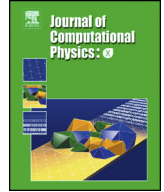




Contents lists available at ScienceDirect

Journal of Computational Physics: X

journal homepage: www.elsevier.com/locate/jcpx

The 3D Elliptical Parcel-In-Cell (EPIC) method

Matthias Frey^{a,*}, David Dritschel^{a,*}, Steven Böing^b^a Mathematical Institute, University of St Andrews, KY16 9SS, UK^b University of Leeds/Met Office Strategic Research Group, School of Earth and Environment, University of Leeds, LS2 9JT, UK

ARTICLE INFO

Article history:

Received 18 March 2023

Received in revised form 7 November 2023

Accepted 8 November 2023

Available online 15 November 2023

Keywords:

PIC

Elliptical parcels

Lagrangian methods

Density stratified flows

Turbulent flows

ABSTRACT

We present the three-dimensional version of the Elliptical Parcel-In-Cell (EPIC) method for the simulation of fluid flows and analogous continuum systems. The method represents a flow using a space-filling set of ellipsoidal parcels, which move, rotate and deform in the flow field. Additionally, parcels may carry any number of attributes, such as vorticity, density, temperature, etc, which generally evolve in time on the moving parcels. An underlying grid is used for efficiency in computing the velocity field from the interpolated vorticity field, and in obtaining parcel attribute tendencies. Mixing is enabled by permitting parcels to split when excessively deformed, and by merging very small parcels with the nearest other parcel. Several tests are provided which illustrate the behaviour of the method and demonstrate its effectiveness in modelling complex, buoyancy-driven turbulent fluid flows. The results are compared with a large eddy simulation (LES) and a direct numerical simulation (DNS) model.

© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In a previous paper, we developed the Elliptical Parcel-In-Cell (EPIC) method for two-dimensional (2D) flows, and applied it to complex density-stratified turbulent flows [1]. In particular, we showed that it can faithfully represent turbulent mixing without explicit diffusion or a subgrid mixing parametrisation. Unlike previous Particle/Parcel-in-Cell (PIC) methods, the EPIC method preserves the volume occupied by parcels in each grid cell over long integration times without ever needing to re-grid the parcels. Here, we present the corresponding three-dimensional (3D) method and apply it to a variety of test cases, including a moist buoyant bubble studied in [2,3].

PIC methods are hybrid Lagrangian–Eulerian numerical methods that use moving elements to represent certain flow properties (Lagrangian), and use a grid (Eulerian) for operations like constructing the velocity field. The grid also plays a role in computing the tendencies of certain parcel properties, such as vorticity. This hybrid approach combines accurate and efficient aspects of each representation (Lagrangian and Eulerian), and typically benefits from strong numerical stability without explicit stabilisers. A more detailed discussion can be found in [1]; here we focus specifically on the 3D extension of EPIC.

The plan of the paper is as follows. In Section 2, we discuss the flow model and provide an overview of the numerical approach. We then discuss the various new aspects of the 3D EPIC method in detail, some of which differ non-trivially from the 2D method in [1]. In Section 3, the method is then applied to several demanding test cases and compared with standard Eulerian (pseudo-spectral and LES) methods. Our conclusions are offered in Section 4.

* Corresponding authors.

E-mail addresses: mf248@st-andrews.ac.uk (M. Frey), david.dritschel@st-andrews.ac.uk (D. Dritschel).

2. Methodology

2.1. The flow model

We consider an inviscid incompressible fluid having variable density, under the Boussinesq approximation, and include a simple model for the condensation and evaporation of water vapour. This model was used previously by [2,3] to examine the evolution of an idealised cloud in the Moist Parcel-In-Cell (MPIC) method. The MPIC method uses non-deformable spherical parcels rather than ellipsoids, and differs in a number of other respects as discussed below. The flow model considered here is sufficiently complex to illustrate all of the key features of the EPIC method; we anticipate however that the method may be much more widely applicable, e.g. to geophysical and astrophysical fluid dynamics and beyond (see Section 4).

In a Cartesian coordinate system $\mathbf{x} = (x, y, z)$ with z pointing upwards, the prognostic (evolution) equations for velocity $\mathbf{u}(\mathbf{x}, t) = (u, v, w)$, liquid-water buoyancy $b_l(\mathbf{x}, t)$ and total water (i.e. water vapour plus liquid water) specific humidity $q(\mathbf{x}, t)$ (ignoring precipitation) are

$$\frac{D\mathbf{u}}{Dt} + 2\boldsymbol{\Omega} \times \mathbf{u} = -\frac{\nabla p}{\rho_0} + b \hat{\mathbf{e}}_z, \quad (1)$$

$$\frac{Db_l}{Dt} = 0, \quad (2)$$

$$\frac{Dq}{Dt} = 0, \quad (3)$$

where t is time, $DF/Dt = F_t + \mathbf{u} \cdot \nabla F$ is the material derivative of any quantity F (subscripts t, x, y and z denote partial differentiation), p is the pressure, ρ_0 is a constant reference density, $\boldsymbol{\Omega}$ is the (constant) background rotation vector, and $\hat{\mathbf{e}}_z$ is the vertical unit vector. These are supplemented by the relations

$$\nabla \cdot \mathbf{u} = 0, \quad (4)$$

$$b = b_l + b_c \max(0, q/q_0 - e^{-\lambda_c z}) \quad (5)$$

where λ_c is the reciprocal of the condensation scale height, q_0 is the surface saturation humidity, and b_c is a constant, in units of buoyancy (acceleration), proportional to the latent heat of condensation L_v . Specifically,

$$b_c = \frac{gL_v q_0}{c_p \theta_{v0}} \quad (6)$$

where g is the acceleration due to gravity, c_p is the specific heat at constant pressure and θ_{v0} is a constant reference liquid-water potential temperature. Note, the liquid-water buoyancy is defined by $b_l = g(\theta_l - \theta_{v0})/\theta_{v0}$ where θ_l is the (variable) liquid-water potential temperature.

The form of the total buoyancy b in (5) models the latent heating (increase in buoyancy) associated with condensation as a parcel rises above its condensation level $z_c(q) = \lambda_c^{-1} \log(q_0/q)$ (assuming the ground is at $z=0$), and likewise evaporative cooling when z falls below $z_c(q)$. Further discussion of the physical model can be found in [3].

We can eliminate the pressure p by instead evolving the vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, which satisfies

$$\frac{D\boldsymbol{\omega}}{Dt} = (2\boldsymbol{\Omega} + \boldsymbol{\omega}) \cdot \nabla \mathbf{u} + \nabla b \times \hat{\mathbf{e}}_z. \quad (7)$$

Notably, buoyancy gradients alter only the horizontal vorticity components since $\nabla b \times \hat{\mathbf{e}}_z = (b_y, -b_x, 0)$. In variables $\boldsymbol{\omega}$, b_l and $\tilde{q} \equiv q/q_0$, the flow model depends only on the parameters $\boldsymbol{\Omega}$, λ_c and b_c (besides domain-related parameters). In the simplest case of a dry, non-rotating stratified flow, we have $q=0$; then $b_l = b$ is the total buoyancy and the variable \tilde{q} is not needed.

We consider a domain which is periodic in both x and y , and bounded by free-slip surfaces in z . In general, $x_{\min} \leq x \leq x_{\max}$, and similarly for y and z . Along the z boundaries, the normal flow must vanish: if $\mathbf{u} = (u, v, w)$ in Cartesian coordinates, then $w=0$ at $z=z_{\min}$ and z_{\max} . There is no restriction on the values of $\boldsymbol{\omega}$, b_l and \tilde{q} on these boundaries – in particular, the boundaries are not in general shear-free [4]. Indeed, variations of buoyancy along the boundaries imply non-zero tendencies for the horizontal components of the vorticity, and hence non-zero values of u_z and v_z . However, since the boundaries are free-slip in the absence of viscosity, there can be no net generation of vorticity, implying that the domain integral of $\boldsymbol{\omega}$ remains constant in time (see [5,6] and references therein). This is trivially satisfied by the vertical vorticity component $\zeta = v_x - u_y$ on account of the periodic boundary conditions in x and in y (in fact the horizontal average of ζ is zero at each height z). For the other two components, constancy of their domain integrals implies that the difference between the values of the horizontal momentum on the two boundaries remains constant.

2.2. Vorticity inversion

The use of vorticity as a prognostic variable entails recovering the velocity \mathbf{u} by inverting the definition $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, subject only to homogeneous boundary conditions on the vertical velocity component w . In particular, one cannot generally assume any symmetry properties for the other field components when considering impermeable free-slip boundaries, and this significantly complicates matters. The procedure developed to find \mathbf{u} , detailed in [4] and summarised below, is based on a novel field decomposition that permits an arbitrary vertical dependence of fields near the boundaries.

Working in Cartesian coordinates, we take $\boldsymbol{\omega} = (\xi, \eta, \zeta)$ and $\mathbf{u} = (u, v, w)$. We consider a horizontally-periodic domain and Fourier transform all field components in x and y : $f(x, y, z, t) \rightarrow \hat{f}(k, l, z, t)$ where k and l are the x and y wavenumbers, respectively. Derivatives in x and y then become multiplications by ik and il in spectral space (here $i = \sqrt{-1}$). Since the relation $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is linear, the component equations can be readily combined to form a Poisson equation for \hat{w} :

$$\hat{w}'' - (k^2 + l^2)\hat{w} = i l \hat{\xi} - i k \hat{\eta}, \quad (8)$$

subject to homogeneous boundary conditions, $\hat{w} = 0$ at $z = z_{\min}$ and z_{\max} (here a prime denotes z differentiation). From \hat{w} and the vertical vorticity component $\hat{\zeta}$, the horizontal velocity components are recovered from

$$\hat{u} = \frac{ik\hat{w}' + il\hat{\zeta}}{k^2 + l^2}, \quad \hat{v} = \frac{il\hat{w}' - ik\hat{\zeta}}{k^2 + l^2}. \quad (9)$$

Notably, using the field decomposition introduced in [4], the vertical derivative of \hat{w} is obtained from a wavenumber multiplication of the sine series part followed by a cosine transform, and from the analytic differentiation of the harmonic part.

For $(k, l) = (0, 0)$, i.e. for the part of the flow independent of x and y , instead $\hat{w} = 0$ and the horizontal velocity components are found by integrating $\hat{u}' = \hat{\eta}$ and $\hat{v}' = -\hat{\xi}$. Again, this is done analytically. Full details and tests can be found in [4].

2.3. Elliptical parcels: basic representation and dynamics

As in the 2D model [1] we propagate the centre of an elliptical Lagrangian parcel i , located at $\mathbf{x}_i(t)$, and with velocity $\mathbf{u}_i(t) \equiv \mathbf{u}(\mathbf{x}_i, t)$ at time t , by the advection equation

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}_i(t), \quad (10)$$

and evolve its shape by (see [7])

$$\frac{d\mathbf{B}_i}{dt} = \mathbf{B}_i \mathbf{S}_i^\top + \mathbf{S}_i \mathbf{B}_i. \quad (11)$$

Here the 3×3 symmetric positive-definite matrix \mathbf{B} (upon dropping the subscript i) defines the shape of the ellipsoidal parcel through

$$\mathbf{x}^\top(t) \mathbf{B}^{-1}(t) \mathbf{x}(t) = 1, \quad (12)$$

with \top denoting a transpose, while \mathbf{S} is the local velocity gradient matrix,

$$\mathbf{S} = \nabla \mathbf{u} = \begin{pmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{pmatrix}. \quad (13)$$

(Note that \mathbf{x} in Eq. (12) refers to points on the ellipsoid and is not to be confused with the parcel centre \mathbf{x}_i .) Thanks to incompressibility and symmetry, we need to store only five elements of \mathbf{B} per parcel. This corresponds to three additional parcel attributes compared to the 2D model [1].

Notably, the evolution of materially-conserved quantities like b_i and q in Eqs. (2) and (3) is done *almost for free*: the corresponding parcel attributes are simply constant – they never need to be updated. Hence, as the parcels move and deform, they exactly preserve all conserved attributes. Only merger (see below) mixes parcel attributes. Critically, the EPIC method is automatically monotone [1]. This is in sharp distinction to Eulerian advection methods, where significant effort is often required to preserve monotonicity, besides numerical stability [8]. Moreover, the advantages of Lagrangian advection grow as more attributes are added, as would be required in more realistic atmospheric models.

Lagrangian advection is also advantageous for attributes which are not materially conserved yet are advection dominated (e.g. chemical and biological species, etc). In the present model, the vector vorticity is a prime example. As in [2], we evolve the vorticity $\boldsymbol{\omega}_i(t) = (\xi_i, \eta_i, \zeta_i)$ on each parcel i using the mathematically-equivalent flux form of Eq. (7) (see also [5,6]),

$$\frac{d\xi_i}{dt} = \nabla \cdot (\omega_a u + b \hat{\mathbf{e}}_y), \quad (14)$$

$$\frac{d\eta_i}{dt} = \nabla \cdot (\omega_a v - b \hat{\mathbf{e}}_x), \quad (15)$$

$$\frac{d\zeta_i}{dt} = \nabla \cdot (\omega_a w), \quad (16)$$

where $\omega_a \equiv \omega + 2\Omega$ is the absolute vorticity, and the right-hand sides are interpolated from the grid to the parcels (see Section 2.5). This appears to be the most stable approach based on extensive numerical tests. We have tried using the usual expression for the vortex stretching term in Eq. (7), but this is numerically unstable. The flux form above was previously advocated by [9] in Lagrangian PIC methods; other equivalent formulations discussed in [9] were found to be numerically unstable.

Additionally, to reduce aliasing errors when computing products of gridded quantities, we filter the vorticity ω (before inverting to find the velocity \mathbf{u}) using the filter recommended by [10]. This filter takes the form $\exp(-36(k/k_{\max})^{36})$ in each wavenumber direction (the total filter is a product of these one-dimensional filters). Here, it is applied to the decomposed vorticity field (see details in [4]), so is 2D for the surface-associated part and 3D for the remaining interior part. This filter is much less aggressive than the usual de-aliasing filter where the upper third of all modes are set to zero (the latter was used previously in MPIC [2]). Our results however are not sensitive to the choice of the filter. Both filters preserve numerical stability.

An important feature of the algorithm is the preservation of the domain mean vorticity,

$$\langle \omega \rangle = \frac{\sum_i \omega_i V_i}{\sum_i V_i}, \quad (17)$$

which is required for mathematical consistency [5,6]. We have found that numerical errors result in a drift of $\langle \omega \rangle$, which may strongly affect energy conservation – as found in the Beltrami test case in Section 3.3. In EPIC, we remove the domain-mean parcel vorticity at the beginning of each time step.

Finally, the gridded vorticity field obtained by interpolation from the parcels (see Section 2.5 below) is not exactly solenoidal, as required by the definition $\omega = \nabla \times \mathbf{u}$. Moreover, evolving all three components of vorticity is redundant since the constraint $\nabla \cdot \omega = 0$ is not guaranteed. A way around this is to evolve the horizontal curl of the horizontal vorticity, $\chi = \eta_x - \xi_y$, alongside the vertical component ζ , then recover ξ and η from the relations

$$\xi_x + \eta_y = -\zeta_z \quad \text{and} \quad \eta_x - \xi_y = \chi, \quad (18)$$

which become simple algebraic relations (at each z independently) in semi-spectral space, and are easily solved (details omitted). Note that because $\zeta = v_x - u_y$, both ζ and ζ_z have zero horizontal mean values, as does χ . Thus Eq. (18) can always be inverted for horizontal wavenumbers k and l not both zero (we do not use the above relations for the horizontal mean parts of ξ and η ; these are not modified). However, the dynamical equation satisfied by χ is much more complicated than those for ξ and η . Instead, we continue evolving all three components of vorticity, but correct the horizontal vorticity whenever needed by forming the horizontal curl $\chi = \eta_x - \xi_y$ then inverting the relations in Eq. (18) for ξ and η (excluding the horizontal mean part, $k = l = 0$). For the derivative ζ_z , we use centred differences with linear extrapolation at the boundaries as in [4].

2.4. Characteristics of the shape matrix

For various procedures, the EPIC model needs to obtain the elliptical semi-axis lengths, as well as their orientations, from the shape matrix \mathbf{B} . Conveniently, the eigenvalues of the shape matrix \mathbf{B} are the squared values of the semi-axes a^2 , b^2 and c^2 , where we assume $a \geq b \geq c$ without loss of generality [7]. The axis orientations are the corresponding *normalised* eigenvectors $\hat{\mathbf{a}}$, $\hat{\mathbf{b}}$ and $\hat{\mathbf{c}}$.

While in principle the eigenvalues a^2 , b^2 and c^2 of the 3×3 symmetric matrix \mathbf{B} may be found using the exact solution to the cubic first recorded by Cardano [11] (see also [12] for some of the fascinating history), it turns out not to be stable enough numerically due to occasionally severe round-off errors. Typically, we require a large number of eigenvalue/eigenvector solves each time step, and our tests show that some of the matrices having two nearby eigenvalues result in unacceptable errors, far exceeding numerical precision. This problem is well known, and has led to alternative iterative approaches e.g. using the Jacobi algorithm [13,14]. Here we use the approach of Scherzinger and Dohrmann [15] that combines the efficiency of analytical methods with the stability and accuracy of numerical methods.

2.5. Interpolation

A core feature of PIC methods is the transfer of parcel properties to the underlying grid, and vice-versa. Gridded fields are obtained by summing weighted parcel properties at each grid point, while the velocity of a parcel for example is interpolated from nearby grid points. In EPIC, parcels are ellipsoids of finite volume, and thus the contribution of a given

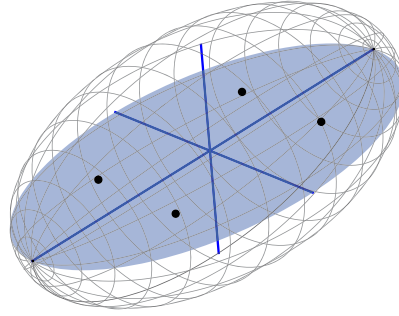


Fig. 1. Location of the four support points (black dots) in the major-middle plane (blue shaded ellipse) of a general ellipsoid.

parcel depends on its specific shape. In principle, one must integrate over each parcel, since different parts of it contribute differently to nearby grid points (moreover a parcel may straddle grid cell boundaries). In 2D, this integration can be done efficiently using 2-point Gaussian quadrature, where the points lie along the major axis of the ellipse at half the focal distance from the ellipse centre [1]. In 3D, the analogous quadrature uses 4 points positioned within the plane containing the major and middle axes of the ellipsoid [7]. This is in fact the simplest quadrature (apart from using a single point); more accurate quadrature formulae use 7, 13 or more points. However, as a compromise between accuracy and efficiency, only 4 points are used here.

The 4 points inside the ellipsoid are called “support points”. For parcel i centred at \mathbf{x}_i , they are located in space at the positions

$$\mathbf{x}_i^m = \mathbf{x}_i + \Xi_i \cos \vartheta_m \hat{\mathbf{a}}_i + \Upsilon_i \sin \vartheta_m \hat{\mathbf{b}}_i \tag{19}$$

where $\Xi_i \equiv \sqrt{2(a_i^2 - c_i^2)}/5$, $\Upsilon_i \equiv \sqrt{2(b_i^2 - c_i^2)}/5$, and $\vartheta_m \equiv m\pi/2 - \pi/4$ for $m = 1, 2, 3$ and 4 . The support points are shown in the plane spanned by the eigenvectors $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{b}}_i$ in the right panel of figure 2 in [7] (note the reference uses the reverse axis order $c_i \geq b_i \geq a_i$). An alternative 3D view is provided in Fig. 1. The support points always lie inside the ellipsoid.

