

## MIT Open Access Articles

*An Eulerian label advection method for conservative volume-based tracking of bubbles/droplets*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Gaylo, Declan B., Hendrickson, Kelli and Yue, Dick K.P. 2022. "An Eulerian label advection method for conservative volume-based tracking of bubbles/droplets." *Journal of Computational Physics*, 470.

**As Published:** 10.1016/j.jcp.2022.111560

**Publisher:** Elsevier BV

**Persistent URL:** <https://hdl.handle.net/1721.1/155738>

**Version:** Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

**Terms of use:** Creative Commons Attribution-Noncommercial-ShareAlike



# An Eulerian label advection method for conservative volume-based tracking of bubbles/droplets

Declan B. Gaylo<sup>a</sup>, Kelli Hendrickson<sup>a</sup> and Dick K.P. Yue<sup>a,\*</sup>

<sup>a</sup>*Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

---

## ARTICLE INFO

### Keywords:

Two-phase flow  
Volume-of-fluid  
Eulerian tracking  
Fragmentation  
Coalescence  
Structure tracking

## ABSTRACT

We develop a robust volume-conservative framework for tracking blob evolution in complex two-phase flow that accurately and uniquely obtains the volume transfer among bubbles/droplets (blobs). This new framework is built on a volume-tracking matrix (VTM) that quantifies the volume transfer between any two blobs in two separated instances (snapshots) during the evolution, and an efficient Eulerian label advection (ELA) algorithm that explicitly provides the unique, consistent, volume-conservative VTM. Given a set of blobs defined at a snapshot by, say, a connected-component labeling (CCL) method and the grid-level volume-fraction flux from the conservative Volume of Fluid (cVOF) method [1], ELA gives the VTM by solving the Eulerian flow of each blob's fluid through time. Due to its grid-level Eulerian nature, ELA is independent of the complexity of the blob-level evolution, including high-arity (tertiary, quaternary, etc.) events and cycles which prevent previous methods from obtaining the VTM. We prove theoretically that ELA is volume-conservative to machine precision, with the same Courant restriction as cVOF. Furthermore, we show that, by allowing a diffusive error, multiplying the VTM obtains volume-conservative tracking over longer intervals without increasing the computational cost of ELA. We verify all these results using extensive simulations of evolving blob populations in flows with prescribed velocity and isotropic homogeneous turbulence (IHT).

---


## 1. Introduction

Bubbles and droplets, hereafter both referred to generally as blobs, in two-phase flows are important in a wide variety of natural and engineering applications. Examples include blobs formed by breaking waves [2, 3, 4], ship wakes [5], ocean spray [6], and combustion [7]. For all these applications, a key parameter of interest is the size distribution of the blobs and the transfer of volume between sizes. To develop accurate predictions of blob sizes and the transfer of volume, a detailed understanding of the mechanisms that evolve the blob population is necessary. To study these evolution mechanisms in experiments or simulations, accurate quantification of evolution events is required; however, even in simulations where all resolved properties of the flow are available, there remains a challenge in relating the available grid-level Eulerian description of the flow to the Lagrangian evolution of blobs. While a blob is defined as a volume enclosed by a continuous material interface, numerical methods, such as volume of fluid or level set, do not explicitly define this interface, making blob tracking a challenge despite the availability of information compared to experiments. This challenge has two separate parts: identification of blobs at an individual instant in time (snapshot) based on grid-level descriptions and tracking how these identified blobs evolve between two adjacent snapshots. The first part is, more broadly, the process of connected-component labeling (CCL) [8]. Recent CCL methods have been proposed with connectivity criteria designed specifically for two-phase fluid simulations [9, 10, 11]. In this paper, to avoid entangling these two parts, we take the blob identification provided by CCL as a given and address the second part of the challenge.

Previously, two Lagrangian blob-tracking methods for numerical simulations have been proposed [11, 12]. Their inputs from the simulation are Lagrangian integral quantities of blobs, e.g., volume, centroid, and total momentum. From such information about a population of blobs at two successive snapshots, the methods seek to find a possible evolutionary path between the two populations, either by physical and numerical constraints [11] or by minimizing a prescribed error function [12]. A limitation of both these methods is that they assume that all events are binary, meaning events involve at most two blobs from one snapshot and one blob from the other snapshot. Applied to entrainment of bubbles by a free surface, the Lagrangian methods have had some success describing fragmentation away from the

---

\*Corresponding author: Tel.: +1-617-253-6823;

 yue@mit.edu (D.K.P. Yue)

ORCID(s): 0000-0001-6198-7003 (D.B. Gaylo); 0000-0002-3596-6556 (K. Hendrickson); 0000-0003-1273-9964 (D.K.P. Yue)

free surface [12, 13]. In a region away from an air-entraining free surface, the flow can be approximated as isotropic homogeneous turbulence (IHT) [14], where the majority of fragmentation events are indeed binary [15, 16]. However, when near an entraining free surface, non-binary events are important. In the case of plunging breaking waves, large, roughly cylindrical, air cavities are often observed [17]. Gao et al. [18] show that these unstable structures rapidly undergo high-arity fragmentation, which Lagrangian methods are unable to track [11, 12]. In general, Lagrangian methods are unable to accurately describe the high-arity evolution of large complex air structures near the free surface. Application of the binary assumption to non-binary events introduces erroneous creation/extinction events [11]. Near the free surface, such error is inseparable from entrainment and degassing statistics. While high-arity events are theoretically avoidable by choosing a sufficiently small time interval  $\Delta t_s$  between adjacent snapshots, the effects of resolution limitations on CCL methods mean that small  $\Delta t_s$  will create erroneous fragmentation and coalescence events [11, 12]. Although a Lagrangian method could theoretically be extended to capture higher-arity events, the complexity would increase significantly [12].

A different approach to blob tracking is an Eulerian approach. Unlike Lagrangian methods, the available velocity field  $\mathbf{u}(\mathbf{x}, t)$  is used to determine the evolution of blobs. Because an Eulerian approach focuses on grid-level detail to describe blob evolution, the complexity of the formulation can be independent of the arity of the blob-level events. A few methods are available which leverage the velocity field [19, 20]; however, like Lagrangian methods, they attempt to describe the evolution of blobs in terms of individual Lagrangian events. We will refer to these as event-based Eulerian descriptions. We will prove that, while such a description can provide *which* blobs contribute volume to a blob at a later time, such a description is insufficient to ascertain *how much* volume is contributed by each blob to a blob at a later time.

In this work we provide a new volume-based Eulerian tracking framework capable of describing any blob evolution. Unlike previous event-based methods, we describe the evolution of the blob population by first providing a complete description for the movement of volume between blobs. This Eulerian volume-based tracking approach uniquely describes the evolution of the blob population through a volume-tracking matrix (VTM), which can describe evolution regardless of the complexity. From this more general description of blob evolution, individual events can be extracted. To uniquely provide the VTM, we build upon the volume-fraction fluxes provided by the conservative Volume of Fluid method (cVOF) [1] to create the Eulerian Label Advection method (ELA), a volume-conservative numerical implementation of volume-based tracking. By leveraging fluxes already calculated by cVOF, ELA minimizes additional computational cost and the only overhead associated is the memory cost, which scales with  $\Delta t_s$ . Using the VTM, we show that long snapshot intervals can be approximated from shorter, less memory-intensive snapshot intervals. These results are validated using simple prescribed velocity simulation and the canonical problem of bubbles fragmenting in IHT.

## 2. Eulerian volume-based tracking

In immiscible two-phase flow, the composition of fluid at a given location in space  $\mathbf{x}$  at time  $t$  can be described by a fluid color function defined at any time by

$$c(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \text{'dark' fluid} \\ 0 & \text{if } \mathbf{x} \in \text{'light' fluid} \end{cases} \quad (1)$$

Without loss of generality, let the dark fluid make up the blobs of interest. For incompressible flows, the evolution of  $c$  must satisfy

$$\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c = 0. \quad (2)$$

We now extend this color function description to incorporate the information given by a CCL method [8]. We assume that, from an approximation of  $c$  at time  $t^n$ , a CCL method provides a set of non-overlapping blobs  $\mathcal{B}^n = \{1 \dots M^n\}$ , where  $M^n$  is the number of blobs, and labels the dark volume within with a corresponding label  $l \in 0 \dots M^n$ . Here  $l = 0$  is reserved for dark fluid not identified as being part of a blob by the chosen CCL method. From these labels, we define a vector color function  $c^n(\mathbf{x}, t)$  with elements initially defined at time  $t^n$  by

$$c_l^n(\mathbf{x}, t^n) = \begin{cases} 1 & \text{if } \mathbf{x} \in \text{blob } l \\ 0 & \text{else} \end{cases} \quad \text{for } l \in 0 \dots M^n. \quad (3)$$

Non-periodic boundary conditions can be implemented by adding a blob (or blobs) to  $\mathcal{B}^n$  and setting the corresponding element(s) of  $\mathbf{c}^n$  at the boundaries. Equivalent to (2), the evolution of  $\mathbf{c}^n$  must satisfy

$$\frac{\partial \mathbf{c}^n}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{c}^n = \mathbf{0}. \quad (4)$$

For a dark fluid particle located at any  $\mathbf{x}$  at any time  $t$ ,  $\mathbf{c}^n(\mathbf{x}, t)$  provides the blob  $l \in \mathcal{B}^n$  that contained the particle at time  $t^n$ . Thus,  $\mathbf{c}^n$  provides a complete description of the flow of dark fluid.

For  $\mathbf{c}^n$  to also provide an accurate description of the flow of dark fluid, the advection of  $c$  and  $\mathbf{c}^n$  must be identical, i.e., the velocity  $\mathbf{u}$  must be the same in (2) and (4). From this, we develop two core requirements that a numerical implementation of (4) must satisfy to provide accurate tracking of dark fluid. As blobs are defined to be non-overlapping at  $t^n$ , the following consistency requirement must be true at all times and locations:

$$\sum_{l=0}^{M^n} c_l^n(\mathbf{x}, t) = c(\mathbf{x}, t). \quad (5)$$

Based on incompressibility, (2) requires that  $c$  is conserved, thus each component  $c_l^n$  of  $\mathbf{c}^n$  must also be conserved, the second core requirement. If the advection of  $c$  is not strictly conservative, a weaker form of this volume-conservation requirement is necessary: that the erroneous addition/loss of  $c$  must be reflected in an addition/loss in  $\sum_l c_l$ .

We note in passing that while the color function based formulation we have used here lends itself well to a VOF-based numerical implementation, (4) could be implemented using a variety of two-phase advection methods. For example, splitting the color function into a vector color function and applying (4) has been used to address numerical coalescence using a variety of two-phase methods [21, 22, 23]. For this section we assume that the solution  $\mathbf{c}^n$  which satisfies (4) is known, as Eulerian volume-based tracking could be performed by any consistent and volume conservative numerical implementation of (4).

## 2.1. A volume-tracking matrix (VTM) description of blob evolution

Integrating (4) over time, the vector color function  $\mathbf{c}^n(\mathbf{x}, t)$  originally defined by (3) at time  $t^n$ , i.e.,  $\mathbf{c}^n(\mathbf{x}, t^n)$ , can be advanced in time to the next snapshot  $t^{n+1} = t^n + \Delta t_s$  to give  $\mathbf{c}^n(\mathbf{x}, t^{n+1})$ . For a dark fluid particle located at  $\mathbf{x}$  at time  $t^{n+1}$ ,  $\mathbf{c}^n(\mathbf{x}, t^{n+1})$  provides the blob  $l \in \mathcal{B}^n$  that contained the particle at time  $t^n$ . At  $t^{n+1}$ , we assume CCL provides a new set  $\mathcal{B}^{n+1} = \{1 \dots M^{n+1}\}$  of blobs, from which we use (3) again to define a new vector color function  $\mathbf{c}^{n+1}(\mathbf{x}, t)$ , with the initial value defined at  $t^{n+1}$ . Based on  $\mathbf{c}^n$  and  $\mathbf{c}^{n+1}$ , both available at time  $t^{n+1}$ , the volume of dark fluid from a blob  $l \in \mathcal{B}^n$  that ends up in a blob  $m \in \mathcal{B}^{n+1}$  is

$$q_{ml} = \int_{\forall} c_m^{n+1}(\mathbf{x}, t^{n+1}) c_l^n(\mathbf{x}, t^{n+1}) dV, \quad (6)$$

where  $\forall$  is the whole domain. Applying (3) for  $c_m^{n+1}(\mathbf{x}, t^{n+1})$  gives

$$q_{ml} = \int_{\mathbf{x} \in \text{blob } m} c_l^n(\mathbf{x}, t^{n+1}) dV. \quad (7)$$

For  $l = 0 \dots M^n$  and  $m = 0 \dots M^{n+1}$ , we define the matrix  $\mathbf{Q}^{(n \rightarrow n+1)} = \{q_{ml}\}$ , which provides a complete description of the flow of dark fluid from blobs  $\mathcal{B}^n$  to blobs  $\mathcal{B}^{n+1}$ . Each element  $q_{ml}$  in the matrix  $\mathbf{Q}^{(n \rightarrow n+1)}$  provides the (absolute) volume of dark fluid that transfers from blob  $l$  to blob  $m$  over the interval  $t^n$  to  $t^{n+1}$ .

