The Parcels v2.0 Lagrangian framework: new field interpolation schemes

Philippe Delandmeter[1] and Erik van Sebille[1]
[1]Utrecht University, Institute for Marine and Atmospheric Research, Princetonplein 5, 3584 CC Utrecht, The Netherlands
Correspondence: Philippe Delandmeter (p.b.delandmeter@uu.nl)

Dear Editor,

Please find enclosed our revised manuscript "The Parcels v2.0 Lagrangian framework: new field interpolation schemes". We have amended the manuscript and figures to address the issues raised by both reviewers, and have included a detailed reply to their comments. You will also find hereinafter our point by point responses to both reviewers and a version of the manuscript where the differences with the previous version are highlighted. Note that apart from the manuscript modifications, we now also provide animations of the simulations in the supplementary materials.

Thank you again for considering our manuscript for publication. We are indebted to Dr Knut-Frode Dagestad and Dr Joakim Kjellsson for their constructive comments. We appreciate your time and look forward to hearing from you.

Yours faithfully,
Philippe Delandmeter and Erik van Sebille

Response to Dr Knut-Frode Dagestad's review

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

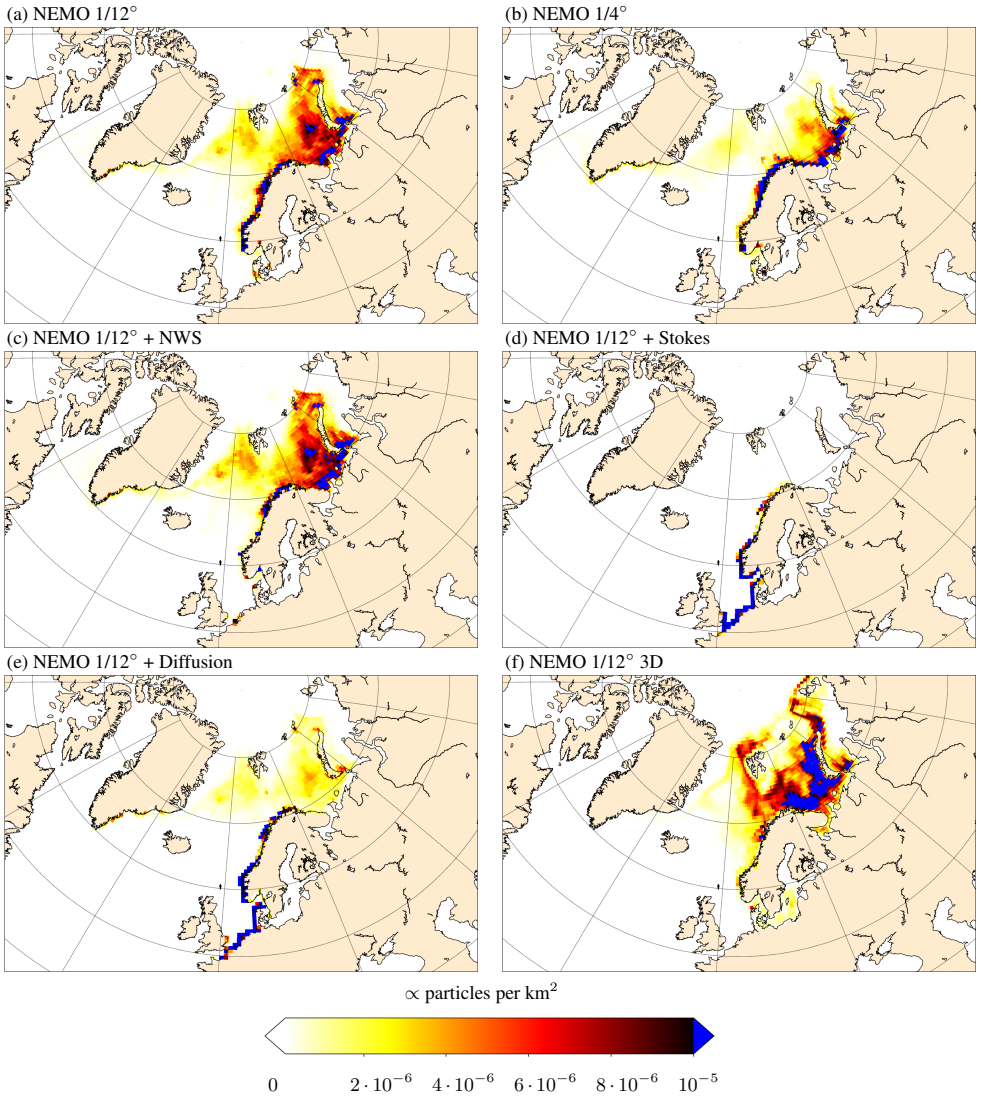

Figure 1: Floating microplastic density, averaged over the third year of particle age, for the different simulation scenarios: (a) NEMO 1/12°, (b) NEMO 1/4°, (c) NWS nested into NEMO 1/12°, (d) NEMO 1/12° coupled with Stokes drift from WaveWatch III, (e) NEMO 1/12° coupled with diffusion. Note the linear scale different from the manuscript.

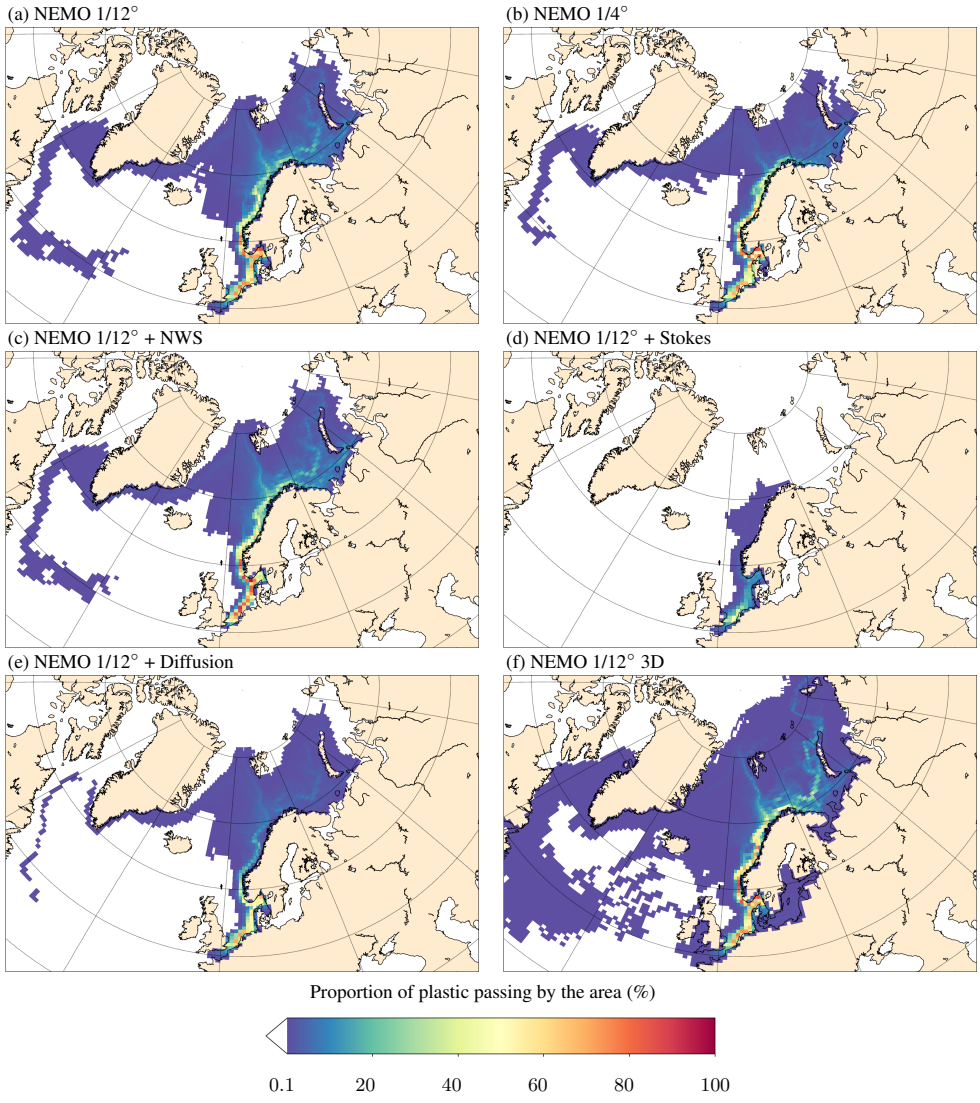

Figure 2: Fraction of floating microplastic originally released in the Thames and Rhine estuaries reaching the domain. It is computed for each cell of a 1/4° longitude x 1/8° latitude grid, as the proportion of the particles that have visited at least once the cell. Note the linear scale different from the manuscript.

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.

Response to Dr Joakim Kjellsson's review

We would like to thank Dr Joakim Kjellsson for his careful reading and its constructive comments. Please find our replies below.

Philippe Delandmeter and Erik van Sebille

Summary

The paper describes the new version of Parcels, v2.0. The new version includes new interpolation schemes for tracing particles which allow for various vertical coordinates and staggered grids. As the paper presents these new and necessary features of Parcels I recommend it should be published, but only after some re-working of the text and also using a 3D test case rather than a 2D case.
Thank you. Please find our answers to your different comments below.

Major comments

The authors spend quite some time deriving and explaining the new interpolation schemes for z and s coordinates on staggered grids, which is a new feature of Parcels. I'm therefore a bit puzzled that the showcase in the Results section is only for surface drift. I would strongly recommend the authors to change the showcase to some example with subsurface 3D flow, so that we can see the z or s coordinates in action.
Following your comments and the ones from Dr Dagestad, we haved added a simulation with 3D passive particles in the North Sea to compare surface and 3D transport in the North West European continental shelf. In this run, the particles dynamics follow the NEMO 1/12° data, which are discretised on a curvilinear C-grid with z-levels.

The writing is in need of revising (see specific comments further down). In particular, I struggled with Page 6, Line 23 to Page 9 Line 20, which did not flow well and was at times confusing. This section needs a bit of re-writing and re-structuring. I also found Section 2.1.1 to be very abstract and I had to wait until Section 3 before the methods to be described in a more practical sense. I would strongly urge the authors to either put Section 3 directly after 2.1.1 or somehow merge the two sections so that the definitions of interpolation schemes are directly followed by how it is done in practice.
Section 2.1 introduced a lot of definitions and notations, which made it hard to read. Based on your comments (here and below), we have restructured the section to help the reader along with the construction of the interpolation schemes, highlighting the main four steps:

1. define a mapping between the physical cell and a unit cell;
2. compute the fluxes on the unit cell interfaces, as a function of the velocities on the physical cell interfaces;
3. interpolate those fluxes to obtain the relative velocity;
4. transform the relative velocity to the physical velocity.

Before we were describing the scheme by starting with what we want (the velocity at given x, y, z) and progressively building the variables we need, ending with the gridded input data. We have reversed that order for the 2D case, starting from what we have and reaching what we want. This structure follows the one in Section 3, that you commented as easier to understand. For the 3D case, we keep the original order since there is no point to introduce the different fluxes (U_0, $U_{1/2}$, U_1, V_0, ...) before motivating why we need them, but we keep referring to the four main steps of the interpolation. Section 3 is a validation of the schemes, and therefore appears after the complete description.

Minor overall comments

On the large scale I find that the authors over-use the word "different" as a synonym for "various", and I often found that the word could simply be omitted to make the paper easier to read.
We have removed the unnecessary occurrences of this word as well as others for an easier reading.

Parcels is referred to as being developed to meet the exa-scale challenge, when velocity fields and tracer fields become massive and traditional Lagrangian codes will struggle. However, none of the examples in the paper are what I would refer to as "very large data sets", and there are no results regarding Parcels run time, memory use etc. I would therefore change the focus a little bit and re-phrase the introduction and also throughout the paper to describe Parcels as flexible and user-friendly, which seems to be the big advantage of using Parcels, rather than focusing too much on computational efficiency.
The aim of Parcels is to build an efficient and flexible framework for Lagrangian ocean analysis (Lange et al., 2017). We simply remind this in the paper introduction, then do not insist on it since it is not the point of the paper.

Specific comments

Page 1:
Line 19: I would re-write to say "can, in turn, be used to analyse the global ocean dynamics given the flow field from the model." Followed by "The flow field can also be taken from observations, e.g. land-based measurement... "
We have modified the sentence.

Page 2:
Line 2: "and many other types", "etc." Line 26: "We then validate... "
Line 29: "the results."
Done

Page 5:
I'm wondering if Fig 2 is really necessary. The staggering of grids is also shown in Fig 3, and indices could also be added to Fig 3, thus making Fig 2 redundant.
Fig. 3 is already quite busy with other information. We preferred to not overload it, and keep the global indexing separately, in Fig. 2.

Page 6:
Line 15: "in the cells, and interpolating"
Line 16: "formulation. For instance, such interpolation"
Line 18: While I enjoy citations, it it enough to just cite Jonsson et al. 2015 (the Tracmass code) and Doos et al 2017 (a thorough model description paper).
Done.

Line 23: If I understand this section correctly, you calculate fluxes on the cell faces, then interpolate fluxes to particle position, and then interpolate cell face area to the particle position, and divide flux by area to find velocity? The section starts by defining the velocity and ends by defining the Jacobian, which makes it a bit confusing for the reader to follow how velocities are interpolated from the model grid to the particle position. It would make more sense to start by defining the fluxes U,V, then how they are interpolated to the particle position, and then describe how velocities are found. Line 24: "(Fig 2b). Velocities are not found by linear interpolation but, like in finite-volume schemes, they are approximated by linearly interpolating the fluxes (U0,U1,V0,V1) at the cell faces (fig 3b) and dividing by the cell face area."
Line 26: Should it be ". . . the velocity and any position (x,y) is derived as a function..."?

The interpolation schemes consist in (1) building a mapping between the physical and the unit cell; (2) computing the fluxes on the cell edges (in 2D) or faces (in 3D); (3) interpolating the relative velocity using the fluxes and the Jacobian of the transformation; (4) transform the velocity to the physical coordinates. We have rewritten the section to highlight the important parts of the interpolation process, reordering the description as you suggested.

Page 7:
Line 5: These are the velocities on the model grid? The section should end with an expression for how u,v are found. Line 9-11: The last two sentences seem out of place. Instead, you could add ", where indices are chosen to conform with the NEMO model (Madec et al)" on Page 6, Line 24. Line 17: l is the model vertical index? Conforming with NEMO model?
Those were the relative velocities. After the restructuring of the section, we now end with the expression for u and v as you suggested. The comment on the NEMO indexing was also moved following your comment. Yes, we use NEMO conforming indexing in both horizontal and vertical directions. l was the vertical index, since k was used somewhere else, but we have changed it such that we use the more common i, j and k notations for the grid indexing.

Page 8:
Line 1: what is meant by "do not resolve exactly a uniform velocity"? Do you mean "do not result in a uniform velocity"
Line 6: remove "different"
We meant that if the C-grid field was representing a uniform velocity, it was not possible to obtain for all x, y and z that velocity while interpolating the field with a linear interpolation scheme in 3D. We have reformulated that sentence.

