Geoscientific
Model Development
Discussions

Interactive comment on "Simple algorithms to compute meridional overturning and barotropic streamfunction on unstructured meshes" *by* Dmitry Sidorenko et al.

Mark R. Petersen

mpetersen@lanl.gov

Received and published: 11 May 2020

Please see attached pdf for a better formatted version of these comments.

This paper provides a clear explanation on how to compute the meridional overturning and barotropic streamfunction on unstructured meshes. It is well-written and clearly communicates the motivation, the mesh structure, and mathematics of the computation, and the sensitivity of the parameters. I have not seen these details discussed in other publications, so it will be a useful reference to ocean modelers reproducing this calculation.

In MPAS-Ocean, we compute the meridional overturning circulation (MOC) streamfunction by integrating the vertical velocity in latitude bins, level-wise, as described in this paper as Method A. I adapted this method from our previous model, POP, which is a structured quadrilateral mesh but usually ran on a tri-pole grid, so the generality of latitude binning was required for that model as well. We have the ability to run the MOC calculation as an in-situ calculation during the simulation: https://github.com/MPAS-Dev/MPAS-Model/blob/master/src/core_ocean/analysis_members/mpas_ocn_moc_streamfunction.F#L! or as a post-processing calculation from the time-averaged velocity fields: https://github.com/MPAS-Dev/MPAS-Analysis/blob/develop/mpas_analysis/ocean/streamfunction_moc.py#L436 We describe the philosophy of the in-situ computation here: Woodring, J., Petersen, M., Schmeisser, A., Patchett, J., Ahrens, J., Hagen, H., In Situ Eddy Analysis in a High-Resolution Ocean Climate Model, IEEE Transactions on Visualization and Computer Graphics, vol.22, no.1, pp.857-866, Jan. 31 2016 http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7192723 If you feel it is relevant, you are free to add any of that background to your paper.

We have found that the python post-processing method is fine at lower resolutions, but at high resolutions ($\sim$ 1 million horizontal cells) it is extremely slow or fails, and we rely on our in-situ calculation. It would be good to hear in section 5 about any performance limitations in your calculation. On page 9 line 20, are these python routines just the plotting part, or do they reproduce the functionality of the Fortran and c code? If the later, did you run into performance bottlenecks, and how did you run python at high resolution – some combination of xarray and dask?

Your equations are missing numbers.

I find the notation of the streamfunction definition confusing (page 4, top). I prefer to see the $d\theta$ and dz at the very end, to indicate what is included in the integral. The first one, in particular, could be confused with the multiplication of two integrations,

rather than one case of a double integral. Another way to clarify is to write w and v as: w(x,\theta',z) and v(x,\theta,z') and integrate in \theta' on the first equation and z' on the second, to explicitly state that the primes are dummy integration variables, not to be confused with the upper bound and independent variable on the left side.

For the new reader, it would be useful to review the definition of a streamfunction at the beginning of section 3. That is, if we have zonally-integrated velocities W and V, then for an incompressible fluid we can define a streamfunction \Psi as d\Psi/d\theta = W (1) d\Psi/dz = -V (2) and refer to some fluid dynamics textbook – my favorite is Kundu, https://www.amazon.com/Fluid-Mechanics-Pijush-K-Kundu/dp/0123821002 (I forget which line is negative, I always have to work through it. I assume one of your streamfunctions has a negative sign absorbed in a coordinate or variable definition, but that would be good to note as well).

From my equations above, it is clear that one may compute \Psi by either integrating (1) in \theta, or integrating (2) in z. This is identical analytically, but leads to different implementations and numerical results, as you describe.

Please state the boundary conditions for the two equations, top of page 4. For Method A and a global MOC it is easy – just set \psi to zero at the Southern boundary. But for regional MOCs, like the Atlantic, we use the second equation with v along a connected set of edges near \theta_0 (33S, say) and integrate from the bottom up. That is our southern boundary condition \Psi(z,theta_0) to then integrate northward with the first equation. For Method B, the stream function is just zero at the top or bottom.

Although it is implied, it would be useful to state in each figure caption whether you are using Method A or B.

Grammatical comments Please change MPAS-o to MPAS-Ocean throughout. p.1: original conversation, which recent years, considerable double)): 2019) and 2009) p.2 easy-to-implement of the dual mesh differ from that p.3 faces of the scalar control of the edge in the model, however care same way as in the model if fluxes are to be

properly analysed. and, we would guess, generally known. p. 4 In these definitions southern latitude, in radians, because the full velocity The computational procedure is straightforward p.5 in more general cases. p.6 Using the coarsest bin p.10 The general idea of the simple We emphasize that the algorithms described are still sensitive to parameter choices, for in each case

Please also note the supplement to this comment: https://www.geosci-model-dev-discuss.net/gmd-2019-336/gmd-2019-336-SC1-supplement.pdf

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-336, 2020.