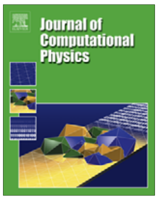
Journal of Computational Physics 229 (2010) 745–759

Contents lists available at ScienceDirect

Journal of Computational Physics

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



A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure

M. Herrmann

School of Mechanical, Aerospace, Chemical, and Materials Engineering, Arizona State University, Tempe, AZ 85287-6106, USA

ARTICLE INFO

Article history:
Received 26 January 2009
Received in revised form 2 October 2009
Accepted 7 October 2009
Available online 3 November 2009

Keywords:
Atomization
Two phase flow
Eulerian/Lagrangian coupling
Level set method
Refined level set grid method
Spray

ABSTRACT

This paper presents a parallel Eulerian/Lagrangian multi-scale coupling procedure for two-phase flows. At the fully resolved scale, the dynamically evolving phase interface is tracked using a Eulerian approach. In regions of the flow, where the phase interface geometry can no longer be resolved adequately, separated, small scale liquid structures are described by a Lagrangian point particle approach. The coupling procedure of these two descriptions consists of an efficient parallel algorithm that identifies tracked liquid candidate structures, removes them from the resolved Eulerian description, and inserts them into the Lagrangian description preserving their position, mass, momentum, and lower order shape. While in principle applicable to level set, Volume of Fluid, and marker particle interface tracking methods for the fully resolved scale, this paper focuses on examples from atomization simulations using the refined level set grid method.

1. Introduction

Interfaces, like liquid/gas or liquid/solid phase interfaces, play a dominant role in many technical applications and natural phenomena. Often times, the dynamics of these interfaces exhibit a vast range of length and timescales that make them difficult to model and numerically simulate. A prime example for this is the atomization of liquids both in technical applications like engine injectors and natural phenomena like wave breaking. Here, length scales can span from the cm range on the scale of the atomizer down to fractions of μm on the scale of the smallest drops. This represents a tremendous challenge to attempts to simulate in detail the processes occurring during atomization, even on the massively parallel computational resources available today [1].

To study atomization processes in realistic applications, it is thus necessary to introduce some level of modeling of the phase interface dynamics that determine the outcome of the breakup process. From a modeling perspective atomization is often split into two subsequent processes: primary atomization followed by secondary atomization. During primary atomization, a liquid jet or sheet exits the injector, the phase interface deforms, topology change occurs, and separated small and large scale liquid structures are generated. The liquid surface area during primary atomization is usually small whereas the liquid volume loading is large, since large scale compact regions of pure liquid exist. The liquid structures then continue to break up during secondary atomization into ever smaller drops. Due to surface tension forces becoming dominant on small scales, these small scale liquid structures are typically of a simple geometry, i.e. either spherical or ellipsoidal. During secondary atomization, the total liquid surface area is usually large, whereas the liquid volume loading is small.

A wide variety of models has already been developed for the secondary breakup process. These are either based on Lagrangian descriptions of the individual drops or parcels [2–8], or introducing an additional level of modeling, are based on Eulerian methods derived from the Williams spray equation [9] describing the resulting spray's statistics [10,11]. These

E-mail address: marcus.herrmann@asu.edu

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.10.009

models have been very successful in describing flows where their underlying assumptions are met (simple drop geometry, small individual drops). Unfortunately, these assumptions fail in the initial primary atomization region. To overcome this shortcoming, a wide range of phenomenological models aiming to represent statistically the essential features of primary atomization have been proposed. Although conventional phenomenological models introduce the mechanisms of the primary atomization of the liquid jet or sheet, i.e. surface instabilities [6,12–14], drop shedding [15], spontaneous breakup [16], and jet turbulence [17], they still rely on a Lagrangian description using round blobs injected from the nozzle exit and hence neglect all details of the phase interface dynamics and the resulting momentum and energy exchange between the liquid and the gas. As such they typically employ adjustable parameters to tune the atomization outcome to experimentally available data for representative flow conditions, a serious shortcoming to predictive simulations.

Alternatively, some efforts have been made in recent years to simulate the near injector region in detail [1,18–20]. To achieve this, the position, geometry, and motion of the phase interface has to be described accurately, necessitating a tracking of the phase interface. Here the term tracking is used loosely, not only including traditional marker particles or interface surface grids [21], but also encompassing methods that keep track of the position of the phase interface implicitly by solving for additional marker variables, like the Volume of Fluid method [22], or the level set method [23]. In their original versions, each method has its advantages and disadvantages, however, modern enhancements to each of the approaches [24–27] result in comparable accuracy and applicability to the problem of atomization.

However, due the range of length scales present during atomization and the vast number of generated small spray drops and resulting large phase interface area, it is virtually impossibly to track the phase interface from the near injector region all the way to the smallest generated drops far downstream of the injector. Detailed simulations tracking the phase interface are thus limited to the primary atomization region [1].

