Environmental Modelling and Software 125 (2020) 104621

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: http://www.elsevier.com/locate/envsoft

Modelling marine particle dynamics with LTRANS-Zlev: implementation and validation

Célia Laurent, Stefano Querin, Cosimo Solidoro *, Donata Melaku Canu

National Institute of Oceanography and Applied Geophysics - OGS, Borgo Grotta Gigante 42/C, 34010, Sgonico, TS, Italy

ARTICLE INFO

Keywords:
LTRANS
MITgcm
Lagrangian modelling
Larva
Oil spill

ABSTRACT

This paper presents the release of LTRANS-Zlev, which is a new version of the off-line Lagrangian ocean particle-tracking model LTRANS v.2b that is compatible with a Z-coordinate (constant-depth layers) discretization of the hydrodynamic equations. The model capitalizes on and massively extends the capabilities of the original code LTRANS, which is already quite popular, but can be used only adopting a sigma-coordinate (terrain-following layers) discretization. Among the additional features included in LTRANS-Zlev, there are the backward-in-time particle-tracking algorithm and some new customizable larval behaviour options. The new version also includes the OILTRANS-module for oil spill simulations. The new implementations were validated by using the output of the Z-coordinate Massachusetts Institute of Technology general circulation model (MITgcm) for an idealized case study describing a cyclonic gyre in a mid-latitude closed basin. Another test-case, in which larval dispersal is modelled in the northern Adriatic Sea, illustrates some of the new features of LTRANS-Zlev.

1. Introduction

Lagrangian particle-tracking algorithms provide a way to simulate, analyse and describe oceanographic transport processes that can be of particular interest in a number of studies, from larval transport (Cowen and Sponaugle, 2009; Werner et al., 2007), to pollution dispersion and water quality assessment (James, 2002), to oil spill modelling (Canu et al., 2015; North et al., 2011). In the Lagrangian description (Pope, 1985; Taylor, 1922) the transport processes are tracked by following fluid particles in order to identify the trajectories starting from or arriving at a given point. The transport equations are discretized along a moving frame of reference and particle positions and velocities are considered also in between the Eulerian grid points. Therefore, Lagrangian approaches provide a finer description of the transport processes compared with the one traditionally obtained in purely Eulerian transport models, in which the dynamics of the flow is described by computing the hydrodynamic properties at every grid element of a fixed (geo) reference grid. However, the Lagrangian approach still requires the input of advection velocity fields, which are used to update the particle position and must be computed in advance by an Eulerian hydrodynamic model.

In an Eulerian model, the vertical discretization of the computational domain can be based mainly on three kinds of coordinate systems: constant depth levels (Z), terrain-following (σ) layers, or isopycnal (ρ) layers. In the Z-case, each layer is characterized by a constant depth over the whole domain and therefore the computational domain has more vertical layers in the deeper areas than in the shallower ones. Examples of such a kind of models are MOM (Griffies et al., 2000, 2004), NEMO (Madec and the NEMO Team, 2016) and MITgcm (Adcroft et al., 2017; Marshall et al., 1997a, 1997b). Conversely, in the terrain following σ-models such as POM (Blumberg and Mellor, 1987; Mellor, 2003) and ROMS (Shchepetkin and McWilliams, 2009), the number of layers is the same all over the domain, so that in deeper areas layers are thicker. Isopycnal models such as MICOM (Bleck et al., 2002) use a vertical discretization based on the potential density ρ referenced to a given pressure value.

Each discretization technique (Z, σ or ρ) has both advantages and disadvantages (Chassignet et al., 1996; Griffies et al., 2000) and the choice between these approaches is not straightforward (Shapiro et al., 2013). It depends on several factors such as the basin topography, the relative importance of the various processes in a given study-site and the numerical schemes implemented in the model (Shchepetkin and McWilliams, 2009). The Z-level discretization is usually more accurate in representing the surface mixed layer and the diabatic processes it includes, while the σ coordinates allows a better representation of the bottom boundary layer. Isopycnal models are recommended for tracer

* Corresponding author.
E-mail addresses: claurent@inogs.it (C. Laurent), squerin@inogs.it (S. Querin), csolidoro@inogs.it (C. Solidoro), dcanu@inogs.it (D.M. Canu).

https://doi.org/10.1016/j.envsoft.2020.104621
Received 31 May 2019; Received in revised form 3 December 2019; Accepted 6 January 2020
Available online 10 January 2020

advection and diffusion along sloping density surfaces in the ocean. Isopycnal models are less efficient in representing the bottom boundary layer and the surface mixing layer, which are mostly unstratified; therefore, Z or σ models may be preferred when simulating water basins whose dynamics are strongly influenced by the processes involved in these layers. For basins showing remarkable variations of topography, a Z-grid model is computationally more expensive than a σ model, as it requires a higher number of vertical levels than a σ model to solve both the surface and bottom boundary layer dynamics (Blumberg and Mellor, 1987). Conversely, using σ coordinates may cause inconsistencies in the areas of sloping topography, where numerical truncation errors in the computation of the pressure gradient force may become important and where the condition of hydrostatic consistency is more critical (Haney, 1991). To handle such cases, the vertical resolution of the σ grid must be increased (Mellor et al., 1994) at the cost of reducing model performance.

Most Eulerian geophysical dynamics models already have their own native on-line Lagrangian tracer tracking modules. However, for studies involving multiple experiments, large or high-resolution time or space intervals, or ensemble simulations, the computational cost of using on-line tracking modules can be very high, and often not affordable, since the whole hydrodynamic flow fields have to be recomputed for every realization. On the contrary, the use of off-line Lagrangian particle transport models allows to perform multiple transport modelling studies, using a single set of stored flow fields with very affordable computational costs. It also permits backtracking applications. Off-line models can be used as long as it is assumed that the transported materials have no effect on the geophysical flow (passive tracers) and as long as the hydrodynamic stored fields have a sufficiently refined temporal resolution to approach high frequency phenomena. The use of Lagrangian models is -indeed- growing and recent reviews and books illustrating the wide use and popularity of the Lagrangian approach in oceanography can be found in recent literature (Lynch et al., 2014; Prants et al., 2017; van Sebille et al., 2018). Among the principal off-line Lagrangian models we can cite ICHTHYOP (Lett et al., 2008), which was developed for σ coordinates systems; Parcels (Lange and Sebille, 2017) is instead compatible with both σ and Z coordinates, but does not yet include ready-to-use larval behaviour and oil spill weathering modules, while CMS (Paris et al., 2013) offers these possibilities, but it is compatible only with Z-grid models.

The Lagrangian TRANSport model LTRANS (LTRANS v.2b) is one of the most popular off-line three-dimensional particle tracking modules (North et al., 2013; Schlag and North, 2012). It has been developed to simulate the movement of passive tracers, particles with sinking or floating behaviour like sediment or oil droplets, and planktonic organisms like oyster larvae. The source code is written in FORTRAN 90, it includes the popular 4^{th} order Runge-Kutta scheme for particle advection and a random displacement model to account for sub-grid scale turbulent diffusion. LTRANS v.2b estimates two-dimensional water properties (such as sea surface height or bottom depth) at the particle location using bi-linear interpolation. Instead, three dimensional quantities (fluid velocity, temperature, etc…) are estimated by (see subsection 2.3) (a) interpolating them bi-linearly at the particle horizontal location for every vertical level and (b) creating a water column profile using a tension spline curve (North et al., 2006). Sticking and reflective boundary conditions on solid walls, specific particle behaviour routines, and settlement algorithms are also included. The transport model can be used together with an oil spill weathering module named OILTRANS and developed within LTRANS v.2 (Berry et al., 2012). OILTRANS considers mechanical spreading, advection and diffusion of the oil particles by winds and currents, as well as evaporation, emulsification and vertical dispersion.

Since hydrodynamic fields are usually stored with a too low frequency to drive particle tracking algorithms, LTRANS considers the time interval between two fields (e.g., hours or days) as an external forcing time-step. The particle motion time-scale is instead based on a much shorter internal time step (e.g., minutes, satisfying the CFL condition $U\Delta t/\Delta x < 1$) allowing particles to move over smaller space and time intervals, in order to ensure a good accuracy of the solution. At every internal time-step, the advection velocities at the particle position are interpolated in time and space from the Eulerian fields, and boundary condition algorithms ensure that the particles remain within the volume composed by the water cells of the domain. As an output, LTRANS provides NetCDF files containing the coordinates of the trajectories followed by the particles and, if requested, the flow field properties (temperature, salinity, etc…) sampled along these trajectories.

The LTRANS v.2b model was originally developed to run adopting the stored hydrodynamic fields produced by the Regional Ocean Modeling System (ROMS), which is based on a terrain-following σ-layer discretization. Due to this reason, LTRANS v.2b employs grids, bathymetry and velocity fields based on a σ coordinate system.

