Notes on the creation of a model grid for ROMS

Pierre St-Laurent

September 18, 2023

Corresponding author: `psl@fastmail.net`

1 Introduction

New implementations of the Regional Oceanic Modeling System (ROMS, *Shchepetkin and McWilliams*, 2005) require the creation of a *ROMS grid file* that contains the geometry of the model domain. I wrote down my notes on how I create this file in the particular context of *estuaries that are only partially resolved by the mesh of the model.* The notes assume that the reader consults `myroms.org` and Kate Hedström's manual (*Hedström*, 2018) to read about the ROMS concepts that are not explained here. Note also that there are software packages designed for semi-automatic creation of ROMS grids; the present notes are geared toward individuals preferring to go through the steps manually.

The document is a work in progress—I will be adding more details gradually.

2 Model domain

The geographical extent of the computational mesh is what I'll refer to as the *model domain.* ROMS requires a rectangular domain in the ξ-η space. I have a personal preference toward Cartesian model domains where the horizontal mesh size is the same in both horizontal directions (ξ and η) and across the model domain. In this case, the model domain can be succintly defined with a center (`lon0`, `lat0`), a mesh size ($dx = 1/pm = 1/pn$), a number of grid cells along ξ (`lm + 2`), and a number of cells along η (`mm + 2`). Next, the latitude and longitude corresponding to those grid cells is obtained by defining a map projection. Any conformal projection would do but a few convenient choices are:

- Universal Transverse Mercator (UTM) is a reasonable choice for Cartesian grids and it is readily understood by GIS software if you use the appropriate UTM zone;
- Stereographic has the convenient property that a circle on the sphere remains a circle on the projection;
- Lambert Conformal Conic has the convenient property that a straight line on the projection matches the shortest route on the sphere.

I use the program `proj` (`https://proj.org/`) to translate the x,y positions of the Cartesian model grid into actual longitudes and latitudes. Once they are obtained, the topography of the model domain (ROMS variable h) can be extracted from online public datasets such as USGS CoNED, GEBCO, Etopo... The two-dimensional array h can have positive and negative values (assuming WET_DRY is activated) with the values referenced to a geopotential surface. The vertical datum NAVD88 is one definition of such a geopotential surface. If the topographic datasets are relative to something else (*e.g.* long-term averaged sea level, Mean Lower Low Water...), NOAA provides site-specific conversions between these datums so that a map covering the model domain can be constructed and the topography can be transformed to be relative to NAVD88. Values of h increase downward so that grid cells positioned on land often have negative values.

3 Aliasing

Topographic datasets nearly always feature details at spatial scales finer than the mesh size selected above (dx) which leads to aliasing. One way to avoid aliasing is to filter (discard) the unresolved spatial scales prior to sampling the topographic dataset. However, in the case of interest here (estuaries that are only partially resolved by the mesh of the model), this approach artificially raises the bed of small channels of width $O(dx)$, leading to a model grid that severely underestimates the connectivity across different parts of the estuary.

An alternative approach is to define a sub-grid inside the existing model grid, for example a 3×3 sub-grid (of mesh size $dx/3$) inside the cells of the original grid. The topographic dataset can then be sampled over the fine sub-grid to provide information about the details that exist in real life but are aliased in the coarse dx grid (see Fig. 1). In this figure, the coarse grid would assume values ≈ 0 throughout the area while the sub-grid is likely to capture the thalweg of the channel.

I use the information from the 3×3 sub-grid to mitigate the aliasing of the dx grid. Specifically, I replace the topography of a given $dx \times dx$ grid cell by the x^{th} percentile of the 9 bathymetric values inside that grid cell. Here, x is an arbitrary constant; $x = 50$ is equivalent to selecting the median value of the 9 sub-grid cells, while $x = 100$ is equivalent to selecting the maximum value. From experience, $x = 50$ is insufficient to create much benefit, $x = 100$ exaggerates the width of small channels and makes the model overestimate connectivity across grid cells, and $x \approx 70$ seems to be an adequate compromise. This procedure typically relieves the need for manual dredging in areas where the connectivity is broken (except perhaps in a few odd locations).

4 Mask

ROMS requires the definition of a mask discriminating between cells that are never inundated ($mask = 0$) and cells that could potentially be inundated ($mask = 1$). I generate a first draft of $mask$ by setting all grid cells with $h < h_{min}$ to zero (where h_{min} is an arbitrary threshold representing the minimum acceptable h value for a potentially-inundated grid cell) and one otherwise. Some of the grid cells having $mask = 1$ will correspond to lakes positioned inland with no connection to the estuary. I use a function such as:
http://www.nordet.net/etc/remove_ponds.m
to discard these irrelevant lakes and ponds. This first draft of $mask$ will be further modified after the smoothing of the bathymetry.

