

A baroclinic discontinuous Galerkin finite element model for coastal flows

Tuomas Kärnä^{a,b,*}, Vincent Legat^{a,b}, Eric Deleersnijder^{a,c}

^a *Université catholique de Louvain,
Institute of Mechanics, Materials and Civil Engineering (IMMC),
Bte L4.05.02, Avenue Georges Lemaître 4,
B-1348 Louvain-la-Neuve, Belgium*

^b *Université catholique de Louvain
G. Lemaître Centre for Earth and Climate Research (TECLIM),
Bte L7.01.11, Chemin du Cyclotron 2,
B-1348 Louvain-la-Neuve, Belgium*

^c *Université catholique de Louvain
Earth and Life Institute (ELI),
G. Lemaître Centre for Earth and Climate Research (TECLIM),
Bte L7.01.11, Chemin du Cyclotron 2,
B-1348 Louvain-la-Neuve, Belgium*

Abstract

Numerical modelling of coastal flows is a challenging topic due to complex topography of the coastal zone, rapid flow dynamics and large density variations. Such phenomena are best simulated with unstructured grid models due to their highly flexible spatial discretisation. This article presents a three-dimensional discontinuous Galerkin finite element marine model. The spatial discretisation and explicit mode splitting time integration scheme are described. Free surface movement is accounted for by means of an arbitrary Lagrangian Eulerian (ALE) moving mesh method. Mass and volume are conserved. The conservation properties and baroclinic adjustment under gravity are tested with numerical benchmarks. Finally, the model is applied to the Rhine river plume in an idealised setting.

Keywords: Marine model, Finite element method, Discontinuous Galerkin, Baroclinic processes, River plume

1. Introduction

Although there has been interest in regional and coastal marine modelling for decades, simulating coastal flows and complete river-estuary-plume-shelf systems still poses several challenges. In coastal domains it is crucial to capture the complicated topographical and bathymetric features, which favours highly flexible unstructured mesh models. Coastal areas

*Corresponding author

Email address: tuomas.karna@uclouvain.be (Tuomas Kärnä)

often feature relatively rapid currents as the flow (e.g. tides) tends to amplify in shallow seas and funnel-like embayments. Due to riverine freshwater input, coastal flows are often significantly impacted by buoyancy and feature strong density gradients in estuaries and river plumes. Strong density gradients and rapid dynamics often lead to numerical stability issues that do not appear in deep ocean applications. Excessive numerical diffusion, inherent to some types of models, may smear out fronts and thus prevent models from capturing essential features of the flow (Hetland, 2005; Baptista et al., 2005; Burchard and Rennau, 2008). As the water density is governed by temperature and salinity, the quality of these fields has an impact on both accuracy and stability, stressing the importance of monotonous tracer advection schemes. In long term simulations strict tracer mass and water volume conservation is also important, yet not always satisfied.

The most commonly used marine models rely on structured grids (MITgcm, Marshall et al. 2004; ROMS, Shchepetkin and McWilliams 2005; POM, Mellor 2004; MOM, Griffies 2010; GETM, Burchard and Bolding 2002; TRIM, Casulli and Cheng 1992). Consequently the disadvantage of these models is the lack of flexibility in the computational grid and poor representation of the coastline. Embedded grids (Debreu and Blayo, 2008; Warner et al., 2010), curvilinear coordinates (Blumberg and Mellor, 1999) and composite grids (Warner et al., 2010) can be used to improve the quality of the spatial discretisation, but it is unclear whether such approaches can be used in highly complex domains (e.g. tidal river networks, de Brye et al. 2010, or the Great Barrier Reef, Lambrechts et al. 2008). Some structured grid models have been extended to unstructured grids by imposing orthogonality between the cells, e.g. TRIM and UnTRIM (Casulli and Walters, 2000) model classes. However, satisfying the orthogonality criterion in practical applications is very difficult, hindering the accuracy of such models (Zhang and Baptista, 2008).

Unstructured mesh models are usually based on finite volume (FV) (FV-COM, Chen et al. 2006; UnTRIM) or finite element (FE) (FESOM, Wang 2007; ICOM; SELFE, Zhang and Baptista 2008) method. Although some of these models are already widely used, unstructured mesh models generally are not as mature as structured grid ones. For example, finding an optimal spatial discretisation in terms of numerical stability, control of spurious modes, low numerical dissipation and computational efficiency still remains a challenge.

In this article we present a discontinuous Galerkin finite element (DG-FE) marine model and its application to baroclinic coastal flows. In general DG-FE can be seen as a hybrid formulation between FV and continuous FE formulations. In contrast to continuous FE, the DG method is well suited for solving advection dominated problems, but requires stabilisation for diffusion terms (Arnold et al., 2002). As the equations are solved element-by-element, DG methods are locally conservative by construction. Generally DG methods are also less diffusive than finite difference (FD) or FV models, especially with high order discretisations. Furthermore, completely discontinuous elements offer some attractive numerical properties, such as straightforward parallelisation of explicit schemes and extension to *hp*-adaptivity (Cockburn, 2003). Monotonic advection schemes can be achieved by means of slope limiters (Cockburn, 2003; Kuzmin, 2010; Aizinger, 2011).

Existing marine models use various time integration methods. The widely used mode

splitting technique (e.g. Griffies (2004)) relies on solving the fast propagating surface gravity waves in a depth averaged 2D framework, coupled to the slower 3D dynamics. In *split-explicit* schemes the 2D mode is treated explicitly with a high temporal resolution (Killworth et al., 1991). *Split-implicit* schemes (e.g. Dukowicz and Smith, 1994), on the other hand, solve the 2D mode implicitly and the same time step is used for both the 2D and the 3D mode. Mode splitting always introduces some discrepancy. Alternatively there exists semi-implicit models without mode splitting (TRIM, UnTRIM, SELFE). Some of these models, however, do not conserve volume or lack proper representation of the non-linear free surface dynamics. In this work, we rely on the split-explicit approach (Shchepetkin and McWilliams, 2005), which has shown to be robust in practice, and can be parallelised efficiently.

To correctly account for the free surface movement, the 3D mesh has to move in the vertical direction to match the instantaneous surface position. Here, an arbitrary Lagrangian Eulerian (ALE) formulation is used to represent the mesh movement. This formulation results in a generic framework, where volume and tracer mass conservation is guaranteed for any type of vertical mesh. In this work, a terrain following σ -coordinate-like mesh is used. Another important property is the consistency between discrete tracer and continuity equations, which ensures that a uniform tracer field is preserved (White et al., 2008b). This consistency criterion is sometimes referred to as constancy preserving property (Shchepetkin and McWilliams, 2005).

The article is organised as follows. Mathematical notation and the primitive equations are presented in Section 2. In the next section the ALE formulation and moving mesh algorithm are described. Section 4 presents the DG-FE discretisation. A matrix form of the discrete equations is given in Section 5. The time integration scheme is presented in Section 6. Finally Section 7 presents numerical results on conservation properties, gravitational adjustment flow and an application to a river plume simulation.

2. Governing equations

2.1. Domain

The three-dimensional computational domain is denoted by Ω . The position vector in Cartesian coordinates is $\mathbf{x} = [x, y, z]$, and the associated velocity components are u , v and w . The horizontal velocity vector is denoted by $\mathbf{u} = [u, v, 0]$. The domain is delimited by the bottom boundary Γ_b defined by the bathymetry $z = -h$, the closed lateral boundary Γ_c , open horizontal boundary Γ_o and the time dependent free surface Γ_s corresponding to the free surface height $z = \eta$. Due to the free surface movement, Ω is time dependent (Figure 1).

For the mathematical formulation it is convenient to define a time-independent reference domain Ω_0 that corresponds to $\eta = 0$. Coordinates associated with the reference domain are denoted by $\boldsymbol{\xi} = [x, y, z_0]$. The reference sea surface of Ω_0 is denoted by $\Gamma_{s,0}$ and the coordinates restricted on $\Gamma_{s,0}$ are $\mathbf{x}_h = [x, y, 0]$.

Traditionally the bathymetric features and free surface movement has been taken into account by introducing specific vertical coordinate systems, such as terrain following σ -coordinates (e.g. POM, Blumberg and Mellor, 1999) or isopycnal coordinates (e.g. Bleck

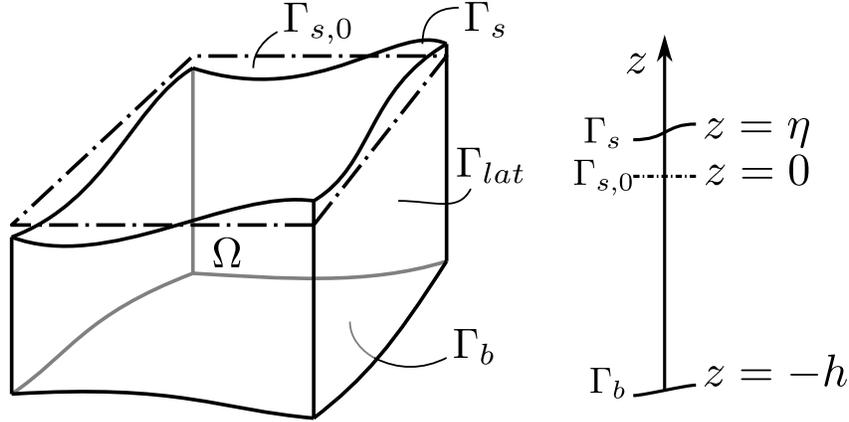


Figure 1: Possible configuration of the computational domain Ω at some time t . The reference surface $\Gamma_{s,0}$, marked with a dashed line, corresponds to $\eta = 0$.

and Smith, 1990). In equipotential z -coordinate models, the grid displacement is usually restricted to the top most cells. As each vertical coordinate system has its advantages and disadvantages, generalised coordinate transformations have been introduced (e.g. Kasahara, 1974). Currently hybrid vertical coordinates are widely used (Song and Haidvogel, 1994; Pietrzak et al., 2002), including dynamically adaptive vertical grids (e.g. Hofmeister et al., 2010).

Here, a generic Arbitrary Lagrangian Eulerian (ALE, Donea et al., 2004) formulation is adopted to facilitate the movement of the grid. In contrast to the aforementioned vertical coordinate systems, the equations are solved in the Cartesian (x, y, z) coordinates, and the grid deformation is taken into account by introducing a mesh velocity in the equations. Below the primitive equations are presented without the mesh velocity (i.e. for static domain), while the ALE formulation is described later in Section 3.

2.2. Momentum equation

In this work the 3D hydrostatic Boussinesq equations are considered. The horizontal momentum equation reads

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \nabla_h \cdot (\mathbf{u}\mathbf{u}) + \frac{\partial (w\mathbf{u})}{\partial z} + f\mathbf{e}_z \wedge \mathbf{u} + \frac{1}{\rho_0} \nabla_h p \\ = \nabla_h \cdot (\nu_h \nabla_h \mathbf{u}) + \frac{\partial}{\partial z} \left(\nu \frac{\partial \mathbf{u}}{\partial z} \right), \end{aligned} \quad (1)$$

where ∇_h is the horizontal gradient operator, f the Coriolis factor, \mathbf{e}_z is vertical unit vector, p is the pressure and ν_h and ν are the horizontal and vertical diffusivity, respectively.

Under the hydrostatic assumption the vertical momentum equation reduces to

$$\frac{\partial p}{\partial z} = -g\rho, \quad (2)$$

where g is the gravitational acceleration.

The water density ρ is computed by means of an equation of state (Jackett et al., 2006) as a function of the temperature, salinity and pressure. It is expressed as a sum of a constant reference density ρ_0 and a deviation $\rho' \ll \rho_0$:

$$\rho = \rho_0 + \rho', \quad (3)$$

Integrating (2) from η to z results in

$$p = g\rho_0(\eta - z) + g \int_z^\eta \rho' d\zeta + p_a,$$

where p_a is the atmospheric pressure acting on the sea surface. Defining the baroclinic head as

$$r = \frac{1}{\rho_0} \int_z^\eta \rho' d\zeta, \quad (4)$$

the pressure gradient term appearing in (1) can be written as

$$\frac{1}{\rho_0} \nabla_h p = g \nabla_h \eta + g \nabla_h r + \frac{1}{\rho_0} \nabla_h p_a. \quad (5)$$

In this article the effect of the atmospheric pressure is omitted as it is irrelevant for the presented applications.

2.3. Continuity and free surface equations

The continuity equation is given by

$$\nabla_h \cdot \mathbf{u} + \frac{\partial w}{\partial z} = 0, \quad (6)$$

from which the vertical velocity w is computed.

Integrating the continuity equation over the water column and taking into account the impermeability boundary conditions (defined later) on Γ_b and Γ_s , one obtains the free surface equation,

$$\frac{\partial \eta}{\partial t} + \nabla_h \cdot \int_{-h}^\eta \mathbf{u} dz = 0. \quad (7)$$

2.4. Tracer equations

The evolution of temperature T and salinity S are simulated with an advection-diffusion equation. Denoting the horizontal and vertical diffusivity by μ_h and μ , respectively, the equations read

$$\frac{\partial T}{\partial t} + \nabla_h \cdot (\mathbf{u}T) + \frac{\partial (wT)}{\partial z} = \nabla_h \cdot (\mu_h \nabla_h T) + \frac{\partial}{\partial z} \left(\mu \frac{\partial T}{\partial z} \right), \quad (8)$$

$$\frac{\partial S}{\partial t} + \nabla_h \cdot (\mathbf{u}S) + \frac{\partial (wS)}{\partial z} = \nabla_h \cdot (\mu_h \nabla_h S) + \frac{\partial}{\partial z} \left(\mu \frac{\partial S}{\partial z} \right). \quad (9)$$

2.5. Turbulence model

To close the aforementioned set of equations, the vertical eddy viscosity ν and diffusivity μ must be determined in terms of the flow state. In this work, the $k - \varepsilon$ turbulence closure model (Rodi, 1987) is used with the stability functions of Canuto et al. (2001) (Model A). The evolution of turbulent quantities is solved with GOTM¹ (General Ocean Turbulence Model, Burchard et al., 1999) turbulence model library that has been coupled to the present finite element model (Kärnä et al., 2012).

2.6. Boundary conditions

On the surface and bottom boundaries the conventional impermeability boundary conditions are prescribed

$$w + \mathbf{u} \cdot \nabla_h h = 0, \quad \mathbf{x} \in \Gamma_b \quad (10)$$

$$w - \frac{\partial \eta}{\partial t} - \mathbf{u} \cdot \nabla_h \eta = 0, \quad \mathbf{x} \in \Gamma_s. \quad (11)$$

Further, a slip condition is enforced to take into account the bottom and surface stresses,

$$\nu \frac{\partial \mathbf{u}}{\partial z} = \frac{\boldsymbol{\tau}_b}{\rho_0}, \quad \mathbf{x} \in \Gamma_b \quad (12)$$

$$\nu \frac{\partial \mathbf{u}}{\partial z} = \frac{\boldsymbol{\tau}_s}{\rho_0}, \quad \mathbf{x} \in \Gamma_s. \quad (13)$$

The bottom stress is given by

$$\frac{\boldsymbol{\tau}_b}{\rho_0} = C_d \|\mathbf{u}_{\text{bf}}\| \mathbf{u}_{\text{bf}}, \quad (14)$$

$$C_d = \left[\frac{\kappa}{\ln((z_{\text{bf}} + h)/\delta_0)} \right]^2,$$

$$\mathbf{u}_{\text{bf}} = \mathbf{u}(x, y, z_{\text{bf}}),$$

where κ is the von Karman constant, δ_0 is the bottom roughness length and z_{bf} denotes the vertical coordinate where the bottom velocity \mathbf{u}_{bf} is defined.

On the closed lateral boundaries Γ_c , assumed to be strictly vertical, impermeability is imposed

$$\mathbf{u} \cdot \mathbf{n}_h = 0, \quad (15)$$

where $\mathbf{n}_h = [n_x, n_y, 0]$ is the horizontal unit normal vector.

¹www.gotm.net

2.7. Depth averaged equations

For computational efficiency, the fast propagating surface gravity waves are simulated with two-dimensional depth averaged equations. Introducing the depth averaged horizontal velocity $\bar{\mathbf{u}}$ and the decomposition $\mathbf{u} = \bar{\mathbf{u}} + \tilde{\mathbf{u}}$, equation (7) can equivalently be written as

$$\frac{\partial \eta}{\partial t} + \nabla_h \cdot (H\bar{\mathbf{u}}) = 0, \quad (16)$$

where $H = \eta + h$ is the total depth.

