

Documentation for beom

Pierre St-Laurent
pierrestlau@globetrotter.net

April 24, 2016

Frequently Asked Questions

1. *What is beom?*

The Back of Envelope Ocean Model (beom) is a numerical solver for the multi-layer shallow-water equations. It simulates rotating basins with a free surface, wetting-drying, and layered stratification under the hydrostatic approximation. The name refers to the physical assumptions of the model (an inviscid, incompressible, one-component stably-stratified fluid) and to the streamlined code ($\sim 2,000$ lines of Fortran 95 distributed under a copyleft license; see Appendix K). The model targets computers ranging from laptops to workstations.

2. *How does it relate to existing ocean models?*

The basic model dynamics and numerics are similar to those of MICOM¹ or HIM². The model mostly differs by the way it treats vanishing layers (wetting/drying and isopycnal outcrops, see Appendix C) and by its regular unstructured grid on the horizontal (suitable for estuaries, Appendix F). The former allows the model to accurately handle large thickness gradients while preserving potential vorticity conservation (see Salmon, 2002). The code is able to reproduce several analytical solutions (see section Test-cases).

3. *Why not use models that already exist?*

The motivation behind beom is to simulate tides in rotating basins of complex geometry with high accuracy and reasonable efficiency. Coastal models such as FVCOM (Chen et al., 2007) use irregular grids whose design requires particular care in complex estuaries. Computational performance and conservation of enstrophy/energy are also problematic with irregular grids (Danilov, 2013). On the other end of the spectrum, General Circulation

Models (GCMs) focus on the larger scales with numerical methods (regular structured grids, implicit and/or linearized free surface, implicit vertical advection, no wetting/drying, no-slip lateral boundary conditions) that are conservative and efficient but unadapted for coastal regions (Ketefian and Jacobson, 2009; Chen et al., 2007). beom is somewhat of a hybrid in the sense that it combines finite-difference methods with a regular unstructured grid (Appendix F) to achieve acceptable performance in realistic estuaries (e.g., Fig. 19).

4. *How fast is beom?*

The code is optimized for basins of complex geometry and idealized stratification and for a multi-processor workstation. beom is most appropriate for calculations with a few layers (e.g. < 5) and a number of processors $4 < N < 64$. In this specific niche, the computational performances of the code are comparable or superior to other ocean models. If your primary concern is speed and not accuracy then you should look into other isopycnal models (e.g. GOLD for massively-parallel clusters or MICOM for a serial computer).

5. *How do I use the model? What do I need?*

All that is needed is a Fortran 95 compiler. I successfully used beom with Sun, Intel, and GNU Fortran compilers. On my laptop I simply type:

```
gfortran -Ofast -fopenmp shared_mod.f95 private_mod.f95 main.f95
export OMP_NUM_THREADS=2
./a.out
```

...and the calculation starts³.

¹The Miami Isopycnal Coordinate Ocean Model. See Bleck and Smith (1990).

²The Hallberg Isopycnal Model. See Hallberg and Rhines (1996).

³The Intel compiler ifort requires additionally the flags `-free -Tf` because it does not recognize the file extension `.f95` natively.

The ideal way to learn about the model is to begin with the various test-cases described in the next section. Each test-case comes with a Matlab⁴ script that creates all the files required for a calculation and then plots the results. The appendices at the end of this documentation provide additional insight on the model physics and numerics.

6. *Can beom be used in a global configuration?*

No. The model equations are discretized using finite-differences on a C-grid of uniform resolution ($\Delta x = \Delta y = \text{constant}$). Therefore beom can only be used in a regional configuration. Both periodic and non-periodic open boundaries (Flow Relaxation Scheme, Lavelle and Thacker, 2008) are implemented.

7. *Is beom funded/sponsored by an agency?*

beom is a personal, unfunded, late night/rainy Sunday afternoon project. I do it on my own time and for the fun of it. Feel free to use and modify beom (within the limits of the license; see Appendix K).

8. *What is the status of beom?*

The code has been successfully tested against a suite of test-cases for which analytical solutions were available (see next section). The focus is on adding test-cases, streamlining the code, and improving the documentation.

Test-case 1: Stommel 1948

The first test-case is taken from the classic paper of Stommel (1948). The physics are linear (no momentum advection) and correspond to a flat rectangular basin filled with homogeneous water. Winds, linear bottom friction and non-uniform rotation (β -plane) are included. A Octave/Matlab script named `testcases/stommel1948.m` creates the input files for the test-case and shows how the model results compare to the analytical solutions (Fig. 2).

Test-case 2: Seaward wind & Coastal upwelling

The case of a uniform seaward wind is particularly simple as it leads to a steady 2-D solution after a transient period $t \gg f^{-1}$ (Millot and Crépon, 1981). In this test-case, the shore-line is located at $x = 0$ and a uniform wind stress τ_x grows to 0.1 Pa over a period of 4 days.

⁴A free alternative to Matlab is GNU Octave (<https://www.gnu.org/software/octave>). The two software share their scripting syntax and thus the scripts are compatible. Octave is readily available as a package in most Linux distributions.

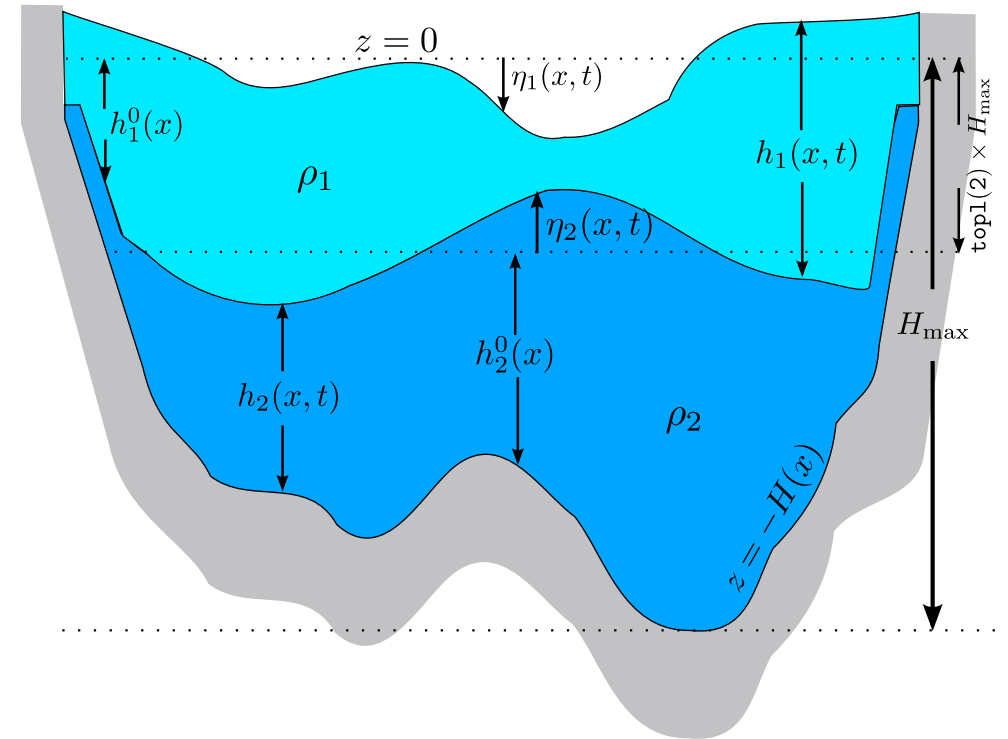


Figure 1: Geometry of the model for two layers in the x - z plane. The figure shows a case with outcrops ($h_2 \rightarrow 0$ near the coasts) but no wetting-drying ($h_1 \gg 0$ everywhere, see Fig. 13 for comparison). The vertical scale is exaggerated.

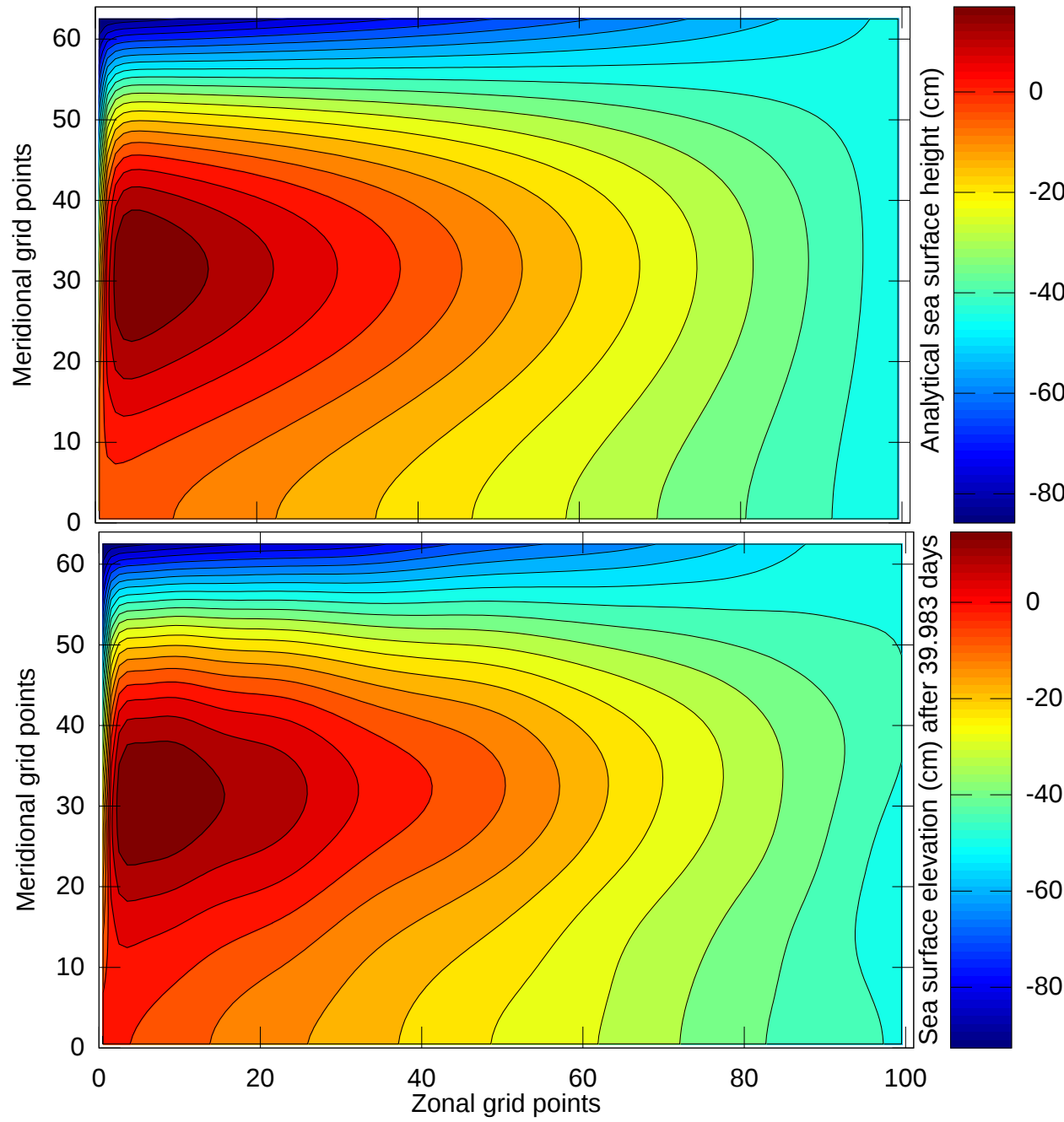


Figure 2: Sea surface elevation η_1 for $f = 10^{-11} \text{ y s}^{-1}$ in the test-case of Stommel (1948).

This produces an upwelling at the western boundary ($x = 0$). The problem is essentially 2-D and periodic boundary conditions are used in the longshore direction (y). For simplicity, the eastern boundary is a solid wall where the opposite physics occurs (downwelling). See the file `testcases/upwelling_seaward_wind.m` and Fig. 3.

Test-case 3: Conservation of properties

In absence of forcing and dissipation, the total mass, vorticity, potential enstrophy and energy of a physical layer should (ideally) be conserved by the model over time (see Ketefian and Jacobson, 2009). Mass and volume conservations are interchangeable in the context of beam since we deliberately neglect diapycnal exchanges and make the approximation of an incompressible flow and fluid. The simple spatial discretization used in the code guarantees the conservation of volume/mass and mechanical energy in any given layer (Appendix F). Vorticity and potential enstrophy are only approximately conserved by the spatial discretization. Round-off errors, land boundaries (Appendix G) and the time-discretization (Appendix E) are additional sources of errors that we will ignore for now.

The goal of the test-case is to provide some insight on the magnitude of these numerical errors. We begin by defining the volume, potential enstrophy and vorticity of a given layer i (see Appendix A for the definition of certain variables):

$$V_i(t) \equiv \iint h_i dx dy, \quad (1)$$

$$e_i(t) \equiv \iint \frac{1}{2} q_i^2 h_i dx dy, \quad (2)$$

$$\omega_i(t) \equiv \iint \zeta_i dx dy. \quad (3)$$

The mechanical energy of the basin ($K + P$) is defined as:

$$K \equiv \sum_i \iint \frac{1}{2} \rho_i h_i (u_i^2 + v_i^2) dx dy, \quad (4)$$

$$P \equiv \iint \frac{1}{2} \rho_1 g \eta_1^2 dx dy, \quad (5)$$

where K is the kinetic energy and P the barotropic potential energy (the energy contained in the baroclinic mode is neglected for simplicity). I further define the volume anomaly, the potential

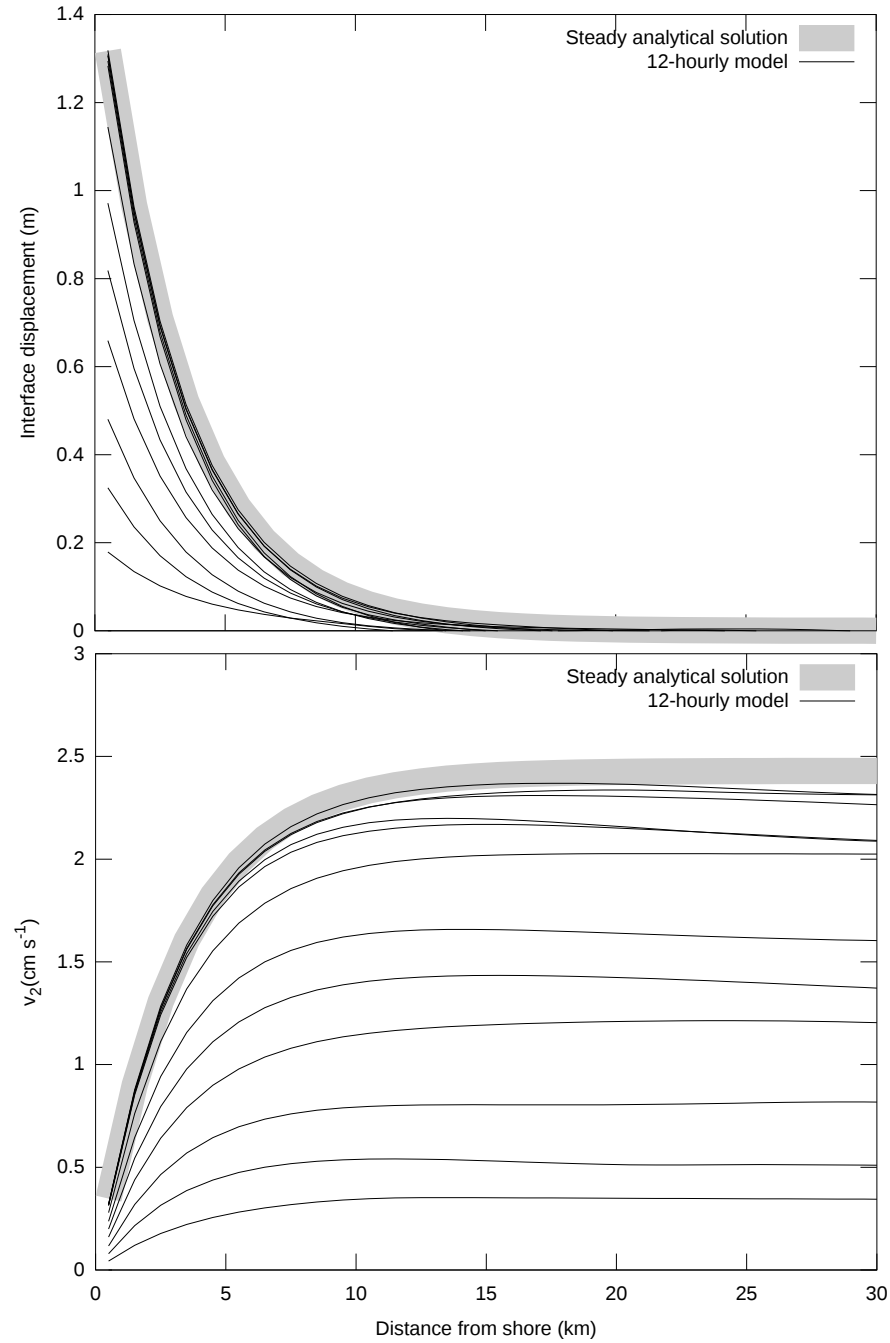


Figure 3: Comparison between analytical asymptotic values (thick gray line; Millot and Crépon 1981, their Eqs. 4.8 and 4.16) and model outputs sampled every 12 hours in test-case of upwelling driven by seaward wind.

