GMD-2023-45 Response to Reviewers

We appreciate the time and efforts the two reviewers dedicated in the second round of review to providing feedback on our manuscript "*Intercomparisons of five ocean particle tracking software packages in the Regional Ocean Modeling System*". We have incorporated or responded to all the comments and suggestions made by the reviewers. Please see below, in blue, for a point-by-point response to the reviewers' comments and concerns. The line numbers refer to the tracked version of our manuscript.

Reviewer 1:

Review of the revised version of "Intercomparisons of five ocean particle tracking software packages" by Xiong and MacCready.

I thank the authors for their extensive and detailed replies to my queries and comments. While many of the issues that I raised have been solved in the revised version of the manuscript, I am left with the most important one (original comment 1): what does this new Lagrangian code add? Or, in other words, what niche does it fill? The authors do not sufficiently answer that, in my opinion. In the reply, they discuss comparison to other codes, but in a diverse ecosystem of Lagrangian codes, each one has its niche (at least when it was first developed). I don't see that niche for Tracker: what it can do that other codes can't.

I leave it to the editor whether this manuscript, despite my reservation, still meets the bar for Geoscientific Model Development; or whether it would be better suited for a (Diamond Open Access) journal like the Journal of Open Source Software.

Response: we thank both reviewers' comments on the niche of Tracker that inspire us to clarify the uniqueness of Tracker and emphasize the major findings of our study. We think that the most unique thing about Tracker, compared to other packages compatible with ROMS modeling system, lies in its much faster execution time, a feature attributed to the efficiency of the nearest neighbor searching algorithm. This performance enhancement is especially pronounced in the large model domain we tested (please see Figure 9).

Within a forecast system such as LiveOcean, a reasonable computational burden is important when the offline particle tracking based on forecasted hydrodynamics can be finished in a timely manner. Even though Particulator (one of the tracking packages that were tested in this study and is written in MATLAB) can run fast with millions of particles, the commercial software MATLAB is less accessible than the open-source Python. Therefore, besides saving computation time, platform independence is another unique thing about Tracker (Table 1, under the category of 'Ease of use').

In addition, the most important finding in this study is that although all tested particle tracking packages have different choices in e.g., interpolation schemes, advection schemes, boundary conditions, or compatibilities with different numerical models or forcing data sources that make them unique, they do not end up with very different results.

In the revised (tracked) manuscript, we added these statements in the sections of abstract (lines 13-14, 18), section 3.4 (lines 313-325), and conclusions (lines 351-352).

Reviewer 2:

In general, I find the paper to be improved and with many added details. I think the paper should be of interest to the community, and I recommend publication after some minor revisions.

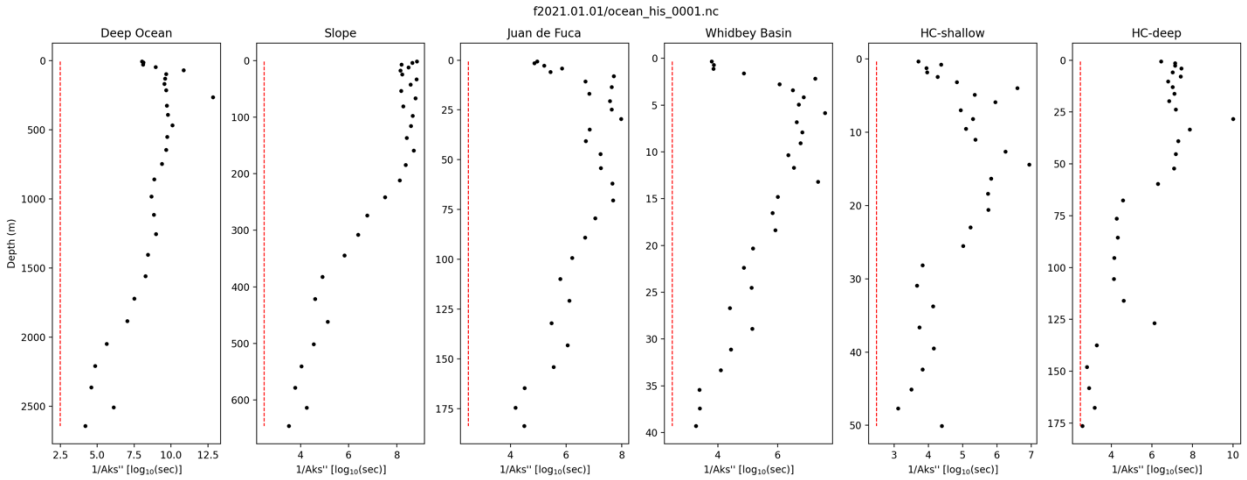
Regarding the use of the 3-point Hanning window: As far as I can tell, North et al. do not in fact use this approach, but rather discuss a 4-point and 8-point moving average. If this is the case, then I would suggest adding a reasonably clear explanation of what a Hanning window is, and how it is used to smooth a function. Or alternatively, a reference to a clear explanation. I think this is an important point to make clear, as this is one of the few implementation details where the different models compared are known to be different.