2.5.1. Interpolation from parcel values to the grid

As in the 2D model, we use linear interpolation in each coordinate direction (here tri-linear interpolation) to obtain a gridded field quantity, say \bar{q} , from the parcel attributes q_i and volumes V_i . Here, there are 4 support points per parcel (rather than 2 in the 2D model) and each such point gets an equal weight of 1/4. First, we initialise the gridded volume $\bar{V} = 0$ and an auxiliary field $\bar{m} = 0$; after interpolation the ratio of these defines \bar{q} . Second, we sum over all parcels and support points, interpolate, and accumulate \bar{V} and \bar{m} . For each support point \mathbf{x}_i^m , we locate the grid box containing it – this is easily done on a regular grid. Let $\bar{\mathbf{x}}$ denote the corner of this grid box having the smallest \bar{x} , \bar{y} and \bar{z} . The interpolation weight between \mathbf{x}_i^m and $\bar{\mathbf{x}}$ is

$$\phi_i^m = \frac{1}{4} \left(1 - \frac{|\bar{x}_i^m - \bar{x}|}{\Delta x} \right) \left(1 - \frac{|\bar{y}_i^m - \bar{y}|}{\Delta y} \right) \left(1 - \frac{|\bar{z}_i^m - \bar{z}|}{\Delta z} \right). \tag{20}$$

The analogous weights at the other 7 corners of the grid cell are found by replacing \bar{x} with $\bar{x} + \Delta x$, or \bar{y} with $\bar{y} + \Delta y$, or \bar{z} with $\bar{z} + \Delta z$ above. These weights are multiplied by the parcel volume V_i and added to \bar{V} at each corner of the grid cell. Likewise the weights are multiplied by $q_i V_i$ and added to \bar{m} . Once we have summed over all parcels and support points, the gridded quantity \bar{q} is found simply by the ratio \bar{m}/\bar{V} at all grid points $\bar{\mathbf{x}}$. Note that the same weights ϕ_i^m can be used to generate all fields at once for efficiency.

As in 2D, we must double the gridded values at the boundaries $i_z = 0$ and n_z to ensure the total gridded volume is conserved and equal to the domain volume. For other gridded quantities this factor of two cancels out, as required, when taking the ratio \bar{m}/\bar{V} . A downside to this interpolation is that fields having a linear variation in z near the boundaries are in general poorly represented, unless the parcel attributes are chosen carefully. But we can only choose the parcel attributes at the initial time to match input gridded values (see Section 2.10 below), and have no control at later times. All of our attempts to use higher-order extrapolation led to numerical instability (divergence of kinetic energy) in the Beltrami test case discussed in Section 3.3.

This boundary interpolation is over-ridden for the vertical vorticity component ζ when it is initially zero and when it cannot become non-zero theoretically (i.e. when the vertical component of the background rotation vector $\boldsymbol{\Omega}$ is zero). In this case, we set the gridded values of ζ as well as its gridded tendency $D\zeta/Dt$ at $i_z = 0$ and n_z to zero.

2.5.2. Interpolation from the grid to parcel values

The reverse operation, interpolating grid values \bar{q} to each parcel support point, is done using the same weights in Eq. (20), now with contributions from all 8 corners of the grid box containing the support point. Then the values accumulated at the 4 support points per parcel are summed together to define the parcel attribute q_i , usually a parcel tendency or velocity.

2.6. Parcel splitting

A parcel is split if either the parcel aspect ratio λ (the ratio of the largest to smallest semi-axis length) exceeds a user-specified parameter λ_{\max} , or the maximum parcel semi-axis length a (suppressing the index i) exceeds $a_{\max} = \sqrt[3]{3/4\pi} \min\{\Delta x, \Delta y, \Delta z\}$. In 3D, the aspect ratio $\lambda = a/c$, where the semi-axis lengths satisfy $a \geq b \geq c$. The value of a_{\max} is chosen to limit the maximum parcel volume V to the grid cell volume $V_{\text{cell}} = \Delta x \Delta y \Delta z$. This is a change from the 2D model (see [1]) which simply limited the parcel volume directly.

After splitting, the shape matrices of the two new parcels are given by

$$\mathbf{B}_1 = \mathbf{B}_2 = \mathbf{B} - \frac{3}{4}a^2\hat{\mathbf{a}}\hat{\mathbf{a}}^T \quad (21)$$

just as in the 2D model. Only the semi-major axis length a of the original parcel is divided in two; the other axis lengths remain the same. Each new parcel has half the volume of the original parcel. The only difference with the 2D model is that the new parcel centres are separated by a distance $2h = \sqrt{3/5}a$ rather than $\sqrt{3/4}a$ in 2D. The new parcel centres lie at $\mathbf{x} \pm h\hat{\mathbf{a}}$ (details may be found in Appendix A). Note, parcel splitting preserves the zeroth, first and second-order spatial moments.

2.7. Parcel merging

The criterion used for parcel merging is identical to that used in the 2D model. A parcel m is marked available for merging if its volume $V_m < V_{\min} = \tilde{V}_{\min}V_{\text{cell}}$, where \tilde{V}_{\min} is another user-controlled parameter (all numerical parameters may be found in Table 1 below or in [1]). Moreover, the procedure to find and merge groups of merging parcels remains the same as that described in appendix D of [1].

Merger preserves the total volume (zeroth moment) $V = 4\pi abc/3 = \sum_i V_i = 4\pi/3 \sum_i a_i b_i c_i$, the centroid (first moment) $\mathbf{x} = V^{-1} \sum_i V_i \mathbf{x}_i$ and, approximately, the second-order spatial moments, where i ranges over all parcels in the group of merging parcels. The only difference in the entire merging procedure stems from the fact that the parcels are ellipsoids rather than ellipses. This means that the (tentative) shape matrix of a merged parcel must now be computed from

$$\begin{aligned} B_{11}^* &= \frac{1}{V} \sum_i V_i (5\check{x}_i^2 + B_{11,i}) \\ B_{12}^* &= \frac{1}{V} \sum_i V_i (5\check{x}_i \check{y}_i + B_{12,i}) \\ B_{13}^* &= \frac{1}{V} \sum_i V_i (5\check{x}_i \check{z}_i + B_{13,i}) \\ B_{22}^* &= \frac{1}{V} \sum_i V_i (5\check{y}_i^2 + B_{22,i}) \\ B_{23}^* &= \frac{1}{V} \sum_i V_i (5\check{y}_i \check{z}_i + B_{23,i}) \\ B_{33}^* &= \frac{1}{V} \sum_i V_i (5\check{z}_i^2 + B_{33,i}), \end{aligned} \quad (22)$$

where $\check{\mathbf{x}}_i \equiv \mathbf{x}_i - \mathbf{x}$ (these are the only unique components since \mathbf{B}^* is symmetric). The final shape matrix is found from

$$\mathbf{B} = \mathbf{B}^* \left(\frac{a^2 b^2 c^2}{\det(\mathbf{B}^*)} \right)^{1/3}, \quad (23)$$

which ensures that the volume of the merged parcel equals the sum of the volumes of the merging parcels ($\det(\mathbf{B}) = (abc)^2$ is proportional to the squared parcel volume).

2.8. Enforcing boundary conditions on parcel operations

In the 3D model, the domain is periodic in x and in y , and periodicity is handled in precisely the same manner as in the 2D model [1] (where only x is periodic). At both free-slip z boundaries, the grid extends beyond the domain by one halo grid layer ($i_z = -1$ and $i_z = n_z + 1$) to accommodate parcel centres and support points that may appear outside during an advection step. These boundaries are also handled the same way apart from one difference: the halo grid points are filled by extrapolation. In particular, all velocity and strain components are extrapolated, rather than mirrored assuming symmetry or anti-symmetry as in the 2D model. Tests using extrapolation in the 2D model (see [1]) in fact show generally negligible differences. This is because the halo grid points are both rarely needed and receive a small weight when they are. Nonetheless, in the 3D model, we have opted for extrapolation based on the exact test flow discussed in Section 3.3, for which neither the horizontal velocity components nor their z derivatives vanish at the boundaries. As the vertical velocity component $w = 0$ on the boundaries, extrapolation is equivalent to assuming anti-symmetry across the boundaries.

2.9. Incompressibility

As in the 2D model, at each time step we correct for errors in the gridded volume \bar{V} arising from the parcel motion. Any finite number of parcels cannot in general ensure $\bar{V} = V_{\text{cell}}$ at all times, and a novel feature of the EPIC method is that these errors are corrected every time step by a minor adjustment of the parcel centres (see further discussion in [1]). This adjustment is done in two complementary ways to accelerate the convergence of \bar{V} to V_{cell} . First, a displacement field $\nabla\phi$ is found by solving Poisson's equation $\nabla^2\phi = \delta \equiv (\bar{V} - V_{\text{cell}})/V_{\text{cell}}$, subject to the boundary condition $\phi_z = 0$ at $z = z_{\text{min}}$ and z_{max} to prevent parcels from leaving the domain. In semi-spectral space, after a horizontal Fourier transform, each Fourier amplitude $\hat{\phi}(k, l, z)$ solves

$$\hat{\phi}'' - (k^2 + l^2)\hat{\phi} = \hat{\delta} \quad (24)$$

subject to $\phi'(k, l, z_{\text{min}}) = \phi'(k, l, z_{\text{max}}) = 0$. This is solved numerically using centred differences resulting in a tridiagonal problem analogous to that described in the appendix of [1], only not using compact differencing.

Once the parcel centres are adjusted by $\nabla\phi$, the gridded volume \bar{V} is recomputed for the next stage of the parcel correction algorithm. In this stage, parcels are displaced in a way which depends on their location within the grid cell and the discrepancies between \bar{V} and V_{cell} at the eight corners of the grid cell. Each parcel coordinate is displaced independently; so for x_i , the displacement is $C_x \tilde{x}_i (1 - \tilde{x}_i) \Delta x$ where $\tilde{x}_i = (x_i - \bar{x})/\Delta x$ varies from 0 to 1, and C_x is a dimensionless pre-factor dependent on $\bar{V} - V_{\text{cell}}$ at the corners of the grid cell. For a cell edge which is oriented in the x -direction, between grid points \bar{x}_n and \bar{x}_{n+1} , C_x is given by

$$C_x = -\beta \frac{\bar{V}_{n+1} - \bar{V}_n}{V_{\text{cell}}}. \quad (25)$$

The default value of β is set to 1.8, as in [1]. For a point in the cell interior, linear interpolation is used to determine C_x , again following [1]. The magnitudes of the pre-factors C_x , C_y and C_z are limited to C_{max} , a user-specified numerical parameter with default value 0.5.

These two stages are repeated once to keep volume errors small, typically below $10^{-3} V_{\text{cell}}$. This is demonstrated below (see Fig. 8 in Section 3.1).

2.10. Initialisation

A simulation normally begins by reading in field data from a grid for each prognostic quantity in the model, here b_l , q and ω . These fields are then used to define the corresponding parcel attributes by interpolation from the grid as explained in Section 2.5.2. By default, we place a regular $2 \times 2 \times 2$ array of parcels in each grid cell, each having a volume of $V_{\text{cell}}/8$. As the parcels are typically spherical, we simply interpolate the gridded values to the parcel centres to define the parcel attributes. Note that the initialisation does not guarantee that the reverse interpolation, from parcel attributes to gridded values in Section 2.5.1, matches the given field values. An algorithm designed to match field values was proposed in section 3.1 of [1], but in applications here we have found that it can lead to overshoots and undershoots of parcel attributes. Such overshoots and undershoots have been found to degrade energy conservation in the test flows examined below. Exceptionally, we may bypass this algorithm and directly assign parcel attributes from analytical functions as in [2,3].

2.11. Time stepping

Time stepping is carried out by a low-storage 4th-order Runge-Kutta method, exactly as described in [1]. The time step Δt is adapted at the beginning of each time step to ensure that the maximum (generalised) buoyancy frequency N_{max} and maximum rate of deformation γ_{max} are both resolved in time. Here $N \equiv \sqrt{\|\nabla b\|}$ while γ is the maximum eigenvalue of the symmetrised velocity gradient matrix $(\mathbf{S} + \mathbf{S}^T)/2$, as used in the 2D method [1]. Note that N and γ are evaluated at each grid point, and the maximum is taken over all grid points.

Table 1

Numerical parameters and their default values. The default value of the initial number of parcels per cell and the minimum parcel volume fraction are different compared to the 2D method [1]. Also, the 3D model has a slightly different parcel splitting criterion which does not require the maximum parcel volume fraction parameter anymore, see Section 2.6 for further details.

Numerical parameter	Value
Initial number of parcels per cell	8
Maximum aspect ratio, λ_{\max}	4
Minimum parcel volume fraction, \tilde{V}_{\min}	1/20
Number of iterations to correct for incompressibility	2
Gradient correction pre-factor, β	1.8
Gradient correction max. compression, C_{\max}	0.5
Time step pre-factor, α	0.2

The time step is chosen as

$$\Delta t = \min\left(\frac{\alpha}{N_{\max}}, \frac{\alpha}{\gamma_{\max}}\right) \quad (26)$$

where α is a prescribed dimensionless numerical constant. The recommended default, $\alpha = 0.2$, is the result of extensive tests carried out in [1]. Tests with the 3D method continue to show that $\alpha = 0.2$ is adequate – a compromise between accuracy and efficiency.

3. Results

We next illustrate the performance of the EPIC method in four distinct test cases. We assess accuracy by the rate of convergence of early-time solutions, gridded-volume errors in time, probability density functions of vorticity, etc. We also compare with the Moist Parcel-In-Cell (MPIC) method previously developed to study moist convection [2,3], as well as with both a standard large eddy simulation (LES) method and a pseudo-spectral method (PS3D) [4,16] to exhibit some of the distinct advantages of EPIC. For reference, we list all of the numerical parameters used in Table 1. In Appendix E, the cost and accuracy of the model is examined with respect to the minimum parcel volume fraction and the maximum parcel aspect ratio using the rotating Rayleigh-Taylor test case. For PS3D, we use $\alpha = 0.1$ for time stepping unless otherwise indicated.

In several of the test cases, we assess convergence through the evolution of the kinetic and potential energy components, \mathcal{K} and \mathcal{P} , as well as the enstrophy Υ . These are defined per unit volume by

$$\mathcal{K} = \frac{1}{2V_{\text{dom}}} \int \int \int |\mathbf{u}|^2 dV \quad (27)$$

$$\mathcal{P} = -\frac{1}{V_{\text{dom}}} \int \int \int bz dV \quad (28)$$

$$\Upsilon = \frac{1}{2V_{\text{dom}}} \int \int \int |\boldsymbol{\omega}|^2 dV, \quad (29)$$

where $V_{\text{dom}} = L_x L_y L_z$ is the volume of the domain. Since the potential energy \mathcal{P} may be dominated by the background hydrostatic equilibrium state (see in particular the internal wave test case in Section 3.2), it is more useful here to consider the *available* potential energy $\mathcal{A} = \mathcal{P} - \mathcal{P}_{\text{ref}}$, where \mathcal{P}_{ref} is the constant potential energy of the buoyancy (i.e. density) field re-arranged into hydrostatic equilibrium (see [17–19] and §2.9 in [1]). For small departures from hydrostatic equilibrium, numerical inaccuracies in the discrete calculation of the reference state can swamp the actual available potential energy, a known issue [18,19].

Where possible, we make use of the known hydrostatic reference state, $b_{\text{ref}}(z)$, to accurately compute \mathcal{A} from the form proposed in [20]. There, the available potential energy *density*, $a(b, z)$, is defined by

$$a(b, z) = \int_{b_{\text{ref}}(z)}^b (\mathcal{Z}(\tilde{b}) - \mathcal{Z}(b_{\text{ref}}(z))) d\tilde{b} \quad (30)$$

where $\mathcal{Z}(b)$ is the inverse function corresponding to $b_{\text{ref}}(z)$, i.e. $\mathcal{Z}(b_{\text{ref}}(z)) = z$. From $a(b, z)$, the available potential energy per unit volume is calculated from

$$\mathcal{A} = \frac{1}{V_{\text{dom}}} \int \int \int a(b, z) dV. \quad (31)$$

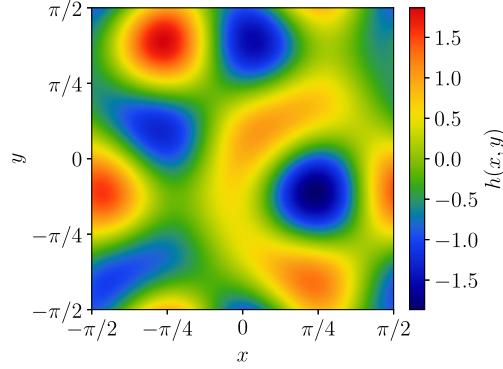


Fig. 2. Horizontal perturbation of the rotating Rayleigh-Taylor flow test case.

3.1. Rayleigh-Taylor flow test case

We first consider a buoyancy-driven flow in $[-\pi/2, \pi/2]^3$ with a perturbed unstable buoyancy profile

$$b(x, y, z) = -\sin z + \epsilon h(x, y) \cos^2 z \quad (32)$$

where $\epsilon = 0.1$ and

$$h(x, y) = \cos 4x \cos(2y + \pi/6) + \sin(2x + \pi/6) \sin 4y \quad (33)$$

(cf. Fig. 2). This smooth function is chosen to be void of symmetry. We also consider a vertical background rotation $\mathbf{\Omega} = (0, 0, 1/2)$, corresponding to a Coriolis frequency of unity, which is also equal to the maximum of $\sqrt{|b_z|}$ when $\epsilon = 0$. The instability manifested by this flow is known as the Rayleigh-Taylor instability, first identified by Rayleigh [21] in 1882 and Taylor [22] in 1950 (see also [23] for the rotating case).

In this test case, the analytical form of the available potential energy density $a(b, z)$ can be found from Eq. (30), noting that the hydrostatic reference state $b_{\text{ref}}(z) = \sin z$. One may show that

$$a = b \sin^{-1} b + \sqrt{1 - b^2} - zb - \cos z, \quad (34)$$

and that this function is positive definite over $|z| \leq \pi/2$ and $|b| \leq 1$ except along $b = \sin z = b_{\text{ref}}(z)$ where $a = 0$. Moreover, for $b' = b - b_{\text{ref}}$ sufficiently small, $a \approx (b')^2 / (2N^2)$ where $N^2 = db_{\text{ref}}/dz$, the correct asymptotic form for small perturbations.

