Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2018-339-AC1, 2019

Interactive comment on "The Parcels v2.0 Lagrangian framework: new field interpolation schemes" *by* Philippe Delandmeter and Erik van Sebille

Philippe Delandmeter and Erik van Sebille

p.b.delandmeter@uu.nl

Received and published: 4 April 2019

We would like to thank Dr Knut-Frode Dagestad for his careful reading and its constructive comments. Please find our replies below.

Philippe Delandmeter and Erik van Sebille

General comments

This manuscript describes the upgrades leading to version 2.0 of the Parcels lagrangian drift code, with emphasis on new field interpolation schemes. A short section is devoted to mathematical validation of the new schemes, and a longer section is devoted to a demonstration study of floating microplastics. The manuscript is well written, with few typos. Figures, in particular the field diagrams, are very clear and useful. In general the paper demonstrates and documents nicely the accuracy, power and flexibility of the Parcels software, and is as such worthy publication In GMD. My only major concern about this paper, as commented below, is that the study of floating microplastics does not address very directly the new field interpolation schemes, which are the major focus of this paper.

Thank you. In the answer below, we address the different comments, with a special care about the main one concerning the relevance of the North Sea microplastic application.

Specific comments

The new interpolation schemes addresses z and sigma as vertical coordinates. However, the widely used ocean model HYCOM uses a hybrid combination, with z-coordinates near the surface and sigma-coordinates below. Please comment whether the existing schemes are applicable to native HYCOM-output, or if such an extension would be simple/feasible, and eventually whether it is planned for the (near) future.

State-of-the-art ocean models use a large variety of vertical coordinate systems and vertical mesh discretisations. In Parcels, we differentiate them into two categories: z- and s- levels. The z-levels are defined as a vector of the depth of the mesh levels. The s-levels are a 3D-array of the mesh level depths, depending on lon-lat position and vertical levels. The combination of z- and σ- levels in HYCOM is then naturally read by Parcels, that considers them as general s-levels. We added this discussion in the revised manuscript (page 11, line 16 of the revised manuscript).

It could also be commented whether there are any specific plans to support unstructured grids in the (near) future.

Incorporating unstructured grids involves three main developments: (1) interpolating a field in a cell; (2) finding in which cell the particle is located; (3) a new data structure. We already do (1) for general quadrilaterals (2D) and hexahedra (3D), but other common meshes (triangles and prisms) would be required to support unstructured grids. To find where is our particle (2), we use some properties of the mesh, that do not hold on unstructured meshes. The current data structure is built for structured data and should then be generalised (3). The Parcels team is still a relatively small group and there is no plan to support unstructured grids in the near future.

The paper addresses various types of fields, but does not mention file formats. E.g. would any CF-compatible netCDF-file be directly ingestible by Parcels 2.0? What would be the approach for using model-specific output formats, including non-netCDF?

The most common way to provide input data to Parcels is indeed to use netCDF files, although other formats such as numpy or xarray objects are also available. The Parcels input data loading functions have been improved since version 0.9 (Lange et al., 2017), but their general structure has remained the same. Following your comment, we have added a note on the input data format (page 11, line 22).

In section 2.1.1 velocities are referred to as “zonal” and “meridional”, and in Section 2.2.1 it is said that longitude and latitude are 1D arrays for rectilinear grids. Does this imply that only lon-lat (Plate Caree) and Mercator grids are supported, and not other projections such as e.g. polar stereographic?

Parcels is not specifically designed for lon-lat coordinates, even if those are the coordinates in most of the applications. As long as the coordinates (which are abusively called lon, lat, depth in Parcels) are in the same system as the velocities, there is nothing to do. If it is not the case, velocities can be rescaled using UnitConverters as it is explained in page 12 line 16. UnitConverters between lon/lat and metric systems are already implemented, but other can be added by the user. Finally, if the velocity need a more specific transformation (for example a rotation), this needs to be done within a kernel, that can also be added by the user.

Is the coastline always considered to be land-pixels of the ocean model, or does Parcels 2.0 also support using e.g. vector coastlines such as GSHHS? This affects e.g. the question about impermeability (i.e. that ocean currents do not have an onshore component). The interpolation schemes assures impermeability at the coast, which is a good property, but this would be difficult to assure if the same ocean models is not used for the landmask. And with nested grids, each ocean model would then need to use their own coastline to assure impermeability?

This is a very interesting comment. Parcels does not support directly vector coastlines. We prefer to read the data as they were generated by the OGCM for a better consistency. But as you pointed out, when multiple fields from different datasets are used, this can result in situations where the particle is considered on the beach according to one field, but not the other. The solution to this situation is problem dependent. This is again where Parcels kernels show their importance. For example in the 2D north sea microplastic application, we do have multiple fields with different coastlines. We define a particle as on beach if it is out of NEMO or NWS boundaries. A kernel is naturally dealing this situation (https://github.com/OceanParcels/Parcelsv2.0PaperNorthSeaScripts/blob/master/northsea_mp_kernels.py). If the user wants to use GSHHS data for example, it can easily take advantage of the Python language of Parcels, to rasterize the data on a grid and provide it to the model.

