

Documentation for beom

Pierre St-Laurent psl@fastmail.net
<http://nordet.net/beom.html>

June 11, 2023

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 idealized nature of the model equations (restricting beom to process-oriented studies) and to the streamlined code (only $\sim 2,000$ lines of Fortran 95, distributed under a copyleft license; see §K). The model is appropriate for computers ranging from laptops to multi-core workstations.

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

The basic model dynamics and numerics are similar to traditional isopycnal models such as MICOM¹ or HIM². The model differs by the way it treats vanishing layers (wetting/drying and isopycnal outcrops, see §C) and by its regular unstructured grid on the horizontal (efficient in sparse domains such as estuaries; §E). The former allows the model to accurately represent large thickness gradients while preserving potential vorticity conservation (see Salmon, 2002). The respect of conservation laws and the ability to reproduce important analytical solutions (see the test-cases) are primary goals behind beom.

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

The code is geared toward process-oriented experiments of coastal dynamics in large (Coriolis-influenced) basins with simple (layered) stratification and complex geometries (such as estuaries). In an ideal world, the code would feature full (unsplit, unfiltered) barotropic-baroclinic coupling, Lagrangian (isopycnal) coordinates on the vertical, no

numerical diffusion/dissipation, and a strict respect of conservation laws. The code would also take advantage of multi-core commodity hardware for parallel execution (devices as inexpensive as a Raspberry Pi already provide four computational cores.)

Existing “coastal models” such as FVCOM (Chen et al., 2007) use irregular grids whose design requires particular care in complex domains. Computational performance and conservation of enstrophy/energy are also problematic with irregular grids (Danilov, 2013). At 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-slip lateral boundary conditions, no wetting/drying) that are computationally inexpensive 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 schemes with a regular unstructured grid (§E) to meet (most of) its design goals, particularly in sparse horizontal domains, with only 2,000 lines of code and no library dependency.

4. *How fast is beom?*

The code is optimized for domains having a sparsely-populated horizontal grid, a few layers (say, ≤ 4), and for a multi-core workstation ($4 < N < 128$, N the number of cores) with shared memory. In this niche, the code provides a highly accurate solution to the primitive equations at a competitive cost. Outside of this niche, computational performances will rapidly degrade from suboptimal use of memory cache, the lack of barotropic-baroclinic splitting/filtering, and the fine-grain parallelization.

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:

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

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

```
gfortran -O3 -fopenmp shared_mod.f95 private_mod.f95 main.f95
export OMP_NUM_THREADS=2
./a.out
```

... and the calculation starts.

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 Matlab³ 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.

Note: Moderate-to-large grids may require an increase in the stack size limit (see Appendix I). Also, the Intel compiler `ifort` would require the flags `-free -Tf` for compilation because it does not recognize the file extension `.f95` natively (type `man ifort` for additional information.)

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. *Why not use a GPU rather than relying on OpenMP parallelization?*

I have no experience of parallel coding over a graphics processing unit (GPU) but I can appreciate their massive potential. Maybe one day, or maybe someone else will do it and share their modified code.

8. *Is beom funded/sponsored by an agency or institution?*

beom is a personal, unfunded, late night/rainy Sunday afternoon project. I do it on my own time, at my own pace, and for the fun of it. Feel free to use, modify and redistribute beom under the same license (see §K).

9. *How should the use of beom be acknowledged?*

There isn't a formal publication for beom because it is an implementation of various algorithms that are already published. The present document is maintained in parallel with the Fortran code and is the most appropriate reference for beom.

³A free alternative to Matlab is GNU Octave (<https://www.gnu.org/software/octave>). The two software share the same syntax and thus their scripts are compatible. Octave is readily available as a package in most Linux distributions. Alternatively, the scripts can be ported to other languages (e.g., Python) since they are fairly basic.

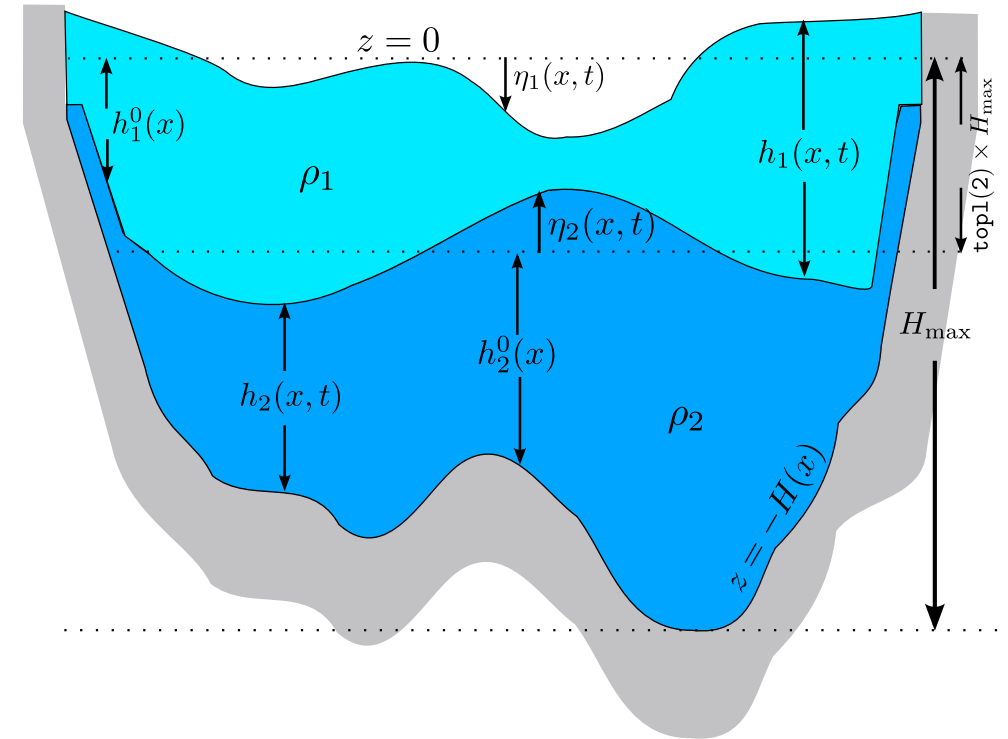


Figure 1: Geometry of the model in the x - z plane for a case with two layers. The figure shows a case with outcrops (h_2 has a small (but non-zero) thickness near the coasts) but no wetting-drying ($h_1 \gg 0$ everywhere; see Fig. 13 for comparison). The vertical scale is exaggerated.

10. *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 new test-cases and improving the documentation. beom is not meant to be a ‘community code’ (à la github) because of a lack of resources, and I do not plan to implement additional functionalities at this time, but you are welcome to modify beom (e.g., implement a rigid lid; Zhao et al., 2019) and to redistribute the modified code under the same license (see §K).

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 Matlab/Octave 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 $t \gg f^{-1}$ (Milot 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. 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 spatial discretization used in the code guarantees the conservation of volume and mechanical energy in any given layer while vorticity and potential enstrophy are approximately conserved (§E.1). Round-off errors, land boundaries (§E.3) and the time-discretization (§D) 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 §A for

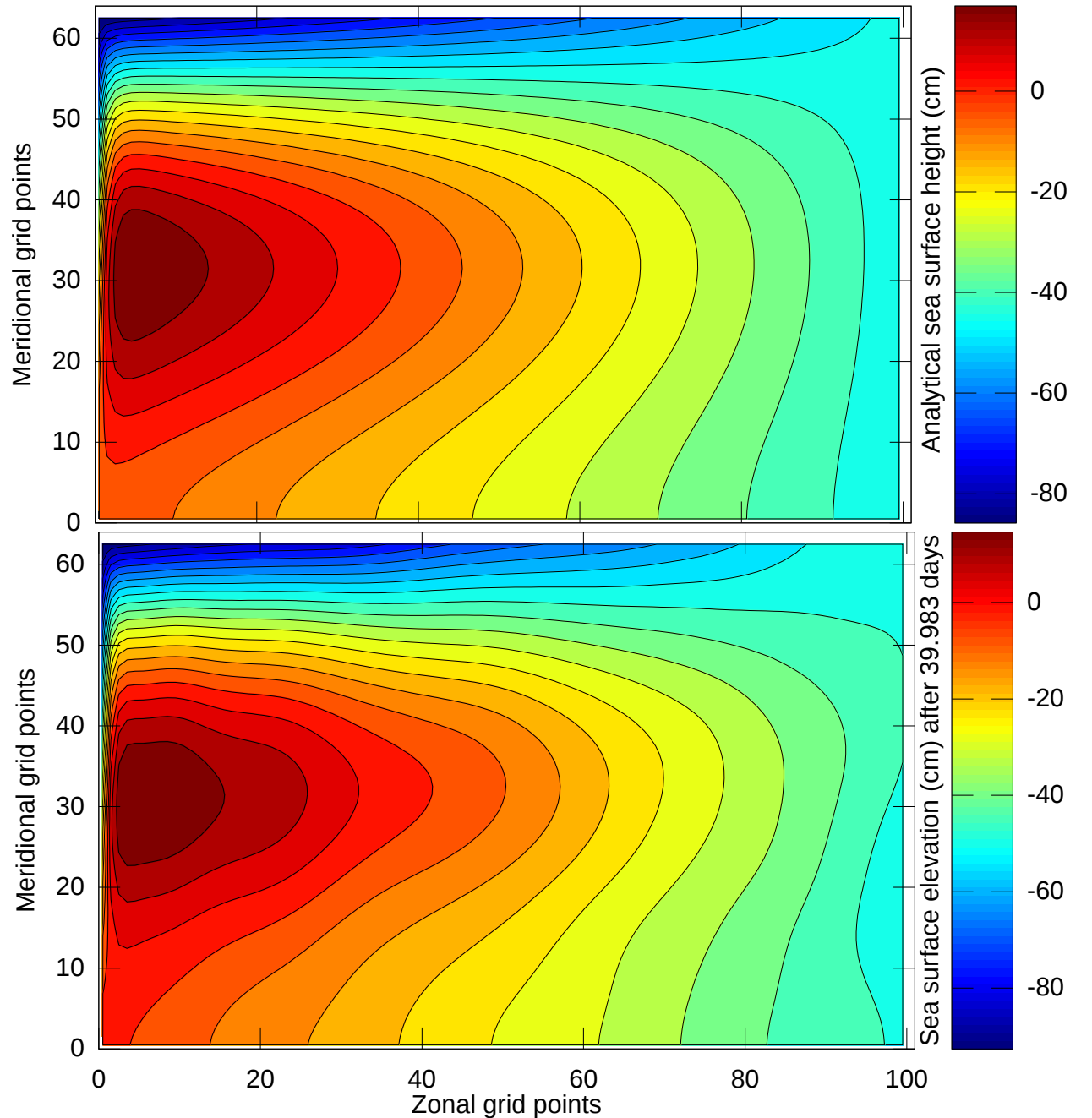


Figure 2: Sea surface elevation η_1 for $f = 10^{-11} \text{ y s}^{-1}$ in the test-case of Stommel (1948).

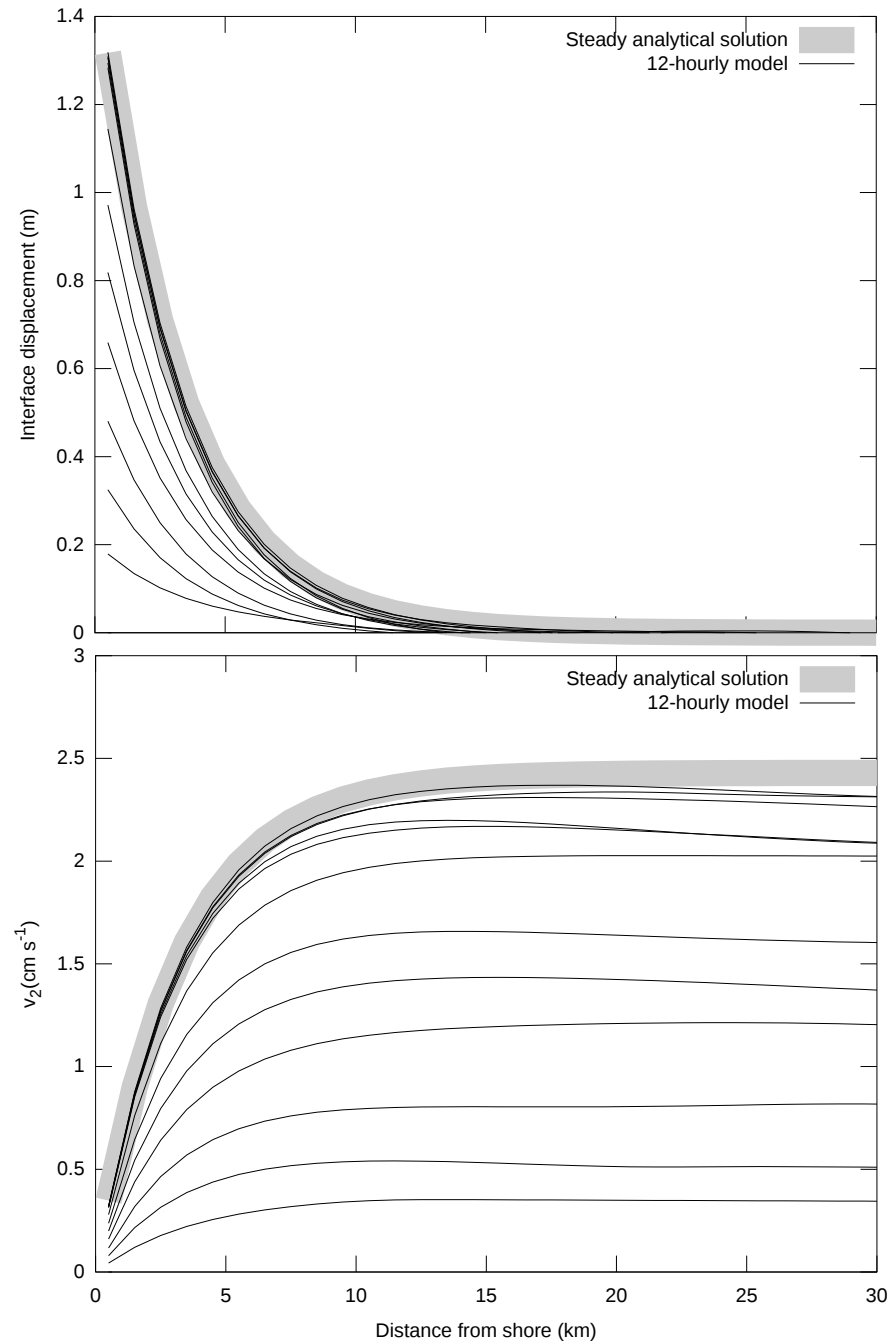


Figure 3: Comparison between analytical asymptotic values (thick gray line; Millot and Crépon 1981, their Eqs. 4.8 and 4.16) and model outputs sampled every 12 hours in test-case of upwelling driven by seaward wind.

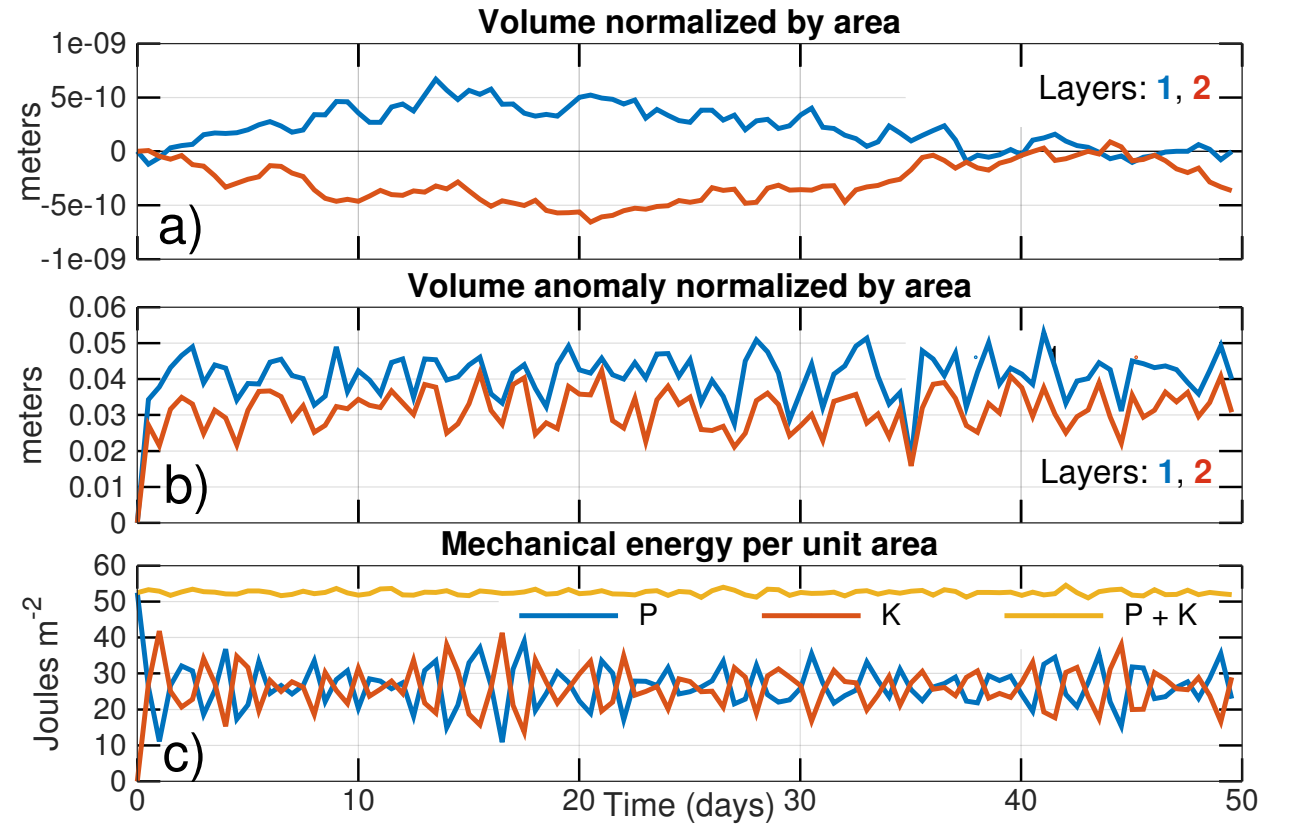


Figure 4: Numerical errors in volume and mechanical energy. In (a,b) the curves are referenced to their value at $t = 0$ and normalized by the area of the basin $\iint dx dy$. See Test-case 3 for definition of quantities. The results were obtained with the ‘standard’ forward-backward scheme (§D).

