Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2018-339
Manuscript under review for journal Geosci. Model Dev.
Discussion started: 15 January 2019

Geoscientific Model Development
Discussions

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 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 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 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 analyse the ocean dynamics given the flow field. The flow field results from numerical models or from other sources. Such sources are for example processed data 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), fish (e.g Phillips et al., 2018), icebergs (e.g. Marsh et al., 2015), and many more different types. 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 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 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 validate the new developments through a study of the sensitivity of floating microplastic dispersion on the North West European continental shelf and discuss those 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.

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 types of staggered grids have their pros and cons (Cushman-Roisin and Beckers, 2011). Parcels reads the more popular distributions occurring in geophysical fluid dynamics: A-, B- and C- grids. A- and C- are fundamentally different (Fig. 2) and the method to interpolate fields on those grids is defined below. 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 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.

2.1.1 A-grid

The A-grid is the un-staggered distribution of Arakawa's grids: 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.

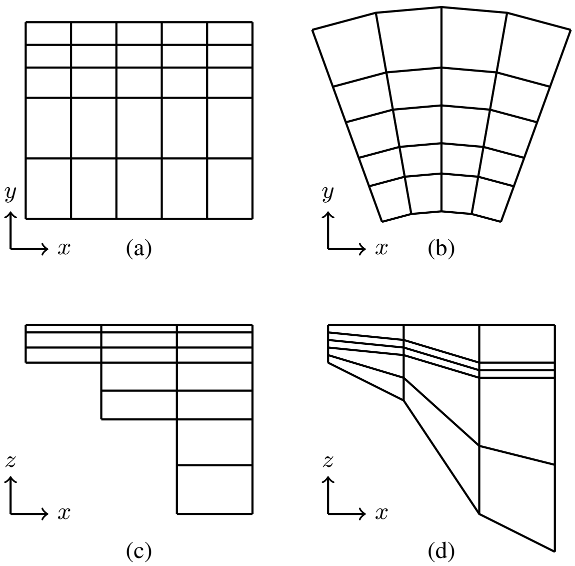


Figure 1. Different grid discretisations processed by Parcels. In the horizontal plane: (a) rectilinear, (b) curvilinear; in the vertical: (c) z-levels and (d) s-levels.

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} and the four nodal values F_k with $k = 0,...,3$, surrounding the cell, resulting in the following expression:

$$f(x,y) = \sum_k \phi_k^{2D}(\xi,\eta)\, F_k, \tag{1}$$

$$\text{with } \xi,\eta \text{ s.t.} \begin{cases} x = \sum_k \phi_k^{2D}(\xi,\eta)\, X_k \\ y = \sum_k \phi_k^{2D}(\xi,\eta)\, Y_k. \end{cases} \tag{2}$$

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) are the coordinates of the cell vertices. The two-dimensional Lagrange polynomials ϕ_k^{2D} are the bi-linear functions:

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

In a rectilinear mesh, solving Eq. 2 reduces to the usual solutions:

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

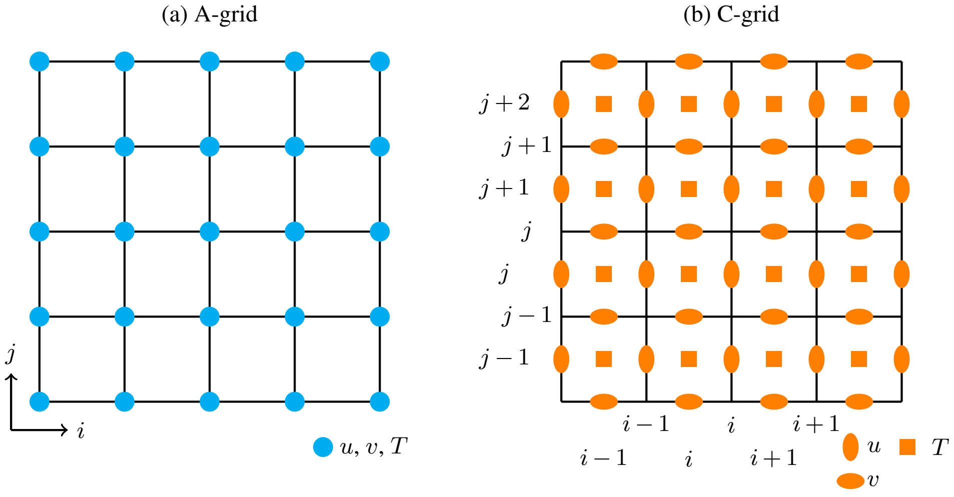


Figure 2. Different variable distributions (Arakawa and Lamb, 1977): (a) A-grid and (b) C-grid. In C-grid, i and j represent the variable column and row indexing in arrays where the variables are stored. The indexing of the C-grid follows the NEMO notations (Madec et al., 2016).