Page 9:
Line 5: "and their respective fluxes are"
Done

Line 6: I like this Table. Could you do the same thing for the 2D case and also add to Fig 3? It would be a lot shorter, but I think it could be informative.
For the 2D case, the situation is much easier since there are only 4 fluxes, and the Jacobians reduce to the simple edge length. We have added the 2D computed fluxes definition in new Eq. 7.

Line 9-12: Replace with "We can compute the fluxes through grid faces [12,13,14,15] (in blue, Fig 4), [16,17,18,19] (in red), and [8,9,10,11] (in green) using the continuity equation. The flux through [12,13,14,15] is..."
We have reformulated that sentence.

Line 13-14: Is this only for fixes z coordinates? In the case of z or σ coordinates, the cell thickness varies in time which must be taken into account. If the time-varying part is taken into account here, please explain how.*
Thank you for pointing that comment. Indeed, this is only valid for a fixed mesh. For a moving mesh, the mesh expansion should be added to the continuity equation, which we don't do in Parcels. We have added this note to the document, with also a reference to Kjellsson and Zanna 2017, that quantifies the error induced if a moving mesh is assumed as fixed.

Line 16: What does the "+" superscript mean? What is the difference between U+ and U?
There is an important difference between $U_{1/2}$ and $U^+_{1/2}$. $U^+_{1/2}$ corresponds to the flux going through the physical interface. From this flux, the velocity $u_{1/2}$ is computed as the flux $U^+_{1/2}$ divided by the interface area, which corresponds to the Jacobian evaluated at $\eta = 0.5$, $\zeta = 0.5$. Then as for the other interfaces, the flux to be interpolated $U_{1/2}$ is the product of $u_{1/2}$ with the Jacobian $J_0^{2D,f}(0.5, \eta, \zeta)$. This distinction results from the fact that the Jacobian varies as a function of the relative coordinates. Volume is not distorted evenly within the cell. For a rectangular parallelepiped, the Jacobian is constant all over the cell and $U_{1/2} = U^+_{1/2}$.

We explained this in the revised manuscript (page 10, line 13).

Page 10:
Line 5: "four grid objects"
Line 13: "which should not be used for C-grids"
Line 16: "... describe the new objects which were added..." Line 27: "... regions, which may overlap or not... "
Line 28: remove "different"
Line 29: "... order in which they ..."
Thank you, done.

Page 11:
Line 3: "... is the velocity given in ..."
Done.

Page 13:
Line 5: "... transported through... "
Line 10-13: "... studies have focused on marine littler in the southern part of the North Sea (Neumann) and have included diffusion and wind drift to their model as well as used a higher resolution."
Done.

Line 23: "NEMO-N006" is not a standard I am familiar with, at least it does not ship with NEMO v3.6. Are you referring to the ORCA0083-N006 simulation, which is similar to ORCA0083-N001 used by Grist et al 2014 and Kjellsson and Zanna 2017?
Line 26: "... at horizontal resolutions of nominally 1/4 and 1/12."
Line 26: What is the vertical coordinate used, i.e. which interpolation scheme is used here?
Yes, we use the ORCA0083-N006 and ORCA025-N006 models. The models use 75 z-layers, with the variables distributed on a C-grid. We have added those informations in the manuscript.

Page 14:
Line 7: Full disclosure: this is the reason why I'm often sceptical about CMEMS data for particle modelling. All data is interpolated from the model grid to some other grid and not necessarily in a conservative way. Could you say a few words here about how this interpolation was done by CMEMS. Would you get identical results if the CMEMS data came on the native C-grid from the NEMO model?
Line 9: "...data, which will be ..."
Line 10: Again, which vertical coordinate is used? Is this also interpolated by CMEMS?
When transposing a field from a C to a A-grid, one should be particularly careful to preserve some properties, such as conservation, consistency and the boundary conditions. But even if those properties hold, information will be lost and one cannot generally obtain the same solution by interpolating the original C field and the A-transformed one, whatever being the transposing method (in CMEMS all the quantities are interpolated on the tracer grid).
So yes, we agree with you that when available, the original grid should always be used for Lagrangian modelling. However, many data are provided as a A-grid, like in CMEMS, and Parcels can also interpolate such datasets. The North West shelf reanalysis data are discretised on a z-level mesh, but we only use the surface surface field from this data set.

Line 28-29: After "(Gutow 2018).": "Here we distinguish two flow types, the first based on NEMO and CMEMS data which has impermeable boundary conditions at the coast, and the second which includes Stokes drift and diffusion thus allowing beaching."
Done

Page 15 Line 5: "...advection." Line 6: "...are run, where if the particle beaches, it stops moving."

Done

Page 16:
Line 5: what is meant by "travelled at least once by a cell"? That the cell has been visited by at least one particle?
You're right, this sentence is not precise. We mean that for each cell, we computed the fraction of particles that have visited it at least once. We have reformulated this sentence in the revised manuscript.

Line 10: remove "different"
Done

Line 15: "... no validation of mesoscale dispersion has been done for those simulations there... ". I am fairly sure Andrew C has done some validation (AMOC strength, AABW volume etc.), but probably not for particle dispersion near the grid scale.
By validation, we meant "validation for MP simulations". Even if the model was validated for climate quantities, this validation does not hold for particle modelling, especially at coastal scales. The sentence was reformulated to avoid any confusion.

Line 20-22: I don't fully understand this sentence. By "differences generated in the first year" you mean "within the first year we see more transport into the Kattegat and Skaggerak leaving fewer particles for transport along the Norwegian coast"? There are some differences along the Norwegian coast and Barents Sea (less deep blue regions for Fig 6c).
The NWS do not cover latitudes further North than 65°N, which are reached after around one year by the particles travelling to the Arctic. So basically, most of the differences between run (a) and (c) result from the first year dynamics, since further North both run do only interpolate NEMO 1/12° data. We explained that better in the revised manuscript (page 17, line 23).

Line 27-28: "... how adding diffusion impacts the face of MP."
Line 28: "reduced by 68%" is in relation to NEMO 1/12?
Yes, we precised it in the manuscript.

Page 18:
I don't understand the caption of the figure. Do you mean "fraction of particles visiting each different region at least once" or "for each grid cell, fraction of particles that have visited that grid cell".?
See previous comment about the same confusion in the main text. The caption was reformulated.

The Parcels v2.0 Lagrangian framework: new field interpolation schemes

Philippe Delandmeter[1] and Erik van Sebille[1]

[1]Utrecht University, Institute for Marine and Atmospheric Research, Princetonplein 5, 3584 CC Utrecht, The Netherlands

Correspondence: Philippe Delandmeter (p.b.delandmeter@uu.nl)

Abstract. With the increasing amount of data produced by numerical ocean models, so increases the need for efficient tools to analyse these data. One of these tools is Lagrangian ocean analysis, where a set of virtual particles are released and their dynamics is integrated in time based on fields defining the ocean state, including the hydrodynamics and biogeochemistry if available. This popular methodology needs to adapt to the large variety of models producing these fields at different formats.

This is precisely the aim of Parcels, a Lagrangian ocean analysis framework designed to combine (1) a wide flexibility to model particles of different natures and (2) an efficient implementation in accordance with modern computing infrastructure. In the new Parcels v2.0, we implement a set of interpolation schemes to read various types of discretised fields, from rectilinear to curvilinear grids in the horizontal direction, from z- to s- levels in the vertical and ~~different variable distributions such as~~ using grid staggering with the Arakawa's A-, B- and C- grids. In particular, we develop a new interpolation scheme for a three-dimensional curvilinear C-grid and analyse its properties.

Parcels v2.0 capabilities, including a suite of meta-field objects, are then illustrated in a brief study of the distribution of floating microplastic in the North West European continental shelf and its sensitivity to ~~different~~ various physical processes.

1 Introduction

Numerical ocean modelling has evolved tremendously in the past decades, producing more accurate results, with finer spatial and time resolutions (Prodhomme et al., 2016). With the accumulation of very large data sets resulting from these simulations, the challenge of ocean analysis has grown. Lagrangian modelling is a powerful tool to analyse flows in ~~different~~ several fields of engineering and physics, including geophysics and oceanography (van Sebille et al., 2018).

While Lagrangian modelling can be used to simulate the flow dynamics itself (e.g. Monaghan, 2005), most of the modelling effort in geophysical fluid dynamics is achieved with an Eulerian approach. Lagrangian methods can, in turn, be used ~~for ocean analysis, i.e. to globally~~ to analyse the ocean ~~dynamics~~ dynamic given the flow field from an Eulerian model. The flow field ~~results from numerical models or from other sources. Such sources are for example processed data from~~ can also be taken from land-based measurement such as high frequency radar (Rubio et al., 2017), and satellite imagery that measure altimetry (Holloway, 1986) or directly the currents using Doppler radar (Ardhuin et al., 2018).

Lagrangian analysis simulates the pathways of virtual particles, that can represent water masses, tracers such as temperature, salinity or nutrients, or particulates like sea grass (e.g. Grech et al., 2016), kelp (e.g. Fraser et al., 2018), coral larvae (e.g.

Thomas et al., 2014), plastics ~~(e.g. Lebreton et al., 2012; Onink et al., 2018)~~(e.g. Lebreton et al., 2012; Onink et al., 2019), fish (e.g Phillips et al., 2018), icebergs (e.g. Marsh et al., 2015), ~~and many more different types~~etc. It is used at a wide range of time and spatial scales as for example the modelling of plastic dispersion, from coastal applications (e.g. Critchell and Lambrechts, 2016) to regional (e.g. Kubota, 1994) or global scales (e.g. Maximenko et al., 2012).

The method consists in advancing, for each particle, the coordinates and other state variables by first interpolating fields of interest, as the velocity or any tracer, at the particle position and integrating in time the ordinary differential equations defining the particle dynamics.

A number of tools are available to track virtual particles, with ~~different~~ diverse characteristics, strengths and limitations including Ariane (Blanke and Raynaud, 1997), TRACMASS (Döös et al., 2017), CMS (Paris et al., 2013) and OpenDrift (Dagestad et al., 2018). An extensive list and description of Lagrangian analysis tools is provided in van Sebille et al. (2018). One of the tools is Parcels.

Parcels ("Probably A Really Computationally Efficient Lagrangian Simulator") is a framework for computing Lagrangian particle trajectories (http://www.oceanparcels.org, Lange and van Sebille, 2017). The main goal of Parcels is to process the continuously increasing amount of data generated by the contemporary and future generations of ocean general circulation models (OGCMs). This requires two important features of the model: (1) not to be dependent on one single format of fields and (2) to be able to scale up efficiently to cope with up to petabytes of external data produced by OGCMs. In Lange and van Sebille (2017), the concept of the model was described and the fundamentals of Parcels v0.9 were stated. Since this version, the model essence has remained the same, but many features were added or improved, leading to the current version 2.0. Among all the developments, our research has mainly focused to develop and implement into Parcels ~~different~~ interpolation schemes to provide the possibility to use a set of fields discretised on various types of grids, from rectilinear to curvilinear in the horizontal direction, with z- or s- levels in the vertical, and using ~~different variable distributions through~~ grid staggering with A-, B- and C- Arakawa staggered grids (Arakawa and Lamb, 1977). In particular an interpolation scheme for curvilinear C-grids, that was not defined in other Lagrangian analysis models, was developed for both z and s-levels.

In this paper, we detail the ~~different~~ interpolation schemes implemented into Parcels, with a special care in the description of the new curvilinear C-grid interpolator. We describe the new meta-field objects available into Parcels for easier and faster simulations. We prove some fundamental properties of the interpolation schemes. ~~Then we~~ We then validate the new developments through a study of the sensitivity of floating microplastic dispersion and 3D passive particles distribution on the North West European continental shelf and discuss ~~those~~ the results.

2 Parcels v2.0 development

To simulate particle transport in a large variety of applications, Parcels relies on two key features: (1) interpolation schemes to read external data sets provided on different formats and (2) customisable kernels to define the particle dynamics.

The interpolation schemes are necessary to obtain the field value at the particle 3D location. They have been vastly improved in this latest version 2.0. Section 2.1 describes the interpolation of fields and Section 2.2.1 their structure in Parcels.

The kernels are already available since the earliest version of Parcels (Lange and van Sebille, 2017). They are the core of the model, integrating particle position and state variables. The built-in kernels and the development achieved are briefly discussed in Section 2.2.2.

2.1 Field Interpolation

External data sets are provided to Parcels as a set of ~~different~~ fields. Each field is discretised on a structured grid that provides the node locations and instants at which the field values are given. It is noteworthy that the ~~different~~ fields in a field set are not necessarily based on the same grid. In the horizontal plane, rectilinear (Fig. 1a) and curvilinear (Fig. 1b) grids are implemented. Three-dimensional data are built as the vertical extrusion of the horizontal grid, either using z-levels (Fig. 1c) or s-levels (Fig. 1d). A three-dimensional mesh is a combination of a rectilinear or a curvilinear in the horizontal direction mesh with either z- or s- levels in the vertical. Another class of meshes are the unstructured grids (e.g. Lambrechts et al., 2008) , which are not yet supported in Parcels.

Fields can be independent from each other (e.g. water velocity from one data set and wind stress from a different data set) and interpolated separately. Often, fields come from a same data set, for example when they result from a numerical model, and form a coherent structure, that must be preserved in Parcels; an example is the zonal and meridional components of the ~~flow~~ velocity field. A coherent field structure is discretised on the same grid, but the ~~different~~ variables are not necessarily distributed evenly, leading to a so-called staggered grid (Arakawa and Lamb, 1977). The ~~different~~ various types of staggered grids have their pros and cons (Cushman-Roisin and Beckers, 2011). Parcels reads the more popular ~~distributions~~ staggered grids occurring in geophysical fluid dynamics: A-, B- and C- grids. While A- and C- are fundamentally different (Fig. 2)~~and the method to interpolate fields on those grids is defined below. The~~ , the B-grid ~~, used by OGCMs such as MOM (Griffies et al., 2004) or POP (Smith et al., 2010),~~ can be considered ~~for Parcels~~ as a hybrid ~~distribution in between A and C~~ configuration in between A- and C- grids. D- and E- grids are less common and are not implemented into the framework so far. ~~Another class of grids are the unstructured meshes (e.g. Lambrechts et al., 2008), which are not yet supported in Parcels~~The methods to interpolate fields on the A-, C- and B- grids are presented in this section.

2.1.1 A-grid

The A-grid is the un-staggered ~~distribution of~~ Arakawa's ~~grids~~grid: zonal velocity (u), meridional velocity (v) and tracers (T) are collocated (Fig. 2a). This grid is used in many reanalysis data sets for global currents (e.g. Globcurrent, Rio et al., 2014) or tidal dynamics (e.g. TPXO, Egbert and Erofeeva, 2002), as well as regridded products such as OFES (Sasaki et al., 2008) or data sets on platforms like the Copernicus Marine Environment Monitoring Service. In this section we consider first the two-dimensional case before describing three-dimensional fields.