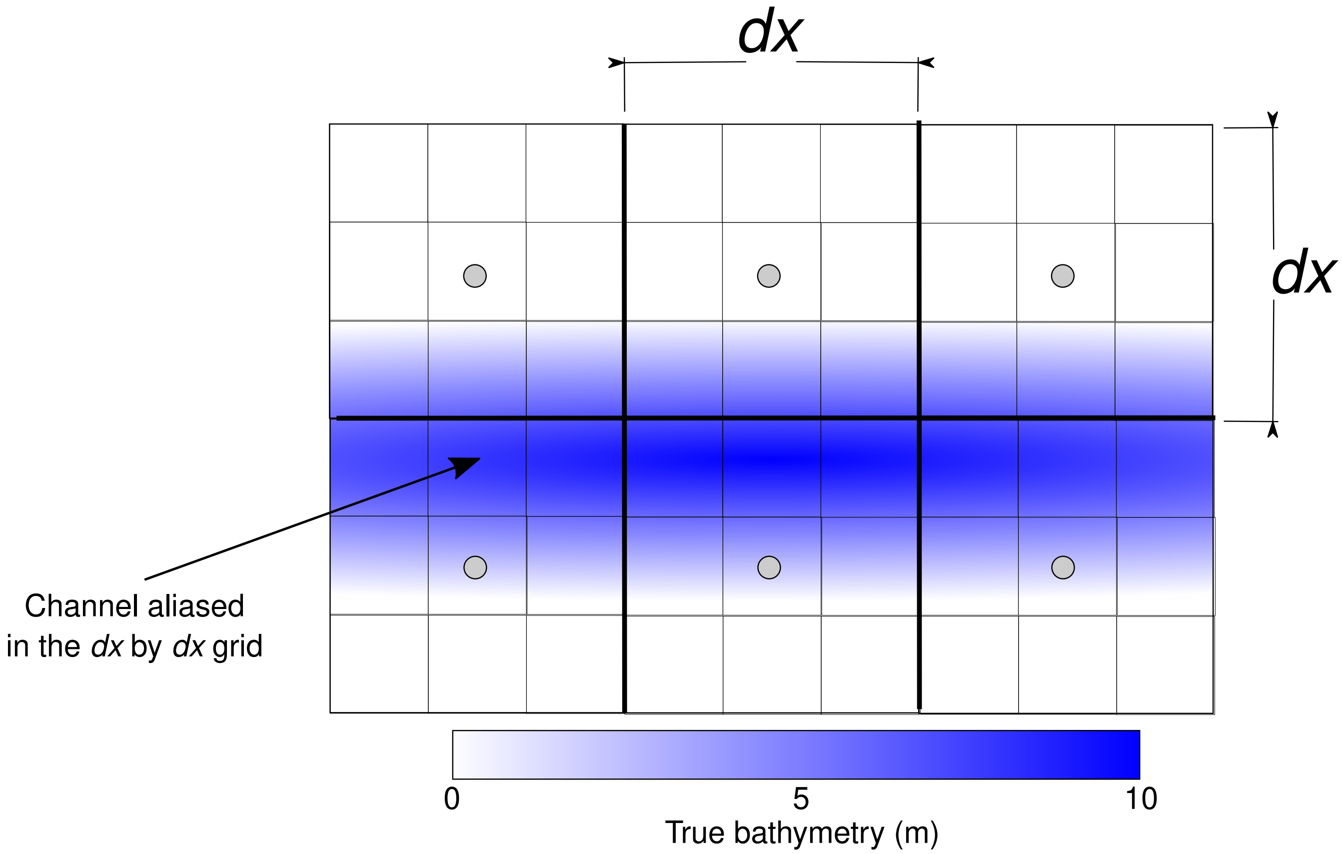

Figure 1: Example of a channel that is aliased in the coarse $dx \times dx$ grid but recovered by the finer 3×3 sub-grid. The gray circles denote the center of the coarse $dx \times dx$ grid cells. Note that the channel has a width comparable to dx and that the aliasing arises primarily because the channel happens to be positioned between the gray circles.

5 Smoothing of h

ROMS has two metrics to identify locations of the grid that are insufficiently smooth: `rx0` and `rx1`. Metric `rx0` is computed from h, while `rx1` is computed from the three-dimensional grid box thicknesses. I typically focus on `rx0` because regions of excessively high `rx1` often coincide with high `rx0`. Following *Beckmann and Haidvogel* (1993):

$$\texttt{rx0} = \frac{\Delta h}{2\overline{h}}, \tag{1}$$

where Δh is the difference in h between two neighboring grid cells and $\overline{h}$ is their arithmetic mean.