We note that based on  $\mathbf{c}^n$  we can express the volume of all the blobs at time  $t^n$  as a vector  $\mathbf{v}^n$  of length  $M^n$ ,

$$\mathbf{v}^n = \int_{\forall} \mathbf{c}^n(\mathbf{x}, t) dV. \quad (8)$$

Recalling (5), the column and row sums of  $\mathbf{Q}^{(n \rightarrow n+1)}$  give the blob volumes at  $t^n$  and  $t^{n+1}$  respectively:

$$\sum_m q_{ml} = v_l^n, \quad (9a)$$

$$\sum_l q_{ml} = v_m^{n+1}. \quad (9b)$$

Normalizing the columns of  $\mathbf{Q}^{(n \rightarrow n+1)}$  by  $\mathbf{v}^n$ ,

$$a_{ml} = q_{ml} / v_l^n, \quad (10)$$

we define the volume-tracking matrix (VTM),  $\mathbf{A}^{(n \rightarrow n+1)} = \{a_{ml}\}$ . The VTM is a left stochastic matrix,

$$\sum_m a_{ml} = 1, \quad (11)$$

which describes the evolution of volume from blobs  $\mathcal{B}^n$  to blobs  $\mathcal{B}^{n+1}$  as

$$\mathbf{v}^{n+1} = \mathbf{A}^{(n \rightarrow n+1)} \mathbf{v}^n. \quad (12)$$

We note that if instead the rows of  $\mathbf{Q}^{(n \rightarrow n+1)}$  are normalized by  $\mathbf{v}^{n+1}$ , one gets a functionally equivalent right stochastic matrix description of the reverse evolution.

Because the VTM provides only a summary of  $\mathbf{c}^n(\mathbf{x}, t^{n+1})$ , on the particle level, each entry  $a_{ml}$  of the VTM can be interpreted as the probability a particle of dark fluid is in blob  $m \in \mathcal{B}^{n+1}$  at  $t^{n+1}$  given that it was in blob  $l \in \mathcal{B}^n$  at  $t^n$ . However, on the blob level, each entry  $a_{ml}$  provides the (deterministic) proportion of the volume of blob  $l \in \mathcal{B}^n$  that ends up in blob  $m \in \mathcal{B}^{n+1}$ . Thus, the VTM  $\mathbf{A}^{(n \rightarrow n+1)}$  provides a complete Lagrangian description of the evolution of blobs. From the VTM, individual events of arbitrary arity are available. For example, fragmentation of one parent blob into  $d$  child blobs within a larger simulation will create a subset of  $\mathbf{A}^{(n \rightarrow n+1)}$  that has  $d$  non-zero entries in a single column, and coalescence of  $d$  blobs into one will create a subset that has  $d$  entries of 1 in a single row.

## 2.2. Limitations of existing blob-tracking approaches providing the VTM

We first consider how the VTM could be obtained from information provided by existing methods. Lagrangian methods of blob tracking [11, 12] seek to identify a solution to which blobs at  $t^n$  contributed to a blob at  $t^{n+1}$ . In terms of our VTM description, they seek to identify which entries of  $\mathbf{Q}^{(n \rightarrow n+1)}$  are non-zero. Similarly, event-based Eulerian methods [19, 20] directly provide which entries are non-zero, but not the values. Based on conservation, (9) provides an equation to which an implicit solution for the values of  $\mathbf{Q}^{(n \rightarrow n+1)}$  could be sought given the knowledge of which entries of  $\mathbf{Q}^{(n \rightarrow n+1)}$  are non-zero, as provided by previous tracking methods. However, (9) will not always provide a unique solution. Therefore, the unique VTM is not always obtainable using previous tracking methods.

For example, consider the case where two large blobs of volume  $v_1^n = v_2^n$  exchange two smaller blobs of volume  $0.05v_1^n$  each over a time  $T$ , as shown in figure 1. For this given flow, we examine the performance of tracking methods if only snapshots at time  $t^n = 0$  and  $t^{n+1} = \Delta t_s > T$ , are available. Two blobs will be identified at  $t^n$  and two blobs identified at  $t^{n+1}$ , representing a  $2 \times 2$  tracking matrix with four non-zero entries. The resulting set of (column normalized) equations from (9),

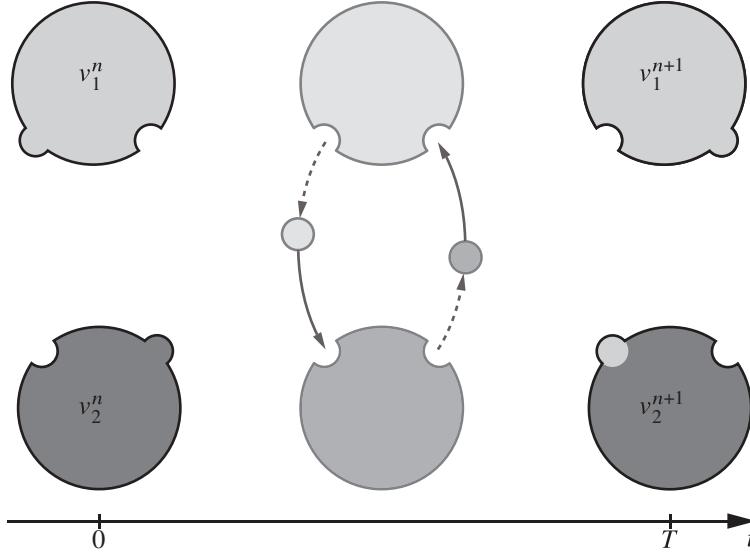
$$\begin{Bmatrix} v_1^{n+1} \\ v_2^{n+1} \end{Bmatrix} = \begin{bmatrix} a_{11} & (1 - a_{22}) \\ (1 - a_{11}) & a_{22} \end{bmatrix} \begin{Bmatrix} v_1^n \\ v_2^n \end{Bmatrix}, \quad (13)$$

does not have a unique solution. We note that this problem is equivalent to solving for the currents in a graph given the net current into each node. Here, the nodes are  $\{\mathcal{B}^n, \mathcal{B}^{n+1}\}$ , the edges are the non-zero entries of  $\mathbf{Q}^{(n \rightarrow n+1)}$ , and the currents are the respective element values. Any cycle in the graph allows a loop current, introducing a null space in the solution for the currents. It is possible that a Lagrangian method, where the non-zeros are not explicitly known, may pick the solution  $a_{11} = a_{22} = 1$  (or 0) suggesting no cycles, but there is still a physical exchange of volume and the uniqueness problem remains. In §4.1 we show that a test case similar to figure 1 generates such cycles for  $\Delta t_s > T$ . While choosing a small enough  $\Delta t_s$  could theoretically eliminate this issue, in §4.2.2 we show that for a realistic  $\Delta t_s$ , cycle generation is common.

Figure 1 illustrates that by splitting the dark fluid at  $t^n$  using the vector color function  $\mathbf{c}^n$  and then advecting it with (4), the values for the flow of volume ( $\mathbf{Q}^{(n \rightarrow n+1)}$ ) are explicitly available at time  $t^{n+1}$  through (6). Given connectedness provided by CCL, Eulerian volume-based tracking, unlike previous methods, provides the unique and correct solution to the VTM. For this example:

$$\begin{Bmatrix} v_1^{n+1} \\ v_2^{n+1} \end{Bmatrix} = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \begin{Bmatrix} v_1^n \\ v_2^n \end{Bmatrix}. \quad (14)$$

As volume-based Eulerian tracking provides the non-zero entries in the VTM directly from the underlying advection of dark fluid, all resulting cycles represent real exchange of volume between the blobs identified by CCL.



**Figure 1:** Illustration of two blobs of equal volume exchanging 5% of their volume over a time  $T$  less than the snapshot interval,  $\Delta t_s$ . This creates a  $2 \times 2$  VTM with 4 non-zero entries, meaning a unique solution to the values of the non-zero entries is not obtainable from conservation laws. By marking the volume of the blobs with a split color function, illustrated here using green and blue, the values of the non-zero entries are available explicitly.

### 3. Eulerian label advection (ELA)

We describe here the Eulerian label advection (ELA) method which is a volume-conservative numerical advection scheme for the vector color function in (4) which allows direct calculation of the VTM  $\mathbf{A}^{(n \rightarrow n+1)}$ . The numerical representation of the vector color function is based on the Volume-of-Fluid (VOF) representation of the scalar color function. The advection scheme itself is based on and consistent with the conservative Volume-of-Fluid (cVOF) method of Weymouth and Yue [1], while ELA maintains the volume-conservative nature of that approach.

#### 3.1. Defining the vector source fraction field

For a three-dimensional domain size  $N_i \times N_j \times N_k$ , let  $\Omega_{ijk}$  be the region of each cell with volume  $\Delta\Omega_{ijk} = \int_{\Omega_{ijk}} dV$ . The volume fraction field  $f_{ijk}$  is defined as

$$f_{ijk}(t) = \frac{\int_{\Omega_{ijk}} c(\mathbf{x}, t) dV}{\Delta\Omega_{ijk}}. \quad (15)$$

A vector source fraction field  $\mathbf{s}$  is defined analogously using the vector color function:

$$\mathbf{s}_{ijk}^n(t) = \frac{\int_{\Omega_{ijk}} \mathbf{c}^n(\mathbf{x}, t) dV}{\Delta\Omega_{ijk}}. \quad (16)$$

Therefore, given the results of a CCL method run at time  $t^n$ ,  $\mathbf{s}_{ijk}^n(t^n)$  describes which blobs  $l \in \mathcal{B}^n$  make up the dark fluid in the cell, including (unlike previous Eulerian methods) the relative volume contribution of each. Typical CCL algorithms assign all of the dark fluid in a cell to a single blob, so at  $t^n$ ,  $\mathbf{s}^n(t^n)$  can be initialized using

$$(s_l^n)_{ijk}(t^n) = \begin{cases} f_{ijk}(t^n) & \text{if } \Omega_{ijk} \in \text{blob } l \\ 0 & \text{otherwise} \end{cases} \quad \text{for } l \in 0 \dots M^n. \quad (17)$$

#### 3.2. Derivation of the ELA algorithm

Inspired by the approach of Weymouth and Yue [1], we seek a volume-conservative numerical advection scheme to model (4). For conciseness, we consider a single cell and drop the  $ijk$  subscript. First, we integrate (4) over the cell

to avoid the discontinuities in  $\mathbf{c}^n(\mathbf{x})$ , then integrate by parts and apply the divergence theorem to obtain

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{c}^n \, dV + \oint_{\partial\Omega} \mathbf{c}^n u_n \, dS = \int_{\Omega} \mathbf{c}^n \nabla \cdot \mathbf{u} \, dV. \quad (18)$$

Writing in terms of  $\mathbf{s}^n$  using (16) and rearranging,

$$\frac{\partial \mathbf{s}^n}{\partial t} \Delta\Omega = -\mathbf{F}_{net} + \int_{\Omega} \mathbf{c}^n \nabla \cdot \mathbf{u} \, dV, \quad (19)$$

where the vector  $\mathbf{F}_{net}$  describes the net flux of  $\mathbf{s}^n$  out of  $\Omega$ . As we discretize (19) in time, to avoid confusion of the snapshot index  $n$  based on  $\Delta t_s$  and the fluid-solver time index  $k$  based on  $\Delta t$ , we will omit  $n$  for the rest of §3.2 and §3.3, e.g.,  $\mathbf{s}^k = \mathbf{s}^n(t^k)$ .

