

I am not writing a full review, but I have just a three comment to the issues which caught my attention.

Lines 60-71, Sec. 2.1 Model grid: "The horizontal resolution of Norkyst is approximately 800 m. ROMS uses C grid staggering, and the domain size is 2747 x 1148 horizontal grid points. About 42% of the grid is land....", also Figure 1:

This is, indeed, an inefficient way to set up a grid.

The goal of this project, as I understand it, is to make an operational model, which both ambitious and computationally challenging, as it involves a large, high-resolution grid, obviously very expensive to run, and even to handle the associated data. And is intended to be used operationally on daily basis.

It happened, that Norway has the most complicated coastline in the World, almost fractal, and going inside these fjords is what is desired, but the grid must be fine enough to even get crude idea about what is going on there. So the authors choose to setup a simple rectangular grid, covering the entire country, and, obviously, facing the need to make a compromise between grid resolution and how far offshore you can go. Cutting Lofoten basin in half also means that you are cutting eddies in half. Lofoten basin is famous for its eddies.

My suggestion: make it curvilinear:

http://91.225.115.25/norkyst/v6.1/roms_grid.pdf

The grid is designed to cover larger area, but in such a way that resolution contracts when approaching Norwegian coast - kind of alternative to 2-way nesting, except that it has no steps and is perfectly smooth.

In this variant the dimensions are 3844 x 802, which slightly less than the total number of points of your grid 2747 x 1148. Note that only one out of several grid lines is shown (it is not practical to plot them all), but the land mask is the actual land mask as it would be seen by ROMS code,

running using this grid.

Its grid resolution is shown here

http://91.225.115.25/norkyst/v6.1/roms_grid_res.pdf

By the design the grid has the same resolution in both directions, locally $pm=pn$ at any point, even though both of them change by more than order of magnitude, depending on the location. 500 means 500 meters; 1k2 means 1.2km.

Topography looks like this:

http://91.225.115.25/norkyst/v6.1/roms_grid_topo.pdf

There are three variants of this grid, which differ only by resolution:

<http://91.225.115.25/norkyst/v6.1/> xi_rho=3844 eta_rho=802

<http://91.225.115.25/norkyst/v6.2/> xi_rho=4611 eta_rho=962

<http://91.225.115.25/norkyst/v6.3/> xi_rho=3076 eta_rho=642

The content of these directories is as follow:

"roms_grid.nc" -- netCDF file for ROMS grid, which can be made runnable. This file already contains all finalized variables defining grid as geometric object (lon_rho, lat_rho, lon_psi, lat_psi, pm, pn, angle, f) and place-holders for land mask and topography (currently "mask_rho" is generated from USGS GSHHS data. Alternatively EMODNET Europe_coastline_2020_OSM.shp can be used. No effort was made thus far to do any manual editing of land mask, so some narrow passages should be inspected and opened as needed), and "hraw" interpolated/averaged from GEBCO_2024.nc.

All other files are precursors, configuration files, diagnostics, orthogonality check, etc...

Many thanks, this is a very generous effort to improve our grid! The forecasting system is already operational, producing daily forecasts since 2012, and we have admittedly been quite conservative when updating our grid for this version 3. Our strategy so far has been to use the nesting options to improve the resolution where needed, and we have also been thinking about using nesting options to reduce the percentage of land points. There are some open questions about forcing data and ability to resolve some aspects of the open ocean dynamics that we would need to look into, but you have certainly given us something to think about for version 4.

Lines 73-76, "The main challenge with regards to stability is not associated with horizontal resolution, but with occasional large vertical velocities in regions of strong convergence (e.g. at the Kattegat-Skagerrak front), which in turn leads to violations of the CFL criterion in the vertical tracer equations, hence the minimum depth of 10 m"

This problem has been solved 10 years ago:

<https://www.sciencedirect.com/science/article/pii/S1463500315000530>

Indeed, once the horizontal resolution becomes finer and finer, at certain point vertical advection becomes the most restrictive factor. Investigation of where and how exactly model blows-up due to computational instability reveals that vertical Courant number is generally very small everywhere, except just in few places, "hot spots" - literally, a handful of grid points out of one million or so holds the simulation hostage. What is going on there is some physical process resulting in something highly non-equilibrium. It may be a front with strong vertical mixing next to no mixing. A supercritical flow (horizontal advection velocity exceeds internal wave speed) becoming subcritical, resulting in violent generation of internal waves (similar to . An internal wave has nowhere to go because of the bottom raise,

resulting in growth of amplitude, and eventual shoaling and crashing. Back-and-forth tidal movements over a topographic ridge causing

generation of internal waves of supercritical (too steep topography) nature.

Whatever.

This is a very accurate problem description for our case. We just realised that your fix is in line for the next development version of ROMS and we are very much looking forward to testing it.

Lines 142-144, <<For the sea surface height and barotropic velocities, we use the "Chapman explicit" and "Shchepetkin" options, respectively...>> -- perhaps the proper way to call it "Riemann boundary conditions", or something of this sort, see Sec. 2.1.2 in

<https://www.sciencedirect.com/science/article/pii/S146350031000082X>

what is relatively new here, is that all Riemann solvers (or numerical methods relying on the idea of propagating something along characteristics) always use non-staggered grids: this makes sense, because Riemann invariants are linear combinations of prognostic variables, and it is natural to make them co-located on the grid. However, ROMS uses staggered grid, and there is no way around it. So this needs to be dealt with, and Sec. 2.1.2 is about this. Another point to make is that one cannot separate b.cs. for free surface and for barotropic velocities: they are part of the same algorithm - the only reason why the discrete value of free surface needs to be "radiated-out" is to end up at the same point in space and time as the normal velocity component, so the two can be added/subtracted to form Riemann invariant. Free surface value needs to travel only half-grid interval in space and one time step up in time, so if traced back to time step "n", the characteristic for free surface may be outside the grid interval (see Fig. 2 from Mason et al 2010). As the result, a naive (say "Chapman explicit") scheme causes more restrictive limitation on barotropic time step than the time stepping algorithm of barotropic mode.

Many thanks for these insights. We find that this treatment of the boundaries gives us much less artifacts compared to using the old "Flather" and "Chapman" conditions. We did have to spend some time tuning in the nudging time scales too, of course, and also have to match the exterior bathymetry in the relaxation zone.