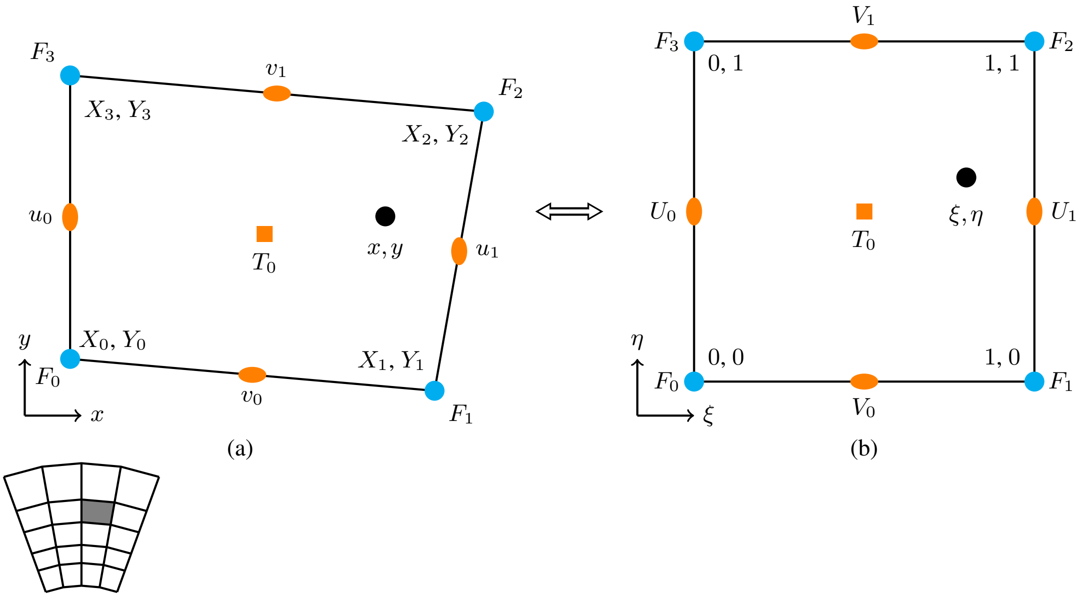


Figure 3. Variable distribution for a A-grid (cyan nodes) and C-grid (orange nodes) cell with (a) physical coordinates in the mesh cell and (b) relative coordinates in the unit cell.

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}:

$$\begin{array}{ll}
\phi_0^{3D}(\xi,\eta)=(1-\xi)(1-\eta)(1-\zeta), & \phi_1^{3D}(\xi,\eta)=\xi\,(1-\eta)(1-\zeta),\\
\phi_2^{3D}(\xi,\eta)=\xi\,\eta\,(1-\zeta), & \phi_3^{3D}(\xi,\eta)=(1-\xi)\,\eta\,(1-\zeta),\\
\phi_4^{3D}(\xi,\eta)=(1-\xi)(1-\eta)\,\zeta, & \phi_5^{3D}(\xi,\eta)=\xi\,(1-\eta)\,\zeta,\\
\phi_6^{3D}(\xi,\eta)=\xi\,\eta\,\zeta, & \phi_7^{3D}(\xi,\eta)=(1-\xi)\,\eta\,\zeta.
\end{array}$$

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

$$z_0=\sum_{k=0}^{3}\phi_k^{2D}Z_k, \qquad z_1=\sum_{k=4}^{7}\phi_k^{2D}Z_k. \tag{5}$$

2.1.2 C-grid

In a C-grid discretisation, the velocities are distributed 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 zonal and meridional velocities as if they would be on two different A-grids, shifted from half a cell, would not be in accordance with the C-grid formulation: in particular, such interpolation would break impermeability at the coastal boundary conditions. Here, 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), 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 are defined below.

2D field

For readability, the velocities at the $(j,\,i)$ cell edges are $(u_0,\,u_1,\,v_0,\,v_1)$, 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 but like in finite volume schemes they are approximated by interpolating linearly 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:

$$\begin{cases}
u=\dfrac{\partial x}{\partial t}=\sum_k\left(\dfrac{\partial\phi_k^{2D}}{\partial\xi}\dfrac{\partial\xi}{\partial t}+\dfrac{\partial\phi_k^{2D}}{\partial\eta}\dfrac{\partial\eta}{\partial t}\right)X_k=\dfrac{\partial x}{\partial\xi}\dfrac{\partial\xi}{\partial t}+\dfrac{\partial x}{\partial\eta}\dfrac{\partial\eta}{\partial t},\\
v=\dfrac{\partial y}{\partial t}=\sum_k\left(\dfrac{\partial\phi_k^{2D}}{\partial\xi}\dfrac{\partial\xi}{\partial t}+\dfrac{\partial\phi_k^{2D}}{\partial\eta}\dfrac{\partial\eta}{\partial t}\right)Y_k=\dfrac{\partial y}{\partial\xi}\dfrac{\partial\xi}{\partial t}+\dfrac{\partial y}{\partial\eta}\dfrac{\partial\eta}{\partial t},
\end{cases} \tag{6}$$