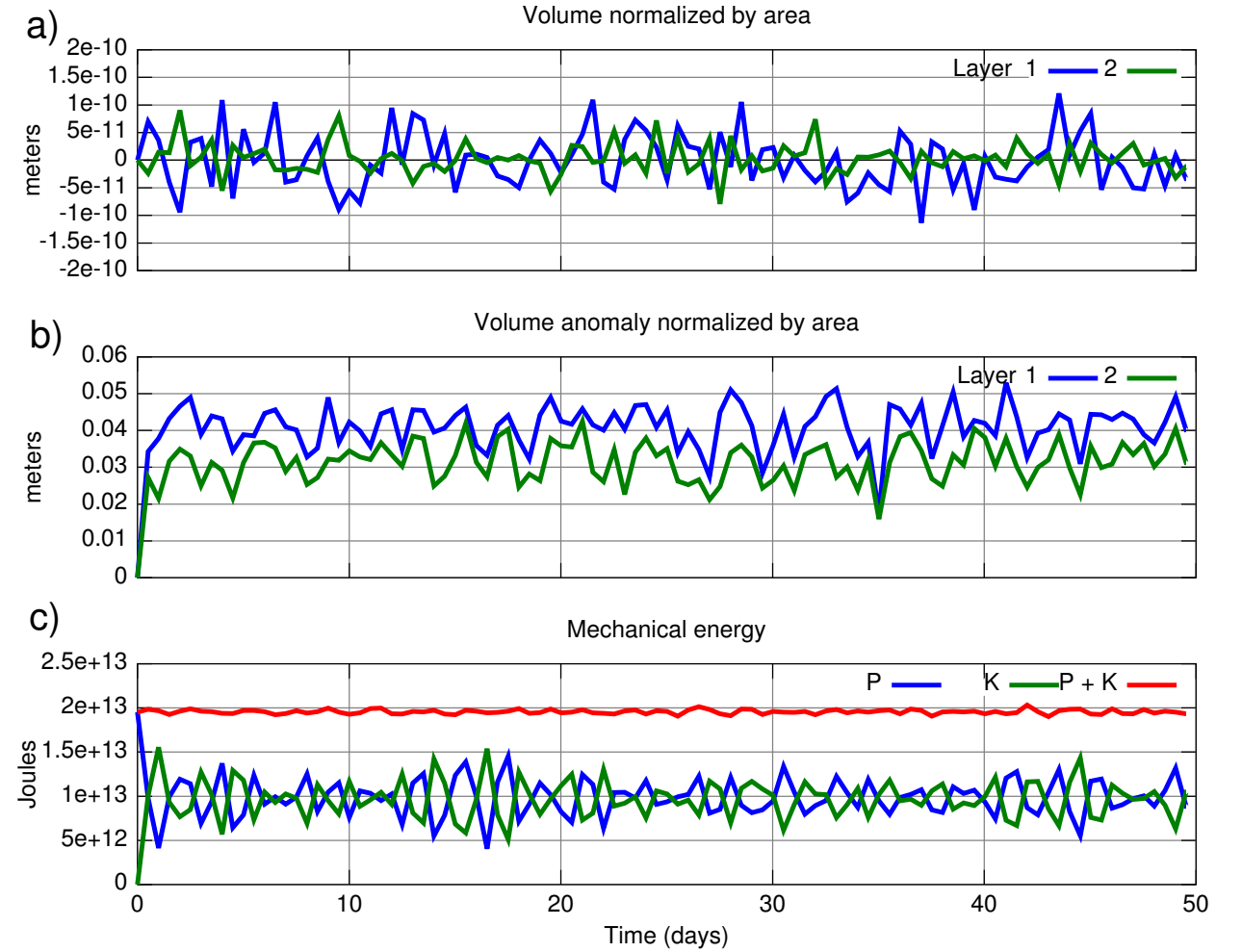


Figure 4: Numerical errors in volume and mechanical energy. In (a,b) the curves are referenced to their value at $t = 0$ and normalized by the area of the basin $\iint dx dy$. See Test-case 3 for definition of quantities. The results were obtained with the ‘standard’ forward-backward scheme (Appendix E).

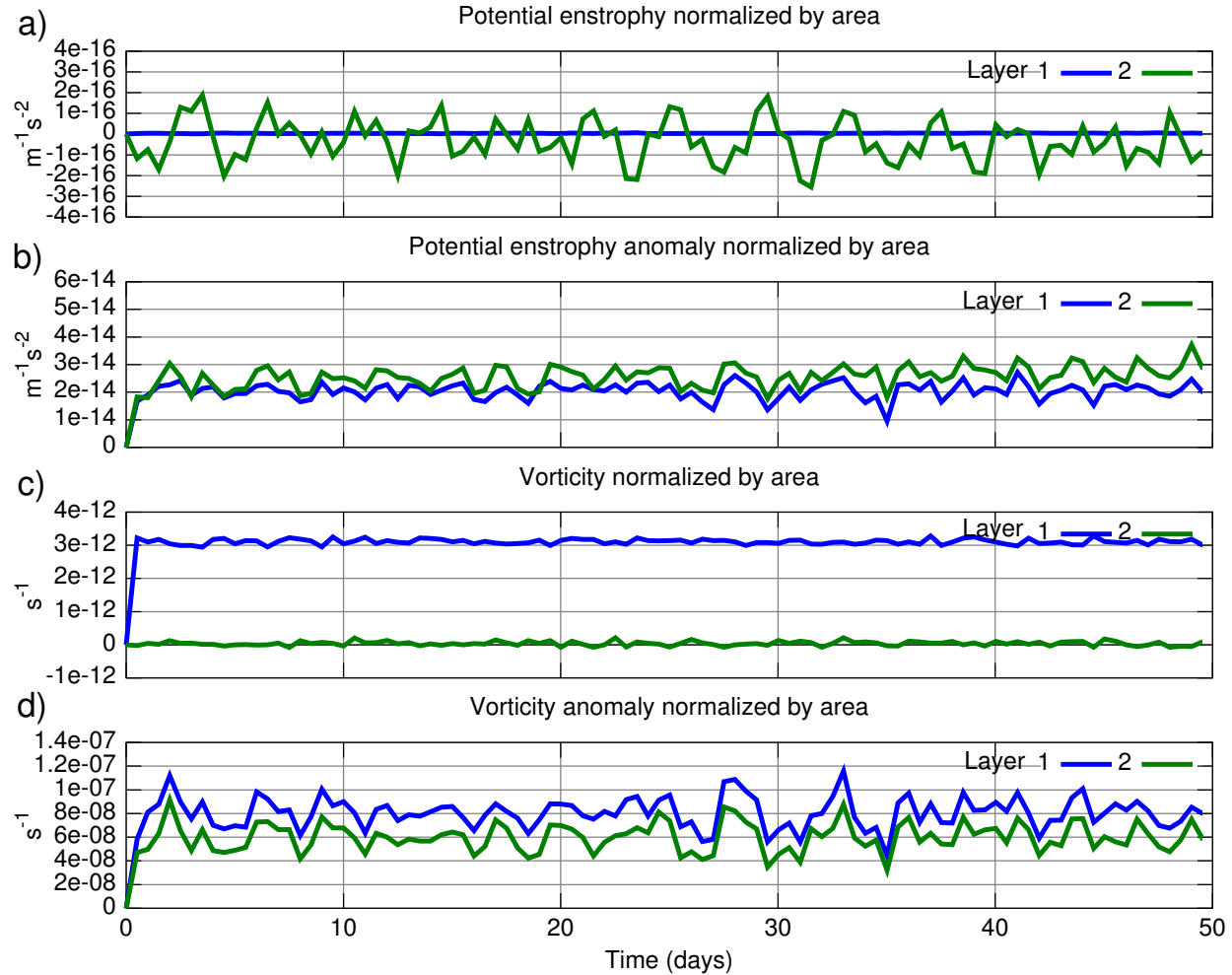


Figure 5: Numerical errors in potential enstrophy and vorticity. In (a,b,c,d) the curves are referenced to their value at $t = 0$ and normalized by the area of the basin $\iint dx dy$. See Test-case 3 for definition of quantities. The results were obtained with the ‘standard’ forward-backward scheme (Appendix E).

enstrophy anomaly, and the vorticity anomaly as:

$$V'_i(t) \equiv \iint |h_i - h_i(x, y, t = 0)| dx dy, \quad (6)$$

$$e'_i(t) \equiv \iint \left| \frac{1}{2} q_i^2 h_i - \frac{1}{2} q_i^2(x, y, t = 0) h_i(x, y, t = 0) \right| dx dy, \quad (7)$$

$$\omega'_i(t) \equiv \iint |\omega_i| dx dy. \quad (8)$$

The anomalies are meant to represent a typical deviation from the mean state. We can think of the anomalies as the ‘signal’ resolved by the model, while round-off and discretization errors represent a background noise. The accuracy of the model can be gauged by the signal-to-noise ratio.

The test-case described in the file `testcases/conservation.m` represents a semi-realistic scenario where round-off and discretization errors are tracked during the collapse of a Gaussian-shaped mound of water. The basin is double-periodic, it has a two-layer stratification, and it includes a seamount contained within the deep layer. The vertical stratification is sufficiently strong that the baroclinic Rossby radius of deformation is resolved with a coarse mesh of 10 km. The basin is initially at rest and the collapse of the surface mound excites several normal modes of the basin (e.g., Rao, 1966). For the purpose of this test-case the integration time is limited to 50 days. The model is executed with no explicit dissipation and no forcing.

The average layer thickness (i.e. the volume of the layer normalized by the area of the basin) oscillates randomly and stays within 10^{-10} m of its initial value during the 50 days of simulation (Fig. 4a). This small numerical error is eight orders of magnitude smaller than the ‘signal’ resolved by the model ($O(10^{-2}$ m), Fig. 4b). The mechanical energy ($K + P$) displays oscillations $O(3\%)$ (Fig. 4c). These oscillations are primarily due to the time-stepping scheme and their magnitude decrease if the time-step Δt is reduced (Appendix E). Note that Fig. 4c was obtained with the standard forward-backward scheme (`g_fb = 0` in file `shared_mod.f95`; see Appendix E). The ‘generalized’ forward-backward scheme produces similar results except that the high-frequency wave motion is gradually damped (e -folding time of 75 days). The numerical dissipation is inherent to the generalized scheme and exacerbated by the coarse resolution of the test-case (Appendix E).

The potential enstrophy oscillates randomly with an amplitude $O(10^{-16} m^{-1} s^{-2})$ (Fig. 5a). This small numerical error is two orders of magnitude smaller than the typical oscillation resolved by the model (Fig. 5b). The numerical error for the vorticity (Fig. 5c) is five orders of magnitude smaller than the typical variations resolved by the model (Fig. 5d).

The option `outc = 1` in the file `testcases/conservation.m` increases the height of the seamount so that it intersects the interface separating the two layers. The results are generally

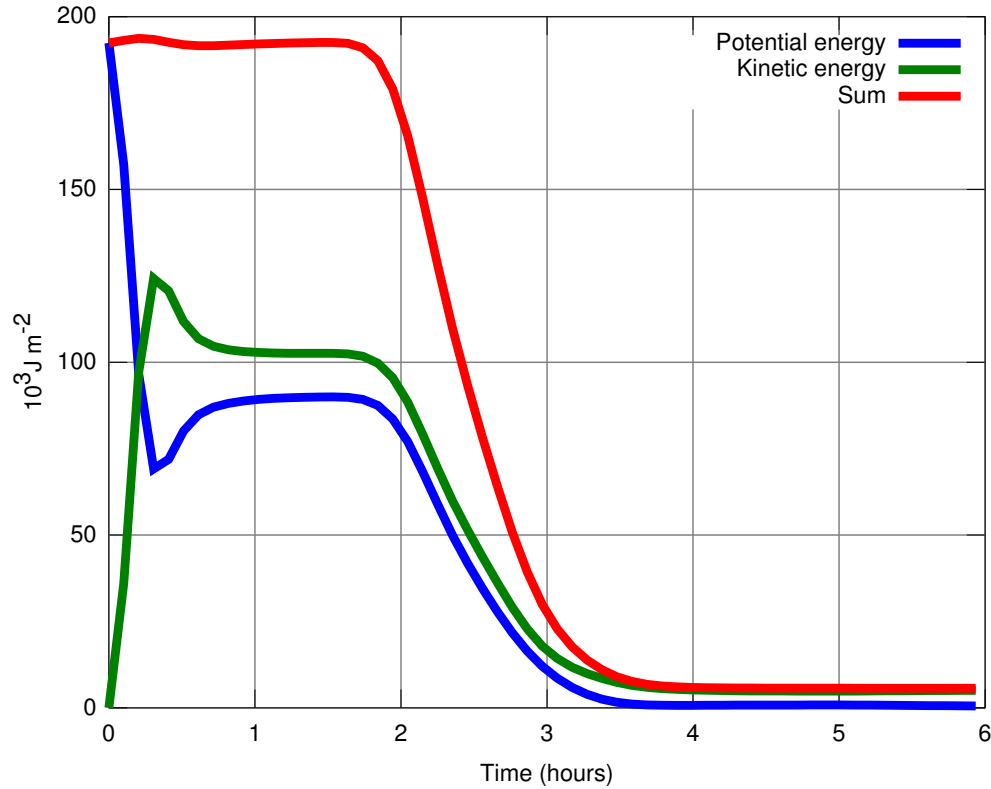


Figure 6: Absorption of the outward-propagating wavefront by the sponge zones. The residual energy corresponds to the small geostrophically-balanced mound (e.g., Lavelle and Thacker, 2008, their Fig. 3). See Test-case 4 (Wave sponge at model boundaries).

similar to the case without outcrop since the scheme of Salmon (2002) preserves the important conservation laws.

Test-case 4: Wave sponge at model boundaries

Open boundaries are special boundaries through which signals generated inside the model domain can propagate freely, as if the domain would extend to infinity. The open boundary scheme implemented in beom is called the flow relaxation scheme (Davies, 1976). In this approach, the model domain is extended with ‘sponge zones’ in which the outward-propagating signals are gradually absorbed. The sponge is typically implemented as an additional term on the right hand

side of the prognostic equations (we consider a one-dimensional space for simplicity):

$$\frac{\partial \phi}{\partial t} = \dots + \frac{\phi_{\text{ext}} - \phi}{T_{\text{relax}}}, \quad (9)$$

where $\phi(x, t)$ is a prognostic variable, ϕ_{ext} its (estimated) value outside of the model domain, and $T_{\text{relax}}(x)$ a timescale for the relaxation. Prognostic variables are not conserved within the sponge zones because of this additional term. The choice of the variables to be relaxed, and the shape of $T_{\text{relax}}(x)$ within the sponge zone, are active topics of research (see Lavelle and Thacker, 2008, for a review) and they depend on the problem at hand. In the case of a flow in near geostrophic balance, a simple ‘no-gradient’ boundary condition with very weak relaxation ($T_{\text{relax}} \geq \text{month}$) is often acceptable (e.g., Williams et al., 2001). A more general case that includes large barotropic waves generated inside the domain will require a strong sponge at one of the open boundaries ($T_{\text{relax}} \rightarrow \Delta x / c_{\text{ext}}$, where c_{ext} is the phase speed of the waves).