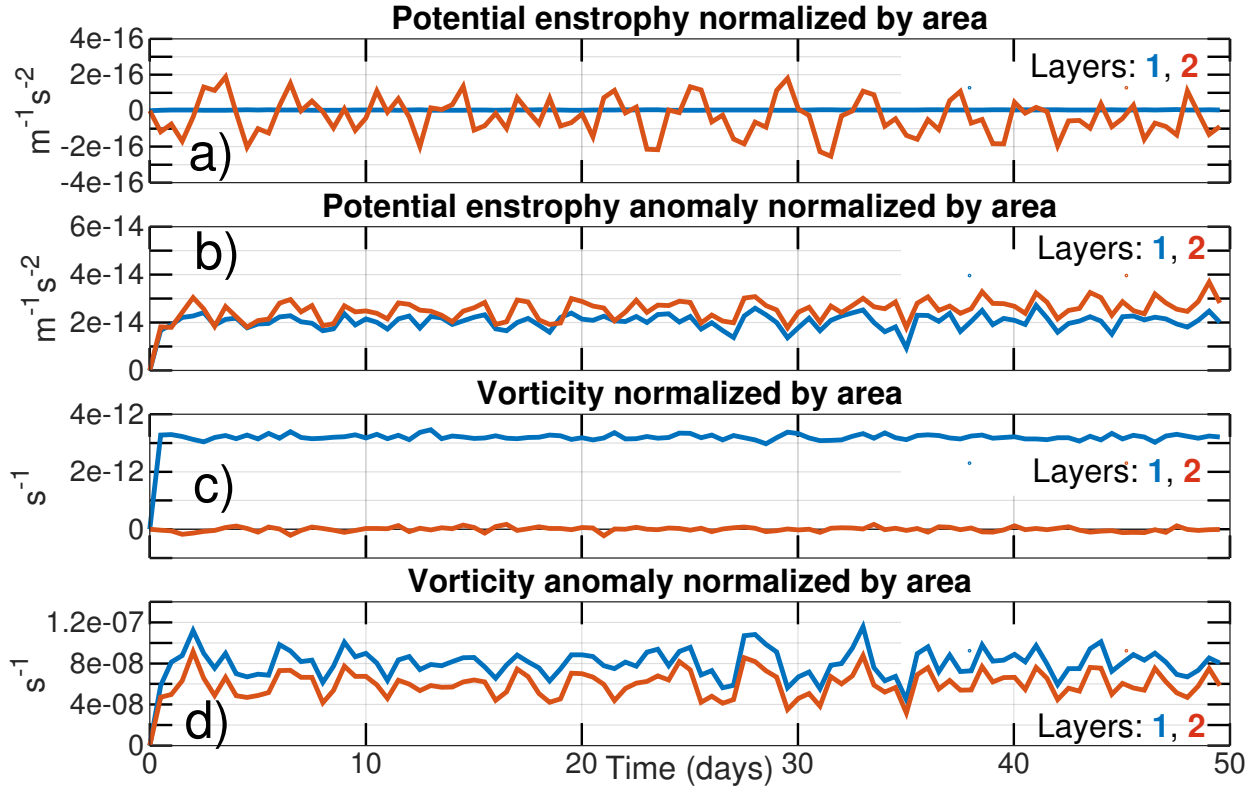


Figure 5: Numerical errors in potential enstrophy and vorticity. In (a,b,c,d) the curves are referenced to their value at $t = 0$ and normalized by the area of the basin $\iint dx dy$. See Test-case 3 for definition of quantities. The results were obtained with the ‘standard’ forward-backward scheme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

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 area-averaged volume anomaly, potential enstrophy anomaly and vorticity anomaly:

$$A^{-1} V'_i(t) \equiv A^{-1} \iint |h_i - h_i(x, y, t = 0)| dx dy, \quad (6)$$

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

$$A^{-1} \omega'_i(t) \equiv A^{-1} \iint |\omega_i| dx dy. \quad (8)$$

where $A = \iint dx dy$ is the area of the model domain. These anomalies are meant to represent a typical fluctuation from the initial model state over the course of the calculation. We can think of these anomalies as the ‘signal’ that we try to simulate, 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 a seamount at its center. 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 runs with no explicit dissipation, no forcing, and the energy-conserving ‘standard’ forward-backward time-discretization (`g_fb = 0.` in file `shared_mod.f95`; see §D). The values (and figures) reported below are obtained with the parameter `rw` set to `r4` (see §B.1).

The average layer thickness (i.e. the volume of the layer normalized by the area of the basin) oscillates and stays within $\pm 5 \times 10^{-10}$ m of its initial value during the 50 days of simulation (Fig. 4a). This numerical error is eight orders of magnitude smaller than the ‘signal’, i.e. the typical fluctuations in the thickness of the layer ($O(10^{-2}$ m), Fig. 4b). The mechanical energy ($K + P$) displays oscillations $O(3\%)$ (Fig. 4c). These oscillations are primarily due to the time discretization and their magnitude would decrease if the time-step Δt were reduced (§D). If the ‘generalized’ forward-backward scheme is used instead of the ‘standard’ forward-backward scheme, the high-frequency wave motion is gradually damped (e -folding time of 75 days). This numerical dissipation is inherent to the generalized scheme and is exacerbated by the coarse resolution of the test-case and the dominance of divergent motion (see §D).

The potential enstrophy oscillates around its initial value with an amplitude $O(10^{-16} \text{ m}^{-1} \text{ s}^{-2})$ (Fig. 5a). This 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).

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 beam is called the flow relaxation scheme (Davies, 1976). In this approach, outward-propagating signals such as waves are gradually absorbed within special regions of the model domain called ‘sponge zones’. 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 mean flow in near geostrophic balance and no waves, a simple ‘no-gradient’ boundary condition with very weak

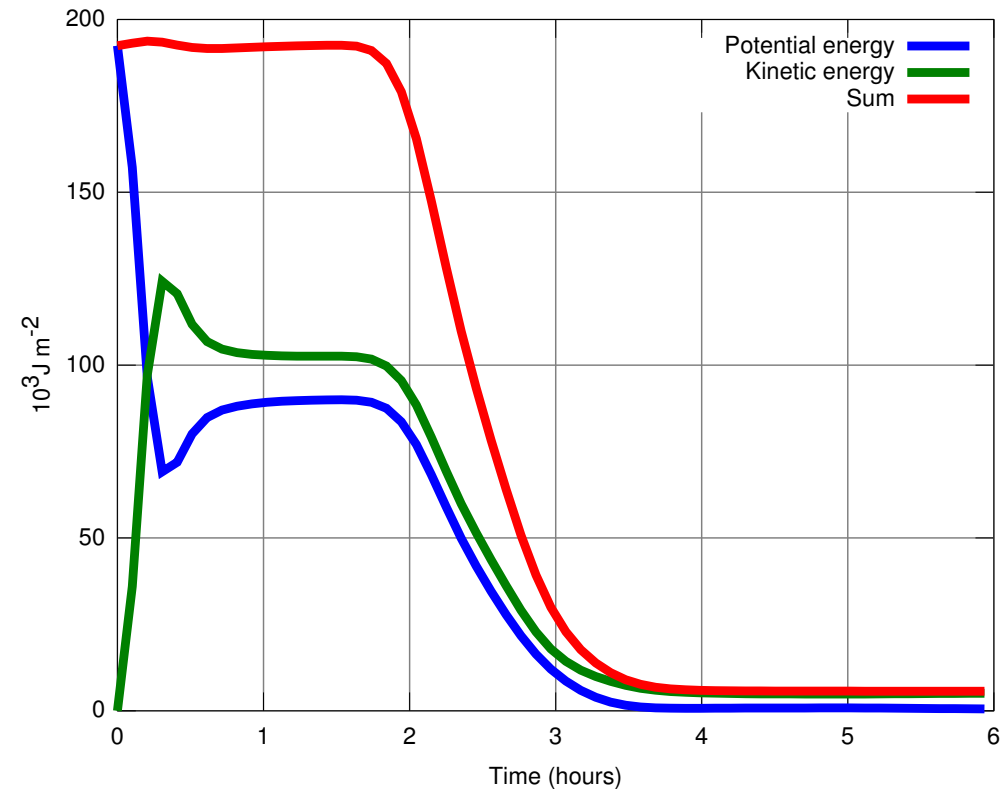


Figure 6: Absorption of the outward-propagating wavefront by the sponge zones. The residual energy corresponds to the small geostrophically-balanced mound (e.g., Lavelle and Thacker, 2008, their Fig. 3). See Test-case 4 (Wave sponge at model boundaries).

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 at least 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 (Test-case 3) by replacing the periodic boundaries with sponge zones that completely surround the domain⁴. 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 inverse of the relaxation timescale (T_{relax}^{-1}) 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).

Although the present test-case is entirely focused on the absorption of outward-propagating waves, the same mechanics (Eq. 9) can be used to prescribe a signal to be advected into the model domain (see, for example, Test-cases 7,11,12). I use the general term “nudging zone” (also known as ‘relaxation zone’ or ‘buffer zone’) to describe a zone where Eq. 9 is implemented for either purpose (absorption of outward signals or prescription of inward signals). beam identifies the presence and location of nudged open boundary segments by looking for grid cells satisfying three conditions: (1) a $T_{\text{relax}}^{-1}(x, y)$ field is provided as input to the calculation (see the file `testcases/wave_sponge.m` for an example), (2) the grid cell is associated with a value $T_{\text{relax}}^{-1}(x, y) > 0$, (3) the grid cell has a bottom depth > 0 while its neighboring grid cell has a depth $= 0$ (boundary of computational domain). This set of conditions allows for defining an arbitrary number of nudged open boundaries that can be positioned anywhere (not necessarily at the edges of the model domain).

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

⁴In more realistic model domains, it is often sufficient to have a wave sponge along only one segment of the domain’s boundaries; see Test-case #11.

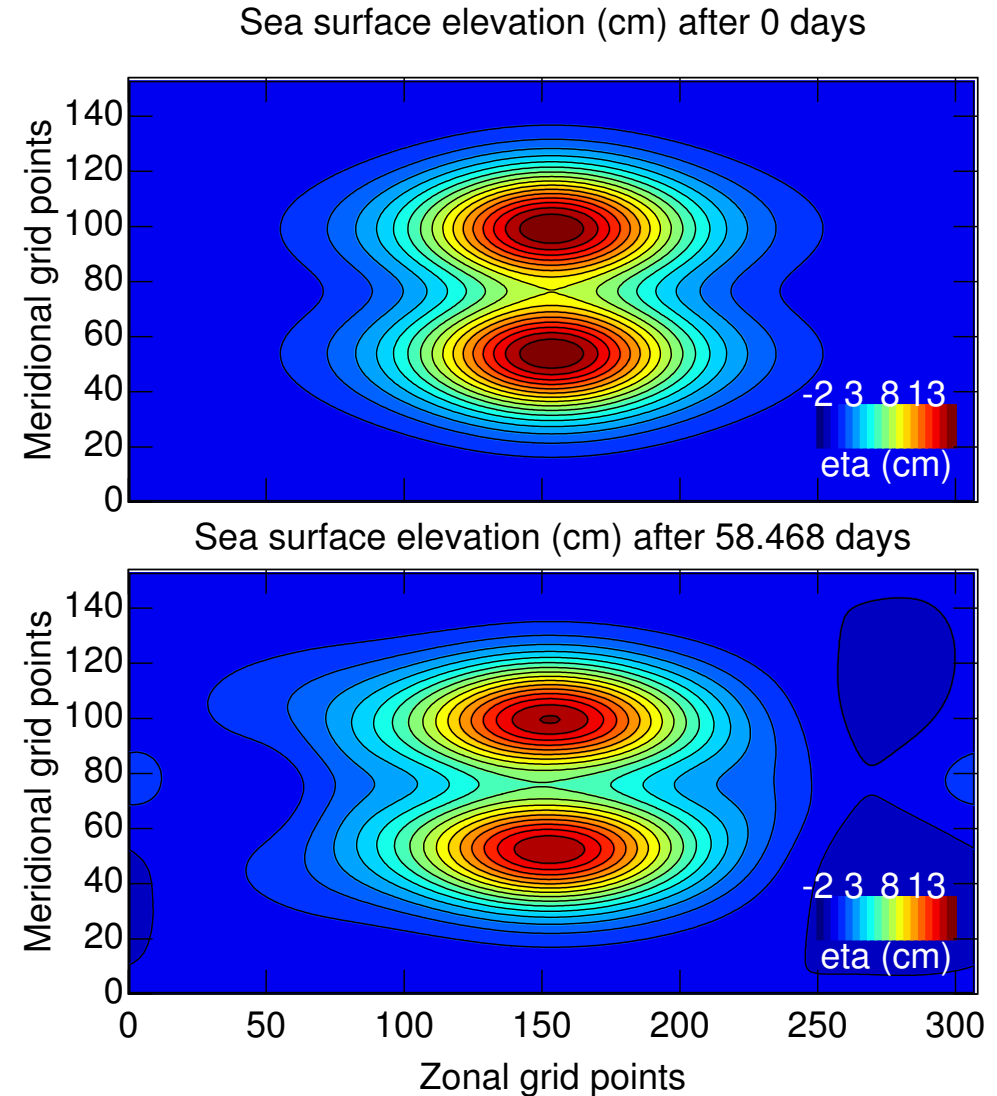


Figure 7: See Test-case on Equatorial soliton.

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 `shared_mod.f95`; see §D). The ‘standard’ forward-backward scheme cannot handle the test-case unless a strong dynamic viscosity is introduced (`dvis=1.`).

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 $U_0 =$

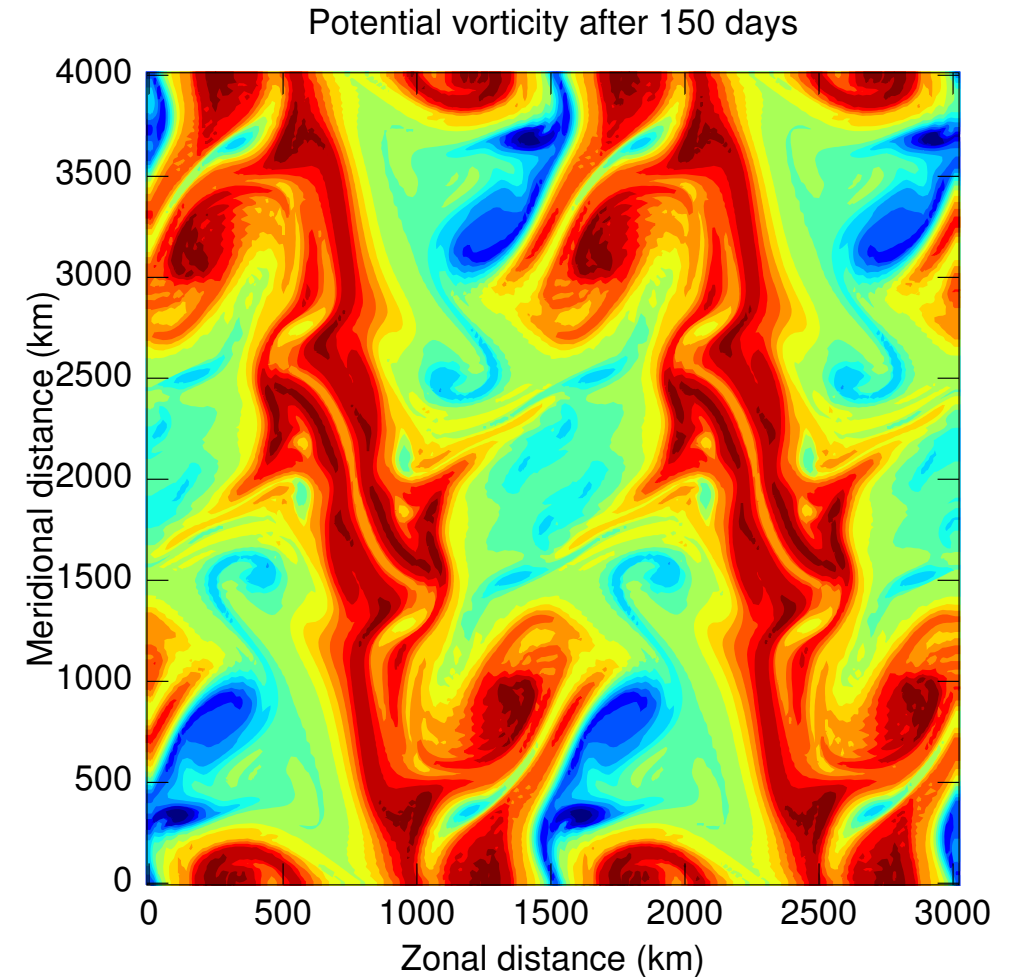


Figure 8: Two-dimensional turbulence in the test-case of barotropic instability.

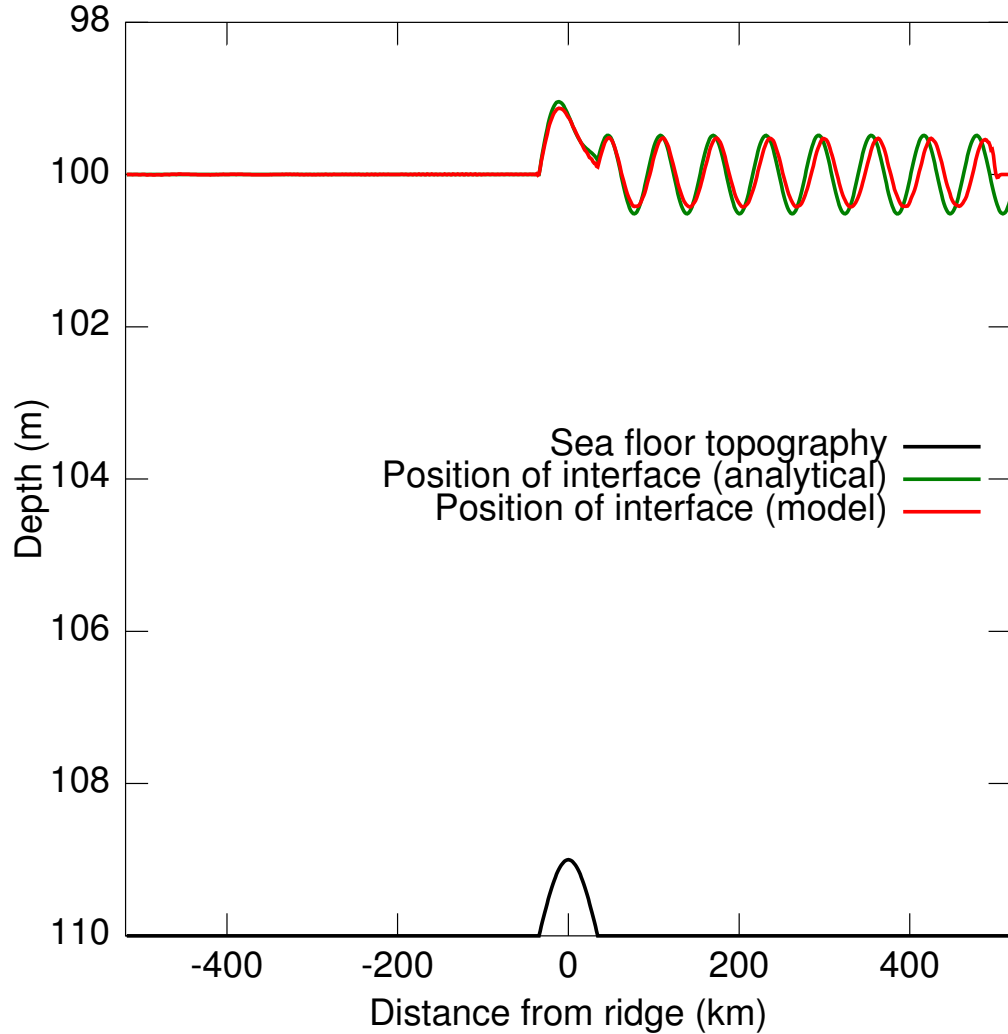


Figure 9: Test-case for stratified rotating flow over topography.

1.2 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.

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 §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 sloping walls. The test-case verifies that the numerical scheme (§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 (“inundation”). 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 §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 (Fig. 11b), 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 increasingly noisy as h_{sal} is reduced.