Weymouth and Yue [1] provide an operator-split method to solve (2) using the volume fraction field  $f$  on Cartesian-grids which ensures volume conservation. To go from  $f^k$  to  $f^{k+1}$  at  $t^{k+1} = t^k + \Delta t$  in an  $\mathcal{N}$  dimensional domain,

$$\frac{\Delta\Omega}{\Delta t} (f^{(d)} - f^{(d-1)}) = F_{d+1/2} - F_{d-1/2} + \tilde{c} \frac{\partial u_d}{\partial x_d} \Delta\Omega \quad \text{for } d \in 1 \dots \mathcal{N}, \quad (20)$$

where  $f^{(0)} = f^k$  and  $f^{(\mathcal{N})} = f^{k+1}$ . Note that the superscript  $d$  references the operator split step and takes values 0 to  $\mathcal{N}$  and the subscript  $d$  in the dilation term indicates the vector component in the associated direction. For each direction, the scalar flux on the positive face ( $F_{d+1/2}$ ) and negative face ( $F_{d-1/2}$ ) are calculated using a second-order interface reconstruction based on  $f^{(d-1)}$ , which ensures the flux terms are conservative. Weymouth and Yue [1] approximate the dilation term using the cell center value of the color function at  $t^{(0)}$ ,

$$\tilde{c} = \begin{cases} 1 & \text{if } f^{(0)} > 1/2 \\ 0 & \text{otherwise} \end{cases}. \quad (21)$$

For volume conservation, it is critically important that  $\tilde{c}$  remains the same throughout the operator-split steps [1].

We now solve (19) using a similar operator-split equation,

$$\frac{\Delta\Omega}{\Delta t} (\mathbf{s}^{(d)} - \mathbf{s}^{(d-1)}) = \mathbf{F}_{d+1/2} - \mathbf{F}_{d-1/2} + \tilde{\mathbf{c}} \frac{\partial u_d}{\partial x_d} \Delta\Omega \quad \text{for } d \in 1 \dots \mathcal{N}, \quad (22)$$

where  $\mathbf{F}_{d+1/2}$  and  $\mathbf{F}_{d-1/2}$  are vector flux terms on the positive and negative faces and  $\tilde{\mathbf{c}}$  is a vector dilation term. As the consistency requirement states (5) is always true, we require

$$\sum_l (s_l)^{(d)} = f^{(d)}. \quad (23)$$

Summing (22) over its vector components and comparing to (20), (23) is always satisfied if  $\mathbf{F}$  and  $\tilde{\mathbf{c}}$  are defined such that they sum to the scalar equivalents, i.e.,  $\sum_l (F_l) = F$  and  $\sum_l (\tilde{c}_l) = \tilde{c}$ . For convenience, we define the normalized vector source fraction  $\hat{\mathbf{s}}$  as

$$\hat{s}_l = \frac{s_l}{\sum_i s_i} \quad \text{for } l \in 0 \dots M, \quad (24)$$

which has the property  $\sum_l (\hat{s}_l) = 1$ . An explicit conservative upwind scheme is used to determine the composition of the dark-fluid flux based on the previous operator-split step's  $\mathbf{s}^{(d-1)}$  and the scalar flux  $F$  from cVOF:

$$\mathbf{F}_{d+1/2} = F_{d+1/2} \cdot \begin{cases} \hat{\mathbf{s}}_d^{(d-1)} & \text{if } F_{d+1/2} > 0 \\ \hat{\mathbf{s}}_d^{(d+1)} & \text{if } F_{d+1/2} < 0 \end{cases}, \quad (25a)$$

$$\mathbf{F}_{d-1/2} = F_{d-1/2} \cdot \begin{cases} \hat{\mathbf{s}}_d^{(d-1)} & \text{if } F_{d-1/2} > 0 \\ \hat{\mathbf{s}}_{d-1}^{(d-1)} & \text{if } F_{d-1/2} < 0 \end{cases}. \quad (25b)$$

To describe the vector dilation term based on the scalar dilation term  $\tilde{c}$  from cVOF,

$$\tilde{\mathbf{c}} = \tilde{c} \hat{\mathbf{s}}^{(0)}. \quad (26)$$

For volume conservation, as well as now consistency, it is similarly critical that  $\tilde{\mathbf{c}}$  be based on the initial  $\mathbf{s}^{(0)}$  and remain the same throughout the operator-split steps.

For implementation of (22), the flux calculations can be performed such that  $\hat{\mathbf{s}}^{(d-1)}$  is calculated as needed one grid cell at a time as  $\mathbf{s}^{(d)}$  overwrites  $\mathbf{s}^{(d-1)}$ . The dilation  $\tilde{\mathbf{c}}$  must be stored throughout the operator-split steps, which comes with an additional memory requirement no larger than storing  $\mathbf{s}^{(0)}$ . We highlight that, from the scalar terms already required by cVOF in (20), only a normalization and a multiplication operation is needed to create the vector terms required by ELA in (22).

### 3.3. ELA volume conservation

The requirements for a volume-conservative (to machine precision) operator-split advection scheme are [1]:

1. flux terms are conservative,
2. the dilation terms sum to zero, and
3. there is no over or under filling.

The consistent extension of the scalar cVOF equation (20) to a vector ELA equation (22) using (25) and (26) maintains satisfaction of requirements 1 and 2.

For requirement 3, cVOF with the Courant restriction

$$C = \Delta t \sum_{d=1}^{\mathcal{N}} \left| \frac{u_d}{\Delta x_d} \right| \leq \frac{1}{2} \quad (27)$$

guarantees  $0 \leq f^{(d)} \leq 1$  [1]. Because (23) is always satisfied by (22), this Courant restriction guarantees

$$0 \leq \sum_l (s_l)^{(d)} \leq 1. \quad (28)$$

This only establishes that the sum of  $s^{(d)}$  cannot over or under fill. To guarantee the same for the individual components  $(s_l)^{(d)}$ , we must establish a Courant restriction for our vector advection equation, using the work of Weymouth and Yue [1] as a baseline. First, (28) shows that by proving  $0 \leq (s_l)^{(d)}$  for all  $l$ , we prove  $0 \leq (s_l)^{(d)} \leq 1$ , satisfying requirement 3. In appendix A, by considering all possible combinations of the sign of the velocity on either face, we prove that  $0 \leq (s_l)^{(d)}$  for all  $l$  provided the Courant restriction (27) is true. As this is the same Courant restriction needed by cVOF, no change in  $\Delta t$  is needed to use ELA with cVOF versus cVOF alone.

We note that after using a VOF advection scheme, it is sometimes useful to filter very small volume fractions of dark/light fluid, artifacts of precision limitations in the flux calculations [24]. This filter is expressed in terms of a chosen (very small) zero-threshold value  $\epsilon$  as

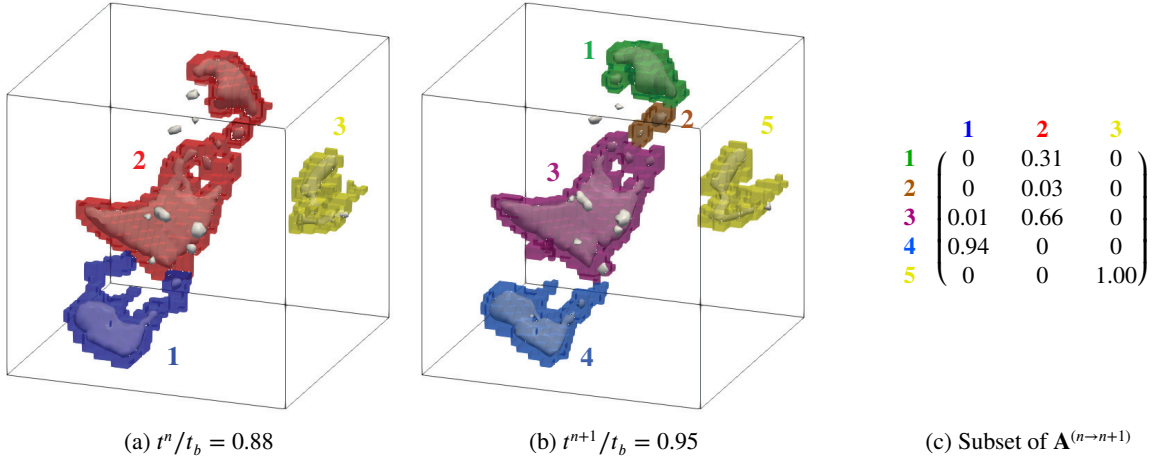
$$f'_{ijk} = \begin{cases} 1 & \text{if } 1 - f_{ijk} < \epsilon \\ 0 & \text{if } f_{ijk} < \epsilon \\ f_{ijk} & \text{otherwise} \end{cases}. \quad (29)$$

Here,  $\epsilon$  larger than machine precision ensures approximate symmetry between floating point operations on  $f$  versus  $1 - f$ . To ensure (23) is always true, as required for ELA volume conservation, if the filter (29) is used  $\mathbf{s}$  must be similarly adjusted by the same  $\epsilon$ :

$$\mathbf{s}'_{ijk} = \begin{cases} \hat{\mathbf{s}}_{ijk} & \text{if } 1 - f_{ijk} < \epsilon \\ \mathbf{0} & \text{if } f_{ijk} < \epsilon \\ \mathbf{s}_{ijk} & \text{otherwise} \end{cases}. \quad (30)$$

While neither VOF nor ELA are strictly conservative if  $\epsilon \neq 0$ , the total volume addition/loss in  $\mathbf{s}_{ijk}$  is equal to that in  $f_{ijk}$ . Therefore, by applying (30), ELA continues to satisfy the consistency requirement and satisfies the necessarily weakened volume-conservation requirement in the sense that it tracks all the dark fluid, including that artificially added/subtracted by (29).





**Figure 2:** ELA tracking from a simulation (see §4.2 for details). The  $f = 0.5$  iso-surface is shown and grid cells are highlighted corresponding to the label assigned by the CCL method. Note that an iso-surface is itself a CCL method, which does not necessarily align with the chosen method. 3 resolved parent bubbles (a) have their volume distributed among 5 resolved child bubbles (b). ELA gives the associated VTM (c).

### 3.4. Extracting the VTM

We start by rewriting (6) as a summation over all cells in the domain,

$$q_{ml} = \sum_{\Omega_{ijk} \in \forall} \int_{\Omega_{ijk}} c_m^{n+1}(\mathbf{x}, t^{n+1}) c_l^n(\mathbf{x}, t^{n+1}) dV. \quad (31)$$

As done for (17), we assume that a CCL algorithm assigns all of the dark fluid in a cell to a single blob, giving

$$q_{ml} = \sum_{\Omega_{ijk} \in \text{blob } m} \int_{\Omega_{ijk}} c_l^n(\mathbf{x}, t^{n+1}) dV. \quad (32)$$

Recall that  $n$  is the snapshot interval index, i.e.,  $t^{n+1} = t^n + \Delta t_s$ . Choosing an integer  $K$  based on the desired snapshot interval,  $\Delta t_s = K \Delta t$ , (22) is performed every simulation time step to advance  $\mathbf{s}^n(t^n)$  in time to  $\mathbf{s}^n(t^{n+1})$ , which provides

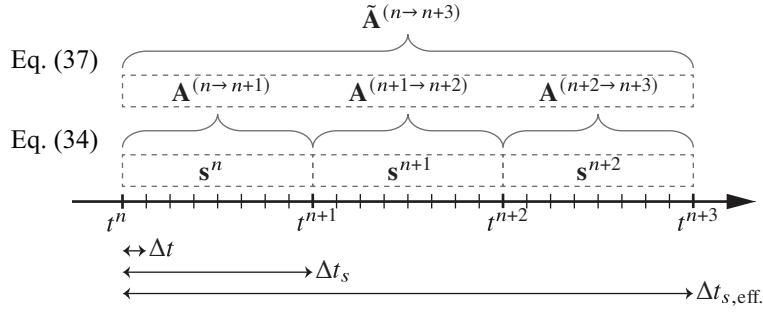
$$\mathbf{s}_{ijk}^n(t^{n+1}) = \frac{\int_{\Omega_{ijk}} \mathbf{c}^n(\mathbf{x}, t^{n+1}) dV}{\Delta \Omega_{ijk}}, \quad (33)$$

which gives

$$q_{ml} = \sum_{\Omega_{ijk} \in \text{blob } m} \Delta \Omega_{ijk} (s_l^n)_{ijk}(t^{n+1}). \quad (34)$$

After normalizing using (10), we obtain the VTM  $\mathbf{A}^{n \rightarrow n+1}$ .

