



**HAL**  
open science

# A vertically-Lagrangian, non-hydrostatic, multilayer model for multiscale free-surface flows

Stéphane Popinet

► **To cite this version:**

Stéphane Popinet. A vertically-Lagrangian, non-hydrostatic, multilayer model for multiscale free-surface flows. *Journal of Computational Physics*, 2020, 10.1016/j.jcp.2020.109609 . hal-02365730v1

**HAL Id: hal-02365730**

**<https://hal.science/hal-02365730v1>**

Submitted on 15 Nov 2019 (v1), last revised 2 Jun 2020 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A vertically-Lagrangian, non-hydrostatic, multilayer model for multiscale free-surface flows

BY STÉPHANE POPINET

Sorbonne Université, CNRS, Institut Jean le Rond d'Alembert, 75005 Paris, France

*November 15, 2019*

## Abstract

This article presents a semi-discrete, multilayer set of equations describing the three-dimensional motion of an incompressible fluid bounded below by topography and above by a moving free-surface. This system is a consistent discretisation of the incompressible Euler equations, valid without assumptions on the slopes of the interfaces. Expressed as a set of conservation laws for each layer, the formulation has a clear physical interpretation and makes a seamless link between the hydrostatic Saint-Venant equations, dispersive Boussinesq-style models and the incompressible Euler equations. The associated numerical scheme, based on an approximate vertical projection and multigrid-accelerated column relaxations, provides accurate and efficient solutions for all regimes. The same model can thus be applied to study metre-scale waves, even beyond breaking, with results closely matching those obtained using small-scale Euler/Navier–Stokes models, and coastal or global scale dispersive waves, with an accuracy and efficiency comparable to extended Boussinesq wave models. The implementation is adaptive, parallel and open source as part of the Basilisk framework and the documented source codes sufficient to reproduce all results and figures are provided.

Keywords: 3D incompressible Euler; free-surface; approximate projection; adaptive; wave breaking; Boussinesq

## 1 Introduction

Deformations of different materials separated by sharp interfaces occur in many fundamental phenomena. The physical understanding and modelling of these systems require the introduction of suitable mathematical representations of the geometry of the interfaces and, more generally, space and time. In the general case of arbitrarily large deformations, characteristic of fluid motion, this is not a trivial question. It is also well-known, since the time of Euler and Lagrange, that the structure (and complexity) of the equations describing motion, as well as their physical interpretation, is closely connected to the choice of geometrical representation (frame of reference, coordinate system etc.).

The geometric concept of “layered” materials is particularly useful in solid mechanics (composites, shells and membranes etc.), fluid mechanics (boundary layers, liquid films and lubrication etc.) and geophysical fluid dynamics (avalanches, glaciers, ice shelves, ocean, atmosphere etc.). The advantages of this type of representation are twofold: 1) the geometries of material interfaces are directly described as graphs of uni-valued/differentiable functions, 2) the dimensional anisotropy characteristic of layers (i.e. quantities varying

much more slowly/rapidly along the layers than across) and the associated coordinate system can be used to decouple physical effects and obtain simplified models and equation sets.

A typical example of this approach are the “shallow-water equations” which describe the motion of a single layer bounded vertically by two material interfaces: the bathymetry and free-surface. They were first derived by Saint-Venant using conservation of mass and momentum in vertical slices and the assumption that the vertical pressure profile is essentially hydrostatic. An important limitation of the Saint-Venant equations is that they are strictly consistent only for a vertically-constant velocity profile (a “plug flow”). Lifting this limitation requires the ability to describe the vertical velocity profile, which can be done by considering multiple layers.

Since these additional layers are not generally bounded by material interfaces, their vertical motion can be chosen freely and does not have to follow vertically-Lagrangian dynamics. This vertically-Eulerian point of view is dominant amongst the shallow-water models used to describe oceanic or atmospheric dynamics. It has however the important drawback of introducing vertical advection terms which break the horizontal/vertical decoupling due to anisotropy (point 2 above) and thus complicate the equations, their physical interpretation and numerical approximation.

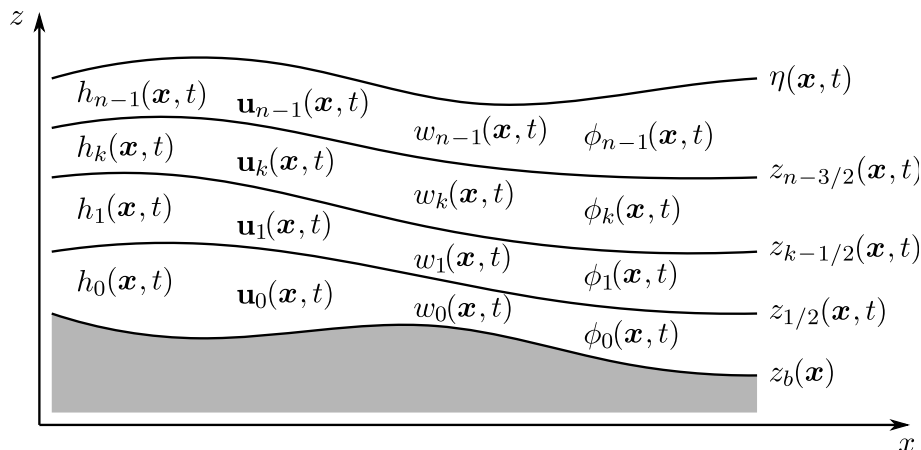
The hydrostatic assumption is another obvious limitation of Saint-Venant systems, which, as a consequence, cannot represent the dispersion of short gravity waves. If this limitation is lifted, multilayer systems could in principle become consistent with layer integration of the (incompressible) Euler equations with a free-surface and bathymetry. The resulting physical, mathematical and numerical model would then be valid without assumptions on the anisotropy of the flow, while preserving the ability to remove terms (and the corresponding numerical code) in anisotropic cases. It would thus provide a seamless link between the (single layer) Saint-Venant equations and a consistent discretisation of the incompressible free-surface Euler equations. This would be of considerable practical interest generally, but in particular for geophysical flows, since the resulting model hierarchy would be applicable and efficient for scales ranging from metres to the global scale.

The goal of this article is to present such a model. We first give its formal description in Section 2 and relate it to existing work in Section 2.1. The associated numerical scheme is described in Section 3, split into hydrostatic and non-hydrostatic parts, with a particular emphasis on accurate dispersion relations (Section 3.5). Section 4 discusses several test cases and applications, for a broad range of spatial scales and with measures of accuracy and performance.

The source code necessary to reproduce all results and figures is freely accessible (GPL license) as part of the Basilisk framework [1].

## 2 Model equations

We consider  $n$  layers of an incompressible (inviscid) fluid bounded vertically by a fixed bottom topography and a free surface (Figure 1). The properties of each layer are functions of time and two-dimensional space only. The resulting model is thus *semi-discrete* i.e. horizontally continuous and two-dimensional, and vertically discrete.



**Figure 1.** Definition of the  $n$  layers.  $h_k$  is the layer thickness,  $(\mathbf{u}, w)_k$  the velocity vector in the  $\mathbf{x}$ - $z$  reference frame,  $\phi_k$  the non-hydrostatic pressure,  $z_{k-1/2}$  the height of the layer interface and  $\eta$  the free-surface height. Bold letters are used for vector fields.

We propose the following semi-discrete approximation of the incompressible Euler equations with a free-surface and gravity.

$$\partial_t h_k + \nabla \cdot (h \mathbf{u})_k = 0, \quad (1)$$

$$\partial_t (h \mathbf{u})_k + \nabla \cdot (h \mathbf{u} \mathbf{u})_k = -g h_k \nabla \eta - \nabla (h \phi)_k + [\phi \nabla z]_k, \quad (2)$$

$$\partial_t (h w)_k + \nabla \cdot (h w \mathbf{u})_k = -[\phi]_k, \quad (3)$$

$$\nabla \cdot (h \mathbf{u})_k + [w - \mathbf{u} \cdot \nabla z]_k = 0, \quad (4)$$

with  $k$  the index of the layer,  $h_k$  its thickness,  $\mathbf{u}_k, w_k$  the horizontal and vertical components of the velocity,  $g$  the acceleration of gravity,  $\phi_k$  the non-hydrostatic pressure (divided by the fluid density),

$$\eta = z_b + \sum_k h_k,$$

the free-surface height, with  $z_b$  the height of the bottom topography,

$$z_{k+1/2} = z_b + \sum_{l=0}^k h_l,$$

the height of each layer interface (see Figure 1) and

$$[f]_k = f_{k+1/2} - f_{k-1/2},$$

the “vertical difference” operator. Note that since all fields are two-dimensional, so is the  $\nabla$  operator.

Equation (1) is the layer thickness evolution equation, equations (2) and (3) express conservation of the horizontal and vertical momentum, and (4) is the semi-discrete continuity equation (or semi-discrete incompressibility condition) which expresses volume (and thus mass) conservation.

The system must then be closed by assuming relations between the layer-averaged values  $\phi_k, w_k$  and the layer-interface values  $\phi_{k+1/2}, w_{k+1/2}$ . We will outline a strategy to optimise this choice in section 3.5.

Note that the formal derivation of this system is not entirely trivial, in particular for the “slope terms”  $\nabla z$  in (2) and (4). We give further details on their derivation in Appendix A.

**Remark 1.** One may feel that equations (1) and (4) are redundant (since they appear to both express volume conservation). If this was the case, the system would not be closed since there are four unknowns per layer ( $h$ ,  $\mathbf{u}$ ,  $w$  and  $\phi$ ). Note however that there is no vertical advection term (of the form  $\partial_z(h \mathbf{u} w)_k$ ) in either equations (2) or (3). This is of course because we have assumed that the layer interfaces (as defined by  $z_{k+1/2}$ ) are material surfaces which follow the vertical motion of the fluid i.e. the discretisation is Lagrangian vertically. This implies a kinematic condition, relating  $z$  and  $w$ , which must be added to (2) and (3). This condition can be written

$$w = \partial_t z + \mathbf{u} \cdot \nabla z \quad (5)$$

Using the vertical difference operator, the fact that  $h_k = [z]_k$  and (4) we get

$$\begin{aligned} [w]_k &= \partial_t [z]_k + [\mathbf{u} \cdot \nabla z]_k, \\ [w - \mathbf{u} \cdot \nabla z]_k &= \partial_t h_k, \\ -\nabla \cdot (h \mathbf{u})_k &= \partial_t h_k, \end{aligned}$$

which is the layer evolution equation (1). Equation (1) is then equivalent to condition (5) and comes from the particular choice of a Lagrangian vertical coordinate system. It does not represent a specific physical property of the system (with the exception of the bottom and top boundary conditions). Equation (4) on the other hand is true irrespective of the choice of vertical coordinate and is necessary and sufficient to enforce volume/mass conservation. This point is further discussed in relation to previous work in Section 2.1.

**Remark 2.** While the evolution equations for  $h$ ,  $\mathbf{u}$  and  $w$  are obvious, the evolution equation for  $\phi$  is not. This is classical for incompressible fluids for which the pressure becomes a gauge field used to project the velocity onto the space of divergence-free solutions [2, 3]. Equation (4) is the equivalent semi-discrete “divergence-free” condition which will be used to build the Poisson-like equation which  $\phi$  must verify.

**Remark 3.** Setting  $\phi = 0$  and dropping the requirement of conservation of the vertical momentum (equation (3)) gives

$$\partial_t h_k + \nabla \cdot (h \mathbf{u})_k = 0, \quad (6)$$

$$\partial_t (h \mathbf{u})_k + \nabla \cdot (h \mathbf{u} \mathbf{u})_k = -g h_k \nabla \eta, \quad (7)$$

$$\nabla \cdot (h \mathbf{u})_k + [w - \mathbf{u} \cdot \nabla z]_k = 0, \quad (8)$$

which are the standard hydrostatic, multilayer Saint-Venant (or stacked shallow-water) equations. The vertical velocity  $w$  becomes a diagnostic variable which can be obtained from  $\mathbf{u}$  and  $h$  using equation (8). Note that (5) could be used instead, however the resulting value of  $w$  would then not necessarily verify the discrete continuity equation (8).

**Remark 4.** The system of equations (1), (2), (3) and (4) is consistent even for a single layer for which it can be re-written

$$\partial_t h + \nabla \cdot (h \mathbf{u}) = 0,$$

$$\partial_t (h \mathbf{u}) + \nabla \cdot (h \mathbf{u} \mathbf{u}) = -g h \nabla \eta - \nabla (h \phi) - \phi_{-1/2} \nabla z_b,$$

$$\partial_t (h w) + \nabla \cdot (h w \mathbf{u}) = \phi_{-1/2}, \quad (9)$$

$$\nabla \cdot (h \mathbf{u}) + w_{1/2} - \mathbf{u}_{1/2} \cdot \nabla \eta = 0, \quad (10)$$

where we have used the free-surface boundary condition  $\phi_{1/2} = 0$  and bottom boundary impermeability condition  $(w - \mathbf{u} \cdot \nabla z)_{-1/2} = 0$ . The system is then closed by assuming functional relationships between the bottom pressure  $\phi_{-1/2}$  and free surface velocity  $(\mathbf{u}_{1/2}, w_{1/2})$  and the “layer-averaged” pressure  $\phi$  and velocity  $(\mathbf{u}, w)$ ; the simplest of which could be  $\phi_{-1/2} = \phi$ ,  $(\mathbf{u}_{1/2}, w_{1/2}) = (\mathbf{u}, w)$ . Better choices are discussed in section 3.5 in the general multilayer case. For a single layer, they reduce to  $\phi_{-1/2} = \frac{8}{3} \phi$ ,  $(\mathbf{u}_{1/2}, w_{1/2}) = (\mathbf{u}, w)$  for the staggered scheme of section 3.6.1, and  $\phi_{-1/2} = 2 \phi$ ,  $(\mathbf{u}_{1/2}, w_{1/2}) = (\mathbf{u}, 2w)$  for the Keller box scheme of section 3.6.2.

**Remark 5.** Although we have not considered density variations and other terms such as Coriolis forces or viscosity for simplicity, their inclusion does not fundamentally change the structure of the system. We will discuss later how to include the (dominant) vertical viscosity using well established techniques. Note also that additional tracers, such as temperature, salinity etc. are easily added through advection equations of the form

$$\partial_t(ch)_k + \nabla \cdot (ch \mathbf{u})_k = S_k,$$

where  $c$  is a volumetric concentration and  $S$  source terms.

## 2.1 Related work

Given its basic nature it is clear that the semi-discrete system we propose is related to a broad range of approaches which we attempt to review here.

### 2.1.1 Hydrostatic system

The single layer limit of the hydrostatic system (6), (7) and (8) is the classical Saint-Venant system [4]. Its multilayer generalisation is also well-known and is discussed in many textbooks on geophysical fluid dynamics, in particular in connection with the theoretical study of internal wave dynamics, using the two-layer version [5, 6, 7]. From a modelling perspective, this system is also the basis for isopycnal hydrostatic discretisations of the equations of oceanic motion [8] or isentropic discretisations of the atmospheric equations [9, 10].

Another classical variant of the hydrostatic system (6), (7) and (8) is the corresponding general “quasi Eulerian” vertical discretisation for which the layer positions  $z_{k+1/2}$  are diagnosed from the other variables rather than computed as part of the solution (using equation (6)). In this case, the vertical advection fluxes must be added to equation (7) and the layer evolution equation (6) is replaced by the evolution equation for the total depth. Classical examples of this type of discretisation are the  $z$ - and  $\sigma$ -coordinates models typically used for ocean modelling [11]: for  $z$ -coordinates the layer positions are constant i.e. the discretisation is strictly Eulerian vertically, while for  $\sigma$ -coordinates the layer positions are proportional to the total depth and thus vary in both space and time (i.e. quasi-Eulerian).

Interesting overviews of the pros and cons of the purely Lagrangian (i.e. isopycnal) vertical discretisation versus Eulerian or quasi-Eulerian discretisations ( $z$ - or  $\sigma$ -coordinates) are given by Griffies et al. (2000) [11], Adcroft & Hallberg (2006) [12] and Chassignet et al. (2011) [13]. Either versions of the multilayer hydrostatic equations have also been studied fairly recently by applied mathematicians in a different context, more closely linked to hydraulic engineering, seeking to clarify the consistency of shock-like solutions (hydraulic jumps) in the presence of viscous dissipation [14, 15, 16].

### 2.1.2 Non-hydrostatic system

The generalisation of the layered equations to include non-hydrostatic effects is more recent. In ocean modelling this effort has been driven in particular by the increase in computing power which allows to resolve spatial scales for which non-hydrostatic effects are not negligible or even dominant. This “convergence of scales” also means that the approaches developed in scientific communities which were formerly interested in different scales are now also converging.

For example coastal engineers have long been interested in describing oceanic motion (mostly waves) at scales of order of metres. This motivated some of the earliest developments in the non-hydrostatic description of the motion of fluid layers, starting from Boussinesq [17] and much later, Serre (for hydraulic engineering in dams rather than the ocean) [18], Peregrine [19] and Green and Naghdi [20]. Since then these “extended-Boussinesq” wave models have known major developments which are summarised in the interesting review by Brocchini (2013) [21].

The equations typically used in these models are closely related to the hydrostatic Saint-Venant system, often with a single layer (see Lynett & Liu (2004) [22] or Chazel et al. (2009) [23] for multilayer versions), but without the additional vertical momentum or continuity equations (9) and (10). Rather, the effects of the non-hydrostatic pressure are modelled by including source terms on the right-hand-side of the mass and/or momentum equations (6) and (7). These terms are typically obtained through asymptotic expansion/truncation of the Euler velocity potential. While these equations are thus structurally different from the non-hydrostatic single or multi-layer equations above, a common goal is the accurate description of short waves and in particular their dispersion relation as well as non-linear behaviour.

In parallel with these developments, interest in small-scale hydrodynamics with free surfaces and interfaces also led to extensive work on suitable numerical methods. Key developments in this area are the invention of the staggered (MAC or C-grid) discretisation and associated projection method for incompressible flows. Interestingly these were discovered independently in different communities by Harlow and Welch (1965) [24] (small-scale free-surface incompressible flows) and Arakawa (1966) [25] (large-scale geophysical flows). This gap between communities was only closed relatively recently, for example with the work of Marshall and Adcroft to include non-hydrostatic terms in the MITgcm ocean model, using the MAC projection scheme within a  $z$ -coordinate ocean model [26].