The WaveWatch Stokes drift is only available up to 80 degree North, nevertheless the simulations including Stokes drift shows particles further north. Is the Stokes drift simply neglected northwards of 80 degrees, instead of deactivating the particles?

Indeed, the Stokes drift was simply neglected northwards of 80°N. However, when revising the code, we spotted a bug that was highly affecting the Stokes dynamics: due to a typo, the time step had been removed from the Stokes dynamical kernel. We have now fixed this bug and revised entirely the code to ensure that such error was only present there. After this fix, the dynamics of the MP including Stokes drift have changed drastically. The new results are discussed in the revised manuscript. The particles going North are now negligible.

With the Nemo 1/4 degree model, more MP is trapped along the Norwegian Coast than with the 1/12 model. Can you explain why this is the case?

This is interesting. Indeed, more MP is trapped along the Norwegian coast with the low resolution model. This is not a consequence of beaching particles, since any particle accidentally beaching would be pushed back towards the water, and we observed that such beaching was negligible. When observing the particle dynamics now available in the supplementary materials, we observe that the particle main path stays further away from the coast with the HR data. This might be a consequence of the HR data, which produce higher lateral shear in the surface velocities, that acts as a barrier protecting the beach from open waters. We discuss this difference in the revised manuscript (page 18, line 6).

The sensitivity study of Section 4 addresses "floating microplastics". It is not stated specifically, but I assume this means that the MP is considered to be at the very surface all the time? If so, this has some implications which should be commented. E.g. have *in situ measurements (e.g. Kukulka, 2012, Kooi, 2016) shown that MP particles are mixed deeper in the water column with increasing wind speeds. This implies that the real Stokes drift is less than the surface Stokes drift, which is presumably used in this study?*

Yes, the MP stays the whole time at the surface in those simulations. It is a strong assumption, which is taken in many studies of the MP distribution (Onink et al., 2019). Better approximation of the plastic dynamics are currently being developed, and we are working on this as well. But this paper, which focuses on the numerics, simply aims to show how the distribution of the surface MP could be affected by the use of different datasets. As we have written, it is meant as an illustration of the Parcels framework, not a comprehensive study of MP in the North Sea.

Following the comments of both reviewers, we also run 3D passive particle dynamics in this revised version. This also shows the effect of 3D dynamics on particle transport (see also new figure showing vertical distribution), even though this second simulation is not specifically dedicated towards plastic transport, which do not follow passively the 3D hydrodynamics.

Figure 8 shows e.g. that the effect of adding Stokes drift, is that more particles are kept at the Norwegian coast, and less is advected into the Arctic Ocean. However, it is very hard to see this from Figures 6 a) and d), probably due to the logarithmic scale. What is the reason for choosing a logarithmic scale on Figure 6 and 7?

We have first post-processed the data using linear scales (see Figs 1 and 2 of this comment). Such graphs give a lot of details in a certain range of concentration, but the multiscale dimension of the distribution is hard to see. With the logarithmic scale, we have exactly the opposite situation. That is why we chose the second option, but have Figure 8 which, as you pointed, provides information that is hardly seen on the maps.

Section 4.2 explains that impermeability condition applies to the ocean current advection, but that Stokes drift and diffusion allow stranding. However, if diffusion is regarded as unresolved (sub-grid) ocean currents, these should in principle also not have any onshore component. This would imply that the amount of stranding is overestimated.

Ocean current data (NEMO and NWS) have impermeable boundary condition, that we keep in our Lagrangian simulation. Stokes drift data do not have impermeable boundary condition, such that we allow Stokes drift to beach the particles. The third source of transport is the diffusion. It parameterises unresolved processes, which includes sub-grid scales and the coastal boundary layer, the last one allowing beaching. We then chose a boundary condition that allows beaching. Since the simulation has not been calibrated against observation, it should not be seen as a realistic simulation of the MP transport, especially close to the coast, but as a study of the importance of the diffusion term.

It would also be nice to have some short comments about the implementation of the new interpolation schemes. Are these programmed in Python, or in C? Are they programmed at a lowest level (i.e. the equations as shown in this paper), or are some external higher level Python libraries used? Any comments about computational time/performance would also be welcome, either in general terms (fast, very fast, slow. . .) or as numerical metrics.

The schemes are implemented at the lowest level, as they are described in the manuscript. Parcels can be run either in a Python-C coupled way (for efficiency) or fully in Python (development mode), such that the schemes are implemented twice: in Python and in C. Interpolation schemes for A-grid are available in the scipy library, but this is not the case with our new C-grid interpolator, such that we do not use any external interpolation library.