In this paper, we present LTRANS-Zlev, which is a new version of LTRANS that handles the grids and hydrodynamic flow fields of geophysical models based on a Z-coordinate discretization, with staggered Arakawa-C grid. The modification required an extensive editing of the code, but it significantly improved and widened its use, since many oceanographic models are -indeed- based on Z-levels. Furthermore, several new features and options have been implemented and/or included in the code: the backward-in-time particle tracking capacity, the capability of simulating coastal stranding in selected areas and at prescribed distances from the coast, new larval behaviour options and customizable larval scenarios (including near-bottom release, diel vertical migration followed by seabed transport), the capacity to handle multiple independent (i.e., not connected) water basins and the OILTRANS module for oil spill simulations. The Zlev version can handle Z* coordinates as well, by adjusting the height of every level along the vertical coordinate, according to the sea-surface elevation changes. Furthermore, LTRANS-Zlev still offers the possibility to use the vertical layer discretization based on σ instead of Z vertical coordinates.

The new version of LTRANS described in this paper has been tested with the hydrodynamic flow fields provided by a set of MITgcm (Massachusetts Institute of Technology general circulation model) simulations. The code provided in the repository includes a short user manual, a test-case ready to be used and some pre-processing tools to build the grids using the MITgcm outputs.

The details of the main developments implemented in LTRANS-Zlev in order to handle the Z-levels of the MITgcm model, instead of the σ discretization of ROMS, are described in section 2, together with the new features available in the Zlev version. Section 3.1 gives technical and computational information regarding LTRANS-Zlev, while section 3.2 presents a test-case performed on an idealized cyclonic gyre in a mid-latitude closed basin, in which we compared particle trajectories computed by LTRANS-Zlev off-line, coupled to the 3D Eulerian flow fields provided by the MITgcm, versus particle trajectories computed by the online native tracking module of the MITgcm. Then, section 3.3 presents a realistic application considering particles with larval behaviour, in which particles are released close to the seabed, then they rise towards the surface, perform diel vertical migration for a few days, after which they sink back to the bottom, where they are transported along the seabed, searching for the environmental conditions suitable for settlement. Conclusions and future perspectives are finally drawn in section 4.

2. Model description

2.1. Overview

The original LTRANS v.2b model comprises an initialization phase, which builds the grids and boundaries of the water domain, and a running phase loading the hydrodynamic flow fields at every external time-step and then updating the particles positions at every internal time-step. In LTRANS-Zlev the global structure of the LTRANS v.2b code

is unchanged, but most parts of code handling the grids, hydrodynamic flow fields and boundary conditions have been deeply rewritten. Besides, new functions have been implemented to calculate, for example, the depth of the deepest grid nodes, which define the Z-level bathymetry, and to interpolate the bottom depth at the particle position, leading to the new version of LTRANS represented in the diagram of Fig. 1.

In addition to the source code, an external pre-processing Python tool (included in the LTRANS-Zlev package) creates the Z-grid file required by LTRANS-Zlev, starting from the output file provided by the off-line Z-level hydrodynamic model (MITgcm, in our case). This external tool rewrites the grid coordinates and water/land masks at every node of the Arakawa-C grid in a unique file in NetCDF format, as requested by the initial version of LTRANS. The native MITgcm binary output files (or the merged-in-time ones) are instead read directly by LTRANS-Zlev (at step (*7) in Fig. 1).

2.2. Boundary conditions

While tracking the movement of Lagrangian particles, particular care has to be given to the boundary conditions. In fact, it might happen that the numerical integration of a particle path (i.e., the computation of where the interpolated velocity field projects a particle after a time step) moves a particle outside the water-volume of the computational domain. For this reason, it is fundamental to have well-defined boundaries, whose position and typology define what happens along the borders of the water-volume.

2.2.1. Z-level Arakawa-C grid

Most geophysical models such as ROMS or MITgcm use Arakawa-C grids, where the scalar properties (e.g., water density ρ) and the u, v and w components of velocity are defined on staggered nodes. In the Arakawa-C staggered grid the ρ nodes are at the centre of cubic (land or water) cells, while the u, v and w nodes lie on the cell edges (in the x, y and z directions, respectively). Fig. 2(a)–(b) illustrate the vertical levels

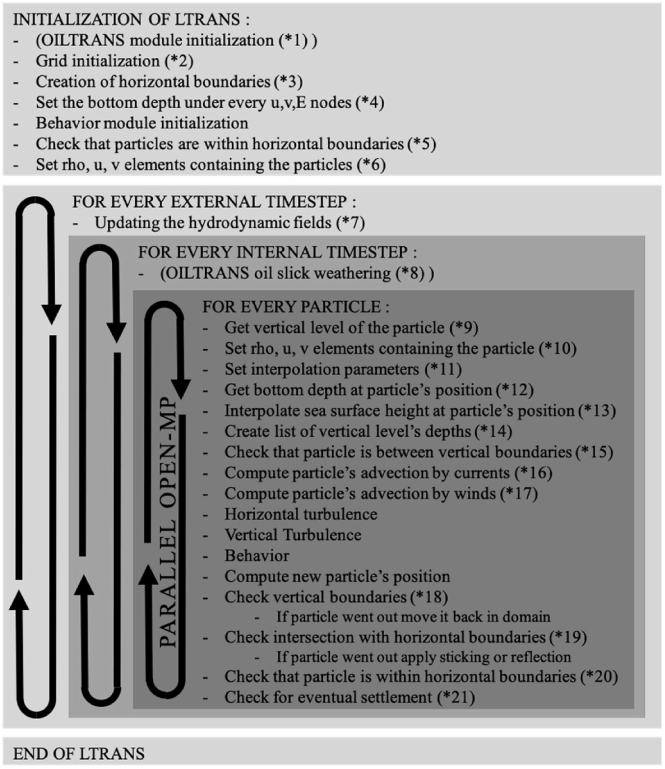


Fig. 1. LTRANS-Zlev flow diagram. Numbers with asterisk between parentheses (*.) identify the processes that have been modified.

and Arakawa-C staggering, respectively for a σ-level grid and a Z-level grid without partial bottom cells. In both cases, every level along the vertical direction is made by a planar layer whose grid is defined by ρ, u, v and w nodes. The horizontal position of the nodes does not vary with the vertical level, meaning that the nodes of two different vertical levels have the same horizontal (x, y) location but a different vertical (z) depth. In every grid cell, they store different values of the three-dimensional flow fields at every vertical level. The sea-surface elevation and the bottom depth are stored at ρ node positions, but they are two-dimensional quantities depending only on the horizontal (x, y) position.

As shown in Fig. 2(a)–(b), the water-volume of the computational domain is enclosed by two kinds of boundaries: the upper and lower boundaries made of horizontal planes delimiting the extremities of the water column (i.e., sea surface and bottom) and the lateral boundaries made of vertical planes delimiting the longitudinal and latitudinal extension of the basin.

2.2.2. Lateral boundaries

σ-level and Z-level grids represent lateral boundaries in a different way. In σ-coordinates, the number of water cells at any vertical (horizontal) section remains constant, but the height of each grid cell varies according to the local bottom depth. The planar layers can tilt vertically at all depths to follow the bathymetry, and the lateral land/water boundaries are the same for all layers (as shown in red in Fig. 2(a)). In contrast, Z-grids layer boundaries change with depth (Fig. 2(b)) and therefore different vertical transects might have a different number of wet (i.e., non-land) cells according to the horizontal position of the transect, and horizontal sections at different depths might have a different number of wet cells.

From the Eulerian grids shown in Fig. 2(a)–(b), LTRANS elaborates the lateral boundaries as illustrated in Fig. 3(a)–(b) (for σ and Z grids, respectively) by linking together at every level the u and v nodes at the interface between land and water cells to form a sequence of vertical planes that the particles cannot cross. The nodes positioned at the centre of every element made by 4 adjacent ρ nodes are called E nodes; their projection on the bottom is represented by the magenta circles in Fig. 3.

LTRANS v.2b considers the lateral borders of an ocean basin on a σ-level grid as a series of vertical walls whose horizontal (x, y) coordinates are identical at every vertical level. Fig. 3(a) shows that the water volume is enclosed by the same vertical-red planes for any vertical level. Conversely, for a Z-level grid the position of the lateral solid boundaries is instead dependent on the vertical level (Fig. 3(b)). As a consequence, on Z-level grids, there might be multiple isolated sub-basins at depth (as well as water basins nested inside island areas), due to the presence of rifts and trenches at the bottom of the sea; LTRANS-Zlev is designed to handle such kinds of water sub-basins.

Particles are often projected outside the water domain (across a lateral boundary) during the numerical integration of their position. In such a case, the particles will stop along the boundary, to simulate, for example, coastal stranding of a pollutant or bottom contamination. Alternatively, to simulate larval dispersion one might assume that the particles bounce against the boundary, and they are reflected back into the water. The two assumptions are labelled as sticking and reflective conditions, respectively.

2.2.3. Upper and lower boundaries

The sea surface height and the bottom depth at a particle location are estimated at every time-step in order to maintain the particle within the upper and lower boundaries of the water column. On a σ-level grid LTRANS v.2b interpolates these quantities by running bi-linear interpolations of the sea surface height and the bottom depth (represented by planar crosses and circles in Fig. 2(a), respectively), which are both two-dimensional quantities stored at the 4 ρ nodes surrounding horizontally a particle. By doing so, when a particle is moved laterally in an area shallower than its previous position, it is also risen up towards the surface, so that it cannot -by construction- cross the bottom boundary (i.