In this test-case I revisit the collapse of the Gaussian-shaped mound of water by replacing the periodic boundaries with sponge zones that surround the domain. The wavefront gradually propagates outward and is ultimately absorbed within the sponge zones. The final state in the interior domain is an ocean at rest except for a weak geostrophically-balanced gyre where the mound was located. Following Lavelle and Thacker (2008), all prognostic variables are relaxed within the sponge zones except tangential velocities. Modave et al. (2010) suggest a sponge zone with a width of $n \sim 15$ grid points in which the relaxation timescale varies as (their Eq. 29):

$$T_{\text{relax}}^{-1}(i) = \frac{c_{\text{ext}}}{n \Delta l} \frac{i}{n - i}, \quad (10)$$

where c_{ext} is the phase speed of surface gravity waves, Δl is the mesh size, and i is the position within the sponge zone in unit of grid points. The file `testcases/wave_sponge.m` creates the input files and illustrates the efficiency of the sponge (Fig. 6).

Test-case 5: Equatorial soliton

The Equatorial soliton is a solitary wave that propagates westward at fixed speed along the Equator. The wave depends on non-linearities to preserve its shape and so this test-case is often used to evaluate ocean models. The model domain corresponds to a periodic zonal channel on a β -plane and it is initialized with analytical fields for η, u, v (Lavelle and Thacker, 2008, their Eqs. 21–23).

The model is integrated for two months during which the soliton travels the whole channel and returns to its initial position. The file `testcases/soliton.m` creates the input files and compares the wave at initial and final times (Fig. 7). The test-case does not require any dissipation if the ‘generalized’ forward-backward scheme is used (`dvis = 0.` and `g_fb = 1.` in file

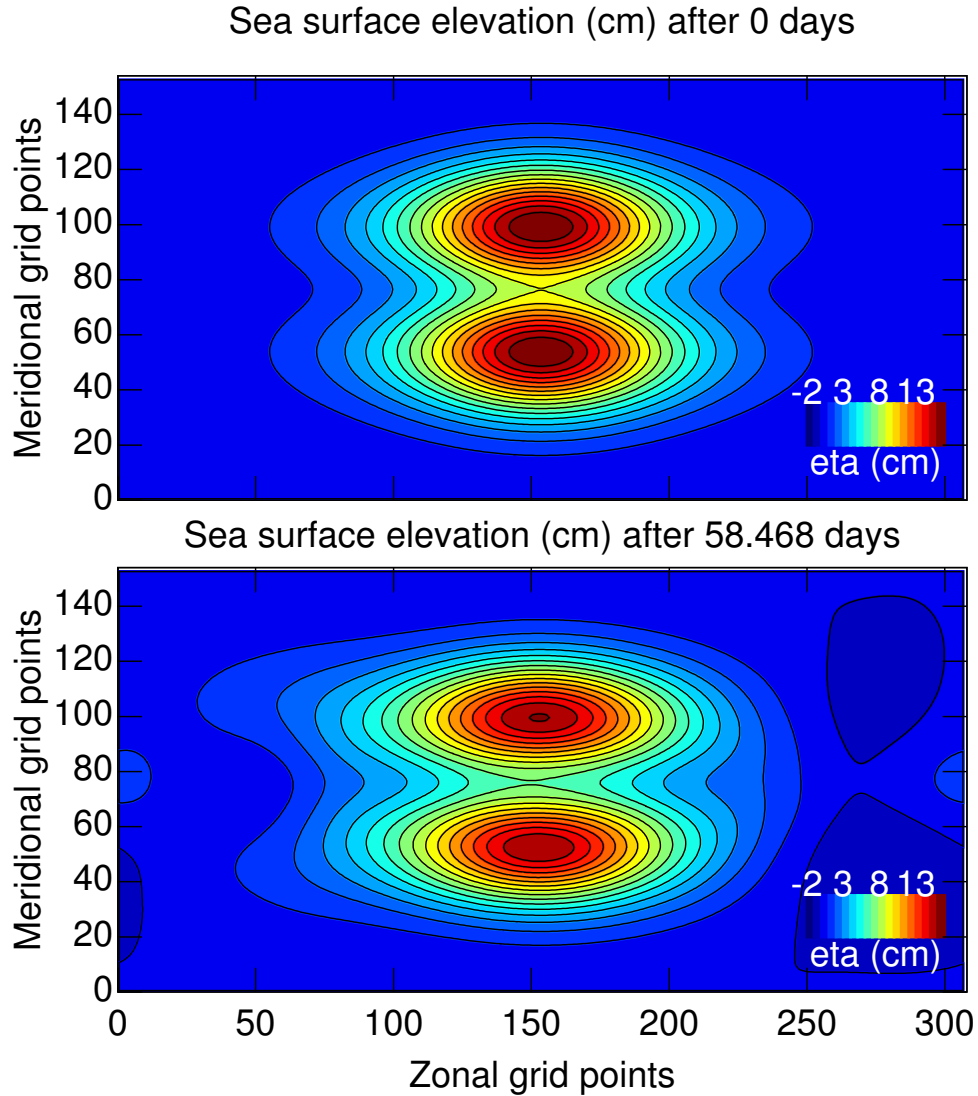


Figure 7: See Test-case on Equatorial soliton.

shared_mod.f95; see Appendix E). The ‘standard’ scheme cannot handle the test-case without dissipation.

Test-case 6: Barotropic instability and 2-D turbulence

The test-case `testcases/unstable_jet.m` considers the case of two zonal jets flowing in opposite directions and initially in geostrophic balance with a Gaussian-shaped ridge of water along x . The large relative vorticity of the jets makes them dynamically unstable. A perturbation corresponding to sinusoidal corrugations at the bottom is used to trigger the growth of the wave. The domain is a double-periodic rectangle with uniform planetary rotation.

The two jets start to meander after $O(1 \text{ day})$ and rapidly evolve into two-dimensional turbulence (Fig. 8). The test-case checks that the periodic boundary conditions, momentum advection, and non-linear viscosity are properly working.

Test-case 7: Stratified rotating flow over topography

Baines and Leonard (1989) provide an analytical solution for the case of a stratified ($1\frac{1}{2}$ layer) rotating flow over a small semi-infinite ridge. Their analytical solution corresponds to the steady state (valid for times $t \gg f^{-1}$) and is analog to the atmospheric flow that develops above a mountain range. They neglect vertical shear and assume that both layers share the same initial speed U_0 . The analytical solution neglects derivatives in the cross-stream direction (y) and assumes that $\mathbf{u} \cdot \nabla \mathbf{u}$ can be linearized as $U_0 \partial u / \partial x$.

The file `testcases/baines_ridge.m` reproduces the setup of Baines by prescribing a two-layer flow along x with periodic conditions along y . The $1\frac{1}{2}$ layer dynamics are mimicked by having the upper layer 10 times thicker than the lower layer. The height of the ridge is $1/10$ of the lower layer thickness since the analytical solution is only valid for small topography. Sponges are placed upstream and downstream of the ridge to absorb the waves produced during the transient adjustment. The Coriolis term $q\mathbf{e}_3 \times h\mathbf{u}$ is balanced by a constant and uniform body force applied in the cross-stream (y) direction. Fig. 9 compares the analytical and modeled interface for $U_0 = 1.2 \text{ m s}^{-1}$ and $\Delta\rho = 5 \text{ kg m}^{-3}$ (corresponding to a Froude number $F_0 = 1.7$) after 10 days of simulation.

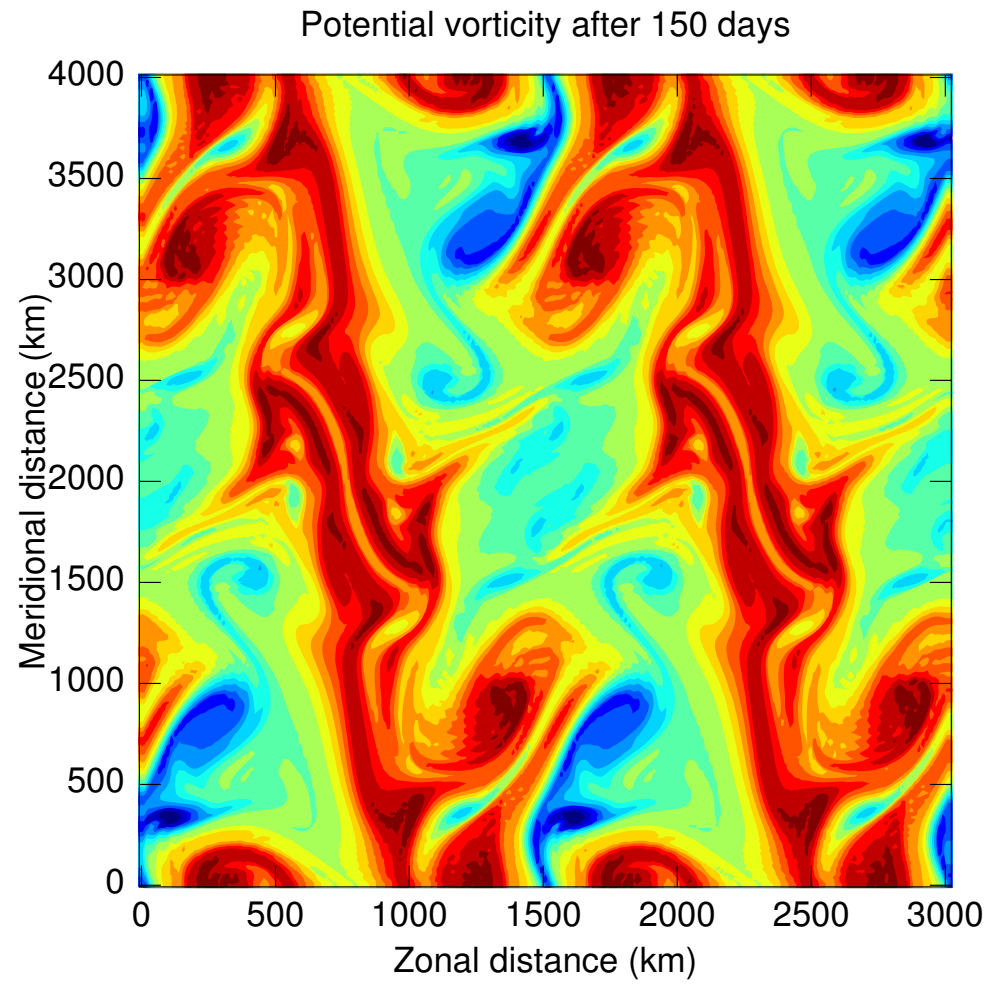


Figure 8: Two-dimensional turbulence in the test-case of barotropic instability.

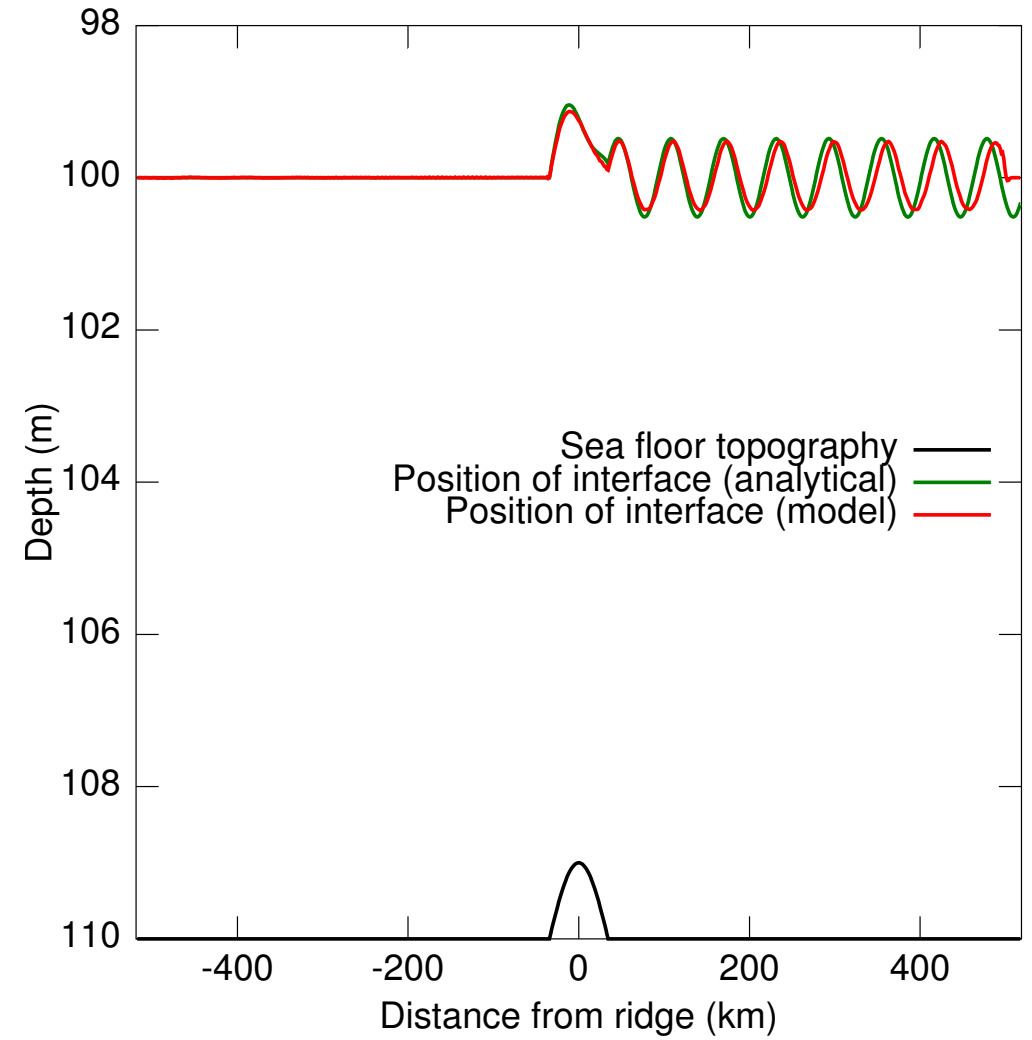


Figure 9: Test-case for stratified rotating flow over topography.

Test-case 8: Isopycnal outcropping

An isopycnal surface becomes discontinuous (undefined) wherever it intersects the sea floor, the sea surface, or another isopycnal. Bretherton (1966) interprets such discontinuities as the layer being still active but infinitesimally thin (a δ -sheet). The same concept is referred to as ‘mass-less layers’ by Hsu and Arakawa (1990). From a modelling point of view, the difficulty is exchanged from handling spatial discontinuities to handling very thin thickness values (see Appendix C).

The test-case `testcases/outcrop_seamount.m` considers the case of a closed square basin with sloping walls and a seamount at its center (Fig. 10). The model starts from rest and has five layers that intersect both the seamount and the sloppy walls. The test-case verifies that the numerical scheme (Appendix C) is stable despite the steep isopycnal slopes, i.e. that it does not generate artificial currents that grow over time. Such test-case is commonly used to evaluate pressure-gradient errors in σ -coordinate models.

The test-case is an example of simple isopycnal outcropping with no wetting-drying. These two concepts are compared in Fig. 11. In simple isopycnal outcropping, the shoreline corresponds to a vertical wall and the minimum depth of the basin is such that the upper layer thickness h_1 is always $\gg h_{\text{sal}}$. Here h_{sal} is the ‘Salmon thickness’, a problem-dependent parameter to be determined by the user prior to a calculation (see Appendix C). For example in Fig. 11a the minimum depth of the basin is set to:

$$H(i = 1) = 10 h_{\text{sal}} + (N - 1) h_{\text{sal}}, \quad (11)$$

where $(N - 1) h_{\text{sal}}$ was included to take into account the vertical space occupied by layers 2–4 (outcropped layers have a thickness $h \sim h_{\text{sal}}$). In cases with wetting-drying there is no restriction on the minimum depth of the basin (Fig. 11b).

The Salmon thickness essentially determines the thickness of the outcropped layers. The choice of h_{sal} depends on the horizontal mesh size, the topographic slope and the stratification. In cases with wetting-drying, Salmon suggests (Salmon, 2002, page 625):

$$h_{\text{sal}} > \left| \frac{\partial H}{\partial x} \right| \Delta x, \quad (12)$$

where $|\partial H / \partial x|$ is the bottom slope (Fig. 11b). Equation 12 ensures that the dynamical balance of the top layer changes smoothly from ‘wet’ to ‘dry’ grid points. Salmon shows that violating this criterion leads to solutions that are well-behaved but increasingly noisy as h_{sal} is reduced. Experience with realistic topography and dynamic viscosity confirms that Equation 12 can be sometimes violated by as much as two orders of magnitude at certain grid points and still lead to well-behaved fields.

