Université catholique de Louvain
Ecole Polytechnique de Louvain
Institut de Mécanique, Matériaux et Génie Civil

Development of a Baroclinic Discontinuous Galerkin Finite Element Model for Estuarine and Coastal Flows

---

Doctoral Dissertation presented by

Tuomas Kärnä

in partial fulfillment of the requirements
for the degree of
Doctor in Applied Sciences

Thesis committee:

Prof. Eric Deleersnijder, Université catholique de Louvain (Advisor)
Prof. Vincent Legat, Université catholique de Louvain (Advisor)
Prof. Hans Burchard, Leibniz Institute for Baltic Sea Research
Prof. Eric Delhez, Université de Liège
Prof. Emmanuel Hanert, Université catholique de Louvain
Prof. Jean-François Remacle, Université catholique de Louvain
Prof. Sandra Soares Frazao, Université catholique de Louvain
Prof. Grégoire Winckelmans, Université catholique de Louvain (President)

Louvain-la-Neuve, May 2012

My story in Louvain-la-Neuve began in January 2008. During these years, conducting the research and finalising this thesis would not have been possible without the collaboration and help of many people.

First of all, I am greatly indebted to my supervisors, Prof. Eric Deleersnijder and Prof. Vincent Legat for their advice and support along the way. Working as a part of the SLIM and TIMOTHY teams has been interesting and fruitful. This is all thanks to the (past and present) colleagues that I've had the privilege of working with, Prof. J.F. Remacle, Prof. Emmanuel Hanert, Olivier Gourgue, Richard Comblen, Olivier Lietaer, Jonathan Lambrechts, Benjamin de Brye, Anouk de Brauwere, Sébastien Blaise, Bruno Seny, Samuel Melchior, Christopher Thomas, Karim Slaoui, Alice Pestiaux and all the other people of the Euler building.

I am indebted to Prof. Hans Burchard for taking the time to work with me on the turbulence closure model coupling. I am also grateful the other members of the jury, Professors Eric Delhez, Sandra Soares Frazao and Grégoire Winckelmans for their insightful comments and remarks on the thesis.

Lastly I'd like to express my gratitude to my family, who have always been supportive and encouraged me on my path, wherever it has led me.

This work has been carried out with the support of the Belgian Fund for Research in Industry and Agriculture (FRIA), the project "Taking up the challenges of multi-scale marine modelling", funded by the Communauté Française de Belgique under contract ARC 10/15-028, and the project "Tracing and Integrated Modelling of Natural and Anthropogenic Effects on Hydrosystems" (TIMOTHY), IAP6.13, funded by the Belgian Federal Science Policy Office (BELSPO).

Contents

CHAPTER 1

Introduction

1.1 Coastal ocean

The coastal ocean extends from the outer edge of the continental shelf to the shoreline, and up to the furthest salt water influence in estuaries and rivers. It occupies only about 8% of Earth's surface, and 0.2 % of the global ocean volume, but it is the most biologically active part of the ocean. Shelf seas, coral reefs, estuaries, rivers, wetlands and shores are habitats for countless species, and are known for markedly high primary production and biodiversity. In general, it is the coastal ocean that most influences, and is most influenced by, human activity.

The dynamics of the coastal ocean differ markedly from those of the deep ocean. The horizontal scales of motion are much smaller than in the large ocean basins, and the coastlines impose a severe constraint on the flow. As continental shelf seas are relatively shallow (typically some 100 m deep), surface effects, such as wind stress or surface cooling or heating, affect a large part, if not the entire, water column. At the same time, the horizontal extent of coastal seas is large enough to feel the influence of Earth's rotation, in contrast to smaller basins.

The circulation in coastal seas is mainly forced by the prevailing large scale ocean currents, tides and winds. Vertical mixing, induced by frictional effects in the bottom and wind stress on the surface, plays an important role in the dynamics. Solar radiation and heat exchange with the atmosphere have a strong impact on the water temperature, and coastal waters can warm up considerably especially during summer months. The salinity distribution is governed by the exchange with the oceans and other basins, as well as riverine (or glacial) freshwater input and, to a lesser extent, evaporation and precip-

itation. Due to freshwater influence, less saline (i.e. lighter) water is often trapped in the surface layer, hence increasing stratification. Strong pycnocline inhibits mixing and thus plays an important role in the dynamics and primary production. As the pycnocline is also affected by the atmospheric heat fluxes, and destroyed by turbulence, the stratification conditions in coastal waters vary greatly in space and time.

Most of the human population is located in the vicinity of coastal areas. It is estimated that 60% of the world population lives within 60 km of the coasts, and this percentage is predicted to increase in the future due to growing urbanisation (Rao et al., 2008). The majority of the world's arable lands and urban areas are located in coastal plains and lower river valleys. Power plants and industrial complexes are often situated near coasts because of the proximity of cities, transportation and available cooling water (Walsh, 1988). The coastal ocean is heavily exploited by our societies for food, natural resources and transportation.

High population density and intensive industrial and agricultural activities have a strong impact on the water quality and marine ecosystems. The coastal ocean is particularly vulnerable to the effects of climate change and changes in land use and freshwater consumption. In many parts of the world, the coastal ecosystems are stressed and degraded, largely due to anthropogenic pressure (Rao et al., 2008). Heavy metals, petrochemicals, synthetic chemicals (such as organochlorides) and radionuclides find their way to the riverine and marine environment through both coastal and atmospheric sources. Such pollutants are toxic to many species, affecting the survival or fertility of marine organisms. Furthermore, the pollutants may be assimilated in the organisms and can accumulate in the food web, causing more severe consequences on higher trophic levels. Elevated levels of nutrients, due to agriculture, deforestation or insufficient waste water treatment, boost the (phytoplankton) primary production in coastal waters, that can reach high levels compared to the pristine state (Jørgensen and Richardson, 1996). Moreover, changes in land use tend to intensify erosion and thus increase water turbidity and sedimentation in coastal areas (Wolanski, 2007). Physical transport of pollutants, nutrients and suspended particulate matter, and their interaction with the marine ecosystem have become increasingly complex problems in the coastal ocean.

The state of the coastal ocean also has (direct or indirect) effects on the economy, human health, protection against natural hazards and recreation. For sustainable management of the marine environment, it is therefore necessary to gain a better understanding of the most significant physical, chemical and biological processes that take place at various spatial and temporal scales in the coastal ocean.

1.2 Necessity of numerical ocean modelling

Numerical modelling has become an indispensable tool in ocean sciences. The hydrodynamic equations are too complex to be analytically tractable. Moreover, considering the vastness of marine systems and the wide spectrum of relevant time and length scales, observations are too scarce to provide a complete picture of the current state of the oceans.

Numerical marine models solve the primitive equations with appropriate approximations and parametrisations of unresolved physics. Consequently they represent (up to a certain degree) our current understanding of the essential ocean dynamics. Numerical models can therefore be used in conjunction with measurements to fill in the gaps in observational data to build plausible realisations of the ocean state.

Numerical modelling also serves to advance basic research as it can improve our understanding of the physical processes of the oceans. By designing specific simulations, which would be impossible to achieve or observe in reality, it is possible to isolate certain processes and analyse their driving mechanisms and effects in detail.

Perhaps most importantly, marine models allow to predict the future state of the ocean, providing valuable information for decision making and risk management. Predictions range from daily forecasts and natural hazard warnings to climate projections (or hindcasts) covering hundreds or even thousands of years.

In terms of environmental management, one central goal is to assess the anthropogenic impacts on the marine environment on various temporal and spatial scales. In such work, numerical marine modelling plays a key role. Knowledge of currents and diffusion parameters in the coastal ocean is essential for predicting the fate of contaminants and the response of marine ecosystems. In general, processes driving the flow on the whole coastal sea must be understood, before the impact of contaminants on biota and water quality can be predicted. As predicting the evolution of complex systems, such as food webs, always involves uncertainty, various statistical approaches, such as data assimilation and scenario analysis, are being developed in combination with computational fluid dynamics.

In this thesis, the focus is on modelling the hydrodynamics of the coastal ocean. However, reactive transport modules can be introduced in the discussed hydrodynamic models, thus permitting biogeochemical modelling of desired complexity.

1.3 Challenges in multi-scale coastal modelling

Whilst the first ocean models were designed to simulate the global ocean circulation (e.g. Bryan, 1969), there has always been interest in developing regional and coastal models. Open source marine models aimed specifically at regional applications include ROMS (Shchepetkin and McWilliams, 2005), GETM (Burchard and Bolding, 2002), Delft3D (Del, 2012), FVCOM (Chen et al., 2006) and SELFE (Zhang and Baptista, 2008). General circulation models (e.g. MITgcm, Marshall et al. 2004) and large scale ocean models (e.g. POM, Mellor 2004) are also often applied to regional settings. Despite the many efforts in past decades, several challenges still remain in modelling coastal flows.

1.3.1 Characteristics of coastal flows

Ocean currents and tidal signals tend to be amplified in shallow regions and funnel-like embayments due to the contraction of the flow. As a consequence, estuaries and coastal areas exhibit stronger currents and more rapid dynamics than those found in the deep sea. The water density typically varies considerably due to riverine freshwater input and, to a lesser extent, atmospheric heat fluxes. The strong density gradients are, on the one hand, modified by advection processes, but, on the other hand, affect the velocity field through the internal pressure gradient. Vertical mixing, generated by winds and bottom friction, yet inhibited by stratification, also plays an important role. Combined with rotational effects, coastal flows exhibit complex and highly non-linear behaviour.

The region of freshwater influence (ROFI Simpson et al., 1993) is a regime where riverine buoyancy input is comparable with, or greater than, seasonal heat induced buoyancy. ROFIs are characterised by a competition between buoyancy and stirring induced by winds or currents. Under the influence of the Coriolis force, river plumes tend to turn to the right (in the Northern hemisphere), forming coastal currents that can retain their characteristics for hundreds of kilometres. Consequently they are extremely important in transporting sediments, nutrients, and contaminants in coastal seas.

Highly dynamic coastal flows pose challenges to numerical modelling. Strong advective velocities and steep density gradients can lead to stability issues, which may be difficult to amend. Excessive numerical dissipation, either inherent to some model types or introduced to ensure stability, can diffuse the density fronts, changing the flow dynamics. As the density is governed by temperature and salinity, the quality of these fields have a direct impact on accuracy and numerical stability, stressing the importance of robust tracer advection schemes.

In coastal waters, the free surface movement cannot be assumed to be negligible compared to the total water depth, an assumption that is often made in order to simplify the numerical methods. In sufficiently shallow water, the free surface movement becomes a dominant factor. Ultimately, tidal areas often feature sand banks, salt marshes or mudflats that are exposed at low water and submerged at high water. Although currents in such intertidal regions remain relatively small (due to bottom friction), the varying extent of the wetted area has an impact on the flow that cannot be neglected. Large intertidal areas tend to slow down the tidal signal propagation in estuaries (Friedrichs, 2010) and also serve as a reservoir for tracers.

Representing shoreline movement in Eulerian shallow water models turns out to be a complicated task. Essentially the problem arises from the fact that numerical shallow water models break down at zero depth and there is no automatic mechanism that prevents negative depths from arising. A considerable number of wetting-drying methods is present in the literature, with various approaches aiming to overcome the problem. Typically, the proposed methods are specific for a single model and compatible only with a certain (implicit or explicit) time integration scheme. Their properties also vary; some impose additional time step restrictions, some have poor conservation properties or exhibit spurious behaviour and some rely on additional dissipation or damping to remain stable. Moreover, most of the existing methods have been developed for FD or FV methods, while there are not as many candidates for FE models. Consequently it seems that wetting-drying methods are still case specific and it is difficult to propose a general purpose method.

1.3.2 Multi-scale models and unstructured meshes

Traditionally, there have been separate model classes for rivers, estuaries and shelf seas. However, as these domains are tightly linked, separate models are often insufficient to capture all the interactions across the different length scales. As the spatial resolution of numerical models is constantly increasing due to advances in computer technology, the consequences of ignoring such multi-scale interaction are becoming more and more apparent.

Most marine models solve the primitive equations on a structured grid using the finite difference (FD) method. The advantages of such an approach are evident. Using finite differences on a regular and orthogonal grid yields relatively simple and efficient numerical methods. The disadvantage is the relative inflexibility of the grid. Even on a modern deformed orthogonal grid, the spatial resolution cannot be increased arbitrarily. Although nested grids can be used to increase resolution locally, nesting can introduce significant errors at the model interfaces. Moreover, it is questionable whether nested grids could be used in complex domains such as multiple estuary systems or reefs (e.g. Lambrechts et al., 2008b). Furthermore, structured grids have the

disadvantage of representing coastlines as a staircase-like boundary which can have a significant impact on the flow (Adcroft and Marshall, 1998).

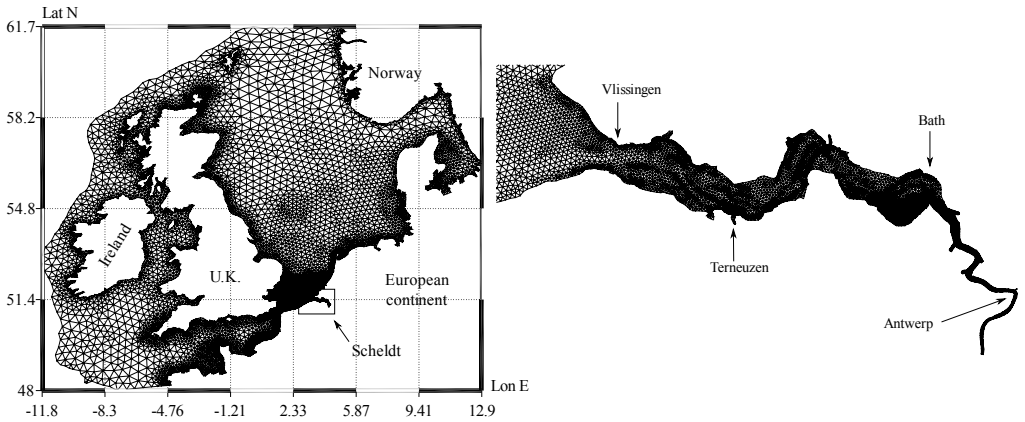


Figure 1.1: Top: An unstructured mesh of the Scheldt estuary and the adjacent North-western European Continental Shelf Sea.

Due to the disadvantages of structured grids, unstructured mesh models have gained more and more attention in recent years (e.g. Deleersnijder and Lermusiaux, 2008). The inherent flexibility of unstructured meshes is indeed crucial for multi-scale coastal modelling. The spatial resolution can be increased arbitrarily where needed (restricted only by the computational cost), and coastlines are represented more realistically as a piecewise linear boundary. Consequently it becomes feasible to capture river networks, estuaries, archipelagos, straits, reefs, and shelf seas with sufficient resolution in a single model (Figure 1.1). In such applications the element size can vary over several orders of magnitude.

Resolution can also be adapted based on other criteria, such as topographical features or numerical error estimates. Typical examples include mesh refinement in areas of steep bathymetry gradients (to overcome pressure gradient errors), near river mouths (to resolve small scale flow features), or in areas of high posteriori error. Moreover, with adaptive mesh techniques, it is possible to minimise numerical errors or to track complex phenomena evolving in time (e.g. cyclones or fronts). Mesh adaptivity, however, is not dealt with in this work.

Another advantage of unstructured meshes is the ease of mesh generation. With recent advances in meshing algorithms, creating unstructured meshes for marine applications is no longer a major obstacle. Versatile mesh generation software such as GMSH (Geuzaine and Remacle, 2009) now exist which have specific tools for marine modelling (Lambrechts et al., 2008a). Consequently, dealing with multi-scale domains is, in general, easier with unstructured meshes than with nested or curvilinear grids.

Some structured grid models have been extended to unstructured meshes by assuming that the mesh is orthogonal[1] (e.g. TRIM, Casulli and Cheng 1992, and UnTRIM, Casulli and Walters 2000, model classes). The drawback is that, in practical applications, the orthogonality criterion is very difficult to meet, introducing an additional disretisation error. In this work we will concentrate on general unstructured mesh models, that do not require orthogonality and are valid also on highly anisotropic meshes.

Unstructured mesh models are usually based on the finite volume (FV) or the finite element (FE) method. In the FV method, the conservation law is imposed on the elements, and communication between elements is achieved through numerical fluxes. Consequently FV methods are locally conservative by construction. Moreover, monotonicity can be ensured with slope (or flux) limiters. Both of these properties are essential for advection dominated problems (Cockburn and Shu, 2001). FE methods, on the other hand, provide high order accuracy through more elaborate polynomial basis functions.

In the continuous Galerkin (CG) FE formulation, the fields are continuous between elements. With the discontinuous Galerkin (DG) method, the continuity requirement is relaxed and numerical inter-element fluxes are introduced similarly to the FV method. As such, the DG method can be seen as a hybrid of the FV and continuous FE approaches.

DG methods are promising because they are locally conservative, while allow for high order accuracy. As in the FV method, monotonicity is achievable with slope limiters, although they tend to be more complicated for high order discretisation. Moreover, as the elements are discontinuous in DG, it is straightforward to construct hybrid meshes where the element type (e.g. triangle or quadrangle) or the accuracy (degree of the basis functions) varies. This property makes DG ideal for *hp*-adaptivity.

A major drawback of unstructured grid models is their computational cost. The unstructured nature of the mesh is also reflected in numerical algorithms and memory structure that are bound to be more complicated than with regular grid FD (or FV) models. Furthermore, FE methods require numerical integration over the elements which introduces an additional computational overhead. As such, on similar configurations, unstructured grid models tend to be slower than their structured grid counterparts. Consequently, improving computational efficiency is a major objective of current unstructured mesh model development. One possible route is to improve the time integration methods to allow larger time steps, for instance by moving towards implicit schemes or by using a Lagrangian-Eulerian advection method.

[1]Here orthogonality means that element centres can be chosen in such a way that each interface is perpendicular to the line connecting the neighbouring centres.

1.4 The SLIM project

This thesis deals with the development of discontinuous Galerkin shallow water models, with focus on multi-scale shelf sea-estuary-river network modelling. The work is carried out in the framework of the Second-generation Louvain-la-Neuve Ice-ocean Model (SLIM[2]), developed at the Université catholique de Louvain, Belgium. SLIM consists of 1D section averaged (de Brye et al., 2010), 2D depth averaged (Gourgue et al., 2009; de Brye et al., 2010; Comblen et al., 2010b; Kärnä et al., 2010, 2011) and 3D shallow water models (White et al., 2008a,b; Blaise et al., 2010; Comblen et al., 2010a; Kärnä et al., 2012; Kärnä et al.).

As a part of the inter-university TIMOTHY[3] project, the 2D depth averaged model was applied to simulating the Scheldt estuary, located between the Netherlands and Belgium. Due to heavy industrial activity and a high population density in the basin, the water quality in the Scheldt river and estuary is poor, with elevated concentrations of heavy metals, pathogenic micro-organisms and other contaminants (Baeyens et al., 1997; de Brauwere et al., 2011b). Consequently there is a clear need to better understand the sources and the fate of the contaminants in the Scheldt continuum, as well as the governing biochemical dynamics.

The modelling domain extends from the shelf break to the North Sea, and further into the Scheldt estuary and the river network (Figure 1.1). Upstream of Antwerp, the 2D model is replaced by a 1D section averaged river model. Extending the domain far up- and downstream of the estuary is advantageous as the boundary conditions can be reliably assigned using global tidal model data at the shelf break and discharge data at the end of the tidal rivers.

In Kärnä et al. (2010), tracer advection and diffusion properties of the model were validated against analytical solutions in simplified geometries. The hydrodynamics of the Scheldt was validated with water surface and salinity measurements in de Brye et al. (2010). The water renewal time scales were estimated in de Brauwere et al. (2011a) and de Brye et al. (2012), making use of the Constituent-oriented Age and Residence time Theory (CART[4]). The fate of fecal bacteria (*E. coli*) in the Scheldt was dealt with in de Brauwere et al. (2009); de Brauwere et al. (2011b).

1.5 Scope of this thesis

This thesis addresses two key issues of DG-FE marine modelling: wetting-drying in depth-averaged 2D models and 3D modelling of buoyancy driven

[2] http://www.climate.be/SLIM

[3] Tracing and Integrated Modeling of Natural and Anthropogenic Effects on Hydrosystems : The Scheldt River basin and adjacent coastal North Sea http://www.climate.be/TIMOTHY

[4] http://www.climate.be/CART

flows. In **Chapter 2** a novel implicit wetting-drying method is presented with a focus on the Scheldt application. A 3D baroclinic model is presented in **Chapter 3** and **Chapter 4**.

1.5.1 Implicit wetting-drying method

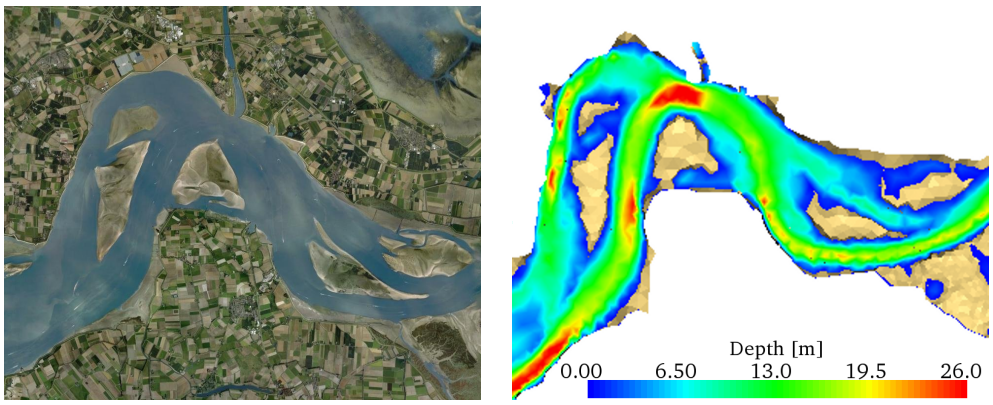


Figure 1.2: Exposed sandbanks in the Scheldt estuary at low tide. Left: A satellite image of the estuary near Hansweert (source: Google Earth) Right: The same area as modelled by SLIM.

As the Scheldt estuary features relatively large intertidal sand banks and salt marshes, wetting-drying has to be taken into account in the numerical model (Figure 1.2). A novel wetting-drying method developed for the depth averaged 2D model is presented in **Chapter 2**.

The central idea of the proposed method is quite different from those commonly in use: positive water depth is guaranteed by redefining water depth in such a way that it cannot reach zero. This is achieved by artificially lowering the bed as the water level falls. It follows that only a small modification in the governing equations is required. As the method can be formulated on the level of the equations, it is not specific for a single model type but is applicable to a wide range of models. The method is mass conservative and consistent with tracers (i.e. a constant tracer field is preserved), both of which are important properties for long term environmental modelling.

Usually wetting-drying methods introduce a threshold depth that is used to identify dry elements or nodes. Consequently, the numerical scheme abruptly changes at the threshold, which leads to unstable oscillations if the (explicit or implicit) time step is too large. This drawback also applies for the first flux-limiting wetting-drying method developed for SLIM (Gourgue et al., 2009). In the proposed wetting-drying method, the transition between wet and dry areas is smooth by construction. The smoothness is controlled by a parameter, which is relatively easy to estimate.

Due to the gradual change between wet and dry areas, it is possible to compute the Jacobian of the numerical system, which permits fully implicit

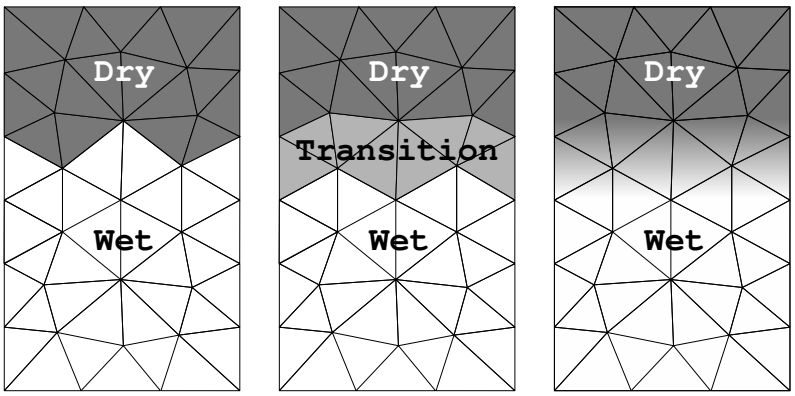


Figure 1.3: Illustration of different wetting-drying approaches. Left: Elements are inactivated when they reach dry state. Centre: Nodal fluxes are inactivated at dry state. Right: Gradual wet-dry transition.

time integration. Consequently the time step is no longer restricted by the CFL (Courant–Friedrichs–Lewy) condition, which leads to significantly smaller computational cost. In the Scheldt application the largest permissible explicit time step is roughly 1 s, whilst in the implicit model a time step of 10 min to 20 min is commonly used. As a result the total CPU cost of the implicit model is two orders of magnitude smaller, allowing the various long term simulations discussed in the references above to be carried out.

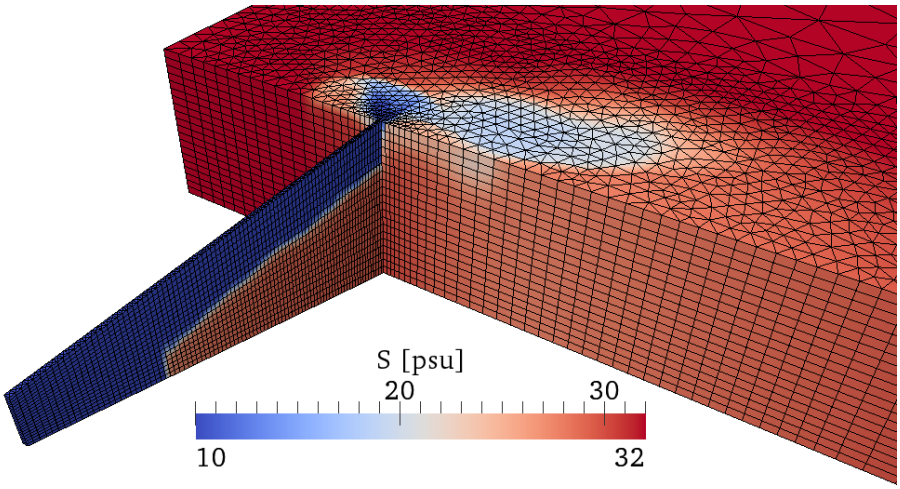


Figure 1.4: Simulation of the Rhine river plume in an idealised geometry.

1.5.2 Development of a 3D baroclinic model

The latter part of the thesis is devoted to the development of a hydrostatic, baroclinic discontinuous Galerkin model. The model uses prismatic elements

that have become customary in unstructured grid marine models (White et al., 2008a; Wang et al., 2008; Blaise et al., 2010). In terms of the vertical discretisation, a terrain following mesh is used, although the computational kernel also supports equipotential grids and their generalisations. An example mesh is shown in Figure 1.4. To accommodate free surface fluctuation, the mesh moves in the vertical direction. An Arbitrary Lagrangian Eulerian (ALE, Donea et al., 2004) formulation is adopted for solving the equations in the moving frame of reference.

Because the 3D baroclinic primitive equations are much more complex than the 2D shallow water equations, fully implicit time integration schemes are out of reach for the moment. For computational efficiency, a mode splitting approach is chosen, where the dynamics are split to a fast 2D (external) mode and a slower 3D (internal) mode. The fast propagating surface gravity waves are solved with inexpensive 2D equations, thus circumventing the the most restrictive CFL condition. Furthermore, vertical diffusion is treated semi-implicitly to overcome certain stability issues due to high vertical resolution. The next most restrictive conditions, related to advection and internal wave propagation, have to be satisfied.

Mode splitting techniques can be divided in two main classes depending on whether the free surface is treated explicitly or implicitly. In *split-explicit* approach (Gadd, 1978; Blumberg and Mellor, 1999; Killworth et al., 1991), the 2D mode is marched in time with an explicit scheme using a high temporal resolution. This method is used in ROMS (Shchepetkin and McWilliams, 2005), GETM (Burchard and Bolding, 2002), POM (Mellor, 2004), MOM (Griffies et al., 2001) and FVCOM (Chen et al., 2006), among others. *Split-implicit* models (e.g. Dukowicz and Smith, 1994), on the other hand, solve the free surface equation (or the complete 2D shallow water equations) implicitly with the same time step as the 3D mode. MITgcm (Marshall et al., 2004; Campin et al., 2004) and FEOM (Wang, 2007) belong to this category, as well as TRIM (Casulli and Cheng, 1992), UnTRIM (Casulli and Walters, 2000) and SELFE (Zhang and Baptista, 2008).

The 3D model development presented here is a continuation to the work of White et al. (2008a,b) and Blaise et al. (2010); Comblen et al. (2010a). The first 3D SLIM implementation was the barotropic model by White et al. (2008b). A continuous Galerkin discretisation, with linear non-conforming elements, were used in the horizontal direction, while a linear DG discretisation was applied in the vertical direction. Most 3D terms were advanced with a first order explicit scheme, except for vertical diffusion that was semi-implicit. The external gravity waves were treated with the split-implicit method, and the free surface movement was taken into account with a conservative ALE formulation in the 3D mode. In terms of vertical mixing, a simple parametrisation was used, although in Blaise et al. (2007) Mellor-Yamada level 2.5 turbulence closure was introduced.

White et al. (2008a) deals with tracer conservation issues, and especially the discrete compatibility of the elevation, continuity and tracer equations. In addition to global tracer conservation, in the absence of sources or sinks, a uniform tracer field should remain constant throughout the simulation. In Griffies et al. (2001), this property is called local conservation, while Shchepetkin and McWilliams (2005) call it constancy preserving property. Owing to the conservative ALE formulation and the time integration scheme, the model of White et al. (2008a) is indeed both globally conservative and constancy preserving.

A DG discretisation of the baroclinic equations is presented in Blaise et al. (2010), while the associated implicit-explicit (IMEX) Runge-Kutta time integration method is detailed in Comblen et al. (2010a). Also in this scheme, the 2D mode is treated implicitly. Regarding the 3D mode, the external pressure gradient, Coriolis forcing, vertical advection and diffusion are treated implicitly. Furthermore, additional Lagrange multipliers are introduced to ensure the consistency of the 2D and the 3D modes. The major drawback of this model is the computational cost, due to the chosen time integration method. Also, due to a non-conservative ALE formulation, the model is not strictly mass conservative, yet constancy preserving. The model is validated with mildly baroclinic internal gravity wave test case, using both linear and quadratic elements. However, it is not applicable to strongly baroclinic flows, because the tracer advection scheme is not stabilised. Consequently, in the presence of strong salinity/temperature gradients, spurious extrema may arise in the corresponding tracer fields, eventually rendering the model unstable.

The 3D model presented here relies on a full DG spatial discretisation, similar to that of Blaise et al. (2010). In contrast to the previous models, however, the split-explicit paradigm is adopted here, following Shchepetkin and McWilliams (2005). Moreover, of the 3D terms, only vertical diffusion is treated implicitly. A major advantage of explicit time integration is that the model is easy to parallelise and scales well in parallel applications. The latter is vital for realistic 3D simulations where the use of computing clusters is unavoidable.

For parallel computing, the domain is partitioned in the horizontal direction so that the prismatic columns are always kept in the same partition. Therefore the implicit vertical diffusion, which does not include any lateral exchange, can be solved locally column by column.

As mentioned earlier, the quality of the temperature and salinity fields is crucial in baroclinic applications. Here, monotonic tracer advection is achieved by means of a slope limiter, as usual in FV and DG schemes (Cockburn, 2003; Kuzmin, 2010; Aizinger, 2011). The limiter filters out spurious tracer maxima and ensure stability in the presence of strong density gradients. The solver is designed in such a way that both tracer conservation and consistency can be achieved, although the latter is eventually satisfied only approximately.

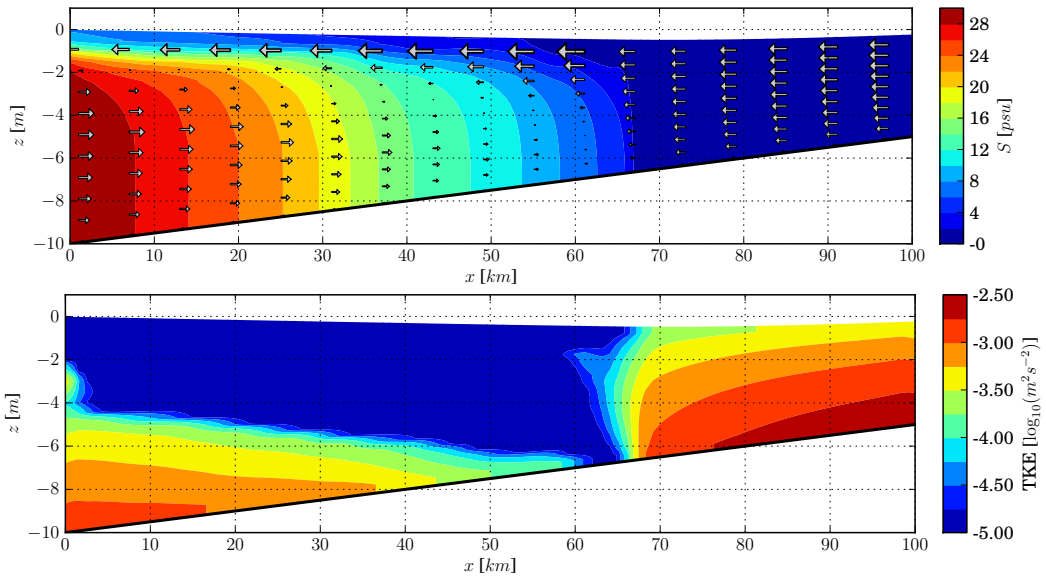


Figure 1.5: Idealised estuary simulation at the beginning of the rising tide. Top: Salinity and horizontal velocity. Bottom: Turbulent kinetic energy. The two-way flow typical to estuarine circulation is clearly visible. Turbulent mixing is inhibited in strongly stratified areas.

To account for vertical mixing, the 3D model uses the General Ocean Turbulence Model (GOTM[5]), which is a FD 1D water column model. The coupling is presented in **Chapter 3**. Owing to the vertically aligned 3D mesh, an online coupling is possible once a robust mapping between the 3D DG-FE fields and 1D vertical FD grid is created. The coupled model is validated with standard turbulence closure benchmarks, as well as an idealised estuary simulation (Warner et al., 2005), where the main features of estuarine circulation are well reproduced (Figure 1.5).

In **Chapter 4** the full baroclinic model is presented. The DG discretisation and the split-explicit time integration methods are described. The model is validated first with simplified flows to test the mode-splitting, conservation properties and performance under gravitational adjustment. Finally a simulation of the Rhine river plume is carried out in a simplified geometry (Figure 1.4).

[5] http://www.gotm.net/

Supporting publications

T. Kärnä, E. Deleersnijder, and A. de Brauwere. Simple test cases for validating a finite element unstructured grid fecal bacteria transport model. *Applied Mathematical Modelling*, 34(10):3055–3070, 2010. ISSN 0307-904X. doi: 10.1016/j.apm.2010.01.012.

T. Kärnä, B. de Brye, O. Gourgue, J. Lambrechts, R. Comblen, V. Legat, and E. Deleersnijder. A fully implicit wetting-drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):509–524, 2011. ISSN 0045-7825.
doi: 10.1016/j.cma.2010.07.001.

T. Kärnä, V. Legat, E. Deleersnijder, and H. Burchard. Coupling of a discontinuous Galerkin finite element marine model with a finite difference turbulence closure model. *Ocean Modelling*, 47:55–64, 2012. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2012.01.001.

T. Kärnä, V. Legat, and E. Deleersnijder. Baroclinic discontinuous Galerkin finite element model for coastal flows. *Ocean modelling*, submitted.

A. de Brauwere, F. De Ridder, O. Gourgue, J. Lambrechts, R. Comblen, R. Pintelon, J. Passerat, P. Servais, M. Elskens, W. Baeyens, **T. Kärnä**, B. de Brye, and E. Deleersnijder. Design of a sampling strategy to optimally calibrate a reactive transport model: Exploring the potential for Escherichia coli in the Scheldt Estuary. *Environmental Modelling & Software*, 24(8):969–981, Aug 2009. ISSN 1364-8152.
doi: 10.1016/j.envsoft.2009.02.004.

B. de Brye, A. de Brauwere, O. Gourgue, **T. Kärnä**, J. Lambrechts, R. Comblen, and E. Deleersnijder. A finite-element, multi-scale model of the Scheldt tributaries, river, estuary and ROFI. *Coastal Engineering*, 57 (9):850–863, 2010. ISSN 0378-3839.
doi: 10.1016/j.coastaleng.2010.04.001.

B. de Brye, S. Schellen, M. Sassi, B. Vermeulen, **T. Kärnä**, E. Deleersnijder, and T. Hoitink. Preliminary results of a finite-element, multi-scale model of the Mahakam Delta (Indonesia). *Ocean Dynamics*, pages 1–14, 2011. ISSN 1616-7341.
doi: 10.1007/s10236-011-0410-y.

O. Gourgue, R. Comblen, J. Lambrechts, **T. Kärnä**, V. Legat, and E. Deleersnijder. A flux-limiting wetting-drying method for finite-element shallow-water models, with application to the Scheldt Estuary. *Advances in Water Resources*, 32(12):1726–1739, Dec 2009. ISSN 0309-1708.
doi: 10.1016/j.advwatres.2009.09.005.

CHAPTER 2

Implicit wetting-drying method

This Chapter reproduces an updated version of the following paper, first submitted on 23 February 2010:

T. Kärnä, B. de Brye, O. Gourgue, J. Lambrechts, R. Comblen, V. Legat, and E. Deleersnijder. A fully implicit wetting-drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary. *Computer Methods in Applied Mechanics and Engineering*, 200 (5-8):509–524, 2011. ISSN 0045-7825.
doi: 10.1016/j.cma.2010.07.001.

Abstract

Resolving the shoreline undulation due to tidal excursion is a crucial part of modelling water flow in estuaries and coastal areas. Nevertheless, maintaining positive water column depth and numerical stability has proved out to be a very difficult task that requires special attention. In this paper we propose a novel wetting-drying method in which the position of the sea bed is allowed to fluctuate in drying areas. The method is implemented in a Discontinuous Galerkin Finite Element Model (DG-FEM). Unlike most methods in the literature our method is compatible with fully implicit time-marching schemes, thus reducing the overall computational cost significantly. Moreover, global and local mass conservation is guaranteed which is crucial for long term environmental

applications. In addition consistency with tracer equation is also ensured. The performance of the proposed method is demonstrated with a set of test cases as well as a real-world application to the Scheldt Estuary. Due to the implicit time integration, the computational cost in the Scheldt application is reduced by two orders of magnitude. Although a DG-FEM implementation is presented here, the wetting-drying method is applicable to a wide variety of shallow water models.