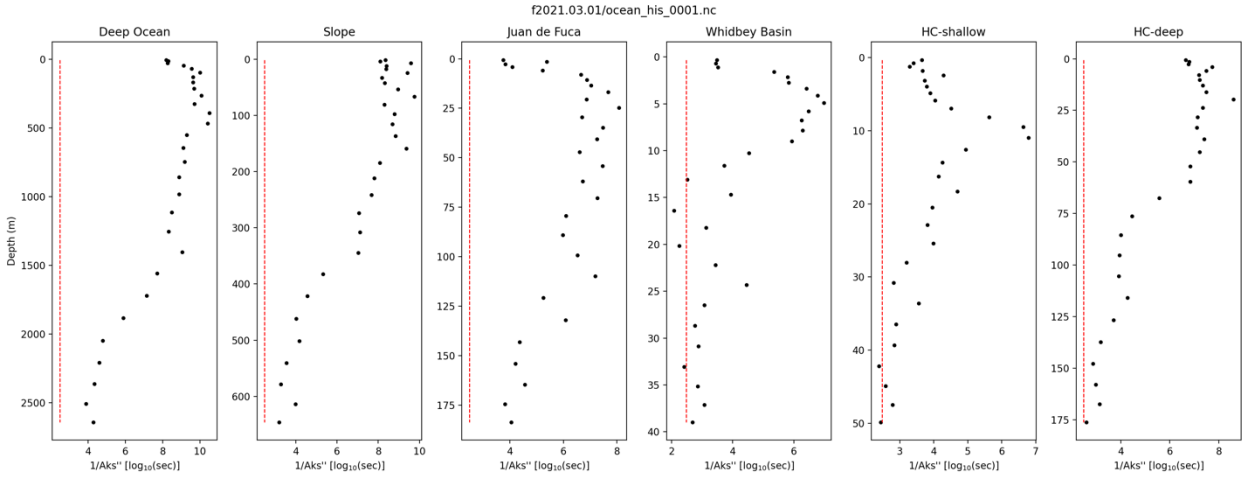
Response: we thank the review for pointing out lack of details for the 3-point Hanning window method. More explanations and the reference to this method were added in lines 98-101 in the tracked manuscript. The implementation of 3-point Hanning window can also be found in line 188 in trackfun.py (https://github.com/parkermac/LO/blob/main/tracker/trackfun.py).