Depth averaging (1), the momentum equation becomes (see e.g. (White et al., 2008a))

$$\begin{aligned} \frac{\partial \bar{\mathbf{u}}}{\partial t} + \bar{\mathbf{u}} \cdot \nabla_h \bar{\mathbf{u}} + f \mathbf{e}_z \wedge \bar{\mathbf{u}} \\ + g \nabla_h \eta + g \frac{1}{H} \int_{-h}^{\eta} \nabla_h r dz = \bar{\mathbf{A}}_H + \bar{\mathbf{D}}_H + \frac{\boldsymbol{\tau}_b + \boldsymbol{\tau}_s}{\rho_0 H}, \end{aligned} \quad (17)$$

with

$$\begin{aligned} \bar{\mathbf{D}}_H &= \frac{1}{H} \int_{-h}^{\eta} \nabla_h \cdot (\nu_h \nabla_h \mathbf{u}) dz \\ \bar{\mathbf{A}}_H &= -\frac{1}{H} \frac{\partial}{\partial x} \int_{-h}^{\eta} \tilde{u} \tilde{u} dz - \frac{1}{H} \frac{\partial}{\partial y} \int_{-h}^{\eta} \tilde{v} \tilde{u} dz. \end{aligned}$$

Above $\bar{\mathbf{A}}_H$ represents the advection of $\tilde{\mathbf{u}}$, while $\bar{\mathbf{D}}_H$ stands for the horizontal viscosity of momentum. For the sake of simplicity the latter is parametrised by the conventional two-dimensional diffusion operator

$$\bar{\mathbf{D}}_H = \frac{1}{H} \nabla_h \cdot (H \nu_h \nabla_h \bar{\mathbf{u}}). \quad (18)$$

This parametrisation is justified given the small contribution and large relative uncertainty of the viscosity term. Moreover, this form is purely dissipative, which is not necessarily the case for the exact operator in (17).

3. ALE formulation

The 3D mesh adapts to the instantaneous position of the free surface. Consequently, the top boundary Γ_s will coincide to the $z = \eta$ surface while the bottom Γ_b remains static. Following Farhat et al. (2001) and White et al. (2008b), we define a mapping from the static domain Ω_0 to the time dependent domain Ω :

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}, t) = [x, y, z(x, y, z_0, t)].$$

The mapping is assumed to be invertible with the Jacobian $\mathbf{J} = \partial \mathbf{x} / \partial \boldsymbol{\xi}$ and $J = \det(\mathbf{J}) = \partial z / \partial z_0 > 0$.

The primitive equations are expressed in such a way that they can be solved in the moving domain Ω , given the instantaneous domain (or mesh) velocity

$$w_m = \frac{\partial z}{\partial t} \Big|_{\boldsymbol{\xi}}.$$

The *conservative* ALE formulation of the tracer equation (8) reads (Formaggia and Nobile, 2004)

$$\frac{1}{J} \frac{\partial(JT)}{\partial t} \Big|_{\boldsymbol{\xi}} + \nabla_h \cdot (\mathbf{u}T) + \frac{\partial((w - w_m)T)}{\partial z} = \mathbf{D}_H, \quad (19)$$

where \mathbf{D}_H denotes the right hand side of (8).

Consequently, the conservative ALE formulation implies a change in the time derivative and modification in the vertical advective velocity. As the time derivative is taken with respect to the static coordinates $\boldsymbol{\xi}$, this formulation is useful in the FE method. Taking an arbitrary test function φ , and noting that $Jd\boldsymbol{\xi} = d\mathbf{x}$, the weak formulation of the time derivative can be expressed as

$$\int_{\Omega} \frac{1}{J} \frac{\partial(JT)}{\partial t} \Big|_{\boldsymbol{\xi}} \varphi d\mathbf{x} = \frac{d}{dt} \left(\int_{\Omega} T \varphi d\mathbf{x} \right) \Big|_{\boldsymbol{\xi}}.$$

This approach is applied to all the 3D prognostic variables (T, S, u and v).

Alternatively, using a *non-conservative* ALE form, the tracer equation reads

$$\frac{\partial T}{\partial t} \Big|_{\boldsymbol{\xi}} + T \frac{\partial w_m}{\partial z} + \nabla_h \cdot (\mathbf{u}T) + \frac{\partial((w - w_m)T)}{\partial z} = \mathbf{D}_H, \quad (20)$$

where the additional second term accounts for the volumetric change. A numerical scheme based on this formulation does not conserve mass, but can be used in cases where $\partial T / \partial t$ is computed in a static geometry.

3.1. Moving mesh algorithm

In this work a σ -coordinate-like moving mesh algorithm is adopted, which distributes the vertical perturbation linearly over the water column. The presented methodology can be easily generalised for other grid types as well. Using the vertical coordinates $z_0 \in [-h, 0]$ of the static domain Ω_0 , the time dependent vertical coordinates are then obtained as

$$z = z_0 + \eta \frac{z_0 + h}{h} \quad (21)$$

implying $z \in [-h, \eta]$ and $J = (\eta + h)/h$. The vertical mesh velocity w_m can be deduced from the impermeability boundary conditions. At the surface, $w_m = \partial \eta / \partial t$ which can be computed as

$$w_m \Big|_{\Gamma_s} = w - \nabla_h \eta \cdot \mathbf{u}. \quad (22)$$

At the bottom $w_m = 0$ because Γ_b is static. In the interior w_m then becomes:

$$w_m = w_m|_{\Gamma_s} \frac{z_0 + h}{h}, \quad \boldsymbol{\xi} \in \Omega_0. \quad (23)$$

Expressing w_m in the time dependent coordinates yields

$$w_m = w_m|_{\Gamma_s} \frac{z + h}{\eta + h}, \quad \boldsymbol{x} \in \Omega. \quad (24)$$

The non conservative ALE formulation also requires $\partial w_m / \partial z$ which is given by

$$\frac{\partial w_m}{\partial z} = w_m|_{\Gamma_s} \frac{1}{\eta + h}, \quad \boldsymbol{x} \in \Omega. \quad (25)$$

4. Finite element discretisation

4.1. Function spaces

The sea surface $\Gamma_{s,0}$ is divided into a set of triangles \mathcal{T} . A piecewise discontinuous polynomial function space \mathbb{W} is defined on \mathcal{T} so that each function in \mathbb{W} is a linear polynomial inside the triangles $Q \in \mathcal{T}$, and discontinuous on the interfaces $e = Q \cap Q'$. The space \mathbb{W} is defined by means of Lagrangian basis functions $\varphi_i : \mathbb{R}^2 \rightarrow \mathbb{R}$, $i = 1, \dots, N_T$, where N_T is the total number of nodes in the triangulation \mathcal{T} . The basis functions φ_i are non-zero in exactly one element and attain unity only in a single node $\boldsymbol{x}_{h,i} = [x_i, y_i, 0]$, being zero in all the others, $\varphi_i(\boldsymbol{x}_{h,j}) = 0$, $\forall i \neq j$.

The triangular surface mesh is extruded in vertical direction towards the bottom Γ_b resulting in columns of triangular prisms. Each column is further divided equally into N_σ prisms, thus forming a terrain following grid similar to conventional σ -coordinates².

The set of N_P prisms is denoted by \mathcal{P} . A piecewise polynomial function space \mathbb{V} is defined on \mathcal{P} by a set of Lagrangian basis functions $\psi_i : \mathbb{R}^3 \rightarrow \mathbb{R}$. Each ψ_i is a linear polynomial both in the horizontal (x, y) and vertical (z) direction, and non-zero only in a single prism K . Also here the Lagrangian property holds on the nodes \boldsymbol{x}_i , $i = 1, \dots, N_P$: $\psi_i(\boldsymbol{x}_i) = 1$, $\psi_i(\boldsymbol{x}_j) = 0$, $\forall i \neq j$.

In the present DG discretisation all the scalar fields (including u and v) belong to the same function space \mathbb{V} (in the 3D mesh) or \mathbb{W} (in the 2D mesh). Therefore all the fields share the same basis functions and same nodes. In this article first order basis functions are used, where the nodes correspond to the vertices of the prisms (in 3D) or triangles (in 2D). Consequently there are 6 degrees of freedom associated to each prism (3 for each triangle), which is substantially more than in a continuous Galerkin discretisation.

²Note that the equations are still solved in Cartesian coordinates, in contrast to σ -coordinate models.

4.2. Interfaces

In the set of prisms, all interfaces associated to an element $K \in \mathcal{P}$ are defined as

$$\mathcal{I}(K) := \{K \cap K' | K' \in \mathcal{P}\}. \quad (26)$$

For each interface $I \in \mathcal{I}(K)$, the unit normal vector is $\mathbf{n} = [n_x, n_y, n_z]$ chosen to point from K to K' . The lateral and horizontal interfaces, respectively, are denoted by

$$\mathcal{I}_{lat}(K) := \{K \cap K' | K' \in \mathcal{P}, \mathbf{n} \cdot \mathbf{e}_z = 0\}, \quad (27)$$

$$\mathcal{I}_h(K) := \{K \cap K' | K' \in \mathcal{P}, \mathbf{n} \cdot \mathbf{e}_z \neq 0\}. \quad (28)$$

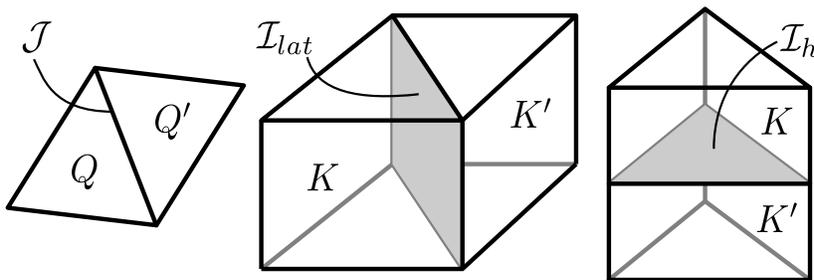


Figure 2: Interfaces for triangular surface mesh (left) and prismatic 3D mesh (right).

In the case of a prismatic mesh, $\mathcal{I}_{lat}(K)$ correspond to the vertical quadrilateral faces, and $\mathcal{I}_h(K)$ to the top/bottom triangles of a prism K (Figure 2). It is noteworthy that $\mathcal{I}_{lat}(K)$ are always vertical, but due to the movement, $\mathcal{I}_h(K)$ are not strictly horizontal (i.e. n_x and n_y are not necessarily zero).

On the triangulation \mathcal{T} the interfaces are defined similarly,

$$\mathcal{J}(Q) := \{Q \cap Q' | Q' \in \mathcal{T}\}. \quad (29)$$

On an interface, all variables have two different values associated to the elements K and K' . In DG method it is essential to compute the interface fluxes carefully in order to ensure numerical stability of the scheme. Possible methods include (approximate) Riemann solvers for advective fluxes (Toro, 2009) and interior penalty methods for diffusive fluxes (Arnold et al., 2002). Denoting the variables on the “left” (corresponding to K) by subscript L and on the “right” (corresponding to K') by subscript R , arithmetic mean, difference and maximum operators are then defined as

$$\begin{aligned} \{\bullet\} &= \frac{\bullet_L + \bullet_R}{2}, \\ [[\bullet]] &= \frac{\bullet_L - \bullet_R}{2}, \\ [\bullet] &= \max(\bullet_L, \bullet_R), \end{aligned}$$

respectively. This notation is used for both 2D and 3D elements.

In the derivation of the weak form, the following shorthand notation is used for spatial integrals (dA and dS denote the infinitesimal area and line elements on \mathcal{I} and \mathcal{J} , respectively):

$$\begin{aligned}\langle \bullet \rangle_K &= \int_K \bullet d\mathbf{x}, & K \in \mathcal{P} \\ \langle\langle \bullet \rangle\rangle_{\mathcal{I}} &= \int_{\mathcal{I}} \bullet dA, & \mathcal{I} \in \mathcal{I}(K) \\ \langle\langle \bullet \rangle\rangle_Q &= \int_Q \bullet d\mathbf{x}_h, & Q \in \mathcal{T} \\ \langle\langle\langle \bullet \rangle\rangle\rangle_{\mathcal{J}} &= \int_{\mathcal{J}} \bullet dS, & \mathcal{J} \in \mathcal{J}(Q)\end{aligned}$$

In practice, the integrals are evaluated with numerical quadrature rules. In 2D, a Hammer quadrature (?) of order $2o+1$ is used on the triangles. In 3D, a combination of $2o+1$ order Hammer quadrature (in the horizontal) and Gauss-Legendre quadrature (in the vertical direction) is used.

4.3. Depth averaged equations

The weak (or variational) formulation for the depth averaged equations is obtained by multiplying (16) by a test function $\varphi \in \mathbb{W}$ and (17) by $\boldsymbol{\varphi} \in \mathbb{W} \times \mathbb{W}$ and integrating over $\Gamma_{s,0}$. As the basis functions are non-zero only within a single element, the weak formulation can be written separately for each element $Q \in \mathcal{T}$.

$$\langle\langle \frac{\partial \eta}{\partial t} \varphi \rangle\rangle_Q + \underbrace{\langle\langle \nabla_h \cdot ((\eta + h)\bar{\mathbf{u}}) \varphi \rangle\rangle_Q}_{\varepsilon_u} = 0, \quad (30)$$

$$\begin{aligned}\langle\langle \frac{\partial \bar{\mathbf{u}}}{\partial t} \cdot \boldsymbol{\varphi} \rangle\rangle_Q &+ \underbrace{\langle\langle \bar{\mathbf{u}} \cdot \nabla_h \bar{\mathbf{u}} \cdot \boldsymbol{\varphi} \rangle\rangle_Q}_{\bar{u}_u} + \langle\langle f \mathbf{e}_z \wedge \bar{\mathbf{u}} \cdot \boldsymbol{\varphi} \rangle\rangle_Q \\ &+ \underbrace{\langle\langle g \nabla_h \eta \cdot \boldsymbol{\varphi} \rangle\rangle_Q}_{\bar{u}_\eta} + \langle\langle g \frac{1}{H} \int_{-h}^{\eta} \nabla_h r \cdot \boldsymbol{\varphi} \rangle\rangle_Q \\ &= \langle\langle \bar{\mathbf{A}}_H \cdot \boldsymbol{\varphi} \rangle\rangle_Q + \underbrace{\langle\langle \frac{1}{H} \nabla_h \cdot (H \nu_h \nabla_h \bar{\mathbf{u}}) \cdot \boldsymbol{\varphi} \rangle\rangle_Q}_{\bar{u}_v^h} \\ &+ \langle\langle \frac{\boldsymbol{\tau}_b}{\rho_0 H} \cdot \boldsymbol{\varphi} \rangle\rangle_Q + \langle\langle \frac{\boldsymbol{\tau}_s}{\rho_0 H} \cdot \boldsymbol{\varphi} \rangle\rangle_Q.\end{aligned} \quad (31)$$

Note that these equations are purely local in Q , as there is no dependency on the neighbouring elements Q' . Inter-element fluxes arise when the underbraced terms are replaced by the following terms, obtained by integrating by parts:

Divergence of $H\bar{\mathbf{u}}$:

$$\mathcal{E}_u = -\left\langle\left\langle (\eta + h)\bar{\mathbf{u}} \cdot \nabla_h \varphi \right\rangle\right\rangle_Q + \sum_{\mathcal{J}(Q)} \left\langle\left\langle (\eta^* + h)(\bar{\mathbf{u}}^* \cdot \mathbf{n}_h) \varphi \right\rangle\right\rangle_{\mathcal{J}} \quad (32)$$

Horizontal advection of momentum:

$$\bar{U}_u = -\left\langle\left\langle \nabla_h \cdot (\bar{\mathbf{u}} \varphi) \cdot \bar{\mathbf{u}} \right\rangle\right\rangle_Q + \sum_{\mathcal{J}(Q)} \left\langle\left\langle (\bar{\mathbf{u}}^* \cdot \mathbf{n}_h) \bar{\mathbf{u}}^* \cdot \varphi \right\rangle\right\rangle_{\mathcal{J}} \quad (33)$$

External pressure gradient:

$$\bar{U}_\eta = -\left\langle\left\langle g\eta \nabla_h \cdot \varphi \right\rangle\right\rangle_Q + \sum_{\mathcal{J}(Q)} \left\langle\left\langle g\eta^* \varphi \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{J}} \quad (34)$$

Diffusion of momentum³:

$$\begin{aligned} \bar{U}_\nu^h = & -\left\langle\left\langle \nu_h (\nabla_h \bar{\mathbf{u}}) : (\nabla_h \varphi)^T \right\rangle\right\rangle_Q + \left\langle\left\langle \nu_h \frac{1}{H} (\nabla_h H) \cdot (\nabla_h \bar{\mathbf{u}}) \cdot \varphi \right\rangle\right\rangle_Q \\ & + \sum_{\mathcal{J}(Q)} \left\langle\left\langle (\nu_h \nabla_h \bar{\mathbf{u}})^* \cdot \mathbf{n}_h \cdot \varphi \right\rangle\right\rangle_{\mathcal{J}} \end{aligned} \quad (35)$$

In the interface terms of (32)-(35), the variables marked with an asterisk are unknown, and must be deduced from the state variables on both sides of the interface. The variables η^* and $\bar{\mathbf{u}}^*$ are related to the propagation of surface gravity waves and are solved with an approximate Riemann solver described below. The flux $(\nu_h \nabla_h \bar{\mathbf{u}})^*$, on the other hand, is related to the horizontal diffusion operator, described in Section 4.3.2.