Figure 2 shows an example from the simulation in §4.2 of a VTM extracted using ELA. Based on the first column of  $\mathbf{A}^{(n \rightarrow n+1)}$ , we see that the majority of the volume from the parent with label 1 at time  $t^n$  went to the child with label 4 at time  $t^{n+1}$ , apart from a small portion that went to child 3. For clarity, under-resolved bubbles have been excluded, causing the first column not sum to 1, cf. (11). Based on the second column, the volume from parent 2 went to three different children, 1, 2, and 3. Based on the third column, all of the volume from parent 3 went to child 5. The events described by this VTM are complex in two ways: there is a non-binary fragmentation event, and both fragmentation and coalescence occurred to form child 3.



**Figure 3:** Illustration of the flow of information using ELA and matrix multiplication with  $K = 8$  and  $N = 3$ . After using ELA over  $K$  simulation time steps, the information in  $\mathbf{s}^n$  is summarized in a VTM using (34). The information in  $N$  of these VTMs is summarized in an effective VTM using (37).

### 3.5. Approximating long snapshot intervals through matrix multiplication

While there is limited additional computational effort required for ELA (see §3.2), there is a memory cost related to storing the source vector  $\mathbf{s}^n$ . Here, we provide an upper bound for how this memory cost scales with  $K$ . Although in practice the cost is substantially less than this upper bound, the memory cost of ELA is still non-trivial. Should this cost be prohibitive, we present a matrix-multiplication method that employs small, inexpensive  $K$  to approximate a larger  $K$ . It is important to note that using matrix multiplication with ELA represents a trade-off between accuracy and cost, as matrix multiplication is only an approximation of Eulerian volume-based tracking.

To provide the upper bound on ELA memory cost, we begin with how  $\mathbf{s}^n$  is stored. The number of bubbles  $M^n$  is generally only known *a posteriori*, so the length of  $\mathbf{s}^n$  cannot be bounded *a priori*. Rather than attempt to store the entire length for each cell, we store the index and value of each non-zero entry of  $\mathbf{s}^n$  in a cell. The same approach applies to storing  $\tilde{\mathbf{c}}$ . Assuming the CCL method used will not generate more than one label per cell, and noting that the Courant restriction limits advection of  $\mathbf{s}^n$ , the maximum number of entries of  $\mathbf{s}^n$  needed to be stored to run ELA over a given snapshot interval  $\Delta t_s = K \Delta t$  is

$$\text{nnz}(\mathbf{s}_{ijk}^n) \leq \min \left\{ (1 + 2K)^{\mathcal{N}}, \quad M^n \right\}, \quad (35)$$

where  $\text{nnz}(\mathbf{s}_{ijk}^n)$  is the maximum number of non-zero entries in any  $\mathbf{s}_{ijk}^n$ . By (35), the smallest memory requirement for ELA comes with a small  $K$ ; however, the resulting small  $\Delta t_s$  introduces significant noise due to resolution limits inherent in CCL algorithms [11]. For simulations in §4.2,  $K \sim \mathcal{O}(100)$  when following published guidance on selecting  $\Delta t_s$  [11]. Although the upper bound suggests that memory requirements scale with  $K^3$  for a three-dimensional simulation, in §4.2.3 we will show that in practice the scaling is much lower than  $K^3$ .

To approximate a desired  $\Delta t_s$  with a smaller  $K$ , we introduce the matrix-multiplication method. Recalling that the VTM in (12) acts like a left stochastic matrix, we can write

$$\mathbf{v}^{n+N} = \left[ \prod_{m=n+N-1}^n \mathbf{A}^{(m \rightarrow m+1)} \right] \mathbf{v}^n, \quad (36)$$

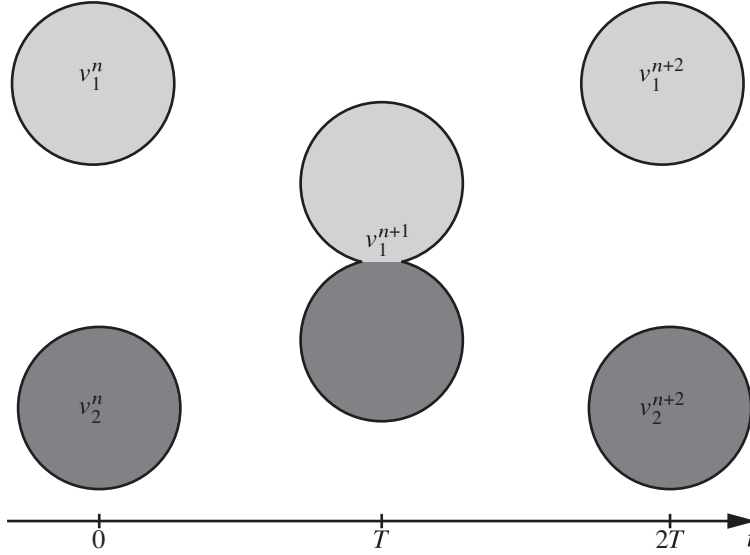
for  $N$  snapshot intervals. This defines an effective VTM

$$\tilde{\mathbf{A}}^{(n \rightarrow n+N)} = \prod_{m=n+N-1}^n \mathbf{A}^{(m \rightarrow m+1)}, \quad (37)$$

with an effective snapshot interval  $\Delta t_{s,\text{eff}} = N \Delta t_s$ . Therefore,

$$\Delta t_{s,\text{eff}} = N K \Delta t. \quad (38)$$

With  $\Delta t$  constrained by the flow solver, the effective snapshot interval can be adjusted through the true snapshot interval during the simulation ( $K$ ) or through multiplying the matrices in post-processing ( $N$ ), as illustrated in figure 3.



**Figure 4:** Illustration of two blobs at  $t = 0$  whose dark fluid briefly coalesces into one blob at  $t = T$  but then fragments along the boundary between the two original blobs such that at  $t = 2T$  all of the volume from one original blob is in only one of the final blobs.

Although the effective VTM describes the same transition from  $\mathbf{v}^n$  to  $\mathbf{v}^{n+N}$ , we note that there is a loss of information when using matrix multiplication versus a larger  $K$ . Through the vector color function  $\mathbf{c}^n$ , the source of each particle of air is known. However, when  $\mathbf{A}$  is calculated using (6), we integrate  $\mathbf{c}^n$  over the volume of each blob, losing the spatial distribution of  $\mathbf{c}^n$  within the blob. In effect, matrix multiplication assumes that after each true snapshot interval  $\Delta t_s$ ,  $\mathbf{c}^n$  is homogeneous within a blob, increasing the apparent entropy. This is consistent with the stochastic interpretation of the VTM: it provides a probability given only that a particle is within a blob, not the specific location of the particle within the blob. However, this implied diffusion is inconsistent with (4), which comes from the VTM providing less information than  $\mathbf{s}^n$ . A result of this loss of information is that effective VTMs depend on  $N$ , and are thus not unique.

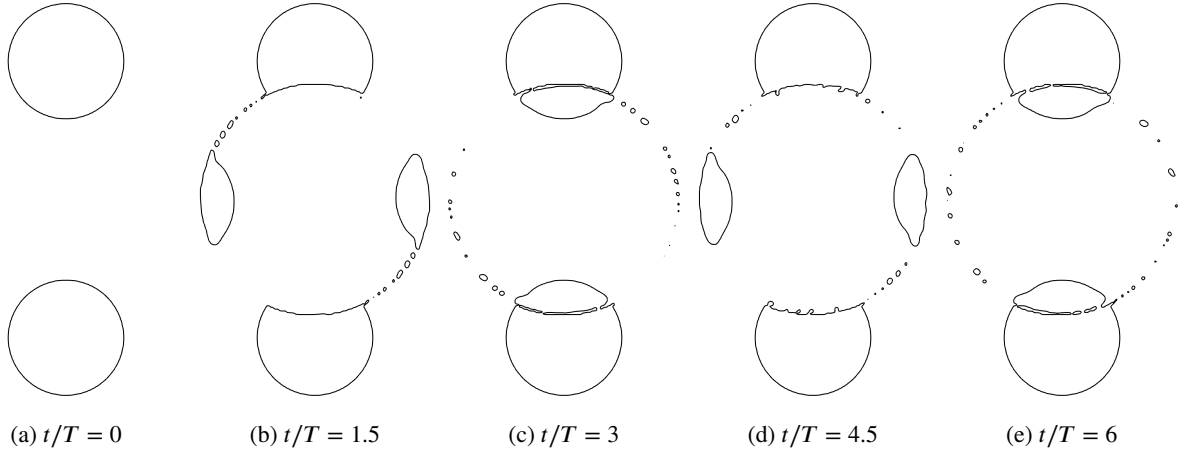
As an example, consider the case illustrated in figure 4. Two blobs of equal volumes  $v_2^n = v_1^n$  briefly coalesce at  $t = T$  to form a single blob of volume  $v_1^{n+1} = v_1^n + v_2^n$ . At  $t = 2T$  the single blob fragments along the original boundary such that no dark fluid was mixed. Here we consider connectedness provided by CCL a given, but note that such an event could be either physical or spurious, as the connectedness accuracy for closely passing blobs depends on the CCL method and its parameters [11]. First, consider the case where  $\Delta t_s = 2T$ . Eulerian volume tracking produces the correct VTM:

$$\mathbf{A}^{(0 \rightarrow 2T)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (39)$$

Even if this were a spurious event, ELA with  $\Delta t_s \geq 2T$  would correctly identify no exchange of volume and no cycles. Second, consider the case where  $\Delta t_s = T$ , with  $N = 2$  to get  $\Delta t_{s,\text{eff}} = 2T$ :

$$\tilde{\mathbf{A}}^{(0 \rightarrow 2T)} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}. \quad (40)$$

While  $\tilde{\mathbf{A}}^{(0 \rightarrow 2T)}$  still satisfies (12), the loss of spatial information when the vector color function is collapsed into a VTM at  $t = T$  creates a more diffuse tracking matrix. This loss of information means that there is no reliable way to decrease the diffusive error apart from decreasing  $N$  or introducing more information. Keeping  $N$  constant, one would have to make assumptions about the underlying flow to determine a likely evolutionary path from the possible events present in the diffuse effective VTM. This is equivalent to a (only slightly more constrained) Lagrangian tracking approach, and would have the same challenges. We note that, based on the interpretation of the VTM as a graph (see §2.2), it can



**Figure 5:** Evolution of volume fraction  $f = 0.5$  iso-surface in the two-dimensional vortical exchange simulation.

be shown that for the special case where  $\tilde{\mathbf{A}}$  (or a connected component of it) found using  $N > 1$  has no cycles, it must be equal to the tracking matrix  $\mathbf{A}'$  (or a corresponding connected component of it) found using a larger true snapshot interval and no multiplication over the same time period ( $K' = NK$ ).

#### 4. Numerical verification and demonstration of ELA using VTM

To verify and demonstrate the properties of ELA, we perform two types of simulations. In section §4.1 we perform two-phase prescribed-velocity simulations of a vortical flow and in §4.2 we perform two-phase Navier-Stokes simulations of the canonical problem of bubbles fragmenting in IHT. Both simulations are performed on a staggered Cartesian grids using the cVOF method with  $\epsilon = 10^{-12}$  used for (29) and (30).

