

Simulating salt intrusion with SELFE

Author: Joseph Zhang

Date: August 2012

Salt intrusion represents a challenging test for hydrodynamics models; here are some suggestions on how to apply SELFE to accurately simulate this process.

1. Version

The info in this document is relevant for SELFE v3.2.0 and up.

2. Unstructured grid generation

The transport of salt and heat is heavily influenced by channels, and therefore their accurate representation in the grid is of paramount importance. A cross section of a typical channel can be seen in Fig. 1. The flexibility of unstructured grids allows us to faithfully represent the channel edges in the grid (Fig. 2); this has important implications for tidal prism as well because the total volume is better represented in Fig. 2b. We usually do not smooth bathymetry.

While the definition of edges is usually self-evident in main channels (e.g., examine the highest gradient zone), it becomes obscure in shallower water areas; e.g. should the edge be located at 5m or 3m in upstream rivers? This depends on the system; e.g., if you wish to accurately simulate the sharp gradients in shallow water (an example would be many Texas coastal rivers), you then need to carefully follow edges at a depth that you believe still can stratify significantly. Otherwise you do not have to adhere to the channel edges in these regions. In deeper ocean representation of channels is not necessary.

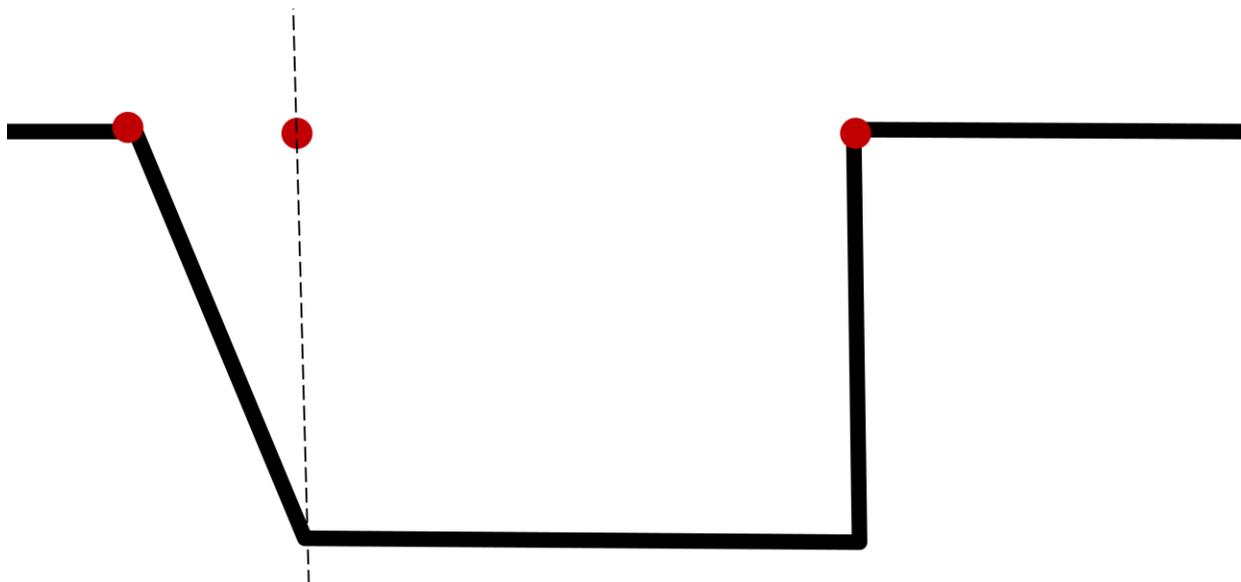


Figure 1: Side view of a typical channel. The 3 red dots represent the channel edges and should be followed faithfully in generating the unstructured grid (e.g. in SMS the arcs must go thru these dots).

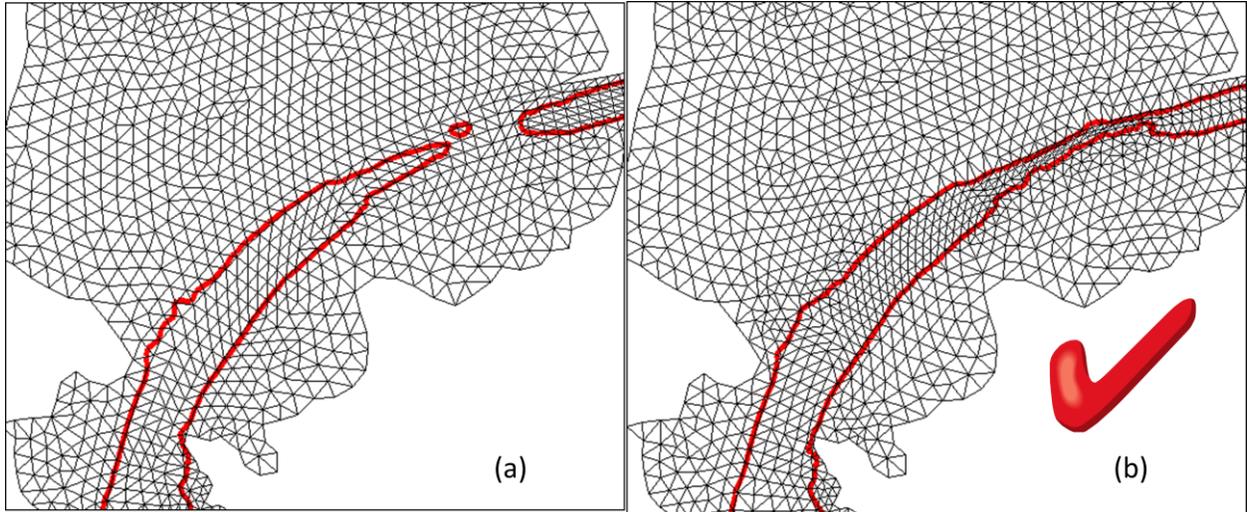


Figure 2: Two different representations of a channel; the red line is 10m isobaths representing the edges of a channel. The channel edges are faithfully followed in (b) but not in (a).