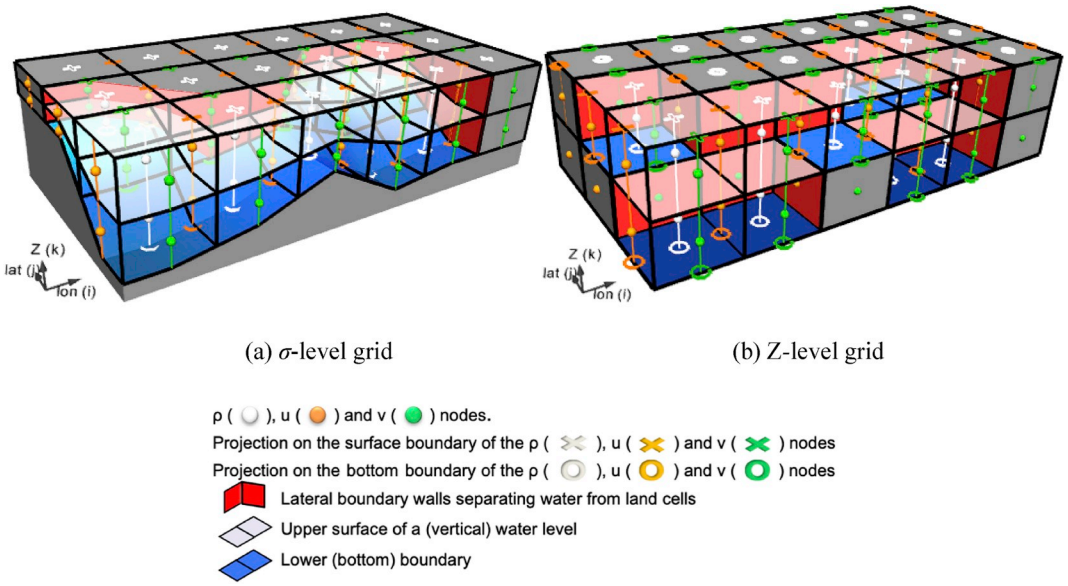


Fig. 2. Representation of a part of an Eulerian Arakawa-C grid with ρ nodes denoted by white dots, u and v staggered nodes by orange and green dots, respectively. The projection of the nodes on the bottom and surface boundaries are represented by planar circles and crosses, respectively. Black lines delimit the cubic cells surrounding every ρ node. The lower boundaries are identified by blue planes while white-transparent planes lie on top of every water layer. The red vertical surfaces represent the lateral boundary walls separating land from water cells. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

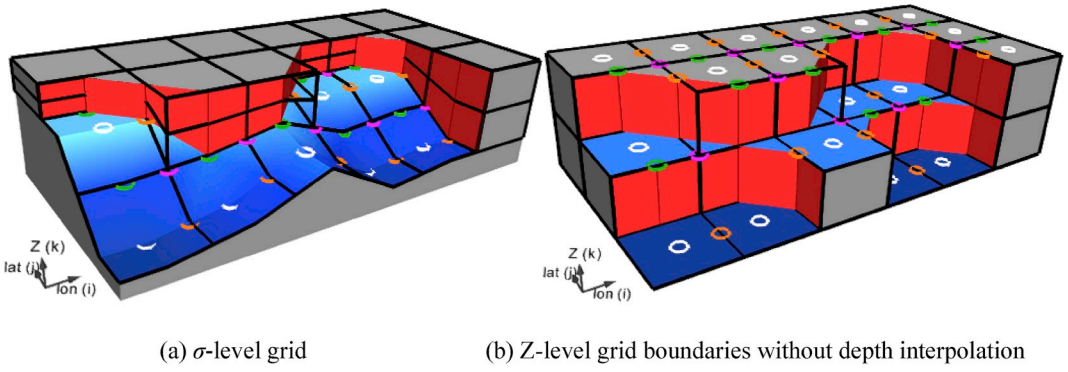


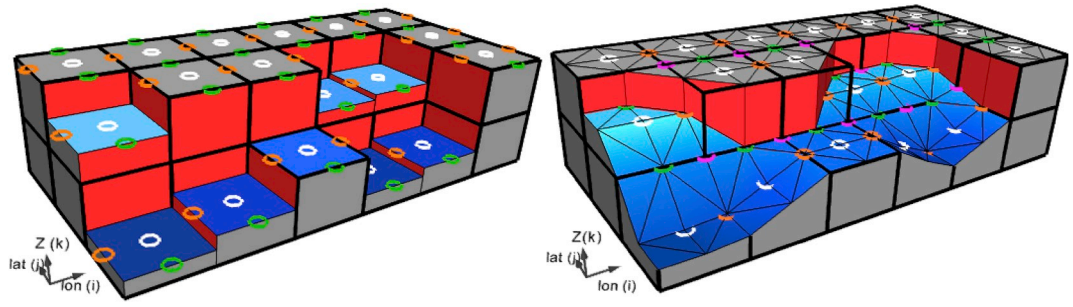
Fig. 3. Horizontal (red) and vertical-bottom (blue) boundaries elaborated by LTRANS. The projection of the ρ, u, v and E nodes on the bottom boundary are represented by planar circles (white, orange, green and magenta, respectively). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

e., flow below the deepest layer). Therefore, in LTRANS v.2b on a σ-level grid, the lateral boundary reflection conditions are applied only along the lateral (vertical) walls, that is when the particles reach a lateral-coastal (land or island) boundary represented in red in Fig. 2(a).

Using a Z-grid with depth dependent lateral boundaries, a direct linear interpolation of the bottom depth from the surrounding ρ nodes (planar white circles in Figs. 2(b)–3(b)) is no longer possible. In fact, such a seabed surface would pass across the deeper boundary walls and particles moving downward from one vertical level to another could find themselves trapped behind one of the vertical walls composing the lateral boundaries.

We implemented two possible configurations: the first one uses full cells with a bathymetry made by horizontal planes and vertical walls, while the second one is characterized by sloping surfaces and takes into account the partial cells. In the first approach (no partial cells) the LTRANS-Zlev bathymetry has to be compliant with the red boundary walls positioned along the u and v nodes represented in Fig. 3(b). To do so, a grid refinement is applied to the bottom boundary depth matrix (originally defined at the position of the ρ nodes) using the position of the ρ, u, v and E nodes as new grid points. The depth of the bottom under the u, v and E nodes is defined once for the whole simulation at step (*4) of Fig. 1; it depends on the shape of the lateral boundaries at the various vertical levels. The sharp bathymetry profile associated to the non-interpolated Z-grid boundaries shown in Fig. 3(b) causes multiple reflections along lateral walls, slowing down the particles that run into bathymetry elevations. This can be advantageous when running simulations for basins characterized by steep slopes in the bathymetry (such as submarine canyons), or when studying the interaction of marine particles with any kind of vertical man-built underwater structure (for example dikes or oil and gas extraction terminals). Unfortunately, this approach cannot accommodate a partial bottom cells discretization of the Eulerian grid.

A second alternative configuration of LTRANS-Zlev allows to avoid this limitation by implementing partial cells and considering the seabed as made by sloping surfaces. In this second case, the partial cells discretization of an Eulerian Z-grid (like the one shown in Fig. 4(a)) is represented in the Lagrangian simulation by a smoother interpolated bathymetry presenting tilted planes along the bottom surface such as the one illustrated in Fig. 4(b). In this configuration, the local depth is evaluated by a linear interpolation of the three nodes composing the



a) Eulerian bathymetry (b) LTRANS-Zlev interpolated bathymetry

Fig. 4. Z-grid bathymetry with partial bottom cells as considered by (a) the hydrodynamic model and by (b) LTRANS-Zlev using the depth interpolation option.

triangle containing the particle. With this approach, the particles will rise whenever they encounter bathymetry elevations (blue sloping surfaces in Fig. 4(b)); conversely, they will experience reflections along the lateral boundaries (red walls in Fig. 4(b)) only in the upper layers, when moving across the continent or island borders. This configuration may be preferred for simulations in most oceanic basins characterized by gentle slopes or flat grounds, unlikely to retain particles.

When using the interpolated bathymetry, the depth of the bottom under some of the ρ, u, v and E nodes is subject to a re-computation (at step (*4) of Fig. 1) to reduce depth discontinuities.

LTRANS-Zlev offers the possibility to use either the first (non-interpolated) configuration represented in Fig. 3(b), or the second (interpolated) configuration represented in Fig. 4(b). If the user adopts the first non-interpolated approach on a Z-grid containing partial bottom cells, these partial cells are either converted into full cells or cancelled (masked as land points) by the model, depending on the relative height of the partial cell.

2.2.4. Application of the boundary conditions

The position of every particle is updated at every internal time step by computing the velocity associated to its former position and the related displacement within the integration time step (step (*16) of Fig. 1). The movement related to waves (stokes drift), wind (step (*17) of Fig. 1) and turbulence is added as well. At step (*19) of Fig. 1, the particle is reflected horizontally, whenever the trajectory between the previous position and the new position crosses any lateral-boundary of the vertical level associated to the earlier particle position.