To simulate in detail the atomization process in its entirety, some form of multi-scale coupling between the tracked phase interface description suitable for the primary atomization region, and the Lagrangian description suitable for the secondary atomization region is thus required. This coupling procedure is the focus of this paper. The resulting simulation strategy tracks the details of the phase interface dynamics during primary atomization in the near injector region and transfers small scale, nearly spherical, separated liquid structures into a standard Lagrangian spray description.

Yet, even following the above outlined approach, computational requirements are extreme, necessitating the use of massively parallel computer systems. It is therefore crucial that the proposed coupling procedure is both highly scalable and efficient.

This paper is structured in the following way. After first introducing the parallel multi-scale Eulerian/Lagrangian coupling procedure, test cases are presented that verify the procedure and demonstrate its overall efficiency and excellent parallel scalability to thousands of processors. Then, an example application of the coupling procedure is presented, namely the atomization of a turbulent liquid jet under diesel engine conditions. Finally, conclusions are presented.

2. Method

In this section, the Eulerian/Lagrangian coupling procedure for the case of a liquid/gas flow whose phase interface is tracked by a level set approach is described. Regions of the computational domain where the level set scalar G is $G > 0$ are denoted as liquid, whereas regions with $G < 0$ are denoted as gas. The purpose of the coupling procedure is to identify separated regions Ω_d of liquid in gas (or gas in liquid) fulfilling certain criteria, to remove these from the level set tracking representation, and to insert them into a Lagrangian framework, preserving their volume V_d, center of mass $\boldsymbol{x}_d$, and momentum $\rho_d \boldsymbol{u}_d$:

$$V_d = \int_{\Omega_d} d\boldsymbol{x}, \quad \boldsymbol{x}_d = \frac{1}{V_d} \int_{\Omega_d} \boldsymbol{x} d\boldsymbol{x}, \quad \boldsymbol{u}_d = \frac{1}{V_d} \int_{\Omega_d} \boldsymbol{u}(\boldsymbol{x}) d\boldsymbol{x}, \tag{1}$$

where ρ_d is the constant liquid density. Removal criteria considered in this paper are based on the derivation assumptions of typical Lagrangian spray models, i.e. that the individual drops be small and nearly spherical. Other transfer criteria are possible, for example criteria based on threshold Weber number, Capillary number, or drop internal fluctuation kinetic energy. These could be used to indicate whether a liquid drop will continue to atomize. In the case of atomization, Lagrangian models for continued atomization of initially spherical drops do exist, see the discussion in the introduction, hence these additional criteria are not employed, here. However, ultimately, the choice of specific transfer criteria depends on the level of complexity of the Lagrangian model employed.

The size criterium is expressed in terms of a threshold volume V_{cut}, in that only those separated liquid structures with volume $V_d \leqslant V_{cut}$ are candidates for removal. The shape criterium will be expressed in terms of an eccentricity measure threshold e_{cut}, where the eccentricity measure e is defined as the normalized maximum distance of a separated liquid structure's phase interface Γ_d to its center of mass,

$$e = \max_{\Gamma_d} \frac{\|\boldsymbol{x} - \boldsymbol{x}_d\|}{\max(\Delta x_G, R_d)} \tag{2}$$

where $R_d = 3/(4\pi)V_d^{1/3}$ and Δx_G is the grid size. The maximum function in the denominator ensures that very small drops at the resolution limit will not result in erroneously large eccentricity measures.

The following sections outlining the coupling algorithms are based on the refined level set grid (RLSG) method introduced in [25]. In addition to a flow solver grid, this method employs a separate, independent Cartesian grid, termed G-grid, for tracking the phase interface using the level set approach. For efficiency a domain decomposed parallel dual-narrow-band approach is employed. The key features of this method important for the proposed algorithm are presented in Fig. 1 and are the following: the dynamically evolving RLSG G-grid is split into individual blocks that are distributed among the processors partaking in the simulation. Each block consists of a Cartesian grid, of which only those nodes that are within a certain distance around the $G = 0$ interface are active and exist. These active nodes form a narrow band surrounding the phase interface. Similarly, only those blocks exist, that contain active nodes, forming a secondary block narrow band surrounding the phase interface. Because the $G = 0$ interface dynamically evolves in time, the location of active nodes and thus the location of active blocks is constantly changing.

It is one of the assumptions in the derivation of standard Lagrangian point particle methods that the volume occupied by the Lagrangian drop be significantly smaller than the local flow solver cell volume V_{fs}, which requires $V_{cut} \ll V_{fs}$. There exists thus an apparent inconsistency since in the pre-transfer tracked representation one would like the drop to be well resolved by the flow solver, implying $V_d \geqslant V_{fs}$, whereas in the post-transfer Lagrangian representation one would like the same drop to be significantly smaller than the local flow solver cell, $V_d \ll V_{fs}$.