4.3.1. Riemann solver for surface gravity waves

A Riemann problem consist of solving a hyperbolic conservation law subject to a discontinuous initial condition. Therefore Riemann solvers (Toro, 2009) are a natural choice for computing the interface fluxes in FV and DG-FE methods.

If advection of momentum is negligible and $\eta \ll h$, surface gravity waves can be modelled with the linear shallow water equations:

$$\begin{aligned} \frac{\partial \eta}{\partial t} + \nabla_h (h\bar{\mathbf{u}}) &= 0, \\ \frac{\partial \bar{\mathbf{u}}}{\partial t} + g \nabla_h \eta &= 0 \end{aligned}$$

³The colon denotes the Frobenius inner product: $\mathbf{A} : \mathbf{B} = \sum_{i,j} A_{ij} B_{ij}$

Denoting the wave celerity by $c = \sqrt{gh}$, the well-known Riemann solution to these equations is (LeVeque, 2002; Comblen et al., 2010)

$$\eta^* = \{\eta\} + \frac{h}{c}[[\bar{\mathbf{u}}]], \quad (36)$$

$$\bar{\mathbf{u}}^* = \{\bar{\mathbf{u}}\} + \frac{c}{h}[[\eta]]. \quad (37)$$

Combined with the DG method, the jump operators on the right hand side introduce sufficient dissipation to guarantee numerical stability.

If the advection is not negligible, a more complex non-linear Riemann solver corresponding to the full shallow water equations should be used. In this work, non-linear solver presented in Comblen et al. (2010) is used for the 2D mode, while the above linear solution is used in the 3D mode.

4.3.2. Interior penalty stabilisation

In DG-FE diffusion operators require additional stabilisation. In this work the Incomplete Interior Penalty Method (IIPM, Rivière, 2008) is adopted. In IIPM the interface flux is replaced by a mean flux $\{\nu_h \nabla_h \bar{\mathbf{u}}\}$ and an additional penalty term $\sigma \nu_h [[\bar{\mathbf{u}}]]$ is introduced, with the penalty factor defined as (Shahbazi, 2005)

$$\sigma_d = \frac{(o+1)(o+d)}{d} \frac{N_0}{2L_{min}}, \quad (38)$$

where d stands for the dimension of the problem and N_0 is the number of neighbours of an element ($N_0 = 3$ for triangles). L_{min} approximates the effective element length scale normal to the interface. In 2D, $L_{min} = \min(|Q|, |Q'|)/|I|$, where $|Q|$ is the element area and $|I|$, $I = Q \cap Q'$ the interface length.

Thus the whole diffusion operator of the 2D momentum equation (35) becomes

$$\begin{aligned} \bar{U}_\nu^h = & - \left\langle\left\langle \nu_h (\nabla_h \bar{\mathbf{u}}) : (\nabla_h \boldsymbol{\varphi})^T \right\rangle\right\rangle_Q + \left\langle\left\langle \nu_h \frac{1}{H} (\nabla_h H) \cdot (\nabla_h \bar{\mathbf{u}}) \cdot \boldsymbol{\varphi} \right\rangle\right\rangle_Q \\ & + \sum_{\mathcal{J}(Q)} \left\langle\left\langle \left\{ \nu_h \nabla_h \bar{\mathbf{u}} \right\} \cdot \mathbf{n}_h \cdot \boldsymbol{\varphi} \right\rangle\right\rangle_{\mathcal{J}} + \sum_{\mathcal{J}(Q)} \left\langle\left\langle \sigma_2 [\nu_h] [[\bar{\mathbf{u}}]] \right\rangle\right\rangle_{\mathcal{J}}. \end{aligned} \quad (39)$$

4.4. Momentum equation

The weak formulation of the horizontal momentum equation is obtained by multiplying (1) by a test function $\boldsymbol{\psi} \in \mathbb{V} \times \mathbb{V}$, integrating over the time dependent domain Ω . Here the equations are developed for the conservative ALE formulation (19) for brevity, as the non-conservative formulation (20) leads to very similar equations. Element-wise weak formulation then reads:

$$\begin{aligned} \frac{d}{dt} \left\langle \mathbf{u} \cdot \boldsymbol{\psi} \right\rangle_K + \underbrace{\left\langle \nabla_h \cdot (\mathbf{u}\mathbf{u}) \cdot \boldsymbol{\psi} \right\rangle_K}_{\mathcal{U}_u} + \underbrace{\left\langle \frac{\partial ((w - w_m)\mathbf{u})}{\partial z} \cdot \boldsymbol{\psi} \right\rangle_K}_{\mathcal{U}_w} + \left\langle f \mathbf{e}_z \wedge \mathbf{u} \cdot \boldsymbol{\psi} \right\rangle_K + \\ \underbrace{\left\langle g \nabla_h \eta \cdot \boldsymbol{\psi} \right\rangle_K}_{\mathcal{U}_\eta} + \left\langle g \nabla_h r \cdot \boldsymbol{\psi} \right\rangle_K = \underbrace{\left\langle \nabla_h \cdot (\nu_h \nabla_h \mathbf{u}) \cdot \boldsymbol{\psi} \right\rangle_K}_{\mathcal{U}_\nu^h} + \underbrace{\left\langle \frac{\partial}{\partial z} \left(\nu \frac{\partial \mathbf{u}}{\partial z} \right) \cdot \boldsymbol{\psi} \right\rangle_K}_{\mathcal{U}_\nu^v}. \end{aligned} \quad (40)$$

For a complete DG weak formulation, underbraced terms are replaced by the following terms. Horizontal advection

$$\begin{aligned} \mathcal{U}_u = & -\left\langle \nabla_h \psi : \mathbf{u}\mathbf{u} \right\rangle_K + \sum_{\mathcal{I}_{lat}(K)} \left\langle\left\langle \psi \cdot \mathbf{u}^* \mathbf{u}^* \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{I}_{lat}} \\ & + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \psi \cdot \mathbf{u}^{uw} (\mathbf{u}^d \cdot \mathbf{n}_h) \right\rangle\right\rangle_{\mathcal{I}_h}, \end{aligned} \quad (41)$$

vertical advection

$$\mathcal{U}_w = -\left\langle (w - w_m) \mathbf{u} \cdot \frac{\partial \psi}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle (w^d - w_m) \mathbf{u}^{uw} \cdot \psi n_z \right\rangle\right\rangle_{\mathcal{I}_h}, \quad (42)$$

and external pressure gradient

$$\mathcal{U}_\eta = -\left\langle g\eta \nabla_h \cdot \psi \right\rangle_K + \sum_{\mathcal{I}_{lat}(K)} \left\langle\left\langle g\eta^* \psi \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{I}_{lat}} + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle g\eta \psi \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{I}_h}. \quad (43)$$

In the \mathcal{I}_h integrals \mathbf{u}^d, w^d stand for values in the lower element, while \mathbf{u}^{uw} refers to the conventional upwind value. Note that η and w_m are unique on \mathcal{I}_h .

The horizontal diffusion operator becomes

$$\mathcal{U}_\nu^h = -\left\langle \nu_h (\nabla_h \psi) : (\nabla_h \mathbf{u})^T \right\rangle_K + \sum_{\mathcal{I}} \left\langle\left\langle \psi \cdot \{ \nu_h \nabla_h \mathbf{u} \} \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{I}}, \quad (44)$$

and the vertical diffusion operator

$$\mathcal{U}_\nu^v = -\left\langle \nu \frac{\partial \psi}{\partial z} \cdot \frac{\partial \mathbf{u}}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \psi \cdot \left\{ \nu \frac{\partial \mathbf{u}}{\partial z} \right\} n_z \right\rangle\right\rangle_{\mathcal{I}_h}. \quad (45)$$

4.4.1. Riemann solver for the 3D mode

Similarly to the 2D mode, in (41) and (43) the Riemann values η^* and \mathbf{u}^* are needed. In this work η^* is computed with the linear Riemann solver (36), while \mathbf{u}^* is computed as

$$\mathbf{u}^* = \{\mathbf{u}\} + \frac{c}{h} [[\eta]]. \quad (46)$$

The above formulation is consistent with the linear Riemann solver, because the depth average of (46) reduces to (37). Here we are using the linear solver as it is not possible to derive a similar formula for \mathbf{u}^* with the full non-linear 2D Riemann solver. As the surface gravity waves are essentially solved in the 2D mode and imposed in the 3D mode, this is not a major drawback.

4.4.2. Interior Penalty stabilisation for the 3D mode

Also in 3D the diffusion operators are stabilized with IIPM. An additional penalty term is added to the right hand side (Ern et al., 2009),

$$\begin{aligned} & \sum_{\mathcal{I}(K)} \left\langle \left\langle \sigma_3 (\mathbf{n} \cdot \mathbf{D}_\nu \cdot \mathbf{n}) \boldsymbol{\psi} \cdot [[\mathbf{u}]] \right\rangle \right\rangle_{\mathcal{I}} = \\ & \sum_{\mathcal{I}(K)} \left\langle \left\langle \sigma_3 \nu_h (n_x^2 + n_y^2) \boldsymbol{\psi} \cdot [[\mathbf{u}]] \right\rangle \right\rangle_{\mathcal{I}} + \sum_{\mathcal{I}_h(K)} \left\langle \left\langle \sigma_3 \nu n_z^2 \boldsymbol{\psi} \cdot [[\mathbf{u}]] \right\rangle \right\rangle_{\mathcal{I}_h}, \end{aligned} \quad (47)$$

where $\mathbf{D}_\nu = \text{diag}(\nu_h, \nu_h, \nu)$ is the diffusivity tensor and the penalty factor σ_3 is computed with (38). Here $L_{min} = \min(|K|, |K'|)/|I|$, with the element volume $|K|$ and interface area $|I|$. Note that (47) is defined on $\mathcal{I} = \mathcal{I}_h \cup \mathcal{I}_{lat}$. Due to the fact that the \mathcal{I}_h interfaces are not strictly horizontal the whole diffusivity tensor has to be taken into account.

The final diffusion operators read

$$\begin{aligned} \mathcal{U}_\nu^h &= - \left\langle \nu_h (\nabla_h \boldsymbol{\psi}) : (\nabla_h \mathbf{u})^T \right\rangle_K + \sum_{\mathcal{I}(K)} \left\langle \left\langle \boldsymbol{\psi} \cdot \{ \nu_h \nabla_h \mathbf{u} \} \cdot \mathbf{n}_h \right\rangle \right\rangle_{\mathcal{I}} \\ &+ \sum_{\mathcal{I}(K)} \left\langle \left\langle \sigma_3 \nu_h (n_x^2 + n_y^2) \boldsymbol{\psi} \cdot [[\mathbf{u}]] \right\rangle \right\rangle_{\mathcal{I}}, \end{aligned} \quad (48)$$

$$\begin{aligned} \mathcal{U}_\nu^v &= - \left\langle \nu \frac{\partial \boldsymbol{\psi}}{\partial z} \cdot \frac{\partial \mathbf{u}}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle \left\langle \boldsymbol{\psi} \cdot \left\{ \nu \frac{\partial \mathbf{u}}{\partial z} \right\} n_z \right\rangle \right\rangle_{\mathcal{I}_h} \\ &+ \sum_{\mathcal{I}_h(K)} \left\langle \left\langle \sigma_3 \nu n_z^2 \boldsymbol{\psi} \cdot [[\mathbf{u}]] \right\rangle \right\rangle_{\mathcal{I}_h}. \end{aligned} \quad (49)$$

4.5. Tracer Equations

The weak formulations of the temperature and salinity equations are obtained by multiplying (8) and (9) by a test function $\psi \in \mathbb{V}$, integrating over Ω and adopting the ALE form. As (8) and (9) are equivalent only the T equation is developed for brevity. Element-wise weak formulation reads:

$$\begin{aligned} \frac{d}{dt} \left\langle T \psi \right\rangle_K &+ \underbrace{\left\langle \nabla_h \cdot (\mathbf{u} T) \psi \right\rangle_K}_{c_u} + \underbrace{\left\langle \frac{\partial ((w - w_m) T)}{\partial z} \psi \right\rangle_K}_{c_w} = \\ &\underbrace{\left\langle \nabla_h \cdot (\mu_h \nabla_h T) \psi \right\rangle_K}_{c_\mu^h} + \underbrace{\left\langle \frac{\partial}{\partial z} \left(\mu \frac{\partial T}{\partial z} \right) \psi \right\rangle_K}_{c_\mu^v}. \end{aligned} \quad (50)$$

After integration by parts, the advection terms become

$$\begin{aligned} \mathcal{C}_u &= -\left\langle T \mathbf{u} \cdot \nabla_h \psi \right\rangle_K + \sum_{\mathcal{I}_{lat}(K)} \left\langle\left\langle T(\mathbf{u}^* \cdot \mathbf{n}_h) \psi \right\rangle\right\rangle_{\mathcal{I}_{lat}} \\ &+ \sum_{\mathcal{I}_h(K)} \left\langle\left\langle T^{uw} \mathbf{u}^d \cdot \mathbf{n}_h \psi \right\rangle\right\rangle_{\mathcal{I}_h}, \end{aligned} \quad (51)$$

$$\mathcal{C}_w = -\left\langle (w - w_m) T \frac{\partial \psi}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle (w^d - w_m) T^{uw} \psi n_z \right\rangle\right\rangle_{\mathcal{I}_h}, \quad (52)$$

where \mathbf{u}^d , w^d are the velocity in the lower element and T^{uw} stands for upwind value. For consistency the same Riemann value \mathbf{u}^* (46) must be used as in the momentum equation. The diffusion terms are treated similarly as before

$$\mathcal{C}_\mu^h = -\left\langle \mu_h (\nabla_h \psi) \cdot (\nabla_h T) \right\rangle_K + \sum_{\mathcal{I}(K)} \left\langle\left\langle \{\mu_h \nabla_h T\} \cdot \mathbf{n}_h \psi \right\rangle\right\rangle_{\mathcal{I}}, \quad (53)$$

$$\mathcal{C}_\mu^v = -\left\langle \mu \frac{\partial T}{\partial z} \frac{\partial \psi}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \left\{ \mu \frac{\partial T}{\partial z} \right\} n_z \psi \right\rangle\right\rangle_{\mathcal{I}_h}. \quad (54)$$

Defining the diffusivity tensor $\mathbf{D}_\mu = \text{diag}(\mu_h, \mu_h, \mu)$, the additional interior penalty term is

$$\begin{aligned} \sum_{\mathcal{I}(K)} \left\langle\left\langle \sigma_3 (\mathbf{n} \cdot \mathbf{D}_\mu \cdot \mathbf{n}) \psi [[T]] \right\rangle\right\rangle_{\mathcal{I}} &= \sum_{\mathcal{I}(K)} \left\langle\left\langle \sigma_3 \mu_h (n_x^2 + n_y^2) \psi [[T]] \right\rangle\right\rangle_{\mathcal{I}} \\ &+ \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \sigma_3 \mu n_z^2 \psi [[T]] \right\rangle\right\rangle_{\mathcal{I}_h}. \end{aligned} \quad (55)$$

Including the interior penalty terms, the final horizontal and vertical diffusion operators, respectively, are

$$\begin{aligned} \mathcal{C}_\mu^h &= -\left\langle \mu_h (\nabla_h \psi) \cdot (\nabla_h T) \right\rangle_K + \sum_{\mathcal{I}(K)} \left\langle\left\langle \{\mu_h \nabla_h T\} \cdot \mathbf{n}_h \psi \right\rangle\right\rangle_{\mathcal{I}} \\ &+ \sum_{\mathcal{I}(K)} \left\langle\left\langle \sigma_3 \mu_h (n_x^2 + n_y^2) \psi [[T]] \right\rangle\right\rangle_{\mathcal{I}}, \end{aligned} \quad (56)$$

$$\mathcal{C}_\mu^v = -\left\langle \mu \frac{\partial T}{\partial z} \frac{\partial \psi}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \left\{ \mu \frac{\partial T}{\partial z} \right\} n_z \psi \right\rangle\right\rangle_{\mathcal{I}_h} + \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \sigma_3 \mu n_z^2 \psi [[T]] \right\rangle\right\rangle_{\mathcal{I}_h} \quad (57)$$

4.6. Computing the vertical velocity

Vertical flow velocity is computed diagnostically from the continuity equation (6). At the bottom boundary, w is determined by the impermeability boundary condition:

$$w = -\mathbf{u} \cdot \nabla_h h, \quad \mathbf{x} \in \Gamma_b. \quad (58)$$

In the interior, w is obtained by integrating $\partial w / \partial z = -\nabla_h \cdot \mathbf{u}$ over the vertical. In practice this is solved with the following weak formulation over an element K :

$$\left\langle \frac{\partial w}{\partial z} \psi \right\rangle_K = -\left\langle \nabla_h \cdot \mathbf{u} \psi \right\rangle_K \Leftrightarrow \quad (59)$$

$$\begin{aligned} \left\langle\left\langle w^d \psi n_z \right\rangle\right\rangle_{\mathcal{I}_{top}} - \left\langle w \frac{\partial \psi}{\partial z} \right\rangle_K &= -\left\langle\left\langle w^d \psi n_z \right\rangle\right\rangle_{\mathcal{I}_{bot}} - \sum_{\mathcal{I}_{lat}(K)} \left\langle\left\langle \psi \mathbf{u}^* \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{I}_{lat}} \\ &\quad - \sum_{\mathcal{I}_h(K)} \left\langle\left\langle \psi \mathbf{u}^d \cdot \mathbf{n}_h \right\rangle\right\rangle_{\mathcal{I}_h} + \left\langle \mathbf{u} \cdot \nabla_h \psi \right\rangle_K \end{aligned} \quad (60)$$

The top and bottom faces of the element K are denoted by \mathcal{I}_{top} and \mathcal{I}_{bot} , respectively. As w is unknown on the top interface, the left hand side of (60) is assembled in a modified mass matrix⁴, while the other terms are assembled on the right hand side.