While this manuscript focuses on the interpolation schemes, we are currently working on the model performance and develop a parallel implementation of the code. So far the overall CPU time is dominated by IO communication, such that we do not see any difference in terms of performance between A- and C- grids.

As mentioned under the General Comments, the study of floating microplastics does not appear to test specifically the new interpolation schemes, apart from demonstrating that fields of different grids may be used and combined. NEMO input is on a curvilinear C-grid, whereas CMEMS input is on an A-grid (rectilinear?), but the vertical coordinates (z or sigma) are not mentioned. However, if the plastics is considered to be at the surface all the time, the vertical interpolation is not even used in this study. A more direct validation/test, would e.g. be to compare two simulations with 3D-drift with the CMEMS/NEW data set in respectively native coordinates (C-grid curvilinear), and the regridded (A-grid rectilinear) data as used in this study. In addition to quantifying the differences (hopefully small?) of the spatial distribution/drift for such a case, it would be interesting the get an idea about the difference in computational time, where using pre-regridded data would be expected to be faster.

After reading your comments and the ones from Dr Kjellsson, we have included a new 3D simulation, that advects passive particles interpolating the NEMO 1/12° data, which are discretised on a C-grid with z-levels. This new simulation is not a proof that the interpolation scheme dynamics, the analytical proofs being provided in Section 3, but an illustration of a 3D run in Parcels.

We did not directly compare our simulation on a C-grid, with a simulation with the same regridded data: first, we did not have such data; but more importantly, as we commented to Dr Kjellsson, this would not be a proof of our interpolation schemes dynamics, but a validation of the regridding algorithm. If A-grid data are generated in a conservative form with correct boundary conditions, there will still be some difference between the two dynamics, but this difference will be small. The advantage of our approach is that no regridding is necessary, but the users can interpolate the data they have access for the different types of grids.

Technical comments

Figure1: it could be commented that all 4 combinations of horizontal and vertical grids are possible.

Indeed. We added such comment.

Both in Section 2.1.1 and 2.1.2 there are unnumbered sub-headings names “2D field” and “3D field”. This could lead to confusion when jumping back and forth between pages/sections.

While we agree with your comment, technical instructions from Copernicus require to not use paragraphs but only subsubsections. We have emphasized at the beginning of 2.1.1 and 2.1.2 that we consider separately the 2D and 3D cases.

Section 2.1.1, line 15. The meaning of this sentence is unclear: “The interpolation must use local information in the cells.”

Indeed, this sentence was not clear and have been rephrased.

Section 2.2.1 says that data is read lazily, which is a nice property. Is this based on external libraries such as dask, or is it explicitly programmed in Parcels?

The Python-C structure of Parcels currently prevents us from using lazy loading functionalities of xarray and dask. Indeed, the data needs to be loaded into memory before being manipulated by the C-library. We still implement lazy loading, by only loading the data time steps when required. This is directly implemented into Parcels.

There are links to the interpolation code, which is said to be independent of Parcels. Does this mean that it is implemented as a stand-alone library which is used by Parcels, or is it (also) directly included in the Parcels codebase?

The simple interpolation code provides information to the reader who wants to see a Python implementation of the schemes developed in this paper. In Parcels, the interpolation schemes are directly implemented in the code, independently from the small library.

Throughout the paper, Microplastics is abbreviated as MP, which is fine. However, for the figure captions it might be useful to be explicit, as figures are sometimes used/read out of direct context of the paper.

You are right. We modified the figure captions in that sense.

Figure 7: Could also comment here that scale is logarithmic.

Done.

Page 14, line 14: please give the Ifremer FTP address (or refer to the data availability section at the end)

We added at the beginning of the data section that all links are provided in the data availability section.

Page 15, line 9: "consisting at" $->$ "consisting of".

Done, thank you.

Page 16, line 4: could specify that 1/4 degree is longitude, and 1/8 degree is latitude.

We modified the sentence.

Page 16m line 12: “even if this” $->$ “even if the”

Done.

The North West shelf reanalysis is referred to as “CMEMS”. However, CMEMS provides a lot of different data, also including NEMO. Thus I would recommend using a more specific reference, such as e.g. “NWS”.

You are right. We now refer to the North West shelf reanalysis data as NWS.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-339, 2019.