Also in the late 1990s, coastal engineers started adapting these small-scale non-hydrostatic numerical methods to specific aspects of their field. Representative examples of this trend are the work of Casulli [27] and Stelling & Zijelma [28]. Both of these schemes still use a  $z$ -coordinate in the vertical so do not differ fundamentally from standard MAC/C-grid incompressible solvers (with a free-surface). Later work by Zijelma & Stelling, however, generalises the numerical scheme to a  $\sigma$  vertical coordinate [29]. The resulting system of equations is thus related to the semi-discrete system (1), (2), (3) and (4), with the fundamental difference of a quasi-Eulerian treatment of the vertical coordinate.

Larger-scale ocean modellers have also been interested in including non-hydrostatic effects, in particular in the  $\sigma$ -coordinate models often used for “regional” ocean studies [30]. These schemes are generally based on the approaches developed by Casulli, Stelling and Zijelma and the resulting set of equations do not differ fundamentally.

Non-hydrostatic, vertically-Lagrangian descriptions of the equations of motion are thus rare, which may be partly due to the confusion between the layer evolution equation and the continuity equation mentioned in Remark 1 above (see in particular [12] and [31] for a clarification). A notable exception is the article by Vitousek & Fringer (2014) [32] which presents a mild slope approximation of the non-hydrostatic equations expressed in



isopycnal coordinates. The authors derive an interesting scaling analysis to justify the validity of the mild slope assumption for the oceanic scales they are interested in. We will show later that these terms cannot be neglected for typical coastal-engineering-scale processes, even when the slopes are relatively mild. We will also show how these terms can be retained while keeping the structure and numerical efficiency of the simplified system.

In parallel with these developments, the search for efficient short-wave models applicable to coastal engineering led to the idea that these multilayer non-hydrostatic models, used with as few layers as possible (ideally a single one), could be an efficient alternative to Boussinesq-type models. Zijelma and Stelling showed in particular (somewhat empirically) that a particular version of their numerical scheme had good dispersion properties even for (very) short waves using only two layers [28, 33]. This led Yamazaki et al. (2008) [34] to propose a single-layer model similar to the single-layer model in Remark 4 above, with the notable difference of the absence of the convective term in the vertical momentum equation (9) and neglecting all metric/slope terms (essentially a linearised version). This single-layer model was also re-derived more recently by Bristeau et al. (2015) [35] who pointed out that the inclusion of the convective term is necessary for energy consistency.

## 2.2 Properties

The semi-discrete system (1), (2), (3) and (4) has a number of favourable features. It only relies on layer-averaging of the incompressible Euler equations, without any other approximation. In particular it is consistent irrespective of the slope of the layers. The vertically-Lagrangian discretisation eliminates any numerical vertical diffusion which is particularly important for long-term integration of ocean models [8]. When using flux-based finite-volume schemes for transport, horizontal momentum as well as tracers are exactly conserved in each layer. Vertical momentum should also be conserved, although this is more difficult to guarantee in practice (than for horizontal momentum) because it depends on the details of the numerical scheme used in the vertical (see section 3.6). Finally, the main advantage of this system is that it provides a clear and natural hierarchy of models, of increasing accuracy (and complexity): single layer Saint-Venant, multilayer (hydrostatic) Saint-Venant, (non-hydrostatic) multilayer free-surface Euler (and Navier–Stokes).

The drawbacks are few. The most obvious one is the addition of the layer thicknesses as prognostic variables. The added complexity and computational cost are low however, since they are passive tracers which will be transported using the same scheme as other tracers. The multilayer Saint-Venant system is known to be non-hyperbolic [14] with possibly non-unique solutions for the layer thicknesses. This is not surprising and reflects the lack of physical coupling between layers. This is cured in the non-hydrostatic case and/or when using vertical remapping (see section 3.3).

## 3 Numerical scheme

The numerical scheme is designed in a modular fashion, to match the increasing complexity and accuracy of the successive theoretical approximations. The goal is a numerical tool where mathematical terms (and the corresponding code), can be added one-by-one depending on the balance required between accuracy, complexity and numerical efficiency. We also aim for generality and try to optimise the following properties: accurate dispersion relation for gravity waves, conservation and robustness to shocks and wetting and drying. Note that the description below is for two dimensions (one horizontal and one vertical) but the generalisation to two horizontal dimensions is straightforward (and automatic when using the Basilisk framework).



### 3.1 Hydrostatic scheme

We first build a numerical scheme to approximate the multi-layer, hydrostatic system (6), (7) and (8). This system is an hyperbolic system of conservation laws and is often solved based on characteristic (approximate) Riemann solutions. This is important in particular when considering the consistency of shock-like solutions [36, 37]. The resulting numerical schemes are classical but reasonably complex and expensive. They also have the drawback that special measures must be taken to guarantee that the trivial “lake-at-rest” solution is recovered when bathymetry is included [36]. We follow a simpler approach, often taken when building ocean models, which does not explicitly take into account the existence of shock-like solutions.

The time integration of (6), (7) and (8) is split into a first step where all the conserved quantities  $s_k$  are advected according to

$$\begin{aligned} \frac{h_{i,k}^{n+1} - h_{i,k}^n}{\Delta t} &= \frac{F_{i-1/2,k}^{n+1/2} - F_{i+1/2,k}^{n+1/2}}{\Delta}, \\ \frac{(h s)_{i,k}^{n+1} - (h s)_{i,k}^n}{\Delta t} &= \frac{(s F)_{i-1/2,k}^{n+1/2} - (s F)_{i+1/2,k}^{n+1/2}}{\Delta}, \end{aligned}$$

where  $s_{i,k}$  is either the horizontal velocity, vertical velocity or any additional tracer and  $F_{i+1/2,k}$  is the thickness flux  $(h u)_{i+1/2,k}$ , evaluated using the (Lax-Wendroff-like) Bell–Collela–Glaz third-order-upwind advection scheme [38], with a staggered velocity  $u_{i+1/2,k}$  to be defined later. Each index  $i$  such that space is discretised in intervals  $x_{i+1/2} - x_{i-1/2} = \Delta$ . For simplicity, all the conserved variables are collocated and treated as cell-averages over  $[x_{i-1/2}, x_{i+1/2}]$ . Note that this step is standard and could be performed using a broad range of advection schemes. Desirable properties include: exact conservation (which is guaranteed for flux-based finite-volume schemes as here), monotonicity/positivity which is necessary for consistency in particular when “drying” of the layers occur (i.e.  $h_k \rightarrow 0$ ) (this is guaranteed here by upwinding and min-mod limiting).

In a second step, the pressure gradient term (r.h.s. of (7)) is added to the horizontal momentum, with some care. The free-surface position is first calculated as

$$\eta_i^{n+1} = z_{i,b} + H_i^{n+1} \quad \text{with} \quad H_i^{n+1} = \sum_k h_{i,k}^{n+1}$$

The horizontal staggered acceleration is then obtained as

$$a_{i+1/2,k} = -g \frac{\eta_{i+1,k} - \eta_{i,k}}{\Delta} \quad (11)$$

provided the “wet condition”

$$\begin{aligned} H_i > \epsilon \quad \text{and} \quad H_{i+1} > \epsilon \quad \text{or} \\ H_i > \epsilon \quad \text{and} \quad \eta_i \geq z_{i+1,b} \quad \text{or} \\ H_{i+1} > \epsilon \quad \text{and} \quad \eta_{i+1} \geq z_{i,b} \end{aligned} \quad (12)$$

is verified (with  $\epsilon$  a small value, typically  $10^{-6}$ ), otherwise  $a_{i+1/2,k} = u_{i+1/2,k}^{n+1} = 0$ . The staggered velocity field is then computed as

$$u_{i+1/2,k}^{n+1} = \frac{(h u)_{i+1,k}^{n+1} + (h u)_{i,k}^{n+1}}{h_{i+1,k}^{n+1} + h_{i,k}^{n+1}} + \Delta t a_{i+1/2,k} \quad (13)$$

Note that the robustness of this step relies on the fact that  $u$  and  $h$  are collocated and transported using the same advection scheme. Finally the pressure gradient term is added to velocity as

$$u_{i,k}^{n+1} \leftarrow u_{i,k}^{n+1} + \Delta t \frac{a_{i+1/2,k} + a_{i-1/2,k}}{2} \quad (14)$$

Since time integration is fully explicit, we can expect that the classical stability condition

$$\Delta t < \frac{\Delta}{\sqrt{gH}} \quad (15)$$

applies. This scheme is simple, conservative and robust to wetting and drying. We will also show that it can recover shock-like solutions. It is of course only formally first-order accurate in time and is constrained by the maximum speed of gravity waves. Note however that it could be replaced by more sophisticated schemes, using for example implicit time integration [27] or barotropic/baroclinic mode-splitting [39], without affecting its articulation with the (viscous, non-hydrostatic etc.) extensions presented later.

### 3.2 Vertical viscosity

In the case of flows with large aspect ratios, diffusion is expected to be controlled essentially by its vertical component which can be added to the momentum equations (2) and (3) as

$$\begin{aligned} \partial_t(h \mathbf{u})_k + \nabla \cdot (h \mathbf{u} \mathbf{u})_k &= -g h_k \nabla \eta - \nabla (h \phi)_k + [\phi \nabla z]_k + [\nu \partial_z \mathbf{u}]_k, \\ \partial_t(h w)_k + \nabla \cdot (h w \mathbf{u})_k &= -[\phi]_k + [\nu \partial_z w]_k, \end{aligned}$$

with  $\nu$  the kinematic viscosity.

Boundary conditions on the top and bottom layers need to be added to close the system for the viscous stresses. We chose to impose a Neumann condition on the free-surface i.e.

$$\partial_z u|_{t=\dot{u}_t}$$

and a Navier slip condition on the bottom i.e.

$$u|_{b=u_b} + \lambda_b \partial_z u|_b \quad (16)$$

The default boundary conditions are free-slip on the free-surface (i.e.  $\dot{u}_t = 0$ ) and no-slip on the bottom (i.e.  $u_b = \lambda_b = 0$ ).

For stability, we discretise the viscous friction term implicitly as

$$\frac{(h u_k)^{n+1} - (h u_k)^*}{\Delta t} = \nu \left( \frac{u_{k+1} - u_k}{h_{k+1/2}} - \frac{u_k - u_{k-1}}{h_{k-1/2}} \right)^{n+1}$$

This can be expressed as the linear system

$$\mathbf{M} \mathbf{u}^{n+1} = \mathbf{rhs},$$

where  $\mathbf{M}$  is a tridiagonal matrix. A more detailed description of the general algorithm can be found in [16] and in the documented source code (<http://basilisk.fr/src/layered/diffusion.h>).

A significant improvement compared to [16] regards the treatment of the Navier slip bottom condition. The discretisation proposed in [16] is only second-order accurate near the bottom. This can be easily improved using the third-order-accurate discretisation of (16) corresponding to

$$\begin{aligned} b_0 &= h_0 + 2 \Delta t \nu \left( \frac{1}{h_0 + h_1} + \frac{h_1^2 + 4 h_0 h_1 + 4 h_0^2}{\det} \right), \\ c_0 &= -2 \Delta t \nu \left( \frac{1}{h_0 + h_1} + \frac{h_0^2}{\det} \right), \\ \text{rhs}_0 &= (h u_0)^* + 2 \Delta t \nu u_b \frac{h_1^2 + 4 h_0 h_1 + 3 h_0^2}{\det}, \\ \det &= h_0 h_1 (8 \lambda_b + h_1) + h_0^2 (6 \lambda_b + 3 h_1) + 2 (h_1^2 \lambda_b + h_0^3), \end{aligned}$$

with  $a_k$ ,  $b_k$  and  $c_k$  the lower, principal and upper diagonals of  $\mathbf{M}$ . The resulting scheme is third-order accurate (everywhere) for a regular vertical spacing (identical  $h_k$ ) and can, for example, exactly represent the viscous terms for a parabolic velocity profile.

### 3.3 Vertical remapping

Lagrangian methods have the well-known drawback that as times passes discretisation elements become more and more distorted, up to a point where the results loose consistency. A solution is to periodically “remap” the solution onto a “clean” discretisation. The quality of this remapping then largely controls the overall quality of the method. In multiple dimensions this is not a simple operation, in particular when conservation of the remapped quantities is required [40]. In the one-dimensional case we are interested in, this is much simpler and can be done using for example Piecewise Polynomial Reconstruction [41]. In our implementation we have used the quadratic piecewise remapping of the PPR library of Engwirda and Kelley (2016, <https://github.com/dengwirda/PPR>).

The choice of the target remapped discretisation is entirely flexible and could be based on physical considerations (i.e. the wish to resolve certain features: e.g. boundary layers, material interfaces etc.) and/or approximation error analysis [42, 43]. For the moment, we have only applied a simple remapping to “ $\sigma$ -coordinates” i.e. equal layer thicknesses proportional to the local water depth. This remapping is applied at every timestep just after the advection step of the hydrostatic algorithm.

Note that a subset of the possible remappings can be interpreted as fluxes of the layer-integrated quantities through the layer interfaces (see [43] for an example). The associated “remapping” velocity is the time derivative of the (desired) vertical displacement of the layers. This is in fact a discrete representation of the vertical advection term required when considering the equations in a moving (vertical) reference frame, and is known as an Arbitrary Lagrangian–Eulerian (ALE) (vertical) discretisation. We prefer to avoid this terminology since it is more restrictive than the remapping concept and has historically caused significant confusion (due to an erroneous interpretation of the physical meaning of the remapping velocity: it has none in general). See also [31] for a recent discussion.

### 3.4 Adaptivity and parallelism

The numerical schemes are implemented within the Basilisk framework [1] which provides transparent quadtree adaptivity and OpenMP/MPI parallelism for Cartesian numerical schemes. A detailed description is beyond the scope of this paper but we refer the interested reader to the web site as well as [44, 45].

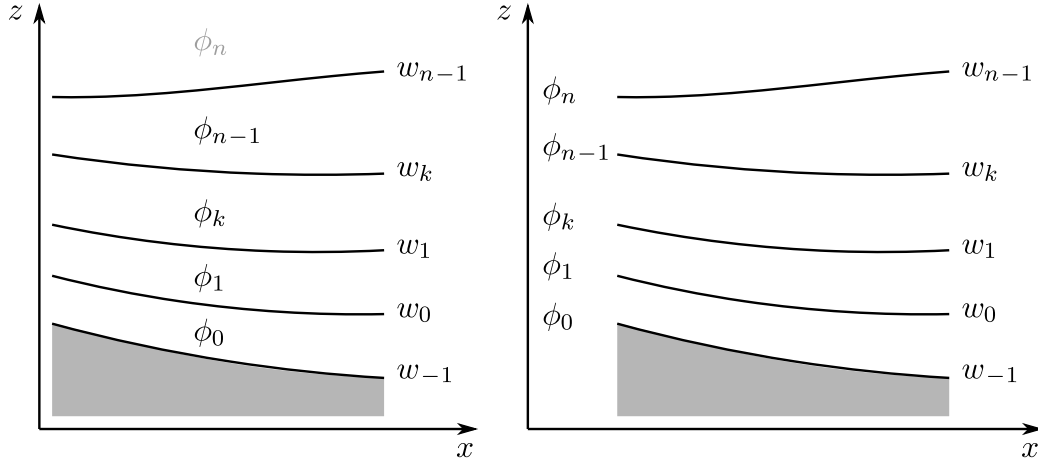
### 3.5 Generic dispersion analysis

The semi-discrete formulation (1), (2), (3) and (4) is well-suited for a generic study of the dispersive properties of various choices of vertical discretisations. Using standard linear perturbation analysis, we look for wave-like solutions of the form

$$\begin{aligned} h_k &= \bar{h}_k + h'_k e^{i(\hat{k}x - \omega t)}, \\ u_k &= u'_k e^{i(\hat{k}x - \omega t)}, \\ w_k &= w'_k e^{i(\hat{k}x - \omega t)}, \\ \phi_k &= \phi'_k e^{i(\hat{k}x - \omega t)}, \end{aligned}$$

with  $\omega$  the frequency,  $\hat{k}$  the wavenumber and  $h'_k$ ,  $u'_k$ ,  $w'_k$  and  $\phi_k$  vanishingly-small perturbations.

The vertical discretisation scheme essentially amounts to the choice of functional relationships between the layer interface values  $\phi_{k+1/2}$ ,  $w_{k+1/2}$ , ... and the ‘‘layer-averaged’’ values  $\phi_k$ ,  $w_k$ , ... (as for the single layer model in Remark 4 above).



**Figure 2.** Conventions used for the indices of vertically staggered fields, for the generic dispersion analysis. (left) Staggered vertical discretisation. (right) Keller box scheme.

#### 3.5.1 Staggered vertical discretisation

We first consider the simple choice

$$\begin{aligned} \phi_{k-1/2} &= \phi_k, \\ w_{k+1/2} &= w_k, \end{aligned}$$

which can be interpreted as a vertically-staggered discretisation of the velocity (i.e. pressure at the centre of the layer and vertical velocity on the layer interface, see Figure 2, left). The perturbations are then solutions of the linearised version of (1), (2), (3) and (4) (for a flat bathymetry)

$$-\omega h'_k + \bar{h}_k u'_k \hat{k} = 0, \quad (17)$$

$$-\bar{h}_k u'_k \omega = -g \hat{k} \bar{h}_k \sum h'_k - \bar{h}_k \hat{k} \phi'_k, \quad (18)$$

$$-\bar{h}_k w'_k \omega i = \phi'_k - \phi'_{k+1}, \quad (19)$$

$$\bar{h}_k u'_k \hat{k} i + w'_k - w'_{k-1} = 0 \quad (20)$$

The system must be closed with the (linearised) bottom impermeability condition

$$w'_{-1} = 0,$$

and a condition for the top pressure  $\phi_n$ . With this choice of variable staggering, the top pressure is above the free surface and we must use an approximation of the free-surface boundary condition  $\phi=0$ , such as the third-order accurate interpolation (valid for identical layer depths)

$$\phi_n = \frac{\phi_{n-2} - 6\phi_{n-1}}{3} \quad (21)$$

The dispersion relation can then be obtained by considering non-trivial solutions of (17), (18), (19) and (20) i.e. solutions for which the determinant of the system vanishes. For a single layer, this gives

$$\omega_1^2(\hat{k}) = 8g \frac{h_0 \hat{k}^2}{3h_0^2 \hat{k}^2 + 8} \quad (22)$$

(where we have dropped the  $-$  for convenience) and for two layers one finds

$$\omega_2^2(\hat{k}) = 8g \frac{h_0 h_1^2 \hat{k}^4 + (h_1 + h_0) \hat{k}^2}{3h_0^2 h_1^2 \hat{k}^4 + 3(3h_0^2 + h_0 h_1 + h_1^2) \hat{k}^2 + 8}$$

The calculation for an increasing number of layers becomes rapidly cumbersome but can be carried out using a symbolic algebra package. We give explicit formulae for a larger number of layers in Appendix B. The general form is the rational function

$$\omega_n^2(\hat{k}) = g \frac{\sum_{i=1}^n a_{2i}(h_k) \hat{k}^{2i}}{\sum_{i=0}^n b_{2i}(h_k) \hat{k}^{2i}},$$

with  $a_{2i}$  and  $b_{2i}$  polynomial functions of  $h_k$ .