For these simulations we choose the Informed Component Labeling (ICL) algorithm developed by Hendrickson et al. [10] as the CCL method to provide ELA with contiguous blobs because it is volume conservative, i.e., all dark fluid is marked as part of a blob. This allows us to validate that ELA is volume conservative. In addition to computational cells being adjacent and having  $f > 0$  for them to be considered part of a connected blob, ICL requires that the calculated interface normals (available from cVOF interface reconstruction) must not oppose. This additional connectivity criteria creates a bias towards representing unresolved structures as many separate blobs as opposed to a thin liquid bridge. However, as discussed by Chan et al. [11], differentiating between separate blobs and a thin liquid bridge is an unresolved computational issue for CCL.

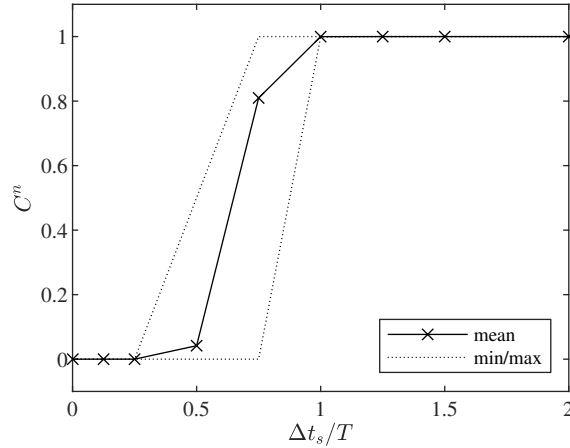
##### 4.1. Example involving two-dimensional vortical exchange

To illustrate the challenge of blob tracking when complex interactions are present, we consider a flow similar to the conceptual one shown in figure 5. Consider two circular blobs of radius  $a$  whose centers are both a distance  $L$  from the center of a forced vortex of radius  $R$  and rotation  $\Omega$ , giving an angular velocity field  $v_\theta(r)$  as a function of the radial distance  $r$  from the vortex center,

$$v_\theta(r) = \Omega\pi r \mathcal{H}(R - r), \quad (41)$$

where  $\mathcal{H}$  is the Heaviside step function. The vortex rips volume from one blob, creating smaller blobs which are then transferred to the other blob. The time it takes for this exchange of volume gives the characteristic time  $T = 1/\Omega$ . As discussed in §2.2, if  $\Delta t_s > T$ , cycles are formed which prevent a unique solution using previous methods. Here, we perform a two-dimensional simulation of  $L/R = 6/5$  and  $a/R = 1/2$  with resolution  $\Delta x = R/32$  over  $0 < t/T < 8$  and study the effect of  $\Delta t_s/T$  on cycle production. Figure (1) shows the evolution of the  $f = 0.5$  isosurface.

To quantify cycle production, we generate a matrix  $\mathbf{B}^{(n \rightarrow n+1)}$  by removing all elements of the VTM  $\mathbf{A}^{(n \rightarrow n+1)}$  that are not involved in a cycle. This is achieved by iteratively setting any element to zero if it is the only non-zero entry in a row or column until no such cases exist. To avoid inflating the count of cycles by including under-resolved events, we remove columns of  $\mathbf{A}^{(n \rightarrow n+1)}$  relating to under-resolved parent blobs,  $v_i^n < v_{\text{res}, 2D}$ , as well as rows relating to



**Figure 6:** Statistics for  $C^n$ , the proportion of total volume involved in cycles, using different snapshot intervals for the vortical exchange simulation over  $0 < t/T < 6$ .

under-resolved child blobs,  $v_j^{n+1} < v_{\text{res}, 2D}$ , where  $v_{\text{res}, 2D} = \pi(2\Delta x)^2$ . The proportion of the (resolved) volume of dark fluid involved in cycles can then be written

$$C^n = \frac{\sum [\mathbf{B}^{(n \rightarrow n+1)} \mathbf{v}^n]}{\sum [\mathbf{A}^{(n \rightarrow n+1)} \mathbf{v}^n]}. \quad (42)$$

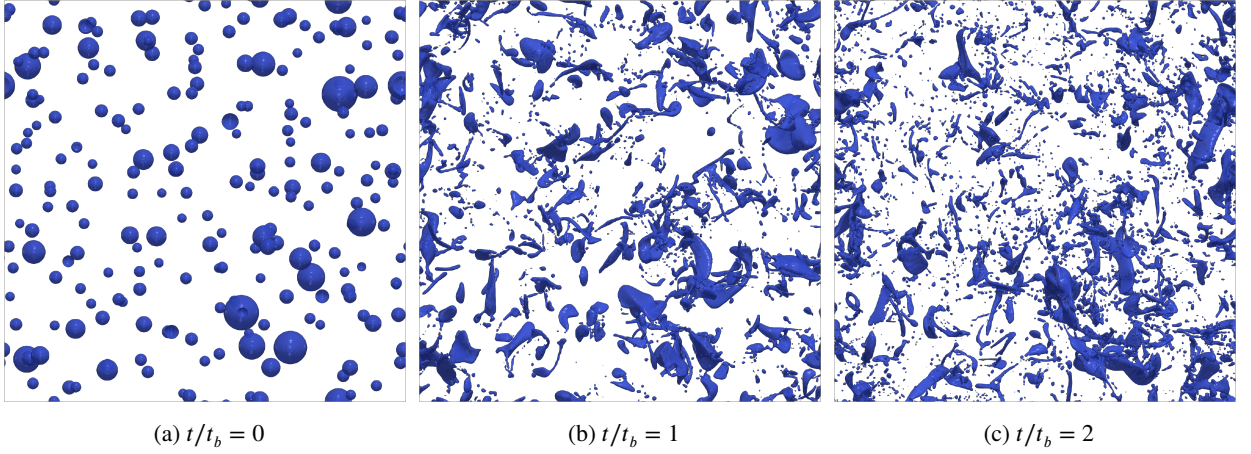
Figure 6 shows there is no cycle production for  $\Delta t_s \ll T$ , a jump in cycle production at  $\Delta t_s \sim T$ , and all volume is involved in a cycle for  $\Delta t_s > T$ . Thus, unless  $\Delta t_s \ll T$ , previous tracking methods would be unable to provide unique solutions to the flow of volume between blobs.

#### 4.2. Example involving three-dimensional bubble fragmentation in IHT

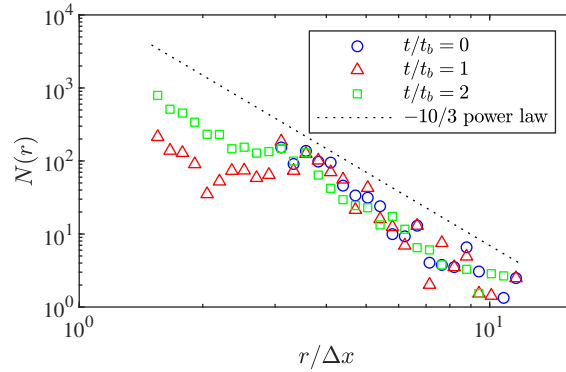
As a practical demonstration of ELA, we consider the canonical problem of a low void-fraction distribution of air-bubbles in water (density ratio  $\rho_a/\rho_w = 1000$ ) fragmented by strong IHT (Weber number  $We = \infty$ ). We choose IHT as it is spatially homogeneous, quasi-steady, and well understood, allowing simple measurement of averaged turbulence properties, particularly the turbulent dissipation rate  $\varepsilon$ . Additionally, multi-phase IHT has been well studied experimentally (e.g., [16, 25, 26]), and serves as a building block for understanding bubbly-flow near an air-entraining free surface, such as in ship wakes [14]. Using this canonical problem, we verify that ELA is volume conservative, quantify the abundance of loop currents, and demonstrate the use of matrix multiplication and the associated trade-off between cost and accuracy.

We perform three-dimensional direct numerical simulations on a triply periodic grid of  $256^3$ . IHT is created and maintained using a linear forcing method [27, 28] resulting in a turbulent Reynolds number  $Re = u_{\text{rms}}^4/\varepsilon\nu = 200$  and Kolmogorov scale  $\eta \approx \Delta x$ . After turbulence is initialized for one phase, a population of spherical bubbles with radii between  $r_{\text{min}} = 3\Delta x$  and  $r_{\text{max}} = 15\Delta x$  following a  $r^{-10/3}$  power law is randomly distributed without overlap at  $t = 0$  such that the void fraction is 1%. The details of this canonical problem and the description and validation of the flow solver are provided by Yu et al. [14].

For fragmentation of bubbles in IHT, a characteristic time is  $t_b = 0.42 \varepsilon^{-1/3} r_{\text{min}}^{2/3}$ , corresponding to the typical lifetime of the smallest bubbles [4, 25]. We run our simulations over  $0 < t/t_b < 2$ , over which we observe an increase from 215 to 588 resolved bubbles (defined as bubbles with a volume larger than  $v_{\text{res}, 3D} = 4/3\pi(1.5\Delta x)^3$ ). The evolution of the bubble field is shown in figure 7 and the bubble-size distribution is shown in figure 8. We perform a series of otherwise identical runs using different  $\Delta t_s$  (see table 1), then using (37) we extract tracking matrices describing the same effective snapshot interval  $\tau_{\text{eff}} = \Delta t_{s,\text{eff}}/t_b = 0.07$ , consistent with recommendations by Chan et al. [11]. Figure 2 provides an event observed by ELA over this snapshot interval and the corresponding tracking matrix from a subset of the domain. To evaluate the effect of the number of bubbles and events, two different time periods are considered: an early period  $0 < t/t_b < 1$  covering 15 effective snapshot intervals with fewer bubbles and events, and a late period  $1 < t/t_b < 2$  covering 15 effective snapshot intervals with more bubbles and events. Noting



**Figure 7:** Evolution of volume fraction  $f = 0.5$  iso-surface in the three-dimensional IHT simulation.



**Figure 8:** Evolution of the bubble-size distribution  $N(r)$  in the IHT simulation versus the expected  $-10/3$  power law. Note, only resolved bubbles ( $r/\Delta x \geq 1.5$ ) are considered.

that we track all bubbles, not just those that are resolved, the largest value of  $M^n$  is  $9 \times 10^4$  for the early and  $2 \times 10^5$  for the late period.

#### 4.2.1. Validating ELA volume conservation

Using case La and Lb ( $N = 1$ ), we first seek to verify the conservative nature of ELA. Following Weymouth and Yue [1], a measure of the typical change in the volume of dark fluid per simulation time step  $\Delta t$  is

$$(L_1)_{\text{cVOF}} = \frac{1}{TK} \sum_{n=0}^{T-1} |V^{n+1} - V^n|, \quad (43)$$

where  $V^n$  is the sum of all the blob volumes  $\mathbf{v}^n$ . Here,  $TK$  is the total number of simulation time steps. The total relative change over the entire simulation is

$$(\text{Change})_{\text{cVOF}} = \frac{V^0 - V^T}{V^0}. \quad (44)$$

We define equivalent metrics for ELA based on the tracking matrices:

$$(L_1)_{\text{ELA}} = \frac{1}{TK} \sum_{n=0}^{T-1} \left\| \mathbf{v}^{n+1} - \mathbf{A}^{(n \rightarrow n+1)} \mathbf{v}^n \right\|_1, \quad (45)$$

**Table 1**

Summary of different runs performed using the same flow but different ELA settings. Note, the  $\Delta t$  was chosen dynamically based on the flow solver's restrictions, so the average value based on all  $\Delta t_s$  intervals is reported for  $K$ .

	$N = \Delta t_{s,\text{eff.}} / \Delta t_s$	$K = \langle \Delta t_s / \Delta t \rangle$	Normalized with (47)?
Case La	1	109.9	no
Case Lb	1	109.9	yes
Case S1	2	54.96	yes
Case S2	4	27.50	yes
Case S3	8	13.75	yes
Case S4	16	6.870	yes
Case S5	32	3.434	yes
Case S6	64	1.717	yes

$$(\text{Change})_{\text{ELA}} = \frac{\sum [\mathbf{v}^T - \tilde{\mathbf{A}}^{(0 \rightarrow T)} \mathbf{v}^0]}{\sum [\mathbf{v}^T]}. \quad (46)$$

By comparing the new volume predicted by the VTM to the new volume calculated from the void fraction field, we measure any volume error in ELA separate from that related to cVOF.

**Table 2**

The  $L_1$  and relative change metrics for volume conservation error separated into ELA contribution and cVOF contribution for IHT simulations over  $0 < t/t_b < 2$  corresponding to  $T = 30$  snapshot intervals. Note that the  $L_1$  errors are per fluid solver step while relative change errors are over the entire simulation ( $\sim 3000$  steps).