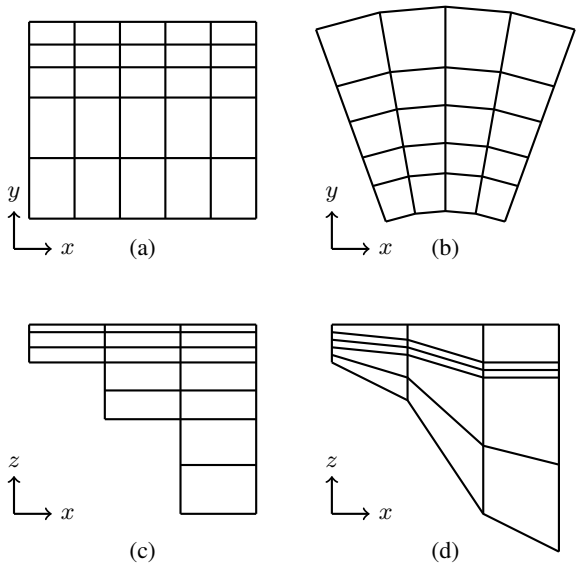

Figure 1. ~~Different grid~~ Grid discretisations processed by Parcels. In the horizontal plane: (a) rectilinear, (b) curvilinear; in the vertical: (c) z-levels and (d) s-levels. Four combinations of horizontal and vertical grids are possible to form a three-dimensional mesh.

2D field

In a two-dimensional context, field f is interpolated in cell (j,i) where the particle is located, with the bi-linear Lagrange polynomials ~~ϕ_k^{2D}~~ ϕ_n^{2D} and the four nodal values ~~F_k with $k=0,...,3$~~ F_n with $n=0,...,3$, surrounding the cell, resulting in the following expression:

$$f(x,y) = \sum_{\substack{k\\ n=0}}^{3} \phi^{2D}_{\substack{k\\ n}}(\xi,\eta)\, F_{\substack{k\\ n}}, \tag{1}$$

$$\text{with } \xi,\eta \text{ s.t. } \begin{cases} x = \sum_n \phi_n^{2D}(\xi,\eta)\, X_n \\ y = \sum_n \phi_n^{2D}(\xi,\eta)\, Y_n. \end{cases} \tag{2}$$

ξ, η are the relative coordinates in the unit cell (Fig. 3b), corresponding to the particle relative position in the physical cell (Fig. 3a). ~~(X_k, Y_k)~~ (X_n, Y_n) are the coordinates of the cell vertices. The two-dimensional Lagrange polynomials ~~ϕ_k^{2D}~~ ϕ_n^{2D} are the bi-linear functions:

$$\phi_0^{2D}(\xi,\eta) = (1-\xi)(1-\eta), \qquad \phi_1^{2D}(\xi,\eta) = \quad \xi \quad (1-\eta),$$
$$\phi_2^{2D}(\xi,\eta) = \quad \xi \quad \eta, \qquad \phi_3^{2D}(\xi,\eta) = (1-\xi) \quad \eta.$$

~~In~~ Note that in a rectilinear mesh, solving Eq. 2 reduces to the usual solutions:

$$\xi = \frac{x - X_0}{X_1 - X_0}, \qquad \eta = \frac{y - Y_0}{Y_3 - Y_0}. \tag{3}$$

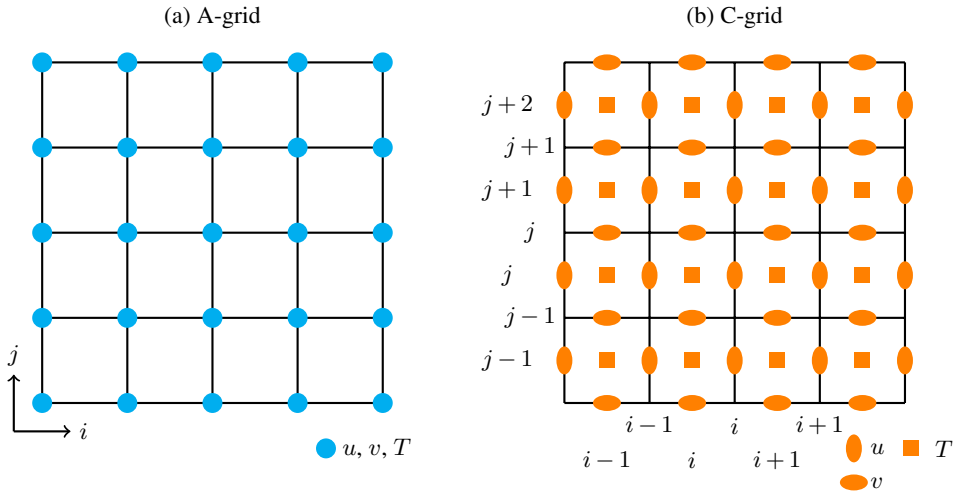

Figure 2. ~~Different variable distributions~~ Arakawa's staggered grids (Arakawa and Lamb, 1977): (a) A-grid and (b) C-grid. In C-grid, i and j represent the variable column and row indexing in arrays where the variables are stored. The indexing of the C-grid follows the NEMO notations (Madec et al., 2016).

3D field

To read three-dimensional fields, for both z- and s- levels, the scheme interpolates the eight nodal values located on the hexahedron vertices using the tri-linear Lagrangian polynomials ~~ϕ_k^{3D}~~ ϕ_w^{3D}:

$$\phi_0^{3D}(\xi,\eta,\zeta) = (1-\xi)(1-\eta)(1-\zeta), \qquad \phi_1^{3D}(\xi,\eta,\zeta) = \quad \xi \ (1-\eta)(1-\zeta),$$

$$\phi_2^{3D}(\xi,\eta,\zeta) = \quad \xi \quad \eta \ (1-\zeta), \qquad \phi_3^{3D}(\xi,\eta,\zeta) = (1-\xi) \quad \eta \ (1-\zeta),$$

$$\phi_4^{3D}(\xi,\eta,\zeta) = (1-\xi)(1-\eta) \quad \zeta, \qquad \phi_5^{3D}(\xi,\eta,\zeta) = \quad \xi \ (1-\eta) \quad \zeta,$$

$$\phi_6^{3D}(\xi,\eta,\zeta) = \quad \xi \quad \eta \quad \zeta, \qquad \phi_7^{3D}(\xi,\eta,\zeta) = (1-\xi) \quad \eta \quad \zeta.$$

ξ and η are obtained as in the 2D case, using the first four vertices of the hexahedron. The vertical relative coordinate is obtained as:

$$\zeta = \frac{z - z_0}{z_1 - z_0}, \tag{4}$$

with

$$z_0 = \sum_{k=0 n=0}^{3} \phi^{2D}{}_{kn} Z_{kn}, \qquad z_1 = \sum_{k=4 n=4}^{7} \phi^{2D}{}_{kn} Z_{kn}. \tag{5}$$

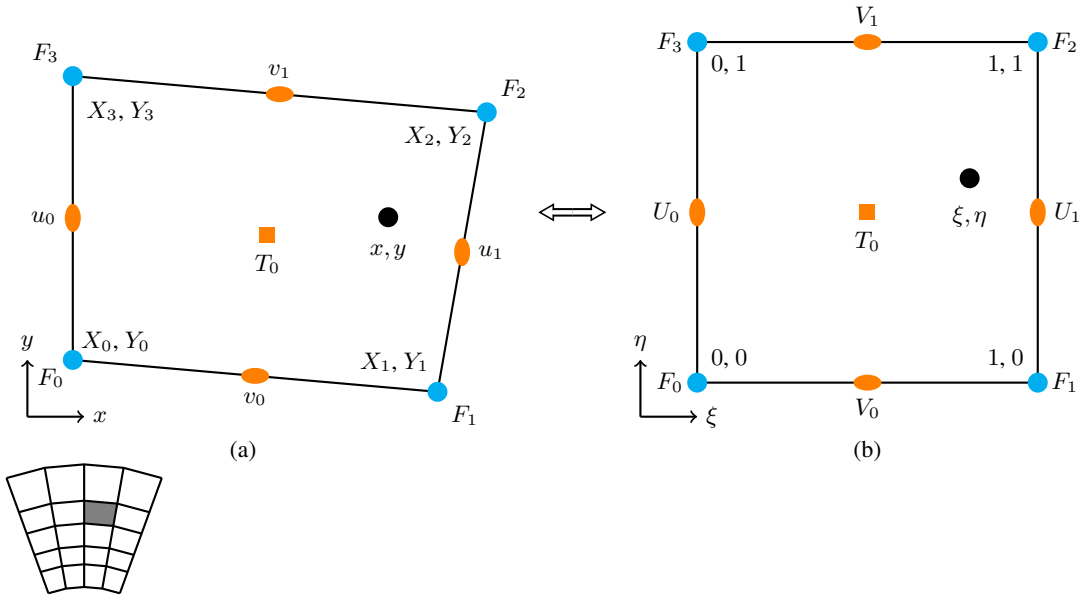

Figure 3. ~~Variable distribution~~ Positioning of the variables for a A-grid (~~cyan~~ blue nodes) and C-grid (orange nodes) cell with (a) physical coordinates in the mesh cell and (b) relative coordinates in the unit cell.

The interpolation results in:

$$f(x,y,z) = \sum_{n=0}^{7} \phi_n^{3D}(\xi,\eta,\zeta)\, F_n. \tag{6}$$

2.1.2 C-grid

In a C-grid discretisation, the velocities are ~~distributed~~ located on the cell edges and the tracers are ~~located~~ at the middle of the cell (Fig. 2b). ~~The interpolation must use local information in the cells. Indeed, interpolating~~ One could attempt to interpolate zonal and meridional velocities as if they would be on two different A-grids, shifted from half a cell, but this would not be in accordance with the C-grid formulation: in particular, such interpolation would break impermeability at the coastal boundary conditions. ~~Here,~~ Instead, we use the interpolation principle ~~is~~ based on the scheme used in the Lagrangian model TRACMASS ~~(Döös et al., 2010; Jönsson et al., 2015; Döös et al., 2017)~~(Jönsson et al., 2015; Döös et al., 2017), but generalised to curvilinear meshes.

The tracer is computed as a constant value all over the cell, in accordance with the mass conservation schemes of C-grids. The formulations for the two-dimensional and three-dimensional velocities consist of four steps:

1. define a mapping between the physical cell and a unit cell (as for the A-grid, Fig. 3);

2. compute the fluxes on the unit cell interfaces, as a function of the velocities on the physical cell interfaces;

3. interpolate those fluxes to obtain the relative velocity;

4. transform the relative velocity to the physical velocity.

Step 1 consists in applying Eq. 2 for the 2D case and Eqs. 2 and 4 for 3D fields. The other three steps are defined below, for both 2D and 3D fields.

2D field

~~For readability, the~~ The velocities at the edges of cell (j, i) ~~cell edges are (~~are $(u_{j+1,i}, u_{j+1,i+1}, v_{j,i+1}, v_{j+1,i+1})$ (Fig. 2b). This indexing was chosen to be consistent with the notation used by the NEMO model (Madec et al., 2016). For readability, they will be renamed in this section to (u_0, u_1, v_0, v_1), using local indices (Fig. 3a) instead of the global indices ~~$(u_{j+1,i}, u_{j+1,i+1}, v_{j,i+1}, v_{j+1,i+1})$ (Fig. 2b). Those velocities are not directly interpolated linearly butlike in finite volume schemesthey are approximated by interpolating linearly~~ . The velocity at (x, y) is not obtained by linear interpolation but, like in finite-volume schemes, it is approximated by linearly interpolating the fluxes (U_0, U_1, V_0, V_1) through the cell edges (Fig. 3b)~~. In a first step, the velocity is derived as a function of the relative velocity following Eq. 2, leading to:~~

$$\begin{cases} u = \dfrac{\partial x}{\partial t} = \sum_k \left(\dfrac{\partial \phi_k^{2D}}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_k^{2D}}{\partial \eta} \dfrac{\partial \eta}{\partial t} \right) X_k = \dfrac{\partial x}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial x}{\partial \eta} \dfrac{\partial \eta}{\partial t}, \\ v = \dfrac{\partial y}{\partial t} = \sum_k \left(\dfrac{\partial \phi_k^{2D}}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_k^{2D}}{\partial \eta} \dfrac{\partial \eta}{\partial t} \right) Y_k = \dfrac{\partial y}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial y}{\partial \eta} \dfrac{\partial \eta}{\partial t}, \end{cases}$$

~~with~~

$$\mathbf{J}^{2D}(\xi, \eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_k \frac{\partial \phi_k^{2D}}{\partial \xi} X_k & \sum_k \frac{\partial \phi_k^{2D}}{\partial \eta} X_k \\ \sum_k \frac{\partial \phi_k^{2D}}{\partial \xi} Y_k & \sum_k \frac{\partial \phi_k^{2D}}{\partial \eta} Y_k \end{bmatrix},$$

, that read (Step 2):

$$\begin{cases} U_0 = L_{03}\, u_0, \\ U_1 = L_{12}\, u_1, \\ V_0 = L_{01}\, v_0, \\ V_1 = L_{23}\, v_1, \end{cases} \tag{7}$$

where L_{03}, L_{12}, L_{01}, L_{23} are the edge lengths. Secondly, $\mathbf{J}^{2D}(\xi, \eta)$~~being~~ , the Jacobian matrix of the transformation from the physical cell to the unit cell (Fig. 3) ~~. The relative velocities are defined as the linear interpolation of the fluxes:~~ is defined:

~~The scaling factor in the relative velocity definition is $J^{2D}(\xi,\eta) = \det\left(\mathbf{J}^{2D}\right)$, the~~

$$\mathbf{J}^{2D}(\xi,\eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_n \frac{\partial \phi_n^{2D}}{\partial \xi} X_n & \sum_n \frac{\partial \phi_n^{2D}}{\partial \eta} X_n \\ \sum_n \frac{\partial \phi_n^{2D}}{\partial \xi} Y_n & \sum_n \frac{\partial \phi_n^{2D}}{\partial \eta} Y_n \end{bmatrix}. \tag{8}$$

The determinant of the Jacobian ~~Matrix~~matrix, that will be called Jacobian is computed: $J^{2D}(\xi,\eta) = \det\left(\mathbf{J}^{2D}\right)$. The Jacobian defines the ratio between an elementary surface in the physical cell and the corresponding surface in the unit cell. The relative velocity in the unit cell is defined as (Step 3):

$$\begin{cases} \frac{\partial \xi}{\partial t} = & \frac{(1-\xi)U_0 + \xi U_1}{J^{2D}(\xi,\eta)}, \\ \frac{\partial \eta}{\partial t} = & \frac{(1-\eta)V_0 + \eta V_1}{J^{2D}(\xi,\eta)}. \end{cases} \tag{9}$$

Finally (Step 4), the velocity is obtained by transforming the relative velocity (Eq. 9) to the physical coordinate system:

$$\begin{cases} u = \frac{\partial x}{\partial t} = \sum_n \left(\frac{\partial \phi_n^{2D}}{\partial \xi} \frac{\partial \xi}{\partial t} + \frac{\partial \phi_n^{2D}}{\partial \eta} \frac{\partial \eta}{\partial t} \right) X_n = \frac{\partial x}{\partial \xi} \frac{\partial \xi}{\partial t} + \frac{\partial x}{\partial \eta} \frac{\partial \eta}{\partial t}, \\ v = \frac{\partial y}{\partial t} = \sum_n \left(\frac{\partial \phi_n^{2D}}{\partial \xi} \frac{\partial \xi}{\partial t} + \frac{\partial \phi_n^{2D}}{\partial \eta} \frac{\partial \eta}{\partial t} \right) Y_n = \frac{\partial y}{\partial \xi} \frac{\partial \xi}{\partial t} + \frac{\partial y}{\partial \eta} \frac{\partial \eta}{\partial t}. \end{cases} \tag{10}$$

~~This interpolation scheme has the advantage of preserving the impermeability boundary conditions by construction. Fig. 2b illustrates the indexing of the velocities on the grid.This indexing was chosen to be consistent with the notation used by the NEMO model (Madec et al., 2016).~~

3D field

The three-dimensional interpolation on C-grids is different for z- and s- levels.

