Ocean Modelling 176 (2022) 102065

Contents lists available at ScienceDirect

Ocean Modelling

journal homepage: www.elsevier.com/locate/ocemod

A high accuracy/resolution spectral element/Fourier–Galerkin method for the simulation of shoaling non-linear internal waves and turbulence in long domains with variable bathymetry

Theodoros Diamantopoulos [a,*], Sumedh M. Joshi [b,1], Greg N. Thomsen [c], Gustavo Rivera-Rosario [a], Peter J. Diamessis [a], Kristopher L. Rowe [d]

[a] School of Civil and Environmental Engineering, Cornell University, Ithaca, NY, USA
[b] Center of Applied Mathematics, Cornell University, Ithaca, NY, USA
[c] Wandering Wakhs Research, Austin, TX, USA
[d] Leadership Computing Facility, Argonne National Laboratory, Lemont, IL, USA

ARTICLE INFO

Keywords:
Spectral methods
Domain decomposition
Deflation
Non-linear internal waves

ABSTRACT

A high-order hybrid continuous-Galerkin numerical method, designed for the simulation of non-linear, non-hydrostatic internal waves and turbulence in long computational domains with complex bathymetry, is presented. The spatial discretization in the non-periodic wave-propagating directions, utilizes the nodal spectral element method. Such a high-order element-based discretization allows the highly accurate representation of complex domain geometry along with the flexibility of concentrating resolution in areas of interest. Under the assumption of the normal-to-isobath propagation of non-linear internal waves, a third periodic direction is incorporated via a Fourier–Galerkin discretization. The distinct non-hydrostatic nature of non-linear internal waves and, any instabilities and turbulence therein, necessitates the numerically challenging solution of the pressure Poisson problem. A defining feature of this work is the application of a domain decomposition approach, combined with block-Jacobi/deflation-based preconditioning to the pressure Poisson problem. Such a combined approach is particularly suitable for the long high aspect-ratio complex domains of interest and enables the efficient high-accuracy reproduction of the non-hydrostatic dynamics of non-linear internal waves. Implementation details are also described in the context of the stability of the solver and its parallelization strategy. A series of benchmarks of increasing complexity demonstrate the robustness of the flow solver. The benchmarks culminate with the three-dimensional simulation of a convectively breaking mode-one non-linear internal wave over a realistic South-China-Sea bathymetric transect and background current/stratification profiles.

1. Introduction

1.1. Non-linear internal waves and turbulence

Non-linear internal waves (NLIWs) in the ocean are at the receiving end of the cascade of energy originally injected into the water-column by winds and tides (Alford et al., 2015; Egbert and Ray, 2000). Propagating over long distances, NLIWs dissipate and mix the water column along their propagation path and deposit energy away from their generation site through breaking as a result of turbulence and instabilities in wave interior and wave-seafloor interactions (Rivera-Rosario et al., 2017; Boegman and Stastna, 2019). Inherently strongly non-linear and non-hydrostatic phenomena, NLIWs are often encountered in the form of mode-one internal solitary waves (ISWs) of depression, characterized by a precarious balance between their non-linearity and physical dispersion. The breaking of an ISW is a process of critical significance to ocean energetics where the associated turbulence and mixing are induced via either a shear instability (Moum et al., 2003) or a convective instability (Lien et al., 2012, 2014; Lamb et al., 2019) as they shoal into shallower coastal waters.

The simulation of a shoaling ISW over an actual bathymetry poses a challenge to state-of-the-art numerical modeling. Essentially, the wave-scale response to the bathymetry, given the particular background shear and stratification, has to be effectively captured over long propagation distances without being spuriously altered by numerical dispersion and dissipation. Furthermore, the numerical discretization has to be carefully chosen for any finer-scale motions developing within

* Correspondence to: School of Civil and Environmental Engineering, 220 Hollister Hall, Cornell University, Ithaca, NY, 14853, USA
E-mail address: td353@cornell.edu (T. Diamantopoulos).
[1] Deceased.

https://doi.org/10.1016/j.ocemod.2022.102065
Received 16 December 2021; Received in revised form 24 May 2022; Accepted 14 June 2022
Available online 18 June 2022

the simulated wave to be as close as possible to their oceanic counterparts by reproducing, for a given grid resolution, the broad range of scales within a turbulent flow occurring inside the wave.

1.2. High-order element-based techniques

In this regard, high-accuracy pseudo-spectral discretizations have emerged as powerful approaches for simulating stratified flows. Aside from triply periodic discretizations (de Bruyn Kops, 2015; Winters et al., 2004), features such as complex boundary conditions and geometries, necessary for the simulation of NLIWs, have been recently developed for Fourier-based (Winters and de la Fuente, 2012) and Chebyshev-collocation techniques (Subich et al., 2013). An alternative to global spectral discretizations, such as those discussed above, are high-order element-based techniques. Sporadically used for the simulation of stratified flows (Diamessis et al., 2005; Rivera-Rosario et al., 2020; Özgökmen et al., 2004), high-order element-based discretizations are attractive for the simulation of NLIWs and their resultant instabilities/turbulence since they support the exponential convergence of global spectral methods with the inherent flexibility of elemental discretizations that can localize resolution in areas of interest (Deville et al., 2002; Kopriva, 2009). As such, high-order element-based discretizations are a powerful potential alternative to Adaptive-Mesh-Refinement techniques (Barad et al., 2009; Santilli and Scotti, 2015).

In this study, a continuous Galerkin Spectral Element Method (SEM) (Patera, 1984) is used for discretizing the along-wave propagation and vertical directions as well as representing any complex topographic features by employing curvilinear quadrilateral elements. First, localized flow resolution in the vertical allows the reliable representation of background stratification and currents along with any finer-scale motions in wave interior, typically concentrated in the upper layer of the water column for ISWs of depression (Lien et al., 2012) or at the bottom boundary when studying their bottom boundary layer induced by the wave (Sakai et al., 2020a,b). Second, as in the case for a convectively breaking, shoaling ISWs, grid resolution in the along-wave propagating direction is increased in the breaking region (i.e., shallower waters) while remaining coarse in deep waters, enabling a reduction to the computational cost compared to the uniform grid case. Lastly, by adhering to a high polynomial order ($N \geq 7$) within each element, the smallest resolved scales are not numerically damped as any artificial dissipation is minimized at these scales for the particular resolution. In tandem, a high polynomial order also ensures that any error introduced by the commonly used linear interpolation of the bathymetry is avoided (Steinmoeller et al., 2016). It is worth noting that high polynomial orders introduce minimal numerical dispersion which does not artificially impact the physical dispersion of the wave. This is essential for preserving the ISW's waveform over long propagating distances and ensuring that any numerical phase errors, which may lead to the development of artificial trailing waves, are mitigated.

Note that the authors have established a significant knowledge base in the context of high-order element discretizations, through the adaptation and implementation of the Spectral Multidomain Penalty Method (SMPM), i.e., a discontinuous collocation-based numerical method (Hesthaven, 1998; Escobar-Vargas et al., 2014; Diamessis et al., 2005), in the framework of the incompressible Navier–Stokes equations (INSE). Nonetheless, on account of its inherently discontinuous nature, the SMPM is found to develop elevated spurious divergence at the elemental interfaces for highly under-resolved simulations (Joshi et al., 2016a). Consequently, the spatial discretization is replaced by the SEM, i.e, a weak-form-based continuous numerical method.

1.3. Pressure Poisson equation

A crucial aspect of implementation which has to be addressed when using a high-order element-wise discretization for the INSE, is the numerical linear algebra associated with the solution of the pressure Poisson equation (PPE) which is essential for the accurate reproduction of non-hydrostatic dynamics (Joshi et al., 2016b; Scotti and Mitran, 2008; Lamb et al., 2019). Admittedly the most computationally costly part of an INSE solver, the PPE solver design is constrained by the leptic grids (Scotti and Mitran, 2008) which are inherited from the high aspect ratio nature of NLIWs and the long, shallow computational domains connected to their propagation properties. Consequently, when simulating NLIWs, the PPE solver to be used, and its adopted preconditioning strategy, need to carefully take into account these design considerations, to achieve convergence with the minimum number of iterations possible.

As such, the numerical solution of the pressure Poisson problem in this work is broken down into the following steps. First, the discretized PPE is reformulated via the use of multi-level static condensation/iterative substructuring (Couzy and Deville, 1995; Karniadakis and Sherwin, 2013), a domain decomposition technique which takes into account the geometry of the discretization and local support of the elemental basis functions (Canuto et al., 2007). This domain decomposition approach, which leads to a smaller global system of equations, is one of the main advantages of the SEM when solving an elliptic equation, compared to other high-order Galerkin discretizations (Yakovlev et al., 2016). Hence, the discretized system of equations is condensed, via block Gaussian eliminations, into a hierarchy of smaller Schur complement problems where each matrix division takes place locally except for the last level of the condensation. Secondly, the last Schur complement problem is solved iteratively via a two-level deflated preconditioned conjugate gradient solver. Based on the work of Joshi et al. (2016b), albeit for a symmetric discretization in this case, deflation enchanced by a block-Jacobi preconditioner is an effective preconditioning approach irrespective of the grid's aspect-ratio, number of elements used, and polynomial order of the approximation.

1.4. Hybrid element-based fourier numerical methods

The extension of the SEM two-dimensional solver, and the respective PPE computational kernel, into three dimensions is guided by *in situ* observations regarding the propagation of NLIWs (Lien et al., 2012, 2014). Since the propagation of NLIWs may be reasonably approximated as operating along a trajectory normal to the isobaths, an additional third periodic direction is incorporated via a Fourier–Galerkin (FG) discretization (Kopriva, 2009). Although hybrid SEM/FG have been developed in the past (Karniadakis, 1990; Moxey et al., 2020; Blackburn et al., 2019), to the best of the authors knowledge, a high-order hybrid SEM/FG discretization specifically designed for high-aspect ratio complex, long domains combined with a robust PPE preconditioning strategy has yet to appear in the literature.

1.5. Implementation

The development of a new solver is not a light undertaking and requires significant resources to develop a method, scalably implement the technique, verify correctness, and maintain the tool over its lifetime so that it is both flexible and performant. As such, utilizing or adapting existing high performance frameworks (Balay et al., 2021; Paul F. Fischer and Kerkemeier, 2008; Moxey et al., 2020) for one's application is the obvious, and best, approach for most. While a fully three-dimensional element-based discretization in the form of hexahedral or tetrahedral elements offers a more general framework, the discretized three-dimensional SEM operator has a larger condition number than its two-dimensional counterpart (Canuto et al., 2007). In addition to the increased iteration counts of the conjugate gradient solver used for

the PPE and viscous/diffusive solves (cf. Section 1.3), the work done per iteration increases as the elemental matrix is proportional to the problem size, $O(N^3)$ where N is the order of the polynomial expansion per element.

In the flow solver considered in this paper, the choice of a two-dimensional discretization, motivated by physical considerations, enables the use of an identical computational grid for of each transverse plane. As such each $x-z$ plane may be solved concurrently rather than having to resort to the fully coupled solve of its three-dimensional counterpart. To best of the authors' knowledge there are no existing solvers that allow exploitation of high aspect-ratio, long-domain physics of interest, and, thus, creating SEM/FG in terms of discretization and implementation to solve loosely coupled two-dimensional subproblems was a necessary path forward.

From an implementation standpoint the predecessor SMPM flow solver uses a one-dimensional parallelization strategy (Joshi et al., 2016b) where a subset of the computational domain in the x-direction is assigned to each parallel process. This choice restricts the overall resolution and parallel scalability for three-dimensional simulations. Therefore, in the newly developed SEM flow solver, the parallelization strategy takes place in two dimensions enabling the use of much larger number of parallel processes with the associated decrease of the memory footprint per parallel process.

1.6. Paper structure

This paper is organized as follows: In Section 2, the main model equations are presented. Section 3 presents the implicit–explicit stiffly stable splitting scheme of the INSE along with the necessary implicit solves needed to integrate the field variables in time. Section 4 discusses the spatial discretization and the associated domain decomposition and preconditioning strategies adopted for the pressure Poisson problem along with a brief discussion of the viscous term treatment. Section 5 discusses the implementation of the discretization and analyzes its major components to understand its potential performance bottlenecks and scalability. Section 6 covers the stabilization techniques used in the SEM/FG discretization. In Section 7, a series of unstratified and stratified benchmarks are shown, with the latter culminating in the simulation of a convectively breaking, shoaling ISW, thereby demonstrating the efficacy of the flow solver. Section 8 consists of the concluding remarks.

2. Equations of motion

The equations of motion of an incompressible fluid under the Boussinesq approximation are typically written as:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u}\cdot\nabla\mathbf{u} - \frac{g}{\rho_0}\rho'\mathbf{k} - \frac{1}{\rho_0}\nabla p' + \nu\nabla^2\mathbf{u} \tag{1}$$

$$\frac{\partial \rho'}{\partial t} = -\mathbf{u}\cdot\nabla\left(\rho' + \bar{\rho}\right) + \kappa\nabla^2\boldsymbol{\rho}' \tag{2}$$

$$\nabla\cdot\mathbf{u} = 0\ , \tag{3}$$

where $\mathbf{u}$ is the velocity vector field with components (u, v, w) in the (x, y, z) directions, respectively. For stratified flows, the horizontal velocity component u is decomposed into a fluctuating u' and a steady state background current $U = U(z)$ which leads to the following notation

$$\begin{aligned} u &= u' + U \\ w &= w' \\ v &= v'\ , \end{aligned} \tag{4}$$

where the prime symbol denotes the deviation of the velocity field from the background state, ρ_0 is the reference density, g is the gravitational acceleration and $\mathbf{k}$ is the unit vector in the upward (positive) vertical direction. In the momentum equation (1), p' is the pressure perturbation from the background hydrostatic pressure and ν, κ are the molecular diffusivities which are assumed to be constant. Note that the representation of the viscous terms, i.e., vector Laplacian, is a direct result of incompressibility (Diamantopoulos et al., 2021). Additionally, ρ' is the density perturbation which originates from the decomposition (Kundu and Cohen, 2008),

$$\rho = \rho_0 + \bar{\rho}(x, y, z) + \rho'(x, y, z, t) \text{ with } \rho' \ll \bar{\rho} \ll \rho_0\ , \tag{5}$$

where $\bar{\rho}$ is the background density.

3. Time discretization

The semi-discrete equations which arise from the velocity projection scheme (Karniadakis et al., 1991) are comprised of three fractional steps, treated separately and in succession: (1) the advective (non-linear) term, (2) the pressure and (3) the viscous term.

3.1. Non-linear term

The non-linear term is advanced in time using an explicit extrapolation ($J = 3$) (Karniadakis et al., 1991) where an intermediate velocity field $\tilde{\mathbf{u}}$ is obtained

$$\frac{\tilde{\mathbf{u}} - \sum_{i=0}^{J-1}(\alpha_i \mathbf{u}^{n-i})}{\Delta t} = -\sum_{i=0}^{J-1}\beta_i\left(\mathbf{u}^{n-i}\cdot\nabla\mathbf{u}^{n-i} + \frac{g}{\rho_0}\left(\rho'\right)^{n-i}\mathbf{k}\right)\ , \tag{6}$$

where α_i and β_i are the respective time-stepping coefficients which depend on the time discretization order and time-step Δt for a variable time-step (Peyret, 2002). Details regarding the spatial discretization and implementation of the non-linear term are provided in Appendix B.