The vertical velocity is first solved for the bottom most elements, using the bottom boundary condition. Consequently w is known at the top of these elements which is used as a “bottom boundary condition” for the next layer of elements. Repeating the procedure until all elements are integrated results in a fully discontinuous w field in the whole domain.

In the right hand side of (60), a value of \mathbf{u}^* is required in the lateral interfaces. To ensure discrete consistency with the horizontal momentum equation and the tracer equation it is essential that \mathbf{u}^* of (46) is used also here.

4.7. Computing the baroclinic head

The definition of the baroclinic head (4) contains a vertical integral. It is solved in a manner similar to w , except that here the solution is known at the surface:

$$r = 0, \quad \mathbf{x} \in \Gamma_s. \quad (61)$$

Therefore, the integration is performed from the surface to the bottom, i.e. $r = -(1/\rho_0) \int_{\eta}^z \rho' dz$. The corresponding weak form is

$$\left\langle\left\langle r^u \psi n_z \right\rangle\right\rangle_{\mathcal{I}_{bot}} - \left\langle r \frac{\partial \psi}{\partial z} \right\rangle_K = -\left\langle\left\langle r^u \psi n_z \right\rangle\right\rangle_{\mathcal{I}_{top}} - \frac{1}{\rho_0} \left\langle \rho' \psi \right\rangle_K, \quad (62)$$

where r^u denotes the value above the interface.

Assembling the two first terms to the modified mass matrix, r can be solved with a similar recursive procedure starting from the top most elements.

4.8. Computing the internal pressure gradient

The horizontal gradient of r appears in the momentum equation (1). As in the case of the external pressure gradient, $g \nabla_h \eta$, obtaining a robust estimate of $g \nabla_h r$ is essential for

⁴It is noteworthy that the associated matrix does not depend on vertical scaling of the mesh and thus it is not necessary to recompute it as the mesh is updated.

numerical stability. In the DG method this must be done carefully as the discontinuities of r should also be taken into account in the gradient. For the external pressure gradient $g\nabla_h\eta$, numerical stability is achieved by integrating the term by parts and using an approximate Riemann solver for the 2D surface gravity waves as explained in Section 4.3.1.

For the baroclinic head r , however, it is not straightforward to derive an approximate Riemann solver, because r is not a local variable (i.e. it depends on ρ' in the interval $[z, \eta]$). Here $\nabla_h r$ is solved with the following weak form:

$$\left\langle \nabla_h r \cdot \boldsymbol{\psi} \right\rangle_K = \sum_{\mathcal{I}(K)} \left\langle \left\langle r^* \boldsymbol{\psi} \cdot \mathbf{n}_h \right\rangle_{\mathcal{I}} \right\rangle_K - \left\langle r \nabla_h \cdot \boldsymbol{\psi} \right\rangle_K \quad (63)$$

In above, r^* is still required at the interface. As no Riemann solution is available, an arithmetic mean $r^* = (r_L + r_R)/2$ is used. Although the arithmetic mean ignores potentially significant physical processes (e.g. advection and gravitational adjustment) across the interface, (63) still produces a better estimate of $\nabla_h r$ than using the local gradient (i.e. field r and gradients of the basis functions).

For the 2D momentum equation, a depth average of the internal pressure gradient must be computed. Once $\nabla_h r$ is known, it is obtained from the definition

$$\overline{\nabla_h r} = \frac{1}{H} \int_{-h}^{\eta} \nabla_h r dz, \quad (64)$$

where the vertical integral is computed by summing the nodal values, weighted by the element heights.

5. Matrix form

5.1. 2D equations

A discrete system of the 2D free surface equation is obtained from (30) by replacing η , h and $\bar{\mathbf{u}}$ by the respective DG-FE approximations $\hat{\eta} = \sum_i \eta_i \varphi_i$, $\hat{h} = \sum_i h_i \varphi_i$ and $\hat{\bar{\mathbf{u}}} = \sum_k \bar{\mathbf{u}}_k \varphi_k$.

Using the basis functions φ_j as the test function and summing over all the elements, the weak formulation can then be written as

$$\begin{aligned} \left\langle \left\langle \frac{\partial \sum_i \eta_i \varphi_i}{\partial t} \varphi_j \right\rangle_{\Gamma_{s,0}} \right\rangle_Q &= \sum_Q \left\langle \left\langle \sum_i (\eta_i + h_i) \varphi_i \sum_k \bar{\mathbf{u}}_k \varphi_k \cdot \nabla_h \varphi_j \right\rangle_Q \right\rangle_Q \\ &\quad - \sum_Q \left\langle \left\langle (\eta^* + \sum_i h_i \varphi_i) \bar{\mathbf{u}}^* \cdot \mathbf{n}_h \right\rangle_{\mathcal{J}(Q)} \varphi_j \right\rangle_{\mathcal{J}(Q)}, \quad \forall j. \end{aligned} \quad (65)$$

As $\Gamma_{s,0}$ and φ_i do not depend on time, the latter can be expressed in a matrix form

$$\frac{d\mathbf{E}}{dt} = (\mathbf{M}_{2D})^{-1} \mathbf{B}_\eta =: \tilde{\mathbf{B}}_\eta, \quad (66)$$

where $[\mathbf{M}_{2D}]_{ij} = \langle\langle \varphi_i \varphi_j \rangle\rangle_\Omega$ is the 2D mass matrix, $[\mathbf{E}]_i = \eta_i$ and \mathbf{B}_η denotes the right hand side (RHS) of (65).

The mass matrix is block diagonal, each block $\mathbf{M}_{2D}^Q \in \mathbb{R}^{N_{2D} \times N_{2D}}$ corresponding to an element Q with N_{2D} nodes. Consequently \mathbf{M}_{2D}^{-1} can be easily obtained by inverting the blocks \mathbf{M}_{2D}^Q . As the 2D mesh does not depend on time, the inverses can be precomputed.

In the case of the depth averaged momentum equation (31) a matrix form is derived in a similar manner. Taking $\boldsymbol{\varphi} = [\varphi_j, 0]$ and $\boldsymbol{\varphi} = [0, \varphi_j]$ as the test function leads into equations for \bar{u}_i and \bar{v}_i , respectively. Denoting the nodal values by $[\bar{\mathbf{U}}_u]_i = \bar{u}_i$ and $[\bar{\mathbf{U}}_v]_i = \bar{v}_i$, and the corresponding RHS by $\mathbf{B}_{\bar{u}}$ and $\mathbf{B}_{\bar{v}}$, one obtains

$$\begin{aligned} \widetilde{\mathbf{M}}_{2D} &= \begin{bmatrix} \mathbf{M}_{2D} & 0 \\ 0 & \mathbf{M}_{2D} \end{bmatrix}, & \bar{\mathbf{U}} &= \begin{bmatrix} \bar{\mathbf{U}}_u \\ \bar{\mathbf{U}}_v \end{bmatrix}, & \mathbf{B}_{\bar{u}\bar{v}} &= \begin{bmatrix} \mathbf{B}_{\bar{u}} \\ \mathbf{B}_{\bar{v}} \end{bmatrix}, \\ \frac{d\bar{\mathbf{U}}}{dt} &= \left(\widetilde{\mathbf{M}}_{2D}\right)^{-1} \mathbf{B}_{\bar{u}\bar{v}} =: \widetilde{\mathbf{B}}_{\bar{u}\bar{v}}. \end{aligned} \quad (67)$$

5.2. 3D equations

In the case of the 3D equations, a discrete system is obtained from the weak formulation by introducing the DG-FE approximation $\hat{\eta} = \sum_i \eta_i \psi_i$ for all the scalar fields (η , w , w_m , $\partial w_m / \partial z$, T, S, ν_h , ν) and $\hat{\mathbf{u}} = \sum_i \mathbf{u}_i \psi_i$ for the vector fields \mathbf{u} and $\nabla_h r$.

Summing over all the prisms, the weak form of the tracer equation (50) becomes

$$\sum_i \frac{d}{dt} \langle T_i \psi_i \psi_j \rangle_\Omega = \sum_K B_u^{K,j}(\hat{T}, \hat{\mathbf{u}}, \dots) + \sum_K C_u^{K,j}(\hat{T}, \dots), \quad \forall j, \quad (68)$$

where $C_T^{K,j}$ contains the terms related to vertical diffusion (49), while all the other terms are grouped in $B_T^{K,j}$.

Denoting the nodal values by $[\mathbf{T}]_i = T_i$, equation (68) can be written in matrix form

$$\frac{d}{dt} (\mathbf{M}_{3D} \mathbf{T}) = \widetilde{\mathbf{B}}_T + \widetilde{\mathbf{C}}_T, \quad (69)$$

where $\widetilde{\mathbf{B}}_T$ and $\widetilde{\mathbf{C}}_T$ denote the contribution of $B_T^{K,j}$ and $C_T^{K,j}$, respectively, and $[\mathbf{M}_{3D}]_{ij} = \langle \psi_i \psi_j \rangle_\Omega$ the 3D mass matrix. The mass matrix is again block diagonal with blocks $\mathbf{M}_{3D}^K \in \mathbb{R}^{N_{3D} \times N_{3D}}$, N_{3D} being the number of nodes in a prism K . Due to the moving mesh, however, \mathbf{M}_{3D} and its inverse must be recomputed after each mesh update.

The momentum equation (40) is treated in a similar fashion. With the nodal values $[\mathbf{U}_u]_i = u_i$ and $[\mathbf{U}_v]_i = v_i$, the discrete system can be expressed as

$$\begin{aligned} \widetilde{\mathbf{M}}_{3D} &= \begin{bmatrix} \mathbf{M}_{3D} & 0 \\ 0 & \mathbf{M}_{3D} \end{bmatrix}, & \mathbf{U} &= \begin{bmatrix} \mathbf{U}_u \\ \mathbf{U}_v \end{bmatrix}, & \widetilde{\mathbf{B}}_{uv} &= \begin{bmatrix} \widetilde{\mathbf{B}}_u \\ \widetilde{\mathbf{B}}_v \end{bmatrix}, & \widetilde{\mathbf{C}}_{uv} &= \begin{bmatrix} \widetilde{\mathbf{C}}_u \\ \widetilde{\mathbf{C}}_v \end{bmatrix}, \\ \frac{d}{dt} (\widetilde{\mathbf{M}}_{3D} \mathbf{U}) &= \widetilde{\mathbf{B}}_{uv} + \widetilde{\mathbf{C}}_{uv}, \end{aligned} \quad (70)$$

where $\widetilde{\mathbf{C}}_u$ and $\widetilde{\mathbf{C}}_v$ correspond to the vertical diffusion terms (49), while all the remaining terms are encapsulated in $\widetilde{\mathbf{B}}_u$ and $\widetilde{\mathbf{B}}_v$.

6. Time integration

This section outlines the time integration method for the coupled 2D-3D shallow water equations. First, the typical time step constraints are outlined. Then the overall time integration method is presented, followed by a scheme for the 3D and 2D equations.

6.1. Maximum admissible time step

The long surface gravity waves travel at speed $c = \sqrt{gH}$. If the length scale of an element in the triangulation \mathcal{T} is L_h , the longest admissible time step according to the CFL (Courant–Friedrichs–Lewy) condition is

$$\Delta t_c \propto \frac{L_h}{c + U}, \quad (71)$$

where $U \geq 0$ is the maximal advective speed. In many marine applications $c \gg U$, i.e. the surface gravity waves are more restrictive than horizontal advection, or internal wave propagation alone.

Due to the aspect ratio of the horizontal and vertical length scales of the ocean, the stability constraint due to vertical advection is comparable to that of horizontal advection. Vertical diffusion, on the other hand, may impose a stricter condition on a fine vertical mesh

$$\Delta t_\nu \propto \frac{L_z^2}{\nu}. \quad (72)$$

Consequently, for computational efficiency, the split-explicit method is adopted for treating the external (2D) and internal (3D) modes. The surface gravity waves are solved with relatively inexpensive 2D depth averaged equations with a high temporal resolution satisfying (71). The restriction imposed by (72), on the other hand, is circumvented by treating the vertical diffusion semi-implicitly.

6.2. Overview of the time integration method

Starting from an initial condition at time t_0 , the temporal discretisation is defined on interval $[t_0, t_{end}]$, with constant time increments Δt . The variables at $t_n = t_0 + n\Delta t$ are denoted by a superscript n . The domain at t_n is denoted by Ω^n .

The overall time marching scheme is illustrated in Figure 3. The 3D equations advanced in time with a third order Leap-Frog-Adams-Moulton (LF-AM3) predictor-corrector scheme following Shchepetkin and McWilliams (2009). The 2D equations are solved in a separate sub-routine described in Section 6.4. Considering only the tracer T , the complete time marching procedure can be summarized as follows

- Prediction stage (in Ω^n)

$$\begin{aligned} \mathbf{T}^{n-1/2} &= \left(\frac{1}{2} - 2\gamma\right) \mathbf{P}_{n-1}^n \mathbf{T}^{n-1} + \left(\frac{1}{2} + 2\gamma\right) \mathbf{T}^n \\ \mathbf{T}^{n+1/2,*} &= \mathbf{T}^{n-1/2} + \Delta t(1 - 2\gamma)(\mathbf{M}_{3D}^n)^{-1} \mathbf{B}_T^n \end{aligned} \quad (73)$$

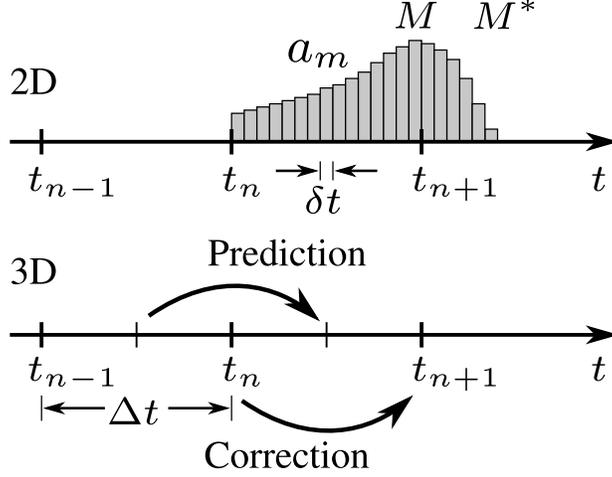


Figure 3: Schematic of the time integration method. During each macro time step, the 2D mode is advanced M^* iterations with a time step $\delta t = \Delta t/M$. The 2D variables are averaged with a filter defined by the weights a_m (see Section 6.4.1).

- Advancing the 2D equations
- Correction stage (in Ω^{n+1})

$$\mathbf{T}^{n+1,\dagger} = \mathbf{P}_n^{n+1} \mathbf{T}^n + \Delta t (\mathbf{M}_{3D}^{n+1})^{-1} \mathbf{B}_T^{n+1/2} \quad (74)$$

$$\mathbf{T}^{n+1} = \mathbf{T}^{n+1,\dagger} + \Delta t \Theta (\mathbf{M}_{3D}^{n+1})^{-1} \mathbf{C}_T^{n+1} + \Delta t (1 - \Theta) (\mathbf{M}_{3D}^n)^{-1} \mathbf{C}_T^n. \quad (75)$$

Above, γ and Θ are parameters related to the temporal scheme and the operator $\mathbf{P}_b^a = (\mathbf{M}_{3D}^a)^{-1} \mathbf{M}_{3D}^b$.

The 2D mode provides η and $\bar{\mathbf{u}}$ for the 3D mode. The free surface elevation η is used both to update the geometry Ω_n and to compute the external pressure gradient. The depth averaged velocity $\bar{\mathbf{u}}$ is used to adjust the 3D horizontal velocity as explained in Section 6.5. The 3D mode, on the other hand, affects the 2D mode through bottom friction, internal pressure gradient and advection of $\tilde{\mathbf{u}}$ as seen in equation (17).

