

Living with ROMS

Deepak Cherian

Last updated: April 20, 2020

Contents

1	Checklist	2
2	Description	2
3	Misc	2
4	Forcing	3
5	Numerical artifacts / noise	4
6	GLS turbulence	6
7	Calculating budgets	7
8	Output variables to file	7
9	Time stepping	8
10	Diagnostics	10
11	CPP Options	10
12	Useful code files	11
13	Vertical Mode Splitting	12
14	Pressure Gradient Error	13
15	Model Spinup	15
16	Nudging	15
17	Open Boundary Conditions	16
18	Rho0 and R0	19
19	Horizontal Mixing	21

20	Lagrangian Analyses	22
21	Passive Tracers	23
22	NETCDF-4 output	24
23	Performance	24
24	References	25

1 Checklist

- Compile with `-xHost -no_heap_arrays`, set `stacksize` to unlimited with `ulimit -s unlimited`.
- In header file:


```
#define PERFECT_RESTART
#define HDF5
#define DEFLATE
#define DJ_GRADPS
```
- Switch tracer advection scheme from U3 to anything else (too diffusive). Avoid SPLINES.
- If using GLS, set `GLS_Kmin` to $1e-10$ so that simulation reaches background diffusivity values.
- Check that Barotropic Courant number to ≤ 0.85 or so, $r \times 1 \leq 5$ or so.
- Check min, max values for all grid spacing and all fields.
- For forcing, boundary fields make sure that the *second* timestep that ROMS reads in is correct. Otherwise there is an issue with decoding time units.
- Smooth sharp breaks in bathymetry. Check $\max N^2$

2 Description

Shchepetkin and McWilliams (2009)

3 Misc

- Advection schemes have an impact on instability growth rates. See Durski & Allen (2005). Numerical mixing?
- Use `ncgen -b *.cdl` to create a netcdf file based on `trunk/Data/ROMS/CDL/*.cdl`.

- Nonlinear/prsgrd.F describes which header file contains code for each pressure gradient CPP option.
- ANANAME (*) - gives path to analytical header files used (search for ANANAME in analytical.f90). Also present in stdout output.
- Use NINF0=1 to force ROMS to check for bad values of variables at every timestep. This forces it to write a restart record when it blows up.
- If it can't find libraries, you might have to add something to LD_LIBRARY_PATH. The command `ldd oceanM` will tell you what libraries it is linked to.
- In FORTRAN files, `x1(ng)` and `e1(ng)` will give you domain extents. You will need to add a `include mod_grid` statement at the top of the file.
- `Hz = diff(z_w,1,3)` in MATLAB. See ROMS/Nonlinear/set_depth.F Line 239. Hz is the height of the grid cell.
- Apparently, ROMS applies no-slip boundary conditions to masked areas. This is hard-coded in. See Utility/metrics.F and <https://www.myroms.org/forum/viewtopic.php?t=238>
- Interpolation with ROMS grids in MATLAB is apparently tricky. See <https://www.myroms.org/projects/src/ticket/565>.
- ANA_DRAG requires UV_DRAG_GRID. Actually, UV_DRAG_GRID says that you want spatially varying drag coefficients. Then ANA_DRAG says you want it specified analytically and not read from netCDF file.
- Use units of `' 'days since ...'` for all boundary forcing. Hours doesn't seem to work.
- I have found that it's good to ramp up the wind stress over a week or two. Otherwise the initial barotropic adjustment seems to limit time step. Blowups are associated with barotropic waves moving around.

4 Forcing

You can specify surface forcing variables at a single point by providing 1D timeseries. The code just checks number of dimensions so definitely make sure it's 1D. <https://www.myroms.org/forum/viewtopic.php?t=616>

If not using BULK_FLUXES, ROMS needs `shflux` (net surface heat flux), `swflux` (surface freshwater flux), `swrad` (with some CPP options), and wind stresses: `sustr`, and `svstr`. Apparently bulk

fluxes can be unstable with OBCs (!) **arango**: <https://www.myroms.org/forum/viewtopic.php?t=3003>

In some applications with open boundary conditions like T/S radiation, the recommendation is to not use BULK_FLUXES.

The explanation is kind of complicated. The bulk flux formulation in ROMS uses the Monin-Obukhov similarity parameters of Liu et al. (1979) to compute stability functions that compute the turbulent fluxes for wind (W_{star}), heat (T_{star}), and moisture (Q_{star}). There are stable and unstable regimes for these functions. This computation can be highly nonlinear. Any bias or errors in the open boundary conditions for temperature, due to radiation conditions, may result in a loss or gain of heat at the boundary. This may create bogus upwelling/downwelling at the open boundary edges. The model becomes unstable quite rapidly and it blows up.

Therefore, I avoid to use BULK_FLUXES in applications with dynamically active circulation regimes.

5 Numerical artifacts / noise

5.1 Sponge layers

I am seeing noise in places where `visc_factor` changes even if `VISC2 == 0`

5.2 Internal waves

Mostly from Batteen and Han (1981),

1. Accurate dispersion of IGWs required to simulate geostrophic adjustment.
2. Spatial averaging to compute the Coriolis force in C-grids introduces computational noise and distorts the dispersion relation.
3. Two grid point internal waves in the C-scheme, behave like gravity waves in a non-rotating fluid, due to the averaging of the Coriolis force. Because of the non-dispersive characteristic of the gravity waves, these waves do not disperse the energy, resulting in the two grid noise pattern.

5.3 OBC noise

Sponge layers work - need to be careful about reflections. Comparing a periodic run with an OBC run helps identify if the open boundary is really the source of noise.

5.4 2dx noise

For viscosity parameter choices, see section 19.1.

kate: <https://www.myroms.org/forum/viewtopic.php?f=14&t=2208&p=7955>

I believe the vertical mixing schemes work pointwise, so that any 2dx junk will not be smoothed by it. You need to investigate horizontal mixing schemes.