2.1 Introduction

Most coastal areas of continental shelf seas are significantly influenced by tides. When approaching the coast, the tidal signal tends to amplify, especially in funnel-shaped embayments where the tidal range may reach considerable magnitudes. Combined with the fact that many estuaries and embayments also feature gradually sloping bathymetry, the total area submerged under water may vary significantly during the tidal cycle.

Any hydrodynamical model that is being applied to such a tidally influenced domain needs to correctly take into account the sequential exposure and submerging of the seabed.

Ever since the 1970s hydrodynamical models have been equipped with wetting-drying (WD) algorithms. However, the multitude of WD methods found in the literature reveals that numerical modelling of shoreline undulation is far from being trivial. In what follows, we will mostly concentrate on methodology applicable to unstructured grid models, i.e. finite element (FE) and finite volume (FV) formulations.

Perhaps the most natural approach would be to track the WD interface in time and move the boundary nodes, or deform the entire mesh, accordingly. These moving mesh methods probably yield the most appropriate description of the wetting-drying process, but are faced with some difficulties: First of all, one needs to come up with a parametrisation for moving the domain boundary as a function of the flow in boundary elements (Marchandise and Remacle, 2006; Sobey, 2009). Secondly, sophisticated algorithms are needed for maintaining good mesh quality in long simulations. However, re-meshing becomes increasingly difficult if changes in the topography, such as emerging ponds or islands, need to be taken into account. Nevertheless, the greatest drawback is that deforming the mesh is computationally expensive. For the latter reason, deforming mesh wetting-drying has seldom been applied to real-world problems (Yuan et al., 2008; Zheng et al., 2003; Zijlema and Stelling, 2008).

Most of the available WD methods have been developed for fixed meshes. The fixed mesh approaches can be further sub-divided into two main categories. In the first category, either nodes or entire elements are deactivated when becoming dry. The dry state is detected with special criteria usually based on total water depth. The first attempts were element reduction methods, where entire elements are tagged as “dry” or “wet” and dry elements are

excluded from the computational domain. This implies that the WD interface can only be located at element edges which has an impact on the boundary layer behaviour (Bates and Hervouet, 1999). Moreover, the sudden inclusion/exclusion of elements may break mass and momentum conservation and also trigger numerical instability (van't Hof and Vollebregt, 2005).

In order to describe a coastline that does not coincide with the element edges, most authors introduce transition (or partially wet) elements that require special treatment. By far the most popular method is to leave a thin layer of water in the dry areas to ensure positive water depth and maintain a continuum across the WD interface (Bates and Hervouet, 1999; Bunya et al., 2009; Gourgue et al., 2009; Nikolos and Delis, 2009). The transition elements are then defined as elements for which some nodes, but not all are dry, i.e. have water level under a prescribed threshold value.

The main difficulty in thin layer methods lies in the way that the transition elements are being treated. The transition elements typically have "hanging nodes" that remain on a higher level than the free surface. In such case the transition elements have spurious water surface slope and thus experience an artificial pressure gradient that tends to drive the water down (Heniche et al., 2000). Without any treatment, the hanging nodes will eventually dry out. Commonly the transition elements are being explicitly detected and the pressure gradient term is being ignored. Sometimes the elements are further divided into "dam-break" and "flooding" types, where the pressure gradient term needs to be cancelled only in the latter case (Bates and Hervouet, 1999; Bunya et al., 2009). Typically several more or less complicated rules are needed for detecting the transitory (or dry) elements and the local physics is changed accordingly. However, such discontinuous switches (such as cancelling the pressure gradient or outward fluxes under a threshold water depth) render these methods highly non-linear and may introduce oscillations and numerical instability. Indeed it is not uncommon to increase bottom friction or dissipation to circumvent such problems. Nevertheless, thin layer methods can produce accurate results even in very difficult applications.

The thin layer methods rely on explicit detection of dry elements, so the WD front can propagate only by one element per time step. Moreover, the time steps are usually constrained by the Courant–Friedrichs–Lewy (CFL) criterion, which is needed to ensure positive water depth or numerical stability (Casulli, 2009). Consequently the computational cost grows significantly, especially in large scale simulations that incorporate both deep and shallow areas (Stelling and Duinmeijer, 2003). Due to the time step constraints, explicit time integration is commonly used, although semi-implicit schemes also exist. Fully implicit time marching is generally not applicable as it requires that the Jacobian of the system can be computed (exactly or approximately), which is not possible for the discontinuous switches.

Another class of fixed grid WD methods is the artificial porosity approach. In these methods, the bed is assumed to be porous and non-zero water fluxes

are allowed for negative depths. These methods include an additional porous layer (Ip et al., 1998; Heniche et al., 2000; Nielsen and Apelt, 2003; van't Hof and Vollebregt, 2005; Yuan et al., 2008), narrow connecting channels (Kennedy et al., 2000; Jiang and Wai, 2005) and also bear close resemblance to sub-grid scale bathymetry (Defina, 2000; Bates and Hervouet, 1999) methods. The main advantage is that the artificial pressure gradient problem can be naturally circumvented. With porous media, free surface will fall under the bed in drying phase and eventually the spurious surface slope will disappear. Naturally the drawback is that "virtual water" is generated as depth attains negative value, although mass conservative formulations are reported to exist (Jiang and Wai, 2005; Defina, 2000). The key advantage of porosity methods is that the transition between wet and dry areas is smooth and the modifications can be expressed in the primitive equations. Due to the latter reason many methods of this type are compatible with semi-implicit or implicit (Ip et al., 1998; Heniche et al., 2000; van't Hof and Vollebregt, 2005) time integration, reducing computational cost significantly.

Some WD methods are exploiting the fact that the WD process is essentially dominated by the pressure gradient and bottom friction, which can be used to simplify the equations; see for example (Ip et al., 1998; Burchard et al., 2004). Requiring a balance between the two will ultimately lead to the so called diffusive wave approximation of the shallow water equation (Alonso et al., 2008; Santillana and Dawson, 2010) that can be used to model the WD interface. Such an approach, however, is not applicable to coastal waters in general.

In this paper we present an alternative "negative-depth" implicit WD formulation for FE shallow water models. However, in contrast to the porous media methods, our method is based on the idea that the bed is allowed to move in time as water elevation drops, which leads to a very similar formulation but without the need to introduce the concept of porosity. This yields simpler implementation and most importantly the proposed method has only one unknown parameter whose value can be estimated fairly reliably.

Although the notion of moving the bathymetry may seem unusual, similar modifications have been applied locally in certain WD methods. An example of an FD model where bathymetry is temporally modified for the computation of elevation gradient is presented in Burchard et al. (2004). Similar modifications in FV models can be found in Castro et al. (2005); Nikolos and Delis (2009). In all these references the motivation to modify the bathymetry is to avoid spurious pressure forces at the WD interface, precisely as in our case. The major difference is that here the modification is formulated already in the primitive equations.

As our goal is to perform long term environmental simulations, WD treatment should not jeopardise mass conservation nor introduce spurious tracer transport. Therefore special care is taken to guarantee strict mass conservation

property. In addition, a consistent tracer equation and its FE implementation is presented.

The proposed WD methods is implemented in the FE Second-generation Louvain-la-Neuve Ice-ocean Model (SLIM)[1] (Gourgue et al., 2009; de Brye et al., 2010) and its validity is demonstrated with a set of standard numerical tests. However, in many cases, the test cases have proved out to be too easy to solve compared to simulations with real-world bathymetry, and thus more challenging test cases are also introduced.

The paper is organised as follows. The shallow water equations with the moving bathymetry are introduced in Section 2.2 and a fully implicit FE implementation is presented in Section 2.3. A consistent tracer equation is then presented in Section 2.4 followed by numerical tests in Section 2.5 and a real-world application in Section 2.6.

2.2 Shallow water equations allowing moving bathymetry

Given Cartesian horizontal coordinates $\boldsymbol{x}_h = [x, y]^T$, the depth averaged shallow water equations (SWEs) in non-conservative form are:

$$\frac{\partial \eta}{\partial t} + \boldsymbol{\nabla} \cdot (H\ \bar{\boldsymbol{u}}) = 0\ , \tag{2.1}$$

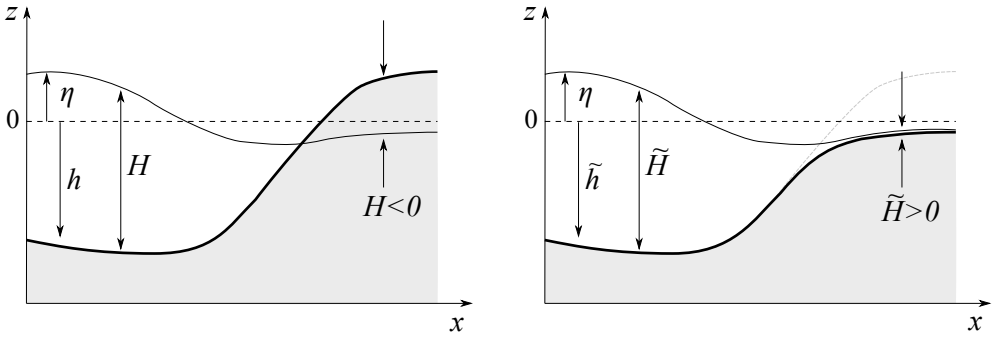
$$\frac{\partial \bar{\boldsymbol{u}}}{\partial t} + (\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla})\, \bar{\boldsymbol{u}} + F_c \boldsymbol{e}_z \times \bar{\boldsymbol{u}} + g \boldsymbol{\nabla} \eta = \frac{\boldsymbol{\tau}_s - \boldsymbol{\tau}_b}{\rho H}\ . \tag{2.2}$$

where the water column depth is $H(\boldsymbol{x}_h, t) = \eta(\boldsymbol{x}_h, t) + h(\boldsymbol{x}_h)$, η being the free surface elevation versus a reference level, h the original static bathymetry, $\bar{\boldsymbol{u}} = [\bar{u}, \bar{v}]^T$ the horizontal (depth averaged) velocity, g the gravity acceleration, F_c the Coriolis factor, ρ the density of water, and $\boldsymbol{\tau}_b$ and $\boldsymbol{\tau}_s$ the bottom and surface stress vectors, respectively. Here, the horizontal diffusion term in equation (2.2) has been omitted because it plays little role in WD processes. Proper DG-FEM treatment of the horizontal diffusion term can be found in Rivière (2008), for example. In this work the Chézy-Manning formulation for bottom friction is used (n denotes the Manning coefficient of dimensions $\mathrm{sm}^{-1/3}$):

$$\frac{\boldsymbol{\tau}_b}{\rho} = g n^2 \frac{\|\bar{\boldsymbol{u}}\| \bar{\boldsymbol{u}}}{H^{1/3}} \tag{2.3}$$

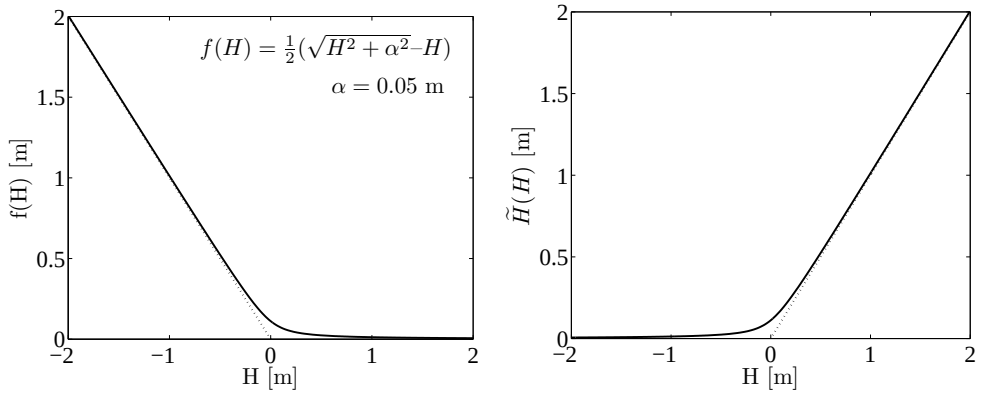
To ensure positive water depth, we introduce a smooth (at least once continuously differentiable) function f and redefine the bathymetry as $\tilde{h} = h + f(H)$. The function f is chosen so that the redefined total depth remains positive, i.e. $\tilde{H} = \eta + \tilde{h} > 0$ is always satisfied (see Figures 2.1 and 2.2). Thus $\tilde{h}$ is a function of the elevation and static bathymetry:

$$\tilde{h} = \tilde{h}(\eta, h) = h + f(\eta + h). \tag{2.4}$$



(a) Variables for fixed bathymetry (b) Variables for modified bathymetry

Figure 2.1: Redefinition of the total water column depth.



(a) Function $f(H)$ (b) Modified total depth $\tilde{H}$

Figure 2.2: Functions for defining moving bathymetry.

Now the SWEs are modified in such a way that the bed fluctuation is properly taken into account, i.e. the redefined total depth $\tilde{H}$ is being conserved:

$$\frac{\partial \eta}{\partial t} + \frac{\partial \tilde{h}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\tilde{H} \bar{\boldsymbol{u}} \right) = 0 \ , \tag{2.5}$$

$$\frac{\partial \bar{\boldsymbol{u}}}{\partial t} + \left(\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla} \right) \bar{\boldsymbol{u}} + F_c \boldsymbol{e}_z \times \bar{\boldsymbol{u}} + g \boldsymbol{\nabla} \eta = \frac{\boldsymbol{\tau}_s - \boldsymbol{\tau}_b}{\rho \tilde{H}} \ , \tag{2.6}$$

where the second term in the continuity equation (2.5) is due to the fact that $\tilde{h}$ is not static. Note that the momentum equation is modified simply by replacing H by $\tilde{H}$, which appears only in the bottom and surface stress, as well as possible forcing terms.

[1] http://www.climate.be/SLIM

The continuity equation can be further developed by noting that $\frac{\partial \eta}{\partial t} + \frac{\partial \tilde{h}}{\partial t} = (1 + f'(H))\frac{\partial \eta}{\partial t} := A(H)\frac{\partial \eta}{\partial t}$ which leads to

$$A(H)\frac{\partial \eta}{\partial t} + \boldsymbol{\nabla} \cdot \left(\tilde{H}\bar{\boldsymbol{u}}\right) = 0. \tag{2.7}$$

The above formulation is closely related to scaling of continuity equation that is presented in porous media methods (Heniche et al., 2000; Ip et al., 1998; Jiang and Wai, 2005) and also in sub-grid scale methods (Bates and Hervouet, 1999). Indeed, it is easy to see that $0 < A < 1$ so that A is a smooth indicator that ranges from dry ($A \approx 0$) to wet ($A \approx 1$) conditions. In the context of porous media and sub-grid scale wetting-drying, A is interpreted as the "active" (or "wet") fraction, i.e. proportion of total element area that is penetrable by water. In contrast to methods where elements are either wet or dry, it is precisely the smoothness of A that prevents spurious oscillations and allows implicit time marching.

Despite the illustrative power of the scaled formulation (2.7), equation (2.5) will be implemented in the numerical system thus avoiding the computation of $f'(H)$.

When comparing to other WD methods it is useful to plot the total water column depth on the original static bathymetry. In this case the elevation is given by

$$\tilde{\eta} = \tilde{H} - h = \eta + f. \tag{2.8}$$

2.3 Numerical FE implementation

In the previous Section the proposed WD method was presented on the level of the primitive equations. In this section it is shown how the method can be implemented in DG-FEM shallow water models.

2.3.1 Weak formulation

In order to derive a weak formulation for the latter equations, we define a suitable triangulation $\mathcal{T}$ of the domain Ω and piecewise discontinuous polynomial function spaces $\mathcal{V}^p$ and $\mathcal{W}^p$, such that every function $a : \mathbb{R}^2 \to \mathbb{R}$, $a \in \mathcal{V}^p$, $\boldsymbol{b} : \mathbb{R}^2 \to \mathbb{R}^2$, $\boldsymbol{b} \in \mathcal{W}^p$ is polynomial of order p (at most) inside the triangles $K \in \mathcal{T}$ and discontinuous at the interfaces $e = K \cap K'$. Since the functions are discontinuous at the interfaces we can define a set of polynomial basis functions φ_i and $\boldsymbol{\psi}_i$ that are non-zero only in a single element. Therefore we have representations for $a = \sum_i a_i \varphi_i$ and $\boldsymbol{b} = \sum_i \boldsymbol{b}_i^T \boldsymbol{\psi}_i$.

We now require that the numerical solution to (2.5) belongs to these spaces, i.e. $\hat{\eta} \in \mathcal{V}^p$ and $\hat{\bar{\boldsymbol{u}}} \in \mathcal{W}^p$. Multiplying (2.5) and (2.6) by test functions $\varphi \in \mathcal{V}^p$ and $\boldsymbol{\psi} \in \mathcal{W}^p$, respectively, integrating by parts and denoting the element-wise

surface and contour integrals as

$$\int_{K\in\mathcal{T}} \cdot \, d\boldsymbol{x}_h = \Big\langle \cdot \Big\rangle_K, \tag{2.9}$$

$$\int_{e=K\cap K'} \cdot \, dS = \Big\langle\!\Big\langle \cdot \Big\rangle\!\Big\rangle_e, \tag{2.10}$$

respectively, the weak formulation of the system (2.5)-(2.6) becomes

$$\Big\langle \frac{\partial \eta}{\partial t}\varphi \Big\rangle_K + \Big\langle \frac{\partial \tilde{h}}{\partial t}\varphi \Big\rangle_K + \sum_{e=K\cap K'} \Big\langle\!\Big\langle (\tilde{H}\bar{\boldsymbol{u}})^* \cdot \boldsymbol{n}\varphi \Big\rangle\!\Big\rangle_e - \Big\langle \tilde{H}\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla}\varphi \Big\rangle_K = 0, \tag{2.11}$$

$$\begin{aligned} \Big\langle \frac{\partial \bar{\boldsymbol{u}}}{\partial t} \cdot \boldsymbol{\psi} \Big\rangle_K &+ \sum_{e=K\cap K'} \Big\langle\!\Big\langle \bar{\boldsymbol{u}}^* \cdot \boldsymbol{n}\ \bar{\boldsymbol{u}}^* \cdot \boldsymbol{\psi} \Big\rangle\!\Big\rangle_e - \Big\langle \boldsymbol{\nabla} \cdot (\bar{\boldsymbol{u}}\boldsymbol{\psi}) \cdot \bar{\boldsymbol{u}} \Big\rangle_K \\ &+ \sum_{e=K\cap K'} \Big\langle\!\Big\langle g\eta^* \boldsymbol{\psi} \cdot \boldsymbol{n} \Big\rangle\!\Big\rangle_e - \Big\langle g\eta \boldsymbol{\nabla} \cdot \boldsymbol{\psi} \Big\rangle_K \\ &+ \Big\langle F_c \boldsymbol{e}_z \times \bar{\boldsymbol{u}} \cdot \boldsymbol{\psi} \Big\rangle_K = \Big\langle \big(\frac{\boldsymbol{\tau}_s - \boldsymbol{\tau}_b}{\rho\tilde{H}}\big) \cdot \boldsymbol{\psi} \Big\rangle_K. \end{aligned} \tag{2.12}$$

The major difference between the above weak formulation and standard FE SWE formulation is the novel mass correction term due to the moving bathymetry, i.e. the second term in equation (2.11).

Due to the discontinuity, the values of η and $\bar{\boldsymbol{u}}$ are defined twice at element interfaces. Thus the corresponding values in the contour integrals are ambiguous and are marked with an asterisk. These values are solved with an approximate Riemann solver that is derived in the next section.

2.3.2 Approximate Riemann solver

In this work we are using an approximate Riemann solver that is based on the Roe averages (Roe, 1997), which is commonly used in SWE models, e.g. in Gourgue et al. (2009); Nikolos and Delis (2009); Zhao et al. (1996); Comblen et al. (2010b). Another DG-FEM WD method by Bunya et al. (2009) utilises Lax-Friedrichs flux. In Ern et al. (2008) Harten-Lax-van Leer-Contact (HLLC) flux, which ensures non-negative water depth, is used. Such flux, however, contains many conditional statements and is thus difficult to implement in an implicit model.

Consider two neighbouring elements, left K_l and right K_r, $e = K_l \cap K_r \neq \varnothing$, such that the x axis is oriented to the normal direction from K_l to K_r and the element interface is along the y axis. The corresponding velocities are denoted $\bar{u}$ and $\bar{v}$, respectively.

The Riemann problem is derived with the conservative form of the shallow water equations. As usual in FE and FV methods, only the transport in the normal direction is taken into account. This leads to a 1D Riemann problem

(perpendicular to the interface), and consequently the partial derivatives with respect to y can be omitted. Considering first the non-modified SWEs, one obtains the following set of equations:

$$\frac{\partial H}{\partial t} + \frac{\partial (H\bar{u})}{\partial x} = 0 \tag{2.13}$$

$$\frac{\partial (H\bar{u})}{\partial t} + \frac{\partial (H\bar{u}^2)}{\partial x} + \frac{\partial (\frac{1}{2}gH^2)}{\partial x} = gH\frac{\partial h}{\partial x} \tag{2.14}$$

$$\frac{\partial (H\bar{v})}{\partial t} + \frac{\partial (H\bar{u}\bar{v})}{\partial x} = 0. \tag{2.15}$$

In what follows, we do not take into account the bathymetry gradient $\frac{\partial h}{\partial x}$, because incorporating its influence in approximate Riemann solvers is still a quite difficult task.[2] There are indeed numerous publications on the topic, mainly with respect to FV methods (see e.g. George, 2008; Nikolos and Delis, 2009, and references therein). However, such a difficulty is not analysed here for the sake of simplicity.

With this assumption the right hand side of equation (2.14) disappears and one ends up with a homogeneous set of equations. By defining a state vector $\boldsymbol{q} = [H, H\bar{u}, H\bar{v}]^T$ the system can be written as (using the shorthand notation $\boldsymbol{q}_t := \frac{\partial \boldsymbol{q}}{\partial t}$)

$$\boldsymbol{q}_t + \boldsymbol{F}_x = \boldsymbol{q}_t + \boldsymbol{V}(\boldsymbol{q})\boldsymbol{q}_x = 0, \tag{2.16}$$

where

$$\boldsymbol{V}(\boldsymbol{q}) = \begin{bmatrix} 0 & 1 & 0 \\ -\bar{u}^2 + gH & 2\bar{u} & 0 \\ -\bar{u}\bar{v} & \bar{v} & \bar{u} \end{bmatrix}. \tag{2.17}$$

Approximate Riemann solvers are based on the idea of linearising (2.16), i.e. replacing $\boldsymbol{V}(\boldsymbol{q})$ with a local constant. Given the state vectors on both sides of the boundary, $\boldsymbol{q}_l$ and $\boldsymbol{q}_r$, the linearisation $\boldsymbol{V}_{\text{lin}}$ can be found by requiring that it satisfies the Rankine-Hugoniot equation: $\boldsymbol{V}_{\text{lin}}(\boldsymbol{q}_l - \boldsymbol{q}_r) = \boldsymbol{F}_l - \boldsymbol{F}_r$. This leads to the Roe averages (Roe, 1997)[3]:

$$H_{\text{Roe}} = (H_l + H_r)/2, \tag{2.18}$$

$$\bar{u}_{\text{Roe}} = \frac{\sqrt{H_l}\bar{u}_l + \sqrt{H_r}\bar{u}_r}{\sqrt{H_l} + \sqrt{H_r}}, \tag{2.19}$$

$$\bar{v}_{\text{Roe}} = \frac{\sqrt{H_l}\bar{v}_l + \sqrt{H_r}\bar{v}_r}{\sqrt{H_l} + \sqrt{H_r}}. \tag{2.20}$$

[2] It is worth stressing that the bathymetry gradient is taken into account in the DG formulation (2.11), and its effect is neglected only in the stabilizing Riemann solver. As such the simplification does not have a major impact on the flow.

[3] The problem is in fact under-determined, so (2.18) is chosen rather than deduced. Using an additional equation will lead to a fully determined system with the same result, see e.g. Zhao et al. (1996)

The $\boldsymbol{V}_{\text{lin}}$ is obtained by substituting (2.18)-(2.20) to (2.17). The linear system can now be solved by diagonalising $\boldsymbol{V}_{\text{lin}} = \boldsymbol{RDR}^{-1}$, where $\boldsymbol{R}$ contains the right eigenvectors. The eigenvalues of $\boldsymbol{V}$ are $\{\lambda_i\}_{i=1}^{3} = \{\bar{u}, \bar{u}+c_{\text{Roe}}, \bar{u}-c_{\text{Roe}}\}$, $c_{\text{Roe}} = \sqrt{gH_{\text{Roe}}}$, corresponding to speeds at which waves propagate across the boundary. Introducing new state vectors, $\boldsymbol{Q} = \boldsymbol{R}^{-1}\boldsymbol{q}$, leads to a system of independent equations $\boldsymbol{Q}_t + \boldsymbol{DQ}_x = 0$. These are solved simply by taking the upwind value of $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$:

$$Q_i^* = \begin{cases} Q_{i,l}, & \lambda_i > 0 \\ Q_{i,r}, & \lambda_i < 0 \end{cases}, \quad i = 1, \ldots, 3. \tag{2.21}$$

Defining a jump operator $[[a]] = (a_l - a_r)/2$ and a mean operator $\{a\} = (a_l + a_r)/2$ the solutions can be formulated as

$$\begin{aligned} Q_i^* &= s_i[[Q_i]] + \{Q_i\}, \quad i = 1, \ldots, 3, \\ s_i &= \text{sign}(\lambda_i). \end{aligned} \tag{2.22}$$

Finally, one can compute the solution $\boldsymbol{q}^* = \boldsymbol{RQ}^*$:

$$H^* = \{H\} + \frac{s_2 + s_3}{2}[[H]] + \frac{s_2 - s_3}{2c_{\text{Roe}}}\left([[H\bar{u}]] - \bar{u}_{\text{Roe}}[[H]]\right) \tag{2.23}$$

$$\begin{aligned} (H\bar{u})^* = \{H\bar{u}\} &+ \frac{s_2 + s_3}{2}[[H\bar{u}]] + \frac{s_2 - s_3}{2}c_{\text{Roe}}[[H]] \\ &+ \frac{(s_2 - s_3)\bar{u}_{\text{Roe}}}{2c_{\text{Roe}}}\left([[H\bar{u}]] - \bar{u}_{\text{Roe}}[[H]]\right) \end{aligned} \tag{2.24}$$

$$\begin{aligned} (H\bar{v})^* = \{H\bar{v}\} &+ s_1[[H\bar{v}]] + \frac{s_2 + s_3 - 2s_1}{2}\bar{v}_{\text{Roe}}[[H]] \\ &+ \frac{(s_2 - s_3)\bar{v}_{\text{Roe}}}{2c_{\text{Roe}}}\left([[H\bar{u}]] - \bar{u}_{\text{Roe}}[[H]]\right) \end{aligned} \tag{2.25}$$

$$\eta^* = H^* - h \tag{2.26}$$

$$\bar{u}^* = (H\bar{u})^*/H^* \tag{2.27}$$

$$\bar{v}^* = (H\bar{v})^*/H^* \tag{2.28}$$

Generally, the horizontal coordinate axes do not coincide with the element interface and an appropriate rotation is applied to the coordinate system so that the above formulation remains valid.

In the Riemann solver presented above it is necessary to compute the square root of the total depth. When extending the solver to the moving bathymetry WD method, we therefore use the modified depth $\tilde{H}$ for which positivity is guaranteed. In other words, H is replaced by $\tilde{H}$ in equations (2.23)-(2.25). After computing $\tilde{H}^*$, $(\tilde{H}\bar{u})^*$ and $(\tilde{H}\bar{v})^*$, the final solution is obtained as:

$$\eta^* = \tilde{H}^* - \{\tilde{h}\} \tag{2.29}$$

$$\bar{u}^* = (\tilde{H}\bar{u})^*/\tilde{H}^* \tag{2.30}$$

$$\bar{v}^* = (\tilde{H}\bar{v})^*/\tilde{H}^* \tag{2.31}$$

As stated earlier, the bathymetry source term is ignored in this formulation. Including the bathymetry gradient is not obvious as it leads to an inhomogeneous system that cannot be represented as fluxes (George, 2008). In the case of moving bathymetry this problem is even more complicated as the bathymetry $\tilde{h}$ is discontinuous and also depends non-linearly on the elevation through (2.4).

The presented procedure (2.29)-(2.31) is equivalent to ignoring the possible jumps in the bathymetry and assuming that the two water columns $\tilde{H}_l$ and $\tilde{H}_r$ stand on a common, continuous bed. In practice the procedure appeared to be sufficient as long as the discontinuity in $\tilde{h}$ remains small compared to the total depth.

2.3.3 Implicit time marching

Equations (2.11-2.12) are expressed in continuous time and discrete space. It must be noted that these equations are specific compared to common finite element formulations. Indeed, the mass correction term, i.e. the time derivative of the bathymetry $\tilde{h}$, is non-linear. In general there are two ways of solving non-linear PDEs: Either explicit time integration with high temporal resolution or implicit time integration featuring a non-linear solver. In order to solve the equations accurately with reasonable computational cost, an implicit time-stepping scheme is adopted here.

Diagonally Implicit Runge-Kutta (DIRK) methods provide high order accuracy and high flexibility, as the time step can be easily varied. Runge-Kutta methods involve multiple stages, and with DIRK schemes a non-linear system of equations is to be solved at each stage.

For brevity, the system (continuous in time and discrete in space) is written as follows:

$$\left\langle \varphi \frac{\partial \left(\eta + \tilde{h}(\eta, h)\right)}{\partial t} \right\rangle_K = \boldsymbol{S}_\eta(\eta, \bar{\boldsymbol{u}}, \varphi), \tag{2.32}$$

$$\left\langle \boldsymbol{\psi} \cdot \frac{\partial \bar{\boldsymbol{u}}}{\partial t} \right\rangle_K = \boldsymbol{S}_{\bar{\boldsymbol{u}}}(\eta, \bar{\boldsymbol{u}}, \boldsymbol{\psi}), \tag{2.33}$$

where $\boldsymbol{S}_\eta(\eta, \bar{\boldsymbol{u}})$ and $\boldsymbol{S}_{\bar{\boldsymbol{u}}}(\eta, \bar{\boldsymbol{u}})$ are the discrete spatial operators for η and $\bar{\boldsymbol{u}}$, respectively.

Given the solution $(\eta^{n-1}, \bar{\boldsymbol{u}}^{n-1})$ at time t^{n-1}, the solution at time t^n is obtained with a fully implicit Runge-Kutta time marching defined below. The superscript index n refers to time steps while superscript i is used to denote the Runge-Kutta stages.

- For each stage $i = 1 \dots s$, solve the non-linear system of equations:

$$\begin{aligned} \left\langle \varphi \eta^i \right\rangle_K &= -\left\langle \varphi \tilde{h}(\eta^i) \right\rangle_K + \left\langle \varphi \left(\eta^{n-1} + \tilde{h}(\eta^{n-1}) \right) \right\rangle_K \\ &\quad + \Delta t \sum_{j=1}^{i} a_{i,j} \boldsymbol{S}_\eta(\eta^j, \bar{\boldsymbol{u}}^j), \end{aligned} \tag{2.34}$$

$$\left\langle \boldsymbol{\psi} \cdot \bar{\boldsymbol{u}}^i \right\rangle_K = \left\langle \boldsymbol{\psi} \cdot \bar{\boldsymbol{u}}^{n-1} \right\rangle_K + \Delta t \sum_{j=1}^{i} a_{i,j} \boldsymbol{S}_{\bar{\boldsymbol{u}}}(\eta^j, \bar{\boldsymbol{u}}^j) \tag{2.35}$$

- The final stage reads:

$$\begin{aligned} \left\langle \varphi \eta^n \right\rangle_K &= -\left\langle \varphi \tilde{h}(\eta^n) \right\rangle_K + \left\langle \varphi \left(\eta^{n-1} + \tilde{h}(\eta^{n-1}) \right) \right\rangle_K \\ &\quad + \Delta t \sum_{j=1}^{s} b_j \boldsymbol{S}_\eta(\eta^j, \bar{\boldsymbol{u}}^j), \end{aligned}$$

$$\left\langle \boldsymbol{\psi} \cdot \bar{\boldsymbol{u}}^n \right\rangle_K = \left\langle \boldsymbol{\psi} \cdot \bar{\boldsymbol{u}}^{n-1} \right\rangle_K + \Delta t \sum_{j=1}^{s} b_j \boldsymbol{S}_{\bar{\boldsymbol{u}}}(\eta^j, \bar{\boldsymbol{u}}^j)$$

In comparison to usual implicit DG-FEM implementation the major difference here is the treatment of the non-linear mass correction term shown in equation (2.32). As a consequence a new term $\left\langle \varphi \tilde{h}(\eta^i) \right\rangle_K$ appears in equation (2.34) that depends non-linearly on the state variable η^i.

The weights $a_{i,j}$, b_j and c_i are the Runge-Kutta coefficients. The time corresponding to each stage i is $t^i = t^n + c_i \Delta t$ which is used when computing the external forcings. Here we only use schemes for which $a_{s,j} = b_j$, $c_s = 1$ and thus the s-th RK stage gives the final solution directly, i.e. $\eta^n = \eta^s$, $\bar{\boldsymbol{u}}^n = \bar{\boldsymbol{u}}^s$. In this paper we are using second and fourth order accurate schemes, namely DIRK22 (Ascher et al., 1997, section 2.6) and ESDIRK64 (Jothiprasad et al., 2003, Appendix A) that are presented in A.1.

2.3.4 Newton solver

The equations (2.34)-(2.35) are fully implicit and non-linear. They are solved with a conventional Newton iteration. The non-linear system is linearised, the required Jacobian matrix of the system being approximated numerically. The Jacobian of equations (2.34)-(2.35) is given by

$$\boldsymbol{J}(\eta^i, \bar{\boldsymbol{u}}^i) = \begin{bmatrix} -\left\langle \varphi \tilde{h}(\eta^i) \right\rangle_K + \Delta t \; a_{i,i} \boldsymbol{S}_\eta(\eta^i, \bar{\boldsymbol{u}}^i) \\ \Delta t \; a_{i,i} \boldsymbol{S}_{\bar{\boldsymbol{u}}}(\eta^i, \bar{\boldsymbol{u}}^i) \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \eta^i} & \frac{\partial}{\partial u^i} & \frac{\partial}{\partial v^i} \end{bmatrix}.$$

To compute the Jacobian reliably the presented numerical system has to be continuously differentiable with respect to the state variables η, u, v. In

this work the partial derivatives are approximated by means of finite differencing where the perturbation to the state variables was of order 10^{-6}. Such a numerical differentiation is valid for continuously differentiable functions. At discontinuities, however, numerical approximation is not reliable and it often causes oscillations that prevent the Newton solver from converging.

The advantage of the proposed WD method is that all the modifications are indeed smooth without any discontinuities or switches. In what has been presented, the only non-differentiable operator is the sign function found in the Riemann solver in equation (2.22). The sign function is replaced with a smooth approximation:

$$\operatorname{sign}(u) \approx \tanh \beta \frac{u}{U}, \quad \beta \approx 60, \; U \approx 1\text{m/s}. \tag{2.36}$$

For discontinuous Galerkin discretisation, the Jacobian matrix is composed of blocks, where each block corresponds to an element. This block structure enables an efficient Incomplete LU (ILU) factorisation. Using an ILU factorisation with no fill-in as preconditioner for GMRES (Generalized Minimal Residual Method, Saad and Schultz, 1986) iteration appears to be sufficient to ensure convergence. In our implementation, such an efficiency can only be achieved with DG elements.

In practice the Newton solver appeared to be robust and it was observed to converge rapidly. Stopping criterion was the relative error versus initial state, i.e. $\varepsilon_{rel,i} = \varepsilon_i / \varepsilon_0$, where ε_i is the residual of iteration i. Typically only 3 to 6 iterations were required to reach $\varepsilon_{rel} = 10^{-7}$. In the inner loop, the GMRES solver required roughly 50 iterations on average to converge to tolerance $\varepsilon_{rel} = 10^{-3}$ (compared to the beginning of the GMRES iteration).

2.3.5 Mass conservation

Equation (2.5) represents mass conservation of the entire water column. The corresponding weak formulation (2.11) will result in a mass conservative scheme if the term $\left\langle \varphi \, \partial\tilde{h}/\partial t \right\rangle_K$ is computed accurately. The presented Runge-Kutta time marching will conserve mass because at each subiteration $\partial\eta/\partial t$ and $\partial\tilde{h}/\partial t$ are treated similarly. Thus only requirement is that the term $\left\langle \varphi \, \tilde{h}(\eta^i) \right\rangle_K$ is computed accurately in equation (2.34).

The latter imposes two restrictions: First, due to the non-linearity of $\tilde{h}$ an iterative solver, such as the Newton method proposed here, must be used. Explicit methods can only provide approximate mass conservation. Secondly, the numerical quadrature of $\left\langle \varphi \, \tilde{h}(\eta^i) \right\rangle_K$ must be accurate.

In this work conventional Hammer quadrature rules are used, that are accurate up to polynomials of order $2p + 1$, where p is the degree of the FE discretisation (the extra p is required for the test function). However, no polynomial can meet the desired properties of $\tilde{h}$ outlined in Section 2.2 and indeed $f(H)$ must be of infinite order. Therefore simply evaluating $\tilde{h}$ at integration

points implies that the quadrature is not exact and mass conservation breaks down.

Up to this point the polynomial order of the elevation field has been free but now we restrict ourselves to $p = 1$. Linear elements have the unique property that the extrema are found at the vertices and thus it suffices to ensure that water depth remains positive at the nodes. The moving bathymetry is therefore defined by computing the nodal values using (2.4) with linear interpolation in between:

$$\hat{\tilde{h}}^n = \sum_{i=1}^{3} \tilde{h}(\eta_i^n, h_i)\varphi_i(\boldsymbol{x}_h), \tag{2.37}$$

where φ_i are now linear discontinuous basis functions. Equation (2.37) results in a mass conservative scheme. For velocity field higher order elements can still be used. In this work first order elements are used for all fields.