For z-levels, the horizontal and vertical directions are independent. The horizontal velocities in cell (k, j, i) are thus interpolated as in the 2D case, using the data at level k: $(u_{k,j+1,i}, u_{k,j+1,i+1}, v_{k,j,i+1}, v_{k,j+1,i+1})$, and the vertical velocity is interpolated as:

$$w = \zeta w_0 + (1-\zeta) w_1, \tag{11}$$

with ~~$w_0 = w_{i+1,j+1,l}$, and $w_1 = w_{i+1,j+1,l+1}$.~~ $w_0 = w_{k,j+1,i+1}$, and $w_1 = w_{k+1,j+1,i+1}$. The 3D indexing is again consistent with NEMO notation.

For s-levels, the three velocities must be interpolated at once. The three-dimensional interpolation is similar to its two-dimensional version, but it is not the straightforward extension, which would linearly interpolate the fluxes as in Eq. 9 and divide this result by the Jacobian. Indeed, in 2D the ~~Jacobian is a bi-linear function of ξ and η,~~ interpolation scheme is built specifically such that a uniform velocity field is exactly interpolated by Eq. 10 ~~, which is~~ (see demonstration in Section 3.1).

This is made possible since when developing the right hand side of Eq. 10, one obtains a numerator which is a bi-linear function of ξ and η, and a denominator which is the Jacobian, precisely bi-linear in ξ and η. ~~But~~ Doing a similar approach in 3D ~~,~~ would lead to a tri-linear numerator, but the Jacobian J^{3D} is a tri-quadratic function of the coordinates ξ, η and ζ:

$$J^{3D}(\xi,\eta,\zeta) = \det\left(\mathbf{J}^{3D}\right), \quad \text{with } \mathbf{J}^{3D}(\xi,\eta,\zeta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \sum_n \frac{\partial \phi_n^{3D}}{\partial \xi} X_n & \sum_n \frac{\partial \phi_n^{3D}}{\partial \eta} X_n & \sum_n \frac{\partial \phi_n^{3D}}{\partial \zeta} X_n \\ \sum_n \frac{\partial \phi_n^{3D}}{\partial \xi} Y_n & \sum_n \frac{\partial \phi_n^{3D}}{\partial \eta} Y_n & \sum_n \frac{\partial \phi_n^{3D}}{\partial \zeta} Y_n \\ \sum_n \frac{\partial \phi_n^{3D}}{\partial \xi} Z_n & \sum_n \frac{\partial \phi_n^{3D}}{\partial \eta} Z_n & \sum_n \frac{\partial \phi_n^{3D}}{\partial \zeta} Z_n \end{bmatrix}, \tag{12}$$

~~such that a linear interpolation of the fluxes does not resolve exactly a uniform velocity. The velocities~~ The interpolation order must then be increased as well to a quadratic function. Here, as in 2D, the velocities u, v and w are still derived from the coordinate transformation~~:~~ (Step 4):

$$\begin{cases} u = \dfrac{\partial x}{\partial t} = \sum_n \left(\dfrac{\partial \phi_n^{3D}}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_n^{3D}}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial \phi_n^{3D}}{\partial \zeta}\dfrac{\partial \zeta}{\partial t} \right) X_n = \dfrac{\partial x}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial x}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial x}{\partial \zeta}\dfrac{\partial \zeta}{\partial t}, \\ v = \dfrac{\partial y}{\partial t} = \sum_n \left(\dfrac{\partial \phi_n^{3D}}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_n^{3D}}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial \phi_n^{3D}}{\partial \zeta}\dfrac{\partial \zeta}{\partial t} \right) Y_n = \dfrac{\partial y}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial y}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial y}{\partial \zeta}\dfrac{\partial \zeta}{\partial t}, \\ w = \dfrac{\partial z}{\partial t} = \sum_n \left(\dfrac{\partial \phi_n^{3D}}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial \phi_n^{3D}}{\partial \zeta}\dfrac{\partial \zeta}{\partial t} \right) Z_n = \dfrac{\partial z}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial z}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial z}{\partial \zeta}\dfrac{\partial \zeta}{\partial t}, \end{cases} \tag{13}$$

but this time the relative velocities interpolate the fluxes using quadratic Lagrangian functions (Step 3):

$$\begin{cases} \dfrac{\partial \xi}{\partial t} = & \dfrac{\left(2\xi^2 - 3\xi + 1\right) U_0 + \left(-4\xi^2 + 4\xi\right) U_{1/2} + \left(2\xi^2 - \xi\right) U_1}{J^{3D}(\xi,\eta,\zeta)}, \\ \dfrac{\partial \eta}{\partial t} = & \dfrac{\left(2\eta^2 - 3\eta + 1\right) V_0 + \left(-4\eta^2 + 4\eta\right) V_{1/2} + \left(2\eta^2 - \eta\right) V_1}{J^{3D}(\xi,\eta,\zeta)}, \\ \dfrac{\partial \zeta}{\partial t} = & \dfrac{\left(2\zeta^2 - 3\zeta + 1\right) W_0 + \left(-4\zeta^2 + 4\zeta\right) W_{1/2} + \left(2\zeta^2 - \zeta\right) W_1}{J^{3D}(\xi,\eta,\zeta)}. \end{cases} \tag{14}$$

~~The different~~ Step 2 is more complex than in the 2D case. The fluxes interpolated in Eq. 14 are the product of the velocities with the face Jacobian ~~$J^{2D,f}$~~ $J_i^{2D,f}$, as illustrated in Fig. 4.

~~The Jacobian $J_j^{2D,f}$~~ The Jacobian $J_i^{2D,f}$ is defined as follow (Weisstein, 2018):

$$J^{2D,f}{}_{ji}(\xi,\eta,\zeta) = \sqrt{\left(M_{0,j}(\mathbf{J}^{3D})\right)^2 + \left(M_{1,j}(\mathbf{J}^{3D})\right)^2 + \left(M_{2,j}(\mathbf{J}^{3D})\right)^2} \sqrt{\left(M_{0,i}(\mathbf{J}^{3D})\right)^2 + \left(M_{1,i}(\mathbf{J}^{3D})\right)^2 + \left(M_{2,i}(\mathbf{J}^{3D})\right)^2}, \tag{15}$$

with ~~$M_{i,j}(\mathbf{J}^{3D})$, the i,j~~ $M_{j,i}(\mathbf{J}^{3D})$, the j,i minor of $\mathbf{J}^{3D}$, i.e. the determinant of the Jacobian matrix $\mathbf{J}^{3D}$ from which row ~~i and column j were removed~~ j and column i were removed, leading to a 2x2 matrix. Fluxes through a face that is normal to ξ, η or ζ in the unit cell are computed with Jacobian $J_0^{2D,f}$, $J_1^{2D,f}$ or $J_2^{2D,f}$, respectively. Using the node indices defined in Fig. 4b,

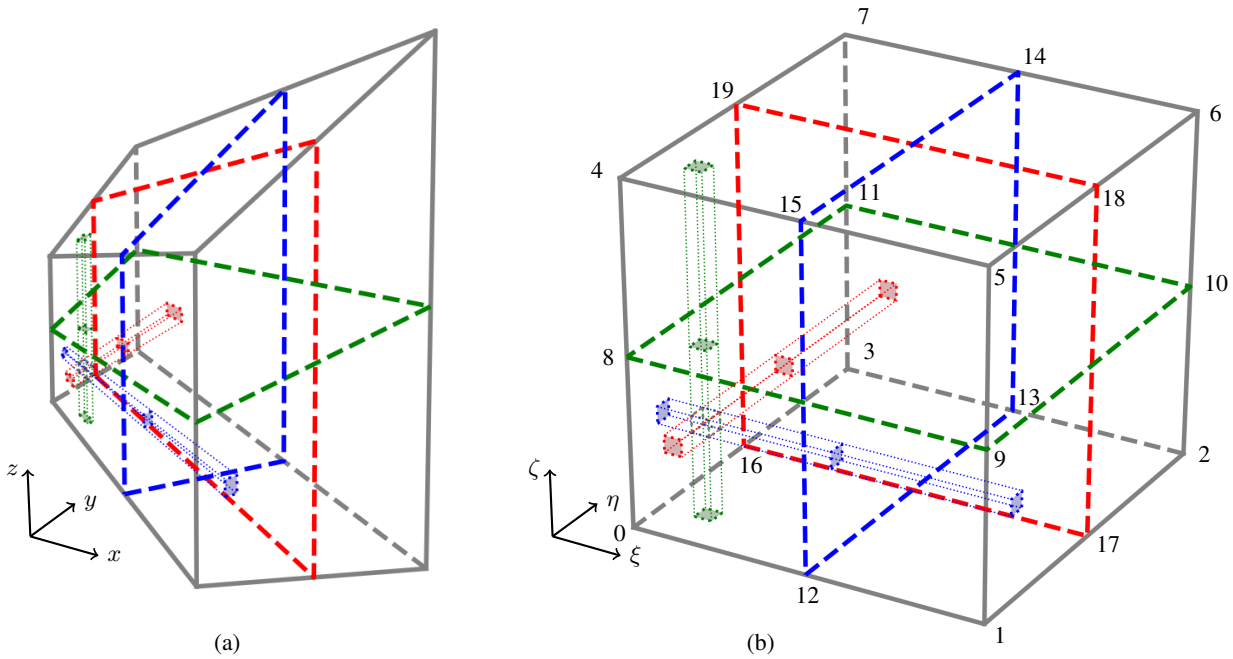

Figure 4. Fluxes used for 3D interpolation on a C-grid for (a) the physical cell and (b) the unit cell. The node indices in panel b are used to define the faces in Eq. 16.

the Jacobians and ~~the different~~ their respective fluxes read then:

face	Jacobian	flux
$[0,3,7,4]$	$J_0^{2D,f}(0,\eta,\zeta)$	$U_0 = u_0\, J_0^{2D,f}(0,\eta,\zeta)$
$[1,2,6,5]$	$J_0^{2D,f}(1,\eta,\zeta)$	$U_1 = u_1\, J_0^{2D,f}(1,\eta,\zeta)$
$[12,13,14,15]$	$J_0^{2D,f}(1/2,\eta,\zeta)$	$U_{1/2} = u_{1/2}\, J_0^{2D,f}(1/2,\eta,\zeta)$
$[0,1,5,4]$	$J_1^{2D,f}(\xi,0,\zeta)$	$V_0 = v_0\, J_1^{2D,f}(\xi,0,\zeta)$
$[3,2,6,7]$	$J_1^{2D,f}(\xi,1,\zeta)$	$V_1 = v_1\, J_1^{2D,f}(\xi,1,\zeta)$
$[16,17,18,19]$	$J_1^{2D,f}(\xi,1/2,\zeta)$	$V_{1/2} = v_{1/2}\, J_1^{2D,f}(\xi,1/2,\zeta)$
$[0,1,2,3]$	$J_2^{2D,f}(\xi,\eta,0)$	$W_0 = w_0\, J_2^{2D,f}(\xi,\eta,0)$
$[4,5,6,7]$	$J_2^{2D,f}(\xi,\eta,1)$	$W_1 = w_1\, J_2^{2D,f}(\xi,\eta,1)$
$[8,9,10,11]$	$J_2^{2D,f}(\xi,\eta,1/2)$	$W_{1/2} = w_{1/2}\, J_2^{2D,f}(\xi,\eta,1/2)$

(16)

It is important to note another difference between the 2D and the 3D approaches here. While in 2D, the fluxes were simply the product of the velocity and the edge length and were independent from ξ and η, this is not the case in 3D anymore. The Jacobian is a function of the relative coordinates. This is because an interface will not be evenly distorted from the physical to

the unit cell. In fact, the 2D is a particular case of the 3D, where the edge length corresponds to the edge Jacobian, which is independent of the relative coordinates.

Finally, the velocities u_0, u_1, v_0, v_1, w_0 and w_1 are provided by the grid, but this is not the case for $u_{1/2}$, $v_{1/2}$, $w_{1/2}$. For ocean applications under the Boussinesq approximation, mass conservation is reduced to volume conservation (Cushman-Roisin and Beckers, 2011). Under this assumption, leading to the continuity equation. Using this equation, the flux through faces $[12,13,14,15]$ (in blue on Fig. 4), $[16,17,18,19]$ (in red) and $[8,9,10,11]$ (in green) can be computed . if the mesh is fixed. Otherwise, for example with vertically adaptive grids, the change in volume of the mesh must be considered, which is currently not implemented in Parcels. Ignoring this term in vertically adaptive grids would lead to errors on the order of 1% (Kjellsson and Zanna, 2017).

We write here the development for face $[12,13,14,15]$. The two other face faces follow the same process. Let consider the hexahedron $[0,12,13,3,4,15,14,7]$, the Jacobians referring to it will be noted J^*. The flux going through $[12,13,14,15]$ reads:

$$U^+_{1/2} = u_0\, J_0^{*2D,f}(0,1/2,1/2) + v_0\, J_1^{*2D,f}(1/2,0,1/2) - v_1\, J_1^{*2D,f}(1/2,1,1/2) + w_0\, J_2^{*2D,f}(1/2,1/2,0) - w_1\, J_2^{*2D,f}(1/2,1/2,1),$$

from which flux $U_{1/2}$ can be computed:

$$U_{1/2} = U^+_{1/2} \frac{J_0^{*2D,f}(1,\eta,\zeta)}{J_0^{*2D,f}(1,1/2,1/2)}. \tag{17}$$

$U^+_{1/2}$ corresponds then to the flux through the physical face $[12,13,14,15]$ and $U_{1/2}$ is the flux that should be interpolated, which results from $U^+_{1/2}$ transformed to $u_{1/2}$, the velocity at the physical face, before computing the flux in the relative cell interface. It is noteworthy that $J_0^{*2D,f}(1,\eta,\zeta) = J_0^{2D,f}(1/2,\eta,\zeta)$.