Cases with simple outcropping allow for considerably thinner values since the free surface is

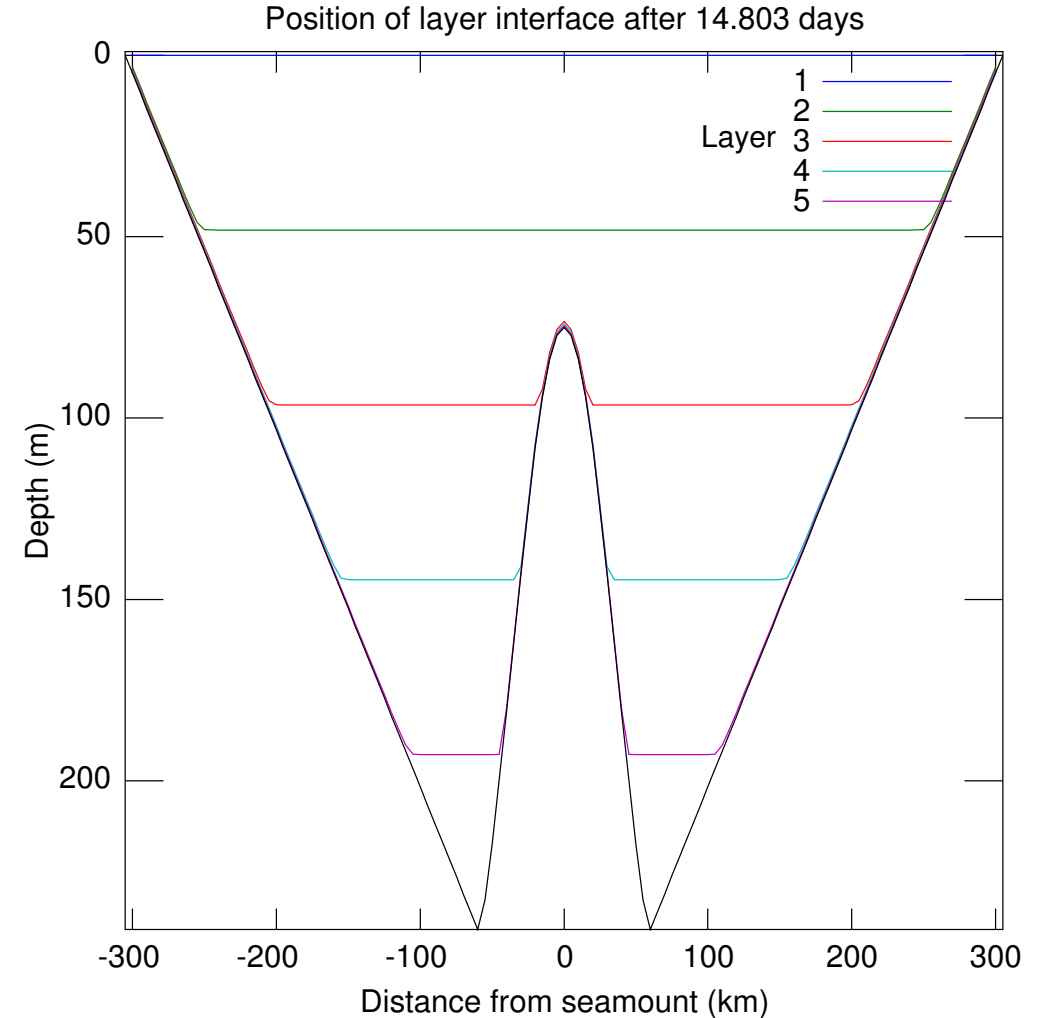
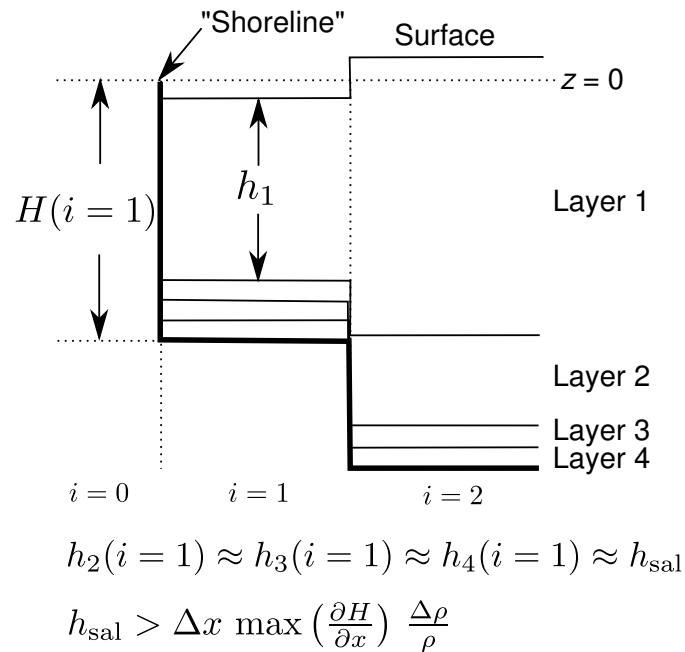


Figure 10: Position of the layer interfaces in the test-case of simple isopycnal outcropping.

a) Outcrop, no wetting-drying



b) Outcrop + wetting-drying

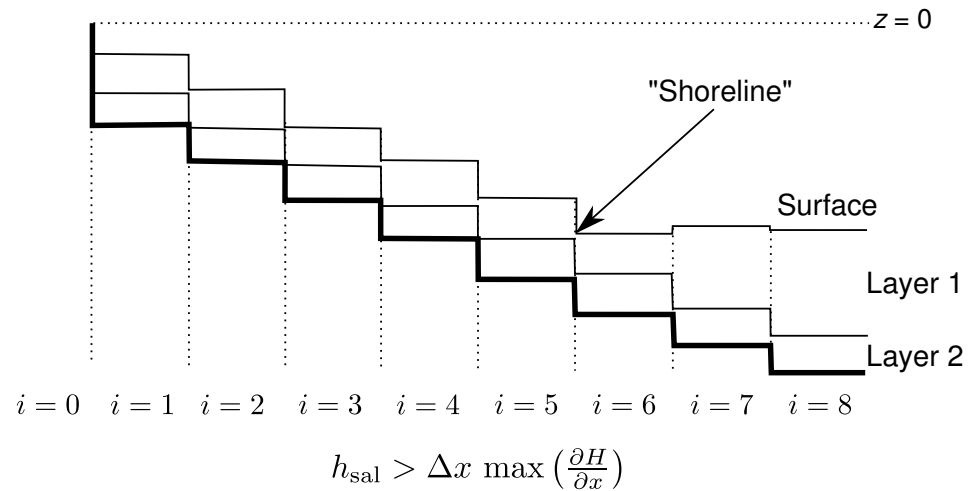


Figure 11: Comparison between simple isopycnal outcropping and wetting-drying. The scales are exaggerated.

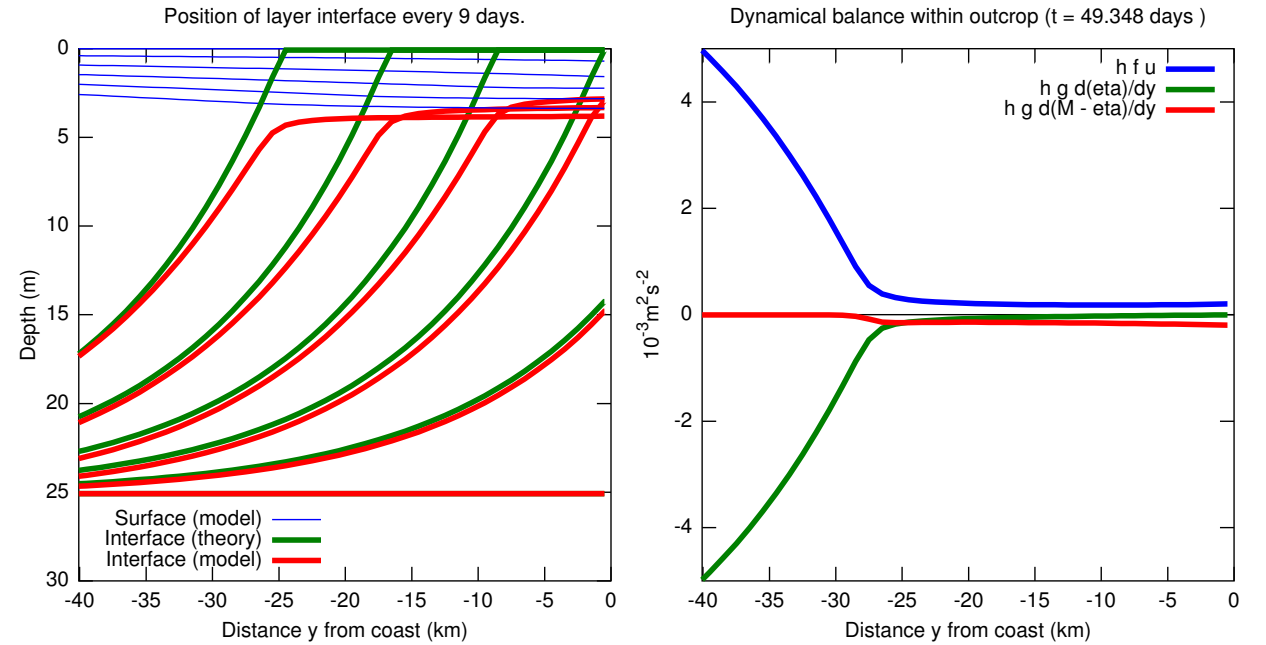


Figure 12: (left) Migration of the surface front during the upwelling event (see Test-case 10). (right) Dynamical balance within the region of outcrop.

not involved:

$$h_{\text{sal}} > \max \left| \frac{\partial H}{\partial x} \right| \Delta x \frac{\Delta \rho}{\rho}, \quad (13)$$

where $\max |\partial H / \partial x|$ is the maximum bottom slope over the model domain (Fig. 11a).

In test-case `testcases/outcrop_seamount.m` the Salmon thickness is set to $h_{\text{sal}} = \max |\partial H / \partial x| \Delta x (\rho_2 - \rho_1) / \rho_2 = 14 \text{ cm}$ and the layers are a few cm thick in the areas of outcrop (Fig. 10). The maximum spurious currents are $O(10^{-5} \text{ m s}^{-1})$. The kinetic energy of the model domain remains stable over the 15 days of simulation without any recourse to dissipation (the viscosity `dvis` is zero). Note that the energy-conserving scheme of Sadourny (1975, his Eq. 3) is vital for this result (see Appendix F).

Test-case 9: Coastal upwelling with outcrop

Morel et al. (2006) provide an analytical solution for a coastal upwelling where the pycnocline outcrops the sea surface. Their setup is a flat basin with a rigid lid, a two-layer stratification, and a

constant and uniform longshore wind applied impulsively at $t = 0$. Derivatives in the along-shore direction are assumed zero and non-linear terms are neglected.

The file `testcases/morel_upwelling.m` reproduces the setup of Morel et al. (2006) as closely as possible. The wind forcing is mimicked with a constant body force applied on the upper layer (both before and after the pycnocline reaches the surface). No force is applied to the lower layer (see their Eq. 25). One grid point and periodic conditions are used in the along-shore direction (x). The offshore boundary condition is a solid wall located one barotropic Rossby radius away from the upwelling.

The position y of the model ‘front’ (i.e. where the interface intersects the surface) closely follows the theory (Fig. 12). Differences between the two are essentially due to the rigid lid approximation (the surface slopes downward in both model and reality) and to the finite value of h_{sal} (0.5 m). This model-theory comparison illustrates the accuracy and robustness of the scheme.

It is instructive to examine the dynamical balance within the outcropped areas (i.e. in the upper layer and shoreward of the front). With the periodic condition in the alongshore direction, the depth-integrated longshore momentum equation reduces to:

$$h \frac{\partial u}{\partial t} \approx h f v + h b_x, \quad (14)$$

where b_x represents the ‘wind stress’ (here a constant and uniform body force). In the across-shore direction the momentum equation is:

$$h \frac{\partial v}{\partial t} \approx 0 \approx -h f u - h g \frac{\partial \eta_1}{\partial y} - h \frac{\partial}{\partial y} (M - g \eta_1), \quad (15)$$

where $M - g \eta_1$ represents the artificial term suggested by Salmon (see Eq. 34). Outside the outcrop area the dynamical balance is essentially geostrophic (Fig. 12). Within the outcrop area, the surface becomes flat ($\partial \eta_1 / \partial y \rightarrow 0$, green curve) and geostrophic velocities $f \partial \eta_1 / \partial y$ vanish. The actual longshore transport $h u$ is also close to zero (blue curve) and balanced by a smooth and weak Salmon term (red curve).

Test-case 10: Wetting and Drying

An important application of Salmon’s method (Appendix C) are cases where the land/ocean boundary is free to move horizontally (‘wetting and drying’). In order to accommodate such a displacement of the shoreline, the model domain must be extended horizontally and vertically to include the land areas that will be potentially flooded (Figs. 11,13). In a realistic case, one would use a topographic database that includes both the water depth and the land elevation. Then,

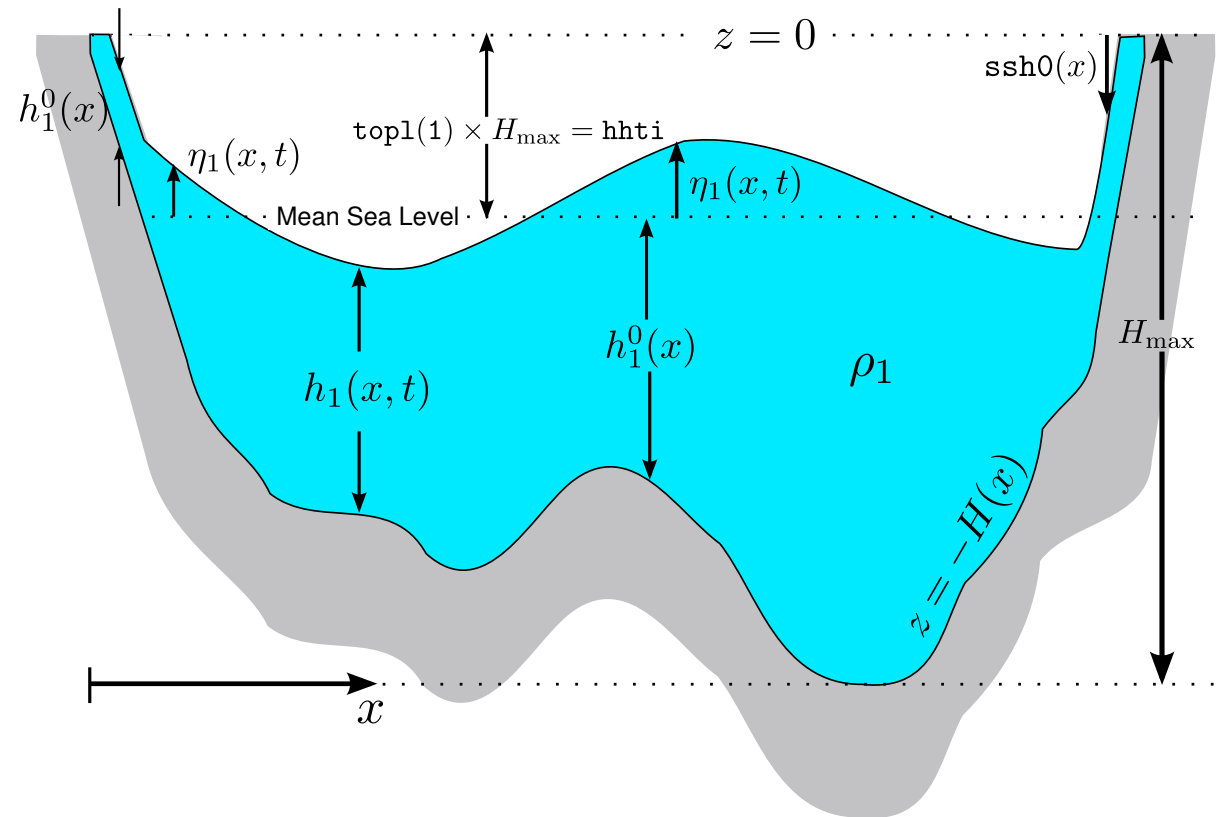


Figure 13: Basin configuration in cases with wetting-drying. Scales are exaggerated. The figure shows a case with no stratification for simplicity.

we define a maximum value for the height of the flood (in the tidal literature this would correspond to the ‘Highest High Tide’, `hhti`). The model domain is then extended to include all the land areas whose elevation is $\leq \text{hhti}$. Finally, the top of the upper model layer (`topl(1)`) is lowered to coincide with the mean sea level (Fig. 13). The model shoreline corresponds to the point where the layer thickness falls below the Salmon thickness h_{sal} (Appendix C).