There are two different approaches to address this issue. One way is to use non-uniform flow solver meshes. Ideally, one would adaptively adjust the local mesh size in such a way that directly after transfer, the flow solver grid is coarsened to such an extend that the new Lagrangian drop becomes much smaller than the new flow solver grid cell. This is easily achieved by using adaptive mesh refinement (AMR) techniques using the $G = 0$ iso-surface as the refinement indicator. Without an AMR approach for the flow solver, the employed static mesh has to be generated in such a way that the secondary atomization region uses significantly coarser grids than the primary atomization region. This is a natural thing to do, since, after all, the ability to use such a grid layout for efficiency reasons is the primary motivation for the proposed coupling procedure. However, since in most applications, primary and secondary atomization regions are geometrically not clearly separated, in some regions of the flow there will exist Lagrangian drops that can be significantly larger than the local flow solver grid cells. Whether the thus incurred, spatially confined error is of significance depends on the actual application.

Using the RLSG method to track the phase interface offers an additional way to mitigate the potential problem. Because the phase interface is tracked on a separate grid which can be refined independent of the flow solver grid, sub-flow solver phase interface scales can be resolved and tracked. However, in flows that involve surface tension forces, significant refinement of the RLSG grid relative to the flow solver grid by typically more than 64 RLSG grid cell per flow solver grid cell, necessitates the use of additional models to accurately capture their sub-flow solver effect and avoid unphysical interface artifacts [28].

In cases where the size inconsistency does play an important role, extensions to the standard Lagrangian point particle approach can be employed. If the drop is smaller than the local flow solver cell, yet occupies a significant portion of it, volumetric effects can be incorporated, see for example [29]. If the drops is larger than the local flow solver cell, the geometry of the drop can be taken into account in the Lagrangian framework following the approaches in [30–32].

Although this paper focuses in the following on the level set method to track the phase interface, some guidance on how to apply the procedure to other interface tracking methods like the Volume of Fluid method or marker particle tracking are outlined in Section 2.5.

2.1. Identify and tag separated candidate structures

The first step in the coupling procedure is to identify each contiguous liquid structure and tag it by a unique identifier number *id*. To simplify discussion of the parallel, multi-block algorithm, we will first introduce the serial, single block algo-

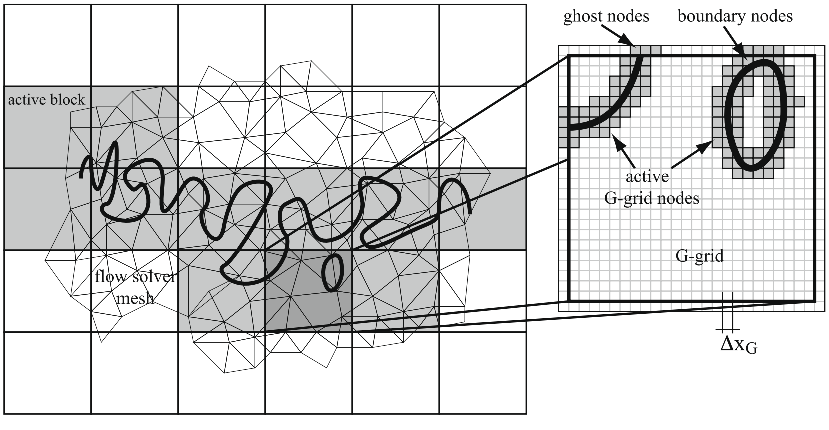


Fig. 1. Refined level set grid dual-narrow-band block and grid structure [25].

```
set $id = 1$
for $i = 1, n_G$ do
    if $G > 0$ and node untagged then
        tag node with $id$ and mark as $\mathcal{S}$
        while any nodes are marked as $\mathcal{S}$ do
            for all nodes marked $\mathcal{S}$ do
                tag all untagged $G > 0$ neighbors with $id$ and mark as $\mathcal{C}$
            end for
            unmark all $\mathcal{S}$ nodes
            mark all $\mathcal{C}$ nodes as $\mathcal{S}$
        end while
        set $id = id + 1$
    end if
end for
```

Fig. 2. Basic serial single block identification algorithm.

rithm, summarized in Fig. 2. Here, n_G is the total number of existing G-nodes in the narrow band structure. The algorithm is based on the band generation algorithm introduced in [25]. After finding the first untagged node inside the liquid, i.e. the first node with $G > 0$, the algorithm proceeds to grow inside the liquid in layers denoted by $\mathscr{S}$ and $\mathscr{C}$ until the continuous structure is filled, see Fig. 3. Although this could be achieved using a recursive algorithm, the slightly more complicated iterative scheme is preferred, in order to avoid the overhead typically associated with recursive function calls. These might otherwise limit the size of the computational domain per processor. After filling the first continuous structure, the *id* tag is increased, and the next untagged node inside the liquid is searched for. To allow for easy, fast access to all nodes belonging to a specific liquid structure, lookup lists are generated during this growth phase.

