

Tracking and analysis of interfaces and flow structures in multiphase flows

A. Bußmann^a, J. Buchmeier^b, M.S. Dodd^c, S. Adami^{a,d}, I. Bermejo-Moreno^{b,*}^a Institute of Aerodynamics and Fluid Mechanics, School of Engineering and Design, Technical University of Munich, 85747 Garching, Germany^b Department of Aerospace and Mechanical Engineering, University of Southern California, Los Angeles, CA 90089, USA^c Center for Turbulent Research, Stanford University, Stanford, CA 94305-3024, USA^d Munich Institute of Integrated Materials, Energy and Process Engineering (MEP), Technical University of Munich, 85747 Garching, Germany

ARTICLE INFO

Keywords:

Tracking

Multiphase

Droplet breakup

Homogeneous isotropic turbulence

ABSTRACT

A methodology is introduced to study the dynamics of fluid interfaces in multiphase flows, emphasizing their break-up and coalescence. The algorithm tracks surfaces, here obtained by isocontouring an interface-describing scalar field (e.g., VOF) from a time series of volumetric snapshots. Physical and geometric information of the surfaces is used to find correspondences in a higher-dimensional space. Events are derived from found correspondences to describe the interactions among isosurfaces of closed fluid structures extracted at consecutive tracking time steps. The correspondences and events are filtered based on physical realizability, accounting for geometric constraints between consecutive time instances, as well as temporal constraints on the relations between surfaces in previous tracking steps. The resulting events are used to map the time evolution of all surfaces and their interactions into a graph, which is then queried to retrieve information on the dynamics of the fluid interfaces. The methodology is applied to a DNS dataset of droplet break-up in forced homogeneous isotropic turbulence (HIT). Emphasis is placed on the statistics of split and merge events, the lifetime of surfaces, and their geometric evolution in relation to the background flow fields.

1. Introduction

The understanding of the flow feature dynamics in physical space, including multiphase flow structures, can greatly benefit from tracking algorithms that automatically correlate the flow structures as they evolve in time, based on physical quantities [1]. Tracking results enable a quantitative assessment of the interactions among flow features extracted from a single or multiple physical fields. This is especially relevant in turbulent multiphase flows, for which the interface between two fluids and the turbulent flow structures (e.g., vortex tubes or shear-dominated sheets) move, deform, break up into smaller fragments, and coalesce with nearby features over time by the action of the underlying turbulence.

Explicit tracking methods solve the so-called *correspondence problem* to match objects (e.g., structures extracted from a flow field), between two consecutive time instances [2]. In implicit extraction methods, no explicit calculations are carried out to find correspondences, which instead are obtained implicitly by the applied method of extraction [3]. Both explicit and implicit methods combine found correspondences into events that represent structure interactions between consecutive time instances. From the time series of events, information about the dynamics of the structures can be retrieved, such as eddy growth over

time or the break-up process of a bubble influenced by its surrounding flow field [2].

Implicit methods can be divided into *spatio-temporal* and *predictive methods*. Spatio-temporal approaches use a higher-dimensional space that combines the three spatial dimensions and time to extract features and find correspondences. Interactions can be found, for instance, using isosurfaces in this space [4,5]. An additional scale-space was introduced by Bauer and Peikert [6] to track vortices and their dynamics in time. Another approach uses integrated streamlines of a higher-dimensional vector field to provide feature tracking [7]. Predictive methods, on the other hand, incorporate information of already found correspondences to estimate future interactions. Muelder and Ma [3] use information of previous time steps to predict the future position of structures by linear and quadratic functions. Then, corresponding objects are extracted at this estimated position. Sauer et al. [8] use a similar approach when particle information is present in the flow field. At each tracking time step, particles are assigned to an extracted object and correspondences are implicitly found by following the particles in time.

In contrast to implicit methods, correspondence-based approaches require the *extraction* of all structures before the correspondence problem can be solved. In multiphase datasets, where different fluids are present at the same time, an interface-describing scalar quantity, such

* Corresponding author.

E-mail addresses: alexander.bussmann@tum.de (A. Bußmann), bermejom@usc.edu (I. Bermejo-Moreno).

as the level set or the volume fraction, allows an immediate extraction of structures. In turbulent flows, structural analyses have focused on identifying regions of high turbulence intensity, dissipation, and enstrophy, placing emphasis on their geometric characterization as tube- or sheet-like structures [9,10]. Such regions are classified by local, pointwise measurements of the velocity gradient tensor [11]. This can be used to identify stretched vortices [12], vortex tubes (vorticity-dominant) [13], curved vortex sheets (strain-dominant) or flat vortex sheets (high vorticity and strain) [14,15]. To overcome the locality of such criteria, points with a similar identity can be grouped, using e.g. box-counting methods on sets of points [16], vortex clusters [17] or quadrant events [18]. Here, we use the non-local geometrical analysis introduced by Bermejo-Moreno and Pullin [9], where joint probability distributions of differential geometric attributes mapped on extracted isosurfaces are used to characterize the structure geometry non-locally. This method has been applied to turbulent flow datasets [19–22]. Leung et al. [10] used Minkowski functionals from integral geometry [23] to investigate non-local geometries and the interaction of structures in homogeneous isotropic turbulence (HIT).

Physics-based approaches use detailed data from the flow field to track structures in time. Clyne et al. [24] define the point of minimum dissipation on extracted structures and use streamlines of this point to find interactions. Elsinga and Marusic [25] describe the evolution of flow topologies by evolving second- and third-order invariants of the velocity gradient tensor in a Lagrangian frame of reference. Oster et al. [26] track the flame front in combustion processes following micro-patches by an advection equation. Since not all flow features allow such physics-based tracking, *regional-based methods* use only regional information of the extracted objects to find correspondences. A variety of methods exist, such as affine transformation matrices [27], spiral search for tube skeletons [28], or a radius search for hairpin vortices [29]. Regional overlap of features, introduced by Silver and Wang [1,30], has been applied successfully to understand the evolution of coherent structures in turbulent channel flows [31,32]. To overcome the problem of requiring spatial overlap and allow larger tracking steps, *attribute-based approaches* track structures based on their different attributes and represent a higher level of abstraction [33]. Possible attributes are, for example, the centroid position, the volume/mass, or the circulation of an extracted objects, as introduced by Samtaney et al. [34]. To find the correct correspondence under all possible interactions, Reinders et al. [35] sum all attribute criteria into a weighted correspondence function that finds matches based on user-defined tolerances and a confidence index. Recently, Chan et al. [36,37,38] used attribute-based tracking to investigate bubble breakup and coalescence in breaking waves, using as attributes the bubble centroid position and volume, and detecting events based on volume conservation.

Methodologies like the particle tracking algorithm of Sauer et al. [8] or the feature field approach of Theisel and Seidel [7] have limited applicability, since they require special data to be available from the simulation. A general approach allows to be used on a large number of datasets and it need not be restricted to fluid-mechanical applications. Overlapping approaches, used by Silver and Wang [30] or Lozano-Durán and Jiménez [32], and spatio-temporal methodologies, require a relatively small tracking time step, and pose difficulties to track small, fast moving objects. Hence, they are not robust against time step variations. The attribute-based approaches of Samtaney et al. [34] and Reinders et al. [35] introduce user-defined tolerances that enable highly-flexible and interactive tracking, applicable to different scenarios. Depending on the complexity of the involved attributes, the higher level of abstraction in attribute-based approaches has advantages compared to the more physics-based approaches of [24,26], where the full flow field must be present at each tracking instance.

The present approach combines the advantages of attribute- and regional-based tracking. Spatial and non-local geometric attributes of volumetric extracted structures are used to find correspondences between two consecutive tracking time steps [39]. Those interactions are

then constrained on physical realizability using global spatial and physical attributes, along with more detailed regional information of the structures, without requiring spatial overlap as in solely regional-based tracking approaches (e.g., Lozano-Durán and Jiménez [32] or Silver and Wang [30]). Similar to Reinders et al. [35], robustness is ensured by user-defined tolerances on the changes of attributes. All found correspondences are then formed into events. The final set of events is used to map the time evolution of all surfaces and their interactions into a directed graph, which is then queried to retrieve information on the dynamics of the fluid interfaces.

In contrast to Reinders et al. [35] and other mentioned methods, correspondence determination and event detection is not combined into a single step in the present work, which provides a more robust tracking approach and allows the formation of compound events. The explicit nature of the proposed tracking algorithm requires a decoupling strategy to decompose such complex, compound events. Here, compound event refer to a set of structures that are temporarily connected to each other. In the literature, compound events mostly refer to spatially connected structures in a single time instance identified by so-called labeling algorithms [40–43]. Guo et al. [44] provide a tracking technique of vortex lines in superconductors that also allows the inherent detection of compound events. However, they use operations tailored to that specific type of simulations and assume a certain mesh quality, such that no decomposition of those complex events is needed.

This paper is organized as follows: in Section 2 the developed tracking methodology is presented, emphasizing extraction, correspondence determination, event detection and graph mapping. In Section 3, the numerical setup of the problem of droplet breakup in HIT is shown together with the approach to determine a suitable set of tracking parameters. Section 4 describes how the simulation and tracking results are validated. In Section 5, results obtained from the application of the tracking algorithm for simulations of droplet breakup are presented, highlighting the geometric changes of structures during the breakup process.

2. Methodology of the tracking algorithm

Owing to the explicit nature of the proposed tracking algorithm, it requires a preceding extraction step (see Fig. 1). During this step, structures are extracted from the three-dimensional simulation data and characterized geometrically (see Section 2.1). This geometrical information is then used in the tracking methodology to find correspondences between structures of each pair of consecutive time instances.

During the whole tracking period $t \in [t_{\text{start}}, t_{\text{end}}]$, where t is the time, t_{start} the start and $t_{\text{end}} = t_{\text{start}} + \sum_{i=0}^{N_t} (\Delta t)_i$ the end time, in total, N_t tracking steps with a variable time interval of $(\Delta t)_i$ are carried out. A tracking step is defined by its initial t^n and consecutive time instance t^{n+1} , where $n \in [0, N_t - 1]$. In each tracking step, interactions are found between structures from the previous instance n (source structures or sources) and the consecutive time $n + 1$ (target structures or targets). Correlations between structures from the same time step are not considered.

The actual tracking algorithm consists of four main stages: (i) structure pre-processing, (ii) correspondence determination, (iii) event detection, and (iv) graph mapping. In the first stage, structures are prepared for the tracking algorithm, e.g., to allow tracking under periodic boundary conditions (Section 2.4) or to use certain constraints during the correspondence search (Section 2.2.2). Afterwards, correspondences between structures are found (Section 2.2), which give one-to-one interactions between structures at consecutive tracking time steps. These correspondences are then filtered based on constraints accounting for physical realizability. During the process of event detection (Section 2.3), the remaining correspondences form events which describe the interaction of a group of correspondences. These events are categorized as continuation events (one structure evolves as a single structure), splits (break-up of a structure into several smaller ones) or

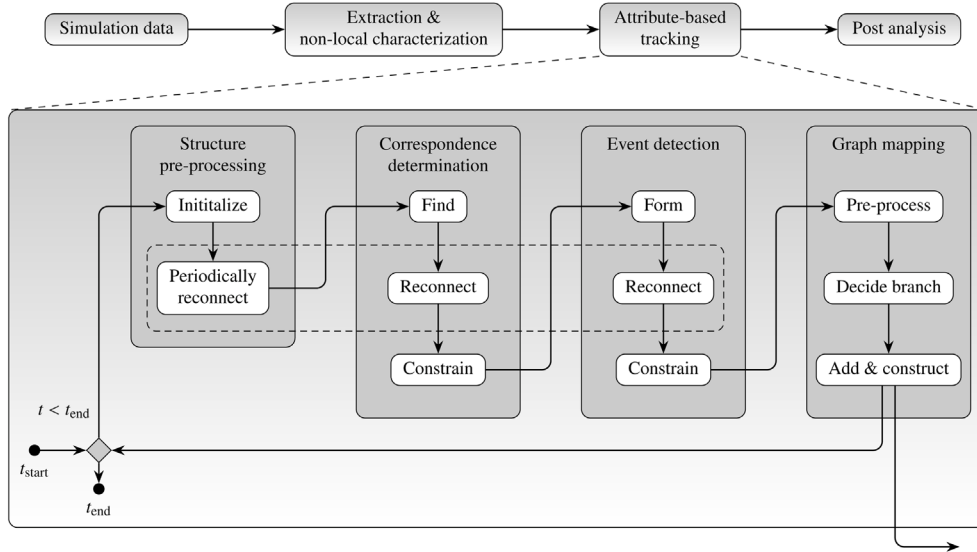


Fig. 1. Schematic representation of the subsequent steps from the simulation data towards the post analysis of the structural dynamics (top) and detailed steps carried out during the attribute-based tracking algorithm (bottom). The tracking block consists of four parts: (i) structure pre-processing, (ii) correspondence determination, (iii) event detection, and (iv) graph mapping. In each of the first three blocks periodic reconnection can optionally be applied (dashed block).

merges (coalescence of smaller structures into a single, larger one). As previously done with found correspondences, newly formed events are filtered based on physically realizable constraints (e.g., volume conservation). These event constraints can consider not only the geometry and physics of the structures involved in the event between consecutive time instances, but also temporal relations among structures in previous tracking steps. The final stage of each tracking step is to map the events into a dynamically-generated graph (Section 2.5), which involves pre-processing of the found events, assigning an adequate branch for each structure and appending the new events to the existing graph. The graph is then queried to retrieve information on the dynamics of the involved structures

2.1. Extraction of structures and non-local geometric characterization

Structures are extracted from the flow field by isocontouring a selected scalar field [45]. The non-local characterization of each structure is done using two differential geometry parameters, the absolute shape index, S , and the dimensionless curvedness, C , [46], plus a third dimensionless quantity, the compactness parameter, λ , also referred to as sphericity and stretching. The shape index and curvedness are point-wise quantities obtained on the structure surface as a function of the principal curvatures (κ_1, κ_2)

$$S = \left| -\frac{2}{\pi} \arctan \left(\frac{\kappa_1 + \kappa_2}{\kappa_1 - \kappa_2} \right) \right|, \quad C = \frac{3V}{A} \sqrt{\frac{\kappa_1^2 + \kappa_2^2}{2}}. \quad (1)$$

V and A are the volume enclosed by the structure and its surface area, respectively [47]. The shape index provides dimensionless information about the geometrical type of point (saddle-like, ridge- or cap-/cup-like), whereas the dimensional curvedness is the inverse of a representative local radius of curvature obtained from the principal radii of curvature at that point. In addition, the compactness parameter, indicative of the global compactness of the structure, is defined as

$$\lambda = \sqrt[3]{\frac{36\pi V^{2/3}}{A}} \quad (2)$$

and equals unity for spheres. For all other structures it is in the range $\lambda \in [0, 1]$, meaning that lower values correspond to more stretched structures, occupying more area for the same volume. The compactness parameter is a global characteristic of the structure, whereas the curvedness and shape index are local, point-wise geometric quantities

of the surface. To obtain a non-local characterization, the area-based joint probability density function (jpdf) of the dimensionless curvedness and the absolute shape index is calculated. Following [9] a feature center ($\hat{S}, \hat{C}$) is obtained from first- and second-order moments of the jpdf. The feature center better accounts for skewed/asymmetric jpdfs, commonly found for smoothly corrugated surfaces in turbulent flow structures. A feature space of geometric parameters ($\hat{S}, \hat{C}, \lambda$), invariant against translations, rotations and scaling of the reference system, is suitable to compare the structures based on their geometry [9].

With the three scalar parameters of the feature space, structures can be classified into different groups. Blob- or sphere-like structures are found in the region around (1, 1, 1). Elongated cylinders of nearly circular cross-section have a shape index of $S = 1/2$ and a curvedness of $C = 3/2\sqrt{2} \approx 1.06$. Hence, tube-like structures, predominantly consisting of such cylindrical shapes, occupy the region around the axis (1/2, 1, λ), becoming increasingly stretched as the compactness parameter decreases. The transition towards sheet-like structures occurs as the compactness and curvedness further decrease. Planar structures correspond to the plane of zero curvedness.