6.2.1. Conservation and consistency

The prediction stage is solved in a single domain Ω^n using the non-conservative ALE formulation. The correction stage, on the other hand, includes both Ω^n and Ω^{n+1} and thus the conservative ALE formulation is used in this stage. This choice leads to a mass conservative and consistency preserving scheme.

Using the conservative ALE formulation in the prediction stage would break tracer consistency, because the volumetric change is not properly taken into account. The non-conservative formulation in this stage, on the other hand, preserves tracer consistency at the expense of losing mass conservation. Mass conservation, however, is required only for the correction stage, for the whole scheme to be mass conservative (Shchepetkin and McWilliams,

2005). Therefore only the correction stage is required to be both mass conservative and tracer consistent.

6.3. Advancing the 3D equations

The prediction stage consists of updating \mathbf{T} from $t_{n-1/2}$ to $t_{n+1/2}$, with RHS \mathbf{B}_T evaluated at t_n . The initial condition $\mathbf{T}^{n-1/2}$ is obtained by temporal interpolation, controlled by the parameter γ . Setting $\gamma = 0$ results in a centred average and the LF-TR (Leap-Frog–Trapezoidal Rule) scheme. The third order accurate LF-AM3 scheme is obtained with $\gamma = 1/12$ (Shchepetkin and McWilliams, 2005), which eliminates the numerical modes of LF schemes.

After the 2D equations have been advanced from t_n to t_{n+1} , new geometry Ω^{n+1} becomes available. The corrector stage then updates \mathbf{T} to time t_{n+1} , where the RHS is computed with the predicted state ($\mathbf{U}^{n+1/2,*}$, $\mathbf{T}^{n+1/2,*}$, etc.).

For both prediction and correction stages, the 3D velocity is updated first, followed by a similar update of the tracers.

The correction stage is finalized by a semi-implicit evaluation of the vertical diffusion of momentum and the tracers. Setting $\Theta = 0.5$ corresponds to the classical second order Crank-Nicolson method. In this work a slightly higher value $\Theta = 0.6$ is chosen to damp possible spurious oscillations. In (75) \mathbf{C}_T^n is originally evaluated in the previous time step with geometry Ω^n , and consequently it is multiplied by the old inverse mass matrix. Note that the right hand sides \mathbf{C}_{uv} and \mathbf{C}_T consist only of terms (49) and (57), respectively. As there are no terms involving the lateral interfaces, \mathcal{I}_{lat} , these equations are independent for each column of prisms and can be solved separately.

6.4. Advancing the 2D equations

The depth averaged equations are advanced in time with a standard third order Adams-Bashforth (AB3) scheme. The time step of the 2D mode is denoted by δt and the micro time steps are indicated with an index $m \in [0, M^*]$ with the corresponding time $t_m = t_n + m\delta t$. For consistency, the 3D time step must be a multiple of the 2D time step $\Delta t = M\delta t$, where M is the time step ratio. In practice the 2D model will be advanced further than the next 3D time step t_{n+1} , so that $M^* > M$ (Figure 3).

Using the notation of Section 5.1, the AB3 iteration is then given by

$$\mathbf{E}^{m+1} = \mathbf{E}^m + \delta t \left[\left(\frac{3}{2} + \beta \right) \tilde{\mathbf{B}}_\eta^m - \left(\frac{1}{2} + 2\beta \right) \tilde{\mathbf{B}}_\eta^{m-1} + \beta \tilde{\mathbf{B}}_\eta^{m-2} \right], \quad (76)$$

$$\bar{\mathbf{U}}^{m+1} = \bar{\mathbf{U}}^m + \delta t \left[\left(\frac{3}{2} + \beta \right) \tilde{\mathbf{B}}_{\bar{u}\bar{v}}^m - \left(\frac{1}{2} + 2\beta \right) \tilde{\mathbf{B}}_{\bar{u}\bar{v}}^{m-1} + \beta \tilde{\mathbf{B}}_{\bar{u}\bar{v}}^{m-2} \right]. \quad (77)$$

For AB3 scheme $\beta = 5/12$, while setting $\beta = 0$ results in second order accurate AB2 scheme. Given the state $(\mathbf{E}^0, \bar{\mathbf{U}}^0)$ at macro time step n , the iteration is initialized with a forward Euler and an AB2 step.

6.4.1. Computing temporal averages

The 3D mode requires η and $\bar{\mathbf{u}}$ at macro time steps t_n and $t_{n+1/2}$. However, as the 2D mode is solved with a higher temporal resolution than the 3D equations, η and $\bar{\mathbf{u}}$ contain high frequency components that cannot be resolved in the 3D iteration. Consequently, in order to avoid signal aliasing, the 2D variables are low-pass filtered in time to represent their evolution in time scales close to Δt (or above).

Given the fields η , $\bar{\mathbf{u}}$ at the micro time steps m , the temporal average centred at time $n + 1$ is defined by a set of weights $\{a_m\}_{m=1}^{M^*}$. The following averages are defined

$$\eta_{\otimes}^{n+1} = \sum_{m=1}^{M^*} a_m \eta^m, \quad (78)$$

$$\bar{\mathbf{u}}_{\otimes}^{n+1} = \sum_{m=1}^{M^*} a_m \bar{\mathbf{u}}^m, \quad (79)$$

$$(H\bar{\mathbf{u}})_{\otimes}^{n+1} = \sum_{m=1}^{M^*} a_m H^m \bar{\mathbf{u}}^m. \quad (80)$$

The domain geometry is updated with η_{\otimes}^{n+1} . The temporal averages η_{\otimes}^{n+1} and $\bar{\mathbf{u}}_{\otimes}^{n+1}$ are used as initial conditions for the next 2D iteration. Finally $(H\bar{\mathbf{u}})_{\otimes}^{n+1}$ and $(H\bar{\mathbf{u}})_{\otimes}^{n+1/2}$ (defined below) are used to adjust the 3D velocity field.

The weights must fulfil the following normalization and centroid conditions (Shchepetkin and McWilliams, 2005)

$$\sum_{m=1}^{M^*} a_m = 1, \quad \sum_{m=1}^{M^*} \frac{m}{M} a_m = 1. \quad (81)$$

The latter condition means that the centroid of the filter corresponds to t_{n+1} , implying that the temporal averaging must extend beyond the next macro time step, i.e. $M^* > M$.

The temporal average of $H\bar{\mathbf{u}}$ centred at $n + 1/2$ is defined by another set of weights $\{b_m\}_{m=1}^{M^*}$

$$(H\bar{\mathbf{u}})_{\otimes}^{n+1/2} = \sum_{m=1}^{M^*} b_m H^{m-1/2} \bar{\mathbf{u}}^{m-1/2}, \quad (82)$$

subject to similar conditions

$$\sum_{m=1}^{M^*} b_m = 1, \quad \sum_{m=1}^{M^*} \frac{m}{M} b_m = \frac{1}{2}. \quad (83)$$

This temporal average should be in agreement with a single macro time step update of the elevation

$$\eta_{\otimes}^{n+1} = \eta_{\otimes}^n + \Delta t \nabla_h \cdot (H\bar{\mathbf{u}})_{\otimes}^{n+1/2}. \quad (84)$$

Shchepetkin and McWilliams (2005) show that the above condition holds if the weights b_m are chosen as

$$b_{m'} = \frac{1}{M} \sum_{m=m'}^{M^*} a_m, \quad (85)$$

so that b_m are unique and depend only on a_m . The exact form of the filter coefficients a_m b_m is presented in Appendix A.

It is worth noticing that, split-explicit models have the advantage that the filters can be specially designed for desired properties (low dissipation and dispersion in active range, sufficient damping of high frequencies). Shchepetkin and McWilliams (2005) show that even a highly dissipative constant filter $a_i = 1/M^*$, $M^* = 2M$ is in fact less dissipative than a backward Euler implicit free surface model, that is still used in some models.

6.5. Adjusting the 3D velocity

Because the horizontal momentum equation is defined both in 3D and in 2D, the corresponding velocity fields \mathbf{u} and $\bar{\mathbf{u}}$ do not automatically agree. Therefore the 3D velocity field \mathbf{u} is corrected such that its depth average matches $\bar{\mathbf{u}}$, which is in agreement with the η field. The adjusted velocity can be written as

$$\mathbf{u}_{adj}^n = \mathbf{u}^n + \frac{1}{H^n} \left((H\bar{\mathbf{u}})_{\otimes}^n - \int_{-h}^{\eta} \mathbf{u}^n dz \right). \quad (86)$$

The 3D velocity is adjusted after each update in the prediction, correction and semi-implicit stages. At the end of each macro time step, the final \mathbf{u}^{n+1} is adjusted with $(H\bar{\mathbf{u}})_{\otimes}^{n+1}$. During the prediction stage, at $t_{n+1/2}$, the 2D mode is not yet solved and consequently $(H\bar{\mathbf{u}})_{\otimes}^{n+1/2}$ is not yet available. In this case older $(H\bar{\mathbf{u}})_{\otimes}^n$ is used instead, before executing the prediction stage of the tracers. After the 2D mode is solved, $\mathbf{u}^{n+1/2,*}$ is adjusted again with the correct $(H\bar{\mathbf{u}})_{\otimes}^{n+1/2}$.

6.6. Slope limiter

Monotonicity preserving advection schemes used with marine models include Total Variation Diminishing (TVD) schemes in finite volumes (Pietrzak, 1998) or flux-corrected schemes in continuous finite elements (Kuzmin et al., 2005). The monotonicity property ensures that local maxima (minima) do not increase (decrease) in time.

Many slope limiter families exist in the literature for DG-FEM, for example the minmod limiter by Cockburn and Shu (1998). However, the implementation of many such filters depends on the element type and dimension. Therefore in this work a mass conservative, geometry-independent slope limiter is used. In its simple form, with only one degree of freedom per element, the limiter is similar to the one of Kuzmin (2010) and Aizinger (2011). An optimal version, that modifies the nodal values as little as possible, similarly to the minmod limiter, is also tested. These two versions are briefly outlined in Appendix B.

6.7. Updating mesh geometry

The water elevation η , as computed by the depth averaged equations, belongs to the space \mathbb{W} and is thus discontinuous. The mesh, however, must remain conforming, so that the lateral faces on \mathcal{I}_{lat} are the same on both sides. A conforming 3D mesh can be achieved by updating the mesh with a continuous elevation field η_c (Aizinger and Dawson, 2007), obtained from the discontinuous η . In this work a mass conserving weighted average filter is used. Another possibility is to project η on continuous basis functions with L_2 projection, which also conserves mass. However, the L_2 projection tends to create overshoots, while the average operation is purely diffusive.

Consequently, the time dependent domain Ω corresponds to η_c , and in all the equations related to surface boundary conditions and the mesh movement – namely (11) and (21) – η is replaced by η_c . Furthermore all the vertical integrals are defined on interval $z \in [-h, \eta_c]$. This leads into a small discrepancy in the numerical implementation, due to the error $\eta - \eta_c$.

7. Numerical tests

The performance of the presented 3D model were tested in a sequence of numerical benchmarks. Conservation and consistency properties were assessed in a barotropic test case, followed by a baroclinic gravitational adjustment benchmark. Finally, the model was applied to simulating the Rhine river plume in Section 7.3.

7.1. Surface gravity waves and conservation properties

Volume and mass conservation properties were tested with propagating surface gravity waves in a rectangular channel 10 km long, 1 km wide and 50 m deep. All lateral boundaries were set impermeable. Initially a free surface perturbation was prescribed along the channel: $\eta_0 = a_0 \exp(-(x/\sigma_0)^2)$, with $a_0 = 0.1$ m and $\sigma_0 = 2000$ m. Salinity evolution was computed with (8) imposing a constant initial value $S_0 = 4$ PSU. Temperature was taken as a constant 10°C throughout the simulation. Bottom friction and vertical diffusion were omitted in this test.

The domain was discretised with 100 m horizontal resolution and 20 vertical layers. The propagation of the gravity waves were simulated for 8000 s with 0.2 s 2D time step and $M = 30$. The observed relative error in volume and total tracer mass was of order 10^{-14} . The deviation in tracer concentration was higher, 10^{-6} . This error is due to the fact that the 3D fields and the free surface elevation are not exactly compatible, thus breaking tracer consistency. The dominant source of error is proportional to $\eta_c - \eta$, related to the smoothing of the free surface.

The simulation was repeated for a non-constant tracer field. Initially $S = 4$ PSU was prescribed at the bottom and $S = 3$ PSU at the surface boundaries with linear transition in between. Also in this case, the relative error in total tracer mass was of order 10^{-14} , thus verifying mass conservation. During the simulation, the tracer extrema were decreasing monotonically, suggesting that numerical dissipation alone was sufficient to filter out the spurious extrema in the tracer field.

A mesh refinement analysis was further carried out for the same setup. Horizontal resolutions ranged from 1000 m to 100 m and number of vertical layers were increased from 2 to 20. The finest resolution was used as a reference solution. Figure 4 presents the L_2 error in horizontal velocity versus horizontal mesh resolution, verifying the theoretical second order convergence.

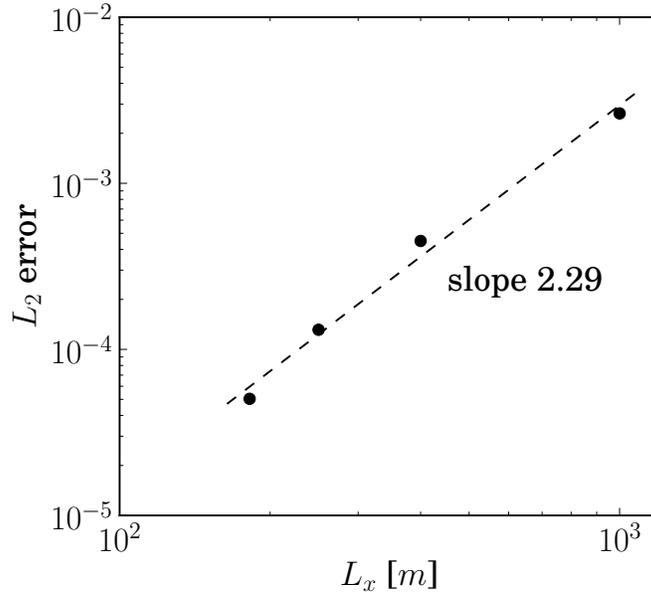


Figure 4: Convergence analysis of u in the surface gravity wave test.

7.2. Gravitational adjustment of a density front

The model’s ability to simulate buoyancy driven flows was assessed in a standard non-rotating gravitational adjustment test (e.g. Wang (1984); Haidvogel and Beckmann (1999); Jankowski (1999)). Initially a rectangular domain, 64 km long and 20 m deep, is filled with two fluids of slightly different densities. In the left half of the domain $\rho'_1 = 2.15 \text{ kgm}^{-3}$, while on the right $\rho'_2 = -2.15 \text{ kgm}^{-3}$ is prescribed. At time $t = 0$ the barrier separating the fluids is removed, and a density driven exchange flow develops, driving the dense fluid under the lighter fluid. Assuming that $\Delta\rho = \rho'_1 - \rho'_2 \ll \rho_0$ and that all potential energy is transformed into movement, it can be shown that the top and bottom fronts advance with speed $c = (1/2)\sqrt{g'H}$, where $g' = g\Delta\rho/\rho_0$ is the reduced gravity (Jankowski, 1999).

The exchange flow was simulated with horizontal resolution $L_h = 833$ m and $N_\sigma = 12$ sigma layers. Vertical diffusion, bottom friction and Coriolis force were neglected. The simulation was carried out using both the simple and the optimal slope limiter.

Figure 5 shows the initial density distribution and the situation after 31 500 s of simulation for both limiters. It is seen that the solution oscillates in both cases, suggesting that the

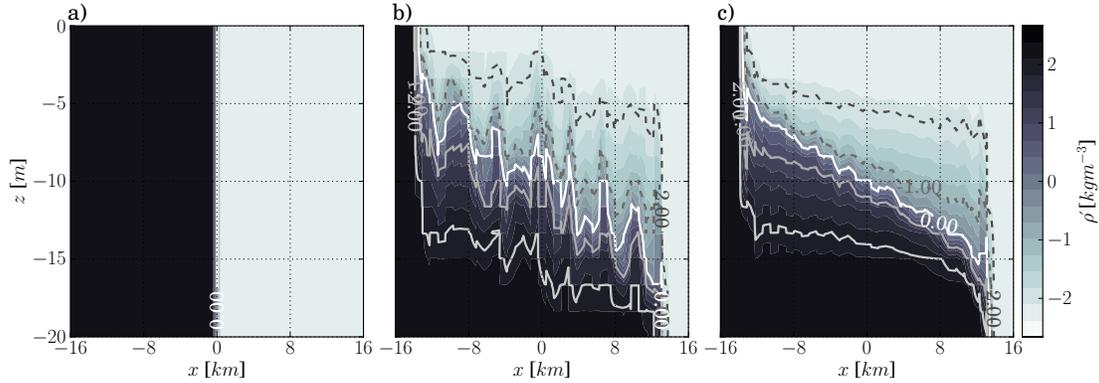


Figure 5: Gravitational adjustment without background diffusion. a) Initial density. Density after 31500 s of simulation in the case of the b) simple and c) optimal slope limiter.

internal pressure gradient term, that couples the tracer and momentum equations, is not fully stabilised. However, it is worth noticing that the slope limiter also plays a role: oscillations are much larger for the simple limiter. This is due to the fact that it tends to alter the solution excessively, which leads to noise in the tracer field. Therefore for the subsequent tests, only the optimal limiter is considered.