3.2. Pressure Poisson equation and projection step

The equation for the pressure arises from the time-advancement of the first intermediate velocity field $\tilde{\mathbf{u}}$ into a second intermediate, divergent-free velocity field $\hat{\mathbf{u}}$,

$$\frac{\hat{\mathbf{u}} - \tilde{\mathbf{u}}}{\Delta t} = -\nabla p^{n+1}, \tag{7}$$

where p is the temporal mean of the pressure perturbation calculated over one time-step (not to be confused with the instantaneous pressure (Diamessis et al., 2005)). Eq. (7) can be reformulated as

$$\frac{1}{\Delta t}\hat{\mathbf{u}} + \nabla p^{n+1} = \frac{1}{\Delta t}\tilde{\mathbf{u}}\ . \tag{8}$$

The equation above is effectively the Helmholtz decomposition of the first intermediate velocity field $\tilde{\mathbf{u}}$ into a solenoidal vector field $\hat{\mathbf{u}}$ and an irrotational vector field ∇p^{n+1}. Taking the divergence of Eq. (8), a Poisson equation arises for the pressure

$$-\nabla^2 p^{n+1} = -\frac{1}{\Delta t}\nabla\cdot\tilde{\mathbf{u}}\ . \tag{9}$$

The consistent boundary conditions for the pressure Poisson equation aimed towards ensuring $\mathcal{O}\left(\Delta t^2\right)$ accuracy of the time-discretization scheme are specified in Karniadakis et al. (1991). Once the pressure is computed, the second intermediate velocity $\hat{\mathbf{u}}$ is computed from the projection

$$\hat{\mathbf{u}} = \tilde{\mathbf{u}} - \Delta t\nabla p^{n+1}\ . \tag{10}$$

3.3. Viscous equation

The evaluation of the velocity field at the next time step is computed implicitly by solving a vector Helmholtz equation. Notice that as written in Eq. (1), the viscous term of the incompressible Navier–Stokes equations is expressed in the form of a vector Laplacian, under the assumption of a constant kinematic viscosity

$$-\alpha\nabla^2\mathbf{u}^{n+1} + \mathbf{u}^{n+1} = \mathbf{f}\ . \tag{11}$$

Here, $\alpha = \nu\Delta t/\gamma_0$ is a positive scaling coefficient, with γ_0 a time stepping coefficient, and $\mathbf{f} = \hat{\mathbf{u}}/\gamma_0$ is the right-hand side. Note that for viscous flows with complex boundaries of non-zero curvature, enforcement of free-slip boundary condition is performed by solving each component of Eq. (11), separately (Diamantopoulos et al., 2021).

3.4. Density equation

The advection–diffusion equation for the density perturbation, Eq. (2), follows the same implicit–explicit temporal discretization. First an intermediate density perturbation is computed through the explicit advancement of the advective term. Second, an implicit solve of a scalar Helmholtz equation, i.e., the diffusive term takes place for computing the density perturbation at the next time-step. It is during this last step where boundary conditions are imposed for the density perturbation. Throughout this work, the boundary conditions for the density perturbation are considered of the homogeneous Neumann-type, i.e, zero flux.

4. Spatial discretization of elliptic equation

In this section, the spatial discretization of the pressure Poisson equation (PPE) and the ensuing domain decomposition technique, along with its underlying numerical linear algebra, are presented in detail. To avoid redundancy, the rest of the field variables adhere to the same spatial discretization as for the pressure during their respective implicit steps of the solver (see Eq. (11) and Section 3.4). Nevertheless, the overall computational kernel for the PPE differs drastically in terms of complexity with respect to the velocity/density implicit solves (see Section 4.5). Therefore, emphasis is given to the numerical linear algebra when solving the Poisson problem which is inextricably linked to long computational complex domains encountered in the simulation of NLIWs and their non-hydrostatic dynamics. The overall preconditioning strategy, along with the adopted domain decomposition/parallelization scheme, which are tailored specifically to solve robustly the PPE under these computational meshes, are then further discussed.

4.1. Periodic direction

Denoting as $\Omega \subset \mathbb{R}^2$ a two-dimensional domain and as $\Omega' = \Omega \times [0, L_y]$, the extruded three-dimensional domain in the transverse periodic direction $y \in [0, L_y]$, and a field variable p, which can be any of the field variables, is approximated by the truncated Fourier series as

$$p(x, z, y) = \sum_{n=0}^{N_y/2-1} \hat{p}_n(x, z)\phi_n(y) \,, \tag{12}$$

where $\phi_n = e^{ik_n y}$ are the Fourier basis functions, N_y the number of grid points in the transverse direction, k_n is the wavenumber, and $\hat{p}_n(x, k_n, z) = \hat{p}_n(x, z) \in \mathbb{C}$ is the respective Fourier coefficient. Following a Fourier Galerkin discretization in y (Kopriva, 2009), the Poisson problem (see Eq. (9)) for a single transverse wavenumber becomes

$$-\nabla^2_{xz}\hat{p}_n(x, z) + k_n^2\hat{p}_n(x, z) = \hat{f}_n \,, \tag{13}$$

where ∇^2_{xz} stands for the Laplacian operator in the non-homogeneous directions, and $\hat{f}_n(x, z) = \hat{f}_n(x, z, k_n)$ corresponds to the Fourier coefficients of the right-hand-side of the Poisson problem. Now, by virtue of the orthogonality of the Fourier basis functions and for all of the transverse non-zero wavenumbers k_n, Eq. (13) is a two-dimensional complex Helmholtz equation in Fourier space. Therefore, a series of two-dimensional solves in the $x - z$ plane of the Helmholtz equation Eq. (13), one for each transverse wavenumber, has to be performed to compute p in Ω'.

4.2. Non-homogeneous directions

The discretization in the non-homogeneous directions $x - z$ is presented for a single transverse wavenumber with each of the two parts (real and imaginary) of the complex Fourier coefficient $\hat{p}_n$ following naturally the discretization herein. Let $\mathcal{V}^N \subset H^1(\Omega)$, be a finite subspace where $p, v \in \mathcal{V}^N$ are a part of the solution, i.e., the real or imaginary part of a Fourier coefficient and the test function respectively. Note that the wavenumber dependence of p and right-hand-side f are not included here to simplify notation. Accordingly, the weak form of Eq. (13) under the Galerkin approximation becomes

$$\int_\Omega \nabla v \cdot \nabla p d\Omega + k_n^2 \int_\Omega v p d\Omega = \int_\Omega v f d\Omega + \oint_{\partial\Omega} v \nabla p \cdot \mathbf{n} dS \,, \tag{14}$$

where $\nabla p \cdot \mathbf{n} = \partial p/\partial \mathbf{n}$ is the natural boundary condition (Deville et al., 2002). By defining $\mathcal{V}^N$ as the finite subspace spanned by two-dimensional Lagrangian basis functions up to order N, $\mathcal{V}^N = \text{span}\{h_1(x, z), \dots, h_n(x, z)\}$, p and v are approximated as $p = \sum_k p_k h_k$ and $v = \sum_k v_k h_k$ where $k \in \{1, \dots, n\}$ is the corresponding index set, and n is the total number of degrees of freedom in the $x - z$ plane. Note that each element has the same polynomial order. Thus, the discretized Eq. (14) is written in a matrix form

$$H(k_n)\mathbf{p} = M\mathbf{f} \Rightarrow H(k_n)\mathbf{p} = \mathbf{g} \,, \tag{15}$$

where $H(k_n)$ is the discrete-weak-form-based Helmholtz operator

$$H(k_n) = K + k_n^2 M \,, \tag{16}$$

and $K_{ij} = \int_\Omega \nabla h_i \cdot \nabla h_j d\Omega$ and $M_{ij} = \int_\Omega h_i h_j d\Omega$ are the respective entries of the assembled stiffness and mass matrices (Deville et al., 2002) where $i, j \in \{1, \dots, n\}$. For Eq. (15), and without loss of generality, homogeneous Neumann boundary conditions are assumed for p, since non-homogeneous Neumann boundary conditions contribute only on the left-hand-side of the equation (Deville et al., 2002). Note that the viscous/diffusive equations for the velocity and density field follow the same weak-form-based formulation and discretization as further discussed in Section 4.5.

4.2.1. Domain-decomposition of the PPE and the Schur complement problem

A non-overlapping, domain decomposition (DD) method with iterative substructuring/static condensation is used for each transverse wavenumber when solving for the pressure (Karniadakis and Sherwin, 2013). In tandem with a logically Cartesian topology, Eq. (15) is broken down into a hierarchy of smaller, trivially parallel, problems with homogeneous Dirichlet boundary conditions for the two levels of the condensation (Karniadakis and Sherwin, 2013; Huismann et al., 2017; Deville et al., 2002). Once the second and last stage of the DD is reached, a Schur complement problem on the vertical interfaces, Γ^v, of the subdomains is iteratively solved. In the context of hierarchy of problems, a subsequent backwards sweep ensures the solution on the global computational domain. An example grid demonstrating the decomposition is shown in Fig. 1.

Using a conforming nodal SEM discretization, the solution of Eq. (15) is sought for exactly at the grid points. For the DD to take place, the unknowns are first ordered in groups based on their grid point location on the $x - z$ plane. Three groups of unknowns are therefore identified: (a) the interior elemental unknowns $p_i \in \Omega^i$, (b) the unknowns which reside on the horizontal edges of the computational domain $p_h \in \Gamma^h$ and (c) the vertical edge's degrees of freedom $p_v \in \Gamma^v$ where $\Gamma^v = \Omega \cap (\Omega^i \cup \Gamma^h)$. The resulting diagonal block-matrices which then emerge in Eq. (17), and are shown in Fig. 2, correspond to the self-interaction between unknowns of the same group of unknowns with the off-diagonal block-matrices expressing the coupling across different groups

$$\begin{bmatrix} H_{ii} & H_{ih} & H_{iv} \\ H_{hi} & H_{hh} & H_{hv} \\ H_{vi} & H_{vh} & H_{vv} \end{bmatrix} \begin{bmatrix} p_i \\ p_h \\ p_v \end{bmatrix} = \begin{bmatrix} g_i \\ g_h \\ g_v \end{bmatrix} \,. \tag{17}$$

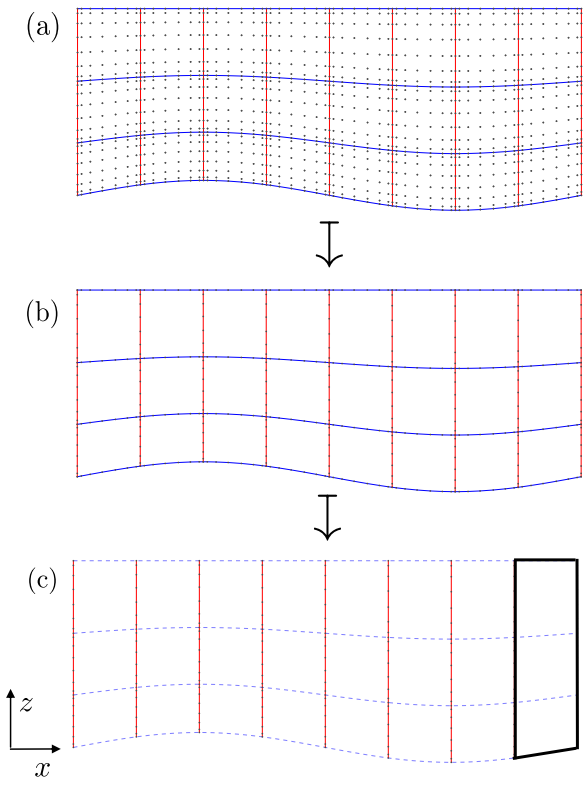


Fig. 1. Grid example for each level of the domain decomposition. (a) The full computational grid where grid points are color-coded based on their location. Black dots denote the interior elemental grid points (Ω_i), whereas blue and red solid lines represent the horizontal (Γ_h) and vertical edges (Γ_v), respectively. (b) The skeleton mesh of $S^{(1)}$ matrix (Eq. (18) and Fig. 3). (c) The vertical edges (red lines) of the second Schur problem $S^{(2)}$ matrix (Eq. (19) and Fig. 4). The blue dashed lines denote the elimination of the horizontal edges from the last level of the condensation. The black outline which encloses a vertical strip of elements corresponds to a subdomain.

Fig. 2 demonstrates the sparsity structure of the block-based representation of the H matrix in Eq. (17) for the computational grid shown in the top panel of Fig. 1. Although this corresponds to the assembled, i.e, global matrix, in practice, global matrices are not explicitly built due to memory limitations for large problem sizes. Nevertheless, for the better illustration of the DD technique all of the resulting matrices and their sparsity structures correspond to the assembled ones shown in Fig. 2.

As a first stage of the DD, the elemental interior unknowns u_i are eliminated, via a block Gaussian elimination, condensing Eq. (17) to a Schur complement matrix $S^{(1)}$ on the horizontal and vertical edges, skeleton mesh, with the following algebraic structure:

$$\begin{bmatrix} S^{(1)}_{hh} & S^{(1)}_{hv} \\ S^{(1)}_{vh} & S^{(1)}_{vv} \end{bmatrix} \begin{bmatrix} p_h \\ p_v \end{bmatrix} = \begin{bmatrix} g^{(1)}_h \\ g^{(1)}_v \end{bmatrix} , \tag{18}$$

where $S^{(1)}_{mn} = H_{mn} - H_{mi}H^{-1}_{ii}H_{in}$ are the block matrices of the first Schur problem, $g^{(1)}_m = g_m - H_{mi}H^{-1}_{ii}g_i$ is its right-hand-side and $m, n = \{h, v\}$. Note that each level of condensation entails the division by the self-interaction block matrix of the group of unknowns eliminated by the condensed equations. In this first level of condensation, the elimination group refers to the interior elemental unknowns p_i and thus the inversion of the H_{ii} matrix is required. Now, since the H_{ii} matrix is block-diagonal (Canuto et al., 2007; Karniadakis and Sherwin, 2013) (cf. black blocks of Fig. 2), with each block corresponding strictly to the interior unknowns of each quadrilateral element, its inversion is trivially parallel. Consequently, the block entries of the $S^{(1)}$ matrix in Eq. (18) are computed first in an element-wise fashion with a subsequent assembly of the shared interfaces, i.e., direct stiffness summation, to enforce continuity (Deville et al., 2002; Patera, 1986).

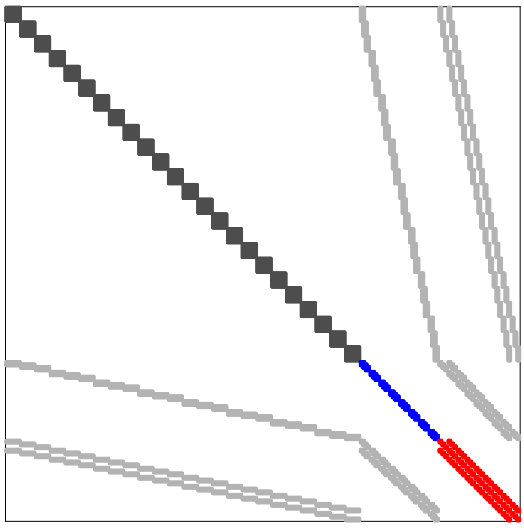


Fig. 2. Sparsity structure of the reordered Helmholtz matrix of Eq. (17) for the grid shown in Fig. 1 with the block diagonal matrix H_{ii} in black. The horizontal H_{hh} and vertical H_{vv} block matrices are shown in blue and red, respectively. The off-diagonal block matrices are shown in gray.

The second, and last, level of condensation refers to the computation of a second Schur matrix (Eq. (19)) for the vertical edges of the computational domain (cf. Fig. 1(c)) which results from the block-Gaussian elimination when solving for the p_v group of unknowns in Eq. (19)

$$\begin{aligned} S^{(2)} &= S^{(1)}_{vv} - S^{(1)}_{vh}(S^{(1)}_{hh})^{-1}S^{(1)}_{hv} \\ S^{(2)}p_v &= g^{(2)}_v \ . \end{aligned} \tag{19}$$

where $g^{(2)}_v = g^{(1)}_v - S^{(1)}_{vh}S^{(1)^{-1}}_{hh}g^{(1)}_h$ is the respective right-hand-side. Notice the division with the $S^{(1)}_{hh}$ matrix for the computation of the second Schur matrix and its right-hand-side. Similar to the case of H_{ii}, $S^{(1)}_{hh}$ is block diagonal but now each block geometrically belongs to the horizontal edges of a vertical collection of elements, i.e., subdomain (cf. Fig. 3). Once the second Schur system of equations is solved iteratively, two successive Dirichlet problems (Eq. (20)) are solved directly, following the backward substitution step of the block-Gaussian elimination (Eq. (17)–(18)). First, the horizontal-edge unknowns are computed with a subsequent computation of the elemental interior unknowns (Eq. (20)). These inversions are local at a subdomain/element level with corresponding Dirichlet boundary conditions enforced by "lifting" the known solution (Karniadakis and Sherwin, 2013). Lifting is a commonly used strategy for prescribing the contribution of the Dirichlet boundary conditions to the interior unknowns by updating accordingly the right-hand-side of the system of equations (cf. Eq. (20)) (Blackburn et al., 2019)

$$\begin{aligned} p_h &= (S^{(1)}_{hh})^{-1}(g^{(1)}_h - S^{(1)}_{hv}p_v) \\ p_i &= (H_{ii})^{-1}(g_i - H_{ih}p_h - H_{iv}p_v) \ . \end{aligned} \tag{20}$$

4.3. Solving the second schur problem

The second Schur problem of Eq. (19) represents geometrically the coupling among the vertical edges of the computational domain. $S^{(2)}$ is block-tridiagonal (Fig. 4), since each vertical edge has at most two neighboring edges, with a bandwidth directly proportional to the vertical grid point count, N_z. For high-aspect ratio grids, where the number of elements in x, m_x, is much larger than that of the elements in z, m_z, $m_x \gg m_z$, as in the simulation of highly NLIWs, $S^{(2)}$ is a large matrix with a relatively small bandwidth. Furthermore, $S^{(2)}$ is symmetric positive definite a property inherited from the original

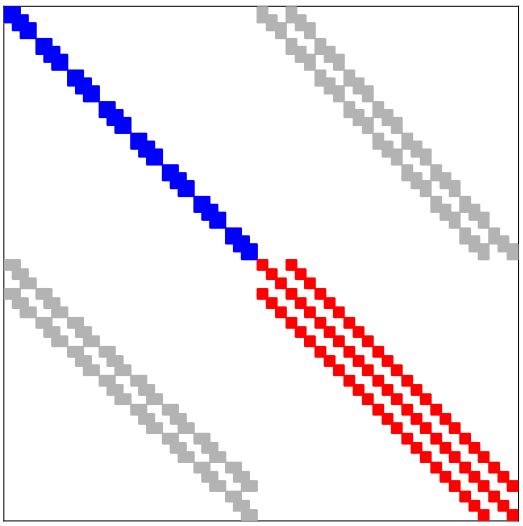


Fig. 3. Sparsity structure of the first Schur complement matrix $S^{(1)}$ of Eq. (18) with $S^{(1)}_{hh}$ in blue, $S^{(1)}_{vv}$ in red, and the off-diagonal block matrices are delineated in gray, following the same stylistic structure as Fig. 2. Note that the $S^{(1)}_{hh}$ matrix is block-diagonal with each block corresponding to the coupling among the horizontal edges within a subdomain.