2.2. Correspondence determination

After the structure pre-processing, the next stage of the tracking methodology finds interactions between structures extracted at two consecutive time instances. Correspondences are first found between the two instances and, then, constrained based on physical realizability. The clear separation of search and constraining introduces more flexibility and better control to find such correspondences, compared to best match algorithms, e.g. in Reinders et al. [35] or Chan et al. [36], where both steps are combined into a single one. Instead of computing a single score for all found correspondences and applying a best-match algorithm, we filter physical unrealizable interactions immediately.

We distinguish between candidate, filtered, and actual correspondences. The correspondence search gives a set of possible candidate correspondences, that are then filtered by constraints, and finally grouped into events of actual correspondences. To quantify the reliability of a structure-structure interaction, a confidence index $c \in [0, 1]$ is introduced, which is comparable to the correspondence factor introduced by Reinders et al. [33,35]. Higher values of c indicate higher reliability. In each stage of the correspondence determination (search and constraining) a confidence value c_j is computed. The overall confidence

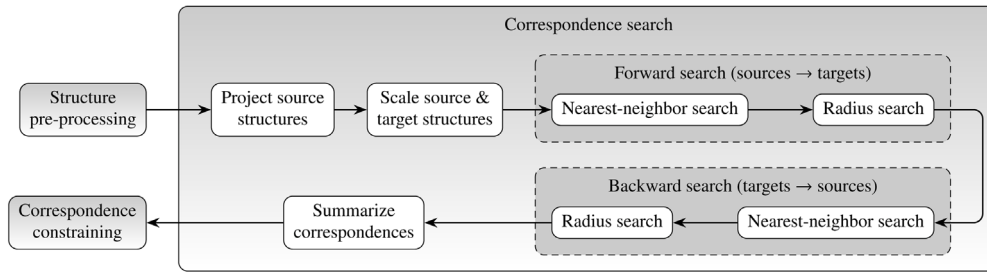


Fig. 2. Schematic representation of the different steps of the correspondence search. The solid arrows represent the data flow between the steps highlighted with white background color. The nearest-neighbor and radius search are executed in forward and backward directions.

of a correspondence, c_i^* , is calculated as the weighted sum of all its N_c computed confidence values:

$$c_i^* = \sum_{j=0}^{N_c} \omega_{j,c} c_{i,j} \quad \text{with} \quad \sum_{j=0}^{N_c} \omega_{j,c} = 1, \quad (3)$$

where $c_{i,j} \in [0, 1]$ is the individual confidence of the different correspondence searches and constraints. The weight $\omega_{j,c}$ for each confidence index can be chosen independently based on the priority given to each contributor.

2.2.1. Correspondence search

A nearest-neighbor and radius search is used to find all candidate correspondences. The basic assumption of the nearest-neighbor search is that continuation events do not alter strongly the structure attributes, independently from the search parameters. Therefore, the search is performed in a six-dimensional parameter space consisting of the three spatial coordinates and the geometrical feature space of the structure. On the contrary, in split and merge events, where structures break up or coalesce, the attributes of a structure vary more significantly. Therefore, the radius search captures such correspondences by finding all the spatially proximate neighbors surrounding the reference structure (source for split and target for merge events, respectively). This approach is similar to the search window used by Elsinga et al. [29], Silver and Wang [30], to account for fast-moving objects in the flow field, or the spiraling search around vorticity tubes used by Vilasenor and Vincent [28]. To account equivalently for split and merge events, a two-way search is carried out, forward and backward (Fig. 2). Being attribute-based, both the nearest-neighbor and radius searches guarantee that no spatial overlap is required, compared to regional-based approaches [1,30,32]. To increase the accuracy of the searches, a projection step and a scaling operation can be used (Fig. 2). As a final step, doubly-found correspondences from the two-way search are consolidated by summing their confidence indices with equal weights to reduce redundant operations in subsequent tracking stages.

Nearest-neighbor search

The nearest-neighbor search (nn) finds, for every structure, its closest neighbor in a space of dimensions N_{nn} . Here, the nearest-neighbor space is six-dimensional ($N_{nn} = 6$), consisting of the Cartesian coordinates of the structure x , y and z , and the three geometric parameters $\hat{S}$, $\hat{C}$, and λ . The spatial position of a structure is determined by the mean coordinates from its axis-aligned bounding box (AABB), $\bar{x} = (x_{\max} + x_{\min})/2$, where $x_{\min}$ and $x_{\max}$ are the minimum and maximum box points accounting for the extent of the structure in physical domain. Other types of bounding boxes, e.g., oriented boxes (OBB), could be adopted for the search [39].

Each coordinate in the 6D space is normalized onto the range $[0, 1]$ for the search. At each tracking step, a linear projection of the point in the 6D space corresponding to each source structure is conducted with its average surface velocity $\bar{v}$ to estimate the position in the target frame, $\bar{x}_{\text{proj}}$. The projection is used to increase the accuracy of the nearest-neighbor search [29,32]. Higher-order projections could be

adopted to involve multiple preceding tracking time steps [3], requiring special treatment for newly created structures that do not exist in previous time steps. Local projection of each point of the discrete structure in physical domain with its velocity could be used instead, followed by the geometric characterization and mapping of the projected structure onto the 6D space before the search. A fixed confidence value is assigned to each correspondence found by the nearest-neighbor search, $c_{i,nn} = c_{nn} \in [0, 1]$.

Radius search

The nearest-neighbor search, which finds only one neighbor for each structure, is not sufficient to identify break-up or coalescence events by which structures split into smaller ones or merge into a larger element, leading to more involved geometrical deformations. An additional radius search is conducted to identify structures that are in close spatial proximity and could be involved in such events. The search radius must be large enough to capture all involved structures but small enough to reduce the number of false positive correspondences. Possible search radius choices are a fixed value that depends on physical processes, e.g. boundary layers [29], or the sampling frequency [30]. Here, the search radius is defined as the longest Cartesian dimension of the structure's AABB

$$r = \frac{1}{2} \max(\{x_{i,\max} - x_{i,\min}, i = 1, 3\}) \varepsilon_r, \quad (4)$$

This choice of search radius accounts for the different sizes of the involved structures. The tolerance ε_r restricts the search and can be made application-dependent to account, for example, for fast-moving structures.

The confidence of the radius search is computed using the relation of the found spatial distance, $d_{i,r}$, and the radius, r_i , of the structure around which the search is applied

$$c_{i,r} = 1 - \frac{d_{i,r}}{r_i} \in [0, 1]. \quad (5)$$

The confidence value increases for nearby structures. Since the distance is at most equal to the radius, the confidence is bounded by $[0, 1]$.

2.2.2. Correspondence constraints

After the correspondence search, all candidate interactions have been found, but not all are physically realizable. In the next step, unrealizable interactions are rejected based on constraints that reflect consistency during the evolution of a structure, i.e. the change of attributes of a structure is limited from one tracking time step to the next. Each correspondence constraint is described by its constraint function f fulfilling

$$f(p_A, p_B) \leq T_f, \quad (6)$$

where a certain set of parameters p of two structures A and B is related and compared to a specified tolerance T_f . Correspondences that exceed the tolerance are rejected and not combined into a single correspondence factor as in Reinders et al. [35], because the correspondence

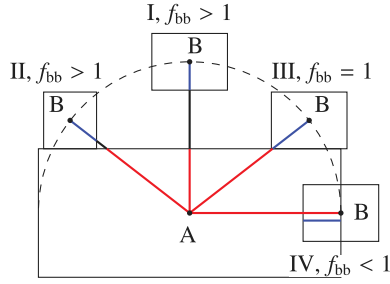


Fig. 3. Representation of different scenarios and values for the bounding-box constraint function f_{bb} between two structures A and B. The lines connecting the filled dots represent the distance between the mean positions of each structure pair, $d_{A,B}$. The colored segments represent the size S_i of each structure. The dashed circle indicates that the AABBs of all B structures are placed equidistant to structure A.

search is already finished. The confidence value for each constraint function is computed by [35]

$$c_{i,f} = 1 - \frac{f(p_A, p_B)}{T_f} \in [0, 1]. \quad (7)$$

In this work, two spatial constraints are applied on the correspondences: a bounding-box constraint and a local proximity constraint.

Bounding-box constraint

In the radius search, the largest dimension of the structure AABB is used as the search radius. This may lead to false correspondences for highly-stretched structures, such as tubes or sheets. For example, two elongated tubes with parallel cylindrical shape that are far apart a distance smaller than the tube length, can still be identified as a correspondence by the radius search. Additionally, the nearest-neighbor search can also result in correspondences found between structures that are spatially far apart, because spatial coordinates are not generally the only parameters used in that search. The bounding-box constraint partially eliminates these false positives by rejecting correspondences found for any two structures, A and B, for which the distance $d_{A,B}$ between their mean positions is too large compared to the sum of their sizes, S_A and S_B

$$f_{bb} = \frac{d_{A,B}}{S_A + S_B} \leq T_{bb}. \quad (8)$$

The spatial distance between two structures is defined as the Euclidean norm between their mean positions $d_{A,B} = \|\bar{x}_B - \bar{x}_A\|$. The size of a structure S_i is referred to the fraction of the total distance $d_{A,B}$ lying inside each bounding box. Hence, the bounding-box constraint measures how much longer $d_{A,B}$ is compared with the fraction contained inside the AABBs of A and B, which can be expressed as

$$f_{bb} = \frac{d_{A,B}}{S_A + S_B} = \frac{d_{A,B}}{d_{A,B,inside}} = 1 + \frac{d_{A,B,outside}}{d_{A,B,inside}}. \quad (9)$$

In Fig. 3 different correspondence scenarios between two structures A and B are represented, leading to different values of the constraint function f_{bb} for the same distance $d_{A,B}$ in between them. Far distant structures (I) always result in function values $f_{bb} > 1$. If the AABBs overlap (IV) then $f_{bb} < 1$. When the structures only overlap slightly or touch each other, the structure size depends on the actual size of the bounding boxes (II and III). In both shown cases, the AABBs touch each other, but only in III the bounding-box constraint function takes the expected value of $f_{bb} = 1$. In II the function value is larger than one, because the intersections of their bounding boxes with the distance vector lie on different planes. Since the AABB is only a rough approximation of the structure, contact between bounding boxes does not necessarily mean that the structures touch each other. Hence, the given approximation is a good first estimation of their distances and can be used to eliminate far distant structures.

Local proximity constraint

Depending on the chosen tolerance T_f , the bounding-box constraint rejects only truly separated structures. In Fig. 4 different arrangements are illustrated to indicate the importance of a second constraint that takes into account the local (i.e., pointwise) proximity of two structures. If structures are enclosed by each other (Fig. 4(a)), the bounding-box constraint cannot distinguish true overlap and inclusion. The same applies for small overlaps, where both structures are clustered on the opposite sides of their AABB (Fig. 4(b)). For truly overlapping (Fig. 4(c)) or separated, but proximal, structures (Fig. 4(d)), the bounding-box constraint gives accurate results, which are then further confirmed by the local proximity analysis.

The local-proximity constraint function is calculated as the minimum distance $d_{\min,A,B}$ between structures A and B, divided by the distance traveled in between the pair of tracking time steps at the maximum surface-averaged velocity $\bar{v}_i$ of the two structures

$$f_{lc} = \frac{d_{\min,A,B}}{\max(\bar{v}_A, \bar{v}_B)\Delta t}, \quad \text{where} \quad \bar{v}_i = \left(\frac{1}{3} \sum_i u_i^2 \right)^{1/2} \quad (10)$$

and u_i are the surface-averaged velocity components of the structure in each Cartesian direction. The distance $d_{\min,A,B}$ is the global minimum pointwise distance between points on the surface of structures A and B. Alternatively to the surface-averaged (global) velocity, a local velocity of (or averaged around) the points defined by the local distance of the two structures could be used to provide a local comparison of the distances.

The use of a structure velocity (local or global) poses challenges when the resulting distance in the denominator of Eq. (10) is small compared to the actual distance in the numerator. In such cases, the constraint function f_{lc} yields large values which would result in a rejection of those correspondences, even if the distance is small. Larger tolerances can be specified to avoid such rejections. Then, the weight of the local proximity constraint ω_{lc} should be lowered, avoiding the local constraint to dominate the overall confidence value of the identified correspondence.

As the number and size (i.e., number points) of structures increases, the computation of pointwise minimum distances becomes computationally expensive. A kd -tree specifically designed for fast computations in three dimensions is used [48]. Owing to the increased computational expense of the local proximity constraint, it is always advantageous to apply first the less accurate but more computationally efficient bounding-box constraint, to reject correspondences between structures that are clearly too distant, and thus reduce the number of correspondences to be considered by the local proximity constraint.

2.2.3. Criteria for confidence weight selection

The confidence of a correspondence, c_i^* , describes the reliability of the structure-structure interaction by a single numerical value that weighs the confidence values for the individual correspondence searches and constraints (see Eq. (3)). The weights, $\omega_{j,c}$, can be chosen independently based on the accuracy and physical information of the different searches and constraints used in the tracking for the dataset under consideration.

The radius search solely uses spatial information to find correspondences, whereas the nearest-neighbor search additionally relies on geometrical attributes. Furthermore, large structures have a large search radius (see Eq. (4)) which can result in a large number of found correspondences. Therefore, the weight of the nearest-neighbor search, ω_{nn} , should be prioritized over the weight of the radius search, ω_r . The bounding-box constraint only uses global geometrical information to analyze the correspondences. It is designed to reject truly separated structures. The local proximity constraint, on the other hand, uses more detailed surface information (pointwise distances) and can identify unrealizable correspondences more accurately. Hence, the weight of the local constraint, ω_{lc} , should be prioritized over the bounding-box constraint, ω_{bb} .

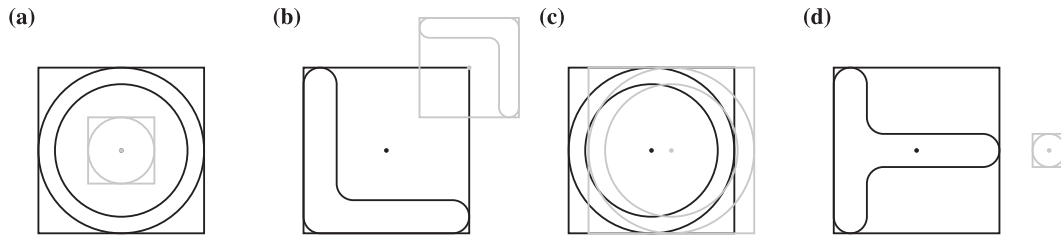


Fig. 4. Schematic representation of structure-structure arrangements emphasizing the use of the local proximity constraint. The interactions represent (a) enclosed structures, (b) structures lying on the opposite side of the overlapping bounding-boxes, (c) truly overlapping structures, and (d) separated structures.

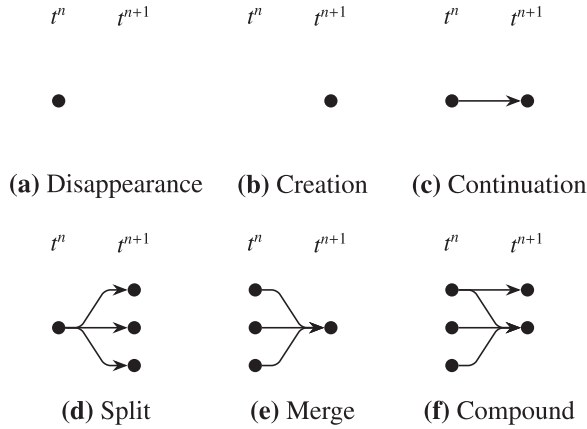


Fig. 5. Categorization of different events used in the tracking methodology. The black circles represent source and target structures at the previous t^n and consecutive time instance t^{n+1} , respectively. The black arrows indicate the directed interaction between sources and targets.