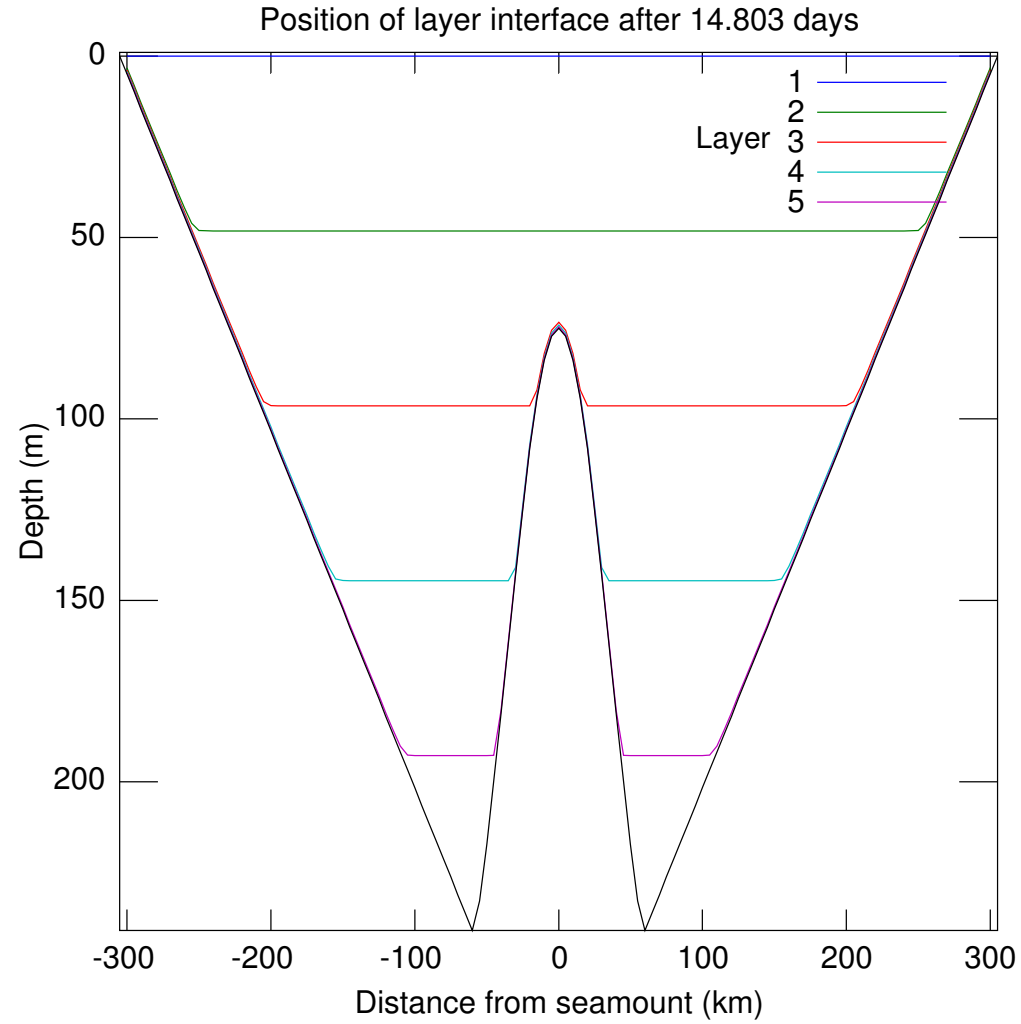
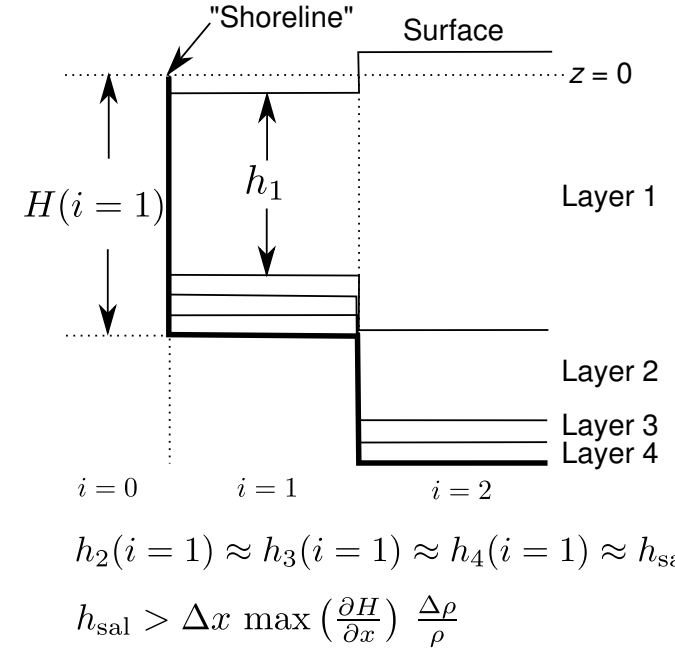


Figure 10: Position of the layer interfaces in the test-case of simple isopycnal outcropping.

a) Outcrop, no wetting-drying



b) Outcrop + wetting-drying

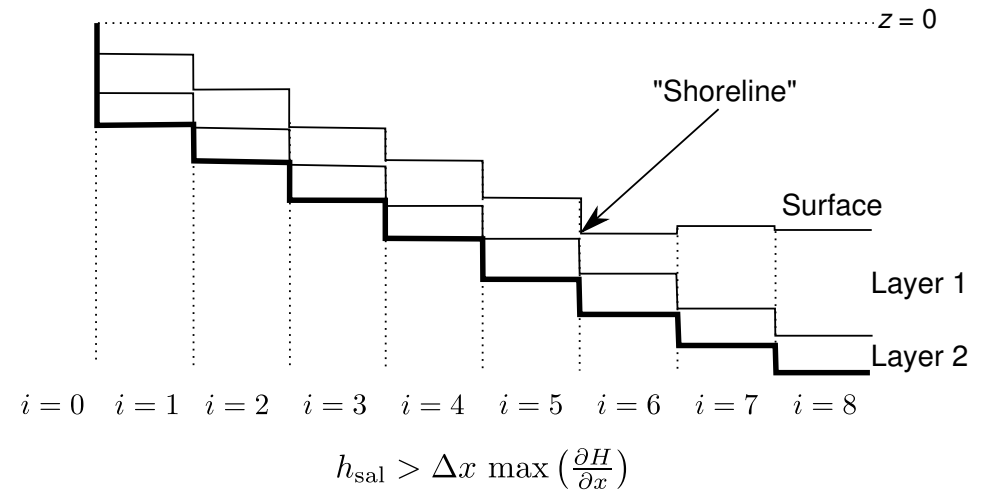


Figure 11: Comparison between simple isopycnal outcropping and wetting-drying. The scales are exaggerated.

In cases with simple outcropping, Eq. 42 becomes (assuming a two-layer stratification for simplicity) in the bottom layer:

$$-\frac{\rho_2 - \rho_1}{\rho_2} \left(H(x, y) - h_2^0(x, y) \right) - \frac{h_{\text{sal}}}{3} \left(\frac{h_{\text{sal}}}{h_2^0(x, y)} \right)^3 = C_2, \quad (13)$$

which is the same equation that led Salmon to his criterion (Eq. 12) except for the density factor. The latter allows for considerably thinner values (Fig. 11a):

$$h_{\text{sal}} > \max \left| \frac{\partial H}{\partial x} \right| \Delta x \frac{\Delta \rho}{\rho}, \quad (14)$$

where $\max |\partial H / \partial x|$ is the maximum bottom slope over the model domain and $\Delta \rho$ is a value representative of the density stratification.

Equations 12, 14 are only rules of thumb grounded on the idea that the dynamical balance (and the relative importance of Salmon’s term within the “pressure gradient”, see Eq. 45) shouldn’t change too abruptly from one grid cell to the next. They can be used as an initial guess for h_{sal} , but this guess should be validated by running the configuration under its state of rest (no forcing, no dissipation) and by verifying that spurious currents are negligible and not growing over time (see next paragraph). Growing currents would indicate that h_{sal} should be increased. Given that Eqs. 12–14 both involve topographic slopes, users of realistic bathymetries can avoid trouble by slightly smoothing the interpolated bathymetric grid so that excessively abrupt transitions in dynamical balance are avoided.

In test-case `testcases/outcrop_seamount.m` the Salmon thickness is set to $h_{\text{sal}} = \max |\partial H / \partial x| \Delta x (\rho_2 - \rho_1) / \rho_2 = 14$ cm and the layers are a few cm thick in the areas of outcrop (Fig. 10). The maximum spurious currents are $O(10^{-5} \text{ m s}^{-1})$ and the potential enstrophy of the layers is constant (within machine precision). The kinetic energy of the model domain remains stable over the 15 days of simulation without any recourse to dissipation (the viscosity parameters `bvis` and `dvis` are zero). Note that Eq. 14 represents a lower bound on h_{sal} and that more realistic settings (that include vigorous flows and/or poorly resolved topography) may require thicker values. The user is expected to test the sensitivity of their results to this important parameter.

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.

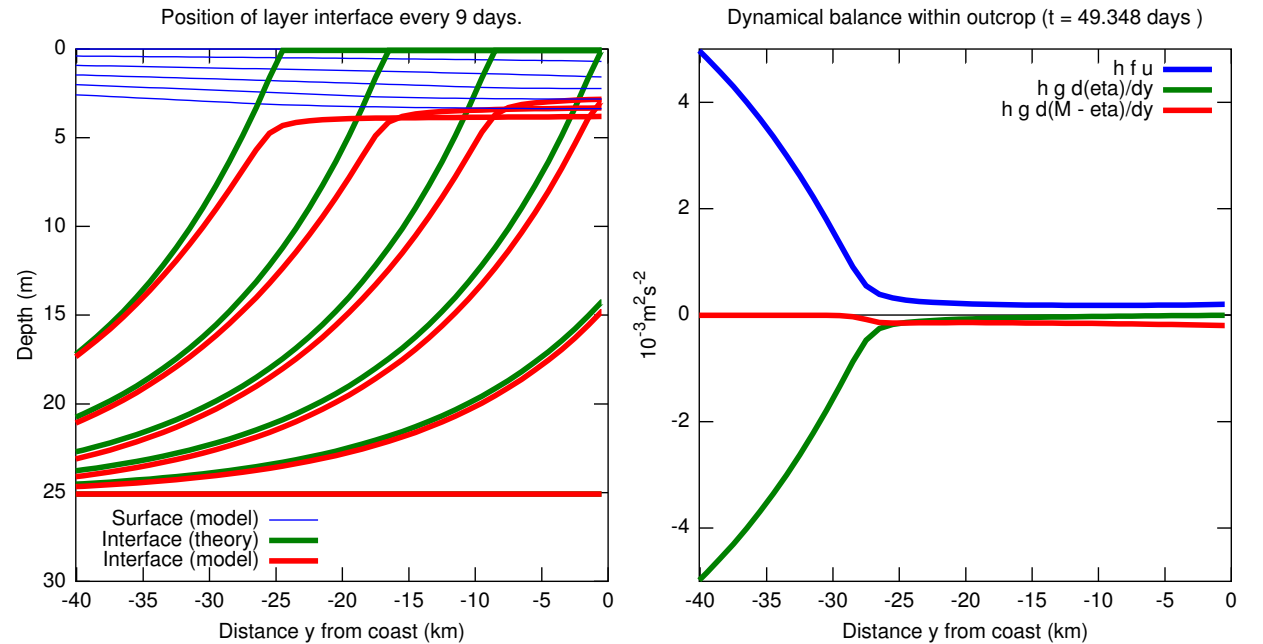


Figure 12: (left) Migration of the surface front during the upwelling event (see Test-case 10). (right) Dynamical balance within the region of outcrop.

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:

$$h \frac{\partial u}{\partial t} \approx h f v + h b_x, \quad (15)$$

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

$$h \frac{\partial v}{\partial t} \approx 0 \approx -h f u - h g \frac{\partial \eta_1}{\partial y} - h \frac{\partial}{\partial y} (M - g \eta_1), \quad (16)$$

where $M - g \eta_1$ represents the artificial term suggested by Salmon (see Eq. 45). 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 (§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, 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

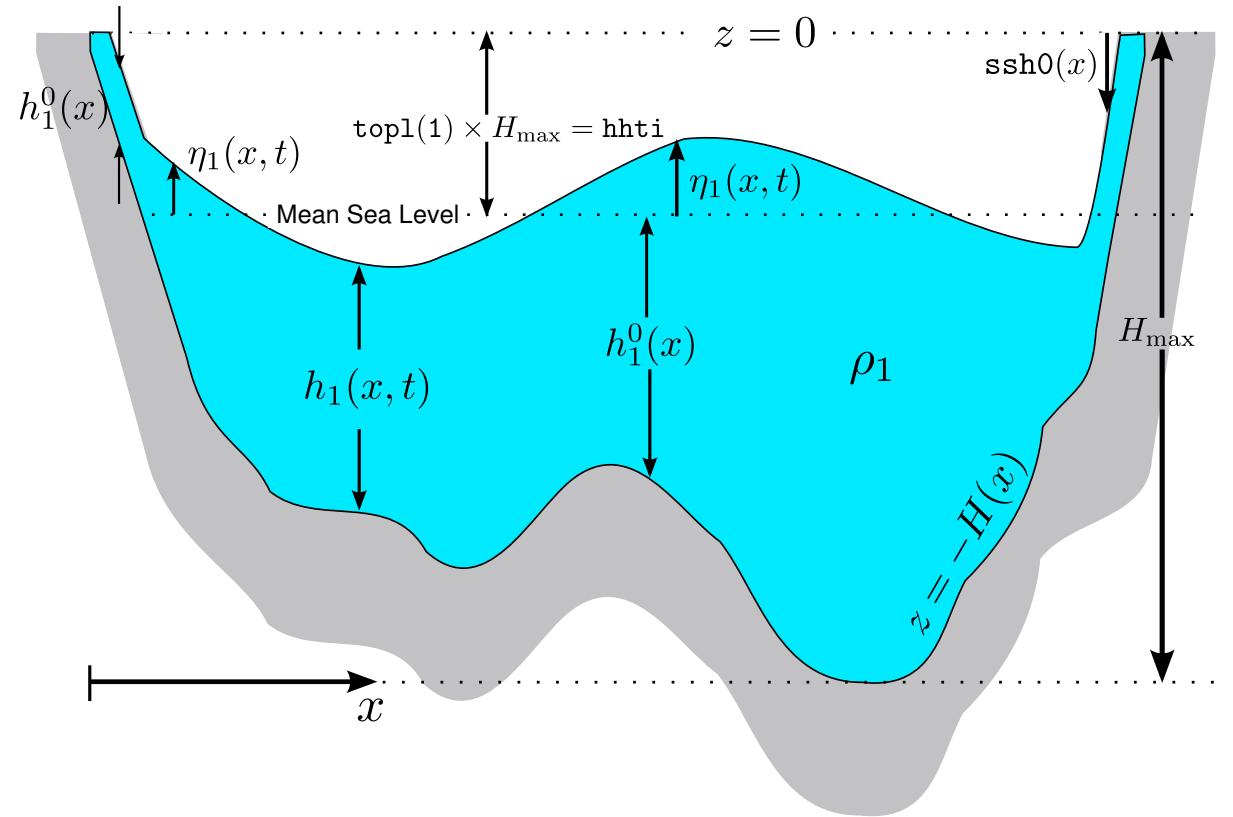


Figure 13: Basin configuration in cases with wetting-drying. Scales are exaggerated. The figure shows a case with no stratification for simplicity.

is $\leq \text{hhti}$. Finally, the top of the upper model layer ($\text{topl}(1)$) 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} (§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 $\text{ssh}0(x)$ does not coincide with the geopotential surface $z = 0$ (Figs. 11,13):

$$\text{ssh}0(x) \equiv \sum_{i=1}^{\text{nlay}} h_i^0(x) - H(x) \approx \begin{cases} -\text{hhti} & \text{away from coastlines,} \\ -H(x) + h_{\text{sal}} & \text{over ‘dry’ areas,} \end{cases} \quad (17)$$

and the top of the (undisturbed) upper layer is positioned at $\text{topl}(1) = \text{hhti} / H_{\text{max}}$. 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 $h_{\text{sal}} = 0.2 |\partial H/\partial x| \Delta x = 1.2 \text{ cm}$ (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;

$$\frac{\partial \psi}{\partial x} = 0, \quad (18)$$

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. In beam, the ‘no-gradient’ obc is applied by default at all open boundaries that are not periodic (see the discussion of Test-case 4 on how beam identifies the existence and location of non-periodic open boundaries).

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

$$\frac{\partial \psi}{\partial t} = \dots + \frac{\psi_{\text{ext}} - \psi}{T_{\text{relax}}}, \quad (19)$$

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

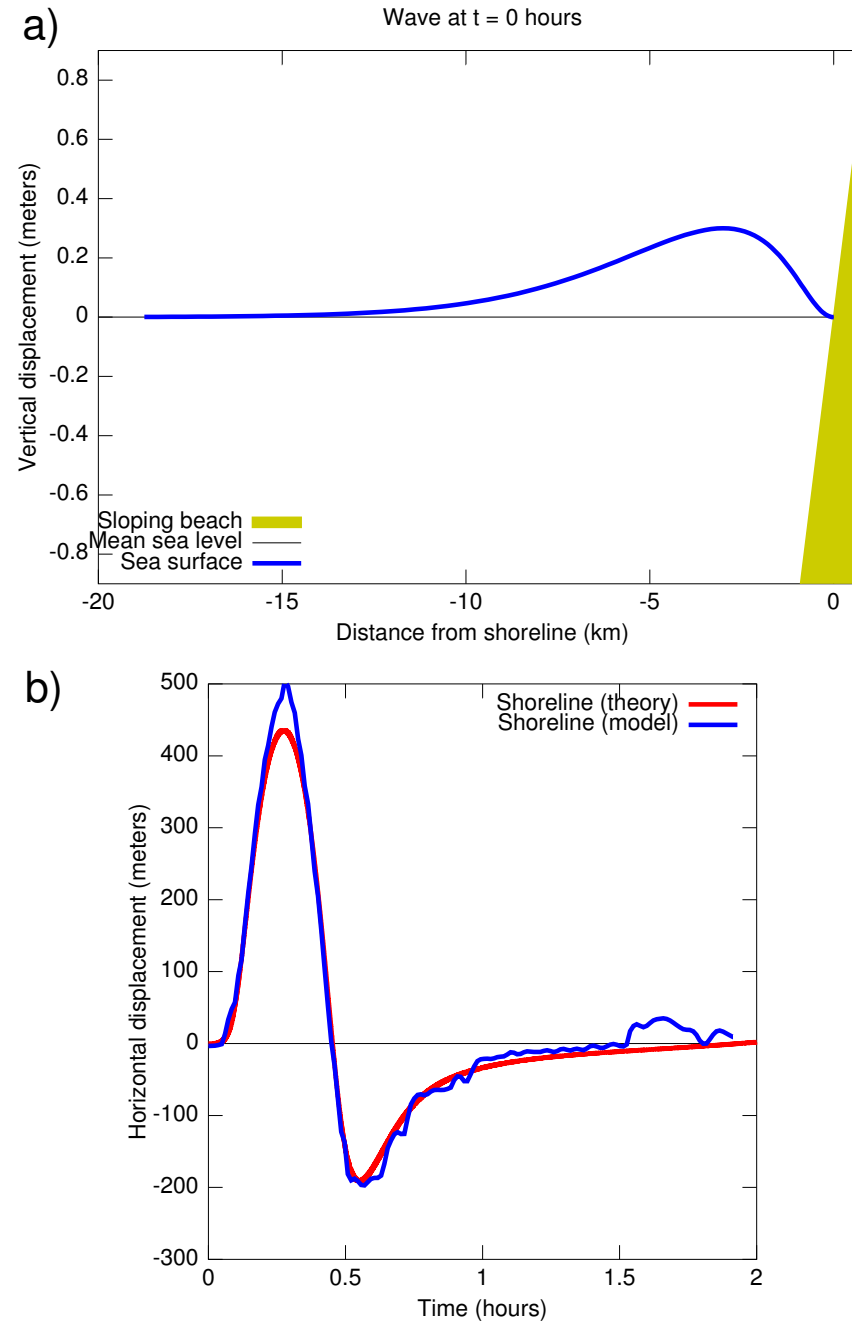


Figure 14: Test-case for wetting and drying on a sloping beach. (a) Initial displacement $\eta_1(x, t = 0)$ of the sea surface. (b) Model-theory comparison for the horizontal displacement of the shoreline.

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

$$\psi_{\text{ext}} : \quad \eta_i = 0, \quad \mathbf{u}_1 = -\mathbf{e}_3 \times \frac{\boldsymbol{\tau}_{\text{wind}}}{f \rho_0 h_1}, \quad \mathbf{u}_2 = 0. \quad (20)$$

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 any wave that would be 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. 18,19,20) 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:

$$u^{\text{tide}} = u_0 \cos(\phi_u - \omega t), \quad v^{\text{tide}} = v_0 \cos(\phi_v - \omega t), \quad \eta_1^{\text{tide}} = \eta_0 \cos(\phi_\eta - \omega t), \quad (21)$$

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.