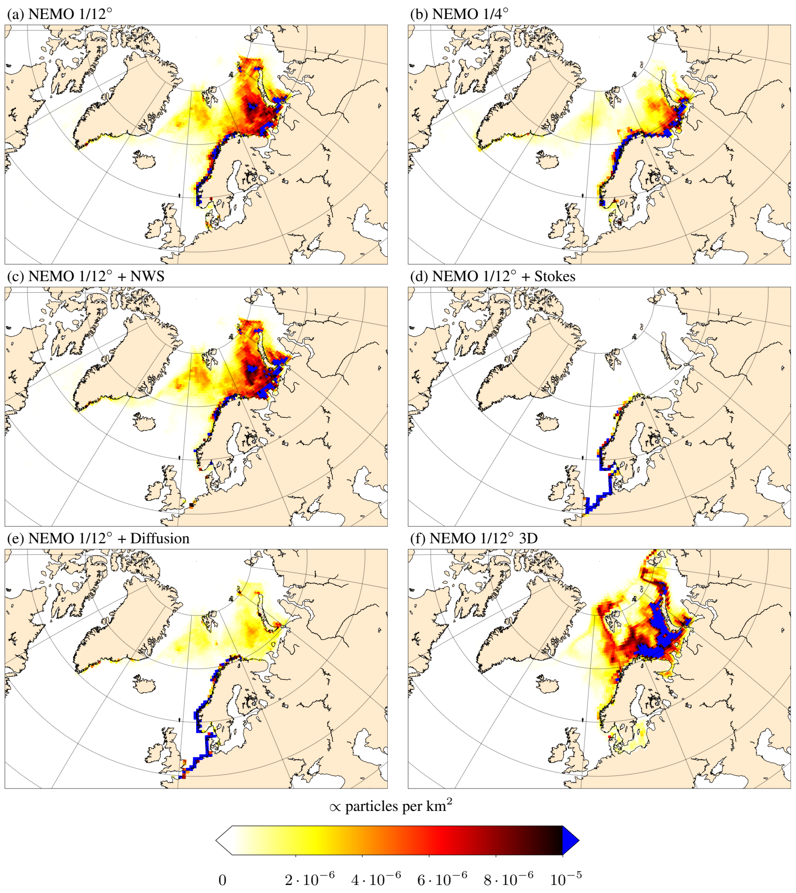
(a) NEMO 1/12°
(b) NEMO 1/4°
(c) NEMO 1/12° + NWS
(d) NEMO 1/12° + Stokes
(e) NEMO 1/12° + Diffusion
(f) NEMO 1/12° 3D
∝ particles per km²
0
2 · 10⁻⁶
4 · 10⁻⁶
6 · 10⁻⁶
8 · 10⁻⁶
10⁻⁵

Fig. 1.

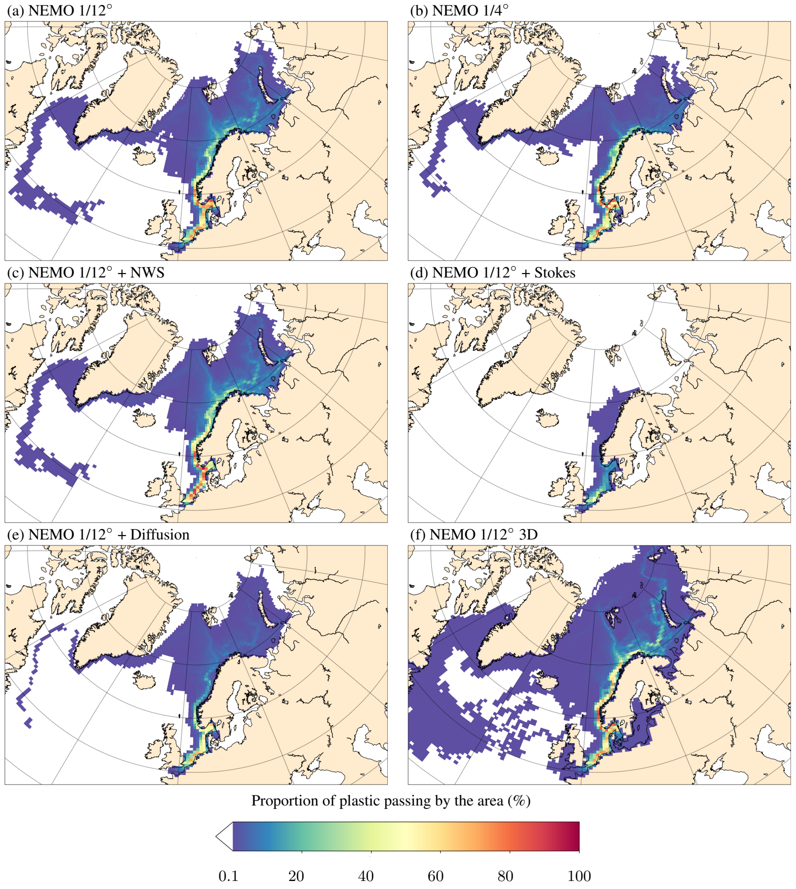

Fig. 2.

Printer-friendly version

Discussion paper