Since the initial vertical buoyancy profile has density *increasing* with height z , the flow overturns in an attempt to restore hydrostatic equilibrium. During this process the fluid undergoes strong mixing which decreases the buoyancy magnitude and leads to a sharp decay in kinetic energy (see below). Fig. 3 illustrates $y = 0$ cross sections of the gridded buoyancy field b computed by EPIC at several representative times. For comparison, Fig. 4 shows results at approximately the same times, but now from a 128^3 simulation using the Eulerian pseudo-spectral method 'PS3D' described in [4]. Notably, the detailed flow structure obtained with EPIC is closely similar to that obtained with PS3D at early times. During the turbulent phase when strong mixing occurs, the simulation results diverge as expected in part due to loss of predictability, but nonetheless remain qualitatively similar. As hydrostatic equilibrium is approached at later times, the results more closely resemble each other again.

We next examine the energy and enstrophy evolution, comparing EPIC and PS3D both for a grid resolution of 128^3 . Enstrophy Υ peaks at a substantially larger value in EPIC than in PS3D – see the lower right panel of Fig. 5. The enstrophy growth corresponds to the destabilisation of the unstable base flow. Otherwise, the kinetic energy \mathcal{K} (shown in the upper left panel of the same figure) exhibits closely similar behaviour, with the steep decay occurring at nearly the same time in both methods (slightly later in EPIC). The initial decay in available potential energy \mathcal{A} closely compensates the growth in \mathcal{K} , as required for total energy conservation. In Table 2 we list the energy loss at time $t = 4$ relative to the initial total energy at $t = 0$ as a function of the grid resolution. For EPIC, the energy loss reduces by at least $\sqrt{8}$ when doubling the grid resolution in all dimensions.

Notably, the PS3D simulation exhibits buoyancy overshoots and undershoots as found previously in 2D simulations [1], while EPIC by construction exhibits monotonically increasing and decreasing buoyancy extreme values $b_{\text{min}}(t)$ and $b_{\text{max}}(t)$, respectively – see Fig. 6. The non-physical undershoots and overshoots in PS3D are caused by hyperdiffusion, and are most prominent near the boundaries, clearly visible in Fig. 4. This feature is especially undesirable in situations where it is important to preserve monotonicity, e.g. when a field like specific humidity q must remain positive definite (see Section 3.4 below). While the pseudo-spectral method normally has excellent energy conservation properties, these may be outweighed by the method's inability to preserve monotonicity. Alternatively, one can use an explicitly monotonic advection scheme as in the large eddy simulation (LES) model featured in Section 3.4, but this comes with a substantial increase in numerical

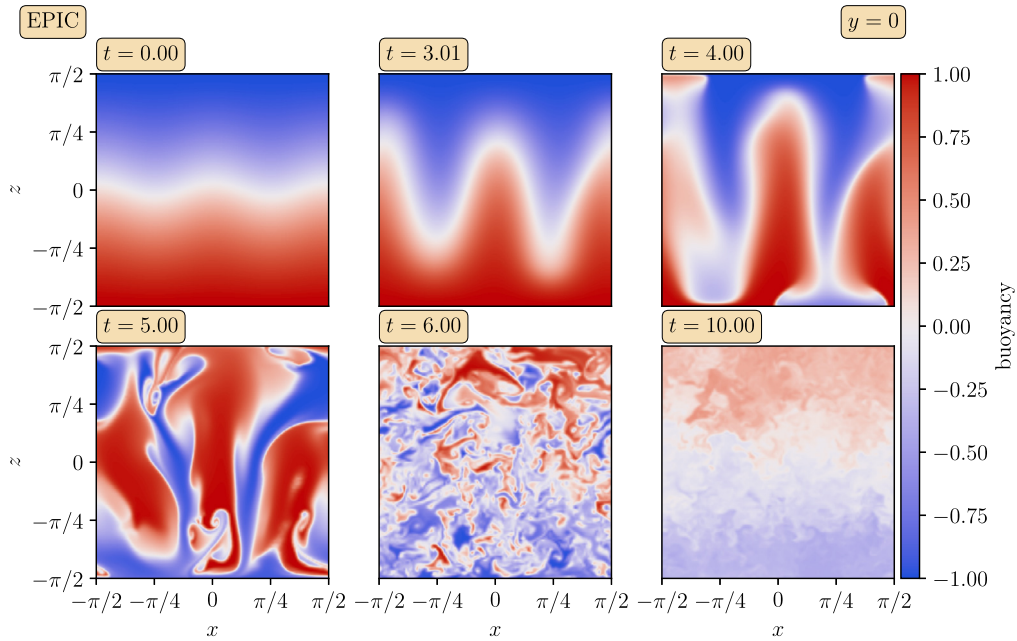


Fig. 3. Buoyancy cross sections in the xz -plane at $y = 0$ for various time steps of the Rayleigh-Taylor instability using EPIC at 128^3 grid resolution.

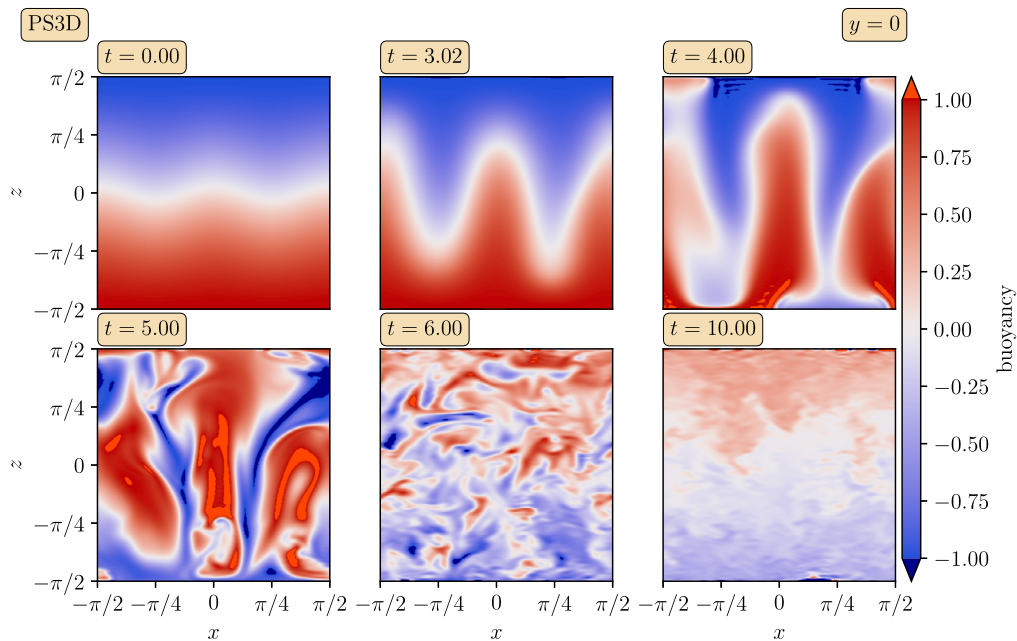


Fig. 4. As in Fig. 3 but using the PS3D method. Here, we have used the same colourmap and range, except that values outside the initial range of buoyancy $[-1, 1]$ are plotted with a uniform, different colour. These unphysical values can be far outside the initial range of buoyancy (see text for details).

cost. The vertical profile of mean buoyancy (mean with respect to the horizontal directions x and y) in Fig. 7 shows good agreement between EPIC and PS3D at early times, but at later times ($t > 5$) PS3D exhibits larger – and unphysical – mean buoyancy values at the bottom and top boundaries compared to EPIC, see in particular times $t = 6$ and 10 .

Finally, we verify that EPIC closely maintains incompressibility (local volume conservation) in this demanding test case. As explained in Section 2.9, we apply two parcel correction methods after every time step to maintain a homogeneous and space-filling parcel distribution inside the domain. To demonstrate the effectiveness of these parcel correction methods, we evaluate the (relative) r.m.s. volume error $\sqrt{\langle (\bar{V} - V_{\text{cell}})^2 \rangle} / V_{\text{cell}}$ as a function of time. As shown in Fig. 8, EPIC keeps the

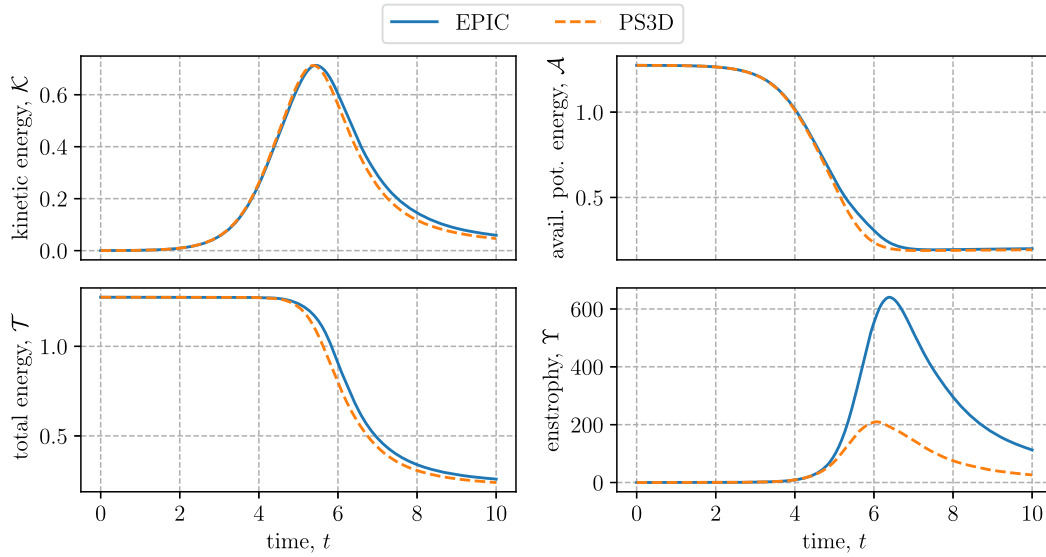


Fig. 5. Evolution of kinetic energy \mathcal{K} , available potential energy \mathcal{A} , total energy $\mathcal{T} = \mathcal{K} + \mathcal{A}$ and enstrophy Υ for the rotating Rayleigh-Taylor test case. Note that we used the parcels to evaluate these quantities in EPIC.

Table 2

Relative total energy loss for the Rayleigh-Taylor test case between the initial time $t = 0$ and time $t = 4$ as a function of the grid resolution. At this early time, energy is still well conserved.

Number of grid cells	Energy loss (%)	
	EPIC	PS3D
32^3	1.45	1.55
48^3	0.81	0.69
64^3	0.51	0.43
96^3	0.25	0.24
128^3	0.15	0.18
384^3	0.02	-

relative error below $1.5 \cdot 10^{-3}$ throughout the simulation, despite the strong mixing that takes place. This error is similar to what was found previously in [1] for the 2D method.

Note that mixing in EPIC is fully represented by explicit splitting and merging of parcels, and no additional parameters (such as a Smagorinsky constant or turbulent Prandtl number) are needed. With the exception of the initial parcel number per cell, these parameters are the same as used in the 2D method [1], where a thorough parameter-sensitivity analysis was carried out. In Appendix C, a stochastic model is used to argue that the choice of a_{\max} is not important for a well-developed flow.

3.2. Internal wave test case

In this test case we consider a resting basic state with linear stratification $\bar{b} = N^2 z$ in a uniformly rotating flow with (vertical) Coriolis frequency $f = 2\Omega$ and buoyancy frequency N . The perturbed flow has velocity (u, v, w) , vorticity (ξ, η, ζ) and buoyancy $\bar{b} + b'$. All quantities are considered infinitesimal except for \bar{b} .

The linearised equations for vorticity and buoyancy are

$$\xi_t = b'_y + f u_z \quad (35)$$

$$\eta_t = -b'_x + f v_z \quad (36)$$

$$\zeta_t = f w_z \quad (37)$$

$$b'_t = -N^2 w. \quad (38)$$

We now seek eigensolutions of the form

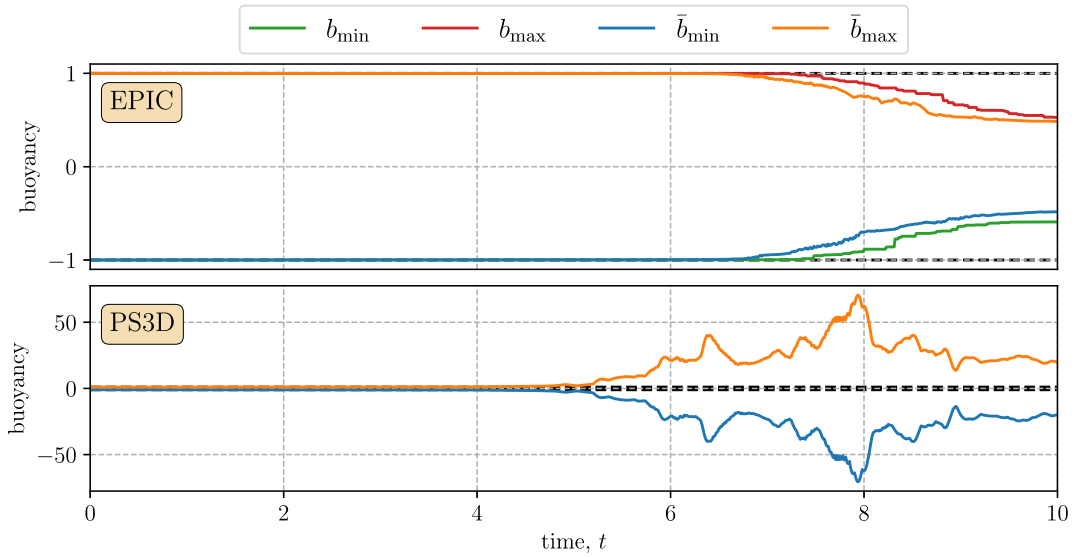


Fig. 6. Evolution of the gridded buoyancy and parcel buoyancy extrema (\bar{b}_{\min} , \bar{b}_{\max}) and (b_{\min} , b_{\max}), respectively. The theoretical bounds $[-1, 1]$ are indicated by the dashed lines. PS3D exhibits large undershoots and overshoots, greatly exceeding these bounds.

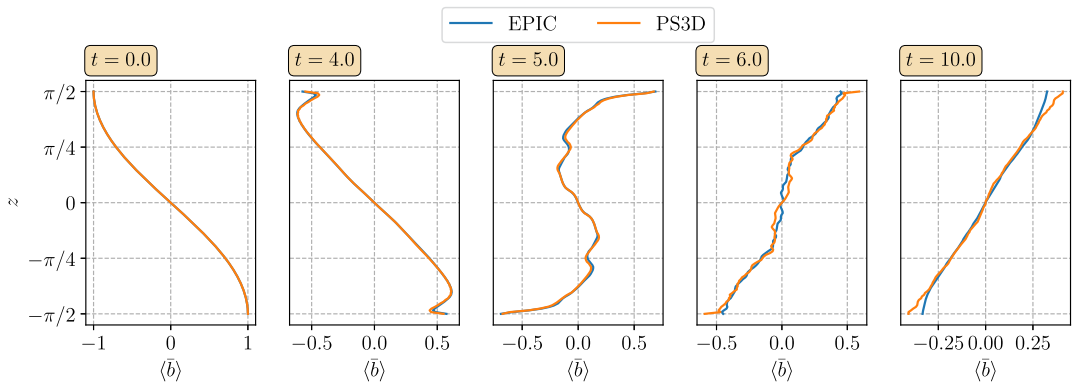


Fig. 7. Gridded mean buoyancy vertical profile at various times for the rotating Rayleigh-Taylor test case.

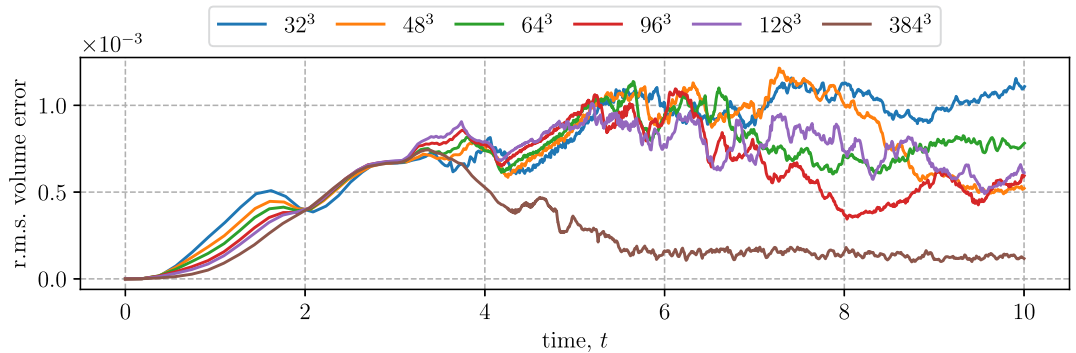


Fig. 8. Gridded relative r.m.s. volume error for the rotating Rayleigh-Taylor test case with different grid resolutions. EPIC manages to preserve incompressibility very well – the error remains smaller than $1.5 \cdot 10^{-3}$ – despite the strongly turbulent overturning flow. Note that the simulation having 384^3 grid cells does not use the default value of the minimum parcel volume fraction (cf. Table 1), but $\tilde{V}_{\min} = 1/40$.

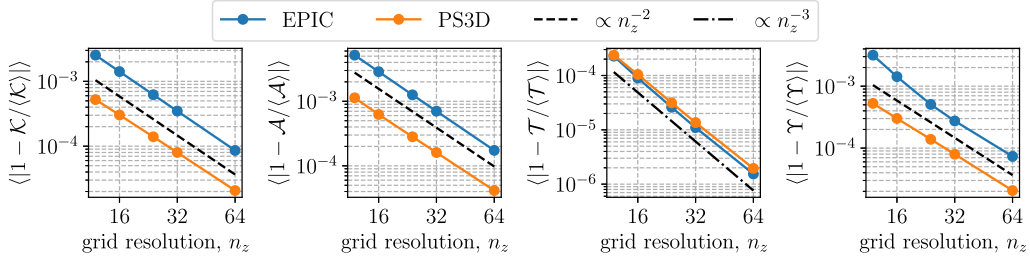


Fig. 9. Convergence analysis of the time-averaged relative error to the mean values of kinetic energy \mathcal{K} , available potential energy \mathcal{A} , total energy $\mathcal{T} = \mathcal{K} + \mathcal{A}$ and enstrophy Υ .

$$\mathbf{u} = \hat{u} \sin mz e^{i\varphi} \quad (39)$$

$$\mathbf{v} = \hat{v} \sin mz e^{i\varphi} \quad (40)$$

$$w = \hat{w} \cos mz e^{i\varphi} \quad (41)$$

$$b' = \hat{b}' \cos mz e^{i\varphi}, \quad (42)$$

where $\varphi = kx + ly - \sigma t$. These forms are consistent with $w = 0$ at $z = \pm L_z/2$, as well as the definition of vorticity and the incompressibility relations for velocity and vorticity, so long as $m = (2j - 1)\pi/L_z$ with j a positive integer.