The mean exchange flow velocity is estimated as the front displacement divided by the elapsed time. For results shown in Figure 5, one obtains roughly 0.44 ms^{-1} which is comparable, and – as expected due to numerical dissipation – slightly smaller than the theoretical value $c = 0.46 \text{ ms}^{-1}$.

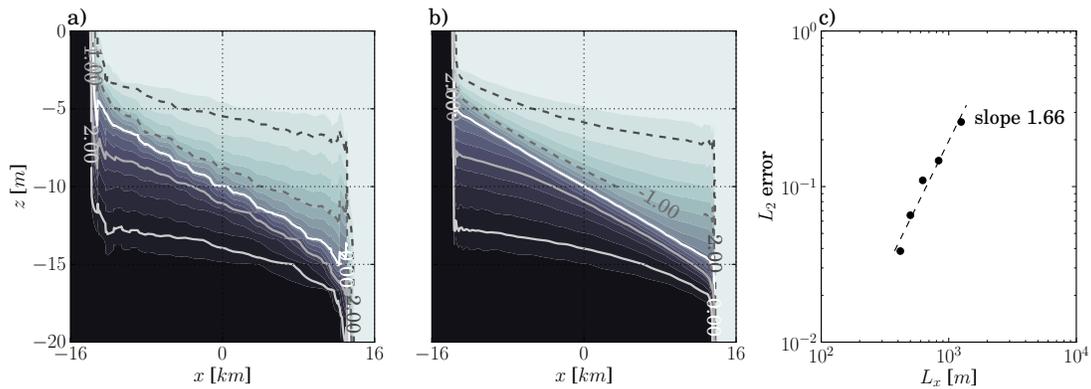


Figure 6: Gravitational adjustment with a constant background diffusion $\nu_h = \kappa_h = 5 \text{ m}^2\text{s}^{-1}$. Density at 31 500 s for a) coarse mesh ($L_h = 833 \text{ m}$, $N_\sigma = 12$) b) fine mesh ($L_h = 312 \text{ m}$, $N_\sigma = 32$). c) Convergence plot.

Also here a convergence analysis of the density distribution was performed. The mesh resolution varied between $L_h = 1250 \text{ m}$ ($N_\sigma = 8$) and $L_h = 312 \text{ m}$ ($N_\sigma = 32$). Because no analytical solution is available, the finest solution was used as a reference. However, owing to the oscillations, the simulation became unstable for finer meshes and as a remedy a small

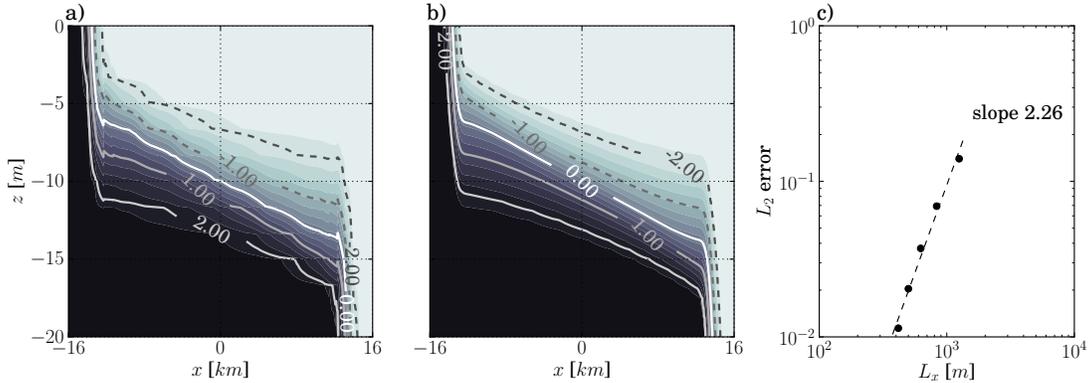


Figure 7: Gravitational adjustment with the Smagorinsky diffusion. Density at 31 500 s for a) coarse mesh ($L_h = 833$ m, $N_\sigma = 12$) b) fine mesh ($L_h = 312$ m, $N_\sigma = 32$). c) Convergence plot.

horizontal diffusion was added.

The results obtained with a constant background diffusivity are presented in Figure 6. Alternatively, a Smagorinsky diffusivity,

$$\nu_h = \kappa_h = \frac{C_s^2 L_h^2}{\pi^2} \sqrt{\left(\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2},$$

was applied⁵, with $C_s = 1.0$. This case is illustrated in Figure 7. It is seen that the oscillations are damped and the solution on the finest mesh is very smooth. Using the Smagorinsky scheme appears to be more diffusive – especially the fronts are smoother – but yields better rate of convergence, 2.26. In the case of a constant diffusion, small wiggles remain in the finest solution near the fronts, and the rate of convergence is only 1.66. It should be noted however, that the wiggles in the reference solution may hamper the convergence analysis in this case.

It can be concluded that under gravitational adjustment the model produces realistic results. However, some diffusion may be necessary to damp oscillations, especially with high resolution meshes. The oscillations are most likely related to insufficient stabilisation of the internal pressure gradient feedback. The rate of convergence is super-linear but not necessarily second order.

7.3. River plume

Areas where freshwater induced buoyancy plays an important role are often referred to as regions of freshwater influence (ROFI, Simpson, 1997). Typically ROFIs feature strong density gradients, that on the one hand drive the water motion, but on the other hand are affected by advection and mixing processes. Thus the flow exhibits a highly non-linear

⁵Note that although the Smagorinsky scheme was developed for viscosity, it is sometimes applied to tracers as well (e.g. Mellor, 2004).

behaviour, arising from the interaction of the tides, the Coriolis force, buoyancy and vertical mixing. In all the complexity ROFIs provide a good benchmark for baroclinic marine models.

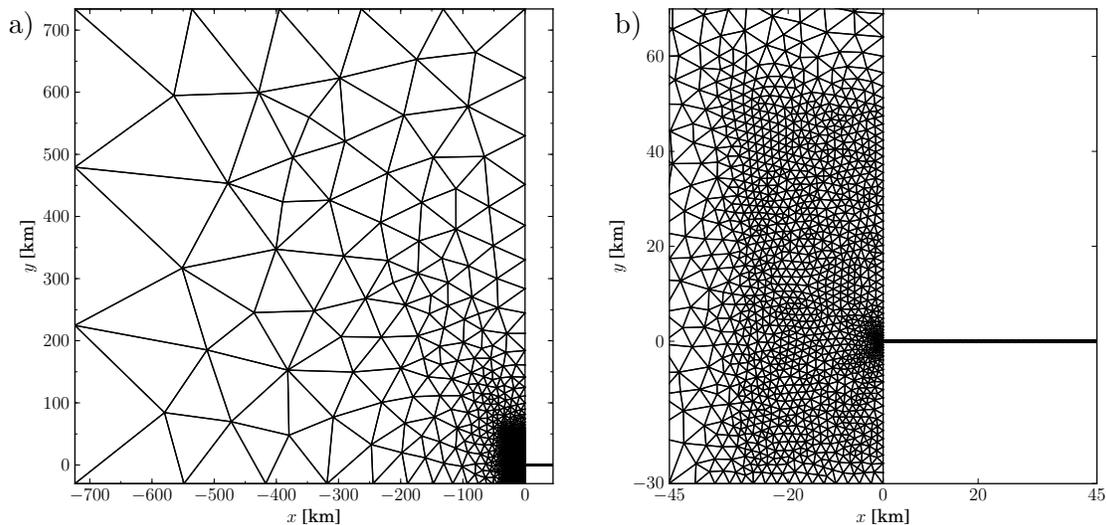


Figure 8: Horizontal mesh for the Rhine ROFI simulation. a) The whole domain (3167 triangular elements) b) Enlargement near the area of interest. The resulting 3D mesh, with 20 sigma layers, consists of 63 340 prisms and 380 040 degrees of freedom.

Numerical 3D modelling of the Rhine ROFI has been discussed in Ruddick et al. (1994, 1995); de Boer et al. (2006); de Boer et al. (2008); Fischer et al. (2009), among others. Here a simulation is carried out in an idealised geometry following de Boer et al. (2006). In this configuration the Dutch coastal zone is modelled as a 20 m deep rectangle extending 100 km in the alongshore (“north”) and 35 km in the across-shore (“west”) directions. The river is represented by a 45 km long and 500 m wide perpendicular channel whose depth reduces linearly to 5 m at the inlet. Such a simplistic geometry is acceptable as the Rhine ROFI is relatively unaffected by coastal and bathymetric features (de Ruijter et al., 1997).

To avoid issues caused by waves reflecting on the open boundaries, the domain is extended some 700 km north and west from the river inlet, gradually decreasing the mesh resolution (Figure 8). Near the river mouth the horizontal resolution is similar to the setup of de Boer et al. (2006). The element edge length is set to 500 m in the river, increasing to roughly 1200 m in the rest of the domain of interest (roughly 50 by 30 km in the alongshore and cross-shore directions). In vertical direction 20 sigma layers are used. The mesh is generated with GMSH mesh generation tool (Geuzaine and Remacle, 2009).

Initially the salinity is set to a constant 32 PSU value and a constant freshwater discharge $1500 \text{ m}^3\text{s}^{-1}$ is prescribed at the Rhine inlet. Following Fischer et al. (2009) the water elevation is forced at the three open boundaries with an M_2 Kelvin wave (amplitude 1.0 m, period $\tau = 44714 \text{ s}$). The Coriolis factor, taken as a constant, corresponding to latitude 52.2° north. Wind forcing is neglected.

The freshwater inflow causes a well formed salt wedge in the river. Once released from the

river, the plume turns to the right and forms a typical freshwater source for the northward coastal current. After roughly 20 tidal cycles the main plume shows nearly periodic behaviour. The results discussed in this section are of the 31st tidal cycle, similarly to de Boer et al. (2006); Fischer et al. (2009).

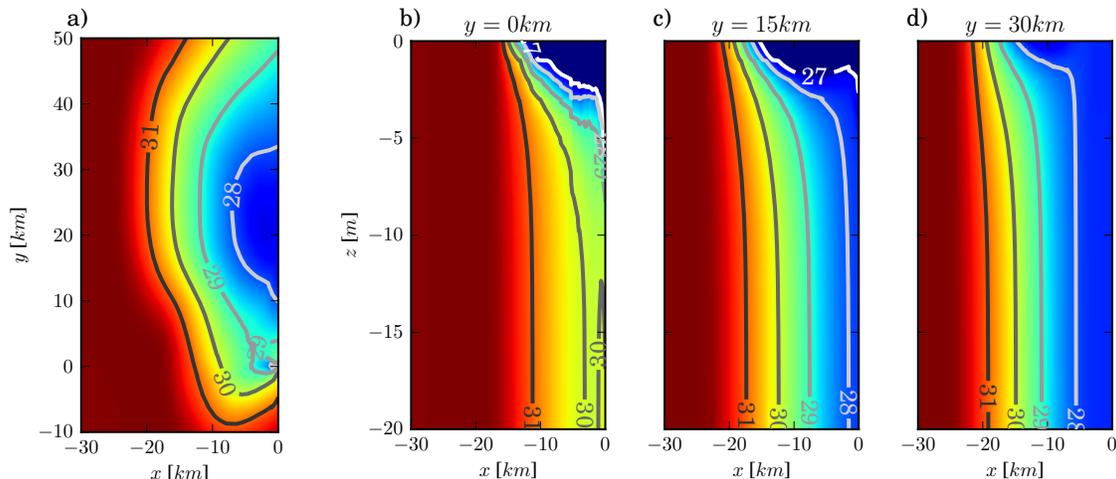


Figure 9: Tidally averaged salinity distribution. a) Depth average. Vertical transect at b) river mouth $y=0$, c) 15 km downstream, d) 30 km downstream.

Tidally averaged salinity is presented in Figure 9. The main plume extends to some 30 km offshore and 100 km northward alongshore from the river mouth. In the “upstream” direction, the river plume extends roughly 20 km southward. In the surface layer, roughly 5 m deep, the water column is strongly stratified with fresh water trapped near the surface. Further below bottom friction induced mixing dominates and the water column becomes nearly homogeneous in vertical direction, salinity decreasing toward the coast.

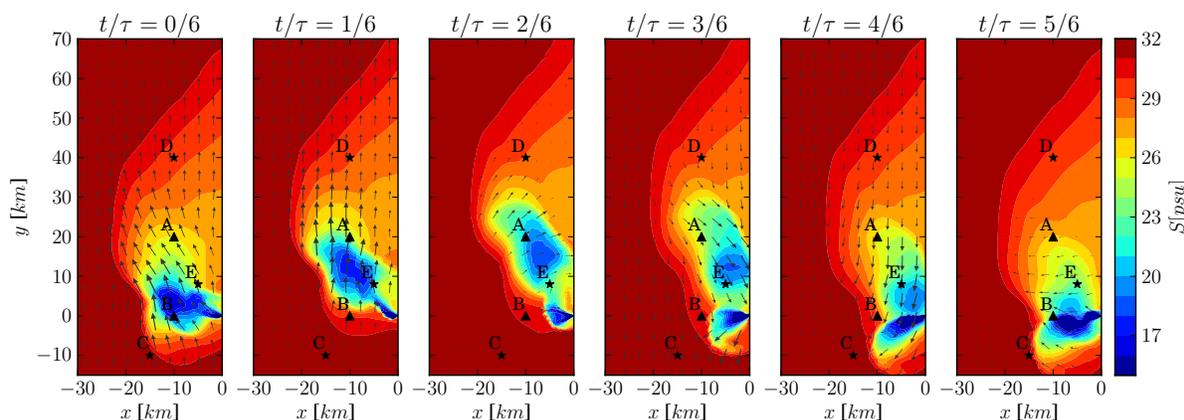


Figure 10: Evolution of surface salinity (0.5 m below surface) over a tidal cycle. Arrows illustrate the horizontal flow velocity. Maximal velocity is roughly 1.3 ms^{-1} . Temporal evolution of velocity is examined in stations A and B, and stratification in stations C, D and E.

de Ruijter et al. (1997) showed that the Rhine river plume exhibits a pulsed freshwater discharge due to two reasons. First, the along-river tidal current is stronger than the mean discharge velocity at the mouth, pinching off the river discharge periodically. Second, the river mouth is narrow compared to the inertial trajectory radius of the discharged water, so that a freshwater lens is separated from the river mouth before a new pulse is generated. The tidal evolution of the surface salinity and currents is presented in Figure 10. The freshwater lens, released south west from the river mouth during rising tide ($t/\tau = 3/6$ to $t/\tau = 5/6$), is clearly visible. It is transported northward during the ebb, merging with the main plume at low tide.

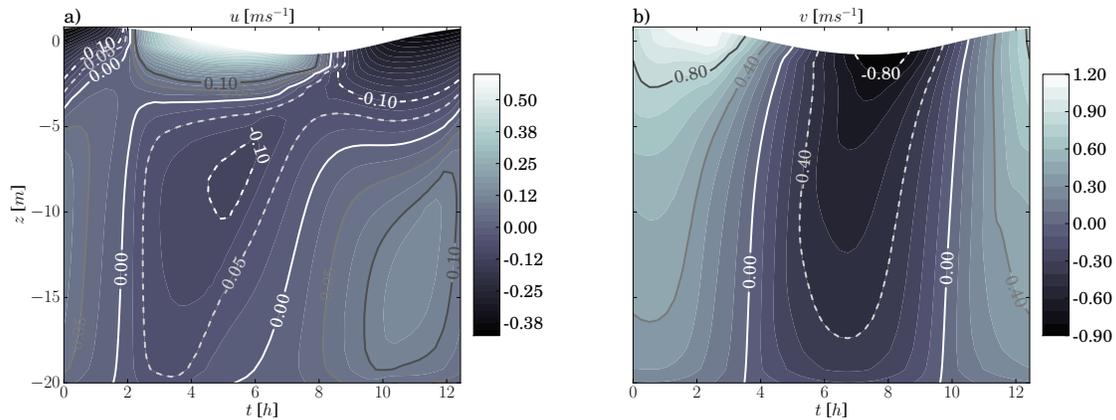


Figure 11: Time series of velocities at Station A (10 km offshore and 20 km north of the river mouth). a) Cross-shore velocity. b) Alongshore velocity.