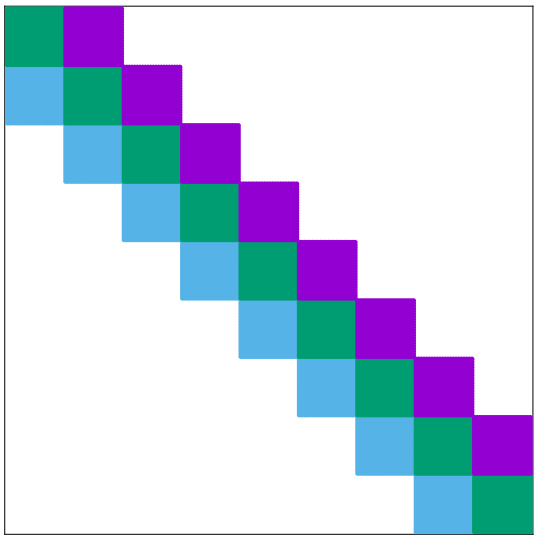


Fig. 4. The second block-tridiagonal Schur complement matrix $S^{(2)}$ for the vertical edges of the computational domain shown in Fig. 1. The tridiagonal structure of $S^{(2)}$ denotes that each vertical edge is coupled with at most two of its adjacent vertical edges.

matrix H of Eq. (17). Note that for $k_n = 0$, the $S^{(2)}$ matrix is positive semi-definite and thus it is regularized via a nullspace projection of the right-hand-side vector (Pozrikidis, 2001). Essentially, the inverse of $S^{(2)}$ is a full matrix which effectively enforces the coupling of all of the vertical-edges degrees of freedom, therefore imposing a global communication pattern among computational processes (Tufo and Fischer, 2001). Consequently, once a solution is obtained for the vertical edges, the remaining solves (Eq. (20)) are local and embarrassingly parallel.

The block-tridiagonal structure of the condensed Schur matrix $S^{(2)}$ is amenable to the use of a direct solver in a purely serial framework. Nonetheless, since a naive parallel block Thomas algorithm performs poorly (Hirshman et al., 2010), various techniques have been proposed for the solution of statically condensed/coarse grid problems with efforts being concentrated on mitigating memory limitations and overall parallelization bottlenecks (Borrell et al., 2011; Tufo and Fischer, 2001; Lee and Wright, 2014; Seal et al., 2013). Nevertheless, for computationally intensive simulations the use of an iterative solver is preferred due to efficiency considerations (Offermans et al., 2019).

4.4. Deflation block-Jacobi preconditioned conjugate gradient solver

A deflation-based, block-Jacobi preconditioned, conjugate gradient solver is utilized for the solution of the second Schur system of equations (see Eq. (19)) (Aubry et al., 2008; Saad et al., 2000; Mansfield, 1990). As demonstrated in Joshi et al. (2016b) for high-aspect ratio leptic grids, the combination of a block-Jacobi preconditioner with deflation, ensures convergence independent of the domain aspect ratio, the polynomial order of the elemental expansion, and the number of elements used in the x-direction. Although the convergence properties of this combined approach have been demonstrated for a high-order collocation discontinuous element based method, the effectiveness of this approach is transferable to symmetric discretizations like the spectral element method (Nicolaides, 1987).

As such, for the domain-decomposed second Schur matrix, its main block-diagonal (see Fig. 4) is used as a block-Jacobi preconditioner. Note that the choice of a non-overlapping block-Jacobi preconditioner is a commonly adopted strategy (Fischer and Rønquist, 1994; Couzy and Deville, 1995; Joshi et al., 2016b; Huismann et al., 2017) which ensures that no communication overhead among processes is introduced during its application in parallel. More specifically, the block preconditioner is factorized during the set-up stage of the solver with the overall factorization cost being amortized over the many pressure Poisson solves per simulation.

For the deflation component of the iterative solver, a deflation matrix $Z \in \mathbb{R}^{n_v \times n_c}$ is defined, where $n_v = \dim(S^{(2)})$ is the total number of unknowns which reside on all of the vertical edges and $n_c = m_x + 1$ is the number of vertical edges, whose columns are the deflation vectors z_j with their entries given by Joshi et al. (2016b),

$$(z_j)_i = \left\{ \begin{array}{ll} 1, & \text{if } x_i \in \Gamma^v_j \\ 0, & \text{if } x_i \notin \Gamma^v_j \end{array} \right\} , \tag{21}$$

where $i \in \{1, \dots, n_v\}$, $j \in \{1, \dots, n_c\}$ and x_i is a grid point that if resides on the Γ^v_j vertical edge then the respective entry of the deflation vector is unity and zero if otherwise. Note that the choice of deflation vectors may affect the convergence rate of the iterative solver (Vermolen et al., 2004) and the investigation of optimal deflation vectors is left for future work. Essentially, Z and Z^T can been interpreted as the prolongation and restriction operators commonly used in multigrid methods (Tang et al., 2009). A coarse grid matrix $C(k_n) = Z^T S^{(2)}(k_n) Z \in \mathbb{R}^{n_c \times n_c}$ of the second Schur matrix is then constructed. This coarse Schur matrix C is a good approximation of $S^{(2)}$ and is subsequently used to update the search direction of the deflated preconditioned conjugate gradient method, achieving faster convergence (Aubry et al., 2008; Saad et al., 2000). Note that the coarse version of the second Schur complement $C(k_n)$ is tridiagonal and can be seen as the one-dimensional x-direction equivalent of Eq. (13) using linear basis functions, i.e., the Helmholtz matrix for the FEM (Joshi et al., 2016b). Particularly for a high polynomial order where fewer elements are used in the x-direction than in the equivalent in resolution lower polynomial order case, the C matrix is much smaller in size than the original Schur complement matrix, i.e., $n_c \ll n_v$. Therefore, the coarse problem is solved directly and redundantly within each process to avoid any introduced communication overhead.

Notice that the rank deficiency of the second Schur complement matrix $S^{(2)}$ for the zeroth mode is passed onto $C(k_n = 0)$. Nonetheless, since the coarse matrix C is solved directly, a regularization technique via Householder matrices (Escobar-Vargas, 2012), equivalent to the right-hand-side null-space projection of the $S^{(2)}$ matrix (see also Section 4.3), is used (Pozrikidis, 2001). However, since the regularization technique on C is not sparsity-preserving, a one-dimensional DD technique is adopted between the interior and the two boundary unknowns of the $C(k_n = 0)$ matrix. First, a 2×2 Schur complement

matrix is computed corresponding to the first and last unknowns of $C(k_n = 0)$. Once this Schur complement matrix is regularized and the two boundary values are computed, a final tridiagonal solve is performed for the interior unknowns of $C(k_n = 0)$ matrix.

4.5. Viscous/diffusive solves

A simple strategy is adopted for the numerical solution of the viscous (Eq. (11)) and diffusive (see Section 3.4) parts of the solver. It is during this step of the solver where boundary conditions for the velocity field and the density perturbation are enforced (Diamantopoulos et al., 2021). Consequently, adhering to a DD approach for these implicit solves, such as that described in Section 4.2.1, would introduce an additional complexity regarding the enforcement of different types of boundary conditions. For instance, the dimensionality of the discretized system of equations will be accordingly reduced; as for the case of Dirichlet boundary conditions where the solution is already known at the boundaries (Karniadakis and Sherwin, 2013). Following the discretization presented in Section 4.1–4.2, the respective Helmholtz matrix $\mathbf{H}^u(k_n)$ of Eq. (11) for a given transverse wavenumber k_n is given by

$$\mathbf{H}^u(k_n) = \alpha\mathbf{K} + (1 + k_n^2)\mathbf{M} \,, \tag{22}$$

where the time-step coefficient α (Eq. (11)) scales linearly with $\nu\Delta t$ or $\kappa\Delta t$. Thus, for high Reynolds number flows, $\mathbf{H}^u$ is diagonally dominant and $\mathbf{H}^u \approx (1 + k_n^2)\mathbf{M}$. Therefore, the implicit viscous/diffusive solves are performed iteratively for the uncondensed system of equations using the diagonal mass matrix $\mathbf{M}$ as a preconditioner (Blackburn et al., 2019) which is an efficient preconditioning strategy for achieving convergence in a few iterations.

5. Implementation

This section assesses the implementation's scalability and ability to tackle environmental scale domains for the temporal and spatial discretizations outlined in Section 4. Specifically the implementation of the SEM/FG flow solver is evaluated against its design goals to understand its bottlenecks and identify areas of interest for a future scaling study.

The SEM/FG flow solver's implementation was guided by three objectives to support scaling to large node and core counts: (1) loose coupling between two-dimensional subproblems, (2) minimize communication, and (3) maximize arithmetic intensity of computations. Nonetheless, these goals alone are not sufficient for an efficient implementation (e.g. use of efficient vendor libraries for computation is required), but provide the foundations that frame its scalability. Both loose coupling and minimizing communication are necessary to maximize parallel execution and set an upper bound on performance through Amdahl's law (Magoulès et al., 2016), though the former is necessary to maximize gains from the method's parallel domain decomposition. Structuring computations to maximize arithmetic intensity provides opportunities for efficient execution, often through external acceleration libraries for FFTs and dense linear algebra. Each of these objectives drives towards an implementation that can be applied to large environmental scale problems.

The SEM/FG solver was designed as a pure MPI solver targeting a single rank per core on multi-core HPC systems. The structured nature of solver's computations admits flexible distribution of work to either MPI ranks or threads, though a hybrid MPI/OpenMP implementation was not pursued since the anticipated performance was comparable. Moreover, a hybrid approach increases complexity in both maintenance and execution.

The analysis presented is limited to the parallel domain decomposition and the viscous and pressure solvers, as their implementations directly reflect the discretization as outlined in Section 4.2.1. Components such as solver setup, input and output, and post-processing are omitted as they are independent of the method. For each component of interest, an outline on how the method maps onto its implementation is provided along with a qualitative assessment of bottlenecks with respect to the aforementioned objectives.

5.1. Parallel domain decomposition

Following the hybrid approach of Bolis et al. (2016) a two-dimensional parallel decomposition in the x and y directions is used. The y dimension is a natural choice due to the Fourier discretization, allowing the solution of a three-dimensional system to be pursued through solving parallel two-dimensional subproblems. The x dimension is chosen over the z since SEM/FG target problems have a high horizontal aspect ratio ($\sim 10:1$) and the parallel decomposition along x provides greater opportunities for scaling due to significantly more horizontal elements. Additionally, this results in minimizing the data exchanged between ranks as the vertical surface area between subdomains is significantly smaller than the horizontal surface area.

Two communication patterns emerge when decomposing in x and y: (1) along $x-z$ planes with point-to-point exchanges of vertical subdomain edges and collective communications during conjugate gradient iterations, and (2) along x–y planes with all-to-all data exchanges coupling all the transverse Fourier modes during non-linear advection term computations. Each two-dimensional subproblem spends the majority of its time communicating with the ranks in its $x-z$ plane, with the coupling communications in the y direction occurring a fixed number of times per time step, ultimately framing SEM/FG's execution as a collection of loosely coupled, two-dimensional subproblems.

From an implementation standpoint, SEM/FG maps its MPI ranks, P_i, onto a two-dimensional Cartesian topology, $P_i \rightarrow P_{xy}$. Subsets of the computational domain are assigned to each process, both in the subdomain (x) and wavenumber (y) sense, such that each process P_{xy} holds one or more subdomains in x and one or more transverse wavenumbers in y directions, respectively. Fig. 5 illustrates the aforementioned decomposition, topology, and communication patterns.

As part of the pressure and viscous solves, vertical edge exchanges are performed between adjacent ranks within an $x-z$ plane to enforce C^0 continuity and are performed simultaneously across all ranks within a local two-dimensional subproblem. Due to the parallel decomposition in x, the data size transferred between adjacent ranks is fixed at $N_v = m_z N + 1$ values, i.e., the grid resolution in the vertical direction. Collective communication within each two-dimensional subproblem occurs as a small, fixed number of reductions to compute dot products during conjugate gradient iterations. Due to deflation-based preconditioning, the pressure solve also has a collective exchange to distribute deflation vectors.

Computation of non-linear advective terms is done in physical space which necessitates transforming grid points from Fourier space and back. Since wavenumbers are distributed along y, collective exchange amongst ranks in the transverse dimension is required to compute each Fourier transform. The exact number of FFTs depends whether the flow is stratified and, thus, Eq. (2) is solved, though needs to compute $O(10m_z N^2)$ N_y-point real-valued FFTs per timestep. The volume of data exchanged between ranks is $O(8m_z N_y N^2)$ bytes per transform. Advection terms are computed once per timestep and result in the only synchronization point across all ranks in the y dimension.

To better understand loose coupling and explore SEM/FG's scalability limitations, the aforementioned communication patterns are qualitatively analyzed. This is performed by examining the choices for allocating subdomains to individual ranks and assessing the bottlenecks that emerge. For the scaling analysis discussed below, one dimension, either x or y, is fixed to an arbitrary size and the other dimension is varied between extremes.

Choices for distributing vertical subdomains amongst ranks in x range from all subdomains on a single rank to one subdomain per rank. While it eliminates communications, the extreme where a single rank

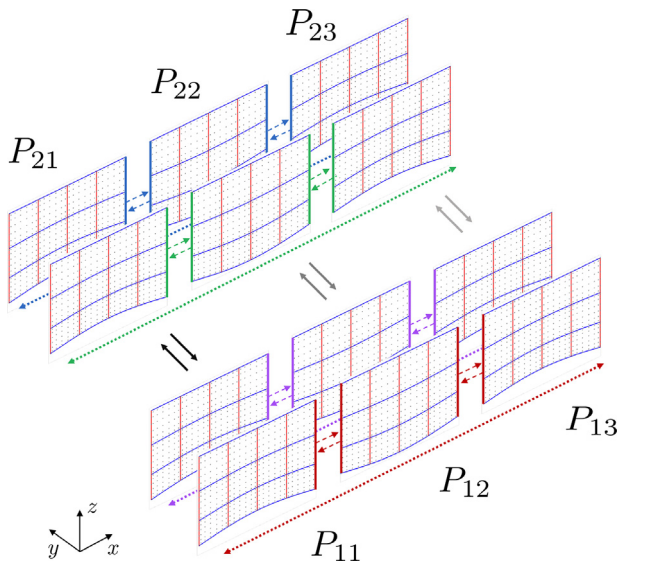


Fig. 5. Example of the parallel domain decomposition and its Cartesian topology. Each process owns two points in the y-direction and four subdomains per $x-z$ plane. Data transposition (solid arrows) takes place during the forward and inverse Fourier transforms among processes which reside on the same column. The vertical edge overlap between two adjacent processes of each $x-z$ plane is exchanged (dashed arrows) via point-to-point non-blocking communications. Each $x-z$ point-to-point communication takes place during the direct stiffness summation, typical of all fractional steps of the time-splitting scheme, and a global communication among processes residing on the same $x-z$ plane during the deflation preconditioning of the pressure Poisson problem (dotted arrows).

has an entire $x-z$ plane is infeasible due to memory requirements, and is not considered further. On the other extreme, mapping a single subdomain to one rank results in minimal computation per rank with a slight increase in communications overhead as more ranks participate in the collective reductions. The time saved in per-rank computation is expected to exceed the increased communication overhead for all environmental scale problems, resulting in an overall smaller time per timestep. Finally, execution with more than one subdomain per rank results in a larger time per timestep than in the single subdomain per rank, though by how much is unknown without a strong scaling study.