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 experiment is sometimes referred to as a ‘dam break’. The density contrast between the two fluids forces a slow

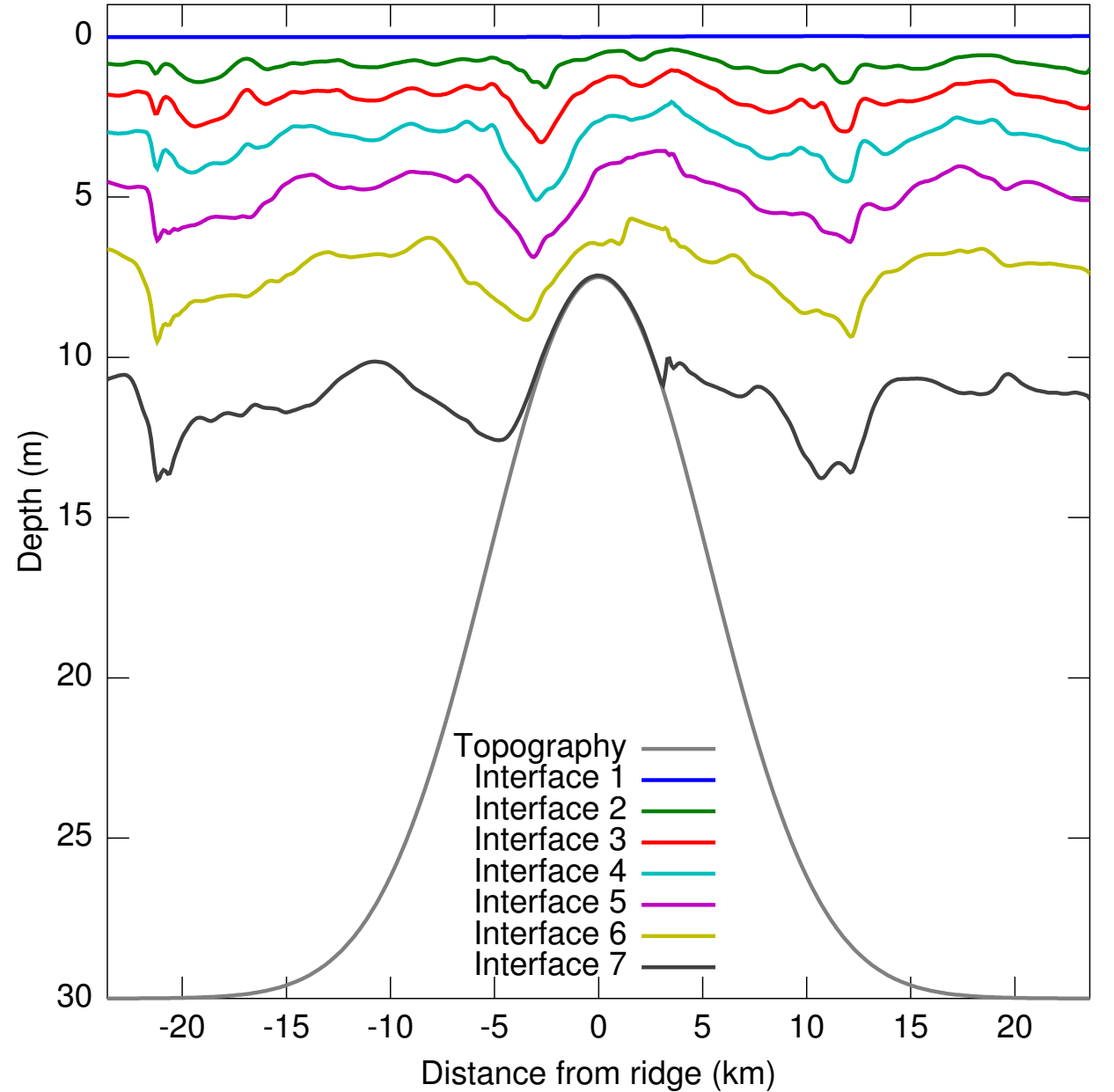


Figure 15: Test-case for tidal flow over a ridge.

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

$$\frac{1}{2} \sqrt{g H \Delta \rho / \rho_0} \quad (22)$$

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).

Test-case 14 Remotely-forced current

The script `testcases/remote_current.m` can be used as a starting point for a generic three-dimensional regional continental shelf/slope domain with a slope current (or a coastal current) having specific characteristics. The current is assumed to be remotely-forced, i.e., generated *outside* of this regional model domain, by larger-scale forcings (e.g., Antarctic westerlies) not represented here. The domain is a f -plane in the Southern Hemisphere ($f = \text{constant} < 0$) and has no topography (for the sake of simplicity). The domain is open on all four sides with the remote current (RC) flowing to the left (“westward”, $u < 0$). The characteristics of the RC are prescribed (and maintained over time) at the upstream boundary (eastern side).

To minimize the computational cost of the test-case, the bottom depth is set to an unusually shallow value ($H = 10$ m) and the stratification is unusually strong (two layers with $\Delta \rho = 30 \text{ kg m}^{-3}$). The current in the top layer is parameterized as:

$$u_1(y) = -u_{\max} \exp\left(-\frac{1}{2} \frac{y^2}{(\gamma L_R)^2}\right), \quad (23)$$

where u_{\max} and γ define the speed and width of the RC (respectively). The current is assumed to be barotropic ($u_2 = u_1$, $\eta_2 = 0$, $v_1 = v_2 = 0$) and in geostrophic balance with a tilted free surface:

$$u_1 = -g f^{-1} \partial \eta_1 / \partial y. \quad (24)$$

This configuration is used as the initial condition and also the condition toward which the model fields are relaxed inside the nudging zones.

The script defines the relaxation/nudging applied to the model variables (η , u , v) at the four open boundaries. At the southern and western sides, a weak relaxation (timescale $T_{\text{relax}} \approx O(1 \text{ month})$) is applied on all three variables and this relaxation decreases linearly away from the boundary over a distance of 15 grid points. The purpose of the weak relaxation is to prevent the model variables from drifting over time.

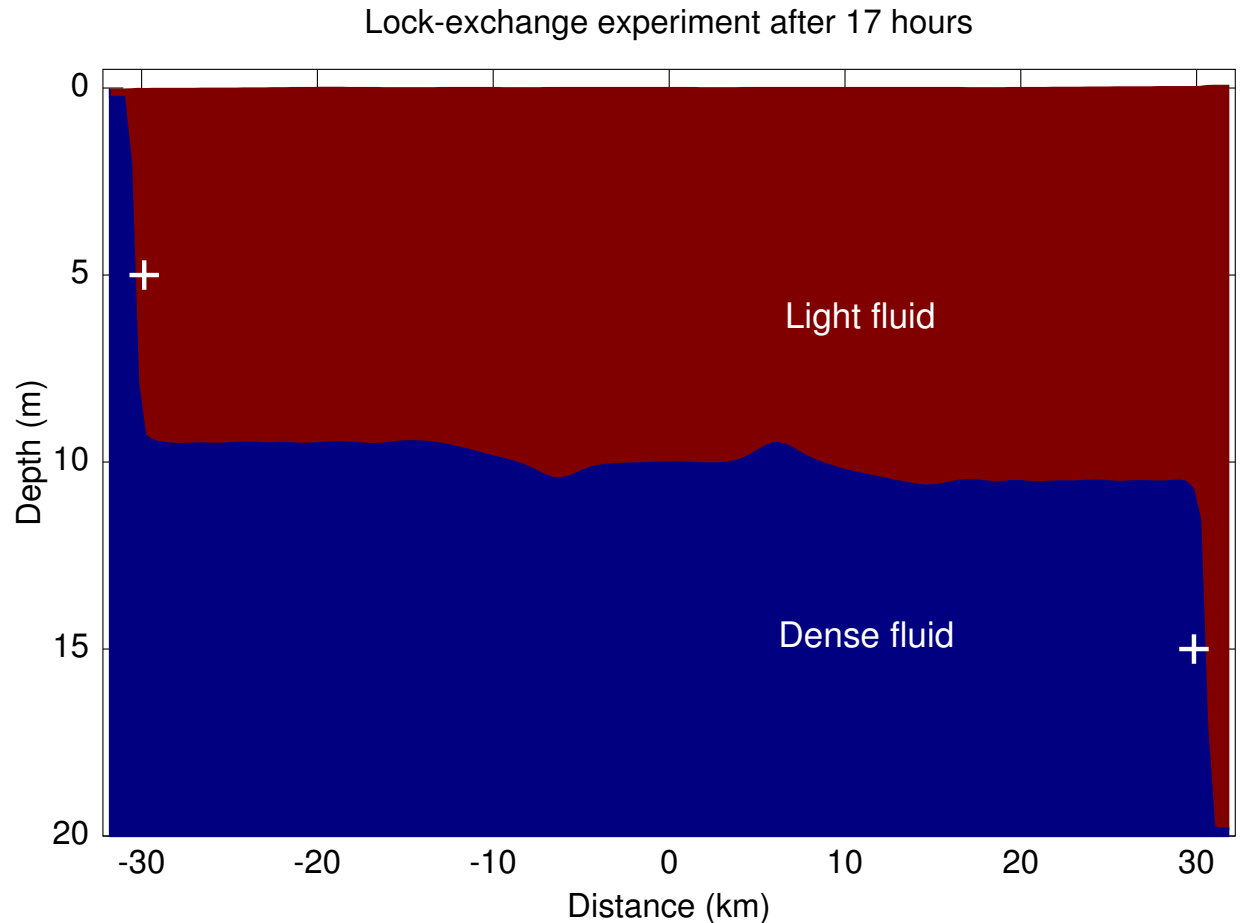


Figure 16: Test-case for the lock-exchange experiment. The symbol + marks the theoretical position of the front (Eq. 22).

At the eastern side ($x = L_x$), the same weak relaxation is applied everywhere except within the RC. There, the relaxation is strengthened to:

$$T_{\text{relax}}(x = L_x, y) \approx \Delta t \frac{|u_1(y)|}{u_{\text{max}}}, \quad (25)$$

where Δt is the model time-step. The stronger relaxation ensures that the characteristics of the RC are faithfully reproduced and maintained over time. The relaxation decreases away from the boundary as in Eq. 10. Finally, a wave sponge (see test-case #4) is applied at the northern side so that a gravity wave produced during the simulation can safely leave the domain.

The key feature apparent in the first days of simulation is the current adjusting itself from a strict geostrophic balance at the upstream boundary ($\eta_2 = 0$ and Eq. 24) to a more realistic universe that includes ageostrophic terms (notably a small viscosity $\mu \neq 0$). The adjustment is small (deviations in η_2 are $O(1 \text{ cm})$) but generates gravity waves that eventually disappear through the wave sponge at the northern boundary. The adjustment process is also accompanied by the formation of vortices within the current after day 20. The vortices travel westward with the current and disappear through the western boundary around day 70. After approximately 100 days, the domain reaches a quasi-steady state where the RC maintains its characteristics indefinitely with minor deviations from the state prescribed at the upstream boundary. Note that the test-case is set for a duration of only 10 days in its default configuration ($\text{dt_s}=10$).

Test-case 15 Locally-forced slope current

The script `testcases/easterlies_current.m` can be used as a starting point for a generic regional continental shelf/slope domain with a flow driven by the *local* winds (in contrast with test-case #14). The setup of the test-case is similar to that of Stewart and Thompson (2012) but with major simplifications (described below). The domain is a f -plane in the Southern Hemisphere ($f = \text{constant} < 0$) and uses a hyperbolic tangent function to mimic the cross-shelf bathymetry. The domain is periodic in the along-slope direction, closed at the onshelf (southern) boundary, and includes a wave sponge (see test-case #4) at the offshelf (northern) boundary. The sole forcing is a time-invariant zonal wind stress applied over the model domain (Fig. 17a) mimicking the strong westerlies offshore ($\tau_x > 0$) and the comparatively weaker Antarctic easterlies over the continental slope ($\tau_x < 0$). A weak linear bottom friction ($1.1 \times 10^{-3} \text{ m s}^{-1}$) is the primary mechanism for the dissipation of kinetic energy. The test-case assumes a very weak background stratification with a potential density profile of the form:

$$\rho(z) = \rho_{\text{bottom}} - \Delta\rho \exp(z/H_{\text{strat}}), \quad (26)$$

where z is positive upward, $\rho_{\text{bottom}} = 1027 \text{ kg m}^{-3}$ would be the density at the ocean's bottom, $H_{\text{strat}} = 1000 \text{ m}$ is a e -folding scale for this exponential profile, and $\Delta\rho = 1 \text{ kg m}^{-3}$ is the assumed density contrast between the surface and the bottom. This continuous density profile is then discretized over six layers each representing a density class of width $\Delta\rho/6$. With this choice, the thickness of the layers in their state of rest varies between 183 m and 1208 m (Fig. 17c). To minimize the computational cost of the test-case, the domain is limited to one grid point in the zonal dimension ($\text{lm}=1$) which makes it a 2.5-D domain (the $\frac{1}{2}$ refers to the fact that non-zero velocities along the missing dimension (x) are allowed, i.e. $u \neq 0$). Additional simplifications from the setup of Stewart and Thompson (2012) include the absence of $\partial f/\partial y$, surface heat fluxes, or dense water formation on the shelf.

The test-case is initiated from rest and the wind forcing is gradually introduced (“ramped”) over a period of one day ($\text{dt_r}=1$) to minimize the formation of inertial waves (e.g., Millot and Crépon, 1981). Over the following days, the wind stress accelerates the upper layers and produces a surface-intensified eastward current at $y = 1400 \text{ km}$ (“Antarctic Circumpolar Current”, ACC) and a surface-intensified westward current at $y = 500 \text{ km}$ (“Antarctic Slope Current”, ASC; see Fig. 17c). Simultaneously, Ekman pumping generated by $\partial\tau_x/\partial y$ leads to upward/downward transport and a corresponding vertical displacement of the isopycnals (Fig. 17a,c). The upward and downward displacements on the two sides of the ACC are roughly symmetrical (i.e. similar in magnitude but of opposite sign). In contrast, the presence of the continental slope strongly enhances the downward displacement of isopycnals on the southern side of the ASC. Note that the layers’ interface at $y = 0 \text{ km}$ and at $y = 2000 \text{ km}$ are mostly undisturbed (i.e. they remain at their initial depth) which is consistent with the lack of wind forcing ($\tau_x \rightarrow 0$) at these locations.

The sea surface elevation reaches a quasi-stationary state after approximately 750 days, consisting of a northward slope within the ACC and a steep gradient within the ASC (Fig. 17b). (Note that the default duration of the test-case is only 20 days; $\text{dt_s}=20$.) On the other hand, the baroclinic field does not show sign of reaching a stationary state as the isopycnals become steeper and steeper over time (and start to outcrop at the surface; see layer 1 in Fig. 17c) in response to Ekman pumping. This unrealistic runaway behavior is a consequence of the 2.5-D approximation ($\text{lm}=1$) that inhibits the mechanism through which the ocean normally releases its potential energy: baroclinic instability and the formation of mesoscale vortices (“eddies”). Eddies play an essential role in the 3-D setup of Stewart and Thompson (2012) and in more realistic models of the ACC that include the oceanic ridges (Ward and Hogg, 2011). These essential processes can be activated by simply increasing the value of lm inside `testcases/easterlies_current.m` and re-running the test-case (albeit at a much higher computational cost).

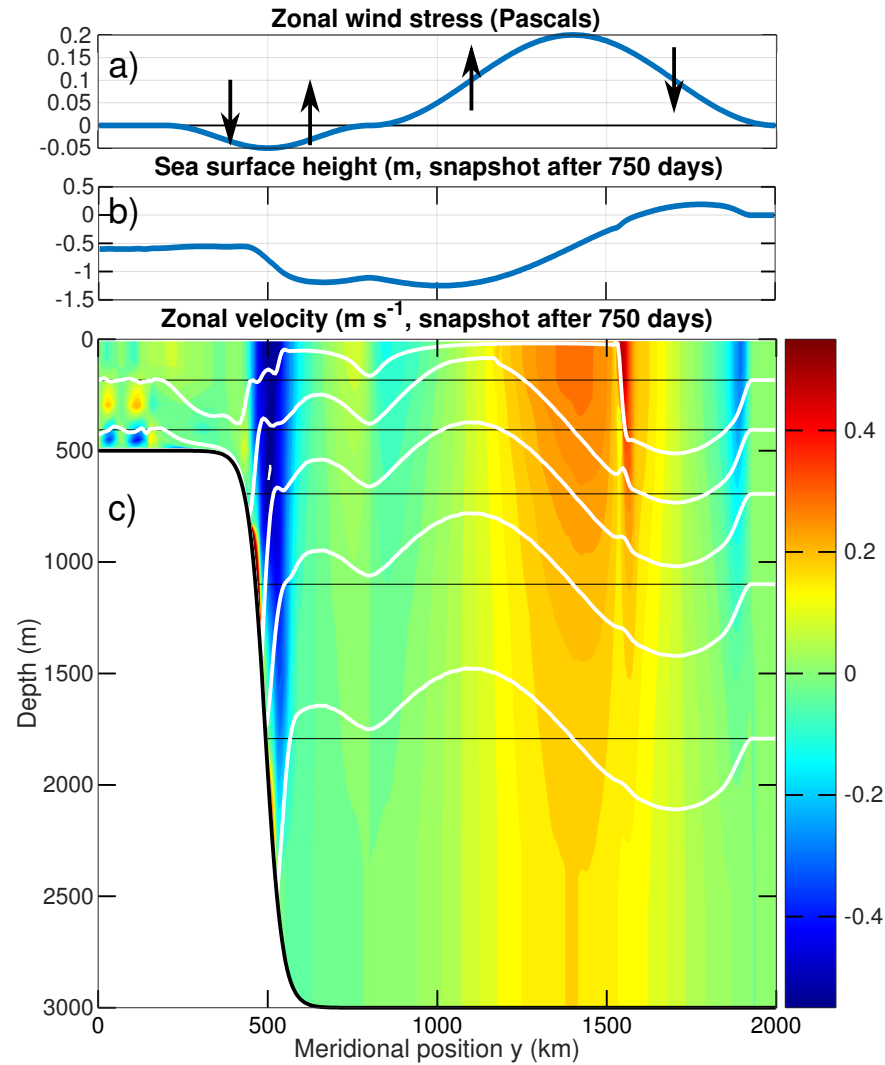


Figure 17: Test-case for a locally-forced slope current. (a) Constant wind stress forcing applied on the model domain, with black arrows representing upward/downward Ekman pumping. (b) Sea surface height after this variable has reached a stationary state. (c) Snapshot of the zonal velocity u (shading, in m s^{-1} , positive eastward) and the position of the layers' interface (thick white lines) for the same date as in (b). The thin horizontal black lines represent the position of the layers' interface at $t = 0$ days. Note the correspondence between the vertical displacement of the interfaces and the regions of upward/downward Ekman pumping in (a).

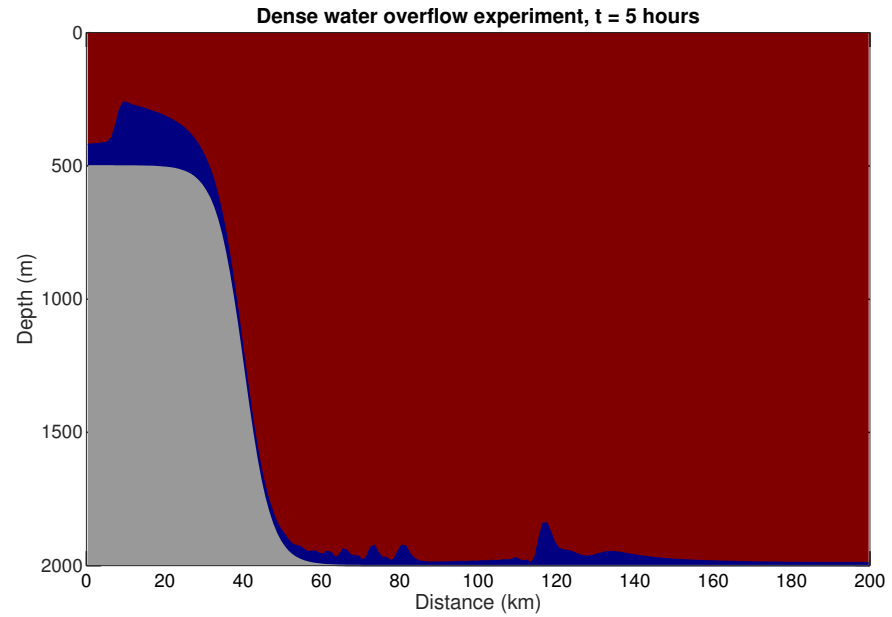


Figure 18: Test-case for dense water overflow on a continental slope (test-case 16).