The discrete dispersion relations above are an approximation of the exact linear dispersion relation for waves in arbitrary water depth

$$\omega^2(\hat{k}) = g \hat{k} \tanh(\hat{k} h)$$

It is clear that a larger number of layers should improve the approximation of the hyperbolic function since the degree of the rational function increases with the number of layers. The details of this convergence will matter in practice however, and will depend on the details of the discretisation. The analysis of this convergence is thus a good way of choosing between different potential schemes.

For example, the single-layer system approximates the hyperbolic function as (according to (22))

$$\tanh(x) \approx \frac{x}{\frac{3}{8}x^2 + 1}, \quad (23)$$

which can be compared with the Padé [1/2] approximant

$$\tanh(x) \approx \frac{x}{\frac{3}{9}x^2 + 1}$$

The choice of the second-order top pressure interpolation

$$\phi_n = -\phi_{n-1}, \quad (24)$$

instead of (21) gives a different discrete dispersion relation and the approximation

$$\tanh(x) \approx \frac{x}{\frac{1}{2}x^2 + 1},$$

which is obviously less accurate than (23).

Note also that for multi-layer systems, the choice of the layer thicknesses  $h_k$  may be optimised to obtain better approximations of the dispersion relation [33]. It is also easy to show that adding a “geometrical parameter” for each layer — which can be justified as modelling the relation between layer-interface values and layer-averaged values — for example the  $\alpha_k$  in the modified vertical momentum equation

$$\partial_t(h w)_k + \nabla \cdot (h w \mathbf{u})_k = -\alpha_k [\phi]_k,$$

can be used in combination with the layer thicknesses to get enough degrees of freedom to completely tune the discrete dispersion relation i.e. to obtain any rational function with a degree set by the number of layers. One could for example use this approach to exactly match the Padé approximants of the hyperbolic function [23]. We do not pursue this route however because it is likely to break the overall consistency of the scheme (for example its convergence properties) and an accurate linear dispersion relation is only one of the aspects we are interested in.

### 3.5.2 Keller box scheme

Stelling and Zijelma (2003) [28] noted that their  $z$ -coordinate implementation of the “Keller box scheme” [46] had much better dispersion characteristics than another implementation (similar to the one above but with a different treatment of boundary conditions). Zhu et al. (2014) [33] more recently derived an analytical dispersion relation for this scheme and proposed optimised layer thicknesses. We repeat briefly this analysis here, but using the simpler framework presented above.

The box scheme relies on a collocated discretisation of the vertical velocity  $w$  and non-hydrostatic pressure  $\phi$ , both defined on the layer interfaces (see Figure 2, right). The layer-averaged horizontal pressure gradient required in the horizontal momentum equation (2) is then obtained by averaging the corresponding layer-interface gradients. In a similar manner, the evolution equation (3) for the vertical momentum in each layer is written as the average of the evolution equations for the layer-interface momenta (see Section 3.6.2 for a detailed description). Using the notations above, the corresponding linearised perturbation system can be written

$$\begin{aligned} -\omega h'_k + \bar{h}_k u'_k \hat{k} &= 0, \\ -\bar{h}_k u'_k \omega &= -g \hat{k} \bar{h}_k \sum h'_k - \bar{h}_k \hat{k} \frac{\phi'_{k+1} + \phi'_k}{2}, \\ -\bar{h}_k \frac{w'_k + w'_{k-1}}{2} \omega i &= \phi'_k - \phi'_{k+1}, \\ \bar{h}_k u'_k \hat{k} i + w'_k - w'_{k-1} &= 0, \end{aligned}$$

with the (linearised) bottom impermeability condition

$$w'_{-1} = 0,$$

and the (exact) top pressure boundary condition

$$\phi'_n = 0$$

Computing the determinant then gives the dispersion relations (see Appendix B). For a single layer, the corresponding approximation is

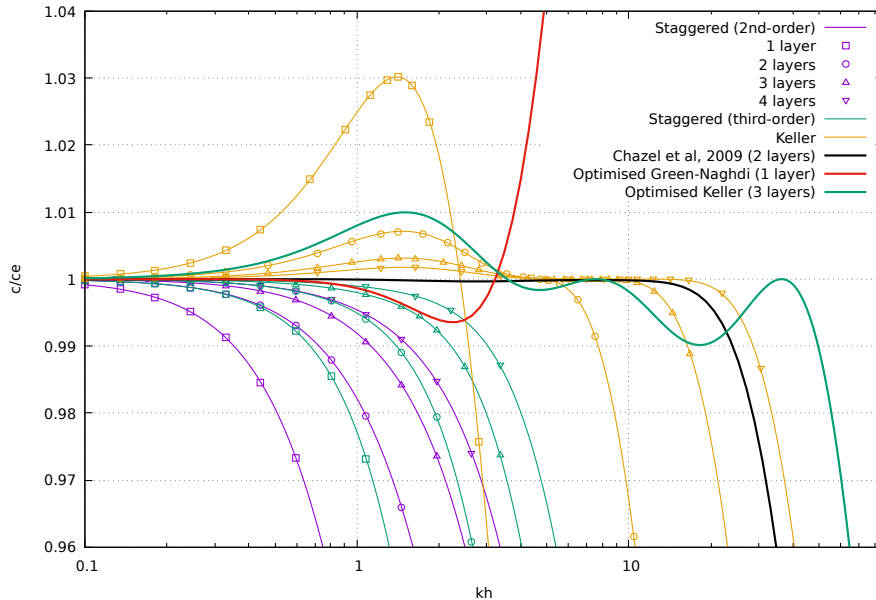
$$\tanh(x) \approx \frac{x}{\frac{1}{4}x^2 + 1} \quad (25)$$

which seems worse than (23) (using the Padé approximant as reference).

To better appreciate the difference between schemes, we display in Figure 3 the ratio of the discrete phase velocity  $c = \omega/\hat{k}$  to the exact Stokes phase velocity

$$c_e = \sqrt{\frac{g}{\hat{k}} \tanh(\hat{k}h)}$$

as functions of  $\hat{k}h$ . We have also included the dispersion relations of the extended Boussinesq two-layer model of Chazel et al. (2009) [23] and of the optimised one-parameter Serre–Green–Naghdi model [47].



**Figure 3.** Ratio of the discrete phase velocity to the exact phase velocity for a range of numerical schemes and number of layers.

The two-layer model of Chazel et al. (2009) has excellent dispersion properties, which is expected since Boussinesq models are designed based on their dispersion characteristics. The range of validity of the staggered schemes of section 3.5.1 and of the Keller box scheme both increase with an increasing number of layers, as expected. Using the third-order free-surface pressure condition (21) instead of the second-order condition (24) also significantly improves the range of validity of the staggered scheme. The range of validity of the Keller box scheme is however much broader than that of the staggered schemes. One can note in particular that the two-layers Keller scheme is superior to the single-parameter optimised Green-Naghdi model and that the three- and four-layers (uniform thicknesses) Keller schemes are competitive with the two-layer Boussinesq model of Chazel et al. (2009). Using an “optimal” layer thickness distribution of 68%–26.5%–5.5%, as in [33], gives a three-layer Keller model with less than 1% error for a very broad range of  $\hat{k}h$ .



It is also interesting to note that the one-layer Keller box scheme, with a dispersion relying on approximation (25), has a much broader range of validity than the one-layer, third-order, staggered scheme, even though (23) is closer to the Padé approximant. Of course, this increase in the range of validity comes at the expense of a lesser accuracy for small values of  $\hat{k}h$ .

### 3.6 Non-hydrostatic scheme

The non-hydrostatic extension is derived from the classical “projection method” [2, 3]. Note that when using Eulerian discretisations and in particular  $z$ -coordinates, the natural projection scheme is a straightforward extension of the staggered MAC or C-grid scheme, applied to the vertical coordinate [26]. This corresponds to the linearised scheme studied in section 3.5.1 and Figure 2 (left). For general coordinates, including the Lagrangian formulation we use here, the slope of the layers must be taken into account.

#### 3.6.1 Staggered vertical discretisation

To illustrate the principle of the method, we start with the simpler staggered vertical discretisation of section 3.5.1. Using the semi-discrete notation introduced above, the non-hydrostatic scheme can be derived as the following extension of the hydrostatic scheme:

1. Compute  $h_k^{n+1}$ ,  $(hu)_k^\star$ ,  $(hw)_k^\star$  using  $h_k^n$ ,  $u_k^n$ ,  $w_k^n$  and the hydrostatic scheme described in the previous section. Note that the vertical momentum ( $hw$ ) has been added to the list of advected tracers. The  $\star$  denotes intermediate values which do not include the hydrostatic and non-hydrostatic pressure terms.
2. Project the  $(hu_k^\star, hw_k^\star)$  vector using  $\phi_k$  so that the incompressibility condition (4) is verified. Following (2) and (3), the momentum at  $n+1$  is obtained as

$$(hu)_k^{n+1} = (hu)_k^\star - \Delta t (\partial_x(h\phi) - [\phi \partial_x z])_k, \quad (26)$$

$$(hw)_k^{n+1} = (hw)_k^\star - \Delta t [\phi]_k. \quad (27)$$

Replacing the corresponding expressions in (4) then gives the Poisson-like equation

$$\left[ w^\star - \Delta t \frac{[\phi]}{h} - \left( u^\star - \Delta t \frac{\partial_x(h\phi) - [\phi \partial_x z]}{h} \right) \partial_x z \right]_k = 0,$$

with  $u_k^\star$  computed using (13) and  $w_k^\star = (hw)_k^\star / h_k^{n+1}$ . Re-arranging the terms, this can be written as the (unknown) l.h.s. and (known) r.h.s

$$\left[ \frac{[\phi]}{h} - \left( \frac{\partial_x(h\phi) - [\phi \partial_x z]}{h} \right) \partial_x z \right]_k = \frac{\partial_x(hu^\star)_k + [w^\star - u^\star \partial_x z]_k}{\Delta t}, \quad (28)$$

where the terms in colour depend on the slope of the layers. If these terms are neglected, the standard structure of a discrete Poisson problem is recovered (the first term in the l.h.s. is the horizontal second derivative, the second term is the vertical second derivative and the r.h.s. is the divergence of the intermediate velocity field).

3. Use (26) and (27) to compute the final values of  $(hu)_k^{n+1}$  and  $(hw)_k^{n+1}$ .

While the pressure gradient term in (27) is clearly defined (due to the vertically staggered discretisation), the discretisation of the horizontal pressure gradient in (26) must be described. The classical projection scheme uses the MAC or C-grid staggered velocity/pressure arrangement to avoid velocity/pressure decoupling. On the other hand, as discussed previously, the natural discretisation of the (hyperbolic) hydrostatic system of conservation laws relies on a collocated discretisation of the velocity components (i.e. Arakawa A- or B-grid). We solve this problem using collocated pressure and horizontal velocity/momentum components and an ‘‘approximate projection’’ method [48], which matches the computation of the hydrostatic pressure gradient in (11) and (14). The non-hydrostatic pressure gradient is first added to the staggered acceleration as

$$a_{i+1/2,k} \leftarrow a_{i+1/2,k} - 2 \frac{h_{i+1,k}^{n+1} \phi_{i+1,k} - h_{i,k}^{n+1} \phi_{i,k} - [\phi \partial_x z]_{i,k}}{h_{i+1,k}^{n+1} + h_{i,k}^{n+1}},$$

the updated acceleration is then used to compute  $u_{i+1/2,k}^{n+1}$  and  $(h u)_{i,k}^{n+1}$  according to (13) and (14).

As for any incompressible flow solver, the overall efficiency of the scheme depends on the linear solver used to invert the Poisson-like system (28). We use the generic geometric multigrid solver built into Basilisk, which also works in parallel and on quadtree-adaptive grids. A more detailed description of this solver is given in [49] where it is used to invert the pseudo-elliptic Green–Naghdi operator. To improve robustness and convergence rate, column relaxation is used in the vertical direction: in the discretised version of the l.h.s. of (28), the off-diagonal values of  $\phi$  in the horizontal directions,  $\phi_{i+1,k}$  and  $\phi_{i-1,k}$  are supposed to be known, which then gives a linear, tri-diagonal system of equations for  $\phi_{i,k}$  which can be inverted efficiently using Thomas algorithm. This vertical column relaxation is combined with multigrid acceleration to obtain the solution to within a specified tolerance. The value of this tolerance is also important for efficiency. A good balance between accuracy and efficiency is obtained using the following criterion

$$\frac{|h_k (\nabla \cdot (h \mathbf{u})_k + [w - \mathbf{u} \cdot \nabla z]_k)|_\infty}{|h_k|_1^2} \Delta t < \epsilon, \quad (29)$$

where the max norm  $\|\cdot\|_\infty$  and average  $\|\cdot\|_1$  are computed over all layers and all elements. The tolerance  $\epsilon$  can be interpreted as the maximum relative change of volume allowed for any discretisation element during one timestep. Unless otherwise indicated it will be set to  $10^{-3}$  in the numerical applications below.

This scheme works well in practice but is reasonably complicated to implement, in particular due to the slope terms in (28). Its extension to the more complicated Keller box scheme is thus not straightforward. Thankfully all the slope terms in (28) are not equally important. The terms in blue and green in the l.h.s. are in particular  $O(\Delta t)$  less important than the term in red in the r.h.s. Indeed neglecting them would only contribute to increasing the existing  $O(\Delta t)$  time-splitting error of the projection scheme. Furthermore the blue term depends on the vertical difference of the layer slopes (i.e. a second-derivative), while the green terms depend on even higher-order derivatives.

To test these assumptions, we compared the results obtained using the full scheme and a simplified scheme neglecting the slope terms (green and blue) in (28), for all the test cases presented in section 4 and found negligible differences. As expected, ignoring the blue term has the largest (but still negligible) impact. Note that all other slope terms must be retained, in particular in the r.h.s. of (28) and in (26).

It is important to note that this approximation is possible only because of the choice of a Lagrangian vertical discretisation. In the case of an Eulerian or quasi-Eulerian discretisation the vertical velocity is used to transport tracers in the vertical direction. Volume, mass and tracer conservation at the discrete level thus rely on verifying *exactly* the discrete non-divergence condition. The corresponding discrete Laplacian operator, used in the l.h.s. of the Poisson equation, must then be the exact composition of the discrete divergence and discrete gradient operators and not just an approximation.

For the Lagrangian vertical discretisation on the other hand, volume, mass and tracer conservation do not depend on the vertical velocity field (which is not used for transport) but is guaranteed by the consistent advection of the layer thicknesses and of the tracers. Using an “approximate vertical projection” may thus affect the accuracy of the non-hydrostatic terms but not the conservation properties.

### 3.6.2 Keller box scheme

We have seen in section 3.5 that the staggered vertical scheme is not competitive with extended Boussinesq models, even when a (prohibitively) large number of layers is used. This motivates the generalisation of the non-hydrostatic algorithm to the Keller scheme. The horizontal momentum and vertical velocity are now given by

$$\begin{aligned} (h u)_k^{n+1} &= (h u)_k^* - \Delta t (\partial_x(h \phi) - [\phi \partial_x z])_k, \\ \left( \frac{w_{k-1/2} + w_{k+1/2}}{2} \right)^{n+1} &= w_k^* - \Delta t \frac{[\phi]_k}{h_k^{n+1}}, \end{aligned}$$

and the pressure gradient in the middle of the layer is approximated as

$$\partial_x(h \phi)_k = \frac{\partial_x(h_k \phi_{k-1/2}) + \partial_x(h_k \phi_{k+1/2})}{2} \quad (30)$$

Using the incompressibility condition (4), the pressures  $\phi_{k+1/2}$  and velocities  $w_{k+1/2}$  are then solutions of the system

$$\begin{aligned} \frac{w_{k-1/2} + w_{k+1/2}}{2} &= w_k^* - \Delta t \frac{\phi_{k+1/2} - \phi_{k-1/2}}{h_k}, \\ \partial_x(h u)_k^* - [u^* \partial_x z]_k - \Delta t \frac{\partial_x \partial_x(h_k \phi_{k-1/2}) + \partial_x \partial_x(h_k \phi_{k+1/2})}{2} + [w]_k &= 0, \end{aligned} \quad (31)$$

where we have neglected the terms equivalent to the [green](#) and [blue](#) terms in (28), and dropped the  $n + 1$  indices for convenience. To obtain a system of equations for  $\phi$ , we need to express  $[w]_k$  as a function of  $\phi$ . This is readily done using the recursion relation (derived from (31))

$$w_{k+1/2} = 2 \left( w_k^* - \Delta t \frac{\phi_{k+1/2} - \phi_{k-1/2}}{h_k} \right) - w_{k-1/2},$$

which gives

$$w_{k+1/2} = 2 \sum_{l=0}^k (-1)^{k+l} \left( w_l^* - \Delta t \frac{\phi_{l+1/2} - \phi_{l-1/2}}{h_l} \right),$$

and

$$[w]_k = 2 \left( w_k^* - \Delta t \frac{\phi_{k+1/2} - \phi_{k-1/2}}{h_k} \right) + 4 \sum_{l=0}^{k-1} (-1)^{k+l} \left( w_l^* - \Delta t \frac{\phi_{l+1/2} - \phi_{l-1/2}}{h_l} \right)$$

The pressure equation is then

$$\partial_x(hu)_k^* - [u^* \partial_x z]_k - \Delta t \frac{\partial_x \partial_x (h_k \phi_{k-1/2}) + \partial_x \partial_x (h_k \phi_{k+1/2})}{2} + 2 \left( w_k^* - \Delta t \frac{\phi_{k+1/2} - \phi_{k-1/2}}{h_k} \right) + 4 \sum_{l=0}^{k-1} (-1)^{k+l} \left( w_l^* - \Delta t \frac{\phi_{l+1/2} - \phi_{l-1/2}}{h_l} \right) = 0$$

or equivalently the l.h.s. and r.h.s.

$$\begin{aligned} & \partial_x \partial_x (h_k \phi_{k-1/2}) + \partial_x \partial_x (h_k \phi_{k+1/2}) + \\ & 4 \frac{\phi_{k+1/2} - \phi_{k-1/2}}{h_k} + 8 \sum_{l=0}^{k-1} (-1)^{k+l} \frac{\phi_{l+1/2} - \phi_{l-1/2}}{h_l} = \\ & \frac{2}{\Delta t} \left( \partial_x (hu)_k^* - [u^* \partial_x z]_k + 2w_k^* + 4 \sum_{l=0}^{k-1} (-1)^{k+l} w_l^* \right) \end{aligned} \quad (32)$$

We see that the value of the pressure in a given layer is now dependent on the values in all the layers below it. Compared to the staggered scheme derived previously (equation (28)), the structure of the linear problem has thus changed from a sparse, multi-diagonal system to a dense, block-Hessenberg system. This has important implications for the expected performance of the linear solver and lead Stelling & Zijelma [29] to propose an approximation of their quasi-Eulerian box scheme which neglects this vertical coupling. We tried their approach without success. In particular, all our implementations of their scheme were unstable for a number of layers larger than five. Similar unstable results were obtained when using their open-source implementation SWASH [50].