Distributing transverse subdomains amongst ranks in y follows a similar argument, though differs in that the time per timestep is limited by the slowest two-dimensional subproblem. Since each subproblem's wavenumber correlates to its condition number (cf. Eq. (16), (22)), which is proportional to the number of conjugate gradient iterations required, the time per timestep is $\max_i T_i$, where the ith rank's total time, $T_i = \sum_k T_{i_k}$, is the sum total time of each of its k-many two-dimensional subproblems. As a result, the extreme case where each rank solves a single two-dimensional subproblem minimizes the total time per timestep. The case where ranks solve multiple subproblems results in a load imbalance requiring partitioning wavenumbers across ranks to avoid such an inefficiency. The current implementation of SEM/FG naively partitions wavenumbers in sequential order and leaves load balancing in y as a future effort.

Given the aforementioned communication patterns the method described in Section 4 maps well to the parallel domain decomposition. Increasing ranks in y reduces total time per timestep, while increasing ranks in x trades per-rank computation for slightly increased collective communication overhead. Increasing ranks in both x and y will reduce time per timestep for environmental scale problems, though cannot be quantified without a scaling study.

5.2. Pressure solve

The method's domain decomposition results in a smaller, isolated problem where only the shared vertical edges are collectively solved, allowing the shared horizontal edges and interior elements to be computed locally and without additional communication as a direct result of the two-level static condensation. Since $\kappa(S^{(2)}) \ll \kappa(H)$ (Carvalho et al., 2001), this results in fewer iterations of deflation-based preconditioned conjugate gradient (DPCG), each operating only on the vertical edges rather than the entire subdomain. Both solving $D^{-1}x_v = f$, where D stands for the block-Jacobi preconditioner, and computing $S^{(2)}x_v$ operate on matrices sized $N_v \times N_v$, which are a factor of $(N+1)^2$ smaller than the entire subdomain. Additionally, these operations map directly onto LAPACK and BLAS routines, `dpotrs()` and `dgemv()` respectively, that are accelerated by vendor implementations.

In addition to reducing the size of the collective problem, the method's two-level condensation also trades communications for local work. First, the coarse direct solve of $C^{-1}Z^T x_v$ in the DPCG iterations is redundantly computed in each rank rather than collectively. Given the tridiagonal, symmetric positive definite nature of $C^{-1}Z^T$ this can be efficiently solved $O(8m_x)$ FLOPS (Golub and Van Loan, 2013) and is an obvious trade for fewer collective communications within each iteration. Once an iterative solution is computed for the shared vertical edges, each rank can compute solutions to horizontal edges and interior elements without any further communication within the $x-z$ plane.

The matrices involved in lifting and solving the RHS for the interior the elements (cf. Eq. (20)), sized $4N \times (N-1)^2$ and $(N-1)^2 \times (N-1)^2$, respectively, lends itself to exploring acceleration libraries tailored to small sizes and batched execution. Both BLASFEO (Frison et al., 2018) and libxsmm (Heinecke et al., 2016) provide tuned implementations for small dense linear algebra problems, including matrix–vector operations, while BLASFEO also solves systems of equations with Cholesky factorization.

While increasing the efficiency of the DPCG iterations reduces the total time per timestep, potentially larger gains are available by reducing the number of iterations required. As seen in Eq. (15) and described in Section 4, SEM/FG's time per timestep is directly limited by the number of iterations required by the $k_y = 0$ wavenumber subproblem. As such, improvements to the preconditioning strategy have a significant impact in reducing the time spent in the pressure solve. Identifying a better selection of deflation vectors is an area of future work.

5.3. Viscous solve

Compared to the pressure solve, SEM/FG's viscous solve is both a smaller and better conditioned problem due to its dominant diagonal structure (c.f. second term of right-hand-side of Eq. (22)). Application of boundary conditions and lifting the RHS result in per-element matrix–vector multiplications, of M and H, all of size $(N+1)^2 \times (N+1)^2$. Preconditioned conjugate gradient (PCG) iterations consist of per-element matrix–vector multiplications with H though the conjugate gradient solution operates on vertical edges due to the domain decomposition, resulting in scalar operations of size $N_v \times (m_{x_s} N + 1)$, where m_{x_s} stands for the number of subdomains per MPI-process. Of note, the block-Jacobi preconditioner is applied through an element-by-element Hadamard product (Golub and Van Loan, 2013) since the matrix M^{-1} can be precomputed during solver setup due to its diagonal structure. As a result, all of the viscous solve's operations are either matrix–vector multiplications that map directly onto BLAS routines or vectorizable array operations. Similar to the pressure problem, the viscous solve may benefit from a BLAS implementation tuned for small matrices so as to accelerate the small matrix–vector multiplications contained in PCG iterations.

6. Stabilization techniques

In under-resolved simulations, which is predominately the case in environmental flow modeling, SEM/FG uses an exponential filter (Diamessis et al., 2005; Vandeven, 1991) as a surrogate of a hyperviscous operator (Moura et al., 2016) both in the periodic y, and

non-homogeneous directions $x-z$. Each velocity component and density perturbation are filtered once every time-step after the computation of the non-linear (Eq. (6)) and advective terms respectively. The filter orders in y, q_y and $x-z$ directions, q_{xz} are reported for each benchmark in Section 7. Additional stabilization is achieved in the $x-z$ non-homogeneous directions via polynomial over-integration, or equivalently referred to as polynomial dealiasing (Kirby and Karniadakis, 2003), of the non-linear and advective terms (Malm et al., 2013). Details regarding the stabilization techniques used in this work are provided on Appendices A and B.

7. Benchmarks

In this section, the overall accuracy of the solver is demonstrated for a series of benchmarks of increasing complexity, ultimately building up to a three-dimensional benchmark of a convectively breaking ISW propagating over a realistic bathymetry (Rivera-Rosario et al., 2022). Results are validated, when applicable, with analytical solutions or with other studies for both unstratified and stratified flows.

7.1. Kovasznay flow

The steady-state Kovasznay flow (Kovasznay, 1948) is a commonly used test case for the validation of high-order two-dimensional incompressible flow solvers. It is based on an exact solution of the incompressible Navier–Stokes equations for a laminar flow past a two-dimensional grid (Eq. (23)) (Hesthaven and Warburton, 2007; Manzanero et al., 2020; Kirby and Sherwin, 2006) on $\Omega = [-0.5, 1] \times [-0.5, 1.5]$ (m^2) . A time step of $dt = 10^{-3}$ s is chosen for a constant number of equally sized elements 8×8 and a varying polynomial order, i.e., *p-refinement* (Hesthaven and Warburton, 2007). Simulations are initialized with the exact solution of the velocity field, given in Eq. (23) for non-homogeneous Dirichlet boundary conditions and performed over a time span of $t = [0, 1]$ (s). Lastly, viscosity is uniform and set equal to $\nu = 1/40\,\mathrm{m}^2/\mathrm{s}$. No exponential filtering is used since the flow is laminar and well-resolved. The L_2 error norm of the field variables is shown in Fig. 6 at time $t = 1$ s as a function of the polynomial order per element per dimension. Notice the exponential convergence (Kopriva, 2009) of SEM, similar to a global spectral method, achieving an error close to machine precision as the polynomial order within each element increases.

$$\begin{aligned} \lambda &= \frac{1}{2\nu} - \sqrt{\frac{1}{4\nu^2} + 4\pi^2} \\ u &= 1 - \exp(\lambda x)\cos(2\pi z) \\ w &= \frac{\lambda}{2\pi}\exp(\lambda x)\sin(2\pi z) \\ p &= (1 - \exp(2\lambda x))/2 \ . \end{aligned} \tag{23}$$

7.2. Normal collision of a vortex dipole with a no-slip wall

The normal collision of a vortex dipole enables the validation of the fully non-linear time-dependent INSE by examining the propagation of a distinct flow feature, namely the vortex dipole, and its interaction with the finer-scale boundary-induced vorticity as it reaches the bottom wall. The work of Subich et al. (2013) is used as a reference since it offers a detailed comparison with previous numerical experiments in the context of high-order discretizations (Clercx and Bruneau, 2006; Kramer et al., 2007). The two-dimensional collision is initially investigated with a subsequent extension to three-dimensions by incorporating a third periodic transverse direction as in Subich et al. (2013). In this study, simulations were performed for a moderate Reynolds number of $Re = UH/\nu = 1250$ (Clercx and Bruneau, 2006), where $H = 1$ m is the half length of the normalized computational domain $\Omega = [-1, 1] \times [-1, 1]$ and U is the computed full-domain-averaged RMS velocity

$$U = \sqrt{\frac{1}{4}\int_\Omega \mathbf{u}^2(\mathbf{x}, t=0)\, d\Omega} = 1 \ (\mathrm{m/s}) \ . \tag{24}$$

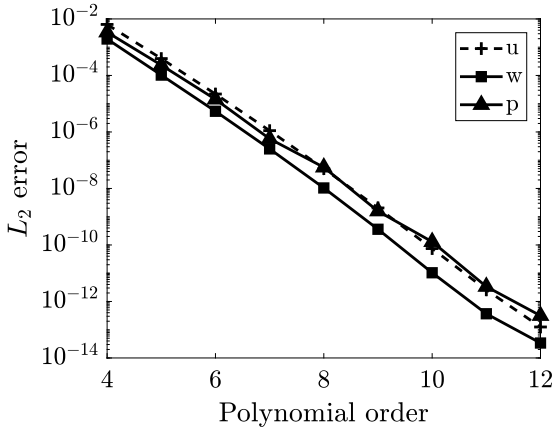


Fig. 6. Lin-log plot of the L_2 error norm of the (u, w) velocity components and instantaneous pressure p as a function of the polynomial order of the element approximation at $t = 1$ s.

The initial two-dimensional velocity field $\mathbf{u} = (u, w)$ induced by the dipole is given by the following expressions

$$\begin{aligned} \frac{u}{U} &= \frac{\omega_0}{2}(z - z_1)\exp\left(-r_1^2/r_0^2\right) - \frac{\omega_0}{2}(z - z_2)\exp\left(-r_2^2/r_0^2\right) \\ \frac{w}{U} &= \frac{\omega_0}{2}(x - x_2)\exp\left(-r_2^2/r_0^2\right) - \frac{\omega_0}{2}(x - x_1)\exp\left(-r_1^2/r_0^2\right) \ , \end{aligned} \tag{25}$$

where $(x_1, z_1) = (-0.1, 0)$ is the initial dimensionless location of the positive monopole, $(x_2, z_2) = (0.1, 0)$ is the initial dimensionless location of the negative monopole, r_1, r_2 is the distance from the center of the positive and negative monopole respectively, r_0 is the radius of each monopole and $\omega_0 \approx 299.5284$ is the amplitude of the monopole's vorticity ω computed from Eq. (24). Note that the $x-z$ plane is used for this benchmark. Therefore, the y-component of vorticity is reported throughout the presentation of two-dimensional and three-dimensional results. In particular, the sign of vorticity, not its direction, is inverted due to a different choice of Cartesian axes when compared to previous studies (Clercx and Bruneau, 2006; Kramer et al., 2007; Subich et al., 2013).

A constant polynomial order of $N = 8$ per element per direction is used for a varying number of elements (m_x, m_z) with $m_x = m_z$. Additionally, a grid stretching strategy is used in the bottom half of the domain, to better resolve the fine bottom boundary layer features resulting from the dipole collision with the wall. The ratio of the heights of two adjacent elements in the vertical is equal to a constant $\varphi_z = \Delta z^{i+1}/\Delta z^i \leq 1$. Varying width elements with a constant ratio of $\varphi_x = 0.92$, symmetrically placed around the center of the domain, are used for two of the cases (Fig. 7). In total, three simulations were performed by increasing the grid resolution in x and z directions through an increase of the number of elements with the corresponding minimum horizontal (Δx_{min}) and vertical grid spacings (Δz_{min}) shown in Table 1. A weak filter of $q_{xz} = 14$ order is used for both the two-dimensional and three-dimensional simulations with a transverse filter order of $q_y = 48$, affecting only the top 23% of the transverse wavenumber range, is used for the three-dimensional test-case.

Three metrics are used to assess the efficacy of the simulations, the total kinetic energy

$$KE(t) = \frac{1}{2}\int_\Omega \left(\frac{\mathbf{u}(\mathbf{x}, t)}{U}\right)^2 d\Omega \ , \tag{26}$$

the magnitude and time of occurence of the maximum enstrophy

$$E(t) = \frac{1}{2}\int_\Omega \left(\frac{\omega(\mathbf{x}, t)}{U/H}\right)^2 d\Omega \ , \tag{27}$$

and the location and strength of the maximum negative vorticity of the primary monopole as described in Subich et al. (2013).

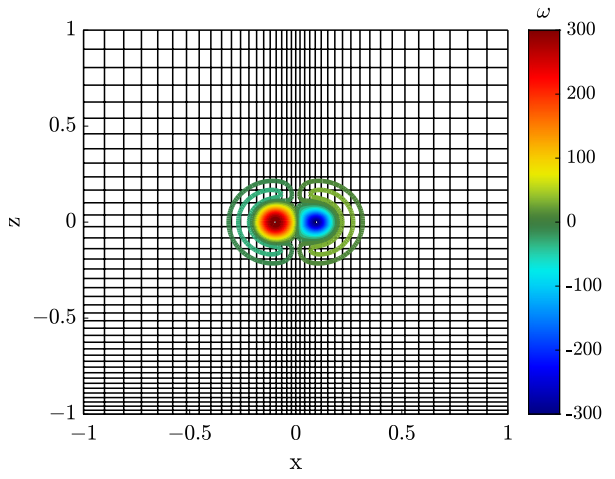


Fig. 7. Initial vorticity of the dipole overlapped by the computational mesh for the 321 × 321 case. Each element has a 9 × 9 Gauss–Lobatto-Legendre grid.

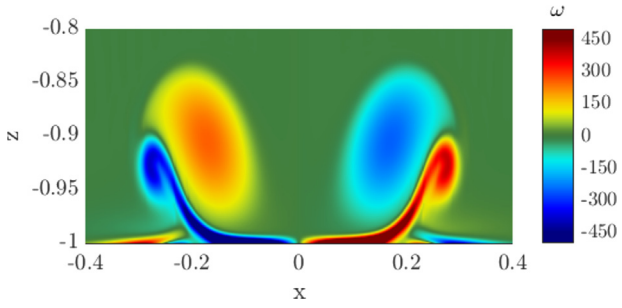


Fig. 8. Vortex dipole interaction with the bottom boundary induced vorticity shortly after the maximum enstrophy time of occurence ($t/(H/U) = 0.363$).

Table 1
Grid parameters and resolution for the two-dimensional vortex dipole wall-normal collision. Polynomial order N, number of elements in the horizontal m_x and vertical m_z directions and the respective number of grid points N_x, N_z in the x and z directions respectively are shown. The grid stretching constants (φ_x, φ_z) are also reported along with the minimum resolution (Δx_{min}, Δz_{min}). Note that the grid points within each element are the Gauss–Lobatto-Legendre points which are non-equidistant.

(N, m_x, m_z)	$N_x \times N_z$	φ_x	φ_z	Δx_{min}	Δz_{min}
(8, 40, 40)	321 × 321	0.92	0.96	2.506×10^{-3}	1.014×10^{-3}
(8, 64, 64)	513 × 513	0.92	0.96	3.249×10^{-4}	3.306×10^{-4}
(8, 144, 144)	1153 × 1153	1	0.985	6.961×10^{-4}	3.114×10^{-4}

As the dipole propagates downward and comes sufficiently close to the wall, a bottom boundary layer is formed with an induced vorticity of opposite sign from that of the primary monopole. Once the boundary-layer-induced vorticity interacts with the primary monopole (Fig. 8), the total enstrophy assumes a maximum value and dissipation of kinetic energy increases (Fig. 9). A secondary smaller increase of enstrophy occurs later in time when the primary vortex recirculates and interacts again with the bottom-boundary-generated vortices. The maximum enstrophy and time of its occurence for different grid resolutions, as compared to previous studies (Clercx and Bruneau, 2006; Kramer et al., 2007; Subich et al., 2013), are reported in Table 2. Lastly, the strength of the maximum negative vorticity of the primary monopole at $t/(H/U) = 0.6$ and the corresponding location of this maximum are reported in Table 3. For all the reported quantities in both Tables 2 and 3 agreement with previous studies improves for increasing resolution.