with

$$\mathbf{J}^{2D}(\xi,\eta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_k \frac{\partial \phi_k^{2D}}{\partial \xi} X_k & \sum_k \frac{\partial \phi_k^{2D}}{\partial \eta} X_k \\ \sum_k \frac{\partial \phi_k^{2D}}{\partial \xi} Y_k & \sum_k \frac{\partial \phi_k^{2D}}{\partial \eta} Y_k \end{bmatrix}, \tag{7}$$

$\mathbf{J}^{2D}(\xi,\eta)$ 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:

$$\begin{cases} \frac{\partial \xi}{\partial t} = \frac{(1-\xi)U_0 + \xi U_1}{J^{2D}(\xi,\eta)}, \\ \frac{\partial \eta}{\partial t} = \frac{(1-\eta)V_0 + \eta V_1}{J^{2D}(\xi,\eta)}. \end{cases} \tag{8}$$

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

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 are thus interpolated as in the 2D case, and the vertical velocity is interpolated as:

$$w = \zeta w_0 + (1-\zeta) w_1, \tag{9}$$

with $w_0 = w_{i+1,j+1,l}$, and $w_1 = w_{i+1,j+1,l+1}$.

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. 8 and divide this result by the Jacobian. Indeed, in 2D the Jacobian is a bi-linear function of ξ and η, such that a uniform velocity field is exactly interpolated by Eq. 6, which is precisely bi-linear in ξ and η. But in 3D, the Jacobian J^{3D} is a tri-quadratic function of the coordinates ξ, η and ζ:

$$J^{3D}(\xi,\eta,\zeta) = \det\left(\mathbf{J}^{3D}\right), \quad \text{with } \mathbf{J}^{3D}(\xi,\eta,\zeta) = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \sum_k \frac{\partial \phi_k^{3D}}{\partial \xi} X_k & \sum_k \frac{\partial \phi_k^{3D}}{\partial \eta} X_k & \sum_k \frac{\partial \phi_k^{3D}}{\partial \zeta} X_k \\ \sum_k \frac{\partial \phi_k^{3D}}{\partial \xi} Y_k & \sum_k \frac{\partial \phi_k^{3D}}{\partial \eta} Y_k & \sum_k \frac{\partial \phi_k^{3D}}{\partial \zeta} Y_k \\ \sum_k \frac{\partial \phi_k^{3D}}{\partial \xi} Z_k & \sum_k \frac{\partial \phi_k^{3D}}{\partial \eta} Z_k & \sum_k \frac{\partial \phi_k^{3D}}{\partial \zeta} Z_k \end{bmatrix}, \tag{10}$$