It turns out that system (32) can be solved efficiently using a multigrid-accelerated column relaxation technique similar to that used to solve (28). Assuming known values of  $\phi_{i+1,k+1/2}$  and  $\phi_{i-1,k+1/2}$  leads to an Hessenberg system for  $\phi_{i,k+1/2}$  in each column  $i$ . An Hessenberg matrix is the sum of a triangular and a tri-diagonal matrix. The resulting system can be solved efficiently using the algorithm of Henry (1994) [51]. Note however that the cost of solution for each column relaxation will scale like  $O(n^2)$  rather than  $O(n)$  for a tridiagonal system. We will demonstrate later that this does not preclude efficient solutions.

The resulting algorithm can be summarised as follows:

1. Compute  $h_k^{n+1}$ ,  $(hu)_k^*$ ,  $(hw)_k^*$  using  $h_k^n$ ,  $u_k^n$ ,  $w_k^n$  and the hydrostatic scheme (without pressure gradients).
2. Compute  $\phi_{k+1/2}$  using multigrid-accelerated Hessenberg column-relaxations to solve (32).
3. Update the staggered accelerations and velocities

$$\begin{aligned} \alpha_{i+1/2,k} & \leftarrow 2 \frac{\partial_x (h\phi)_{i+1/2,k} - [\phi \partial_x z]_{i+1/2,k}}{h_{i+1,k}^{n+1} + h_{i,k}^{n+1}} \\ a_{i+1/2,k} & \leftarrow a_{i+1/2,k} - \alpha_{i+1/2,k} \\ u_{i+1/2,k} & \leftarrow u_{i+1/2,k} - \alpha_{i+1/2,k} \Delta t \end{aligned}$$

where  $\partial_x (h\phi)_{i+1/2,k}$  is approximated using (30).

4. Compute  $(hu)_{i,k}^{n+1}$  and  $(hw)_{i,k}^{n+1}$  using (14) and (27).

The corresponding documented code is available on the Basilisk web site (<http://basilisk.fr/src/layered/nh.h>).

### 3.6.3 Stability of the non-hydrostatic scheme

Since we use explicit timestepping, we can expect the timestep to be limited according to

$$\Delta t < \frac{\Delta}{|\mathbf{u}| + c},$$

with  $c$  the phase speed of gravity waves. Note that  $|\mathbf{u}|$  does not include the vertical velocity component since vertical advection is Lagrangian and thus not subject to a CFL constraint.

The phase speed should take into account dispersion and thus be  $\sqrt{g/\hat{k} \tanh(\hat{k} H)}$  rather than just  $\sqrt{gH}$  (as for the hydrostatic case). The wavenumber  $\hat{k}$  can be taken as the inverse of the smallest possible wavelength i.e.  $1/\Delta$ . Re-arranging the terms then gives the condition

$$\Delta t < \frac{\Delta}{|\mathbf{u}| + \sqrt{gH} \sqrt{(\Delta/H) \tanh(H/\Delta)}} \quad (33)$$

The difference with the hydrostatic case can be better understood using the approximation

$$\sqrt{(\Delta/H) \tanh(H/\Delta)} \approx \begin{cases} 1 & \text{if } \Delta \geq H \\ \sqrt{\Delta/H} & \text{if } \Delta < H \end{cases}$$

which shows that the increase in the timestep is proportional to  $\sqrt{H/\Delta}$  whenever the mesh size is smaller than the water depth. We will see later that this relaxed stability condition can lead to large gains for practical cases.

Note that this condition is based on the exact dispersion relation, not its numerical approximation (as displayed on Figure 3). We can expect the schemes for which the exact dispersion relation is an (approximate) upper bound to respect this condition: that is all schemes on Figure 3 with the exception of the optimised Green–Naghdi model.

### 3.6.4 Wave breaking

Since  $\eta(\mathbf{x}, t)$  describes the free surface as the graph of a (uni-valued) function, it cannot represent the overturning motion characteristic of breaking waves. This is a classical issue for Boussinesq models of wave motion [52]. In the case of the Saint-Venant equations, the corresponding limiting case is the formation of an hydraulic jump (i.e. a shock-like solution) which is known to be a remarkably good model of breaking (given its simplicity) [53]. The classical approach in Boussinesq models is thus to turn off dispersive terms when breaking is detected [49]. This approach does not seem to generalise easily to the present model. We found instead that the somewhat *ad hoc* following recipe works well.

Breaking is characterised both by steep free-surface slopes and large vertical velocities. Dissipation, linked to the unresolved turbulent motions occurring during breaking, can be introduced by limiting the maximum vertical velocity. This is done after application of the vertical non-hydrostatic pressure gradient (equation (27)) using

$$w_k^{n+1} \leftarrow \text{sign}(w_k^{n+1}) \min(|w_k^{n+1}|, b \sqrt{g|H|_\infty}), \quad (34)$$

where  $\sqrt{g|H|_\infty}$  is assumed to be the characteristic horizontal velocity scale of the breaking wave and the *breaking parameter*  $b$  is a number smaller than one. In the case of strong breaking (see the example in section 4.7) this is not sufficient to guarantee the stability of the non-hydrostatic solver and the following slope-limiting is added to  $\partial_x z$  in equations (26) and (32)

$$\partial_x z \leftarrow \begin{cases} \partial_x z & \text{if } |\partial_x z| < 1 \\ \text{sign}(\partial_x z) & \text{otherwise} \end{cases} \quad (35)$$

### 3.6.5 Wetting and drying

Although wetting and drying, and the associated “coastline representation”, has often been considered a difficult issue in ocean models, it is now well-known that the Saint-Venant equations are physically and mathematically well-posed even in the limit of vanishing water depth [36]. The idea of a “coastline” separating dry from wet (or “topography” from “bathymetry”) thus appears to be a misleading concept, as demonstrated in particular by tsunami/flood models based on the Saint-Venant equations [44, 54]. The same approach is taken here. The only adaptations specific to the wet/dry transition are the “wet condition” (12) and the solution of the Poisson equation (28) or (32) which is first multiplied by  $h_k$  to avoid any division by zero.

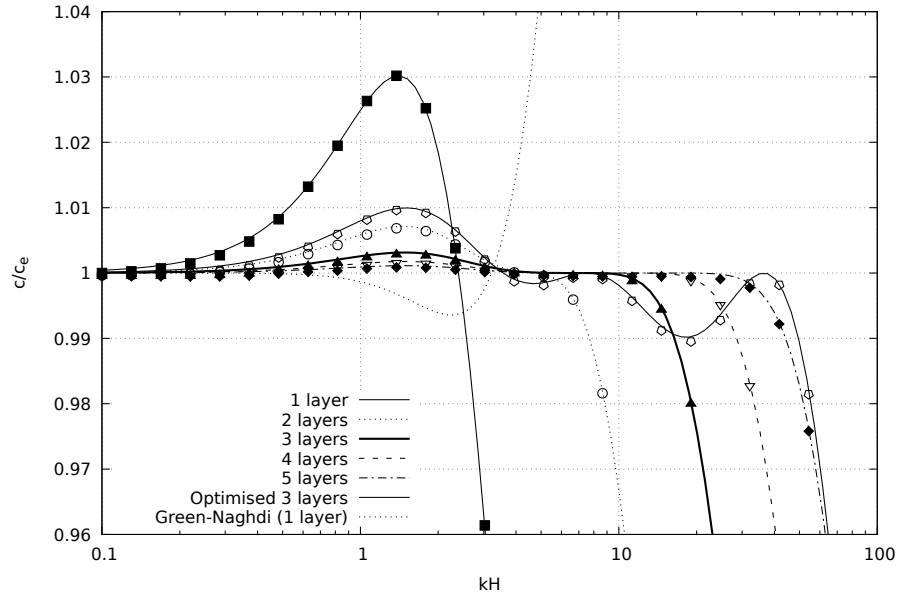
## 4 Numerical results

We present a range of numerical results which illustrate the generality and efficiency of the model. When not otherwise indicated, results have been obtained using the non-hydrostatic Keller box scheme and a timestep given by the stability condition (33). The layer-slope limiting (35) is always turned on but the vertical-velocity limiting (34) is turned off by default. The performance indicated were obtained on an 8 cores Intel® Core™ i7-8550U CPU @ 1.80GHz with 8 GB of memory. The links to the source codes sufficient to reproduce all the results and figures are given in the captions. Note also that other tests and examples are available at <http://basilisk.fr/src/layered/hydro.h#usage>

### 4.1 Dispersion relation

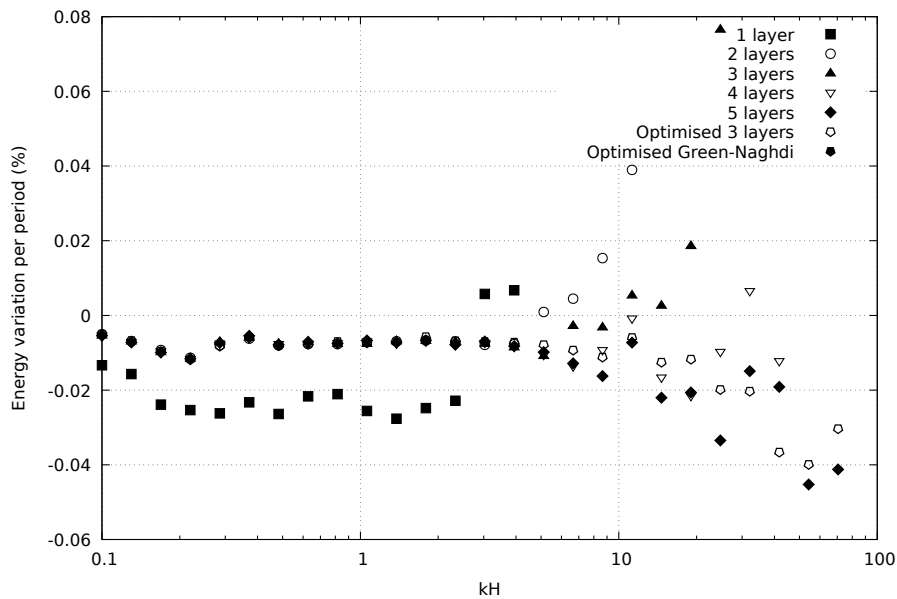
This first case verifies that the numerical dispersion relation follows the linear analysis of section 3.5 and Figure 3. A standing wave is initialised in a periodic domain and its frequency is measured using the zero-crossings of the wave amplitude over ten periods of oscillation. To ensure spatially-converged results, the wavelength is resolved using 128 grid points. The total water depth  $H$  is varied to obtain results for a broad range of  $\hat{k}H$ . The convergence criterion (29) is decreased to  $\epsilon = 10^{-6}$ . This is necessary to obtain accurate results at large  $\hat{k}H$ . Figure 4 illustrates the results obtained for a range of numerical schemes and a varying number of layers. The Green–Naghdi results were obtained using the numerical scheme described in [49]. Close agreement is obtained with the theoretical dispersion relations.

All results were obtained using a timestep given by the stability condition (33), with the exception of the Green–Naghdi model which proved unstable, as expected, and had to use the more restrictive hydrostatic condition (15). Note that for the maximum value of  $\hat{k}H \approx 60$  displayed on the figure and for 128 grid points per wavelength, the ratio of the non-hydrostatic timestep to hydrostatic timestep is approximately  $\sqrt{H/\Delta} = \sqrt{128 \hat{k}H/2\pi} \approx 35$ .



**Figure 4.** Dispersion relation. The lines are the analytical solutions and the symbols the numerical results (<http://basilisk.fr/src/test/dispersion.c>).

The same test case is used to estimate the numerical total energy variation per period, computed as the sum of the discrete potential and kinetic energies. The results are displayed in Figure 5. All schemes appear to conserve energy very well.



**Figure 5.** Relative variation in total energy per oscillation period (<http://basilisk.fr/src/test/dispersion.c>).



### 4.1.1 Profile of vertical velocity

The theoretical vertical profile of the vertical velocity component under a linear wave is given by Stokes' theory as [23]

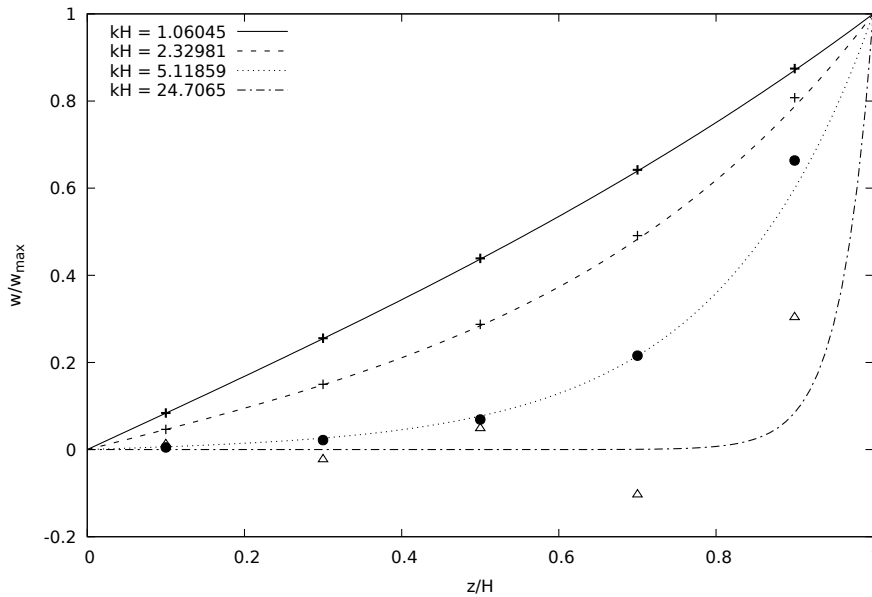
$$w_s(z) = \frac{A g k}{\omega} \frac{\sinh(k z)}{\cosh(k H)} \sin(\omega t - k x),$$

with  $A$  the amplitude of free-surface displacement. It reaches its maximum for multiples of  $T/4$  with  $T$  the Stokes wave period. Figure 6 displays the normalised profiles of vertical velocity obtained numerically at  $t = 37 T/4$ , for different values of  $k H$ . As the wave shortens, its velocity field becomes shallower and thus harder to resolve with a fixed number of equidistant layers.

It is interesting to note that the behaviour of the Keller box scheme used here is quite different from that of the staggered scheme (not shown). As illustrated in Figure 3, the staggered scheme would have much larger dispersion errors. This would lead to large phase errors at  $t = 37 T/4$  and it would not be possible to compare the velocity profiles with linear theory, unless a very large number of layers was used.

Remarkably, with the Keller scheme the phase is accurate, even though the number of layers and their distribution are insufficient to estimate an accurate vertical velocity profile, as illustrated for  $k H = 24.7065$  in Figure 6 (triangular symbols). This profile also suffers from a clear grid-scale oscillation which is probably due to the vertical collocation of pressure and vertical velocity. This could be a concern but has not caused problems in any of the cases we present. Furthermore, the amplitude of these spurious oscillations vanishes with increasing vertical resolution.

As expected, these grid-scale oscillations are absent when using the vertically-staggered scheme, however the velocity profile extends much deeper than predicted by theory, which also explains the corresponding dispersion/phase error.

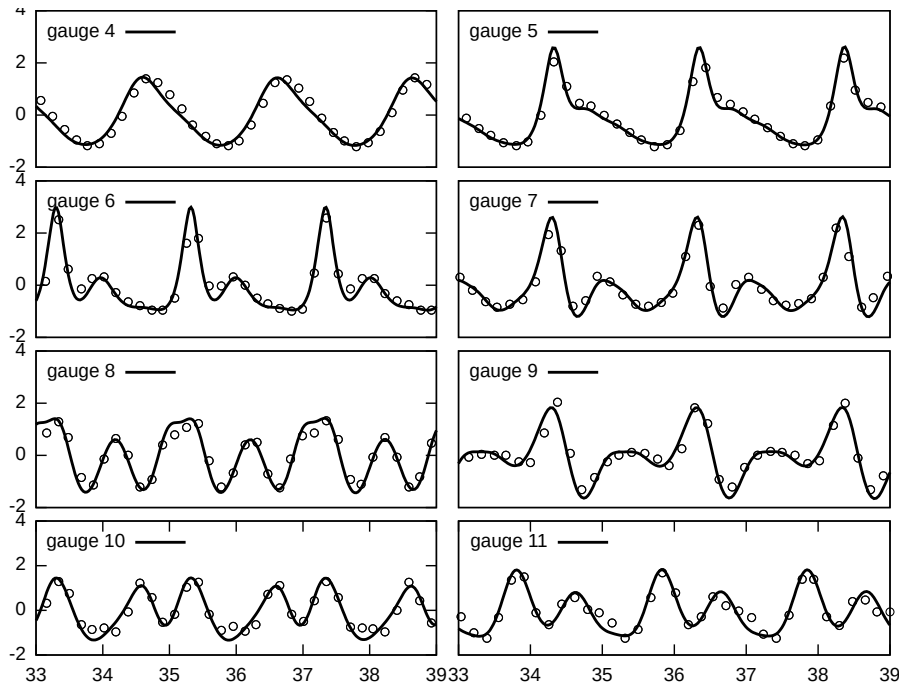


**Figure 6.** Normalised profiles of the vertical component of velocity under a linear wave, for different values of  $k H$ . The numerical results (symbols) are obtained using the Keller box scheme with five layers (<http://basilisk.fr/src/test/dispersion.c>).