5.5 Mixing schemes and blowups

wilkin <https://www.myroms.org/forum/viewtopic.php?t=4921#p19106>

Generally speaking, I've found k-kl within GLS to be more robust than k-epsilon, with no real difference in the physics solution.

ggerbi: <https://www.myroms.org/forum/viewtopic.php?t=1540>

I have had trouble with k-epsilon being a bit less stable than the other closure schemes. Reducing the time step has helped fix this. Not an entirely satisfactory solution, I know.

arango: <https://www.myroms.org/projects/src/ticket/736>

My experience with the vertical mixing parameterizations is that often they yield high vertical diffusion/viscosity values in trouble spots. Usually, their values are from 3 to 4 orders of magnitude greater than those inferred from ocean observations. I have talked with various observationalists in meetings, and the consensus is that vertical diffusion/viscosity in the ocean rarely exceeds a nominal value of 0.001. I have been worry about this for years, and I mentioned it to Hans Burchard when he visited us several years ago. Apparently, the turbulence algorithm goes unstable. I suspected that this was due to the nonlinear 3D advection of turbulent state variables. Hans mentioned that they rarely activate 3D advection in GOTM (General Ocean Turbulence Model). I tried to turn off the 3D advection in the GLS with CPP

options, but the model blow-up. I didn't spend much time debugging. It is still an item in my extensive priority list.

6 GLS turbulence

- may be a problem with PERFECT_RESTART and GLS <https://www.myroms.org/forum/viewtopic.php?f=19&t=3330&p=13419>

ROMS has a problem with high vertical diffusivities if you use the GLS schemes. Explanation is here (Scully et al., 2011): <https://www.myroms.org/forum/viewtopic.php?f=17&t=2310#p8458>.

The "Galperin correction" limits turbulence length scale to be less than the Ozmidov scale i.e., vertical length scale for turbulence in stratified flow Galperin et al. (1988). This is imposed numerically and is **not** an intrinsic feature of the closure model or the stability functions. It looks like there is some observational evidence for this (Scully et al., 2011).

Briefly, in most cases,

$$AK_t \sim AK_v \sim \frac{0.05 \text{ TKE}_{\min}}{N}$$

Now, the TKE_min parameter is too high (7.6e-6) resulting in high diffusivities depending on stratification. For e.g. with $N^2 = 1e - 5$, we get $AK_t \sim 1.3 \times 10^{-4} \gg AK_{t_BAK} = 1 \times 10^{-6}$ So, we reduce that such that the background values specified are attainable i.e., change GLS_KMIN to 1×10^{-10} in the input file. This value comes from GOTM apparently, but I haven't been able to track that down.

Other points:

- Also, in general, all 2 equation models should behave about the same. By default gls-gen scheme has much lower TKE_min in ROMS and so, it might be different. *The results are more sensitive to stability functions chosen* (Scully et al., 2011).
- Scully et al. (2011) say that field observations compare well against 2 equation models with *non-equilibrium* KC94 stability functions *as derived by Burnchard and Böning*. Now, I don't think that the non-eqbm version is in ROMS, so CANUTO might be best - second best in Scully et al. (2011).
- Choosing between A and B versions of CANUTO is sort-of-arbitrary. The constants are derived by two different methods and are very similar. The first is a unique prediction using some theory and the second involves some derivations (Canuto et al., 2001). Maybe A is better?
- KANTHA_CLAYSON restricts mixing to $Ri < 0.2$ while CANUTO let's it go until $Ri = 1$.

7 Calculating budgets

Caveats:

1. Ω that ROMS outputs is actually $H_z \cdot \Omega$ i.e., it has units of m/s. Really, $\Omega = \frac{D\sigma}{Dt}$ has units of 1/s.
2. $H_z = \frac{\partial z}{\partial \sigma}$ is just `diff(z_r, 1, 3)`. Similarly $\frac{\partial \zeta}{\partial \sigma} = \text{diff}(\zeta, 1, 3)$ for ζ being some variable. I think this is because that $d\sigma$ is everywhere and just drops out. So, I need to be very careful.
3. Continuity budget doesn't work at $t=0$ for some reason.
4. Estimating tendency terms from history / average files may be strongly inaccurate based on frequency of output.

7.1 Continuity

For **history files**, I've managed to make it work with equation:

$$\frac{\partial H_z}{\partial t} + \left(\frac{\partial u H_z}{\partial x} \right)_\sigma + \left(\frac{\partial v H_z}{\partial y} \right)_\sigma + \frac{\partial \Omega H_z}{\partial \sigma} = 0$$

with the caveats listed above.

For **average files**, the following equation should work:

$$\frac{1}{t_2 - t_1} \log \left\{ \frac{H_z(x, y, z, t_2)}{H_z(x, y, z, t_1)} \right\} + \left(\frac{\partial \langle u \rangle}{\partial x} \right)_\sigma + \left(\frac{\partial \langle v \rangle}{\partial y} \right)_\sigma + \frac{\partial \langle \Omega \rangle}{\partial \sigma} = 0$$

8 Output variables to file

Basic info here: https://www.myroms.org/wiki/index.php/Frequently_Asked_Questions#How_to_add_a_new_variable_to_ROMS_output_files.3F

Some notes

1. Time dependent variables are defined in `ROMS/Utility/def_his.F` and written in `ROMS/Utility/wrt_his.F`. Instructions in above URL.
2. Variables that don't depend on time are defined in `ROMS/Utility/def_info.F` and written in `ROMS/Utility/wrt_info.F`

9 Time stepping

9.1 Bottom drag explicit criterion

Need

$$r_D < \frac{\Delta z_{\min}}{\Delta t_{3D}}$$

unless LIMIT_BSTRESS is defined.

9.2 Choosing dt, ndtfast

$$\text{dtfast} = \frac{\text{dt}}{\text{ndtfast}} \quad (1)$$

Selection of ndtfast has implications for the filtering etc. that I don't understand yet.