Test-case 16 Dense water overflow

The test-case is from Ilıcak et al. (2012) and represents a cascade of dense water in a non-rotating ($f = 0$), 2-D (x - z) shelf-slope system with no transverse velocities ($v = 0$). The topography consists of a 500 m deep continental shelf, a 2000 m deep abyssal plain, and a continental slope with maximum slope of 11% (Fig. 18). The initial condition has the water at rest ($u = v = 0$) and homogeneous water everywhere ($\rho = \rho_1 = 1028 \text{ kg m}^{-3}$) except for a 'reservoir' of denser water ($\rho_2 = 1030 \text{ kg m}^{-3}$) occupying most of the continental shelf. The interface separating the two layers is initially vertical and thus defines a 'lock-exchange' experiment with a non-flat topography. No analytical solution are available for this test-case.

For this test-case, the top of layer 2 at the deepest location of the model domain (array `top1` in `shared_mod.f95`) is positioned one Salmon's thickness (h_{sal}) above the seabed. Therefore, layer 2 is initially "absent" from the abyssal plain and continental slope. On the continental shelf, the top of layer 2 is positioned $5h_{\text{sal}}$ away from the surface, leaving enough room for layer 1 that Salmon's artificial term is dynamically inactive on the continental shelf, and thus it doesn't influence the speed at which layer 2 is being drained out of the shelf. This configuration mimics the initial condition of Ilıcak et al. (2012).

After the calculation begins, layer 2 reaches the shelf break within a few hours and forms a

cascade down the slope. As pointed out by Haidvogel and Beckmann (1999) regarding a similar model configuration, the vertical velocities within the cascade are such that the dynamics are fundamentally non-hydrostatic and lie outside of beam’s domain of validity. Nevertheless, the cascade proceeds until the ‘reservoir’ on the shelf is emptied and all of layer 2 has reached the abyssal plain (dt_o is set to one day in the test-case). The cascading dense water forms bores when it reaches the flat abyssal plain, qualitatively similar to descriptions by Ilıcak et al. (2012) and Haidvogel and Beckmann (1999). The test-case includes a nudging zone at the end of the abyssal plain to allow the dense water to exit the domain gracefully.

The test-case doesn’t play to the strengths of Salmon’s approach. The code requires an unusually thick Salmon’s thickness ($h_{sal} = 10.3$ m, considerably thicker than the rule of thumb (Eq. 14) would suggest) in order to survive the cascading dense water. Although 10.3 m is equivalent to only 0.5% of the water depth in the abyssal plain, this value is likely large enough to influence the rate at which the dense reservoir is being drained. A much thinner h_{sal} would be possible if different algorithmic choices were made. For example, Ilıcak et al. (2012) used monotonicity-enforcing advection schemes (MPDATA for ROMS, PLM for GOLD) and an unusually short timestep for their experiments (1s for the barotropic mode). While such choices benefitted their experiments, they go against the original focus and goals of Salmon (2002) (low diffusivity, low computational cost, conservativeness).

Test-case 17 Double gyre

The “double gyre spin-up experiment” mimics the subtropical and subpolar gyres with a flat, 2-layer, rotating ocean driven by meridionally-varying zonal winds. The experiment has appeared in multiple forms in the literature and the test-case specifically follows the configuration of Salmon (2002). The domain is rectangular, closed on all four sides, 4000 m deep, and has a linearly-increasing Coriolis parameter $f = \beta y$ (the southern boundary corresponds to the equator). We assume no normal flow at the lateral boundaries and free slip at lateral/bottom boundaries. The two layers have potential densities of $1030(1 - 2 \times 10^{-3})$ and 1030 kg m^{-3} (respectively), and layer 1 occupies the upper 100 m in the state of rest. The zonal winds vary sinusoidally with y : the southern third of the domain corresponds to easterlies, and the northern 2/3 to westerlies. This wind distribution eventually gives rise to a cyclonic gyre in the northern portion of the domain and an anticyclonic gyre in the southern portion (hence the “double gyre”).

The configuration of Salmon (2002) features a coarse 100×200 grid and a mesh size $\Delta x = \Delta y = 40$ km, or approximately twice the baroclinic Rossby radius of deformation ($L_{\text{Rossby}} = 18.7$ km). Salmon parameterizes the unresolved scales by prescribing (in both layers) a linear friction of the form $\partial \mathbf{u} / \partial t = \dots - \lambda \mathbf{u}$, with λ chosen so that the Stommel boundary layer on the western edge

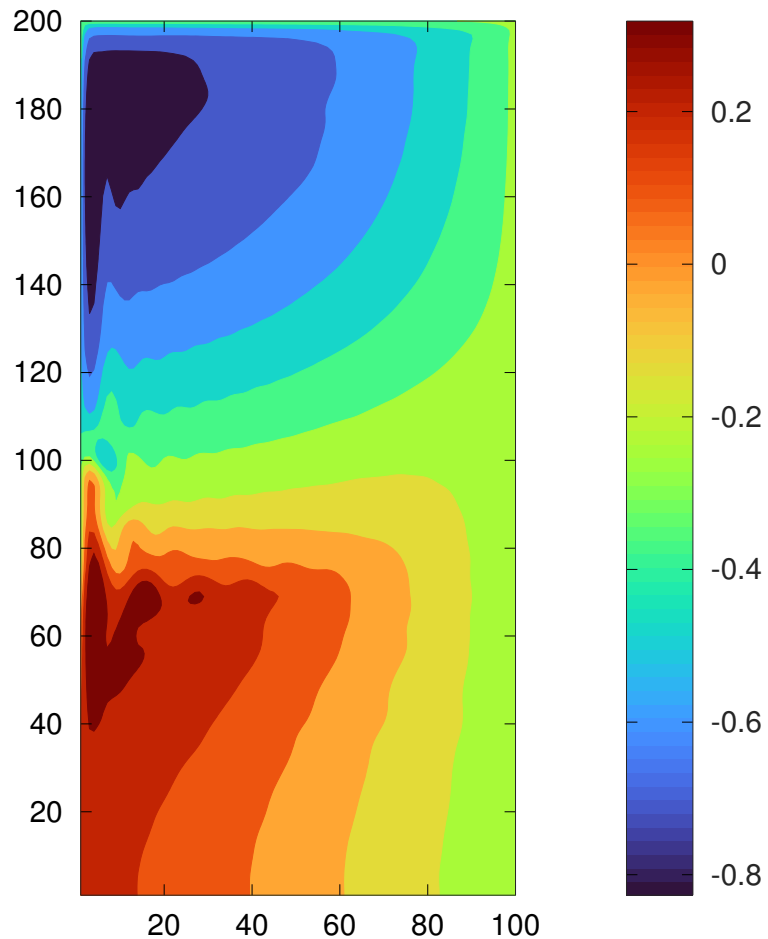


Figure 19: Test-case for the “double gyre” (test-case 17). The figure illustrates $g^{-1} M_1$ (units: meters) averaged over the duration of the calculation (12 years). The model domain has 100×200 grid points and a mesh size $\Delta x = \Delta y = 40$ km. The top (bottom) gyre mimics the subpolar (subtropical) gyre of the North Atlantic Ocean.

of the domain is just resolved (1.5 grid cells thick). Such a Rayleigh drag is not implemented in beam and thus the test-case departs from Salmon (2002) by instead using a dynamic viscosity (`dvis=0.1` in `shared_mod.f95`). It produces very high viscosities within the western boundary layer, $O(1000 \text{ m}^2 \text{ s}^{-1})$, and very low values in the rest of the domain, allowing for more mesoscale variability than the original Rayleigh drag. Salmon (2002) also reduces Earth’s external gravity by a factor of 20 to lengthen his model timestep by a factor of 4.5. Here, the test-case uses the normal gravity (9.8 m s^{-2}) and we simply assume that the user takes advantage of beam’s parallelization to accelerate the calculation.

The calculation starts from the state of rest while the winds are gradually introduced (‘ramped’) over a period of two days (`dt_r=2.` in `shared_mod.f95`). A video of q_1 (available on nordet.net) illustrates the spin-up of the double gyre over a period of 12 years. The initial meridional gradient of q_1 (due to β) is rapidly disturbed by mesoscale activity originating from the southern half of the eastern boundary (corresponding to “northern Africa”). The mesoscale structures propagate westward and contribute to the homogenization of q_1 in the southern half of the domain. Note that the presence of mesoscales in such a coarse grid ($\Delta x = 2.14 L_{\text{Rossby}}$) is most likely facilitated by beam’s algorithms (the sole diffusion in the layers’ thickness advection scheme is the hyperdiffusion associated with the third-order scheme; see §E.1). In the northern half of the domain, the interface gradually rises (eventually outcropping at the surface; $h_1 \sim h_{\text{sal}}$) and lead to high q_1 values throughout the subpolar gyre.

The overall “mean surface circulation” can be visualized by averaging $g^{-1} M_1$ over a long period of time (Figure 19). Note the large gradient of $g^{-1} M_1$ within the thin western boundary layer of the two gyres, and the ‘overshoot’ of the Gulf Stream at the point of separation. In the subpolar gyre, the term $g^{-1} M_1$ includes a non-negligible contribution from Salmon’s artificial term since the upper layer is outcropped in that area. Although our use of a dynamic viscosity prevents an apple-to-apple comparison with Salmon (2002), the key features of the solution appear to be consistent with his description of the test-case.

It is instructive to examine the budget of mechanical energy during the spin-up of the double gyre (Fig. 20). The budget corresponds to Eq. 38 (discussed in §A.1) applied to the whole model domain. Note that the divergence term in Eq. 38 drops out because of the closed lateral boundaries (divergence theorem), leaving us with a 3-term balance between the change in mechanical energy, the winds’ rate of work, and the dissipation associated with the sub-grid scale epipycnal viscosity. The latter term plays a significant role at all times owing to the high velocities at the western boundary as well as the coarseness of the grid used in this test-case. Over the first 4 years, the dissipation is responsible for \approx halving the net energy transfer from the winds to the ocean. The basin’s energy stabilizes around year 8, leading to the expected balance between the work done by winds and dissipation.

The budget is punctuated by oscillation of period $O(70 \text{ days})$ corresponding to the mesoscale

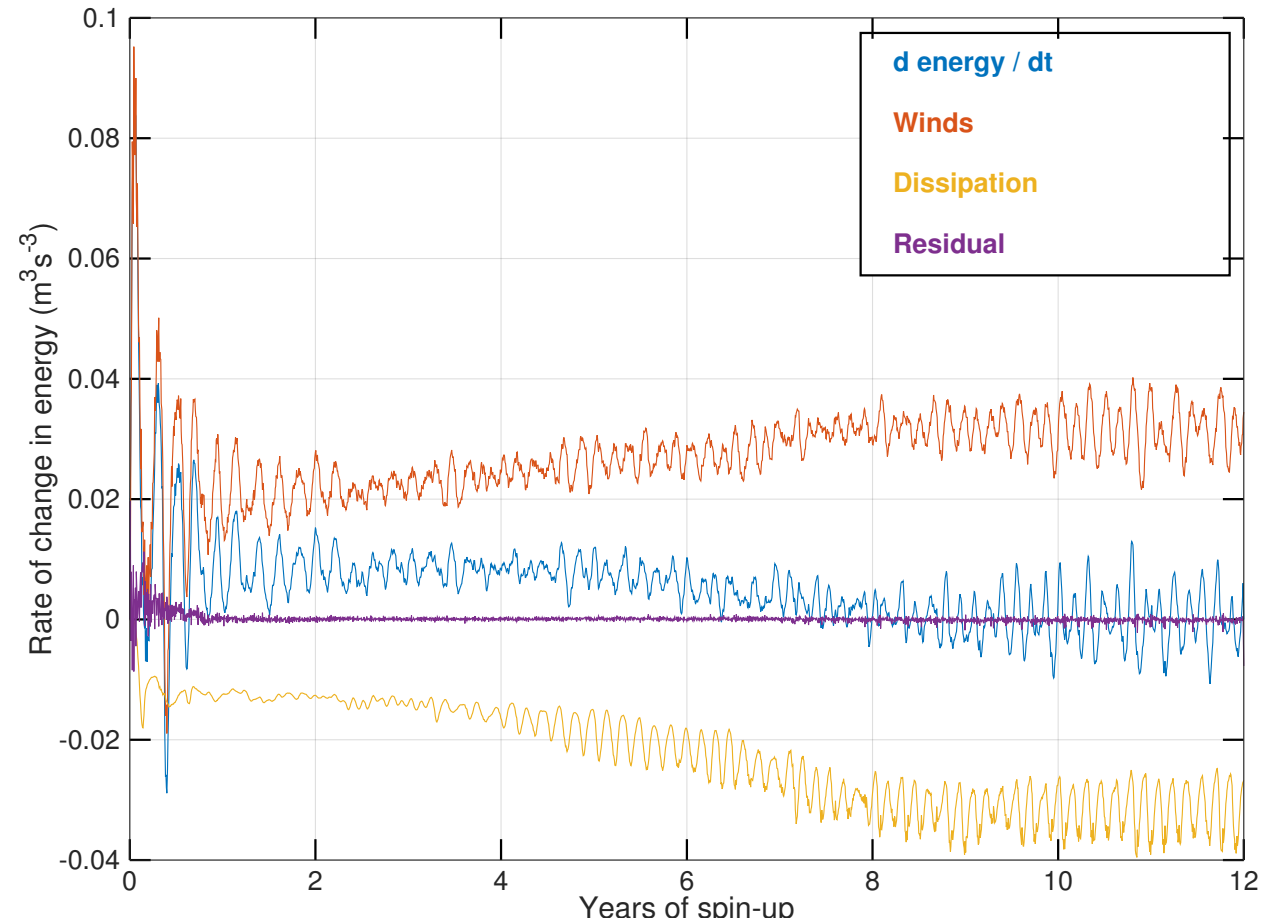


Figure 20: Test-case for the “double gyre” (test-case 17). The figure illustrates the budget of mechanical energy (Eq. 38) over the duration of the calculation (12 years). The budget is given by $\partial(\text{energy})/\partial t = \text{winds} + \text{dissipation} + \text{residual}$, with $\text{dissipation} < 0$. The budget is for the two isopycnal layers combined.

disturbances described earlier (and apparent in the video of q_1). As the disturbances modulate the flow velocities (\mathbf{u}) at the boundaries, they (in turn) leave their imprint on the wind’s rate of work and on the dissipation term (Eq. 38). Figure 20 also illustrates the imbalance in the budget (‘residual’). This term reflects the implicit numerical dissipation associated with the ‘generalized’ forward-backward scheme (§D) and with hyperdiffusion in the layers’ thickness advection scheme (§E.1). Note that in the convention of Fig. 20, an implicit dissipation will produce a positive residual. Approximations made while assembling the energy budget also contribute to the non-zero residual term. For example, in `testcases/double_gyre_nat1.m` the temporal derivatives of Eq. 38 are computed *a posteriori* from outputs saved every two days (`dt_o = 2` in `shared_mod.f95`).

The budget reveals additional features not immediately apparent in Fig. 20. When the change in mechanical energy is broken down into kinetic and potential, it becomes apparent that changes in potential energy dominate over the first 8 years of the spin-up before becoming comparable afterward. The budget terms can also be broken down for the two layers individually. The upper layer generally dominates the budget terms and tends to mask what is happening in the lower layer. The lower layer does not stabilize as quickly as the thin upper layer, and shows actually a decrease in potential energy throughout the 12 years of the calculation. In other words, more than 12 years (or additional dissipation mechanisms, such as bottom friction) would be required for the thick bottom layer to reach its stationary state. The file `testcases/double_gyre_nat1.m` provides a template on how to compute all the terms of such an energy budget.

Test-case 18 Southern Ocean

This test-case combines a realistic topography and a realistic wind stress field (albeit constant in time) to be obtained by the user from public online datasets. It represents the Southern Ocean from 90°S to 39.7°S with a user-adjustable mesh size Δl . In its default configuration, the test-case is initialized *from a state of rest* and assumes a coarse $\Delta l = 50$ km sufficient to illustrate the very first stage of the oceanic response to Southern Westerlies. The test-case offers the choice of a two-layer or a three-layer stratification and a realistic Coriolis parameter is assumed. The three-layer profile is based on World Ocean Atlas data at 53°S (central latitude of the Westerlies) and corresponds to a first baroclinic Rossby radius of 14.2 km (consistent with Chelton et al., 1998, their Fig. 6). The potential density of the three layers is 1027.1 kg m⁻³ (250 m thick), 1027.5 kg m⁻³ (750 m thick), and 1027.8 kg m⁻³ (remainder of water column). Abundant comments in the file `southern_ocean.m` discuss the steps (and pitfalls) surrounding the generation of a conformal map projection, spatial interpolation of the topography and wind stress, *etc.* The test-case is therefore a general template for creating semi-realistic configurations.

Visualization of the results in the test-case’s default configuration ($\Delta l = 50$ km, duration of 1 month) reveals a quick initial adjustment of the sea surface in response to the Westerlies. The sea surface dominates the Montgomery potential of the three layers in this early stage, indicating primarily barotropic dynamics. The baroclinic response to the winds is limited at this point to the continental slope where the easterlies are prevalent. Currents are dominated by the westward Antarctic Slope Current (driven by the easterlies) while the Antarctic Circumpolar Current (ACC) and the familiar basin-wide tilt of the isopycnals (shoaling southward) are yet to appear. If we assume a vertical tilt of 1 000 m between 53°S (*i.e.*, where the Westerlies are at their maximum and thus where the divergence of the Ekman transport shifts from negative (upwelling) to positive (downwelling)) and 40°S (northernmost extent of the Westerlies), the volume of water that must be transported across 53°S to generate this tilt is a whopping $\sim 1.5 \times 10^{16}$ m³. With a Ekman transport of $\tau/(\rho f) \approx 1.69$ m² s⁻¹ acting across the 53°S parallel (circumference $2\pi R \approx 2.4 \times 10^7$ m), the tilting of the isopycnals would require no less than 12 years of spin-up.

Ward and Hogg (2011) describe the sequence of events for the spin-up of the Southern Ocean in a configuration that shares similarities with the present test-case. The key differences are their domain geometry (a re-entrant zonal channel), more idealized topography, more idealized wind stress, and a stratification that is unrealistically strong (at least for the Southern Ocean). Their strong stratification leads to a fairly large Rossby radius of deformation that is fully resolved by their mesh size. Ward and Hogg (2011) show that the fully spun up Southern Ocean relies on form stress (aka interfacial stress) to transmit the zonal wind stress across the isopycnal layers in such a way that the Ekman dynamical balance between zonal stress and meridional transport eventually vanishes. The interfacial stress of isopycnal models like *beom* is effectively zero until mesoscale eddies begin to populate the model domain⁵, so eddies are an essential ingredient in the spin-up of the ACC. Ward and Hogg (2011) estimate that their channel configuration attains a statistical equilibrium after > 30 years of spin-up.

When the mesh size Δl of the test-case is reduced to $\Delta l = 10$ km (*i.e.*, slightly smaller than the Rossby radius of 14.2 km) the test-case becomes ‘eddy-permitting’ (but not ‘eddy-resolving’). The first eddies appear after ~ 1 year of spin-up in specific regions offshore associated with large stationary cyclonic gyres. These sites dominate the ‘eddy production’ during the first 3 years of spin-up until isopycnal slopes become sufficiently steep for the production to spread to other areas. The tilting of the isopycnals (upwelling southward of 53°S, downwelling northward of 53°S) proceeds very gradually and reaches 500 m in each direction after 5 years. By that time, the top layer (of initial thickness 250 m) has vanished over the continental slope ($h_1 \sim h_{\text{sal}}$) and the second layer (1027.5 kg m⁻³) is now exposed to the wind stress (Fig. 21).

This 5 year spin-up of the 10 km configuration is slowly evolving in the right direction but

⁵This is in contrast with traditional Earth System Models that use some variations of Eulerian vertical coordinates with relatively high sub-grid scale vertical viscosities.

it is also far from the equilibrium described by Ward and Hogg (2011). The zonal dynamical balance necessary for the ACC to emerge (Ward and Hogg, 2011) is still in its infancy (and may require a mesh size < 10 km when using the realistic 3-layer configuration) and we know from hydrographic surveys that the tilt of the isopycnals will eventually be larger (thousands of meters, e.g., Orsi et al., 1995). Ocean models are very rarely initialized from rest and more commonly use an initial density field that is already spun-up and resembles observations. The same could be done here by prescribing an initial condition with non-zero interface elevations and velocities (see, *e.g.*, Test-case 5 for an example).