2.3. Event detection

2.3.1. Event formation

After the correspondence determination, the transition from single structure-structure interactions (correspondence) to a collection of interactions (event) takes place. The latter may involve multiple other source and target structures (split, merge and compound events), but also none (creation and disappearance). During the event formation process, each structure is assigned uniquely to one event. Each interaction between a source and target structure is represented by a directed edge, storing the confidence indices assigned during the correspondence determination. Here, events are first formed and afterwards filtered based on constraints, compared to other methods, where both steps are combined into a single one [30,32,35]. In general, six different types of events are distinguished in this work (Fig. 5). A *creation* represents the birth of a structure, where no correspondence can be found for a target structure. The counterpart is a *disappearance*, where no forward connection can be found for a source structure. *Continuation* events connect one source and one target structure with each other and represent the physical translation of a single structure including rotations and deformations without interactions with other structures. *Splits* describe the break up of a source structure into smaller sub-structures, whereas *merge* events indicate a fusion/coalescence of smaller structures into a larger one. *Compound* events are a combination of simultaneously occurring continuations, splits and merges. Physically, a compound event represents the partial separation of a large structure into smaller ones, where one or more of the broken structures merge with others at the same time.

The presence of compound events sets this methodology apart from others, where the detection of such events was not considered or not possible [35]. This further provides more robustness against variations of the tracking time step, since larger time steps increase the possibility

of such compound events. In some applications, the presence of such events is even physically motivated. For instance, consider two miscible (source) structures initially separated and with different chemical composition that evolve in time to first coalesce and then break up into two other (target) structures, each containing a mixture of the original compositions. If the tracking time step is long enough to skip the initial coalescence, this scenario will lead to the occurrence of a compound event: the two source structures would be related to the two target structures by a single compound event. The mixture of chemical composition might be required in this case as a parameter in the search for correspondences.

2.3.2. Decomposition of compound events

During the application of event constraints, single events (e.g., split and merge) have no complex interactions since one side of the event always contains only one structure. Compound events reveal more complex behavior due to their cross connections between several sources and targets.

Guo et al. [44] provide a tracking technique that also allows the detection of compound events. In their approach, the underlying mesh of the simulation is used together with operations specifically designed for the extraction of vortex lines in superconductors. Furthermore, the mesh must be continuous in space and time, and fine enough to represent all physical phenomena correctly, so there is no requirement to decompose found compound events. In the present work, no assumption is made on the underlying mesh and captured physical phenomena. Hence, an algorithm is required to decouple false-positive compound events into simpler sub-events based on dataset-dependent attributes.

A compound event involves at least two sources and two targets, whereby at least one of the targets has more than one backward connection (i.e., from target to source). This allows a further categorization into single and multiple compound events, depending on whether one or more backward connections (edges) are assigned to any of its targets, respectively. Targets in a compound event can be classified into (i) regular targets, (ii) split targets, and (iii) merge targets (Fig. 6).

In regular targets, only a single backward connection to a source structure exists. Split targets are connected with at least two source structures, where each source has two outgoing edges. A merge target has multiple backward connections and one of the sources has only one forward edge, i.e., it solely merges into the target. Compound events with double connections are here denoted as non-linear, in contrast to linear events, which have no double connections.

During the application of event constraints, candidate compound events can be decomposed into simpler sub-events (continuations, split, merges) if the resulting sub-events are physical realizable, but creations and disappearances are not allowed to be formed. In Fig. 7 two compound events are shown involving a split target and a merge target respectively. Since at least one source structure that interacts with the merge target has only one forward connection, it cannot exist without that target, if disappearances are not allowed. Therefore, the decomposition algorithm splits such events into a single merge and remaining splits/continuations (Fig. 7(a)). On the contrary, split targets can be moved to all sub-events, since all sources have multiple forward connections (Fig. 7(b)). If multiple split and merge targets are involved

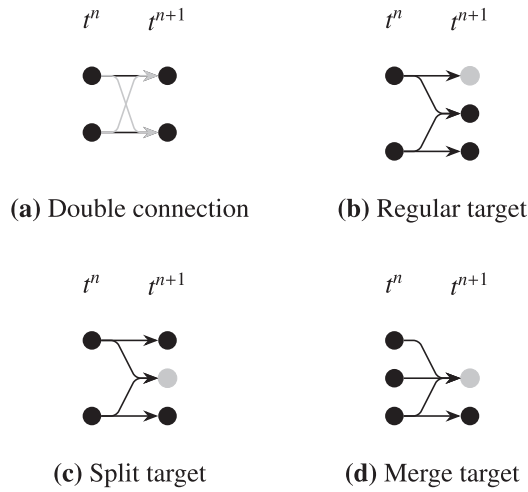


Fig. 6. Schematic representation of different properties of compound events. The special targets and double connections are highlighted through light gray colors.

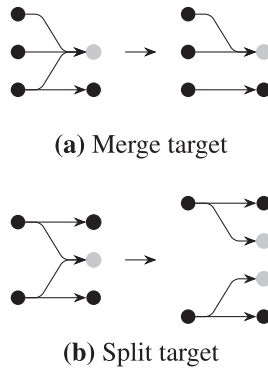


Fig. 7. Schematic representation of the process to decouple a compound event in simpler sub-events. The process is shown for compound events involving one merge target (a) and one split target (b). The special targets are represented by gray circles, where all others have black color. The interactions are shown by directed black arrows.

in a linear compound event, the decomposition algorithm is applied recursively.

After decomposition, all sub-events are tested on physical realizability. If all sub-events are marked as realizable, the decomposition has been successful. Otherwise, the compound event is kept. Instead of applying the recursive loop up to the lowest level, where only single events remain, intermediate checks are carried out after a single target node has been broken up. Then, recursiveness is applied on the remaining events. This allows the partial simplification of compound events into sub-events. In non-linear compositions, several double connections can exist, wherefore the recursive loop is not able to decompose the event. To reduce non-linear compositions to linear ones, for each block of double connections, the edge with the lowest confidence is erased. To avoid the deletion of edges with a high confidence value, a minimum threshold $c_{\min}$ can be assigned above which edges are not discarded.

2.3.3. Event constraints

The constraints applied on the correspondences allow to pre-filter single structure–structure interactions based on their spatial separation. Additional event constraints provide a last step to reject structures from the final event to ensure physical realizability. The type and number of constraints depend on the application. Directional constraints may be suitable for flows with a mean dominant directionality, but the same constraint can be counterproductive when applied to isotropic flows.

The task of an event constraint is to decompose a single event into realizable sub-events. If the constraint is not fulfilled, the original event is kept. Since decomposition is done immediately, the order of application of different event constraints is important. Hence, constraints acting on compound events should always be applied first, to reduce their complexity. A recombination of events is not allowed, because otherwise it is not guaranteed that previously applied constraints are fulfilled anymore.

In several applications, structures rotate, deform and interact with other structures during their lifetime. For example, in incompressible multiphase flows a split and merge between structures requires volume conservation. This fact is used in the volume constraint applied in this work. In the overall event, volume conservation must be guaranteed between all involved source (N_{sources}) and target structures (N_{targets})

$$V_{\text{sources}} = \sum_{i=0}^{N_{\text{sources}}} V_i^n = \sum_{j=0}^{N_{\text{targets}}} V_j^{n+1} = V_{\text{targets}}. \quad (11)$$

Since this conservation cannot be assured completely, especially if the tracking time step increases, a tolerance ε_V can be specified to relax the conservation criterion

$$\Delta V = \frac{\min(V_{\text{sources}}, V_{\text{targets}})}{\max(V_{\text{sources}}, V_{\text{targets}})} \geq \varepsilon_V. \quad (12)$$

For a tolerance of 0.9, for example, 90% of the volume will need to be conserved over each tracking step, regardless of whether volume is gained or lost, for any event to be accepted. In the tracking algorithm of Chan et al. [38] the conservation tolerance between two snapshots is connected to the CFL condition of the simulation. Since this only holds true for certain type of simulations, here, we provide a more relaxed criterion independent of the applied simulation methods.

2.4. Tracking under periodic boundary conditions

For datasets that include periodicity at the boundaries, special algorithmic steps are required to handle structures that possibly intersect and/or cross the periodic boundaries at consecutive tracking steps. When structures intersect the domain boundaries, they are not closed and cannot be used for the geometrical classification unless they are properly reconnected with their periodic continuations on the opposite boundaries. Therefore, periodic reconnection is applied on such structures and the resulting closed structure is moved to one arbitrary side of the domain [49].

The correspondence search is partly based on the spatial coordinates of the structures. If a structure crosses the boundaries, the searches are not able to identify the structures on the opposite side (assuming the structure and the search radius are small enough). Therefore, structures are duplicated on sides where they potentially cross the domain boundaries. Here, duplication means that the spatial position of structures is modified but the geometrical attributes stay the same.

Potentially boundary-crossing structures are identified by their future position, estimated with the linear projection also used for the nearest-neighbor and radius search, and the search radius of the structure. To account for split- and merge events similarly, source structures are projected forward and target structures backward. A crossing occurrence is triggered if either the search radius (Eq. (4)) or the AABB of the projected structure intersects a domain boundary. The latter is used to increase the robustness of the algorithm in cases where the search radius is smaller than the AABB. Then, duplicated structures are placed at the opposite boundary outside the domain. If a structure simultaneously crosses a domain on two sides that are not topologically parallel, the structure is duplicated three times: one for each individual direction and one for the combination of both directions.

Duplicated structures are treated as single objects during the correspondence search and require special treatment when constraints are applied and events are formed. Correspondences that define the same interaction between the same structures are reconnected to a single correspondence. When events are formed, a second reconnection step is done to ensure that each structure is only present once under all events.

2.5. Graph mapping

After all final events have been formed, they are stored into a directed graph [50] to map the temporal evolution of the structures. This resembles the approaches of Lozano-Durán and Jiménez [32], Reinders et al. [35], Laney et al. [51]. A single event gives information of the structure interaction between two consecutive tracking time steps. The evolution of structures over a longer time can be understood by further following the graph connections to previous time steps. At any given tracking time step, each structure is represented in the graph by a node, whereas directed edges connecting nodes indicate correspondences (interactions) between structures in consecutive tracking steps. Here, a strictly directed graph is used, i.e., only forward connections from t^n to t^{n+1} are allowed. Such graph data structure further allows to encode cross-interactions between structures corresponding to different physical fields (e.g., the interplay of turbulent eddies and interfacial surfaces in multiphase flows).

2.5.1. Clustering into branches

All nodes are represented in the graph object, consisting of several subgraphs, which cluster the connections of a set of structures. A subgraph can have several branches, each representing the lifetime of an individual structure. If a structure splits into smaller structures (or is part of a merge event), one of the target (source) structures must be identified as *dominant* in such event, consequently marking the corresponding dominant branch. Branches can be classified into five different types. *Primary* branches represent structures that either continue without any interaction or other structures merge into or split from. In *outgoing* or *ingoing* branches, structures split from other or merge into, respectively. *Connecting* branches represent structures that split from one structure and merge into another. *Reconnecting* branches are the same as connecting branches, but such that the origin and end of the structure is the same primary branch.

The mapping of interactions into connected branches represents a clustering in space-time. Two structures belong to the same cluster either by being contiguous in one of the spatial dimensions or by sharing a common backward/forward connection [32]. To determine the dominant branch of an event involving multiple branches, a weighted sum of N_{db} criteria can be used to formulate a score s_{db} for each connection

$$s_{db} = \sum_{i=0}^{N_{db}} \omega_{i,db} s_i \quad \text{with} \quad \sum_{i=0}^{N_{db}} \omega_{i,db} = 1. \quad (13)$$

Similar to the confidence indices, the weight $\omega_{i,db}$ can be specified to prioritize different criteria for each application. In this work three criteria are used to determine the dominant branch of an event: the confidence value of the interaction, the volume and the lifetime of the structure. To allow the use of the confidence, all weights must also be bounded by $[0, 1]$ and must add to unity.

The volumetric criterion estimates the dominant branch by comparing the volume of a structure to the reference volume of the event V_{ref}

$$s_{i,V} = \frac{\min(V_i, V_{ref})}{\max(V_i, V_{ref})} \in [0, 1]. \quad (14)$$

The reference volume depends on the event type. In merge events the reference volume is that of the target structure and a score is calculated for each source, whereas in split events the reverse applies. For merges, an additional lifetime decider is applied by taking into account the time period a source structure existed before the current tracking step. A functional approach of the form

$$s_{i,life} = 1 - \exp\left(-\frac{t - t_0}{T_{ref}}\right) \in [0, 1] \quad (15)$$

is used, where t is the current tracking time of the source structure, t_0 is the time its branch was created, and T_{ref} is a specified reference time. Hence, longer-living structures have a higher score.

2.5.2. Graph visualization

To further facilitate the visual analysis of the graph object, each structure is represented by a certain node shape. The shape of a target structure in a current tracking step is defined by the event in which it is involved (see legend in Fig. 8). For compound events, a distinction is made between primary and secondary nodes. In the former, structures truly result from a simultaneous split and merge occurring at the same time (gray target nodes in Fig. 6), whereas secondary do not (black target nodes in Fig. 6). Connections are represented by solid or dashed arrows in the graph, originating in one structure and ending in another. Dashed arrows indicate connections with a low confidence value, i.e., they have a low reliability. An example graph is shown in Fig. 8 representing the initial break-up process of a droplet in HIT. For longer simulation times and a larger number of structures, the visualization of the graph becomes more involved.

3. Droplet break-up in HIT – numerical setup

The geometrical characterization and tracking algorithms presented in Section 2 are applied to study the break-up and coalescence of a liquid drop in forced HIT. An initially spherical drop is subjected to the dynamic forces imposed by the surrounding turbulent fluctuations. This leads the drop to deform and move in the flow field. When the disruptive forces from the background turbulence exceed the limit of the restorative force of surface tension, the drop breaks up into smaller fragments. These fragments can then break up into even smaller ones or coalesce with each other forming larger drops again.

3.1. Numerical setup

Governing equations and implementation

The database comprising different tracking time steps used for the present analysis of droplet break-up is obtained from a three-dimensional DNS. We follow the methodology described in Dodd and Ferrante [52,53], Dodd [54], where a full description of the numerical simulation framework is provided, along with systematic verification and validation. Here, we provide only a short summary of the solved equations, numerical discretization, and computational setup of the simulation.

For the DNS of HIT of two immiscible fluids separated by an interface, the continuity equation

$$\nabla \cdot \mathbf{u} = 0 \quad (16)$$

and the dimensionless incompressible Navier–Stokes equations

$$\partial_t \mathbf{u} + \nabla \cdot (\mathbf{u}\mathbf{u}) = \frac{1}{\rho} \left(-\nabla p + \frac{1}{\text{Re}} \nabla \cdot (2\mu \mathbf{S}) + \frac{1}{\text{We}} \mathbf{f}_\sigma + (1 - C) \mathbf{A} \mathbf{u} \right) \quad (17)$$

are solved. Here, $\mathbf{u}$ is the Cartesian velocity vector, p the pressure, ρ the density, μ the dynamic viscosity of the fluid, and $\mathbf{S} = 1/2 (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the strain-rate tensor. $\text{Re} = \bar{\rho}_c \bar{U} \bar{L} / \bar{\mu}_c$ is the Reynolds number, $\text{We} = \bar{\rho}_c \bar{U}^2 \bar{L} / \bar{\sigma}_c$ is the Weber number, where tilde denotes reference dimensional variables: length ($\bar{L}$), velocity ($\bar{U}$), surface tension ($\bar{\sigma}$), carrier-fluid density ($\bar{\rho}_c$) and dynamic viscosity ($\bar{\mu}_c$). Statistically stationary HIT is enforced in Eq. (17) by a linear velocity forcing term in the carrier phase only [55]. As described in Bassenne et al. [55], the time dependent coefficient A is modulated to keep the turbulent kinetic energy of the carrier-phase flow constant. Because A is spatially uniform, the turbulence remains homogeneous and isotropic. The application of the forcing term only in the carrier phase is controlled by the constant C , which equals unity in the drop phase [54]. The equations are solved on a uniform staggered mesh with second-order central finite differences. Time integration is applied by a pressure correction method with a second-order Adams–Bashforth scheme [52].