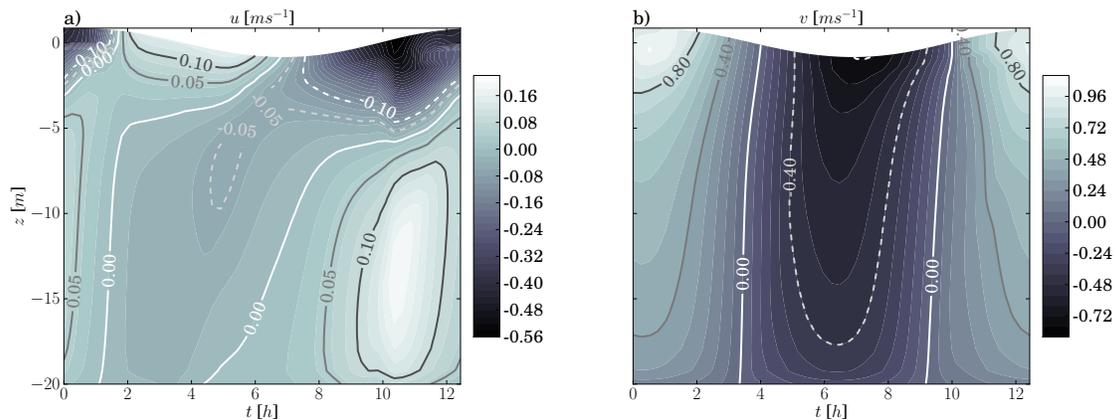


Figure 12: Time series of velocities at Station B (10 km offshore and 0 km north of the river mouth). a) Cross-shore velocity. b) Alongshore velocity.

In the ROFI the flow velocity shows asymmetric pattern. On the surface, the tidal velocity rotates clockwise (anti-cyclonically) while in bottom layer (below 5m depth) the rotation is anti-clockwise (cyclonic). Further downstream, where the influence of the stratification is

small, the flow reduces to nearly unidirectional (i.e. cross-shore component is zero) as in the case of pure Kelvin waves. Figures 11 and 12 present time series of the horizontal velocity at two locations, marked A and B in Figure 10. In station A the asymmetric velocity pattern is visible as the cross-shore velocity has opposite sign in the surface and bottom layers. This behaviour is related to the movement of the freshwater lens in the cross-shore direction. Intensified river outflow associated with the freshwater pulse, compensated by a similar saline intrusion in the bottom layer, can be seen in station B (Figure 12).

It is well known that in the ROFI the stratification conditions vary significantly in time and space (Simpson et al., 1993; de Boer et al., 2008). Figure 13 presents the minimal and maximal gradient Richardson number $Ri = N^2/R^2$ (where N and R are the Buoyancy and (vertical) shear frequency respectively) in the surface layer over the tidal cycle. Comparing the two plots, it is evident that in several locations the surface layer alternates between stratified and mixed conditions. The temporal evolution of the stratification is examined in more detail in the locations C,D and E in Figure 13.

Figure 14 presents the temporal evolution of stratification at station C. This station is located at the southern boundary of the main “bulge” of the plume. It is influenced by the freshwater outflow during the rising tide (see the two last panels in Figure 10). Otherwise the water column is well mixed, and unstably stratified ($N^2 < 0$) after high water. The unstable stratification is a good example of Strain Induced Periodic Stratification (SIPS, Simpson et al., 1990). The tidal currents, being stronger near the surface, tilt the horizontal salinity gradient. As $\partial S/\partial y$ is negative at this location, flood currents push saline water over lighter water giving rise to $N^2 < 0$. In de Boer et al. (2008) the different processes affecting stratification in the Rhine ROFI are analysed using the potential energy anomaly equation in a numerical 3D model. Their results also confirm that in this part of the plume, the stratification is dominated by alongshore straining and advection.

The situation at station D is presented in Figure 15. Because this location lies already 40 km North of the river mouth, the water column is quite homogeneous throughout the tidal cycle. However, here the SIPS pattern is opposite: as $\partial S/\partial y$ is positive it is the ebb that induces the unstable stratification.

The stratification pattern is more complicated at station E (Figure 16) due to the passing fronts related to the fresh water lens. The surface layer remains strongly stratified, but the shear is also significant due to stronger surface velocities (Figure 10). The competition of N^2 and R^2 result in a patch of $Ri < 0.25$ during the falling tide at depth 5 m below surface.

8. Conclusions

Discontinuous Galerkin methods are well-suited for solving advection dominated problems and have advantageous numerical properties, but so far they have not been extensively applied to ocean modelling. This article presents a DG baroclinic marine model that fulfils the essential requirements for simulating coastal density driven flows. Water volume and tracer mass are conserved up to machine precision. Spurious extrema in tracer fields are filtered by means of a slope limiter.

A split-explicit predictor-corrector time integration method similar to that of Shchepetkin and McWilliams (2005) is used. In the DG framework, explicit models bring some important advantages, such slope limiters and straightforward parallelisation. As computational cost is a major challenge for emerging unstructured grid models, the latter is an important feature for large scale and/or high resolution applications.

Vertical mesh movement is taken into account by means of ALE (Arbitrary Lagrangian Eulerian) formulation. A conservative ALE formulation is used for the correction stage, while a non-conservative formulation is used in the prediction stage. This choice leads to a strictly mass conservative scheme. However, as it is cumbersome to ensure discrete compatibility of the 2D free surface equation and the 3D equations, the tracer consistency criterion is satisfied only approximately. Alternatively, one could use the non-conservative formulation in both stages, yielding exact tracer consistency at the expense of losing global mass conservation. In this work we have chosen to retain mass conservation as it is presumably a more important property in environmental applications.

The model is tested with a gravitational adjustment benchmark, where the general features of the flow are well represented. Nevertheless some oscillations are visible, due to the lack of rigorous stabilisation of the internal pressure gradient. More work is needed to tackle this issue, as it seems that deriving a stabilisation term for internal wave processes is not straightforward. However, using an optimal slope limiter, that modifies the nodal values as little as possible, produces significantly better results. Moreover, a small diffusivity can be further introduced to reduce the oscillations.

Finally, the model is applied to the Rhine river plume in an idealised geometry following the setup of de Boer et al. (2006). The river plume results are well in agreement with other results in the literature, e.g. obtained with Delft-3D (de Boer et al., 2006; de Boer et al., 2008) and GETM (Fischer et al., 2009). The plume exhibits pulsating behaviour, releasing a clearly defined lens of riverine water each rising tide. In several locations, the stratification conditions change over the tidal cycle, altering between fully mixed and strongly stratified states.

Acknowledgements

Tuomas Kärnä is a Research fellow with the Belgian Fund for Research in Industry and Agriculture (FRIA). Eric Deleersnijder is an honorary Research associate with the Belgian National Fund for Scientific Research (FNRS). The present study was carried out in the framework of the project "Taking up the challenges of multi-scale marine modelling", which is funded by the Communauté Française de Belgique under contract ARC 10/15-028 (Actions de Recherche Concertée) with the aim of developing and using SLIM (www.climate.be/slim) and the project Tracing and Integrated Modelling of Natural and Anthropogenic Effects on Hydrosystems (TIMOTHY), an Interuniversity Attraction Pole (IAP6.13) funded by the Belgian Federal Science Policy Office (BELSPO). Computational resources have been provided by the supercomputing facilities of the Universite catholique de Louvain (CISM/UCL) and the Consortium des Equipements de Calcul Intensif en Federation Wallonie Bruxelles (CECI) funded by the Fond de la Recherche Scientique de Belgique (FRS-FNRS).

Appendix A. Temporal filter coefficients

In this work we are using a filter defined by the following generating function (Shchepetkin and McWilliams, 2005)

$$A(\tau) = A_0 \left[\left(\frac{\tau}{\tau_0} \right)^p \left(1 - \left(\frac{\tau}{\tau_0} \right)^q \right) - r \left(\frac{\tau}{\tau_0} \right) \right], \quad \tau = \frac{m}{M}, \quad (\text{A.1})$$

with parameters $p, q \in \mathbb{N}$ and $r \in \mathbb{R}$. Let τ^* be the largest real root of $A(\tau)$. Then the length of the filter is obtained as $M^* = \tau^* M$ and weights are $a_m = A(m/M)$, $m \in [1, M^*]$. The constant A_0 is found by imposing the constraint

$$\sum_{m=1}^{M^*} a_m = 1. \quad (\text{A.2})$$

The constant τ_0 is obtained from the second constraint

$$\sum_{m=1}^{M^*} \frac{m}{M} a_m = 1, \quad (\text{A.3})$$

which leads to a non-linear problem. It is solved with a secant method using the initial guess

$$\tau_0 = \frac{(p+2)(p+q+2)}{(p+1)(p+q+1)}. \quad (\text{A.4})$$

Optimizing τ_0 also ensures that b_m sums to unity and its centroid is close to half.

Notice that the root τ^* depends on the filter type (i.e. parameters p, q and r), but not on mode split ratio M . In contrast A_0 and τ_0 do depend on M , and must be computed in the model initialization. In this work the filter $p = 2, q = 2, r = 0.2846158$ is used (Shchepetkin and McWilliams, 2009).

Appendix B. 3D vertex-based slope limiter

In this work a mass conservative vertex-based slope limiter is used. It is applicable to deformed prismatic elements where the top and bottom faces are not necessarily horizontal.

Given a certain node \mathbf{x}_i in the 3D mesh, a ‘‘neighbourhood’’ of \mathbf{x}_i is defined as a set of elements sharing the node: $\mathcal{P}(\mathbf{x}_i) = \{K \in \mathcal{P} | \mathbf{x}_i \in K\}$. Consider a scalar field T whose nodal values in element K are $\{T_j^K\}_{j=1}^{N_{3D}}$. A mapping $\chi(j, K)$ maps a node j in element K to i in the global mesh indexing, so that T_j^K corresponds to node $\mathbf{x}_{\chi(j, K)}$.

The idea of a vertex based slope limiter is to ensure that no nodal value T_j^K at $\mathbf{x}_{\chi(j, K)}$ can exceed the minimum/maximum *mean value* of the elements sharing the node $\mathbf{x}_{\chi(j, K)}$. These bounds are denoted by $T_{\chi(j, K)}^{\text{Min}}$ and $T_{\chi(j, K)}^{\text{Max}}$, respectively. Defining the total mass in the

element K by $\|T^K\| := \sum_j V_j^K T_j^K$, where $V_j^K = \langle \psi_j \rangle_K$ is the volume associated with node j , the limiter consists of finding limited values \tilde{T}_j^K so that

$$\begin{aligned} T_{\chi(j,K)}^{\text{Min}} &\leq \tilde{T}_j^K \leq T_{\chi(j,K)}^{\text{Max}}, \\ \|\tilde{T}^K\| &= \|T^K\|. \end{aligned}$$

The major advantage of such vertex based limiter is that it is geometry independent, applicable to any dimension and all types of elements (Kuzmin, 2010; Aizinger, 2011).

Appendix B.1. Simple one-parameter limiter

A simple choice for the finding \tilde{T}_j^K is to consider the convex combination between the original values ($\lambda = 1$) and the mean value ($\lambda = 0$),

$$\tilde{T}_j^K = \lambda T_j^K + (1 - \lambda) \frac{\|T^K\|}{\sum_j V_j^K}.$$

Clearly the mass conservation criterion is met for all $\lambda \in [0, 1]$. Knowing the bounds, the maximum acceptable λ_j is determined for each node j , and the solution is taken as $\lambda = \min_j(\lambda_j)$ (Kuzmin, 2010).

The advantage of this approach is that the single parameter λ can be easily found. The drawback is that the solution is not optimal; if only a certain node needs to be limited, the solution may change significantly in other nodes as well. This disadvantage becomes more severe as the dimension of the problem is high, as with 3D elements.

Appendix B.2. Optimal limiter

An better limited solution can be found by requiring that the modification on each node remains minimal. Using a conventional quadratic penalisation, one obtains the following quadratic programming problem

$$\begin{aligned} \text{minimize} \quad & \sum_j |T_j^K - \tilde{T}_j^K|^2, \\ & \|\tilde{T}^K\| = \|T^K\|, \\ & T_{\chi(j,K)}^{\text{Min}} \leq \tilde{T}_j^K \leq T_{\chi(j,K)}^{\text{Max}}. \end{aligned}$$

This problem has $N_{3D} - 1$ degrees of freedom. Consequently it is more complicated to solve but provides better results in 3D applications.

References

Aizinger, V., 2011. A Geometry Independent Slope Limiter for the Discontinuous Galerkin Method, in: Krause, E., Shokin, Y., Resch, M., Kröner, D., Shokina, N. (Eds.), Computational Science and High Performance Computing IV. Springer Berlin / Heidelberg. volume 115 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pp. 207–217.

- Aizinger, V., Dawson, C., 2007. The local discontinuous Galerkin method for three-dimensional shallow water flow. *Computer Methods in Applied Mechanics and Engineering* 196, 734–746.
- Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D., 2002. Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. *SIAM Journal on Numerical Analysis* 39, 1749–1779.
- Baptista, A.M., Zhang, Y., Chawla, A., Zulauf, M., Seaton, C., Myers, E.P.I., Kindle, J., Wilkin, M., Burla, M., Turner, P.J., 2005. A cross-scale model for 3D baroclinic circulation in estuary–plume–shelf systems: II. Application to the Columbia River. *Continental Shelf Research* 25, 935–972.
- Bleck, R., Smith, L., 1990. A wind-driven isopycnic coordinate model of the north and equatorial Atlantic Ocean 1. Model development and supporting experiments. *Journal of Geophysical Research* 95, 3273–3285.
- Blumberg, A.F., Mellor, G.L., 1999. A description of a three-dimensional coastal ocean model, in: Heaps, N.S. (Ed.), *Three Dimensional Coastal Ocean Models*. American Geophysical Union. pages 1–16.
- de Boer, G., Pietrzak, J., Winterwerp, J., 2006. On the vertical structure of the Rhine region of freshwater influence. *Ocean Dynamics* 56, 198–216.
- de Brye, B., de Brauwere, A., Gourgue, O., Kärnä, T., Lambrechts, J., Comblen, R., Deleersnijder, E., 2010. A finite-element, multi-scale model of the Scheldt tributaries, river, estuary and ROFI. *Coastal Engineering* 57, 850–863.
- Burchard, H., Bolding, K., 2002. GETM - a general estuarine transport model. Scientific documentation. Technical Report EUR 20253 EN. European Commission.
- Burchard, H., Bolding, K., Villarreal, M.R., 1999. GOTM, a general ocean turbulence model. Theory, implementation and test cases. Technical Report EUR 18745. European Commission.
- Burchard, H., Rennau, H., 2008. Comparative quantification of physically and numerically induced mixing in ocean models. *Ocean Modelling* 20, 293–311.
- Canuto, V.M., Howard, A., Cheng, Y., Dubovikov, M.S., 2001. Ocean Turbulence. Part I: One-Point Closure Model – Momentum and Heat Vertical Diffusivities. *Journal of Physical Oceanography* 31, 1413–1426.
- Casulli, V., Cheng, R.T., 1992. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids* 15, 629–648.

- Casulli, V., Walters, R.A., 2000. An unstructured grid, three-dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids* 32, 331–348.
- Chen, C., Beardsley, R.C., Cowles, G., 2006. An unstructured grid, finite-volume coastal ocean model: FVCOM User Manual. Technical Report 06-0602. SMAST/UMASSD.
- Cockburn, B., 2003. Discontinuous Galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* 83, 731–754.
- Cockburn, B., Shu, C., 1998. The Runge-Kutta discontinuous Galerkin method for conservation laws V : multidimensional systems. *Journal of Computational Physics* 141, 199–224.
- Comblen, R., Lambrechts, J., Remacle, J.F., Legat, V., 2010. Practical evaluation of five partly discontinuous finite element pairs for the non-conservative shallow water equations. *International Journal for Numerical Methods in Fluids* 63, 701–724.
- de Boer, G.J., Pietrzak, J.D., Winterwerp, J.C., 2008. SST observations of upwelling induced by tidal straining in the Rhine ROFI. *Continental Shelf Research* 29, 263–277.
- Debreu, L., Blayo, E., 2008. Two-way embedding algorithms: a review. *Ocean Dynamics* 58, 415–428.
- Donea, J., Huerta, A., Ponthot, J.P., Rodríguez-Ferran, A., 2004. *Arbitrary Lagrangian–Eulerian Methods*. John Wiley & Sons, Ltd.
- Dukowicz, J.K., Smith, R.D., 1994. Implicit free-surface method for the Bryan-Cox-Semtner ocean model. *J. Geophys. Res.* 99, 7991–8014.
- Ern, A., Stephansen, A.F., Zunino, P., 2009. A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity. *IMA Journal of Numerical Analysis* 29, 235–256.
- Farhat, C., Geuzaine, P., Grandmont, C., 2001. The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids. *Journal of Computational Physics* 174, 669–694.
- Fischer, E., Burchard, H., Hetland, R., 2009. Numerical investigations of the turbulent kinetic energy dissipation rate in the Rhine region of freshwater influence. *Ocean Dynamics* 59, 629–641.
- Formaggia, L., Nobile, F., 2004. Stability analysis of second-order time accurate schemes for ALE–FEM. *Computer Methods in Applied Mechanics and Engineering* 193, 4097–4116.
- Geuzaine, C., Remacle, J.F., 2009. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 79, 1309–1331.