	$(L_1)_{\text{ELA}} / V^0$	$(\text{Change})_{\text{ELA}}$	$(L_1)_{\text{cVOF}} / V^0$	$(\text{Change})_{\text{cVOF}}$
Case La	$8.2 \times 10^{-16}$	$-1.9 \times 10^{-12}$	$5.7 \times 10^{-13}$	$1.9 \times 10^{-9}$
Case Lb	$2.2 \times 10^{-17}$	$4.8 \times 10^{-14}$	"	"

The results for the IHT simulations are shown in table 2. We first run the normal ELA method described in §3.2 (case La). As expected,  $(L_1)_{\text{cVOF}} / V^0 \sim \mathcal{O}(\epsilon)$ , indicating the growth in cVOF volume error is due to  $\epsilon$  in (29). Therefore, if  $\epsilon$  is changed we expect  $(L_1)_{\text{cVOF}}$  and  $(\text{Change})_{\text{cVOF}}$  to change proportionally. As  $s$  remains consistent with  $f$  through (30),  $\epsilon$  does not affect the ELA metrics in table 2. The growth of the ELA volume conservation error per step,  $(L_1)_{\text{ELA}} / V^0$ , is approximately machine precision, validating that the ELA method is volume conservative to machine precision. To minimize the effect of machine precision, we run a second simulation (case Lb) where, after every operator-split step (22), we normalize by

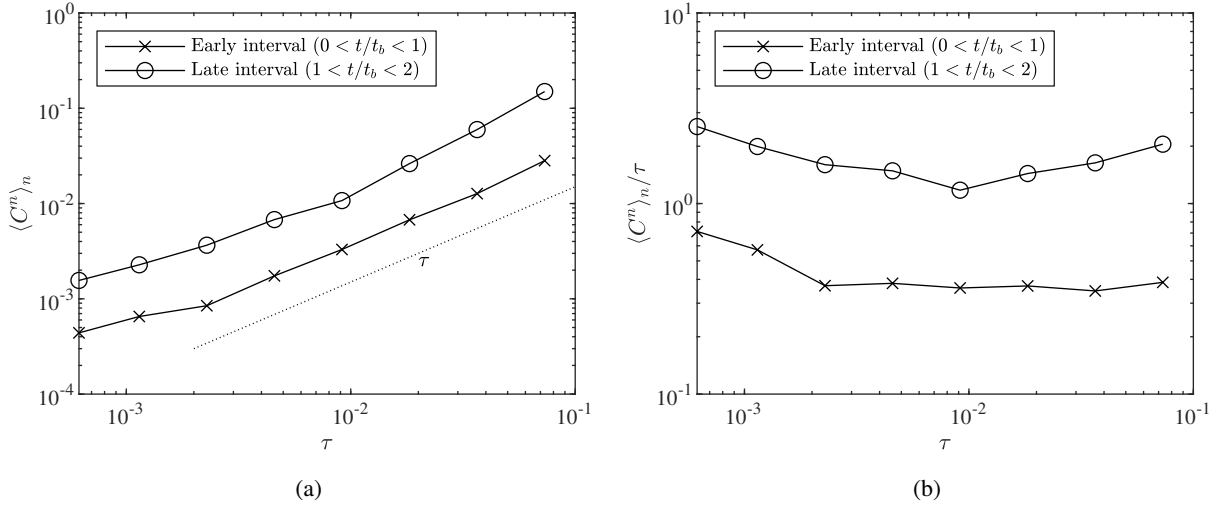
$$\mathbf{s}^{(d)} = f^{(d)} \mathfrak{s}^{(d)} \quad (47)$$

to limit the accumulation of error related to machine precision. This achieves a 1/40 reduction in the (already near machine precision) error. For the rest of the IHT simulations, we use this normalization.

#### 4.2.2. Relationship between cycle generation and snapshot interval

We now evaluate cycle production in the IHT simulation using the same analysis as in §4.1. Again, to avoid inflating the count of cycles by including under-resolved events, we remove columns of  $\mathbf{A}^{(n \rightarrow n+1)}$  relating to under-resolved parent bubbles,  $v_i^n < v_{\text{res}, 3\text{D}}$ , and rows relating to under-resolved child bubbles,  $v_j^{n+1} < v_{\text{res}, 3\text{D}}$ . We use VTM from the true snapshot intervals of each run and do not use matrix multiplication, providing a range of  $\tau = \Delta t_s / t_b$ . The results are shown in figure 9.

Previous work on numerical tracking [11, 12, 19] has identified that over small time intervals, CCL causes chains of spurious fragmentation and coalescence, as CCL methods struggle to consistently identify distinct regions of dark fluid separated by lengths on the order of the grid. Generally, interfaces can be arbitrarily close, making this a fundamental



**Figure 9:** The average proportion of the volume of resolved bubbles involved in cycles  $C^n$  (a), and the average rate of change  $C^n/\tau$  (b), for IHT simulations with different true snapshot intervals  $\tau = \Delta t_s/t_b$  (see table 1) compared to a linear relationship.

limitation of CCL [9]. When tracking blobs identified by CCL, imposing a minimum  $\Delta t_s$  typically mitigates the inclusion of spurious events [11, 12]. As this strategy is adopted to improve the CCL information provided to tracking, it applies to all tracking methods, including ELA. Treating figure 4 as an idealized example of a spurious event caused by CCL in a given flow, we see that  $\Delta t_s \geq 2T$  eliminates the spurious event from the VTM. Note that the characteristic time period of spurious events, e.g.,  $T$ , is likely sensitive to the fluid solver and the CCL method.

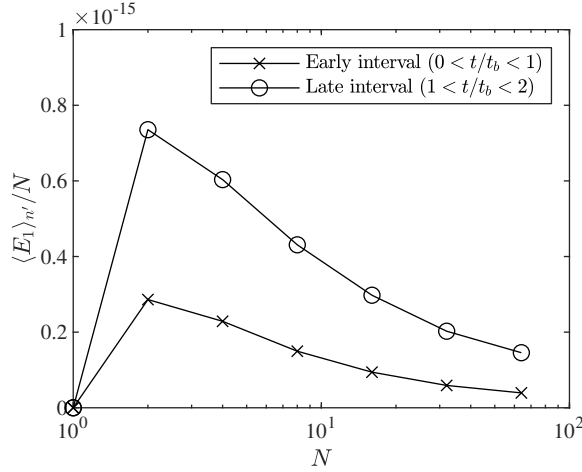
Cycle production depends on the number of events and number of blobs, as illustrated by the difference between the early and late time periods; however over both time periods an approximately linear scaling  $C^n \propto \tau$  is observed. Considering the magnitude of  $C^n$ , we observe that, for  $\tau = \mathcal{O}(0.1)$  proposed by Chan et al. [11], 5% of the resolved volume is involved in cycles for the early period and 20% of the resolved volume is involved in cycles for the late period. This means previous tracking methods would be unable to provide unique solutions to the tracking matrix, and the differences between solutions could be significant. In § 2.2, we establish that volume-based Eulerian tracking does not introduce cycles, thus these cycles are the result of either physical exchanges of volume or spurious events caused by CCL. Because tracking considers CCL given, ELA cannot directly quantify what portion of these cycles fall into each category; however, the results in figure 9 show that physical cycles are abundant in this simulation. As  $C^n$  provides a measure of the total cycle production over an interval  $\tau$ ,  $C^n/\tau$  provides a measure of the rate of cycle production. For small  $\tau \lesssim 10^{-2}$ ,  $C^n/\tau$  decreases with increasing  $\tau$ , consistent with the decreasing probability of spurious events. For the late interval and  $\tau > 10^{-2}$ ,  $C^n/\tau$  increases with increasing  $\tau$ , consistent with the increasing probability of physical cycles (see § 4.1). This suggests that physical cycles dominate at large  $\tau$  in the late interval of this simulation.

On selecting  $\Delta t_s$  in practice, Lagrangian tracking methods also apply an upper bound on  $\Delta t_s$  because their binary assumption precludes identifying events over long time periods [11] and their identification of advection is inaccurate over large displacements [12]. Due to its Eulerian volume-tracking nature, ELA does not suffer these limitations and is accurate for large  $\Delta t_s$ . With accurate large  $\Delta t_s$  now possible, ELA reduces the effect of spurious events caused by CCL. Separately, a physical motivation for selecting  $\Delta t_s$  is that it defines the distinction between a single event and multiple events, and thus directly affects measured event statistics [26]. The proper choice of  $\Delta t_s$  for this purpose is an area of ongoing investigation [29].

#### 4.2.3. Effect of matrix multiplication on accuracy and memory cost

To study the effects of approximating long snapshot intervals through matrix multiplication, we perform seven otherwise identical simulations with different  $\Delta t_s$ , then multiply the resulting VTMs to achieve the same  $\Delta t_{s,\text{eff}}/t_b = 0.07$  (see table 1). Defining  $n'$  to index the effective snapshot interval, i.e.,  $t^{n'+1} = t^{n'} + \Delta t_{s,\text{eff}}$ , each simulation and





**Figure 10:** The average normalized  $E_1$  error for two intervals of the IHT simulations using different true snapshot intervals (see table 1).

subsequent matrix multiplication generates VTMs describing

$$\mathbf{v}^{n'+1} = \tilde{\mathbf{A}}^{(n' \rightarrow n'+1)} \mathbf{v}^{n'}. \quad (48)$$

For case Lb,  $(\Delta t_s)_{\text{Lb}} = (\Delta t_{s,\text{eff}})_{\text{Lb}}$ , so no matrix multiplication is necessary and  $(\tilde{\mathbf{A}})_{\text{Lb}} = (\mathbf{A})_{\text{Lb}}$ . Using this case as a reference, the difference for each shorter true snapshot interval case (Case S1, Case S2, etc.) is defined

$$\left( \mathbf{D}^{n'} \right)_{\text{SX}} = \left( \tilde{\mathbf{A}}^{(n' \rightarrow n'+1)} \right)_{\text{SX}} - \left( \mathbf{A}^{(n' \rightarrow n'+1)} \right)_{\text{Lb}}. \quad (49)$$

We remove columns of  $\mathbf{D}^{n'}$  and entries of  $\mathbf{v}^{n'}$  relating to under-resolved parent bubbles,  $v_i^{n'} < v_{\text{res}, 3\text{D}}$ .

We first validate that matrix multiplication is a volume conservative description of the transition shown in (48). We expect  $\mathbf{D}^{n'} \mathbf{v}^{n'} = \mathbf{0}$ , so the normalized error in  $\tilde{\mathbf{A}}^{(n' \rightarrow n'+1)}$ 's prediction of the volume  $\mathbf{v}^{n'+1}$  due to matrix multiplication is

$$(E_1)^{n'} = \left\| \mathbf{D}^{n'} \mathbf{v}^{n'} \right\|_1 / \left\| \mathbf{v}^{n'+1} \right\|_1. \quad (50)$$

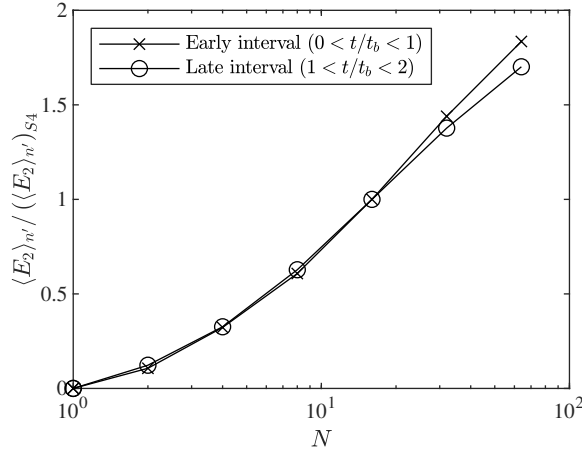
Figure 10 shows the average  $E_1$  error for the early and late time period normalized by  $N$ , the number of matrix multiplications required to calculate  $\tilde{\mathbf{A}}^{(n' \rightarrow n'+1)}$ . The  $\mathcal{O}(10^{-15})$  normalized error for all simulations shows that matrix multiplication is volume conservative to machine precision.