2.3.6 Choosing the function $f(H)$

The function f should meet the following properties:

- $\tilde{H} = H + f(H) > 0$ for all H
- $f \approx 0$ for $H >> 1$ m
- f must be continuously differentiable

The first property was already mentioned in Section 2.2. The second property states that the modification is restricted to shallow areas only. And the third property is needed for ensuring convergence of the Newton iteration. In this work the following function, that fulfils the desired properties, is used:

$$\begin{aligned} f(H) &= \frac{1}{2}(\sqrt{H^2 + \alpha^2} - H), \\ \Rightarrow \tilde{H} &= H + f(H) = \frac{1}{2}(\sqrt{H^2 + \alpha^2} + H). \end{aligned} \tag{2.38}$$

Clearly, f is monotonously decreasing, continuously differentiable and satisfy $f(H) > \max\{-H, 0\}\ \forall H \in \mathbb{R}$. The free parameter controls the smoothness of the transition: $f(H) \to \max\{-H, 0\}$, as $\alpha \to 0$ (see Figure 2.2a). The dimension of both f and α is meters.

In practice the α parameter affects the width of the transition zone between wet ($A(H) \approx 1$, $\tilde{h} \approx h$) and dry ($A(H) \approx 0$) areas. Noting that for $H = 0$, $\tilde{H} = \alpha/2$, it is easy to see that α also directly controls the water depth in dry areas. As such α is similar to the threshold depth parameter used in thin-layer WD methods, as both are expressed in meters and determine the remaining water layer depth.

For robust operation, the WD interface should be smooth implying that the transition zone should encompass more that one element. Estimating the variation in bathymetry within one element by

$$\epsilon := L_x |\boldsymbol{\nabla} h|, \tag{2.39}$$

where L_x is the horizontal length scale, one can conclude that $\alpha \approx \epsilon$, which can be used as a rule of thumb for gradually sloping domains.

In terms of numerical stability, other more restrictive constraints may exist. The system should be smooth enough to ensure convergence of the Newton solver. It is also plausible that rapidly varying flows and dealing with shock waves require larger values for α. These restrictions are, however, more difficult to estimate a priori.

2.4 Tracer consistency

The tracer equation is implemented in conservative form and thus taking into account the moving bathymetry is straightforward. Denoting the tracer concentration by C, the depth-averaged tracer equation in conservative form is given by

$$\frac{\partial HC}{\partial t} + \boldsymbol{\nabla} \cdot (HC\bar{\boldsymbol{u}}) = S + \boldsymbol{\nabla} \cdot (\kappa H \boldsymbol{\nabla} C) \tag{2.40}$$

where κ is the tracer horizontal diffusivity and S contains the sources and sinks.

We now show that it is possible to formulate the tracer equation in such a way that it is consistent with the continuity equation (2.5). Consistency in this context means that setting $C = 1$ in the tracer equation should lead to exactly the same numerical procedure that is used to solve the continuity equation (e.g. White et al., 2008a).

It is clear that replacing H by $\tilde{H}$ in equation (2.40) leads to an equation that is consistent with (2.5). The weak form of the modified tracer equation becomes

$$\begin{aligned} \left\langle \frac{\partial \tilde{H}C}{\partial t}\varphi \right\rangle_K + \sum_{e=K\cap K'} \left\langle\!\!\left\langle (\tilde{H}C\bar{\boldsymbol{u}})^* \cdot \boldsymbol{n}\varphi \right\rangle\!\!\right\rangle_e - \left\langle \tilde{H}C\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla}\varphi \right\rangle_K = \\ \sum_{e=K\cap K'} \left\langle\!\!\left\langle \kappa \tilde{H}^* \boldsymbol{\nabla} C^* \cdot \boldsymbol{n}\varphi \right\rangle\!\!\right\rangle_e - \left\langle \kappa \tilde{H} \boldsymbol{\nabla} C \cdot \boldsymbol{\nabla}\varphi \right\rangle_K . \end{aligned} \tag{2.41}$$

Clearly, setting $C = 1$ in above leads to (2.11). Again the unknown values $(\tilde{H}C)^*$ in the element interfaces are computed with an approximate Riemann solver. A solution for $(\tilde{H}C)^*$ can be derived by adding the tracer equation (with zero diffusivity) to the system (2.16). Because in this context the tangential velocity $\bar{v}$ is essentially treated as a passive tracer, one obtains a similar

solution:

$$(HC)^* = \{HC\} + s_1[[HC]] + \frac{s_2 + s_3 - 2s_1}{2} C_{\text{Roe}}[[H]] + \frac{(s_2 - s_3)C_{\text{Roe}}}{2c_{\text{Roe}}} \left([[H\bar{u}]] - \bar{u}_{\text{Roe}}[[H]]\right) \tag{2.42}$$

In the above equation C_{Roe} denotes the tracer concentration value used in the linearised equations. Similarly to $\bar{v}_{\text{Roe}}$, solving the Rankine-Hugoniot relation results in the Roe average:

$$C_{\text{Roe}} = \frac{\sqrt{H_l}C_l + \sqrt{H_r}C_r}{\sqrt{H_l} + \sqrt{H_r}}, \tag{2.43}$$

Clearly, for an arbitrary constant tracer concentration $C = C_0$, one gets $C_{\text{Roe}} = C_0$ and $(HC)^* = C_0 H^*$, which confirms consistency.

If a first order time discretisation is used the first term of equation (2.41) becomes

$$\left\langle \frac{\partial \tilde{H}C}{\partial t} \right\rangle_K = \left\langle \frac{(\tilde{h}^{n+1} + \eta^{n+1})C^{n+1} - (\tilde{h}^n + \eta^n)C^n}{\Delta t} \right\rangle_K, \tag{2.44}$$

which is linear in C^{n+1} and no iterative solver is needed. Equation (2.41) can therefore be solved in a conventional manner when $\tilde{H}^{n+1}$ is known. However, for consistency the same fully implicit Runge-Kutta time integration presented in Section 2.3.3 is used for solving the tracer equation.

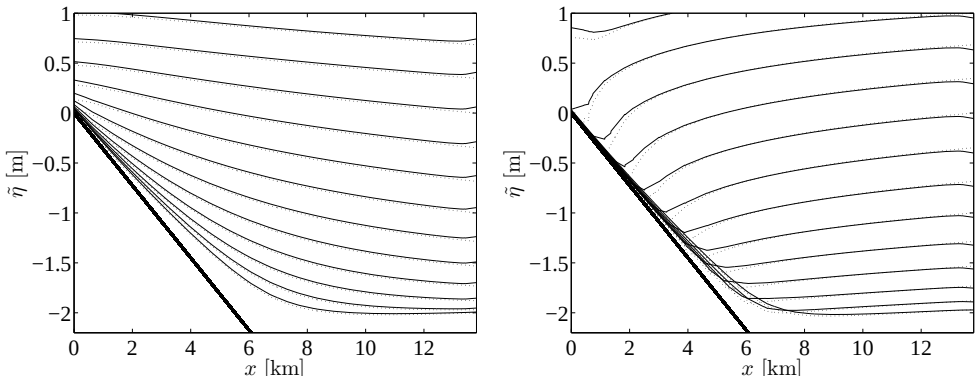
2.5 Numerical tests

The presented DG-FEM WD method was tested with several test cases commonly used in the literature and also a couple of novel ones. All the tests were solved with DIRK22 time integration with 600 second time step unless otherwise noted.

2.5.1 Balzano test cases

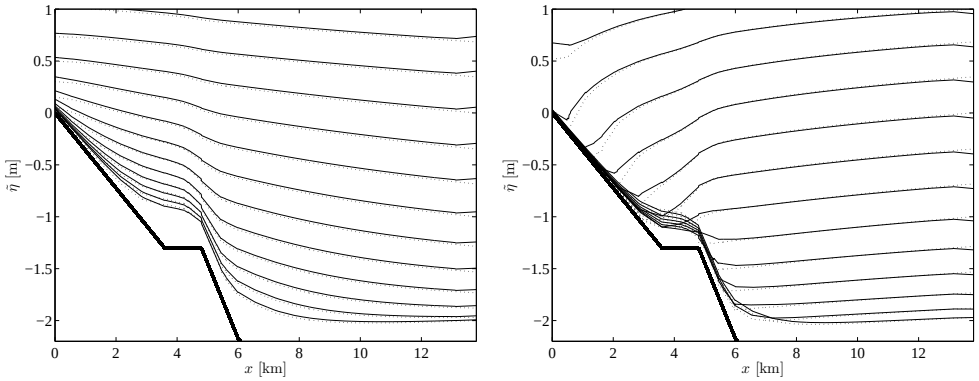
The first test cases considered are those by Balzano (1998). They feature a rectangular 13800 m long basin, with mildly sloping bathymetry ranging from zero to 5 meters. As forcing, water level perturbation is prescribed at the deep end. The Manning bottom friction coefficient is set to 0.02 $\text{s/m}^{1/3}$ while Coriolis force, viscosity and free surface stress are being ignored. The test cases are originally one dimensional but here they are solved in a 2D basin whose width is 7200 meters. For exact description of the bathymetry the reader should refer to Balzano (1998) or Gourgue et al. (2009).

The first test case considers a wave run up on a uniformly sloping bed. A sinusoidal water level perturbation with an amplitude of 2 m and period of 12



(a) drying phase (b) flooding phase

Figure 2.3: Balzano 1 test case. Elevation field at 20 minutes intervals for (a) the drying phase and (b) flooding phase. The thick line corresponds to the static bed. For comparison results obtained with an explicit WD method are show with a dotted line.



(a) drying phase (b) flooding phase

Figure 2.4: Balzano 2 test case. Elevation field at 20 minutes intervals for (a) the drying phase and (b) flooding phase. The thick line corresponds to the static bed. For comparison results obtained with an explicit WD method are show with a dotted line.

hours is imposed. Here we are using a uniform mesh of 1200 m horizontal resolution, similar to the original one dimensional tests by Balzano. The vertical length scale $\epsilon = 0.43$ m and a slightly smaller value is used for the α smoothness parameter $\alpha = 0.3$ m. Figure 2.3 shows the modified elevation field $\tilde{\eta}$ at 20 minutes intervals for the drying and flooding phase. It is noteworthy that the $\tilde{\eta}$ elevation curves correspond to the total water column depth and thus these plots are directly comparable to other results in literature (e.g. Balzano, 1998; Nielsen and Apelt, 2003). In this test case, the bed does not become

completely dry during the drying phase (Figure 2.3 a), due to the gradually sloping bed and the tidal period.

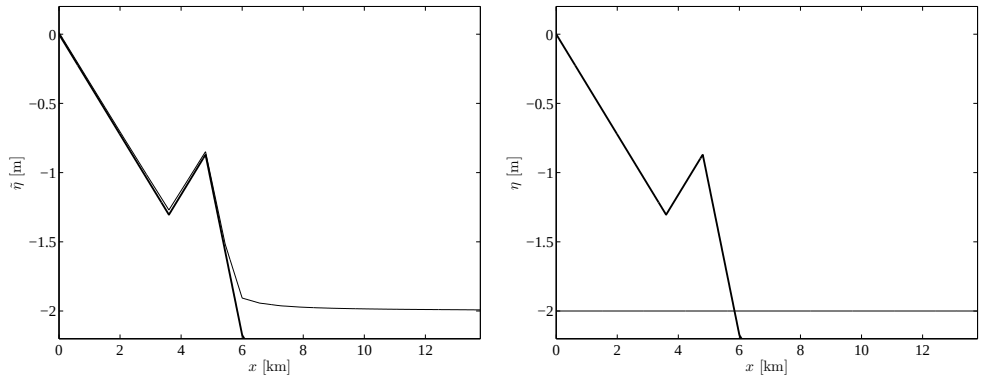
For comparison the results obtained with a fully explicit flux-limiting WD method (see Gourgue et al., 2009) are also shown in Figure 2.3. Clearly the two methods are indeed in good agreement. Moreover, the retention volume (water trapped in the dry areas) is not very large, in fact smaller than with some methods presented in Balzano (1998) (Figure 2 in their manuscipt) and no oscillations or wiggles are present. The shocks at the flooding front appear to be milder than with the explicit method, which is due to the smooth WD transition. There are also some differences near the open boundary, essentially due to the implicit treatment of the boundary conditions.

The second test is similar, except that the bed now features a flat shelf. The corresponding elevation field is plotted in Figure 2.4. The results are again in good agreement with the fully explicit WD method and similar to those presented in literature.

The third test case is different because the bed now features a small "pond" that retains water in the dry phase. In this simulation the water level at the open boundary is dropped sinusoidally to the minimal level, holding it there for an indefinitely long time in order to test whether water is leaking through the dry area. The elevation field after 100 hours, which essentially corresponds to the static solution, is shown in Figure 2.5. Because water fluxes do not vanish as long as the pressure gradient term operates, the pond eventually dries up. The flux depends on the parameter α (the smaller the value of α, the smaller the flux) and the bottom friction parametrisation, but neither can prevent the pond from drying as time goes to infinity. Figure 2.5b presents the η field that is used in the numerical implementation. Indeed it is seen that the static case corresponds to situation where the elevation field is horizontal and thus the pressure gradient term is zero. This test reveals that water is leaking through dry areas which can be seen as a drawback of the presented method. However, it must be stressed that such drawback is common also to all porous media methods (Nielsen and Apelt, 2003).

2.5.2 Steeper slopes

During the model development is was noted that the standard test cases proposed in the literature e.g. in Balzano (1998) and Leclerc et al. (1990) tend to be too easy compared to real-world simulations with complex topography. This is especially true with a steep bathymetry (Brufau et al., 2002). In this section we therefore present similar tests as the first test by Balzano but with increasing difficulty by steepening the bed slope. The proposed method was tested with several different mesh resolutions and the goal was to determine the smallest α values for which the method remains stable. To ensure similar conditions the tests were designed such that in each case at least one element would be completely dry at low water. The results are presented in Table 2.1.



(a) $\tilde{\eta}$ (b) η

Figure 2.5: Balzano 3 test case. The static solution after 100 h of simulation time. a) Total water column depth superimposed on the original bathymetry. b) Elevation field as seen by the pressure gradient term. The thick line corresponds to the static bed.

	L_x [m]				
$\lvert\boldsymbol{\nabla} h\rvert$ [m/m]	25	50	100	250	500
0.10	0.3	0.3	0.3	0.3	0.5
0.05	0.3	0.3	0.3	0.3	0.5
0.01	0.2	0.2	0.3	0.3	0.3
0.001	0.2*	0.2*	0.3*	0.5*	0.5*

Table 2.1: Smallest stable α parameter values (in meters) for various bed slopes and mesh resolutions. *) A mild shock wave appears at flooding phase, which requires larger α for coarse mesh.

It is seen in Table 1 that the smallest possible α value remains relatively constant in all the cases despite the fact that the nature of the WD process varies dramatically. Indeed, the vertical length scale ϵ varies from 50 m (upper right corner) to 0.025 m (lower left corner in Table 2.1). In the case of very steep slopes the surface remains nearly horizontal and no wave propagation effects are visible at the WD front. For more gradual bathymetry wave effects become apparent and mild shocks appear at the flooding phase. To deal with such shocks a larger retention depth (water column depth in the dry area) may be needed to maintain stability. Larger retention depth implies faster wave propagation at the WD front which naturally smooths the shocks.

2.5.3 Thacker test case

Thacker (1981) presented an analytical solution for water oscillation in a paraboloid bowl. The solution has been derived for the non-linear shallow

water equations, and has been used as a test case by many authors (Lynett et al., 2002; Balzano, 1998; Bates and Horritt, 2005; Ern et al., 2008; Casulli, 2009). The test case consists of a regular basin, a paraboloid of revolution, in which the free surface oscillates without any external forcing. The basin is large, 430.620 km in diameter, but very shallow as the depth is not more than 50 m at the centre. Initially the free surface is also a paraboloid of revolution. The dimensions of the bowl are chosen such that the free surface oscillation has a 12 hour period. For exact description of the test case see e.g. Gourgue et al. (2009); Balzano (1998).

This is a challenging test case as no bottom friction (or other dissipation) is present. Moreover the horizontal length scales are larger than in a typical application. Due to the large diameter of the basin, a common grid resolution ranges from 4 km to 10 km. Consequently, although the bed slope is very gradual, the vertical WD length scale is large: $\epsilon = 2.4$ m for a 10 km mesh. A slightly smaller value $\alpha = 2.2$ m was used for the smoothness parameter. Using a smaller α resulted in oscillations at the boundary, that in the absence of any dissipation eventually deteriorated the solution in the whole domain. In order to ensure that the boundary conditions do not affect the solution, the computational domain was extended to diameter 495.2 km so that the boundary remains dry at all times.

A cross section of the elevation at three time instances is shown in Figure 2.6a. In the centre of the domain, the numerical model is very close to the analytical solution while the difference increases towards the WD front. The smooth transition between the wet and dry regimes is clearly visible. As stated earlier the smoothness of the solution is controlled by the parameter α and thus smaller α yields more accurate solution. However, the smallest feasible α value appeared to depend on the horizontal resolution, and α was observed to be proportional to ϵ. Thus more accurate solutions can be obtained by decreasing α but the mesh has to be refined accordingly.

Figure 2.6b shows the water elevation at the centre of the domain versus time. It is seen that the proposed method is stable without noticeable additional dissipation. Moreover, as the signal does not attenuate significantly in time, the numerical dissipation also remains moderate. This test also justifies the higher order Runge-Kutta time integration presented in Section 2.3.3: Clearly first order implicit Euler method is too dissipative to be used in practical applications. Here the same time step was used for the implicit Euler method, but similar excessive dissipation was observed also for shorter time steps.

2.5.4 Mass conservation

Because the Thacker test case features a closed basin it is well suited for testing the mass conservation property presented in Section 2.3.5. Denoting the total mass at time t by $M(t)$ the relative error $\mathcal{E} = (M(0) - M(t))/M(0)$ was of order

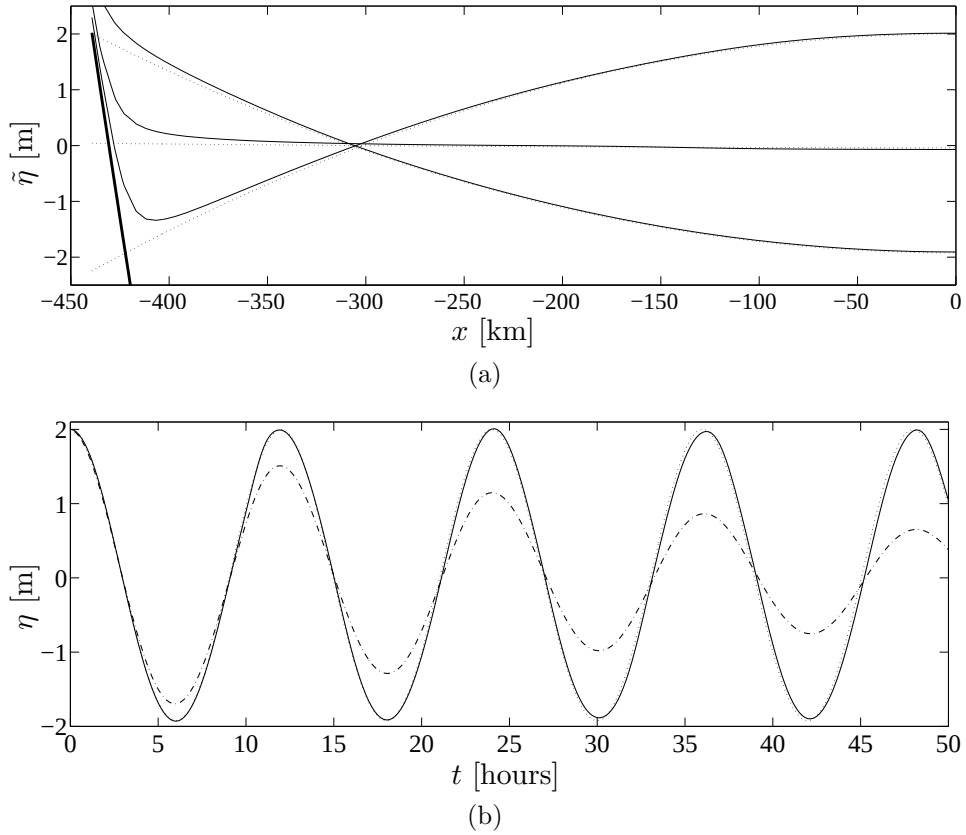


Figure 2.6: Thacker test case. a) Solid thin line: elevation after 6, 9, and 12 hours. Dotted line: analytical solution. Thick line: bathymetry. b) Water level at the centre of domain versus time. Solid line: Numerical solution. Dotted line: analytical solution. Dash-dotted line: numerical solution with implicit Euler time integration.

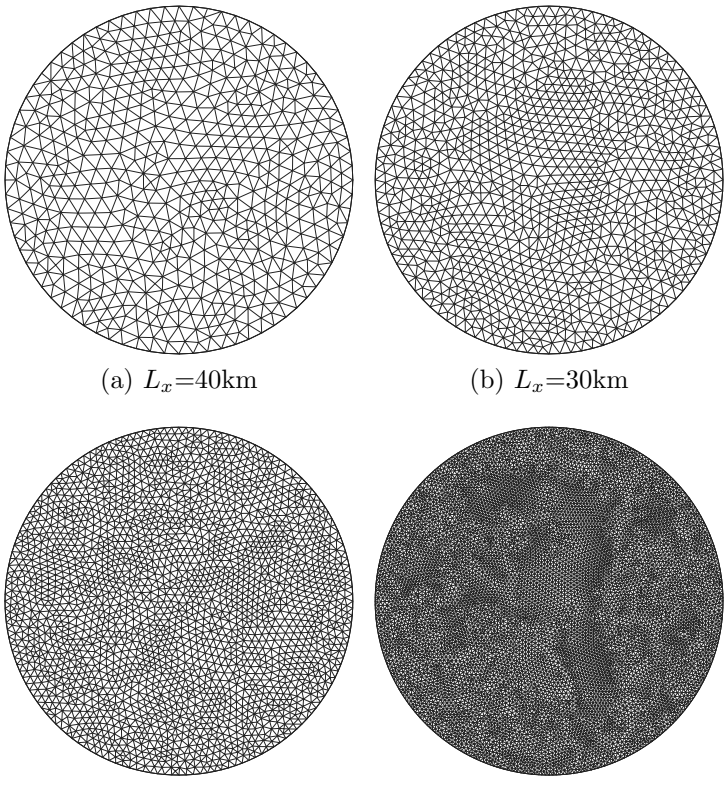
10^{-11} throughout the 25 hour simulation period, demonstrating that mass is conserved with sufficient accuracy.

2.5.5 Rate of convergence

The Thacker test case was also used to test how fast the proposed method converges toward the analytical solution when the mesh is refined. The test case was run with several meshes with increasing resolution $L_x = \{10,\ 15,\ 20,\ 30,\ 40\}$ km. As the smoothness parameter α affects the solution, it was tuned to match the length scale: the corresponding variable values were set to $\alpha = 0.9\epsilon$ i.e. $\alpha = \{2.16,\ 3.24,\ 4.32,\ 6.48,\ 8.64\}$ m. These values are close to the smallest stable values in each case because decreasing α significantly caused oscillations. Examples of the different meshes are illustrated in Figure 2.7.

The L_2 error of the elevation field $\tilde{\eta}$ was used as an error measure:

$$\mathcal{E}_{L_2} = \sqrt{\langle(\tilde{\eta} - \eta_a)^2\rangle_\Omega / \langle 1\rangle_\Omega}, \tag{2.45}$$



(a) L_x=40km (b) L_x=30km

(c) L_x=20km (d) L_x=10km

Figure 2.7: Meshes for the Thacker convergence test case.

where η_a denotes the analytical solution that takes into account the dry bed, i.e. $\eta_a = \max\{\eta_{exact}, -h\}$. It is noteworthy that the error measure encompasses the entire domain, both wet and dry areas.

The observed L_2 error is shown on Figure 2.8, showing roughly 1.5 rate of convergence. It has to be stressed, however, that the convergence test is not entirely fair as we are comparing two different solutions. Indeed the analytical solution presented in Thacker (1981) was developed for the original SWEs while in here the modified equations (2.5)-(2.6) are being solved. Moreover, the error is the largest at the WD transition, where the numerical solution is smoother than the analytical (see Figure 2.6a). Therefore the error is mainly dominated by the parameter α instead of the spatial discretisation. Nevertheless, based on this test it can be stated that refining the mesh allows smaller α values leading to smaller discrepancies, and thus the solution does converge towards the solution of the original SWEs.

2.5.6 Test on tracer consistency and conservation

Tracer consistency was tested with the Thacker test case by adding a passive tracer in the simulation. The tracer concentration was initially set to unity

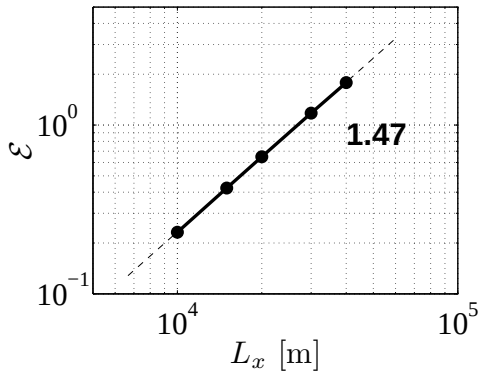


Figure 2.8: Convergence of the L_2 error versus spatial resolution. The observed rate of convergence is mainly due to the α parameter values.

throughout the domain. As demonstrated in Section 2.4, the concentration should remain equal to unity at all times. In fact, the solution for such a tracer is trivial, and therefore the numerical method should converge instantly without any iterations. Such behaviour was indeed observed. During the simulation the error in tracer concentration was of order 10^{-8} which is the same magnitude as the residual tolerance used in the Newton iteration.

Tracer mass conservation was also tested. Figure 2.9 illustrates the relative error in total tracer mass versus time. The relative error is defined by $\mathcal{E}_C = (M_C(0) - M_C(t))/M_C(0)$, where $M_C(t)$ denotes the total tracer mass at time t. The tracer mass is conserved up to precision 10^{-15} which is the same order of magnitude as the numerical precision of the model. Similar precision was obtained for a non-uniform tracer field as well.

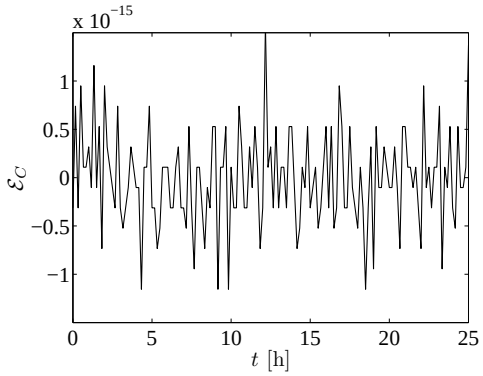
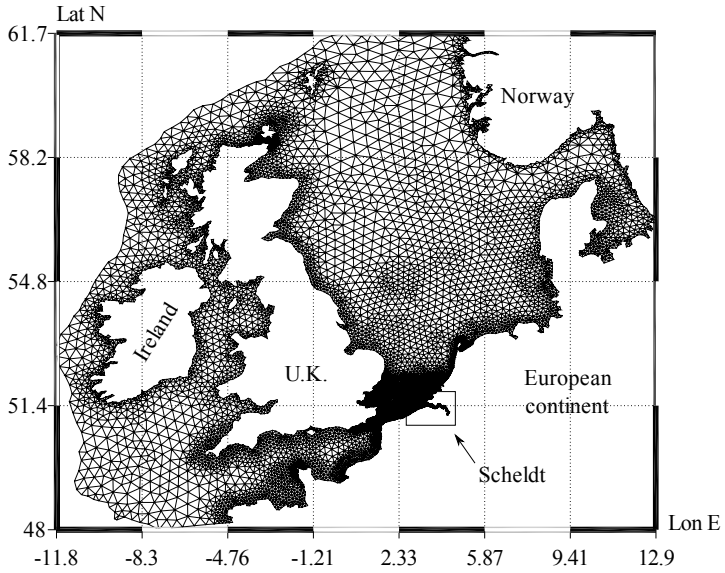
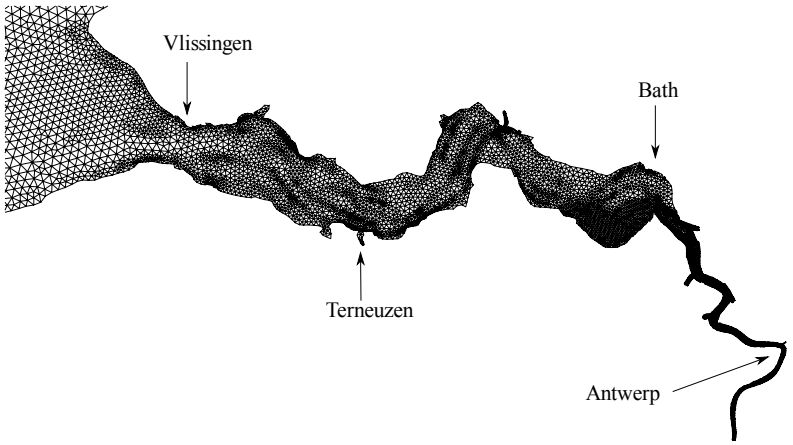


Figure 2.9: Error in total tracer mass versus time. The error is the same magnitude as the numerical precision.



(a) Triangular mesh of the entire 2D domain



(b) Enlarged view of the Scheldt Estuary

Figure 2.10: Computational domain of the Scheldt application. The mesh contains 29130 elements. The upstream river network is modelled with 1D version of the SLIM (not shown here).

2.6 Application to the Scheldt Estuary

The Scheldt Estuary is situated between Belgium and the Netherlands (Figure 2.10). The entire Scheldt River catchment area in northern France, Belgium and the Netherlands hosts approximately 7 million people and also features heavy industrial activity. Due to substantial and partly untreated discharges, the water quality in the river is generally poor with elevated levels of heavy metals, fecal bacteria and nutrients (Baeyens et al., 1997).

The Scheldt is a macrotidal estuary driven by the semi-diurnal tides on the North Sea. Indeed, the residual water flux due to river discharge is roughly two orders of magnitude smaller than that of the tidal flow. The width reduces from 6 km near the mouth to 500 m near Antwerp, roughly 80 km upstream. The tidal signal is amplified as it travels up the estuary, the tidal range near the mouth in Vlissingen being 3.8 meters and 5.2 meters further upstream in Antwerp. The estuary is generally very shallow (mean depth is roughly 10 m) but features deep flood and ebb channels (see Figure 2.11) that can reach the depth of 60 meters (Swinkels et al., 2009). The main estuary also features large tidal flats (mainly Saeftinge and Ballatsplaat) and sand banks between the two main channels, both of which are submerged during high water.

It is clear that taking wetting-drying phenomenon into account in such a domain is essential. The model domain is discretised with a triangular mesh that not only encompasses the Scheldt Estuary but also most of the North-western European Continental Shelf Sea (NWECSS) extending all the way to the shelf break (see Figure 2.10). Although the computational domain is extended drastically, the increase in computational cost remains moderate as roughly half of the elements are located inside the area of interest, i.e. the Scheldt. The mesh was generated with the GMSH software (Geuzaine and Remacle, 2009; Lambrechts et al., 2008a).

The advantage of such domain extension is that the tide can be prescribed at the shelf break using data from global tidal models. Moreover, the meteorological events in the North Sea can easily be incorporated in the model. In this work ETOPO1 bathymetry data[4] (Amante and Eakins, 2009) is used for the NWECSS while the Scheldt bathymetry is obtained from KustZuid model[5]. The tidal signal at the open boundary is defined using TPXO7.1 model[6] (Egbert et al., 1994). The meteorological forcings (wind stress and atmospheric pressure) are from global NCEP reanalysis data[7] (Kalnay et al., 1996). At the upstream boundary the 2D model is coupled with a 1D river network model that covers all the connected tidal rivers and tributaries. The tidal rivers are forced with observed river discharge at the upstream boundaries. In addition the freshwater discharges of Thames, Seine, Rhine and Meuse rivers are prescribed using daily average discharge data. More detailed information on the model setup can be found in Gourgue et al. (2009) and de Brye et al. (2010).

The hydrodynamics were solved using the DIRK22 time marching with 20 minutes time-step. The WD the smoothness parameter was set to $\alpha = 0.5$ m.

Although model validation is not the main purpose of this paper, the bottom friction coefficient was tuned to obtain a good match of the tidal signal propagation by comparing water elevation to measurements in several stations in Scheldt and North Sea. The Manning coefficient ranges from 0.0235 $s^{-1}m^{1/3}$

[4] http://www.ngdc.noaa.gov/mgg/global/

[5] Courtesy of M. Zijlema of the National Institute for Coastal and Marine Management (RIKZ)

[6] http://www.oce.orst.edu/research/po/research/tide/index.html

[7] http://www.cdc.noaa.gov/cdc/data.ncep.reanalysis.surfaceflux.html

in the shelf sea to 0.026 $s^{-1}m^{1/3}$ near Antwerp, increasing linearly along the estuary. Examples of two time series versus observations are presented in Figure 2.14. The model is clearly in good agreement with the data and there is no evidence that the moving bathymetry WD process would deteriorate the solution by affecting wave propagation, for example. A more detailed validation of the Scheldt model is presented in de Brye et al. (2010), where the same implicit WD method is used.

Snapshots of free surface elevation at high and low water can be seen in Figure 2.11 where the drying sand banks are clearly visible. Moreover, in Figure 2.12 it can be seen that the modification of the bathymetry is indeed restricted to the dry areas which also support the notion that the physical processes in the wet domain remain essentially unmodified. The depth averaged velocity field is shown in Figure 2.13. The flow circulates smoothly around the sand banks and no spurious noise is visible. Asymmetric flow patterns between ebb and flood channels is clearly observed (Swinkels et al., 2009).

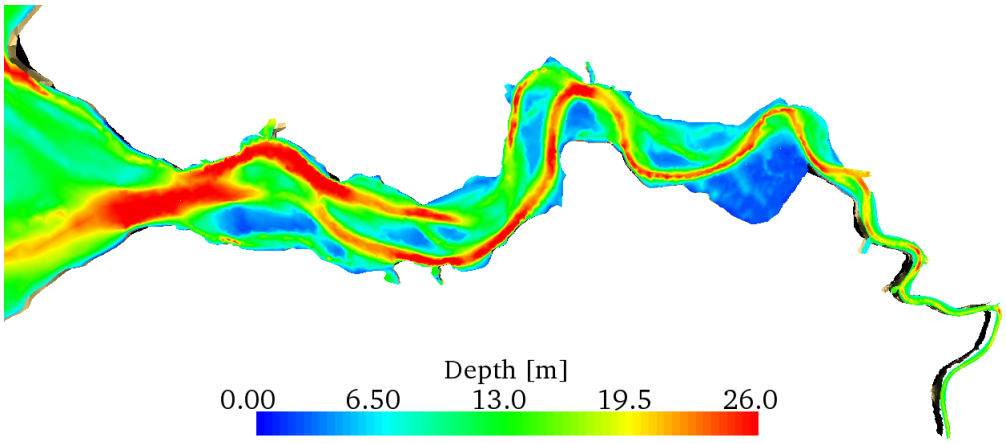
An explicit flux-limiting WD method (in conjunction with SLIM) is applied to a similar Scheldt application in Gourgue et al. (2009). Qualitatively the flux-limiting method yields similar results to what has been reported here, and in de Brye et al. (2010). The major difference is in the computational cost. As most of the WD methods rely on explicit time integration, the longest permissible time step is heavily restricted by the CFL condition. In the Scheldt Estuary the time step is less than one second due to the deep channels and relatively small elements.

With the proposed fully implicit time marching, however, there is no intrinsic restriction on the time step, it is only required to resolve the tide and the forcing signals. In this simulation therefore a much longer time step of 20 minutes is used.

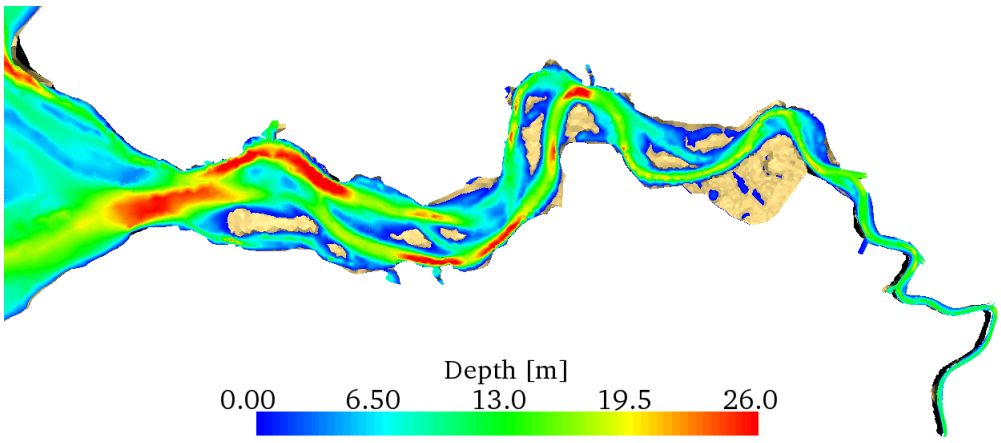
In Table 2.2 the overall CPU time is compared to the explicit flux-limiting WD method (Gourgue et al., 2009). The comparison was carried out for Balzano test 1 and the Scheldt application running the simulations for a short period of time (T). These tests were run on an Intel Xeon processor using four nodes. It is seen that already in the Balzano benchmark implicit code runs much faster, while in the Scheldt application the explicit time stepping is becoming far too expensive, running more than 200 times slower. It must be noted, however, that such a comparison is not entirely fair as neither the code nor the model setup is optimised for explicit computation. For example, one should eliminate too small elements in the mesh, which has not been carried out. Nevertheless, in order to be able to capture the tidal dynamics in the estuary, sufficiently fine mesh is required (de Brye et al., 2010) and such a setup is very demanding for explicit methods.

Simulation	Number of elements	Method	Δt	T	CPUtime	Gain
Balzano 1	561	expl.	≈ 5 s	12 h	116.9 s	
	561	impl.	600 s	12 h	20.46 s	5.7
Scheldt	29130	expl.	< 0.2 s	1 h	34592 s	
	29130	impl.	1200 s	1 h	129.64 s	267

Table 2.2: Comparison of CPU times for flux-limiting and moving bathymetry wetting-drying. The presented method runs significantly faster due to the implicit time stepping.