7.2.1. Extension to three-dimensions

The three-dimensional benchmark of the vortex-dipole collision is performed by extruding the $x - z$ computational domain Ω into $\Omega' = \Omega \times [0, 0.4]$. The $x - z$ resolution corresponds to the 513 × 513 grid shown in Table 1 along with a discretized y-direction of $N_y = 96$ grid points. White Gaussian noise is added onto every velocity component at each grid point of the computational domain, following the same approach as in Subich et al. (2013). In total, two three-dimensional runs are performed for the same grid resolution albeit with different noise amplitudes determined by the standard deviation (σ) of the Gaussian white noise which is set equal to $\sigma = 10^{-2}$, as in (Subich et al., 2013), and $\sigma = 10^{-3}$ for the stronger and for the weaker noise run, respectively.

Table 2
Maximum enstrophy and its time of occurence for different grid resolutions and reported values from previous studies (Clercx and Bruneau, 2006; Kramer et al., 2007; Subich et al., 2013).

$N_x \times N_z$	Enstrophy	Time
321 × 321	1883.550	0.341672
513 × 513	1893.337	0.341576
1153 × 1153	1897.243	0.341454
Clercx and Bruneau (2006)	1899	0.3414
Kramer et al. (2007)	1899.2	0.3414
Subich et al. (2013)	1899.921	0.341369

Table 3
Maximum absolute value of vorticity of the primary negative monopole and its location (x_m, z_m) for different grid resolutions at $t = 0.6$ and reported values from previous studies (Clercx and Bruneau, 2006; Kramer et al., 2007; Subich et al., 2013). The reported values of vorticity correspond to the same dipole location albeit for a different sign of vorticity but same direction as defined by the right-hand rule for a given coordinate system. The coordinate z_m is the vertical distance of the primary monopole from the bottom wall.

Grid points	(x_m, z_m)	Vorticity
321 × 321	(0.1495, 0.1269)	218.8496
513 × 513	(0.1489, 0.1258)	219.0806
1153 × 1153	(0.1505, 0.1257)	219.1692
Clercx and Bruneau (2006)	(0.151, 0.126)	219.4
Kramer et al. (2007)	(0.1506, 0.126)	219.29
Subich et al. (2013)	(0.1514, 0.1257)	219.2434

The evolution of averaged-in-y enstrophy and kinetic energy (KE) for the three-dimensional benchmarks is shown in Fig. 9. The $\sigma = 10^{-2}$ test-case follows the same behavior as the two-dimensional simulation up until the first interaction of the dipole with the no-slip-driven vorticity at the bottom boundary. At this point, enstrophy continues increasing which is a signature of fine structure, assuming larger values due to vortex stretching, with a concurrent increase of KE dissipation when compared to its two-dimensional counterpart. Although, the standard deviation of the noise is equal to the one reported in Subich et al. (2013), the three-dimensionalization of the flow starts much earlier, where any initial secondary transverse instabilities (Harris and Williamson, 2012) are bypassed due to the strong amplitude noise. This deviation is attributed to the weak filtering order ($q_y = 48$) in the periodic direction which is further enforced by examining the evolution of the enstrophy and KE for the $\sigma = 10^{-3}$ run and possible subtle changes in the structure of the noise.

In that case due to the weaker perturbations which drive a slower growth rate of any transverse flow structure, enstrophy starts deviating later in time from the two-dimensional benchmark, after the first interaction of the dipole with the bottom-generated vorticity. A second peak of enstrophy of almost equal strength with the first one is now observed during the second interaction of the dipole with the bottom boundary-generated vorticity similar to the one reported in Subich et al. (2013). The second enstrophy peak drives an increase in KE dissipation as in the $\sigma = 10^{-2}$ case, albeit less intense. Lastly, in Fig. 10 two isosurfaces of the transverse vorticity ω_y and one iso-surface of the magnitude of vorticity $|\omega| = \left(\omega_x^2 + \omega_y^2 + \omega_z^2\right)^{1/2}$ are shown for the $\sigma = 10^{-3}$ test-case soon after the second enstrophy peak. The almost parallel to the $x - z$ plane generated vorticity, indicates the three-dimensionalization of the flow; similar patterns are also observed in Subich et al. (2013).

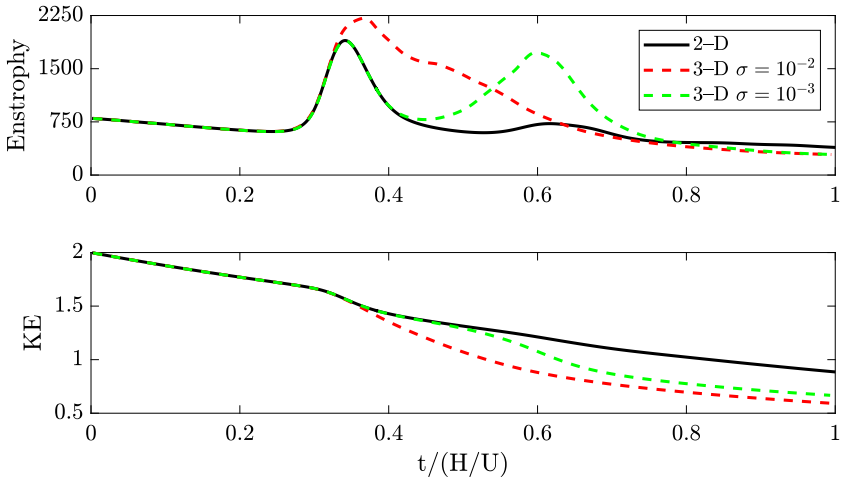


Fig. 9. Evolution of enstrophy (top) and kinetic energy (bottom) of the vortex-dipole normal wall collision for a resolution of $N_x \times N_z = 1153 \times 1153$ grid points (solid black line). Dashed lines correspond to the three-dimensional time evolution of, averaged in y-direction, enstrophy and kinetic energy for an additive white Gaussian noise of a 10^{-2} standard deviation (red) and 10^{-3} (green) respectively for a resolution of $N_x \times N_y \times N_z = 513 \times 96 \times 513$ grid points.

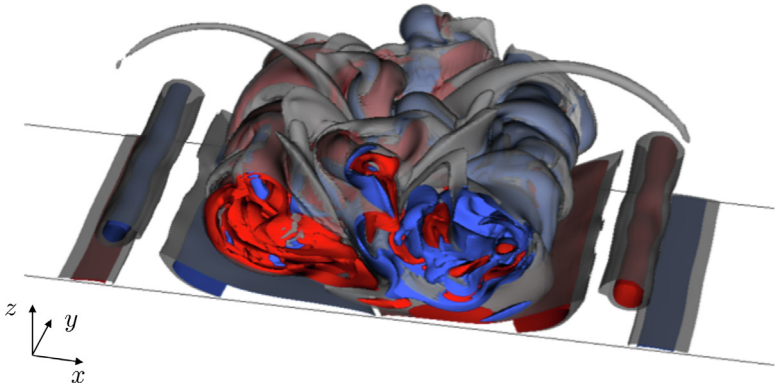


Fig. 10. Isosurfaces of the transverse vorticity $\omega_y = -50$ (blue), $\omega_y = 50$ (red) and the magnitude of vorticity $|\omega| = 40$ (gray), soon after the second interaction of the vortex dipole with the no-slip-driven generated vorticity at $t/(H/U) = 0.605$ for $\sigma = 10^{-3}$.

7.3. Stratified test-cases

The stratified test-cases share the common theme of the propagation of an internal solitary wave (ISW) of depression under different conditions. Starting from an idealized configuration, i.e., two-layer continuous stratification with no background current, uniform depth and inviscid flow, the complexity of the simulations is gradually increased. Initially, a small bathymetric feature is incorporated, culminating into a study of the three-dimensional propagation of an ISW over a realistic bathymetry, with realistic background currents and stratification, inspired by actual measurement in the South China Sea (SCS) (Lien et al., 2012, 2014). The propagation of an ISW is used as a platform to assess the robustness of the flow solver because of the delicate inherent balance between their non-linear steepening and physical dispersion. Consequently, if a numerical discretization is not carefully chosen, any spurious flow features, such as non-physical dispersive trailing waves, erroneous propagation speed or artificial amplitude attenuation due to numerical diffusion, introduced by the flow solver, will eventually compromise the simulation. By virtue of the hybrid SEM/FG numerical method used, the dynamic nature of the two-dimensional convectively breaking ISW is successfully captured along with the three-dimensional scales down to grid resolution.

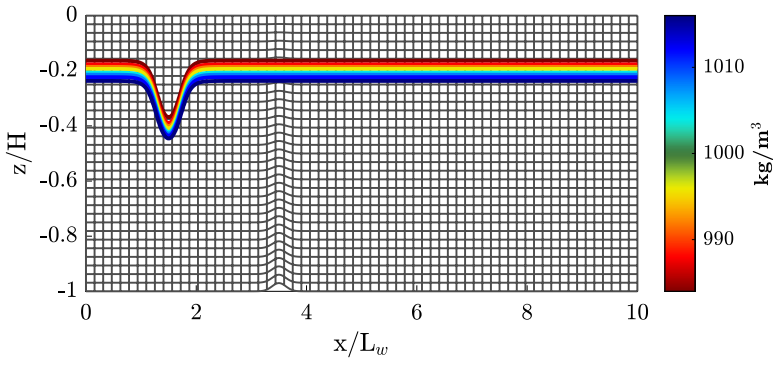
7.3.1. Tankscale ISW in uniform depth water

The initial ISW satisfies the Dubreil-Jacotin-Long (DJL) equation (Long, 1953; Dunphy et al., 2011) for an initial available potential energy (APE) of $APE = 0.05$ J/m under a continuous two-layer stratification

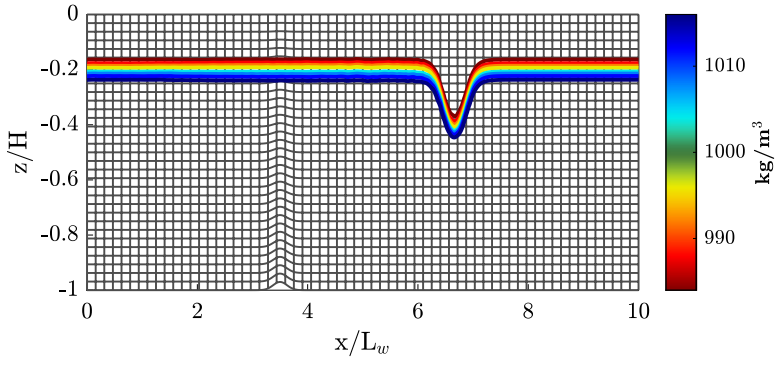
$$\frac{\rho}{\rho_0} = 1 - \frac{\Delta\rho}{2\rho_0} \tanh\left(\frac{z + h_1}{\delta}\right), \tag{28}$$

where $\rho_0 = 1000\,\mathrm{kg/m^3}$ is the reference density, $\Delta\rho = 40\,\mathrm{kg/m^3}$ is the difference between the lower and upper densities, $h_1 = 3\,\mathrm{cm}$ is the upper layer depth and $\delta = 0.5\,\mathrm{cm}$ is the interface thickness. The ISW propagates *inviscidly* where the viscous/diffusive terms in Eqs (1),(2) are deactivated and the time discretization is adjusted accordingly. The wave is expected to propagate in a uniform depth tank of $H = 15\,\mathrm{cm}$ and length $L = 10 \times L_w = 6.9\,\mathrm{m}$, where $L_w \approx 0.69\,\mathrm{m}$ is the ISW's width, with a DJL-prescribed constant propagation speed of $c = 0.1145412$ m/s. In regards to spatial discretization, a resolution of $N_x \times N_z = 513 \times 257$ grid points is used in the x and z directions respectively for a polynomial order $N = 8$ per element in each dimension and uniform sized quadrilateral elements. Fig. 11(a) shows the DJL-based ISW which is used as an initial condition for the tankscale uniform depth propagation as well as for the wave propagation over a small bathymetric feature as examined in Section 7.3.2.

The overall numerical dissipation of the flow solver, as generated by the numerical filtering, is investigated by calculating the kinetic energy $KE = \rho_0/2 \int_\Omega \mathbf{u}^2(\mathbf{x}, t)\, d\Omega$ of the ISW as a function of time. In the case of the inviscid propagation of an ISW in a constant depth waveguide, on purely physical grounds the KE of the ISW should remain constant. In practice however, when a high-order discretization is used, mitigation of any aliasing induced instabilities attributed to the non-linear terms



(a) $\mathbf{t/(L_w/c) = 0}$



(b) $\mathbf{t/(L_w/c) = 5.14}$

Fig. 11. The computational mesh for the simulation of the inviscid propagation of a tankscale ISW of depression over a Gaussian bump at $x/L_w = 3.5$. Shown in color are the isopycnal contours of the total density, equispaced from the reference density, for two different snapshots of the simulation. Grid resolution, domain dimensions and initial conditions are the same as for the flat-bottom test-case of Section 7.3.1.

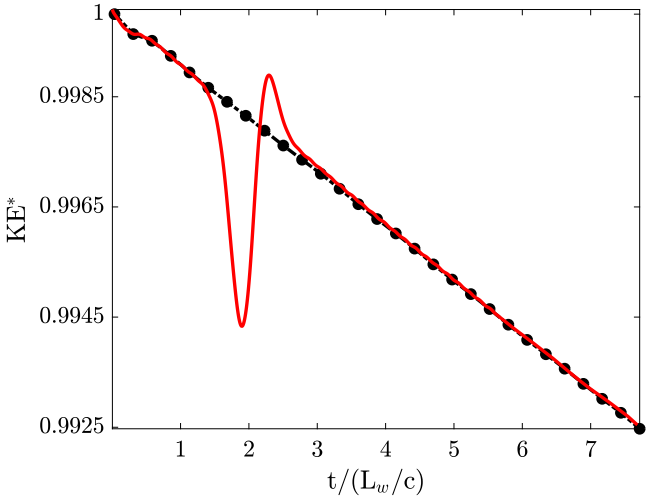


Fig. 12. Evolution of the kinetic energy of the inviscidly propagating tankscale ISW for the uniform depth (black dots) and the bump (solid red) test-cases, non-dimensionalized by the initial kinetic energy of the wave.

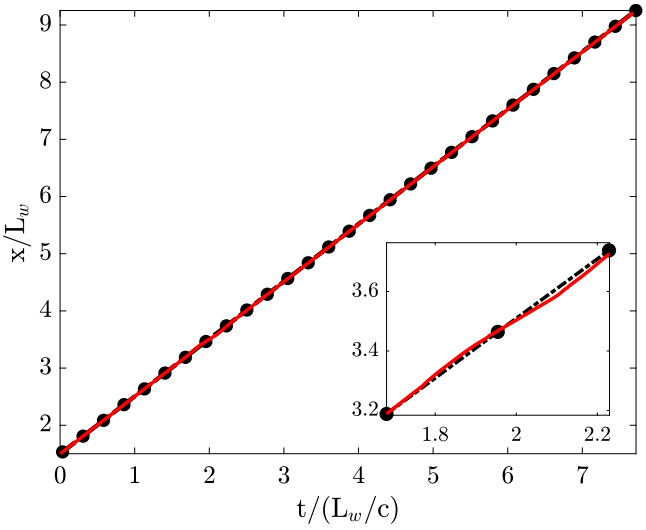


Fig. 13. Position of the trough of the ISW as a function of time for the flat bottom (black dots) and the bump (solid red). Bottom right panel is a detail of the position of the wave as it propagates over the bump. Notice the change of slope in the timeseries, i.e, propagation speed, as the wave goes over the bump and how the ISW recuperates its initial propagation speed as it moves back into uniform-depth water.

of INSE requires the introduction of an artificial, albeit controlled, dissipative mechanism. An initial increase of the KE dissipation rate is observed due to the reduced order of the time integration scheme for the first two time-steps of the solver (Fig. 12). Once the high-order time integration is established, the dominant source of dissipation is the weak exponential filter, of order $q_{xz} = 14$, used. More specifically, the KE assumes a constant value of $\partial \mathrm{KE}/\partial t \approx -1.6\times10^{-4}$ which corresponds to $\sim 0.1\%$ KE* loss per width of propagation where KE* is the ISW's kinetic energy normalized by the initial KE of the wave. Note that the overall loss of KE* can be used as a useful metric to assess the artificial dissipation introduced by a stratified flow solver.