To quantify the loss of accuracy due to matrix multiplication (see discussion in §3.5), we examine how individual columns of the VTM differ. Based on  $\mathbf{D}^{n'} = \{(d_{ij})^{n'}\}$ , we define the average difference per column for each matrix

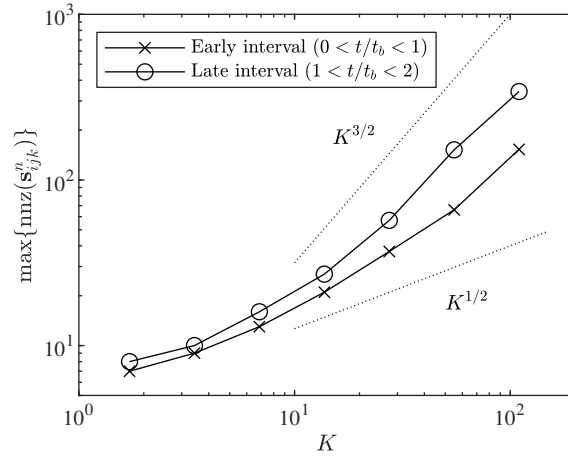
$$(E_2)^{n'} = \frac{1}{2} \left\langle \sum_i |(d_{ij})^{n'}| \right\rangle_j. \quad (51)$$

The factor of 1/2 guarantees  $0 \leq E_2 \leq 1$ . Figure 11 shows that the trend of the growth of the average  $E_2$  error for both the early and late time periods behaves similarly with changing  $N$ , apart from a scaling coefficient. This scaling factor is likely dependent on the number of bubbles and number of events. Additionally, although we do not examine these here, it could also potentially depend on the type of CCL method used, the  $\Delta t_{s,\text{eff}}$  chosen, and the general properties of a flow. For  $N \gg 1$ ,  $E_2$  exhibits approximately logarithmic growth. We highlight that large  $N$ s, and therefore small  $K$ s, can be used with only moderate loss of accuracy, but representing a significant reduction in memory needed to store  $\mathbf{s}$  without decreasing  $\Delta t_{s,\text{eff}}$ .

We now examine the scaling of the memory requirement to store  $\mathbf{s}^n$ . Each non-zero entry of  $\mathbf{s}^n$  within a grid cell is recorded in terms of its index (4-byte integer) and value (8-byte double). In our implementation memory is pre-allocated



**Figure 11:** The growth of the average  $E_2$  error with  $N$  for two intervals of the IHT simulations normalized by the value for  $N = 16$  (case S4):  $\langle (E_2)_{n'} \rangle_{S4} = 0.029$  for the early and  $\langle (E_2)_{n'} \rangle_{S4} = 0.079$  for the late interval.



**Figure 12:** The growth of largest number of non-zeros in the source vector field  $\mathbf{s}^n$  with  $K$  for two intervals of the IHT simulations.

equally among grid cells, so the memory requirement per grid cell is  $12 \times \max\{\text{nnz}(\mathbf{s}^n)\}$  bytes. Noting that  $\tilde{\mathbf{c}}$  must also be stored during advection, approximately twice this memory is needed in total for ELA. Figure 12 shows that the actual scaling of  $\max\{\text{nnz}(\mathbf{s}^n)\}$  is well less than the  $K^3$  scaling predicted by the upper-bound (35). While the upper-bound is based only on the number of bubbles and  $K$ , like  $E_2$ , the actual value of  $\max\{\text{nnz}(\mathbf{s}^n)\}$  likely depends on a variety of factors. Comparing the early and late intervals in figure 12, the consistent increase in magnitude corresponding to an increase in the number of bubbles and the otherwise similar scaling with  $K$  demonstrate number of bubbles and  $K$  are important factors. For the late interval of Case Lb, the largest  $K$  considered, (35) gives  $\max\{\text{nnz}(\mathbf{s}^n)\} < 2 \times 10^5$ , but we find  $\max\{\text{nnz}(\mathbf{s}^n)\} = 342$ , corresponding to a (theoretical) total ELA memory requirement of only 140 GB for the entire grid. This shows that, in practice, ELA can be significantly less memory intensive than (35) would suggest, making ELA feasible with large  $K$  and no matrix multiplication. Still, as  $\max\{\text{nnz}(\mathbf{s}^n)\}$  is only known *a posteriori*, some trial and error may be necessary to determine a sufficient memory allocation for a given application. Alternatively, dynamic storage of the non-zero entries of  $\mathbf{s}^n$  could be used, e.g., a linked list. While not implemented here, this would remove the need to estimate  $\max\{\text{nnz}(\mathbf{s}^n)\}$  *a priori*, and allow the memory usage of a grid cell to be set by its own  $\text{nnz}(\mathbf{s}^n)$  rather than the global maximum.

Finally, we show that the total computational costs of ELA are reasonable, even for large  $\Delta t_s$  without matrix multiplication ( $N = 1$ ). For example, our simulations are run on two nodes of an HPE SGI 8600 cluster, where each

node has 48 cores at 2.7 GHz and 170 GB of usable memory. Without ELA, the memory usage is 33 GB and the total wall time required is 9 hours. For Case Lb, we allocate space for 448 non-zero entries of  $\mathbf{s}^n$  per grid cell, slightly larger than the actual maximum of 342. The memory usage is 268 GB and the total wall time required is 19 hours. We note that ELA is quite computationally efficient considering that it is solving up to  $M^n = 2 \times 10^5$  advection equations in (4). Given an appropriately selected  $\Delta t_s$  (see discussion in §4.2.2), one should select the smallest  $N$  (and thus the largest  $K$ ) that is possible given one's computational resources. Using  $N = 1$  with an appropriate  $\Delta t_s$  is often feasible on scientific computing cluster. On the other hand, if using matrix multiplication ( $N > 1$ ), the effect of diffusive error on the VTM must be evaluated (e.g., figure 11).

## 5. Conclusions

We develop a new volume-based tracking framework that can completely describe the flow of volume between blobs during their evolution. The description is in the form of a volume-tracking matrix (VTM) that provides the volume flow between any two blobs at adjacent snapshots in time. To calculate this VTM, we develop ELA. Unlike previous blob-tracking methods [11, 12, 19, 20], ELA explicitly provides the unique VTM and requires no assumptions about the arity of the events. Even without arity assumptions, we demonstrate that previous Lagrangian and event-based Eulerian methods could only reliably provide which pairs of blobs had volume flow between them, not how much volume had flowed. The VTM provided by ELA always provides the unique flow of volume between blobs, creating a more complete description of their evolution, from which individual events can later be extracted.

To develop ELA, we extend the cVOF method [1] from a scalar equation for advecting a volume fraction  $f$  to a vector equation for advecting a source fraction  $\mathbf{s}^n$ , which describes which previous blobs contributed to the volume in a cell. To remain consistent and minimize additional computational operations, the vector flux and dilation terms for ELA are based directly on the equivalent scalar terms calculated for cVOF. Additionally, this reuse of the terms allows ELA to be volume conservative to machine precision given the same Courant condition required by cVOF. We prove the conservation theoretically and verify this using the canonical problem of bubbles fragmenting in IHT simulations [14].

Although ELA requires limited computational operations, there is a memory cost related to storing  $\mathbf{s}^n$  over long intervals, even when only non-zero entries are stored for each cell. We show how a long-interval VTM can be approximated by multiplying the VTMs from consecutive shorter intervals. While still conservative, this matrix multiplication introduces a diffusive error into the approximated long-interval VTM when cycles are present. From IHT simulations, we demonstrate this tradeoff between memory requirements and diffusive error, and find that, while theoretical bounds suggest long intervals are impractical, in practice the actual memory requirements are reasonable. Understanding the practical performance for additional cases, particularly free-surface entrainment, is an area of future work.

The volume-based description of blob evolution generated uniquely by ELA provides a complete description of the flow of volume among blobs in two-phase simulations through a VTM, a matrix form that is convenient for post analysis. ELA will allow detailed analysis of blob evolution mechanisms (e.g., as fragmentation and entrainment) in complex flows with many blobs and interacting evolution mechanisms, with no necessary assumptions about the evolution mechanisms, i.e., arity. Because ELA is capable of capturing arbitrarily-high arity, it is well suited for evaluating and improving existing models of these evolution mechanisms. Large ELA data sets of these inherently stochastic mechanisms will pave the way for elucidating and quantifying the underlying physics of these mechanisms and their interactions.

## Acknowledgments

This work was funded by the U.S. Office of Naval Research grant N00014-20-1-2059 under the guidance of Dr. W.-M. Lin. The computational resources were funded through the Department of Defense High Performance Computing Modernization Program.

## A. Proof of ELA volume Conservation

Following the proof of volume conservation of cVOF by Weymouth and Yue [1], we prove the same for ELA by showing that there will be no over or under filling, i.e.,  $0 \leq (s_l)^{(d)} \leq 1$  for all  $l$ . We will prove that, with an appropriate Courant restriction, if  $0 \leq (s_l)^{(d-1)} \leq 1$  for all  $l$  then ELA cannot under fill, i.e.,  $(s_l)^{(d)} \geq 0$  for all  $l$ . Because the sum of the components of  $\mathbf{s}^{(d)}$  is bounded by (28), this single inequality is sufficient to prove  $0 \leq (s_l)^{(d)} \leq 1$  for all  $l$ .

For each operator-split step, we consider four general cases based on the velocity on the positive ( $u_r$ ) and negative side ( $u_l$ ) side of the cell:

- a.  $u_r, u_l > 0$
- b.  $u_r > 0, u_l < 0$
- c.  $u_r < 0, u_l > 0$
- d.  $u_r, u_l < 0$ .

Case (d) is symmetric with case (a), so proving case (a) cannot under fill also proves case (d) cannot under fill. Further, two sub-cases of case (a) are considered: (a1)  $u_r \geq u_l$  and (a2)  $u_r < u_l$ . For convenience we scale the velocities and fluxes to local Courant numbers,

$$\bar{u} = \frac{u\Delta t}{\Delta x_d}, \quad (52a)$$

$$\bar{\mathbf{F}} = \mathbf{F} \frac{\Delta t}{\Delta \Omega}. \quad (52b)$$

Thus, (22) becomes

$$\mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} = \bar{\mathbf{F}}_{d+1/2} - \bar{\mathbf{F}}_{d-1/2} + \tilde{\mathbf{c}} \Delta \bar{u}_d \quad \text{for } d \in 1 \dots \mathcal{N}, \quad (53)$$

where  $\Delta \bar{u}_d = \bar{u}_r - \bar{u}_l$ . Substituting in (25) and (26) and introducing absolute values to illustrate the sign of each term,

$$\text{case (a1)} \quad \mathbf{s}_d^{(d)} - \mathbf{s}_d^{(d-1)} = -\hat{\mathbf{s}}_d^{(d-1)} \left| \bar{F}_{d+1/2} \right| + \hat{\mathbf{s}}_{d-1}^{(d-1)} \left| \bar{F}_{d-1/2} \right| + \hat{\mathbf{s}}_d^{(0)} \tilde{\mathbf{c}} \left| \Delta \bar{u}_d \right|, \quad (54a)$$

$$\text{case (a2)} \quad \mathbf{s}_d^{(d)} - \mathbf{s}_d^{(d-1)} = -\hat{\mathbf{s}}_d^{(d-1)} \left| \bar{F}_{d+1/2} \right| + \hat{\mathbf{s}}_{d-1}^{(d-1)} \left| \bar{F}_{d-1/2} \right| - \hat{\mathbf{s}}_d^{(0)} \tilde{\mathbf{c}} \left| \Delta \bar{u}_d \right|, \quad (54b)$$

$$\text{case (b)} \quad \mathbf{s}_d^{(d)} - \mathbf{s}_d^{(d-1)} = -\hat{\mathbf{s}}_d^{(d-1)} \left| \bar{F}_{d+1/2} \right| - \hat{\mathbf{s}}_{d-1}^{(d-1)} \left| \bar{F}_{d-1/2} \right| + \hat{\mathbf{s}}_d^{(0)} \tilde{\mathbf{c}} \left| \Delta \bar{u}_d \right|, \quad (54c)$$

$$\text{case (c)} \quad \mathbf{s}_d^{(d)} - \mathbf{s}_d^{(d-1)} = +\hat{\mathbf{s}}_{d+1}^{(d-1)} \left| \bar{F}_{d+1/2} \right| + \hat{\mathbf{s}}_{d-1}^{(d-1)} \left| \bar{F}_{d-1/2} \right| - \hat{\mathbf{s}}_d^{(0)} \tilde{\mathbf{c}} \left| \Delta \bar{u}_d \right|. \quad (54d)$$