The fastest wave for the "fast" time step to resolve is the gravity wave: \sqrt{gH} . dtfast is predicatable using the courant number condition so you set ndtfast appropriately.

for the slow time step, it is the internal wave speed (NH/π for constant N).

arango:

the rationale for dt and ndtfast was to force the user to think and have the baroclinic time-steps (dt) divisible by the number of barotropic time steps (dtfast). That is, $\text{dt} = \text{ndtfast} * \text{dtfast}$. We wanted to have a full number. <https://www.myroms.org/forum/viewtopic.php?f=19&t=2512&p=9419#p9419>

We actually recommend having a value for NDTEFAST to be larger than 30. A value of 10 is too small. You need to fine tune DT accordingly to get the appropriate baroclinic time-step. Of course, you need to first set your barotropic time-step according to the fast gravity-wave dynamics which primarily controls the CFL in ROMS. <https://www.myroms.org/forum/viewtopic.php?f=14&t=2068#p7367>

shchepet: <https://www.myroms.org/forum/viewtopic.php?f=14&t=2960&p=11236r#p11236>

What is your barotropic gravity wave Courant number,

$$\text{Cg}_{\max} = \max_{i,j} \{ \text{dtfast} \sqrt{gh(1/dx^2 + 1/dy^2)} \}$$

computed by metrics.F and reported as "Maximum barotropic Courant Number = ???" in ROMS print out?

While for the 3D mode the maximum allowed time step is hard to predict apriori because most likely it is limited by vertical advection and that depends on too many factors (topography, vertical grid stretching, overall flow regime), for the barotropic mode it is plain and simple: it is dominated by Cg_{max} , and the maximum barotropic Courant number should be 0.8 or slightly more: anything less than that is waste of computing power.

m.hadfield

I find that to avoid mysterious crashes in ROMS 3D simulations it is a good idea to:

- Set the short time step so that the maximum barotropic Courant number is 0.9 or so. You can go higher without triggering a crash in the barotropic part of the model, but pushing the limits here seems to reduce the stability of the baroclinic part.
- Calculate a Courant number for the grid based on a baroclinic velocity of 7 m/s and set the time step so this Courant number has a maximum value of 1. The baroclinic velocity will depend on the situation, obviously, but 7 m/s has worked pretty well for me.

If you're getting crashes, find out where they occur. If they're near the boundary pay attention to grid smoothness in this area and to the numerical formulation of the boundary conditions. Otherwise the first thing to try is reduce the long time step.

Time step is generally dependent on the vertical advection and horizontal diffusion.

<https://www.myroms.org/forum/viewtopic.php?f=14&t=2352#p8978>

arango:

The horizontal diffusion tensor is rotated along geopotentials or isopycnals. This in turns puts a time-step limitation due to the density slope approximation of the rotated tensor. The numerical scheme is not implicit so you need to lower the time step accordingly to satisfy the CFL (usually vertical) condition. This not necessarily implies that the system is unconditionally stable! It is just that you need to use a very small time-step due to you stratification and flow regime. Therefore, you need to be more precise when using such terminology.

I have been using rotated (geopotentials and isopycnic) tensors in all my realistic applications for years without problems with both harmonic and biharmonic. I just need to use the appropriate time-step and viscosity with the stress tensor.

By the way, when doing isopycnic rotation we first rotate the tensor along geopotentials so the time-step limitation is even more stringent.

10 Diagnostics

src ticket here: <https://www.myroms.org/projects/src/ticket/486>

File: Nonlinear/diag.F

Max. speed is calculated over *whole* domain, not at location of max. Courant number!

11 CPP Options

- Pressure Gradient schemes: table 1. From Shchepetkin and McWilliams (2001) DJ_GRADPS

CPP	.F file	Details	Paper
DJ_GRADPS	prsgrd32.F	Density Jacobian (Cubic H)	Shchepetkin (2003) - Section 5
PJ_GRADPQ4	prsgrd44.F	Quartic + quartic within grid box	"
PJ_GRADPQ2	prsgrd42.F	Quartic + parabolic within grid box	"
PJ_GRADP	prsgrd40.F	Pressure Jacobian	Lin (1997)
WJ_GRADP + default	prsgrd30.F	Standard and density weighted jacobian	Song (1998)

Table 1: CPP options and corresponding .F file. Lifted from Nonlinear/prsgrd.F

- TS_U3HADVECTION
- SPLINES

arango: <https://www.myroms.org/forum/viewtopic.php?f=17&t=1205#p4215>

It has also been reported in this forum that the SPLINES option may affect the downward flux in the vertical diffusion/viscosity terms. Many users get better and the correct results by not activating this option.

- TS_SVADVECTION

arango: <https://www.myroms.org/forum/viewtopic.php?f=17&t=1205#p4215>

TS_SVADVECTION This option must be only used in shallow, high resolution applications with a lot of vertical levels. It should not be used in coarse, basin scale applications with high stretched vertical grids. I have mentioned this several times in the past. Use TS_C4VADVECTION instead

- TS_A4HADVECTION

- TS_MPDATA: I have found that it definitely helps preserve limits on tracer esp. one with range [0,1].

schepet: <https://www.myroms.org/forum/viewtopic.php?t=2479#p9250>

This sounds like a fairly standard answer in ROMS community (and, in fact, in general oceanographic community), however, it should be noted that contrary to a very popular belief MPDATA is not a monotonicity-preserving, total-variation-diminishing (TVD) scheme.

The only special property MPDATA has is sign preservation, i.e., all-non-negative tracer field remains all non-negative, and vice-versa about all-non-positive.

MPDATA can, in principle, create a new extremum or amplify an existing extremum in the vicinity of a sharp transition. Nevertheless, much of positive experience with MPDATA is due to its relatively low numerical dispersion (similar to that of Lax-Wendroff scheme) combined with some left-over numerical diffusion which makes it very robust.

...Try to avoid sharp front in your initial condition by spreading in across three grid points.