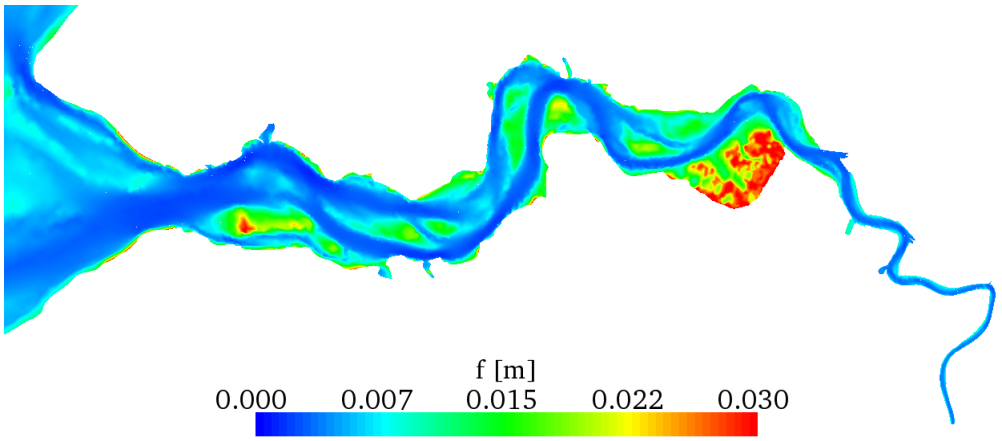


(a)

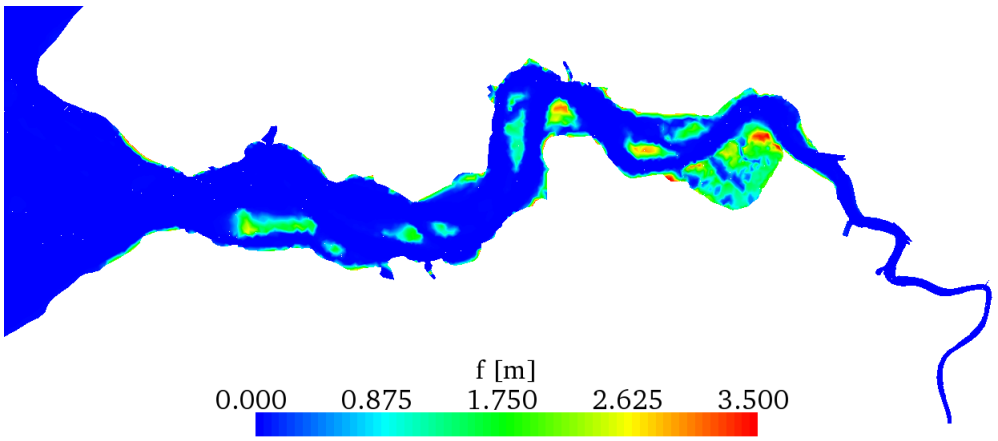


(b)

Figure 2.11: Wetting-drying in the Scheldt Estuary during spring tide. a) Depth at high water b) Depth at low water. Exposed sand banks are clearly visible. Maximum depth is roughly 60 m.



(a)



(b)

Figure 2.12: Bathymetry displacement $|\tilde{h} - h|$ during a) high water b) low water. Note that the color scale is different in the two cases. The modification remains very small in the deep areas.

2.7 Conclusions

In this paper a novel fully implicit WD method has been proposed. Although the method relies on artificially moving the bathymetry in dry areas, both the numerical test and the real-world application in the Scheldt Estuary confirms that the WD processes are modelled with good accuracy. No spurious behaviour was noticed in the dry areas and wave propagation properties seem intact.

Although the moving bathymetry WD method resembles porous media approach, there are several differences. Since our model relies on artificial displacement of bathymetry, there is no need to introduce the concept of porosity nor properties of the porous layer. Generally this leads to a simpler numerical formulation and fewer number of unknown parameters than in porous media

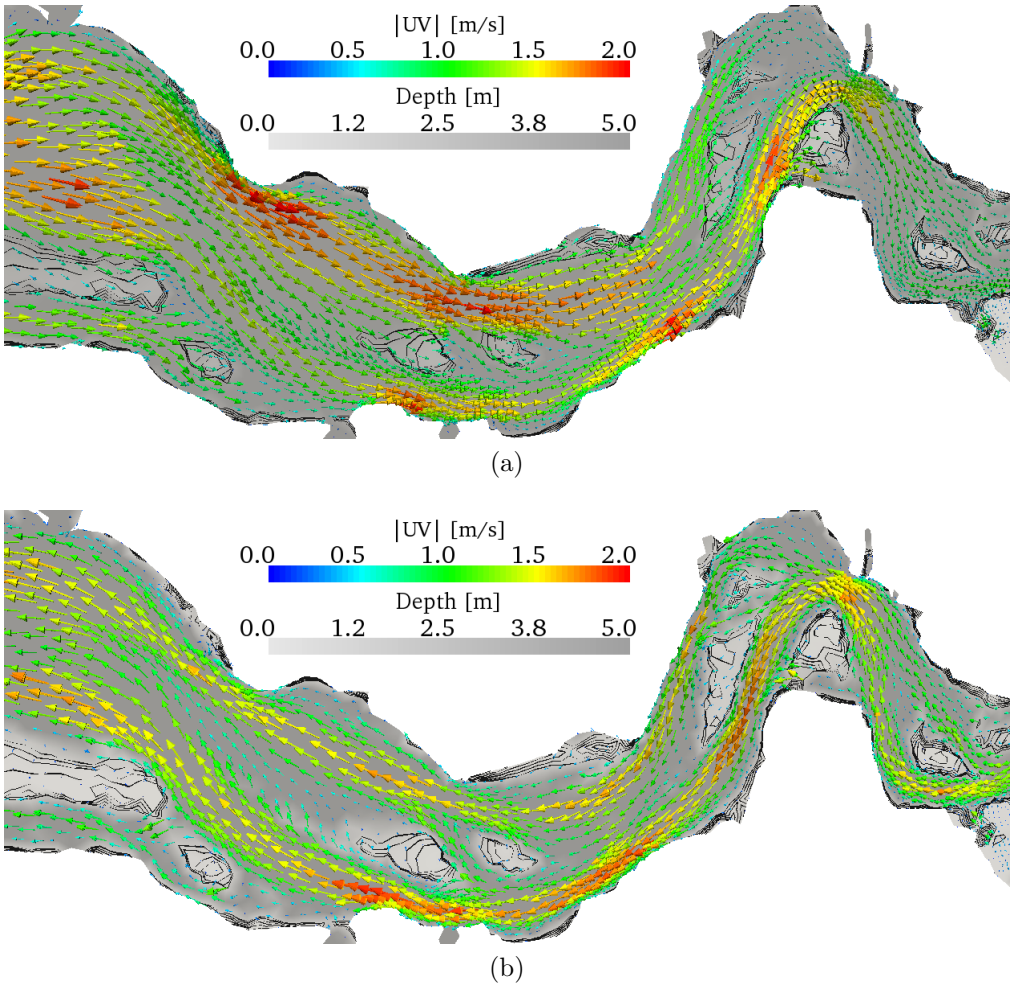
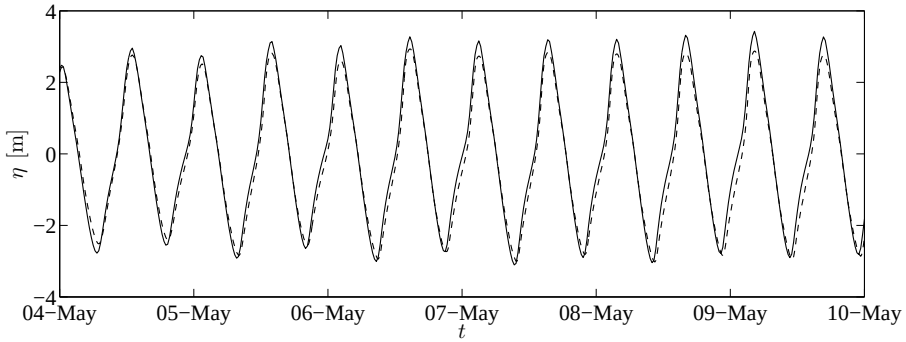


Figure 2.13: Velocity fields at a) flood and b) ebb tide.

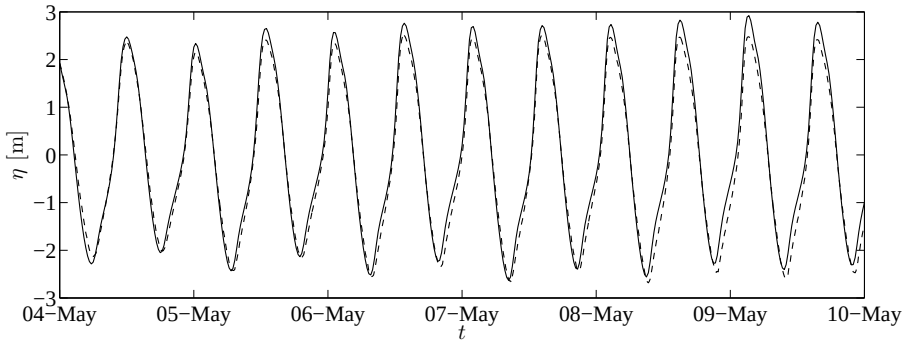
WD methods (Kennedy et al., 2000; Jiang and Wai, 2005; Nielsen and Apelt, 2003). Indeed, the moving bed was defined by means of a single-parameter function, which has proved out to be sufficient in all the various test cases. Moreover, in the presented formulation, mass is conserved up to machine precision (with respect to the modified total depth). The artificial retention volume due to the modified bathymetry remains small, and does not accumulate in time. In practice, the method has proved out to be very robust in various applications.

Since the main field of application of the SLIM is long-term transport simulations, we have paid special attention to ensure strict mass conservation and tracer consistency properties, which (especially for the latter) is not common in WD literature. The numerical tests confirm that the scheme is mass conservative, tracer mass is conserved up to numerical precision and no spurious transport due to the WD method appears.

The drawback of the moving bed approach is that water can leak through dry regions, as indicated by the third Balzano test case. The leakage is con-



(a) Bath



(b) Terneuzen

Figure 2.14: Simulated water elevation (solid line) versus measurements (dashed line) in two stations.

trolled by the parameter α, which generally depends on the mesh resolution and the bathymetry gradient. Consequently, the method is not well-suited for applications where the flow near lakes or other isolated reservoirs needs to be resolved with high accuracy. It should be stressed, however, that the same weakness is common to all "negative depth" WD methods.

The main advantage of the presented method lies in the computational efficiency due to the implicit time integration. Indeed, it is not obvious to develop a fully implicit WD method that is strictly mass conservative and consistent with tracer equation, especially in FE framework. To be able to really exploit the advantages of implicit time integration, higher order DIRK schemes are suggested here. When using long time steps of order ten minutes, simple implicit Euler time marching scheme proved out to be far too dissipative.

In the Scheldt application the total CPU time was reduced by a factor of 200 in comparison to explicit time stepping, that is often required in WD simulations. As the unstructured grid FE models tend to be slower than established FD models, this speed-up is essential in practical applications.

It should be stressed that although this paper deals with FE implementation, other formulations are not excluded, because the moving bed method can be described already at the level of the primitive equations.

In this work, we have concentrated on first order DG discretisation, although DG formulation is known to be computationally more efficient in higher order. However, designing WD treatment for high order models is still difficult, and in fact the author is not aware of any method applicable to order $p > 1$. Nevertheless, as the method presented here does not depend on the discretisation, it could potentially be extended to high order elements as well. In practice, the difficulty arises from the fact that for higher order basis functions the extrema are not necessarily located at the nodes.

CHAPTER 3

Coupling with a turbulence closure model

This Chapter reproduces an updated version of the following paper, first submitted on 31 July 2011:

T. Kärnä, V. Legat, E. Deleersnijder, and H. Burchard. Coupling of a discontinuous Galerkin finite element marine model with a finite difference turbulence closure model. *Ocean Modelling*, 47:55–64, 2012. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2012.01.001.

Abstract

This paper describes an online coupling between a 3D discontinuous Galerkin finite element marine model and a 1D vertical turbulence closure model based on finite differences. The coupling exploits the topology of the 3D mesh, that is formed by stacking layers of prisms in the vertical direction. A robust mapping between the finite difference grid and the finite element function space is designed, taking into account the discontinuities in the latter. The coupling is tested with two horizontally homogeneous flows and an idealised 3D estuary simulation. The results are in good agreement with those obtained with a finite difference model using the same turbulence closure, indicating that the coupling does not deteriorate the performance of the turbulence model.

3.1 Introduction

Three-dimensional marine models usually rely on Fourier-Fick parametrisations to represent vertical fluxes due to unresolved fluctuations. The relevant eddy coefficients are obtained by means of turbulence closure schemes. While in some applications simple formulations, such as algebraic expressions of eddy viscosity/diffusivity, can be sufficient, in general more sophisticated models are needed to account for the time-space evolution of the turbulent fluxes.

Most popular high-level turbulence models consist of two partial differential equations, one for the turbulent kinetic energy (TKE) and another one for an accompanying variable that determines the relevant length scale. Such turbulence closures include the widely used model by Mellor and Yamada (1982) (level 2.5), k–ε (Rodi, 1987), k–ω (Wilcox, 1988; Umlauf et al., 2003) and a recent generic length scale (GLS) model by Umlauf and Burchard (2003). Choosing a turbulence model is not trivial as it may have a high impact on mixing and circulation (Ruddick et al., 1995; Luyten et al., 1996; Burchard et al., 1998; Burchard, 2002a; Wijesekera et al., 2003; Warner et al., 2005).

GOTM[1] (General Ocean Turbulence Model, Burchard et al. (1999)) is a library that implements a generic turbulence closure model, in which all the above models can be easily obtained by changing parameters. GOTM is based on a finite difference (FD) formulation on a 1D vertical grid. It has been extensively tested and validated in numerous studies. Offering the flexibility to easily switch from one closure to another, GOTM is an advantageous tool for marine modelling.

GOTM has been coupled to many structured grid FD or finite volume (FV) models, including GETM (General Estuarine Transport Model, Burchard and Bolding (2002)) and MOM (Modular Ocean Model, Griffies (2010)). Enstad et al. (2008) studied CO_2 transport in a lake using GOTM with MITgcm (Massachusetts Institute of Technology general circulation model). Rygg et al. (2009) used GOTM with both MITgcm and BOM (Bergen Ocean Model) in a similar study. Also POLCOMS (Proudman Oceanographic Laboratory Coastal Ocean Modelling System) has been coupled to GOTM to simulate tidal mixing and stratification in the Northwest European Continental shelf (Holt and Umlauf, 2008). Among unstructured grid models GOTM has been coupled to FVCOM (Finite Volume Coastal Ocean Model, Chen et al. (2006); Tian and Chen (2006)).

Creating an interface between a 1D FD turbulence model and a FD/FV circulation model is fairly straightforward given the similarities in the grid and the mathematical representation of the fields. The purpose of this paper is to investigate the possibility of coupling GOTM to a discontinuous Galerkin (DG) finite element (FE) marine model. Such a coupling is feasible as long as the unstructured FE grid is vertically aligned, such as in the case of the commonly used prismatic mesh. Equipotential z-grids, terrain following σ-grids and their

[1] `www.gotm.net`

generalisations are equally applicable. However, due to the DG formulation, exchanging data becomes more complicated because fields are represented as a piecewise discontinuous polynomial on the domain, in contrast to a set of discrete values in FD.

FE marine models have been equipped with various turbulence closure models. FEOM (Finite Element Ocean Model,Wang (2007)) uses a Richardson number dependent Pacanowski and Philander (1981) parametrisation. A more sophisticated Mellor-Yamada level 2.5 closure has been implemented in a 1D water column model (Hanert et al. (2006, 2007); Blaise and Deleersnijder (2008)) and in a full 3D model in Blaise et al. (2007). The SELFE (Semi-implicit Eulerian-Lagrangian Finite Element) model implements the GLS turbulence closure (Zhang and Baptista, 2008). However, all these FE models are based on *continuous Galerkin* (CG) formulation, i.e. the basis functions are continuous between elements. White et al. (2008b) present a FE model that features discontinuous fields in the vertical direction, but rely on a simple parabolic parametrisation of eddy viscosity. Therefore, to our knowledge, sophisticated turbulence models in *discontinuous Galerkin* framework have not been dealt with so far.

As established models are nowadays mainly based on structured meshes and FD formulation, while unstructured mesh models are still emerging and mostly applied to regional studies, it is clear that there is a need to develop interfaces between the two model classes. This paper is a contribution to such a coupling. The aim is to take the best of both worlds, i.e. combining novel FE ocean model developments with an established FD turbulence library.

The article is organised as follows. The governing equations are presented in Section 3.2. Section 3.3 presents the numerical models: the 3D model and its DG-FE function space are briefly presented in Section 3.3.1, while the spatial discretisation and the interface of GOTM are outlined in Section 3.3.2. The coupling strategy is presented in Section 3.4. Numerical tests and concluding remarks are presented in Sections 3.5 and 3.6, respectively.

3.2 Governing equations

The 3D baroclinic equations are presented in Chapter 4, Section 4.2. To close this set of equations, the vertical eddy viscosity, ν, and diffusivity, μ, are solved by means of a turbulence closure model.

3.2.1 Turbulence closure models

Eddy viscosity and eddy diffusivity are calculated as proportional to a turbulence velocity scale $k^{1/2}$ (where k is the turbulent kinetic energy, TKE, per unit mass) and an integral turbulent length scale l:

$$\nu = c_\nu k^{1/2} l, \quad \mu = c_\mu k^{1/2} l. \tag{3.1}$$

The dimensionless proportionality factors, c_ν and c_μ, are stability functions, depending on non-dimensional shear and buoyancy frequency squared,

$$\alpha_M = \frac{M^2 l^2}{k}, \quad \alpha_N = \frac{N^2 l^2}{k}, \tag{3.2}$$

respectively. Here M^2 stands for the vertical shear frequency squared and N^2 is the vertical buoyancy (Brunt-Väisälä) frequency squared, defined as

$$M^2 = \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2, \tag{3.3}$$

$$N^2 = -\frac{g}{\rho_0}\frac{\partial \rho}{\partial z}. \tag{3.4}$$

Second moment turbulence closure models are derived from the Reynolds Averaged Navier-Stokes (RANS) equations, leading to transport equations for the Reynold stresses and turbulent tracer fluxes (Burchard, 2002a). These equations are then closed by assuming a local equilibrium and finding suitable second moment parametrisations for the unknown third moments. Classical second-moment closures are those by Mellor and Yamada (1982) and Canuto et al. (2001).

In the present framework, k and l are calculated by means of two budget equations. The k-equation is derived form the Navier-Stokes equations under the assumption of vertical shear layers (i.e. horizontal homogeneity of the flow) and the turbulent TKE transport being down-gradient. The resulting transport equation reads as

$$\frac{\partial k}{\partial t} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u} k) + \frac{\partial (w k)}{\partial z} = \frac{\partial}{\partial z}\left(\frac{\nu}{\sigma_k}\frac{\partial k}{\partial z}\right) + P + B - \varepsilon, \tag{3.5}$$

with the constant turbulent Schmidt number, σ_k, the shear and buoyancy production terms

$$P = \nu M^2, \quad B = -\mu N^2, \tag{3.6}$$

respectively, and the viscous dissipation rate per unit mass, ε. The latter can be calculated from k and l by means of

$$\varepsilon = \left(c_\mu^0\right)^3 \frac{k^{3/2}}{l}, \tag{3.7}$$

with the empirical dimensionless parameter c_μ^0.

The internal turbulent length scale l is calculated by means of the generic two-equation turbulence closure model developed by Umlauf and Burchard (2003). It introduces another transport equation for the generic quantity

$$\Psi = \left(c_\mu^0\right)^p k^m l^n, \tag{3.8}$$

with real numbers p, m and n.

Clearly, for $p = 3$, $m = 3/2$ and $n = -1$, $\Psi = \varepsilon$ is obtained. Along with the k-equation, this set of parameters results in the well-known k–ε model (Rodi, 1987). Other well known quantities for which budget equations have been derived are the turbulence frequency $\omega = \left(c_\mu^0\right)^{-1} k^{1/2} l^{-1}$ (Wilcox, 1988; Umlauf et al., 2003) and kl (Mellor and Yamada, 1982). The transport equation for Ψ as derived by Umlauf and Burchard (2003) reads

$$\frac{\partial \Psi}{\partial t} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u}\Psi) + \frac{\partial (w\Psi)}{\partial z} = \frac{\partial}{\partial z}\left(\frac{\nu}{\sigma_\Psi}\frac{\partial \Psi}{\partial z}\right) + \frac{\Psi}{k}\left(c_{\Psi 1} P + c_{\Psi 3} B - c_{\Psi 2}\varepsilon\right), \tag{3.9}$$

with the empirical parameters $c_{\Psi 1}$, $c_{\Psi 2}$, and $c_{\Psi 3}$, and the turbulent Schmidt number σ_Ψ. The transport equation (3.9) has been derived by combining (3.5) with the highly empirical ε-equation (Rodi, 1987). However, the empirical parameters have a clear physical meaning, as discussed by Umlauf and Burchard (2003). The role of $c_{\Psi 3}$ for properly adjusting the balance between increased mixing due to shear and decreased mixing due to stable stratification has been highlighted by Burchard and Baumert (1995); Umlauf and Burchard (2005) by analysing conditions for steady state solutions of (3.5) and (3.9) for homogeneous shear layers (i.e. zero gradients of k and Ψ):

$$P + B = \varepsilon, \quad c_{\Psi 1} P + c_{\Psi 3} B = c_{\Psi 2}\varepsilon, \tag{3.10}$$

which implies

$$R_i^{st} = \frac{c_{\Psi 1} - c_{\Psi 2}}{c_{\Psi 3} - c_{\Psi 2}} \cdot \frac{c_\mu\left(R_i^{st}\right)}{c_\nu\left(R_i^{st}\right)} \tag{3.11}$$

and

$$\Gamma^{st} = \frac{c_{\Psi 1} - c_{\Psi 2}}{c_{\Psi 3} - c_{\Psi 1}}, \tag{3.12}$$

with the steady state gradient Richardson number, $R_i^{st} = N^2/M^2$ and the steady state mixing efficiency, $\Gamma^{st} = -B/\varepsilon$, where M and N are shear and buoyancy frequency fulfilling (3.10). It should be noted that the stability functions c_ν and c_μ are functions of R_i^{st} only for turbulence equilibrium $P+B = \varepsilon$. By means of (3.11), $c_{\Psi 3}$ can be calculated as function of R_i^{st} which is expected to be $R_i^{st} = 0.25$ (Shih et al., 2000). Burchard and Hetland (2010) showed that when calculating $c_{\Psi 3}$ using the stability functions developed by Cheng et al. (2002), a steady-state mixing efficiency of $\Gamma^{st} = 0.22$ is obtained by means of (3.12), a value close to the estimate by Osborn (1980).

The most stable boundary conditions for k and l have proved to be Neumann conditions, which are generally derived from the law of the wall and depend on the surface friction velocity u_*^s and the bottom friction velocity u_*^b, unless surface wave breaking effects are considered (Umlauf and Burchard, 2005).

c_μ^0	$c_{\Psi 1}$	$c_{\Psi 2}$	$c_{\Psi 3}^-$	$c_{\Psi 3}^+$	σ_k	σ_Ψ
0.5265	1.44	1.92	-0.6209	1.00	1.00	1.30

Table 3.1: Parameters of the k–ε turbulence closure model. The parameter $c_{\Psi 3}^-$ (obtained with $R_i^{st} = 0.25$) is used in the case of stable stratification, while $c_{\Psi 3}^+$ is used for unstable stratification. For details refer to Burchard and Bolding (2001) and references therein.

In this work a k–ε turbulence closure is used with the parameters given in Table 3.1. The stability functions are those of Canuto et al. (2001) (Model A), which can be written as (Burchard and Bolding, 2001):

$$c_\nu = \frac{0.1070 + 0.01741\alpha_N - 0.00012\alpha_M}{C_{ca}}, \tag{3.13}$$

$$c_\mu = \frac{0.1120 + 0.004519\alpha_N - 0.00088\alpha_M}{C_{ca}}, \tag{3.14}$$

$$\begin{aligned} C_{ca} = {} & 1 + 0.2555\alpha_N + 0.02872\alpha_M + 0.008677\alpha_N^2 \\ & + 0.005222\alpha_N\alpha_M - 0.0000337\alpha_M^2. \end{aligned}$$

3.3 Numerical models

3.3.1 3D finite element ocean model

In this work we use the Second-generation Louvain-la-Neuve Ice-ocean Model (SLIM[2]). SLIM is based on the discontinuous Galerkin finite element method, and consists of 1D, 2D and 3D shallow water models.

SLIM 2D has been applied to various studies. Modelling the flow in the entire Great Barrier Reef, Australia, is presented in Lambrechts et al. (2008a) with validation against measurement data. Coupled 2D-1D model has been applied to modelling the Scheldt river, estuary and adjacent coastal zone, where the tidal water elevation is simulated with good accuracy across the multi-scale domain (de Brye et al. (2010)).

White et al. (2008b) present an early barotropic version of SLIM 3D model with semi-discontinuous function space. Full DG discretisation of the 3D baroclinic equations is presented in Blaise et al. (2010) and an implicit-explicit Runge-Kutta time integration method in Comblen et al. (2010a). The 3D simulations presented here are conducted using the 3D model described in Chapter 4.

Spatial discretisation of SLIM

For solving boundary value problems with the finite element method, the domain is divided into a finite number of elements. By means of the elements, a

[2] `www.climate.be/slim`

discrete function space, spanned by a set of basis functions, is defined. A FE solution belongs to this function space and approximates the exact solution of the boundary value problem in the sense of the L_2 norm. Consequently, a FE solution is a function defined in the entire domain, instead of a set of discrete values as in FD. Usually Lagrangian basis functions are used, which implies that the nodal values are interpolated inside the elements.

In SLIM the 3D mesh is made up of prismatic elements (see leftmost panel in Figure 3.1). Each element is a triangular prism of order o, that is formed as a product of a o-th order triangle in horizontal and o-th order linear element in vertical. As such, the solution is a piecewise polynomial of degree o in both horizontal and vertical directions. Due to the discontinuous Galerkin (DG) formulation, the solution is discontinuous at the element interfaces. The same spatial discretisation is used for all fields.

Assuming that there are n^E elements in the mesh, an element is identified by an integer $e = 1, \ldots, n^E$. Each element e contains n_e^N nodes, and consequently a node is identified by a pair of integers, (e, ξ), $\xi = 1, \ldots, n_e^N$. The set of all nodes (e, ξ) is denoted by $\mathcal{D}$. The nodal values of a field u are denoted by $u_{(e,\xi)}$. Denoting the ξ-th basis function of the element e as $\psi_{(e,\xi)}$, a field is expressed as $u_h = \sum_{(e,\xi)\in\mathcal{D}} u_{(e,\xi)}\psi_{(e,\xi)}(x, y, z)$, which is a discontinuous piecewise polynomial of degree o. In this work we are concentrating on first order elements, i.e. $o = 1$.

3.3.2 GOTM turbulence closure model

GOTM solves the equations (3.5) and (3.9), except for the advection terms that must be implemented in the 3D circulation model. GOTM uses a 1D vertical staggered grid extending from the bed to the fluctuating free surface, divided into q cells (see the rightmost panel in Figure 3.1). Internally, the mean flow variables are defined in the cell centres (crosses in Figure 3.1), while the turbulent variables are defined at cell interfaces. Thus the mean flow variables are denoted by $u_{m+1/2},\ m = 0, \ldots, q-1$ while the turbulent quantities read $\nu_m,\ m = 0, \ldots, q$.

GOTM interface

The variables required at runtime by GOTM are listed in Table 3.2. The user must provide the input variables at each time step. The output variables evolve in time and are updated by GOTM. If a single GOTM instance is used for computing several 1D segments, the output variables of the previous iteration must be provided. Consequently, one must store the arrays $\nu_{j,m}$, $\mu_{j,m}$, $k_{j,m}$, $\varepsilon_{j,m}$ for each vertical segment j.

User provided input		m
H	total depth	
u_*^s	surface friction velocity	
u_*^b	bottom friction velocity	
z_0^s	surface roughness length	
z_0^b	bottom roughness length	
$d_{m+1/2}$	cell height	$0, \ldots, q-1$
N_m	buoyancy frequency	$1, \ldots, q-1$
M_m	vert. shear frequency	$1, \ldots, q-1$

Output		m
ν_m	turbulent vert. viscosity	$0, \ldots, q$
μ_m	turbulent vert. diffusivity	$0, \ldots, q$
k_m	turbulence kinetic energy	$0, \ldots, q$
ε_m	TKE dissipation rate	$0, \ldots, q$

Table 3.2: Input and output variables in the GOTM interface with q cells.

3.4 Coupling strategy

3.4.1 Mapping nodes between 3D and 1D

In order to couple a 3D model with the 1D turbulence model, one needs to define vertical segments in the discontinuous 3D mesh, and build a mapping between the corresponding nodes.

In the simplest form, a 1D vertical array is built for each (discontinuous) node in the triangular surface mesh. Then there exists a bijective mapping Π_{DG} that maps each 3D node to a position i in a vertical array j. Using the nodal values of a field u in the 3D mesh, $u_{(e,\xi)}$, the values at the 1D array are denoted $u_{j,i} = \Pi_{\mathrm{DG}} u_{(e,\xi)}$, $i = 0, \ldots, n_j^{\mathrm{DG}} - 1$.

The 1D array defined above is discontinuous in the vertical. To account for the fact that the 1D FD grid has only continuous (unique) values at the cell interfaces, another mapping Π_{CG} is needed. Unlike above, Π_{CG} ignores the discontinuities in vertical, and hence it is no longer a bijection. Π_{CG} is used to fetch GOTM generated data back to the 3D fields, denoted by $\nu_{(e,\xi)} = \Pi_{\mathrm{CG}} \nu_{j,m}$. Consequently the field $\nu_{(e,\xi)}$ is continuous in the vertical direction.

The mappings Π_{DG} and Π_{CG} are illustrated in Figure 3.1. For first order elements it holds $n_j^{\mathrm{CG}} = n_j^{\mathrm{DG}}/2 + 1$.

Alternatively, the 1D vertical arrays can be located at the centroids of the surface triangles. In this case there exists only a single 1D array for each column of prisms. In order to fetch data from the 3D mesh to such an array, the FE basis functions are evaluated at the DG points corresponding to the triangle centroid, which can be seen as a generalisation of the map Π_{DG}. To map

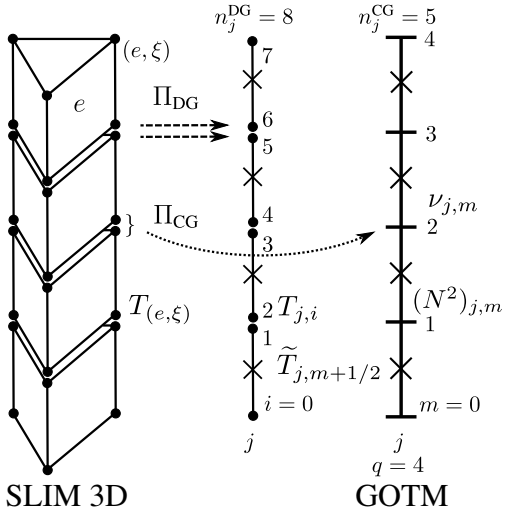


Figure 3.1: Schematic illustration of the coupling. The operator Π_{DG} maps 3D nodes (e, ξ) to DG nodes in the corresponding 1D vertical mesh. Π_{CG} maps 3D nodes to cell interfaces (j, m) in GOTM. For illustration purposes, the discontinuities of the SLIM 3D mesh has been exaggerated; in reality there is no gap between the elements.

turbulent quantities back to the 3D mesh, the 1D values are copied to the entire column of prisms, i.e. the values are constant in the horizontal, and linear continuous in the vertical direction. The 3D field is further smoothed in horizontal direction by taking a nodal average weighted by the volume associated to each node. This approach provides horizontal filtering on the input data, which improves the stability of the turbulence closure model in the presence of strong horizontal gradients.

3.4.2 Computing M and N

The key input parameters for GOTM are the vertical shear frequency M and buoyancy frequency N, defined in (3.3) and (3.4).

The buoyancy frequency requires the computation of the vertical gradient of the potential density, which is obtained by differentiating the equation of state at a constant pressure

$$\frac{\partial \rho}{\partial z} = A(T, S, p)\frac{\partial T}{\partial z} + B(T, S, p)\frac{\partial S}{\partial z}, \tag{3.15}$$

$$A = \frac{\partial \rho(T, S, p)}{\partial T}\Big|_{S,p}, \tag{3.16}$$

$$B = \frac{\partial \rho(T, S, p)}{\partial S}\Big|_{T,p}. \tag{3.17}$$

Consequently, for computing M and N, the vertical gradients of T, S, u and v need to be evaluated at the element interfaces. Here we present two different strategies for obtaining the gradients.

In FE discretisation the most intuitive way to evaluate gradients is by using the gradients of the basis functions. The vertical gradient of a field T is given as $\partial T/\partial z = \sum_{(e,\xi)\in\mathcal{D}} T_{(e,\xi)}\partial\psi_{(e,\xi)}/\partial z$, which is a discontinuous polynomial of degree $o-1$. The nodal values of such a field are given by

$$\left(\frac{\partial T}{\partial z}\right)_{(e,\xi)} = T_{(e,\xi)}\frac{\partial\psi_{(e,\xi)}}{\partial z}\Big|_{\boldsymbol{x}_{(e,\xi)}},\ \forall(e,\xi)\in\mathcal{D}$$

Using the mapping Π_{DG}, an array $\left(\partial T/\partial z\right)_{j,i},\ i=1,\dots,n_j^{\mathrm{DG}}$ is created for each vertical line j, assigning the appropriate values

$$\left(\frac{\partial T}{\partial z}\right)_{j,i} = \Pi_{\mathrm{DG}}\left(\frac{\partial T}{\partial z}\right)_{(e,\xi)}. \tag{3.18}$$

The above values are discontinuous at the element interfaces. An estimate of the gradient can be obtained by taking an arithmetic mean

$$\left(\widehat{\frac{\partial T}{\partial z}}\right)_{j,m} = \frac{1}{2}\left[\left(\frac{\partial T}{\partial z}\right)_{j,2m} + \left(\frac{\partial T}{\partial z}\right)_{j,2m-1}\right],\quad m=1,\dots,n_j^{\mathrm{CG}}-2. \tag{3.19}$$

The drawback of (3.19) is that it only uses the gradient in each element, thus ignoring the jump at the interface. The gradient fields also tend to be more noisy than the corresponding scalar fields.

Another possibility is to use finite differencing across the element interface. First, the field nodal values are fetched in the 1D arrays

$$T_{j,i} = \Pi_{\mathrm{DG}}T_{(e,\xi)}. \tag{3.20}$$

Denoting the z coordinates of each node by $z_{j,i}$, element heights and the total depth are obtained as

$$d_{j,m+1/2} = z_{j,2m+1} - z_{j,2m},\ l=0,\dots,n_j^{\mathrm{CG}}-2, \tag{3.21}$$

$$H_j = z_{j,n_j^{\mathrm{DG}}-1} - z_{j,0}. \tag{3.22}$$

Next, T is evaluated at element centres using the 1D DG-FE basis functions $\psi_i(\zeta),\ i=1,\dots,n_{1D}^N$, defined on a reference element $\zeta\in[-1,1]$. For first order elements $n_{1D}^N=2$ and we can write (see Figure 3.1):

$$\widetilde{T}_{j,m+1/2} = T_{j,2m}\psi_1(0) + T_{j,2m+1}\psi_2(0),\quad m=0,\dots,n_j^{\mathrm{CG}}-2 \tag{3.23}$$

For Lagrangian basis functions, we have $\psi_1(0)=\psi_2(0)=1/2$, implying that the above is equivalent to an arithmetic mean of the nodal values.

Now the vertical gradient of T can be estimated by finite differencing

$$\begin{aligned} \tilde{d}_{j,m} &= \frac{1}{2}\left(d_{j,m+1/2} + d_{j,m-1/2}\right), \\ \left(\widehat{\frac{\partial T}{\partial z}}\right)_{j,m} &= \frac{\widetilde{T}_{j,m+1/2} - \widetilde{T}_{j,m-1/2}}{\tilde{d}_{j,m}}, \\ m &= 1, \ldots, n_j^{\mathrm{CG}} - 2 \end{aligned} \tag{3.24}$$

The formulation (3.24) is more accurate because the field values are more reliable at the element centres. Clearly, (3.24) depends on a stencil of 4 nodal values and thus ignores neither the jump nor the gradient at the interface. To obtain better estimates, more sophisticated interpolation methods could be used, but those are not dealt with in this article.

Once the gradients are obtained, the buoyancy frequency squared is computed as

$$(N^2)_{j,m} = -\frac{g}{\rho_0}\left[A_{j,m}\left(\widehat{\frac{\partial T}{\partial z}}\right)_{j,m} + B_{j,m}\left(\widehat{\frac{\partial S}{\partial z}}\right)_{j,m}\right], \tag{3.25}$$

where $A_{j,m}$ and $B_{j,m}$ are computed with the mean temperature and salinity at the interface.

The vertical shear frequency is computed in a similar fashion. Denoting the shear frequency in x direction by M_x, the following temporal averages are defined with respect to the velocity field of the previous iteration:

$$\begin{aligned} (\bar{M}_x)_{j,m} &= \delta\left(\widehat{\frac{\partial u}{\partial z}}\right)_{j,m} + (1-\delta)\left(\widehat{\frac{\partial u}{\partial z}}\right)^{\mathrm{old}}_{j,m}, \\ (\bar{M}'_x)_{j,m} &= \frac{1}{2}\left(\widehat{\frac{\partial u}{\partial z}}\right)_{j,m} + \frac{1}{2}\left(\widehat{\frac{\partial u}{\partial z}}\right)^{\mathrm{old}}_{j,m}. \end{aligned}$$

The implicity parameter $\delta \in [0,1]$ depends on the temporal scheme of vertical diffusion. Here, the Crank-Nicholson scheme is used for vertical diffusion, so that $\delta = 1/2$.

Now, the square of vertical shear frequency in x direction is:

$$\begin{aligned} (M_x^2)_{j,m} &= (\bar{M}_x)_{j,m}(\bar{M}'_x)_{j,m}, \\ m &= 1, \ldots, n_j^{\mathrm{CG}} - 2 \end{aligned} \tag{3.26}$$

The shear frequency in y direction, M_y, is computed analogously, with the respective velocity field v. Finally the shear frequency is given by

$$M = \sqrt{M_x^2 + M_y^2}. \tag{3.27}$$

The temporal averaging in (3.26), proposed by Burchard (2002b), guarantees that the (Reynolds averaged) kinetic energy is transformed to turbulent kinetic

energy in a conservative manner, which improves the numerical stability of the TKE computation.