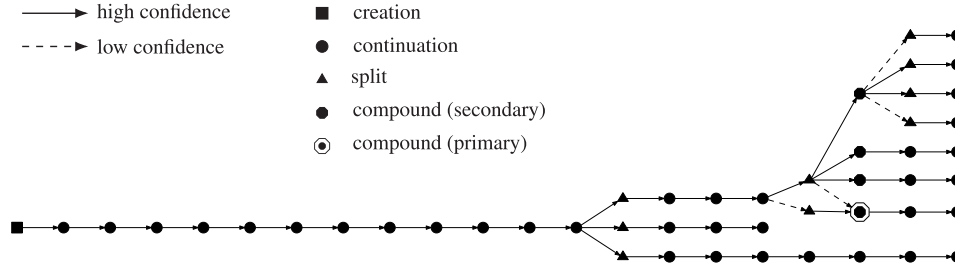


Fig. 8. Example of a graph obtained from the tracking of a lower resolved initial break-up process of a droplet in HIT. The graph shows the occurrence of several continuing structures and includes three split events, a compound event and a disappearing structure. Time increases from left to right.

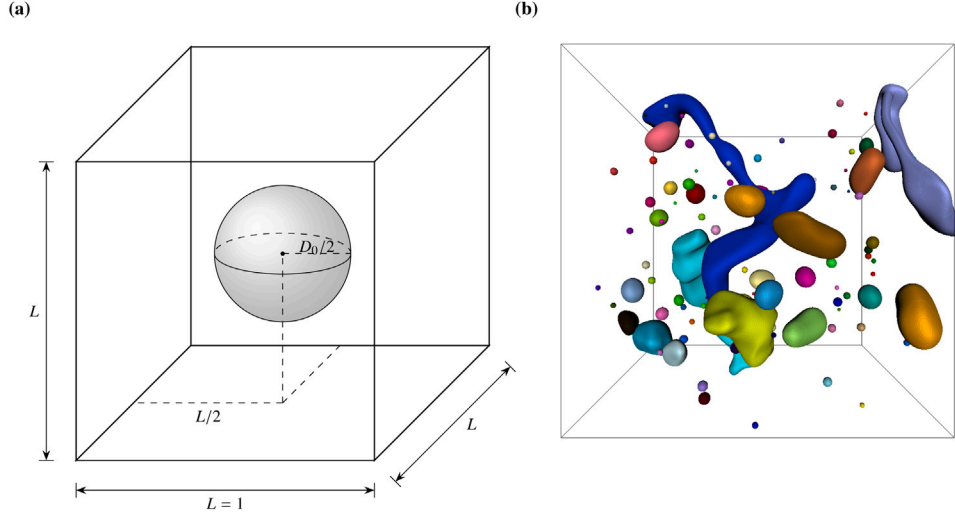


Fig. 9. Triply periodic cubic domain for the DNS of the break-up of an initially spherical droplet (of diameter $D_0 = 0.457$) in three-dimensional HIT (a) and a snapshot of break-up process at the end of the simulation (b). The structures in (b) are extracted at a volume fraction of $f_v = 0.5$ and the different colors represent distinct structures identified in the tracking. Structures intersecting the domain boundaries have been reconnected with their periodic continuation.

Surface tension is considered in Eq. (17) by a continuum surface force

$$\mathbf{f}_\sigma = \kappa \mathbf{n} \delta(\mathbf{x} - \mathbf{x}_s), \quad (18)$$

where κ the interface curvature and $\mathbf{n}$ the interface normal vector [56]. The Dirac δ -function is needed to apply the surface tension force only at the interface position $\mathbf{x}_s$. The interface between the two fluids is captured by a conservative Volume-of-Fluid (VoF) scheme with split advection and a piecewise-linear interface reconstruction (PLIC) [57]. The interface normals are estimated using a Mixed-Youngs-Centered scheme [58] and the curvature is computed using height functions [59, 60].

Initial conditions

The droplet break-up is simulated in a cubic domain of dimensionless side length $L = 1$ and an initial drop diameter of $D_0 = 0.457$ corresponding to an initial volume fraction $\phi_0 = V_0/V_L = 0.05$, where V_L is the volume of the cubic domain and V_0 is the initial volume enclosed by the droplet (see Fig. 9(a)). A Cartesian, structured grid with $N = 560^3$ points is used to discretize the computational domain, with uniform spacing in each coordinate direction. Periodic boundary conditions are applied on all sides of the domain. At the initial state, $t = 0$, the velocity field outside the drop is isotropic and divergence-free, and the velocity cross-correlation spectra $R_{ij}(\kappa)$ satisfies the realizability constraint [61,62]. To ensure these properties, the velocity is derived prescribing an initial energy spectrum $E(\kappa)$ of the turbulent kinetic energy (TKE) of the form [61]

$$E(\kappa) = \left(\frac{3u_{\text{rms},0}^2}{2} \right) \left(\frac{\kappa}{2\pi\kappa_p} \right) \exp \left(-\frac{\kappa}{\kappa_p} \right). \quad (19)$$

κ is the wave number, κ_p the wave number of the peak energy and $u_{\text{rms},0}$ the initial rms-velocity in dimensionless form. To determine the initial energy spectrum, the free parameters κ_p and u_0 must be chosen. Here, they take values of $\kappa_p = 1$ and $u_{\text{rms},0} = 5.095 \times 10^{-2}$. All wave numbers are normalized by the lowest wave number $\kappa_{\min} = 2\pi/L$, where L is the domain length. The dimensionless kinematic viscosity $\nu = 1.16 \times 10^{-4}$ is calculated from the initial Reynolds number based on the Taylor micro-scale, $\text{Re}_{\lambda,0} = 50$, and the initial dimensionless dissipation rate, $\varepsilon_0 = 6.96 \times 10^{-4}$. As turbulence evolves, the Reynolds number initially decays and tends to a value of $\text{Re}_{\lambda,t \rightarrow \infty} = 35.4$. The velocity inside the drop is initially zero.

The initial dimensionless parameters of the DNS are summarized in Table 1. u_{rms} is the rms-velocity fluctuation of the carrier fluid and ν the kinematic viscosity. $\eta = (\nu^3/\varepsilon)^{1/4}$ and $\tau_\eta = (\nu/\varepsilon)^{1/2}$ are the Kolmogorov length and time scales, respectively, Δx is the cell length of the domain and τ_L is the eddy turnover time, defined by the integral length scale and the rms-velocity $\tau_L = L/u_{\text{rms},0}$ [63]. The integral length scale equals the domain side length. Due to the linear forcing term added to the Navier-Stokes equations (Eq. (17)) the energy spectrum will evolve differently from the one used to initialize the simulation (Eq. (19)) [see, for example, 64]. The initial spectrum has been chosen such that the fundamental wave number contains the peak energy. Whereas the choice of initial spectrum is not unique [see 65,66, and references therein], it is expected to affect mainly the transient period of droplet deformation and breakup until a statistically stationary phase is reached.

The density ratio ϕ and the viscosity ratio γ between the fluids are chosen to be unity to neglect effects resulting from variations of these properties. The proposed tracking and analysis methodology is,

Table 1

Initial dimensionless conditions of the flow field (carrier fluid) and the drop (disperse fluid).

Flow conditions	L	N	$u_{rms,0}$	ε_0	ν	λ	η	$Re_{\lambda,0}$	τ_η	τ_L
	1.0	560 ³	0.051	6.96×10^{-4}	1.16×10^{-4}	0.114	6.88×10^{-3}	50	0.408	19.63
Droplet conditions	D_0	D_H	D_0/D_H	D_H/η	$D_H/\Delta x$	φ	γ	ϕ_V	We_{rms}	We
	0.457	0.08	5.7	11.65	44.9	1.0	1.0	0.05	5.84	5000

Table 2

Parameters and tolerances chosen for the tracking algorithm. For the meaning of the parameters refer to the main text and to Section 2.

Structure thresholds			Nearest-neighbor search		Radius search		Bounding-box constraint		Local proximity constraint		Volumetric constraint	
N_{min}	L_{min}	V_{min}	c_{nn}	ω_{nn}	ε_r	ω_r	T_{bb}	ω_{bb}	T_{lc}	ω_{lc}	ε_V	$c_{min,V}$
150	$6\Delta x$	$36\pi(\Delta x)^3$	0.5	0.5	$\sqrt{3}$	0.1	1.5	0.1	0.75	0.3	0.95	0.5

however, applicable to any other choice of these ratios, considered as unitary in this work for simplicity. The Weber number of the flow (inverse of the non-dimensional surface tension) is $We = 1/\sigma = 5000$, resulting in an initial droplet Weber number of $We_{rms} = D_0 u_{rms,0}^2 \rho / \sigma = 5.932$. The droplet is at $Re \gg 1$ such that pressure forces dominate its deformation. Dimensional analysis leads to a critical length scale (Hinze scale) based on the liquid density, surface tension coefficient, and dissipation rate [66]. This critical scale, above which break-up occurs [67], is $D_H = 0.725 (\sigma/\rho)^{3/5} \varepsilon^{-2/5} = 0.08$. Hence, the initial droplet is larger than the Hinze scale, leading to a break-up process. Fig. 9(b) shows isosurfaces of the volume fraction of droplet fluid ($f_V \in [0, 1]$) obtained at an isovalue of 0.5 for the final time of the simulation. Each individual isosurface is indicated with a different color is considered for the tracking.

3.2. Tracking parameters

The choice of tracking parameters used to capture structure interactions in this application is summarized in table Table 2 and discussed in this section. This choice is guided by event and branch statistics obtained by testing different parameter combinations, as well as by inspection of critical break-up and coalescence events (see Section 4 for details).

Structure thresholds

Curvature plays an important role in the numerical simulation of multiphase flows, and in the geometric characterization of the structures used in the tracking methodology. In both cases, minimum conditions are required to provide proper computation of the curvature. The minimum diameter of a droplet in each Cartesian direction is chosen to be at least six times the grid resolution, $L_{min} = 6\Delta x$, to ensure that the interface surface locally contains enough points, accounting for the error generated by the curvature calculation of the VoF scheme. This is done by imposing a minimum volume of individual extracted isosurfaces, $V_{min} = 36\pi(\Delta x)^3$, defined as the volume of a sphere with a diameter of L_{min} . A minimum number of points for each triangulated isosurface, $N_{min} = 150$, is also imposed to ensure sufficiently accurate statistics of the curvedness and shape index in the geometric characterization of each structure.

Nearest-neighbor search

For the nearest-neighbor search, a confidence value and weight must be chosen. Since structures that evolve in primary branches should have the maximum confidence and are the main goal of the nearest-neighbor search, the confidence value and weight are chosen as $c_{nn} = 0.5$ and $\omega_{nn} = 0.5$, respectively. This prioritizes nearest-neighbor search over the subsequent radius search and correspondence constraints.

Radius search

In the radius search, the critical parameter is the definition of the search radius of each structure. In Eq. (4) the radius is defined based on the maximum directional length of the AABB of a structure. This leads to undetected break-up or coalescence events in the corner of an AABB. Therefore, the tolerance of the search radius is increased to $\varepsilon_r = \sqrt{3}$, which allows the detection of such interactions. However, an increased search radius results in a larger number of false interactions, which must be later rejected by the correspondence and event constraints. To minimize the influence of the radius search in the overall confidence and clearly separate it from the nearest-neighbor search, the weight is set to a lower value, $\omega_r = 0.1$.

Spatial correspondence constraints

The spatial constraints acting on the bounding box and the local distance of structures are important to effectively reject found correspondences between structures that can clearly be distinguished by a separation of their AABB's or their local distance. Since the bounding-box constraint is only used to discard truly separated structures, the tolerance is set to $T_{bb} = 1.5$, which also overcomes the difficulties described in Section 2.2.2 for structures that are proximate to each other.

A more significant influence on the overall tracking statistics results from the choice of the tolerance in the local proximity constraint. Therefore, variations in the range $T_{lc} = \{0.5, 0.75, 1.0, 1.5\}$ are investigated to better understand the sensitivity of the constraint. Increasing tolerance values lead to less rejected correspondences and more compound events. Conversely, lower tolerances lead to a larger number of non-matched structures, increasing the number of creation events. These two trends are not completely independent. In cases where more structures are rejected by the local-proximity constraint, less compound events can be formed, which is not always incorrect. Inspection of the resulting events indicates that the increase in compound events from $T_{lc} = 0.5$ to $T_{lc} = 0.75$ is mainly based on this trend. Hence, the tolerance $T_{lc} = 0.75$ gives the best compromise for the present dataset regarding the number of rejected structures and false compound events. Furthermore, the number of incorrectly determined compound events only accounts for around 0.5% of the total number of tracked events.

The confidence weight of the bounding-box constraint is set to $\omega_{bb} = 0.1$ and for the local constraint to $\omega_{lc} = 0.3$, prioritizing the latter for its higher accuracy.

Volumetric constraint

Two parameters can be specified for the volumetric constraint: the minimum confidence value $c_{min,V}$ to decompose compound events, and the maximum relative change of the volume ε_V over an event. A sensitivity analysis is conducted to ascertain the influence of these parameters. The minimum confidence value is increased in the range $c_{min,V} = \{0.3, 0.4, 0.5, 0.7\}$. Up to the value of 0.5, the number of rejected compound events increased successively and inspection of the discarded events confirmed the correctness of these rejections. A

further increase of the confidence value did not increase the number of rejected correspondences, since higher confidence is unusual for split and merge events, primarily resulting from the radius search. For the chosen confidence values and weights of the nearest-neighbor and radius searches, edges with confidence values above $c_{\min,V} > 0.5$ are mostly associated with continuations or correspondences between the source (target) and the largest target (source) structures in a split (merge) event, which should not be affected by this constraint.

The volumetric tolerance is varied in the range $\varepsilon_V = \{0.9, 0.95, 0.99\}$, where higher values indicate a more restrictive constraint and decomposition of compound events into smaller sub-events (splits or merges). A volume tolerance of $\varepsilon_V = 0.9$ leads to stronger activation of the decomposition algorithm of compound events. On the other hand, a value of $\varepsilon_V = 0.99$ raises the number of compound events drastically. This is a result of the exclusion of small, spurious structures from the tracking algorithm, filtered by size or number of points. In split (merge) events, the absence of small target (source) structures (due to filtering) and lead to a violation of such strict volume conservation. Hence, a restrictive tolerance of 0.99 prevents further decomposition of such compound events. To overcome these two described effects—incorrect activation of the decomposition handling for lower tolerances and missing small structures for higher tolerance—the volume tolerance is set to $\varepsilon_V = 0.95$.

Tracking time step variations

We further analyze the behavior of the tracking parameters under different tracking time steps. The tracking step is successively increased in the range $\Delta t = \{0.25, 0.5, 0.75, 1.0\}$, where $\Delta t = 0.25$ corresponds to 280 time steps of the DNS. For larger steps, especially $\Delta t = \{0.75, 1.0\}$, drastically increased numbers of creations and disappearances are found, together with higher numbers of compound events. This confirms the expected behavior: with larger tracking time steps, certain information of the structure interaction is missing (e.g., if two structures merge and then split in between two tracking time steps), leading to incorrectly found correspondences that are then correctly rejected by the spatial constraints. This general behavior is shared with all tracking algorithms found in the literature, as it relates to the loss of information with decreased sampling frequency. If information is missing for certain interactions no proper direct tracking of the underlying events is possible (although it could be modeled). Furthermore, larger tracking time steps lead to larger inaccuracies in the predicted location of the structures using linear projection, which could be mitigated with higher-order projection schemes. A larger number of compound events is also expected for increasing tracking time steps, owing to the greater difficulty of decomposing such events for increased numbers of structures of different sizes and shapes. Therefore, more refined decomposition algorithms are needed to properly reject such interactions. The increased number of compound events obtained for increasing tracking time steps translates into a decreased number of isolated split and merge events.