- HSIMT Wu and Zhu (2010)

hwusklec <https://www.myroms.org/forum/viewtopic.php?f=14&t=4313&p=16762&hilit=HSIMT#p16757>

Sigma-like coordinate, including S coordinate, often brings additional challenges for advection schemes. High order schemes such as HSIMT (or others) uses at least three grid points to calculate the cell-interface tracer flux. If there is a very steep topography, say a sea mount, the three neighbouring grid points on one sigma coordinate are actually at very different vertical location, hence very likely they have distinct temperature. Using them to calculate the advection will make a big mess. This is why sometimes HSIMT works not very good around rough topography. Low-order schemes, on the contrary, may work better in this case, since it uses only two grid points, thus the "sigma titling" effect is not that severe. A good practice is to design the stretching coordinate carefully so that the titling S layers (due to topography) are well below the thermocline.

12 Useful code files

Generally in the Build directory - easiest place to find everything that the compiled code uses

- `analytical.f90` : Contains all analytically specified stuff = BCs , ICs, grid etc.

- `mod_scalars.f90` : Defines all scalars! including constants like g , C_p , MY25 parameters etc.
- `metrics.f90` : Calculates Courant numbers...
- `read_floatpar` : reads floats.in file

Variable	Description
<code>xl(ng)</code>	Total length of XI-axis
<code>el(ng)</code>	Total length of ETA-axis
<code>h</code>	Model topography (bottom depth [m] at RHO-points.)
<code>f</code>	Coriolis parameter [1/s].
<code>fomn</code>	Compound term, $f/[p_m \cdot p_n]$ at RHO points.
<code>angler</code>	Angle [radians] between XI-axis and the direction to the EAST at RHO-points.
<code>latr</code>	Latitude (degrees_north) at RHO-points.
<code>lonr</code>	Longitude (degrees_east) at RHO-points.
<code>xp</code>	XI-coordinates [m] at PSI-points.
<code>xr</code>	XI-coordinates (m) at RHO-points.
<code>yp</code>	ETA-coordinates [m] at PSI-points.
<code>yr</code>	ETA-coordinates [m] at RHO-points.
<code>pm</code>	Coordinate transformation metric "m" [1/meters] associated with the differential distances in XI.
<code>pn</code>	Coordinate transformation metric "n" [1/meters] associated with the differential distances in ETA.
<code>om_u</code>	Grid spacing [meters] in the XI -direction at U-points.
<code>om_v</code>	Grid spacing [meters] in the XI -direction at V-points.
<code>on_u</code>	Grid spacing [meters] in the ETA-direction at U-points.
<code>on_v</code>	Grid spacing [meters] in the ETA-direction at V-points.
<code>dmde</code>	ETA-derivative of inverse metric factor "m", $d(1/M)/d(ETA)$.
<code>dndx</code>	XI-derivative of inverse metric factor "n", $d(1/N)/d(XI)$.
<code>pmon_p</code>	Compound term, p_m/p_n at PSI-points.
<code>pmon_r</code>	Compound term, p_m/p_n at RHO-points.
<code>pmon_u</code>	Compound term, p_m/p_n at U-points.
<code>pnom_p</code>	Compound term, p_n/p_m at PSI-points.
<code>pnom_r</code>	Compound term, p_n/p_m at RHO-points.
<code>pnom_v</code>	Compound term, p_n/p_m at V-points.

Table 2: Misc. grid variables

13 Vertical Mode Splitting

From Ezer et al. (2002),

Both, POM and ROMS models, use time splitting schemes, where the two-dimensional, vertically integrated momentum equations (i.e., the “external mode”) are solved using a short time step to resolve fast moving barotropic waves, and the three-dimensional momentum equations (i.e., the “internal mode”), are solved using a longer time step. This splitting technique is common to many free surface ocean models, since it is more efficient than solving the three-dimensional equations with a short time step required by the Courant-Friedrichs-Lewy (CFL) stability condition (Courant et al., 1967). However, the different truncation errors in the two sets of equations require numerical adjustments or filters to make sure that the equations correctly satisfy the continuity equations and conserve tracers quantities.

13.1 EOS error

Using the Boussinesq approximation along with the full EOS is inconsistent and leads to mode splitting errors. Rutgers ROMS still has this problem (03/12/2012).

13.2 Bottom Stress Error

14 Pressure Gradient Error

(Shchepetkin and McWilliams, 2009, Section 5) has overview.

In vertical co-ordinates that don't line up with z surfaces, the horizontal pressure gradient is never evaluated accurately. Shchepetkin (2003) show a way (“a polynomial reconstruction of the ρ field with subsequent analytical contour integration”) of correcting this such that the error in the chain rule is 4th order,

$$\left(\frac{\partial p}{\partial x}\right)_z = \left(\frac{\partial p}{\partial x}\right)_s - \frac{\partial p}{\partial z} \cdot \left(\frac{\partial z}{\partial x}\right)_z$$

Essentially, you interpolate to a “common level” in the horizontal and use that to calculate derivatives. The point is to be careful while doing the interpolation. Two stencils used for this are: “Standard Jacobian” and the “Weighted Jacobian” that differ in the common level chosen. Choosing a common level in between those for the two methods seems to give better results (Shchepetkin, 2003). Quoting them,

“Yet, it has to be emphasized that this choice has no mathematical foundation and is purely empirical in its nature, for example there is no reason to expect that $\gamma = 0.5$ will be universally optimal for any combination of stratification, bottom topography and grid resolution.”

Higher rx_1 generally means larger PGF error.

The error is amplified by the use of a compressible EOS. Again quoting Shchepetkin (2003),

This indicates that the inclusion of compressibility in the straightforward way most likely causes more inaccuracy in the velocity field due to s-coordinate PGF errors than the inaccuracy caused by unphysically neglecting compressibility effects altogether

14.1 Grid Metrics

Beckmann & Haidvogel Number - rx_0

Haney Number - rx_1

Comment by Shchepetkin <https://www.myroms.org/forum/viewtopic.php?f=14&t=612>

Beckmann-Haidvogel rx_0 number is relevant only because it controls Haney number rx_1 (under all other conditions – number of vertical levels and stretching – other kept the same, the two are proportional to each other), and also because you know Haney number only after you setup your vertical grid, while in practice you have to make decisions about topography much earlier in the process of setting up your configuration.