Most channels in nature are “anisotropic” i.e. with disparate cross- and along-channel length scales. Therefore it’s often useful to represent channels as pseudo structured grids; e.g., in SMS you want to use coons patches for channels and then split all quads into triangles. The grids in Fig. 2 were generated this way. You’ll have to adjust the cross- and along-channel resolution based on your understanding of the system (more in Section 3). The choice of cross-channel resolution is particularly important.

3. Selection of parameters

3.1 Transport scheme

In strongly forced systems with compressed salt wedges (e.g. Columbia River) the higher-order ELM transport (possibly together with higher-order tracking $n_{adv}=2$) usually performs well. In other systems the 2nd-order upwind (TVD) scheme is often useful.

TVD, being an explicit scheme, is not efficient when very high resolution is used. The stability (and max. principle) is only guaranteed by the Courant condition:

$$\Delta t_b = \min_i \frac{V_i}{\sum_{j \in S^-} |Q_j| (1 - \frac{\phi_j}{2} + \delta_j)} \quad (1)$$

where V_i is the volume of a prism, S is all inflow faces, Q_j is the fluxes thru the faces, φ_j is the limiter (function of the upwind ratio), and δ_j is related to the limiter. In SELFE the time step for transport eqs. is different from the main step (Δt), and sub-cycling is automatically used for transport eq. From Eq. (1) the transport time step is not only dependent on the flow but also on the local tracer gradients (i.e. it's adaptive in nature).

An efficient hybrid upwind/TVD scheme is available in new versions of SELFE, which allows you to specify regions where more efficient upwind scheme is used; e.g. when depth is shallower than a specified threshold (h_tvd); additional control can be specified in `tvd.gr3`. In grid generation, gridding along a contour just above the high-order TVD threshold (h_tvd) allows more control of the TVD region and in particular creates a smoother boundary between the region where it is used and where it is not.

So a workflow for adjusting the grid is:

- a) Generate a 1st grid with reasonable resolution and test-run it with TVD and an appropriate h_tvd (e.g. 6m); use average flow condition (it is expected that high-flow condition will lead to less efficiency);
- b) *After a dynamic equilibrium is established for some time*, plot out `fort.17` (format: `time_step, nitr`, where `nitr` is <# of sub-cycles in transport>) to see if on average the # of sub-cycles is reasonable ($\Delta t_b = \Delta t / nitr$). Resist the temptation to do this too soon, as initial re-organization of the salt often leads to small time steps but sometimes the steps become larger later;
- c) If sub-step is too small in general, use `TVD_analyzer.pl` (run in same dir as `hgrid.gr3`) to find "culprit" elements; the outputs from this script is `TVD.prop` (format <element #> <flag of 1 or 0>) which can be loaded into `xmgredit5` for visualization (Special→Properties→Read, and make sure the toggle is on 'Property at elements', and then read in `.prop` file; a dialogue box will appear. Define isolines first by pressing the button, and then depress 'Display current' to show regions with `depth=1`). Specify a tolerance time step (`dtb0`) for this script. Remember: this script will write out cumulative max. up to the most current time step, and so start with a small `dtb0` (e.g. 2 sec) and you'll likely to see only a few violating elements. Re-generate the grid to coarsen these elements to relieve the bottleneck.

With some experience and understanding of your system you may only need a few iterations to get a good grid.

If you are frustrated you may be attempted to use `tvd.gr3` (or even `h_tvd`) to bypass those violating elements. While this is a convenient fix and sometimes can be useful in the process, remember that reverting back to upwind does introduce numerical diffusion in these elements. In general `h_tvd` should be fixed first (based on the typical stratification), and `tvd.gr3` should be

used only to specify regions where you are absolutely not interested in stratification (e.g. freshwater region).

3.2 Other parameters

The design of vertical grid is also very important. In general resolution near the surface and bottom needs to be sufficient. However, in some systems you may want to resolve the surface only ($\theta_b=0$) as surface gradients there are more challenging. Use adequate # of levels to resolve sharp gradients.

Test different turbulence closure schemes including the GOTM options. Lowering `diffmax.gr3` (e.g. $1.e-3 \text{ m}^2/\text{s}$) may also help in some upstream regions; this is effectively an admission that there are errors in the turbulence models (note that unlike other models, SELFIE does not smooth viscosity/diffusivity outputs from closure models). However, this is usually not necessary if your grid resolution is adequate.

Boundary conditions are also relevant in this exercise. Since TVD has stricter time step limits, it's desirable to avoid large velocity along the ocean boundary (often due to non-specification of vel. b.c. there for inflow). A very useful trick to avoid this is to run 2D barotropic SELFIE on the same (or larger grid) and use the vel. info as b.c. (see one way nesting scripts in utility/).