The file `testcases/carrier_beach.m` provides an example of ‘wetting and drying’. In contrast with all the earlier test-cases, the undisturbed sea surface $\text{ssh0}(x)$ does not coincide with the geopotential surface $z = 0$ (Figs. 11,13):

$$\text{ssh0}(x) \equiv \sum_{i=1}^{\text{nlay}} h_i^0(x) - H(x) \approx \begin{cases} -\text{hhti} & \text{away from coastlines,} \\ -H(x) + h_{\text{sal}} & \text{over ‘dry’ areas,} \end{cases} \quad (16)$$

and the top of the (undisturbed) upper layer is positioned at $\text{topl}(1) = \text{hhti}/H_{\max}$. In this case $\text{hhti} = 0.9$ meters.

Carrier and Greenspan (1958) provide an analytical solution for the initial-value problem of a mound of water released in front of the shoreline (Fig. 14a). The mound has an initial height of 30 cm and the bottom has a gentle slope of $|\partial H/\partial x| = 10^{-3}$. The Salmon thickness is set to a very small value $h_{\text{sal}} = 0.2 |\partial H/\partial x| \Delta x = 1.2$ cm (see Eq. 12) and the model is ran with no dissipation at all. The mound collapses without breaking and forces the shoreline to migrate horizontally and vertically along the sloping beach. Waves radiated seaward are absorbed within a wave sponge. Fig. 14b shows that the model accurately reproduces the oscillation of the shoreline.

Test-case 11: Upwelling with mixed open boundary conditions

In this test-case we consider a rectangular domain with a straight coastline on one side and open boundaries on the three other sides. An ideal open boundary condition (obc) would simultaneously allow: (1) the propagation of outgoing perturbations (waves, drifting eddies) and (2) the prescription of a given field (velocity, stratification, sea surface height). This dual requirement is often referred to as a passive-active obc (e.g., Palma and Matano, 1998; Marchesiello et al., 2001).

The test-case combines different types of obc at the edges of the model domain to fulfill the two requirements (see the file `testcases/mixed_open_bc.m`). The first obc is a ‘no-gradient’ condition;

$$\frac{\partial \psi}{\partial x} = 0, \quad (17)$$

where x is the direction normal to the boundary and ψ represents the layers’ thickness h and the two velocity components u, v . This obc is simply an extrapolation of the solution calculated inside the computational domain. It allows for the development of a geostrophic flow normal to the open boundary (Chapman, 1985). Its main drawback is that it reflects outgoing waves. The ‘no-gradient’ obc is automatically applied at all open boundaries that are not periodic.

The default behavior of the model is to always complement the ‘no-gradient’ obc with a relaxation of the type:

$$\frac{\partial \psi}{\partial t} = \dots + \frac{\psi_{\text{ext}} - \psi}{T_{\text{relax}}}, \quad (18)$$

where \dots represents the right hand side of the prognostic equations, T_{relax} is the relaxation time-scale, and ψ_{ext} is a prescribed state (sometimes called the external field). In the test-case this prescribed state is simply the state of rest to which the model automatically adds an Ekman

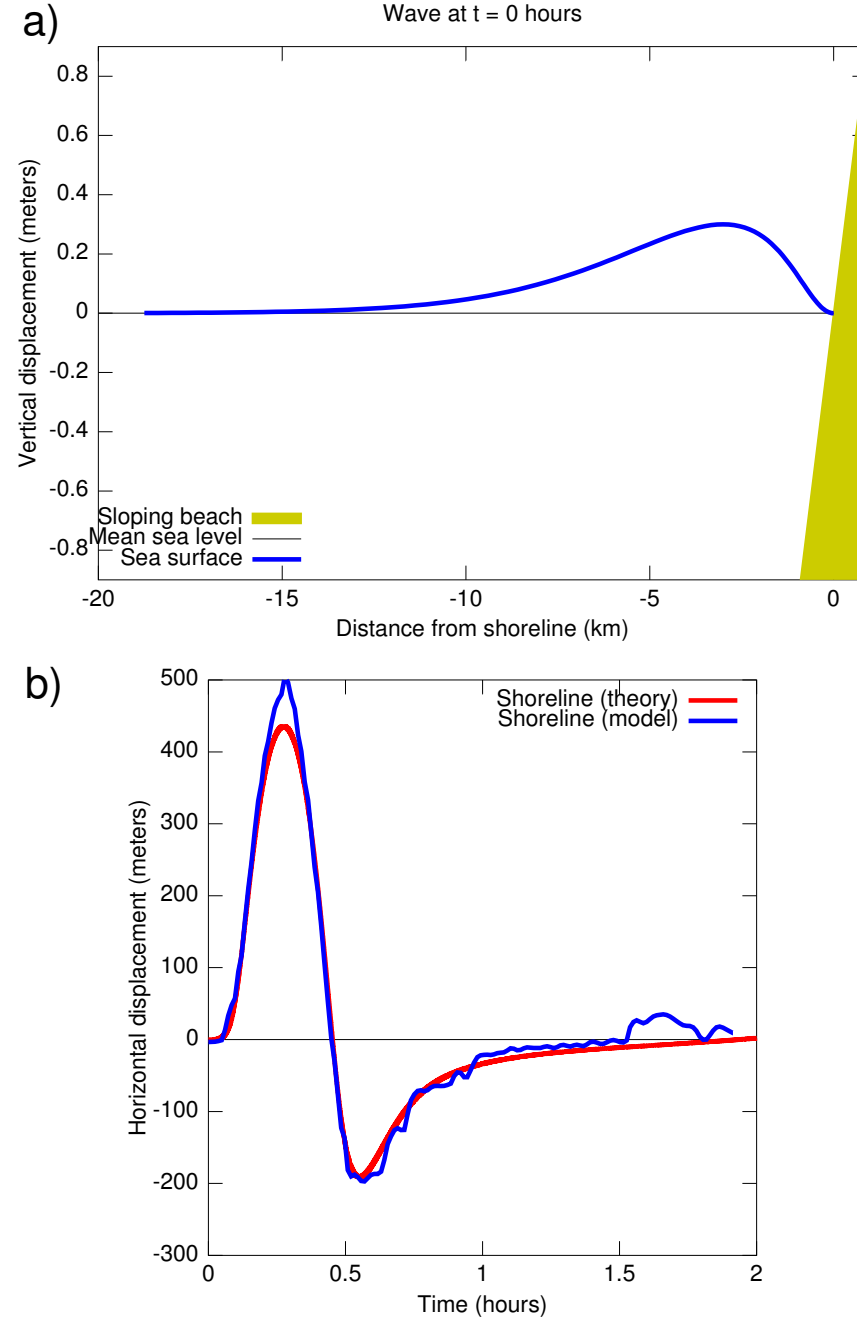


Figure 14: Test-case for wetting and drying on a sloping beach. (a) Initial displacement $\eta_1(x, t = 0)$ of the sea surface. (b) Model-theory comparison for the horizontal displacement of the shoreline.

component in the upper layer ($i = 1$):

$$\psi_{\text{ext}} : \quad \eta_i = 0, \quad \mathbf{u}_1 = -\mathbf{e}_3 \times \frac{\boldsymbol{\tau}_{\text{wind}}}{f \rho_0 h_1}, \quad \mathbf{u}_2 = 0. \quad (19)$$

The Ekman component ensures that wind-induced currents are free to flow across the open boundaries. The relaxation time-scale is set to a relatively long period at the northern and southern open boundaries ($T_{\text{relax}} \sim 1$ month) and its role is to prevent model drifts during long integrations.

The same configuration ('no-gradient' + relaxation) is applied at the offshore open boundary but with time-scales T_{relax} that are much shorter. The strong relaxation acts as a sponge and represents the main outlet for waves generated within the domain (see Test-case 4).

The test-case `testcases/mixed_open_bc.m` is similar to Test-case 2 except that the model domain is now three-dimensional and that non-periodic obc are used. The test-case verifies that the 3-D configuration with mixed obc (Eqs. 17,18,19) converges toward the same solution as the 2-D periodic case.

Test-case 12: Tidal flow over a ridge

The test-case `testcases/tide_ridge.m` simulates a tidal stream over a Gaussian ridge. The ridge has a height of $0.75 H_{\text{max}}$ and blocks the lower two layers. The stratification is represented by seven layers concentrated in the upper part of the water column. The domain is two-dimensional with a mesh size $dx = 100$ m and no rotation ($f = 0$).

The tidal forcing consists of an oscillating barotropic flow of the form:

$$u^{\text{tide}} = u_0 \cos(\phi_u - \omega t), \quad v^{\text{tide}} = v_0 \cos(\phi_v - \omega t), \quad \eta_1^{\text{tide}} = \eta_0 \cos(\phi_\eta - \omega t), \quad (20)$$

where $u_0 = 10 \text{ cm s}^{-1}$, $\phi_u = \pi/2$, $\omega = 12.141 \text{ rad days}^{-1}$, and $v_0 = \eta_0 = 0$. The forcing is applied with nudging at two open boundaries located on both sides of the ridge. A quadratic bottom friction is applied in a 10 m-thick boundary layer (`hbb1=10.`).

The oscillating flow and the stratification produce internal waves that radiate away from the ridge. The interface displacement associated with the waves is a few meters.

Test-case 13: Lock-exchange experiment

The test-case `testcases/lock_exchange.m` simulates the gravitational adjustment of two fluids initially separated by the center in a closed rectangular reservoir. The density contrast between

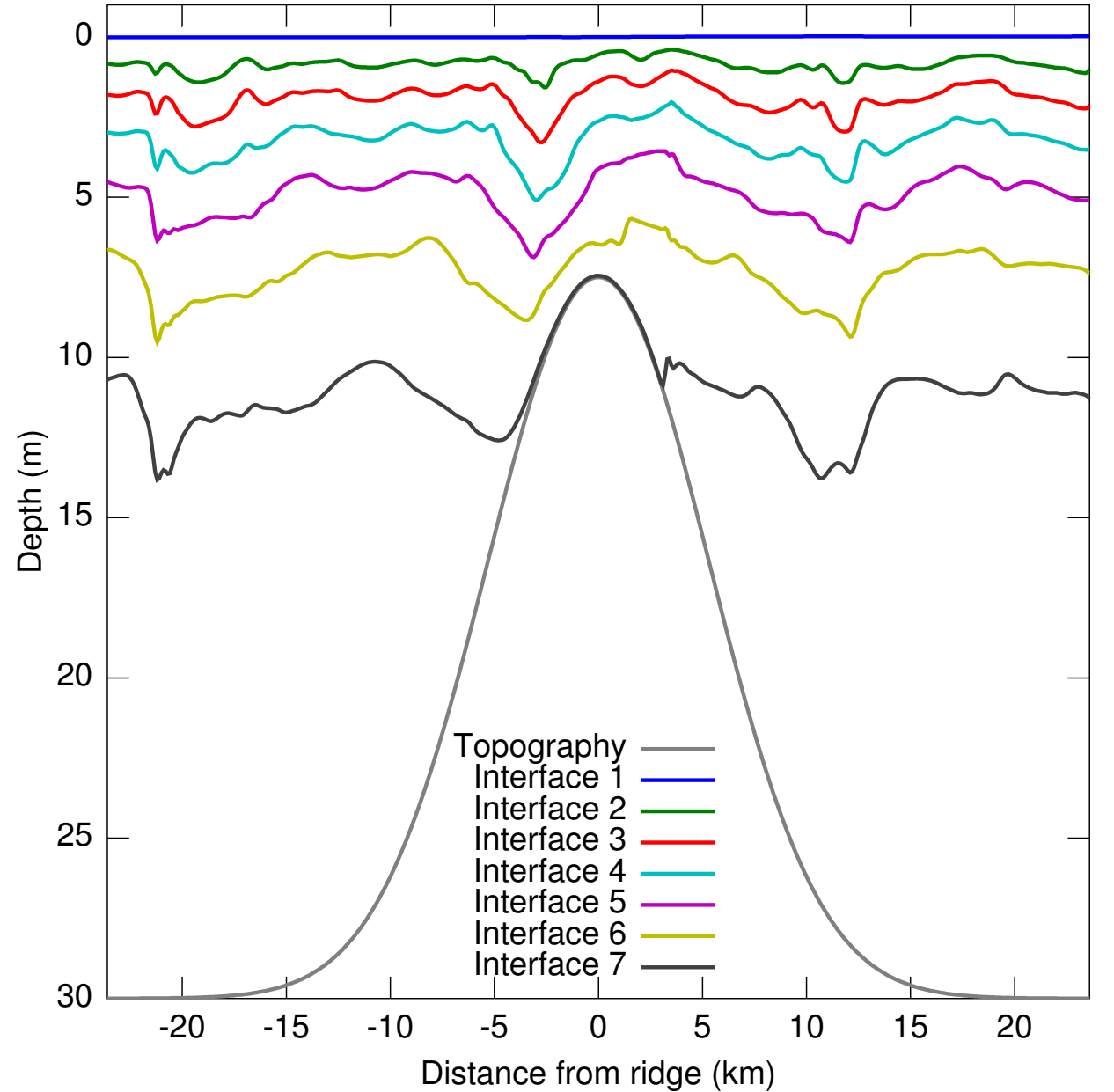


Figure 15: Test-case for tidal flow over a ridge.

the two fluids forces a slow re-distribution of mass and a transient oscillation with a fixed period. The initial density front propagates outward with a velocity of (e.g., Ilıcak et al., 2012):

$$\frac{1}{2} \sqrt{g H \Delta \rho / \rho_0} \quad (21)$$

where H is the reservoir depth and $\Delta \rho$ is the density contrast between the two fluids/layers. Figure 16 shows the position of the interface after 17 hours. The figure can be compared to the corresponding experiment in Shchepetkin (2015).

A Appendix: Model Equations

Under the assumptions of hydrostatic motion, incompressible flow and incompressible Boussinesq fluid, the momentum equation in isopycnal coordinates is (e.g., Hsu and Arakawa, 1990):

$$\frac{\partial \mathbf{u}}{\partial t} = -q \mathbf{e}_3 \times h \mathbf{u} - \nabla_h B + \frac{1}{\rho_0} \frac{\partial \boldsymbol{\tau}}{\partial z} + \nabla \cdot \overline{\mathbf{u}' \mathbf{u}'} \quad (22)$$

where \mathbf{u} is the layer-averaged horizontal velocity, h the layer thickness, $q \equiv (\zeta + f)/h$, ζ , f the potential, relative, and planetary vorticities, respectively. $B \equiv M + \mathbf{u} \cdot \mathbf{u}/2$ is the Bernoulli potential, M the Montgomery potential, and $\boldsymbol{\tau}$, $\overline{\mathbf{u}' \mathbf{u}'}$ the stresses from the Reynolds decomposition. In Fig. 1 I consider an example with two layers ($N = 2$) and decompose the thicknesses as:

$$h_2(x, y, t) = h_2^0(x, y) + \eta_2(x, y, t), \quad (23)$$

$$h_1(x, y, t) = h_1^0(x, y) + \eta_1(x, y, t) - \eta_2(x, y, t), \quad (24)$$

where h^0 is the layer thickness for the ocean at rest, and η is the interface displacement from its equilibrium position. The layers are allowed to become very thin (but non-zero) where they intersect the bottom, the surface, or other layers. The layer thickness evolves as:

$$\frac{\partial h_i}{\partial t} = -\nabla \cdot h_i \mathbf{u}_i. \quad (25)$$

The Montgomery potential is derived from the hydrostatic equation or (equivalently) from the basin potential energy V (Salmon, 2002):

$$V \equiv \iiint \rho g z \, dz \, dx \, dy, \quad (26)$$

$$\frac{M_i}{g}(x, y, t) = \frac{1}{\rho_i g} \frac{\delta V}{\delta h_i} = -H(x, y) + \sum_{k=1}^{N=2} \frac{\min(\rho_i, \rho_k)}{\rho_i} h_k(x, y, t) \quad (27)$$

$$= \eta_1(x, y, t) + \sum_{k=1}^{N=2} h_k^0(x, y) - H(x, y) - \sum_{k=1}^{i-1} \frac{\rho_i - \rho_k}{\rho_i} h_k(x, y, t). \quad (28)$$