Coarser vertical resolution and stretching more concentrated on top usually mean that you can tolerate steeper topography.

rx_1 up to 3 is considered "safe and conservative"; less than that value usually means that you smooth your topography too much in a realistic application; In practice we often run model having maximum values up to 5 or 6; above that you have to use more caution. Values exceeding 8...10 are considered "insane".

Note that when $rx_1 = 1$ (meaning exactly 1), the hydrostatic error of POM density Jacobian scheme vanishes identically.

Also note that even rx_1 is not the whole story: remember that no or weak and nearly uniform stratification means no or small error regardless of steepness of topography (hence you do not have to smooth too much very shallow areas in marginal seas because they are typically too shallow to be strongly stratified). So the above recommendations are to be considered as the outcome of practical experience, and not mathematically well-defined thresholds.

One can always test it by running a rest stratification test using a representative vertical profile and observe levels of artificial currents. Then decide whether they

are acceptable. Representative means to pick a profile at a single point, and do not average it over horizontal directions: averaging usually results in a much smoother vertical profile which is not representative in sense of errors it generates.

15 Model Spinup

Use CPP option `TS_FIXED`. This turns off evolution of **all tracers**.

From <https://www.myroms.org/forum/viewtopic.php?f=1&t=1165>

arango: Contrairily, a diagnostic run does not implies prediction of the fluid state. There is not a time-evolution due to the model equations; the computations are done based on the present fluid state. Sometimes a diagnostic run is done to equilibrate the fluid initial and boundary conditions to the first order dynamic balances implicit in the continuum governing equations but affected by their discretization (bathymetry, topography, land/sea mask, grid resolution, etc). It can also used to extract a particular (diagnostic) information, or to filter dynamics to slow manifolds by removing unrepresented processes. In ROMS, this can be done by activating `TS_FIXED` to supress the time-evolution of density (modeled in terms of potential temperature and salinity). That is, the initial temperature and salinity is hold constant (initial state) for the entire simulation. So you can initialize the model with just zero free-surface, zero momentum, and a given potential temperature and salinity fields. Then, the model will compute the free-surface and momentum that balances the frozen density state subject to the boundary conditions. This must be a short run (usually, 7 to 14 days)... In spin-up runs we check the kinetic energy of the system. We consider the system fully spin-up when the kinetic energy is no longer growing other than to the seasonal signal.

16 Nudging

Need both `*CLIMATOLOGY` and `*CLM_NUDGING` to actually get nudging to work. Reasons here: <https://www.myroms.org/forum/viewtopic.php?f=19&t=3240&p=12296>

First, don't nudge free surface apparently: <https://www.myroms.org/forum/viewtopic.php?f=14&t=2002&p=7190>

Actually, `ANA_SSH` is an option that I forgot to clean. No manipulations in ROMS free-surface are allowed, except during variational data assimilation. It is incorrect

to do any of manipulation in ROMS free-surface because it is an expression of the vertically-integrated continuity equation. You will get in a lot of trouble if such manipulations are done. You will erroneously alter the transport. Any forcing needs to be done instead in the prognostic vertically-integrated momentum equations (u_{bar} , v_{bar}) for consistency. Recall that the free-surface is an expression for the $div(u_{bar}, v_{bar})$. This also applies for tidal forcing which has been discussed so many times this forum.

This is a fundamental concept that should be clear to all ROMS users.

If you put in RadNud boundary conditions and you have `*CLM_NUDGING`, then the code at the end of `ana_nudgcoef.h` automatically assigns the boundary nudging inverse timescale value to be that in the `nudgcoef` array you set earlier i.e., whatever pattern you define in `ana_nudgcoef.h` is always going to be continuous with boundary nudging timescale. `obcfac(ng)` is multiplied in too i.e., your inflow nudging is going to be stronger than what you set in your code.

Marchesiello et al. (2001) suggest that nudging layers should always have timescales larger than boundary nudging to be consistent with the RadNud option.

ROMS writes nudging as $X_{nudgcoef} * (X_{clm} - X)$

17 Open Boundary Conditions

Marchesiello et al. (2001) has a good writeup on how successive approximations to the Navier Stokes equation diminish its hyperbolic character and make the implementation of OBCs harder.

Free surface nudging is apparently a very bad idea. Always nudge only u_{bar}/v_{bar} . see: <https://www.myroms.org/forum/viewtopic.php?f=17&t=1758>

Nudging is a different beast because it will change volume and mass conservation. If you need such nudging, you need to do it via u_{bar} and v_{bar} for consistency with ROMS equations.

To check OBC noise, plot w field and max out the colorbar at $1e-6$ level or so.

From <https://www.myroms.org/forum/viewtopic.php?f=14&t=2454>

wilkin: "What's not clear from the documentation in `ocean.in` is that certain options are available only to certain state variables.

`Cha` is only for zeta

`Fla`, `Red` are only for u_{bar} and v_{bar}

whereas `Per`, `Rad`, `Gra`, `Cla`, `Clo` are for all

Other traps: Nud can only be used in conjunction with Rad, not on its own. Nud cannot be added to other options, i.e. there is no ClaNud or RedNud etc.”

17.1 VolCons

According to Marchesiello et al. (2001), ROMS calculates net change in volume (due to **normal** velocity through boundary?) and divides it equally as a **normal** barotropic flow among the boundaries with VOLCONS specified. The problem is that in the code, the incoming flux is calculated only at those boundaries where VOLCONS is activated instead of all open boundaries. Hence, you can't limit flux changes to only the boundaries you want. More detail here: <https://www.myroms.org/forum/viewtopic.php?f=14&t=646>. This is not adapted to work with periodic boundary conditions also.

17.2 Radiation Condition

Stuff is advected out as a 1D or 2D advective equation, with speeds determined from model output. Using with sponge layer (Marchesiello et al., 2001) and RADIATION_2D is always recommended. The latter allows information to escape obliquely through the boundary rather than just normal to the boundary. The sponge layer helps suppress computational noise associated with the OBC.