2.3. Flow interpolation methods

In LTRANS v.2b, the interpolation routines of the flow field (used for example at step (*16)) are based on the identification of the 4 surrounding ρ nodes (u, v, w nodes for the velocity fields, respectively) forming a squared element containing the particle (e.g., the element formed by the white ρ nodes of coordinates (i,j), $(i,+1j)$, (i,j) and (2,2) in Fig. 2(a)). At every internal time step, optimized algorithms (searching in neighbouring elements) determine the (2-D, planar) element in which the particle is located. Then, among the 4 nodes composing the element itself, the 3 nodes that are closest to the particle are selected, defining a triangle containing the particle. The interpolation coefficients are computed depending on the position of the particle relatively to these 3 nodes. In a first phase, these coefficients are used to interpolate horizontally the water properties at every vertical level, to create a vertical water column profile at the particle horizontal position. In a second phase, the water properties are interpolated at the exact particle depth using 4 vertical levels (2 levels above, 2 levels below the particle) with the TSPACK tension spline curve fitting package (Renka, 1993).

Since the tension spline curve is computed by considering the interpolated values in each layer, for a Z-level mesh the different nodes to be considered for interpolations may have a different number of water layers in areas with uneven topography. Moreover, some areas might have less than 4 vertical water levels above the bottom. In the v.2b version, 4 vertical levels are systematically used for the vertical profile, while in the Zlev version (Z-level mesh) the tension spline is computed only if at least 3 vertical levels are present at the particle position (using at least 3 levels). In case of only 2 vertical levels, a linear interpolation is performed. Otherwise, if the particle is above the last ρ cell-centre, or in case there is only one vertical level above the bottom, the water properties are assumed to be those of the upper level, instead of being linearly interpolated. Besides, in the Zlev version the user can choose to deactivate the tension spline curve fitting and use instead a simple linear interpolation to compute the water properties at the exact depth of the particle.

On Z-grids, in areas of varied topography, the fact that the different nodes to be considered for interpolations have a different number of water layers can affect also the particles transported near the bottom. In fact, such situations might lead to erroneous interpolation of the water properties near the bottom, where the zero-values stored in land nodes might be used to interpolate properties such as current velocities, temperature and salinity. The problem is solved by (fictitiously) assigning to each land node the water properties of its closest water nodes, giving the properties of the upper node to the first land node under the bottom, or averaging the properties of the (water) horizontal neighbours for deeper land nodes. This procedure is applied every time a new hydrodynamic field is loaded, to the quantities stored at the ρ nodes and to the horizontal current velocities stored along every lateral boundary. In a second step, to account for the boundary layer viscous effects, as described in section 2.4.6, the model offers the possibility to parameterize the decrease of the speed of the particles, which are slowed down as they approach the bottom.

2.4. Additional features

2.4.1. Oil spill weathering and effects of wind and Stokes drifts

LTRANS-Zlev integrates also the OILTRANS oil spill weathering module developed within LTRANS (Berry et al., 2012), together with the effects of wind forcing on oil slicks. The wind forcing used to calculate the drift of the particles is two-dimensional, varying on each grid element and interpolated in space for every particle. In the Zlev version, the wind intensity used to compute the oil weathering processes of the OILTRANS module is calculated at every time-step, by averaging (over all the particles of the simulation) the value of the wind intensity interpolated at every particle position. At every time-step, the average (over the particles) of the water temperature and water column height are calculated in the same way and used to compute the weathering effect. The user can choose which percentage of the (instantaneous in time, but spatially averaged over all the particles) wind intensity will be used for the weathering processes. As the weathering is highly responsive to wind, this parameterization allows a better calibration of its intensity, which is useful if we want to take into account and evaluate the

effects of the temporal resolution of the forcing fields. In fact, wind forcing with high (or low) temporal resolution might show peaks of wind intensity (or a relatively smooth variability, respectively). In this way, the user can adapt the wind intensity term, if the calibration of the various processes involved in the weathering has been done for wind fields sampled at a different temporal resolution. Furthermore, in the Zlev version, wind forcing can be enabled even when the oil spill module is turned off. This feature was not included in the original LTRANS, but it is quite useful to model wind effects on the dispersion of inertial floating material, for objects with large portions floating above the sea level and consequently directly exposed to wind drag, such as macroplastics. Additionally, the amplitude of the wind and Stokes drift as well as the deviation angle of the wind drift to the right-hand side of the wind vector can now be customized by the user.

2.4.2. Tracking particles backward-in-time

Backward-in-time computations (negative time step) are possible in LTRANS-Zlev by reading the hydrodynamic files in reverse order, that is from the last one to the first one, and using a negative time-step to reverse the direction of the advection fields. This feature can be used when the goal of the simulation is the identification of the source of the transported particles, in order to estimate, for example, the release position of a pollutant, or the coordinates of an accident, when people or objects are found lost at the sea.

2.4.3. Stranding and settlement

In addition to the possibility of simulating particle settlement by defining settlement polygons (already included in LTRANS v.2b), LTRANS-Zlev parameterizes also particle stranding, by defining the distance from the coast within which a particle must be considered as stranded. The user can also choose a preferential depth range for stranding or settlement. These options increase the versatility of the model and allow to simulate a wider range of larval species, which follow different settling scenarios.

2.4.4. Particle release from the bottom layer

Another new possibility offered by LTRANS-Zlev is the release of particles at a given distance (which can be customized) above the seabed. Within this option, the user must specify only the horizontal positions where the particles should be released and LTRANS will compute for every particle the exact depth of its release point, following the shape of the bathymetry. This feature is useful, for example, when simulating the dispersal of larvae released by organisms living close to the bottom.

2.4.5. Larval behaviour with diel vertical migration and transport along the seabed

In LTRANS-Zlev, an option allows the user to entirely customize the parameters of a new larval scenario representative of the behaviour of the larvae of several species: the particles are released close to the seabed, then rise towards the surface and perform diel vertical migration, after which they sink to the bottom to be transported along the seabed.

By selecting this new behaviour, the user can setup diel vertical migration (DVM) by linking in the input file the name of a data-file containing time varying, spatially average, short wave downward (SWD) radiation data for the computational domain, and setting threshold values for the upward/downward swimming phases of the larvae. In other words, to compute the temporal variability of the target depth range of the particles, LTRANS-Zlev reads the space-averaged SWD radiation file extracted from the Eulerian model and calculates the instantaneous SWD radiation intensity. Then, the model checks if the value is higher than the sunrise threshold or lower than the sunset threshold. Depending on the current SWD radiation, LTRANS-Zlev determines whether the target layer of the particles is the daily one or the nightly one. Both threshold values for sunrise and sunset, as well as the vertical swimming speeds and the depth of the day/night target layers can be entirely set by the user, together with the length of the DVM period.

When the simulated time exceeds the DVM period, the particles sink to the bottom and are transported along the seabed, remaining within a target layer. The maximum height above the bottom of this target layer can be set by the user.

At any moment, the particles that are located outside their target layer range (due to the advection by the currents, or when going from one target layer to another) are pushed back towards it, using a vertical component of swimming speed which is superimposed to the hydrodynamic transport.

The development of this new behaviour option widens the applicability of the new version of LTRANS. The possibility to customize its parameters makes it suitable for modelling different species in the same environmental conditions. The Zlev version still includes the larval scenarios already implemented in the LTRANS v.2b, which were specifically parameterized to model *C. Virginica* and *C. Ariakensis* oysters. For other species, near-surface or near-bottom transport and diel vertical migration (based on time since midnight, instead of short wave downward radiation fields) can still be modelled as single behaviours maintained for the whole simulation, as it was already implemented in LTRANS v.2b.

2.4.6. Advection velocity in the boundary layer

The height of the bottom boundary layer and the advection velocity of any particle entering it can be set by the user. Within the selected boundary layer, the advection of particles transported near the bottom is reduced by applying either the logarithmic law-of-the-wall (already implemented in the v.2b version), or a percentage of the ambient flow velocity.

2.4.7. Diagnostic outputs along particle paths

If set by the user, when printing the output files containing particle paths, LTRANS can extract instantaneous flow field properties, which are critical for larval behaviour, such as temperature and salinity, at every particle position. The Zlev version offers also the possibility to extract the minimum or maximum values (besides the instantaneous values, also available in LTRANS v.2b) encountered by the particles since the last output of the Lagrangian model. Moreover, in LTRANS Zlev, a sediment grain size map, defined on the same grid as the hydrodynamic model, can be loaded. Then, the instantaneous, minimum or maximum values of the grain size encountered by the particle since the last extraction can be written, together with the temperature and salinity values. Finally, the presence of a particle within a polygon can be tracked independently from any settlement behaviour, by storing at every instant the identification number of the polygon containing the particle.

All these options are useful to run, using a single LTRANS simulation, multiple post-processing scenarios, in which the death or settlement of particles representing larvae depend on the encountered grain-size or water properties, when passing through the settlement polygons. The option of running multiple post-processing scenarios is of particular importance when dealing with the sensitivity analysis of the model parameterization, which is always needed and recommended when modelling biologically driven processes (Gibson and Spitz, 2011), such as in larval connectivity studies.

A short user manual accompanies the code in the repository and presents the flags that the user can set to activate the various options described in this section.