My guiding principle is to smooth as little as possible. I apply a fixed number of smoothing iterations over h (typically 1 to 3 iterations) with a simple 5-point Laplacian operator. The flowchart of a given iteration looks like:

1. Compute `rx0` from h, using $\max(h, \texttt{DCRIT})$ while computing the denominator of Eq. 1;
2. Smooth h with the Laplacian operator to obtain h';
3. Compute a two-dimensional array $w(i, j)$ defined by the rules:

(a) Wherever $\mathtt{rx0}(i,j) < \mathtt{rx0}_{min}$, $w = 0$;
$\mathtt{rx0}_{min}$ is an arbitrary threshold below which rx0 is deemed unconcerning;

(b) Wherever $h'(i,j) < h(i,j)$, $w = 0$;

(c) Elsewhere, $w = \min\left(1, \max\left[0, \left(\mathtt{rx0} - \mathtt{rx0}_{min}\right)/\mathtt{rx0}_{min}\right]\right)$.

4. Re-define h as a weighted average of h and h':
$h(i,j) = w(i,j)\, h(i,j)' + (1 - w(i,j))\; h(i,j)$.

The flowchart limits the smoothing to areas where rx0 is deemed "high" (*i.e.* above $\mathtt{rx0}_{min}$), it makes the smoothing proportional to rx0, and it ensures that smoothing only makes the bathymetry deeper (never shallower). The last rule is there to prevent the smoothing from affecting the connectivity between different parts of the estuary.

Once the smoothing is completed, *mask* is set to zero wherever $\mathtt{rx0} > \mathtt{rx0}_{max}$, with $\mathtt{rx0}_{max}$ an arbitrary threshold representing the highest rx0 value deemed "acceptable". Finally, h is capped to the maximum h value found in areas where $mask = 1$ (this step allows for a smaller hc value).

6 Vertical discretization

The ROMS grid file contains only the information necessary for the "barotropic" (vertically-averaged) mode of ROMS. Information about the *vertical discretization* is to be entered inside the input file roms.in, and this information will then be re-copied into all ROMS output files. The vertical discretization is defined by: N, DCRIT (assuming WET_DRY is activated), Vtransform, Vstretching, THETA_S, THETA_B, and TCLINE (listed as hc inside ROMS output files). Note that the constraints surrounding these parameters change depending on the choice of Vtransform and Vstretching (see myroms.org).

7 Testing the grid

The robustness of the grid can be tested by running ROMS in increasingly realistic configurations: (a) closed configuration with no forcings whatsoever, which allows for an evaluation of pressure gradient errors and the magnitude of the resulting spurious currents, (b) add realistic meteorological forcing, (c) add terrestrial ("river") inputs, (d) add open boundaries, (e) add tidal forcing. These tests should be conducted with conservative (*i.e.*, safe) parameters for the model time-step and vertical stretching (optimization of these parameters should come after the grid's resilience has been established).

The location of a crash (excess speed, unrealistic seawater density…) provides a good hint as to which grid cells are problematic, with rx0 being useful to confirm the presence of a grid cell acting as a rotten apple. Troubleshooting these situations would typically involve:

- increasing h_{min}, and/or,
- lowering $\mathtt{rx0}_{max}$, and/or,
- plotting h and rx0 in the vicinity of the crash and manually indentifying which i,j cell is the root of the problem and masking it ($mask(i,j) = 0$).

While going through this process of making the grid resilient, I typically favor masking additional cells over increasing the number of smoothing iterations. The rationale for this preference is that smoothing can generate new problems in a different location, and of course increasing smoothing pushes the model topography farther from reality.

8 Other fields required by ROMS

to be continued

References

Beckmann, A., and D. B. Haidvogel (1993), Numerical simulation of flow around a tall isolated seamount. Part I: Problem formulation and model accuracy, *J. Phys. Oceanogr.*, *23*, 1736–1753.

Hedström, K. S. (2018), Technical manual for a coupled sea-ice/ocean circulation model (Version 5), *Tech. rep.*, U.S. Dept. of the Interior, Bureau of Ocean Energy Management, Alaska OCS Region. OCS Study BOEM 2016-037, https://github.com/kshedstrom/roms_manual/blob/master/roms_manual.pdf.

Shchepetkin, A. F., and J. C. McWilliams (2005), The Regional Oceanic Modeling System (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model, *Ocean Model.*, *9*, 347–404, doi:10.1016/j.ocemod.2004.08.002.