Inserting these forms into Eqs. (35) to (38) and performing much algebra leads to the dispersion relation

$$\sigma^2 = \frac{N^2(k^2 + l^2) + f^2 m^2}{k^2 + l^2 + m^2}. \quad (43)$$

Moreover, if we start with $w = \hat{w} \cos mz \cos(kx + ly)$ and take \hat{w} to be real, then the other fields for any $t \geq 0$ are

$$u = (k^2 + l^2)^{-1} \hat{w} m \sin mz [k \sin \varphi + \sigma^{-1} f l \cos \varphi] \quad (44)$$

$$v = (k^2 + l^2)^{-1} \hat{w} m \sin mz [l \sin \varphi - \sigma^{-1} f k \cos \varphi] \quad (45)$$

$$\xi = (\sigma^2 - f^2)^{-1} \hat{w} \cos mz [\sigma^{-1} f k (N^2 - \sigma^2) \cos \varphi - l (N^2 - f^2) \sin \varphi] \quad (46)$$

$$\eta = (\sigma^2 - f^2)^{-1} \hat{w} \cos mz [\sigma^{-1} f l (N^2 - \sigma^2) \cos \varphi + k (N^2 - f^2) \sin \varphi] \quad (47)$$

$$\zeta = \sigma^{-1} f m \hat{w} \sin mz \sin \varphi \quad (48)$$

$$b = \bar{b} + \sigma^{-1} N^2 \hat{w} \cos mz \sin \varphi. \quad (49)$$

We next use these solutions to test the numerical method. We initialise the flow with the above vorticity and buoyancy fields at $t = 0$, taking a small value of \hat{w} , here 10^{-3} , so that the linearised approximation is valid (nonlinear effects are expected to arise at $\mathcal{O}(\hat{w}^2)$). These fields are then used to obtain the initial parcel attributes as described in Section 2.10. In the test case illustrated, we have taken $k = l = 1/2$ and $m = 1$ in a domain of extent $4\pi \times 4\pi \times \pi$ centred on the origin, and have used an isotropic grid resolution of $48 \times 48 \times 12$, $64 \times 64 \times 16$, $96 \times 96 \times 24$, $128 \times 128 \times 32$ or $256 \times 256 \times 64$. We have also taken $N = 2$ and $f = 1$, giving an eigenfrequency of $\sigma = \sqrt{2}$. The wave period is $2\pi/\sqrt{2} \approx 4.44$.

In linear theory, the flow evolves by uniformly translating in x and y at constant speeds σ/k and σ/l , both equal to $2\sqrt{2}$ in the example illustrated. Simulations using both EPIC and a pseudo-spectral method, PS3D, originally developed in [4] and extended to include buoyancy effects (version 0.0.13 [16]), closely approximate the exact linear solution despite the low resolution employed. Note that for PS3D we evolve the buoyancy perturbation b' instead of the total buoyancy $b = \bar{b} + b'$, which here is dominated by $\bar{b} = N^2 z$. The perturbation buoyancy evolves according to $b'_t + \mathbf{u} \cdot \nabla b' = -N^2 w$. Since only the horizontal gradient of b' is needed for the vorticity tendency, in EPIC we recover b' on the grid by interpolating the parcel values of $b'_i - N^2 z_i$. When the total buoyancy is required for output purposes, we simply add $N^2 z$ to b' . This is done to avoid boundary interpolation errors associated with the dominant mean part of b .

A convergence study of the kinetic energy \mathcal{K} , available potential energy \mathcal{A} , total energy \mathcal{T} and enstrophy Υ evolution is shown in Fig. 9 for both PS3D and EPIC. We make use of the known reference state, $b_{\text{ref}}(z) = N^2 z$, to accurately compute \mathcal{A} from the form proposed in [20], i.e. as an integral over the available potential energy density α , here given by

$$\alpha = \frac{(b - N^2 z)^2}{2N^2} = \frac{(b')^2}{2N^2}. \quad (50)$$

For the linearised solution for b' above, one can show that $\mathcal{A} = N^2 \hat{w}^2 / (8\sigma^2)$, or 2.5×10^{-7} for the parameters chosen. Moreover, the ratio of kinetic to available potential energy is $\mathcal{K}/\mathcal{A} = 1 + (f^2/N^2)(2m^2/(k^2 + l^2))$, or 2 for the parameters chosen.

Table 3
Relative total energy loss (in per mille) for the internal wave test case between the initial time $t = 0.0$ and the final time $t = 4\pi/\sqrt{2}$ versus the grid resolution. Note that the simulation values are interpolated to match the final time.

Number of grid cells	Energy loss (%)	
	EPIC	PS3D
$48^2 \times 12$	0.310	0.514
$64^2 \times 16$	0.124	0.217
$96^2 \times 24$	0.040	0.065
$128^2 \times 32$	0.017	0.029
$256^2 \times 64$	0.002	0.005

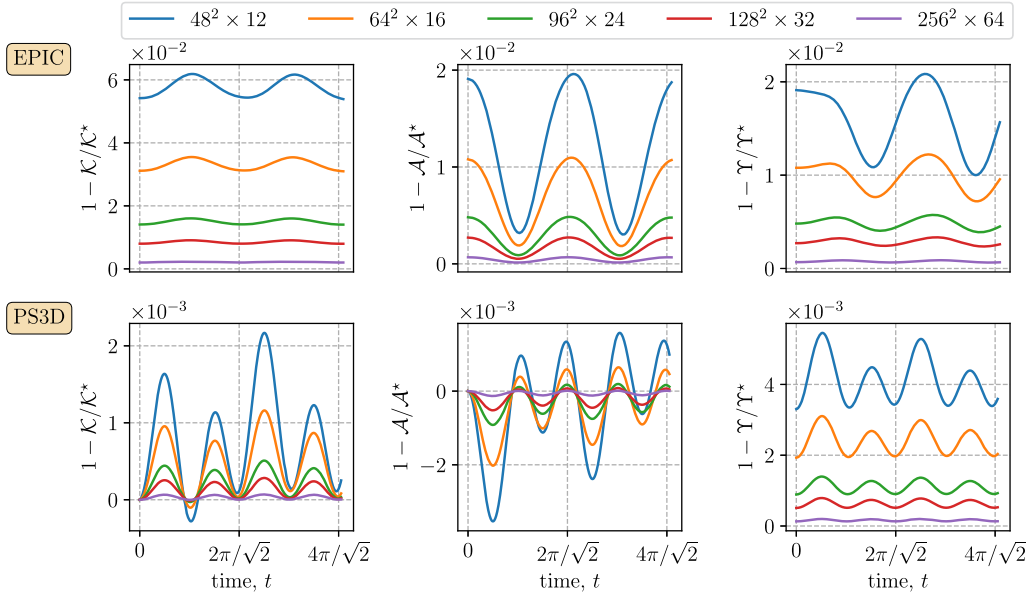


Fig. 10. Convergence analysis of the relative error between the numerical and the theoretical values of kinetic energy \mathcal{K} , available potential energy \mathcal{A} and entropy Υ for the internal wave test case. The analytical values are $\mathcal{K}^* = 5 \cdot 10^{-7}$, $\mathcal{A}^* = 2.5 \cdot 10^{-7}$ and $\Upsilon^* = 7.5 \cdot 10^{-7}$.

In EPIC, \mathcal{K} , \mathcal{A} and Υ are calculated from sums over parcels. In PS3D, \mathcal{A} uses a limited form of b , namely $\min(b_{\max}, \max(b_{\min}, b))$, restricted to lie between the theoretical minimum and maximum buoyancy values b_{\min} and b_{\max} , here -2π and $+2\pi$ respectively. This is not necessary in this test case, but is essential in other cases where PS3D produces unphysical overshoots and undershoots, see in particular Section 3.1. EPIC, by design, does not generate such errors.

As indicated by the dashed lines in Fig. 9, the convergence rate is quadratic in \mathcal{K} , \mathcal{A} and Υ for PS3D and EPIC. The figure shows the relative error with respect to the time-averaged quantities. For the theoretically-conserved quantity \mathcal{T} , the total energy, cubic convergence is found, presumably because the other quantities are only conserved by the linear dynamics. In Table 3 we also list the relative total energy loss in % between the initial time $t = 0$ and the final time $t = 4\pi/\sqrt{2}$ as a function of the grid resolution. Again, we observe an approximate cubic convergence for EPIC and PS3D.

Fig. 10 shows the time evolution of the relative errors in \mathcal{K} , \mathcal{A} and Υ , for various grid resolutions, now with respect to the exact theoretical values. For EPIC, these errors feature a wave period that closely matches the internal wave period, $2\pi/\sqrt{2}$. For PS3D on the other hand, the frequency is twice as high. Notably, in this test case PS3D requires a time step 20 times smaller (i.e. $\alpha = 0.01$) than EPIC to achieve the level of accuracy shown. Otherwise the relative errors increase with a linear trend and do not converge with spatial resolution. The errors reduce by approximately a factor 4, 4 and 3.6 for \mathcal{K} , \mathcal{A} and Υ , respectively, when doubling the grid resolution in EPIC. The higher errors in EPIC are due to the parcel initialisation which cannot exactly represent the initial input fields.

Regarding computational cost, the runtime for both EPIC and PS3D grows as the cube of the vertical grid resolution as shown in Fig. 11. Note that we can only compare the growth rate, but not individual execution times since EPIC was run with 8 MPI (Message Passing Interface) ranks (distributed memory parallelism) and PS3D with 8 OpenMP threads (shared memory parallelism).

Another way to quantify errors is provided in Fig. 12, which shows the vertical vorticity difference $\zeta - \zeta^*$ between the numerical and exact (linear) solution at several times over two wave periods (using $256^2 \times 64$ grid cells). Here this

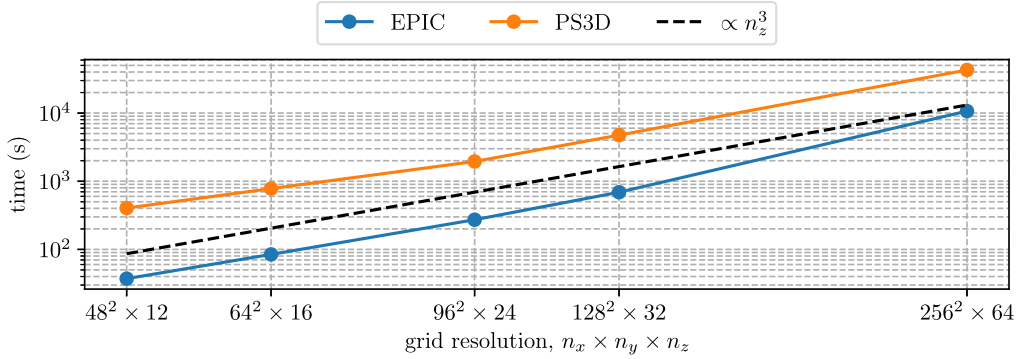


Fig. 11. Cost in CPU seconds versus grid resolution for EPIC and PS3D for the internal wave test case. Note that EPIC is run with 8 MPI ranks and PS3D with 8 OpenMP threads.

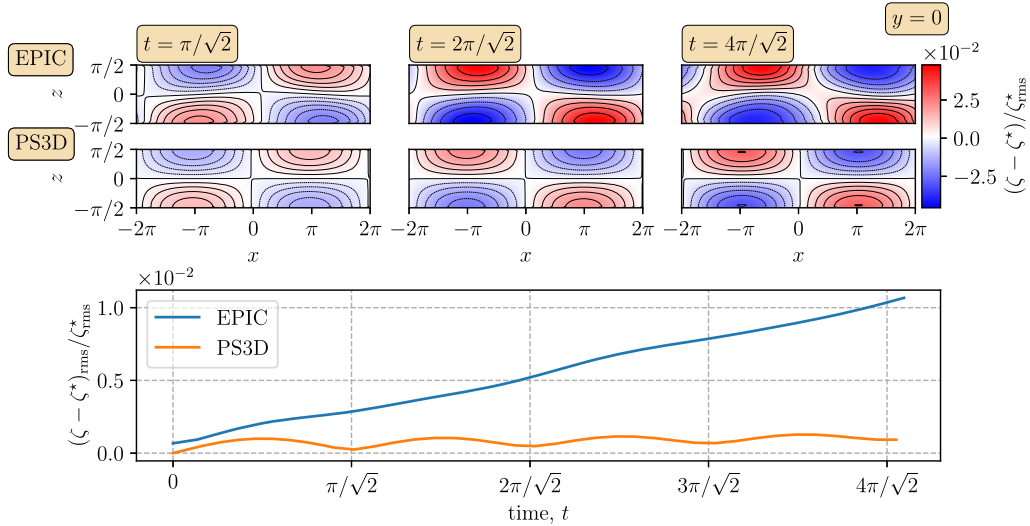


Fig. 12. Cross sections in the xz -plane at $y = 0$ (upper panel) showing the vertical vorticity difference from the analytical result $\zeta^*(t)$ normalised by $\zeta_{rms}^*(t)$, and (lower panel) the r.m.s. error normalised by $\zeta_{rms}^*(t)$. Here results are shown for the simulation with $256^2 \times 64$ cells. Note that the fields are interpolated to exactly match the labelled times.

difference is scaled by ζ_{rms}^* , the r.m.s. value of ζ^* . The larger error in EPIC (at maximum 8.5 times that of PS3D) is partly due to the parcel initialisation, specifically the tri-linear interpolation, which cannot perfectly match the initial fields. Note that the r.m.s. error (lower panel of Fig. 12) remains less than 1.1% of ω_{rms}^* . The growing error in EPIC may be a phase error caused by the imperfect initialisation. This error decreases quadratically with spatial resolution, and is nearly absent in PS3D, which is why the error is so much smaller in those simulations.

3.3. Beltrami flow test case

Next we consider the same Beltrami flow studied in [4]. In the domain $[-\pi/2, \pi/2]^3$, the unperturbed velocity field is

$$\mathbf{u}(\mathbf{x}, 0) = \begin{cases} \frac{1}{4}(\sin z - 3 \cos z) \sin(2x + 2y) \\ \frac{1}{4}(\sin z + 3 \cos z) \sin(2x + 2y) \\ \cos z \cos(2x + 2y), \end{cases} \quad (51)$$

while the corresponding unperturbed vorticity is $\boldsymbol{\omega} = 3\mathbf{u}$. This theoretically steady flow is unstable to small perturbations and breaks down into strongly anisotropic turbulence, with maximum intensity near the boundaries [4]. To force a similar initial instability development for EPIC and PS3D, we explicitly perturb the initial horizontal vorticity components ξ and η by

$$\Delta\xi = A_\xi \cos 2y \cos z \quad \text{and} \quad \Delta\eta = A_\eta \cos 2x \cos z, \quad (52)$$

where we have taken $A_\xi = 1/5$ and $A_\eta = 1/10$ in the results below. This perturbation ensures $\nabla \cdot \boldsymbol{\omega} = 0$.

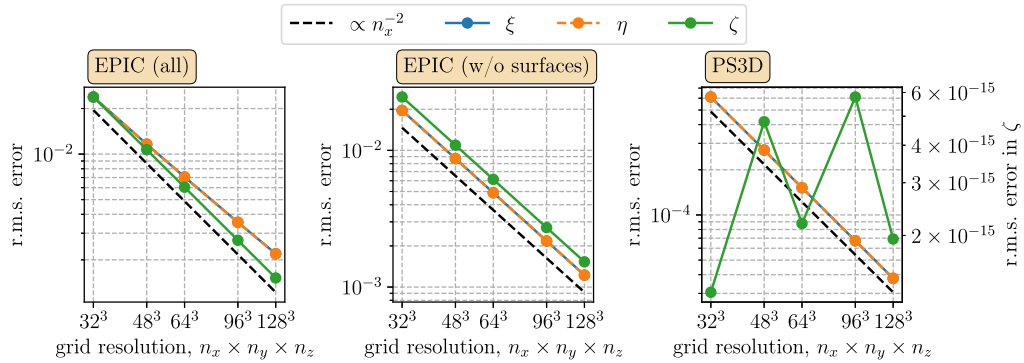


Fig. 13. R.m.s. error in the initial fields compared to the (exact) input fields. For PS3D (rightmost panel), the error in the horizontal vorticity components is due to the solenoidal correction. The vertical vorticity component is not changed by the solenoidal correction and the error remains at machine precision. For EPIC (left and centre panel), this error is mainly due to the parcel-to-grid interpolation. The centre panel excludes the lower and upper surfaces in the r.m.s. error calculation. The error in the horizontal vorticity components is then lower than the error in the vertical component.

Table 4

Comparison of the enstrophy peak value and the slope of the kinetic energy decay between EPIC and PS3D. EPIC is split into parcel (p) and gridded field (f) evaluation. The slope is obtained by a least-squares fit from the data which is 25% around the middle value of kinetic energy.

Grid	Enstrophy peak			Slope of kinetic energy decay		
	EPIC (p)	EPIC (f)	PS3D	EPIC (p)	EPIC (f)	PS3D
32^3	17.12	8.89	11.47	-0.022	-0.0204	-0.0353
48^3	31.28	16.69	19.84	-0.0299	-0.0296	-0.0402
64^3	51.65	27.87	29.29	-0.0309	-0.0297	-0.0435
96^3	99.11	53.07	49.97	-0.0393	-0.0388	-0.0491
128^3	153.96	82.24	73.69	-0.0486	-0.0468	-0.0558

PS3D, developed in [4,16], uses a mixed spectral space representation of the prognostic fields in order to represent shear (non-zero horizontal vorticity) on the free-slip boundaries. The cascade of enstrophy to small scales is diffused in 2D only (by hyperdiffusion). In EPIC, there is no need for hyperdiffusion as the mixing associated with parcel splitting and merging has a weak diffusive effect. Moreover, advection in EPIC is monotone and all fields remain within their theoretical bounds even during mixing.

In both methods, a solenoidal correction is required each time step to maintain $\nabla \cdot \omega = 0$ to machine precision. This correction only modifies (weakly) the horizontal vorticity components [see 4, for details], but it means that even the initial fields (after correction) differ from the input fields. In EPIC, a further error in initialisation occurs due to assigning parcel attributes directly from interpolated input fields. After a parcel-to-grid interpolation, there is generally a slight discrepancy between the initial and input fields, which is most pronounced at the z boundaries (the interpolation is only first-order accurate there, see Section 2.10). This error in initialisation is quantified as a function of resolution for both numerical methods in Fig. 13. PS3D makes essentially no error in ζ (only round-off); otherwise both methods exhibit errors scaling like the square of the grid spacing, as expected. PS3D is not ‘spectrally accurate’ due to the need to do some grid operations in z by finite differences in order to handle shear at the z boundaries [4]. The left and middle panels show that the boundary interpolation in EPIC is significantly less accurate (first-order) than the interior interpolation (second-order), degrading the overall accuracy in capturing the input fields. In summary, the two methods start from slightly different initial fields, and one may anticipate subsequent differences in the flow evolution.

Fig. 14 compares the evolution of kinetic energy and enstrophy between the methods and across a range of resolutions. There is broad agreement in the timing of the instability (i.e. the peak in enstrophy) and in the overall decay in kinetic energy. The PS3D simulations all start from nearly the same value of kinetic energy for all grid resolutions, while the EPIC simulations exhibit lower kinetic energy at coarse resolution – a consequence of the parcel initialisation. Nevertheless, qualitatively the results compare well, with the most significant difference being the magnitude of enstrophy, which is much higher in EPIC than in PS3D (see Table 4 for a quantitative comparison). The middle panel shows the effect of interpolating parcel values to compute these diagnostics from the gridded fields. This interpolation averages the locally intense subgrid vorticity values, resulting in much weaker enstrophy, comparable to that found in PS3D at the same grid resolution. The parcel-based diagnostics in the left panel, however, are appropriate, since they measure the additional degrees of freedom afforded by the parcel representation of subgrid scales.