3. Results and discussion

3.1. Code optimization, parallelization, and other computational aspects

The LTRANS-Zlev code is written in FORTRAN 90. The main time-

demanding loop of the present version has been parallelized with OPEN-MP directives (see Fig. 1). The parallelization of other sections of the code, like the I/O sections or the oil module, is not implemented yet, but may be included in future versions. The choice of OPEN-MP for a first parallel implementation was guided by the fact that this approach parallelizes the instructions, while keeping the memory shared among the parallel threads. This allows a balanced distribution of the workload among the threads, while limiting the communication costs. This kind of parallelization could be further enhanced by an additional MPI parallelization that could allow the increase of the memory available for the simulation by using multiple nodes, but -besides the obvious implementation cost- this might lead to a sub-optimal use of the resources. In fact, by using the domain decomposition, the workload might become unequally divided among the processes, when the distribution of the particles is not uniform over the domain and tends instead to concentrate in specific areas. The option to parallelize with MPI using other memory partition schemes is not straightforward, andwould in any case require a complex implementation to limit the time consuming communications among the processes, since every particle needs to access different portions of the hydrodynamic and grid matrixes at every single time-step of the simulation.

Unlike the v.2b version, LTRANS-Zlev can read the input parameters from files (the name of which can be selected by the user) that are provided, as an input argument, to the software executable. Similarly, the output files generated by the model are named and identified by means of a prefix, which must be specified by the user in the input parameter file. These upgrades allow to simplify the simultaneous run of multiple independent simulations, giving the possibility to run multiple executions from the same directory. In this way, the user can change some parameters of the simulations (for example, release time or position, random seed, larval behaviour options, oil spill weathering properties, etc…) to perform sensitivity studies or to compute ensemble statistics. To reduce the time needed to initialize the model, the user can choose to store the matrixes of adjacent elements during the first run on a new domain, and then re-use this file instead of re-computing the adjacent elements at every new run on the same domain.

The simulations performed in the benchmark presented in Table 1 used these features to run multiple independent simulations launched simultaneously, by means of distinct job submissions. Every job occupied part or the total number of cores of the node, nevertheless, the whole node had been reserved for the simulation.

Table 1 presents the computational resources required by a simulation running on a computational node composed by two x86 64 Intel(R) Xeon(R) CPUs, with a total of 20 cores per node. Namely, the CPU/socket model is E5-2680 v2 @ 2.80 GHz (Turbo Speed 3.60 GHz), with 10 cores for each socket, 25 MB Cache and 62 GB of RAM available. We used gfortran GCC (version 4.8.2) to compile LTRANS-Zlev, together with the hdf5 and NetCDF libraries.

Five groups of simulations are presented in Table 1: they are identified by the letters A, B, C, D, E and are characterized by different Lagrangian parameters. The particles in the Lagrangian simulations do not interact with each other, hence the modelling of large amounts of particles can be split into several simultaneous simulations for various sub-groups of particles. Within each group, Table 1 compares the computational cost for several configurations of simulations, including: multiple simultaneous independent sequential simulations (1, 5 or 20 simultaneous runs) identified by the SEQ label, a single OPEN-MP simulation using 20 threads (OMP-*1), and multiple (5, 7 or 10

Table 1
Resources required by LTRANS-Zlev coupled offline to the MITgcm model. The blue parameters highlight the differences with the reference "A" case. Colour-scale varying from red to green show the efficiency (from low to high) of the simulations.

Simulation name	Computational configuration		Simulation parameters			Resources for the whole ensemble of simultaneous runs	
	CPU(s) = threads per run × N° of runs	N° of particles per run × N° of runs	Total number of particles	MITgcm grid size	Simulated time	Memory used (Gb)	Real time (min)
SEQ-A1	1 × 1	2000×1	2000	4×10^6	5 days	2.4	15.5
SEQ-A2	1 × 5	400×5				12	6.4
SEQ-A3	1 × 20	100×20				48	5.30
OMP-A1	20× 1	2000×1				2.4	3.9
OMP-A2	4 × 5	400×5				12	1.75
OMP-A3	2 × 10	200×10				48	3.2
SEQ-B1	1 × 1	20 000×1	20 000	4×10^6	5 days	2.4	134.0
SEQ-B2	1 × 5	4000×5				12	31.5
SEQ-B3	1 × 20	1000×20				48	10.5
OMP-B1	20× 1	20 000×1				2.4	20.5
OMP-B2	4 × 5	4000×5				12	10.5
OMP-B3	2 × 10	2000×10				48	10.7
SEQ-C1	1 × 1	2000×1	2000	11.5×10^6	5 days	6.9	24.5
SEQ-C2	1 × 5	400×5				34	15.5
OMP-C1	20 × 1	2000×1				6.9	10.0
OMP-C2	4 × 5	400×5				34	11.6
SEQ-D1	1 × 1	20 000×1	20 000	11.5×10^6	5 days	6.9	140
SEQ-D2	1 × 5	4000×5				34	38
SEQ-D3	1 × 7	2857×7				48	28.5
OMP-D1	20 × 1	20 000×1				6.9	24
OMP-D2	4 × 5	4000×5				34	19
OMP-D3	2 × 7	2857×7				48	20
SEQ-E1	1 × 1	20 000×1	20 000	11.5×10^6	30 days	6.9	852
OMP-E1	20 × 1	20 000×1				6.9	153

simultaneous, independent runs) simultaneous OPEN-MP simulations using 4 or 2 threads each.

The parameters differentiating the groups B, C, D and E from the reference case (A) are identified in blue in Table 1. The parameters of the Lagrangian model highlighted in blue are those whose values have been increased to analyse their respective influence on the memory and the time requirements of the simulations. Two distinct grid sizes (about 4×10^6 and 11.5×10^6 elements) were tested, and each set of simulations considered a total of either 2000 or 20000 particles, tracked for 5 or 30 days. Group A is characterized by light simulations using the small grid (4×10^6 elements) and running for only 5 days and 2000 particles. As highlighted in blue in Table 1, the simulations of group B and C differ from those of group A by, respectively, a higher number of particles and a larger number of grid elements. The simulations of group D present both a high number of particles and a large number of grid-elements, while the simulations of the group E are the most demanding ones, differing from those of group D by the length of the simulated time period: 30 days instead of 5.

The results of the comparative analysis between simulations characterized by various hardware or parallel configurations and Lagrangian parameterizations is presented in the last three columns of Table 1. The memory and time consumption of the simulations are highlighted by a colour-scale varying from red to green to identify low to high efficiency of the simulations, in terms of memory and runtime. In particular, we found that:

- For the test-cases presented in Table 1, the memory required by the model is directly proportional to the size of the Eulerian grid (multiplied by the number of simultaneous runs), and it is not impacted by the number of particles to be simulated. For a higher number of particles, the required memory would necessarily increase: it is estimated that a maximum of 500 Bytes of memory space is required per particle, so that 2 million particles would need 1 GB of memory.
- For large grids, the memory cost limits the number of simulations that can be run simultaneously. We choose, for example, a maximum of 7 simultaneous runs for the 11.5×10^6 elements domain. The OPEN-MP parallelization allows a reduction of the computational time without any increase of the memory usage (for example, OMP-E3 has the same memory use as the SEQ-E3 simulation).
- Either in sequential or in parallel mode, unless a very few particles are released, it is preferable to run a few simultaneous simulations (for heavy simulations, as much as it can be allowed by the available memory) and reduce the number of particles per run. Then, the use of the OMP parallelization, with an accurate number of threads, reduces the time requested by the simulations.
- The time required for the simulation (real time) can be minimized by choosing the optimal balance between the number of simultaneous simulations and the number of parallel threads. In fact, Table 1 shows that, given the particular computational node configuration and Lagrangian parameters selected in this study, the most efficient solution is to run 5 OMP simultaneous simulations using 4 threads each (simulations OMP [-]2). Conversely, running either multiple sequential simulations (SEQ [–]), or a single simulation with 20 parallel threads (OMP [-]1) or running 7 (or 10) simultaneous simulations with 2 threads (OMP [-]3) is always more time-consuming. Modelling the whole group of particles in a unique sequential simulation (SEQ [-]1) is by far the most demanding solution and must be avoided.

3.2. Validation of the LTRANS-Zlev implementation

We verified the absence of setup and coding errors in LTRANS-Zlev by running a comparison between the trajectories computed by the native online virtual float tracking module included in the MITgcm model and the particle trajectories computed by LTRANS-Zlev, driven by the same velocity fields provided by MITgcm.

MITgcm is a general circulation model that can solve either the hydrostatic or non-hydrostatic Navier-Stokes equations under the Boussinesq approximation for an incompressible flow. The non-hydrostatic capability allows MITgcm to simulate oceanic processes over a wide range of scales. The model is based on a finite volume discretization on a curvilinear horizontal grid, with vertical Z-coordinate levels and partial cells for the bottom layers.

Our test considered idealized flow conditions obtained by simulating a cyclonic gyre in a mid-latitude, closed basin (Cossarini et al., 2017). The circulation structure is controlled by the shape of the bathymetry (circular pit in a squared domain, Fig. 5) and the flow is in geostrophic equilibrium. The horizontal resolution of the hydrodynamic simulation is 1/128° for 256×256 grid points, so that the domain size is approximately 170×220 km in the zonal and meridional direction, respectively. There are 60 vertical layers, whose height varies from 1.5 m in the surface layer to about 60 m in the deepest region (in the centre of the basin). The model is forced by steady winds and a seasonal cycle of surface heat (downward long-wave and short-wave radiation) and mass (precipitation) fluxes. The horizontal shear in the surface wind field maintains a permanent cyclonic gyre, whereas the surface heat fluxes act on the thermohaline properties of the water column, inducing a yearly cycle (summer stratification – winter mixing). This simulation has been run for several years to reach steady-state conditions (perpetual year simulation).