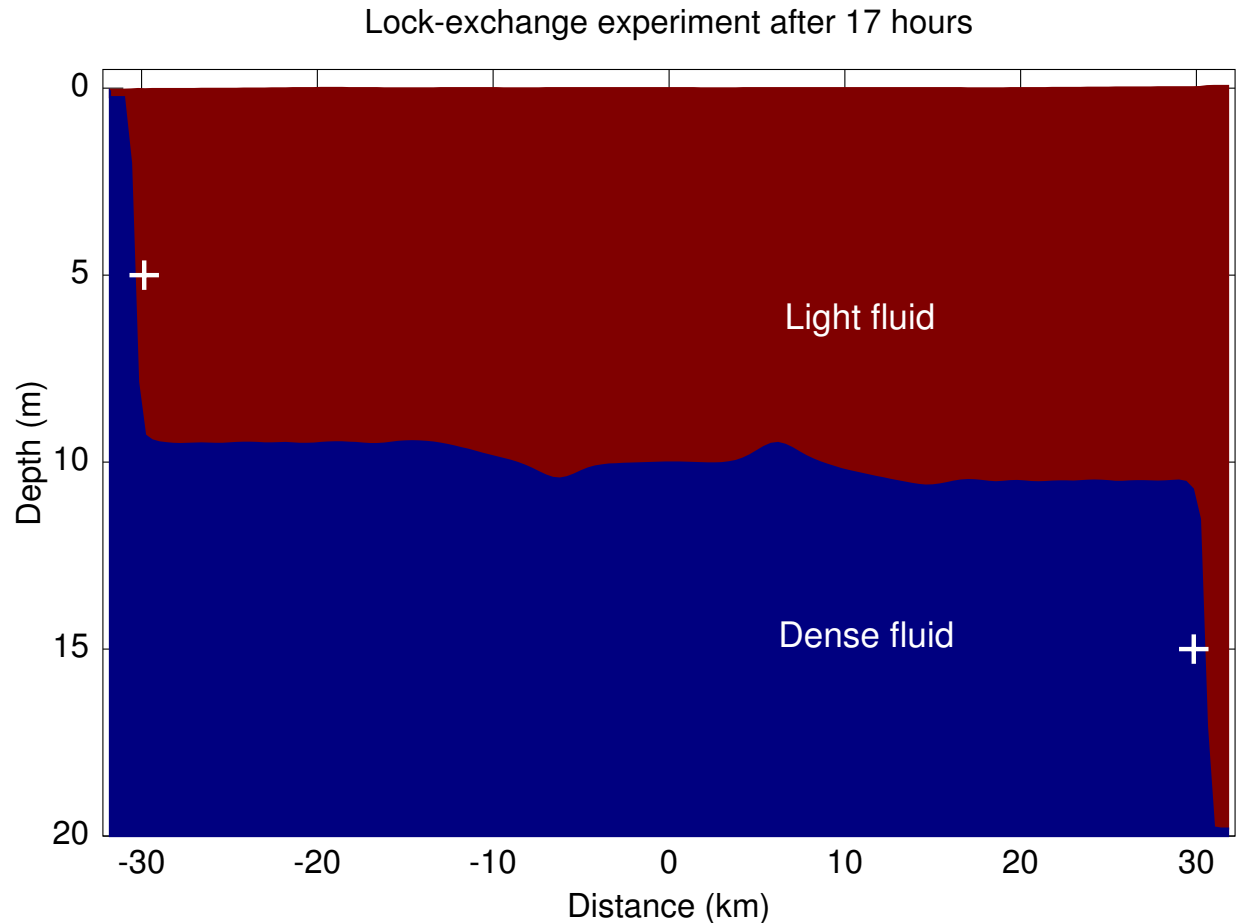


Figure 16: Test-case for the lock-exchange experiment. The symbol + marks the theoretical position of the front (Eq. 21)..

In simple cases the second and third terms of the last line cancel each other, and $\partial M_2 / \partial x$ reduces to the familiar form $\approx \partial / \partial x [g \eta_1 + g (\rho_2 - \rho_1) \eta_2 / \rho_2]$. Following Salmon (2002) an artificial term can be added to M to prevent h from reaching zero (see Appendix C). Eqs. 22,25 are discretized in space as in Sadourny (1975) and time-stepped with either the standard or generalized ‘forward-backward’ algorithm (Bleck and Smith, 1990; Shchepetkin and McWilliams, 2005, see Appendix E). Epipycnal momentum diffusion is parameterized as (Adcroft and Marshall, 1998):

$$\begin{aligned} \nabla \cdot \overline{\mathbf{u}'\mathbf{u}'} &\equiv \mathbf{e}_1 \left[\delta_x (\nu_{cc} D) - \delta_y (\nu_{ll} \zeta) \right] / \Delta l \\ &+ \mathbf{e}_2 \left[\delta_x (\nu_{ll} \zeta) + \delta_y (\nu_{ll} D) \right] / \Delta l \end{aligned} \quad (29)$$

where δ_x, δ_y are the finite-difference operators, $\Delta l = \Delta x = \Delta y$ is the mesh size, $D \equiv \nabla \cdot \mathbf{u}$, and ν_{cc}, ν_{ll} is the eddy viscosity defined at cell-center and lower-left corner, respectively. Eq. 29 can be rewritten as the usual Laplacian friction $\partial / \partial x (\nu \partial \mathbf{u} / \partial x) + \partial / \partial y (\nu \partial \mathbf{u} / \partial y)$ and the two formulations only differ around coastlines. The condition at lateral boundaries is set to free-slip (see Deremble et al., 2011; Ketefian and Jacobson, 2009; Dupont et al., 2003). Following Leith (1996) and Fox-Kemper and Menemenlis (2008) the epipycnal eddy viscosity varies in space and time as:

$$\nu \equiv \text{bvis} + \text{dvis} (\Delta x)^3 \sqrt{|\nabla \zeta|^2 + |\nabla D|^2}, \quad (30)$$

with bvis a constant background viscosity (zero by default) and dvis a non-dimensional coefficient. For most applications $0 \leq \text{dvis} \leq 1$. Leith’s viscosity is similar to the Smagorinsky (1963) viscosity but it is specifically designed for 2–D (or epipycnal) turbulence and gives better results in benchmarks (Graham and Ringler, 2013). Fox-Kemper and Menemenlis (2008) suggested the term $|\nabla D|$ to handle cases where D becomes comparable to ζ .

B Appendix: Files and structure of the code

beom is entirely contained inside three Fortran 95 files: `main.f95`, `shared_mod.f95`, and `private_mod.f95`. The file `main.f95` is the main program. In the simplest cases it only includes a directive to run the ocean model (`call run()`) and then a `quit()` statement. More complex cases would include coupling with other models such as SWAN (surface waves) or CICE (sea ice). All those directives are to be defined in the file `main.f95`.

The second file (`shared_mod.f95`, ≈ 200 lines of code) is a Fortran module that contains all the model parameters. This includes everything from physical constants (`grav = 9.8`), the precision of `real` variables, up to the model grid size (`lm` and `mm`) and the directory where the model results should be written to (`odir = '/tmp/'`). In nearly all cases, the file `shared_mod.f95` is the only file you need to understand and modify for your purposes. The comments within the file (!) clearly identify which parameters can be modified and what they represent.

The last file (`private_mod.f95`, ≈ 2000 lines of code) is a Fortran module that contains the procedures and arrays involved in a calculation. As its name implies, the module is defined `private` and there is no need to modify this file for a calculation. You can think of it as the engine hidden under the hood of a car. All the model parameters and physical constants are defined in `shared_mod.f95` and the only reason to pop the hood is if you are interested in the mechanics. Most subroutines are reasonably documented and include references to the relevant papers. A simplified flowchart looks like this:

```
call run()
1. read_input_data
   (a) check_consistency_options
   (b) read_input_file('h_bo') (if file h_bo.bin is provided)
   (c) index_grid_points
   (d) read_input_file (repeat for remaining files inside directory idir)
   (e) get_equilibrium_thickness_h_0 (if ocrp is activated)
   (f) save_metadata (document the value of each model parameter in this calculation)
   (g) write_outputs (save the initial condition as the first output record)
       i. write_array (repeat for every prognostic variable)
2. integrate_time
   (a) distribute_stress (if ocrp is activated)
   (b) first_three_timesteps (standard forward-backward for first 3 timesteps)
       i. update_n
       ii. update_mont_rvor_pvor_dive_kine
       iii. update_viscosity
       iv. update_u
       v. update_v
   (c) gener_forward_backward (repeat steps i to v for every remaining timesteps)
call quit()
```

C Appendix: Isopycnal outcrops and wetting-drying

Isopycnal surfaces become discontinuous (undefined) wherever they intersects the sea floor, the sea surface, or another isopycnal. Bretherton (1966) interprets such discontinuities as the layer being still active but infinitesimally thin (a δ -sheet). The same concept is referred to as a ‘massless layer’ by Hsu and Arakawa (1990). From a modelling point of view, the difficulty is changed from handling spatial discontinuities to handling very thin thickness values.

Isopycnal models traditionally rely on special advection schemes (e.g., Smolarkiewicz and Margolin, 1998) to handle such near-zero layer thicknesses. beom differs from these models

by following a new approach proposed by Salmon (2002). In this scheme, the Montgomery potential (Appendix A) is modified by adding an artificial term that ‘pushes’ the fluid toward areas where the layer thickness is vanishing. The artificial term is defined to be negligible everywhere except in locations and times where $h \rightarrow 0$. The scheme of Salmon (2002) can be activated by setting `ocrp = 1.` inside the file `shared_mod.f95`.

In the approach proposed by Salmon (2002), the state of rest ($\partial \mathbf{u} / \partial t = 0$) is defined by the Montgomery potential of each layer i being everywhere equal to a given constant C_i :

$$\left[\sum_{k=1}^{N=2} h_k^0(x, y) - H(x, y) \right] - \sum_{k=1}^{i-1} \frac{\rho_i - \rho_k}{\rho_i} h_k^0(x, y) - \frac{h_{\text{sal}}}{3} \left(\frac{h_{\text{sal}}}{h_i^0(x, y)} \right)^3 = C_i, \quad (31)$$

which should be compared to the original equation (Eq. 28). The last term on the left hand side is the artificial term proposed by Salmon, and h_{sal} is a constant named the ‘Salmon thickness’ (Primeau and Newman, 2007). Note that the state of rest implies that $\eta_1(x, y, t) = 0$ and $h_i(x, y, t) = h_i^0(x, y)$. The constants C_i are obtained by evaluating Eq. 31 at the location x, y of maximum depth ($H_{\text{max}} \equiv \max [H(x, y)]$) where the artificial term is assumed negligible ($h_i^0 \gg h_{\text{sal}}$). At this specific location the thicknesses are given by:

$$h_{i < N}^0 = H_{\text{max}} \times [\text{topl}(i + 1) - \text{topl}(i)], \quad h_N^0 = H_{\text{max}} \times [1 - \text{topl}(N)], \quad (32)$$

where `topl` (top of layers) is a vector defined by the user (see Fig. 1). Eq. 31 represents a set of N non-linear equations that must be solved at each x, y position to obtain the ‘equilibrium thickness’ $h_i^0(x, y)$ of the layers. Following Salmon (2002), Eq. 31 is solved iteratively with Newton’s method. Defining \mathbf{h} as the vector containing the values h_i^0 and denoting Eq. 31 as $f(\mathbf{h})$, the thicknesses are iterated as:

$$J(f(\mathbf{h}^l)) (\mathbf{h}^{l+1} - \mathbf{h}^l) = -f(\mathbf{h}^l), \quad (33)$$

where J is the Jacobian matrix of $f(\mathbf{h})$ and the subscript l denotes the iteration. Each iteration corresponds to solving a *linear* set of N equations. The thicknesses are iterated until a tolerance $O(10^{-6})$ meters is reached.

It is important to note that the equilibrium thickness $h_i^0(x, y)$ only needs to be computed once (at the initialization stage of the model). The whole procedure is computationally-cheap since it is a local 1-D (vertical) calculation and the number of layers is typically small (say, $N < 5$). The ‘pressure gradient’ computed at each model time-step is:

$$\nabla \frac{M_i(x, y, t)}{g} = \nabla \left[\left(\sum_{k=1}^N h_k(x, y, t) - H(x, y) \right) - \sum_{k=1}^{i-1} \frac{\rho_i - \rho_k}{\rho_i} h_k(x, y, t) - \frac{h_{\text{sal}}}{3} \left(\frac{h_{\text{sal}}}{h_i(x, y, t)} \right)^3 \right], \quad (34)$$

this particular formulation being suitable for single precision arithmetic (in most cases the three terms in Eq. 34 have the same order of magnitude). The first term is recognized as a barotropic

contribution due to the free surface, the second term represents the baroclinic pressure gradient, and the last term is the artificial term of Salmon (2002).

Salmon (2002) discusses the advantages of his approach. First, it is considerably cheaper (computationally speaking) than a flux-limiting algorithm (MPDATA, FCT, TVD; e.g., Smolarkiewicz and Margolin, 1998) as used in MICOM (Bleck and Smith, 1990). Potential energy (as re-defined by Salmon) and kinetic energy conservations are preserved by Salmon’s method. Vorticity conservation is also preserved since the artificial term is a gradient (i.e. it disappears when calculating the curl of the momentum equation). Finally, the model equations remain differentiable (upwind-biased and FCT schemes are not) which simplifies the calculation of bifurcations (Primeau and Newman, 2007) and model adjoint.

D Appendix: Stress and surface/bottom boundary layers

The traditional multi-layer shallow-water equations (Appendix A) assume that the boundary layers are contained inside the surface and bottom layers. In other words, the top and bottom layers are assumed to be sufficiently thick that the Reynolds stress $\rho^{-1} \partial \boldsymbol{\tau} / \partial z$ goes to zero within the layer. With this assumption, the Reynolds stress reduces to a term of the form $\boldsymbol{\tau} / (\rho h)$ applied to the surface/bottom layers. Outside of the boundary layers, the interfacial stress and the diapycnal viscosity are assumed to be zero. Ekman layers and their spiral are not represented by the model (only their vertically-integrated effect).

The situation becomes more complicated when the surface/bottom layers are allowed to become thin or vanish (`ocrp = 1.` in file `shared_mod.f95`; see Appendix C). The Reynolds stress $\boldsymbol{\tau}(x, y, z)$ enters the zonal momentum equation as:

$$\frac{\partial \mathbf{u}}{\partial t} = \dots + \frac{1}{\rho} \frac{\partial \boldsymbol{\tau}}{\partial z}, \quad (35)$$

and after discretization onto layer i this becomes:

$$\frac{\partial \mathbf{u}_i}{\partial t} = \dots + \frac{1}{\rho_i h_i} \int_{z_i - h_i}^{z_i} \frac{\partial \boldsymbol{\tau}}{\partial z} dz, \quad (36)$$

where $z_i < 0$ is the vertical position of the interface separating layers i and $i - 1$. The model assumes that the Reynolds stress at the surface/bottom decays linearly over a constant and uniform ‘boundary layer thickness’ (`hsbl` for surface, `hbb1` for bottom) set by the user in the file `shared_mod.f95`. For the surface stress $\boldsymbol{\tau}_s$ this is:

$$\int_{z_i - h_i}^{z_i} \frac{\partial \boldsymbol{\tau}}{\partial z} dz = \begin{cases} 0 & \text{if } z_i < -h_{\text{sbl}}, \\ \boldsymbol{\tau}_s \times \{\max [0, \min (h_i - 1.5 h_{\text{sal}}, z_i + h_{\text{sbl}}) / h_{\text{sbl}}]\} & \text{if } z_i > -h_{\text{sbl}}, \end{cases} \quad (37)$$

where the non-dimensional quantity $\{\dots\}$ is named `layt` (layer-top) in the code (see the subroutine `distribute_stress`). This parameterization distributes the surface stress amongst the layers when `ocrp` is activated. Note that $h_i - 1.5 h_{\text{sal}}$ is used in place of h_i to ensure that no stress is applied to layers that have vanished (i.e. layers with $h \leq 1.5 h_{\text{sal}}$). The array `layt` is updated every `dt3d` days during a calculation.

The bottom stress is calculated from the horizontal velocities in the deepest layer having a thickness $\geq 2h_{\text{sal}}$ (i.e. we ignore the deep layers that are ‘mass-less’). Then, the bottom stress is distributed over a boundary layer of thickness h_{bbl} (same way as for the surface stress).