Enstrophy growth occurs as the flow destabilises, characterised by vorticity stretching and intensification. Cross sections of the vorticity magnitude $|\omega|$ in the horizontal mid-plane $z = 0$ are shown in Fig. 15 at three characteristic times around the peak in enstrophy, and for both the EPIC and PS3D methods. Both methods use 128^3 grid cells. The computed fields

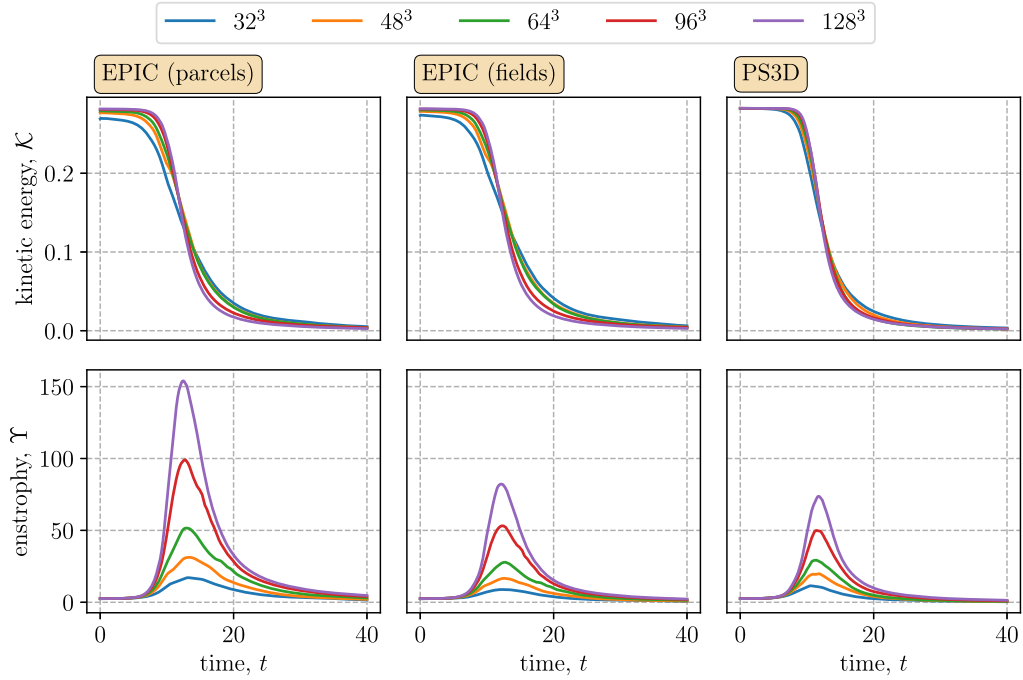


Fig. 14. Evolution of kinetic energy \mathcal{K} and enstrophy Υ for the Beltrami test case.

are closely similar despite the differences in the initial fields. In EPIC, however, the vorticity field is both smaller scale and significantly more intense, consistent with the much higher enstrophy growth seen previously in Fig. 14. A second view is provided in Fig. 16, showing $|\omega|$ in the xz -plane at $y = 0$ at the same times as Fig. 15. EPIC and PS3D again display a similar instability development, breaking down into disorganised turbulence around the time of peak enstrophy. Notably, PS3D exhibits significant vertical Gibbs fringing due to the spectral filter. A final view is provided in Fig. 17 at the bottom surface of the domain ($z = -\pi/2$). The structure of the vorticity field is broader scale but becomes organised into intense fronts (shear line) where the horizontal velocity rapidly switches direction. The PS3D simulation here has more intense boundary vorticity (only at these early to intermediate times), yet the EPIC and PS3D fields are broadly similar in structure. At later times (not shown), the bottom and top surfaces contain the most intense vorticity, mainly because the boundaries restrict vortex line bending and hence dissipation as discussed in [4].

3.4. Moist bubble test case

In the final test case, we compare EPIC with MPIC [2], a PIC method using non-deformable parcels, and with MONC [24], a standard LES model. We examine the evolution of a rising moist buoyant bubble, whose setup is detailed in section 2.4 of [3]. In MONC, we use a Smagorinsky subgrid scheme, and the model configuration is the same as described in [3]. In EPIC, in contrast to those earlier studies, we use a dimensional (rather than dimensionless) setup. However, for ease of comparison with [2,3], we present non-dimensional results when we compare the models below.

We consider a non-rotating flow ($\Omega = \mathbf{0}$) initially at rest within a cubic domain, with x , y and z ranging from 0 to 6280 m. The background state consists of a lower neutrally-stratified layer extending from $z = 0$ to $z = z_b$, where the liquid-water buoyancy $b_l = 0$, and a linearly-stratified region above, where $b_l = N^2(z - z_b)$ (for constant buoyancy frequency N). In the neutral lower layer, the specific humidity is uniform, $q = q_n$, while in the stratified region it decays exponentially, $q = q_n \exp(-\lambda_c(z - z_b))$. Recall, λ_c is the inverse condensation scale height, see Eq. (5).

We next centre a moist, buoyant spherical bubble of radius $R = 800$ m at $(x_o, y_o, z_o) = (3140, 3140, 800)$ m. Within the bubble, we consider a weakly-varying liquid-water buoyancy and specific humidity distribution which smoothly transitions to the environmental values at the edge of the bubble (the smooth edge differs from earlier work using the MPIC model, but makes convergence easier to assess). Specifically, for $\|\mathbf{x}'\| < R$, with $\mathbf{x}' \equiv \mathbf{x} - \mathbf{x}_o$, the liquid-water buoyancy and specific humidity distributions are given by

$$b_l(\mathbf{x}') = b_o \left(1 + \frac{0.3x'y' - 0.4x'z' + 0.5y'z'}{R^2} \right) S(h) \tag{53}$$

$$q(\mathbf{x}') = q_n + (q_o - q_n)S(h), \tag{54}$$

where the edge-smoothing function $S(h)$ takes the form

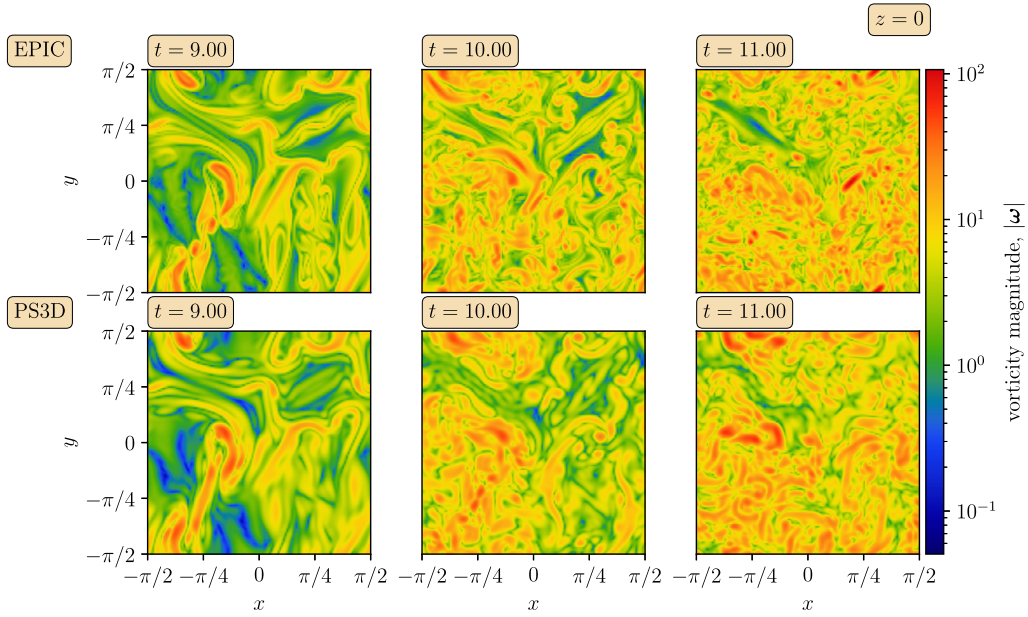


Fig. 15. Cross sections of $|\omega|$ in the xy -plane at $z=0$ at times 9, 10 and 11. The top row is for EPIC and the bottom row is for PS3D.

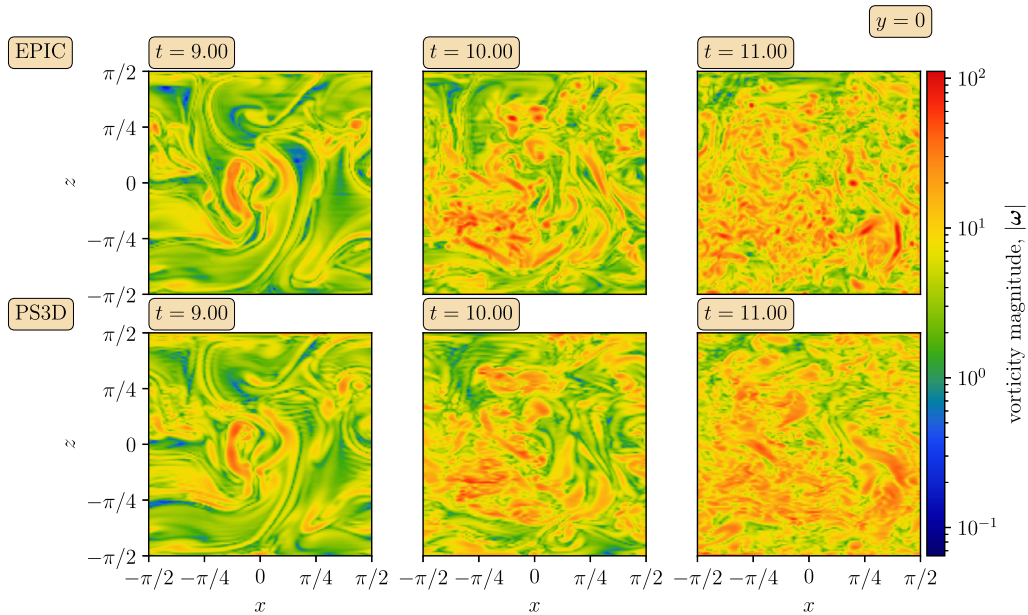


Fig. 16. As in Fig. 15 but for cross sections of $|\omega|$ in the xz -plane at $y=0$.

$$S(h) = \begin{cases} 1 & h \leq 0 \\ 1 - 10h^3 + 15h^4 - 6h^5 & 0 < h < 1 \\ 0 & h \geq 1 \end{cases} \tag{55}$$

with $h \equiv (\|\mathbf{x}'\|/R - f_s)/(1 - f_s)$. Here, $f_s = 0.8$ gives the fraction of the radius of the bubble that is not affected by the edge smoothing. Note that $S(h)$ is continuous at $h = 0$ and 1 , and moreover the first and second derivatives of $S(h)$ vanish there (also $S'(h) = -30h^2(h - 1)^2$, implying S is a monotonically decreasing function of h).

All of the required parameters for this test case and for our model equations, namely Eqs. (2), (3), (5) and (7), are listed in Table 5. These directly correspond to the values used in [3] if one takes a characteristic time scale of 100 s and a characteristic length scale of 1000 m. We use these scales, and a surface humidity $q_0 = 0.015$ kg/kg, to diagnose the flow in terms of non-dimensional quantities, so that figures can be directly compared to those in [2,3].

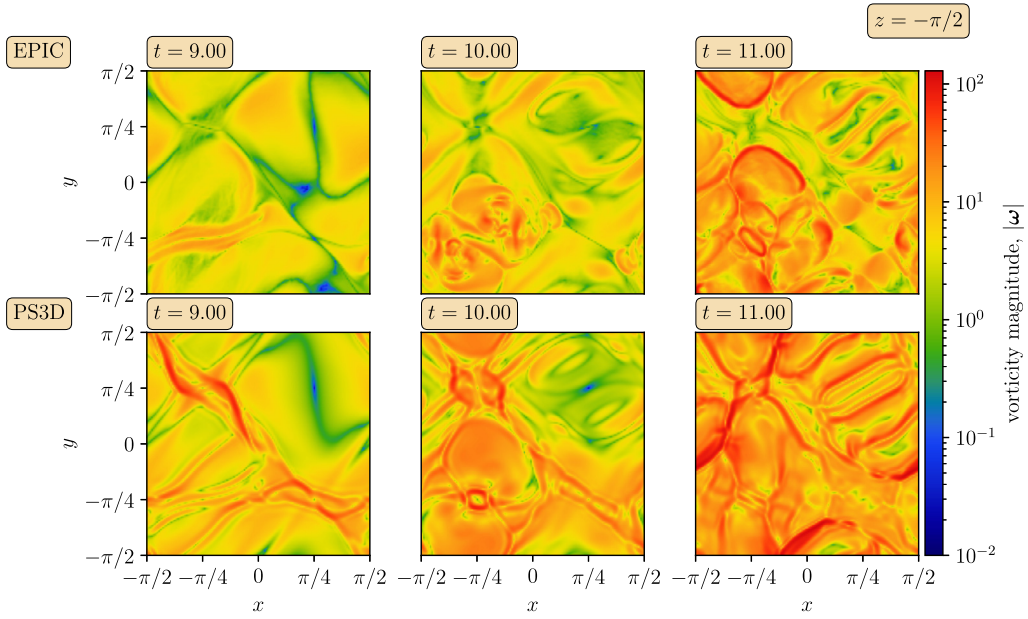


Fig. 17. As in Fig. 15 but for cross sections of $|\omega|$ in the xy -plane at $z = -\pi/2$ (lower surface).

Table 5

Approximate physical constants and parameters used in the moist bubble setup. The exact values are calculated in the programme setup routine.

Quantity	Symbol	Unit	Value
Sat. spec. humidity at ground	q_0	kg/kg	1.5×10^{-2}
Specific humidity in neutral layer	q_n	kg/kg	$\approx 1.1081 \times 10^{-3}$
Specific humidity inside the bubble	q_o	kg/kg	$\approx 1.2313 \times 10^{-3}$
Inverse scale-height	λ_c	1/m	1×10^{-3}
Buoyancy of condensation in Eq. (6)	b_c	m/s ²	1.25
Base of stratified layer	z_b	m	$\approx 2.3822 \times 10^3$
Buoyancy frequency	N	1/s	$\approx 9.7048 \times 10^{-3}$

For MPIC, the parcel attributes are initialised directly from the analytical functions above, following [3]. In EPIC simulations, we found the differences between both methods of initialisation to be minor.

Fig. 18 shows a cross section of the specific humidity field, q , at non-dimensional time $t = 6$, where the left-most panels visualise the individual parcels (only a small part of the domain is shown). The total parcel volume in both simulations is initialised to equal the grid volume, yet notably there are both spaces between parcels and regions of overlap in both simulations. The MPIC simulation has larger empty regions at the edge of the updraft, whereas in EPIC the parcels remain fairly homogeneously distributed. Our earlier work on two-dimensional flows [1] found that both the ellipsoidal shape of the parcels and the use of corrections to the parcel position to enforce incompressibility contribute to the more homogeneous distribution in EPIC.

In general, EPIC resolves the structure of the updraft in significantly more detail. The other panels in the figure show a visualisation using a super-gaussian kernel (as in [1]). The kernel is applied inside an ellipsoid which is scaled by a factor three or five compared to the corresponding ellipsoidal parcel. For EPIC, there are no significant contributions to the grid values beyond three times the parcel edge position. For MPIC, on the other hand, there is a small part of the domain which receives no contributions when using only three times the parcel edge position, and the edge of the updraft looks much more blurred when using a kernel up to five times the parcel radius. We will nevertheless use the more compact kernel in the remainder of this section.

Fig. 19 shows the same cross section of specific humidity q at different grid resolutions. We also show a MONC simulation with the same number of grid points. Note that for large simulations, EPIC simulations are about three times more expensive than MPIC ones (cf. Table 6), whereas MONC simulations have similar cost to MPIC. However, the time step used here in MONC is about 8 times smaller in the 512^3 EPIC and MPIC simulations, and is a relatively conservative estimate (EPIC and MPIC use a fourth-order Runge-Kutta method, whereas MONC uses a leapfrog approach). The relative cost of EPIC and MPIC may increase in longer simulations, where the parcel size distribution will approach equilibrium in the entire domain. As in [3], MPIC and EPIC can resolve the circulation using less than half the number of grid points in each direction. Compared to

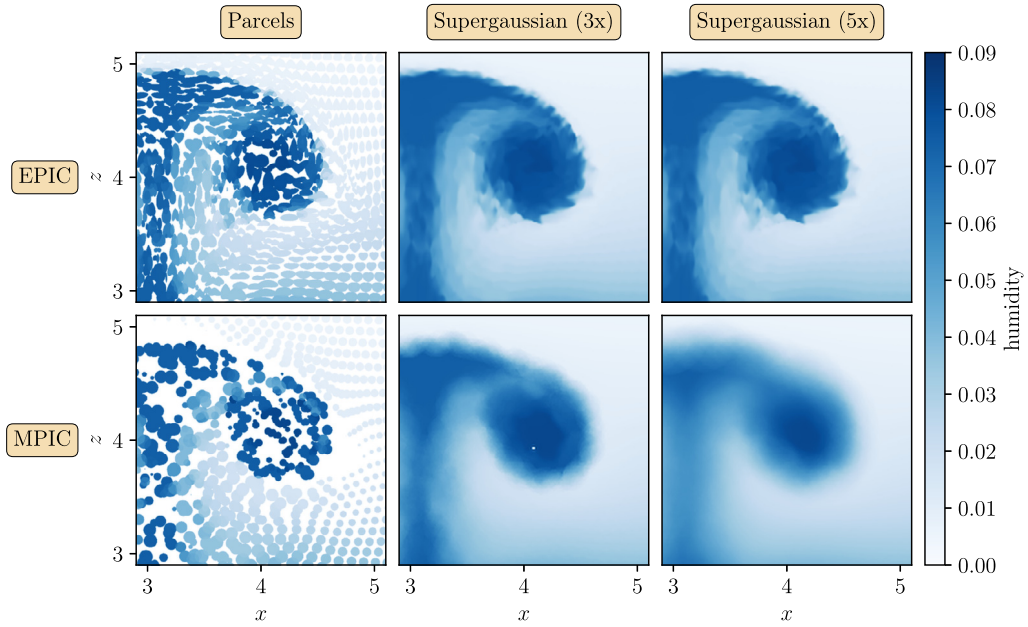


Fig. 18. Cross sections of the specific humidity field q at $t = 6$ in the xz -plane (zoom) in EPIC and MPIC simulations with 32^3 grid points, rendered as parcels with sharp edges, or with the algorithm used in [1] using two different cutoff ellipsoids (see main text).

Table 6

Runtime in seconds of the moist bubble simulations used in this study. The numbers annotated with * indicate I/O-server cores.

Grid	No. of CPU cores			Wall time (s)		
	EPIC	MPIC	MONC	EPIC	MPIC	MONC
32^3	128	128	64 + 8*	11	7	9
64^3	128	128	64 + 8*	86	24	157
128^3	128	256	910 + 114*	1572	244	45
256^3	128	512	910 + 114*	30414	2274	853
512^3	2048	2048	512 + 64*	36698	12006	76999

MPIC, EPIC shows an even sharper contrast between regions of high and low humidity at small scales. Diffusive effects are clearly evident at resolutions less than 512^3 in MONC.