Appendix A 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):

$$\frac{\partial \mathbf{u}}{\partial t} = -q \mathbf{e}_3 \times h \mathbf{u} - \nabla_h B + \frac{1}{\rho_0} \frac{\partial \boldsymbol{\tau}}{\partial z} - \nabla \cdot \overline{\mathbf{u}' \mathbf{u}'} \quad (27)$$

where \mathbf{u} is the Reynolds-averaged and layer-averaged horizontal velocity vector (whose components u, v are aligned with the grid's x, y directions), 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 ($N = 2$) for simplicity, and decompose the thicknesses as:

$$h_2(x, y, t) = h_2^0(x, y) + \eta_2(x, y, t), \quad (28)$$

$$h_1(x, y, t) = h_1^0(x, y) + \eta_1(x, y, t) - \eta_2(x, y, t), \quad (29)$$

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:

$$\frac{\partial h_i}{\partial t} = -\nabla \cdot h_i \mathbf{u}_i. \quad (30)$$

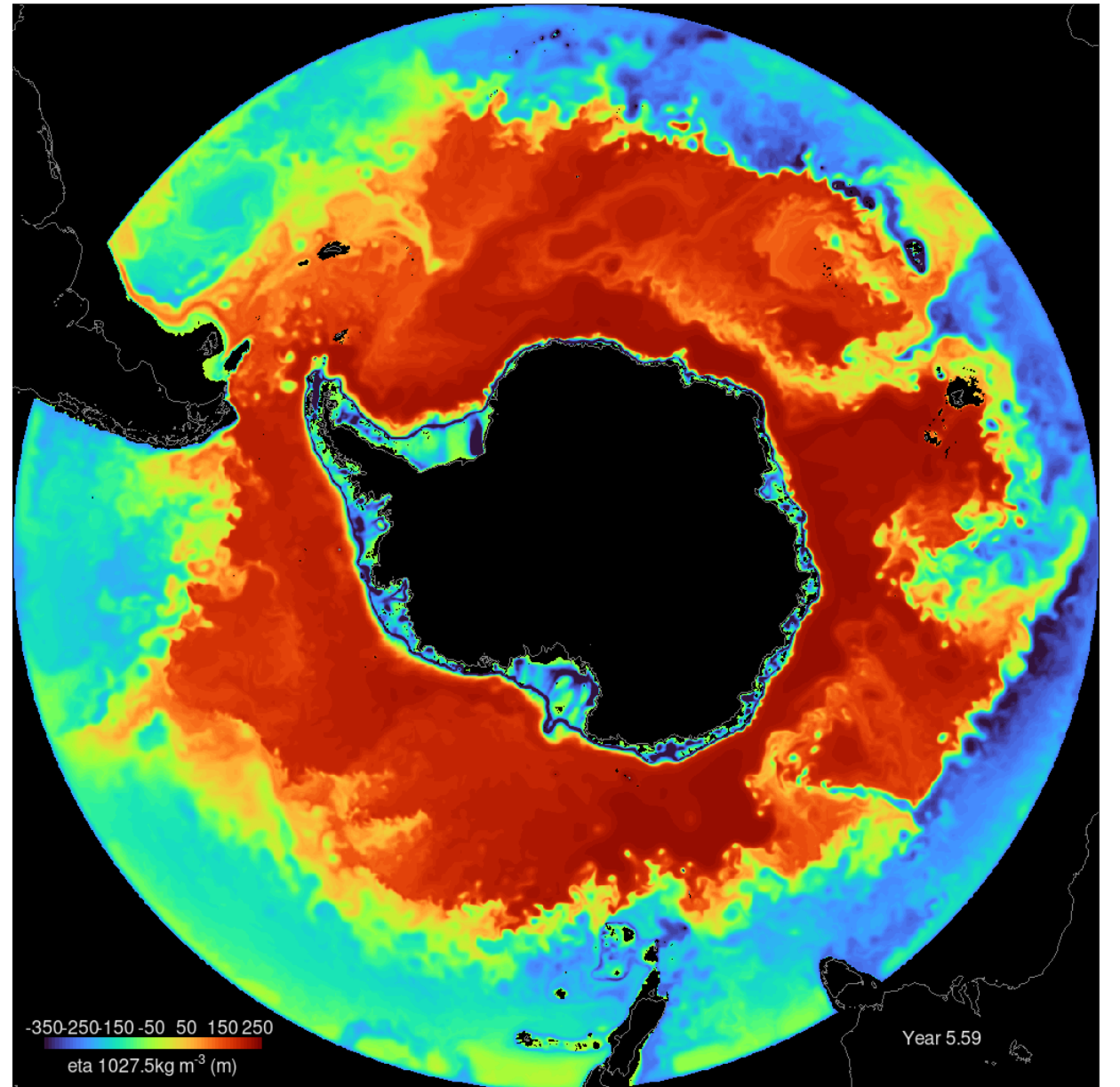


Figure 21: Elevation of the second interface (η_2) after 5 years of spin-up from an initial state of rest (Test-case 18). The interface is initially at a depth of 250 m (corresponding to $\eta_2 = 0$ m). Dark red corresponds to areas where the top isopycnal layer has vanished (outcrop).

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

$$V \equiv \iiint \rho g z \, dz \, dx \, dy, \quad (31)$$

$$\frac{M_i}{g}(x, y, t) = \frac{1}{\rho_i g} \frac{\delta V}{\delta h_i} = -H(x, y) + \sum_{k=1}^{N=2} \frac{\min(\rho_i, \rho_k)}{\rho_i} h_k(x, y, t) \quad (32)$$

$$= \eta_1(x, y, t) + \sum_{k=1}^{N=2} h_k^0(x, y) - H(x, y) - \sum_{k=1}^{i-1} \frac{\rho_i - \rho_k}{\rho_i} h_k(x, y, t). \quad (33)$$

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 §C). The temporal and spatial discretization of Eqs. 27,30 are described in Appendices D and E.1. Following Adcroft and Marshall (1998), the divergence of Reynolds’ stresses is parameterized as:

$$\begin{aligned} -\nabla \cdot \overline{\mathbf{u}'\mathbf{u}'} &\equiv \mathbf{e}_1 \left[\delta_x (\mu_{cc} D) - \delta_y (\mu_{ll} \zeta) \right] / \Delta l \\ &+ \mathbf{e}_2 \left[\delta_x (\mu_{ll} \zeta) + \delta_y (\mu_{cc} D) \right] / \Delta l \end{aligned} \quad (34)$$

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 μ_{cc}, μ_{ll} is the sub-grid scale eddy viscosity computed at cell-center (cc) or at the lower-left corner (ll), respectively. In simple cases where μ is constant, Eq. 34 becomes equivalent to (*i.e.*, it can be rewritten as) the usual Laplacian friction $\partial / \partial x (\mu \partial \mathbf{u} / \partial x) + \partial / \partial y (\mu \partial \mathbf{u} / \partial y)$ and the two formulations only differ around coastlines (see Adcroft and Marshall, 1998). The condition at lateral boundaries is set to free-slip (see Deremble et al., 2011; Ketefian and Jacobson, 2009; Dupont et al., 2003) (see also §E.1). Following Leith (1996) and Fox-Kemper and Menemenlis (2008), the eddy viscosity varies in space and time as:

$$\mu \equiv \text{bvis} + \text{dvis} (\Delta x)^3 \sqrt{|\nabla \zeta|^2 + |\nabla D|^2}, \quad (35)$$

where bvis is a constant background viscosity ($\text{m}^2 \text{s}^{-1}$), the second term on the right-hand side is a time- and space-varying ‘dynamic’ viscosity (also known as a ‘non-linear’ or ‘adaptive’ viscosity), and dvis is a non-dimensional coefficient (≥ 0). The sub-grid scale viscosity of Leith (1996) is similar to the Smagorinsky (1963) viscosity but it is specifically designed for 2–D (eddy) turbulence and gives better results in benchmarks (Graham and Ringler, 2013). Fox-Kemper and Menemenlis (2008) suggested adding the term $|\nabla D|$ to Leith’s viscosity to handle cases where D becomes comparable to ζ . The $|\nabla D|$ part is also the sole contribution to the dynamic viscosity in true 2–D flows of the x - z plane, since $v, \partial u / \partial y$ and ζ are then zero by definition.

The role of the dynamic component (second term on right-hand side of Eq. 35) is to provide high viscosity where/when it is needed and, otherwise, to drop down to a small background value (bvis). The role of bvis is to quash some of the numerical noise before it becomes large enough to be captured by the dynamic component. For most applications $0 \leq \text{dvis} \leq 1$. and $\text{bvis} \rightarrow 0$. Alternatively, one can deactivate the dynamic component by setting dvis to zero and set bvis to a suitably high value (the latter being case-dependent). This approach is useful to replicate existing test-cases with a particular viscosity value or Reynolds number (e.g., Ilıcak et al., 2012).

A.1 Kinetic, potential and mechanical energy

Following Arakawa and Lamb (1981), the rate of change of kinetic energy in a given layer is obtained by multiplying Eq. 27 by \mathbf{u} , invoking $\mathbf{u} \cdot \partial \mathbf{u} / \partial t = \partial (\mathbf{u} \cdot \mathbf{u} / 2) / \partial t$, integrating vertically over the layer’s thickness h , and then using Eq. 30:

$$\frac{\partial}{\partial t} h \frac{\mathbf{u} \cdot \mathbf{u}}{2} = -\nabla \cdot \left(\frac{\mathbf{u} \cdot \mathbf{u}}{2} h \mathbf{u} \right) - h \mathbf{u} \cdot \nabla M + \int \frac{\mathbf{u}}{\rho_0} \cdot \frac{\partial \boldsymbol{\tau}}{\partial z} \, dz - h \mathbf{u} \cdot \nabla \cdot \overline{\mathbf{u}'\mathbf{u}'}, \quad (36)$$

where $-h \mathbf{u} \cdot \nabla M$ is the kinetic-to-potential energy conversion (e.g., Simmons et al., 2004). The rate of change of potential energy is obtained by multiplying Eq. 30 by M :

$$M \frac{\partial}{\partial t} h = -\nabla \cdot M h \mathbf{u} + h \mathbf{u} \cdot \nabla M, \quad (37)$$

and we note that the conversion term appears in both Eqs. 36–37 (with opposite signs). Adding the left hand sides and right hand sides of Eqs. 36–37 leads to the equation for the rate of change of mechanical energy:

$$\frac{\partial}{\partial t} h \frac{\mathbf{u} \cdot \mathbf{u}}{2} + M \frac{\partial}{\partial t} h = -\nabla \cdot \left[h \mathbf{u} \left(\frac{\mathbf{u} \cdot \mathbf{u}}{2} + M \right) \right] + \int \frac{\mathbf{u}}{\rho_0} \cdot \frac{\partial \boldsymbol{\tau}}{\partial z} \, dz - h \mathbf{u} \cdot \nabla \cdot \overline{\mathbf{u}'\mathbf{u}'}, \quad (38)$$

whose left-hand side, in the simplistic case of a single layer and no outcropping, reduces to (e.g., Arakawa and Hsu, 1990):

$$\frac{\partial}{\partial t} \left[h \left(\frac{\mathbf{u} \cdot \mathbf{u}}{2} + \frac{g h}{2} - g H \right) \right] = \frac{\partial}{\partial t} (\text{mechanical energy}), \quad (39)$$

which provides a convenient definition for ‘mechanical energy’. In the more general case of Eq. 38, there is no simple way to define ‘mechanical energy’. The potential energy at time t must then be defined relative to an earlier time t_0 : $\int_{t_0}^t M \partial h / \partial t' \, dt'$. See test-case 17 for an example of the mechanical energy budget.

A few remarks on those relations. (1) Since ρ is homogeneous inside a given layer and constant over time, Eqs. 36–38 can be trivially multiplied by ρ and multiplied by $\Delta x, \Delta y$ to recover the usual units of energy ($\text{kg m}^2 \text{s}^{-2}$, *i.e.*, Joules). (2) We made use of vector and product rule identities that are valid in the continuous case but not always valid in discretized cases. For example, Simmons et al. (2004) discuss some of the pitfalls arising from a barotropic/barocline time-splitting (which isn’t used in beom). (3) When building a volume-integrated budget, the kinetic energy $\mathbf{u} \cdot \mathbf{u}/2$ is typically computed at the center of the grid cell (e.g., Arakawa and Hsu, 1990, their Eq. 3.9) so that it can be directly multiplied by h (also defined at the cell’s center). The divergence term is evaluated in a finite-volume sense with $h \mathbf{u}(\mathbf{u} \cdot \mathbf{u}/2 + M)$ computed at the edges of the cell. The term representing the wind’s rate of work and the bottom friction’s work are computed by evaluating the components of $\boldsymbol{\tau}$ at the location of the velocity points u, v individually. (4) If the algorithms of the model involve “numerical diffusion”, then an extra term (a sink) must be added to Eq. 38. As discussed in §G.3, there are two mild forms of “numerical diffusion” in beom that (overall) should have a minor impact on the energy budget. The magnitude of the numerical sink can be estimated by comparing two calculations (one with and the other without the numerical dissipation), or by computing the residual of the energy budget and attributing it to implicit diffusion. (5) Eqs. 36–38 are the “standard” budget equations to test conservation laws in a numerical code (e.g., Sadourny, 1975; Arakawa and Hsu, 1990; Ketefian and Jacobson, 2009). However, when it comes to interpreting the physical results, the literature proposes many other decompositions such as barotropic/baroclinic components, mean/eddy components, etc. For example, test-case #3 uses a definition of “potential energy” that is less general than Eq. 37 and focuses specifically on the barotropic mode.

A.2 Conservation of vorticity and potential vorticity

Following Arakawa and Lamb (1981), we apply the curl operator on Eq. 27 and invoke $\partial f / \partial t = 0$ to yield:

$$\nabla \times \frac{\partial \mathbf{u}}{\partial t} = \frac{\partial \zeta}{\partial t} = \frac{\partial}{\partial t} (h q) = -\nabla \cdot q h \mathbf{u} + \nabla \times \frac{1}{\rho_0} \frac{\partial \boldsymbol{\tau}}{\partial z} - \nabla \times \nabla \cdot \overline{\mathbf{u}' \mathbf{u}'}, \quad (40)$$

which can be described as a conservation statement for relative vorticity ζ , for absolute vorticity $f + \zeta$, or for the thickness weighted potential vorticity ($h q$). Note that the artificial term of Salmon (2002) (contained in M) does not enter Eq. 40 and thus the conservation statement is unaffected by its presence.

A different conservation statement is obtained if we multiply Eq. 30 by q , combine with Eq. 40, and then divide by h :

$$\frac{\partial q}{\partial t} = -\mathbf{u} \cdot \nabla q + \frac{1}{h} \nabla \times \frac{1}{\rho_0} \frac{\partial \boldsymbol{\tau}}{\partial z} - \frac{1}{h} \nabla \times \nabla \cdot \overline{\mathbf{u}' \mathbf{u}'}. \quad (41)$$

Equation 41 indicates that, in absence of Reynolds stresses, potential vorticity q is materially conserved. For this result to hold, the model requires a layer thickness advection scheme that does not introduce additional diffusion inside Eq. 30. In the case of beom, Eq. 30 has only a weak implicit hyperdiffusion term (§E.1) that leaves Eq. 30 largely unmodified. If a more diffusive thickness advection scheme were used, then an artificial term would need to be added to Eq. 41 to account for the numerical diffusion of h .