## 4.2 Sinusoidal wave propagation over a bar

We next consider a classical test case for the Boussinesq models used in coastal engineering. Beji and Battjes [55] and Luth et al. [56] studied experimentally the transformation of sinusoidal waves propagating over a submerged bar (or reef). This is a good test case for dispersive models as higher harmonics are non-linearly generated and released with phase shifts corresponding to the dispersion relation. Note that wave breaking also occurs experimentally on top of the submerged bar. See section 4.2 of [34] for a detailed description of the setup and results using a one-layer model.

The results displayed on Figure 7 were obtained using the two-layer Keller scheme with a breaking parameter  $b = 0.1$  (see (34)) and 2048 grid points (for a domain length of 50 metres). The agreement with measurements is excellent, in particular for gauge eleven which is the farthest from the reef and thus most affected by dispersion errors. Using only a single layer (not shown), the results are comparable up to gauge nine but dispersion errors clearly affect the results for gauges ten and eleven. The results of Figure 7 can be compared to Figure 4 of [34] and are of comparable quality up to gauge eight, but much better for the last three gauges.

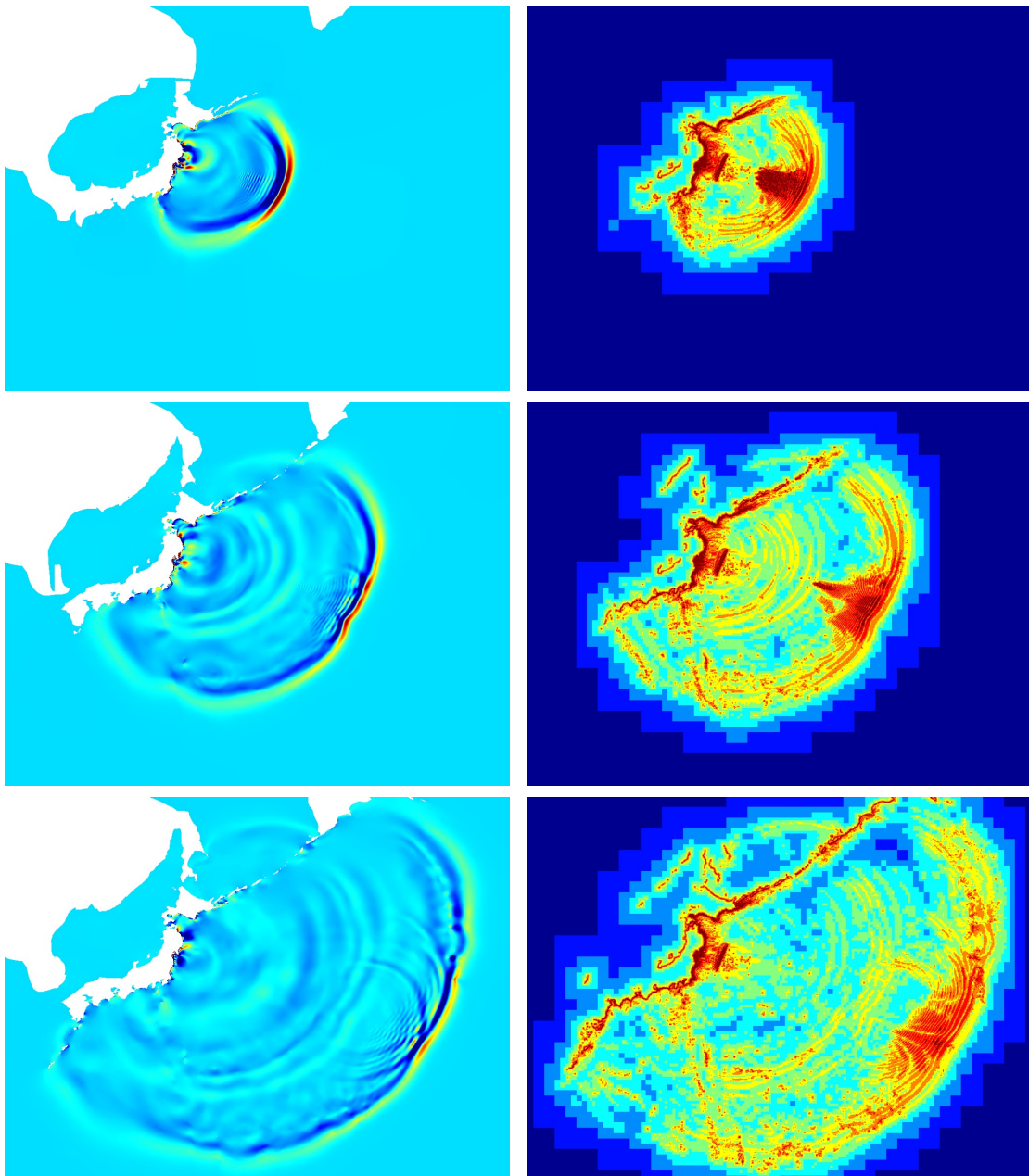


**Figure 7.** Computed (solid line) and measured (symbols) free-surface elevations for the wave propagation over a bar obtained experimentally by Beji and Battjes [55]. The numerical results are obtained with two layers (<http://basilisk.fr/src/test/bar-ml.c>).

## 4.3 Dispersive tsunami

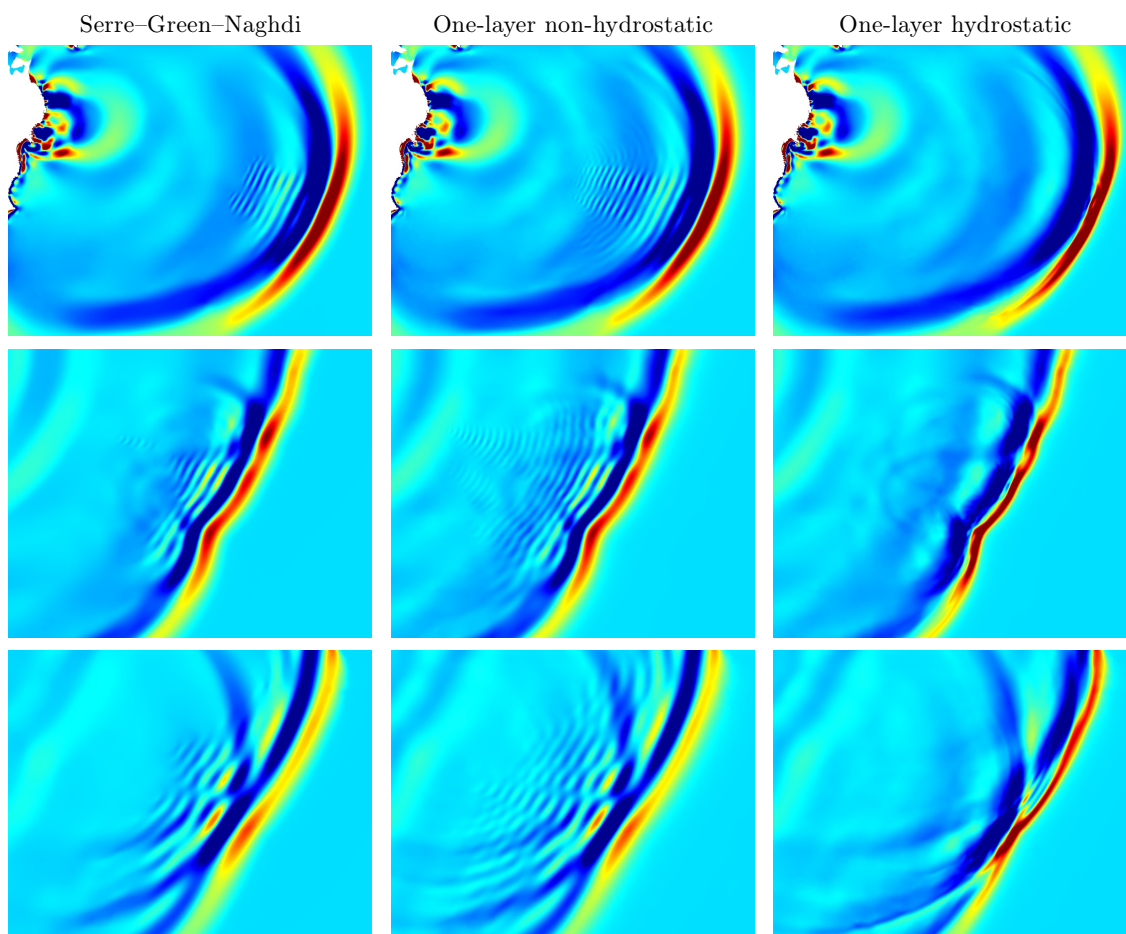
While tsunamis are often modelled using the non-dispersive Saint-Venant equations, it is now well-known that dispersive effects are often not negligible [57, 58, 49]. To illustrate the applicability and robustness of the layered model to such a case, we consider the Japanese

Tohoku tsunami of 11th March 2011, which we already studied using both the Green–Naghdi [49] and Saint-Venant [54] models. The setup is identical to these previous papers where a more detailed description and discussion can be found. The results obtained using the one-layer Keller scheme are displayed on Figure 8. This simulation combines wetting and drying, dispersion, quadtree-adaptivity and complex bathymetry (reconstructed dynamically using the algorithm described in the Appendix of [44]). The results of Figure 8 can be compared with Figure 12 of Popinet (2015) [49].



**Figure 8.** Tohoku tsunami. Evolution of the free-surface elevation (left column) and adaptive grid (right column). The elevation scale ranges from  $-1$  metre (dark blue) to  $+2$  metres (dark red). The level of refinement ranges from 5 ( $\approx 250$  km, dark blue) to 13 ( $\approx 1$  km, dark red). The times are from top to bottom: 1, 2 and 3 hours after the fault rupture (<http://basilisk.fr/src/examples/tohoku.c>).

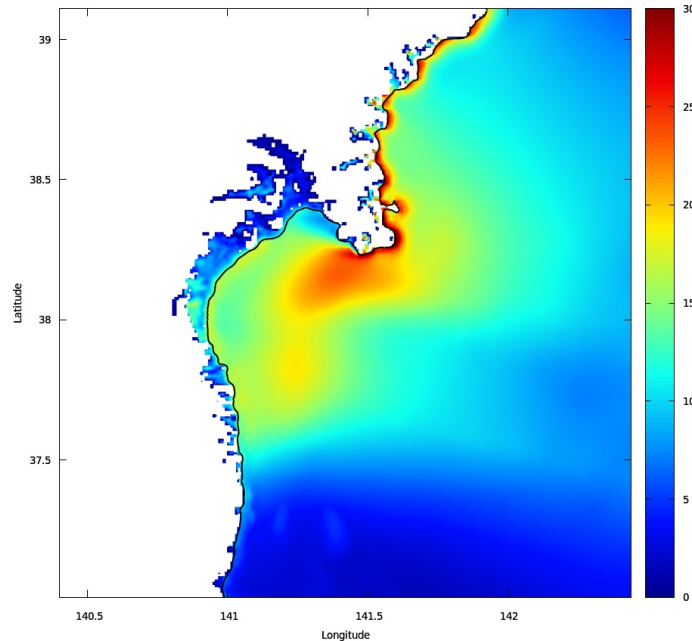
The importance of dispersion on long-distance wave propagation is highlighted on Figure 9 where three different models are compared. The Serre–Green–Naghdi results (obtained using the solver described in [49]) are clearly close to those obtained using the one-layer non-hydrostatic model, the most significant difference being the longer “tail” of dispersive waves obtained using the one-layer model. This reflects the lower numerical dissipation of this scheme, compared to that of the second-order MUSCL/Riemann scheme used in [49]. Besides the absence of a dispersive tail, the hydrostatic scheme also produces a much steeper leading wave: a result consistent with previous studies and the expectation that the lack of dispersive terms leads to over-steepening of wave fronts (and eventually the formation of hydraulic jumps).



**Figure 9.** Close-up views of the leading wave front for the times of Figure 8, obtained using different models. Left-column: Serre–Green–Naghdi; Central-column: non-hydrostatic one-layer model; Right-column: hydrostatic one-layer model (<http://basilisk.fr/src/examples/tohoku.c>).

The ability of the model to deal robustly with wetting and drying is illustrated on Figure 10 which shows a detail of the maximum flood level. The area displayed is approximately  $140 \times 200$  km and includes the extensively-flooded Sendai plain as well as extremely high runup on the Oshika peninsula. These results are consistent with both satellite data and

previous simulations (see e.g. Figure 15 of [54]), although the resolution here is significantly lower (a maximum of 1 km compared to 250 metres)



**Figure 10.** Detail of flooding in the Ofunato, Miyagi and Fukushima districts. The black line indicates the normal coastline. The colorscale is the maximum flood level (maximum water elevation above normal) in meters (<http://basilisk.fr/src/examples/tohoku.c>).

Finally, Table 1 gives the runtimes of the different models, corresponding to six and a half hours (390 minutes) of physical time. Since the mesh is adaptive, the number of grid points indicated is the average over the entire simulation time. Eight cores were used with OpenMP parallelism. While the highest computation speed is obtained with the hydrostatic solver, the shortest computation time is given by the non-hydrostatic solver. This reflects the significant gain in number of timesteps due to the relaxed stability condition (33). The Serre–Green–Naghdi solver [49] is substantially slower, even though it uses less grid points (due to increased numerical dissipation of small waves).

Model	# Timesteps	# Points	Runtime	Speed
Hydrostatic one-layer	17 532	500 666	318	460 000
Non-hydrostatic one-layer	7945	503 230	258	258 000
Serre–Green–Naghdi	17 168	330 161	448	211 000

**Table 1.** Runtimes (minutes) for the dispersive tsunami case. The speeds are given in points  $\times$  timesteps/sec. The number of grid points is a time average.

#### 4.4 Large-amplitude standing wave

After this validation of the numerical scheme in the context of dispersive waves, resolved with only a few layers, we now move to applications illustrating the consistency of the

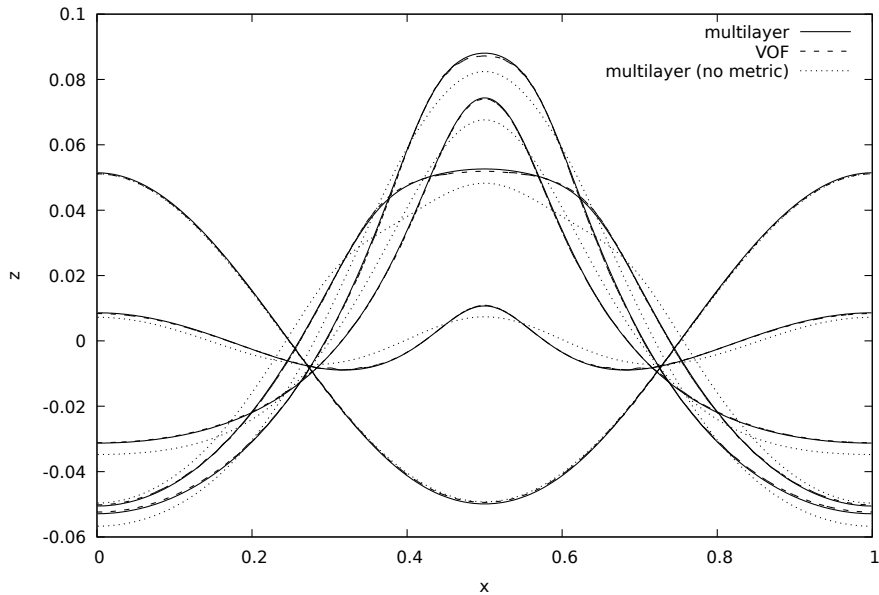


model for the incompressible Euler/Navier-Stokes equations with a free-surface. A simple initial condition is given by a large amplitude sinusoidal perturbation of the free-surface

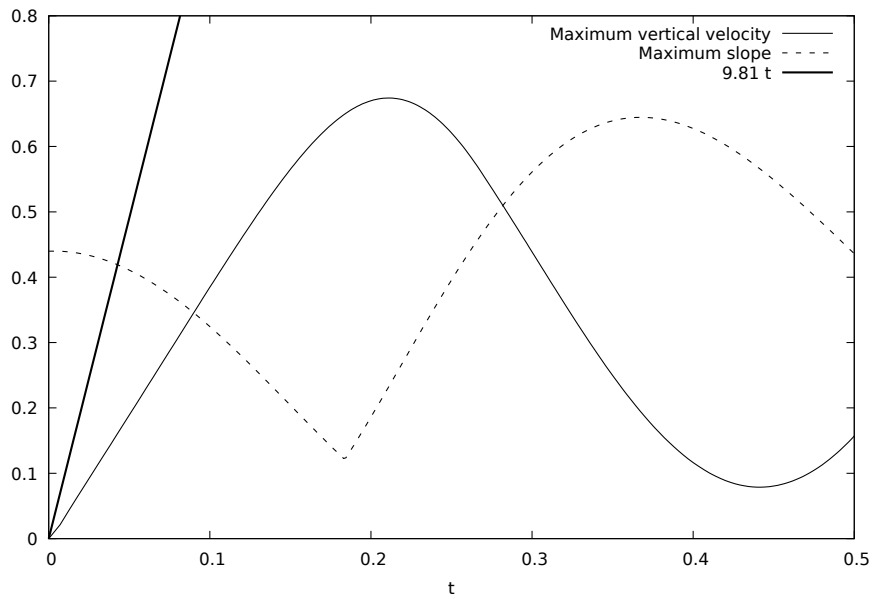
$$\eta(x, 0) = 0.5 + 0.07 \cos(2 \pi x),$$

and zero initial velocity, in a one meter wide domain with free-slip side walls. The acceleration of gravity is set to  $9.81 \text{ m/s}^2$ . The horizontal dimension is discretised with 512 grid points and 20 layers are used in the vertical. The evolution of the free surface is illustrated in Figure 11. Strong non-linearity and vertical accelerations give rise to a vertical “jet” of liquid which then falls back due to gravity. The evolution with time of the maximum vertical velocity and free-surface slope is illustrated in Figure 12. The vertical acceleration is of the order of the acceleration of gravity, i.e. the flow is strongly non-hydrostatic, and the maximum slope is of order one. Clearly in this case the metric/slope terms cannot be neglected, which is confirmed when comparing the results obtained when neglecting the slope terms  $\partial_x z$  in expressions (26) and (32) (dotted line on Figure 11).