The (normalised) histogram of liquid water specific humidity, namely $q_l = \max(0, q - q_0 \exp(-\lambda_c z))$, at $t = 6$ in the three models is shown in Fig. 20. We used bilinear interpolation to obtain better sampling of the MONC simulations with 32^3 and 64^3 points, increasing the number of sampling points in each direction by a factor of 5 and 3 respectively, so all MONC fields are sampled using at least 128^3 points (this is a relatively simple approach; more advanced methods for obtaining histograms are discussed in [25]). For comparison, the results from all 512^3 simulations are shown in the last panel. Both EPIC and MPIC show a peak corresponding to parcels with high liquid water content q_l at low resolution, although the height of this peak is lower in EPIC as compared to MPIC. This indicates there is still scope for improving the representation of small-scale mixing in the lower-resolution EPIC simulations. MONC has many grid points with small amounts of liquid water at low resolution, which are absent in EPIC and in high-resolution MONC reference simulations. This is an advantage of the Lagrangian representation of mixing in EPIC. At the highest resolution, EPIC has more parcels with high q_l compared to MONC, but the distributions are beginning to look similar, despite the variations in EPIC being at much smaller scales, as seen in the cross-sections. MONC under-represents high q_l , as can be seen by the distribution extending outwards with increasing resolution. The significantly greater diffusion in MONC compared to EPIC is visible in the lower right panel, where the MONC distribution is shifted to lower q_l .

A quantitative assessment of the convergence of the liquid water specific humidity histogram at $t = 6$ is presented in Fig. 21. Here, the root-mean-square error over all bins in the liquid water specific humidity distribution is plotted (except for the bin which contains the 0 value, and using linearly spaced bins between $q_l = 0$ and 0.1). In each of the panels of Fig. 21, a different high-resolution model is used as the reference model when calculating the root-mean-square error, so that both convergence of a single model and convergence between models can be assessed.

Each model gradually converges to its own liquid water specific humidity histogram at high resolution. Surprisingly, at the highest resolution, EPIC and MONC have smaller differences using this metric than EPIC and MPIC, even though EPIC and MPIC are both PIC models. This indicates that the mixing occurring in EPIC is substantially different from that occurring

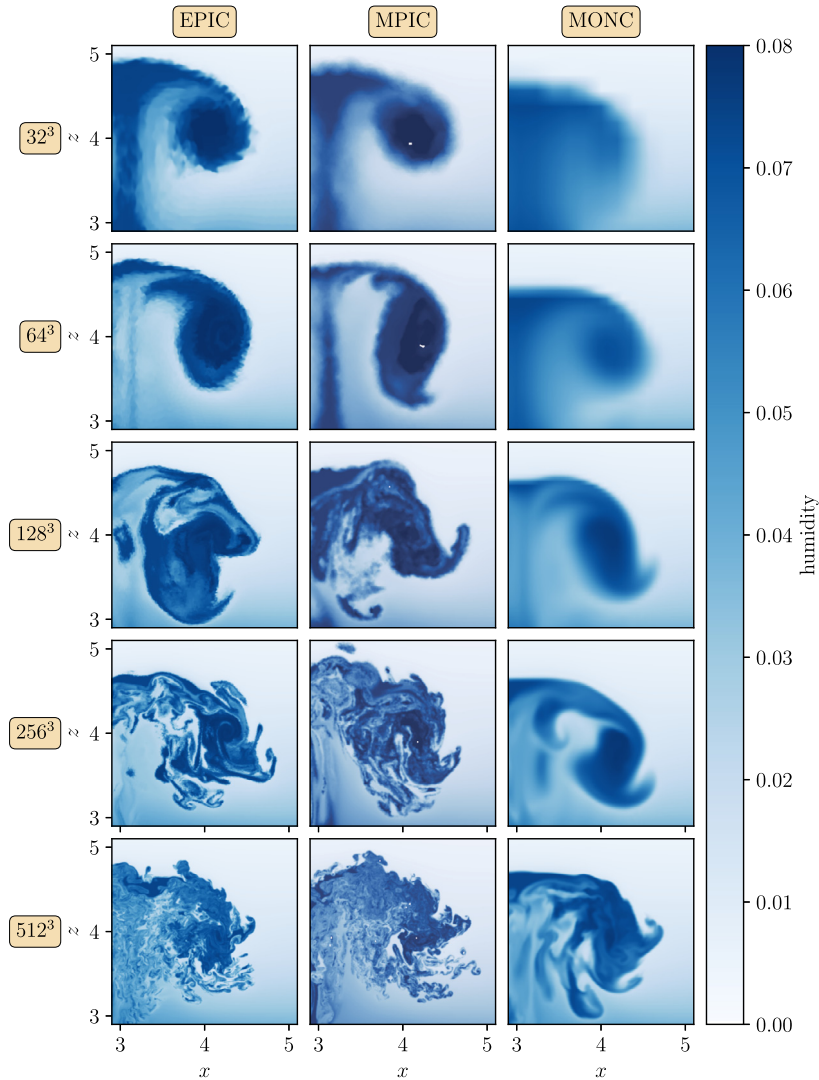


Fig. 19. Cross sections of the specific humidity field q in the xz -plane (zoom) in EPIC, MPIC and MONC simulations with different numbers of grid points.

in MPIC. The difference between the models at the highest resolution is generally smaller than the difference between a high-resolution and a low-resolution simulation of the same model, although there is still a difference between MPIC and MONC at the highest resolution.

The histogram of vorticity magnitude, $|\omega|$, at $t = 6$ is plotted in Fig. 22. The vorticity in MONC is computed from gradients in the velocity field and no subsampling is applied to the MONC data. The maximum values of vorticity magnitude in MONC are significantly smaller than the maximum parcel-based values in MPIC and EPIC (the histograms are plotted on a log-log scale). MONC and MPIC have more very small values ($< 10^{-5}$) of $|\omega|$ than EPIC. For $|\omega|$ between 10^{-3} and 1 (these values largely correspond to regions outside the cloud), the histograms are in good agreement, both between models and across resolutions. For high $|\omega|$, the significantly greater diffusion in MONC is visible by the shift in the distribution to lower values, creating a bulge at moderate values of $|\omega|$. Sampling on a grid also mixes weak vorticity more effectively, giving the impression that there are many weak vorticity values. In EPIC, we sample the PDF (probability density function) using parcels, and this avoids the averaging associated with grid sampling. The spike in the vorticity histogram of EPIC at low $|\omega|$ is due to the mean vorticity correction, cf. Eq. (17). This correction ensures that there is no global vorticity production, i.e. the mean vorticity remains constant. MPIC and MONC do not apply this correction, and therefore their histograms do not exhibit this spike. Nor does EPIC without this correction, and the histogram differs from that plotted in Fig. 22 only for extremely weak vorticity, $|\omega| < 10^{-5}$ (not shown). These values have a negligible influence on the dynamics of the flow.

To emphasise the wide range of scales captured by EPIC, in Fig. 23 we show a cross section of the specific humidity field q in the xz -plane at $t = 6$ including the intersected ellipses of the corresponding ellipsoids. The figure also contains four cascaded zooms having widths of 64, 32, 16 and 8 grid cells. The individual ellipsoids (or their elliptical cross sections) are

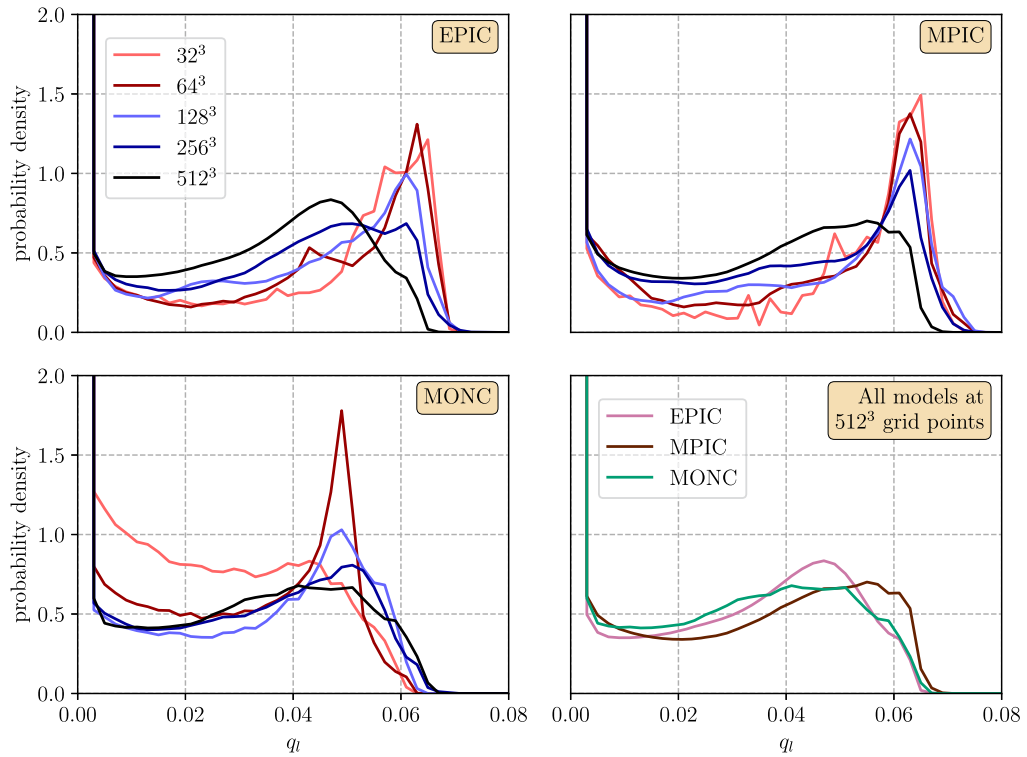


Fig. 20. Liquid water specific humidity histogram in EPIC, MPIC and MONC simulations with different numbers of grid points at $t = 6$. The bin width is 0.002.

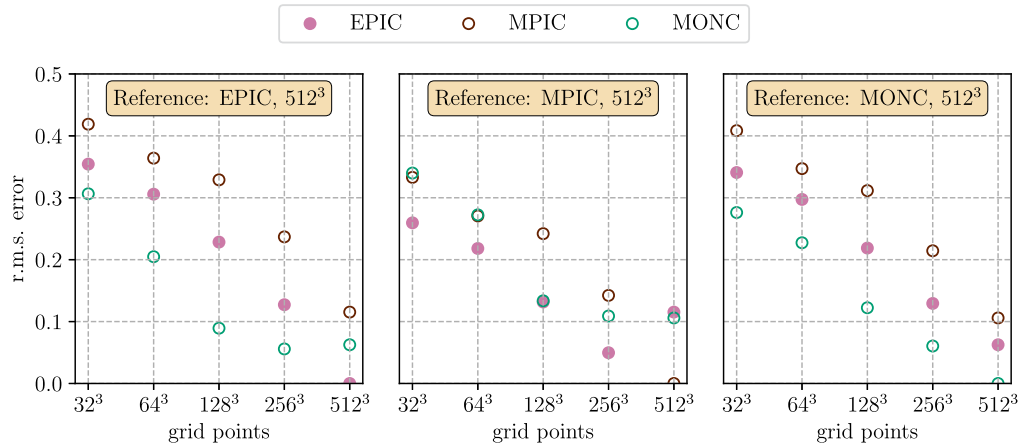


Fig. 21. Root-mean-square error of the EPIC, MPIC and MONC liquid water specific humidity histograms at $t = 6$, as compared to the histogram in the highest-resolution reference EPIC, MPIC and MONC simulations.

visible in the finest zooms. Many ellipsoids fill each grid cell, providing an explicit subgrid model. This ability to resolve at, and below, the grid scale enables EPIC to employ modest grid resolution to capture highly-turbulent flow structures with minimal diffusion – and this process is entirely controlled by parcel splitting and merging, not by any explicit numerical diffusion.

4. Discussion

In this paper, we have extended the Elliptical Parcel-In-Cell (EPIC) method, first developed for two-dimensional flows in [1], to three dimensions. This offers an accurate new, predominantly Lagrangian, approach to simulating very high Reynolds number turbulent, rotating stratified flows – including simple thermodynamics and moisture. Like in two dimensions, the ellipsoidal parcels used to represent key flow attributes (e.g. buoyancy and vorticity) generally deform in the local velocity

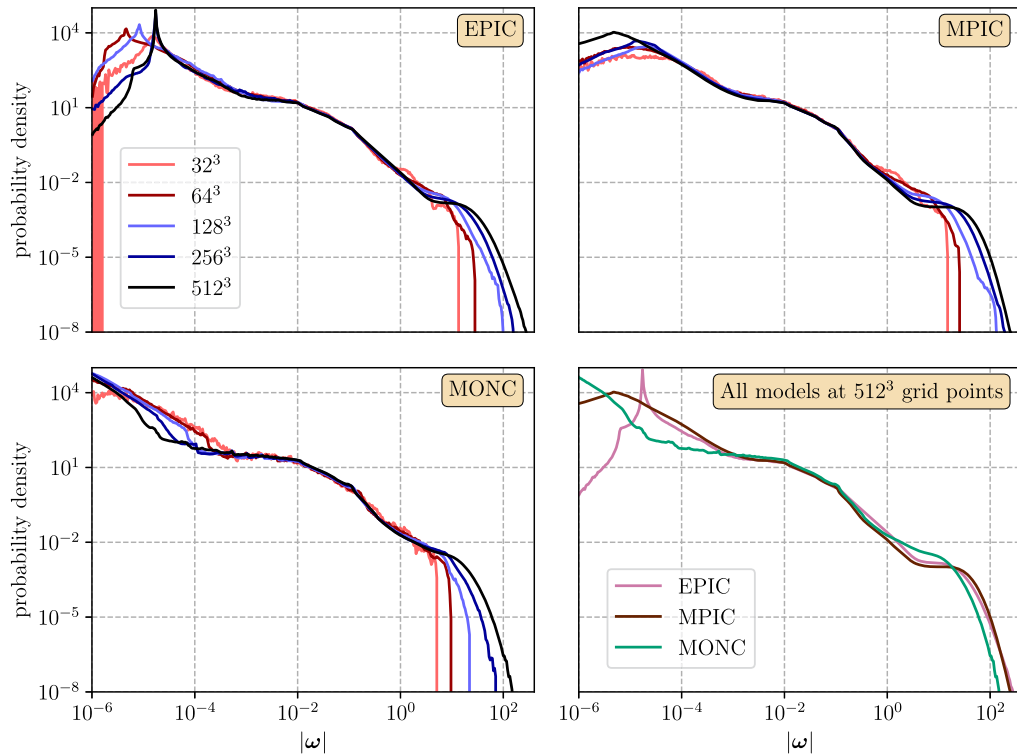


Fig. 22. Histogram of the vorticity magnitude, $|\omega|$, in the EPIC, MPIC and MONC simulations with different numbers of grid points at $t = 6$. The bins widths correspond to 5% increases in vorticity magnitude (i.e. log spacing).

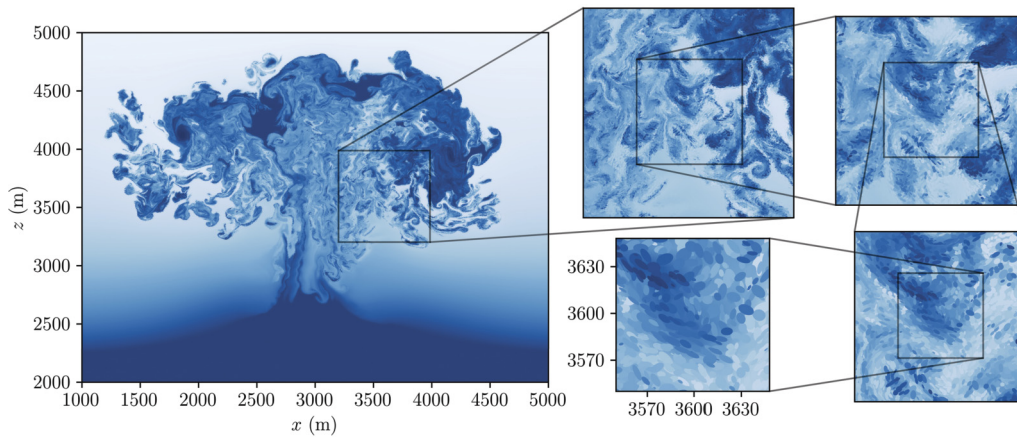


Fig. 23. Cross section of the specific humidity field q in EPIC at $y = 3140$ m in the xz -subdomain $[1000, 5000] \times [2000, 5000]$ m with intersected ellipses at $t = 6$. The zoomed windows correspond to 64, 32, 16 and 8 grid cell widths in both directions.

strain, and furthermore may split and merge at sub-grid scales, thereby providing a natural mixing mechanism. The three-dimensional extension of the associated parcel operations is straightforward and involves only minor code changes. Also, the ellipsoidal parcel shape requires just three additional attributes per parcel. Interpolation from parcel attributes to gridded values (and vice-versa) uses four support points instead of the two needed in the two-dimensional model. Determining the location of these support points requires the solution of a 3×3 eigensystem per parcel, in effect a cubic equation for the squared semi-axis lengths. The famous analytical solution of Cardano (1545) [11] unfortunately does not provide sufficient accuracy for a typical distribution of parcel shapes. Instead, we use the algorithm by Scherzinger and Dohrmann [15] which is comparable in speed to the Cardano method and yet meets our accuracy and robustness criteria.

Here we have benchmarked the method against four diverse test cases that exhibit relevant characteristics of atmospheric and oceanic flows. We have also compared the results with a LES (Large Eddy Simulation), a DNS (Direct Numerical Simulation) and another PIC (Parcel-In-Cell) model. For comparable grid resolutions, EPIC exhibits closely similar conserva-

tion properties, i.e. comparable decay of energy, but a significant improvement in modelling vorticity. As measured by the mean-square vorticity (enstrophy), EPIC permits much higher values (roughly two times larger), owing to its less aggressive treatment of mixing (and lack of explicit numerical diffusion). Another benefit of EPIC is that it preserves monotonicity of tracers whose extrema are bounded theoretically. By construction, the parcel splitting and merging responsible for mixing in EPIC either preserve attributes or create new values between existing ones. This is a particularly important property of a numerical method when tackling problems involving fields that must remain positive-definite (e.g. moisture here, but more generally chemical and biological tracers).

A major challenge in the development of the three-dimensional version of the EPIC model was the accurate assignment of z boundary grid point values during the parcel-to-grid interpolation. As in the two-dimensional model, the boundary values must be doubled after interpolating the parcel properties onto the grid to ensure that the total parcel volume, and the volume-weighted sum of each parcel attribute, are equal to their gridded counterparts. This however results in only first-order accuracy at the boundaries for fields whose derivatives are non-zero there (and not prescribed). Any attempt to improve the accuracy near the boundaries using higher order extrapolation – rather than linear – caused a non-physical energy growth in the Beltrami test case due to greatly increased near-boundary vorticity values that are subsequently swept into the domain. Moreover, such extrapolation appears to be inconsistent with keeping sums over parcels equal to sums over grid points. Two other approaches we are considering to improve the boundary interpolation are (1) the use of 3D ‘ghost’ parcels in vertical halo cells, and (2) two-dimensional, compressible surface parcels (ellipses) that live in the boundary grid layer. We are currently developing the surface parcel approach, and first tests with the Beltrami flow are returning promising results. However, at present there is no mechanism to mix surface parcels with those in the interior. This may be problematic for parcel attributes such as buoyancy, where for example a descending cold region should be able to cool the surface. We are currently formulating a way to mix the attributes of parcels in the grid layer adjacent to each boundary with those of surface parcels. This requires specifying a new mixing rate parameter.