The above formulation for computing N and M is derived for first order Lagrangian DG basis functions, but it similar formulae can be derived for other (e.g. higher order or continuous) function spaces as well.

3.4.3 Advection of k and Ψ

The equations (3.5) and (3.9) contain advection terms for k and Ψ, respectively. These terms are not included in the 1D vertical turbulence closure model that deals only with turbulent processes. Therefore k and Ψ are advected in the 3D model as a passive tracer using the same Adams-Bashforth scheme as for T and S.

For stability it is crucial to ensure strict positivity of these variables which may be challenging due to strong gradients. To avoid spurious extrema in these fields, a slope limiter is used as for other tracers (see Chapter z4). Slightly negative values (of the order of machine precision) may still appear, which are clipped to a small positive value.

3.5 Numerical tests

In order to validate the presented coupling, a number of numerical tests were conducted. All the tests were run on SLIM 3D coupled to GOTM.

In typical estuarine conditions the k–ε, k–ω and (improved) Mellor-Yamada level 2.5 turbulence closures have proven to produce similar results (e.g. Warner et al., 2005). In these tests a k–ε turbulence closure is used with the parameters given in Table 3.1.

3.5.1 Bottom boundary layer

The 3D model's ability to reproduce bottom boundary layer was assessed with pressure gradient driven free flow. The fluid is initially at rest, forced only by a constant free surface slope. In the absence of rotation, the flow velocity near the bottom boundary follows the usual logarithmic profile, which can be expressed as (e.g. Hanert et al. (2007))

$$\boldsymbol{u}(z) = \frac{\boldsymbol{u}_*^b}{\kappa} \log\left(\frac{z_0^b + z + h}{z_0^b}\right), \tag{3.28}$$

where $\boldsymbol{u}_*^b$ is the bottom friction velocity and κ is the von Karman constant.

In accordance with (3.28), the conventional quadratic friction law is imposed at the bottom:

$$\nu \frac{\partial \boldsymbol{u}}{\partial z} = c_d |\boldsymbol{u}_b| \boldsymbol{u}_b, \tag{3.29}$$

$$c_d = \left(\frac{\kappa}{\log(\frac{z_b + h + z_0^b}{z_0^b}))} \right)^2, \tag{3.30}$$

where c_d is the drag coefficient, z_b is the vertical coordinate at the middle of the bottom most element and $\boldsymbol{u}_b = \boldsymbol{u}(z_b)$. Using (3.28) and (3.30), the bottom friction velocity is obtained as $\boldsymbol{u}_*^b = \sqrt{c_d}\boldsymbol{u}_b$.

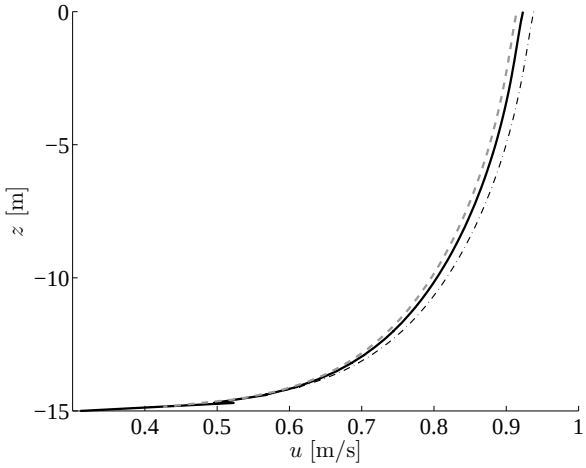


Figure 3.2: Steady-state velocity profile for the bottom boundary layer test. Solid line, SLIM+GOTM; Dashed line, GOTM; Dash-dotted line, SLIM+GOTM with FE gradients.

Pressure gradient driven free flow was simulated in a 10 km by 10 km square domain in horizontal, 15 m deep, with 30 cm vertical resolution. Throughout the simulation the free surface slope was fixed to -10^{-5} in x direction. Bottom roughness length was $z_0^b = 1.5$ mm. The simulation was run for 24 h until it reached a steady state, i.e. a balance between pressure gradient force and friction. The shear frequency was computed either with finite differencing according to (3.24) or by means of the FE basis functions (3.19). For reference, the same simulation was performed with 1D GOTM alone using its FD mean flow module [3].

The steady state vertical velocity profile is presented in Figure 3.2. SLIM+GOTM produce very similar profile to GOTM when the FD gradients are used. With the FE gradients, on the other hand, the flow velocity is overestimated in the upper part of the water column.

[3] Since the flow is horizontally homogeneous, a 1D vertical model and a 3D model produce comparable results. As a 3D FE model (in contrast to common FD models) cannot be reduced to a 1D vertical model, we are using full SLIM 3D here.

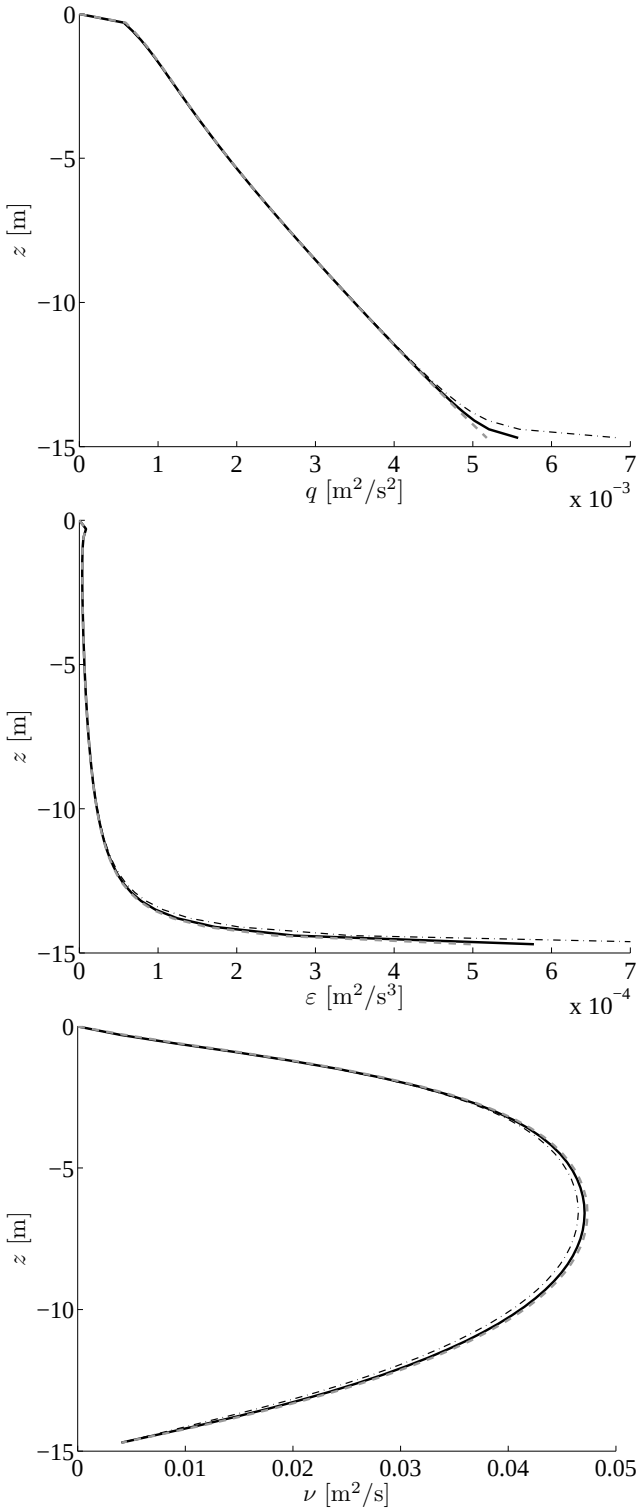


Figure 3.3: Profiles of turbulent kinetic energy (top), TKE dissipation rate (middle) and eddy viscosity (bottom). Solid line, SLIM+GOTM; Dashed line, GOTM; Dash-dotted line, SLIM+GOTM with FE gradients.

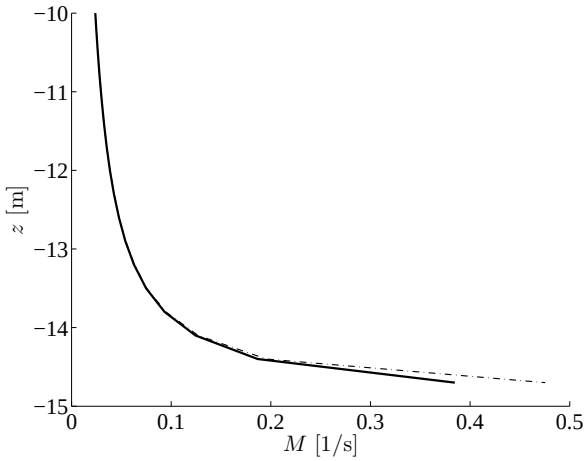


Figure 3.4: Vertical shear frequency at the lower part of the water column computed with the FD gradients (solid line) and the FE gradients (dash-dotted line) for the same velocity profile. Here the velocity profile obtained with SLIM+GOTM and FD gradients (solid line in Figure 3.2) is used.

Vertical profiles of turbulent kinetic energy, dissipation rate and turbulent viscosity are presented in Figure 3.3. The TKE profile obtained with SLIM+GOTM corresponds very closely to that of GOTM, except at the bottom boundary where TKE is higher. It is seen that the deviation is much larger in the case of FE gradients.

Based on these results, the numerical method for computing vertical gradients has a significant impact on the simulated turbulence. Figure 3.4 compares the vertical shear frequency M computed with both FD and FE gradients for the same velocity profile. It is seen that FE gradients produce higher M near the bottom, but the difference is confined only to the two bottom most elements. In the central part of the water column, where the velocity profile is smoother, the two methods are in good agreement.

Because $P = \nu M^2$ appears as a source term for both the k and ε equations, overestimation of M explains the high values of TKE and ϵ near the bottom seen in Figure 3.3. In the central part of the water column, TKE is almost unaffected, but ε is still slightly overestimated by roughly 1% (not shown), which results in smaller eddy viscosity (bottom panel in Figure 3.3), and higher flow velocity (Figure 3.2).

Due to the fact that FD gradients also overestimate the bottom TKE slightly, it is plausible that M is somewhat overestimated in this case as well, but resulting difference in the mixing is much smaller.

3.5.2 Wind-driven entrainment

The next test examines mixed layer deepening due to surface stress, based on the laboratory experiment originally conducted by Kato and Phillips (1969).

Initially the fluid is motionless and linearly stratified. A constant surface stress is applied at the surface. As a consequence, a mixed layer is formed at the surface, growing deeper in time. Price (1979) suggested a formula for the evolution of the mixed layer depth:

$$d_{ML} = 1.05 u_*^s \sqrt{\frac{t}{N_0}}. \tag{3.31}$$

Here the surface friction velocity is taken as a constant $u_*^s = 0.01\ \mathrm{m\,s^{-1}}$ while the initial Brunt-Väisälä frequency is $N_0 = 0.01\ \mathrm{s^{-1}}$ following Deleersnijder and Luyten (1994) and Burchard et al. (1998), among others. In practice N_0 is prescribed by imposing a suitable vertical density gradient.

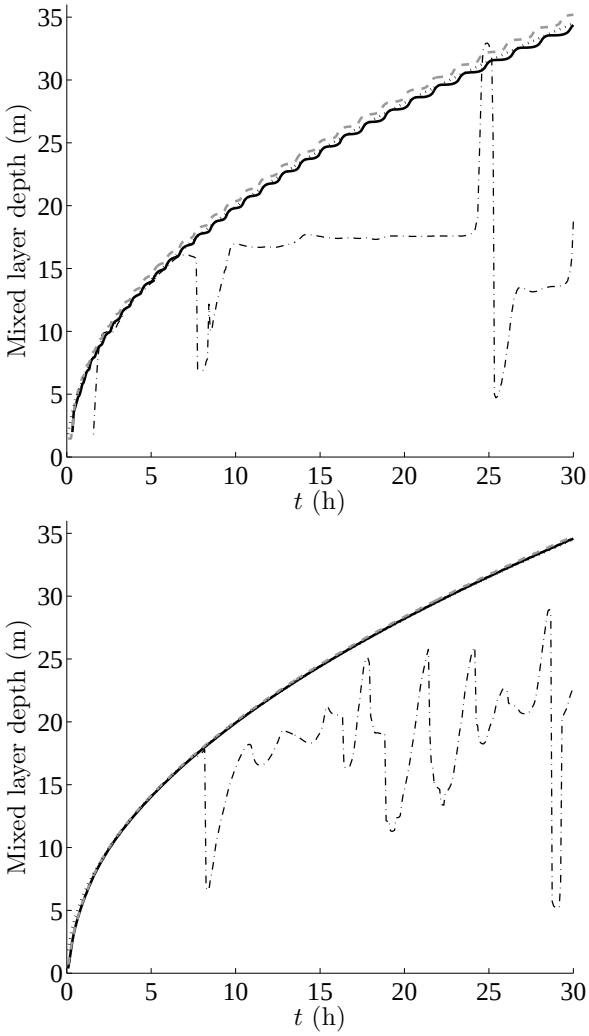


Figure 3.5: Mixed layer depth versus time for the Kato-Phillips test with 1 m (top) and 20 cm (bottom) vertical resolution. Solid line, SLIM+GOTM; Dashed line, GOTM. Dotted line, the solution by Price (3.31); Dash-dotted line, SLIM+GOTM with FE gradients.

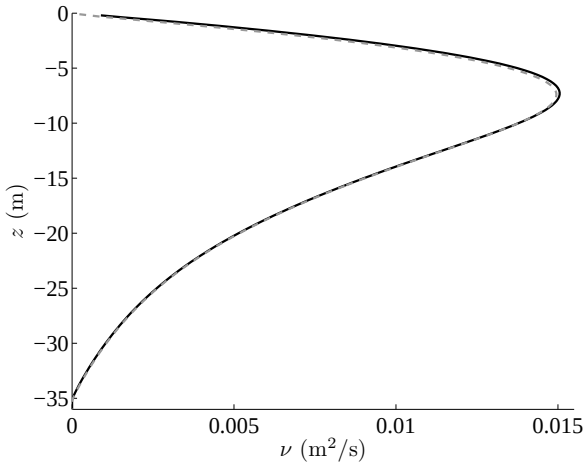


Figure 3.6: Turbulent viscosity with 20cm vertical resolution. Solid line, SLIM+GOTM; Dashed line, GOTM. Results obtained with the FE gradients are omitted.

The mixed layer entrainment was simulated in a 50 m deep water column for 30 h. Two vertical resolutions were investigated, namely 1 m and 20 cm. Again, the vertical gradients were computed either with the FD (3.24) or the FE formulation (3.19).

Previously it has been concluded that the model's capability to predict the mixed layer deepening depends more on the choice of the stability functions rather that the two-equation model itself (Deleersnijder and Luyten (1994), Burchard and Deleersnijder (2001), Umlauf et al. (2003), Deleersnijder et al. (2008)). As stated above, we are using the stability functions by Canuto et al. (2001), that have proved out to perform well in this test case.

Evolution of the mixed layer depth, defined as deepest point where $k > 10^{-5}\,\mathrm{m^2\,s^{-2}}$, is shown in Figure 3.5 for the two resolutions. The results obtained by SLIM+GOTM (using the FD gradients) are very close to those by GOTM, and both agree well with the formula by Price (3.31). For a coarser resolution, the mixed layer depth oscillates as mixing penetrates new cells. This is in line with other results, such as in Burchard and Deleersnijder (2001). Turbulent viscosity profile after 30 h of simulation is shown in Figure 3.6. The profiles are also very similar, SLIM+GOTM appears to produce marginally larger maximum viscosity.

Using the FE gradients, on the other hand, cause spurious behaviour: the mixed layer depth, defined as the deepest point with significant TKE, oscillates with time. This is due to the fact that the N and M fields are noisy, which occasionally triggers the turbulence model to produce high eddy viscosity at certain nodes, thus mixing the water column in a non-homogeneous manner. Due to these defects, only the more robust FD formulation (3.24) is considered in the next 3D test case.

3.5.3 Idealised estuarine circulation

The idealised estuarine scenario proposed by Warner et al. (2005) is used here to verify the behaviour of the turbulence model in a full three-dimensional setting.

Parameter	Symbol	Value
Domain dimensions	D_x, D_y	100 km, 2 km
Mesh lenght scale	L_{xy}	1 km
Vertical layers	n_σ	20
Bathymetry (river)	h_R	5 m
Bathymetry (sea)	h_S	10 m
Salinity (river)	S_R	0 PSU
Salinity (sea)	S_S	30 PSU
Temperature	T_0	10 °C
Tidal period	τ	12 h
Residual velocity	$\bar{U}_0$	-0.08 $\mathrm{m\,s^{-1}}$
Tidal velocity	$\bar{U}_\tau$	$0.4\sin(\frac{2\pi t}{\tau})$ $\mathrm{m\,s^{-1}}$
Depth av. velocity	$\bar{u}_R$	$\frac{\bar{U}_0 h_R}{\eta_R + h_R}$ $\mathrm{m\,s^{-1}}$
Depth av. velocity	$\bar{u}_S$	$\frac{\bar{U}_0 h_S + \bar{U}_\tau h_R}{\eta_S + h_S}$ $\mathrm{m\,s^{-1}}$
Bottom roughness	z_0^b	0.005 m

Table 3.3: Parameters of the estuary simulation. The subscripts R and S stand for river and sea boundary, respectively. $\bar{U}_0$ and $\bar{U}_\tau$ are the depth averaged residual and tidal velocity, respectively, assuming static water depth (h_R and h_S). At the boundaries, $\bar{u}_R$ and $\bar{u}_S$ are prescribed to account for the free surface movement and guarantee constant water volume over a tidal period. Symmetry boundary conditions are used for the three-dimensional velocity. η_R and η_S are taken as the simulated values on the respective boundary.

The domain is a rectangular basin 100 km long, whose depth varies linearly from 10 m in the "ocean" end (x=0 km) to 5 m in the "river" end (x=100 km). In the cross-channel direction the domain is taken to be 2 km in width with impermeable lateral boundaries. The domain is discretised with 1 km horizontal resolution and 20 sigma layers in the vertical, resulting in 0.25 m to 0.50 m vertical resolution. The mesh is illustrated in Figure 3.7 and all the model parameters are presented in Table 3.3. A constant seaward freshwater discharge is applied at the river boundary, while a sinusoidal tidal flow with 12 h period is prescribed at the ocean boundary. The Coriolis force is neglected. Initially the salinity varies linearly from 30 PSU to 0 PSU between 30 km and

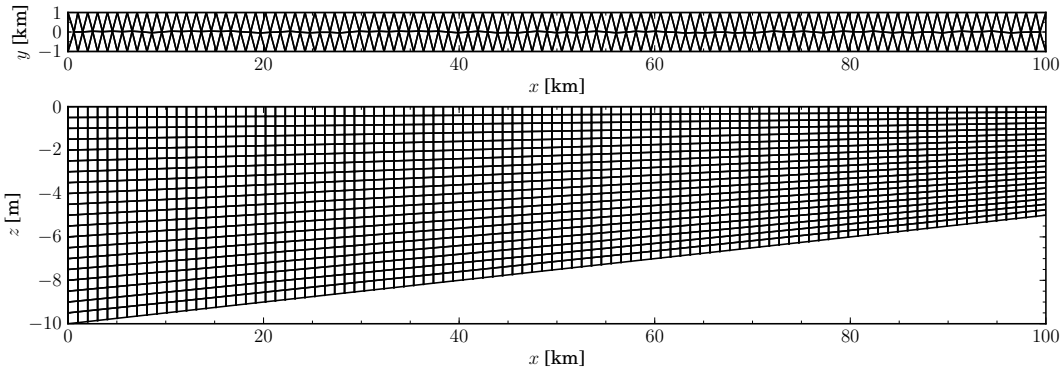


Figure 3.7: Estuary simulation: 3D Mesh. Horizontal edge length is roughly 1 km. The mesh consist of 396 surface triangles and 7920 prisms.

80 km along the channel (Figure 3.8). At the ocean boundary the salinity is prescribed to 30 PSU during inflow.

During the simulation, an estuarine circulation quickly develops, driving the saline water under the fresh water. The forming salt wedge oscillates with the tide. The flow stabilises and becomes nearly periodic after roughly 5 days of simulation, when only a small seaward salt flux persists. Warner et al. (2005) designed this test case to compare different turbulence closure models. Indeed, the salinity distribution is largely controlled by vertical mixing, therefore providing a useful benchmark for the SLIM+GOTM coupling.

In this simulation, the 1D turbulence closure models are placed at the centroids of the triangles, as mentioned in Section 3.4.1. This provides horizontal smoothing of the input fields which improves the stability of the simulation[4]. Similar horizontal filtering is also used in FD models (e.g. Burchard and Bolding, 2002). Furthermore, similarly to Warner et al. (2005), the turbulent length scale l is limited from above following Galperin et al. (1988), which also reduces noise in the turbulent quantities.

Figure 3.8 shows the salt distribution at the end of flood phase, after 16 days of simulation. The salt intrusion is similar to the results by Warner et al. (2005) (Figure 3.9), where the 0 PSU contour line is located around 60 km. With the presented model, the same contour line is somewhat further at 72 km. The surface mixed layer is confined in the first couple of metres of the water column, as in Warner et al. (2005).

Figure 3.10 presents vertical profiles taken at x=30 km after 14.40 days of simulation (corresponding to a typical flood tide). Qualitatively these profiles are similar to those presented in Warner et al. (2005) (Figure 3.11). The salinity profile predicted by SLIM+GOTM is identical, roughly 20 PSU in magnitude. The turbulent length scale is roughly 30% smaller. Turbulent

[4] Note that applying such a filter would have no effect in the two previous test cases where the flow was homogeneous in the horizontal direction.

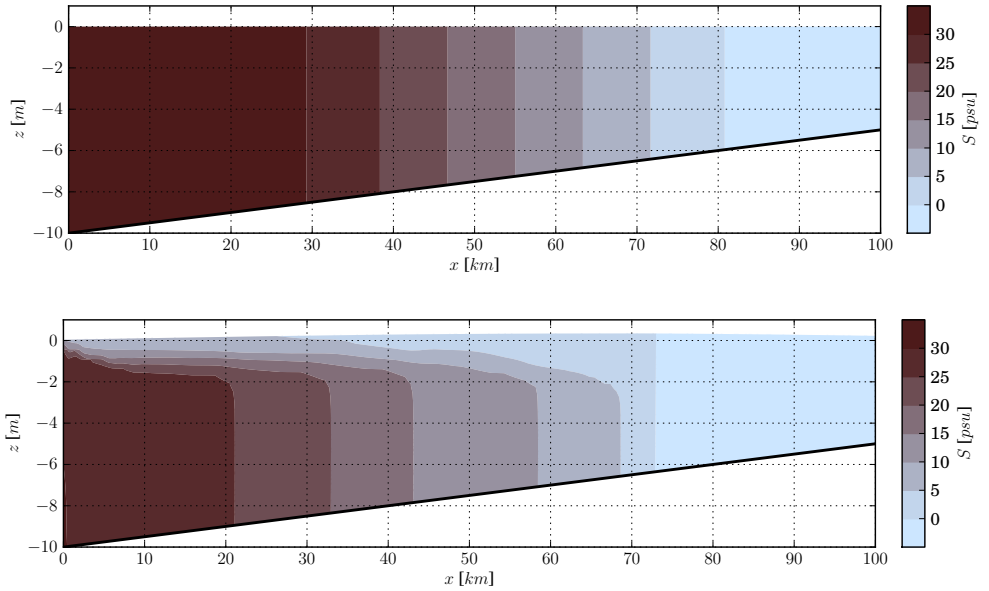


Figure 3.8: Estuary simulation: Top: Initial salinity distribution. Bottom: Salinity after 16 days of simulation, obtained with a k–ε model.

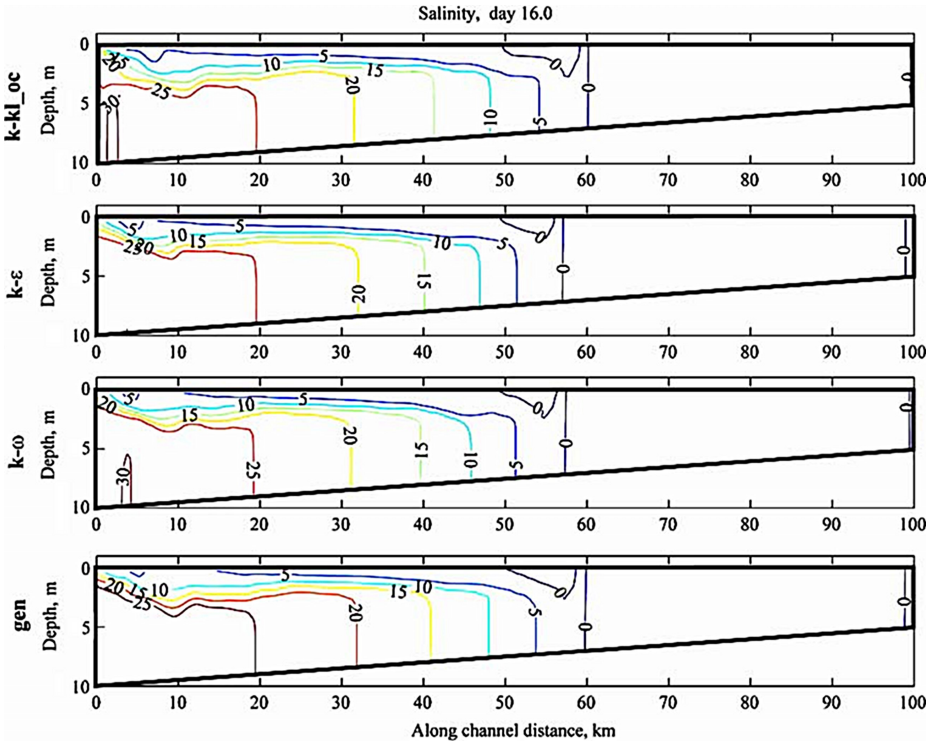


Figure 3.9: Estuary simulation: Salinity distribution from Warner et al. (2005) (Figure 7 in their manuscript), obtained with different turbulence closure models.

eddy viscosity is also underestimated, the maximum value is 0.011 m^2s^{-1} versus nearly 0.02 m^2s^{-1}.

Taking into account that the turbulent quantities are very sensitive to the characteristics of the flow and details of the turbulence closure, one can conclude that the coupled SLIM+GOTM model produces the expected flow features with good accuracy. In addition to differences in the turbulence closure models, also other aspects, such as the boundary conditions or numerical mixing (Burchard and Rennau, 2008; Rennau and Burchard, 2009) may have a significant impact on the distribution of salinity and turbulent quantities. As Warner et al. (2005) used a structured grid FD model, model-dependent features are likely to play a role. However, assessing such differences is out of the scope of the current article.

3.6 Conclusions

Implementing turbulence closure models in a discontinuous Galerkin framework is not often addressed in the literature. We have presented an online coupling between a 3D DG-FE marine model and a FD 1D vertical turbulence model, which exploits the vertically orientated topology of the 3D mesh. We demonstrate that such a coupling is not trivial due to the different mesh topology and the discontinuous functional representation of fields.

Ensuring stability of the turbulence closure model is an essential part of the presented methodology. Stability is achieved by suitable data processing, i.e. computing the vertical gradients more reliably at the element interfaces by taking into account all the nodal values in the elements above and below. Further, in horizontal direction the 1D vertical turbulence models are placed at the centre of each column of prisms, which provides horizontal filtering of the input data.

The coupling has been validated with several test cases. Bottom boundary layer is produced accurately, TKE and viscosity profiles are close to those produced by GOTM. The mixed layer deepening in the Kato-Phillips test case is also correctly predicted. Finally the 3D implementation is validated in an idealised estuary simulation where the results are well in line with those presented in Warner et al. (2005).

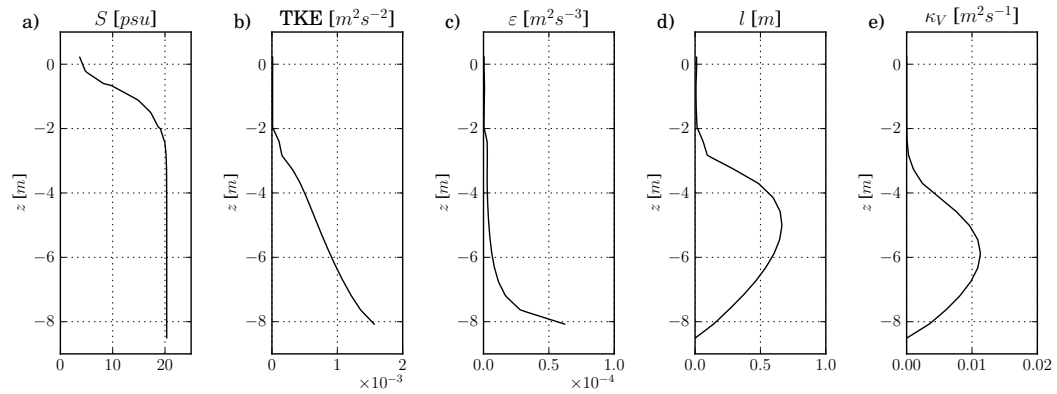


Figure 3.10: Estuary simulation: Vertical profiles after 14.4 days of simulation at $x = 30$ km. a) salinity, b) turbulent kinetic energy, c) TKE dissipation rate, d) turbulent length scale, e) eddy diffusivity.

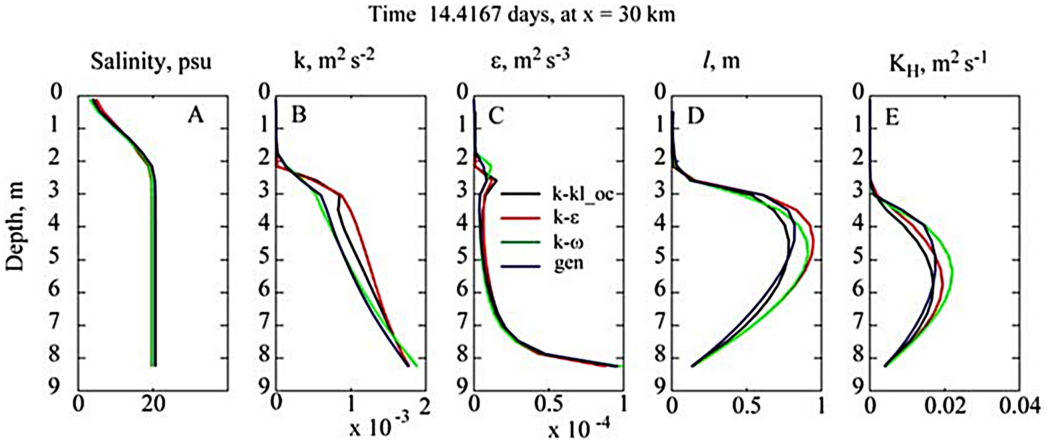


Figure 3.11: Estuary simulation: Vertical profiles from Warner et al. (2005) (Figure 9 in their manuscript), obtained with different turbulence closure models.

CHAPTER 4

Baroclinic three-dimensional marine model

This Chapter reproduces an updated version of the following paper, first submitted in March 26 2012:

T. Kärnä, V. Legat, and E. Deleersnijder. Baroclinic discontinuous Galerkin finite element model for coastal flows. *Ocean modelling*, submitted.

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.

4.1 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 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. 2008b). 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) (FVCOM, 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 an uniform tracer field is preserved (White et al., 2008a). 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 4.2. In the next section the ALE formulation and moving mesh algorithm are described. Section 4.4 presents the DG-FE discretisation. A matrix form of the discrete equations is given in Section 4.5. The time integration scheme is presented in Section 4.6. Finally Section 4.7 presents numerical results on conservation properties, gravitational adjustment flow and an application to a river plume simulation.

4.2 Governing equations

4.2.1 Domain

The three-dimensional computational domain is denoted by Ω. The position vector in Cartesian coordinates is $\boldsymbol{x} = [x, y, z]$, and the associated velocity components are u, v and w. The horizontal velocity vector is denoted by $\boldsymbol{u} = [u, v, 0]$. 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 4.1).

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 $\boldsymbol{x}_h = [x, y, 0]$.

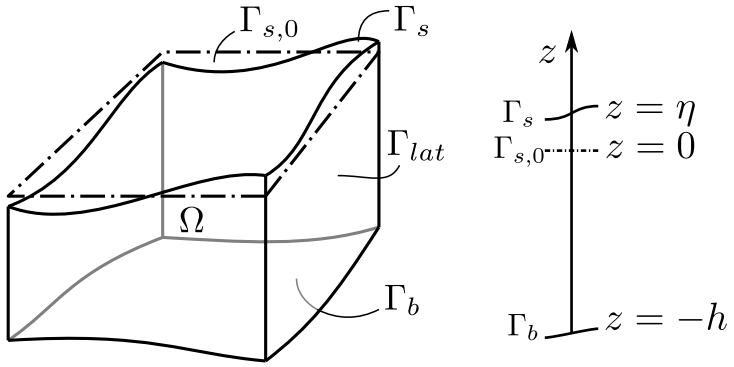


Figure 4.1: Possible configuration of the computational domain Ω at some time t. The reference surface $\Gamma_{s,0}$, marked with a dashed line, corresponds to $\eta = 0$.

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 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 4.3.

4.2.2 Momentum equation

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

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u}\boldsymbol{u}) + \frac{\partial (w\boldsymbol{u})}{\partial z} + f\boldsymbol{e}_z \wedge \boldsymbol{u} + \frac{1}{\rho_0}\boldsymbol{\nabla}_h p \\ = \boldsymbol{\nabla}_h \cdot (\nu_h \boldsymbol{\nabla}_h \boldsymbol{u}) + \frac{\partial}{\partial z}\left(\nu \frac{\partial \boldsymbol{u}}{\partial z}\right), \end{aligned} \tag{4.1}$$

where $\boldsymbol{\nabla}_h$ is the horizontal gradient operator, f the Coriolis factor, $\boldsymbol{e}_z$ 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

$$\frac{\partial p}{\partial z} = -g\rho, \tag{4.2}$$

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 $\rho' \ll \rho_0$:

$$\rho = \rho_0 + \rho', \tag{4.3}$$

Integrating (4.2) 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

$$r = \frac{1}{\rho_0}\int_z^\eta \rho' d\zeta, \tag{4.4}$$

the pressure gradient term appearing in (4.1) can be written as

$$\frac{1}{\rho_0}\boldsymbol{\nabla}_h p = g\boldsymbol{\nabla}_h \eta + g\boldsymbol{\nabla}_h r + \frac{1}{\rho_0}\boldsymbol{\nabla}_h p_a. \tag{4.5}$$

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

4.2.3 Continuity and free surface equations

The continuity equation is given by

$$\boldsymbol{\nabla}_h \cdot \boldsymbol{u} + \frac{\partial w}{\partial z} = 0, \tag{4.6}$$

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,

$$\frac{\partial \eta}{\partial t} + \boldsymbol{\nabla}_h \cdot \int_{-h}^{\eta} \boldsymbol{u} dz = 0. \tag{4.7}$$

4.2.4 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

$$\frac{\partial T}{\partial t} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u} T) + \frac{\partial (wT)}{\partial z} = \boldsymbol{\nabla}_h \cdot (\mu_h \boldsymbol{\nabla}_h T) + \frac{\partial}{\partial z}\left(\mu \frac{\partial T}{\partial z}\right), \tag{4.8}$$

$$\frac{\partial S}{\partial t} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u} S) + \frac{\partial (wS)}{\partial z} = \boldsymbol{\nabla}_h \cdot (\mu_h \boldsymbol{\nabla}_h S) + \frac{\partial}{\partial z}\left(\mu \frac{\partial S}{\partial z}\right). \tag{4.9}$$

4.2.5 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 GOTM[1] (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).

4.2.6 Boundary conditions

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

$$w + \boldsymbol{u} \cdot \boldsymbol{\nabla}_h h = 0, \quad \boldsymbol{x} \in \Gamma_b \tag{4.10}$$

$$w - \frac{\partial \eta}{\partial t} - \boldsymbol{u} \cdot \boldsymbol{\nabla}_h \eta = 0, \quad \boldsymbol{x} \in \Gamma_s. \tag{4.11}$$

[1] `www.gotm.net`

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

$$\nu \frac{\partial \boldsymbol{u}}{\partial z} = \frac{\boldsymbol{\tau}_b}{\rho_0}, \quad \boldsymbol{x} \in \Gamma_b \tag{4.12}$$

$$\nu \frac{\partial \boldsymbol{u}}{\partial z} = \frac{\boldsymbol{\tau}_s}{\rho_0}, \quad \boldsymbol{x} \in \Gamma_s. \tag{4.13}$$

The bottom stress is given by

$$\begin{aligned} \frac{\boldsymbol{\tau}_b}{\rho_0} &= C_d \|\boldsymbol{u}_{\mathrm{bf}}\| \boldsymbol{u}_{\mathrm{bf}}, \\ C_d &= \left[\frac{\kappa}{ln((z_{\mathrm{bf}} + h)/\delta_0)} \right]^2, \\ \boldsymbol{u}_{\mathrm{bf}} &= \boldsymbol{u}(x, y, z_{\mathrm{bf}}), \end{aligned} \tag{4.14}$$

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 $\boldsymbol{u}_{\mathrm{bf}}$ is defined.

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

$$\boldsymbol{u} \cdot \boldsymbol{n}_h = 0, \tag{4.15}$$

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

4.2.7 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 $\bar{\boldsymbol{u}}$ and the decomposition $\boldsymbol{u} = \bar{\boldsymbol{u}} + \tilde{\boldsymbol{u}}$, equation (4.7) can equivalently be written as

$$\frac{\partial \eta}{\partial t} + \boldsymbol{\nabla}_h \cdot (H \bar{\boldsymbol{u}}) = 0, \tag{4.16}$$

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

Depth averaging (4.1), the momentum equation becomes (see e.g. (White et al., 2008b))

$$\begin{aligned} &\frac{\partial \bar{\boldsymbol{u}}}{\partial t} + \bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla}_h \bar{\boldsymbol{u}} + f \boldsymbol{e}_z \wedge \bar{\boldsymbol{u}} \\ &+ g \boldsymbol{\nabla}_h \eta + g \frac{1}{H} \int_{-h}^{\eta} \boldsymbol{\nabla}_h r dz = \bar{\boldsymbol{A}}_H + \bar{\boldsymbol{D}}_H + \frac{\boldsymbol{\tau}_b + \boldsymbol{\tau}_s}{\rho_0 H}, \end{aligned} \tag{4.17}$$

with

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

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

$$\bar{\boldsymbol{D}}_H = \frac{1}{H}\boldsymbol{\nabla}_h \cdot (H\nu_h \boldsymbol{\nabla}_h \bar{\boldsymbol{u}}). \tag{4.18}$$

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 (4.17).