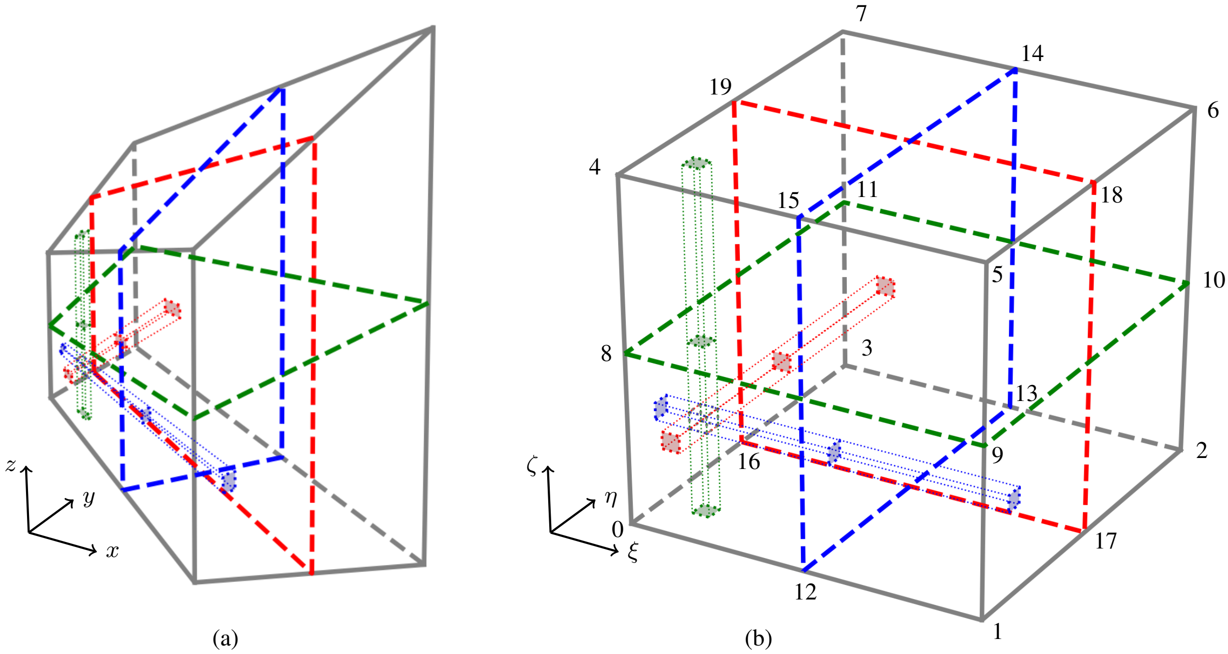


Figure 4. Fluxes used for 3D interpolation on a C-grid for (a) the physical cell and (b) the unit cell. The node indices in panel b are used to define the faces in Eq. 14.

such that a linear interpolation of the fluxes does not resolve exactly a uniform velocity. The velocities are still derived from the coordinate transformation:

$$\begin{cases} u = \dfrac{\partial x}{\partial t} = \sum_k \left(\dfrac{\partial \phi_k^{3D}}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \eta} \dfrac{\partial \eta}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \zeta} \dfrac{\partial \zeta}{\partial t} \right) X_k = \dfrac{\partial x}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial x}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial x}{\partial \zeta}\dfrac{\partial \zeta}{\partial t}, \\ v = \dfrac{\partial y}{\partial t} = \sum_k \left(\dfrac{\partial \phi_k^{3D}}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \eta} \dfrac{\partial \eta}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \zeta} \dfrac{\partial \zeta}{\partial t} \right) Y_k = \dfrac{\partial y}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial y}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial y}{\partial \zeta}\dfrac{\partial \zeta}{\partial t}, \\ w = \dfrac{\partial z}{\partial t} = \sum_k \left(\dfrac{\partial \phi_k^{3D}}{\partial \xi} \dfrac{\partial \xi}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \eta} \dfrac{\partial \eta}{\partial t} + \dfrac{\partial \phi_k^{3D}}{\partial \zeta} \dfrac{\partial \zeta}{\partial t} \right) Z_k = \dfrac{\partial z}{\partial \xi}\dfrac{\partial \xi}{\partial t} + \dfrac{\partial z}{\partial \eta}\dfrac{\partial \eta}{\partial t} + \dfrac{\partial z}{\partial \zeta}\dfrac{\partial \zeta}{\partial t}, \end{cases} \tag{11}$$

but this time the relative velocities interpolate the fluxes using quadratic Lagrangian functions:

$$\begin{cases} \dfrac{\partial \xi}{\partial t} = \dfrac{\left(2\xi^2 - 3\xi + 1\right) U_0 + \left(-4\xi^2 + 4\xi\right) U_{1/2} + \left(2\xi^2 - \xi\right) U_1}{J^{3D}(\xi,\eta,\zeta)}, \\ \dfrac{\partial \eta}{\partial t} = \dfrac{\left(2\eta^2 - 3\eta + 1\right) V_0 + \left(-4\eta^2 + 4\eta\right) V_{1/2} + \left(2\eta^2 - \eta\right) V_1}{J^{3D}(\xi,\eta,\zeta)}, \\ \dfrac{\partial \zeta}{\partial t} = \dfrac{\left(2\zeta^2 - 3\zeta + 1\right) W_0 + \left(-4\zeta^2 + 4\zeta\right) W_{1/2} + \left(2\zeta^2 - \zeta\right) W_1}{J^{3D}(\xi,\eta,\zeta)}. \end{cases} \tag{12}$$

The different fluxes interpolated in Eq. 12 are the product of the velocities with the face Jacobian $J^{2D,f}$, as illustrated in Fig. 4.

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

$$J_j^{2D,f}(\xi,\eta,\zeta) = \sqrt{\left(M_{0,j}(\mathbf{J}^{3D})\right)^2 + \left(M_{1,j}(\mathbf{J}^{3D})\right)^2 + \left(M_{2,j}(\mathbf{J}^{3D})\right)^2}, \tag{13}$$

with $M_{i,j}(\mathbf{J}^{3D})$, the i,j 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. 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, the Jacobians and the different fluxes read then:

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

(14)

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, 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.

We write here the development for face $[12,13,14,15]$. The two other face 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:

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

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

$$U_{1/2} = U_{1/2}^{+} \frac{J_0^{*2D,f}(1,\eta,\zeta)}{J_0^{*2D,f}(1,1/2,1/2)}. \tag{15}$$

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 $[12,1,2,13,15,5,6,14]$.