- Griffies, S.M., 2004. Fundamentals of ocean climate models. Princeton University Press.
- Griffies, S.M., 2010. Elements of MOM4p1. Technical Report No. 6. NOAA/Geophysical Fluid Dynamics Laboratory, GFDL Ocean Group.
- Haidvogel, D., Beckmann, A., 1999. Numerical Ocean Circulation Modeling. Environmental Science and Management, Imperial College Press. 4 edition.
- Hetland, R.D., 2005. Relating River Plume Structure to Vertical Mixing. *Journal of Physical Oceanography* 35, 1667–1688.
- Hofmeister, R., Burchard, H., Beckers, J.M., 2010. Non-uniform adaptive vertical grids for 3D numerical ocean models. *Ocean Modelling* 33, 70–86.
- Jackett, D.R., McDougall, T.J., Feistel, R., Wright, D.G., Griffies, S.M., 2006. Algorithms for Density, Potential Temperature, Conservative Temperature, and the Freezing Temperature of Seawater. *Journal of Atmospheric and Oceanic Technology* 23, 1709–1728.
- Jankowski, J.A., 1999. A non-hydrostatic model for free surface flows. Ph.D. thesis. Institut für Strömungsmechanik und ERiB, Universität Hannover.
- Kärnä, T., Legat, V., Deleersnijder, E., Burchard, H., 2012. Coupling of a discontinuous Galerkin finite element marine model with a finite difference turbulence closure model. *Ocean Modelling* 47, 55–64.
- Kasahara, A., 1974. Various vertical coordinate systems used for numerical weather prediction. *Monthly Weather Review* 102, 509–522.
- Killworth, P.D., Webb, D.J., Stainforth, D., Paterson, S.M., 1991. The Development of a Free-Surface Bryan–Cox–Semtner Ocean Model. *Journal of Physical Oceanography* 21, 1333–1348.
- Kuzmin, D., 2010. A vertex-based hierarchical slope limiter for hp-adaptive discontinuous Galerkin methods. *Journal of Computational and Applied Mathematics* 233, 3077–3085.
- Kuzmin, D., Löhner, R., Turek, S., 2005. Flux–Corrected Transport. Principles, Algorithms and Applications. Scientific Computation, Springer, Berlin.
- Lambrechts, J., Hanert, E., Deleersnijder, E., Bernard, P.E., Legat, V., Remacle, J.F., Wolanski, E., 2008. A multi-scale model of the hydrodynamics of the whole Great Barrier Reef. *Estuarine, Coastal and Shelf Science* 79, 143–151.
- LeVeque, R.J., 2002. Finite-Volume Methods for Hyperbolic Problems. Cambridge University Press.

- Marshall, J., Adcroft, A., Campin, J.M., Dutkiewicz, S., Evangelinos, C., Ferreira, D., Forget, G., Fox-Kemper, B., Heimbach, P., Hill, C., Hill, E., Hill, H., Jahn, O., Losch, M., Maze, G., Menemenlis, D., Molod, A., 2004. MITgcm User Manual. Technical Report. MIT EAPS.
- Mellor, G.L., 2004. Users guide for a three-dimensional, primitive equation, numerical ocean model (June 2004 version). Technical Report. Princeton University.
- Pietrzak, J., 1998. The Use of TVD Limiters for Forward-in-Time Upstream-Biased Advection Schemes in Ocean Modeling. *Monthly Weather Review* 126, 812–830.
- Pietrzak, J., Jakobson, J.B., Burchard, H., Vested, H.J., Petersen, O., 2002. A three-dimensional hydrostatic model for coastal and ocean modelling using a generalised topography following co-ordinate system. *Ocean Modelling* 4, 173–205.
- Rivière, B., 2008. Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation. Number 35 in *Frontiers in Applied Mathematics*, SIAM.
- Rodi, W., 1987. Examples of calculation methods for flow and mixing in stratified flows. *Journal of Geophysical Research* 92, 5305–5328.
- Ruddick, K., Deleersnijder, E., Luyten, P., Ozer, J., 1995. Haline stratification in the Rhine-Meuse freshwater plume: a three-dimensional model sensitivity analysis. *Continental Shelf Research* 15, 1597–1630.
- Ruddick, K., Deleersnijder, E., Mulder, T., Luyten, P., 1994. A model study of the Rhine discharge front and downwelling circulation. *Tellus A* 46, 149–159.
- de Ruijter, W.P.M., Visser, A.W., Bos, W.G., 1997. The Rhine outflow: A prototypical pulsed discharge plume in a high energy shallow sea. *Journal of Marine Systems* 12, 263–276.
- Shahbazi, K., 2005. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics* 205, 401–407.
- Shchepetkin, A.F., McWilliams, J.C., 2005. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling* 9, 347–404.
- Shchepetkin, A.F., McWilliams, J.C., 2009. *Computational Kernel Algorithms for Fine-Scale, Multiprocess, Longtime Oceanic Simulations*. Elsevier. volume Volume 14. pp. 121–183.
- Simpson, J., Brown, J., Matthews, J., Allen, G., 1990. Tidal straining, density currents, and stirring in the control of estuarine stratification. *Estuaries and Coasts* 13, 125–132.

- Simpson, J.H., 1997. Physical processes in the ROFI regime. *Journal of Marine Systems* 12, 3–15.
- Simpson, J.H., Bos, W.G., Shirmer, F., Souza, A., Rippeth, T., Jones, S., Hydes, D., 1993. Periodic stratification in the Rhine ROFI in the North Sea. *Oceanologica Acta* 16, 23–32.
- Song, Y., Haidvogel, D., 1994. A Semi-implicit Ocean Circulation Model Using a Generalized Topography-Following Coordinate System. *Journal of Computational Physics* 115, 228–244.
- Toro, E., 2009. Riemann solvers and numerical methods for fluid dynamics: A practical introduction. Springer Verlag.
- Wang, D.P., 1984. Mutual intrusion of a gravity current and density front formation. *Journal of Physical Oceanography* 14, 1191–1199.
- Wang, Q., 2007. The finite element ocean model and its aspect of vertical discretization. Ph.D. thesis. Universität Bremen.
- Warner, J., Rockwell Geyer, W., Arango, H., 2010. Using a composite grid approach in a complex coastal domain to estimate estuarine residence time. *Computers & Geosciences* 36, 921–935.
- White, L., Deleersnijder, E., Legat, V., 2008a. A three-dimensional unstructured mesh finite element shallow-water model, with application to the flows around an island and in a wind-driven, elongated basin. *Ocean Modelling* 22, 26–47.
- White, L., Deleersnijder, E., Legat, V., 2008b. Tracer conservation for three-dimensional, finite-element, free-surface, ocean modeling on moving prismatic meshes. *Monthly Weather Review* 136, 420–442.
- Zhang, Y., Baptista, A.M., 2008. SELFE: A semi-implicit Eulerian-Lagrangian finite-element model for cross-scale ocean circulation. *Ocean Modelling* 21, 71–96.

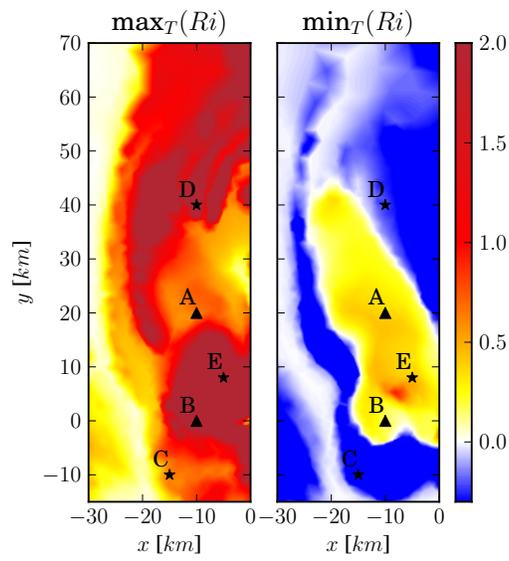


Figure 13: Maximal and minimal gradient Richardson number over the tidal cycle in the surface layer (1 m below surface). The surface layer is strongly stratified in areas where $Ri \gtrsim 1$ (left) and unstably stratified where $Ri < 0$ (right).

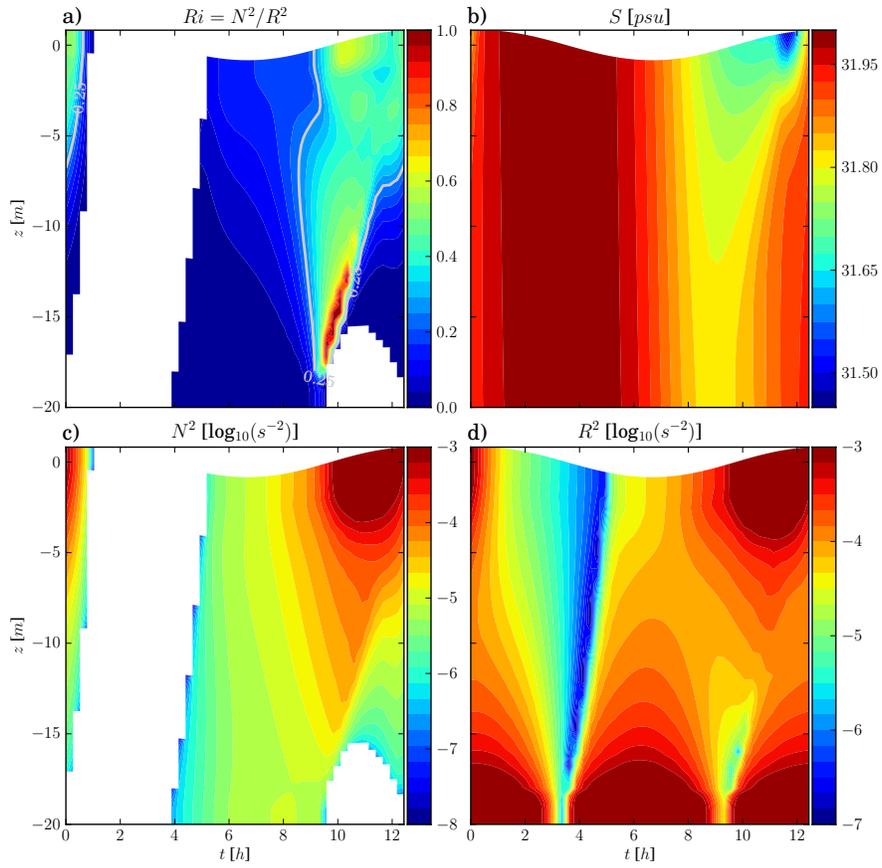


Figure 14: Temporal evolution of the stratification is station C (15 km offshore and 10 km south of the river mouth). a) Gradient Richardson number. The critical Richardson number $Ri_c = 0.25$ is indicated by a contour line. b) Salinity. c) Buoyancy frequency squared. d) Shear frequency squared. White patches indicate unstable stratification ($Ri, N^2 < 0$).

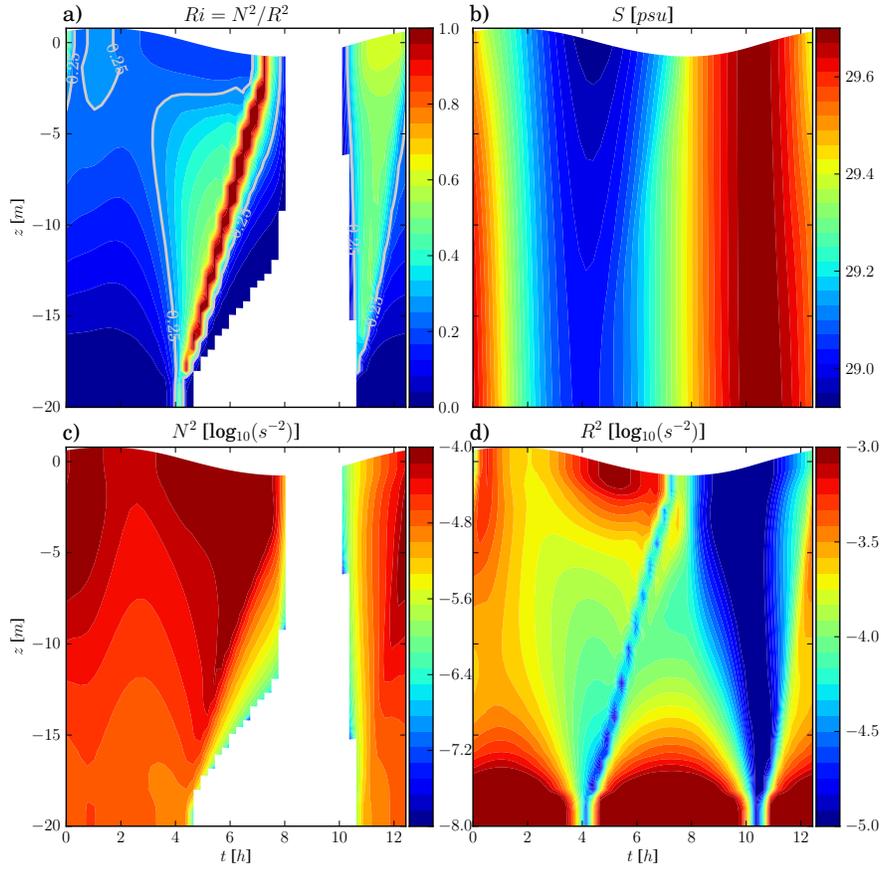


Figure 15: Temporal evolution of the stratification is station D (10 km offshore and 40 km north of the river mouth). a) Gradient Richardson number. The critical Richardson number $Ri_c = 0.25$ is indicated by a contour line. b) Salinity. c) Buoyancy frequency squared. d) Shear frequency squared. White patches indicate unstable stratification ($Ri, N^2 < 0$).

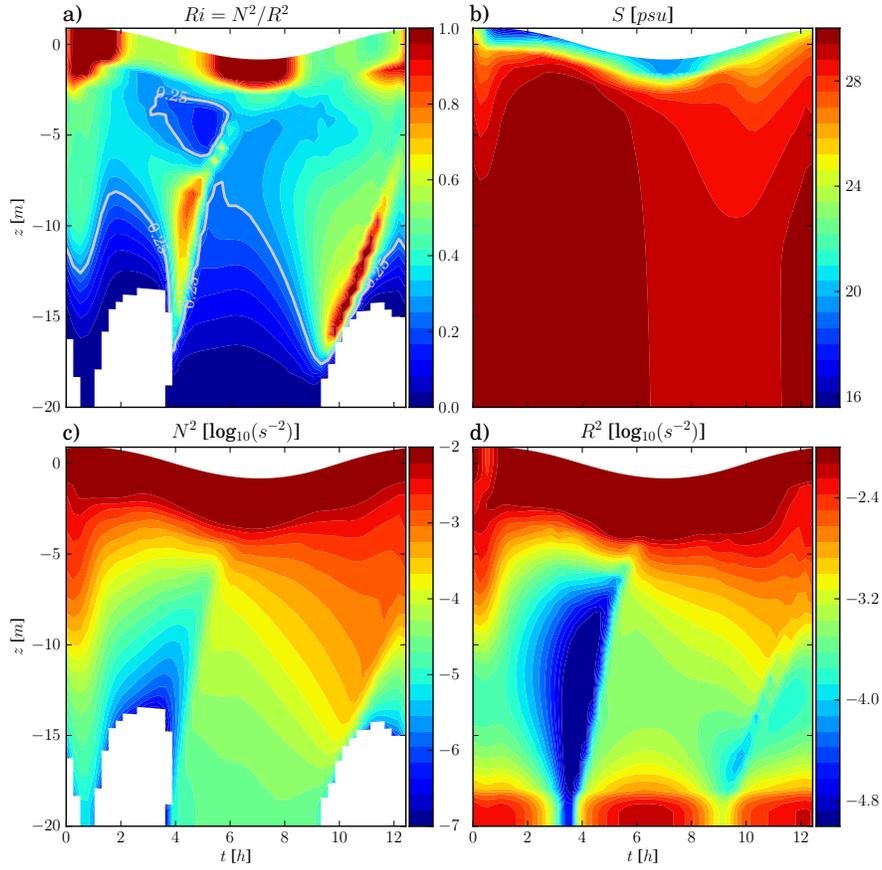


Figure 16: Temporal evolution of the stratification is station E (5 km offshore and 8 km north of the river mouth). a) Gradient Richardson number. The critical Richardson number $Ri_c = 0.25$ is indicated by a contour line. b) Salinity. c) Buoyancy frequency squared. d) Shear frequency squared. White patches indicate unstable stratification ($Ri, N^2 < 0$).