For compressible flows, fluxes $U^+_{1/2}$, $V^+_{1/2}$, $W^+_{1/2}$ cannot be computed using only half of the hexahedron. But since density is constant throughout the entire element, the actual flux $U^+_{1/2}$ can be computed as the average between the flux going through face $[12,13,14,15]$ using hexahedron $[0,12,13,3,4,15,14,7]$ and the one using hexahedron $[12,1,2,13,15,5,6,14]$.

2.1.3 B-grid

The B-grid is a combination between the A- and the C- grids. It is used by OGCMs such as MOM (Griffies et al., 2004) or POP (Smith et al., 2010).

For two-dimensional fields, the velocity nodes are located on the cell vertices as in an A-grid and the tracer nodes are at the centre of the cells as in a C-grid. The velocity field is thus interpolated exactly as for an A-grid (Eq. 1), and the tracer is like in a C-grid, constant over the cell.

For a three-dimensional cell, the tracer node is still at the centre of the cell and the field is constant; the four horizontal velocity nodes are located at the middle of the four vertical edges; the two vertical velocity nodes are located at the centre of the two horizontal faces (Wubs et al., 2006). The horizontal component of the interpolated velocity in the cell thus only varies

as a function of ξ and η and the vertical component is a function of only ζ:

$$\begin{cases} u(x,y,z) = \sum_{n=0}^{3} \phi_n^{2D}(\xi,\eta)\, u_n, \\ v(x,y,z) = \sum_{n=0}^{3} \phi_n^{2D}(\xi,\eta)\, v_n, \\ w(x,y,z) = (1-\zeta) w_0 + \zeta w_1, \\ t(x,y,z) = t_0, \end{cases} \tag{18}$$

with the local indices (u_0, u_1, u_2, u_3) corresponding to global indices ($u_{k,j,i}$, $u_{k,j,i+1}$, $u_{k,j+1,i+1}$, $u_{k,j+1,i}$) and similarly for the v field. w_0 and w_1 correspond to $w_{k,j+1,i+1}$ and $w_{k+1,j+1,i+1}$ and tracer t_0 to $t_{k,j+1,i+1}$. In Parcels v2.0, this field interpolation is only available for z-levels.

2.2 Implementation into Parcels

2.2.1 Fields

General Structure

Parcels relies on a set of `Field` objects, combined in a `FieldSet`, to interpolate different quantities at the particle location. As explained in Section 2.1, a field is discretised on a grid. In Parcels v2.0, four ~~grids~~ grid objects are defined: `RectilinearZGrid`, `RectilinearSGrid`, `CurvilinearZGrid` and `CurvilinearSGrid`.

The main variables of the grids are the time, depth, latitude and longitude coordinates. Longitude and latitude are defined as vectors for rectilinear grids and 2D arrays for curvilinear ones. The depth variable is defined as a vector for z-level grids. A two-dimensional horizontal grid is simply a `RectilinearZGrid` or a `CurvilinearZGrid` in which the depth variable is empty. For s-levels, the depth is defined as a 3D array or a 4D array if the grid moves vertically in time. The time variable can be empty for steady state fields. Note that models such as Hycom using hybrid grids combining z- and σ-levels are treated by Parcels as general `RectilinearSGrid` or `CurvilinearSGrid` objects, for which the depth of the s-levels is provided.

A `Field` has an `interp_method` attribute, which is set to `linear` for ~~A-grid fields~~ fields discretised on a A-grid, `bgrid_velocity`, `bgrid_w_velocity` and `bgrid_tracer` for horizontal and vertical velocity and tracer fields on a B-grid and `cgrid_velocity` and `cgrid_tracer` for velocity and tracer ~~fields~~ on a C-grid. Note that a `nearest` interpolation method is also available, ~~that~~ which should not be used for ~~C-grids~~B- and C- grids.

Parcels can load field data from various input formats. The most common approach consists in reading netCDF files using the `FieldSet.from_netcdf()` method or one of its derivatives. However, Python objects such as xarray or numpy arrays can also be loaded using `FieldSet.from_xarray_dataset()` or `FieldSet.from_data()`, respectively.

Loading a long time series of data often requires a significant memory allocation, which is not always available on the computer. The previous Parcels version circumvented the problem by loading the data step by step. Using `deferred_load` flag which is set by default, this process is fully automated in v2.0 and allows to use long time series while under the hood the time steps of the data are loaded only when they are strictly necessary for computation.

Meta-field objects

In Parcels, a variety of other objects enable to easily read a field. In this section, we describe the ~~different new objects which were added recently~~ new objects recently added to the framework.

The first object is the `VectorField`, that jointly interpolates the ~~different~~ two or three components of a vector field such as velocity. This object is not only convenient but necessary, since u and v fields are both required to interpolate the zonal and meridional velocity in the C-grid curvilinear discretisation.

Another useful object is the `SummedField`, which is a list of fields that can be summed to create a new field. The ~~different~~ fields of the `SummedField` do not necessarily share the same grid. For example, this object can be used to create a velocity which is the sum of surface water current and Stokes drift. This object has no other purpose than simplifying greatly the kernels defining the particle dynamics.

The fields do not necessarily have to cover the entire region of interest. If a field is interpolated outside its boundary, an `ErrorOutOfBounds` is raised, which leads to particle deletion except if this error is processed through an appropriate kernel. A sequence of various fields covering different regions, ~~that can~~ which may overlap or not, can be interpolated under the hood by Parcels with a `NestedField`. In this case, the ~~different~~ fields composing the `NestedField` must represent the same physical quantity. The `NestedField` fields are ranked to set the priority order ~~with~~ in which they must be interpolated.

~~Loading a long time series of data often requires a significant memory allocation, which is not always available on the computer. The previous Parcels version circumvented the problem by loading the data step by step. This process is fully automated in the current version allowing to use long time series, while under the hood the time steps of the data are loaded only when they are strictly necessary for computation.~~

Available data are not always provided with the expected units. The most frequent example is the ~~flow field~~ velocity given in m s^{-1} while the particle position is in degrees. The same problem occurs with diffusivities in m^2 s^{-1}. `UnitConverter` objects allow to convert automatically the units of the data. The two examples mentioned above are defined into Parcels and other `UnitConverter` objects can be implemented by the user for other transformations.

2.2.2 Kernels

The kernels define the particle dynamics (Lange and van Sebille, 2017). Various built-in kernels are already available in Parcels. `AdvectionRK4`, `AdvectionRK45` and `AdvectionEE` implement the Runge-Kutta 4, Runge-Kutta-Fehlberg and explicit Euler integration schemes for advection. While other explicit discrete time schemes can be defined, analytical integration schemes (Blanke and Raynaud, 1997; Chu and Fan, 2014) are not yet available in Parcels.

`BrownianMotion2D` and `SpatiallyVaryingBrownianMotion2D` implement different types of Brownian motion available as kernels. Custom kernels can be defined by the user for an application-dependent dynamics.

The kernels are implemented in Python but are executed in C for efficiency (Lange and van Sebille, 2017) even if a full Python mode is also available. However, the automated translation of the kernels from Python to C somehow limits the freedom in the syntax of the kernels. For advanced kernels, the possibility to call a user-defined C library is available in version 2.0.

3 Validation

3.1 Uniform velocity on a 2D C-grid

In this section, we prove that the C-grid interpolation preserves exactly a uniform velocity in a quadrilateral. To do so, let us define a uniform velocity $\mathbf{u} = (u, v)$ and a quadrilateral with x coordinates $[X_0, X_1, X_2, X_3]$ and y coordinates $[Y_0, Y_1, Y_2, Y_3]$. On such an element, the velocities u_0, u_1, v_0, v_1 are the scalar product of $\mathbf{u}$ and $\mathbf{n}$, the unit vector normal to the edge, and the fluxes U_0, U_1, V_0, V_1 are the velocities multiplied by the edge lengths, leading to:

$$
\begin{aligned}
U_0 &= u\,(Y_3 - Y_0) - v\,(X_3 - X_0) \\
U_1 &= u\,(Y_2 - Y_1) - v\,(X_2 - X_1) \\
V_0 &= u\,(Y_0 - Y_1) - v\,(X_0 - X_1) \\
V_1 &= u\,(Y_3 - Y_2) - v\,(X_3 - X_2).
\end{aligned}
$$

Therefore, developing Eq. 9 results in:

$$
\begin{aligned}
J^{2D}\frac{\partial \xi}{\partial t} &= \quad u\frac{\partial y}{\partial \eta} - v\frac{\partial x}{\partial \eta} \\
J^{2D}\frac{\partial \eta}{\partial t} &= -u\frac{\partial y}{\partial \xi} + v\frac{\partial x}{\partial \xi},
\end{aligned}
$$

and then applying Eq. 10:

$$
\begin{aligned}
\frac{\partial x}{\partial t} &= u \\
\frac{\partial x}{\partial t} &= v,
\end{aligned}
$$

independently from ξ and η. For the 3D case, the same result is obtained numerically. It can be evaluated using the simple Python C-grid interpolator code available at https://doi.org/10.5281/zenodo.2477418.

3.2 z- and s- level C-grid compatibility

As mentioned above, the horizontal and vertical directions in grids using z-levels are completely decoupled, such that horizontal velocity can be computed as for a 2D field, and vertical interpolation is computed linearly. But a z-level grid is a particular case of an s-level grid. We show that the 3D C-grid interpolator reduces to the simpler z-level C-grid when $Z_0 = Z_1 = Z_2 = Z_3$ and $Z_4 = Z_5 = Z_6 = Z_7$.

First, for z-levels, it is noteworthy that:

$$
\sum_{i=0}^{7} \frac{\partial \phi_i^{3D}}{\partial \xi} X_i = \sum_{i=0}^{3} \frac{\partial \phi_i^{2D}}{\partial \xi} X_i = \frac{\partial x}{\partial \xi},
$$

and similarly for $\frac{\partial x}{\partial \eta}, \frac{\partial y}{\partial \xi}, \frac{\partial y}{\partial \eta}$. $\mathbf{J}^{3D}$ is:

$$\mathbf{J}^{3D} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & 0 \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & 0 \\ 0 & 0 & Z_4 - Z_0 \end{bmatrix},$$

and $J^{3D} = J^{2D}(Z_4 - Z_0)$. All the fluxes through the vertical faces reduce to the product of the velocity, the horizontal edge length and the element height, as for U_0:

$$U_0 = u_0 J_0^{2D,f} = u_0 (Z_4 - Z_0) \sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2} = u_0 (Z_4 - Z_0) \sqrt{(X_3 - X_0)^2 + (Y_3 - Y_0)^2}.$$

The fluxes through the horizontal faces are:

$$W_0 = w_0 J^{2D}, \quad W_1 = w_1 J^{2D}.$$

Therefore, the inner fluxes results in:

$$U_{1/2} = 0.5 (U_0 + U_1), \tag{19}$$

and similarly for $V_{1/2}$ and $W_{1/2}$. The first two lines of Eq. 13 reduce to Eq. 10 and the third line is:

$$\frac{\partial z}{\partial t} = (Z_4 - Z_0) \frac{\partial \zeta}{\partial t}. \tag{20}$$

Finally using Eq. 19, Eq. 14 becomes:

$$\begin{cases} \frac{\partial \xi}{\partial t} = & \frac{(1-\xi) U_0 + \xi U_1}{(Z_4 - Z_0) J^{2D}}, \\ \frac{\partial \eta}{\partial t} = & \frac{(1-\eta) V_0 + \eta V_1}{(Z_4 - Z_0) J^{2D}}, \\ \frac{\partial \zeta}{\partial t} = & \frac{(1-\zeta) W_0 + \zeta W_1}{(Z_4 - Z_0) J^{2D}}, \end{cases}$$

from which first two lines correspond to Eq. 9 and third combined with Eq. 20 reduces to Eq. 11.

4 Simulating the sensitivity of North West European continental shelf floating microplastic distribution

Microplastic (MP) is transported ~~trough~~ through all marine environments and has been observed in large quantities both at coastlines (e.g. Browne et al., 2011) and open seas (e.g. Barnes et al., 2009), at the surface and the sea bed. It represents potential risks to the marine ecosystem that cannot be ignored (Law, 2017). At a global scale, high concentrations are reported in the sub-tropics (Law et al., 2010) but also in the Arctic (Obbard et al., 2014). Various studies have already modelled the accumulation of MP in the Arctic (van Sebille et al., 2012, 2015; Cózar et al., 2017), highlighting the MP transport from the North Atlantic

and the North Sea. Meanwhile, at smaller scales, other studies have focused focused on marine litter in the Southern part of the North Sea ~~(Neumann et al., 2014; Gutow et al., 2018), including~~ (Neumann et al., 2014; Gutow et al., 2018; Stanev et al., 2019) and have included diffusion and wind drift to their model ~~and at~~ as well as used a higher resolution~~than the models predicting accumulation in the Arctic mentioned above~~.

Here we study how the modelled accumulation of floating MP in the Arctic depends on the incorporation of ~~processes and~~ physical processes and model resolutions used for the Southern part of the North Sea. Parcels is used to evaluate the sensitivity of the floating MP distribution under those constrains. To do so, virtual floating MP particles are released off the Rhine and Thames estuaries and tracked for three years. The floating MP distribution is then compared with the trajectories of passive 3D particles, which are not restricted to stay at the sea surface. Note that this section is not meant as a comprehensive study of the MP transport off the North Sea, but rather an application of the new features implemented into Parcels, in both two and three dimensions.

4.1 Input data

We study the influence of the different physical processes impacting surface currents like density- and wind- driven currents, tidal residual currents and Stokes drift, but also the impact of mesh resolution and diffusion. The data come from various data sets (Fig. 5), described in this section. Links to access the data are provided in the code and data availability section.

4.1.1 NEMO

The main data we use ~~is NEMO-N006, which is a standard set-up~~ are ORCA0083-N006 and ORCA025-N006, which are standard sets-up from NEMO (Madec et al., 2016), an ocean circulation model forced by reanalysis and observed flux data: the Drakkar Forcing Set (Dussin et al., 2016). The forcings consist of wind, heat and freshwater fluxes at the surface.

The data are available globally at ~~both a resolution~~ resolutions of 1/4° (ORCA025-N006) and $1/12°$ (ORCA0083-N006). They are discretised on an ORCA grid (Madec and Imbard, 1996), a global ocean tripolar ~~C-grid~~grid, which is curvilinear. The mesh is composed of 75 z-levels and the variables are positioned following a C-grid. The temporal resolution is 5 days.

4.1.2 North West shelf reanalysis

The North West shelf reanalysis (Mahdon et al., 2017) is an ocean circulation flow data set based on the Forecasting Ocean Assimilation Model 7 km Atlantic Margin Model, which is a coupling of NEMO for the ocean with the European Regional Seas Ecosystem Model (Blackford et al., 2004). The reanalysis contains tidal residual currents.