4. Droplet break-up in HIT – tracking validation

Besides the definition and sensitivity analyses of the different tracking parameters, the simulation and tracking methodology must be validated to ensure that they accurately represent the underlying flow physics and capture all structure interactions and events. As mentioned earlier, the simulation methodology has been extensively verified and validated in [52–54] and is not the focus of the present work. Nevertheless, in this section a brief assessment of the simulation results is given for the statistically stationary state. The tracking algorithm is then validated from an analysis of the statistics of found interactions along with inspection of such interactions and events. The tracking is applied to a total of 875 instantaneous snapshots (tracking steps) extracted from the simulation at a constant tracking time interval of $\Delta t = 0.25$. The simulation covers a total time period of $t_{\text{final}} \approx 11.15\tau_L$ starting from the initial spherical droplet.

4.1. Extraction of structures

The tracking algorithm operates on extracted isosurfaces from the scalar field defined by the volume fraction f_V . All surfaces represent isocontours of $f_V = 0.5$, obtained by the marching-cubes algorithm followed by a smoothing filter to remove spurious high-frequency surface waves (ripples) that can lead to artificially increased curvedness and reduced shape index values, especially for larger surfaces. An interpolation kernel with a windowed sinc function is used to avoid shrinking of smaller structures towards their centroid, which is a drawback of other smoothing techniques [68].

4.2. Stationary state and structural distribution

The evolution of the number of structures, N_s , over the time, t , normalized by the eddy turnover time, τ_L , is shown in Fig. 10(a). In the beginning, $t/\tau_L \lesssim 0.75$, the initially spherical droplet deforms without any break-up events. At later times, an exponential increase in the number of structures indicates a large number of break-up events with a smaller number of intermediate coalescence events. After approximately $t/\tau_L \approx 6$, the number of structures oscillates around a statistically stationary value, indicative of a balance between break-up and coalescence events.

Fig. 10(b) presents the probability density function (pdf) of the structural sizes in the statistically stationary state ($t/\tau_L \gtrsim 6$). The size of a structure is computed by the characteristic diameter of a sphere with the same structural volume, $D = (6/\pi)V^{1/3}$. Furthermore, the size is normalized by the Hinze scale, D_H , to separate the sub- ($D/D_H < 1$) and super-Hinze regimes ($D/D_H \geq 1$), as indicated by the center dashed line. The black line denotes the mean value over all time steps in the statistically stationary state, whereas the shaded gray area represents half of the standard deviation around that mean. Over the whole stationary state, a bi-modality in sub- and super-Hinze scales can be observed in the distribution of sizes. The drop observed for larger size indicates that structures larger than the Hinze scale are unstable and prone to break up, validating the Hinze scale as a critical diameter for the break-up process. All structure sizes lie in between the initial droplet size (right vertical dashed line in Fig. 10(b)) and the minimum volume chosen for the tracking (left vertical dashed line).

Fig. 11 shows pdfs of all structures considered during the entire simulation mapped onto the geometric feature space $(\hat{S}, \hat{C}, \lambda)$. Only a portion of the whole feature space is filled. Most of the structures lie in the region of spherical droplets ($\{\hat{S}, \hat{C}, \lambda\} = \{1, 1, 1\}$). As structures get more stretched, they move towards lower shape index values, indicating the predominance of filament-like or tube-like structures. Highly stretched or sheet-like structures (low values of $\hat{C}$) are not present in the current dataset of droplets obtained from this simulation. This indicates that sheet-like structures are counteracted by the surface tension forming more tube- and blob-like structures. When structures tend to higher stretching (implying lower λ), splitting is initiated, bringing them back towards a more compact and energetically stable shape. Since a large part of the feature space is empty and no structures can be found in those regions during the whole simulation, only the narrowed subdomain ($0.4 < \hat{S} < 1.0$, $0.5 < \hat{C}$, $0.4 < \lambda < 1$) of this space is shown in further analyses.

4.3. Event and branch statistics

Before the evolution of structures and their dynamics are investigated in more detail, the tracking algorithm is validated based on the found events in the whole simulation and in the stationary state. In Table 3, the number of found interactions is shown together with their order for the full run and for the stationary state.

The order of split events is defined as the number of additionally created structures by the break-up (i.e., in a first order split event, one structure breaks up into two smaller ones). The opposite holds for

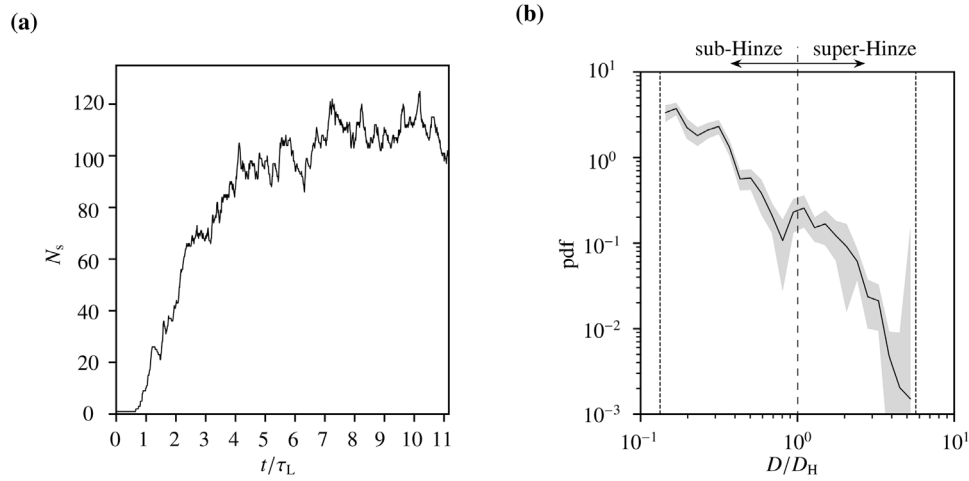


Fig. 10. Number of structures considered by the tracking algorithm over the full simulation period $0 < t/\tau_L \lesssim 11.15$ (a) and overall distribution of structure sizes in the stationary state $6 \lesssim t/\tau_L \lesssim 11.15$ (b). The sizes are normalized by the Hinze-scale. The black line in (b) represents the mean value and the shaded gray area half of the standard deviation around that mean.

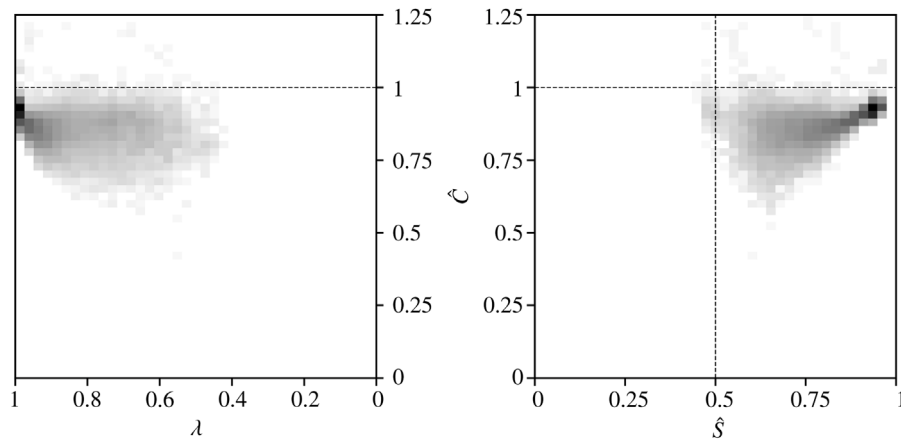


Fig. 11. Feature space of all structures considered by the tracking algorithm over the full simulation period $0 < t/\tau_L \lesssim 11.15$. The dashed lines mark the region of tube-like structures (vertical) and spherical curvedness values (horizontal).

Table 3

Overall statistic of events found by the tracking algorithm for (a) the whole simulation period, $0 < t/\tau_L \lesssim 11.15$, and (b) the stationary state, $5 \lesssim t/\tau_L < 11.15$. Rows represent the order of found events, where an order of 1 indicates that one structure splits into two smaller structures and vice versa. For compound events only the total number is shown as we do not define an order (see main text).

(a)				
Order	Merge	Split	Compound	Total
1	535	398	N/A	933
2	13	108	N/A	121
3	2	6	N/A	8
4	0	5	N/A	5
Total	550	517	44	1111

(b)				
Order	Merge	Split	Compound	Total
1	344	234	N/A	578
2	8	51	N/A	59
3	2	4	N/A	6
4	0	3	N/A	3
Total	354	292	25	671

merge events. In compound events, no order can be specified, since several structures break-up and merge simultaneously. Therefore, only the total number of tracked compound events is shown.

Among the 1,111 events captured for the full simulation period, a slight ($\approx 6\%$) dominance of merge events over split events is observed. The total number of split and merged structures can be calculated by multiplying the number of events, N_o , by their order, o , and taking the difference between split and merge events

$$\Delta N = \sum_{o=1}^4 (N_{o,\text{split}} - N_{o,\text{merge}}) o = 85. \quad (20)$$

The excess of split structures is commensurate with the number of structures found at the end of the simulation, $N_{\text{total}} = 99$, validating the number of events found by the tracking algorithm. The difference between both values stems from unconsidered compound and creation events. It further indicates that most of the events are captured properly. Otherwise, a predominance of creations or disappearances would be present. In the stationary state, the difference yields $\Delta N = -6$, indicating the balance between break-up and coalescence events. In both cases, the differences show that a similar number of source and target structures are involved in compound events. The statistics also emphasize that merge events predominantly occur in first-order coalescences between two structures, whereas approximately 30% of

Table 4

Overall statistic of creations and disappearances found by the tracking algorithm (a) and created branches (b). The data is shown for the full tracking period, $0 < t/\tau_L \lesssim 11.15$.

(a)				
N_c	N_d	$N_{c \rightarrow d}$	$\bar{t}_c/\tau_L$	
207	205	200	0.05	

(b)					
Primary	Secondary				Total
	Incoming	Outgoing	Connecting	Reconnecting	
202	6	102	511	77	898

split events are of order higher than one for the full simulation period. This number decreases to $\approx 25\%$ when considering only the statistically stationary state.

Beside split, merge and compound events, creation and disappearance events also occur. Starting from an initially single droplet, in multiphase flows without reactions, in/outlets, or mass sources/sinks, no structures should be created or disappear during the simulation. Due to the chosen minimum limits of the structure length, $L_{\min}$, volume, $V_{\min}$, and number of points of the surface triangulation, $N_{\min}$, it can happen that at one time instance structures do not fulfill those criteria, whereas they do in the next tracking step (as they deform). Hence, structures can be virtually created from the tracking standpoint. The same reasoning holds for disappearing structures whenever they fall below the specified minimum limits in a subsequent tracking step. To verify that the created and disappearing structures result from those effects, each branch starting from a created structure is followed during its lifetime until the structure either disappears or merges with other structures. In Table 4(a) the number of creations and disappearances together with the number of creations ending in disappearances $N_{c \rightarrow d}$ and the average lifetime of a creation $\bar{t}_c/\tau_L$ are shown.

Only 3% of the creations end in larger structures through merging, and the same amount of disappearances result from previous interactions. All other creations end in disappearances and have an average lifetime of $\bar{t}_c/\tau_L = 0.05$, which corresponds to approximately 3–4 tracking steps. This indicates that such short-lived structures are only artifacts resulting from the chosen minimum limits. In fact, the volume and number of surface points is near the chosen limit for all created and disappeared structures. Since most of the creations do not split or merge during their lifetime and they simply end in disappearances, a simple filtering of such structures is done prior to further analysis.

Missing events are determined by calculating the volume change of structures that do not split or merge with other structures between two consecutive tracking steps. Volume changes above the chosen minimum volume indicate a merge or split with a filtered virtual structure. Similarly, incorrectly captured events are identified when the relative volume conservation over an event is below 99%. By inspection of these structures and events, it is found that application of the proposed tracking algorithm to this dataset results in less than 2% of non-captured events, missing interactions, wrong events and occurring non-physical creations and disappearances.

Table 4(b) shows the number of branches obtained from the tracking. The high number of primary branches only results from the number of created structures, which disappear after only a short time period. The large number of outgoing branches is explained since it accounts for the termination of branches for all the structures that exist at the end of the simulation. The number of incoming branches coincides with the number of created structures in the field merging with other structures at their end of life. 85% of the structures—excluding primary, outgoing and incoming branches—originate from one structure and merge into another (connecting branches), whereas all others evolve and re-merge or reconnect with the same structure. This clearly indicates that, in the present dataset, no predominant structure can be

found to live for the whole simulation time period. Each structure at least splits or merges once in its life. Therefore, no distinction is made between reconnecting and connecting branches.

5. Droplet break-up in HIT – temporal evolution and geometry of structures

The following analysis is focused on the entire simulation period and considers only non-primary branches that originate from the initial single droplet. Due to the periodicity, structures do not leave the domain. Therefore, all structures at some point are part of a split or merge event. In such scenarios, primary branches mainly describe artifacts of the structure thresholding leading to the spurious formation of creations and disappearances. Compound events, which account for only 4% of found events, are not included in this analysis, since the complex structure interactions they represent often cannot be classified into merges or splits, nor unambiguously assign a single branch to determine the lifetime of their involved structures. The chosen isosurface value of $f_V = 0.5$ represents the geometric local midpoint between both phases. Since the numerical thickness of the interface is one grid cell in the present DNS (with approximately 45 grid points per Hinze scale), higher or lower isovalues might result in slightly larger and more convoluted structures in one phase and smaller structures in the other phase. This would increase the number of creation, disappearing and split–merge events. Since those events are excluded from the current analysis, the isosurface value has no impact on the presented results.

5.1. Size ratio of structures over events

The analysis of the sizes of structures before and after break-up or coalescence is relevant, for example, in the study of mixing processes. We define the mean diameter per (split or merge) event j as

$$\bar{D}_j = \frac{1}{N_{s,j}} \sum_{i=1}^{N_{s,j}} D_{j,i}, \quad (21)$$

where $N_{s,j}$ is the number of structures after the split or before the merge event, and $D_{j,i}$ is the equivalent diameter of each structure i resulting from the split or involved in the merge, respectively. The equivalent diameter of each structure, $D_{j,i}$, is then compared to this mean value, $\bar{D}_j$. The mean diameter per event, $\bar{D}_j$, is also compared to the Hinze scale, D_H . Considering all split and merge events, a joint probability density function (jpdf) is formed in terms of $\bar{D}/D_H$ and $D/\bar{D}$, where the event and structure subindices have been dropped. The results are shown in Fig. 12 for structures after split and before merge events. Values of $\bar{D}/D_H > 1$ indicate that at least one structure in the event is larger than the Hinze scale, whereas lower values imply that all structures split into sub-Hinze structures or merge from them. A size ratio $D/\bar{D} = 1$ further indicates a break-up into structures of similar sizes, whereas values greater than unity represent unevenly-sized structures involved in the event.

In split events (Fig. 12(a)), the mean size of the resulting structures is predominantly larger than the Hinze-scale. Split leading to unequal sizes of the structures ($D/\bar{D} \neq 1$) are dominant, but there is also a fair number of split events resulting in equal sizes. Structures below the Hinze scale ($\bar{D}/D_H < 1$) split into more equally-sized structures, but the number of such events decreases, consistent with Martínez-Bazán et al. [69]. In the next section, it will be seen that this is a result of cascade events, where larger structures successively split into smaller ones. The marginal pdf of the mean diameter shows a mild bi-modality between the sub- and super-Hinze regimes, with a less populated region slightly below the Hinze-scale, $\bar{D}/D_H \lesssim 1$.