Appendix B 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 as `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()

```

B.1 Precision of arithmetic operations (parameter `rw`)

The file `shared_mod.f95` contains a user-adjustable parameter `rw` (“Working precision of Real numbers”) that can be set to either `r4` (single-precision arithmetic) or `r8` (double-precision arithmetic). Most arithmetic operations can be safely conducted in single precision, with the benefit that the memory usage is nearly halved (relative to `r8`) and that the runtime is reduced by a factor that is application-dependent. There are, nevertheless, parts of the code that always necessitate double precision arithmetic, notably the treatment of dates in Matlab’s `datetime` format and most operations that involve the layers’ thickness h . These key components of the code were ‘hardened’ so that they use double precision arithmetic irrespective of whether `rw` is set to `r4` or `r8`. This being said, there can always be situations where the use of `r4` can lead to unexpected issues. You should not hesitate to revert to `r8` if you suspect this is a concern. Note that most Fortran compilers can be requested to track down serious floating point exceptions (e.g., overflows).

B.2 Output files & Restarting a model calculation

The code generates output files in single precision (4 bytes per real number) using a custom binary file format that can be subsequently processed in a variety of languages (Octave, Matlab, Python, Fortran...). The output files inherit the endianness of the computer used for the original calculation (i.e., little endian under Linux) but this behavior can be modified in most Fortran compilers by using specific compilation flags. See the Octave/Matlab script `get_field.m` for an example of how to read output files created in little endian format. (Although the file format NetCDF has established itself as the standard for model outputs in the ocean modeling community, this format was voluntarily avoided so that the code would remain self-sufficient.) Note that only ‘wet’ grid points are included in `beom`’s outputs in an effort to limit their size (see §E). Note also that `beom` does not have a mechanism to average the model fields over time. Therefore, the user is expected to save their model variables frequently enough that the shortest physical timescale of interest is

properly sampled (e.g., hourly in a tide-dominated estuary, daily in a regional basin dominated by mesoscales vortices, weekly in a coarse basin-scale domain, etc).

The default behavior of the code is to associate each prognostic model variable with a given output file (η and `eta_.bin`, u and `u____.bin`, v and `v____.bin`) for the entire duration of a model run. If a power failure occurs before the code reaches the end of its calculation (as in the classic experiment of Edward Lorenz), a ‘restart’ mechanism is available to continue a previous calculation. To trigger this mechanism, one only needs to set the parameter `rsta` to 1. inside file `shared_mod.f95` and to re-compile. `beom` will inspect the file `time.txt`, use the model fields corresponding to the last time entry as its initial condition, and add the new results at the end of the existing files (rather than overwriting the files). In a different scenario, the previous calculation may have ended with non-physical results (e.g., NaN, or unrealistically high velocities indicating an imminent crash). The user can easily restart from an earlier point of the calculation (e.g., 2 or 3 records next to the last) by deleting as many entries as desired inside the file `time.txt`. The code will use the last entry that it finds in `time.txt` as the initial condition of the ‘restart’. Note that entries inside `time.txt` always indicate a complete and successful writing of the model variables. If the writing is partial and incomplete (because, say, the computer hard drive ran out of space while writing the last record), the corresponding time-stamp will not appear in file `time.txt`. Finally, note that during a restart, the `dt_s` value inside `shared_mod.f95` sets the duration of the calculation starting from the point of the restart. For example, if a calculation originally had a value `dt_s=365.` and crashed, and `beom` is restarted from day 35 (without changing `dt_s`), then the calculation will continue until it reaches $35 + 365$ days.

Additional model variables are saved to disk when the parameter `diag` is set to 1. inside file `shared_mod.f95`: q (`pvor.bin`), $g^{-1}M$ (`mont.bin`), μ_{cc} (`v_cc.bin`), and μ_{ll} (`v_ll.bin`).

B.3 Reproducibility of results

Results from `beom` are ‘deterministic’ in the sense that the code produces the same exact results when executed repeatedly under identical conditions⁶. Here, ‘identical conditions’ means “while using the same executable, same computer, same input files, and same input parameters”. This is an important point as numerical codes are normally compiled with high compiler optimization levels (`-O2` or higher), in which case any insignificant change in the code (including in `shared_mod.f95`) has the potential to change when/where round-off errors occur. The use of different computers will also introduce differences in the results, unless absolutely all compiler optimizations are deactivated (leading to an unacceptable penalty in speed). In summary, any departure from ‘identical conditions’ will introduce differences at the level of arithmetic precision.

⁶Note that the results wouldn’t be deterministic if one of the algorithms used random number generation to represent, e.g., a diffusion process.

The impact of these minuscule differences between two calculations conducted in non-identical conditions will ultimately depend on the physics. Calculations where the non-linear terms are absent (as in the test-case of Stommel (1948)), or where the flow field is strongly constrained by boundary conditions, should exhibit negligible differences in all circumstances. At the other end of the spectrum, the worst-case scenario would feature a vigorous, unforced flow (with non-linear terms such as momentum advection activated) and subject to baroclinic and/or barotropic instability. In this worst-case scenario, differences in round-off errors between the two calculations may eventually be sufficient to influence when/where dynamical instability occurs (“butterfly effect”). Once this point is reached, the instantaneous flow fields from the two calculations will become visibly “different”. The two calculations will still have a lot in common: they should yield quasi-identical long-term statistics (mean flow/fluxes, spectrums, eddy kinetic energy), but the instantaneous fields of the two calculations will appear to have diverged.

Note that these numerical considerations are not specific to beom and naturally arise in most numerical codes of geophysical fluid dynamics. For example, the ‘butterfly effect’ is routinely exploited in numerical forecasts of hurricane trajectories (or in climate projections) to obtain different (but equally valid) realizations. The different realizations are used to statistically define the most likely outcome and the range of uncertainty.

Appendix C Isopycnal outcrops and wetting-drying

C.1 Salmon’s 2002 scheme

Isopycnal surfaces become discontinuous (undefined) wherever they intersect 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 small thickness values.

Isopycnal models traditionally rely on monotonic or positive-definite advection schemes (e.g., Smolarkiewicz and Margolin, 1998) to handle such near-zero layer thicknesses. beom differs from these models by following an approach proposed by Salmon (2002). In this scheme, the Montgomery potential (§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 :

$$\left[\sum_{k=1}^N h_k^0(x, y) - H(x, y) \right] - \sum_{k=1}^{i-1} \frac{\rho_i - \rho_k}{\rho_i} h_k^0(x, y) - \frac{h_{\text{sal}}}{3} \left(\frac{h_{\text{sal}}}{h_i^0(x, y)} \right)^3 = C_i, \quad (42)$$

which should be compared to the original equation (Eq. 33). 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. 42 at the location x, y of maximum depth ($H_{\text{max}} \equiv \max [H(x, y)]$). At this specific location, the thicknesses are approximately given by (the actual value will deviate slightly if the artificial term is non-zero at this location):

$$h_{i < N}^0 = H_{\text{max}} \times [\text{topl}(i + 1) - \text{topl}(i)], \quad h_N^0 = H_{\text{max}} \times [1 - \text{topl}(N)], \quad (43)$$

where `topl` (top of layers) is a vector defined by the user (see Fig. 1). Eq. 42 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. 42 is solved iteratively with Newton’s method. Defining \mathbf{h} as the vector containing the values h_i^0 and denoting Eq. 42 as $f(\mathbf{h})$, the thicknesses are iterated as:

$$J(f(\mathbf{h}^l)) (\mathbf{h}^{l+1} - \mathbf{h}^l) = -f(\mathbf{h}^l), \quad (44)$$

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:

$$\nabla \frac{M_i(x, y, t)}{g} = \nabla \left[\left(\sum_{k=1}^N h_k(x, y, t) - H(x, y) \right) - \sum_{k=1}^{i-1} \frac{\rho_i - \rho_k}{\rho_i} h_k(x, y, t) - \frac{h_{\text{sal}}}{3} \left(\frac{h_{\text{sal}}}{h_i(x, y, t)} \right)^3 \right], \quad (45)$$

this particular formulation being suitable for single precision arithmetic (in most cases the three terms in Eq. 45 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’s approach assumes that the dynamical balance (and the relative importance of the artificial term inside Eq. 45) doesn’t change too abruptly from one grid cell to the next. An excessively abrupt transition can occur if the ratio $(h_{\text{sal}}/h_i)^3$ (from the last term in Eq. 45) changes

dramatically from one grid cell to the next. This situation can occur, for example, if the model bathymetry includes poorly resolved (i.e., aliased) bathymetric features. Users of realistic bathymetries would thus be advised to slightly smooth the original bathymetric chart to avoid this situation. As in the testcase on ‘Isopycnal outcropping’, the user can test their bathymetric grid in the model’s state of rest and verify that the velocities remain within, e.g., 10^{-4} m s^{-1} and that they do not grow over time.

Salmon (2002) discusses the advantages of his approach. First, it is considerably cheaper (computationally speaking) than a positive-definite 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 is valuable when calculating bifurcations (Primeau and Newman, 2007) or an adjoint operator. A drawback of Salmon’s approach is that it sets a lower bound on the thickness of the layers (Eqs. 12–14) while even a primitive 1st-order upwind scheme would not. Another drawback is that the choice of h_{sal} is problem-dependent and the user is expected to test the sensitivity of their results to this important parameter.

C.2 Stress and surface/bottom boundary layers

The traditional multi-layer shallow-water equations (§A) assume that the surface/bottom boundary layers are fully contained inside the surface/bottom isopycnal layers (respectively). In other words, the top and bottom isopycnal layers are assumed to be sufficiently thick that the Reynolds stress divergence term, $\rho^{-1} \partial \tau / \partial z$, goes to zero within the isopycnal layer. With this assumption, the term reduces to $\tau / (\rho h)$, which is applied to the surface/bottom isopycnal layers. Outside of those surface/bottom isopycnal layers, the term $\rho^{-1} \partial \tau / \partial z$ is assumed to be zero. Ekman layers and their spiral are not explicitly represented in traditional multi-layer shallow-water models; only their vertically-integrated effect is represented.

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 §C.1). The Reynolds stress $\tau(x, y, z)$ enters the zonal momentum equation as:

$$\frac{\partial \mathbf{u}}{\partial t} = \dots + \frac{1}{\rho} \frac{\partial \tau}{\partial z}, \quad (46)$$

and after discretization onto layer i this becomes:

$$\frac{\partial \mathbf{u}_i}{\partial t} = \dots + \frac{1}{\rho_i h_i} \int_{z_i-h_i}^{z_i} \frac{\partial \tau}{\partial z} dz, \quad (47)$$

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, `hbsbl` for bottom) set by the user in the file `shared_mod.f95`. For the surface stress τ_s this is:

$$\int_{z_i-h_i}^{z_i} \frac{\partial \tau}{\partial z} dz = \begin{cases} 0 & \text{if } z_i < -h_{\text{sbl}}, \\ \tau_s \times \{\max[0, \min(h_i - 1.5 h_{\text{sal}}, z_i + h_{\text{sbl}}) / h_{\text{sbl}}]\} & \text{if } z_i > -h_{\text{sbl}}, \end{cases} \quad (48)$$

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_{bsbl} (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.

Appendix D Temporal discretization

D.1 Forward-Backward versus ‘Generalized’ Forward-Backward

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:

$$(\Delta t)_{\text{max}} = \frac{1}{2} \frac{\Delta x}{\sqrt{g H_{\text{max}}}}. \quad (49)$$

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:

$$\eta^{n+1} = \eta^n + \Delta t \times \text{rhs} [u^n, v^n, \eta^n], \quad (50)$$

$$u^{n+1} = u^n + \Delta t \times \text{rhs} [\eta^{n+1}, v^n], \quad (51)$$

$$v^{n+1} = v^n + \Delta t \times \text{rhs} [\eta^{n+1}, u^{n+1}], \quad (52)$$

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):

$$\eta^{n+1} = \eta^n + \Delta t \times \text{rhs} [(u, v, \eta)^n, (u, v, \eta)^{n-1}, (u, v, \eta)^{n-2}], \quad (53)$$

$$u^{n+1} = u^n + \Delta t \times \text{rhs} [\eta^{n+1}, \eta^n, \eta^{n-1}, \eta^{n-2}, v^n], \quad (54)$$

$$v^{n+1} = v^n + \Delta t \times \text{rhs} [\eta^{n+1}, \eta^n, \eta^{n-1}, \eta^{n-2}, u^{n+1}]. \quad (55)$$

The averaging of the different time-levels increases the accuracy of the scheme but also damps high-frequency oscillations (i.e. close to the cutoff $\Delta t^{-1} \sim \sqrt{g H_{\max}}/\Delta x$) and prevents them from ‘contaminating’ the well-resolved scales. This also means that the scheme is slightly dissipative and that the mechanical energy associated with high-frequency divergent motion will decay in time in absence of forcing (slow divergent motion and rotational motion are largely unaffected). The ‘soliton’ test-case particularly benefits from the ‘generalized’ scheme as this test-case can be executed without any explicit dissipation. The ‘standard’ scheme fails in this particular test-case unless strong dynamic viscosity (`dvis=1.`) is introduced.

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`).

D.2 dt3d and the time-stepping of viscous terms

With its default parameters, the model updates all the physical variables with a single time-step Δt set by its most stringent physical process: the propagation of long surface gravity waves (see Appendices A,D). In many cases, the viscous terms representing lateral viscosity and bottom friction evolve at a much slower pace, and we could ‘skip’ a few barotropic Δt with a negligible impact on the physical solution. This is particularly true in non-divergent flows such as the test-case of barotropic instability.

The model parameter `dt3d` allows us to exploit the slow evolution of the viscous terms and to save some CPU cycles. This parameter (set inside the file `shared_mod.f95`) has units of days and represents the time between each update of the viscous terms. This includes (1) the distribution of the surface/bottom drag across the layers (§C.2 and subroutine `distribute_stress`) and (2) the dynamic viscosity (Eq. 34 and subroutine `update_viscosity`). The default value, `dt3d = 0.`, indicates that the viscous terms should be evaluated at every barotropic timestep.

Appendix E Spatial discretization and indexing of grid points

E.1 Spatial discretization

The model follows standard finite-difference methods by having its variables positioned on a regular ‘C’ grid (e.g., Arakawa and Lamb, 1981) of uniform horizontal resolution $\Delta x = \Delta y$. 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. 22). The ‘C’ grid is well suited for coastal modelling where the first Rossby radius of deformation is well resolved.

The momentum equation (Eq. 27) has two terms that must be discretized in space: the Coriolis-advection term ($q\mathbf{e}_3 \times h\mathbf{u}$) and the Bernoulli potential gradient (∇B). The Bernoulli gradient is discretized as in Arakawa and Lamb (1981) while the Coriolis-advection term follows the so-called EEN scheme of Madec (2016) (note that EEN goes back to Burridge and Haseler, 1977). This combination conserves energy in a general flow and conserves potential enstrophy in a non-divergent flow. EEN represents a compromise between the simple discretization schemes of Sadourny (1975) and the fully conservative discretization schemes of Ketefian and Jacobson (2009) and Arakawa and Lamb (1981). Le Sommer et al. (2009) consider EEN a major improvement over the schemes of Sadourny (1975). Arakawa and Hsu (1990) also point out (in their §5) that EEN is better suited than other discretization schemes for cases involving isopycnals of infinitesimal thickness h (and thus infinite q ; see §C).

The continuity equation (Eq. 30) is discretized as in Arakawa and Lamb (1981) but with a higher-order approximation for the thickness flux at the cell edges. The second-order centered

interpolation of h is replaced with:

$$\begin{aligned} u h &\equiv u_i \frac{h_{i-1} + h_i}{2} - \max(u_i, 0) \frac{h_{i-2} - 2h_{i-1} + h_i}{6} - \min(u_i, 0) \frac{h_{i-1} - 2h_i + h_{i+1}}{6}, \\ &= \max(u_i, 0) \left[\frac{h_{i-1} + h_i}{2} - \frac{h_{i-2} - 2h_{i-1} + h_i}{6} \right] + \min(u_i, 0) \left[\frac{h_{i-1} + h_i}{2} - \frac{h_{i-1} - 2h_i + h_{i+1}}{6} \right], \end{aligned} \quad (56)$$

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. 56 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. 56 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. 56) 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. 56 are forced to zero in regions of outcrop ($h < 5h_{\text{sal}}$) to ensure that the hyper-diffusion does not interfere with Salmon’s method.

E.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. 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 $\text{ndeg} + 1$. In the example of Fig. 22, $\text{ndeg} = 32$ and the zonal velocities u are stored as:

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

All the model variables share the same indexing and dimensions $(\text{ndeg} + 1) \times \text{nlay}$. 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 $2 \times \text{hdry}$ 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`.

E.3 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 volume, energy, vorticity and potential enstrophy. An important conclusion from their study is that most schemes perform poorly in presence of realistic land masses. The 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 chose to implement the FS scheme in *beom*. It is worth pointing out that Rutgers’ ROMS currently has a hard-coded no-slip condition along interior land areas (see `Utility/metrics.F`), i.e. the opposite of FS.

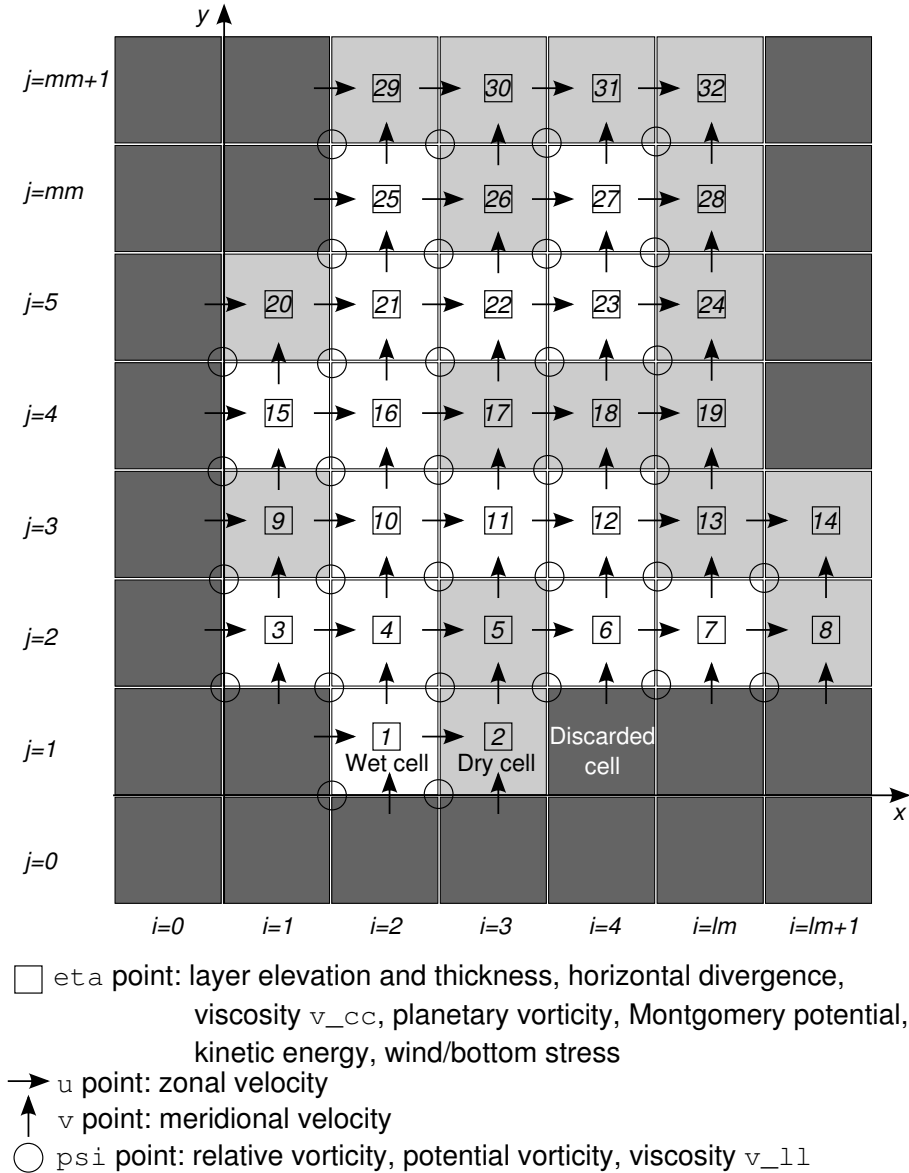


Figure 22: Discretization of variables in horizontal space. White squares are wet cells (ocean) and gray squares are land. Variables from white and light gray cells are indexed into a vector while dark gray cells are discarded. Equations 27,30 are applied at white/light gray cells while fluxes between 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.

Appendix F Scalability

The overhead associated with the unstructured grid 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 under a C shell, one would type:

```
setenv OMP_NUM_THREADS 6
```

and under a bash shell this would become (note the = sign):