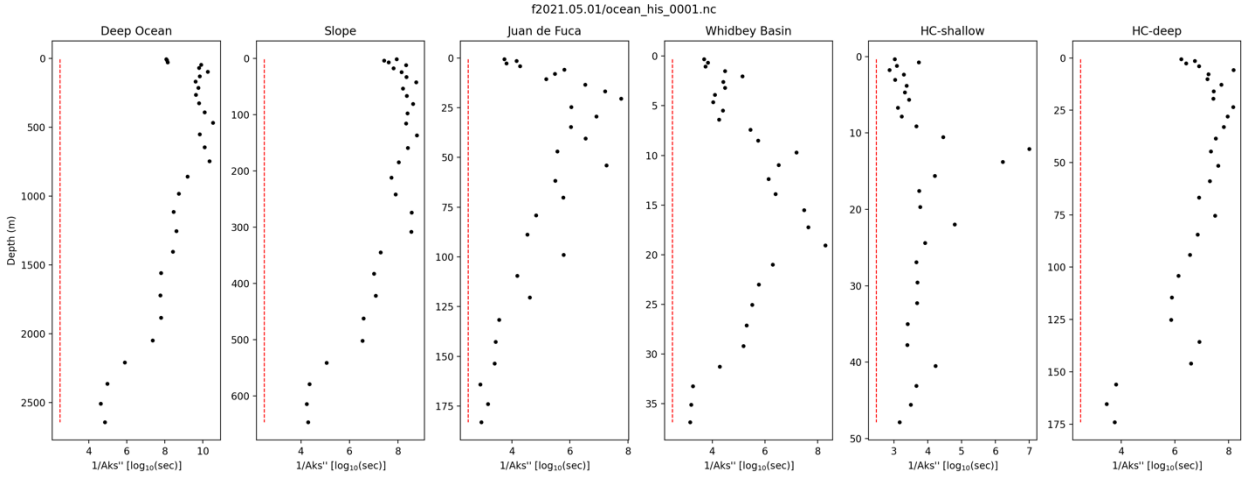
At lines 99-100 in the revised manuscript, you state that Eqs. (1) - (3) are solved with 4th-order Runge-Kutta. However, these equations are not written as ordinary differential equations, but rather already presented as discrete numerical schemes. Specifically, Eqs. (1) and (2) are the forward-Euler representation of the horizontal transport equation, and Eq. (3) is the Visser-scheme implementation of the stochastic differential equation for vertical transport (which can in any case not be solved by 4th-order Runge-Kutta, as that is an ODE method, not an SDE method). This should be corrected, and it would also be useful to clarify here if the same timestep is used for horizontal and vertical motion, as it is fairly common to use a shorter timestep for the vertical transport. See for example discussions on the difference between horizontal and vertical equations and timestep in North et al. (2006) or Rowe et al. (2016), both of which are in your list of references.

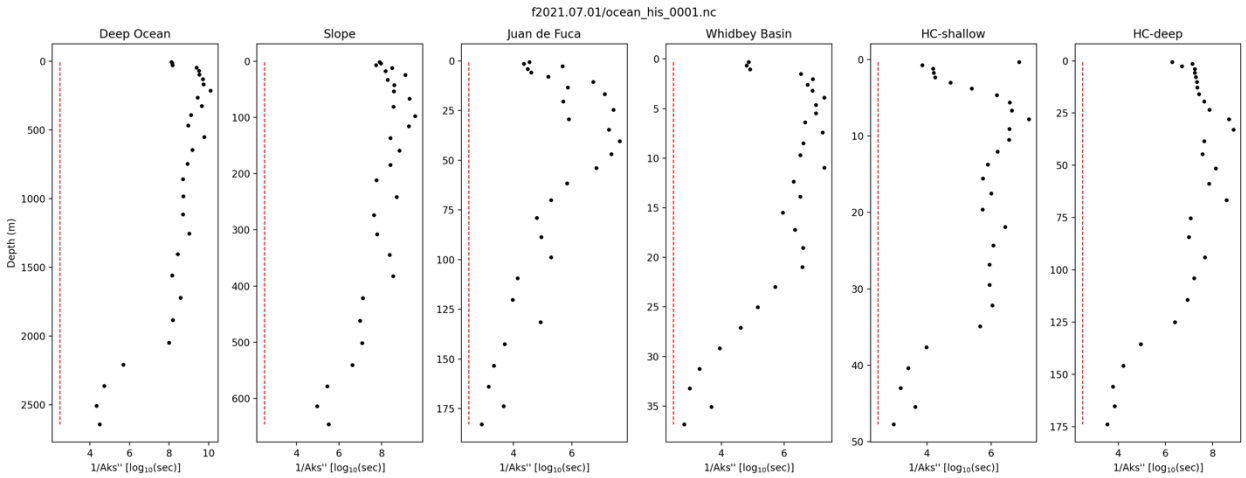
Response: thank you for the suggestions and we apologized for the unclear statement. The velocity components (u, v, w) used in Equations 1-3 are obtained using the 4th-order Runge-Kutta scheme. We corrected the statement in lines 103-105 in the tracked manuscript.