The above outlined algorithm identifies and tags every node with $G > 0$. However, in the application envisioned here, not all continuous liquid structures are candidates for identification and transfer. Depending on the application, varying acceptance/disqualification conditions can be defined. In the atomization scenario, these include maximum liquid volume and

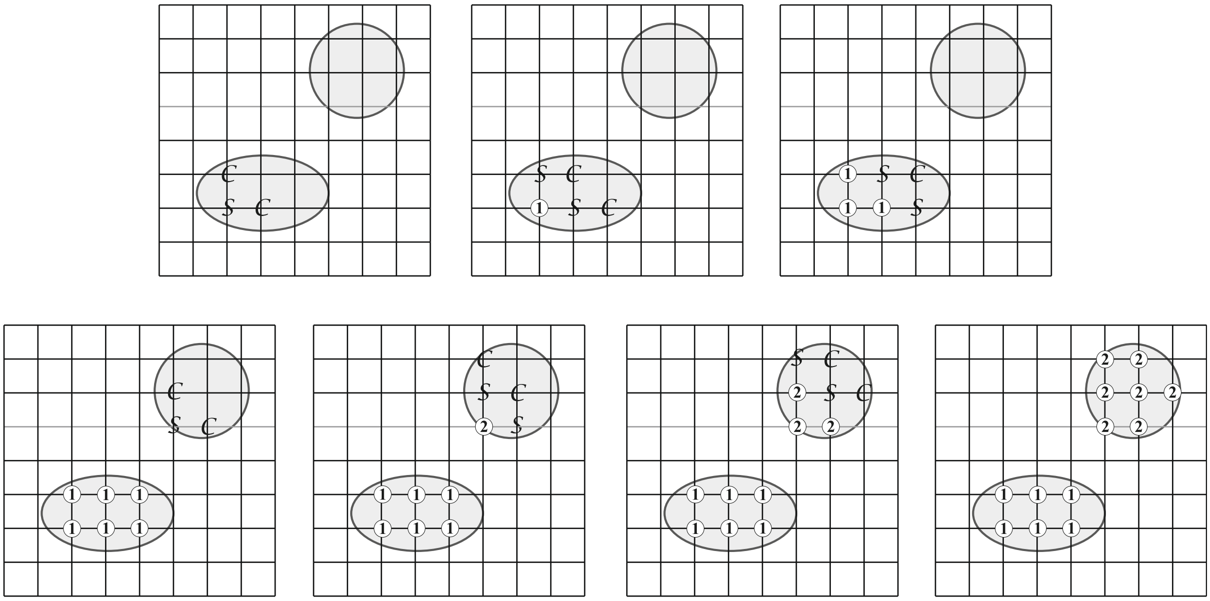


Fig. 3. Basic serial single block identification algorithm steps (from top left to bottom right). Shaded areas indicate regions of $G > 0$, circled numbers are *id*-tag numbers.

shape conditions. To minimize the computational cost, it is imperative to notice and disqualify liquid structures that cannot fulfill the acceptance conditions as early as possible. To achieve this, a running total of the *id* tagged liquid volume, center of mass, and momentum/velocity

$$V_{id} = \sum_{tag_i = id} \psi_i(G) V_{G_i} \qquad \boldsymbol{x}_{id} = \frac{1}{V_{id}} \sum_{tag_i = id} \boldsymbol{x}_i \psi_i(G) V_{G_i} \qquad \boldsymbol{u}_{id} = \frac{1}{V_{id}} \sum_{tag_i = id} \boldsymbol{u}_i \psi(G)_i V_{G_i} \tag{3}$$

is kept during the tag phase (see line 10 of Fig. 4). Here, $\psi_i(G)$ is the cell volume fraction of G-grid cell *i* calculated from the level set field by an analytical formula [33,25], V_{G_i} is volume of the G-grid cell *i*, $\boldsymbol{x}_i$ is that cells centroid coordinate, and $\boldsymbol{u}_i$ is the fluid velocity at that point. Then, if at any time during the tag phase $V_{id} > V_{cut}$, the current drop can be disqualified, by switching all *id* tagged nodes to a tag of $-id$. This is done using the previously generated lookup lists which can then be directly discarded since the continuous structure as a whole has been disqualified. Although tagging with $-id$ still has to continue to exclude the complete continuous liquid structure, additional qualification checks and other drop calculations can then be skipped. Fig. 4 summarizes this enhanced serial single block identification algorithm.