Since analytical solutions are not available for this non-linear problem, a calculation is also done using a completely different solver: the incompressible, two-phase, Volume-Of-Fluid, Euler/Navier–Stokes solver described in [59]. This scheme, also implemented in Basilisk, uses a fixed regular Cartesian grid and a VOF description of the interface separating the two phases. The equations of motion are solved both in the dense phase below the interface and the light phase above the interface. The density ratio is set to  $1/1000$  which is small enough to ensure that the motion of the light phase does not affect the results. The  $1 \times 1$  metre domain is resolved using  $256 \times 256$  grid points. The corresponding interface evolution is displayed in Figure 11 (dashed lines). The agreement with the layered model is clearly excellent at all times.



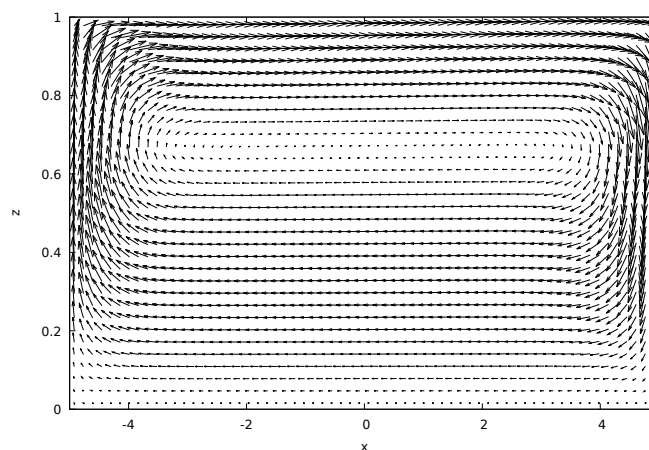
**Figure 11.** Free-surface evolution. The free-surface is displayed at times  $t = 0.1$  sec to  $0.5$  sec in  $0.1$  sec intervals (<http://basilisk.fr/src/test/large.c>).



**Figure 12.** Timeseries of maximum vertical velocity and maximum slope (<http://basilisk.fr/src/test/large.c>).

## 4.5 Wind-driven basin

We next consider a simple case aiming to validate the implementation of both vertical viscosity and remapping. A ten-meters-long and one-meter-deep basin is set in motion by a surface wind stress. Balance is eventually reached between energy injection through surface wind and dissipation by viscosity. The corresponding steady-state velocity field is illustrated on Figure 13: the wind is blowing from left to right and drives an overall parallel circulation with a “downwelling” on the right and (asymmetric) “upwelling” on the left. The free-surface is not flat since an overall horizontal pressure gradient is required to balance the wind stress.



**Figure 13.** Velocity field for a wind-driven basin (<http://basilisk.fr/src/test/wind-driven-nh.c>).

The problem has five dimensional parameters: the length  $L$ , depth  $H$ , acceleration of gravity  $g$ , vertical gradient of surface velocity  $\dot{u}$  and kinematic viscosity  $\nu$ . Only three of



these are dependent, for example

$$\begin{aligned} a &= \frac{L}{H}, \\ \text{Re} &= \frac{\dot{u} H^2}{\nu} = s \frac{g H^3}{\nu^2}, \\ s &= \frac{\dot{u} \nu}{g H} = \frac{\dot{u}^2 H}{\text{Re} g}, \end{aligned}$$

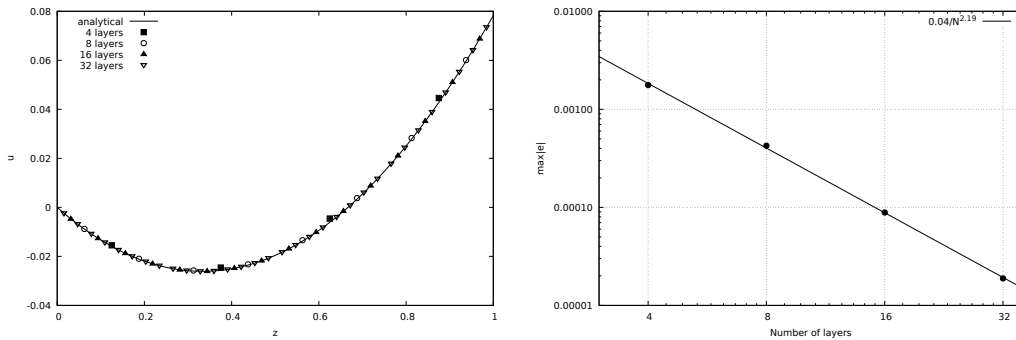
where  $a$  is the aspect ratio,  $\text{Re}$  the Reynolds number and  $s$  the slope of the free-surface. The characteristic time scale over which balance is expected to be reached is

$$t_\nu = \frac{H^2}{\nu}$$

In the case of exactly parallel flow (which is expected in the centre of the domain), it is easy to show that the velocity profile reduces to the parabolic solution

$$u(z) = \sqrt{s \text{Re} g H} \frac{z(3z - 2H)}{4H^2}$$

To ensure that the numerical solution is indeed close to parallel in the middle of the domain, the slope is set to  $s = 1/1000$ . To obtain a stationary regime reasonably quickly, the Reynolds number is set to  $\text{Re} = 10$ . The horizontal dimension is discretised using 64 grid points and the vertical using 4, 8, 16 or 32 layers. The simulation is started from rest and is run for  $10 t_\nu$  to make sure that a steady-state solution is reached. Figure 14 illustrates the convergence toward the theoretical horizontal velocity profile in the centre of the domain. Note that this configuration is a typical case for which a Lagrangian method without vertical remapping would fail, since the layers would quickly collapse toward the top-left and bottom-right corners due to the vertical motion.



**Figure 14.** Velocity profiles (left) and error convergence (right) (<http://basilisk.fr/src/test/wind-driven-nh.c>).

## 4.6 Viscous hydraulic jump

Shocks are a well-known characteristic of the hyperbolic nature of the Saint-Venant equations. It is also well-known that they can be a useful theoretical model of a wide range of real features observed in geophysical flows and/or hydraulic engineering e.g. wave breaking [53], bores and hydraulic jumps [16]. The inclusion of viscous and non-hydrostatic effects into this theoretical model is difficult however, and numerical models (combined with experiments) are a useful way to improve our understanding of these complex processes.

### 4.6.1 Hydrostatic

We first consider the viscous, hydrostatic transcritical flow over a bump, driven by a parabolic velocity profile at inflow and an outflow at a given depth. The inflow rate and

water depth at the outflow are chosen so that the Froude number  $U / \sqrt{gH}$  is smaller than one at inflow, larger than one over the bump and smaller than one at the outflow (i.e. the flow is transcritical). The domain is 30 metres long and roughly one metre deep, so not unlike a typical channel flow one would encounter in hydraulic engineering. The bump geometry is given by the Gaussian function

$$z_b(x) = 0.4 e^{-\frac{(x-10)^2}{5}}$$

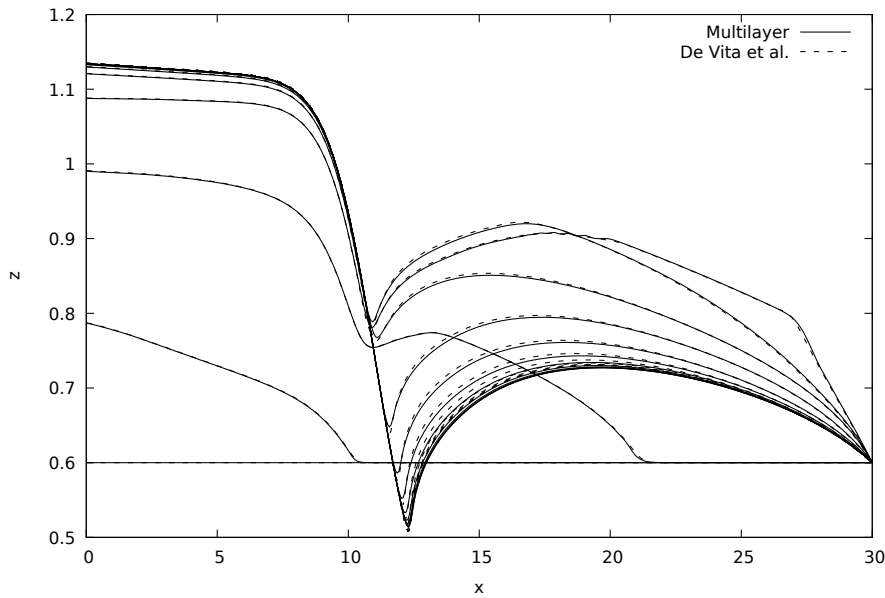
The outflow depth is set to 0.6 metres and the inflow profile is given by

$$u(0, z) = \frac{3}{2} \frac{Q}{H(0)} \left[ 1 - \left( \frac{z}{H(0)} - 1 \right)^2 \right],$$

with the flow rate  $Q$  set to one  $\text{m}^2/\text{s}$ . The (vertical) kinematic viscosity is set to  $10^{-2} \text{ m}^2/\text{s}$  and the acceleration of gravity is  $g = 9.81 \text{ m/s}^2$ . Note that viscosity drastically changes the expected solution, since it controls the position of the (stationary) hydraulic jump: in the absence of viscosity the hydraulic jump is simply advected out of the domain.

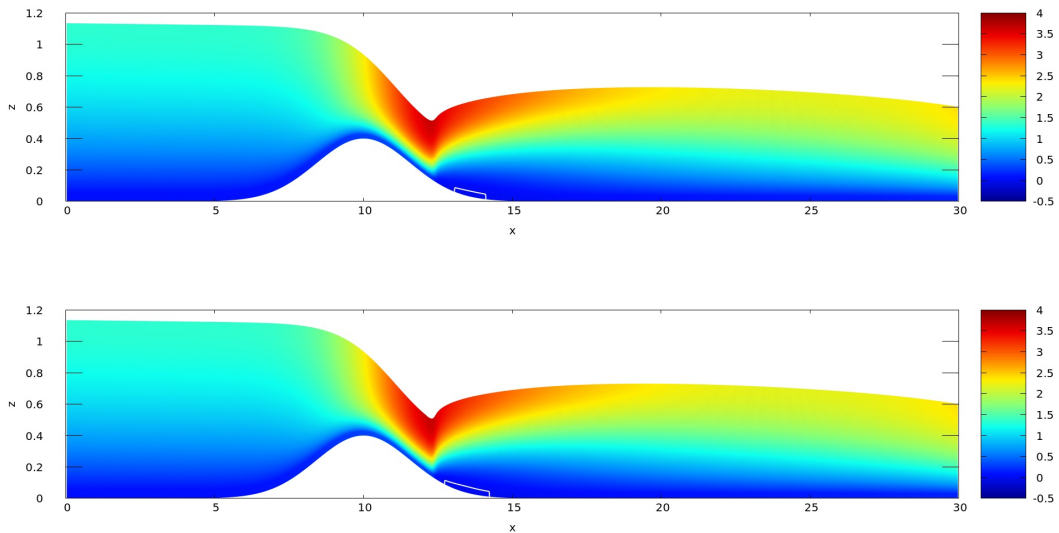
The horizontal is discretised with 512 grid points and the vertical with 20 layers. We consider two different numerical models: the present multilayer model run in hydrostatic mode with vertical viscosity and remapping and the shock-capturing, Riemann, quasi-Eulerian multilayer model described in [16], itself inspired from [15].

The simulation is started from rest with a constant water level of 0.6 metres. The corresponding evolution of the free surface is illustrated on Figure 15. As fluid flows in from the left and hits the fixed-depth right boundary, an hydraulic jump forms just downstream of the bump and eventually reaches a stationary position (after about 60 seconds). The agreement between the two models is clearly very good, both for the time evolution and the stationary solution, with a slightly sharper shock for the Riemann-based scheme.



**Figure 15.** Evolution of the free-surface obtained with the two models indicated in the legend. The time interval between profiles is 5 seconds (<http://basilisk.fr/src/test/gaussian.c>).

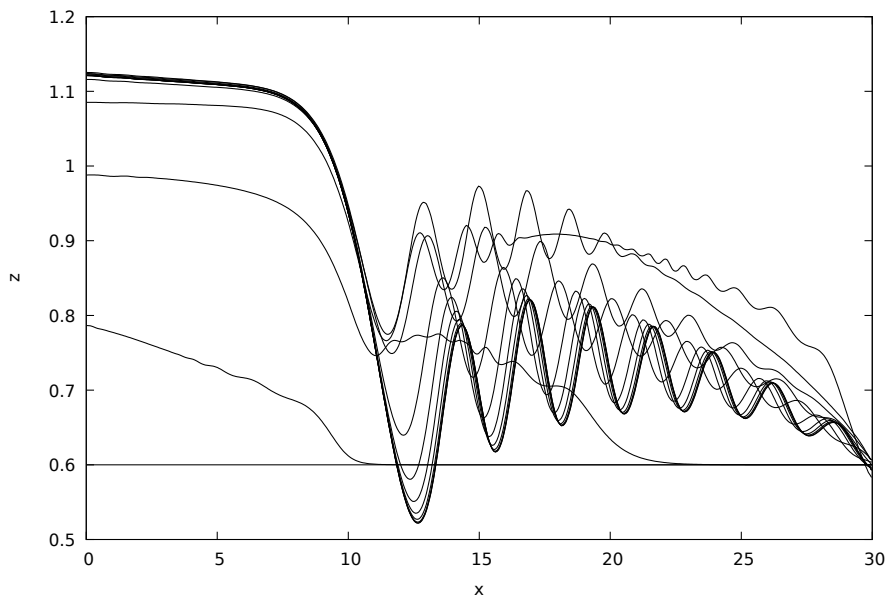
This agreement is confirmed by the comparison of the stationary horizontal velocity fields displayed on Figure 16. The white isoline indicates the presence of a (barely resolved) weak recirculation zone behind the bump.



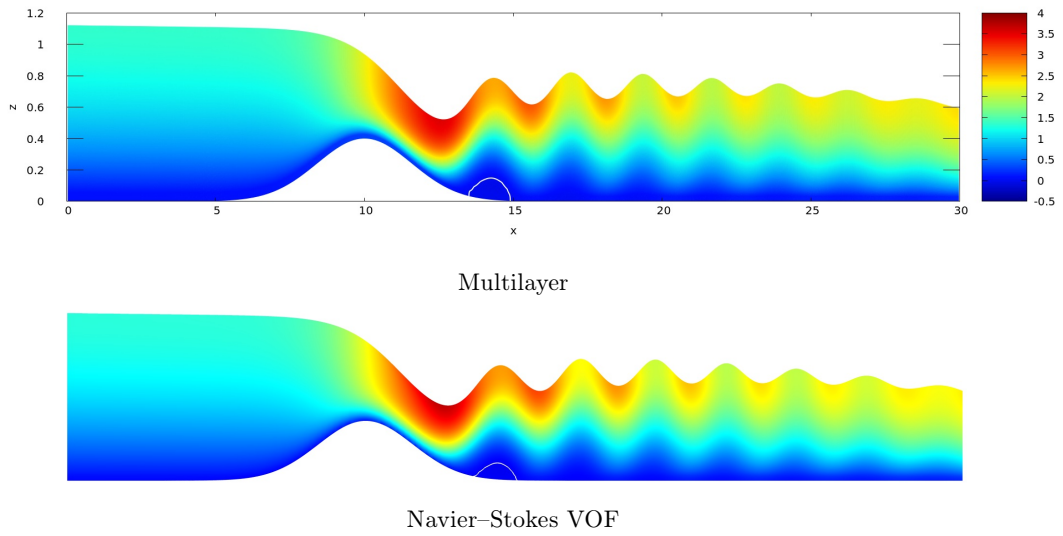
**Figure 16.** Horizontal velocity. (top) Hydrostatic multilayer model. (bottom) De Vita et al. The white isoline indicates  $u=0$  (<http://basilisk.fr/src/test/gaussian.c>).

### 4.6.2 Non-hydrostatic

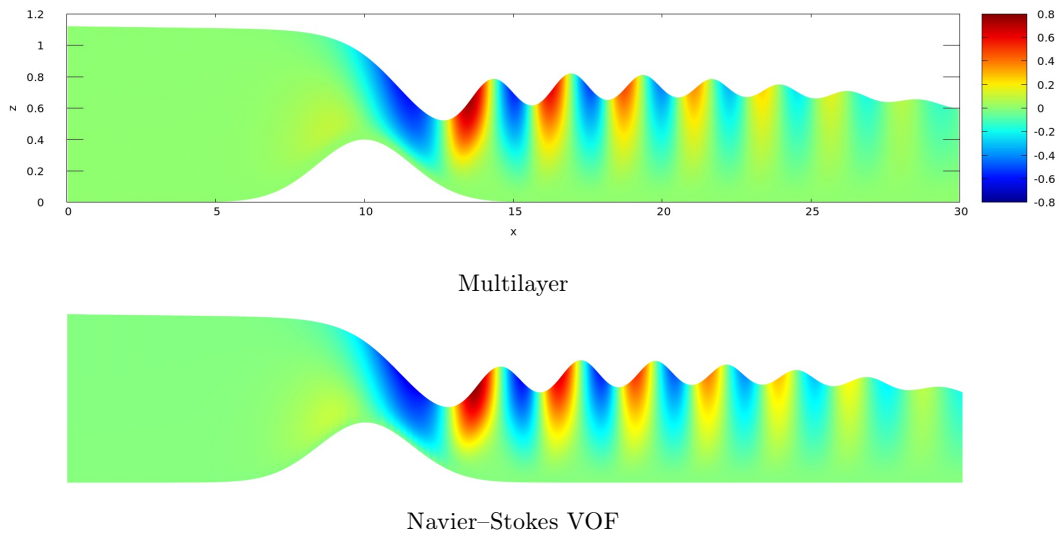
We repeat the same simulation using the non-hydrostatic multilayer solver. This leads to a very different stationary solution, as illustrated in Figures 17, 18 and 19. Vertical inertia is not negligible and leads to the formation of a dispersive wave train rather than a shock-like solution. This is a much more realistic solution than the hydrostatic version and recalls the waves first studied by Favre [60] in the context of dam overflows. The non-hydrostatic pressure field displayed in Figure 20 further confirms that non-hydrostatic effects are important (i.e. the non-hydrostatic pressure divided by the density is of order one, so is comparable to the hydrostatic pressure).