We also intend to improve the parcel initialisation developed in [1]. This currently suffers from higher errors at the vertical boundaries, leading to a discrepancy in the input field values after interpolating parcel attributes to field values. This makes it difficult to compare with analytical solutions, like the linear internal wave test case presented in this paper. The surface parcel approach described above would appear to overcome this issue, leading to a consistent interpolation scheme, an integral part of the EPIC method.

Future developments include a more realistic moist physics model, the addition of precipitation, and a boundary layer to model surface fluxes. We also aim to move from the Boussinesq approximation to the anelastic formulation to be able to study atmospheric convection more realistically (including the near exponential decay of density with height).

Code availability

EPIC and PS3D are open-source and freely available under the BSD 3-Clause license. The results in this paper are obtained with EPIC version 0.13.1 [26] and PS3D version 0.0.13 [16]. The MONC and MPIC source codes used for this publication as well as plot scripts are found in [27].

Third party libraries

Instead of the HDF5 API, EPIC now uses the netCDF API and follows the Climate and Forecast (CF) metadata convention [28,29] (version CF-1.8). Therefore, netCDF C (version 4.9.0) and netCDF Fortran (version 4.5.3) must be installed in addition to HDF5 (version 1.12.2.1). Note that the 3D model uses distributed memory parallelism. The results in this publication are generated with the Cray MPI Fortran library (version 8.1.23).

CRediT authorship contribution statement

Matthias Frey: Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **David Dritschel:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Supervision, Writing – original draft, Writing – review & editing. **Steven Böing:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All source codes are publicly available.

Acknowledgements

This research is supported by the UK Engineering and Physical Sciences Research Council (grant numbers EP/T025301/1 and EP/T025409/1). This work used the ARCHER2 UK National Supercomputing Service (<https://www.archer2.ac.uk>) which is funded and managed by UKRI and supported by EPCC, HPE and The University of Edinburgh.

Appendix A. Parcel splitting

Here we derive the parcel separation and shapes of two equal parcels which just split from a ‘parent’ parcel. The approach we follow here is the direct 3D extension to that in [30] and the 2D EPIC model [1]. Without loss of generality, we consider the case where the parent parcel’s centre is at the origin and its semi-axes align with the coordinate axes. The general case follows from a rotation and translation.

It is simplest to use the following, non-orthogonal coordinate system for an ellipsoid with semi-axis lengths a , b and c ,

$$x = ar \sin \theta \cos \phi \quad (\text{A.1})$$

$$y = br \sin \theta \sin \phi \quad (\text{A.2})$$

$$z = cr \cos \theta \quad (\text{A.3})$$

for which $0 \leq \theta \leq \pi$, $0 \leq \phi \leq 2\pi$ and $0 \leq r \leq 1$ over the ellipsoid interior. The Jacobi transformation matrix is then

$$J = \begin{pmatrix} a \sin \theta \cos \phi & ar \cos \theta \cos \phi & -ar \sin \theta \sin \phi \\ b \sin \theta \sin \phi & br \cos \theta \sin \phi & br \sin \theta \cos \phi \\ c \cos \theta & -cr \sin \theta & 0 \end{pmatrix} \quad (\text{A.4})$$

with determinant $|J| = abc r^2 \sin \theta$. The infinitesimal volume element is thus

$$dV = dx dy dz = |J| dr d\theta d\phi = abc r^2 \sin \theta dr d\theta d\phi. \quad (\text{A.5})$$

As in the 2D model, we want to preserve all spatial moments up to second order after splitting. The zeroth moment is volume, the first moment is proportional to the centroid, and the second moments are the spatial integrals of products of the Cartesian coordinates. The integrals of xy , xz and yz are zero due to symmetry. The only non-zero second moments are

$$\int_V x^2 dV = \int_0^1 \int_0^\pi \int_0^{2\pi} a^3 bc r^4 \sin^3 \theta \cos^2 \phi dr d\theta d\phi = \frac{4\pi a^3 bc}{15} \quad (\text{A.6})$$

$$\int_V y^2 dV = \int_0^1 \int_0^\pi \int_0^{2\pi} ab^3 c r^4 \sin^3 \theta \sin^2 \phi dr d\theta d\phi = \frac{4\pi ab^3 c}{15} \quad (\text{A.7})$$

$$\int_V z^2 dV = \int_0^1 \int_0^\pi \int_0^{2\pi} abc^3 r^4 \cos^2 \theta \sin \theta dr d\theta d\phi = \frac{4\pi abc^3}{15}. \quad (\text{A.8})$$

When the parent ellipsoid is split along its major axis, here the x axis, the two child parcels will be centred on the x axis at the positions $x = \pm h$, with h to be determined (this preserves the first moment). Let a' , b' and c' denote the semi-axis lengths of the identical child parcels. Then, the non-zero second moments of the child parcel centred at $x = h$ are

$$\int_{V'} (h+x)^2 dV' = \frac{4\pi a' b' c'}{15} [(a')^2 + 5h^2] \quad (\text{A.9})$$

$$\int_{V'} (y')^2 dV' = \frac{4\pi a' (b')^3 c'}{15} \quad (\text{A.10})$$

$$\int_{V'} (z')^2 dV' = \frac{4\pi a' b' (c')^3}{15}. \quad (\text{A.11})$$

The same results follow for the child parcel centred at $x = -h$. Thus, we need to double these values and equate them to the results in Eqs. (A.6), (A.7) and (A.8). Moreover, conservation of total volume (the zeroth moment) implies $2a'b'c' = abc$; hence, equating the y^2 moment integrals gives

$$\frac{8\pi a'(b')^3 c'}{15} = \frac{8\pi ab^3 c}{15} \iff b' = b \quad (\text{A.12})$$

Similarly, equating the z^2 moment integrals gives $c' = c$. In other words, the middle and minor axes lengths remain unchanged. It then follows from volume conservation that $a' = a/2$ must hold. As in the 2D model, only the major axis length changes. The half-distance between the child parcel centres, h , is obtained by equating the x^2 moment integrals:

$$\frac{4\pi a^3 bc}{15} = \frac{8\pi}{15} a' b' c' [(a')^2 + 5h^2] \iff a^3 = a[a^2/4 + 5h^2] \iff h = \frac{\sqrt{3}a}{2\sqrt{5}}. \quad (\text{A.13})$$

The separation distance of the child parcels, $2h$, is the only thing which differs between the 2D and 3D models. In 2D we found $2h/a = \sqrt{3/4}$, while in 3D we find $2h/a = \sqrt{3/5}$.

Appendix B. Parcel merging

Equation (2.14) in [7] defines the \mathbf{B} matrix by the following integral over the ellipsoidal volume,

$$\mathbf{B} = \frac{5}{V} \int_V (\mathbf{x} - \mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c)^\top dV \quad (\text{B.1})$$

where $V = 4\pi abc/3$ and \mathbf{x}_c is the centre of the ellipsoid. Note that the shape matrix \mathbf{B} is proportional to the second-order spatial moments. Now consider the merger of two ellipsoids \mathbf{B}_1 and \mathbf{B}_2 with centres at $\mathbf{x}_{c,1}$ and $\mathbf{x}_{c,2}$, respectively. The shape matrix of the merged parcel is tentatively calculated from

$$\mathbf{B}^* = \frac{5}{V} \left[\int_{V_1} (\mathbf{x} - \mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c)^\top dV + \int_{V_2} (\mathbf{x} - \mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c)^\top dV \right] \quad (\text{B.2})$$

where $V = V_1 + V_2$ and $\mathbf{x}_c = (V_1 \mathbf{x}_{c,1} + V_2 \mathbf{x}_{c,2})/V$ to ensure the zeroth and first-order spatial moments are preserved. The tentative shape matrix \mathbf{B}^* preserves the second-order moments as well; however, its determinant is not generally equal to $(abc)^2$, as required for consistency. As preserving volume is deemed most critical, the final form of the shape matrix is scaled as in Eq. (23).

All of the integrals in Eqs. (B.1) and (B.2) can be calculated analytically. In the following we simplify the analysis by using the second-order moments directly:

$$\mathbf{M}_j \equiv \int_{V_j} (\mathbf{x} - \mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c)^\top dV. \quad (\text{B.3})$$

We further shift the coordinates so that $\hat{\mathbf{x}} = \mathbf{x} - \mathbf{x}_{c,j}$ and $\hat{\mathbf{x}}_{c,j} = \mathbf{x}_c - \mathbf{x}_{c,j}$. The coordinates $\hat{\mathbf{x}}$ are now centred on parcel j . We can then rewrite

$$\mathbf{M}_j = \int_{V_j} (\hat{\mathbf{x}} - \hat{\mathbf{x}}_{c,j})(\hat{\mathbf{x}} - \hat{\mathbf{x}}_{c,j})^\top dV. \quad (\text{B.4})$$

Since $\hat{\mathbf{x}}$ is now centred on $\mathbf{0}$, we have

$$\mathbf{M}_j = V_j \hat{\mathbf{x}}_{c,j} \hat{\mathbf{x}}_{c,j}^\top + \frac{1}{5} V_j \mathbf{B}_j \quad (\text{B.5})$$

after using the general relation Eq. (B.1) from [7]. Hence, it follows that

$$\mathbf{B}^* = 5(\mu_1 \hat{\mathbf{x}}_{c,1} \hat{\mathbf{x}}_{c,1}^\top + \mu_2 \hat{\mathbf{x}}_{c,2} \hat{\mathbf{x}}_{c,2}^\top) + \mu_1 \mathbf{B}_1 + \mu_2 \mathbf{B}_2 \quad (\text{B.6})$$

where $\mu_j = V_j/V$. This expression readily extends to any number of merging parcels.

Appendix C. A stochastic model for parcel size distributions

In this section, we argue that the parcel size distributions in EPIC are largely determined by the choice of V_{\min} and a_{\max} alone, and not by the nature or even the dimensionality of the flow. This distribution can be understood from a simple stochastic model of mixing parcels. The current version of EPIC uses a_{\max} , but the stochastic model uses a maximum volume V_{\max} (as an earlier version on EPIC did; the use of a_{\max} in the current version allows for stable simulations with parcels that are on average larger than when using V_{\max}). The stochastic model is used to argue the size distribution will hardly change for values of V_{\max} greater than eight V_{\min} (in earlier EPIC simulations, this ratio was 8.14).

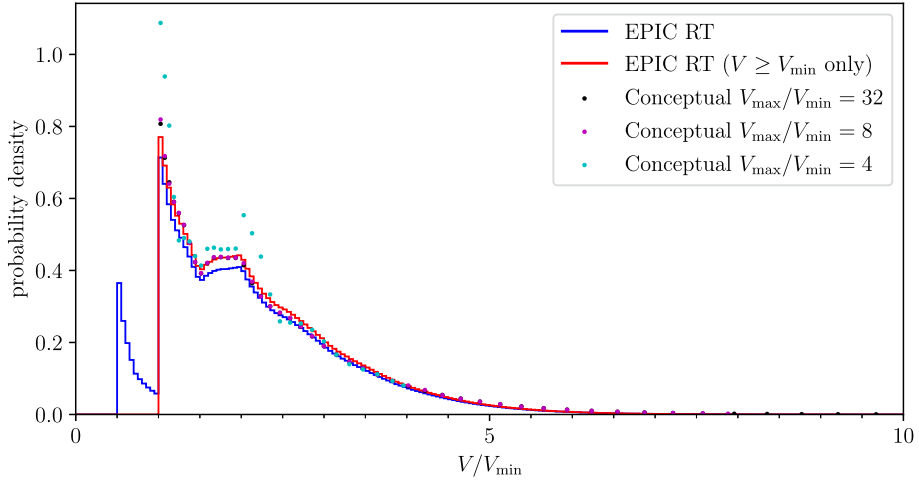


Fig. C.24. Parcel size distribution in the stochastic model, for different values of V_{\max}/V_{\min} , and for a 384^3 Rayleigh-Taylor (RT) flow simulation at $t = 7$.

The simple stochastic model contains a set of parcels, which are equipped only with a volume V and no other properties. Volumes are initialised with a uniform random value between V_{\min} and $2V_{\text{init}} - V_{\min}$, where V_{init} is the mean initial volume (here chosen to be $2V_{\min}$).

A single step of the stochastic model applies the following rules:

1. One parcel is randomly chosen for splitting.
2. If the split parcels are smaller than V_{\min} , they both merge with a randomly chosen parcel from the set (note: there is no information on the location of parcels in the model).
3. If either of the merged parcels has a volume greater than V_{\max} , then this parcel is split (for simplicity, we choose V_{\max} to be at least twice V_{\min} , so that this split will not result in parcels that need to be merged again).

Note that in the stochastic model, the parameters which set the initial and maximum volume can be normalised by V_{\min} . For the results presented here, we use 100000 initial parcels. We sample the distribution every 100000 steps, and consider the distributions between sample times 4000 and 8000 (by this time, any effects of the initial size distribution have disappeared).

The parcel size distribution resulting from a stochastic simulation with these rules is shown in Fig. C.24, and is compared against that from a Rayleigh-Taylor flow simulation on a 384^3 grid with EPIC at time $t = 7$. Because of EPIC's finite time step, and the fact that splitting occurs at the end of the time step, some of the parcels in EPIC are smaller than V_{\min} . If we disregard these parcels, we find excellent agreement between EPIC and the conceptual model using V_{\max} of about $8V_{\min}$. The conceptual model gives a very similar distribution using $V_{\max} = 32V_{\min}$, as few parcels grow to a size larger than $8V_{\min}$ when both splitting and merging occur randomly. If we use $V_{\max} = 4V_{\min}$ in the conceptual model, we find that preventing the larger parcels from forming results in relatively more smaller parcels as compared to EPIC, as would be expected.

Despite the simplicity of the stochastic model, it gives a good approximation of the parcel size distribution in the full EPIC model. The non-monotonic decline of the distribution can be explained using the conceptual model: when a split parcel with volume of at least $V_{\min}/2$ combines with a large parcel, the volume of the merged parcel will be at least $3V_{\min}/2$. Larger parcels are slightly more rare in EPIC as compared to the stochastic model. This is likely a result of the volume corrections in EPIC: these make it less likely for parcels to be close to (and hence merge with) a large neighbour, compared to randomly-distributed parcels that have no regard for local parcel clustering.

Appendix D. Intersection of an ellipsoid with a plane

In this section we describe a method to calculate the intersection of a plane with a general ellipsoid,

$$(\mathbf{x} - \mathbf{x}_c)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}_c) = 1, \quad (\text{D.1})$$

centred at $\mathbf{x}_c = (x_c, y_c, z_c)$. In the following we use $\mathbf{A} = \mathbf{B}^{-1}$. A plane is defined by the equation

$$\mathbf{n}^T (\mathbf{x} - \mathbf{p}) = 0 \quad (\text{D.2})$$

where \mathbf{n} denotes the unit normal vector to the plane and \mathbf{p} a point on that plane.

First we obtain a rotation matrix \mathbf{R} such that $\hat{\mathbf{e}}_z = \mathbf{R}\mathbf{n}$ using Euler's finite rotation formula [31] (or Euler-Rodrigues formula, see [32, and references therein])

$$\mathbf{R} = \mathbf{I} + \mathbf{W} \sin \theta + \mathbf{W}^2 (1 - \cos \theta) \quad (\text{D.3})$$

with identity matrix \mathbf{I} and $\mathbf{W} = \mathbf{w}\mathbf{w}^\top - \mathbf{I}$. Here, the rotation axis and angle are $\mathbf{w} = \mathbf{n} \times \hat{\mathbf{e}}_z$ and $\theta = \cos^{-1}(\mathbf{n}^\top \hat{\mathbf{e}}_z)$, respectively.

The ellipsoid is now centred at $\tilde{\mathbf{x}}_c = \mathbf{R}\mathbf{x}_c$ with $\tilde{\mathbf{A}} = \mathbf{R}\mathbf{A}\mathbf{R}^\top$ (a similar expression applies for $\tilde{\mathbf{B}}$). After rotation the plane is simply $\tilde{z} = \tilde{p}_z$ where \tilde{p}_z is the third component of $\tilde{\mathbf{p}} = \mathbf{R}\mathbf{p}$. After translating the centre of the ellipsoid to the origin, we obtain the equation for the centre of the intersected ellipse from $\mathbf{x}_o = \mathbf{x}_c + \mathbf{R}^\top \tilde{\mathbf{x}}_o$, where

$$\tilde{x}_o = \frac{\tilde{B}_{13}}{\tilde{B}_{33}} \tilde{z}_o, \quad \tilde{y}_o = \frac{\tilde{B}_{23}}{\tilde{B}_{33}} \tilde{z}_o \quad \text{and} \quad \tilde{z}_o = \tilde{p}_z - \tilde{z}_c. \quad (\text{D.4})$$

Notably, \tilde{x}_o and \tilde{y}_o only involve ratios of the co-factor matrix $\text{cof}(\tilde{\mathbf{A}})$ of the matrix $\tilde{\mathbf{A}}$. This co-factor matrix is used to calculate the inverse of $\tilde{\mathbf{A}}$, here $\tilde{\mathbf{B}}$, from $\tilde{\mathbf{B}} = \text{cof}(\tilde{\mathbf{A}})/\det(\tilde{\mathbf{A}})$.

We next use the fact that the centred ellipse in implicit form, i.e.

$$\tilde{A}_{11}\tilde{x}^2 + 2\tilde{A}_{12}\tilde{x}\tilde{y} + \tilde{A}_{22}\tilde{y}^2 + 2\tilde{A}_{13}\tilde{z}_o\tilde{x} + 2\tilde{A}_{23}\tilde{z}_o\tilde{y} + \tilde{A}_{33}\tilde{z}_o^2 = 1, \quad (\text{D.5})$$

is equivalent to

$$\begin{pmatrix} \tilde{x} - \tilde{x}_o \\ \tilde{y} - \tilde{y}_o \end{pmatrix}^\top \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{12} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} \tilde{x} - \tilde{x}_o \\ \tilde{y} - \tilde{y}_o \end{pmatrix} = S \quad (\text{D.6})$$

with $S = 1 - \tilde{z}_o^2/\tilde{B}_{33}$. The $\tilde{\mathbf{B}}_{2 \times 2}$ matrix of the intersected ellipse is therefore simply the inverse of the scaled 2×2 matrix in Eq. (D.6), i.e.

$$\tilde{\mathbf{B}}_{2 \times 2} = \frac{S}{\tilde{A}_{11}\tilde{A}_{22} - \tilde{A}_{12}^2} \begin{pmatrix} \tilde{A}_{22} & -\tilde{A}_{12} \\ -\tilde{A}_{12} & \tilde{A}_{11} \end{pmatrix}. \quad (\text{D.7})$$

Using $\tilde{\mathbf{A}} = \text{cof}(\tilde{\mathbf{B}})/\det(\tilde{\mathbf{B}})$, we can rewrite this fully in terms of $\tilde{\mathbf{B}}$:

$$\tilde{\mathbf{B}}_{2 \times 2} = S \left[\begin{pmatrix} \tilde{B}_{11} & \tilde{B}_{12} \\ \tilde{B}_{12} & \tilde{B}_{22} \end{pmatrix} - \frac{1}{\tilde{B}_{33}} \begin{pmatrix} \tilde{B}_{13} \\ \tilde{B}_{23} \end{pmatrix} \begin{pmatrix} \tilde{B}_{13} & \tilde{B}_{23} \end{pmatrix} \right]. \quad (\text{D.8})$$

The condition for the plane to intersect the ellipsoid is $S > 0$ or $\tilde{B}_{33} > \tilde{z}_o^2$. When this condition is met marginally, i.e. when $\tilde{B}_{33} - \tilde{z}_o^2 \rightarrow 0$, the elements of $\tilde{\mathbf{B}}_{2 \times 2}$ tend to zero, implying that the ellipse shrinks to a point.