Some comments from <https://www.myroms.org/forum/viewtopic.php?f=14&t=2454>

arango: “I am not a fan of 2D radiation boundary conditions. You get into a lot of trouble with volume conservation because you are dealing with gravity wave phenomena and the open boundary quickly becomes unstable. This problem is so ill posed mathematically and physically. You will get much better solutions with the Flather boundary conditions which use the free-surface gradient. It is like 1D reduced physics boundary conditions (barotropic pressure gradient). You need to get the free-surface at the open boundary from a large scale model or a tidal model. For 3D momentum you clamp data from a large scale model. Open boundary conditions are very tricky and are application dependent”

marchesiello: “concerning 2D radiative conditions, I tend to disagree with Hernan that radiative conditions are a bad choice. All these OBCs are ill-posed anyway in PE models. I always had smoother and better results (for non-tidal forcing cases) using 2D radiative conditions with a volume conservation constraint than with Flather conditions (as illustrated in our 2001 paper). However, the enforcement of volume constraint is tricky in a parallel code and when tides are involved, Flather-type conditions are much more appropriate. We therefore also use a Flather type

(characteristic method) condition as a default, although that can easily be changed when tides are not considered.”

17.3 Mixed Radiation-Nudging

Add a nudging term to the RHS of the radiation BC (above section).

Nudging time scales work as follows (replace TNUDG by ZNUDG etc.)

1. TNUDG is read from input file. TNUDG is then reassigned to $1/\text{TNUDG}/86400$ i.e., inverse time scale in secs
2. $\text{Tobc_out} = \text{TNUDG} = \text{inverse time scale in seconds}$
3. $\text{Tobc_in} = \text{TNUDG} * \text{obcfac} = "$

So if $\text{TNUDG} = 50$ days, $\text{obcfac} = 10$, $\text{Tobc_out} = 1/50$ days & $\text{Tobc_in} = 1/50 * 10$ days = $1/5$ days.

The error that was corrected in <https://www.myroms.org/projects/src/ticket/607> works as

1. $\text{Tobc_in} = \text{obcfac} * \text{TNUDG}$ (inverse time scale) = $1/5$ days.
2. next step is to convert to timestep and so $\tau = dt * \text{Tobc_in}$, but this wasn't done. So, essentially, we end up nudging at $\tau = \text{Tobc_in}$ i.e., smaller by a factor of dt in days. effective $\tau = 1/dt * 1/5$ days = $1/(75*5)$ days for me.

17.4 Shc BC

shchepet: from <https://www.myroms.org/forum/viewtopic.php?f=19&t=3281&p=12491#p12491>

the role of these B.C. condition is to merely provide "radiative cooling" of the solution, or, simply put, avoid trapping of surface gravity waves inside the domain. The easiest way to understand this is to compare this with "clamped" boundaries. If normal velocity is specified at the boundary, then waves will be 100% reflected back into the domain, and the entire solution will have seiche-mode oscillation in free surface which will never go away. In contrast, a radiation condition (not necessarily perfect non-reflection) makes seiche mode go away rather quickly – it cannot sustain itself as long as a significant portion of wave is let out of the domain every time.

Mason et al. (2010)

the majority of ocean models use horizontally-staggered grids, where cells for the free surface and velocity components are not co-located,

In virtually all cases the dilemma of placing u , ζ at different locations has been addressed by applying an ad hoc interpolation scheme to translate each variable to the location of the other; see e.g., Chapman (1985) for a review. This procedure has several drawbacks, such as excessive reflections and the imposition of an additional restriction on time step size relative to the stability limit of the main scheme.

17.5 Reduced Physics

kate: <https://www.myroms.org/forum/viewtopic.php?f=19&t=2159#p7673>

The time to use reduced boundary condition is when you have information about surface elevation but not barotropic currents.

17.6 TKE

From <https://www.myroms.org/forum/viewtopic.php?f=17&t=2411#p8938>

arango: Although the radiation boundary conditions for the TKE fields are implemented, it is not wise to use them. You may get a lot of unexpected behavior at the open boundary. The vertical parametrization is highly nonlinear. Gradient conditions will be adequate even if you have open boundary conditions.

18 Rho0 and R0

From <https://www.myroms.org/forum/viewtopic.php?f=1&t=2436#p9063>

The RHO0 is a very important quantity in ROMS and is due to the Boussinesq approximation: the horizontal density gradients are very small so it can be neglected, except where they appear in the governing equations multiplied by the acceleration due to gravity, g , which is assumed constant. This is used in terms like the pressure gradient, buoyancy, etc. The inertia in the fluid is very small compared to the gravity exerted over the fluid. Therefore, this is a very important approximation constant and have values of the order of $RHO0=1025 \text{ kg/m}^3$.

Contrarily, $R0$ is a less important constant and rarely used except when the linear equation of state is invoked. The linear equation of state is only used in highly

idealized ROMS applications. The seawater is a nonlinear function of temperature, salinity, and pressure. If you derive the linear equation of state using calculus of variations, you will see where ρ_0 comes from. ρ_0 is in terms of density anomalies and has a typical value of $\rho_0 = 25 \text{ kg/m}^3$.

Therefore, ρ_{00} and ρ_0 are very different constants...

From <https://www.myroms.org/forum/viewtopic.php?f=14&t=1305>

shchepet: The difference between the 2003 paper and what you see in the code is explained by the fact that equations in the paper are written assuming that density "rho" is the whole density (meaning that it is about 1030 kg/m^3), while the code is written assuming that density is density perturbation (with at least 1000 subtracted). As the result, in the code there is a need to add the 1000 (or so) back wherever is needed – basically just the barotropic contribution of the total pressure.

Splitting density into bulk part in the code is motivated by the need to avoid roundoff errors: keeping density as "whole" would degrade the accuracy of computation of pressure gradient by at least two (actually close to 3) decimal places, provided that everything else is equivalent. [Note that in the actual pressure gradient routine $\rho(i, j, k)$ (which is anomaly) is multiplied by $z_r(i, j, k)$, which is fairly large in comparison with free-surface perturbation, but $1000 * g / \rho_0$ is multiplied only by free surface "zeta". As the result, cancellation of large terms is done by hand, and does not compromise roundoff errors.