2.2. Multi-block identification and tagging: joining candidate structures

Splitting the single block computational domain into multiple blocks introduces several complications. For one, continuous liquid structures might span several blocks and the above outlined identification algorithm cannot tag them with a un-

```
 1: set $id = 1$
 2: for $i = 1, n_G$ do
 3:   if $G > 0$ and node untagged then
 4:     tag node with $id$ and mark as $\mathcal{S}$
 5:     while any nodes are marked as $\mathcal{S}$ do
 6:       for all nodes marked $\mathcal{S}$ do
 7:         for all untagged $G > 0$ neighbors of current $\mathcal{S}$ node do
 8:           tag node with $id$ and mark as $\mathcal{C}$
 9:           if $id > 0$ then
10:             perform drop calculations for current node
11:             if current drop should be disqualified then
12:               mark all nodes with tag $id$ with tag $-id$
13:               set $id = -id$
14:             end if
15:           end if
16:         end for
17:       end for
18:       unmark all $\mathcal{S}$ nodes
19:       mark all $\mathcal{C}$ nodes as $\mathcal{S}$
20:     end while
21:     if $id > 0$ then
22:       $id = id + 1$
23:     else
24:       $id = -id$
25:     end if
26:   end if
27: end for
```

Fig. 4. Enhanced serial single block identification algorithm.

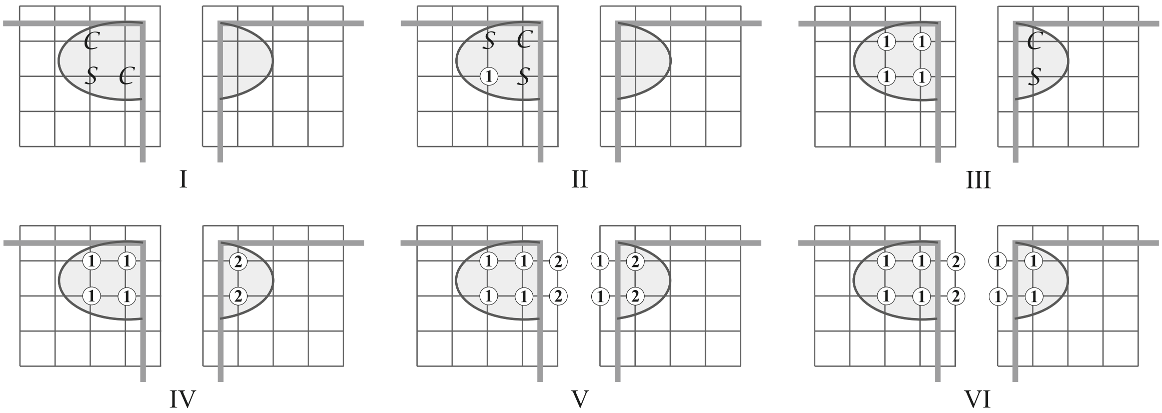


Fig. 5. Basic multi-block identification algorithm. Steps I–IV are the block local steps and steps V and VI are the joining steps. Shaded areas indicate regions of $G > 0$.

ique identifier. Irrespective, to use the identification algorithm in Fig. 4 in the multi-block case, only two modifications are required. First, the loop starting in line 2 has to be replaced by two loops: the outer one over all blocks, and the inner one over all existing G-nodes within the current block. The second modification is due to the fact that the iterative growth algorithm might grow outside the local block structure. Assuming that each block is surrounded by at least one layer of ghost nodes, a potential $\mathscr{C}$ neighbor of an $\mathscr{S}$ node might be inside the first ghost layer. If this is the case, this ghost node will not be marked as $\mathscr{C}$ and no drop calculations will be performed for this node, however it will be added to the lookup list for the current *id*-tagged drop. Fig. 5 depicts in sub-figures I–IV the situation after the modified single block identification algorithm in the multi-block case. The continuous drop split into two by the underlying multi-block structure is tagged by different *id* tag numbers. These drop parts thus have to be joined giving them one unique identifier tag. Note that should a disqualification condition be met in any of the blocks, that local *id* tag will be negative.

Before we discuss how to join the differently tagged parts of the continuous drop, let us first look at what happens if the blocks are distributed among different processors. In that case, each processor will use a locally unique *id* tag. After tagging all $G > 0$ regions with a locally unique *id*-tag, each processor determines a global tag offset and makes all its tags globally unique by adding the processor's tag offset to the local *id* number. To maintain the negative *id*-tag of rejected structures, this is only done for drop tags with $id > 0$.

Fig. 6 summarizes the algorithm to join differently tagged parts across multiple blocks of the same continuous drop. It applies irrespective whether these blocks are local or distributed amongst different processors. We first update the ghost node values of the tag numbers by synching their matching block's internal node tag numbers, see also step V in Fig. 5. If the blocks are local to a single processor, this just involves a local copy, however, if the blocks are distributed amongst different processors, this involves a communication step. Then, for each local block, we loop over the block boundary adjacent nodes with $G > 0$ and $id > 0$ and check if the neighboring ghost cell contains an identifier tag (id_g) smaller than the local node identifier tag (id_b). If this is the case, all block local nodes with tag id_b are changed to the smaller ghost node id_g-tag, see also

```
repeat
    update ghost node $id$-tags of all blocks
    for all local blocks do
        for all block boundary adjacent nodes with $id > 0$ do
            if ghost node $id_g < id_b$ of boundary adjacent node then
                for all block nodes with $id = id_b$ do
                    set tag to $id_g$
                end for
            end if
        end for
    end for
until globally, no drops changed their $id$-tag
```

Fig. 6. Parallel, multi-block tag synchronization algorithm.

step VI in Fig. 5. We use the previously generated lookup lists to do this task efficiently. If any drops changed their *id*-tag in this manner, we need to redo the above process until globally, no drops changed their *id*-tags. Numerical experiments have shown that in all our test cases and applications to date, only between 2 and a maximum of 6 of these "redos" are required.

