Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-205-RC1, 2017

Geoscientific Model Development Discussions

Interactive comment on "OpenDrift v1.0: a generic framework for trajectory modeling" *by* Knut-Frode Dagestad et al.

Anonymous Referee #1

Received and published: 21 September 2017

In this manuscript, the authors present an overview of the design philosophy and some of the use cases for the new OpenDrift v1.0 particle tracking package. They highlight its flexibility and extendability to real-world problems.

This is a very readable, nicely-written manuscript describing an impressive piece of code. I have read it with much excitement, and congratulate the authors for some of the sophisticated software engineering they've done. They have seemingly mastered to make a particle tracking framework that is both very powerful as well as easy-to-use; not an easy achievement!

While I think this manuscript and the associated code is almost in a shape suitable for publication in GMD, I do have a few minor comments that the authors may want to address

- There is little actual benchmarking of the performance of the code. While the authors state that their code is efficient and fast, it would be good to see some quantification of these statements, ideally in comparison to similar other tools available. The advantage of providing benchmarks in this v1.0 paper is that it will also provide the authors a target for future versions of the code

- In general, I would have liked to see some more discussion of technical details and choices. This manuscript will probably become a GMD paper, with a readership of scientific code developers; yet the manuscript reads mostly like a high-level user guide. While I think most of these high-level descriptions are useful and should stay, a bit of 'under-the-hood' detail might please the technicallly-inclined reader

- On lines 6 and 7, is there a difference between 'model' and 'module'? If not, I suggest using the same word as it may be confusing to readers

- line 9: 'subjective estimates' here is a rather vague term. Perhaps explain a bit more what is meant?

- line 41: Another advantage of offline models, as the authors later also acknowledge, is their ability to track backwards in time

- line 61: 'robustness' for operational use is a rather vague term. What do the authors mean with this?

- line 70 and 76: The section header is named 'Reader class' but then later the authors state that Reader is a subclass. This may be confusing to some readers

- Table 1: How were these packages selected? Is this table really necessary? Some of the more widely used packages in large scale oceanography, such as Ariane and the Connectivity Modeling System, are missing

- line 82: Where is this 'ocean_model_output.nc' file located? It does not appear on the GitHub. Would it be good to provide access to these files somewhere, for users who want to recreate the simulation described here?

- line 83: Mention here already that OpenDrift supports ‘lazy reading’ of files? This is a really neat feature, as the authors also proudly acknowledge later, so best to at least mention it here too?

- line 127: Can any more technical details be provided of the caching mechanism? Does it use xarray? Dask? Something custom developed?

- lines 152-167: the frequent forward-referencing to later sections here makes for difficult reading. Can’t the order of the sections be reshuffled somewhat to reduce the amount of forward-references?

- line 160: I am surprised that all history of particles is stored in arrays. Would these arrays not get potentially enormous? Are they kept in memory? Is there any smart memory-management implemented? What does this mean for scalability of the code to very many and/or very long trajectories?

- line 190: All examples of Reader subclasses have only one file. What if the data is in multiple files? Would that be a common use-case?

- line 225: How is ‘evenly’ defined here? Which algorithm is used?

- line 228: Is it also possible to create one’s own seed_elements methods? Can they be overloaded?

- line 236: I assume ‘runge-kutta’ is a 4th order Runge-Kutta integration? Or 2nd order?

- line 252: Is the ‘advective increments’ the same as the time_step argument?

- section 3.1: some of this discussion, especially in the second paragraph, feels a bit out of place in a GMD paper about methodology and code. Is this really all necessary?

- line 287: some words seem missing at the end of this sentence

- line 314: Is stranding the only option available in OpenDrift? Or can one also use e.g. reflective boundary conditions?

- lines 516-540: again, much of this discussion does not seem very pertinent to the code and out of place in a GMD paper.

- There is no mention of paralel capacity for OpenDrift. Is the code parallelised, i.e. through numba? If not, are there any plans for a paralel version?

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-205, 2017.