2.2 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 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.

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

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 to the framework.

The first object is the `VectorField`, that jointly interpolates the different 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 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 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 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. 8 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. 6:

$$\frac{\partial x}{\partial t} = u$$
$$\frac{\partial x}{\partial t} = v,$$

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:

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

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:

$$U_{1/2} = 0.5\left(U_0 + U_1\right), \tag{16}$$

and similarly for $V_{1/2}$ and $W_{1/2}$. The first two lines of Eq. 11 reduce to Eq. 6 and the third line is:

$$\frac{\partial z}{\partial t} = \left(Z_4 - Z_0\right) \frac{\partial \zeta}{\partial t}. \tag{17}$$

Finally using Eq. 16, Eq. 12 becomes:

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

from which first two lines correspond to Eq. 8 and third combined with Eq.17 reduces to Eq. 9.

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

Microplastic (MP) is transported trough 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 subtropics (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 focused on marine litter in the Southern part of the North Sea (Neumann et al., 2014; Gutow et al., 2018), including diffusion and wind drift to their model and at 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 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. 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.

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.

4.1.1 NEMO

The main data we use is NEMO-N006, which is a standard set-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 of 1/4° and $1/12°$. They are discretised on an ORCA grid (Madec and Imbard, 1996), a global ocean tripolar C-grid, which is curvilinear. 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). The data are originally computed on a C-grid, but are re-interpolated on the tracer nodes to form an A-grid dataset which is available on CMEMS, with a temporal resolution of 1 day.

The data, that will be referred to as CMEMS, 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 simulations are run in the following configurations: (a) NEMO hydrodynamics at a 1/12° resolution, (b) NEMO at a 1/4° resolution, (c) CMEMS 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.

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.

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):

$$D = D_0 \, (l/l_0)^{4/3}, \tag{18}$$

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 has impermeable boundary conditions at the coast. The second includes Stokes drift and diffusion. It is a process allowing beaching.