In merge events, source structures are predominantly of super-Hinze size. Nearly all merges involve a large structure merging with a much smaller one. This indicates that most coalescences result from large structures occupying more space in the domain, which increases the probability that other structures will merge with it. Since the size

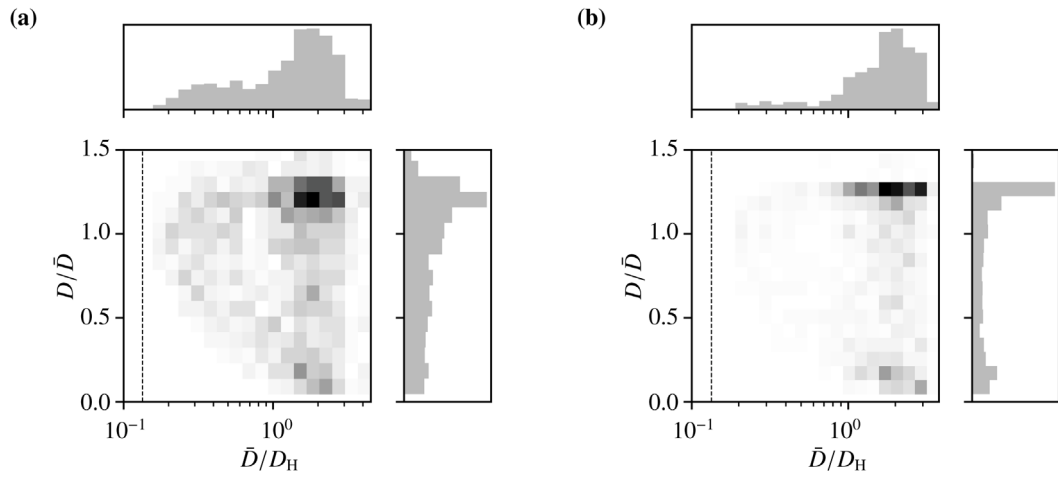


Fig. 12. jpdf of the size ratios with their corresponding marginal pdfs of structures after split (a) and before merge events (b). Equal split or merged structures are represented by a ratio $D/\bar{D} = 1$. The mean equivalent diameter of the involved structures is normalized by the Hinze scale, D_H . The vertical dashed lines represent the minimum equivalent diameter used for the tracking.

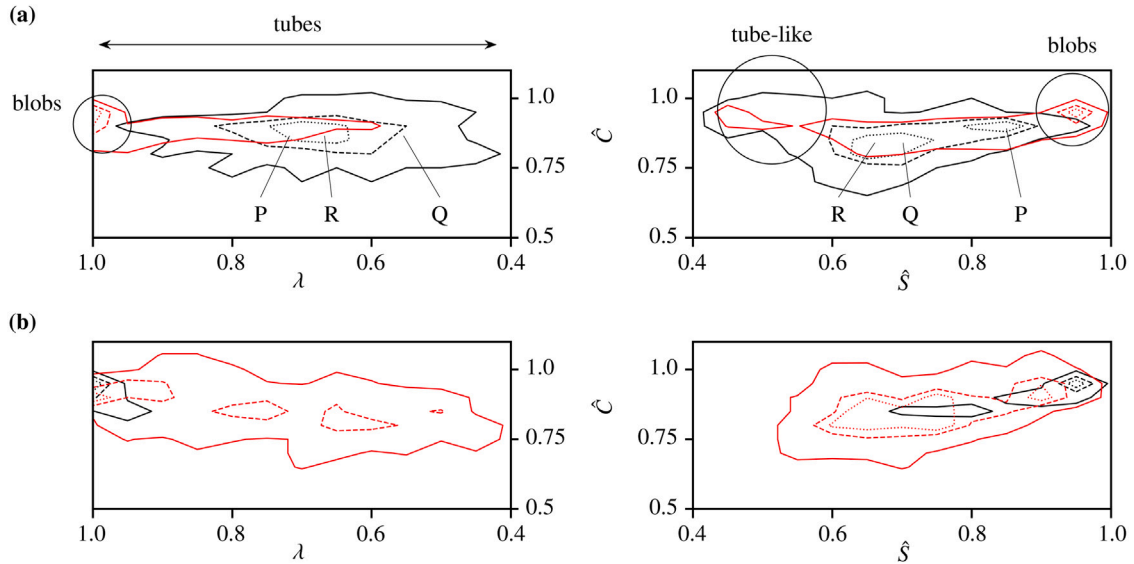


Fig. 13. Contour lines of the jpdfs of structures involved in split (a) and merge (b) events. Source structures are represented in black and target structures in red. The contour lines are given for three fractions of the maximum value: — 0.1; - - - 0.5; ···· 0.75. The letters P, Q, and R refer to structures represented in Fig. 14.

distribution in Fig. 10(b) showed that more sub-Hinze structures are present in the domain, the probability that such smaller structures are part of a merge with a super-Hinze structure increases. The small tail of the marginal pdf of the mean diameter in Fig. 12(b) indicates that merge events between sub-Hinze structures are also present, although, in such scenarios, the involved structures are, more likely, closer in size.

5.2. Geometric and structural changes over events

A variety of tube- and blob-like structures with different stretching is found in the dataset (refer to Fig. 14). To elucidate whether particular structure shapes tend to split or merge more predominantly, Fig. 13 shows contour lines of the jpdfs of source (black) and target (red) structures in the feature space of geometric parameters for three different fractions of the maximum probability.

In merge events, the larger area covered by the resulting pdf of target structures indicates a high variety of structure shapes involved in merge events. Also, the peaks at different λ values do not indicate a predominance of structures resulting from merge events. The distribution shows a dominance of spherical structures coalescing with other

structures. Similar to the results found for the size ratio, this is a simple consequence of the flow homogeneity in all directions (with periodicity applied at all boundaries). Therefore, in the following analysis, only break-up events are considered.

Significant changes in the geometrical shape are present in split events. Before the structures break up, they are mostly tube-like structures of different stretching (indicated by the wide range of λ values). The largest number of structures split at intermediate values of the compactness parameter, λ . In Fig. 14, three structures of different predominant regions of the feature space are visualized, with the distribution of the shape index and curvedness mapped onto their surface. Additionally, the jpdf and corresponding marginal pdfs of S and C are given. The location of those structures in the feature space of geometric parameters is indicated by letters (P, Q, and R) in Fig. 13. The predominance of tube-like structures results from the formation of ligaments and bridges between two larger, more compact structures (structures P and Q in Fig. 14) or evolving on one side of the structures surface (structure R in Fig. 14). The compactness parameter, λ , is then dependent on the structures with which the ligaments are connected and on their size. Lower λ values (i.e., more

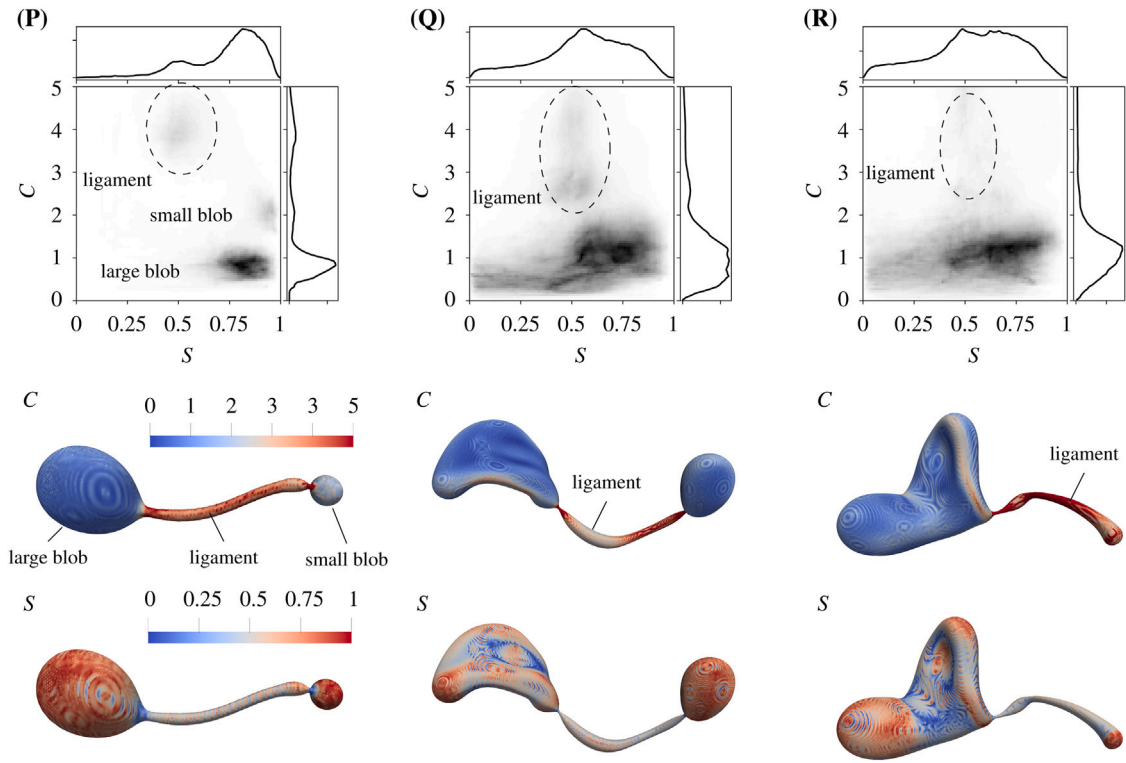


Fig. 14. Representation of different structures (P, Q, R) present immediately before they split into smaller sub-structures. For each structure, the jpdf and corresponding marginal pdfs are shown for the curviness C and shape index S . Darker and lighter colors indicate higher and lower values, respectively. Additionally, the curviness (top) and shape index (bottom) is mapped on the structures surface. The colorbars of C and S shown on the left are common to all three structures. The upper color bar represents the curviness and the lower one the shape index. The letters P, Q, and R correspond to the locations in geometric space of parameters highlighted in Fig. 13(a).

stretched structures) indicate the formation of larger ligaments between the outer elements (e.g., structure Q). In cases in which the outer structures are more spherical (as found for smaller ones), and therefore more compact, the shape index tends to higher values (region of high density indicated by the black dotted contour line above $\hat{S} > 0.8$ on the $\hat{S}\hat{C}$ -jpdf in Fig. 13(a)). If the outer structures are larger and, therefore, often bulgier, the shape index and curviness decrease (corresponding to the other region of high density of the $\hat{S}\hat{C}$ -jpdf, represented by the black dotted contour line around $\hat{S} \approx 0.7$). Immediately before the split occurs, spherical or blob-like structures are unlikely, since they previously must undergo a certain deformation towards more stretched structures to exceed the restoring force of surface tension. In all scenarios, the formation of the ligaments can be observed in the jpdf (Fig. 14). The curviness has predominant values of $C < 2$. The formation of the ligaments, especially the constriction at the outer parts of the tubes, brings the curviness to higher values. More significant is the formation of longer tails of higher curviness for values of the shape index $S \approx 0.5$, indicative of the region where cylindrical shapes are predominant. The formation of the ligaments leads to higher curviness values especially in this region. This increased curviness can be more or less pronounced depending on the shape and relative size of the remainder of the structure at both ends of the ligament. If the side elements are more spherical (structure P in Fig. 14), a clear peak at $S \approx 0.5$ can be observed. The bulgier and larger the side structures are (structures Q and R), and the less circular-cylindrical the ligament is, the less defined this high-curviness peak will be, blending with the pdf of the remainder of the structure. Nevertheless, the ligament formation always leads to an increase of the curviness at a shape index $S \approx 0.5$. Marginal pdfs obtained at values of S along with higher order moments of the jpdf can help identify the formation of these ligaments, which could then be used to predict split events following the evolution of a structure.

The mapped shape index and curviness highlight the distinction between the ligament elements and the outer, compact structures. The

curviness can clearly be distinguished between the side structures (lower values, blue) and the ligament (higher values, red), owing to the smaller radius of curvature of the ligament. Hence, a local analysis of the curviness can help separate those regions and to identify ligament-like regimes. Similarly, the distribution of shape index values can be used to identify ligaments and end regions, although less clearly. In the tube-like ligament connecting the two ends, the shape index uniformly takes values of $S \approx 0.5$ (cylindrical shape), in contrast to the side structures, which can have a broader range of S values (for spherical shapes, predominantly $S \approx 1$). Depending on the shape of those structures, the distinction can be more ambiguous (compare structures Q and R to structure P), but the combination of shape index and curviness allows a clear identification of the ligaments, even in earlier stages of the deformation.

In multiphase flows, spheres represent more stable structures in terms of the effect of surface tension. The dominance of blob-like structures ($\{\hat{S}, \hat{C}, \lambda\} = \{1, 1, 1\}$) as a result of split events observed in Fig. 13(a), indicates that, over break-ups, such stable conditions of the structure geometry are preferred over more cylindrical ones. Nevertheless, the formation of such ligament structures is not completely avoided. A long tail in the jpdf shows that tube-like structures ($\hat{S} \approx 0.5$) are formed (black solid line). Furthermore, structures that break up are also present in the tube-like region (red solid line), indicating the formation of cascade events, where structures split in several consecutive steps towards more spherical ones.

In Fig. 15, a representative cascade event is shown together with the subgraph obtained from the tracking algorithm, indicating the formation of several branches. The temporal evolution of the shape index and the compactness parameter during the cascade event is also given. In that figure, darker gray colors represent more stretched structures (i.e., lower λ). Before the actual split event starts the formation of a long bridge between two more compact structures can be observed, which is a similar behavior seen for other structures in Fig. 14. The whole

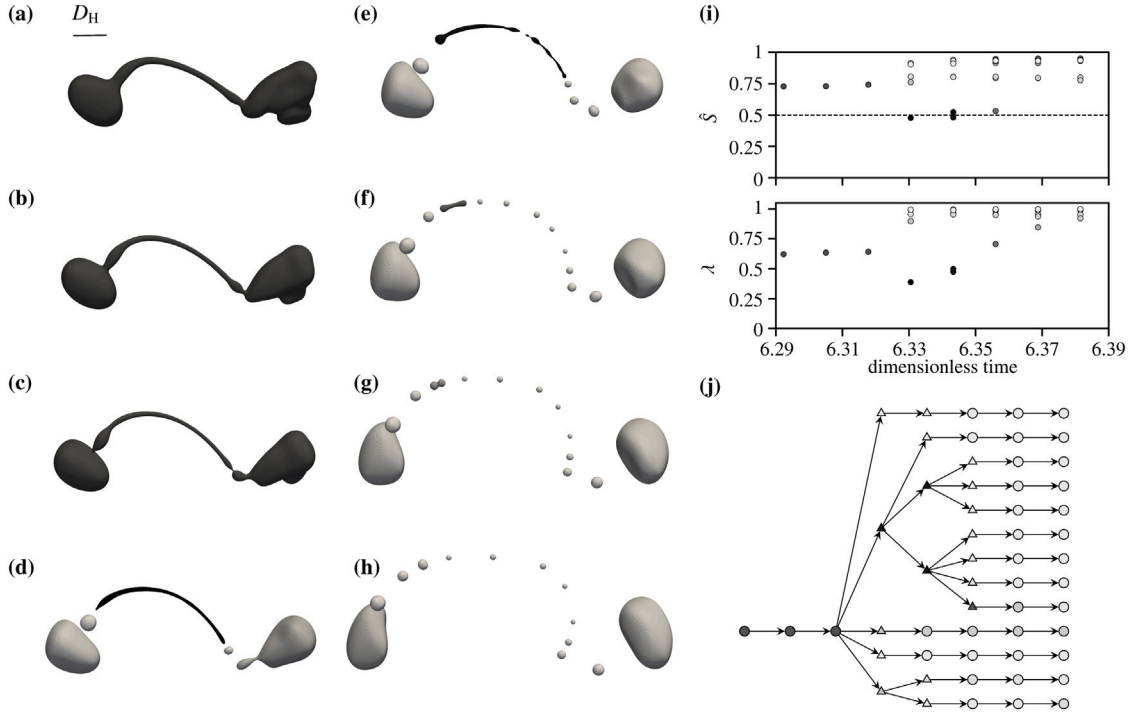


Fig. 15. Example of a cascade event representing a split of a larger structure into a larger number of smaller ones in several steps (c)–(f). Two steps before (a)–(b) and after the cascade (g)–(h) are shown additionally. The black bar in (a) represents the Hinze scale. In (i) the evolution of the structures over the normalized time, t/τ_L , is shown for the compactness parameter, λ , and the shape index, $\hat{S}$. Furthermore, the cascade process is represented in the extract of the graph built by the tracking algorithm (j). In all visualizations, the color represents the compactness parameter, λ . Darker color indicates more stretched structures (lower λ), whereas lighter color indicates more compact structures (higher λ).

structure is larger than the Hinze scale, whereas the side structures are approximately of that scale. As the bridge evolves, a constriction of its region of contact with the end structures initiates the split. The order of the split depends on the length and thickness of the ligament. Larger and thicker bridges tend to split into higher order splits. Due to the constriction at the contact points, a low-frequency wave is formed on the surface, leading to subsequent constriction at different points of the ligament, a manifestation of Rayleigh instability [70]. If the ligament is thick enough, the constriction mainly leads to further splits of spherical structures at the outer side. If the thickness decreases, then intermediate splits into smaller cylindrical sub-parts are also possible (Fig. 15(e)). In the final stage, the whole bridge is broken up into several smaller spherical structures, which are then stabilized by surface tension. In cases where the bridge is small enough, a split-up is avoided by attracting the side structures towards each other and re-merging into a single blob (see evolution of the dark bone-like structure in Fig. 15(f)–(h)). Nevertheless, in all cases, the initial tube-like ligament splits into several blobs in the final stage of the cascade.