2.3. Gathering and removing structures

After the completion of the joining step, continuous liquid structures split across arbitrarily many blocks and processors will have a common unique identifier tag *id*, however, their properties like volume or center of mass are still split amongst blocks and processors. Due to the unique identifier, it is however easy to join each structure's volume, center of mass, and momentum/velocity by simply performing a global reduce operation, yielding V'_{id}, $\boldsymbol{x}'_{id}$, and $\boldsymbol{u}'_{id}$. Next, using V'_{id}, an additional disqualification check using the threshold volume V_{cut} is performed, further narrowing the list of potential transfer candidates. Then, each joined structure's eccentricity measure can be estimated as

$$e_{id} = \max_{tag_i=id} \frac{\|\boldsymbol{x}_i - \boldsymbol{x}_{id}\|}{\max(\Delta x_{G_i}, R_{id})}, \tag{4}$$

where the maximum is taken over all nodes belonging to the *id*-tagged structure. Note that this is a first order approximation to Eq. (2), allowing a further narrowing of the list of candidate structures to those with $e_{id} \leqslant e_{cut}$.

Unfortunately, however, V'_{id}, $\boldsymbol{x}'_{id}$, and $\boldsymbol{u}'_{id}$ do not accurately reflect the structures' true values and are only estimates for the following reason. The identifier algorithm outlined in Section 2.1 considers only nodes with $G > 0$, and thus all drop properties are calculated using only these nodes. However, as can be seen from Fig. 7, even nodes with $G < 0$ can contribute to the liquid volume of the drop and thus must not be neglected.

Simply including neighbor $G < 0$ nodes of the drop into the volume calculation, Eq. (3), will erroneously include liquid patches belonging to other drops, as can be seen in the example of Fig. 7 for the non-shaded region of the right panel. The way to avoid this problem is to follow the algorithm outlined in Fig. 8. It is based on the simple fact that the drop's liquid volume has to be equal to the difference of the total liquid volume before and after the drop's removal from the domain. To remove the drop from the domain, it is sufficient to simply switch the sign of all $G > 0$ nodes belonging to the drop to $-G$. This can be done efficiently by using the lookup tables generated in the identification phase. Note that these lookup tables

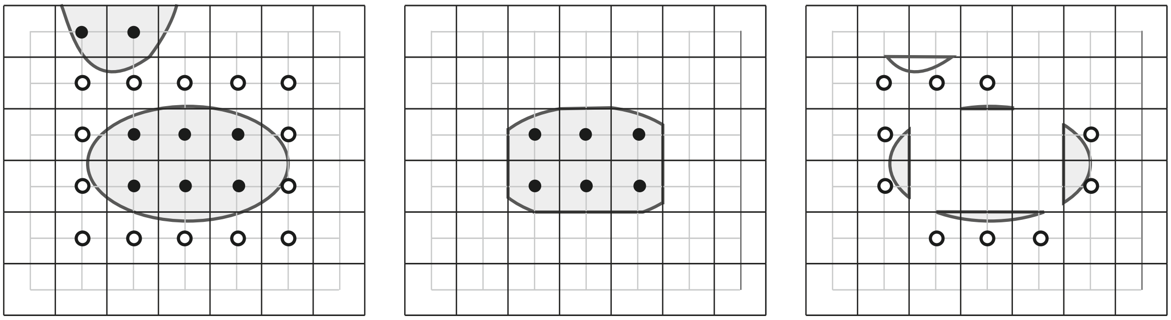


Fig. 7. Split of liquid volume of a drop between $G > 0$ nodes (solid circles) and $G < 0$ nodes (open circles).

```
calculate total liquid volume in each block $V_{bl}$
for each candidate structures do
    for all nodes belonging to current structure do
        set $G$ value of node to $-|G|$
    end for
    calculate liquid volume in each block $V'_{bl}$
    processor local liquid volume of structure is $V_d = \sum_{bl} V_{bl} - V'_{bl}$
    set $V_{bl} = V'_{bl}$
end for
sum structure volumes across processors
```

Fig. 8. Parallel algorithm to determine candidate structure liquid volume.

include $G > 0$ nodes in the ghost cell layer of each block exactly for the reason of being able to switch their sign here. This allows for an accurate liquid volume calculation inside each block, even if only a small part of the drop reaches within it. A final global reduce operation can then determine the liquid structures true V_{id}, $\boldsymbol{x}_{id}$, and $\boldsymbol{u}_{id}$.