If `ocrp` = `0`. (i.e. no outcrops allowed) then the surface (bottom) stress is only applied to the uppermost (bottom) layer.

E Appendix: Time-stepping schemes

The code uses explicit time-stepping schemes where all prognostic variables are advanced with a single time-step Δt . Explicit schemes are simpler than implicit ones, they are more accurate, and they have better scalability on parallel computers. Following Beckers and Deleersnijder (1993, their Eq. 44) and assuming a C-grid of uniform mesh size Δx , the maximum stable time-step is:

$$(\Delta t)_{\text{max}} = \frac{1}{2} \frac{\Delta x}{\sqrt{g H_{\text{max}}}}. \quad (38)$$

It is assumed that the number of active layers (N) is sufficiently small that the model runtime remains competitive with split-explicit schemes. Note also that split-explicit schemes introduce difficulties (Hallberg and Adcroft, 2009; Simmons et al., 2004) that are absent in `beom`.

Two different time-stepping schemes are available in `beom`: the ‘standard’ Forward-Backward (FB) scheme (used for the barotropic mode of MICOM, Bleck and Smith, 1990), and the ‘generalized’ FB scheme (used for the barotropic mode of UCLA-ROMS, Shchepetkin and McWilliams, 2005). The standard FB is a single-stage, single time-level, explicit algorithm. In a simple case with a single layer, no momentum advection, no forcing and no dissipation, the standard FB scheme reduces to:

$$\eta^{n+1} = \eta^n + \Delta t \times \text{rhs} [u^n, v^n, \eta^n], \quad (39)$$

$$u^{n+1} = u^n + \Delta t \times \text{rhs} [\eta^{n+1}, v^n], \quad (40)$$

$$v^{n+1} = v^n + \Delta t \times \text{rhs} [\eta^{n+1}, u^{n+1}], \quad (41)$$

where n is the time-level, and ‘rhs’ represents the right hand side of the prognostic equations. We see that the surface η is calculated ‘forward’ in time (i.e. from u^n, v^n, η^n), and the velocity is calculated ‘backward’ in time (i.e. from the newly updated η^{n+1}). Following Bleck and Smith (1990),

the time-stepping sequence alternates between η, u, v and η, v, u so that both velocity components benefit from a ‘backward’ Coriolis term. The standard scheme is simple, it allows relatively long time-steps Δt , and it can be used on the very first time-steps as it does not require information other than the current state of the model variables ($t = n$). The scheme does not include any numerical dissipation and thus conserves mechanical energy in simple cases.

The generalized FB scheme is a single-stage, four time-levels algorithm (Shchepetkin and McWilliams, 2005, their Eq. 2.49):

$$\eta^{n+1} = \eta^n + \Delta t \times \text{rhs} [(u, v, \eta)^n, (u, v, \eta)^{n-1}, (u, v, \eta)^{n-2}], \quad (42)$$

$$u^{n+1} = u^n + \Delta t \times \text{rhs} [\eta^{n+1}, \eta^n, \eta^{n-1}, \eta^{n-2}, v^n], \quad (43)$$

$$v^{n+1} = v^n + \Delta t \times \text{rhs} [\eta^{n+1}, \eta^n, \eta^{n-1}, \eta^{n-2}, u^{n+1}]. \quad (44)$$

The averaging of the different time-levels damps high-frequency oscillations (i.e. close to the cutoff $\Delta t^{-1} \sim \sqrt{g H_{\text{max}}} / \Delta x$) and prevents them from ‘contaminating’ the well-resolved scales⁵. This also means that the scheme is slightly dissipative and that mechanical energy can slowly decay in time in absence of forcing.

The binary flag `g_fb` (inside module `shared_mod.f95`) determines which scheme is used during a calculation. A value of `1`. corresponds to the generalized scheme, and `0`. to the standard FB scheme. Note that the generalized FB relies on the standard FB scheme for the first few time-steps of a calculation to ‘fill’ the time-levels $n-2$ and $n-1$ (see subroutine `first_three_timesteps`).

F Appendix: Spatial discretization and indexing of grid points

F.1 Spatial discretization

The model follows standard finite-difference methods by having its variables positioned on a regular ‘C’ grid of uniform horizontal resolution $\Delta x = \Delta y$ (Arakawa and Lamb, 1981). The zonal velocity point u is on the left edge, meridional velocity point v on the bottom edge, surface elevation η at cell-center, and vorticity point at lower-left corner (Fig. 17). The ‘C’ grid is well suited for coastal modelling where the first Rossby radius of deformation is assumed to be well resolved.

The momentum equation (Eq. 22) is discretized following the energy-conserving scheme of Sadourny (1975, his Eq. 3). I found this scheme to behave better than the enstrophy-conserving scheme of the same author (Sadourny, 1975, his Eq. 4) as the latter produces spurious currents

⁵For example, the ‘soliton’ test-case requires no explicit dissipation at all when the generalized scheme is used. The same test-case requires the addition of explicit dissipation (`dvis` > 0) to run smoothly with the standard scheme.

growing in time in the test-case `outcrop_seamount.m`. Note that the two schemes only differ by the discretization of the term $-q\mathbf{e}_e \times h\mathbf{u}$. In other words, the model will conserve both energy and potential enstrophy if q, h, u are sufficiently smooth (since both discretizations will produce the same result). When it is activated, the Leith viscosity (Eq. 30) works to this goal by concentrating the dissipation in locations where the vorticity gradients are high (i.e. where the fields are becoming noisy) while leaving the large-scales unaffected.

The continuity equation (Eq. 25) is discretized as in Sadourny (1975) but with a higher-order approximation for the thickness flux at the cell edges. The second-order centered interpolation of Sadourny (1975) is replaced with:

$$\begin{aligned} u h &\equiv u_i \frac{h_{i-1} + h_i}{2} - \max(u_i, 0) \frac{h_{i-2} - 2h_{i-1} + h_i}{6} - \min(u_i, 0) \frac{h_{i-1} - 2h_i + h_{i+1}}{6}, \\ &= \max(u_i, 0) \left[\frac{h_{i-1} + h_i}{2} - \frac{h_{i-2} - 2h_{i-1} + h_i}{6} \right] + \min(u_i, 0) \left[\frac{h_{i-1} + h_i}{2} - \frac{h_{i-1} - 2h_i + h_{i+1}}{6} \right], \end{aligned} \quad (45)$$

where u_i is the zonal flow at the left edge of a cell and h is defined at the center of the cells (indices $i-2$ to $i+1$). Eq. 45 was proposed by Shchepetkin and McWilliams (2005, their page 394) and represents an upstream-biased interpolation with a hyper-diffusive truncation. It is similar to the third-order UTOPIA scheme but with all the quadratic terms neglected (see Shchepetkin and McWilliams, 1998, their Appendix B). The diffusion only occurs at the grid scale and it leaves the well-resolved scales intact. The numerical diffusion implicit to Eq. 45 is the only form of diffusion applied to the layer thickness.

The benefits of the higher-order interpolation (i.e. last two terms of Eq. 45) are most apparent in cases of flow-topography interactions (e.g. Test-case #7, Stratified flow over topography). The hyper-diffusion considerably reduces the noise at the grid level and allows for a smooth solution even with zero viscosity. Note that the last two terms of Eq. 45 are forced to zero in regions of outcrop ($h < 2h_{\text{sal}}$) to ensure that the hyper-diffusion does not interfere with Salmon’s method.

F.2 Indexing of grid points

Large-scale ocean models traditionally used *structured* grids where all grid points are indexed in two-dimensional or three-dimensional arrays that mimic their location in physical space. With such structured grids, neighboring cells are identified simply by incrementing/decrementing the array indices. `beom` differs from traditional finite-difference models by having the grid points of a layer indexed into a one-dimensional array. Most coastal models use such an *unstructured* indexing because it is more flexible and adapted to the complex geometry of estuaries (e.g., Fig. 19). Operations are conducted only in coastal areas and all unnecessary cells representing land are discarded (e.g., Backhaus, 2008).

In the code, each model variable of a given layer is represented by a vector of length `ndeg+1`. In the example of Fig. 17, `ndeg=32` and the zonal velocities u are stored as:

```
real :: u( 0 : ndeg, nlay )
```

All the model variables share the same indexing and dimensions $(ndeg + 1) \times nlay$. The index `0` is used to represent the value of the variable on discarded cells (usually zero). Interactions between neighboring cells necessitate a *table of connectivity* (array `neig` in the code) that provides the index of the 8 neighbouring cells. For instance, the index of the cell located on the right of a given cell `ipnt` is given by:

```
index_to_neighbor_on_the_right = neig(1, ipnt)
```

where the first argument is an integer between 1 and 8 that identifies the neighbors in a counter-clockwise sense (starting at 1 for the neighbor on the right). The array `subc(ipnt, 2)` works the other way around, by providing the i, j indices that correspond to a given wet cell `ipnt`. Those familiar with Matlab can see that `subc` is similar to the function `ind2sub`.

The indexation of the grid cells is done automatically by the code. It only requires as input a raw topographic grid (i.e. a two-dimensional array of real positive numbers representing the water depth in meters; see the test-cases for examples). The code distinguishes ‘wet’, ‘dry’ and ‘discarded’ cells according to depth and position. All cells shallower than $2 \times h_{\text{dry}}$ are either ‘dry’ or ‘discarded’. `hdry` is a parameter that must be set in `shared_mod.f95` and in `get_nbr_deg_freedom.m` (`hdry` is typically 10^{-3} meters).

Note that the code uses static arrays and needs to know the value of `ndeg` (the total number of ‘wet’ and ‘dry’ cells) prior to a calculation. One way to obtain `ndeg` is to use the function `get_nbr_deg_freedom.m` with a raw topographic grid (see the test-cases for examples). As an alternative to using `get_nbr_deg_freedom.m`, one can enter a dummy value for `ndeg` in `shared_mod.f95` and initiate the calculation. The code will immediately stop and return the correct `ndeg` that should have been entered in `shared_mod.f95`.

G Appendix: Treatment of land boundaries

The conservation properties of most finite-difference methods tend to degrade rapidly when land boundaries are introduced. Ketefian and Jacobson (2009) compare the accuracy of such schemes in the context of the inviscid shallow-water equations. For an Arakawa-C grid, the difficulty lies in extrapolating the relative vorticity ζ and the layer thickness h along the land boundaries while (ideally) conserving mass, energy, vorticity and potential enstrophy. An important conclusion from their study is that most schemes perform poorly in presence of realistic land masses. The

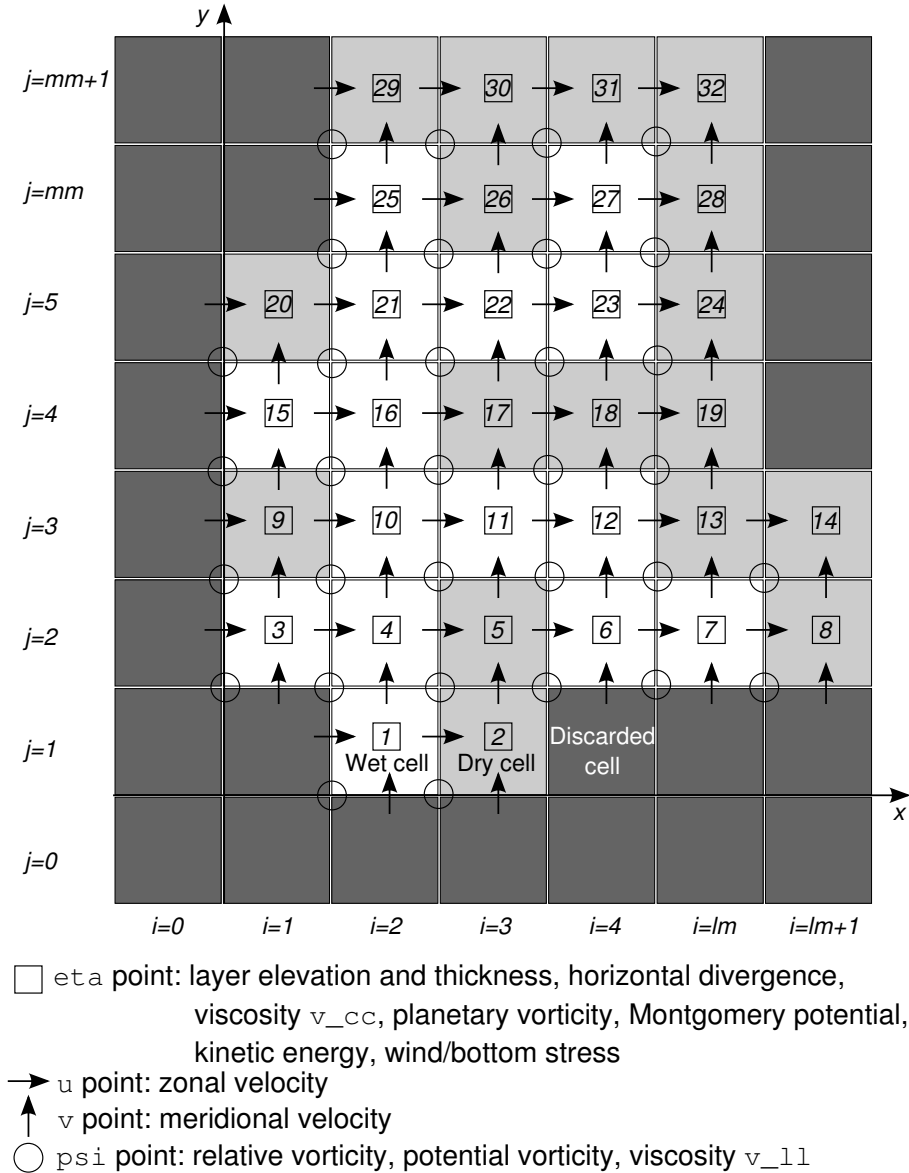


Figure 17: Discretization of variables in horizontal space. White squares are wet cells (ocean) and gray squares are land. Variables from white and light gray cells are indexed into a vector while dark gray cells are discarded. Equations 22,25 are applied at white/light gray cells while fluxes between white and gray cells are zero. Each cell has its η point at the center, u point on the left, v point at the bottom, and ψ point on the lower-left corner. In cases with wetting-drying the wet cells must be extended to include all potentially-flooded areas.

only one that conserves all these quantities (the so-called BVEM scheme) is complicated and requires the solution of a prognostic equation for each ζ grid point located along a land boundary.

A simpler and reasonably accurate alternative to BVEM is named Free-Slip (FS). In this scheme, ζ is simply set to zero along land boundaries which is equivalent to imposing a free-slip lateral boundary condition. The layer thickness h is extrapolated as a weighted average of the neighbouring wet grid points (Ketefian and Jacobson, 2009, their Appendix B). These authors show that the FS scheme performs nearly as well as the BVEM scheme. Similar conclusions were obtained by Deremble et al. 2011 and Dupont et al. 2003. For all these reasons I decided to implement the FS scheme in beam.

H Appendix: Scalability and performance

The overhead associated with the un-splitted time-stepping is partly mitigated by the scalability of the code (i.e., its performance on multi-core computers). As an example, I will consider the case of a closed rectangular basin 600×300 km with a uniform depth of 3,000 m and a 1 km resolution. The domain is on a f -plane and the forcing is a constant 0.1 Pascal westerly wind applied at $t = 0$ (the ocean is initially at rest). The computer is a multi-core workstation with gfortran version 4.7 as the compiler. For the purpose of this test-case I use a high optimization `-Ofast -march=native`. The real (wall-clock) duration of the calculation is presented as a function of model layers and parallel threads. For example for a 6-core calculation one would type (the exact command depends on the shell used):

```
setenv OMP_NUM_THREADS 6
```

Figure 18a illustrates the time T required for one day of simulation. The benchmark demonstrates that a modest number of cores (12) suffice to maintain the run-time below an acceptable level (15 minutes in this case). The runtime is linearly proportional to the number of layers (Fig. 18b). The scalability of the code can be compared to a perfectly-parallel calculation defined by:

$$T(N) = T(1)/N, \quad S(N) = T(1)/T(N), \quad (46)$$

where T is the wall-clock time for a fixed-size calculation, N the number of cores, and S the speedup. Note that, even in the theoretical case of perfectly-parallel code, the time $T(N)$ rapidly reaches an asymptote (Amdahl's law).