For a uniform depth waveguide, inviscidly propagating ISWs should have a constant propagation speed c. Consequently, by tracking the position of the wave as a function of time, one should expect a straight line of slope c. The ability of the flow solver to perform reliably in this context is shown in Fig. 13, where the position of the trough

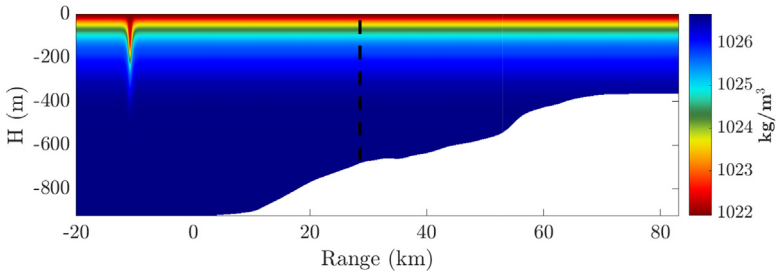


Fig. 14. Density field of the initial ISW shoaling over the realistic SCS bathymetric transect. The 20 km artificial plateau is also shown. The vertical black dashed line denotes where the 3-D simulation is initialized.

of the wave is given as a function of time. Obtained through linear regression, the numerically computed propagation speed of the wave, $c_{num} = 0.1145173\,\text{m/s}$ is in excellent agreement with the DJL-based initial propagation speed of the ISW with a relative error of 2.1×10^{-4}.

7.3.2. Tankscale ISW over a Gaussian bump

An additional level of complexity is now introduced by inserting a small bathymetric feature in the propagation path of the DJL-based ISW considered in Section 7.3.1. To accurately represent the Gaussian bump, the computational mesh and the respective elements in the vicinity of the topographic feature are curvilinear and highly deformed as shown in Fig. 11. Thus, the capability of the solver for supporting complex boundary geometries is tested. The ISW propagates over a Gaussian bump of height $h_b = 0.03H$ and width $L_b = 3L_w/4$, with the initial configuration shown in Fig. 11(a). The dimensions of the bump are chosen for the ISW to propagate over this feature without inducing any convective or shear instabilities. Over such a gentle bathymetric feature, the ISW responds to the bathymetric change, by slightly adjusting its waveform, but recovers its original shape (Fig. 11(b)) and propagation speed as soon as it moves back into uniform-depth water. Therefore, this adjustment and recovery of the ISW waveform are an additional aspect of the flow solver testing provided by this benchmark.

As the wave approaches the bump, it slows down and its KE is converted into available potential energy (APE) (Lamb and Nguyen, 2009). Note that, the computation of the ISW's APE is not included in this work. The reverse energy exchange is observed once the ISW goes over the bump: the wave accelerates, its APE decreases with an accompanying increase of its KE. The evolution of the ISW's KE as it propagates over the bump are shown in Fig. 12. Notice, when the ISW has gone over the bump, its KE assumes values close to the uniform bottom test-case, indicating how the ISW recuperates its original waveform as it propagates again over a uniform bottom. The position of the trough of the wave as function of time is shown in Fig. 13. As the wave propagates over the bump, the slope, i.e., the propagation speed, changes but quickly returns close to the original values as the water depth becomes uniform. Lastly, no error is introduced by the representation of the complex boundary due to the high polynomial order used for each element (Steinmoeller et al., 2016).

7.4. South China sea shoaling ISW

In this section, the three-dimensional inviscid shoaling of an ISW of depression over a realistic gentle bathymetric slope in the South China Sea (SCS) and its three-dimensional convective instability are explored. This benchmark problem consists of the ultimate objective of code development regarding the three-dimensional propagation of ISWs, and NLIWs in general, over realistic configurations. Namely, the flow solver has to successfully capture the ISW adjustment over long distances on a constantly varying waveguide by combining realistic background stratification/current profiles based on the observational work of Lien et al. (2012, 2014) in the SCS. Simulations (Towns et al., 2014) are performed primarily following the methodology of Rivera-Rosario et al. (2020, 2022) albeit for a different numerical method, filtering order used, and value of viscosity.

Table 4
Grid parameters and resolution for the shoaling ISW. Minimum resolution in the $x-z$ directions (Δx_{min}, Δz_{min}) and orders of the exponential filter (q_{xz}, q_y), for different ranges of the SCS transect, are reported. Transverse parameters of the 3-D simulation, as the resolution (Δy), number of transverse points (N_y), and time-step (Δt) are also included. Note that the reported number of elements in the x-direction, m_x, is the total number of elements of the computational domain and not the number of elements per window.

Range (km)	Δx_{min} (m)	Δz_{min} (m)	(q_{xz}, q_y)
0	4.13	0.52	(14, 20)
54	2.07	0.29	(13, 20)
60	2.07	0.24	(13, 20)
(N, m_x, m_z)	Δy (m)	Δt (s)	N_y
(7, 2374, 60)	0.7812	0.227	64

In regards to the three-dimensional shoaling ISW simulation, adhering to a hybrid spatial discretization may at first seem restrictive, since transverse bathymetric variations cannot be represented. Nonetheless, the restriction to such a bathymetric homogeneity in the y-direction is offset by the solver's implementation and the resulting performance gains. As such, the decoupling of the transverse direction during the implicit solves, greatly reduces the computational cost per time-step and hence allows the optimal resolution of as large as possible a range of turbulent length scales (Diamantopoulos, 2021). Consequently, and without deviating from the observed normal-to-isobath propagation of ISWs in the SCS, this hybrid approach allows, in an optimized fashion, the investigation of the onset of the convective instability, and the subsequent development of turbulence, induced solely by bathymetric variations on the $x-z$ plane.

The initial ISW is the baseline wave of amplitude $A = 143\,\text{m}$ as in Rivera-Rosario et al. (2020). Initially, the ISW starts propagating from an artificial plateau of 921 m, to bypass any artificial shoaling response at the wavescale. The simulation is originally performed in two-dimensions up to approximately the 25.7 km of the transect (Fig. 14). At this stage of the simulation, the computational domain is extruded in the y-direction with a width of $L_y = 50\,\text{m}$.

In this work, a weak filtering is utilized until the onset of the convective instability where the flow is highly energetic and thus the filtering becomes slightly stronger in the $x-z$ plane as shown in Table 4. Other modeling parameters and grid resolution for the whole computational domain are shown in the same table. Notice the variation of the resolution in the x-direction for increasing range values. Although the minimum resolution in the x-direction is restricted to $\Delta x \approx 2\,\text{m}$ as the wave becomes convectively unstable, finer-scale three-dimensional structures within the core of the wave can still be identified (cf. Fig. 18). Further quantitative results, outside of the scope of this paper, are needed to determine whether the observed finer-scale three-dimensional structure developed inside the wave is truly turbulent. Such computationally demanding, turbulence-resolving simulations will be the topic of a future study (Diamantopoulos, 2021).

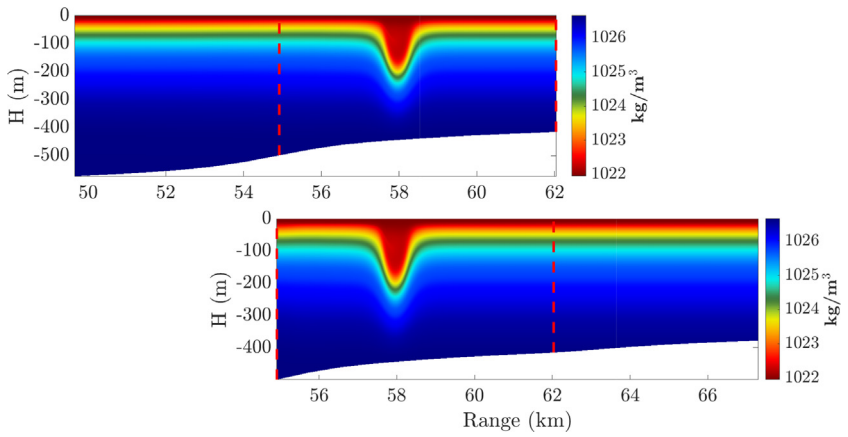


Fig. 15. An example of two successive simulations of the ISW in the SCS using overlapping computational windows. The two red vertical dashed lines denote the limits of the overlapping region among the two computational domains. Once the ISW propagates, from left to right, within the desired overlapping region (top), another simulation is initialized (bottom) where the new computational domain includes the overlapping region of the previous window as well as the subset of the oceanic transect immediately downstream.

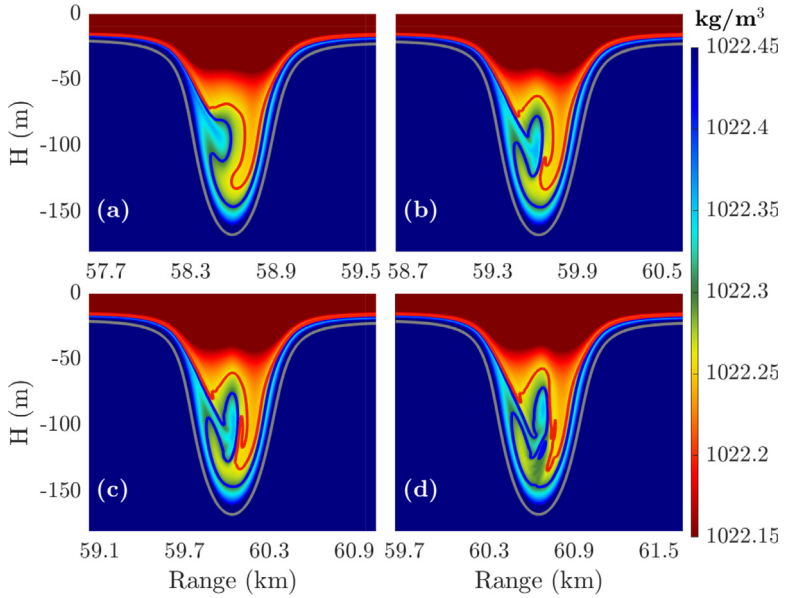


Fig. 16. Evolution of the convective instability as the ISW shoals through visualization of the y-averaged density field. The ISW retains its symmetric shape while heavier fluid detaches from the rear of the wave and plunges on top of lighter fluid within the wave interior. The colored solid lines are the y-averaged isopycnals of 1022.58 kg/m^3 (gray/reference density), 1022.30 kg/m^3 (blue) and 1022.26 kg/m^3 (red) as shown in Fig. 18.

Both the two-dimensional and three-dimensional simulations were performed in overlapping windows, i.e., successive subsets of the computational domain on the $x-z$ plane which encompass the wave as it propagates over approximately 7 wavelengths across each window (Fig. 15). For the three-dimensional runs each window spans approximately 13 km of the transect with an overlapping region of 6 km.

This overlapping technique greatly reduces the overall computational cost per time-step since only a fraction of the degrees of freedom, which account for the full SCS transect, are solved during each fractional step of the solver (Rivera-Rosario et al., 2020). Therefore, the resulting FLOPS, the introduced parallel communication overhead per time-step as well as the memory footprint are substantially smaller when compared to a simulation that would be performed over the full SCS transect.

Upon the initialization of the first three-dimensional windowed run, perturbations (Rivera-Rosario et al., 2022) of amplitude equal to 3×10^{-5} (units are m/s for velocity and kg/m^3 for density) are inserted to each field variable. The noise is added once in every velocity component and in the density perturbation on every grid point of the computational domain. Specifically, white noise is inserted on each $x-z$ plane, adhering to a $-5/3$ slope spectra in the transverse y-direction.

As the ISW shoals and enters shallower waters, its propagation speed decreases below the maximum induced horizontal velocity (c.f. Fig. 19). Consequently, it undergoes a distinct convective instability at a critical depth. Unlike the setup of Vlasenko et al. (2005) where the rear of the wave significantly steepens because of the much higher proximity of pycnocline to the bottom, the SCS shoaling ISW preserves its symmetric waveform. Fig. 16 shows the onset of the trapped core formation and evolution of the convective instability for the SCS shoaling ISW. As the wave shoals, heavier isopycnals than those in its interior detach from the rear of the wave (Fig. 16a) and eventually plunge, in the form of an inclined gravity current, into the interior of the wave (Fig. 16b-c). The resulting flow structure inside the wave resembles the corresponding one reported in recent studies on subsurface trapped cores (He et al., 2019; Rivera-Rosario et al., 2020, 2022). The onset of the convective instability leads to the development of a lateral instability (Fig. 17) which eventually grows in amplitude as the core develops. This heavy-over-light configuration eventually drives the formation of the subsurface core (He et al., 2019) and the associated

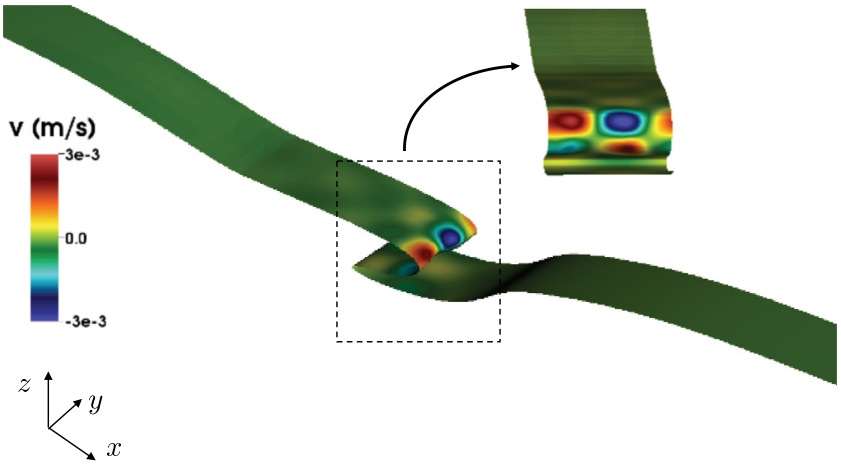


Fig. 17. Structure of the transverse velocity at 58.2 km of the transect on the 1022.26 kg/m^3 density isosurface. The early manifestation of the secondary lateral instability and its structure are shown.

turbulent-like structure and localized mixed regions inside the wave in three-dimensions (Fig. 18).

Once the ISW starts to become convectively unstable, a secondary transverse instability gradually grows in amplitude (Fig. 17) which drives the three-dimensionalization of the flow field as the ISW continues to shoal (Fig. 18). As previously stated, the transition to turbulence from the formation of the trapped core, its persistence, as well as the induced mixing, will be further explored via higher resolution simulations in a future study. Note that the initial transverse mode differs from the one reported in Rivera-Rosario et al. (2022) possibly due to differences in the inserted perturbations. Specifically, perturbations are inserted only in two of the velocity components with the transverse velocity component being computed by virtue of the projection onto a solenoidal velocity field during the time-advancement of INSE (Eq. (10)). The development of three-dimensional transition (Fig. 17) and turbulent-like structure of the ISW's interior (Fig. 18) is delayed as compared to field measurements (Lien et al., 2014) where a fully turbulent trapped core at approximately the 54th km of the transect was observed. For further comparison to field data and different noise insertion strategies for accelerating the three-dimensionalization of the flow field, the reader is referred to the study of Rivera-Rosario et al. (2022).

Fig. 19 shows the variations of the maximum induced horizontal velocity U_{max} and propagation speed c as the wave shoals. For more details on the methodology of computing c the reader is referred to the work of Rivera-Rosario et al. (2020, 2022). Both of these properties are computed from the y-averaged velocity field since it is shown that their differences per $x-z$ plane are negligible. The along propagation-path location of the onset of the convective instability which takes place when the maximum horizontal induced velocity exceeds the propagation speed of the wave, i.e., $U_{max}/c > 1$, is consistent with the locations of snapshots of the isopycnal overturning shown in Fig. 16. Furthermore, as the wave shoals, the U_{max}/c ratio follows the same evolution (Fig. 19) as in Rivera-Rosario et al. (2020).