While it is possible to perform an additional, now accurate, acceptance/disqualification check against the structure's true values, a disqualification at this stage would require the re-insertion of the already removed structure into the level set representation, by switching back all the structure's prior $G > 0$ nodes, now set to $-G$, back to their original G values. In numerical experiments and actual applications to atomization cases, however, it was found that the difference between V_{id} and V'_{id} as well as e_{id} and e is small and decreases under grid refinement, see for example test cases d4g1 and d4g2 in Section 3.1. In light of the fact that in actual applications to atomization simulations, the chosen threshold values are estimates, it is thus legitimate to base the disqualification condition on the close, but not exact estimates V'_{id} and e_{id}. However, to preserve liquid volume/mass during transfer, the true V_{id} has to be used for the inserted Lagrangian drop, necessitating the use of the above algorithm.

2.4. Lagrangian to Eulerian transfer

The prior sections discuss the transfer of Eulerian tracked liquid structures into the Lagrangian description. For completeness, the following briefly outlines the reverse case which might be necessary because of drop growth due to Lagrangian drop/drop collisions, Lagrangian drop collision with tracked structures, indicated by

$$G(\boldsymbol{x}_d) < R_d, \tag{5}$$

and situations where Lagrangian drops migrate from flow solver grid regions of low resolution, where the drop's radius is smaller than the local grid spacing, into regions of high resolution, where the drop's radius becomes significantly larger than the local flow solver grid spacing.

Transfer from the Lagrangian to the Eulerian description of the mass of a drop with radius R_d and center of mass $\boldsymbol{x}_d$ can be achieved by simply re-evaluating the local level set field as

$$G(\boldsymbol{x}) = \max(G(\boldsymbol{x}), R_d - \|\boldsymbol{x} - \boldsymbol{x}_d\|). \tag{6}$$

Transfer of the drop's momentum is a more challenging task. The simplest possible model is to use the solid sphere assumption inherent in many Lagrangian point particle methods by setting

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_d \ \forall \ \|\boldsymbol{x} - \boldsymbol{x}_d\| \leqslant R_d. \tag{7}$$

While this does conserve momentum during transfer, it relies on the pre-transfer gaseous velocities to be consistent with the solid sphere assumption and adequately resolved. This can be ensured by using the Lagrangian point particle extensions described in [30–32].

More advanced models for the post-transfer velocity field can be developed taking for example the internal circulation of the drop or the drag induced gaseous flow field around the drop into account. These models are more challenging and are left to future studies.

2.5. Application to other tracking methods

The algorithms presented in this paper are based on tracking the phase interface using a level set method and rely on the fact that the presence of the phase interface is identifiable by a sign change in the level set scalar value. One option to use the presented methods in the case of Volume of Fluid or marker particle based interface tracking is thus to first reconstruct a binary marker function on a fixed grid. The identification and joining algorithms of Sections 2.1 and 2.2 can then be used directly as presented. In the case of the Volume of Fluid method, running totals of each structure's volume can be directly calculated using the Volume of Fluid scalar, whereas in the case of marker particles, appropriate volume calculation methods have to be employed.

In the gathering step, Section 2.3, one complication arises in identifying a drop's liquid volume associated with non-positive marker function values, see Fig. 7. While level set methods offer an easy way to deal with this, Volume of Fluid methods would have to use geometric reconstruction like PLIC to identify which volume fraction $\psi > 0$ belongs to which separate liquid structure.

3. Results

3.1. Test cases

This section presents a range of artificial test cases to verify the proposed methodology and to ascertain its performance and parallel scalability. Eight different cases are analyzed using four different interface geometries. In each, a unit sized cube is considered in which N spheres are randomly distributed between $0.1 < \boldsymbol{x} < 0.9$.

Interface geometries d1, d2, and d3 consist of $N_{d1} = 1000$, $N_{d2} = 5000$, and $N_{d3} = 25,000$ monodisperse spheres of radius $R = 1/128$, whereas geometry d4 consists of 5000 polydisperse spheres with random radius between $1/512 \leqslant R \leqslant 1/32$. The resulting interface geometries are shown in the left column of Fig. 9.