The data are freely available on the Copernicus Marine Environment Monitoring Service (CMEMS). They have a resolution of about 7 km (1/9° lon x 1/15° lat), from (40° N, 20° W) to (65° N, 13° E), with a temporal resolution of 1 day. The data are originally computed on a C-grid, but are re-interpolated on the tracer nodes to form an A-grid ~~dataset~~ data set with z-levels, which is available on CMEMS~~, with a temporal resolution of 1 day~~.

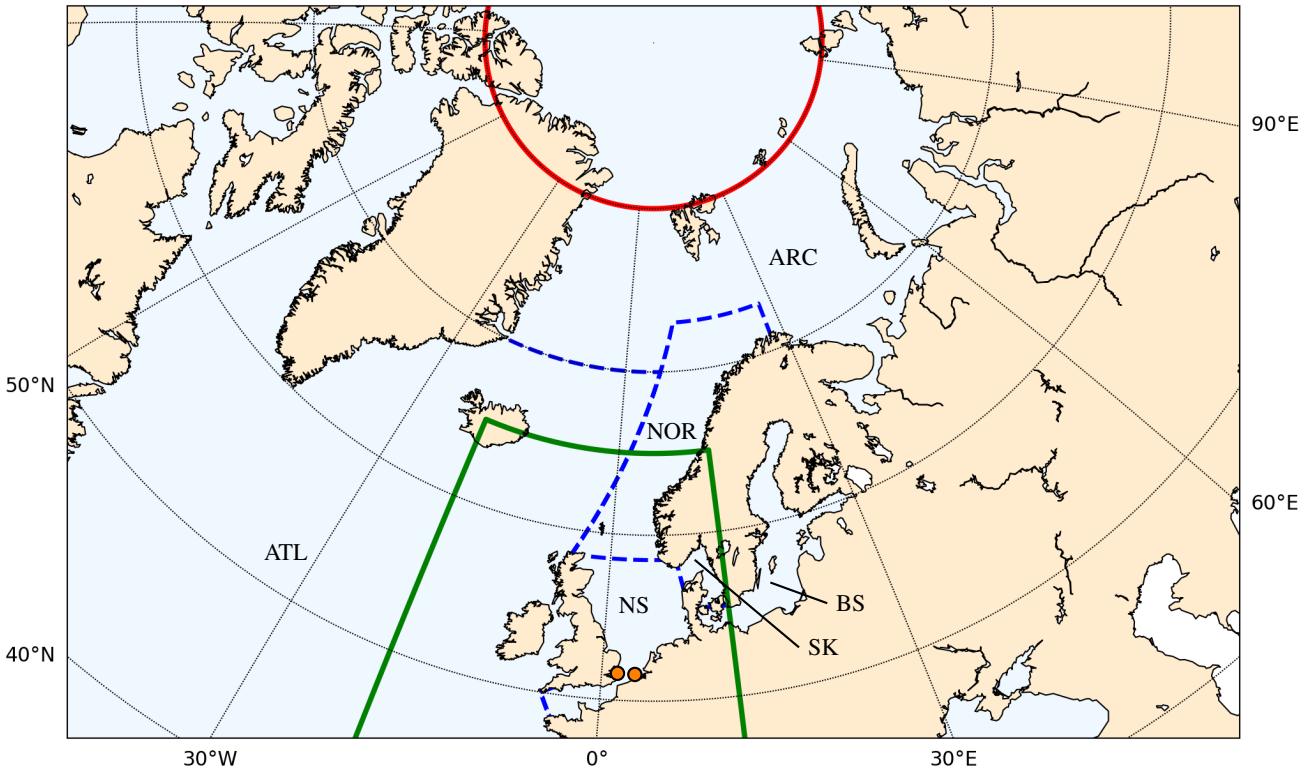

Figure 5. Spatial coverage of the OGCM data used to study North Sea ~~MP~~ microplastic transport. NEMO data are available globally. ~~CMEMS~~ NWS data are available for the North Sea region (green boundaries) and WaveWatch III data are available South of 80° N (red boundary). The particle releasing locations are the Rhine and Thames estuaries (orange dots) and the region is split into ~~different~~ six zones: North Sea (NS), Skagerrak and Kattegat (SK), Baltic Sea (BS), Norwegian Coast (NOR), Arctic Ocean (ARC) and Atlantic Ocean (ATL), that are separated by the dashed blue lines.

The data, ~~that~~ which will be referred to as ~~CMEMS~~NWS, do not cover the entire modelling region, such that a `NestedField` is used to interpolate it within the available region (green zone in Fig. 5), and use the NEMO data for particles outside that region.

4.1.3 WaveWatch III

Stokes drift, i.e. the surface residual current due to waves, was obtained from WaveWatch III (Tolman et al., 2009), that was run using wind forcings from the NCEP Climate Forecast System Reanalysis (CFSR, Saha et al., 2010). The data have a spatial resolution of 1/2°, extending until 80° N, and a temporal resolution of 3 hours. ~~They are freely available on the Ifremer ftp.~~

4.2 Simulations

~~Five~~ Six simulations are run in the following configurations: (a) NEMO hydrodynamics at a 1/12° resolution, (b) NEMO at a 1/4° resolution, (c) ~~CMEMS~~ NWS hydrodynamics in the North Sea nested into NEMO 1/12°, (d) NEMO 1/12° coupled with WaveWatch III Stokes drift~~and finally~~ , (e) NEMO 1/12° with diffusion and finally (f) NEMO hydrodynamics at a 1/12° resolution in which the particles are not constrained to the sea surface.

Every day of year 2000, 100 particles are released in the mouth of the Thames estuary and 100 more particles in mouth of the Rhine, before being tracked for three years. For the two 2D runs, (a) to (e), the particles are released at the sea surface and follow the horizontal surface currents. For the 3D run (f), the particles are released at 0.1, 0.5 and 1 m depths and follow the 3D NEMO flow field.

The diffusion, which parametrises the unresolved processes, is modelled as a stochastic zeroth-order Markov model (van Sebille et al., 2018). The diffusion parameter is proportional to the mesh size, exactly as in the study of North Sea marine litter by Neumann et al. (2014):

$$D = D_0 \, (l/l_0)^{4/3}, \tag{21}$$

with D_0 = 1 m^2 s^{-1} the reference diffusivity, l the square root of the mesh size and l_0 = 1 km. This formulation leads to a same order of magnitude diffusivity as the constant value of Gutow et al. (2018).

The beaching of MP is non-negligible in the North Sea (Gutow et al., 2018) even if it is still poorly understood and often ignored (Neumann et al., 2014) or resulting from the low-resolution current and wind conditions (Gutow et al., 2018). Here ~~,~~ we distinguish two ~~types of dynamics. The first one, that includes NEMO and CMEMS data , is the general ocean circulation. It~~ flow types, the first based on NEMO and NWS data which has impermeable boundary conditions at the coast~~. The second~~ , and the second which includes Stokes drift and diffusion~~. It is a process~~ , thus allowing beaching.

For numerical reasons, due to the integration time step of 15 minutes and the Runge-Kutta 4 scheme, it is theoretically possible that particles beach even with NEMO or ~~CMEMS~~ NWS data. This could happen for example in a region of coastal downwelling, since the ~~MP~~ particles are forced to stay at the surface and could be constantly transported towards the coast. The particle dynamics is thus implemented using separate kernels. At each time step, the particle position is first updated following NEMO or ~~CMEMS advectionis first run~~NWS advection. Then the particle is checked to still be located in a wet cell, otherwise it is pushed back to the sea using an artificial current. In a second step, the Stokes or diffusion kernels are run~~. In this step,~~ , where if the particle beaches, it stops moving. In a final step, the particle age is updated. The kernel code as well as all the scripts running and post-processing the simulations are available at https://doi.org/10.5281/zenodo.2620215.

To compare the simulations, the Parcels raw results, consisting ~~at~~ of particle position, age and beaching status exported every two days, are post-processed into the following maps and budgets.

The ~~MP~~ particle density (Fig. 6) is computed as the number of particles per square kilometre, averaged over the third year of particle age. Note that the absolute value of the concentration is not particularly meaningful, since it is simply proportional to the number of particles released.

To analyse the ~~path of the MP~~particle path, the ocean is discretised into cells of 1/4° longitude x 1/8° ~~resolution and~~ latitude resolution, and for each cell the fraction of particles ~~having travelled~~ that has visited at least once ~~by a~~ the cell is computed (Fig. 7).

~~Finally, to~~ To study the temporal dynamics of the particles, the region is divided into ~~five~~ six zones (Fig. 5): North Sea, Skagerrak and Kattegat, Baltic Sea, Norwegian Coast, Arctic Ocean and Atlantic Ocean, and the evolution of the distribution of ~~MP~~ the particles in those zones is computed (Fig. 8). The time axis represent the particle age in years.

Finally, the integrated vertical distribution (Fig. 9) of the particles as a function of the latitude is computed for the 3D run. For this profile, the domain is divided in bins of 0.5° latitude by 5 m depth and every two days, each particle is mapped to the cell it belongs to, leading to the integrated vertical distribution. Note the linear scale for the upper 50 m depth, and the logarithmic scale for the full vertical profile.

Animations showing the particle dynamics are available in the article supplementary materials.

4.3 Results

The results show various minor and major differences between the ~~different~~ scenarios.

While NEMO 1/12° and NEMO 1/4° show similar dynamics for the first year (Fig. 8), the Norwegian fjords have a higher trapping role in the 1/4° resolution run, even if ~~this~~ the plastic does not beach in both runs. This ~~leads to a reduced amount of plastic~~ increased particle trapping could be a consequence of the data lower resolution that results in a reduced horizontal velocity shear, while a strong shear layer behaves as a barrier isolating the coast from the open waters (Delandmeter et al., 2017) . This is also observed in the supplementary materials animations, in which the main particle path is observed further from the coast for the NEMO 1/12° run. As a consequence of the trapping, the amount or MP reaching the Arctic ~~with~~ is reduced in NEMO 1/4°. This run also produces significantly lower densities North of 80° N. Although none of the runs do resolve coastal dynamics, because of the low temporal and space resolution and the lack of tides, they show important differences. Since no validation was achieved for those ~~simulation~~MP simulations, there is no reason to argue that the 1/12° resolution is high enough to simulate MP dynamics, but this resolution is similar to other studies of plastic litter in the region (Gutow et al., 2018).

The main ~~difference of using CMEMS is the trapping of 6% and 8% of the particles in the Skaggerak and Kattegat and the North Sea zones, respectively, with a decrease of 11% and 19% in the Norwegian coast and Arctic Ocean at the end of the simulation. The differences of the CMEMS run are mainly generated~~ differences of using NWS result from the dynamics during the first year, when the particles are located South of 65° N~~. This explains~~ , explaining the lack of differences in the ~~northern~~ Arctic region in Fig. 6a and Fig. 6c and Fig. 7a and Fig. 7c.

~~As expected, including Stokes drift leads to higher concentrations all along the coast~~ The residence time in the North Sea is increased relatively to NEMO 1/12°, and different peak events occur in the Skagerrak and Kattegat, resulting in final concentrations of 68% in the Arctic and 27% in the Norwegian coast, respectively 7% higher and 8% lower than with NEMO 1/12° (Fig. ~~6d)~~8).

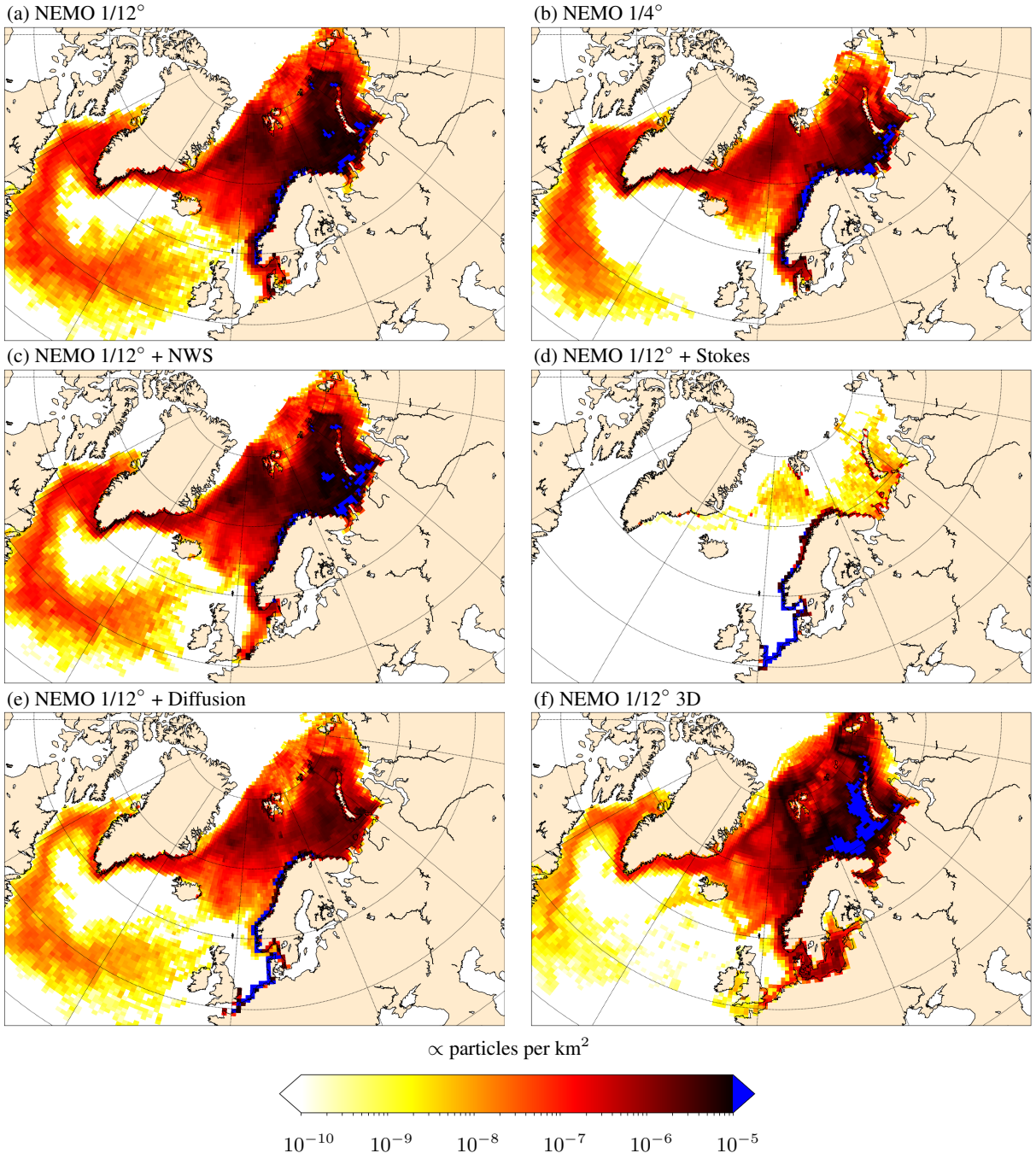

Figure 6. ~~Floating MP density~~Density of floating microplastic (a-e) and passive 3D particles (f), averaged over the third year of particle age, for the different simulation scenarios: (a) NEMO 1/12°, (b) NEMO 1/4°, (c) ~~CMEMS~~ NWS nested into NEMO 1/12°, (d) NEMO 1/12° coupled with Stokes drift from WaveWatch III, (e) NEMO 1/12° coupled with diffusion, (f) NEMO 1/12° 3D. Note the logarithmic scale.