We also clarified that the same timestep is used for horizontal and vertical motion (lines 105-108). The random displacement model criterion requires dt << min(1/abs(AK_s'')) (Visser, 1997; North et al. 2006; Rowe et al. 2016). AK_s'' is the second derivative of vertical diffusivity. We examined the profiles of 1/abs(AK_s'') (please see figures below, the x-axis is log10 scaled) based on the hourly AK_s output at the six stations that were selected to test the vertical well-mixed conditions. AK_s was smoothed using a 3-point Hanning window. The red dash line denotes the timestep of 300s used in Tracker and other offline tracking packages. In most occasions, dt << min(1/abs(AK_s'')) could be achieved in the model.


f2021.01.01/ocean_his_0001.nc
Deep Ocean
Slope
Juan de Fuca
Whidbey Basin
HC-shallow
HC-deep
Depth (m)
1/Aks'' [log10(sec)]


f2021.03.01/ocean_his_0001.nc
Deep Ocean
Slope
Juan de Fuca
Whidbey Basin
HC-shallow
HC-deep
Depth (m)
1/Aks'' [log10(sec)]


f2021.05.01/ocean_his_0001.nc
Deep Ocean
Slope
Juan de Fuca
Whidbey Basin
HC-shallow
HC-deep
Depth (m)
1/Aks'' [log10(sec)]



Lines 127-130: The approach to finding the "significant range" for the WMC test is perfectly fine, but just as an observation (feel free to ignore), it is possible to make this a bit more "rigorous". If you consider any one of the 28 bins, then the probability of a particle ending up in that bin (when positions are drawn uniformly) is 1/28. Hence, drawing 4000 particles and counting the number in a bin is equivalent to drawing a number from a binomial distribution with 4000 trials and probability of success 1/28. The mean of such a distribution is 4000 * (1/28), and the variance is 4000 * (1/28) * (27/28). If you let the "significant range" be for example the mean plus or minus two times the square root of the variance, then you expect the number of particles in that bin to be outside the range about 4.5% of the time.

Response: we appreciate the reviewer's idea and checked that by this way, the mean += 2*sqrt(var) = 119 ~ 166, which is close to 102.5~187.3 used in Figure 2. Therefore, we would like to keep our approach to finding the significant range for the WMC test.

Line 150: It says "decrease their variance", but it should perhaps be "increase"?

Response: it is actually "decrease". We reworded this sentence to make it clear (lines 158-159)

Lines 226-230: It is not clear why you chose to convert the concentration in ROMS to "an equivalent number of particles", at the center of the cell. Is it not better, easier and more accurate to present the horizontally integrated concentrations directly?

Response: We agree with the review that it will be more accurate to present the horizontally integrated contractions of dye directly and will give a smoother vertical distribution in Figure 5. Here, in converting dye mass to an equivalent number of particles, our intention was to give dye and particle distributions from different particle tracking packages the same unit, i.e., particle number, so that they could be compared directly in Figure 5.

Finally, regarding the data: I believe GMD has a data availability policy that includes such things as "data sets for forcing of models". (https://www.geoscientific-model-

development.net/policies/code_and_data_policy.html). You say in the reply to reviewers that the dataset is around 200 GB, but it should be possible to reduce the size quite significantly by "cropping" (with ncks, for example) the data horizontally to include only what is needed to reproduce the results. The model domain shown to the left in Fig 1 appears to be 1000 or so from south to north, but the longest example trajectories shown only need a rectangle of data of about 160 km by 120 km, which should reduce the size a fair bit.

Response: we appreciate the reviewer's constructive comments and uploaded our code and exemplary hydrodynamic outputs to https://github.com/Jilian0717/LPT_intercomparison/tree/main and https://zenodo.org/records/10223144. These two links were also added in the section of Code/Data availability.

Instead of trying to make model extractions to fit the particle tracks, we took the approach of including a subset of the ROMS history files that could be used with our code. We hope that this is sufficient.

Additional private note (visible to authors and reviewers only):
In what conditions/circumstances should Tracker be used instead of any other model and why?

Response: please also see our response to reviewer 1 above. Generally, Tracker runs faster than other particle tracking packages that were tested in the present study, especially in the large model domain (please see Figure 9). In this study, we demonstrated that all tracking packages, although with different specialties, would give very similar results. Therefore, if computation time is the foremost factor, for example, in a forecast system with a large domain size like LiveOcean and millions of particles are required, Tracker would be a good choice. In Figure 9, we also observed that Particulator (written in MATLAB) runs very fast with one million particles, yet MATALB is a commercial software that requires not cheap license, thus much less accessible than Python.