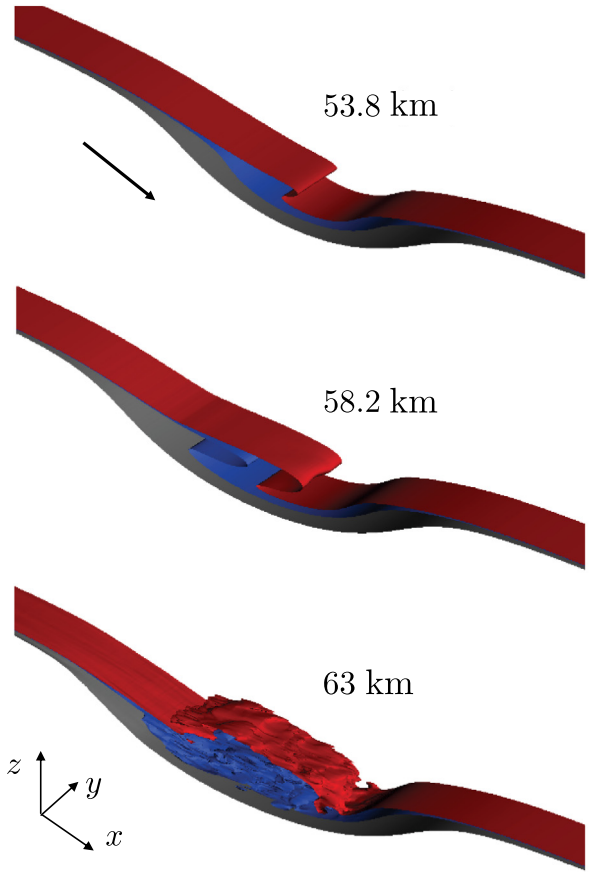


Fig. 18. Evolution of the 3-D convectively breaking, shoaling ISW. Three density isosurfaces are shown: 1022.58 kg/m^3 (gray/reference density), 1022.30 kg/m^3 (blue) and 1022.26 kg/m^3 (red). Top: Onset of the primary convective instability which is mainly two-dimensional. Middle: Heavier fluid clearly detaches from the rear of the wave while the signature of the lateral secondary instability starts becoming noticeable. Bottom: The flow is fully three-dimensional and turbulent-like. The arrow indicates the propagating direction of the ISW.

8. Conclusions and future work

In this work, a continuous hybrid nodal spectral element/Fourier Galerkin flow solver for the simulation of fully non-hydrostatic stratified flows is presented. The solver is specifically designed for the long, high aspect ratio and variable bathymetry computational domains encountered in the normal to isobaths propagation of non-linear internal waves and the resulting turbulence in the wave interior.

A defining feature of the non-linear internal waves is their strong non-hydrostatic nature which necessitates the solution of a pressure Poisson problem. When combined with the high aspect ratio complex bathymetry domains, the efficacy of numerically solving the pressure Poisson equation is of paramount importance for the overall performance of the flow solver. Adopting a domain decomposition technique for the pressure Poisson equation, the resulting condensed Schur complement problem is solved via a block-Jacobi deflated preconditioned

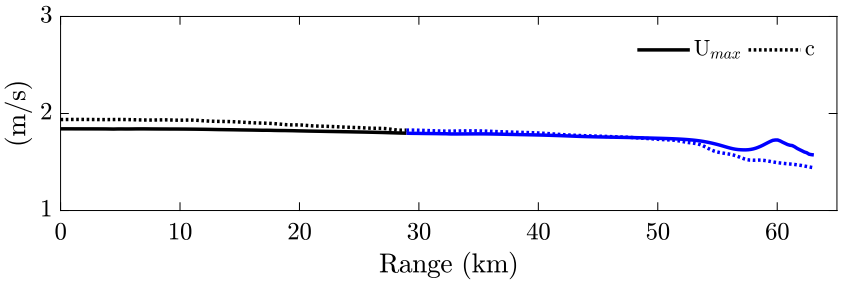


Fig. 19. Maximum horizontal induced velocity (U_{max}) and propagation speed (c) of the wave as it shoals. The black and blue lines denote the two-dimensional and three-dimensional simulations respectively.

conjugate gradient iterative solver. A robust preconditioning technique is used, which ensures favorable convergence properties, i.e., low iteration count for the examined domains and flow problems, as recently demonstrated in Joshi et al. (2016b).

The initial flow solver implementation meets its design goals of producing three-dimensional physics from a collection of coupled two-dimensional problems, minimizing communications, and expressing the method as regions of structured computations. Qualitative assessments indicate the implementation will scale well to tackle environmental scale problems and will be validated with a future scaling study. Future performance improvements include identifying more efficient choices of deflation vectors to reduce the pressure solver's iteration count, as well as exploring acceleration libraries providing tuned implementations for small dense linear algebra problems as reported in Section 5.2.

The overall accuracy of the flow solver is demonstrated through a suite of benchmarks showing a good agreement when compared to analytical solutions and previous studies. The benchmarking effort culminates with the three-dimensional shoaling of an internal solitary wave of depression over realistic bathymetry and background conditions in the South China Sea motivated by *in situ* observations of Lien et al. (2012, 2014). The flow solver captures the key features of the shoaling wave along with turbulent-like structures in its interior, showing a really good agreement with previous studies (Rivera-Rosario et al., 2020, 2022). Future turbulence resolving/computationally intensive simulations of the shoaling internal solitary wave in the South China Sea will be performed utilizing this solver (Diamantopoulos, 2021). Additional applications in stratified flows such as the simulation of internal swash zones (Emery and Gunnerson, 1973; Thorpe and Lemmin, 1999; Davis et al., 2020) and the potentially related boundary-interior exchange (McPhee-Shaw et al., 2021), excited by deep water waves will also be investigated.

CRediT authorship contribution statement

Theodoros Diamantopoulos: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft. **Sumedh M. Joshi:** Conceptualization, Methodology, Software. **Greg N. Thomsen:** Software, Writing – original draft. **Gustavo Rivera-Rosario:** Conceptualization, Methodology, Software, Writing – review & editing. **Peter J. Diamessis:** Conceptualization, Funding acquisition, Project administration, Resources, Supervision, Writing – original draft. **Kristopher L. Rowe:** Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Professor Charles F. Van Loan for his comments on the domain decomposition and Schur complement matrices. The authors would also like to thank Professors Larry Redekopp and Jan Hesthaven for sparking the original inspiration to develop a high-order-element-based flow solver to simulate shoaling nonlinear internal waves. Jorge Escobar-Vargas was instrumental in the very early stages of model development. Sumedh Joshi made critical contributions in the earlier stages of this effort. The authors are additionally grateful to Dr. Ren-Chieh Lien and Professor Kevin Lamb for their patience and ceaseless encouragement in this long development endeavor. The authors would also like to thank Professor Marek Stastna for his suggestions during the revision of the manuscript. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. The shoaling ISW simulations used the Extreme Science and Engineering Discovery Environment (XSEDE) Stampede2 at the Texas Advanced Computing Center (TACC) through allocation TG-EES200010. Financial support is gratefully acknowledged from National Science Foundation - Division of Ocean Sciences (OCE) grant number 1634257.

The work in this paper is the culmination of a 15-year-long grassroots effort obsessively pursued by a small, highly dedicated, number of individuals. We have persevered past adversity and have brought our labor of love to completion. This publication is dedicated to the memory of Sumedh M. Joshi who, sadly, does not get to see the results of his efforts.

Appendix A. Exponential filtering

Since a continuous Galerkin discretization is used in the $x-z$ directions, it is crucial for the filtered quantity to preserve the C^0 continuity of the numerical method (Boyd, 1998). Therefore, a discrete polynomial transform (DPT) takes place before the application of the exponential filter using appropriate basis functions which ensure the C^0 continuity among elemental interfaces. Essentially, a DPT is the mapping from the nodal coefficient space, i.e., Lagrange polynomials, to the modal coefficient space. Instead of the commonly used Legendre basis functions, the modal coefficients of a field variable are computed for the boundary-adaptive basis functions (Canuto et al., 2007; Karniadakis and Sherwin, 2013). In one-dimension, the boundary-adaptive basis functions are comprised of two linear basis functions which are non-zero at the two boundaries and a series of basis functions or bubble modes of increasing polynomial order, i.e, *hierarchical*, which are zero at the boundaries. Thus, a subsequent filtering, only of the respective bubble modes, ensures the continuity among element interfaces.

Let $\boldsymbol{\Sigma} \in \mathbb{R}^{N+1\times N+1}$ be an exponential filtering diagonal matrix with corresponding entries the values of the filtering function evaluated at a given polynomial mode (see Eq. (36) in Diamessis et al., 2005) where N is the polynomial order of the expansion. Then the one-dimensional filtering matrix $\mathbf{F} \in \mathbb{R}^{N+1\times N+1}$ is given by the following expression

$$\mathbf{F} = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^{-1}, \tag{A.1}$$

where $\mathbf{V} \in \mathbb{R}^{N+1\times N+1}$ is the forward discrete-polynomial-transform matrix with columns the computed boundary-adaptive basis at the Gauss–Lobatto-Legendre points (Canuto et al., 2007). Lastly, the inverse $\mathbf{V}^{-1}$ denotes the backward transform from nodal to modal coefficient space. Note that the two-dimensional equivalent filter matrix is simply computed by taking the Kronecker product of the one-dimensional filter matrix $\mathbf{F}$ with itself.

Appendix B. Non-linear terms calculation

The explicit step of the time-splitting scheme (Eq. (6)) for any velocity component u of the velocity vector is shown following a Galerkin approximation in the non-periodic domain Ω. To simplify notation, the density perturbation term is omitted and the intermediate velocity $\tilde{u}$ is time-advanced using a first order time integration scheme ($J = 1$), i.e., Euler method, which can be easily extended to a higher accuracy time integration. Therefore, Eq. (6), for a given test function v, is written as

$$\int_\Omega v\tilde{u}d\Omega = \int_\Omega vud\Omega - \Delta t \int_\Omega vN\{u\}d\Omega \ , \tag{B.1}$$

where $N\{u\} = \mathbf{u}\cdot\nabla u$ is the non-linear operator. Based on the discretization presented on Section 4 the non-linear term $N\{u\}$ is approximated in Ω, as a linear combination of two-dimensional Lagrange polynomials

$$N\{u\}(x,y,z) = \sum_{k=1}^{n} N\{u\}_k(y)h_k(x,z) \ , \tag{B.2}$$

where n are the degrees of freedom within a $x-z$ plane. Note that the test-function v, the intermediate velocity $\tilde{u}$ and the previous time-step velocity u adhere to the same discretization. Thus, Eq. (B.1) is discretized into the following matrix form

$$\mathbf{M}\tilde{\mathbf{u}}(y) = \mathbf{M}\mathbf{u}(y) - \Delta t\mathbf{M}\mathbf{N}\{\mathbf{u}\}(y) \ , \tag{B.3}$$

where $\mathbf{M}$ is the mass matrix and $\mathbf{N}\{\mathbf{u}\}(y) = \{\mathbf{N}\{\mathbf{u}\}_k(y)\}$, $\mathbf{u}(y) = \{u_k(y)\}$, $\tilde{\mathbf{u}}(y) = \{\tilde{u}_k(y)\}$ are the nodal values of $N\{u\}$, u and $\tilde{u}$ respectively.

Now the second term of the right-hand-side of Eq. (B.3) is equal to

$$\mathbf{M}\mathbf{N}\{\mathbf{u}\}(y) = \mathbf{C}\mathbf{u}(y) \ , \tag{B.4}$$

where $C_{ij} = \int_\Omega h_i\mathbf{u}\cdot\nabla h_j d\Omega$ is the weak-form-based discrete non-linear operator. In high Re advection dominated flows, such as the propagation of non-linear internal waves and the associated turbulence, aliasing induced instabilities due to inexact integration of the entries of $\mathbf{C}$ matrix may compromise the simulation. This is of particular importance on complex computational domains where the integrand of C_{ij} includes the associated metric terms for the mapping of the physical domain to the computational domain. Furthermore, since it is a common practice for the use of an isoparametric mapping (Kopriva, 2009) (see Appendix C for more details), the integrand of C_{ij} has an elevated polynomial order and, thus, any inexact numerical integration using the Gauss–Lobatto-Legendre points (Karniadakis and Sherwin, 2013) is exacerbated if not properly treated. On this account, stability is ensured via polynomial over-integration on $\mathbf{C}$ to retain its skew-symmetry (Malm et al., 2013). Briefly, the integrand of C_{ij} is spectrally interpolated into an over-integration grid for its integral to be computed exactly through numerical integration. For more details regarding the computation of $\mathbf{C}$ in deformed geometries the reader is referred to Deville et al. (2002), Malm et al. (2013).

Lastly, the Fourier coefficients of the intermediate velocity $\hat{\tilde{u}}$ are obtained by first inverting the diagonal mass matrix $\mathbf{M}$ of Eq. (B.3) followed by a discrete Fourier transform in y, $\mathcal{F}_y$

$$\hat{\tilde{\mathbf{u}}}(k_n) = \mathcal{F}_y\{\mathbf{u}(y) - \Delta t\mathbf{N}\{\mathbf{u}\}(y)\} \ , \tag{B.5}$$

where k_n is a transverse in y wavenumber.

Appendix C. Mapping to curvilinear elements

An isoparametric approximation (Kopriva, 2009) is adopted for representing the mapping between the computational grid coordinates of the reference quadrilateral element $\xi,\eta \in \Omega_e = [-1,1]\times[-1,1]$ and the physical coordinates $x_e(\xi,\eta), z_e(\xi,\eta) \in \mathbb{R}^2$ for each element

$$\begin{aligned} x_e(\xi,\eta) &= \sum_{i=0}^{N}\sum_{j=0}^{N} x_{ij}l_i(\xi)l_j(\eta) \\ z_e(\xi,\eta) &= \sum_{i=0}^{N}\sum_{j=0}^{N} z_{ij}l_i(\xi)l_j(\eta) \ , \end{aligned} \tag{C.1}$$

where x_{ij}, z_{ij} are the physical grid coordinates within a quadrilateral element and l_i, l_j are the one-dimensional Lagrange polynomials in the reference domain. From Eq. (C.1), the partial derivatives of x_e, z_e with respect to the reference coordinates ξ,η can be easily computed numerically via the use of two-dimensional spectral differentiation matrices (Costa and Don, 2000). Furthermore the derivatives of the inverse mapping in two-dimensions are computed as (Kopriva, 2009)

$$\begin{bmatrix} \frac{\partial\xi}{\partial x} & \frac{\partial\eta}{\partial x} \\ \frac{\partial\xi}{\partial z} & \frac{\partial\eta}{\partial z} \end{bmatrix} = \frac{1}{J}\begin{bmatrix} \frac{\partial z}{\partial\eta} & -\frac{\partial z}{\partial\xi} \\ -\frac{\partial x}{\partial\eta} & \frac{\partial x}{\partial\xi} \end{bmatrix} \ , \tag{C.2}$$

where $J = \frac{\partial x}{\partial\xi}\frac{\partial z}{\partial\eta} - \frac{\partial z}{\partial\xi}\frac{\partial x}{\partial\eta} > 0$ is the Jacobian of the transformation. Notice that both the derivatives of the inverse mapping with respect to the physical coordinates and the Jacobian are used throughout the computation of derivatives and integrals respectively (e.g. first left-hand-side term of Eq. (14)). For more details regarding the computation of the elemental stiffness and mass matrices (cf. Eq. (16)) for curvilinear elements the reader is referred to the textbook of Deville et al. (2002).

In the context of a domain decomposition, the use of curvilinear elements, dictated by complex computational geometries, requires a unique mapping to the reference element Ω_e. As such, each elemental matrix differs from element to element, since the Jacobian of the transformation J assumes different values per grid point (Deville et al., 2002). Consequently, any tensor-product-based calculations for the inversion of the self-interaction matrices H_{ii} and $S^{(1)}_{hh}$ (see Eq. (20)) are not in this case applicable (Huismann et al., 2019; Couzy and Deville, 1995).

References

Alford, M.H., Peacock, T., MacKinnon, J.A., Nash, J.D., Buijsman, M.C., Centurioni, L.R., Chao, S.-Y., Chang, M.-H., Farmer, D.M., Fringer, O.B., et al., 2015. The formation and fate of internal waves in the South China Sea. Nature 521 (7550), 65–69.

Aubry, R., Mut, F., Löhner, R., Cebral, J.R., 2008. Deflated preconditioned conjugate gradient solvers for the Pressure–Poisson equation. J. Comput. Phys. 227 (24), 10196–10208.

Balay, S., Abhyankar, S., Adams, M.F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E.M., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M.G., Kong, F., Kruger, S., May, D.A., McInnes, L.C., Mills, R.T., Mitchell, L., Munson, T., Roman, J.E., Rupp, K., Sanan, P., Sarich, J., Smith, B.F., Zampini, S., Zhang, H., Zhang, H., Zhang, J., 2021. PETSc web page. URL https://petsc.org/.

Barad, M.F., Colella, P., Schladow, S.G., 2009. An adaptive cut-cell method for environmental fluid mechanics. Internat. J. Numer. Methods Fluids 60 (5), 473–514.

Blackburn, H.M., Lee, D., Albrecht, T., Singh, J., 2019. Semtex: A spectral element–Fourier solver for the incompressible Navier–Stokes equations in cylindrical or cartesian coordinates. Comput. Phys. Comm. 245, 106804.