The model runtime is compared to the perfectly-parallel case in Fig. 18c,d. The model reproduces the expected $1/N$ decrease in run-time (Fig. 18c) but its scalability gradually decreases with increasing N (Fig. 18d). This result is expected and is due to synchronization between threads at the end of each parallel loop and to any serial operation present in the code. Further improvements

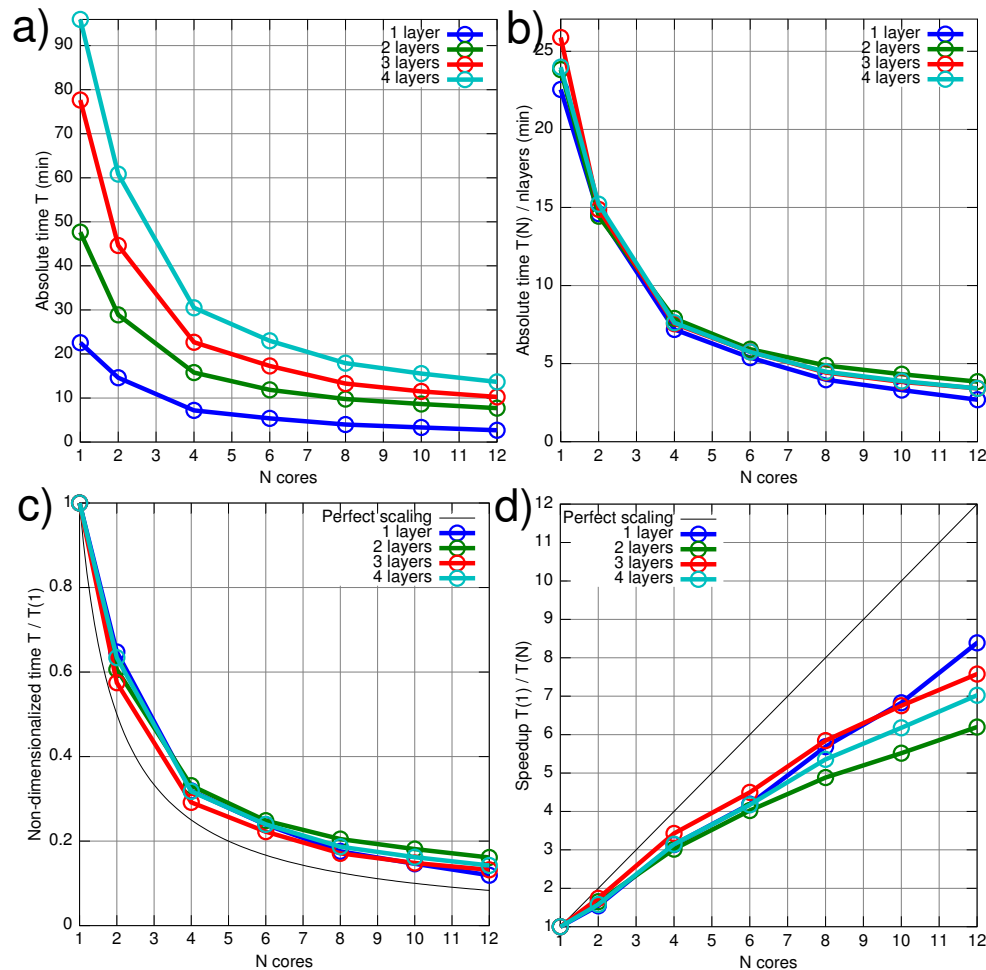


Figure 18: Scalability and performance of the code in a realistic benchmark (see Appendix H). $T(N)$ is the wall-clock time in minutes for a 1-day simulation and N is the number of cores used.

in speed would require a split scheme and/or a coarse-grain parallelization, both modifications representing more work than I am willing to invest.

In a typical calculation the workload is dominated by the following five subroutines: `update_u` and `update_v` (each representing $\sim 28\%$ of the total runtime), `update_mont_rvor_pvor_dive_kine` ($\sim 19\%$), `update_h` ($\sim 14\%$), and `update_viscosity` ($\sim 10\%$).

I Appendix: Examples of calculations

See Fig. 19, ...

J Appendix: Troubleshooting

1. *I cannot reproduce the results of the test-cases.*

First, verify that all the parameters inside `shared_mod.f95` are correct. Also, make sure that the directory containing the input files (variable `idir` inside file `shared_mod.f95`) is *empty* before you execute the `m`-file. The model will try to use *all* the files present in this directory.

2. *The scalability drops when I try to use more than x threads on my Intel workstation.*

Make sure that the number of threads you are using (environment variable `OMP_NUM_THREADS`) is equal or inferior to the number of *physical* cores available on your workstation. Intel chips use the so-called HyperThreading technology to provide additional virtual cores to the operating system. The scalability of most GFD codes drops dramatically when one attempts to use the virtual cores. You must either deactivate HyperThreading on your workstation, or make sure that `OMP_NUM_THREADS` is \leq the number of real (physical) cores.

3. *I get a `Segmentation fault` message when trying to use a large grid.*

A common problem with large grids ($n_{\text{deg}} > 10^6$) is an insufficient stacksize. See the documentation of your shell (`tcsh`, `bash`, etc.) on how to read the current settings and how to increase the stacksize. For example in `bash` you can read the current value by typing:

```
ulimit -a
```

which on my laptop returns 8192 kbytes. I increase the stacksize by typing:

```
ulimit -s 262144
```

where 262144 is the new stacksize in kbytes.

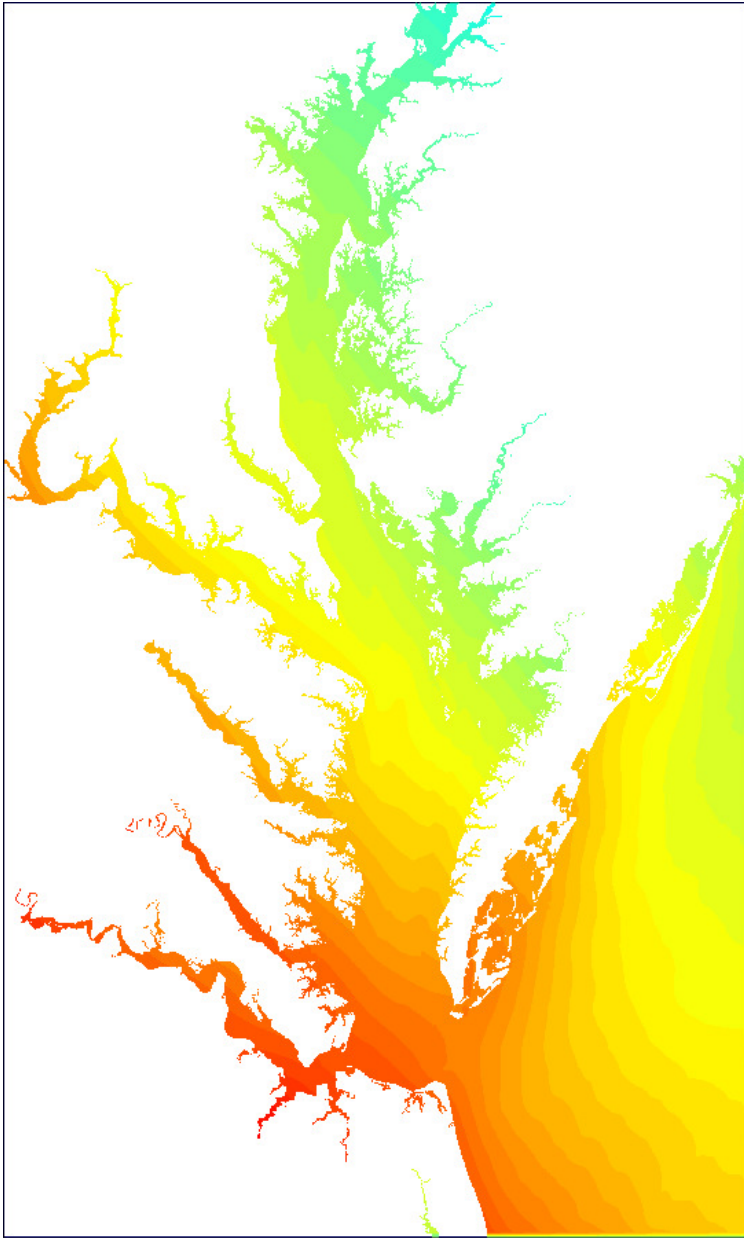


Figure 19: Sea surface elevation during a nor'easter event in the Chesapeake Bay. The grid size is 2019×3047 with a uniform horizontal resolution of 100 m.

With tcsh the syntax is a bit different. You read the current values by typing:

```
limit
```

and you set the new stacksize by typing:

```
limit stacksize 262144
```

4. *I think I encountered a bug.*

The code has been quality-controlled in multiple ways (test-cases with known analytical solutions, running the code in a debugger) but bugs can always slip in. Please email me with a description of the problem and step-by-step instructions for me to reproduce the problem.

K Appendix: Legal information

beom is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

beom is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

A copy of the GNU General Public License is available in the text file COPYING.

References

- Adcroft, A., Marshall, D., 1998. How slippery are piecewise-constant coastlines in numerical ocean models? *Tellus* 50A, 95–108.
- Arakawa, A., Lamb, V. R., 1981. A potential enstrophy and energy conserving scheme for the shallow water equations. *Mon. Weather Rev.* 109, 18–36.
- Backhaus, J. O., 2008. Improved representation of topographic effects by a vertical adaptive grid in vector-ocean-model (VOM). Part I: Generation of adaptive grids. *Ocean Modelling* 22, 114–127, doi:10.1016/j.ocemod.2008.02.003.
- Baines, P. G., Leonard, B. P., 1989. The effects of rotation on flow of a single layer over a ridge. *Q. J. R. Meteorol. Soc.* 115, 293–308.
- Beckers, J.-M., Deleersnijder, E., 1993. Stability of a FBTCS scheme applied to the propagation of shallow-water inertia-gravity waves on various space grids. *J. Comput. Phys.* 108, 95–104.

- Bleck, R., Smith, L. T., 1990. A wind-driven isopycnic coordinate model of the North and Equatorial Atlantic Ocean 1. Model development and supporting experiments. *J. Geophys. Res.* 95 (C3), 3273–3285.
- Bretherton, F. P., 1966. Baroclinic instability and the shortwavelength cut-off in terms of potential vorticity. *Q. J. Royal Met. Soc.* 92, 335–345.
- Carrier, G. F., Greenspan, H. P., 1958. Water waves of finite-amplitude on a sloping beach. *J. Fluid Mech.* 4, 97–109.
- Chapman, D. C., 1985. Numerical treatment of cross-shelf open boundaries in a barotropic coastal ocean model. *J. Phys. Oceanogr.* 15, 1060–1075.
- Chen, C., Huang, H., Beardsley, R. C., Liu, H., Xu, Q., Cowles, G., 2007. A finite volume numerical approach for coastal ocean circulation studies: Comparisons with finite difference models. *J. Geophys. Res.* 112 (C03018), doi:10.1029/2006JC003485.
- Danilov, S., 2013. Ocean modeling on unstructured meshes. *Ocean Modelling*. Doi:10.1016/j.ocemod.2013.05.005.
- Davies, H. C., 1976. A lateral boundary formulation for multi-level prediction models. *Quart. J. R. Met. Soc.* 102, 405–418.
- Deremble, B., Hogg, A., Berloff, P., Dewar, W. K., 2011. On the application of no-slip lateral boundary conditions to ‘coarsely’ resolved ocean models. *Ocean Modelling* 39, 411–415, doi:10.1016/j.ocemod.2011.05.002.
- Dupont, F., Straub, D. N., Lin, C. A., 2003. Influence of a step-like coastline on the basin scale vorticity budget of mid-latitude gyre models. *Tellus* 55A, 255–272.
- Fox-Kemper, B., Menemenlis, D., 2008. Can large eddy simulation techniques improve mesoscale-rich ocean models? In: Hecht, M., Hasumi, H. (Eds.), *Ocean Modeling in an Eddy-ing Regime*. Vol. 177. AGU Geophysical Monograph Series, pp. 319–338.
- Graham, J. P., Ringler, T., 2013. A framework for the evaluation of turbulence closures used in mesoscale ocean large-eddy simulations. *Ocean Modelling* In Press, doi:10.1016/j.ocemod.2013.01.004.
- Hallberg, R., Adcroft, A., 2009. Reconciling estimates of the free surface height in Lagrangian vertical coordinate ocean models with mode-split time stepping. *Ocean Modelling* 29, 15–26, doi:10.1016/j.ocemod.2009.02.008.
- Hallberg, R., Rhines, P., 1996. Buoyancy-driven circulation in an ocean basin with isopycnals intersecting the sloping boundary. *J. Phys. Oceanogr.* 26, 913–940.
- Hsu, Y. G., Arakawa, A., 1990. Numerical modeling of the atmosphere with an isentropic vertical coordinate. *Monthly Weather Review* 118, 1933–1959.
- Ilıcak, M., Adcroft, A. J., Griffies, S. M., Hallberg, R. W., 2012. Spurious dia-neutral mixing and the role of momentum closure. *Ocean Modelling* 45–46, 37–58, doi:10.1016/j.ocemod.2011.10.003.
- Ketefian, G. S., Jacobson, M. Z., 2009. A mass, energy, vorticity, and potential enstrophy conserving lateral fluid-land boundary scheme for the shallow-water equations. *J. Comp. Phys.* 228, 1–32.
- Lavelle, J. W., Thacker, W. C., 2008. A pretty good sponge: Dealing with open boundaries in limited-area ocean models. *Ocean Modelling* 20, 270–292.
- Leith, C. E., 1996. Stochastic models of chaotic systems. *Physica D* 98, 481–491.
- Marchesiello, P., McWilliams, J. C., Shchepetkin, A., 2001. Open boundary conditions for long-term integration of regional oceanic models. *Ocean Modelling* 3, 1–20.
- Millot, C., Crépon, M., 1981. Inertial oscillations on the continental shelf of the Gulf of Lions—Observations and theory. *J. Phys. Oceanogr.* 11, 639–657.
- Modave, A., Deleersnijder, É., Delhez, É. J., 2010. On the parameters of absorbing layers for shallow water models. *Ocean Dyn.* 60, 65–79.
- Morel, Y. G., Darr, D. S., Talandier, C., 2006. Possible sources driving the potential vorticity structure and long-wave instability of coastal upwelling and downwelling currents. *J. Phys. Oceanogr.* 36, 875–896.
- Palma, E. D., Matano, R. P., 1998. On the implementation of passive open boundary conditions for a general circulation model: The barotropic mode. *J. Geophys. Res.* 103 (C1), 1319–1341.
- Primeau, F. W., Newman, D., 2007. Bifurcation structure of a wind-driven shallow water model with layer-outcropping. *Ocean Modelling* 16, 250–263, doi:10.1016/j.ocemod.2006.10.003.
- Rao, D., 1966. Free gravitational oscillations in rotating rectangular basins. *J. Fluid Mech.* 25, 523–555.

- Sadourny, R., 1975. The dynamics of finite-difference models of the shallow-water equations. *J. Atmos. Sci.* 32, 680–689.
- Salmon, R., 2002. Numerical solution of the two-layer shallow water equations with bottom topography. *J. Mar. Res.* 60, 605–638.
- Shchepetkin, A. F., 2015. An adaptive, Courant-number-dependent implicit scheme for vertical advection in oceanic modeling. *Ocean Modelling* 91, 38–69, doi:10.1016/j.ocemod.2015.03.006.
- Shchepetkin, A. F., McWilliams, J. C., 1998. Quasi-monotone advection schemes based on explicit locally adaptive dissipation. *Monthly Weather Rev.* 126, 1541–1580.
- Shchepetkin, A. F., McWilliams, J. C., 2005. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling* 9, 347–404.
- Simmons, H. L., Hallberg, R. W., Arbic, B. K., December 2004. Internal wave generation in a global baroclinic tide model. *Deep Sea Res. II* 51 (25-26), 3043–3068, doi:10.1016/j.dsr2.2004.09.015.
- Smagorinsky, J., 1963. General circulation experiments with the primitive equations. I. the basic experiment. *Monthly Weather Review* 91 (3), 99–164.
- Smolarkiewicz, P. K., Margolin, L. G., 1998. MPDATA: A finite-difference solver for geophysical flows. *J. Comp. Phys.* 140, 459–480.
- Stommel, H., 1948. The westward intensification of wind-driven ocean currents. *Transactions of the American Geophysical Union* 29 (2), 202–206.
- Williams, W. J., Gawarkiewicz, G. G., Beardsley, R. C., 2001. The adjustment of a shelfbreak jet to cross-shelf topography. *Deep-Sea Research II* 48, 373–393.