4.3 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. (2008a), we define a mapping from the static domain Ω_0 to the time dependent domain Ω:

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

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

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

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

The *conservative* ALE formulation of the tracer equation (4.8) reads (Formaggia and Nobile, 2004)

$$\frac{1}{J}\left.\frac{\partial (JT)}{\partial t}\right|_{\boldsymbol{\xi}} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u}T) + \frac{\partial \left((w - w_m)T\right)}{\partial z} = \boldsymbol{D}_H, \tag{4.19}$$

where $\boldsymbol{D}_H$ denotes the right hand side of (4.8).

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 $Jd\boldsymbol{\xi} = d\boldsymbol{x}$, the weak formulation of the time derivative can be expressed as

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

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

$$\frac{\partial T}{\partial t}\Big|_{\boldsymbol{\xi}} + T\frac{\partial w_m}{\partial z} + \boldsymbol{\nabla}_h \cdot (\boldsymbol{u}T) + \frac{\partial\left((w - w_m)T\right)}{\partial z} = \boldsymbol{D}_H, \tag{4.20}$$

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.

4.3.1 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

$$z = z_0 + \eta\frac{z_0 + h}{h} \tag{4.21}$$

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

$$w_m\big|_{\Gamma_s} = w - \boldsymbol{\nabla}_h\eta \cdot \boldsymbol{u}. \tag{4.22}$$

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

$$w_m = w_m\big|_{\Gamma_s}\frac{z_0 + h}{h}, \quad \boldsymbol{\xi} \in \Omega_0. \tag{4.23}$$

Expressing w_m in the time dependent coordinates yields

$$w_m = w_m\big|_{\Gamma_s}\frac{z + h}{\eta + h}, \quad \boldsymbol{x} \in \Omega. \tag{4.24}$$

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

$$\frac{\partial w_m}{\partial z} = w_m\big|_{\Gamma_s}\frac{1}{\eta + h}, \quad \boldsymbol{x} \in \Omega. \tag{4.25}$$

4.4 Finite element discretisation

4.4.1 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 $\varphi_i : \mathbb{R}^2 \to \mathbb{R}$, $i = 1, \ldots, N_T$, 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-coordinates[2].

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 $\psi_i : \mathbb{R}^3 \to \mathbb{R}$. 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 $\boldsymbol{x}_i$, $i = 1, \ldots, N_P$: $\psi_i(\boldsymbol{x}_i) = 1$, $\psi_i(\boldsymbol{x}_j) = 0,\ \forall\, i \neq j$.

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.

4.4.2 Interfaces

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

$$\mathcal{I}(K) := \{K \cap K' | K' \in \mathcal{P}\}. \tag{4.26}$$

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

$$\mathcal{I}_{lat}(K) := \{K \cap K' | K' \in \mathcal{P}, \boldsymbol{n} \cdot \boldsymbol{e}_z = 0\}, \tag{4.27}$$

$$\mathcal{I}_h(K) := \{K \cap K' | K' \in \mathcal{P}, \boldsymbol{n} \cdot \boldsymbol{e}_z \neq 0\}. \tag{4.28}$$

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 4.2). 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,

$$\mathcal{J}(Q) := \{Q \cap Q' | Q' \in \mathcal{T}\}. \tag{4.29}$$

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

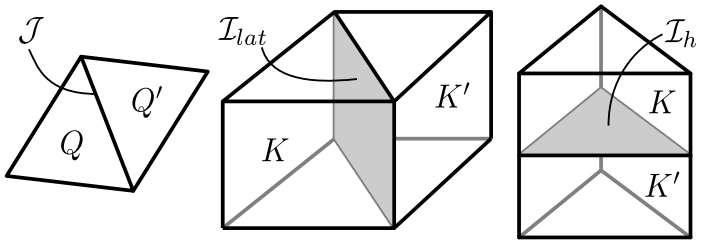


Figure 4.2: Interfaces for triangular surface mesh (left) and prismatic 3D mesh (right).

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

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

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):

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

In practice, the integrals are evaluated with numerical quadrature rules. In 2D, a Hammer quadrature (Solin et al., 2003) 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.

4.4.3 Depth averaged equations

The weak (or variational) formulation for the depth averaged equations is obtained by multiplying (4.16) by a test function $\varphi \in \mathbb{W}$ and (4.17) 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 $Q \in \mathcal{T}$.

$$\left\langle\!\!\left\langle \frac{\partial \eta}{\partial t}\varphi \right\rangle\!\!\right\rangle_Q + \underbrace{\left\langle\!\!\left\langle \boldsymbol{\nabla}_h \cdot ((\eta + h)\bar{\boldsymbol{u}})\,\varphi \right\rangle\!\!\right\rangle_Q}_{\mathcal{E}_u} = 0, \tag{4.30}$$

$$\begin{aligned}
&\left\langle\!\!\left\langle \frac{\partial \bar{\boldsymbol{u}}}{\partial t} \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q + \underbrace{\left\langle\!\!\left\langle \bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla}_h \bar{\boldsymbol{u}} \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q}_{\bar{\mathcal{U}}_u} + \left\langle\!\!\left\langle f \boldsymbol{e}_z \wedge \bar{\boldsymbol{u}} \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q \\
&\quad + \underbrace{\left\langle\!\!\left\langle g \boldsymbol{\nabla}_h \eta \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q}_{\bar{\mathcal{U}}_\eta} + \left\langle\!\!\left\langle g \frac{1}{H} \int_{-h}^{\eta} \boldsymbol{\nabla}_h r \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q \\
&\quad = \left\langle\!\!\left\langle \bar{\boldsymbol{A}}_H \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q + \underbrace{\left\langle\!\!\left\langle \frac{1}{H} \boldsymbol{\nabla}_h \cdot (H \nu_h \boldsymbol{\nabla}_h \bar{\boldsymbol{u}}) \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q}_{\bar{\mathcal{U}}_\nu^h} \\
&\quad + \left\langle\!\!\left\langle \frac{\boldsymbol{\tau}_b}{\rho_0 H} \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q + \left\langle\!\!\left\langle \frac{\boldsymbol{\tau}_s}{\rho_0 H} \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q .
\end{aligned} \tag{4.31}$$

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 $H\bar{\boldsymbol{u}}$:

$$\mathcal{E}_u = -\left\langle\!\!\left\langle (\eta + h)\bar{\boldsymbol{u}} \cdot \boldsymbol{\nabla}_h \varphi \right\rangle\!\!\right\rangle_Q + \sum_{\mathcal{J}(Q)} \left\langle\!\!\left\langle\!\!\left\langle (\eta^* + h)(\bar{\boldsymbol{u}}^* \cdot \boldsymbol{n}_h)\varphi \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}} \tag{4.32}$$

Horizontal advection of momentum:

$$\bar{\mathcal{U}}_u = -\left\langle\!\!\left\langle \boldsymbol{\nabla}_h \cdot (\bar{\boldsymbol{u}}\boldsymbol{\varphi}) \cdot \bar{\boldsymbol{u}} \right\rangle\!\!\right\rangle_Q + \sum_{\mathcal{J}(Q)} \left\langle\!\!\left\langle\!\!\left\langle (\bar{\boldsymbol{u}}^* \cdot \boldsymbol{n}_h)\bar{\boldsymbol{u}}^* \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}} \tag{4.33}$$

External pressure gradient:

$$\bar{\mathcal{U}}_\eta = -\left\langle\!\!\left\langle g\eta \boldsymbol{\nabla}_h \cdot \boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q + \sum_{\mathcal{J}(Q)} \left\langle\!\!\left\langle\!\!\left\langle g\eta^* \boldsymbol{\varphi} \cdot \boldsymbol{n}_h \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}} \tag{4.34}$$

Diffusion of momentum[3]:

$$\begin{aligned}\bar{\mathcal{U}}_\nu^h = -\left\langle\!\!\left\langle \nu_h(\boldsymbol{\nabla}_h\bar{\boldsymbol{u}}) : (\boldsymbol{\nabla}_h\boldsymbol{\varphi})^T \right\rangle\!\!\right\rangle_Q + \left\langle\!\!\left\langle \nu_h \frac{1}{H}(\boldsymbol{\nabla}_h H)\cdot(\boldsymbol{\nabla}_h\bar{\boldsymbol{u}})\cdot\boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q \\ + \sum_{\mathcal{J}(Q)} \left\langle\!\!\left\langle\!\!\left\langle (\nu_h\boldsymbol{\nabla}_h\bar{\boldsymbol{u}})^*\cdot\boldsymbol{n}_h\cdot\boldsymbol{\varphi} \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}}\end{aligned} \tag{4.35}$$

In the interface terms of (4.32)-(4.35), 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 $\bar{\boldsymbol{u}}^*$ are related to the propagation of surface gravity waves and are solved with an approximate Riemann solver described below. The flux $(\nu_h\boldsymbol{\nabla}_h\bar{\boldsymbol{u}})^*$, on the other hand, is related to the horizontal diffusion operator, described in Section 4.4.3.

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:

$$\begin{aligned}\frac{\partial\eta}{\partial t} + \boldsymbol{\nabla}_h(h\bar{\boldsymbol{u}}) = 0, \\ \frac{\partial\bar{\boldsymbol{u}}}{\partial t} + g\boldsymbol{\nabla}_h\eta = 0\end{aligned}$$

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

$$\eta^* = \{\eta\} + \frac{h}{c}[[\bar{\boldsymbol{u}}]], \tag{4.36}$$

$$\bar{\boldsymbol{u}}^* = \{\bar{\boldsymbol{u}}\} + \frac{c}{h}[[\eta]]. \tag{4.37}$$

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. (2010b) is used for the 2D mode, while the above linear solution is used in the 3D mode.

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

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

the interface flux is replaced by a mean flux $\{\nu_h \boldsymbol{\nabla}_h \bar{\boldsymbol{u}}\}$ and an additional penalty term $\sigma\nu_h[[\bar{\boldsymbol{u}}]]$ is introduced, with the penalty factor defined as (Shahbazi, 2005)

$$\sigma_d = \frac{(o+1)(o+d)}{d}\frac{N_0}{2L_{min}}, \tag{4.38}$$

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 (4.35) becomes

$$\begin{aligned}\bar{\mathcal{U}}_\nu^h = &-\left\langle\!\!\left\langle \nu_h(\boldsymbol{\nabla}_h\bar{\boldsymbol{u}}) : (\boldsymbol{\nabla}_h\boldsymbol{\varphi})^T \right\rangle\!\!\right\rangle_Q + \left\langle\!\!\left\langle \nu_h \frac{1}{H}(\boldsymbol{\nabla}_h H)\cdot(\boldsymbol{\nabla}_h\bar{\boldsymbol{u}})\cdot\boldsymbol{\varphi} \right\rangle\!\!\right\rangle_Q \\ &+ \sum_{\mathcal{J}(Q)} \left\langle\!\!\left\langle\!\!\left\langle \{\nu_h\boldsymbol{\nabla}_h\bar{\boldsymbol{u}}\}\cdot\boldsymbol{n}_h\cdot\boldsymbol{\varphi} \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}} + \sum_{\mathcal{J}(Q)} \left\langle\!\!\left\langle\!\!\left\langle \sigma_2\lceil\nu_h\rceil[[\bar{\boldsymbol{u}}]] \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}}.\end{aligned} \tag{4.39}$$

4.4.4 Momentum equation

The weak formulation of the horizontal momentum equation is obtained by multiplying (4.1) 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 (4.19) for brevity, as the non-conservative formulation (4.20) leads to very similar equations. Element-wise weak formulation then reads:

$$\begin{aligned}&\frac{d}{dt}\left\langle \boldsymbol{u}\cdot\boldsymbol{\psi}\right\rangle_K + \underbrace{\left\langle \boldsymbol{\nabla}_h\cdot(\boldsymbol{u}\boldsymbol{u})\cdot\boldsymbol{\psi}\right\rangle_K}_{\mathcal{U}_u} + \underbrace{\left\langle \frac{\partial\left((w-w_m)\boldsymbol{u}\right)}{\partial z}\cdot\boldsymbol{\psi}\right\rangle_K}_{\mathcal{U}_w} + \left\langle f\boldsymbol{e}_z\wedge\boldsymbol{u}\cdot\boldsymbol{\psi}\right\rangle_K + \\ &\underbrace{\left\langle g\boldsymbol{\nabla}_h\eta\cdot\boldsymbol{\psi}\right\rangle_K}_{\mathcal{U}_\eta} + \left\langle g\boldsymbol{\nabla}_h r\cdot\boldsymbol{\psi}\right\rangle_K = \underbrace{\left\langle \boldsymbol{\nabla}_h\cdot(\nu_h\boldsymbol{\nabla}_h\boldsymbol{u})\cdot\boldsymbol{\psi}\right\rangle_K}_{\mathcal{U}_\nu^h} + \underbrace{\left\langle \frac{\partial}{\partial z}\left(\nu\frac{\partial\boldsymbol{u}}{\partial z}\right)\cdot\boldsymbol{\psi}\right\rangle_K}_{\mathcal{U}_\nu^v}.\end{aligned} \tag{4.40}$$

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

$$\begin{aligned}\mathcal{U}_u = &-\left\langle \boldsymbol{\nabla}_h\boldsymbol{\psi} : \boldsymbol{u}\boldsymbol{u}\right\rangle_K + \sum_{\mathcal{I}_{lat}(K)} \left\langle\!\!\left\langle \boldsymbol{\psi}\cdot\boldsymbol{u}^*\boldsymbol{u}^*\cdot\boldsymbol{n}_h \right\rangle\!\!\right\rangle_{\mathcal{I}_{lat}} \\ &+ \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \boldsymbol{\psi}\cdot\boldsymbol{u}^{uw}(\boldsymbol{u}^d\cdot\boldsymbol{n}_h) \right\rangle\!\!\right\rangle_{\mathcal{I}_h},\end{aligned} \tag{4.41}$$

vertical advection

$$\mathcal{U}_w = -\left\langle (w-w_m)\boldsymbol{u}\cdot\frac{\partial\boldsymbol{\psi}}{\partial z}\right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle (w^d - w_m)\boldsymbol{u}^{uw}\cdot\boldsymbol{\psi}n_z \right\rangle\!\!\right\rangle_{\mathcal{I}_h}, \tag{4.42}$$

and external pressure gradient

$$\mathcal{U}_\eta = -\left\langle g\eta \boldsymbol{\nabla}_h \cdot \boldsymbol{\psi} \right\rangle_K + \sum_{\mathcal{I}_{lat}(K)} \left\langle\!\!\left\langle g\eta^* \boldsymbol{\psi} \cdot \boldsymbol{n}_h \right\rangle\!\!\right\rangle_{\mathcal{I}_{lat}} + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle g\eta \boldsymbol{\psi} \cdot \boldsymbol{n}_h \right\rangle\!\!\right\rangle_{\mathcal{I}_h}. \quad (4.43)$$

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

The horizontal diffusion operator becomes

$$\mathcal{U}_\nu^h = -\left\langle \nu_h (\boldsymbol{\nabla}_h \boldsymbol{\psi}) : (\boldsymbol{\nabla}_h \boldsymbol{u})^T \right\rangle_K + \sum_{\mathcal{I}} \left\langle\!\!\left\langle \boldsymbol{\psi} \cdot \{\nu_h \boldsymbol{\nabla}_h \boldsymbol{u}\} \cdot \boldsymbol{n}_h \right\rangle\!\!\right\rangle_{\mathcal{I}}, \quad (4.44)$$

and the vertical diffusion operator

$$\mathcal{U}_\nu^v = -\left\langle \nu \frac{\partial \boldsymbol{\psi}}{\partial z} \cdot \frac{\partial \boldsymbol{u}}{\partial z} \right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \boldsymbol{\psi} \cdot \left\{ \nu \frac{\partial \boldsymbol{u}}{\partial z} \right\} n_z \right\rangle\!\!\right\rangle_{\mathcal{I}_h}. \quad (4.45)$$

Riemann solver for the 3D mode

Similarly to the 2D mode, in (4.41) and (4.43) the Riemann values η^* and $\boldsymbol{u}^*$ are needed. In this work η^* is computed with the linear Riemann solver (4.36), while $\boldsymbol{u}^*$ is computed as

$$\boldsymbol{u}^* = \{\boldsymbol{u}\} + \frac{c}{h}[[\eta]]. \quad (4.46)$$

The above formulation is consistent with the linear Riemann solver, because the depth average of (4.46) reduces to (4.37). Here we are using the linear solver as it is not possible to derive a similar formula for $\boldsymbol{u}^*$ 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.

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),

$$\begin{aligned} &\sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle \sigma_3 (\boldsymbol{n} \cdot \mathbf{D}_\nu \cdot \boldsymbol{n}) \boldsymbol{\psi} \cdot [[\boldsymbol{u}]] \right\rangle\!\!\right\rangle_{\mathcal{I}} = \\ &\sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle \sigma_3 \nu_h (n_x^2 + n_y^2) \boldsymbol{\psi} \cdot [[\boldsymbol{u}]] \right\rangle\!\!\right\rangle_{\mathcal{I}} + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \sigma_3 \nu n_z^2 \boldsymbol{\psi} \cdot [[\boldsymbol{u}]] \right\rangle\!\!\right\rangle_{\mathcal{I}_h}, \end{aligned} \quad (4.47)$$

where $\mathbf{D}_\nu = \mathrm{diag}(\nu_h, \nu_h, \nu)$ is the diffusivity tensor and the penalty factor σ_3 is computed with (4.38). Here $L_{min} = \min(|K|, |K'|)/|I|$, with the element volume $|K|$ and interface area $|I|$. Note that (4.47) 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

$$\begin{aligned}\mathcal{U}_\nu^h = -\Big\langle \nu_h(\boldsymbol{\nabla}_h\boldsymbol{\psi}) : (\boldsymbol{\nabla}_h\boldsymbol{u})^T \Big\rangle_K + \sum_{\mathcal{I}(K)} \Big\langle\!\Big\langle \boldsymbol{\psi}\cdot\{\nu_h\boldsymbol{\nabla}_h\boldsymbol{u}\}\cdot\boldsymbol{n}_h \Big\rangle\!\Big\rangle_{\mathcal{I}} \\ + \sum_{\mathcal{I}(K)} \Big\langle\!\Big\langle \sigma_3\nu_h(n_x^2+n_y^2)\boldsymbol{\psi}\cdot[[\boldsymbol{u}]] \Big\rangle\!\Big\rangle_{\mathcal{I}},\end{aligned} \tag{4.48}$$

$$\begin{aligned}\mathcal{U}_\nu^v = -\Big\langle \nu\frac{\partial\boldsymbol{\psi}}{\partial z}\cdot\frac{\partial\boldsymbol{u}}{\partial z} \Big\rangle_K + \sum_{\mathcal{I}_h(K)} \Big\langle\!\Big\langle \boldsymbol{\psi}\cdot\left\{\nu\frac{\partial\boldsymbol{u}}{\partial z}\right\} n_z \Big\rangle\!\Big\rangle_{\mathcal{I}_h} \\ + \sum_{\mathcal{I}_h(K)} \Big\langle\!\Big\langle \sigma_3\nu n_z^2\boldsymbol{\psi}\cdot[[\boldsymbol{u}]] \Big\rangle\!\Big\rangle_{\mathcal{I}_h}.\end{aligned} \tag{4.49}$$

4.4.5 Tracer Equations

The weak formulations of the temperature and salinity equations are obtained by multiplying (4.8) and (4.9) by a test function $\psi \in \mathbb{V}$, integrating over Ω and adopting the ALE form. As (4.8) and (4.9) are equivalent only the T equation is developed for brevity. Element-wise weak formulation reads:

$$\begin{aligned}\frac{d}{dt}\Big\langle T\psi \Big\rangle_K + \underbrace{\Big\langle \boldsymbol{\nabla}_h\cdot(\boldsymbol{u}T)\psi \Big\rangle_K}_{\mathcal{C}_u} + \underbrace{\Big\langle \frac{\partial\left((w-w_m)T\right)}{\partial z}\psi \Big\rangle_K}_{\mathcal{C}_w} = \\ \underbrace{\Big\langle \boldsymbol{\nabla}_h\cdot(\mu_h\boldsymbol{\nabla}_h T)\,\psi \Big\rangle_K}_{\mathcal{C}_\mu^h} + \underbrace{\Big\langle \frac{\partial}{\partial z}\left(\mu\frac{\partial T}{\partial z}\right)\psi \Big\rangle_K}_{\mathcal{C}_\mu^v}.\end{aligned} \tag{4.50}$$

After integration by parts, the advection terms become

$$\begin{aligned}\mathcal{C}_u = -\Big\langle T\boldsymbol{u}\cdot\boldsymbol{\nabla}_h\psi \Big\rangle_K + \sum_{\mathcal{I}_{lat}(K)} \Big\langle\!\Big\langle T(\boldsymbol{u}^*\cdot\boldsymbol{n}_h)\psi \Big\rangle\!\Big\rangle_{\mathcal{I}_{lat}} \\ + \sum_{\mathcal{I}_h(K)} \Big\langle\!\Big\langle T^{uw}\boldsymbol{u}^d\cdot\boldsymbol{n}_h\psi \Big\rangle\!\Big\rangle_{\mathcal{I}_h},\end{aligned} \tag{4.51}$$

$$\mathcal{C}_w = -\Big\langle (w-w_m)T\frac{\partial\psi}{\partial z} \Big\rangle_K + \sum_{\mathcal{I}_h(K)} \Big\langle\!\Big\langle (w^d-w_m)T^{uw}\psi n_z \Big\rangle\!\Big\rangle_{\mathcal{I}_h}, \tag{4.52}$$

where $\boldsymbol{u}^d$, w^d are the velocity in the lower element and T^{uw} stands for upwind value. For consistency the same Riemann value $\boldsymbol{u}^*$ (4.46) must be used as in the momentum equation. The diffusion terms are treated similarly as before

$$\mathcal{C}_\mu^h = -\Big\langle \mu_h(\boldsymbol{\nabla}_h\psi)\cdot(\boldsymbol{\nabla}_h T) \Big\rangle_K + \sum_{\mathcal{I}(K)} \Big\langle\!\Big\langle \{\mu_h\boldsymbol{\nabla}_h T\}\cdot\boldsymbol{n}_h\psi \Big\rangle\!\Big\rangle_{\mathcal{I}}, \tag{4.53}$$

$$\mathcal{C}_\mu^v = -\Big\langle \mu\frac{\partial T}{\partial z}\frac{\partial\psi}{\partial z} \Big\rangle_K + \sum_{\mathcal{I}_h(K)} \Big\langle\!\Big\langle \left\{\mu\frac{\partial T}{\partial z}\right\} n_z\psi \Big\rangle\!\Big\rangle_{\mathcal{I}_h}. \tag{4.54}$$

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

$$\begin{aligned}\sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle \sigma_3(\boldsymbol{n}\cdot\mathbf{D}_\mu\cdot\boldsymbol{n})\psi[[T]] \right\rangle\!\!\right\rangle_{\mathcal{I}} = \sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle \sigma_3\mu_h(n_x^2+n_y^2)\psi[[T]] \right\rangle\!\!\right\rangle_{\mathcal{I}} \\ + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \sigma_3\mu n_z^2\psi[[T]] \right\rangle\!\!\right\rangle_{\mathcal{I}_h}.\end{aligned} \tag{4.55}$$

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

$$\begin{aligned}\mathcal{C}_\mu^h = &-\left\langle \mu_h(\boldsymbol{\nabla}_h\psi)\cdot(\boldsymbol{\nabla}_h T)\right\rangle_K + \sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle \{\mu_h\boldsymbol{\nabla}_h T\}\cdot\boldsymbol{n}_h\psi \right\rangle\!\!\right\rangle_{\mathcal{I}} \\ &+ \sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle \sigma_3\mu_h(n_x^2+n_y^2)\psi[[T]] \right\rangle\!\!\right\rangle_{\mathcal{I}},\end{aligned} \tag{4.56}$$

$$\mathcal{C}_\mu^v = -\left\langle \mu\frac{\partial T}{\partial z}\frac{\partial\psi}{\partial z}\right\rangle_K + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \left\{\mu\frac{\partial T}{\partial z}\right\} n_z\psi \right\rangle\!\!\right\rangle_{\mathcal{I}_h} + \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \sigma_3\mu n_z^2\psi[[T]] \right\rangle\!\!\right\rangle_{\mathcal{I}_h} \tag{4.57}$$

4.4.6 Computing the vertical velocity

Vertical flow velocity is computed diagnostically from the continuity equation (4.6). At the bottom boundary, w is determined by the impermeability boundary condition:

$$w = -\boldsymbol{u}\cdot\boldsymbol{\nabla}_h h, \quad \boldsymbol{x}\in\Gamma_b. \tag{4.58}$$

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

$$\left\langle \frac{\partial w}{\partial z}\psi\right\rangle_K = -\left\langle \boldsymbol{\nabla}_h\cdot\boldsymbol{u}\psi\right\rangle_K \Leftrightarrow \tag{4.59}$$

$$\begin{aligned}\left\langle\!\!\left\langle w^d\psi n_z\right\rangle\!\!\right\rangle_{\mathcal{I}_{top}} - \left\langle w\frac{\partial\psi}{\partial z}\right\rangle_K = &-\left\langle\!\!\left\langle w^d\psi n_z\right\rangle\!\!\right\rangle_{\mathcal{I}_{bot}} - \sum_{\mathcal{I}_{lat}(K)} \left\langle\!\!\left\langle \psi\boldsymbol{u}^*\cdot\boldsymbol{n}_h\right\rangle\!\!\right\rangle_{\mathcal{I}_{lat}} \\ &- \sum_{\mathcal{I}_h(K)} \left\langle\!\!\left\langle \psi\boldsymbol{u}^d\cdot\boldsymbol{n}_h\right\rangle\!\!\right\rangle_{\mathcal{I}_h} + \left\langle \boldsymbol{u}\cdot\boldsymbol{\nabla}_h\psi\right\rangle_K\end{aligned} \tag{4.60}$$

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 (4.60) is assembled in a modified mass matrix[4], while the other terms are assembled on the right hand side.

[4]It 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.

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 (4.60), a value of $\boldsymbol{u}^*$ is required in the lateral interfaces. To ensure discrete consistency with the horizontal momentum equation and the tracer equation it is essential that $\boldsymbol{u}^*$ of (4.46) is used also here.

4.4.7 Computing the baroclinic head

The definition of the baroclinic head (4.4) contains a vertical integral. It is solved in a manner similar to w, except that here the solution is known at the surface:

$$r = 0, \quad \boldsymbol{x} \in \Gamma_s. \tag{4.61}$$

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

$$\left\langle\!\!\left\langle r^u \psi n_z \right\rangle\!\!\right\rangle_{\mathcal{I}_{bot}} - \left\langle r\frac{\partial \psi}{\partial z}\right\rangle_K = -\left\langle\!\!\left\langle r^u \psi n_z \right\rangle\!\!\right\rangle_{\mathcal{I}_{top}} - \frac{1}{\rho_0}\left\langle \rho' \psi \right\rangle_K, \tag{4.62}$$

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.

4.4.8 Computing the internal pressure gradient

The horizontal gradient of r appears in the momentum equation (4.1). As in the case of the external pressure gradient, $g\boldsymbol{\nabla}_h \eta$, obtaining a robust estimate of $g\boldsymbol{\nabla}_h r$ is essential for 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 $g\boldsymbol{\nabla}_h \eta$, 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 4.4.3.

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 $\boldsymbol{\nabla}_h r$ is solved with the following weak form:

$$\left\langle \boldsymbol{\nabla}_h r \cdot \boldsymbol{\psi} \right\rangle_K = \sum_{\mathcal{I}(K)} \left\langle\!\!\left\langle r^* \boldsymbol{\psi} \cdot \boldsymbol{n}_h \right\rangle\!\!\right\rangle_{\mathcal{I}} - \left\langle r \boldsymbol{\nabla}_h \cdot \boldsymbol{\psi} \right\rangle_K \tag{4.63}$$

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, (4.63) still produces a better estimate of $\boldsymbol{\nabla}_h r$ 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 $\boldsymbol{\nabla}_h r$ is known, it is obtained from the definition

$$\overline{\boldsymbol{\nabla}_h r} = \frac{1}{H}\int_{-h}^{\eta} \boldsymbol{\nabla}_h r dz, \tag{4.64}$$

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

4.5 Matrix form

4.5.1 2D equations

A discrete system of the 2D free surface equation is obtained from (4.30) by replacing η, h and $\bar{\boldsymbol{u}}$ by the respective DG-FE approximations $\hat{\eta} = \sum_i \eta_i \varphi_i$, $\hat{h} = \sum_i h_i \varphi_i$ and $\hat{\bar{\boldsymbol{u}}} = \sum_k \bar{\boldsymbol{u}}_k \varphi_k$.

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

$$\begin{aligned}\left\langle\!\!\left\langle \frac{\partial \sum_i \eta_i \varphi_i}{\partial t}\varphi_j \right\rangle\!\!\right\rangle_{\Gamma_{s,0}} = &\sum_Q \left\langle\!\!\left\langle \sum_i (\eta_i + h_i)\varphi_i \sum_k \bar{\boldsymbol{u}}_k \varphi_k \cdot \boldsymbol{\nabla}_h \varphi_j \right\rangle\!\!\right\rangle_Q \\ &- \sum_Q \left\langle\!\!\left\langle\!\!\left\langle \left(\eta^* + \sum_i h_i \varphi_i\right)\bar{\boldsymbol{u}}^* \cdot \boldsymbol{n}_h)\varphi_j \right\rangle\!\!\right\rangle\!\!\right\rangle_{\mathcal{J}(Q)}, \quad \forall j.\end{aligned} \tag{4.65}$$

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

$$\frac{d\boldsymbol{E}}{dt} = (\mathbf{M}_{2\mathrm{D}})^{-1}\boldsymbol{B}_\eta =: \widetilde{\boldsymbol{B}}_\eta, \tag{4.66}$$

where $[\mathbf{M}_{2\mathrm{D}}]_{ij} = \left\langle\!\!\left\langle \varphi_i \varphi_j \right\rangle\!\!\right\rangle_\Omega$ is the 2D mass matrix, $[\boldsymbol{E}]_i = \eta_i$ and $\boldsymbol{B}_\eta$ denotes the right hand side (RHS) of (4.65).

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

In the case of the depth averaged momentum equation (4.31) 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 $[\bar{\boldsymbol{U}}_u]_i = \bar{u}_i$ and $[\bar{\boldsymbol{U}}_v]_i = \bar{v}_i$, and the corresponding RHS by $\boldsymbol{B}_{\bar{u}}$ and

$\boldsymbol{B}_{\bar{v}}$, one obtains

$$\widetilde{\mathrm{M}}_{2\mathrm{D}} = \begin{bmatrix} \mathrm{M}_{2\mathrm{D}} & 0 \\ 0 & \mathrm{M}_{2\mathrm{D}} \end{bmatrix}, \qquad \bar{\boldsymbol{U}} = \begin{bmatrix} \bar{\boldsymbol{U}}_u \\ \bar{\boldsymbol{U}}_v \end{bmatrix}, \qquad \boldsymbol{B}_{\bar{u}\bar{v}} = \begin{bmatrix} \boldsymbol{B}_{\bar{u}} \\ \boldsymbol{B}_{\bar{v}} \end{bmatrix},$$

$$\frac{d\bar{\boldsymbol{U}}}{dt} = \left(\widetilde{\mathrm{M}}_{2\mathrm{D}}\right)^{-1} \boldsymbol{B}_{\bar{u}\bar{v}} =: \widetilde{\boldsymbol{B}}_{\bar{u}\bar{v}}. \tag{4.67}$$

4.5.2 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 $\hat{\boldsymbol{u}} = \sum_i \boldsymbol{u}_i \psi_i$ for the vector fields $\boldsymbol{u}$ and $\boldsymbol{\nabla}_h r$.

Summing over all the prisms, the weak form of the tracer equation (4.50) becomes

$$\sum_i \frac{d}{dt}\left\langle T_i \psi_i \psi_j \right\rangle_\Omega = \sum_K B_u^{K,j}(\hat{T}, \hat{\boldsymbol{u}}, \ldots) + \sum_K C_u^{K,j}(\hat{T}, \ldots), \quad \forall j, \tag{4.68}$$

where $C_T^{K,j}$ contains the terms related to vertical diffusion (4.49), while all the other terms are grouped in $B_T^{K,j}$.

Denoting the nodal values by $[\boldsymbol{T}]_i = T_i$, equation (4.68) can be written in matrix form

$$\frac{d}{dt}\left(\mathrm{M}_{3\mathrm{D}} \boldsymbol{T}\right) = \widetilde{\boldsymbol{B}}_T + \widetilde{\boldsymbol{C}}_T, \tag{4.69}$$

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

The momentum equation (4.40) is treated in a similar fashion. With the nodal values $[\boldsymbol{U}_u]_i = u_i$ and $[\boldsymbol{U}_v]_i = v_i$, the discrete system can be expressed as

$$\widetilde{\mathrm{M}}_{3\mathrm{D}} = \begin{bmatrix} \mathrm{M}_{3\mathrm{D}} & 0 \\ 0 & \mathrm{M}_{3\mathrm{D}} \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_u \\ \boldsymbol{U}_v \end{bmatrix}, \quad \widetilde{\boldsymbol{B}}_{uv} = \begin{bmatrix} \widetilde{\boldsymbol{B}}_u \\ \widetilde{\boldsymbol{B}}_v \end{bmatrix}, \quad \widetilde{\boldsymbol{C}}_{uv} = \begin{bmatrix} \widetilde{\boldsymbol{C}}_u \\ \widetilde{\boldsymbol{C}}_v \end{bmatrix},$$

$$\frac{d}{dt}\left(\widetilde{\mathrm{M}}_{3\mathrm{D}} \boldsymbol{U}\right) = \widetilde{\boldsymbol{B}}_{uv} + \widetilde{\boldsymbol{C}}_{uv}, \tag{4.70}$$

where $\widetilde{\boldsymbol{C}}_u$ and $\widetilde{\boldsymbol{C}}_v$ correspond to the vertical diffusion terms (4.49), while all the remaining terms are encapsulated in $\widetilde{\boldsymbol{B}}_u$ and $\widetilde{\boldsymbol{B}}_v$.

4.6 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.

4.6.1 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

$$\Delta t_c \propto \frac{L_h}{c + U}, \tag{4.71}$$

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

$$\Delta t_\nu \propto \frac{L_z^2}{\nu}. \tag{4.72}$$

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 (4.71). The restriction imposed by (4.72), on the other hand, is circumvented by treating the vertical diffusion semi-implicitly.

4.6.2 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 4.3. 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 4.6.4. Considering only the tracer T, the complete time marching procedure can be summarized as follows

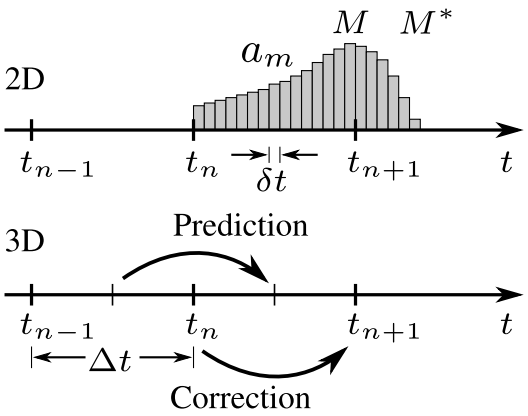


Figure 4.3: Schematic of the time integration method. During each macro time step, the 2D mode is advanced M^* iterations with a time step $\delta t = \Delta t/M$. The 2D variables are averaged with a filter defined by the weights a_m (see Section 4.6.4).

- Prediction stage (in Ω^n)

$$\begin{aligned} \boldsymbol{T}^{n-1/2} &= (\frac{1}{2} - 2\gamma)\boldsymbol{P}^n_{n-1}\boldsymbol{T}^{n-1} + (\frac{1}{2} + 2\gamma)\boldsymbol{T}^n \\ \boldsymbol{T}^{n+1/2,*} &= \boldsymbol{T}^{n-1/2} + \Delta t(1-2\gamma)(\boldsymbol{M}^n_{3\mathrm{D}})^{-1}\boldsymbol{B}^n_T \end{aligned} \tag{4.73}$$

- Advancing the 2D equations

- Correction stage (in Ω^{n+1})

$$\boldsymbol{T}^{n+1,\dagger} = \boldsymbol{P}^{n+1}_n\boldsymbol{T}^n + \Delta t(\boldsymbol{M}^{n+1}_{3\mathrm{D}})^{-1}\boldsymbol{B}^{n+1/2}_T \tag{4.74}$$

$$\boldsymbol{T}^{n+1} = \boldsymbol{T}^{n+1,\dagger} + \Delta t\Theta(\boldsymbol{M}^{n+1}_{3\mathrm{D}})^{-1}\boldsymbol{C}^{n+1}_T + \Delta t(1-\Theta)(\boldsymbol{M}^n_{3\mathrm{D}})^{-1}\boldsymbol{C}^n_T. \tag{4.75}$$

Above, γ and Θ are parameters related to the temporal scheme and the operator $\boldsymbol{P}^a_b = (\boldsymbol{M}^a_{3\mathrm{D}})^{-1}\boldsymbol{M}^b_{3\mathrm{D}}$.

The 2D mode provides η and $\bar{\boldsymbol{u}}$ 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 $\bar{\boldsymbol{u}}$ is used to adjust the 3D horizontal velocity as explained in Section 4.6.5. The 3D mode, on the other hand, affects the 2D mode through bottom friction, internal pressure gradient and advection of $\tilde{\boldsymbol{u}}$ as seen in equation (4.17).

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.

4.6.3 Advancing the 3D equations

The prediction stage consists of updating $\boldsymbol{T}$ from $t_{n-1/2}$ to $t_{n+1/2}$, with RHS $\boldsymbol{B}_T$ evaluated at t_n. The initial condition $\boldsymbol{T}^{n-1/2}$ 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 $\boldsymbol{T}$ to time t_{n+1}, where the RHS is computed with the predicted state ($\boldsymbol{U}^{n+1/2,*}, \boldsymbol{T}^{n+1/2,*}$, 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 (4.75) $\boldsymbol{C}_T^n$ 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 $\boldsymbol{C}_{uv}$ and $\boldsymbol{C}_T$ consist only of terms (4.49) and (4.57), 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.