Dropping the positive terms gives the bounds,

$$\text{case (a1)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq \hat{\mathbf{s}}^{(d-1)} \bar{F}_{d+1/2}, \quad (55a)$$

$$\text{case (a2)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq \hat{\mathbf{s}}^{(d-1)} \bar{F}_{d+1/2} + \hat{\mathbf{s}}^{(0)} \tilde{\mathbf{c}} \Delta \bar{u}_d, \quad (55b)$$

$$\text{case (b)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq \hat{\mathbf{s}}^{(d-1)} \Delta \bar{F}_d, \quad (55c)$$

$$\text{case (c)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq \hat{\mathbf{s}}^{(0)} \tilde{\mathbf{c}} \Delta \bar{u}_d, \quad (55d)$$

where  $\Delta \bar{F}_d = \bar{F}_{d+1/2} - \bar{F}_{d-1/2}$ . As all values of  $\mathbf{s}$  and  $\hat{\mathbf{s}}$  are for the cell of interest, not its neighbors, we have dropped the subscript index. We note that by (23), the normalization (24) is equivalent to

$$\hat{\mathbf{s}}^{(d)} = \frac{\mathbf{s}^{(d)}}{f^{(d)}} \quad (56)$$

We start with case (a1) and case (b). For case (a) and case (b), cVOF ensures [1]

$$\bar{F}_{d+1/2}, \Delta \bar{F}_d \geq -f^{(d-1)}. \quad (57)$$

Substituting into the respective bound and simplifying using (56) obtains

$$\text{case (a1)} \quad \mathbf{s}^{(d)} \geq 0, \quad (58a)$$

$$\text{case (b)} \quad \mathbf{s}^{(d)} \geq 0. \quad (58b)$$

This demonstrates that (57), guaranteed by cVOF, is sufficient to guarantee case (a1) and case (b) cannot under fill.

For case (a2) and case (c), it is straightforward to show that if  $\tilde{c} = 0$ , then  $\mathbf{s}^{(d)} \geq 0$ . We consider  $f^{(0)} > 1/2$  giving  $\tilde{c} = 1$ ,  $f^{(0)} > 1/2$ . For case (a), in addition to (57), cVOF ensures [1]

$$\bar{F}_{d+1/2}, \Delta \bar{F}_d \geq -\bar{u}_r. \quad (59)$$

Writing the bounds from (55) in terms of scaled velocities,

$$\text{case (a2)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq -\hat{\mathbf{s}}^{(d-1)} \bar{u}_r + \hat{\mathbf{s}}^{(0)} (\bar{u}_r - \bar{u}_l), \quad (60a)$$

$$\text{case (c)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq \hat{\mathbf{s}}^{(0)} (\bar{u}_r - \bar{u}_l). \quad (60b)$$

We now further split case (a2) into case (a2.i),  $\hat{\mathbf{s}}^{(0)} < \hat{\mathbf{s}}^{(d-1)}$  and case (a2.ii),  $\hat{\mathbf{s}}^{(0)} \geq \hat{\mathbf{s}}^{(d-1)}$ . As we are only considering  $f^{(0)} > 1/2$ , (56) gives  $\hat{\mathbf{s}}^{(0)} \leq 2\mathbf{s}^{(0)}$ . Thus,

$$\text{case (a2.i)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq -\hat{\mathbf{s}}^{(d-1)} \bar{u}_l, \quad (61a)$$

$$\text{case (a2.ii)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq -2\mathbf{s}^{(0)} \bar{u}_l, \quad (61b)$$

$$\text{case (c)} \quad \mathbf{s}^{(d)} - \mathbf{s}^{(d-1)} \geq 2\mathbf{s}^{(0)} (\bar{u}_r - \bar{u}_l). \quad (61c)$$

We first consider case (a2.ii) and case (c), where a bound is necessary to ensure successive operations cannot under fill. After summing over  $d \leq \mathcal{N}$  operations,

$$\text{case (a2.ii)} \quad \mathbf{s}^{(d)} \geq \mathbf{s}^{(0)} \left[ 1 - 2 \sum_{d'=1}^d |\bar{u}_{d'}| \right], \quad (62a)$$

$$\text{case (c)} \quad \mathbf{s}^d \geq \mathbf{s}^{(0)} \left[ 1 + 2 \sum_{d'=1}^d \min(\Delta \bar{u}_{d'}, 0) \right], \quad (62b)$$

where, to ensure the symmetric case (d) is also captured,  $|\bar{u}_d| = \max(|\bar{u}_l|, |\bar{u}_r|)$ . For a divergence-free flow, (62a) and (62b) guarantee  $\mathbf{s}^d \geq 0$  given the Courant condition

$$C = \sum_{d=1}^{\mathcal{N}} |\bar{u}_d| \leq \frac{1}{2}, \quad (63)$$

for case (a2.ii) and case (c).

Finally, we consider case (a2.i). (61a) can be rewritten

$$\text{case (a2.i)} \quad \mathbf{s}^{(d)} \geq \mathbf{s}^{(d-1)} \left[ 1 - \frac{\bar{u}_l}{f^{(d-1)}} \right]. \quad (64)$$

Therefore,  $f^{(d-1)} \geq |u_d|$  guarantees  $\mathbf{s}^{(d)} \geq 0$ . Rewriting (20) for  $\tilde{c} = 1$  in terms of the scaled velocities and fluxes gives

$$f^{(d)} = f^{(d-1)} + \Delta \bar{F}_d + (\bar{u}_r - \bar{u}_l). \quad (65)$$

Using (59),  $f^{(d)} \geq f^{(d-1)} - \bar{u}_l$ . Summing (65) over  $d - 1 \leq \mathcal{N} - 1$  operations and recalling we are interested in  $f^{(0)} > 1/2$ , it can be shown cVOF guarantees the bound

$$f^{(d-1)} \geq 1/2 - C + |u_d|. \quad (66)$$

Thus  $f^{(d-1)} \geq |u_d|$  and therefore  $\mathbf{s}^{(d)} \geq 0$  is true if  $C \leq 1/2$ , the same condition as (63).

## References

- [1] G. Weymouth, D. K. Yue, Conservative Volume-of-Fluid method for free-surface simulations on Cartesian-grids, *J. Comput. Phys.* 229 (2010) 2853–2865.
- [2] S. A. Thorpe, On the clouds of bubbles formed by breaking wind-waves in deep water, and their role in air-sea gas transfer, *Philos. Trans. R. Soc. London, Ser. A* 304 (1982) 155–210.
- [3] G. B. Deane, M. D. Stokes, Scale dependence of bubble creation mechanisms in breaking waves, *Nature* 418 (2002) 839–844.
- [4] C. Garrett, M. Li, D. Farmer, The connection between bubble size spectra and energy dissipation rates in the upper ocean, *J. Phys. Oceanogr.* 30 (2000) 2163–2171.
- [5] M. V. Trevorrow, S. Vagle, D. M. Farmer, Acoustical measurements of microbubbles within ship wakes, *J. Acoust. Soc. Am.* 95 (1994) 1922–1930.
- [6] F. Veron, *Ocean Spray*, *Annu. Rev. Fluid Mech.* 47 (2015) 507–538.
- [7] F. A. Williams, *Combustion Theory*, 2 ed., The Benjamin/Cummings Publishing Company, Inc., Menlo Park, 1985.
- [8] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, Y. Chao, The connected-component labeling problem: A review of state-of-the-art algorithms, *Pattern Recognit.* 70 (2017) 25–43.
- [9] M. Herrmann, A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure, *J. Comput. Phys.* 229 (2010) 745–759.
- [10] K. Hendrickson, G. D. Weymouth, D. K. Yue, Informed component label algorithm for robust identification of connected components with volume-of-fluid method, *Comput. Fluids* 197 (2020) 104373.
- [11] W. H. R. Chan, M. S. Dodd, P. L. Johnson, P. Moin, Identifying and tracking bubbles and drops in simulations: A toolbox for obtaining sizes, lineages, and breakup and coalescence statistics, *J. Comput. Phys.* 432 (2021) 110156.
- [12] Q. Gao, G. B. Deane, H. Liu, L. Shen, A robust and accurate technique for Lagrangian tracking of bubbles and detecting fragmentation and coalescence, *Int. J. Multiphase Flow* 135 (2021) 103523.
- [13] W. H. R. Chan, P. L. Johnson, P. Moin, J. Urzay, The turbulent bubble break-up cascade. Part 2. Numerical simulations of breaking waves, *J. Fluid Mech.* 912 (2021) A43.
- [14] X. Yu, K. Hendrickson, B. K. Campbell, D. K. Yue, Numerical investigation of shear-flow free-surface turbulence and air entrainment at large Froude and Weber numbers, *J. Fluid Mech.* 880 (2019) 209–238.
- [15] C. Martínez-Bazán, J. L. Montañés, J. C. Lasheras, 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.* 401 (1999) 183–207.
- [16] Y. Qi, A. U. Mohammad Masuk, R. Ni, Towards a model of bubble breakup in turbulence through experimental constraints, *Int. J. Multiphase Flow* 132 (2020) 103397.
- [17] K. T. Kiger, J. H. Duncan, Air-Entrainment Mechanisms in Plunging Jets and Breaking Waves, *Annu. Rev. Fluid Mech.* 44 (2012) 563–596.
- [18] Q. Gao, G. B. Deane, L. Shen, Bubble production by air filament and cavity breakup in plunging breaking wave crests, *J. Fluid Mech.* 929 (2021) A44.
- [19] C. Rubel, M. Owkes, Extraction of droplet genealogies from high-fidelity atomization simulations, *At. Sprays* 29 (2019) 709–739.
- [20] T. R. Langlois, C. Zheng, D. L. James, Toward animating water with complex acoustic bubbles, *ACM Trans. Graph.* 35 (2016) 1–13.
- [21] P. Karnakov, F. Wermelinger, S. Litvinov, P. Koumoutsakos, Aphros: High Performance Software for Multiphase Flows with Large Scale Bubble and Drop Clusters, in: *Proceedings of the Platform for Advanced Scientific Computing Conference*, ACM, New York, NY, USA, 2020, pp. 1–10. doi:[10.1145/3394277.3401856](https://doi.org/10.1145/3394277.3401856).
- [22] N. Balcázar, O. Lehmkuhl, J. Rigola, A. Oliva, A multiple marker level-set method for simulation of deformable fluid particles, *Int. J. Multiphase Flow* 74 (2015) 125–142.
- [23] E. Coyajee, B. J. Boersma, Numerical simulation of drop impact on a liquid–liquid interface with a multiple marker front-capturing method, *J. Comput. Phys.* 228 (2009) 4444–4467.
- [24] A. Baraldi, M. Dodd, A. Ferrante, A mass-conserving volume-of-fluid method: Volume tracking and droplet surface-tension in incompressible isotropic turbulence, *Comput. Fluids* 96 (2014) 322–337.
- [25] C. Martínez-Bazán, J. L. Montañés, J. C. Lasheras, On the breakup of an air bubble injected into a fully developed turbulent flow. part 1. breakup frequency, *J. Fluid Mech.* 401 (1999) 157–182.
- [26] J. Vejražka, M. Zedníková, P. Stanovský, Experiments on breakup of bubbles in a turbulent flow, *AIChE J.* 64 (2018) 740–757.
- [27] T. S. Lundgren, Linearly forced isotropic turbulence, Technical Report, Center for Turbulence Research, Stanford University, 2003.
- [28] C. Rosales, C. Meneveau, Linear forcing in numerical simulations of isotropic turbulence: Physical space implementations and convergence properties, *Phys. Fluids* 17 (2005) 095106.
- [29] J. Solsvik, S. Maaß, H. A. Jakobsen, Definition of the Single Drop Breakup Event, *Ind. Eng. Chem. Res.* 55 (2016) 2872–2882.