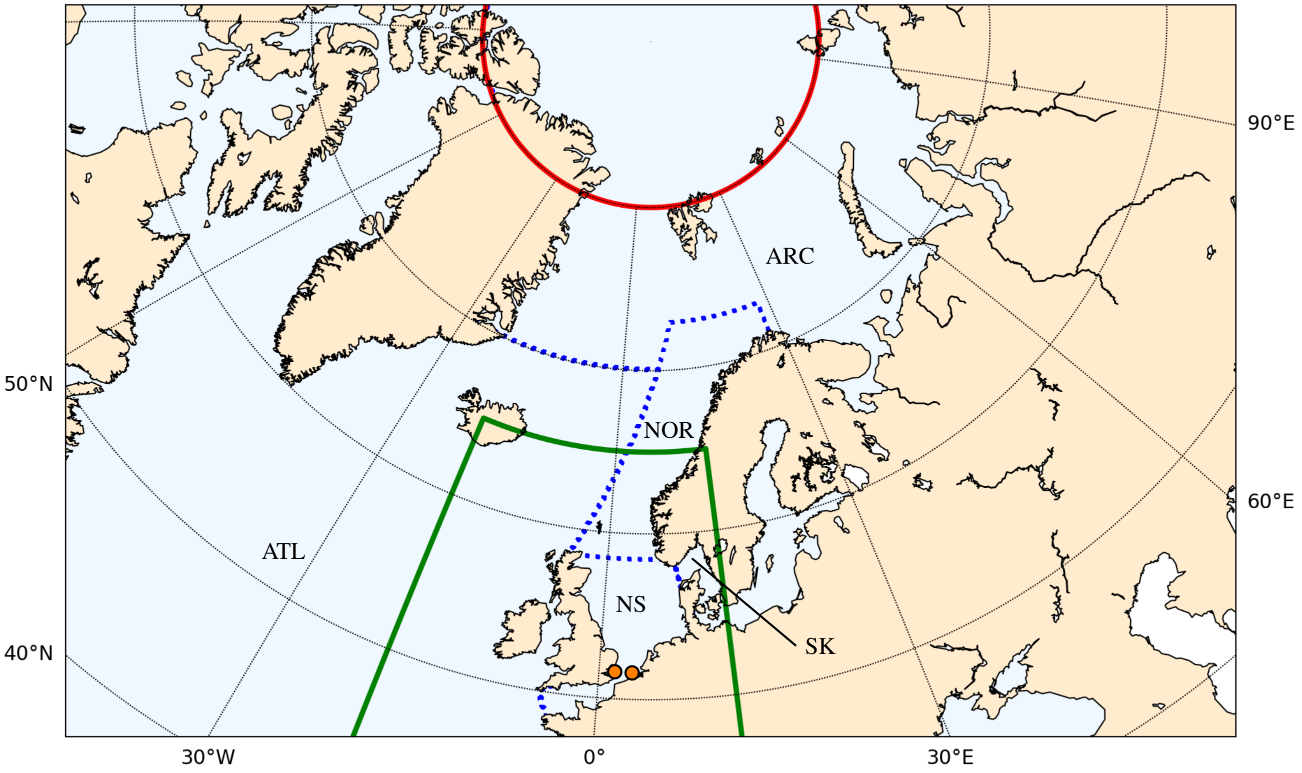


Figure 5. Spatial coverage of the OGCM data used to study North Sea MP transport. NEMO data are available globally. CMEMS data are available for the North Sea region (green boundaries) and WaveWatch III data are available South of 80° N (red boundary). The particle releasing locations are the Rhine and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

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 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 advection is first run. 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, 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.2525279.

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

The MP 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, the ocean is discretised into cells of 1/4° x 1/8° resolution and the fraction of particles having travelled at least once by a cell is computed (Fig. 7).

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

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 plastic does not beach in both runs. This leads to a reduced amount of plastic reaching the Arctic with 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, 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 during the first year, when the particles are located South of 65° N. This explains the lack of differences in the northern 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 (Fig. 6d), due to prevailing westerly winds (Gutow et al., 2018), with 46% of the particles 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.

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. The amount of MP reaching the Arctic is reduced by 68%, 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.

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, which was made possible by the development of the different field interpolation schemes and constructions such as `NestedField` and `SummedField` objects. To validate the MP dynamics observed, it is essential to couple such numerical study with an extensive field study.