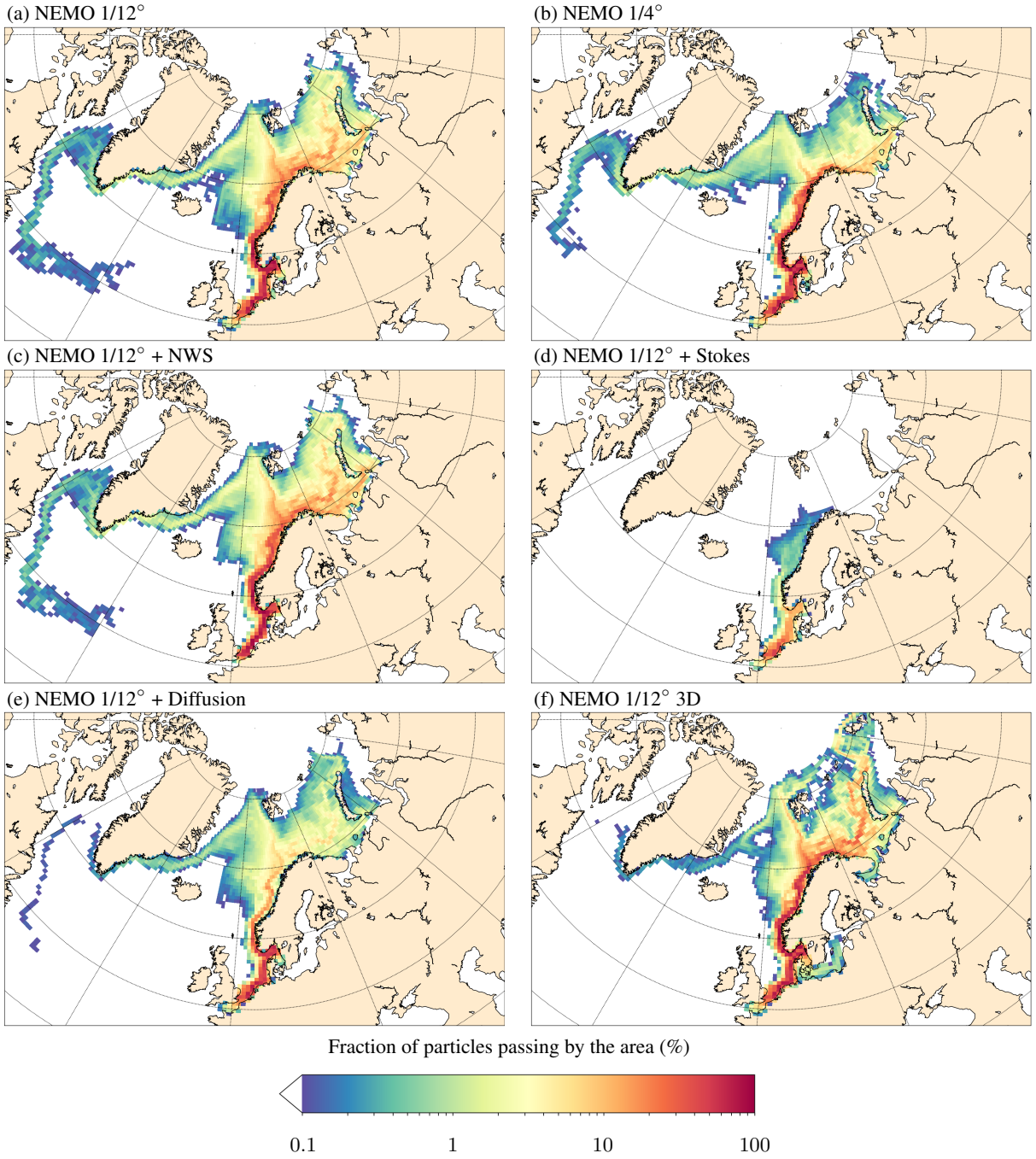

Figure 7. Fraction of floating ~~MP~~ microplastic (a-e) and passive 3D particles (f) originally released in the Thames and Rhine estuaries reaching the domain. It is computed for each cell of a 1/4° longitude x 1/8° latitude grid, as the proportion of the particles that have visited at least once the ~~different region cells~~cell. Note the logarithmic scale.

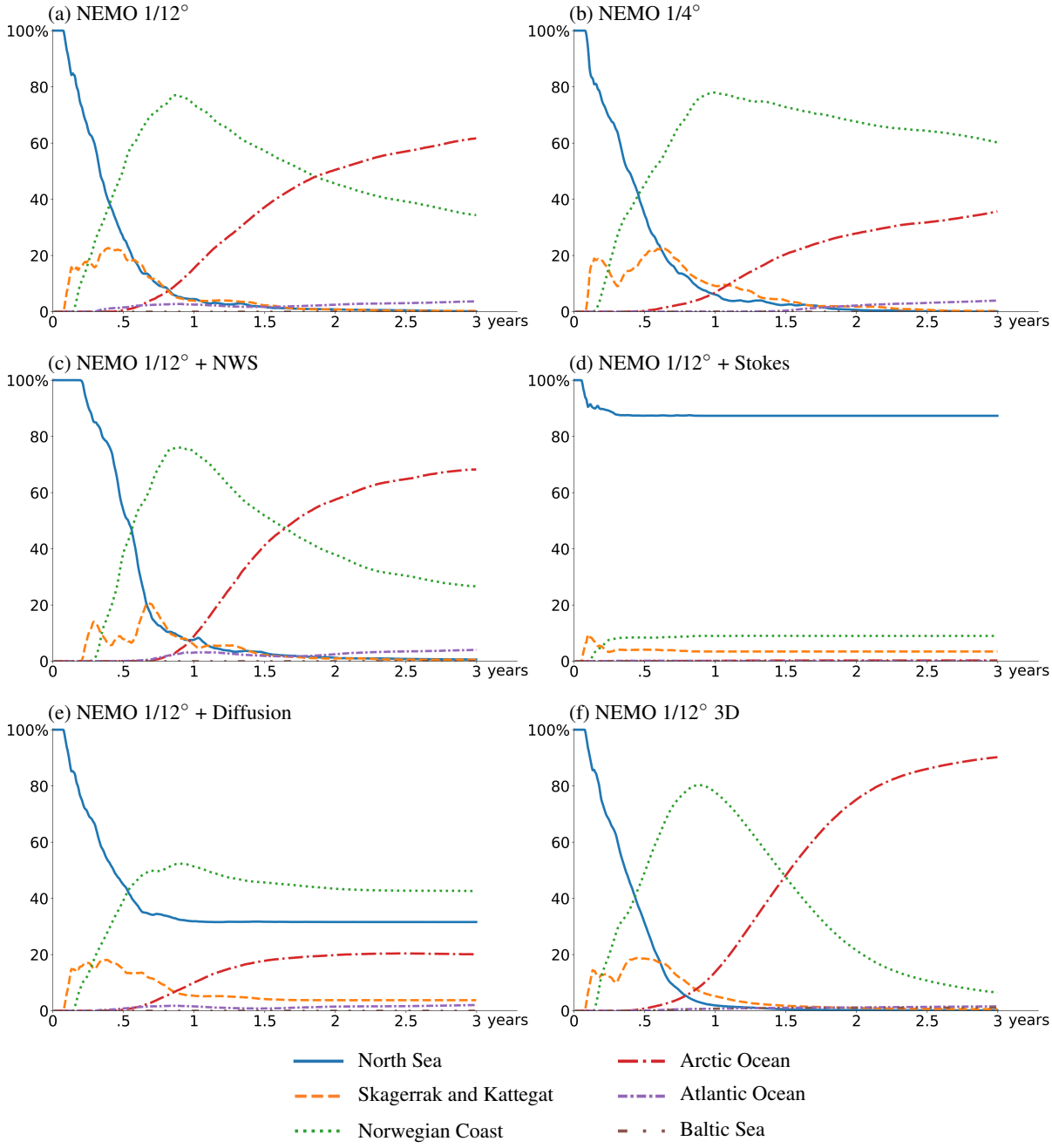

Figure 8. Evolution of the distribution of floating ~~MP~~ microplastic (a-e) and passive 3D particles (f) in the ~~different~~ six zones (Fig. 5) as a function of particle age.

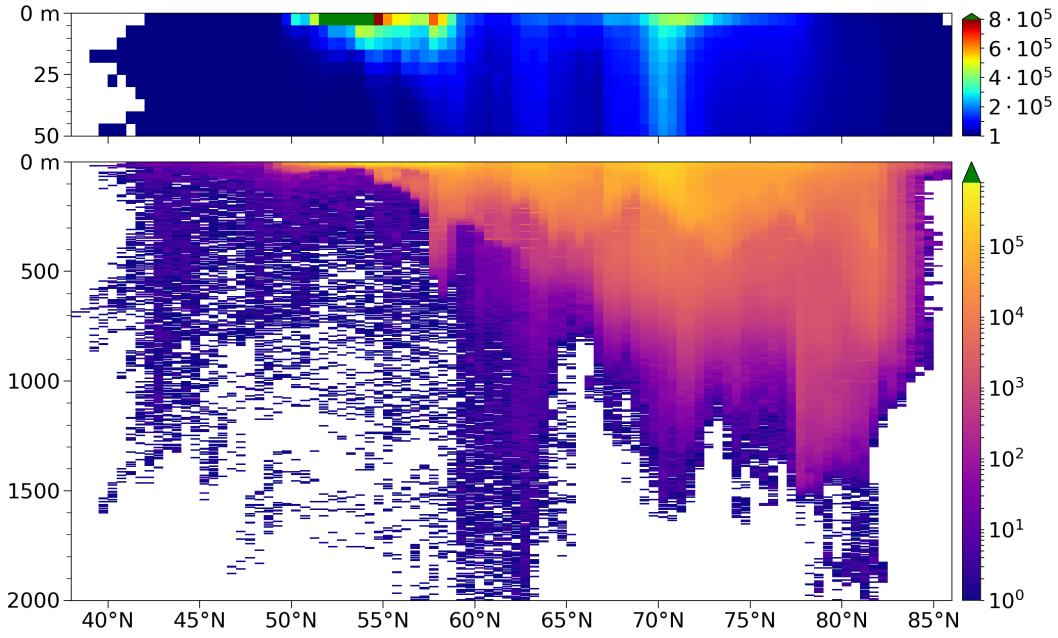

Figure 9. Particle integrated vertical distribution for the 3D NEMO 1/12° simulation. Note the linear scale for the upper 50 m depth, and the logarithmic scale for the full profile.

Including Stokes drift has a major impact on MP dynamics in the North West European continental shelf, due to prevailing westerly winds (Gutow et al., 2018), with ~~46~~close to 90% of the ~~particles~~ plastic staying in the ~~Norwegian zone (Fig. 8d). After 3 years, 45% of the particles have beached in all zones combined. As shown in Fraser et al. (2018), Stokes drift cannot be neglected, especially while considering coastal regions and beaching processes.~~ North Sea, 9% beaching in the Norwegian coast and less than 0.25% reaching the Arctic. Particles are beaching very quickly, with 90% in less than 3.5 months, and 99% within 10 months. While those numbers are not validated here, we can still point out that even if Stokes drift has an important contribution on surface dynamics for large scales (Onink et al., 2019), using it on smaller scales needs a proper validation. Especially the boundary condition should be treated with care. It has a large impact in this application where the particles are released next to the coast.

The parametrisation of sub-grid scales and diffusion is still an important field of research in the Lagrangian community, but it is generally agreed that it cannot be neglected. In this application, we observe how ~~, as expected, adding diffusion which allows beaching has a large impact on the MP fate~~ adding diffusion impacts the fate of MP. The amount of MP reaching the Arctic is reduced by 68% compared to NEMO 1/12°, with large accumulation in the North Sea and Norwegian coast, but not in the Skagerrak and Kattegat. Overall, the proportion of beached particles increases linearly to 73% during the first year, before slowly reaching 83% during the next two years.

Maintaining the MP at the surface is a strong assumption: biofouling, degradation and hydrodynamics affect the plastic depth, which impacts its lateral displacement. In the 3D particle run (Fig. 6f), we do not take all the processes driving MP vertical dynamics, whose parametrisations are currently being developed by the community, but simply model the path of passive particles following the three-dimensional current. Passive particles do not accumulate along the coast as the floating MP (Fig. 6f). This results in an increased transport towards the Arctic (Figs. 7f and 8f). There are overall higher concentrations in the Arctic, including North of 80°N and in the Eastern part of the Kara Sea. Around 1% of the passive particles end up in the Baltic Sea, crossing the Skagerrak and Kattegat using favourable deeper currents (Gustafsson, 1997), while such transport was negligible for floating MP plastic. The vertical distribution of the particles is also analysed (Fig. 9): the majority of the particle, originally released in shallow waters, is still observed close to the surface: 21%, 63% and 97% of the particle records are above 10 m, 100 m and 500 m depths, respectively. The concentration vertical gradient decreases while the latitude increases, with almost no gradient in the Polar region. Passive particles have a significantly different path then surface ones, highlighting the importance of understanding better the vertical dynamics of the plastic to improve the accuracy of its distribution modelling.

This brief study of the sensitivity of North Sea floating MP distribution is an illustration of how Parcels is used to gather and compare flow fields from a multitude of ~~datasets~~data sets in both two and three dimensions, which was made possible by the development of the different field interpolation schemes and ~~constructions such as `NestedField` and `SummedField`~~ meta-field objects. To validate the MP dynamics observed, it is essential to couple such numerical study with an extensive field study.

5 Conclusions

Parcels, a Lagrangian ocean analysis framework, was considerably improved since version 0.9, allowing to read data from multiple fields discretised on different grids and grid types. In particular, a new interpolation scheme for curvilinear C-grids was developed and implemented into Parcels v2.0. This article described this new interpolation as well as the other schemes available in Parcels, including A-, B- and C- ~~variable distributions~~staggered grids, rectilinear and curvilinear horizontal ~~grids~~ meshes and z- and s- vertical levels. ~~Different~~ Numerous features were implemented, including meta field objects, which were described here.

Parcels v2.0 was used to simulate the dynamics of the North West European continental shelf floating microplastic, virtually released during one year off the Thames and Rhine estuaries, before drifting towards the Arctic, and the sensitivity of this transport to various physical processes and numerical choices such as mesh resolution and diffusion parametrisation. While those simulations do not provide a comprehensive study of microplastic dynamics in the area, they highlight key points to consider and illustrate the interest of using Parcels for such modelling.

The next step in Parcels development will involve increasing the model efficiency and developing a fully parallel version of the Lagrangian framework.

–

–

–

–

–

–

Code and data availability.