There are also different ways of dealing with it, regarding what value is subtracted from the real ocean density. Rutgers ROMS traditionally subtracts 1000. As the result, fortran variable "rho" can sometimes (at least at surface) be interpreted as "sigma_t" in the oceanographic tradition, and you have appearance of the expression like $G\rho_0 = 1000 * g / \rho_0$. Recall that in a Boussinesq model density appears only as "buoyancy", that is in combination $-g * \rho / \rho_0$, so the combination

$$G\rho_0 + G\rho * \rho(i, j, k)$$

is merely translation from anomaly to full density with multiplication by g / ρ_0 at the same time. UCLA and Agrif ROMS subtract ρ_0 instead of 1000. As a consequence of this choice, the above expression becomes just

$$g + G\rho * \rho(i, j, k)$$

and "rho" stored in fortran array is no longer interpretable as "sigma_t", but in fact it is much smaller, depending on the choice of ρ_0 , which can be selected by the user.

19 Horizontal Mixing

19.1 Setting parameter values

From <https://www.myroms.org/forum/viewtopic.php?f=1&t=5>

dale: Typically, you would want to have the explicit viscosity have a $2dx$ timescale short enough to be important relative to other dynamics. In practice, this would usually mean a timescale shorter than a day. Therefore, you would want to pick a viscosity something like:

$$\text{visc2} \sim (\Delta x / \pi)^2 / \text{timescale}$$

$$\text{visc4} \sim (\Delta x / \pi)^4 / \text{timescale}$$

for harmonic and biharmonic mixing where timescale is a fraction of a day (a quarter to a half perhaps).

This ensures that viscosity is working on $2dx$ features (which are inaccurate in any case) but not too strongly.

You notice that I have used words like "usually" and "typically" quite frequently. That is because choosing these parameters is more artistry than science. Nonetheless, the above mentioned rules of thumb often work OK.

Also see Ilıcak et al. (2012)

The use of a grid Reynolds number ($U\Delta x/\nu$) below 10 is a sensible and universal recommendation to avoid the saturation level of spurious mixing, but this would not guarantee that the remaining spurious mixing would be insignificant.

A major conclusion from this study is the importance of momentum dissipation for the purpose of reducing spurious diapycnal transport to physically negligible levels. This conclusion is based on observing that the ROMS simulations typically yielded the least spurious mixing amongst the non-isopycnal models considered in the idealized test cases. This reduced spurious diapycnal transport is due to the nonlinear momentum advection scheme of Shchepetkin and McWilliams (1998) that locally adapts momentum closure at the grid-scale to reduce grid scale energy in the velocity field. An alternative method was illustrated in the MITGCM and MOM, in which an adaptive lateral viscosity scheme due to Smagorinsky (1963, 1993) was found to drastically reduce spurious diapycnal mixing

19.2 Numerical Mixing

Burchard and Rennau (2008) - diagnosing spurious mixing

Marchesiello et al. (2009) - for TS_U3D_SPLIT improvements to spurious diapycnal mixing. Arango says using TS_A4HADVECTION, TS_A4VADVECTION is just as good in most cases.

19.3 Biharmonic mixing

Griffies say that you should **never** use biharmonic mixing on tracers, especially with more than one active tracer. Something about downgradient flux is **not** always implied.

19.4 Scaling with grid size

This is what ROMS does in Utility/ini_hmixcoef.F

$$\nu_4 = \text{visc4} \times \left(\frac{\Delta x}{\Delta x_{\max}} \right)^6$$

This visc4 is the value in the .in file. In Utility/ini_hmixcoef.F, the visc4 is the square root of the value in .in.

20 Lagrangian Analyses

Rivas & Samelson (2009) for statistical drifter analysis.

Particle tracking options: <https://www.myroms.org/forum/viewtopic.php?t=2767>

Residence time calculations; <https://www.myroms.org/forum/viewtopic.php?f=14&t=2806>

20.1 ROMS Floats

CPP options:

- FLOATS - required to use floats
- FLOATS_VWALK - vertical random walk - The temperature diffusivity AKt is used to get random displacements.
- FLOATS_STICKY - reflect floats that hit surface, stick floats that hit bottom

Files:

- read_floatpar.F - reads floats.in file and sets seeding time, location - process Fdx,Fdy etc.
- def_floats.F - defines output .nc files
- interp_floats.F - interpolates model fields to float position

Float distributions : by default,

$$x_i = fx_0 + (i - 1)F dx \quad \text{and so on...}$$

Fdx, Fdy, Fdz **are used only if** Fdt = 0

In the output file, Xgrid, Ygrid, Zgrid are (x,y,z) trajectories in (i,j,k) space.

There is no PERFECT_RESTART for floats. FRREC in floats.in file needs to be set for ROMS to read in float output. Float output cannot be split into multiple files yet.

Notes on modifying float deployment strategy in read_floatpar.F

1. G, C, T, N in floats.in corresponds to igrd, Fcoor, Ftype, Fcount in read_floatpar.F.
2. What are mc, nc?

20.2 Using adjoint sensitivity

Firstly, the adjoint code cannot be compiled with OpenMP. Need MPI.

Running floats in reverse/adjoint: <https://www.myroms.org/forum/viewtopic.php?f=16&t=2191>
- see list of papers using adjoint method.

Read Moore et al. (2009) for details and advantages of adjoint method.

Also see tutorial at <http://www.o3d.org/ROMS-Tutorial/Adjoint-Tutorial/TutorialAdjointSensitivityTest.pdf>

21 Passive Tracers

Steps:

- #define T_PASSIVE, ANA_SPFLUX, ANA_BPFLUX for surface, bottom boundary conditions.
- Set number of passive tracers (NPT) in .in file.