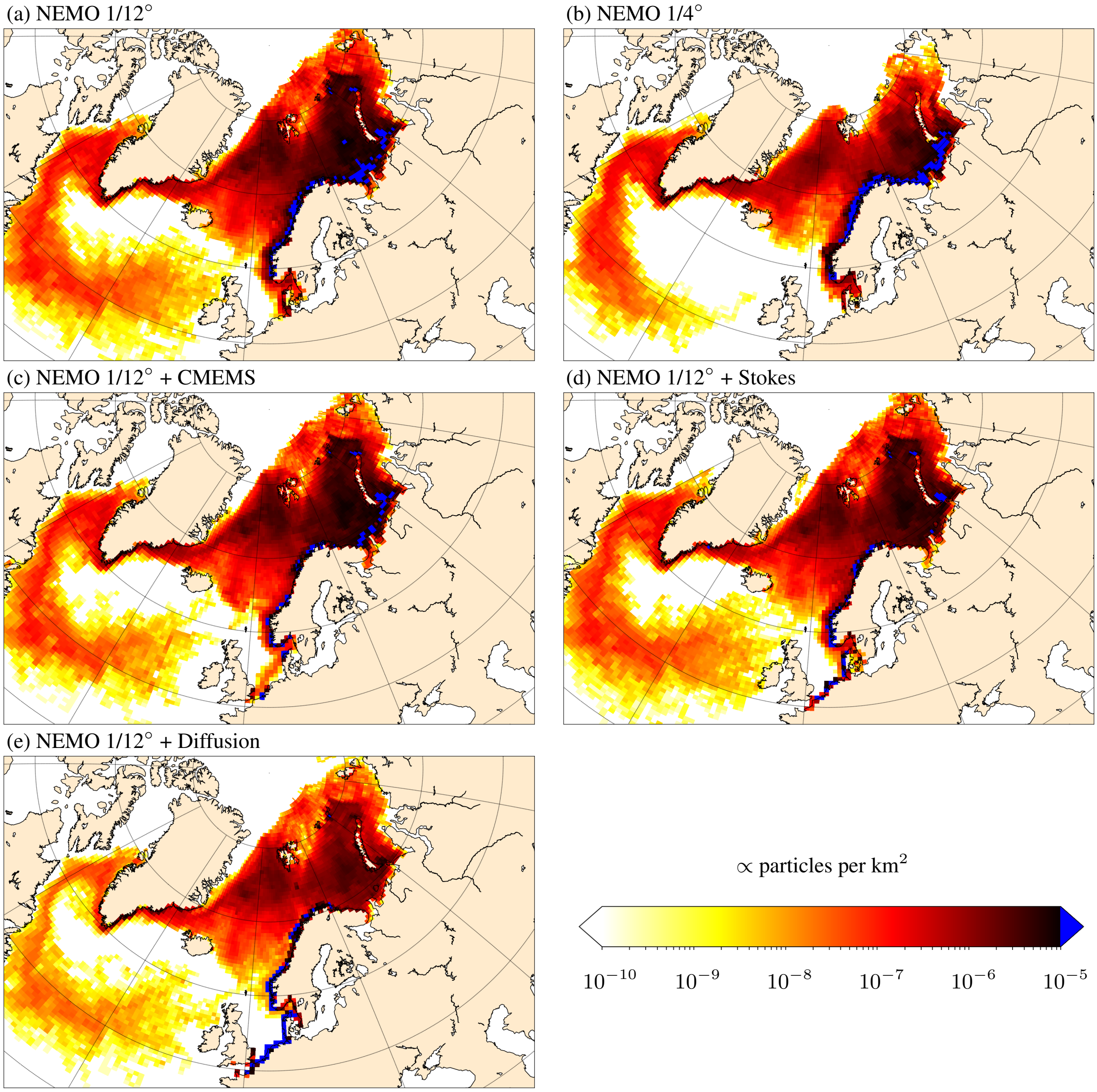
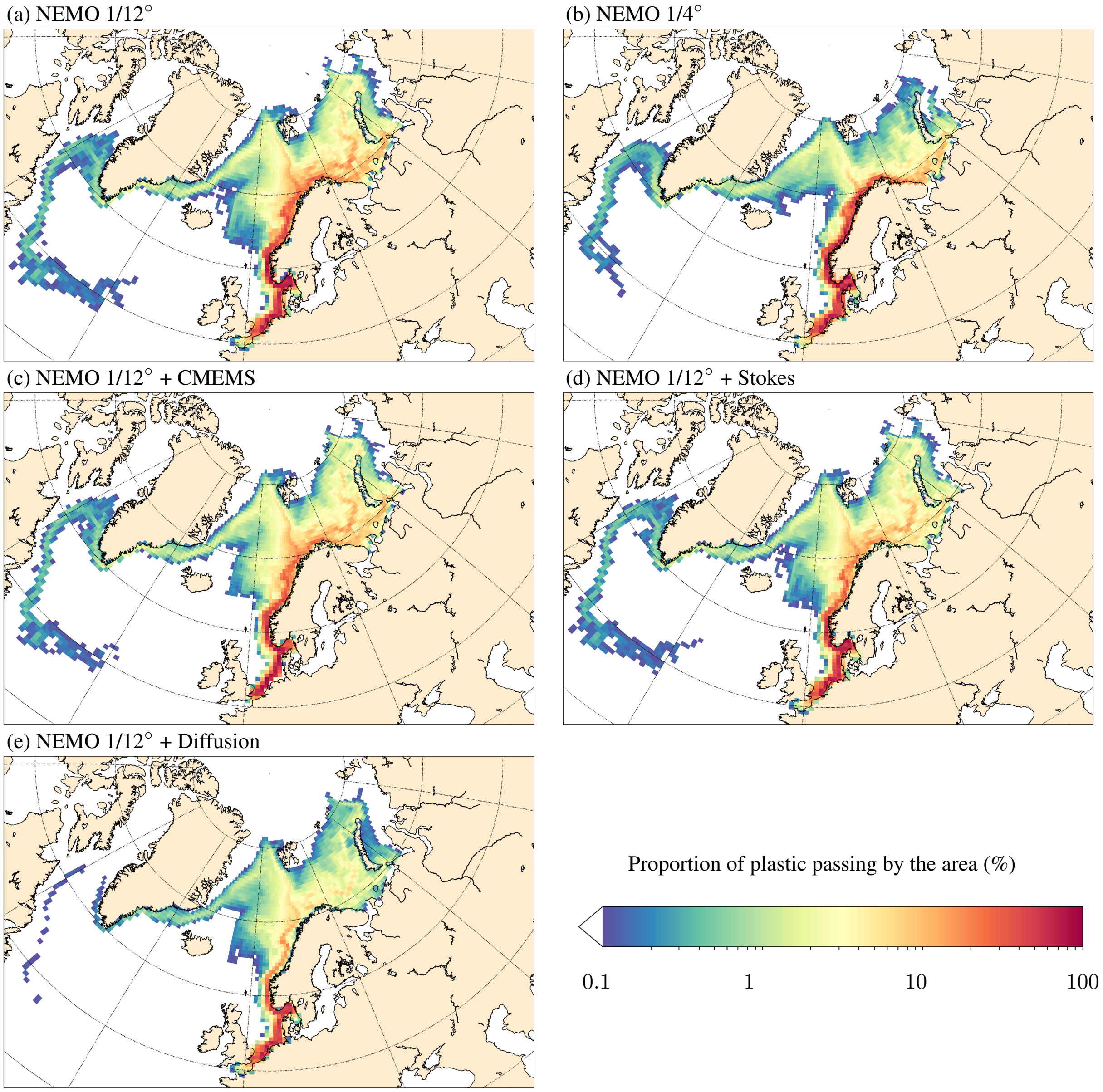


Figure 6. Floating MP density, averaged over the third year of particle age, for the different simulation scenarios: (a) NEMO 1/12°, (b) NEMO 1/4°, (c) CMEMS nested into NEMO 1/12°, (d) NEMO 1/12° coupled with Stokes drift from WaveWatch III, (e) NEMO 1/12° coupled with diffusion. Note the logarithmic scale.