- Parcels code: The code for Parcels is licensed under the MIT licence and is available through GitHub at https://www.github.com/OceanParcels/parcels. The version 2.0 described here is archived at Zenodo at https://doi.org/10.5281/zenodo.2583739. More information is available on the project webpage at http://www.oceanparcels.org.
- Interpolation code: Independently from Parcels, a simple Python code is also implementing all the C-grid interpolation schemes developed in this paper. It is available at https://doi.org/10.5281/zenodo.2477418.
- North Sea floating MP simulations: All the scripts running and post-processing the North Sea MP simulations are available at https://doi.org/10.5281/zenodo.2620215.
- NEMO data: The NEMO N006 data are kindly provided by Andrew Coward at NOC Southampton, UK.
- North West shelf reanalysis data are provided by the Copernicus Marine Environment Monitoring Service (CMEMS). They can be downloaded at http://marine.copernicus.eu/services-portfolio/access-to-products/?option=com_csw&view=details&product_id=NORTHWESTSHELF_REANALYSIS_PHY_004_009.
- WaveWatch III data come from the Ifremer Institute, France. They can be downloaded at ftp://ftp.ifremer.fr/ifremer/ww3/HINDCAST/GLOBAL/.

Author contributions. Philippe Delandmeter and Erik van Sebille developed the code and wrote the paper jointly.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. Philippe Delandmeter and Erik van Sebille are supported through funding from the European Research Council (ERC) under the European Union Horizon 2020 research and innovation programme (grant ~~agreement no~~agreements nos. 715386 and 821926). The North Sea microplastic simulations were carried out on the Dutch national e-infrastructure with the support of SURF Cooperative (project no. 16371). This study has been conducted using E.U. Copernicus Marine Service Information. We thank Henk Dijkstra for the fruitful discussions and Andrew Coward for providing the ~~NEMO N006~~ ORCA0083-N006 and ORCA025-N006 simulation data.

References

Arakawa, A. and Lamb, V. R.: Computational design of the basic dynamical processes of the UCLA general circulation model, General Circulation Models of the Atmosphere, 17, 173–265, 1977.

Ardhuin, F., Aksenov, Y., Benetazzo, A., Bertino, L., Brandt, P., Caubet, E., Chapron, B., Collard, F., Cravatte, S., Delouis, J.-M., et al.: Measuring currents, ice drift, and waves from space: the Sea surface KInematics Multiscale monitoring (SKIM) concept, Ocean Science, 14, 337–354, 2018.

Barnes, D. K., Galgani, F., Thompson, R. C., and Barlaz, M.: Accumulation and fragmentation of plastic debris in global environments, Philosophical Transactions of the Royal Society B: Biological Sciences, 364, 1985–1998, 2009.

Blackford, J., Allen, J., and Gilbert, F. J.: Ecosystem dynamics at six contrasting sites: a generic modelling study, Journal of Marine Systems, 52, 191–215, 2004.

Blanke, B. and Raynaud, S.: Kinematics of the Pacific equatorial undercurrent: An Eulerian and Lagrangian approach from GCM results, Journal of Physical Oceanography, 27, 1038–1053, 1997.

Browne, M. A., Crump, P., Niven, S. J., Teuten, E., Tonkin, A., Galloway, T., and Thompson, R.: Accumulation of microplastic on shorelines woldwide: sources and sinks, Environmental science & technology, 45, 9175–9179, 2011.

Chu, P. C. and Fan, C.: Accuracy progressive calculation of Lagrangian trajectories from a gridded velocity field, Journal of Atmospheric and Oceanic Technology, 31, 1615–1627, 2014.

Cózar, A., Martí, E., Duarte, C. M., García-de Lomas, J., Van Sebille, E., Ballatore, T. J., Eguíluz, V. M., González-Gordillo, J. I., Pedrotti, M. L., Echevarría, F., et al.: The Arctic Ocean as a dead end for floating plastics in the North Atlantic branch of the Thermohaline Circulation, Science advances, 3, e1600 582, 2017.

Critchell, K. and Lambrechts, J.: Modelling accumulation of marine plastics in the coastal zone; what are the dominant physical processes?, Estuarine, Coastal and Shelf Science, 171, 111–122, 2016.

Cushman-Roisin, B. and Beckers, J.-M.: Introduction to geophysical fluid dynamics: physical and numerical aspects, vol. 101, Academic press, 2011.

Dagestad, K.-F., Röhrs, J., Breivik, Ø., and Ådlandsvik, B.: OpenDrift v1.0: a generic framework for trajectory modelling, Geoscientific Model Development, 11, 1405–1420, 2018.

Delandmeter, P., Lambrechts, J., Marmorino, G. O., Legat, V., Wolanski, E., Remacle, J.-F., Chen, W., and Deleersnijder, E.: Submesoscale tidal eddies in the wake of coral islands and reefs: satellite data and numerical modelling, Ocean Dynamics, 67, 897–913, 2017.

Döös, K., Jönsson, B., and Kjellsson, J.: TRACMASS lagrangian trajectory code documentation and manual, Stockholm University, 2010.

Döös, K., Jönsson, B., and Kjellsson, J.: Evaluation of oceanic and atmospheric trajectory schemes in the TRACMASS trajectory model v6. 0, Geoscientific Model Development, 10, 1733–1749, 2017.

Dussin, R., Barnier, B., Brodeau, L., and Molines, J.-M.: The making of the Drakkar Forcing Set DFS5, Tech. rep., DRAKKAR/MyOcean, 2016.

Egbert, G. D. and Erofeeva, S. Y.: Efficient inverse modeling of barotropic ocean tides, Journal of Atmospheric and Oceanic Technology, 19, 183–204, 2002.

Fraser, C. I., Morrison, A. K., Hogg, A. M., Macaya, E. C., van Sebille, E., Ryan, P. G., Padovan, A., Jack, C., Valdivia, N., and Waters, J. M.: Antarctica's ecological isolation will be broken by storm-driven dispersal and warming, Nature Climate Change, 8, 704, 2018.

Grech, A., Wolter, J., Coles, R., McKenzie, L., Rasheed, M., Thomas, C., Waycott, M., and Hanert, E.: Spatial patterns of seagrass dispersal and settlement, Diversity and Distributions, 22, 1150–1162, 2016.

Griffies, S., Harrison, M., Pacanowski, R., and Rosati, A.: A technical guide to mom4 gfdl ocean group technical report no. 5, NOAA, Geophysical Fluid Dynamics Laboratory, 339pp, 2004.

Gustafsson, B.: Interaction between Baltic Sea and North Sea, Deutsche Hydrografische Zeitschrift, 49, 165–183, 1997.

Gutow, L., Ricker, M., Holstein, J. M., Dannheim, J., Stanev, E. V., and Wolff, J.-O.: Distribution and trajectories of floating and benthic marine macrolitter in the south-eastern North Sea, Marine pollution bulletin, 131, 763–772, 2018.

Holloway, G.: Estimation of oceanic eddy transports from satellite altimetry, Nature, 323, 243, 1986.

Jönsson, B., Döös, K., and Kjellsson, J.: TRACMASS: Lagrangian trajectory code, https://doi.org/10.5281/zenodo.34157, https://doi.org/10.5281/zenodo.34157, 2015.

Kjellsson, J. and Zanna, L.: The impact of horizontal resolution on energy transfers in global ocean models, Fluids, 2, 45, 2017.

Kubota, M.: A mechanism for the accumulation of floating marine debris north of Hawaii, Journal of Physical Oceanography, 24, 1059–1064, 1994.

Lambrechts, J., Comblen, R., Legat, V., Geuzaine, C., and Remacle, J.-F.: Multiscale mesh generation on the sphere, Ocean Dynamics, 58, 461–473, 2008.

Lange, M. and van Sebille, E.: Parcels v0.9: prototyping a Lagrangian ocean analysis framework for the petascale age, Geoscientific Model Development, 10, 4175–4186, https://doi.org/10.5194/gmd-10-4175-2017, https://www.geosci-model-dev.net/10/4175/2017/, 2017.

Law, K. L.: Plastics in the marine environment, Annual review of marine science, 9, 205–229, 2017.

Law, K. L., Morét-Ferguson, S., Maximenko, N. A., Proskurowski, G., Peacock, E. E., Hafner, J., and Reddy, C. M.: Plastic accumulation in the North Atlantic subtropical gyre, Science, 329, 1185–1188, 2010.

Lebreton, L.-M., Greer, S., and Borrero, J. C.: Numerical modelling of floating debris in the world's oceans, Marine Pollution Bulletin, 64, 653–661, 2012.

Madec, G. and Imbard, M.: A global ocean mesh to overcome the North Pole singularity, Climate Dynamics, 12, 381–388, 1996.

Madec, G. et al.: NEMO ocean engine, Tech. rep., Note du Pôle modélisation, Inst. Pierre Simon Laplace, 2016.

Mahdon, R., McConnell, N., and Tonani, M.: Product user manual for North-West Shelf Physical Reanalysis Products NORTHWESTSHELF_REANALYSIS_PHYS_004_009 and NORTHWESTSHELF_REANALYSIS_BIO_004_011, http://cmems-resources.cls.fr/documents/PUM/CMEMS-NWS-PUM-004-009-011.pdf, 2017.

Marsh, R., Ivchenko, V., Skliris, N., Alderson, S., Bigg, G. R., Madec, G., Blaker, A. T., Aksenov, Y., Sinha, B., Coward, A. C., et al.: NEMO-ICB (v1. 0): interactive icebergs in the NEMO ocean model globally configured at eddy-permitting resolution, Geoscientific Model Development, 8, 1547–1562, 2015.

Maximenko, N., Hafner, J., and Niiler, P.: Pathways of marine debris derived from trajectories of Lagrangian drifters, Marine pollution bulletin, 65, 51–62, 2012.

Monaghan, J. J.: Smoothed particle hydrodynamics, Reports on progress in physics, 68, 1703, 2005.

Neumann, D., Callies, U., and Matthies, M.: Marine litter ensemble transport simulations in the southern North Sea, Marine pollution bulletin, 86, 219–228, 2014.

Obbard, R. W., Sadri, S., Wong, Y. Q., Khitun, A. A., Baker, I., and Thompson, R. C.: Global warming releases microplastic legacy frozen in Arctic Sea ice, Earth's Future, 2, 315–320, 2014.

Onink, V., Wichmann, D., Delandmeter, P., and Van Sebille, E.: The role of Ekman currents, geostrophy and Stokes drift in the accumulation of floating microplastic, Journal of Geophysical Research: Oceans, 124, 2018.

Onink, V., Wichmann, D., Delandmeter, P., and Van Sebille, E.: The role of Ekman currents, geostrophy and Stokes drift in the accumulation of floating microplastic, Journal of Geophysical Research: Oceans, 124, 2019.

Paris, C. B., Helgers, J., Van Sebille, E., and Srinivasan, A.: Connectivity Modeling System: A probabilistic modeling tool for the multi-scale tracking of biotic and abiotic variability in the ocean, Environmental Modelling & Software, 42, 47–54, 2013.

Phillips, J. S., Gupta, A. S., Senina, I., van Sebille, E., Lange, M., Lehodey, P., Hampton, J., and Nicol, S.: An individual-based model of skipjack tuna (Katsuwonus pelamis) movement in the tropical Pacific ocean, Progress in Oceanography, 164, 63–74, 2018.

Prodhomme, C., Batté, L., Massonnet, F., Davini, P., Bellprat, O., Guemas, V., and Doblas-Reyes, F.: Benefits of increasing the model resolution for the seasonal forecast quality in EC-Earth, Journal of Climate, 29, 9141–9162, 2016.

Rio, M.-H., Mulet, S., and Picot, N.: Beyond GOCE for the ocean circulation estimate: Synergetic use of altimetry, gravimetry, and in situ data provides new insight into geostrophic and Ekman currents, Geophysical Research Letters, 41, 8918–8925, 2014.

Rubio, A., Mader, J., Corgnati, L., Mantovani, C., Griffa, A., Novellino, A., Quentin, C., Wyatt, L., Schulz-Stellenfleth, J., Horstmann, J., et al.: HF radar activity in European coastal seas: next steps toward a pan-European HF radar network, Frontiers in Marine Science, 4, 8, 2017.

Saha, S., Moorthi, S., Pan, H.-L., Wu, X., Wang, J., Nadiga, S., Tripp, P., Kistler, R., Woollen, J., Behringer, D., et al.: The NCEP climate forecast system reanalysis, Bulletin of the American Meteorological Society, 91, 1015–1058, 2010.

Sasaki, H., Nonaka, M., Masumoto, Y., Sasai, Y., Uehara, H., and Sakuma, H.: An eddy-resolving hindcast simulation of the quasiglobal ocean from 1950 to 2003 on the Earth Simulator, in: High resolution numerical modelling of the atmosphere and ocean, pp. 157–185, Springer, 2008.

Smith, R., Jones, P., Briegleb, B., Bryan, F., Danabasoglu, G., Dennis, J., Dukowicz, J., Eden, C., Fox-Kemper, B., Gent, P., et al.: The parallel ocean program (POP) reference manual ocean component of the community climate system model (CCSM) and community earth system model (CESM), Rep. LAUR-01853, 141, 1–140, 2010.

Stanev, E., Badewien, T., Freund, H., Grayek, S., Hahner, F., Meyerjürgens, J., Ricker, M., Schöneich-Argent, R., Wolff, J., and Zielinski, O.: Extreme westward surface drift in the North Sea: Public reports of stranded drifters and Lagrangian tracking, Continental Shelf Research, 2019.

Thomas, C. J., Lambrechts, J., Wolanski, E., Traag, V. A., Blondel, V. D., Deleersnijder, E., and Hanert, E.: Numerical modelling and graph theory tools to study ecological connectivity in the Great Barrier Reef, Ecological Modelling, 272, 160–174, 2014.

Tolman, H. L. et al.: User manual and system documentation of WAVEWATCH III TM version 3.14, Tech. rep., MMAB Contribution, 2009.

van Sebille, E., England, M. H., and Froyland, G.: Origin, dynamics and evolution of ocean garbage patches from observed surface drifters, Environmental Research Letters, 7, 044 040, 2012.

van Sebille, E., Wilcox, C., Lebreton, L., Maximenko, N., Hardesty, B. D., Van Franeker, J. A., Eriksen, M., Siegel, D., Galgani, F., and Law, K. L.: A global inventory of small floating plastic debris, Environmental Research Letters, 10, 124 006, 2015.

van Sebille, E., Griffies, S. M., Abernathey, R., Adams, T. P., Berloff, P., Biastoch, A., Blanke, B., Chassignet, E. P., Cheng, Y., Cotter, C. J., et al.: Lagrangian ocean analysis: Fundamentals and practices, Ocean Modelling, 68, 49–75, 2018.

Weisstein, E. W.: Surface Parameterization, http://mathworld.wolfram.com/SurfaceParameterization.html, from MathWorld – A Wolfram Web Resource, 2018.

Wubs, F. W., Arie, C., and Dijkstra, H. A.: The performance of implicit ocean models on B-and C-grids, Journal of Computational Physics, 211, 210–228, 2006.