4.6.4 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 4.3).

Using the notation of Section 4.5.1, the AB3 iteration is then given by

$$\boldsymbol{E}^{m+1} = \boldsymbol{E}^m + \delta t \left[\left(\frac{3}{2} + \beta \right) \widetilde{\boldsymbol{B}}_\eta^m - \left(\frac{1}{2} + 2\beta \right) \widetilde{\boldsymbol{B}}_\eta^{m-1} + \beta \widetilde{\boldsymbol{B}}_\eta^{m-2} \right], \tag{4.76}$$

$$\bar{\boldsymbol{U}}^{m+1} = \bar{\boldsymbol{U}}^m + \delta t \left[\left(\frac{3}{2} + \beta \right) \widetilde{\boldsymbol{B}}_{\bar{u}\bar{v}}^m - \left(\frac{1}{2} + 2\beta \right) \widetilde{\boldsymbol{B}}_{\bar{u}\bar{v}}^{m-1} + \beta \widetilde{\boldsymbol{B}}_{\bar{u}\bar{v}}^{m-2} \right]. \tag{4.77}$$

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

Computing temporal averages

The 3D mode requires η and $\bar{\boldsymbol{u}}$ 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 $\bar{\boldsymbol{u}}$ 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, $\bar{\boldsymbol{u}}$ 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

$$\eta_\otimes^{n+1} = \sum_{m=1}^{M^*} a_m \eta^m, \tag{4.78}$$

$$\bar{\boldsymbol{u}}_\otimes^{n+1} = \sum_{m=1}^{M^*} a_m \bar{\boldsymbol{u}}^m, \tag{4.79}$$

$$(H\bar{\boldsymbol{u}})_\otimes^{n+1} = \sum_{m=1}^{M^*} a_m H^m \bar{\boldsymbol{u}}^m. \tag{4.80}$$

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

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

$$\sum_{m=1}^{M^*} a_m = 1, \qquad \sum_{m=1}^{M^*} \frac{m}{M} a_m = 1. \tag{4.81}$$

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

$$(H\bar{\boldsymbol{u}})_{\otimes}^{n+1/2} = \sum_{m=1}^{M^*} b_m H^{m-1/2} \bar{\boldsymbol{u}}^{m-1/2}, \tag{4.82}$$

subject to similar conditions

$$\sum_{m=1}^{M^*} b_m = 1, \qquad \sum_{m=1}^{M^*} \frac{m}{M} b_m = \frac{1}{2}. \tag{4.83}$$

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

$$\eta_{\otimes}^{n+1} = \eta_{\otimes}^{n} + \Delta t \boldsymbol{\nabla}_h \cdot (H\bar{\boldsymbol{u}})_{\otimes}^{n+1/2}. \tag{4.84}$$

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

$$b_{m'} = \frac{1}{M} \sum_{m=m'}^{M^*} a_m, \tag{4.85}$$

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 B.1.

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.

4.6.5 Adjusting the 3D velocity

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

$$\boldsymbol{u}_{adj}^{n} = \boldsymbol{u}^{n} + \frac{1}{H^n}\left((H\bar{\boldsymbol{u}})_{\otimes}^{n} - \int_{-h}^{\eta} \boldsymbol{u}^{n} dz\right). \tag{4.86}$$

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 $\boldsymbol{u}^{n+1}$ is adjusted with $(H\bar{\boldsymbol{u}})_{\otimes}^{n+1}$. During the prediction stage, at $t_{n+1/2}$, the 2D mode is not yet solved and consequently $(H\bar{\boldsymbol{u}})_{\otimes}^{n+1/2}$ is not yet available. In this case older $(H\bar{\boldsymbol{u}})_{\otimes}^{n}$ is used instead, before executing the prediction stage of the tracers. After the 2D mode is solved, $\boldsymbol{u}^{n+1/2,*}$ is adjusted again with the correct $(H\bar{\boldsymbol{u}})_{\otimes}^{n+1/2}$.

4.6.6 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 B.2.

4.6.7 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 (4.11) and (4.21) – 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 $\eta - \eta_c$.

4.7 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 4.7.3.

4.7.1 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: $\eta_0 = a_0 \exp\left(-(x/\sigma_0)^2\right)$, with $a_0 = 0.1$ m and $\sigma_0 = 2000$ m. Salinity evolution was computed with (4.8) 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 4.4 presents the L_2 error in horizontal velocity versus horizontal mesh resolution, verifying the theoretical second order convergence.

4.7.2 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 $\rho_1' = 2.15$ kgm^{-3}, while on the right $\rho_2' = -2.15$ 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).

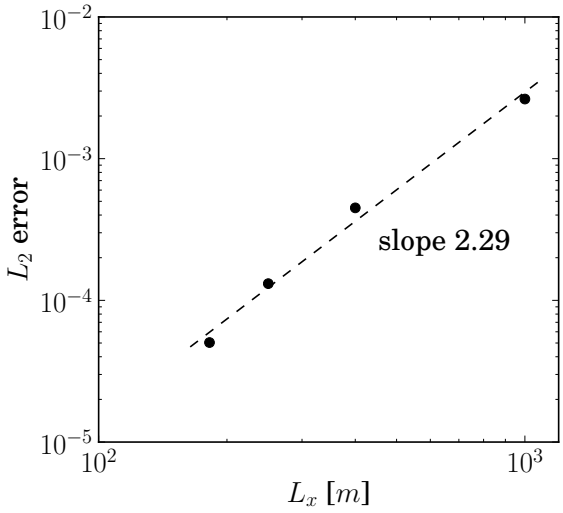


Figure 4.4: Convergence analysis of u in the surface gravity wave test.

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.

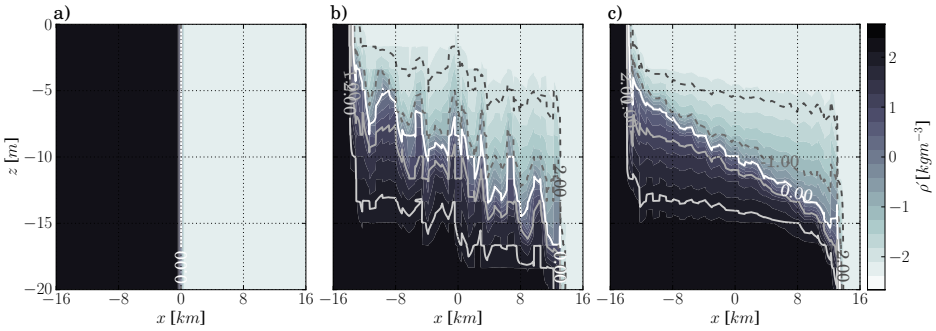


Figure 4.5: Gravitational adjustment without background diffusion. a) Initial density. Density after 31500 s of simulation in the case of the b) simple and c) optimal slope limiter.

Figure 4.5 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 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 4.5, one obtains roughly 0.44 ms^{-1} which is comparable, and – as expected due to numerical dissipation – slightly smaller than the theoretical value $c = 0.46$ ms^{-1}.

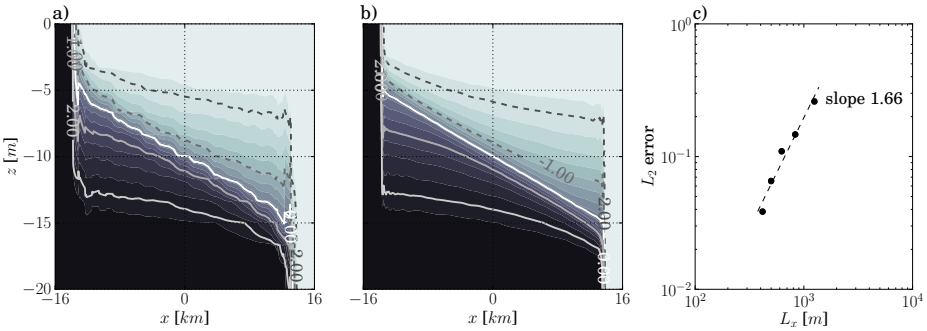


Figure 4.6: Gravitational adjustment with a constant background diffusion $\nu_h = \kappa_h = 5$ $\mathrm{m}^2\mathrm{s}^{-1}$. Density at 31 500 s for a) coarse mesh ($L_h = 833$ m, $N_\sigma = 12$) b) fine mesh ($L_h = 312$ m, $N_\sigma = 32$). c) Convergence plot.

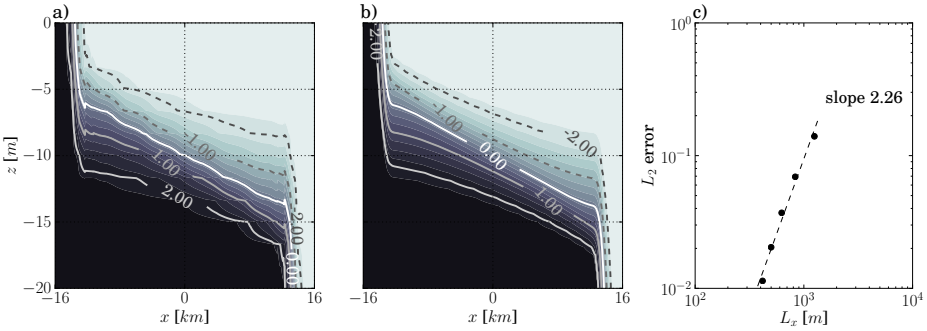


Figure 4.7: Gravitational adjustment with the Smagorinsky diffusion. Density at 31 500 s for a) coarse mesh ($L_h = 833$ m, $N_\sigma = 12$) b) fine mesh ($L_h = 312$ m, $N_\sigma = 32$). c) Convergence plot.

Also here a convergence analysis of the density distribution was performed. The mesh resolution varied between $L_h = 1250$ m ($N_\sigma = 8$) and $L_h = 312$ 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 horizontal diffusion was added.

The results obtained with a constant background diffusivity are presented in Figure 4.6. 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 applied[5], with $C_s = 1.0$. This case is illustrated in Figure 4.7. 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.

4.7.3 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 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.

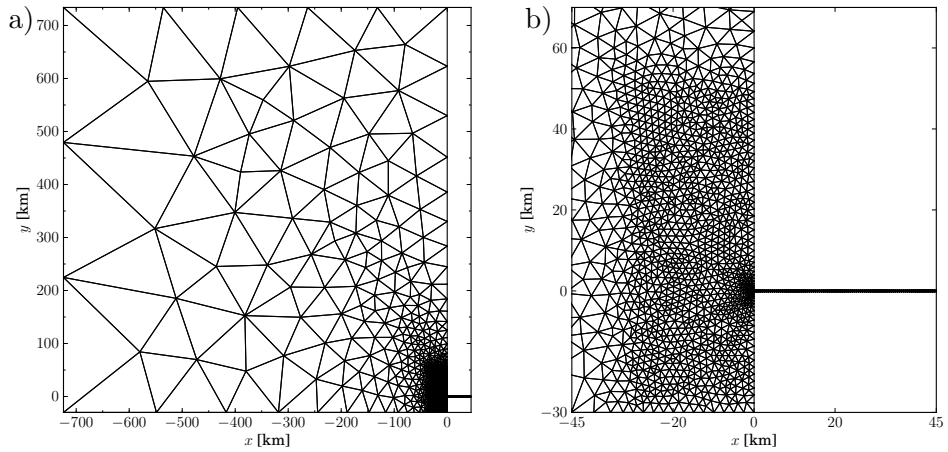


Figure 4.8: Horizontal mesh for the Rhine ROFI simulation. a) The whole domain (3167 triangular elements) b) Enlargement near the area of interest. The resulting 3D mesh, with 20 sigma layers, consists of 63 340 prisms and 380 040 degrees of freedom.

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

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 4.8). 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 m^3s^{-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 $\tau = 44714$ 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).

Tidally averaged salinity is presented in Figure 4.9. 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.

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

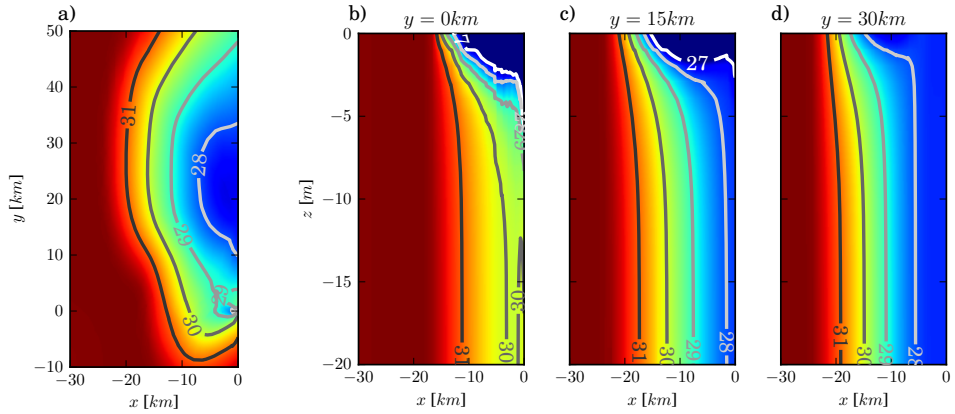


Figure 4.9: Tidally averaged salinity distribution. a) Depth average. Vertical transect at b) river mouth y=0, c) 15 km downstream, d) 30 km downstream.

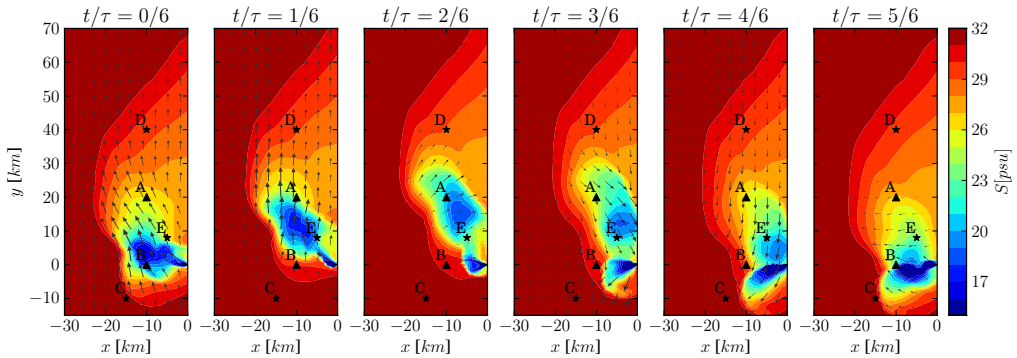


Figure 4.10: Evolution of surface salinity (0.5 m below surface) over a tidal cycle. Arrows illustrate the horizontal flow velocity. Maximal velocity is roughly 1.3 ms^{-1}. Temporal evolution of velocity is examined in stations A and B, and stratification in stations C, D and E.

evolution of the surface salinity and currents is presented in Figure 4.10. 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.

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 4.11 and 4.12 present time series of the horizontal velocity at two locations, marked A and B in Figure 4.10. 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

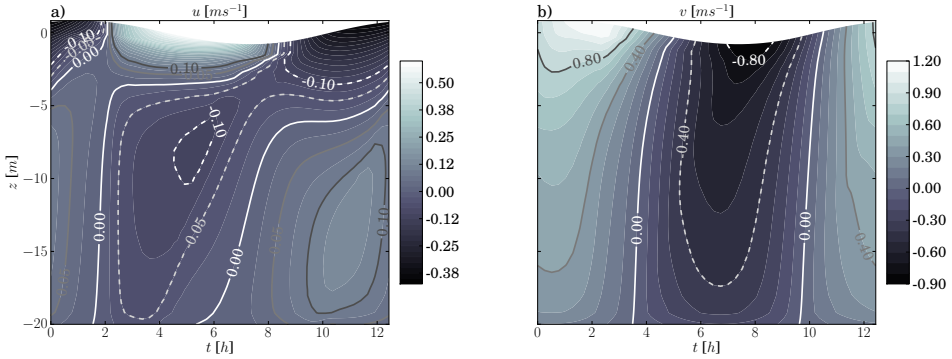


Figure 4.11: Time series of velocities at Station A (10 km offshore and 20 km north of the river mouth). a) Cross-shore velocity. b) Alongshore velocity.

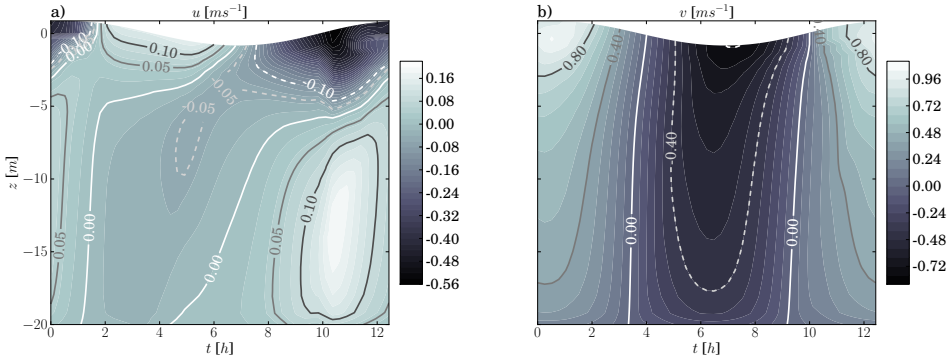


Figure 4.12: Time series of velocities at at Station B (10 km offshore and 0 km north of the river mouth). a) Cross-shore velocity. b) Alongshore velocity.

outflow associated with the freshwater pulse, compensated by a similar saline intrusion in the bottom layer, can be seen in station B (Figure 4.12).

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 4.13 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 4.13.

Figure 4.14 presents the temporal evolution of stratification is 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 4.10). 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

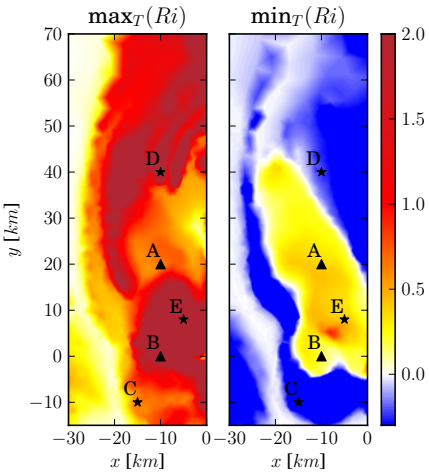


Figure 4.13: Maximal and minimal gradient Richardson number over the tidal cycle in the surface layer (1 m below surface). The surface layer is strongly stratified in areas where $Ri \gtrsim 1$ (left) and unstably stratified where $Ri < 0$ (right).

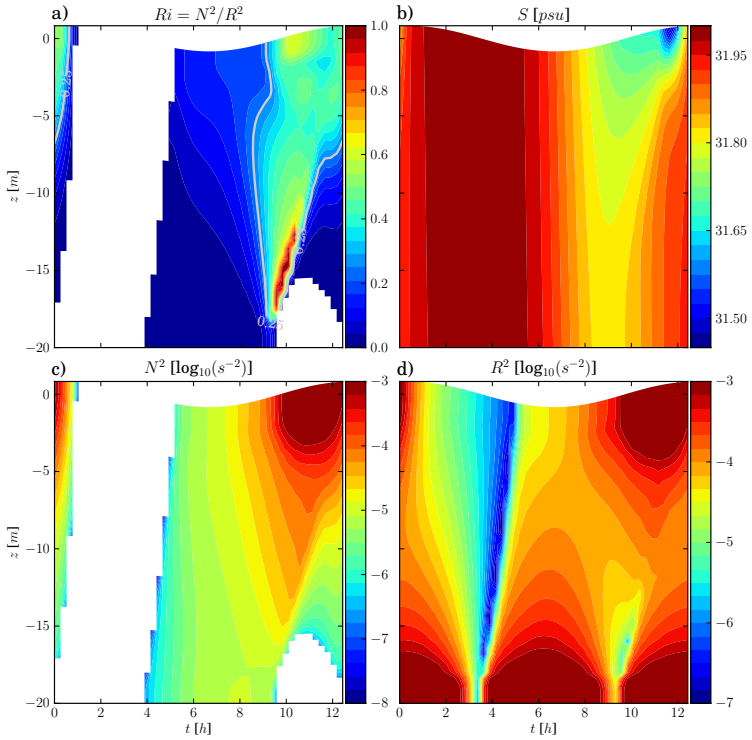


Figure 4.14: Temporal evolution of the stratification is station C (15 km offshore and 10 km south of the river mouth). a) Gradient Richardson number. The critical Richardson number $Ri_c = 0.25$ is indicated by a contour line. b) Salinity. c) Buoyancy frequency squared. d) Shear frequency squared. White patches indicate unstable stratification ($Ri, N^2 < 0$).

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.

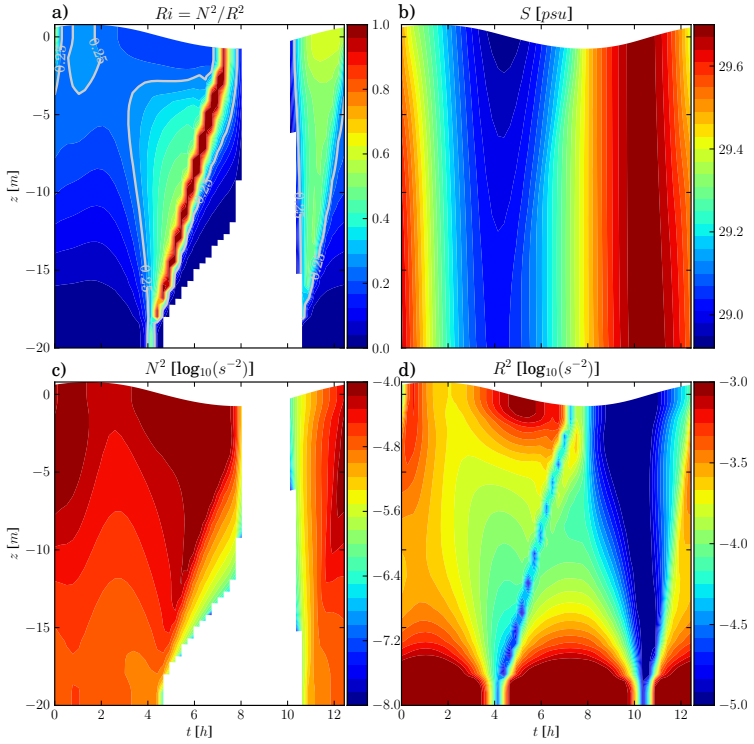


Figure 4.15: Temporal evolution of the stratification is station D (10 km offshore and 40 km north of the river mouth). a) Gradient Richardson number. The critical Richardson number $Ri_c = 0.25$ is indicated by a contour line. b) Salinity. c) Buoyancy frequency squared. d) Shear frequency squared. White patches indicate unstable stratification ($Ri, N^2 < 0$).

The situation at station D is presented in Figure 4.15. 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 4.16) 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 4.10). 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.

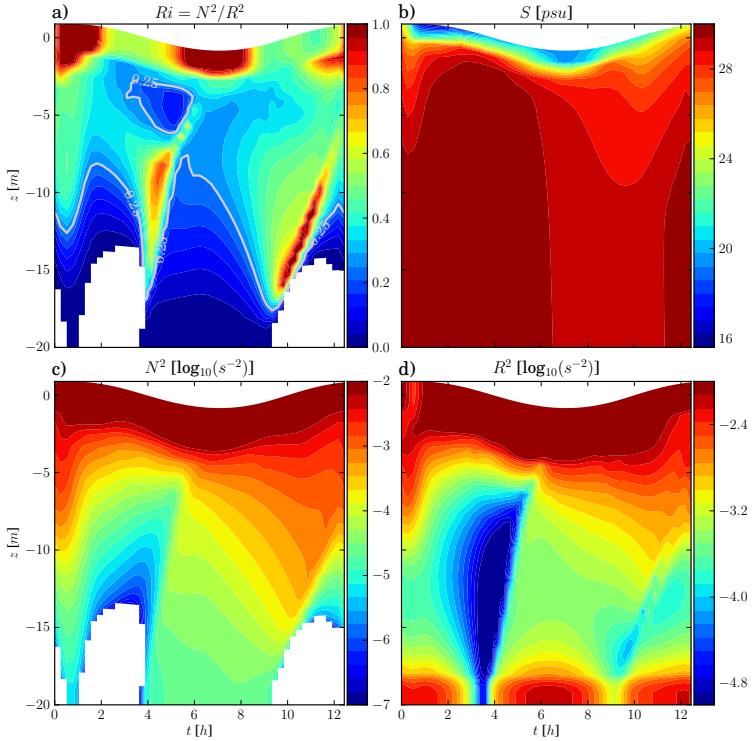


Figure 4.16: Temporal evolution of the stratification is station E (5 km offshore and 8 km north of the river mouth). a) Gradient Richardson number. The critical Richardson number $Ri_c = 0.25$ is indicated by a contour line. b) Salinity. c) Buoyancy frequency squared. d) Shear frequency squared. White patches indicate unstable stratification ($Ri, N^2 < 0$).

4.8 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.

CHAPTER 5

Conclusions

In order to better understand the physical and biochemical processes that take place in coastal waters, and to predict their evolution under anthropogenic pressure, it is necessary to develop regional marine models. In coastal applications it is essential to capture the complex topographical features and the time and length scales relevant to river-estuary-shelf sea systems. Due to their inherent flexibility, novel unstructured grid methods are promising for dealing with such multi-scale problems.

The aim of this thesis was to investigate the feasibility of using the discontinuous Galerkin finite element method in marine modelling, with a focus on estuarine and coastal flows. Due to the characteristics of coastal flows, finding suitable numerical methods is not a straightforward task. In this work two problems have been addressed: the periodic drying and submerging of intertidal areas (i.e. wetting-drying) and the baroclinic modelling of buoyancy driven flows, typical to estuaries and river plumes.

5.1 Wetting-drying

In Chapter 2, a novel implicit wetting-drying method for the depth-averaged shallow water equations is presented. Unlike for most wetting-drying approaches, the proposed method can be formulated directly in the continuous equations. Furthermore, the transition between wet and dry states is smooth, instead of a sharp frontier. This property, although arguably somewhat artificial, is advantageous in terms of numerical implementation. The method involves solving a mildly non-linear free surface equation that ensures strict mass conservation. Consistency with the tracer equation is also guaranteed.

Owing to the smooth wet-dry transition, it is possible to compute the Jacobian of the numerical system. Consequently implicit time integration schemes can be used, which reduces the computational cost significantly. Furthermore, the Jacobian is also required in inverse modelling applications, such as sensitivity analysis and data assimilation, where using conventional wetting-drying methods can be difficult.

The proposed method is well-suited for modelling slow (e.g. tidal) wetting-drying phenomena. However, based on our experience, it does not perform equally well in the case of fast flooding events, such as dam breaks or tsunami propagation. Also, as the Balzano 3 test case illustrates (Section 2.5.1), the method is not appropriate for modelling domains with isolated water reservoirs or lakes. In these cases, more rigorous methods should be used. Possible candidates include the methods by Kesserwani and Liang (2012); Bunya et al. (2009); Gourgue et al. (2009), which all however require explicit time integration.

5.1.1 Perspectives

The proposed wetting-drying method can be implemented in a 3D model as well. In mode-splitting schemes, the 2D mode provides free surface elevation η and depth-integrated velocity $\bar{\boldsymbol{u}}$ to the 3D mode. The elevation is used for updating the 3D geometry and for computing the external pressure gradient. To avoid numerical problems, the pressure gradient should be computed using η in a similar way as in the wetting-drying method. However, the geometry should correspond to $\tilde{\eta}$ (defined in (2.8)) instead. To be strictly volume conservative, the 2D mode should be implicit. However, for certain applications explicit schemes could also be used as the volume discrepancy usually remains small.

Instead of extending the proposed implicit wetting-drying method to 3D, another possibility is to use a semi-implicit free surface formulation similar to the UnTRIM model (Casulli, 2009) and its derivatives. This approach is robust, computationally efficient and unconditionally stable. However, it requires the implementation of a specific non-linear free surface equation that is derived from the discrete equations. Furthermore, the method of Casulli (2009) uses orthogonal unstructured grids, although similar methods have been implemented on general unstructured meshes as well (Zhang and Baptista, 2008). The feasibility of such scheme should be investigated in the DG framework.

In terms of the split-explicit 3D model presented in the latter part of the thesis, perhaps the most natural choice would be to introduce an flux-limiting wetting-drying method in the 2D mode. In this case, as with the UnTRIM-like implicit free surface, the wetting-drying phenomenon could be simulated realistically in all cases, including rapid flooding (e.g. tsunami) events and near isolated water reservoirs. However, some stability issues may arise in the presence of complex or steep bathymetry (Brufau et al., 2002). A major

advantage, in comparison to implicit approaches, is nearly ideal scalability in parallel applications, which is important in large 3D simulations.

5.2 3D baroclinic model

A 3D baroclinic model is presented in Chapters 3 and 4. This work is a continuation to the 3D model developed by White et al. (2008a,b) and later by Comblen et al. (2010a) and Blaise et al. (2010) in the framework of the SLIM project. In this work we have extended the 3D DG-FE model to handle strongly baroclinic flows.

Owing to the DG-FE method and conservative ALE formulation, the model is strictly conservative by construction. Volume and tracer mass conservation is also verified with numerical tests in Section 4.7. Monotonic tracer advection is essential for strongly baroclinic flows, as spurious extrema quickly lead to instabilities. In this work a slope limiter is used to filter the tracer fields.

We have proposed a model based on first order polynomial approximation and split-explicit time integration. The advantage of the chosen approach is that the code is easily parallelisable and slope limiters are readily available. For first order elements, there exist various limiters which can be applied as a post-processing correction at each explicit time step. Here a mass conservative vertex-based limiter is used. While limiters are being developed for high-order discretisation and implicit schemes as well (e.g. Kuzmin, 2006), they tend to be more complicated, especially if extended to 3D fields.

The model is coupled to the General Ocean Turbulence Model (GOTM) library to account for vertical mixing. As GOTM is a 1D vertical finite difference model, coupling between the two model classes must be designed with care in order to avoid creating numerical instabilities in the turbulence closure.

The coupled model is tested with several benchmarks including an idealised estuary simulation following Warner et al. (2005). The model remains stable and is able to produce a realistic estuarine circulation. The flow characteristics compare well with the results of Warner et al. (2005), although some differences are visible.

The most complicated benchmark discussed in this thesis is the Rhine river plume simulation presented in Chapter 4. The plume behaviour is realistically simulated, and the results are well in line with those presented in the literature.

The estuary and river plume simulations reveal a central difficulty in modelling estuarine and coastal flows. As the flow dynamics is highly non-linear due to the interaction of buoyancy, tides, rotational effects and turbulence, numerical models inevitably become sensitive to (lateral) boundary conditions and to the details of the numerical methods. Therefore, although the main characteristics of buoyancy driven flows can be simulated, model-dependent differences are likely to exists. In practical applications these differences may be difficult to assess as it would require high resolution observational data, which is not often available.

5.2.1 Perspectives

Concerning the numerics, more work is needed to find a rigorous stabilisation for the internal pressure gradient, as deriving a conventional Riemann solver for this term is cumbersome. Proper treatment of this term is crucial as it impacts the numerical stability of the model. As it was seen in the case of the gravitational adjustment benchmark (in Section 4.7.2), currently additional diffusivity may be required to suppress oscillations.

Also, the time integration method should be improved to satisfy the tracer consistency criterion. This criterion is often problematic in split-explicit models (Griffies et al., 2001). DG formulation poses additional constraints, as the free surface elevation field is discontinuous, but the adaptive mesh, in general, should remain conforming, i.e. continuous between elements. Currently a constant tracer field is preserved up to a precision of 10^{-6}, implying that some noise may appear in the tracer fields. This level of discrepancy is acceptable in most applications but it is still far from machine precision.

In the future, the feasibility of extending the model to high order should be investigated. Using high-order basis functions in DG offers lower numerical dissipation and a gain in computational efficiency, both of which are attractive features in marine modelling. The main challenge in developing such a model would probably be ensuring monotonicity and numerical stability, as well as dealing with curved boundaries and elements. In principle, extending the presented turbulence closure coupling to high order is feasible, provided that suitable projections between the FE polynomial fields and the FD grid can be constructed.

Furthermore, to increase the computational efficiency, one should also consider circumventing the most restrictive CFL conditions to allow longer time steps. One possibility is to move towards implicit time integration. Based on our experience in shallow coastal applications, solving the 2D mode is relatively inexpensive (of the order 1% of the overall CPU cost). Therefore switching to use an implicit free surface would not bring much gain. In order to significantly reduce the CPU cost, it is likely that the CFL condition related to the 3D advection would need to be relaxed, as the advective velocity tends to be high in coastal areas. For the time being, however, it is unclear whether treating 3D advection implicitly is feasible, as it entails solving a full 3D system in parallel (with both lateral and vertical dependencies). Another possibility would be to use a Lagrangian-Eulerian advection scheme, similarly to the approaches of the UnTRIM (Casulli, 1990; Casulli and Walters, 2000) and SELFE models (Zhang and Baptista, 2008). Such advection schemes, however, should be treated with care, as they have an impact on conservation properties.

As an alternative, multi-rate time integration methods could also be investigated in order to increase the computational efficiency. In multi-rate schemes (e.g. Constantinescu and Sandu, 2007; Schlegel et al., 2009), only the most restrictive elements are marched with a small time step, while the other elements can be iterated with larger time increments. Such schemes are attractive in

unstructured mesh models, as they can exploit the fact that the maximum admissible time step typically varies significantly over the domain (Seny et al., 2012). However, to be applicable in baroclinic 3D models, many improvements are still needed: In addition to ensuring compatibility with the mode splitting scheme and efficient parallelisation, conservation and consistency properties are not easy to meet in multi-rate schemes.

In this work the equations are solved in Cartesian coordinates. In the case of large scale applications, the curvature of Earth has to be taken into account. Possible choices include the traditional latitude-longitude coordinate transform, a stereographic transform or using local element-wise coordinate systems (Comblen et al., 2009; Bernard et al., 2009). Owing to the singularity at the poles, using latitude-longitude coordinates is probably not the best choice. Using local coordinates introduces some computational overhead, which presumably exceeds the cost of the stereographic transformation. Nevertheless, either of these approaches can be implemented in the existing model relatively effortlessly.

Appendices

APPENDIX A

Supplementary material for Chapter 2

A.1 Diagonally Implicit Runge-Kutta schemes

Runge-Kutta schemes are defined by the coefficients $a_{i,j}$, b_j and c_i which are usually gathered in Butcher's tableau:

$$\begin{array}{c|cccc} c_1 & a_{1,1} & 0 & \dots & 0 \\ c_2 & a_{2,1} & a_{2,2} & \dots & 0 \\ c_3 & a_{3,1} & a_{3,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s,1} & a_{s,2} & \dots & a_{s,s} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} .$$

Following (Ascher et al., 1997, section 2.6) and (Jothiprasad et al., 2003, Appendix A), respectively, the implicit Runge-Kutta schemes are defined as follows.

DIRK22

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array}$$

$\gamma = (2 - \sqrt{2})/2$

ESDIRK64

$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$\frac{1}{4}$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
$\frac{17}{20}$	$\frac{15267082809}{155376265600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$

APPENDIX B

Supplementary material for Chapter 4

B.1 Temporal filter coefficients

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

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

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

$$\sum_{m=1}^{M^*} a_m = 1. \tag{B.2}$$

The constant τ_0 is obtained from the second constraint

$$\sum_{m=1}^{M^*} \frac{m}{M} a_m = 1, \tag{B.3}$$

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

$$\tau_0 = \frac{(p+2)(p+q+2)}{(p+1)(p+q+1)}. \tag{B.4}$$

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).

B.2 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 $\boldsymbol{x}_i$ in the 3D mesh, a "neighbourhood" of $\boldsymbol{x}_i$ is defined as a set of elements sharing the node: $\mathcal{P}(\boldsymbol{x}_i) = \{K \in \mathcal{P} | \boldsymbol{x}_i \in K\}$. 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 $\boldsymbol{x}_{\chi(j,K)}$.

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

$$T^{\text{Min}}_{\chi(j,K)} \leq \tilde{T}_j^K \leq T^{\text{Max}}_{\chi(j,K)},$$
$$\|\tilde{T}^K\| = \|T^K\|.$$

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).

B.2.1 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.

B.2.2 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

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

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

References

A. Adcroft and D. Marshall. How slippery are piecewise-constant coastlines in numerical ocean models? *Tellus A*, 50(1):95–108, 1998. Cited on p. 6.

V. Aizinger. A Geometry Independent Slope Limiter for the Discontinuous Galerkin Method. In E. Krause, Y. Shokin, M. Resch, D. Kröner, and N. Shokina, editors, *Computational Science and High Performance Computing IV*, volume 115 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 207–217. Springer Berlin / Heidelberg, 2011. Cited on p. 12, 73, 96, 120.

V. Aizinger and C. Dawson. The local discontinuous Galerkin method for three-dimensional shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(4-6):734–746, 2007. ISSN 0045-7825. doi: 10.1016/j.cma.2006.04.010. Cited on p. 96.

R. Alonso, M. Santillana, and C. Dawson. On the diffusive wave approximation of the shallow water equations. *European Journal of Applied Mathematics*, 19(05):575–606, 2008. Cited on p. 18.

C. Amante and B. W. Eakins. ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis. *NOAA Technical Memorandum NESDIS*, NGDC-24:19, 2009. Cited on p. 39.

D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002. doi: 10.1137/S0036142901384162. Cited on p. 73, 81.

U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25:151–167, 1997. Cited on p. 26, 117.

W. Baeyens, B. van Eck, C. Lambert, R. Wollast, and L. Goeyens. General description of the Scheldt estuary. *Hydrobiologia*, 366(1):1–14, Mar 1997. Cited on p. 8, 38.

A. Balzano. Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. *Coastal Engineering*, 34(1-2):83–107, Jul 1998. ISSN 0378-3839. Cited on p. 30, 31, 32, 34.

A. M. Baptista, Y. Zhang, A. Chawla, M. Zulauf, C. Seaton, E. P. I. Myers, J. Kindle, M. Wilkin, M. Burla, and P. J. Turner. A cross-scale model for 3D baroclinic circulation in estuary–plume–shelf systems: II. Application to the Columbia River. *Continental Shelf Research*, 25(7–8):935–972, 2005. ISSN 0278-4343.
doi: 10.1016/j.csr.2004.12.003. Cited on p. 72.