```
export OMP_NUM_THREADS=6
```

(type `echo $SHELL` to determine whether your workstation uses a C or a bash shell.)

Figure 23a 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. 23b). The scalability of the code can be compared to a perfectly-parallel calculation defined by:

$$T(N) = T(1)/N, \quad S(N) = T(1)/T(N), \quad (57)$$

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. 23c,d. The model reproduces the expected $1/N$ decrease in run-time (Fig. 23c) but its scalability gradually decreases with increasing N (Fig. 23d). 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 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` ($\sim 10\%$).

Appendix G Limitations of beom

G.1 Performance of the code

The code could be considerably faster if I were willing to make fundamental changes (e.g., abandon the unstructured grid geared toward estuarine domains) or if I were willing to introduce additional approximations. Such major developments are, however, unlikely to happen. I consider the code sufficiently fast when used within its intended niche (see ‘Frequently Asked Questions’) and I would rather focus my attention on exploring new test-cases, improving the documentation, and improving the existing code. This being said, you are more than welcome to download the code and make major modifications to suit your needs. For example, a version of beom with a rigid lid and longer timesteps exists (Zhao et al., 2019).

G.2 Conservation of potential enstrophy

As described in §E.1, the spatial discretization of $q\mathbf{e}_3 \times h\mathbf{u}$ (the so-called EEN scheme of Madec, 2016) does not conserve potential enstrophy fully if the flow field includes divergence. In such a case, the budget of potential enstrophy will include a very small residual that cannot be accounted by physical processes. Discretization schemes that fully conserve potential enstrophy and energy do exist (Ketefian and Jacobson, 2009; Arakawa and Lamb, 1981) but they are considerably more cumbersome than the EEN scheme currently used (see §E.1) and make little sense in a ‘back of envelope model’. Moreover, Arakawa and Hsu (1990) (in their §5) emphasize the advantages of the EEN discretization in cases of isopycnal outcrops or wetting/drying ($h \rightarrow 0$, $q \rightarrow \infty$).

G.3 Momentum advection and numerical noise

The ‘vector invariant form’ of the Coriolis and momentum advection terms, $-q\mathbf{e}_3 \times h\mathbf{u} - \nabla_h \mathbf{u} \cdot \mathbf{u}/2$, is widely used in shallow-water models and is the basis for the most conservative schemes currently available (Ketefian and Jacobson, 2009; Arakawa and Lamb, 1981). The drawback of such schemes is that they generate grid-scale numerical noise that necessitates explicit viscosity ($\mu > 0$, Eq. 34) to remain under control. This explicit viscosity μ is larger than the molecular viscosity of water (by orders of magnitude) and is typically justified as representing the contribution of unresolved (sub-grid scale) flow structures causing a forward energy cascade. Once a non-zero μ is introduced, the numerical scheme is no longer conserving energy nor enstrophy, but the contribution of the explicit viscosity to the model’s solution and its budgets can be back-calculated exactly assuming $\mu(x, y, z, t)$ is known (i.e., if this variable was included as part of the model outputs).

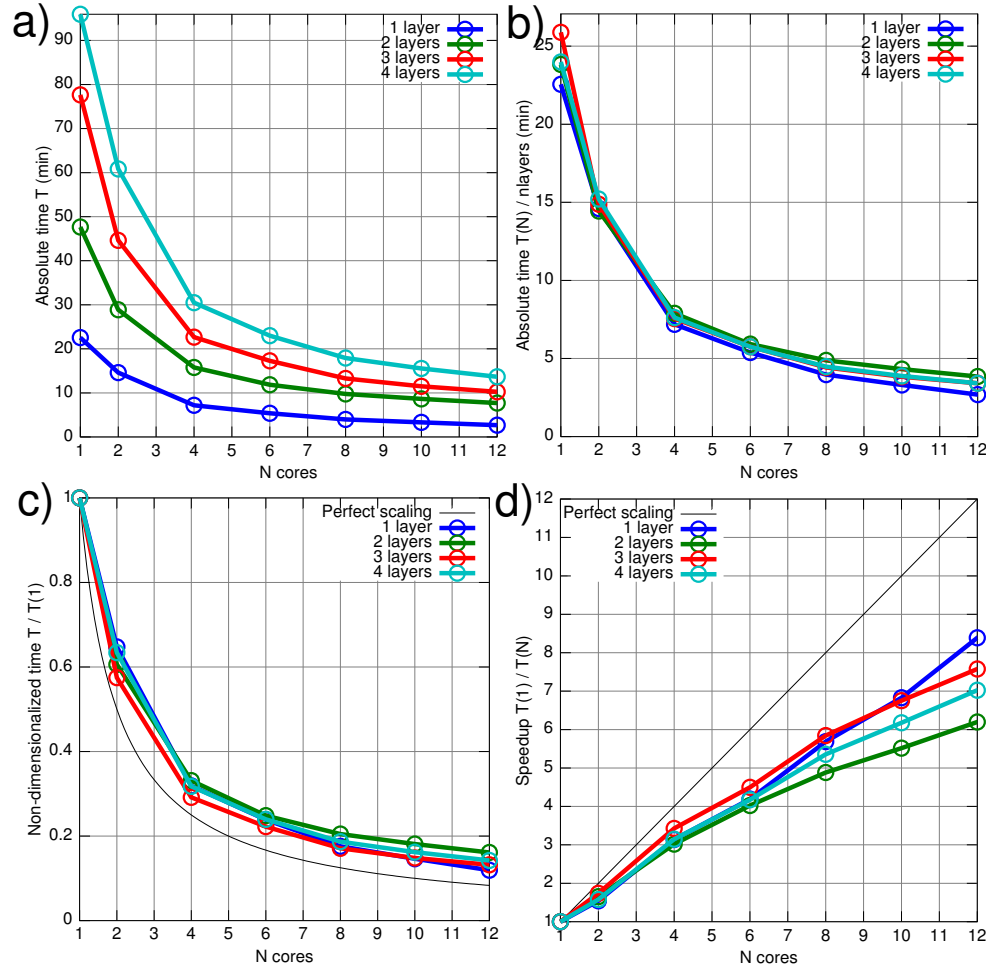


Figure 23: Scalability and performance of the code in a realistic benchmark (see §F). $T(N)$ is the wall-clock time in minutes for a 1-day simulation and N is the number of cores used.

The burden of specifying a suitable explicit viscosity led many modelers to develop numerical schemes with ‘built-in’ (implicit) viscosity and dissipation of energy/enstrophy, either within the spatial discretization of the momentum equation (upwind/upstream bias) or its temporal discretization (e.g., Asselin filter). Examples of this include the flux-form semi-Lagrangian scheme of Lin and Rood (1997) or an upwind-biased scheme in the flux-form momentum equation of ROMS (Shchepetkin and McWilliams, 2005). Such schemes are inherently non-conservative for energy and enstrophy, but this behavior is typically justified in the same way as an explicit viscosity (i.e., sub-grid scale motions and a forward cascade). An advantage of the implicit viscosity is that the user no longer needs to prescribe $\mu(x, y, z, t)$ (the necessary dissipation is built-in the model equations). Additional constraints such as monotonicity can also be enforced within the model code (a valuable feature when representing shock waves). However, quantifying the effect and impact of such a ‘black box’ implicit viscosity on the model’s solution and its budgets is not trivial. This is not a concern in an engineering context, but in a process-oriented study it limits the explanatory power of the numerical model.

There are two forms of ‘implicit’ dissipation in beom. The ‘generalized’ forward-backward scheme uses the suggested weights of Shchepetkin and McWilliams (2005) and is inherently dissipative (§D). This dissipation can be de-activated by reverting to the standard ‘forward-backward’ scheme (`g_fb = 0` in file `shared_mod.f95`). The second form of implicit dissipation is the 3rd-order upstream-bias used in the continuity equation (§E.1). This form of implicit dissipation probably affects the enstrophy budget more than the energy budget and it is hard-coded in beom. To de-activate it, one can comment the extra terms in Eq. 56 and recover a standard 2nd-order centered formulation.

Appendix H Examples of usage

Ken Zhao published two nice studies using a customized version of beom with a rigid lid: Zhao et al. (2019) and Zhao et al. (2021).

The folder `testcases/natl10km/` contains the `param_basin.txt` file (*i.e.* the list of parameters to be entered inside `shared_mod.f95`) and the input files necessary to re-create the 10 km North Atlantic subtropical gyre case portrayed on <http://nordet.net/beom.html>. The actual outputs (≈ 25 gigabytes) are available upon request.

Appendix I Troubleshooting & Questions on general usage

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

First, verify that each and every parameter inside `shared_mod.f95` is entered exactly as it should be (this is the most common error). Also, make sure that the directory containing the input files (variable `idir` inside file `shared_mod.f95`) is *empty* before you execute the test-case m-file. The model will try to use *all* the input files present in this directory (even those that are unrelated to current test-case) and so it is important to ‘clean’ the directory prior to executing one of the m-files. If it still doesn’t work, please email me with a detailed description of the problem and step-by-step instructions for me to reproduce the problem.

2. *I get a Segmentation fault message when trying to use a moderate-to-large grid.*

Operating systems often set a limit on the stack size to prevent potential runaway usage in deeply recursive procedures (which are absent from beom). The default value is typically very low and incompatible with model grids of size `ndeg > 105`, leading to an unexpected `Segmentation fault` without any form of warning. See the documentation of your shell (`tcsh`, `bash`, etc.) on how to read the current settings of your terminal session, and how to change the stack size limit. 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 (32 times the original value).

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
```

3. *My custom application (not a test-case) blows-up. What should I do?*

The ‘restart’ mechanism of beom can be very useful to identify the origin of a crash (and then use this information to devise a solution). In most crashes, one of the prognostic variables will turn NaN at a given grid point at a given timestep. This NaN will then propagate to neighboring grid cells at each timestep (because of horizontal derivatives), giving rise to a steadily-expanding, diamond-shaped zone filled with NaNs. The frequency of the outputs (parameter `dt_o` in file `shared_mod.f95`) is typically too poor to observe this growing outbreak, and thus the outputs will appear to transition abruptly from ‘healthy’ to ‘completely filled with NaNs’, making it nearly impossible to understand what is happening. This is where the restart mechanism can be useful. As described in §B.2, the strategy is to delete a few lines at the end of the output file `time.txt` in order to restart from a point when the

model solution was deemed healthy. However, rather than using the same output frequency as in the original calculation, the restarted calculation would use a smaller `dt_o` value, fine enough to visualize where the NaNs first appear in the model domain and to identify which model variables have suspicious values right before the first NaN appears. Once the user has a reasonable interpretation of what went wrong, they can make additional changes in file `shared_mod.f95` (e.g., tweak the viscosity) and try again. This ‘restart’ procedure can be repeated multiple times to improve the model parameters (with progress being measured by how much farther the code goes before blowing up) and eventually completely prevent such crashes from occurring.

4. *Can I run a calculation for a long time without being logged on all the time?*

Assuming the computer does not have a queuing system, look for the `nohup` command (`man nohup`) which allows the calculation to run in the background without the need to be logged on the system. On a workstation with a C shell, it can be as simple as:
`nohup ./a.out >& log.txt &` (under a bash shell, replace `>&` by `>`).

5. *Can I break up a long calculation of multiple years into sequential calculations?*

You can use the ‘restart’ mechanism of `beom` to do something like that (see §B.2). For example, one can conduct a 8-year long run as four 2-year long runs (`dt_s=730.`). Once the first 2-year segment is completed, you would open `shared_mod.f95` in an editor, change `rsta` from `0.` to `1.`, recompile the code, and then start the calculation. The outputs from the second 2-year segment will be appended to the existing output files from the first 2-year segment, as if it were a continuous run.

6. *The NetCDF4 file format offers data compression. Can we compress beom’s outputs in the same way?*

It is my understanding that NetCDF4 uses the same lossless compression algorithm as `gzip` (LZ77; Ziv and Lempel 1977). Based on one test with a 3.8 gigabytes output file from `beom`, `gzip` reduced the file size by 8%. This is not large enough to motivate the implementation of data compression. I suspect the poor compression rate is a consequence that `beom` only stores non-dry grid points inside its output files.

7. *The scalability drops when I try to use more than x threads on my computer.*

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 computer. Intel and AMD provide additional *virtual* cores to the operating system through technologies like ‘HyperThreading’. For example, my Intel laptop has two physical cores but the operating system allows for up to four threads. Using the additional virtual cores

provides a slight improvement in runtime, but this improvement is well below the scalability we observe on real physical cores. I typically set `OMP_NUM_THREADS` to match the number of physical cores.

8. *Velocities can be unrealistically large inside a region of outcrop.*

The artificial term from Salmon (2002) depends on h_i^{-3} (Eq. 45) and can generate large transient velocities $O(10 \text{ m s}^{-1})$ while accomplishing its role of maintaining the thickness of an outcropped layer to a uniform value $O(h_{sal})$. While these high velocities are typically a fraction of the surface gravity wave speed and therefore do not threaten the computational stability, they are nevertheless artificial in nature and are not a part of the multilayer shallow-water dynamics. When creating visualizations of $|\mathbf{u}_i|$, the artificial velocities can be masked by multiplying the velocity by:

$$\times \max \left[1, \frac{h_i}{10 h_{sal}} \right], \quad (58)$$

which will selectively mask the velocities within the outcropped regions.

9. *I activated all the possible checks in gfortran and I am seeing warnings at compile time.*

There are two types of warnings produced by `gfortran` when compiling with all the possible checks activated. The first one is `Warning: Array reference at...out of bounds in loop beginning at...`. This warning represents the compiler’s best attempt at identifying at compilation time what are possible out-of-bounds operation. What the compiler is missing here is that each of these occurrences are wrapped inside a `if()` conditional that will prevent the out-of-bound operation from occurring at run time. The second type of warning is `Array...at...is larger than limit set by -fmax-stack-var-size, moved from stack to static storage`. This warning represents the compiler identifying variables of constant bounds whose size exceeds an arbitrary limit (65 536 bytes by default). The warning is avoided by setting `fmax-stack-var-size` to a larger value, or by invoking `-frecursive` (which moves all arrays to the stack regardless of their size), or by invoking `-fopenmp` (since `-fopenmp` implicitly includes `-frecursive`).

10. *I activated all the possible checks in my compiler and I am seeing floating point exceptions at run time.*

Not all floating point exceptions are of equal concern. `invalid`, `zero` or `overflow` are worth addressing and their presence can be considered a bug. However, other exceptions like `precision`, `inexact` can arise from normal valid operations and therefore are not cause for concern.

11. *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.

Appendix J Acknowledgments

Individuals who have helped improve beom include: Shasha Cao, Rick Salmon, Andrew L. Stewart, Ken X. Zhao.

Appendix K 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, <https://doi.org/10.1034/j.1600-0870.1998.00007.x>.
- Arakawa, A., Hsu, Y. G., 1990. Energy conserving and potential-enstrophy dissipating schemes for the shallow water equations. *Mon. Weather Rev.* 118, 1960–1969, [https://doi.org/10.1175/1520-0493\(1990\)118<1960:ECAPED>2.0.CO;2](https://doi.org/10.1175/1520-0493(1990)118<1960:ECAPED>2.0.CO;2).
- Arakawa, A., Lamb, V. R., 1981. A potential enstrophy and energy conserving scheme for the shallow water equations. *Mon. Weather Rev.* 109, 18–36, <https://doi.org/10.1175/MWR-D-16-0250.1>.
- 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, <https://doi.org/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, <https://doi.org/10.1002/qj.49711548605>.
- 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, <https://doi.org/10.1006/jcph.1993.1166>.
- 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, <https://doi.org/10.1029/JC095iC03p03273>.
- Bretherton, F. P., 1966. Baroclinic instability and the shortwavelength cut-off in terms of potential vorticity. *Q. J. Royal Met. Soc.* 92, 335–345, <https://doi.org/10.1002/qj.49709239303>.
- Burridge, D. M., Haseler, J., 1977. A model for medium range weather forecasting: Adiabatic formulation. Tech. Rep. 4, European centre for medium range weather forecasts, <https://www.ecmwf.int/node/8495>.
- Carrier, G. F., Greenspan, H. P., 1958. Water waves of finite-amplitude on a sloping beach. *J. Fluid Mech.* 4, 97–109, <https://doi.org/10.1017/S0022112058000331>.
- Chapman, D. C., 1985. Numerical treatment of cross-shelf open boundaries in a barotropic coastal ocean model. *J. Phys. Oceanogr.* 15, 1060–1075, [https://doi.org/10.1175/1520-0485\(1985\)015<1060:NTOCSO>2.0.CO;2](https://doi.org/10.1175/1520-0485(1985)015<1060:NTOCSO>2.0.CO;2).
- Chelton, D. B., deSzoeke, R. A., Schlax, M. G., 1998. Geographical variability of the first baroclinic Rossby radius of deformation. *J. Phys. Oceanogr.* 28, 433–460.
- 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), <https://doi.org/10.1029/2006JC003485>.
- Danilov, S., 2013. Ocean modeling on unstructured meshes. *Ocean Modelling* 73, 195–210, <https://doi.org/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, <https://doi.org/10.1002/qj.49710243210>.
- 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, <https://doi.org/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, <https://doi.org/10.3402/tellusa.v55i3.12094>.
- 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 Eddying Regime*. Vol. 177. AGU Geophysical Monograph Series, pp. 319–338, <https://doi.org/10.1029/177GM19>.
- Graham, J. P., Ringler, T., 2013. A framework for the evaluation of turbulence closures used in mesoscale ocean large-eddy simulations. *Ocean Modelling* 65, 25–39, <https://doi.org/10.1016/j.ocemod.2013.01.004>.
- Haidvogel, D. B., Beckmann, A., 1999. Numerical ocean circulation modeling, 1st Edition. Series on environmental science and management. Imperial College Press, <https://doi.org/10.1142/p097>.
- 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, <https://doi.org/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, [https://doi.org/10.1175/1520-0485\(1996\)026<0913:BDCAIO>2.0.CO;2](https://doi.org/10.1175/1520-0485(1996)026<0913:BDCAIO>2.0.CO;2).
- Hsu, Y. G., Arakawa, A., 1990. Numerical modeling of the atmosphere with an isentropic vertical coordinate. *Monthly Weather Review* 118, 1933–1959, [https://doi.org/10.1175/1520-0493\(1990\)118<1933:NMOTAW>2.0.CO;2](https://doi.org/10.1175/1520-0493(1990)118<1933:NMOTAW>2.0.CO;2).
- Ilicak, 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, <https://doi.org/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, <https://doi.org/10.1016/j.jcp.2010.11.008>.
- 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, <https://doi.org/10.1016/j.ocemod.2007.10.002>.
- Le Sommer, J., Penduff, T., Theetten, S., Madec, G., Barnier, B., 2009. How momentum advection schemes influence current-topography interactions at eddy permitting resolution. *Ocean Modelling* 29, 1–14, <https://doi.org/10.1016/j.ocemod.2008.11.007>.
- Leith, C. E., 1996. Stochastic models of chaotic systems. *Physica D* 98, 481–491, [https://doi.org/10.1016/0167-2789\(96\)00107-8](https://doi.org/10.1016/0167-2789(96)00107-8).
- Lin, S. J., Rood, R. B., 1997. An explicit flux-form semi-Lagrangian shallow-water model on the sphere. *Q.J.R. Meteorol. Soc.* 123, 2477–2498, <https://doi.org/10.1002/qj.49712354416>.
- Madec, G., 2016. Nemo ocean engine. Tech. Rep. 27, Note du Pôle de modélisation de l’Institut Pierre-Simon Laplace, ISSN 1288-1619.
- Marchesiello, P., McWilliams, J. C., Shchepetkin, A., 2001. Open boundary conditions for long-term integration of regional oceanic models. *Ocean Modelling* 3, 1–20, [https://doi.org/10.1016/S1463-5003\(00\)00013-5](https://doi.org/10.1016/S1463-5003(00)00013-5).
- 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, [https://doi.org/10.1175/1520-0485\(1981\)011<0639:IOOTCS>2.0.CO;2](https://doi.org/10.1175/1520-0485(1981)011<0639:IOOTCS>2.0.CO;2).
- Modave, A., Deleersnijder, É., Delhez, É. J., 2010. On the parameters of absorbing layers for shallow water models. *Ocean Dyn.* 60, 65–79, <https://doi.org/10.1007/s10236-009-0243-0>.
- 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, <https://doi.org/10.1175/JPO2899.1>.
- Orsi, A. H., Whitworth, T., Nowlin, W. D., 1995. On the meridional extent and fronts of the Antarctic Circumpolar Current. *Deep-Sea Research I* 42 (5), 641–673.
- 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, <https://doi.org/10.1029/97JC02721>.
- Primeau, F. W., Newman, D., 2007. Bifurcation structure of a wind-driven shallow water model with layer-outcropping. *Ocean Modelling* 16, 250–263, <https://doi.org/10.1016/j.ocemod.2006.10.003>.
- Rao, D. B., 1966. Free gravitational oscillations in rotating rectangular basins. *J. Fluid Mech.* 25, 523–555, <https://doi.org/10.1017/S0022112066000235>.

- Sadourny, R., 1975. The dynamics of finite-difference models of the shallow-water equations. *J. Atmos. Sci.* 32, 680–689, [https://doi.org/10.1175/1520-0469\(1975\)032<0680:TDOFDM>2.0.CO;2](https://doi.org/10.1175/1520-0469(1975)032<0680:TDOFDM>2.0.CO;2).
- Salmon, R., 2002. Numerical solution of the two-layer shallow water equations with bottom topography. *J. Mar. Res.* 60, 605–638, <https://doi.org/10.1357/002224002762324194>.
- Shchepetkin, A. F., 2015. An adaptive, Courant-number-dependent implicit scheme for vertical advection in oceanic modeling. *Ocean Modelling* 91, 38–69, <https://doi.org/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, [https://doi.org/10.1175/1520-0493\(1998\)126<1541:QMASBO>2.0.CO;2](https://doi.org/10.1175/1520-0493(1998)126<1541:QMASBO>2.0.CO;2).
- 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, <https://doi.org/10.1016/j.ocemod.2004.08.002>.
- Simmons, H. L., Hallberg, R. W., Arbic, B. K., 2004. Internal wave generation in a global baroclinic tide model. *Deep Sea Res. II* 51 (25-26), 3043–3068, <https://doi.org/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, [https://doi.org/10.1175/1520-0493\(1963\)091<0099:GCEWTP>2.3.CO;2](https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2).
- Smolarkiewicz, P. K., Margolin, L. G., 1998. MPDATA: A finite-difference solver for geophysical flows. *J. Comp. Phys.* 140, 459–480, <https://doi.org/10.1006/jcph.1998.5901>.
- Stewart, A. L., Thompson, A. F., 2012. Sensitivity of the ocean’s deep overturning circulation to easterly Antarctic winds. *Geophys. Res. Lett.* 39 (L18604), <https://doi.org/10.1029/2012GL053099>.
- Stommel, H., 1948. The westward intensification of wind-driven ocean currents. *Transactions of the American Geophysical Union* 29 (2), 202–206, <https://doi.org/10.1029/TR029i002p00202>.
- Ward, M. L., Hogg, A. M., 2011. Establishment of momentum balance by form stress in a wind-driven channel. *Ocean Modelling* 40, 133–146, <https://doi.org/10.1016/j.ocemod.2011.08.004>.
- 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, [https://doi.org/10.1016/S0967-0645\(00\)00085-0](https://doi.org/10.1016/S0967-0645(00)00085-0).
- Zhao, K. X., Stewart, A. L., McWilliams, J. C., 2019. Sill-influenced exchange flows in ice shelf cavities. *J. Phys. Oceanogr.* 49, 163–191, <https://doi.org/10.1175/jpo-d-18-0076.1>.
- Zhao, K. X., Stewart, A. L., McWilliams, J. C., 2021. Geometric constraints on glacial fjord-shelf exchange. *J. Phys. Oceanogr.* 51, 1223–1246, <https://doi.org/10.1175/JPO-D-20-0091.1>.