In this study, the bathymetry used by LTRANS-Zlev considers the non-interpolated depth in order to have as similar as possible boundary conditions as the MITgcm particle tracking "flt" (i.e., float) package. Both the MITgcm hydrodynamic simulation and float module, as well as the LTRANS implementation, used a 100 seconds internal time discretization. The simulation tracked for a 5 days period the paths of 1302 particles released at 0.75 m depth on a uniform horizontal grid. To present a statistically relevant group of trajectories, the set of 1302 particles was released at a daily frequency for 30 consecutive days and every set was followed over a 5-days-long time interval. The abbreviation MITflt refers to the simulation of particle trajectories computed by the MITgcm online virtual float tracking module. The particle trajectories computed by LTRANS-Zlev were driven by the velocity fields provided by the MITgcm with an hourly frequency.

In order to evaluate the influence of the hydrodynamic fields driving the LTRANS particles, the LTRANS-Zlev simulations were performed six times, using different sets of MITgcm hydrodynamic fields. Initially, we tested whether it would be more accurate to use time averaged or

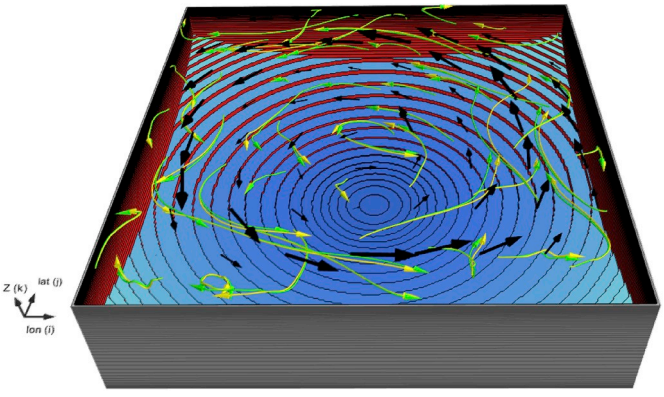


Fig. 5. Representation of the computational domain used in the squared and closed basin experiments. The path of some selected particles of a single set, all released at the same moment, are represented for the first two days of their advection by yellow and green lines for the LTR_hrAV and MITflt simulations, respectively. Black arrows indicate the direction and intensity of the time-averaged surface currents. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

instantaneous variables: the first LTRANS-Zlev case used a set of hourly-averaged hydrodynamic fields (identified as LTR_hrAV) while the second LTRANS-Zlev case was performed using a set of instantaneous states (snapshots) of the hydrodynamic field (identified as LTR_hrSN), still with hourly frequency. We also performed a third, fourth and fifth experiment using hydrodynamic fields delayed by one time-step, 1 hour and 24 hours (identified as LTR_hrAV+dt, LTR_hrAV+1hr, LTR_hrAV+1day, respectively), in order to analyse the differences between the trajectories of particles released in the same point but at different moments, and therefore advected by different hydrodynamic fields. We finally performed a last test using the output velocity fields dumped at every time-step, in order to drive LTRANS with exactly the same velocity fields used by the MITgcm online flt module.

Fig. 5 represents the computational domain, the black arrows show the direction and intensity of the (time-averaged) surface currents, while yellow and green lines represent, for the LTR_hrAV and MITflt simulations, respectively, the paths during the first two days of advection of a selection of particles released at the same moment.

To evaluate the intensity of the transport, the average distance travelled by N particles is calculated through the following expression (where X_n^t, Y_n^t, Z_n^t is the position of the particle n at time t):

$$AD^{3d}(t) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t'=t_0}^{t}\sqrt{\left(X_n^t - X_n^{t-1}\right)^2 + \left(Y_n^t - Y_n^{t-1}\right)^2 + \left(Z_n^t - Z_n^{t-1}\right)^2}$$

The difference between the particle trajectories of the various groups of the two simulations was quantified by computing for every group the three-dimensional separation distance SD^{3d}, which is the average (over all of the N particles) distance between the two trajectories originated at the same time and starting from the same position:

$$SD^{3d}(t)[case\,a, case\,b] = \frac{1}{N}\sum_{n=1}^{N}\sqrt{\left(Xa_n^t - Xb_n^t\right)^2 + \left(Ya_n^t - Yb_n^t\right)^2 + \left(Za_n^t - Zb_n^t\right)^2}$$

The three-dimensional advection distance is represented by the thick grey dashed line in Fig. 6(b) and 6(c) for the MITflt particles only, but it was almost identical for each of the LTRANS cases (i.e., advection distances are very similar). Fig. 6b shows that the advection distance increases almost linearly in time, reaching about 50 km after two days (and 110 km after 5 days). As the domain dimensions span about 170 and 220 km in the zonal and meridional direction, respectively, the size of the gyre allows at least a few days of free transport, before the trajectory of the particles feel the presence of the solid boundaries, being diverted by them.

The evolution in time of the separation distances between the various sets of particles in the different experiments, after one and five days, are represented by dashed coloured lines in Fig. 6b and c, respectively. Fig. 6 (a) represents with orange continuous lines the distance between the particle position computed by LTRANS-Zlev (LTR_hrAV simulation) and the corresponding particle position computed by the MITgcm virtual float module at the end of each of the 5 days of advection. The orange dashed lines in Fig. 6(b) and c shows the evolution of the separation distance in time, which increases almost exponentially, reaching about 15 km after 5 days of simulation (O(10%) of the total advection

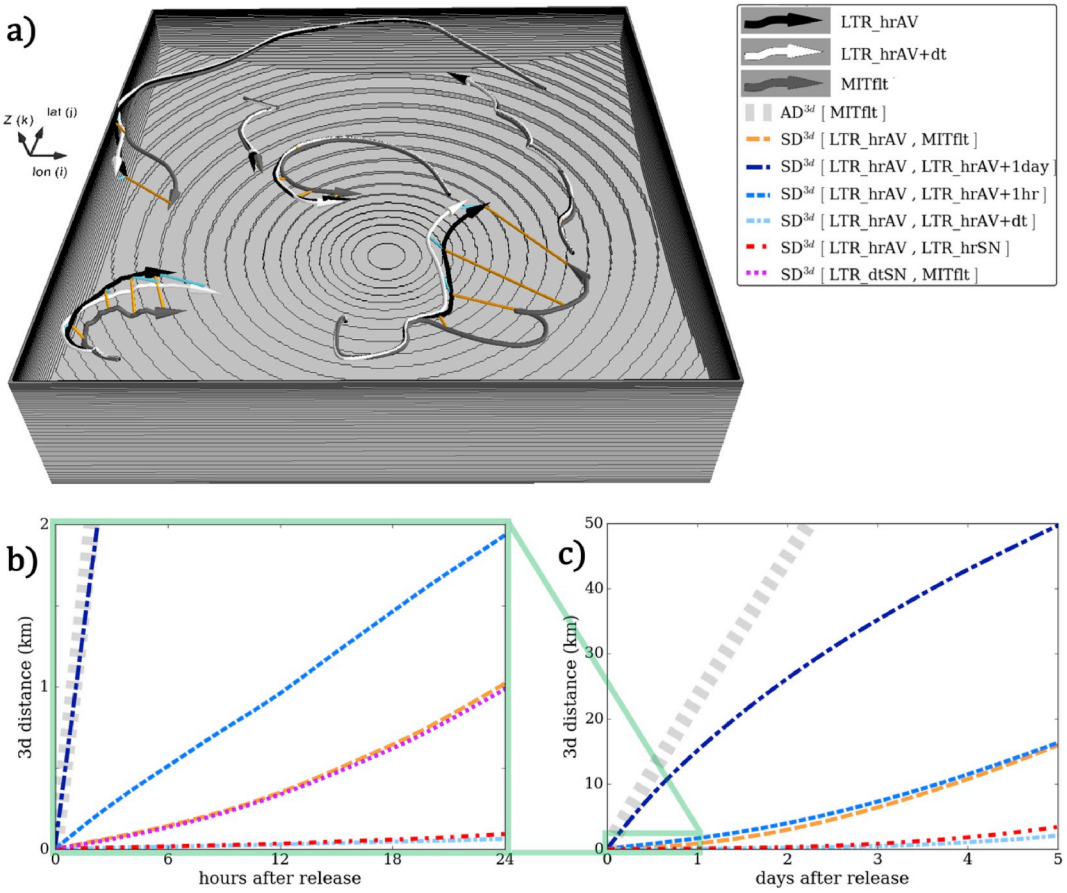


Fig. 6. Three dimensional view of a subset of trajectories (LTR_hrAV, LTR_hrAV+dt and MITflt) and the associated separation distance at the end of each of the 5 simulated days (a). Time series of the advection distance of the MITflt particles (grey dashed lines) and time series of the three-dimensional separation distance between various groups of simulations during 1 (b) and 5 (c) days (coloured lines). A detailed reference is reported in the legend. Continuous lines in plot (a) show the instantaneous 3D distance between particles at selected time intervals. The corresponding dashed lines in plot (b) and (c) show the time evolution of those separation distances.

distance).