P. Bates and J. Hervouet. A new method for moving–boundary hydrodynamic problems in shallow water. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 455(1988):3107–3128, 1999. Cited on p. 17, 18, 21.

P. D. Bates and M. S. Horritt. *Computational Fluid Dynamics*, chapter Modelling Wetting and Drying Processes in Hydraulic Models, pages 121–146. John Wiley & Sons, Ltd, 2005. Cited on p. 34.

P.-E. Bernard, J.-F. Remacle, and V. Legat. Boundary discretization for high order discontinuous Galerkin computations of tidal flows around shallow water islands. *International Journal for Numerical Methods in Fluids*, 59 (5):535–557, 2009. Cited on p. 113.

S. Blaise and E. Deleersnijder. Improving the parameterisation of horizontal density gradient in one-dimensional water column models for estuarine circulation. *Ocean Science*, 4(4):239–246, 2008.
doi: 10.5194/os-4-239-2008. Cited on p. 49.

S. Blaise, E. Deleersnijder, L. White, and J.-F. Remacle. Influence of the turbulence closure scheme on the finite-element simulation of the upwelling in the wake of a shallow-water island. *Continental Shelf Research*, 27(18): 2329–2345, 2007. Cited on p. 11, 49.

S. Blaise, R. Comblen, V. Legat, J.-F. Remacle, E. Deleersnijder, and J. Lambrechts. A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. Part I: space discretization. *Ocean Dynamics*, 60: 1371–1393, 2010. ISSN 1616-7341. Cited on p. 8, 11, 12, 52, 111.

R. Bleck and L. Smith. A wind-driven isopycnic coordinate model of the north and equatorial Atlantic Ocean 1. Model development and supporting experiments. *Journal of Geophysical Research*, 95(C3):3273–3285, 1990. Cited on p. 74.

A. F. Blumberg and G. L. Mellor. A description of a three-dimensional coastal ocean model. In N. S. Heaps, editor, *Three Dimensional Coastal Ocean Models*, pages 1–16. American Geophysical Union, 1999. Cited on p. 11, 72, 74.

P. Brufau, M. E. Vázquez-Cendón, and P. García-Navarro. A numerical model for the flooding and drying of irregular domains. *International Journal for*

Numerical Methods in Fluids, 39(3):247–275, 2002. ISSN 1097-0363. Cited on p. 32, 110.

K. Bryan. A numerical method for the study of the circulation of the World Ocean. *Journal of Computational Physics*, 4(3):347–376, 1969. Cited on p. 4.

S. Bunya, E. J. Kubatko, J. J. Westerink, and C. Dawson. A wetting and drying treatment for the Runge-Kutta discontinuous Galerkin solution to the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20):1548–1562, 2009. ISSN 0045-7825.
doi: 10.1016/j.cma.2009.01.008. Cited on p. 17, 22, 110.

H. Burchard. *Applied turbulence modelling in marine waters*, volume 100 of *Lecture Notes in Earth Sciences*. Springer, Berlin, 2002a. ISBN 9783540437956. Cited on p. 48, 50.

H. Burchard. Energy-conserving discretisation of turbulent shear and buoyancy production. *Ocean Modelling*, 4(3-4):347–361, Jun 2002b. ISSN 1463-5003.
doi: 10.1016/S1463-5003(02)00009-4. Cited on p. 57.

H. Burchard and H. Baumert. On the performance of a mixed-layer model based on the k-ε turbulence closure. *Journal of Geophysical Research*, 100: 8523–8540, 1995. Cited on p. 51.

H. Burchard and K. Bolding. Comparative Analysis of Four Second-Moment Turbulence Closure Models for the Oceanic Mixed Layer. *Journal of Physical Oceanography*, 31(8):1943–1968, 2001.
doi: 10.1175/1520-0485(2001)031. Cited on p. 52.

H. Burchard and K. Bolding. GETM - a general estuarine transport model. Scientific documentation. Technical Report EUR 20253 EN, European Commission, 2002. Cited on p. 4, 11, 48, 65, 72.

H. Burchard and E. Deleersnijder. Stability of algebraic non-equilibrium second-order closure models. *Ocean Modelling*, 3(1-2):33–50, 2001. ISSN 1463-5003.
doi: 10.1016/S1463-5003(00)00016-0. Cited on p. 63.

H. Burchard and R. D. Hetland. Quantifying the contributions of tidal straining and gravitational circulation to residual circulation in periodically stratified tidal estuaries. *Journal of Physical Oceanography*, 40:1243–1262, 2010. Cited on p. 51.

H. Burchard and H. Rennau. Comparative quantification of physically and numerically induced mixing in ocean models. *Ocean Modelling*, 20(3):293–311, 2008. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2007.10.003. Cited on p. 67, 72.

H. Burchard, O. Petersen, and T. P. Rippeth. Comparing the performance of the Mellor-Yamada and the $k - \epsilon$ two-equation turbulence models. *J. Geophys. Res.*, 103(C5):10543–10554, 1998. ISSN 0148-0227. doi: 10.1029/98JC00261. Cited on p. 48, 62.

H. Burchard, K. Bolding, and M. R. Villarreal. GOTM, a general ocean turbulence model. Theory, implementation and test cases. Technical Report EUR 18745, European Commission, 1999. Cited on p. 48, 76.

H. Burchard, K. Bolding, and M. Villarreal. Three-dimensional modelling of estuarine turbidity maxima in a tidal estuary. *Ocean Dynamics*, 54(2): 250–265, 2004. Cited on p. 18.

J.-M. Campin, A. Adcroft, C. Hill, and J. Marshall. Conservation of properties in a free-surface model. *Ocean Modelling*, 6(3-4):221–244, 2004. ISSN 1463-5003. Cited on p. 11.

V. M. Canuto, A. Howard, Y. Cheng, and M. S. Dubovikov. Ocean Turbulence. Part I: One-Point Closure Model – Momentum and Heat Vertical Diffusivities. *Journal of Physical Oceanography*, 31(6):1413–1426, 2001. doi: 10.1175/1520-0485(2001)031. Cited on p. 50, 52, 63, 76.

M. Castro, A. Ferreiro, J. García-Rodriguez, J. González-Vida, J. Macías, C. Parés, and M. Elena Vázquez-Cendón. The numerical treatment of wet/dry fronts in shallow flows: application to one-layer and two-layer systems. *Mathematical and Computer Modelling*, 42(3-4):419–439, 2005. ISSN 0895-7177. Cited on p. 18.

V. Casulli. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *Journal of Computational Physics*, 86(1):56–74, 1990. ISSN 0021-9991. doi: 10.1016/0021-9991(90)90091-E. Cited on p. 112.

V. Casulli. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. *International Journal for Numerical Methods in Fluids*, 60 (4):391–408, 2009. ISSN 1097-0363. Cited on p. 17, 34, 110.

V. Casulli and R. T. Cheng. Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, 15(6):629–648, 1992. ISSN 1097-0363. Cited on p. 7, 11, 72.

V. Casulli and R. A. Walters. An unstructured grid, three-dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids*, 32(3):331–348, 2000. ISSN 1097-0363. doi: 10.1002/(SICI)1097-0363(20000215)32:3. Cited on p. 7, 11, 72, 112.

C. Chen, R. C. Beardsley, and G. Cowles. An unstructured grid, finite-volume coastal ocean model: FVCOM User Manual. Technical Report 06-0602, SMAST/UMASSD, 2006. Cited on p. 4, 11, 48, 72.

Y. Cheng, V. M. Canuto, and A. M. Howard. An improved model for the turbulent PBL. *J. Atmos. Sci.*, 59:1550–1565, 2002. Cited on p. 51.

B. Cockburn. Discontinuous Galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 83(11):731–754, 2003. ISSN 1521-4001.
doi: 10.1002/zamm.200310088. Cited on p. 12, 73.

B. Cockburn and C. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V : multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998. Cited on p. 96.

B. Cockburn and C.-W. Shu. Runge-Kutta Discontinuous Galerkin Methods for convection-dominated problems. *Journal of Scientific Computing*, 16: 173–261, 2001. Cited on p. 7.

R. Comblen, S. Legrand, E. Deleersnijder, and V. Legat. A finite element method for solving the shallow water equations on the sphere. *Ocean Modelling*, 28(1-3):12–23, 2009. ISSN 1463-5003. Cited on p. 113.

R. Comblen, S. Blaise, V. Legat, J.-F. Remacle, E. Deleersnijder, and J. Lambrechts. A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. Part II: implicit/explicit time discretization. *Ocean Dynamics*, 60:1395–1414, 2010a. ISSN 1616-7341. Cited on p. 8, 11, 12, 52, 111.

R. Comblen, J. Lambrechts, J.-F. Remacle, and V. Legat. Practical evaluation of five partly discontinuous finite element pairs for the non-conservative shallow water equations. *International Journal for Numerical Methods in Fluids*, 63(6):701–724, 2010b. ISSN 1097-0363. Cited on p. 8, 22, 83.

E. Constantinescu and A. Sandu. Multirate Timestepping Methods for Hyperbolic Conservation Laws. *Journal of Scientific Computing*, 33:239–278, 2007. ISSN 0885-7474. Cited on p. 112.

G. de Boer, J. Pietrzak, and J. Winterwerp. On the vertical structure of the Rhine region of freshwater influence. *Ocean Dynamics*, 56(3):198–216, 2006. ISSN 1616-7341.
doi: 10.1007/s10236-005-0042-1. Cited on p. 101, 107.

G. J. de Boer, J. D. Pietrzak, and J. C. Winterwerp. SST observations of upwelling induced by tidal straining in the Rhine ROFI. *Continental Shelf Research*, 29(1):263–277, 2008.
doi: 10.1016/j.csr.2007.06.011. Cited on p. 101, 103, 105, 107.

A. de Brauwere, F. De Ridder, O. Gourgue, J. Lambrechts, R. Comblen, R. Pintelon, J. Passerat, P. Servais, M. Elskens, W. Baeyens, T. Kärnä, B. de Brye, and E. Deleersnijder. Design of a sampling strategy to optimally calibrate a reactive transport model: Exploring the potential for Escherichia coli in the Scheldt Estuary. *Environmental Modelling & Software*, 24(8):969–981, Aug 2009. ISSN 1364-8152.
doi: 10.1016/j.envsoft.2009.02.004. Cited on p. 8.

A. de Brauwere, B. de Brye, S. Blaise, and E. Deleersnijder. Residence time, exposure time and connectivity in the Scheldt Estuary. *Journal of Marine Systems*, 84(3–4):85–95, 2011a. ISSN 0924-7963.
doi: 10.1016/j.jmarsys.2010.10.001. Cited on p. 8.

A. de Brauwere, B. de Brye, P. Servais, J. Passerat, and E. Deleersnijder. Modelling Escherichia coli concentrations in the tidal Scheldt river and estuary. *Water Research*, 45(9):2724–2738, 2011b. ISSN 0043-1354.
doi: 10.1016/j.watres.2011.02.003. Cited on p. 8.

B. de Brye, A. de Brauwere, O. Gourgue, T. Kärnä, J. Lambrechts, R. Comblen, and E. Deleersnijder. A finite-element, multi-scale model of the Scheldt tributaries, river, estuary and ROFI. *Coastal Engineering*, 57 (9):850–863, 2010. ISSN 0378-3839.
doi: 10.1016/j.coastaleng.2010.04.001. Cited on p. 8, 19, 39, 40, 52, 72.

B. de Brye, A. de Brauwere, O. Gourgue, E. J. Delhez, and E. Deleersnijder. Water renewal timescales in the Scheldt Estuary. *Journal of Marine Systems*, 94(0):74–86, 2012. ISSN 0924-7963.
doi: 10.1016/j.jmarsys.2011.10.013. Cited on p. 8.

W. P. M. de Ruijter, A. W. Visser, and W. G. Bos. The Rhine outflow: A prototypical pulsed discharge plume in a high energy shallow sea. *Journal of Marine Systems*, 12(1–4):263–276, Aug 1997. ISSN 0924-7963.
doi: 10.1016/S0924-7963(96)00102-9. Cited on p. 101.

L. Debreu and E. Blayo. Two-way embedding algorithms: a review. *Ocean Dynamics*, 58:415–428, 2008. Cited on p. 72.

A. Defina. Two-Dimensional Shallow Flow Equations for Partially Dry Areas. *Water Resources Research*, 36(11):3251–3264, 2000. Cited on p. 18.

E. Deleersnijder and P. Lermusiaux. Multi-Scale Modelling: Nested Grid and Unstructured Mesh Approaches. *Ocean Dynamics (special issue)*, 58(5-6): 335–498, 2008. Cited on p. 6.

E. Deleersnijder and P. Luyten. On the practical advantages of the quasi-equilibrium version of the Mellor and Yamada level 2.5 turbulence closure

applied to marine modelling. *Applied Mathematical Modelling*, 18(5):281–287, 1994. ISSN 0307-904X.
doi: 10.1016/0307-904X(94)90336-0. Cited on p. 62, 63.

E. Deleersnijder, E. Hanert, H. Burchard, and H. Dijkstra. On the mathematical stability of stratified flow models with local turbulence closure schemes. *Ocean Dynamics*, 58:237–246, 2008. ISSN 1616-7341. Cited on p. 63.

Delft3D-FLOW User Manual. Deltares, 2012. Cited on p. 4.

J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. *Arbitrary Lagrangian–Eulerian Methods*. John Wiley & Sons, Ltd, 2004. ISBN 9780470091357. Cited on p. 11, 74.

J. K. Dukowicz and R. D. Smith. Implicit free-surface method for the Bryan-Cox-Semtner ocean model. *J. Geophys. Res.*, 99(C4):7991–8014, 1994. ISSN 0148-0227.
doi: 10.1029/93JC03455. Cited on p. 11, 73.

G. Egbert, A. Bennett, and M. Foreman. TOPEX/Poseidon tides estimated using a global inverse model. *Journal of Geophysical Research*, 99(12):24821–24852, 1994. Cited on p. 39.

L. I. Enstad, K. Rygg, P. M. Haugan, and G. Alendal. Dissolution of a CO2 lake, modeled by using an advanced vertical turbulence mixing scheme. *International Journal of Greenhouse Gas Control*, 2(4):511–519, 2008. ISSN 1750-5836.
doi: 10.1016/j.ijggc.2008.04.001. Cited on p. 48.

A. Ern, S. Piperno, and K. Djadel. A well-balanced Runge-Kutta discontinuous Galerkin method for the shallow-water equations with flooding and drying. *International Journal for Numerical Methods in Fluids*, 58(1):1–25, 2008. ISSN 1097-0363. Cited on p. 22, 34.

A. Ern, A. F. Stephansen, and P. Zunino. A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity. *IMA Journal of Numerical Analysis*, 29(2):235–256, 2009.
doi: 10.1093/imanum. Cited on p. 85.

C. Farhat, P. Geuzaine, and C. Grandmont. 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 (2):669–694, 2001. ISSN 0021-9991.
doi: 10.1006/jcph.2001.6932. Cited on p. 78.

E. Fischer, H. Burchard, and R. Hetland. Numerical investigations of the turbulent kinetic energy dissipation rate in the Rhine region of freshwater

influence. *Ocean Dynamics*, 59:629–641, 2009. ISSN 1616-7341. Cited on p. 101, 107.

L. Formaggia and F. Nobile. Stability analysis of second-order time accurate schemes for ALE–FEM. *Computer Methods in Applied Mechanics and Engineering*, 193(39–41):4097–4116, 2004. ISSN 0045-7825. doi: 10.1016/j.cma.2003.09.028. Cited on p. 78.

C. T. Friedrichs. *Baroclinic tides in channelized estuaries*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511676567. Cited on p. 5.

A. J. Gadd. A split explicit integration scheme for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 104(441): 569–582, 1978. ISSN 1477-870X. doi: 10.1002/qj.49710444103. Cited on p. 11.

B. Galperin, L. H. Kantha, S. Hassid, and A. Rosati. A Quasi-equilibrium Turbulent Energy Model for Geophysical Flows. *Journal of the Atmospheric Sciences*, 45(1):55–62, 1988. doi: 10.1175/1520-0469(1988)045. Cited on p. 65.

D. L. George. Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *Journal of Computational Physics*, 227(6):3089–3113, Mar 2008. ISSN 0021-9991. Cited on p. 23, 25.

C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009. ISSN 1097-0207. Cited on p. 6, 39, 101.

O. Gourgue, R. Comblen, J. Lambrechts, T. Kärnä, V. Legat, and E. Deleersnijder. A flux-limiting wetting-drying method for finite-element shallow-water models, with application to the Scheldt Estuary. *Advances in Water Resources*, 32(12):1726–1739, Dec 2009. ISSN 0309-1708. doi: 10.1016/j.advwatres.2009.09.005. Cited on p. 8, 9, 17, 19, 22, 30, 32, 34, 39, 40, 110.

S. M. Griffies. *Fundamentals of ocean climate models*. Princeton University Press, 2004. Cited on p. 73.

S. M. Griffies. Elements of MOM4p1. Technical Report No. 6, NOAA/Geophysical Fluid Dynamics Laboratory, GFDL Ocean Group, Feb. 2010. Cited on p. 48, 72.

S. M. Griffies, R. C. Pacanowski, M. Schmidt, and V. Balaji. Tracer Conservation with an Explicit Free Surface Method for z-Coordinate Ocean Models.

Monthly Weather Review, 129(5):1081–1098, 2001.
doi: 10.1175/1520-0493(2001)129. Cited on p. 11, 12, 112.

D. Haidvogel and A. Beckmann. *Numerical Ocean Circulation Modeling*. Environmental Science and Management. Imperial College Press, 4 edition, 1999. Cited on p. 97.

E. Hanert, E. Deleersnijder, and V. Legat. An adaptive finite element water column model using the Mellor–Yamada level 2.5 turbulence closure scheme. *Ocean Modelling*, 12(1-2):205–223, 2006. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2005.05.003. Cited on p. 49.

E. Hanert, E. Deleersnijder, S. Blaise, and J.-F. Remacle. Capturing the bottom boundary layer in finite element ocean models. *Ocean Modelling*, 17 (2):153–162, 2007. Cited on p. 49, 58.

M. Heniche, Y. Secretan, P. Boudreau, and M. Leclerc. A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries. *Advances in Water Resources*, 23(4):359–372, Jan 2000. ISSN 0309-1708. Cited on p. 17, 18, 21.

R. D. Hetland. Relating River Plume Structure to Vertical Mixing. *Journal of Physical Oceanography*, 35(9):1667–1688, Sep 2005. ISSN 0022-3670.
doi: 10.1175/JPO2774.1. Cited on p. 72.

R. Hofmeister, H. Burchard, and J.-M. Beckers. Non-uniform adaptive vertical grids for 3D numerical ocean models. *Ocean Modelling*, 33(1-2):70–86, 2010. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2009.12.003. Cited on p. 74.

J. Holt and L. Umlauf. Modelling the tidal mixing fronts and seasonal stratification of the Northwest European Continental shelf. *Continental Shelf Research*, 28(7):887–903, 2008. ISSN 0278-4343.
doi: 10.1016/j.csr.2008.01.012. Cited on p. 48.

J. T. C. Ip, D. R. Lynch, and C. T. Friedrichs. Simulation of estuarine flooding and dewatering with application to Great Bay, New Hampshire. *Estuarine, Coastal and Shelf Science*, 47(2):119–141, 1998. Cited on p. 18, 21.

D. R. Jackett, T. J. McDougall, R. Feistel, D. G. Wright, and S. M. Griffies. Algorithms for Density, Potential Temperature, Conservative Temperature, and the Freezing Temperature of Seawater. *Journal of Atmospheric and Oceanic Technology*, 23(12):1709–1728, 2006.
doi: 10.1175/JTECH1946.1. Cited on p. 75.

J. A. Jankowski. *A non-hydrostatic model for free surface flows*. PhD thesis, Institut für Strömungsmechanik und ERiB, Universität Hannover, 1999. Cited on p. 97.

Y. Jiang and O. W. Wai. Drying-wetting approach for 3D finite element sigma coordinate model for estuaries with large tidal flats. *Advances in Water Resources*, 28(8):779–792, 2005. ISSN 0309-1708.
doi: 10.1016/j.advwatres.2005.02.004. Cited on p. 18, 21, 43.

B. Jørgensen and K. Richardson, editors. Coastal and Estuarine Studies. American Geophysical Union, 1996. ISBN 9780875902661. Cited on p. 2.

G. Jothiprasad, D. J. Mavriplis, and D. A. Caughey. Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes. *Journal of Computational Physics*, 191(2):542–566, Nov 2003. ISSN 0021-9991. Cited on p. 26, 117.

E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, Y. Zhu, A. Leetmaa, R. Reynolds, M. Chelliah, W. Ebisuzaki, W. Higgins, J. Janowiak, K. C. Mo, C. Ropelewski, J. Wang, R. Jenne, and D. Joseph. The NCEP/NCAR 40-Year Reanalysis Project. *Bulletin of the American Meteorological Society*, 77(3): 437–471, 1996.
doi: 10.1175/1520-0477(1996)077. Cited on p. 39.

T. Kärnä, V. Legat, and E. Deleersnijder. Baroclinic discontinuous Galerkin finite element model for coastal flows. *Ocean modelling*, submitted. Cited on p. 8.

T. Kärnä, E. Deleersnijder, and A. de Brauwere. Simple test cases for validating a finite element unstructured grid fecal bacteria transport model. *Applied Mathematical Modelling*, 34(10):3055–3070, 2010. ISSN 0307-904X.
doi: 10.1016/j.apm.2010.01.012. Cited on p. 8.

T. Kärnä, B. de Brye, O. Gourgue, J. Lambrechts, R. Comblen, V. Legat, and E. Deleersnijder. A fully implicit wetting-drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):509–524, 2011. ISSN 0045-7825.
doi: 10.1016/j.cma.2010.07.001. Cited on p. 8.

T. Kärnä, V. Legat, E. Deleersnijder, and H. Burchard. Coupling of a discontinuous Galerkin finite element marine model with a finite difference turbulence closure model. *Ocean Modelling*, 47:55–64, 2012. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2012.01.001. Cited on p. 8, 76.

A. Kasahara. Various vertical coordinate systems used for numerical weather prediction. *Monthly Weather Review*, 102(7):509–522, 1974. Cited on p. 74.

H. Kato and O. M. Phillips. On the penetration of a turbulent layer into stratified fluid. *Journal of Fluid Mechanics*, 37(04):643–655, 1969. Cited on p. 61.

A. B. Kennedy, Q. Chen, J. T. Kirby, and R. A. Dalrymple. Boussinesq Modeling of Wave Transformation, Breaking, and Runup. I: 1D. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 126(1):39–47, Jan 2000. Cited on p. 18, 43.

G. Kesserwani and Q. Liang. *Journal of Scientific Computing*, 50(1):120–144, 2012. Cited on p. 110.

P. D. Killworth, D. J. Webb, D. Stainforth, and S. M. Paterson. The Development of a Free-Surface Bryan–Cox–Semtner Ocean Model. *Journal of Physical Oceanography*, 21(9):1333–1348, 1991. doi: 10.1175/1520-0485(1991)021. Cited on p. 11, 73.

D. Kuzmin. On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. *Journal of Computational Physics*, 219(2): 513–531, 2006. Cited on p. 111.

D. Kuzmin. A vertex-based hierarchical slope limiter for hp-adaptive discontinuous Galerkin methods. *Journal of Computational and Applied Mathematics*, 233(12):3077–3085, 2010. ISSN 0377-0427. doi: 10.1016/j.cam.2009.05.028. Cited on p. 12, 73, 96, 120.

D. Kuzmin, R. Löhner, and S. Turek. *Flux–Corrected Transport. Principles, Algorithms and Applications.* Scientific Computation. Springer, Berlin, 2005. ISBN 3-540-23730-5. Cited on p. 96.

J. Lambrechts, R. Comblen, V. Legat, C. Geuzaine, and J.-F. Remacle. Multiscale mesh generation on the sphere. *Ocean Dynamics*, 58(5):461–473, Dec 2008a. Cited on p. 6, 39, 52.

J. Lambrechts, E. Hanert, E. Deleersnijder, P.-E. Bernard, V. Legat, J.-F. Remacle, and E. Wolanski. A multi-scale model of the hydrodynamics of the whole Great Barrier Reef. *Estuarine, Coastal and Shelf Science*, 79(1): 143–151, 2008b. Cited on p. 5, 72.

M. Leclerc, J.-F. Bellemare, G. Dumas, and G. Dhatt. A finite element model of estuarian and river flows with moving boundaries. *Advances in Water Resources*, 13(4):158–168, 1990. Cited on p. 32.

R. J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002. Cited on p. 83.

P. J. Luyten, E. Deleersnijder, J. Ozer, and K. G. Ruddick. Presentation of a family of turbulence closure models for stratified shallow water flows and preliminary application to the Rhine outflow region. *Continental Shelf Research*, 16(1):101–130, 1996. ISSN 0278-4343. doi: 10.1016/0278-4343(95)93591-V. Cited on p. 48.

P. J. Lynett, T.-R. Wu, and P. L. F. Liu. Modeling wave runup with depth-integrated equations. *Coastal Engineering*, 46(2):89–107, Jul 2002. ISSN 0378-3839. Cited on p. 34.

E. Marchandise and J.-F. Remacle. A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. *Journal of Computational Physics*, 219(2):780–800, Dec 2006. ISSN 0021-9991. Cited on p. 16.

J. Marshall, A. Adcroft, J.-M. Campin, S. Dutkiewicz, C. Evangelinos, D. Ferreira, G. Forget, B. Fox-Kemper, P. Heimbach, C. Hill, E. Hill, H. Hill, O. Jahn, M. Losch, G. Maze, D. Menemenlis, and A. Molod. MITgcm User Manual. Technical report, MIT EAPS, 2004. Cited on p. 4, 11, 72.

G. Mellor and T. Yamada. Development of a turbulence closure model for geophysical fluid problems. *Reviews of Geophysics & Space Physics*, 20(4): 851–875, 1982. Cited on p. 48, 50, 51.

G. L. Mellor. Users guide for a three-dimensional, primitive equation, numerical ocean model (June 2004 version). Technical report, Princeton University, 2004. Cited on p. 4, 11, 72, 100.

C. Nielsen and C. Apelt. Parameters Affecting the Performance of Wetting and Drying in a Two-Dimensional Finite Element Long Wave Hydrodynamic Model. *Journal of Hydraulic Engineering*, 129(8):628–636, Aug 2003. Cited on p. 18, 31, 32, 43.

I. Nikolos and A. Delis. An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. *Computer Methods in Applied Mechanics and Engineering*, 198(47-48):3723–3750, 2009. ISSN 0045-7825. Cited on p. 17, 18, 22, 23.

T. R. Osborn. Estimates of the local rate of vertical diffusion from dissipation measurements. *Journal of Physical Oceanography*, 10:83–89, 1980. Cited on p. 51.

R. C. Pacanowski and S. G. H. Philander. Parameterization of vertical mixing in numerical models of tropical oceans. *Journal of Physical Oceanography*, 11(11):1443–1451, 1981. Cited on p. 49.

J. Pietrzak. The Use of TVD Limiters for Forward-in-Time Upstream-Biased Advection Schemes in Ocean Modeling. *Monthly Weather Review*, 126(3): 812–830, Mar 1998. ISSN 0027-0644.
doi: 10.1175/1520-0493(1998)126. Cited on p. 96.

J. Pietrzak, J. B. Jakobson, H. Burchard, H. J. Vested, and O. Petersen. A three-dimensional hydrostatic model for coastal and ocean modelling using a generalised topography following co-ordinate system. *Ocean Modelling*, 4

(2):173–205, 2002. ISSN 1463-5003.
doi: 10.1016/S1463-5003(01)00016-6. Cited on p. 74.

J. F. Price. On the scaling of stress-driven entrainment experiments. *Journal of Fluid Mechanics*, 90(03):509–529, 1979.
doi: 10.1017/S0022112079002366. Cited on p. 62.

Y. R. Rao, C. R. Murthy, and P. C. Sinha. The Coastal Ocean. In *Modelling and Monitoring of Coastal Marine Processes*, pages 3–10. Springer Netherlands, 2008. ISBN 978-1-4020-8327-3. Cited on p. 2.

H. Rennau and H. Burchard. Quantitative analysis of numerically induced mixing in a coastal model application. *Ocean Dynamics*, 59:671–687, 2009. ISSN 1616-7341. Cited on p. 67.

B. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*. Number 35 in Frontiers in Applied Mathematics. SIAM, 2008. Cited on p. 19, 83.

W. Rodi. Examples of calculation methods for flow and mixing in stratified flows. *Journal of Geophysical Research*, 92:5305–5328, 1987. Cited on p. 48, 51, 76.

P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 135(2):250–258, 1997. ISSN 0021-9991.
doi: 10.1006/jcph.1997.5705. Cited on p. 22, 23.

K. Ruddick, E. Deleersnijder, T. Mulder, and P. Luyten. A model study of the Rhine discharge front and downwelling circulation. *Tellus A*, 46(2):149–159, 1994. Cited on p. 101.

K. Ruddick, E. Deleersnijder, P. Luyten, and J. Ozer. Haline stratification in the Rhine-Meuse freshwater plume: a three-dimensional model sensitivity analysis. *Continental Shelf Research*, 15(13):1597–1630, 1995. ISSN 0278-4343.
doi: 10.1016/0278-4343(95)00034-X. Cited on p. 48, 101.

K. Rygg, L. Enstad, and G. Alendal. Simulating CO2 transport into the ocean from a CO2 lake at the seafloor using a z- and a σ-coordinate model. *Ocean Dynamics*, 59:795–808, 2009. ISSN 1616-7341. Cited on p. 48.

Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, Jul 1986. Cited on p. 27.

M. Santillana and C. Dawson. A local discontinuous Galerkin method for a doubly nonlinear diffusion equation arising in shallow water modeling.

Computer Methods in Applied Mechanics and Engineering, 199(23-24):1424–1436, 2010. ISSN 0045-7825. Cited on p. 18.

M. Schlegel, O. Knoth, M. Arnold, and R. Wolke. Multirate Runge–Kutta schemes for advection equations. *Journal of Computational and Applied Mathematics*, 226(2):345–357, 2009. ISSN 0377-0427. Cited on p. 112.

B. Seny, J. Lambrechts, R. Comblen, V. Legat, and J.-F. Remacle. Multirate time stepping for accelerating explicit discontinuous Galerkin computations with application to geophysical flows. *International Journal for Numerical Methods in Fluids*, Accepted for publication, 2012. ISSN 1097-0363. doi: 10.1002/fld.3646. Cited on p. 113.

K. Shahbazi. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics*, 205(2):401–407, 2005. Cited on p. 84.

A. F. Shchepetkin and J. C. McWilliams. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9(4):347–404, 2005. ISSN 1463-5003. doi: 10.1016/j.ocemod.2004.08.002. Cited on p. 4, 11, 12, 72, 73, 93, 94, 95, 106, 119.

A. F. Shchepetkin and J. C. McWilliams. *Computational Kernel Algorithms for Fine-Scale, Multiprocess, Longtime Oceanic Simulations*, volume Volume 14, pages 121–183. Elsevier, 2009. ISBN 1570-8659. Cited on p. 91, 120.

L. H. Shih, J. R. Koseff, J. H. Ferziger, and C. R. Rehmann. Scaling and parameterisation of stratified homogeneous turbulent shear flow. *Journal of Fluid Mechanics*, 412:1–20, 2000. Cited on p. 51.

J. Simpson, J. Brown, J. Matthews, and G. Allen. Tidal straining, density currents, and stirring in the control of estuarine stratification. *Estuaries and Coasts*, 13(2):125–132, 1990. Cited on p. 103.

J. H. Simpson. Physical processes in the ROFI regime. *Journal of Marine Systems*, 12(1–4):3–15, 1997. ISSN 0924-7963. doi: 10.1016/S0924-7963(96)00085-1. Cited on p. 100.

J. H. Simpson, W. G. Bos, F. Shirmer, A. Souza, T. Rippeth, S. Jones, and D. Hydes. Periodic stratification in the Rhine ROFI in the North Sea. *Oceanologica Acta*, 16(1):23–32, 1993. Cited on p. 4, 103.

R. J. Sobey. Wetting and drying in coastal flows. *Coastal Engineering*, 56: 565–576, May 2009. ISSN 0378-3839. Cited on p. 16.

P. Solin, K. Segeth, and I. Dolezel. *Higher-Order Finite Element Methods*. Studies in Advanced Mathematics. CRC, 2003. Cited on p. 81.

Y. Song and D. Haidvogel. A Semi-implicit Ocean Circulation Model Using a Generalized Topography-Following Coordinate System. *Journal of Computational Physics*, 115(1):228–244, 1994. ISSN 0021-9991. doi: 10.1006/jcph.1994.1189. Cited on p. 74.

G. S. Stelling and S. P. A. Duinmeijer. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *International Journal for Numerical Methods in Fluids*, 43(12):1329–1354, 2003. ISSN 1097-0363. Cited on p. 17.

C. M. Swinkels, C. M. C. J. L. Jeuken, Z. B. Wang, and R. J. Nicholls. Presence of Connecting Channels in the Western Scheldt Estuary. *Journal of Coastal Research*, 25(3):627–640, May 2009. ISSN 0749-0208. Cited on p. 39, 40.

W. C. Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics Digital Archive*, 107(-1):499–508, 1981. Cited on p. 33, 36.

R. Tian and C. Chen. Influence of model geometrical fitting and turbulence parameterization on phytoplankton simulation in the Gulf of Maine. *Deep Sea Research Part II: Topical Studies in Oceanography*, 53(23-24):2808–2832, 2006. ISSN 0967-0645. doi: 10.1016/j.dsr2.2006.08.006. Cited on p. 48.

E. Toro. *Riemann solvers and numerical methods for fluid dynamics: A practical introduction*. Springer Verlag, 2009. Cited on p. 81, 83.

L. Umlauf and H. Burchard. A generic length-scale equation for geophysical turbulence models. *Journal of Marine Research*, 61:235–265(31), 2003. Cited on p. 48, 50, 51.

L. Umlauf and H. Burchard. Second-order turbulence closure models for geophysical boundary layers. A review of recent work. *Continental Shelf Research*, 25(7-8):795–827, 2005. ISSN 0278-4343. Cited on p. 51.

L. Umlauf, H. Burchard, and K. Hutter. Extending the k-[omega] turbulence model towards oceanic applications. *Ocean Modelling*, 5(3):195–218, 2003. ISSN 1463-5003. Cited on p. 48, 51, 63.

B. van't Hof and E. A. H. Vollebregt. Modelling of wetting and drying of shallow water using artificial porosity. *International Journal for Numerical Methods in Fluids*, 48(11):1199–1217, 2005. ISSN 1097-0363. Cited on p. 17, 18.

J. J. Walsh. *On the nature of continental shelves*, volume 508. Academic Press, 1988. ISBN 012733775X. Cited on p. 2.

D.-P. Wang. Mutual intrusion of a gravity current and density front formation. *Journal of Physical Oceanography*, 14(7):1191–1199, 1984. ISSN 0022-3670. doi: 10.1175/1520-0485(1984)014. Cited on p. 97.

Q. Wang. *The finite element ocean model and its aspect of vertical discretization*. PhD thesis, Universität Bremen, 2007. Cited on p. 11, 49, 72.

Q. Wang, S. Danilov, and S. J. Comparison of overflow simulations on different vertical grids using the Finite Element Ocean circulation Model. *Ocean Modelling*, 20(4):313–335, 2008. Cited on p. 11.

J. Warner, W. Rockwell Geyer, and H. Arango. Using a composite grid approach in a complex coastal domain to estimate estuarine residence time. *Computers & Geosciences*, 36(7):921–935, 2010. Cited on p. 72.

J. C. Warner, C. R. Sherwood, H. G. Arango, and R. P. Signell. Performance of four turbulence closure models implemented using a generic length scale method. *Ocean Modelling*, 8(1-2):81–113, 2005. ISSN 1463-5003. doi: 10.1016/j.ocemod.2003.12.003. Cited on p. 13, 48, 58, 64, 65, 66, 67, 68, 69, 111.

L. White, E. Deleersnijder, and V. Legat. Tracer conservation for three-dimensional, finite-element, free-surface, ocean modeling on moving prismatic meshes. *Monthly Weather Review*, 136(2):420–442, 2008a. Cited on p. 8, 11, 12, 29, 73, 78, 111.

L. White, E. Deleersnijder, and V. Legat. 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(1-2):26–47, 2008b. Cited on p. 8, 11, 49, 52, 77, 111.

H. W. Wijesekera, J. S. Allen, and P. A. Newberger. Modeling study of turbulent mixing over the continental shelf: Comparison of turbulent closure schemes. *J. Geophys. Res.*, 108(C3):3103, Mar 2003. ISSN 0148-0227. doi: 10.1029/2001JC001234. Cited on p. 48.

D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26:1299–1310, 1988. Cited on p. 48, 51.

E. Wolanski. *Estuarine Ecohydrology*. Elsevier, 2007. ISBN 9780444530660. Cited on p. 2.

D. Yuan, B. Lin, and R. Falconer. Simulating moving boundary using a linked groundwater and surface water flow model. *Journal of Hydrology*, 349(3-4): 524–535, Feb 2008. ISSN 0022-1694. Cited on p. 16, 18.

Y. Zhang and A. M. Baptista. SELFE: A semi-implicit Eulerian-Lagrangian finite-element model for cross-scale ocean circulation. *Ocean Modelling*, 21

(3-4):71–96, 2008. ISSN 1463-5003.
doi: 10.1016/j.ocemod.2007.11.005. Cited on p. 4, 11, 49, 72, 110, 112.

D. H. Zhao, H. W. Shen, J. S. Lai, and G. Q. Tabios III. Approximate Riemann Solvers in FVM for 2D Hydraulic Shock Wave Modeling. *Journal of Hydraulic Engineering*, 122(12):692–702, 1996.
doi: 10.1061/(ASCE)0733-9429(1996)122:12(692). Cited on p. 22, 23.

L. Zheng, C. Chen, and H. Liu. A modeling study of the Satilla River estuary, Georgia. I: Flooding-drying process and water exchange over the salt marsh-estuary-shelf complex. *Estuaries and Coasts*, 26(3):651–669, Jun 2003. Cited on p. 16.

M. Zijlema and G. S. Stelling. Efficient computation of surf zone waves using the nonlinear shallow water equations with non-hydrostatic pressure. *Coastal Engineering*, 55(10):780–790, Oct 2008. ISSN 0378-3839. Cited on p. 16.