Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2017-167-RC1, 2017

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "Parcels v0.9: prototyping a Lagrangian Ocean Analysis framework for the petascale age" *by* Michael Lange and Erik van Sebille

J. Kjellsson (Referee)

joakim.kjellsson@physics.ox.ac.uk

Received and published: 19 August 2017

The manuscript presents a new framework for a Lagrangian particle model, Parcels. The new particle model is in a testing phase with only the most basic components functional. The authors describe the current workings of the model, test its accuracy, and present where they envision development going further. Overall, the manuscript is well written. The main novelty lies in presenting the new framework to the particle-modelling community and its possible future developments. However, there are very few actual results. I recommend the paper for publication, but only after addressing the comments below.

Printer-friendly version

Discussion paper

Major comments

There is not much discussion about how exactly the new model will be more suited to cope with petascale age computing. The authors spend some time talking about how to optimise the loop over particles to improve performance, but with petascale OGCMs, where velocity fields amount to hundreds or more terabytes, reading and interpolating those fields into the particle model will be a huge bottleneck. Section 5.1 has a paragraph on how reading data from massive files could work, but there is no demonstration. In the practical example, the data file is 6Gb, which is not very large. I understand the authors have not focused on optimisation of PARCELS yet, but are there any examples, not necessarily with particle codes, where spatial indexing has given a performance improvement? I strongly recommend more discussion (in the introduction, design, and discussion sections) about how all current particle codes, e.g. CMS, Tracmass, Ariane, will hit this bottleneck in the peta-scale age, and more details about how PARCELS will overcome it.

Since PARCELS is very flexible, could it be extended to work for atmospheric particles? Perhaps PARCELS should not be presented as a tool for Lagrangian ocean analysis, but rather Lagrangian particle tracking in both atmosphere and oceans? Presenting this kind of framework to the atmospheric modelling community as well could be beneficial, but would mean changing the paper quite a bit. Even if the authors decide to stick with presenting PARCELS as an ocean particle code, atmospheric particle codes still need some mentioning (MetOffice NAME model, FLEXPART) in the introduction.

Minor comments

Throughout the paper, the authors name the model "Parcels". However, more than once I found that the name of the model could be confused with actual parcels. Why not use PARCELS, as any other model (e.g. NEMO, CESM, IFS etc.) to avoid confusion?

Page 1, Line 1: "petascale age" is rather unspecific. The sentence uses future tense, suggesting we are not there yet, even though there are already $> 1petaflops$ computers. Please specify what is meant. OGCMs of a certain horizontal resolution, e.g. global $\sim 1/50$ or $\sim 1/100$?

Page1, Line 19: Add reference for seawater parcels: Doos 1995, Blanke & Raynaud 1997.

Page 2, Line 16: How would it keep up? By being very efficient at reading in velocity data?

Page 2, Line 18: I recommend replacing "functionality such as a myriad of behaviours to the particles" for "active particle behaviours".

Section 2.2: I found the section a bit confusing, and I think it needs some rewriting to become clearer. As I understood it, these are two methods for interpolating data, e.g. velocity, onto the particle position? I'm familiar with the interpolator from SciPy, but what method does the JIT method use? Is that something the user can write himself/herself? Is the SciPy interpolator restricted to nearest-neighbour or linear interpolation methods? Also, what pre-defined macros are you referring to?

Page 6, Line 11-13: Non-compatibility with non-regular grids excludes quite a few OGCMs, which often use rotated pole, tri-polar or cubed-sphere grids. I think this is one of the most important shortcomings of v0.9 that must be addressed soon by the authors or the user community. The authors should say so.

Page 7, Line 5-7: This sentence does not read well with its two parenthesis. I would split into two sentences, e.g. "The bash script get$_ofesdata_agulhas.shprovidedathttps$: $//doi.org/10.5281/zenodowasusedtodownloadsnapshots3165to3289, coveringtheyear2006, ofasubdomainaroundthecoresi$

Page 8, Line 13: 6Gb is actually not a very large file. Many laptops have 16Gb RAM these days and could definitely cope with this while the user sips his/her coffee at some hip cafe.

Section 3.5: Does PARCELS write CF 1.6 compliant data?

Sections 4.2.1 - 4.2.7: The test cases need to be described a bit more. Are the fields generated within PARCELS, or generated and stored as netCDF files and then read into PARCELS? Also, please give $\Delta x, \Delta y, \Delta t$ for all fields.

Page 10, Line 28: "steady-state"

Section 5.1: Are there any tests that show that the optimisations they propose would give some performance improvement? Optimising the reading of velocity fields from very large files would be one of the main strengths of this model. See major comment above.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-167, 2017.

Printer-friendly version

Discussion paper