The structures originally connected by the bridge remain nearly unchanged after the separation of the ligament. This mainly depends on the shape of such structures. Here, both end structures are compact (high λ) structures, which are less prone to split and only deform due to the flow field. This behavior can be seen following the cascade in the time evolution of λ and $\hat{S}$ in Fig. 15. After the separation of the bridge, the outer structures return to values of $\{\lambda, \hat{S}\} \approx \{0.85, 0.75\}$ indicating their compact form. In Fig. 13(a), it is observed that such compact structures are less prone to split again, leading to a continuous evolution in the geometric feature space. The increase of the compactness parameter and shape index for the ligament results from the detachment of the more compact regions connected to the ligament ends before the split. For the duration of the cascade process, stretched tube-like structures are present. The evolution of the bone-like structure can be best seen in the gradual increase towards higher values of λ .

Cascade events, in general, can occur in different scenarios and time spans. In the current dataset, cascades up to a duration of 14

tracking time steps ($\Delta T/\tau_L = 3.5$) are observed. In such cases, intermediate merge events occurred, leading the structures to grow and split successively. The time span of a cascade mostly depends on the length and size of the formed ligament. The longer and thinner the tube is, the faster the cascade occurs. Finally, in all cases, only spherical structures lower than the Hinze scale remain after the cascade. To better understand the behavior and initiation of such cascades, a more detailed analysis of the geometrical distribution and, especially, of the interaction with the surrounding flow field is needed. Additionally, a modal analysis could reveal whether low-frequency surface waves can drive the cascade to break up long ligaments in only a few steps.

5.3. Structure evolution over time

So far, only the change of geometry over events has been investigated, emphasizing the structure evolution present in cascade events. To better understand the evolution of structures towards certain events, Fig. 16 shows the dimensionless lifetime of structures between two events in terms of the jpdf of the lifetime and the size of the structures, along with the corresponding marginal pdfs. A distinction is made between structures that split (Fig. 16(a)) or merge (Fig. 16(b)) at the end. The characteristic average size, D^* , of a structure is given by its mean size over the lifetime

$$D^* = \frac{1}{T} \sum_{i=0}^{N_{\Delta t}} D_i, \quad (22)$$

where $N_{\Delta t}$ is the number of tracking time steps between the events, T the dimensionless lifetime of the structure, and D_i the equivalent diameter of the structure in each tracking step. The predominance of short lifetimes of structures of super-Hinze scales (region B in Fig. 16(a)) indicates the geometric validity of the Hinze scaling, in that such structures are prone to split after a short time, and that the Hinze scale, D_H , represents a critical diameter. Smaller structures (region C in Fig. 16(a)) are less prone to split further, but if they do, the break-up occurs in only one or two tracking steps, which is another indication of

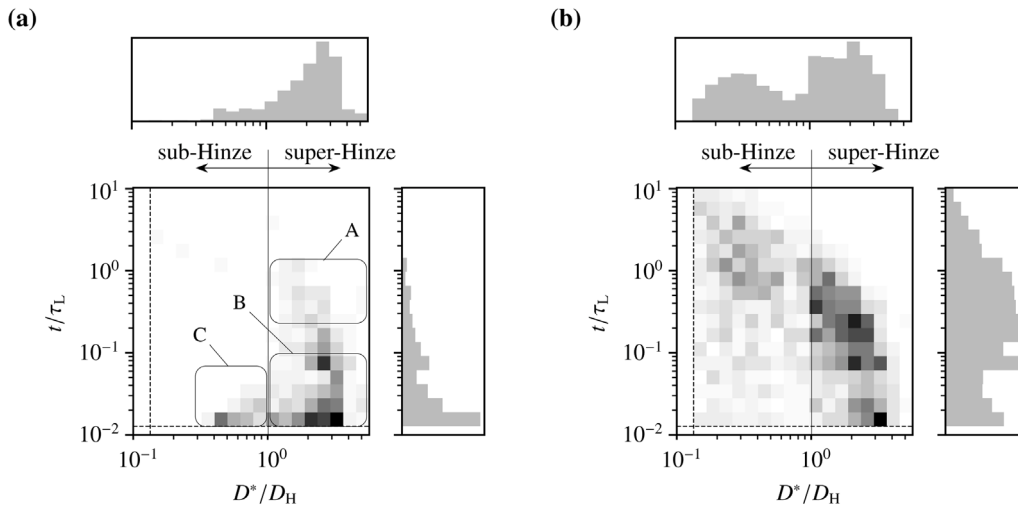


Fig. 16. jpdf and corresponding marginal pdfs for the structure lifetime t/τ_L between two events before the structure splits (a) or merges (b). Darker colors represent higher values and lower ones are given by lighter colors. The vertical dashed line represents the equivalent diameter for the minimum volume used in the tracking, whereas the horizontal dashed line marks the tracking time step, Δt . A unitary dimensionless lifetime is approximately equivalent to 80 tracking steps of Δt .

the occurrence of several cascade events. The low probability density found for long-living structures of sub-Hinze scale indicates the stability of those structures under different flow conditions. Furthermore, the same region is more populated in the jpdf corresponding to merge events (Fig. 16(b)), again indicative of the effect of flow homogeneity (periodicity). Without periodic boundary conditions, the smaller structures are unlikely to split unless they are part of cascade events, for which more tubular shapes are found. Otherwise, surface tension keeps them under stable conditions.

The less populated region for smaller structures and short lifetimes before they merge, again, is mainly affected by the formation of the smaller structures after split events. In the cascade shown in Fig. 15, it can be observed that the smaller structures are always at a certain distance to each other, which results from the split of cylindrical structures and the subsequent re-formation towards spherical ones. When two spheroids split from the ends of a ligament that connected them, a certain traveling time is required before the detached spheroids can re-merge.

Besides the short-living structures, super-Hinze structures can also live for a longer time before they split (region A in Fig. 16(a)). The maximum lifetime of a super-Hinze structure observed in the current dataset is 250 tracking time steps or nearly 30% of the whole simulation period ($\Delta T/\tau_L = 62.5$). Intermediate lifetimes (50–80 tracking time steps, $\Delta T/\tau_L = 12.5$ –20) occur more frequently. This behavior can be linked to the energy spectrum of the turbulence, where the strain rate of the bubbles increases as their sizes decrease [71].

In Fig. 17(e), an illustrative trajectory of such intermediate-living structures is shown in the feature space of geometric parameters. Furthermore, the evolution of the structure is shown with jpdfs and marginal pdfs, along with a visualization of the structure at four specific tracking time steps (Fig. 17(a)–(d)). The evolution follows a common pattern that can especially be observed in the compactness parameter. The initial structure is less stretched (see (a)) and has intermediate shape index and curvedness values. In the first stages, the structure remains compact (i.e., high λ). The stronger variation in curvedness and shape index can be interpreted as a pulsing of the structure (see (b) and (c)). Once the pulsation leads to a longer deformation in one direction, the split is initiated and the structure deforms further. Once the stretching reaches a critical value (i.e., low enough λ), the deformation is large enough to lead the structure to split (d). This marks the final stage of the lifetime of such a structure. In the final stage, again the ligament formation can be identified in the jpdf (similar to the other structures represented in Fig. 14).

The occurrence of the pulsation can be seen as a strong interaction of the interface with the surrounding flow field. Under stable conditions, the background flow leads the structure to deform in one direction. Surface tension then counteracts this change of shape to return the structure to its original state. This behavior repeats until the force applied from the flow field is strong enough to overcome the restoring force of the surface tension, bringing the structure into an unstable state. Finally, this instability leads to the split of such structures into more stable sub-elements. From the given evolution, hypothetically, flow conditions could be found for which even super-Hinze structures can live for a very long time. But as soon as those stable conditions are left, a split of such structures is initiated.

6. Conclusions and future work

In this work, a new hybrid attribute- and regional-based tracking methodology is introduced to study the dynamic behavior and temporal evolution of multi-fluid structures in multiphase flows. From the volume fraction of a DNS simulation, structures are extracted from the three-dimensional field and characterized by three non-dimensional geometrical parameters: shape index, $\hat{S}$, curvedness, $\hat{C}$, and compactness, λ . Those geometric parameters, together with the spatial coordinates of the extracted structures (x, y, z), form the basis of the tracking algorithm. Correspondences between structures of two consecutive time instances are found by using a nearest-neighbor search followed by a radius search. Found correspondences are then filtered based on physical realizability using two spatial constraints. The structures and the resulting set of physically realizable (accepted) correspondences are then grouped into events that can involve: one source and one target structure (continuation), several sources and targets (split, merge and compound events), or none (disappearance and creation). In the final step of the methodology, all events are mapped into a directed graph which can then be queried to retrieve information on the dynamics of the extracted structures. All structures are clustered into branches representing their lifetimes, where the beginning and end of a branch coincide with the creation and disappearance of the structure, respectively. With this information, the evolution of the structures and their mutual interactions can be analyzed.

The tracking methodology is applied to a dataset obtained from DNS of the break-up of a spherical liquid drop in forced HIT [52,53]. Owing to the periodic boundary conditions, the simulation represents an infinite series of drops uniformly spaced ≈ 2.19 diameters apart in each coordinate direction. A validation of the outcome of the tracking

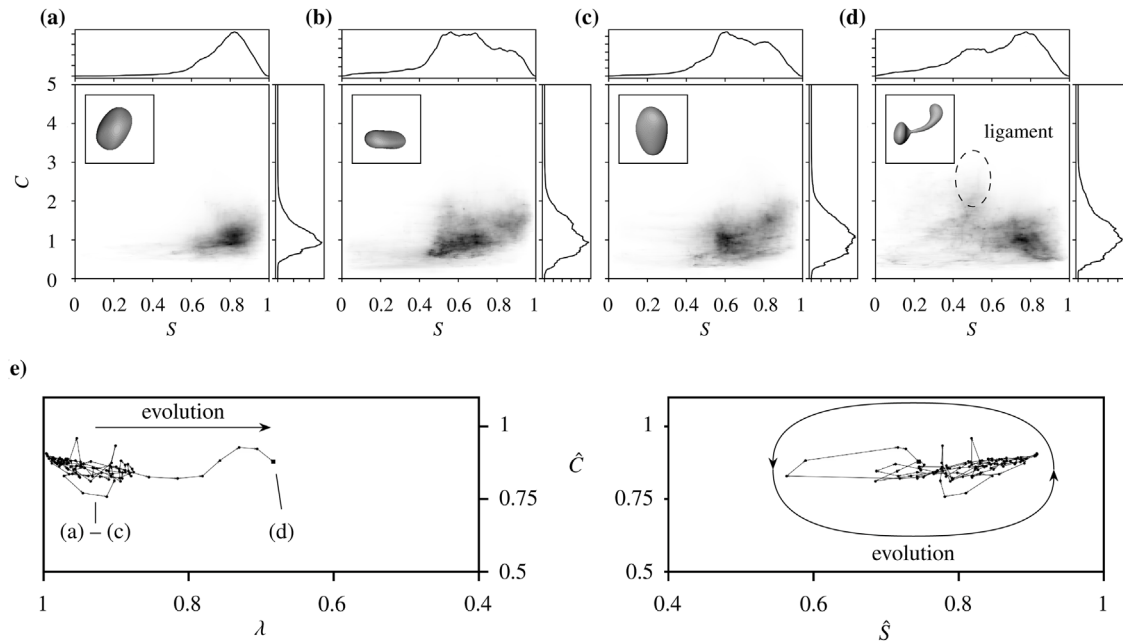


Fig. 17. Representation of the evolution of a long-living (100 tracking time steps, $\Delta T/\tau_L = 25$) super-Hinze structure. The full evolution is shown in projections of the feature space of geometric parameters (e). For four different steps of the evolution, $\Delta T/\tau_L = \{1, 7, 14, 25\}$, the jpdf and marginal pdfs are shown together with a visual representation of the structures (a) – (d).

is done by combining analyses on the overall event statistics and visual assessment of critical events identified. In total, the tracking algorithm results in less than 2% non-captured events for the current application. In the entire simulation, most of the structures are of spherical shape ($\{\hat{S}, \hat{C}, \lambda\} \approx \{1, 1, 1\}$). More stretched structures move towards lower shape index, indicating the predominance of elongated, cylindrical structures (dominated by $S \approx 1/2$). The absence of sheet-like (low $\hat{C}$) or highly stretched (low λ) structures leads to hypothesize that surface tension prevents such deformations, either by restoring the geometry towards more blob-like structures, or by leading to break-ups that form smaller spherical elements.

The analysis of split and merge events focuses on the size distribution and the geometrical change over the event. In merge events, the size ratio and geometrical change lead to the conclusion that flow homogeneity (enforced by periodic boundaries in the simulation) is the main cause for such events. Whenever structures increase in size due to merge events, they occupy more space in the domain, increasing the probability of being part of a merge event with smaller structures. This results in a predominance of structures with unequal size in coalescence events and a broad range of shapes (manifested by disparate locations in the geometrical feature space $\{\hat{S}, \hat{C}, \lambda\}$). Nevertheless, a visual inspection provides a first insight into the formation of merge events. If small structures are proximate to others (independently of their sizes), a small hyperboloidal bridge is formed in between, pulling the structures towards each other by surface tension. Such formations have already been observed experimentally and numerically in coalescence scenarios (e.g., Pergamalis [72] and literature within).

In split events, a mild bi-modality between sub- and super-Hinze scales is observed regarding the sizes of structures in the statistically stationary phase of the break-up process. Furthermore, the distribution of size ratios shows that, in split events, super-Hinze structures predominantly break up into structures with unequal size, where the largest remaining structure after split is of super-Hinze scale. Smaller, sub-Hinze scale structures tend to split into structures with equivalent sizes.

Regardless of the structure size, in break-up scenarios, a significant change of shape occurs. Before a structure splits, a cylindrical ligament is formed between two larger, more compact structures. In the geometrical feature space, those more highly stretched, tube-like structures can

be identified in regions of lower λ values around $\hat{S} \approx 1/2$. In the jpdf and marginal pdfs of S and C of such structures, the ligament formation can be observed by a longer tail of higher curviness values around $S \approx 1/2$. A detailed analysis of the jpdf could help identify and predict split events by tracking in time higher-order moments of the jpdf. Such elongated structures then split predominantly into spherical droplets whose geometric signature concentrates around $\{\hat{S}, \hat{C}, \lambda\} = \{1, 1, 1\}$ in the feature space, as a result of a transition from unstable towards more stable structures by surface tension effects.