We tested the sensitivity of LTRANS to the different hydrodynamic fields by comparing the trajectories computed by using hourly averaged advection fields delayed by one time-step, 1 hour and one day (cyan, light blue and dark blue lines in Fig. 6, respectively). A delay of only one time-step (i.e., 100 s) gives the non-negligible separation distance (cyan lines in Fig. 6) of almost 2 km in 5 days (4320 timesteps), highlighting the turbulent and chaotic features of the system. A 1-hour-delay gives similar results as the comparison between LTRANS hourly averages and MITflt trajectories. A delay of one day produces completely different and strongly divergent trajectories, reaching a separation distance of 50 km in 5 days.

Also the use of instantaneous (LTR_hrSN) or averaged (LTR_hrAV) hourly advection fields give rise to a non-negligible separation distance between LTRANS trajectories (red dashed lines in Fig. 6(b) and (c)): 3 km after 5 days.

The outcomes of the last test, performed by using the MITgcm output velocity fields dumped at every time-step (purple dashed line in Fig. 6 (b)), show that the results are comparable with the hourly averaged test case (LTR_hrAV), suggesting that the separation distance between LTRANS-Zlev and MITflt particles is independent from the time averaging of the MITgcm velocity fields used to drive the LTRANS simulations, at least up to 1 hour. This test was performed for only one day, due to the large dataset made up by the MITgcm hydrodynamic output files dumped at every time-step.

In light of these considerations, and given that both tracer transport modules use the same 4^{th} order Runge-Kutta scheme for particle advection, the difference between LTRANS-Zlev and MITflt trajectories must be ascribed to the different interpolation methods of the velocity fields. In our specific case, and differently from LTRANS-Zlev (see Subsection 2.3), the 3D advection of passive particles in MITgcm is obtained by interpolating the velocity fields using a trilinear scheme (Adcroft et al., 2017).

Overall, the results prove the effectiveness of the new implementation and the satisfactory accuracy of the advection algorithm.

3.3. A realistic test-case

In order to illustrate and check the implementation of some additional features of this new release of the transport model, we performed a simulation focused on the northern Adriatic Sea, reproducing specific larval vertical behaviours and using the interpolated bathymetry configuration of LTRANS-Zlev. The new larval behaviours were implemented in LTRANS-Zlev in the framework of the MANTIS project: a complete description of the experiments is presented in M. Canu et al. (*Nephrops norvegicus* connectivity model to support the design of EFH in the Adriatic Sea. (in preparation)), where *Nephrops Norvegicus* larval dynamics and connectivity in the whole Adriatic are studied. In our test case, the grids, daily hydrodynamic fields and hourly space-averaged short-wave downward (SWD) radiation data were provided by the MITgcm implementation characterized by a 1/64° horizontal resolution. Fig. 7 shows the LTRANS-Zlev interpolated bathymetry together with a representation of the paths of a few particles characterized by specific larval vertical behaviour. The particles are initially released close to the seabed, then an upward movement is superimposed to the hydrodynamic transport, so that particles reach the upper layers, where they start a diel vertical migration for 8 days, before sinking back to the bottom. Afterwards, they are transported along the seabed for further 12 days. During the DVM phase, the depth range of the target layers of the particles is [0–5] meters during the night, and [25–30] meters during the day. The values of the SWD radiation thresholds were chosen analysing the time evolution of the SWD radiation for the selected time period (which depends on the cloud coverage as well as dawn and sunset timetables). After the end of the DVM phase, the particles sink to the bottom and are transported within a 1-m-thick layer above the seabed, with a velocity reduced by 90% to account for the bottom viscous

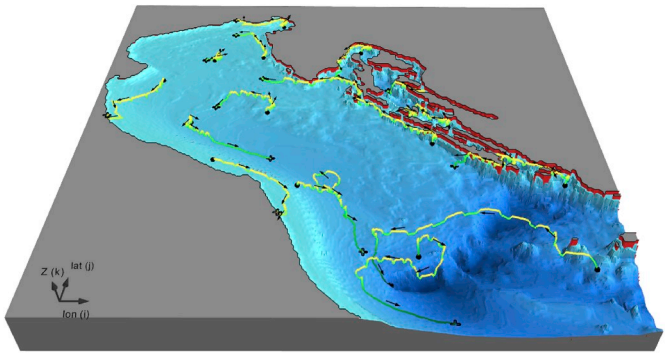


Fig. 7. Representation of the northern Adriatic computational domain used by LTRANS-Zlev, with the interpolated bathymetry option. The paths of the particles are represented by lines with a colour-scale varying from yellow, when the particles are at surface, to dark green for deeper positions. Black arrows indicate the transport direction along the path of the particles, while the release and settlement positions are respectively identified by black dots and crosses. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

boundary effects. In fact, given the absence of precise information regarding the height of the boundary layer or the height above the bottom at which larvae are transported, it may be preferable to set the advection velocity of the particles to a percentage (10% in this case) of the velocity of the currents calculated with the vertical tension spline described in section 2.3.

As expected, the global transport pattern shown in Fig. 7 highlights the typical dynamics of the Adriatic Sea, governed by a basin-scale cyclonic circulation. It appears that during the DVM phase the particles perform longer advection distances when floating close to the surface than in the deeper target layer. The reason lies in the higher intensity of the surface currents with respect to the deeper ones. The effect of the wind is also included in the hydrodynamic flow fields provided by the MITgcm, even though this LTRANS-Zlev simulation was not parametrized to include additional wind forcing on the particles.

This simulation, which includes biologically determined vertical behaviour, would not have been possible in the framework of the built-in MITgcm lagrangian modules, nor in the standard LTRANS v.2b version, that does not allow the user to customize all the described parameters or to set light-thresholds using the shortwave downward radiation fields.

4. Conclusions

This paper presents LTRANS-Zlev, a Lagrangian model working off-line by reading bathymetry and flow fields of either σ or Z-coordinate hydrodynamic models. LTRANS-Zlev tracks marine particles dispersed by oceanographic currents and winds, and can model oil spill transport and weathering through the OILTRANS module. Furthermore, LTRANS-Zlev incudes additional features that were not present in LTRANS v.2b, among which the possibility to customize larval behaviour and environmental dependent behaviour of any transported agent, such as vertical swimming speed, temperature dependent survival and particle growth, time dependent age, diel vertical migration duration and light-related thresholds. The Lagrangian tracking capabilities have been validated by a comparison with the results from the internal MITgcm float module, and model features have been tested and exemplified in both an idealized case study and a more realistic setup. The model runs in parallel on platforms using OpenMP, also in sequential mode on standard workstations, but it requires a computational power that is proportional to the domain size.

The possibility to work with either σ or Z-coordinates increases the versatility of this new version of LTRANS, as the choice of the vertical discretization of the hydrodynamic equations depends on many factors,

such as the topography of the basin, or the kind of processes to be investigated. Moreover, LTRANS-Zlev allows the use of either a smooth interpolated bathymetry, favourable to particle horizontal advection, or a sharp bathymetry made by vertical cliff edges, which may be preferred for simulations in basins presenting sub-marine canyons or vertical underwater structures, such as dikes or oil and gas extraction terminals. In comparison with on-line tracking tools, LTRANS-Zlev allows a remarkable reduction of the computational costs when performing multiple experiments or ensemble simulations, or when adopting high-resolution discretizations in time or space.

5. Software availability

The LTRANS-Zlev model is written in FORTRAN 90 and parallelized with OPEN-MP directives. The code is available on the repository http://github.com/inogs/LTRANS_Zlev since 2019. The software needs as an input the hydrodynamic fields and grid files of an Eulerian model based on Arakawa-C grids (e.g., MITgcm, ROMS, etc…). LTRANS-Zlev does not require particular hardware configurations; its memory usage is proportional to the size of the mesh of the Eulerian input files. The software can be used either in sequential or parallel mode, and requires the hdf5, NetCDF and OPEN-MP libraries. As the original v.2b version, the LTRANS-Zlev is freely available under an MIT/X licence and includes the Mersenne Twister random number generator and the tension spline curve-fitting package (TSPACK) which have their own licenses and are freely available for non-commercial uses.

Declaration of competing interest

This work is original and has not been published, nor is it currently under consideration for publication elsewhere. We also have no conflicts of interest to disclose. After publication the code will be made public. Author of the original version of LTRANS have been already informed about our extension, and already sent us a positive feedback.

Acknowledgments

Author contributions: CL designed and developed the Z-discretization and parallelization of the code with contributions from CS and SQ. DMC defined the new behaviour added to the larval model and its parameterization and CL implemented them, together with the updates of the OILTRANS module. CL performed the Lagrangian simulations using LTRANS-Zlev and SQ performed the Eulerian simulations with the MITgcm and the flt-package particle tracking. CL and CS wrote the manuscript with contributions from the other authors.

Funding: this work was partially funded by the project EU-MANTIS – DG MARE/2014/41 under the theme “Marine protected areas: network (s) for enhancement of sustainable fisheries in EU Mediterranean waters”, by MISE, Italian Ministry of Economic Development in the frame of the “Program Agreement on Research on natural and anthropogenic risks for security and protection of offshore activities. Modeling marine hydrocarbon dispersal”, and by the HPC-TRES.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2020.104621.