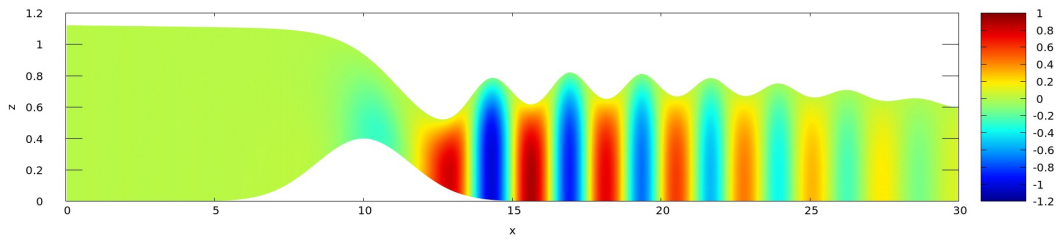
**Figure 17.** Evolution of the free-surface. The time interval between profiles is 5 seconds (<http://basilisk.fr/src/test/gaussian.c>).



**Figure 18.** Horizontal velocity field computed with the multilayer solver (<http://basilisk.fr/src/test/gaussian.c>) and the VOF Navier-Stokes solver (<http://basilisk.fr/src/examples/gaussian-ns.c>). The white isoline indicates  $u = 0$ .

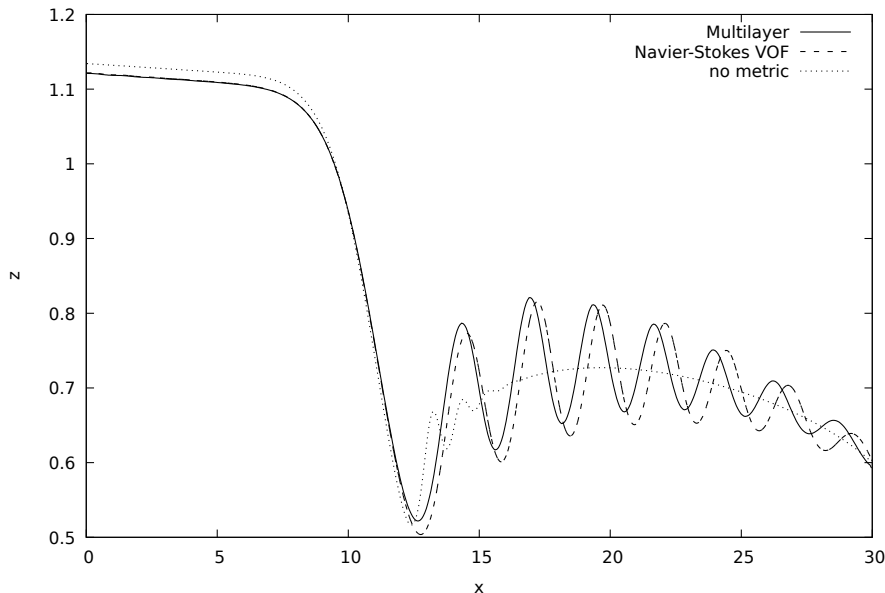


**Figure 19.** Vertical velocity field computed with the multilayer solver and the VOF Navier-Stokes solver.



**Figure 20.** Non-hydrostatic pressure field (divided by the fluid density) computed with the multilayer solver (<http://basilisk.fr/src/test/gaussian.c>).

In Figures 18 and 19, the horizontal and vertical components of the velocity are compared with solutions obtained using the Navier–Stokes/VOF solver introduced in section 4.4. The solution is significantly more difficult to compute in the present case, in particular due to the large horizontal to vertical aspect ratio. Converged results required an equivalent resolution of  $1024 \times 50$  grid points for the Navier–Stokes solver. The agreement between the two solutions is clearly qualitatively good, including for the weak recirculation zone behind the bump, which is now properly resolved. Figure 21 gives a more quantitative comparison of the final free-surface profiles. The importance of including the metric/slope terms is clear, even for this configuration with a relatively mild maximum slope of 0.25. The source of the remaining discrepancies between the Navier–Stokes and multilayer solutions is unclear. An obvious candidate could be the lack of horizontal viscosity in the multilayer model, but its addition does not seem to significantly change the results, which is not so surprising given the  $1/30$  aspect ratio of the flow. An explanation could be some remaining numerical errors in the Navier–Stokes VOF solution.



**Figure 21.** Final free-surface profiles (<http://basilisk.fr/src/test/gaussian.c>).

Finally Table 2 gives a summary of the performances of all solvers for this case. The

gain in number of timesteps due to the relaxed stability condition (33) is significant again, but not sufficient to offset the overhead due to the resolution of the Poisson equation (when comparing the runtimes of the hydrostatic and non-hydrostatic multilayer solvers). The hydrostatic multilayer solver is significantly slower than the quasi-Eulerian, Riemann solver of [16]. This is due to the added cost of vertical remapping, and reflects the poor performance of the PPR library we are using. The gain in runtime between the multilayer and Navier–Stokes/VOF solutions is remarkable, even if the comparison is somewhat unfair since the VOF solver cannot at the moment use anisotropic meshes which would allow to optimise the discretisation.

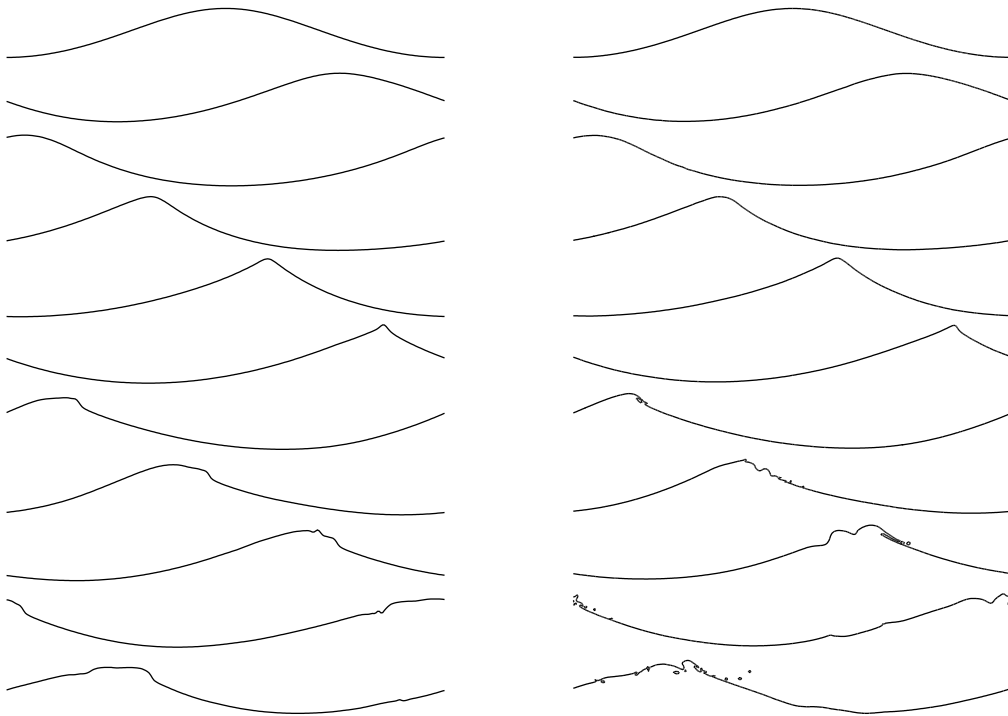
Model	# Timesteps	Runtime	Speed
Hydrostatic (De Vita et al.)	13078	28	4 782 811
Hydrostatic (multilayer)	12945	41	3 233 092
Non-hydrostatic (multilayer)	8448	147	588 486
Non-hydrostatic (NS/VOF)	14754	4448	186 000

**Table 2.** Runtimes (seconds) for the viscous hydraulic jump case. The speeds are given in points  $\times$  timesteps  $\times$  layers/sec. NS/VOF run on eight cores, others run on a single core.

## 4.7 Breaking Stokes wave

Since the study and prediction of (very) short waves is one of the primary motivation for the development of dispersive wave models, we now choose an example which illustrates that the multilayer model can handle robustly and accurately the detailed modelling of very steep water waves up to, and even beyond their threshold of stability. In previous work we have studied in detail the evolution of unstable Stokes waves, including breaking and bubble cloud formation [61, 62]. We reuse this setup here, with an initial free-surface and velocity field given by a third-order Stokes wave solution [63]. Since we do not include surface tension, there are only two parameters: the Reynolds number  $\text{Re} = \sqrt{g\lambda^3}/\nu$  and the initial steepness  $a\hat{k}$ , with  $\lambda = 2\pi/\hat{k}$  the wavelength and  $a$  the wave amplitude. The water depth is set to half the wavelength and is assumed to be deep enough to have a negligible influence on the results. The solution is obtained using both the two-phase Navier–Stokes/VOF solver (with density and viscosity ratios set to air/water) and the free-surface, non-hydrostatic multilayer solver.

The Reynolds number is set to  $\text{Re} = 40000$  and the steepness to  $a\hat{k} = 0.35$ , which corresponds to a spilling/overturning configuration [61]. The multilayer solver uses 256 grid points in the horizontal and 60 vertical layers. The Navier–Stokes/VOF solver uses an adaptive mesh with a maximum equivalent resolution of  $512^2$ . Figure 22 illustrates the free-surface evolution for both solvers. The time interval between profiles is  $T/4$  with  $T = \hat{k}/\sqrt{g\hat{k}}$  the Stokes wave period. The agreement between the two solutions is clearly excellent right up to the point of breaking at  $t \approx 5T/4$ . The multilayer code cannot be expected to match the VOF solution after breaking, since the interface becomes multivalued, however it is still able to robustly compute a solution which is in remarkable qualitative agreement with the VOF solution. The runtimes and speeds given in Table 3 are also favourable.



**Figure 22.** Free-surface evolution. Time increases from top to bottom in intervals of  $T/4$  with  $T$  the Stokes wave period. Left column: layered model. Right column: Navier–Stokes/VOF (<http://basilisk.fr/src/test/stokes.c>).

Model	# Timesteps	Runtime	Speed
Multilayer	1489	69	332 400
Navier-Stokes/VOF	3079	215	97 200

**Table 3.** Runtimes (seconds) for the breaking Stokes wave case. The speeds are given in points $\times$  layers $\times$ timesteps/sec/core. Run on a single core.

## 5 Conclusion and perspectives

The semi-discrete system (1), (2), (3) and (4) is a consistent discretisation of the incompressible Euler equations with topography and a free-surface, valid without assumptions on the slopes of the interfaces. Expressed as a set of conservation laws for each layer, the formulation has a clear physical interpretation and makes a seamless link between the hydrostatic Saint-Venant equations, dispersive Boussinesq-style models and the incompressible Euler equations.

The associated numerical scheme follows this consistent hierarchy and provides accurate and efficient solutions for all regimes. For surface gravity waves the same model can then be used to study metre-scale waves, even beyond breaking, with results closely matching those obtained using small-scale Euler/Navier–Stokes models, and coastal or global scale dispersive waves, with an accuracy and efficiency comparable to extended Boussinesq wave models.



This efficiency is obtained through the introduction of an “approximate vertical projection” which does not compromise conservation properties due to the decoupling provided by the vertically-Lagrangian discretisation. Accurate discrete dispersion relations are obtained thanks to the multigrid-accelerated, Hessenberg column-relaxation solution of a Keller box scheme vertical projection.

The method allows simple and robust wetting and drying on complex topography as demonstrated by adaptive solutions of the dispersive propagation and inundation during the Japanese Tohoku tsunami. Remarkably, for this type of application the introduction of a relaxed timestep restriction, based on the dispersive properties of non-hydrostatic waves, leads to shorter runtimes for the dispersive model than for the hydrostatic version.

This solid basis opens a number of perspectives. From a computational point of view, the performance of the vertical remapping should be improved. Since vertical remapping is comparable to one-dimensional advection (see Section 3.3), there is no reason for its computational cost to be three times as large, as is the case in our implementation using the PPR library. This optimisation should further increase the overall performance.

The flexibility of vertical remapping should also be explored with the goal of developing “vertically adaptive” methods able to automatically track material interfaces, steep gradients and boundary layers.

The ability of the model to provide accurate solutions right up to the point of breaking allows a much deeper exploration of the possible wave breaking parameterisations than what is done in Section 3.6.4.

Finally, as suggested in the introduction, an obvious extension is the inclusion of variable material properties, starting with density. We envisage to take into account both large density differences and Boussinesq-style small density variations. This would allow to efficiently model both multi-material flows, such as wind-driven waves or submarine avalanches and the more classical density/salinity stratified flows typical for oceanic motion. This will be the topic of a follow-up publication.

## Acknowledgements

I would like to thank Pierre-Yves Lagrée for early feedback on this project and all the colleagues who provided comments on the initial versions of this manuscript.

## Bibliography

- [1] Stéphane Popinet and collaborators. Basilisk. <http://basilisk.fr>, 2013–2019.
- [2] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.
- [3] Roger Temam. Une méthode d’approximation de la solution des équations de Navier-Stokes. *Bulletin de la Société Mathématique de France*, 96:115–152, 1968.
- [4] Adhémar Jean-Claude de Saint-Venant. Théorie du mouvement non-permanent des eaux, avec application aux crues des rivières et à l’introduction des marées dans leur lit. *Compte-Rendus de l’Académie des Sciences de Paris*, 73(147-154):5, 1871.
- [5] Geoffrey K Vallis. *Atmospheric and oceanic fluid dynamics*. Cambridge University Press, 2017.
- [6] Vladimir Zeitlin. *Geophysical fluid dynamics: understanding (almost) everything with rotating shallow water models*. Oxford University Press, 2018.
- [7] Knut Klingbeil, Florian Lemarié, Laurent Debreu, and Hans Burchard. The numerics of hydrostatic structured-grid coastal ocean models: State of the art and future perspectives. *Ocean Modelling*, 125:80–105, 2018.
- [8] Rainer Bleck. Ocean modeling in isopycnic coordinates. In *Ocean Modeling and Parameterization*, pages 423–448. Springer, 1998.

- [9] Akira Kasahara. Various vertical coordinate systems used for numerical weather prediction. *Monthly Weather Review*, 102(7):509–522, 1974.
- [10] Rainer Bleck. On the use of hybrid vertical coordinates in numerical weather prediction models. *Monthly Weather Review*, 106(9):1233–1244, 1978.
- [11] Stephen M Griffies, Claus Böning, Frank O Bryan, Eric P Chassignet, Rüdiger Gerdes, Hiroyasu Hasumi, Anthony Hirst, Anne-Marie Treguier, and David Webb. Developments in ocean climate modelling. *Ocean Modelling*, 2(3-4):123–192, 2000.
- [12] Alistair Adcroft and Robert Hallberg. On methods for solving the oceanic equations of motion in generalized vertical coordinates. *Ocean Modelling*, 11(1-2):224–233, 2006.
- [13] Eric P Chassignet. Isopycnic and hybrid ocean modeling in the context of GODAE. In *Operational Oceanography in the 21st Century*, pages 263–293. Springer, 2011.
- [14] Emmanuel Audusse. A multilayer Saint-Venant model: derivation and numerical validation. *Discrete & Continuous Dynamical Systems-B*, 5(2):189–214, 2005.
- [15] Emmanuel Audusse, Marie-Odile Bristeau, Benoît Perthame, and Jacques Sainte-Marie. A multilayer Saint-Venant system with mass exchanges for shallow water flows. Derivation and numerical validation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 45:169–200, 2011.
- [16] F. De Vita, P.Y. Lagrée, S. Chibbaro, and S. Popinet. Beyond shallow water: Appraisal for a numerical approach to hydraulic jumps based upon the boundary layer theory. *European Journal of Mechanics - B/Fluids*, 79/C:233–246, 2019.
- [17] Joseph Boussinesq. Théorie de l’intumescence liquide appelée onde solitaire ou de translation se propageant dans un canal rectangulaire. *Compte-Rendus de l’Académie des Sciences de Paris*, 72(755–759), 1871.
- [18] François Serre. Contribution à l’étude des écoulements permanents et variables dans les canaux. *La Houille Blanche*, 6:830–872, 1953.
- [19] D Howell Peregrine. Long waves on a beach. *Journal of Fluid Mechanics*, 27(4):815–827, 1967.
- [20] Albert E Green and Paul M Naghdi. A derivation of equations for wave propagation in water of variable depth. *Journal of Fluid Mechanics*, 78(2):237–246, 1976.
- [21] Maurizio Brocchini. A reasoned overview on Boussinesq-type models: the interplay between physics, mathematics and numerics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 469(2160):20130496, 2013.
- [22] Patrick J Lynett and Philip L-F Liu. Linear analysis of the multi-layer model. *Coastal Engineering*, 51(5-6):439–454, 2004.
- [23] Florent Chazel, Michel Benoit, Alexandre Ern, and Serge Piperno. A double-layer Boussinesq-type model for highly nonlinear and dispersive waves. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2108):2319–2346, 2009.
- [24] Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, 1965.
- [25] Akio Arakawa. Computational design for long-term numerical integration of the equations of fluid motion: Two-dimensional incompressible flow. Part I. *Journal of Computational Physics*, 1(1):119–143, 1966.
- [26] John Marshall, Alistair Adcroft, Chris Hill, Lev Perelman, and Curt Heisey. A finite-volume, incompressible Navier-Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, 102(C3):5753–5766, 1997.
- [27] Vincenzo Casulli. A semi-implicit finite difference method for non-hydrostatic, free-surface flows. *International Journal for Numerical Methods in Fluids*, 30(4):425–440, 1999.
- [28] Guus Stelling and Marcel Zijlema. An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. *International Journal for Numerical Methods in Fluids*, 43(1):1–23, 2003.
- [29] Marcel Zijlema and Guus S Stelling. Further experiences with computing non-hydrostatic free-surface flows involving water waves. *International Journal for Numerical Methods in Fluids*, 48(2):169–197, 2005.
- [30] Y Kanarska, A Shchepetkin, and JC McWilliams. Algorithm for non-hydrostatic dynamics in the regional oceanic modeling system. *Ocean Modelling*, 18(3-4):143–174, 2007.
- [31] Stephen M. Griffies, Alistair Adcroft, and Robert W. Hallberg. A primer on ocean generalized vertical coordinate dynamical cores based on the vertical Lagrangian-remap method. *Journal of Advances in Modeling Earth Systems*, 2019. Submitted.
- [32] Sean Vitousek and Oliver B Fringer. A nonhydrostatic, isopycnal-coordinate ocean model for internal waves. *Ocean Modelling*, 83:118–144, 2014.
- [33] Ling Zhu, Qin Chen, and Xiaoliang Wan. Optimization of non-hydrostatic Euler model for water waves. *Coastal Engineering*, 91:191–199, 2014.