Boegman, L., Stastna, M., 2019. Sediment resuspension and transport by internal solitary waves. Ann. Rev. Fluid Mech. 51, 129–154.

Bolis, A., Cantwell, C.D., Moxey, D., Serson, D., Sherwin, S.J., 2016. An adaptable parallel algorithm for the direct numerical simulation of incompressible turbulent flows using a Fourier spectral/hp element method and MPI virtual topologies. Comput. Phys. Commun. 206, 17–25.

Borrell, R., Lehmkuhl, O., Trias, F.X., Oliva, A., 2011. Parallel direct Poisson solver for discretisations with one Fourier diagonalisable direction. J. Comput. Phys. 230 (12), 4723–4741.

Boyd, J.P., 1998. Two comments on filtering (artificial viscosity) for Chebyshev and Legendre spectral and spectral element methods: preserving boundary conditions and interpretation of the filter as a diffusion. J. Comput. Phys. 143 (1), 283–288.

Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A., 2007. Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics. Springer Science & Business Media.

Carvalho, L.M., Giraud, L., Meurant, G., 2001. Local preconditioners for two-level non-overlapping domain decomposition methods. Numer. Linear Algebra Appl. 8 (4), 207–227.

Clercx, H.J., Bruneau, C.-H., 2006. The normal and oblique collision of a dipole with a no-slip boundary. Comput. & Fluids 35 (3), 245–279.

Costa, B., Don, W.S., 2000. On the computation of high order pseudospectral derivatives. Appl. Numer. Math. 33 (1–4), 151–159.

Couzy, W., Deville, M.O., 1995. A fast Schur complement method for the spectral element discretization of the incompressible Navier-Stokes equations. J. Comput. Phys. 116 (1), 135–142.

Davis, K.A., Arthur, R.S., Reid, E.C., Rogers, J.S., Fringer, O.B., DeCarlo, T.M., Cohen, A.L., 2020. Fate of internal waves on a shallow shelf. J. Geophys. Res. Oceans 125 (5), e2019JC015377.

de Bruyn Kops, S.M., 2015. Classical scaling and intermittency in strongly stratified Boussinesq turbulence. J. Fluid Mech. 775, 436.

Deville, M.O., Fischer, P.F., Mund, E., et al., 2002. High-Order Methods for Incompressible Fluid Flow. Vol. 9, Cambridge University Press.

Diamantopoulos, T., 2021. A high-order hybrid flow solver for the simulation of nonlinear internal waves in long complex domains. (Ph.D. thesis). Cornell University.

Diamantopoulos, T., Diamessis, P.J., Stastna, M., 2021. On the formulation and implementation of the stress-free boundary condition over deformed bathymetry using a spectral-element-method-based incompressible Navier–Stokes equations solver. Ocean Model. 101834.

Diamessis, P.J., Domaradzki, J., Hesthaven, J.S., 2005. A spectral multidomain penalty method model for the simulation of high Reynolds number localized incompressible stratified turbulence. J. Comput. Phys. 202 (1), 298–322.

Dunphy, M., Subich, C., Stastna, M., 2011. Spectral methods for internal waves: indistinguishable density profiles and double-humped solitary waves. Nonlinear Process. Geophys. 18 (3), 351.

Egbert, G., Ray, R., 2000. Significant dissipation of tidal energy in the deep ocean inferred from satellite altimeter data. Nature 405 (6788), 775–778.

Emery, K.O., Gunnerson, C.G., 1973. Internal swash and surf. Proc. Natl. Acad. Sci. 70 (8), 2379–2380.

Escobar-Vargas, J., 2012. A spectral multidomain penalty method solver for environmental flow processes. (Ph.D. thesis). Cornell University.

Escobar-Vargas, J., Diamessis, P., Sakai, T., 2014. A spectral quadrilateral multidomain penalty method model for high Reynolds number incompressible stratified flows. Internat. J. Numer. Methods Fluids 75 (6), 403–425.

Fischer, P.F., Rønquist, E.M., 1994. Spectral element methods for large scale parallel Navier—Stokes calculations. Comput. Methods Appl. Mech. Engrg. 116 (1–4), 69–76.

Frison, G., Kouzoupis, D., Sartor, T., Zanelli, A., Diehl, M., 2018. BLASFEO. ACM Trans. Math. Software 44 (4), 1–30. http://dx.doi.org/10.1145/3210754.

Golub, G., Van Loan, C., 2013. Matrix Computations. In: Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press.

Harris, D., Williamson, C., 2012. Instability of secondary vortices generated by a vortex pair in ground effect. J. Fluid Mech. 700, 148–186.

He, Y., Lamb, K.G., Lien, R.-C., 2019. Internal solitary waves with subsurface cores. J. Fluid Mech. 873, 1–17.

Heinecke, A., Henry, G., Hutchinson, M., Pabst, H., 2016. LIBXSMM: accelerating small matrix multiplications by runtime code generation. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2016, Salt Lake City, UT, USA, November 13-18, 2016. IEEE Computer Society, pp. 981–991. http://dx.doi.org/10.1109/SC.2016.83.

Hesthaven, J.S., 1998. A stable penalty method for the compressible Navier–Stokes equations: III. Multidimensional domain decomposition schemes. SIAM J. Sci. Comput. 20 (1), 62–93.

Hesthaven, J.S., Warburton, T., 2007. Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications. Springer Science & Business Media.

Hirshman, S.P., Perumalla, K.S., Lynch, V.E., Sanchez, R., 2010. BCYCLIC: A parallel block tridiagonal matrix cyclic solver. J. Comput. Phys. 229 (18), 6392–6404.

Huismann, I., Stiller, J., Fröhlich, J., 2017. Factorizing the factorization–a spectral-element solver for elliptic equations with linear operation count. J. Comput. Phys. 346, 437–448.

Huismann, I., Stiller, J., Fröhlich, J., 2019. Scaling to the stars–a linearly scaling elliptic solver for p-multigrid. J. Comput. Phys. 398, 108868.

Joshi, S.M., Diamessis, P.J., Steinmoeller, D.T., Stastna, M., Thomsen, G.N., 2016a. A post-processing technique for stabilizing the discontinuous pressure projection operator in marginally-resolved incompressible inviscid flow. Comput. & Fluids 139, 120–129.

Joshi, S.M., Thomsen, G.N., Diamessis, P.J., 2016b. Deflation-accelerated preconditioning of the Poisson–Neumann schur problem on long domains with a high-order discontinuous element-based collocation method. J. Comput. Phys. 313, 209–232.

Karniadakis, G., 1990. Spectral element-Fourier methods for incompressible turbulent flows. Comput. Methods Appl. Mech. Engrg. 80 (1–3), 367–380.

Karniadakis, G.E., Israeli, M., Orszag, S.A., 1991. High-order splitting methods for the incompressible Navier-Stokes equations. J. Comput. Phys. 97 (2), 414–443.

Karniadakis, G., Sherwin, S., 2013. Spectral/Hp Element Methods for Computational Fluid Dynamics. Oxford University Press.

Kirby, R.M., Karniadakis, G.E., 2003. De-aliasing on non-uniform grids: algorithms and applications. J. Comput. Phys. 191 (1), 249–264.

Kirby, R.M., Sherwin, S.J., 2006. Stabilisation of spectral/hp element methods through spectral vanishing viscosity: Application to fluid mechanics modelling. Comput. Methods Appl. Mech. Engrg. 195 (23–24), 3128–3144.

Kopriva, D.A., 2009. Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers. Springer Science & Business Media.

Kovasznay, L., 1948. Laminar flow behind a two-dimensional grid. Math. Proc. Camb. Phil. Soc. 44 (1), 58–62.

Kramer, W., Clercx, H., van Heijst, G., 2007. Vorticity dynamics of a dipole colliding with a no-slip wall. Phys. Fluids 19 (12), 126603.

Kundu, P., Cohen, I., 2008. Fluid Mechanics, fourth ed. Elsevier Academic Press.

Lamb, K.G., Lien, R.-C., Diamessis, P.J., 2019. Internal solitary waves and mixing. In: Encyclopedia of Ocean Sciences, third ed. Vol. 3, Academic Press, pp. 533–541.

Lamb, K.G., Nguyen, V.T., 2009. Calculating energy flux in internal solitary waves with an application to reflectance. J. Phys. Oceanogr. 39 (3), 559–580.

Lee, J., Wright, J.C., 2014. A block-tridiagonal solver with two-level parallelization for finite element-spectral codes. Comput. Phys. Comm. 185 (10), 2598–2608.

Lien, R.-C., D'Asaro, E.A., Henyey, F., Chang, M.-H., Tang, T.-Y., Yang, Y.-J., 2012. Trapped core formation within a shoaling nonlinear internal wave. J. Phys. Oceanogr. 42 (4), 511–525.

Lien, R.-C., Henyey, F., Ma, B., Yang, Y.J., 2014. Large-amplitude internal solitary waves observed in the northern South China Sea: properties and energetics. J. Phys. Oceanogr. 44 (4), 1095–1115.

Long, R.R., 1953. Some aspects of the flow of stratified fluids: I. a theoretical investigation. Tellus 5 (1), 42–58.

Magoulès, F., Roux, F.-X., Houzeaux, G., 2016. Parallel Scientific Computing. John Wiley & Sons.

Malm, J., Schlatter, P., Fischer, P.F., Henningson, D.S., 2013. Stabilization of the spectral element method in convection dominated flows by recovery of skew-symmetry. J. Sci. Comput. 57 (2), 254–277.

Mansfield, L., 1990. On the conjugate gradient solution of the Schur complement system obtained from domain decomposition. SIAM J. Numer. Anal. 27 (6), 1612–1620.

Manzanero, J., Rubio, G., Kopriva, D.A., Ferrer, E., Valero, E., 2020. An entropy–stable discontinuous Galerkin approximation for the incompressible Navier–Stokes equations with variable density and artificial compressibility. J. Comput. Phys. 408, 109241.

McPhee-Shaw, E.E., Kunze, E., Girton, J.B., 2021. Submarine canyon oxygen anomaly caused by mixing and boundary-interior exchange. Geophys. Res. Lett. 48 (10), e2021GL092995.

Moum, J., Farmer, D., Smyth, W., Armi, L., Vagle, S., 2003. Structure and generation of turbulence at interfaces strained by internal solitary waves propagating shoreward over the continental shelf. J. Phys. Oceanogr. 33 (10), 2093–2112.

Moura, R.C., Sherwin, S.J., Peiró, J., 2016. Eigensolution analysis of spectral/hp continuous Galerkin approximations to advection–diffusion problems: Insights into spectral vanishing viscosity. J. Comput. Phys. 307, 401–422.

Moxey, D., Cantwell, C.D., Bao, Y., Cassinelli, A., Castiglioni, G., Chun, S., Juda, E., Kazemi, E., Lackhove, K., Marcon, J., et al., 2020. Nektar++: Enhancing the capability and application of high-fidelity spectral/hp element methods. Comput. Phys. Comm. 249, 107110.

Nicolaides, R.A., 1987. Deflation of conjugate gradients with applications to boundary value problems. SIAM J. Numer. Anal. 24 (2), 355–365.

Offermans, N., Peplinski, A., Marin, O., Fischer, P., Schlatter, P., 2019. Towards adaptive mesh refinement for the spectral element solver Nek5000. In: Direct and Large-Eddy Simulation XI. Springer, pp. 9–15.

Özgökmen, T.M., Fischer, P.F., Duan, J., Iliescu, T., 2004. Entrainment in bottom gravity currents over complex topography from three-dimensional nonhydrostatic simulations. Geophys. Res. Lett. 31 (13).

Patera, A.T., 1984. A spectral element method for fluid dynamics: laminar flow in a channel expansion. J. Comput. Phys. 54 (3), 468–488.

Patera, A.T., 1986. Fast direct Poisson solvers for high-order finite element discretizations in rectangularly decomposable domains. J. Comput. Phys. 65 (2), 474–480.

Paul F. Fischer, J.W.L., Kerkemeier, S.G., 2008. Nek5000 web page. https://nek5000.mcs.anl.gov.

Peyret, R., 2002. Spectral Methods for Incompressible Viscous Flow. Vol. 148, Springer Science & Business Media.

Pozrikidis, C., 2001. A note on the regularization of the discrete Poisson–Neumann problem. J. Comput. Phys. 172 (2), 917–923.

Rivera-Rosario, G.A., Diamessis, P.J., Jenkins, J.T., 2017. Bed failure induced by internal solitary waves. J. Geophys. Res. Oceans 122 (7), 5468–5485.

Rivera-Rosario, G., Diamessis, P.J., Lien, R.-C., Lamb, K.G., Thomsen, G.N., 2020. Formation of recirculating cores in convectively breaking internal solitary waves of depression shoaling over gentle slopes in the South China Sea. J. Phys. Oceanogr. 50 (5), 1137–1157.

Rivera-Rosario, G., Diamessis, P.J., Lien, R.-C., Lamb, K.G., Thomsen, G.N., 2022. Three-dimensional perspective on a convective instability and transition to turbulence in an internal solitary wave of depression shoaling over gentle slopes. Environ. Fluid Mech. 1–21.

Saad, Y., Yeung, M., Erhel, J., Guyomarc'h, F., 2000. A deflated version of the conjugate gradient algorithm. SIAM J. Sci. Comput. 21 (5), 1909–1926.

Sakai, T., Diamessis, P.J., Jacobs, G.B., 2020a. Self-sustained instability, transition, and turbulence induced by a long separation bubble in the footprint of an internal solitary wave. I. Flow topology. Phys. Rev. Fluids 5 (10), 103801.

Sakai, T., Diamessis, P.J., Jacobs, G.B., 2020b. Self-sustained instability, transition, and turbulence induced by a long separation bubble in the footprint of an internal solitary wave. II. flow statistics. Phys. Rev. Fluids 5 (10), 103802.

Santilli, E., Scotti, A., 2015. The stratified ocean model with adaptive refinement (SOMAR). J. Comput. Phys. 291, 60–81.

Scotti, A., Mitran, S., 2008. An approximated method for the solution of elliptic problems in thin domains: Application to nonlinear internal waves. Ocean Model. 25 (3–4), 144–153.

Seal, S.K., Perumalla, K.S., Hirshman, S.P., 2013. Revisiting parallel cyclic reduction and parallel prefix-based algorithms for block tridiagonal systems of equations. J. Parallel Distrib. Comput. 73 (2), 273–280.

Steinmoeller, D., Stastna, M., Lamb, K., 2016. Discontinuous Galerkin methods for dispersive shallow water models in closed basins: Spurious eddies and their removal using curved boundary methods. Ocean Model. 107, 112–124.

Subich, C.J., Lamb, K.G., Stastna, M., 2013. Simulation of the Navier–Stokes equations in three dimensions with a spectral collocation method. Internat. J. Numer. Methods Fluids 73 (2), 103–129.

Tang, J.M., Nabben, R., Vuik, C., Erlangga, Y.A., 2009. Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods. J. Sci. Comput. 39 (3), 340–370.

Thorpe, S., Lemmin, U., 1999. Internal waves and temperature fronts on slopes. In: Annales Geophysicae. Vol. 17, Springer, pp. 1227–1234.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G.D., Roskies, R., Scott, J.R., Wilkins-Diehr, N., 2014. XSEDE: Accelerating scientific discovery. Comput. Sci. Eng. 16 (5), 62–74.

Tufo, H.M., Fischer, P.F., 2001. Fast parallel direct solvers for coarse grid problems. J. Parallel Distrib. Comput. 61 (2), 151–177.

Vandeven, H., 1991. Family of spectral filters for discontinuous problems. J. Sci. Comput. 6 (2), 159–192.

Vermolen, F., Vuik, K., Segal, G., 2004. Deflation in preconditioned conjugate gradient methods for finite element problems. In: Conjugate Gradient Algorithms and Finite Element Methods. Springer, pp. 103–129.

Vlasenko, V., Stashchuk, N., Hutter, K., 2005. Baroclinic Tides: Theoretical Modeling and Observational Evidence. Cambridge University Press.

Winters, K.B., de la Fuente, A., 2012. Modelling rotating stratified flows at laboratory-scale using spectrally-based DNS. Ocean Model. 49, 47–59.

Winters, K., MacKinnon, J., Mills, B., 2004. A spectral model for process studies of rotating, density-stratified flows. J. Atmos. Ocean. Technol. 21 (1), 69–94.

Yakovlev, S., Moxey, D., Kirby, R.M., Sherwin, S.J., 2016. To CG or to HDG: a comparative study in 3D. J. Sci. Comput. 67 (1), 192–220.