- Default tracer variable names must be dye_01, dye_02 etc. as defined in varinfo.dat.

wilkin says: <https://www.myroms.org/forum/viewtopic.php?f=17&t=514>

1. Sharp fronts: Initial conditions with sharp spatial gradients are a severe test of the model algorithms, particularly advection. The Gibbs phenomenon (especially in 2nd-order schemes) noted in the posting by ilicakme is a well-known example. This effect is exacerbated by sharp fronts. Keep in mind that you can't expect the model to resolve processes at the grid scale, and certainly not a 1-grid-point wide front. So setting initial conditions that are abruptly discontinuous on the grid scale are almost doomed to failure. In our successful dye-tracing simulations we initialize a smooth Gaussian shaped patch with length scale of at least a few grid spacings - the evolution of the patch is modeled well.

21.1 Late start passive tracer

<https://www.myroms.org/forum/viewtopic.php?f=14&t=2611>

22 NETCDF-4 output

<https://www.myroms.org/projects/src/ticket/248>

Use #define HDF5 instead of #define NETCDF4. <https://www.myroms.org/projects/src/ticket/418>

Env variable LD_LIBRARY_PATH must contain path to nc4 libraries at runtime always - needed when ROMS executable can't find the libraries.

NF90_HDF5 is deprecated as creation mode option. Use NF90_NETCDF4 instead. See http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-f90/NF90_005fCREATE.html

23 Performance

With ifort, compile with `-xHost -no_heap_arrays`

arango: <https://www.myroms.org/projects/src/ticket/769>

We also found that additional improvements (around 8 percent) in ROMS solutions compiled with ifort when the option `-heap_arrays` is removed. However, we need to set the `stacksize` option to a large value in some computers:


```
FFLAGS += -Wl,-stack_size,0x64000000
```

or set the environmental variable `stacksize` to unlimited in the login script. For example, I have the following command in my `.tcshrc`:

```
limit stacksize unlimited
```

If I type the `limit` UNIX command on a Linux cluster, I get:

```
  cputime unlimited
  filesize unlimited
  datasize unlimited
  stacksize unlimited
  coredumpsize 0 kbytes
  memoryuse unlimited
  vmemoryuse unlimited
  descriptors 1024
  memorylocked unlimited
  maxproc 1024
```

ROMS has lots of automatic arrays, so one has the option to allocate those arrays on heap or stack. Usually, the stack option is faster but we need to have enough of it. Otherwise, ROMS will blow-up because memory corruption.

On bash, I have to use `ulimit -s unlimited`.

24 References

- Batteen, M.L., Han, Y.J., 1981. On the computational noise of finite-difference schemes used in ocean models. *Tellus* 33, 387–396. doi:10.1111/j.2153-3490.1981.tb01761.x.
- Burchard, H., Rennau, H., 2008. Comparative quantification of physically and numerically induced mixing in ocean models. *Ocean Model.* 20, 293–311. doi:10.1016/j.ocemod.2007.10.003.
- Canuto, V.M., Howard, A., Cheng, Y., Dubovikov, M.S., 2001. Ocean Turbulence. Part I: One-Point Closure Model—Momentum and Heat Vertical Diffusivities. *J. Phys. Oceanogr.* 31, 14.
- Ezer, T., Arango, H., Shchepetkin, A.F., 2002. Developments in terrain-following ocean models: Intercomparisons of numerical aspects. *Ocean Model.* 4, 249–267.
- Galperin, B., Kantha, L.H., Hassid, S., Rosati, A., 1988. A Quasi-equilibrium Turbulent Energy Model for Geophysical Flows. *J. Atmospheric Sci.* 45, 55–62. doi:10/b6mtdf.

- Ilicak, M., Adcroft, A.J., Griffies, S.M., Hallberg, R.W., 2012. Spurious diapycnal mixing and the role of momentum closure. *Ocean Model.* 45-46, 37–58. doi:10.1016/j.ocemod.2011.10.003.
- Marchesiello, P., Debreu, L., Couvelard, X., 2009. Spurious diapycnal mixing in terrain-following coordinate models: The problem and a solution. *Ocean Model.* 26, 156–169. doi:10.1016/j.ocemod.2008.09.004.
- Marchesiello, P., McWilliams, J.C., Shchepetkin, A., 2001. Open boundary conditions for long-term integration of regional oceanic models. *Ocean Model.* 3, 1–20. doi:10.1016/S1463-5003(00)00013-5.
- Mason, E., Molemaker, J., Shchepetkin, A.F., Colas, F., McWilliams, J.C., Sangrà, P., 2010. Procedures for offline grid nesting in regional ocean models. *Ocean Model.* 35, 1–15. doi:10.1016/j.ocemod.2010.05.007.
- Moore, A.M., Arango, H.G., Di Lorenzo, E., Miller, A.J., Cornuelle, B.D., 2009. An Adjoint Sensitivity Analysis of the Southern California Current Circulation and Ecosystem. *J. Phys. Oceanogr.* 39, 702–720. doi:10.1175/2008JP03740.1.
- Scully, M.E., Geyer, W.R., Trowbridge, J.H., 2011. The Influence of Stratification and Nonlocal Turbulent Production on Estuarine Turbulence: An Assessment of Turbulence Closure with Field Observations. *J. Phys. Oceanogr.* 41, 166–185. doi:10.1175/2010JP04470.1.
- Shchepetkin, A.F., 2003. A method for computing horizontal pressure-gradient force in an oceanic model with a nonaligned vertical coordinate. *J. Geophys. Res.* 108. doi:10.1029/2001JC001047.
- Shchepetkin, A.F., McWilliams, J.C., 2001. A Family of Finite-Volume Methods for Computing Pressure Gradient Force in an Ocean Model with a Topography-Following Vertical Coordinate. Technical Report.
- Shchepetkin, A.F., McWilliams, J.C., 2009. Computational Kernel Algorithms for Fine-Scale, Multiprocess, Longtime Oceanic Simulations, in: *Handbook of Numerical Analysis*. Elsevier. volume 14, pp. 121–183.
- Wu, H., Zhu, J., 2010. Advection scheme with 3rd high-order spatial interpolation at the middle temporal level and its application to saltwater intrusion in the Changjiang Estuary. *Ocean Model.* 33, 33–51. doi:10/cmr5r9.