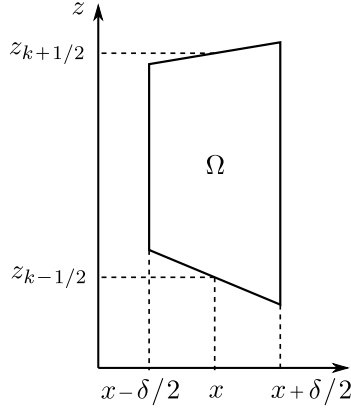
- [34] Y. Yamazaki, Z. Kowalik, and K. F. Cheung. Depth-integrated, non-hydrostatic model for wave breaking and run-up. *Int. J. Numer. Meth. Fluids*, 61:473–497, 2008.
- [35] Marie-Odile Bristeau, Anne Mangeney, Jacques Sainte-Marie, and Nicolas Seguin. An energy-consistent depth-averaged Euler system: Derivation and properties. *Discrete & Continuous Dynamical Systems-B*, 20(4):961–988, 2015.
- [36] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoit Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.
- [37] Emmanuel Audusse and Marie-Odile Bristeau. Finite-volume solvers for a multilayer Saint-Venant system. *International Journal of Applied Mathematics and Computer Science*, 17(3):311–320, 2007.
- [38] John B Bell, Phillip Colella, and Harland M Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 85(2):257–283, 1989.
- [39] Alexander F Shchepetkin and James 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.
- [40] Miodrag Rančić. An efficient, conservative, monotonic remapping for semi-Lagrangian transport algorithms. *Monthly Weather Review*, 123(4):1213–1217, 1995.
- [41] Laurent White and Alistair Adcroft. A high-order finite volume remapping scheme for nonuniform grids: The piecewise quartic method (PQM). *Journal of Computational Physics*, 227(15):7394–7422, 2008.
- [42] Richard Hofmeister, Hans Burchard, and Jean-Marie Beckers. Non-uniform adaptive vertical grids for 3D numerical ocean models. *Ocean Modelling*, 33(1-2):70–86, 2010.
- [43] Mark R Petersen, Douglas W Jacobsen, Todd D Ringler, Matthew W Hecht, and Mathew E Maltrud. Evaluation of the arbitrary Lagrangian–Eulerian vertical coordinate method in the MPAS-ocean model. *Ocean Modelling*, 86:93–113, 2015.
- [44] S. Popinet. Quadtree-adaptive tsunami modelling. *Ocean Dynamics*, 61(9):1261–1285, 2011.
- [45] J. Antoon van Hooft, Stéphane Popinet, Chiel C. van Heerwaarden, Steven J.A. van der Linden, Stephan R. de Roode, and Bas J.H. van de Wiel. Towards adaptive grids for atmospheric boundary-layer simulations. *Boundary-Layer Meteorology*, 167(3):421–443, 2018.
- [46] Herbert B Keller. A new difference scheme for parabolic problems. In *Numerical Solution of Partial Differential Equations–II*, pages 327–350. Elsevier, 1971.
- [47] Didier Clamond, Denys Dutykh, and Dimitrios Mitsotakis. Conservative modified Serre–Green–Naghdi equations with improved dispersion characteristics. *Communications in Nonlinear Science and Numerical Simulation*, 45:245–257, 2017.
- [48] Ann S Almgren, John B Bell, and William Y Crutchfield. Approximate projection methods: Part I. inviscid analysis. *SIAM Journal on Scientific Computing*, 22(4):1139–1159, 2000.
- [49] S. Popinet. A quadtree-adaptive multigrid solver for the Serre–Green–Naghdi equations. *Journal of Computational Physics*, 302:336–358, 2015.
- [50] Marcel Zijlema. SWASH code. <http://swash.sourceforge.net>, 2019. Accessed November 2019.
- [51] Greg Henry. *The shifted Hessenberg system solve computation*. Cornell Theory Center, Cornell University, 1994.
- [52] Per A Madsen, OR Sørensen, and HA Schäffer. Surf zone dynamics simulated by a Boussinesq type model. Part I. Model description and cross-shore motion of regular waves. *Coastal Engineering*, 32(4):255–287, 1997.
- [53] Maurizio Brocchini and Nicholas Dodd. Nonlinear shallow water equation modeling for coastal engineering. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 134(2):104–120, 2008.
- [54] S. Popinet. Adaptive modelling of long-distance wave propagation and fine-scale flooding during the Tohoku tsunami. *Natural Hazards and Earth System Sciences*, 12(4):1213–1227, 2012.
- [55] S. Beji and J.A. Battjes. Experimental investigation of wave propagation over a bar. *Coastal Engineering*, 19:151–162, 1993.
- [56] H.R. Luth, G. Klopman, and N. Kitou. Project 13G: kinematics of waves breaking partially on an offshore bar. Technical report, Delfts hydraulics laboratory, 1994.
- [57] FI González and Ye A Kulikov. Tsunami dispersion observed in the deep ocean. In *Tsunamis in the World*, pages 7–16. Springer, 1993.
- [58] S Glimsdal, GK Pedersen, CB Harbitz, and F Løvholt. Dispersion of tsunamis: does it really matter? *Nat. Hazards Earth Syst. Sci.*, 13:1507–1526, 2013.
- [59] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228:5838–5866, 2009.
- [60] Henri Favre. Etude théorique et expérimentale des ondes de translation dans les canaux découverts. *Dunod*, 150, 1935.

- [61] Luc Deike, Stéphane Popinet, and W Kendall Melville. Capillary effects on wave breaking. *Journal of Fluid Mechanics*, 769:541–569, 2015.
- [62] Luc Deike, W Kendall Melville, and Stéphane Popinet. Air entrainment and bubble statistics in breaking waves. *Journal of Fluid Mechanics*, 801:91–129, 2016.
- [63] H. Lamb. *Hydrodynamics*. The University Press, 1932.
- [64] Maxima. A computer algebra system. Version 5.34.1. <http://maxima.sourceforge.net>, 2014.

## Appendix A Derivation of slope terms

The set of equations (1), (2), (3) and (4) can be derived using different methods: for example one could start from the incompressible Euler equations, written in Cartesian coordinates and use the general vertical coordinate transformation proposed by Kasahara (1974) [9], followed by formal vertical integration over the layer thicknesses (see also [12]).

Another approach is to apply conservation principles directly to thin vertical slices for each of the layers, a technique close to that used by Saint-Venant to derive the single-layer system. We will assume here that the hydrostatic equations (6) and (7) are given and will first deal with the non-hydrostatic pressure terms in the horizontal momentum equation (2).



**Figure 23.** Thin vertical slice through a layer  $k$ .

If we consider a thin vertical slice  $\Omega$  of width  $\delta$  through a layer bounded vertically by  $[z_{k-1/2}, z_{k+1/2}]$  (Figure 23), the variation in the total horizontal momentum of this slice due to pressure can be written

$$\partial_t \int_{\Omega} \rho u \, d\Omega = - \int_{\Omega} \partial_x p \, d\Omega = - \int_{\partial\Omega} p \, dz,$$

where  $\partial\Omega$  denotes the boundary of  $\Omega$ . This can be further developed as

$$\begin{aligned} \int_{\partial\Omega} p \, dz &= p_k^+(z_{k+1/2} - z_{k-1/2})^+ - p_k^-(z_{k+1/2} - z_{k-1/2})^- \\ &\quad - p_{k+1/2}(z^+ - z^-)_{k+1/2} + p_{k-1/2}(z^+ - z^-)_{k-1/2} \\ &= p_k^+ h_k^+ - p_k^- h_k^- - [p(z^+ - z^-)]_k, \end{aligned}$$

where we have used the definition of the layer thicknesses  $h_k = z_{k+1/2} - z_{k-1/2}$  and the vertical difference operator. The  $+$  and  $-$  superscripts denote average values on the  $x \pm \delta/2$  vertical boundaries. For a vanishing  $\delta$ , we have  $f^+ - f^- = \delta \partial_x f$ , where  $f$  is a differentiable function. This readily gives

$$\int_{\partial\Omega} p \, dz = \delta \partial_x (p h)_k - \delta [p \partial_x z]_k$$

Since the integrated horizontal momentum can be written

$$\int_{\Omega} \rho u d\Omega = \delta \rho h_k u_k,$$

with  $u_k$  the layer-averaged horizontal velocity, we get

$$\begin{aligned} \partial_t \int_{\Omega} \rho u d\Omega &= - \int_{\partial\Omega} p dz, \\ \partial_t(\rho h_k u_k) &= -\partial_x(p h)_k + [p \partial_x z]_k, \end{aligned}$$

which, for  $\rho$  constant and  $\phi = p/\rho$ , corresponds to the non-hydrostatic pressure terms in equation (2). Note that the corresponding derivation for the vertical momentum equation (3) is trivial since the  $x \pm \delta/2$  boundaries are strictly vertical.

We next consider the continuity equation (4). Conservation of volume can be written

$$\begin{aligned} 0 &= \int_{\Omega} \nabla \cdot \mathbf{u} d\Omega = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} d\partial\Omega \\ &= u_k^+ (z_{k+1/2} - z_{k-1/2})^+ - u_k^- (z_{k+1/2} - z_{k-1/2})^- \\ &\quad - u_{k+1/2} \delta \partial_x z_{k+1/2} + u_{k-1/2} \delta \partial_x z_{k-1/2} \\ &\quad + w_{k+1/2} \delta - w_{k-1/2} \delta \\ &= \delta \partial_x (u h)_k - \delta [u \partial_x z]_k + \delta [w]_k \\ 0 &= \partial_x (h u)_k + [w - u \partial_x z]_k, \end{aligned}$$

which is the one-dimensional version of (4).

## Appendix B Discrete dispersion relations

The discrete dispersion relation is given by

$$\omega_n^2(\hat{k}) = g \frac{\sum_{i=1}^n a_{2i} \hat{k}^{2i}}{\sum_{i=0}^n b_{2i} \hat{k}^{2i}}$$

The following coefficients can be computed using the Maxima [64] script available at <http://basilisk.fr/src/test/dispersion.mac>

### B.1 Keller box scheme

One layer:

$$\begin{aligned} a_2 &= 4 h_0 \\ b_0 &= 4 \\ b_2 &= h_0^2 \end{aligned}$$

Two layers:

$$\begin{aligned} a_2 &= 16 (h_1 + h_0) \\ a_4 &= 4 h_1 (h_0^2 + h_0 h_1) \\ b_0 &= 16 \\ b_2 &= 4 (h_1^2 + 4 h_0 h_1 + h_0^2) \\ b_4 &= h_0^2 h_1^2 \end{aligned}$$

Three layers:

$$\begin{aligned}
a_2 &= 64(h_2 + h_1 + h_0) \\
a_4 &= (16h_1 + 16h_0)h_2^2 + (16h_1^2 + 64h_0h_1 + 16h_0^2)h_2 + 16h_0h_1^2 + 16h_0^2h_1 \\
a_6 &= (4h_0h_1^2 + 4h_0^2h_1)h_2^2 + 4h_0^2h_1^2h_2 \\
b_0 &= 64 \\
b_2 &= 16h_2^2 + (64h_1 + 64h_0)h_2 + 16h_1^2 + 64h_0h_1 + 16h_0^2 \\
b_4 &= (4h_1^2 + 16h_0h_1 + 4h_0^2)h_2^2 + (16h_0h_1^2 + 16h_0^2h_1)h_2 + 4h_0^2h_1^2 \\
b_6 &= h_0^2h_1^2h_2^2
\end{aligned}$$

Four layers:

$$\begin{aligned}
a_2 &= 256(h_3 + h_2 + h_1 + h_0) \\
a_4 &= (64h_2 + 64h_1 + 64h_0)h_3^2 + (64h_2^2 + (256h_1 + 256h_0)h_2 + 64h_1^2 + 256h_0h_1 + 64h_0^2)h_3 + \\
&\quad (64h_1 + 64h_0)h_2^2 + (64h_1^2 + 256h_0h_1 + 64h_0^2)h_2 + 64h_0h_1^2 + 64h_0^2h_1 \\
a_6 &= ((16h_1 + 16h_0)h_2^2 + (16h_1^2 + 64h_0h_1 + 16h_0^2)h_2 + 16h_0h_1^2 + 16h_0^2h_1)h_3^2 + ((16h_1^2 + \\
&\quad 64h_0h_1 + 16h_0^2)h_2^2 + (64h_0h_1^2 + 64h_0^2h_1)h_2 + 16h_0^2h_1^2)h_3 + (16h_0h_1^2 + 16h_0^2h_1)h_2^2 + \\
&\quad 16h_0^2h_1^2h_2 \\
a_8 &= ((4h_0h_1^2 + 4h_0^2h_1)h_2^2 + 4h_0^2h_1^2h_2)h_3^2 + 4h_0^2h_1^2h_2^2h_3 \\
b_0 &= 256 \\
b_2 &= 64h_3^2 + (256h_2 + 256h_1 + 256h_0)h_3 + 64h_2^2 + (256h_1 + 256h_0)h_2 + 64h_1^2 + 256h_0h_1 + \\
&\quad 64h_0^2 \\
b_4 &= (16h_2^2 + (64h_1 + 64h_0)h_2 + 16h_1^2 + 64h_0h_1 + 16h_0^2)h_3^2 + ((64h_1 + 64h_0)h_2^2 + (64h_1^2 + \\
&\quad 256h_0h_1 + 64h_0^2)h_2 + 64h_0h_1^2 + 64h_0^2h_1)h_3 + (16h_1^2 + 64h_0h_1 + 16h_0^2)h_2^2 + (64h_0h_1^2 + \\
&\quad 64h_0^2h_1)h_2 + 16h_0^2h_1^2 \\
b_6 &= ((4h_1^2 + 16h_0h_1 + 4h_0^2)h_2^2 + (16h_0h_1^2 + 16h_0^2h_1)h_2 + 4h_0^2h_1^2)h_3^2 + ((16h_0h_1^2 + \\
&\quad 16h_0^2h_1)h_2^2 + 16h_0^2h_1^2h_2)h_3 + 4h_0^2h_1^2h_2^2 \\
b_8 &= h_0^2h_1^2h_2^2h_3^2
\end{aligned}$$

## B.2 Staggered scheme (third-order)

One layer:

$$\begin{aligned}
a_2 &= 8h_0 \\
b_0 &= 8 \\
b_2 &= 3h_0^2
\end{aligned}$$

Two layers:

$$\begin{aligned}
a_2 &= 8(h_1 + h_0) \\
a_4 &= 8h_0^2h_1 \\
b_0 &= 8 \\
b_2 &= 3(h_1^2 + h_0h_1 + 3h_0^2) \\
b_4 &= 3h_0^2h_1^2
\end{aligned}$$

Three layers:

$$\begin{aligned}
a_2 &= 8(h_2 + h_1 + h_0) \\
a_4 &= 8((h_1^2 + h_0 h_1 + h_0^2) h_2 + h_0^2 h_1) \\
a_6 &= 8 h_0^2 h_1^2 h_2 \\
b_0 &= 8 \\
b_2 &= 3 h_2^2 + (3 h_1 + 3 h_0) h_2 + 9 h_1^2 + 9 h_0 h_1 + 8 h_0^2 \\
b_4 &= (3 h_1^2 + 3 h_0 h_1 + 3 h_0^2) h_2^2 + 3 h_0^2 h_1 h_2 + 9 h_0^2 h_1^2 \\
b_6 &= 3 h_0^2 h_1^2 h_2^2
\end{aligned}$$

Four layers:

$$\begin{aligned}
a_2 &= 8(h_3 + h_2 + h_1 + h_0) \\
a_4 &= 8((h_2^2 + (h_1 + h_0) h_2 + h_1^2 + h_0 h_1 + h_0^2) h_3 + (h_1^2 + h_0 h_1 + h_0^2) h_2 + h_0^2 h_1) \\
a_6 &= 8(((h_1^2 + h_0 h_1 + h_0^2) h_2^2 + h_0^2 h_1 h_2 + h_0^2 h_1^2) h_3 + h_0^2 h_1^2 h_2) \\
a_8 &= 8 h_0^2 h_1^2 h_2^2 h_3 \\
b_0 &= 8 \\
b_2 &= 3 h_3^2 + (3 h_2 + 3 h_1 + 3 h_0) h_3 + 9 h_2^2 + (9 h_1 + 9 h_0) h_2 + 8 h_1^2 + 8 h_0 h_1 + 8 h_0^2 \\
b_4 &= (3 h_2^2 + (3 h_1 + 3 h_0) h_2 + 3 h_1^2 + 3 h_0 h_1 + 3 h_0^2) h_3^2 + ((3 h_1^2 + 3 h_0 h_1 + 3 h_0^2) h_2 + 3 h_0^2 h_1) h_3 + (9 h_1^2 + 9 h_0 h_1 + 9 h_0^2) h_2^2 + 9 h_0^2 h_1 h_2 + 8 h_0^2 h_1^2 \\
b_6 &= ((3 h_1^2 + 3 h_0 h_1 + 3 h_0^2) h_2^2 + 3 h_0^2 h_1 h_2 + 3 h_0^2 h_1^2) h_3^2 + 3 h_0^2 h_1^2 h_2 h_3 + 9 h_0^2 h_1^2 h_2^2 \\
b_8 &= 3 h_0^2 h_1^2 h_2^2 h_3^2
\end{aligned}$$