(a) NEMO 1/12°

(b) NEMO 1/4°

(c) NEMO 1/12° + CMEMS

(d) NEMO 1/12° + Stokes

(e) NEMO 1/12° + Diffusion

Figure 7. Fraction of floating MP originally released in the Thames and Rhine estuaries reaching at least once the different region cells.

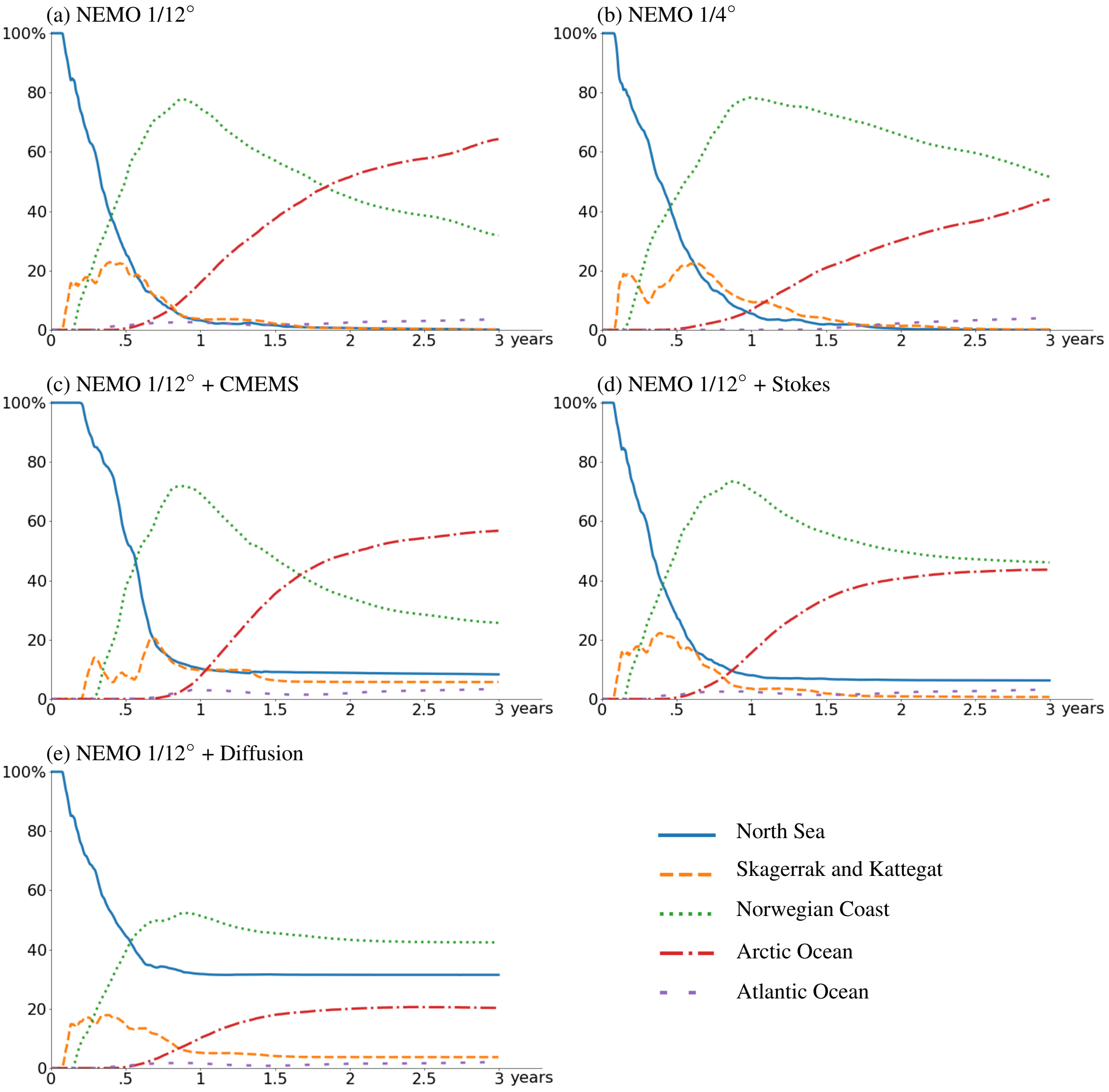


Figure 8. Evolution of the distribution of floating MP in the different zones (Fig. 5) as a function of particle age.

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, rectilinear and curvilinear horizontal grids and z- and s-vertical levels. Different 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.2422179. 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.2525279.
- 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=NORTHWESTSH
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. 715386). 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 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.

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.

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.

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, (under revision), 2018.

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.

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.