References

Adcroft, A., Campin, J.-M., Dutkiewicz, S., Constantinos, E., Ferreira, D., Forget, G., Fox-Kemper, B., Heimbach, P., Hill, C., Hill, E., Hill, H., Jahn, O., Losch, M., Marshall, J., Maze, G., Menemenlis, D., Molod, A., 2017. MITgcm User Manual. Intern. Doc.

Berry, A., Dabrowski, T., Lyons, K., 2012. The oil spill model OILTRANS and its application to the Celtic Sea. Mar. Pollut. Bull. https://doi.org/10.1016/j.marpolbul.2012.07.036.

Bleck, R., Rooth, C., Hu, D., Smith, L.T., 2002. Salinity-driven thermocline transients in a wind- and thermohaline-forced isopycnic coordinate model of the north atlantic. J. Phys. Oceanogr. https://doi.org/10.1175/1520-0485(1992)022<1486:sdttia>2.0.co;2.

Blumberg, A., Mellor, G., 1987. A description of a three-dimensional coastal ocean circulation model, three-dimensional coastal ocean models. Coast Estuar. Sci. 4 https://doi.org/10.1029/CO004p0001.

Canu, D.M., Solidoro, C., Bandelj, V., Quattrocchi, G., Sorgente, R., Olita, A., Fazioli, L., Cucco, A., 2015. Assessment of oil slick hazard and risk at vulnerable coastal sites. Mar. Pollut. Bull. 94 (1–2), 84–95. https://doi.org/10.1016/j.marpolbul.2015.03.006.

Chassignet, E.P., Smith, L.T., Bleck, R., Bryan, F.O., 1996. A model comparison: numerical simulations of the north and equatorial atlantic oceanic circulation in depth and isopycnic coordinates. J. Phys. Oceanogr. 26, 1849–1867. https://doi.org/10.1175/1520-0485(1996)026<1849:amcnso>2.0.co;2.

Cossarini, G., Querin, S., Solidoro, C., Sannino, G., Lazzari, P., DI Biagio, V., Bolzon, G., 2017. Development of BFMCOUPLER (v1.0), the coupling scheme that links the MITgcm and BFM models for ocean biogeochemistry simulations. Geosci. Model Dev. (GMD). https://doi.org/10.5194/gmd-10-1423-2017.

Cowen, R.K., Sponaugle, S., 2009. Larval dispersal and marine population connectivity. Annu. Rev. Mar. Sci. 1, 443–466.

Gibson, G.A., Spitz, Y.H., 2011. Impacts of biological parameterization, initial conditions, and environmental forcing on parameter sensitivity and uncertainty in a marine ecosystem model for the Bering Sea. J. Mar. Syst. 88 (2), 214–231. https://doi.org/10.1016/j.jmarsys.2011.04.008.

Griffies, S.M., Böning, C., Bryan, F.O., Chassignet, E.P., Gerdes, R., Hasumi, H., Hirst, A., Treguier, A.-M., Webb, D., 2000. Developments in ocean climate modelling. Ocean Model. https://doi.org/10.1016/s1463-5003(00)00014-7.

Griffies, S.M., Harrison, M.J., Pacanowski, R.C., Rosati, A., 2004. A technical guide to MOM4. GFDL Ocean Gr. Tech. Rep.

Haney, R.L., 1991. On the pressure gradient force over steep topography in sigma coordinate ocean models. J. Phys. Oceanogr. 21, 610–619. https://doi.org/10.1175/1520-0485(1991)021<0610:otpgfo>2.0.co;2.

James, I.D., 2002. Modelling pollution dispersion, the ecosystem and water quality in coastal waters: a review. Environ. Model. Softw 17 (4), 363–385. https://doi.org/10.1016/S1364-8152(01)00080-9.

Lange, M., Sebille, E. Van, 2017. Parcels v0.9: prototyping a Lagrangian ocean analysis framework for the petascale age. Geosci. Model Dev. (GMD). https://doi.org/10.5194/gmd-10-4175-2017.

Lett, C., Verley, P., Mullon, C., Parada, C., Brochier, T., Penven, P., Blanke, B., 2008. A Lagrangian tool for modelling ichthyoplankton dynamics. Environ. Model. Softw. https://doi.org/10.1016/j.envsoft.2008.02.005.

Lynch, D., Greenberg, D.A., Bilgili, A., Mcgillicuddy, D., Manning, J., Aretxabaleta, A., 2014. Particles in the Coastal Ocean: Theory and Applications. https://doi.org/10.1017/CBO9781107449336.

Madec, G., the NEMO Team, 2016. NEMO ocean engine. Note du Pôle modélisation. https://doi.org/10.1016/j.joms.2014.06.438.

Marshall, J., Adcroft, A., Hill, C., Perelman, L., Heisey, C., 1997a. A finite-volume, incompressible Navier Stokes model for, studies of the ocean on parallel computers. J. Geophys. Res. C Ocean. https://doi.org/10.1029/96JC02775.

Marshall, J., Hill, C., Perelman, L., Adcroft, A., 1997b. Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling. J. Geophys. Res. C Ocean. https://doi.org/10.1029/96JC02776.

Mellor, G., 2003. Users guide for A three-dimensional, primitive equation. Numer. Ocean Model.

Mellor, G., Ezer, T., Oey, L.-Y., 1994. The pressure gradient conundrum of sigma coordinate ocean models. J. Atmos. Ocean. Technol. - J ATMOS Ocean TECHNOL 11, 1126–1134. https://doi.org/10.1175/1520-0426(1994)011<1126:TPGCOS>2.0.CO;2.

North, E.W., Hood, R.R., Chao, S.Y., Sanford, L.P., 2006. Using a random displacement model to simulate turbulent particle motion in a baroclinic frontal zone: a new implementation scheme and model performance tests. J. Mar. Syst. https://doi.org/10.1016/j.jmarsys.2005.08.003.

North, E.W., Adams, E.E., Schlag, Z., Sherwood, C.R., He, R., Hyun, K.H., Socolofsky, S. A., 2011. “Simulating Oil Droplet Dispersal from the Deepwater Horizon Spill with a Lagrangian Approach” in Monitoring and Modeling of the Deepwater Horizon Oil Spill: A Record-Breaking Enterprise. Amer. Geophys. Union, Washington, DC, pp. 217–226.

North, E.W., Adams, E.E., Schlag, Z., Sherwood, C.R., He, R., Hyun, K.H., Socolofsky, S. A., 2013. Simulating oil droplet dispersal from the deepwater horizon spill with a Lagrangian approach. Monit. Model. Deep. Horiz. Oil Spill A Rec. Break. Enterp. 195, 217–226. https://doi.org/10.1029/2011GM001102.

Paris, C.B., Helgers, J., van Sebille, E., Srinivasan, A., 2013. Connectivity Modeling System: a probabilistic modeling tool for the multi-scale tracking of biotic and abiotic variability in the ocean. Environ. Model. Softw. https://doi.org/10.1016/j.envsoft.2012.12.006.

Pope, S.B., 1985. Lagrangian modelling for turbulent flows. In: Al, et, D.L.D. (Eds.), Theoretical Approaches to Turbulence. Springer, pp. 369–373. https://doi.org/10.1007/978-1-4612-1092-4_19.

Prants, S.V., Uleysky, M.Y., Budyansky, M.V., 2017. Lagrangian oceanography. https://doi.org/10.1007/978-3-319-53022-2.

Renka, R.J., 1993. Algorithm 716: TSPACK: tension spline curve-fitting package. ACM Trans. Math Software 19 (1), 81–94.

Schlag, Z.R., North, E.W., 2012. LTRANS Lagrangian TRANSport Model User ’ S Guide. Science vol.80.

Shapiro, G., Luneva, M., Pickering, J., Storkey, D., 2013. The effect of various vertical discretization schemes and horizontal diffusion parameterization on the performance of a 3-D ocean model: the Black Sea case study. Ocean Sci. https://doi.org/10.5194/os-9-377-2013.

Shchepetkin, A.F., McWilliams, J.C., 2009. Computational kernel algorithms for fine-scale, multiprocess, longtime oceanic simulations. In: Handbook of Numerical Analysis. https://doi.org/10.1016/S1570-8659(08)01202-0.

Taylor, G.I., 1922. Diffusion by continuous movements. In: Proceedings of the London Mathematical Society, pp. 196–212. https://doi.org/10.1112/plms/s2-20.1.196.

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., Deleersnijder, E., Döös, K., Drake, H.F., Drijfhout, S., Gary, S.F., Heemink, A.W., Kjellsson, J., Koszalka, I.M., Lange, M., Lique, C., MacGilchrist, G.A., Marsh, R., Mayorga Adame, C.G., McAdam, R., Nencioli, F., Paris, C.B., Piggott, M.D., Polton, J.A., Rühs, S., Shah, S.H. A.M., Thomas, M.D., Wang, J., Wolfram, P.J., Zanna, L., Zika, J.D., 2018. Lagrangian ocean analysis: fundamentals and practices. Ocean Model. https://doi.org/10.1016/j.ocemod.2017.11.008.

Werner, F.E., Cowen, R.K., Paris, C.B., 2007. Coupled biological and physical models present capabilities and necessary developments for future studies of population connectivity. Oceanography 20, 54–69.