Towards an adaptive vertical coordinate in MOM6

Angus Gibson

7 May 2018

Background

- Hybrid coordinates are currently in use in MOM6 (HyCOM-like)
- Require a prescriptive and sensitive configuration:
  - Expected densities and a nominal depth for all layers
  - Depths chosen conservatively to avoid problems with surface boundary layer
- Biggest shortcomings: overflows and exchanges

Regridding/remapping

- Specify an arbitrary grid $z_k(x, y)$ for the entire domain with a *regridding function*
- Given the previous grid $z_k^n(x, y)$ and the model state (T, S, . . .), generate a new grid z_k^{n+1}
- State is conservatively *remapped* from the source grid to the destination grid
- Very generalised operation
  - This does not even require that the grids have the same number of levels!

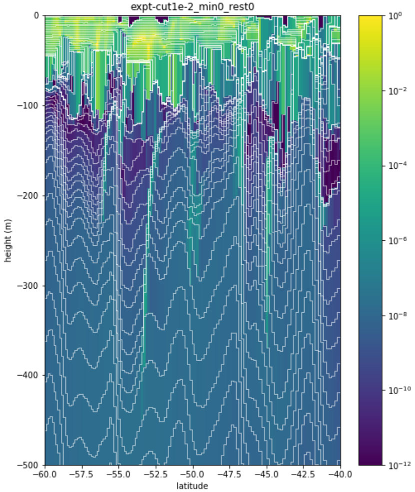
Equation

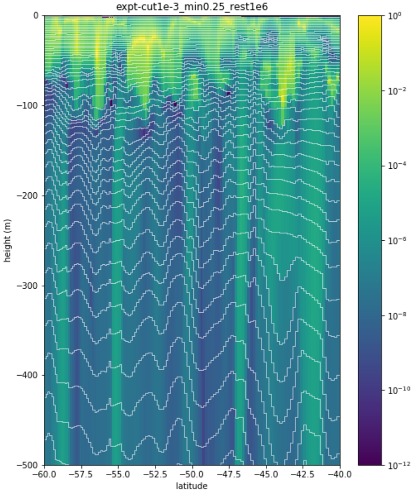
- We want to represent our grid by a single equation that can be understood mathematically and intuitively

$$\partial_t z_k = -\vec{\nabla} \cdot \left(\omega_\sigma \frac{\kappa \nabla \sigma}{\sigma_z} + \omega_z \kappa \nabla z_k \right) + \tau^{-1} \left(\langle z_k \rangle - z_k \right) + ???$$

Examples

- Iterate regridding/remapping on a z-space snapshot





Density adaptivity

$$\partial_t z_k = -\vec{\nabla} \cdot \left(\omega_\sigma \frac{\kappa \nabla \sigma}{\sigma_z} + \omega_z \kappa \nabla z_k \right) + \tau^{-1} \left(\langle z_k \rangle - z_k \right)$$

- Using a stencil surrounding the current gridpoint (i.e. the grid communicates laterally)
- Conserves mean layer height (what goes up in one cell must come down in another)
- Strongly limited (think of a see-saw on a hill)

Horizontal smoothing

$$\partial_t z_k = -\vec{\nabla} \cdot \left(\omega_\sigma \frac{\kappa \nabla \sigma}{\sigma_z} + \omega_z \kappa \nabla z_k \right) + \tau^{-1} \left(\langle z_k \rangle - z_k \right)$$

- *Smooth* layers horizontally (not always flat!) and conservatively
- Split into strongly limited and weakly limited contributions
  - If ω_z is constant, only limited by top and bottom surfaces
- Use an isopycnal slope to decide regions of smoothing or adaptivity
  - ω_σ vs. ω_k

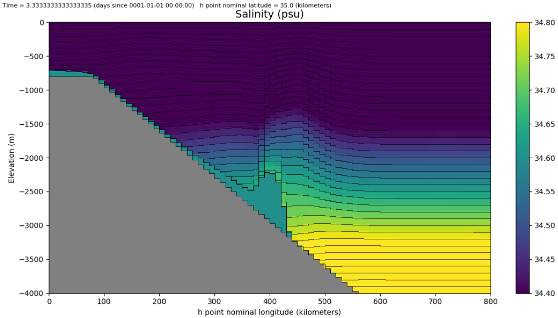
Restoring

$$\partial_t z_k = -\vec{\nabla} \cdot \left(\omega_\sigma \frac{\kappa \nabla \sigma}{\sigma_z} + \omega_z \kappa \nabla z_k \right) + \tau^{-1} \left(\langle z_k \rangle - z_k \right)$$

- Source term – doesn't have to conserve mean layer height
- Tried to design this in such a way to ensure resolution doesn't drift from where it was initially
- Turns out this is approximately equivalent to computing APE – much too expensive
- Would more realistically be used to gently restore toward a target grid, but maintain adaptivity perturbations

Another adjustment term?

- The previous terms miss one key process:
  - Layers with different k-index (i.e. not laterally connected) but the same density
- e.g. flow downslope – how do we know which layer we should be in?



What's next

- Models are Murphy's Law machines: the overflow problem will show up at some point
- We can test the coordinate in an idealised configuration and/or 1° global