Note, the resulting ellipse is plotted using the eigenvalues and eigenvectors of $\tilde{\mathbf{B}}_{2 \times 2}$ as in [1], and by rotating the coordinates back to the original frame of reference.

Appendix E. Code performance and parameter choices

In this subsection we examine of how key numerical parameters, specifically the maximum parcel aspect ratio λ_{\max} and the minimum parcel volume fraction \tilde{V}_{\min} , affect computational cost and accuracy. This analysis is used to refine the default values in [1] for the 3D model. Before proceeding with this section, we recommend reading Section 3.1 first, as this introduces the test case used here in detail.

We conducted two series of simulations at a fixed resolution of 64^3 . In the first series, λ_{\max} is varied from 3 to 5 while keeping $\tilde{V}_{\min} = 1/20$, the default value. In the second series, \tilde{V}_{\min} is varied from 1/40 to 1/10 while keeping $\lambda_{\max} = 4$, the default value. Results are summarised in Table E.7, listing the simulation time, the average computational cost per time step and the average number of parcels per grid box. There is a small decrease in cost when λ_{\max} is increased due to the fact that fewer parcels are generated (less splitting occurs). A much larger decrease in cost occurs when \tilde{V}_{\min} is increased. First, fewer parcels are generated but also fewer time steps are required. The latter occurs because using fewer parcels represents high straining (velocity gradients) less well. Hence, one would expect less accuracy (see below), but the greater efficiency means that this could be compensated by an increase in grid resolution. That is, what matters is the time to solution for a prescribed accuracy.

As found for the 2D model [1], the (relative) r.m.s. volume error, shown in Fig. E.25, does not vary significantly for different maximum parcel aspect ratios λ_{\max} . This is because the average number of parcels per cell is approximately equal (cf. Table E.7). The same is true for the minimum parcel volume fraction \tilde{V}_{\min} in Fig. E.26, apart from $\tilde{V}_{\min} = 1/10$, where about 6 parcels per cell are found.

Table E.7

The cost per time step and the average number of parcels per cell when either the maximum parcel aspect ratio λ_{\max} or the minimum parcel volume fraction \tilde{V}_{\min} is varied, keeping the other at its default value in Table 1. These results were generated for the rotating Rayleigh-Taylor flow test case using 64^3 grid cells. Each simulation ran with 64 cores.

	λ_{\max}			\tilde{V}_{\min}			
	3	4	5	1/10	1/20	1/30	1/40
Wall time (s)	1086	1124	1181	577	1124	1667	2627
No. of time steps	1337	1542	1743	1188	1542	1731	2284
Cost per time step (s)	0.81	0.73	0.68	0.49	0.73	0.96	1.15
Avg. no. parcels per cell	11.3	10.4	10.6	6.2	10.4	14.4	19.4

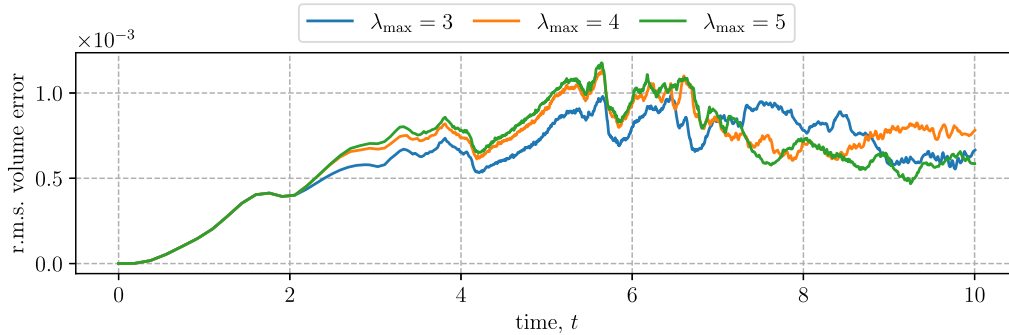


Fig. E.25. Gridded relative r.m.s. volume error for different maximum parcel aspect-ratios λ_{\max} , at a fixed grid resolution of 64^3 and minimum parcel volume fraction $\tilde{V}_{\min} = 1/20$.

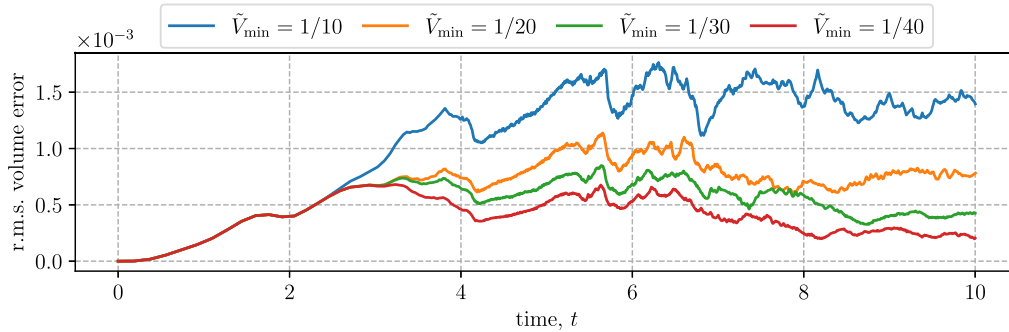


Fig. E.26. Gridded relative r.m.s. volume error for different minimum parcel volume fractions \tilde{V}_{\min} , at a fixed grid resolution of 64^3 and maximum parcel aspect-ratio $\lambda_{\max} = 4$.

We turn next to accuracy. In the upper and lower panels of Fig. E.28, we show the evolution of the r.m.s. error of buoyancy b and of the vertical vorticity component ζ , respectively. The r.m.s. errors are evaluated by coarsening the results of a 384^3 grid resolution simulation that we consider as the ground truth. A cross section of the gridded buoyancy, including zoomed square images of widths 192, 96, 48 and 24 grid cells, is illustrated in Fig. E.27. Superimposed on the gridded data, the figure shows ellipses obtained from the intersection between the $y = 0$ plane and the surrounding ellipsoids. At early times ($t = 4$), the r.m.s. errors in Fig. E.28 converge at a rate that is roughly proportional to $n_z^{-1.2}$. During the turbulent phase of the flow, the error in the vertical vorticity component ζ actually increases with grid resolution because vorticity rapidly intensifies at small scales, a known feature of buoyancy-driven flows. These small scales are systematically removed with increasing coarsening level. The coarsest grid resolution of 32^3 exhibits therefore the lowest r.m.s. error at late times. Since a higher value of \tilde{V}_{\min} implies larger small parcels, small scales are removed, thus generally resulting in lower r.m.s. error as well. The differences between the various \tilde{V}_{\min} thresholds are practically insignificant, yet a value of $\tilde{V}_{\min} = 1/10$ is discouraged as volume conservation (cf. Fig. E.26) is less well preserved compared to the others. The recommended choice that keeps the computational cost low but accuracy high is $\tilde{V}_{\min} = 1/20$. A maximum parcel aspect ratio of $\lambda_{\max} = 4$ also ensures a low cost-accuracy ratio.

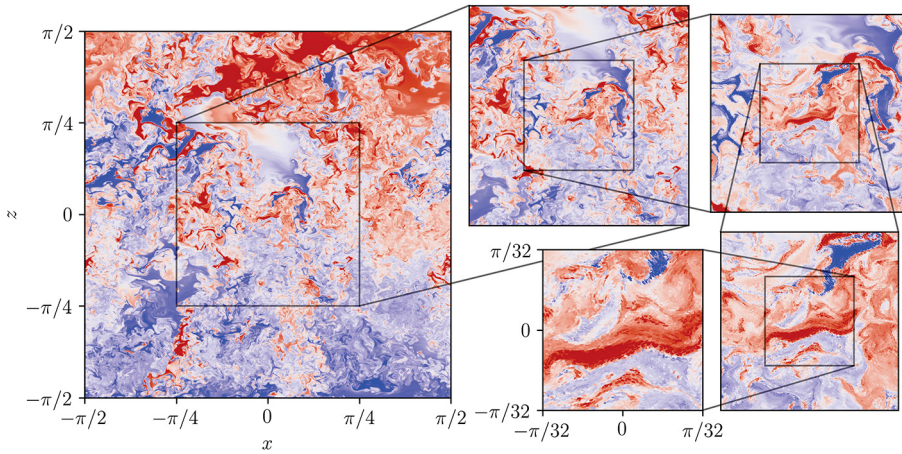


Fig. E.27. Cross section of the buoyancy field for the simulation with 384^3 grid cells at $y = 0$ in the xz -plane with intersected ellipses at $t = 6$. The zoomed windows correspond to widths of 192, 96, 48 and 24 grid cells in both directions.

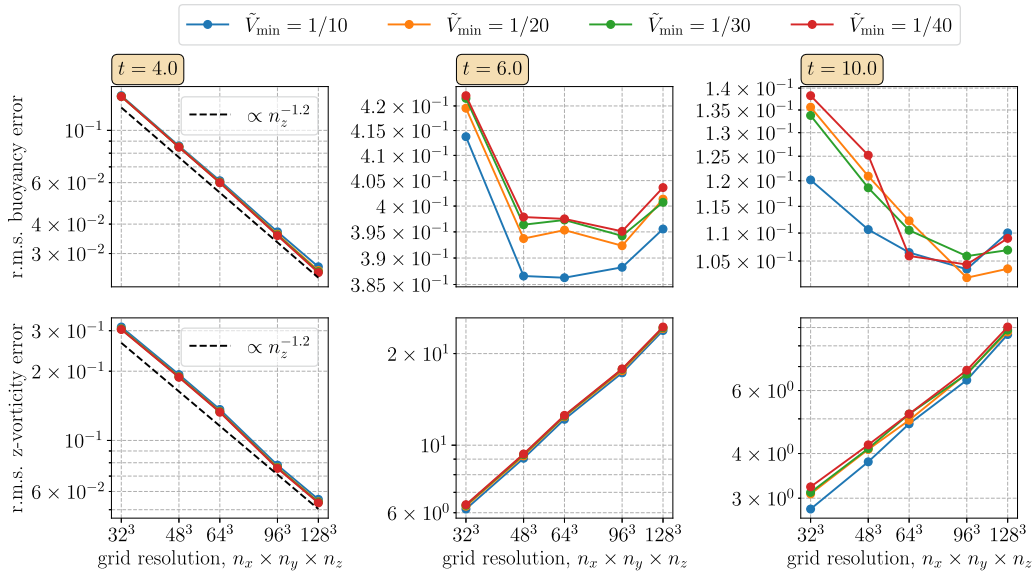


Fig. E.28. R.m.s. errors in buoyancy (top row) and in vertical vorticity (bottom row) versus resolution at three characteristic times (left to right). Curves are shown for different values of \tilde{V}_{\min} , as indicated. The errors are computed relative to a simulation run at 384^3 grid resolution and with $\tilde{V}_{\min} = 1/40$, coarsened to compare with the lower resolutions.

Appendix F. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.jcp.2023.100136>.

References

- [1] M. Frey, D. Dritschel, S. Böing, EPIC: the Elliptical Parcel-In-Cell method, *J. Comput. Phys. X* 14 (2022) 100109, <https://doi.org/10.1016/j.jcp.2022.100109>.
- [2] D.G. Dritschel, S.J. Böing, D.J. Parker, A.M. Blyth, The moist parcel-in-cell method for modelling moist convection, *Q. J. R. Meteorol. Soc.* 144 (715) (2018) 1695–1718, <https://doi.org/10.1002/qj.3319>.
- [3] S.J. Böing, D.G. Dritschel, D.J. Parker, A.M. Blyth, Comparison of the Moist Parcel-in-Cell (MPIC) model with large-eddy simulation for an idealized cloud, *Q. J. R. Meteorol. Soc.* 145 (722) (2019) 1865–1881, <https://doi.org/10.1002/qj.3532>.
- [4] D. Dritschel, M. Frey, The stability of inviscid Beltrami flow between parallel free-slip impermeable boundaries, *J. Fluid Mech.* 954 (2023) A31, <https://doi.org/10.1017/jfm.2022.1007>.
- [5] S. Terrington, K. Hourigan, M. Thompson, The generation and diffusion of vorticity in three-dimensional flows: Lyman’s flux, *J. Fluid Mech.* 915 (2021) A106, <https://doi.org/10.1017/jfm.2021.179>.
- [6] S. Terrington, K. Hourigan, M. Thompson, Vorticity generation and conservation on generalised interfaces in three-dimensional flows, *J. Fluid Mech.* 936 (2022) A44, <https://doi.org/10.1017/jfm.2022.91>.

- [7] D.G. Dritschel, J.N. Reinaud, W.J. McKiver, The quasi-geostrophic ellipsoidal vortex model, *J. Fluid Mech.* 505 (2004) 201–223, <https://doi.org/10.1017/S0022112004008377>.
- [8] T. Yamaguchi, D.A. Randall, M.F. Khairoutdinov, Cloud modeling tests of the ULTIMATE–MACHO scalar advection scheme, *Mon. Weather Rev.* 139 (10) (2011) 3248–3264, <https://doi.org/10.1175/MWR-D-10-05044.1>.
- [9] G. Cottet, P. Koumoutsakos, *Vortex Methods: Theory and Practice*, Cambridge University Press, Cambridge, 2000, <https://doi.org/10.1017/CBO9780511526442>.
- [10] T.Y. Hou, R. Li, Computing nearly singular solutions using pseudo-spectral methods, *J. Comp. Physiol.* 226 (1) (2007) 379–397, <https://doi.org/10.1016/j.jcp.2007.04.014>.
- [11] G. Cardano, *Ars Magna*, 1545.
- [12] L. Guilbeau, The history of the solution of the cubic equation, *Math. News Lett.* 5 (4) (1930) 8–12, <https://doi.org/10.2307/3027812>.
- [13] H. Rutishauser, The Jacobi method for real symmetric matrices, *Numer. Math.* 9 (1) (1966) 1–10, <https://doi.org/10.1007/BF02165223>.
- [14] J. Kopp, Efficient numerical diagonalization of Hermitian 3×3 matrices, *Int. J. Mod. Phys. C* 19 (3) (2008) 523–548, <https://doi.org/10.1142/S0129183108012303>.
- [15] W.M. Scherzinger, C.R. Dohrmann, A robust algorithm for finding the eigenvalues and eigenvectors of 3×3 symmetric matrices, *Comput. Methods Appl. Mech. Eng.* 197 (45) (2008) 4007–4015, <https://doi.org/10.1016/j.cma.2008.03.031>.
- [16] M. Frey, D. Dritschel, matt-frey/ps3d, ps3d version 0.0.13, <https://doi.org/10.5281/zenodo.8338215>, Sep 2023.
- [17] K. Winters, P. Lombard, J. Riley, E. D'Asaro, Available potential energy and mixing in density-stratified fluids, *J. Fluid Mech.* 289 (1995) 115–128, <https://doi.org/10.1017/S002211209500125X>.
- [18] D. Kang, O. Fringer, On the calculation of available potential energy in internal wave fields, *J. Phys. Oceanogr.* 40 (2010) 2539–2545, <https://doi.org/10.1175/2010JPO4497.1>.
- [19] M.J. Molemaker, J.C. McWilliams, Local balance and cross-scale flux of available potential energy, *J. Fluid Mech.* 645 (2010) 295–314, <https://doi.org/10.1017/S0022112009992643>.
- [20] D. Holliday, M.E. McIntyre, On potential energy density in an incompressible, stratified flow, *J. Fluid Mech.* 107 (1981) 221–225, <https://doi.org/10.1017/S0022112081001742>.
- [21] Rayleigh, Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density, *Proc. Lond. Math. Soc.* s1–14 (1) (1882) 170–177, <https://doi.org/10.1112/plms/s1-14.1.170>.
- [22] G.I. Taylor, The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. I, *Proc. R. Soc. Lond. Ser. A, Math. Phys. Sci.* 201 (1065) (1950) 192–196, <https://doi.org/10.1098/rspa.1950.0052>.
- [23] M.M. Scase, K.A. Baldwin, R.J.A. Hill, Rotating Rayleigh–Taylor instability, *Phys. Rev. Fluids* 2 (2017) 024801, <https://doi.org/10.1103/PhysRevFluids.2.024801>.
- [24] N. Brown, M. Weiland, A. Hill, B. Shipway, C. Maynard, T. Allen, M. Rezny, A highly scalable Met Office NERC Cloud model, in: *Proceedings of the 3rd International Conference on Exascale Applications and Software, EASC '15, University of Edinburgh, GBR, 2015*, pp. 132–137.
- [25] B. Duffy, H. Carr, T. Möller, Integrating isosurface statistics and histograms, *IEEE Trans. on Vis. and Comput. Graph.* 19 (2) (2013) 263–277, <https://doi.org/10.1109/TVCG.2012.118>.
- [26] M. Frey, S. Böing, D. Dritschel, EPIC-model/epic: EPIC version 0.13.1, <https://doi.org/10.5281/zenodo.8348474>, Sep 2023.
- [27] S. Böing, M. Frey, MONC and PMPIC comparison code for EPIC, <https://doi.org/10.5281/zenodo.10156409>, Nov 2023.
- [28] D. Hassell, J. Gregory, J. Blower, B.N. Lawrence, K.E. Taylor, A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1), *Geosci. Model Dev.* 10 (12) (2017) 4619–4646, <https://doi.org/10.5194/gmd-10-4619-2017>.
- [29] B. Eaton, J. Gregory, B. Drach, K. Taylor, S. Hankin, J. Blower, J. Caron, R. Signell, P. Bentley, G. Rappa, H. Höck, A. Pamment, M. Juckes, M. Raspaud, R. Horne, T. Whiteaker, D. Blodgett, C. Zender, D. Lee, NetCDF Climate and Forecast (CF) metadata conventions v1.8, <http://cfconventions.org/Data/cf-conventions/cf-conventions-1.8/cf-conventions.html>, Feb. 2020.
- [30] H. Samuel, A deformable particle-in-cell method for advective transport in geodynamic modelling, *Geophys. J. Int.* 214 (3) (2018) 1744–1773, <https://doi.org/10.1093/gji/ggy231>.
- [31] H. Cheng, K.C. Gupta, An historical note on finite rotations, *J. Appl. Mech.* 56 (1) (1989) 139–145, <https://doi.org/10.1115/1.3176034>.
- [32] J.S. Dai, Euler–Rodrigues formula variations, quaternion conjugation and intrinsic connections, *Mech. Mach. Theory* 92 (2015) 144–152, <https://doi.org/10.1016/j.mechmachtheory.2015.03.004>.