The predominance of ligament-like structures before a break-up and the bi-modality of size ratios afterwards indicate the occurrence of cascade events, where larger, elongated super-Hinze structures break up successively into smaller ones. Analysis of a prototypical cascade, identified from the outcomes of the tracking algorithm, shows that such sequences of events mainly occur if long ligaments are present. A constriction in the region of contact with the more compact end structure initiates the split. In consequence, a low-frequency wave (Rayleigh instability [70]) evolves from this constriction and eventually causes the ligament to break up subsequently into several smaller spherical droplets. The side structures, originally connected by the ligament, are nearly unaffected in shape by this effect due to their compact form.

Beside the geometrical change of structures over an event, the evolution of structures without intermediate break-up or coalescence events is investigated. Due to the infinite series of droplets in the periodic domain, no clear conclusion can be drawn for structures that finally merge with others. Before split events, most of the structures only lived for a short period of time. The short lifetime of super-Hinze scale structures validates the Hinze scale as a critical diameter in break-up scenarios. Nevertheless, a reasonable number of super-Hinze structures is also found to have a longer lifetime (up to 30% of the total simulation period or approximately 3.35 times the eddy turnover time, $\Delta t \approx 3.35\tau_L$). In those cases, pulsations in alternating spatial directions keep the structures geometrically in stable conditions. Once those stable conditions are left, the split occurs after a few tracking steps. Currently, it is unclear what triggers the transition from stable to unstable conditions.

To better understand under which conditions such super-Hinze structures can persist, in future work, the study of the interaction of the

fluid interface with the background flow field is needed. The proposed tracking algorithm could be used not only to track the interface but also the surrounding turbulent structures, identifying mutual interactions. A more detailed analysis of the cascade process could be obtained by using a modal analysis of the evolving wave. The use of the $jpdf$ between S and C could also help predict the ligament formation that initiates cascade events. Furthermore, the dimensional curvedness can help identify critical sizes of such ligaments.

The possibility to use the tracking methodology on any scalar field with the same underlying physical assumptions allows a consistent analysis for different applications. Moreover, simple modifications and new implementations of constraints, branch decisions or tolerances provide a good opportunity to use the proposed tracking methodology in different fields.

CRedit authorship contribution statement

A. Bußmann: Methodology, Software, Formal analysis, Validation, Writing—original draft. **J. Buchmeier:** Methodology, Software. **M.S. Dodd:** Investigation, Data curation. **S. Adami:** Writing—review & editing. **I. Bermejo-Moreno:** Conceptualization, Supervision, Project administration, Software, 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

Data will be made available on request.

Acknowledgments

Computational resources for the work described in this paper were provided by the University of Southern California's Center for Advanced Research Computing. Partial financial support for I. Bermejo-Moreno and J. Buchmeier was provided by the Army Research Office, United States (Grant W911NF2010096, Program Manager M. Munson) and is gratefully acknowledged. This research was performed at the University of Southern California during A. Bußmann's stay as a visiting researcher. A. Bußmann was supported by German Academic Exchange Service (DAAD), Walther-Blohm Foundation and Prof. Dr.-Ing. Erich Müller Foundation.

References

- [1] Silver D, Wang X. Volume tracking. In: Proceedings of the 7th conference on visualization '96. IEEE Computer Society Press; 1996, p. 157–64.
- [2] Post FH, Vrolijk B, Hauser H, Laramée RS, Doleisch H. The state of the art in flow visualisation: Feature extraction and tracking. *Comput Graph Forum* 2003;22(4):775–92.
- [3] Muelder C, Ma K-L. Interactive feature extraction and tracking by utilizing region coherency. In: IEEE Pacific visualization symposium, Pacificvis 2009 - proceedings. 2009, p. 17–24.
- [4] Weigle C, Banks DC. Extracting iso-valued features in 4-dimensional scalar fields. In: Proceedings of the 1998 IEEE symposium on volume visualization. ACM; 1998, p. 103–10.
- [5] Ji G, Shen H-W, Wenger R. Volume tracking using higher dimensional isosurfacing. In: IEEE visualization. 2003, p. 209–16.
- [6] Bauer D, Peikert R. Vortex tracking in scale-space. In: Data visualisation 2002 (proc. joint eurographics - IEEE TCVG symposium on visualization), Vol. 2. 2002, p. 233–41.
- [7] Theisel H, Seidel H-P. Feature flow fields. In: Proceedings of the symposium on data visualisation 2003. 2003, p. 141–8.
- [8] Sauer F, Yu H, Ma K. Trajectory-based flow feature tracking in joint particle/volume datasets. *IEEE Trans Vis Comput Graphics* 2014;20(12):2565–74.
- [9] Bermejo-Moreno I, Pullin DI. On the non-local geometry of turbulence. *J Fluid Mech* 2008;603:101–35.
- [10] Leung T, Swaminathan N, Davidson PA. Geometry and interaction of structures in homogeneous isotropic turbulence. *J Fluid Mech* 2012;710:453–81.
- [11] Chakraborty P, Balachandar S, Adrian RJ. On the relationships between local vortex identification schemes. *J Fluid Mech* 2005;535:189–214.
- [12] Ashurst WT, Kerstein AR, Kerr RM, Gibson CH. Alignment of vorticity and scalar gradient with strain rate in simulated Navier-Stokes turbulence. *Phys Fluids* 1987;30(8):2343–53.
- [13] Hunt JCR, Wray AA, Moin P. Eddies, streams, and convergence zones in turbulent flows. *Center for Turbulence Research*; 1988, p. 193–208.
- [14] Joeng J, Hussain F. On the identification of a vortex. *J Fluid Mech* 1995;285:69–94.
- [15] Horiuti K. A classification method for vortex sheet and tube structures in turbulent flows. *Phys Fluids* 2001;13(12):3756–74.
- [16] Moisy F, Jiménez J. Geometry and clustering of intense structures in isotropic turbulence. *J Fluid Mech* 2004;513:111–33.
- [17] del Álamo JC, Jiménez J, Zandonade P, Moser RD. Self-similar vortex clusters in the turbulent logarithmic region. *J Fluid Mech* 2006;561:329–58.
- [18] Lu SS, Willmarth WW. Measurements of the structure of the Reynolds stress in a turbulent boundary layer. *J Fluid Mech* 1973;60:481–511.
- [19] Yang Y, Pullin DI, Bermejo-Moreno I. Multi-scale geometric analysis of Lagrangian structures in isotropic turbulence. *J Fluid Mech* 2010;654:233–70.
- [20] Yang Y, Pullin DI. On Lagrangian and vortex-surface fields for flows with Taylor-Green and Kida-Pelz initial conditions. *J Fluid Mech* 2010;661:446–81.
- [21] Bermejo-Moreno I, Pullin DI, Horiuti K. Geometry of enstrophy and dissipation, grid resolution effects and proximity issues in turbulence. *J Fluid Mech* 2009;620:121–66.
- [22] Hari Krishnan A, Anson C, Klein R, Vercauteren N. Geometry and organization of coherent structures in stably stratified atmospheric boundary layers. 2021, arXiv preprint arXiv:2110.02253.
- [23] Sahni V, Sathyaprakash BS, Shandarin SF. Shapefinder: A new shape diagnostic for large-scale structure. *Astrophys J* 1998;495:L5–8.
- [24] Clyne J, Mininni P, Norton A. Physically-based feature tracking for CFD data. *IEEE Trans Vis Comput Graph* 2013;19(6):1020–33.
- [25] Elsinga GE, Marusic I. Evolution and lifetimes of flow topology in a turbulent boundary layer. *Phys Fluids* 2010;22(1):01502.
- [26] Oster T, Abdelsamie A, Motejat M, Gerrits T, Rössl C, Thévenin D, et al. On-the-fly tracking of flame surfaces for the visual analysis of combustion processes. *Comput Graph Forum* 2018;37(6):358–69.
- [27] Kalivas DS, Sawchuk AA. A region matching motion estimation algorithm. *Image Underst* 1991;54(2):275–88.
- [28] Villaseñor J, Vincent A. An algorithm for space recognition and time tracking of vorticity tubes in turbulence. *CVGIP: Image Underst* 1992;55(1):27–35.
- [29] Elsinga GE, Poelma C, Westerweel J, Schröder A, Geisler R, Scarano F. Tracking of vortices in a turbulent boundary layer. *J Fluid Mech* 2012;697:273–95.
- [30] Silver D, Wang X. Tracking and visualizing turbulent 3D features. *IEEE Trans Vis Comput Graphics* 1997;3(2):129–41.
- [31] Lozano-Durán A, Jiménez J. Time-resolved evolution of the wall-bounded vorticity cascade. *J Phys Conf Ser* 2011;318(6):062016.
- [32] Lozano-Durán A, Jiménez J. Time-resolved evolution of coherent structures in turbulent channels: characterization of eddies and cascades. *J Fluid Mech* 2014;759:432–71.
- [33] Reinders F, Post FH, Spoelder HJW. Attribute-based feature tracking. In: Data visualization '99. Springer Vienna; 1999, p. 63–72.
- [34] Santaney R, Silver D, Zabusky N, Cao J. Visualizing features and tracking their evolution. *Computer* 1994;27:20–7.
- [35] Reinders F, Post FH, Spoelder HJW. Visualization of time-dependent data with feature tracking and event detection. *Vis Comput* 2001;17(1):55–71.
- [36] Chan WHR, Dodd MS, Johnson PL, Urzay J, Moin P. Formation and dynamics of bubbles in breaking waves: Part I. Algorithms for the identification of bubbles and breakup/coalescence events. *Center for Turbulence Research, Center for Turbulence Research, Stanford University*; 2018, p. 3–20.
- [37] Chan WHR, Dodd MS, Johnson PL, Urzay J, Moin P. Formation and dynamics of bubbles in breaking waves: Part II. The evolution of the bubble size distribution and breakup/coalescence statistics. *Center for Turbulence Research, Stanford University*; 2018, p. 21–34.
- [38] Chan WHR, Dodd MS, Johnson PL, Moin P. Identifying and tracking bubbles and drops in simulations: A toolbox for obtaining sizes, lineages, and breakup and coalescence statistics. *J Comput Phys* 2021;432:110156. <http://dx.doi.org/10.1016/j.jcp.2021.110156>.
- [39] Buchmeier J, Bußmann A, Gao X, Bermejo-Moreno I. Geometry and dynamics of passive scalar structures in compressible turbulent mixing. *Phys Fluids* 2021;33(10):105126.
- [40] He L, Ren X, Gao Q, Zhao X, Yao B, Chao Y. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognit* 2017;70:25–43. <http://dx.doi.org/10.1016/j.patcog.2017.04.018>.
- [41] Hendrickson K, Weymouth GD, Yue DK-P. Informed component label algorithm for robust identification of connected components with volume-of-fluid method. *Comput & Fluids* 2020;197:104373. <http://dx.doi.org/10.1016/j.compfluid.2019.104373>.

- [42] Cabaret L, Lacassagne L, Etienne D. Parallel light speed labeling: an efficient connected component algorithm for labeling and analysis on multi-core processors. *J Real-Time Image Process* 2016;15(1):173–96. <http://dx.doi.org/10.1007/s11554-016-0574-2>.
- [43] Herrmann M. A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure. *J Comput Phys* 2010;229(3):745–59. <http://dx.doi.org/10.1016/j.jcp.2009.10.009>.
- [44] Guo H, Phillips CL, Peterka T, Karpeyev D, Glatz A. Extracting, tracking, and visualizing magnetic flux vortices in 3D complex-valued superconductor simulation data. *IEEE Trans Vis Comput Graph* 2016;22:827–36. <http://dx.doi.org/10.1109/tvcg.2015.2466838>.
- [45] Schröder W, Martin K, Lorensen B. The visualization toolkit. fourth ed.. Kitware Inc.; 2006.
- [46] Koendrick JJ, van Doorn AJ. Surface shape and curvature scales. *Image Vis Comput* 1992;10:557–64.
- [47] Alyassin AM, Lancaster JL, Hunter Downs III J, Fox PT. Evaluation of new algorithms for the interactive measurement of surface area and volume. *Med Phys* 1994;21(6):741–52.
- [48] Blanco JL, Rai PK. Nanoflann: a C++ header-only fork of FLANN, a library for nearest neighbor (NN) with KD-trees. 2014, <https://github.com/jlblancoc/nanoflann>.
- [49] Bermejo-Moreno I. On the non-local geometry of turbulence (Ph.D. thesis), California Institute of Technology; 2008.
- [50] Ganser ER, North SC. An open graph visualization system and its applications to software engineering. *Softw - Prac Exp* 1999;1–29.
- [51] Laney D, Bremer P-T, Mascarenhas A, Miller P, Pascucci V. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Trans Vis Comput Graph* 2006;12(5):1053–60.
- [52] Dodd MS, Ferrante A. A fast pressure-correction method for incompressible two-fluid flows. *J Comput Phys* 2014;273:416–34.
- [53] Dodd MS, Ferrante A. On the interaction of Taylor length scale size droplets and isotropic turbulence. *J Fluid Mech* 2016;806:356–412.
- [54] Dodd MS. Direct numerical simulation of droplet-laden isotropic turbulence (Ph.D. thesis), University of Washington; 2017.
- [55] Bassenne M, Urzay J, Park GI, Moin P. Constant-energetics physical-space forcing methods for improved convergence to homogeneous-isotropic turbulence with application to particle-laden flows. *Phys Fluids* 2016;28(035114).
- [56] Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *J Comput Phys* 1992;100:335–54.
- [57] Weymouth GD, Yue DK-P. Conservative volume-of-fluid method for free-surface simulations on cartesian grids. *J Comput Phys* 2010;229:2853–65.
- [58] Aulisa E, Manservigi S, Scardovelli R, Zaleski S. Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry. *J Comput Phys* 2007;225(2):2301–19.
- [59] Cummins SJ, Francois MM, Kothe DB. Estimating curvature from volume fractions. *Comput Struct* 2005;83:425–34.
- [60] López J, Hernández J. On reducing interface curvature computation errors in the height function technique. *J Comput Phys* 2010;229:4855–68.
- [61] Ferrante A, Elgobashi S. On the physical mechanisms of two-way coupling in particle-laden isotropic turbulence. *Phys Fluids* 2003;15(2):315–29.
- [62] Schumann U. Realizability of Reynolds-stress turbulence models. *Phys Fluids* 1977;20(5):721–5.
- [63] Perlekar P, Biferale L, Sbragaglia M, Srivastava S, Toschi F. Droplet size distribution in homogeneous isotropic turbulence. *Phys Fluids* 2012;24(065101).
- [64] Lundgren TS. Linearly forced isotropic turbulence. *Annu Res Briefs, Center Turbul Res* 2003;461–73.
- [65] Rosales C, Meneveau C. Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties. *Phys Fluids* 2005;17(9):095106.
- [66] Rivière A, Mostert W, Perrard S, Deike L. Sub-hinze scale bubble production in turbulent bubble break-up. *J Fluid Mech* 2021;917.
- [67] Hinze JO. Fundamentals of the hydrodynamic mechanism of splitting in dispersion processes. *AIChE J* 1955;1(3):289–95.
- [68] Taubin G, Zhang T, Golub G. Optimal surface smoothing as filter design. In: *Computer vision — ECCV '96*. Springer Berlin Heidelberg; 1996.
- [69] Martínez-Bazán C, Montañez JL, Lasheras JC. On the breakup of an air bubble injected into a fully developed turbulent flow. Part 2. Size PDF of the resulting daughter bubbles. *J Fluid Mech* 1999;401:183–207. <http://dx.doi.org/10.1017/s0022112099006692>.
- [70] Lord Rayleigh F. On the instability of jets. *Proc Lond Math Soc* 1878;s1-10(1):4–13. <http://dx.doi.org/10.1112/plms/s1-10.1.4>.
- [71] Garrett C, Li M, Farmer D. The connection between bubble size spectra and energy dissipation rates in the upper ocean. *J Phys Oceanogr* 2000;30(9):2163–71.
- [72] Pergamalis H. Droplet impingement onto quiescent and moving liquid surfaces (Ph.D. thesis), Imperial College London; 2002.