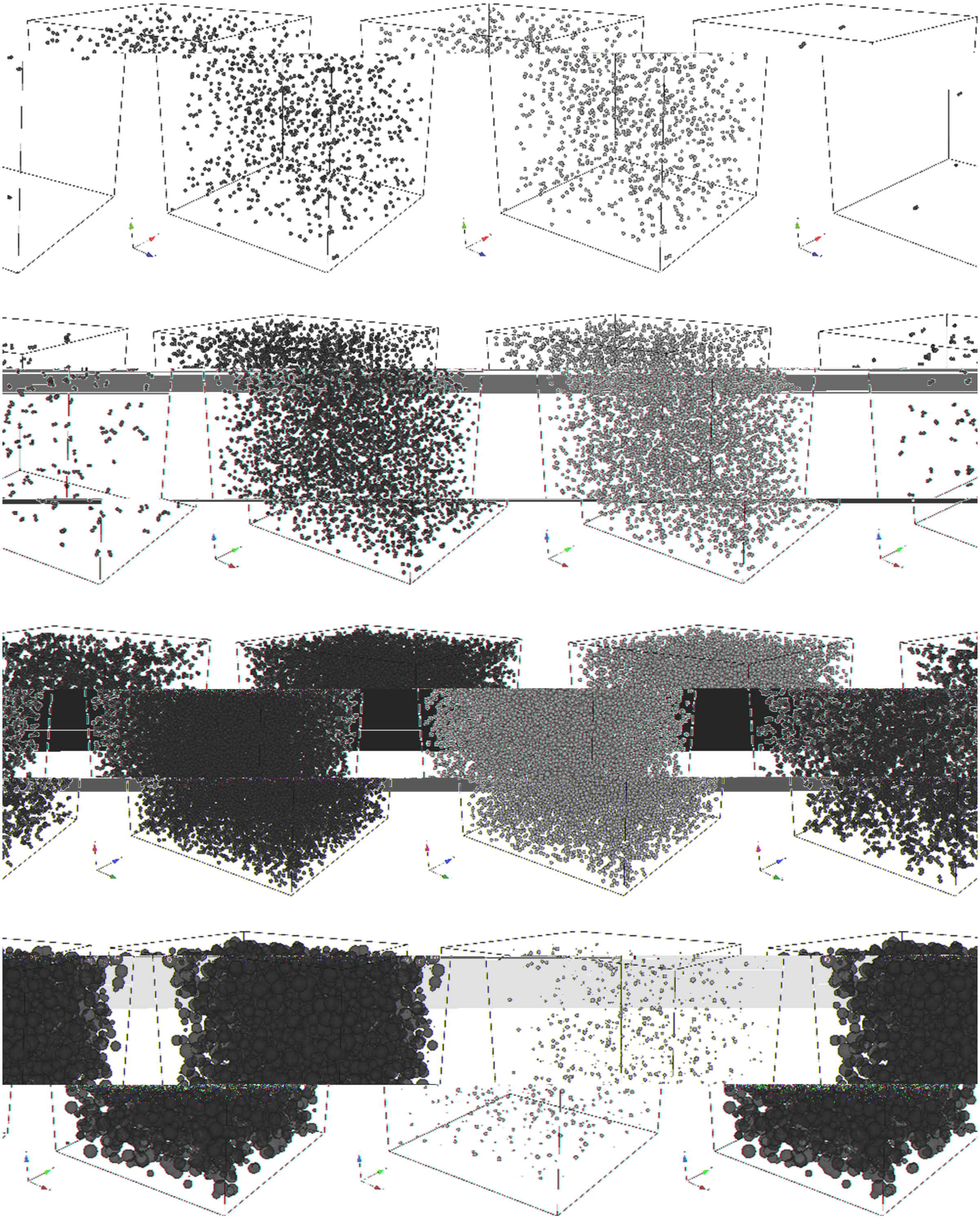


Fig. 9. Tracked phase interface geometry before (left) and after (right) drop identification and identified Lagrangian drops (center) for cases d1g1, d2g1, d3g1, and d4g1 (from top to bottom).

The total number of separated liquid structures M in each case is less than the number of seeded spheres N. This is due to the fact that the randomly placed spheres can result in partial overlapping, creating dumbbell structures in the case of 2 spheres and ever more complex structure for increasing numbers of partially overlapping spheres. Table 1 thus gives the number of isolated spheres N_1, dumbbell structures N_2, and agglomerations resulting from 3 (N_3) and more $(N_{>3})$ spheres for the geometry d1 that will be used for verification of the algorithm.

The four different interface geometries are each analyzed on two different G-grids. In the first, g1, the unit sized cube is resolved by G-grid cells of size $\Delta x_G = 1/1024$, in the second, g2, by G-grid cells of size $\Delta x_G = 1/2048$. Block sizes in the RLSG method are 32^3 and 64^3, respectively. Table 2 gives the resulting number of active G-grid nodes in the RLSG method in millions for each of the eight cases, ranging from 34 million up to 2.8 billion.

The threshold radius R_{cut} defining the threshold identification volume, $V_{cut} = 4/3\pi R_{cut}^3$, in sphere distributions d1, d2, and d3 is $R_{cut} = 0.02$, and $R_{cut} = 0.01$ in distribution d4. The eccentricity measure threshold in all cases is $e_{cut} = 1.5$.

Fig. 9 shows the initial interface geometry of the seeded spheres, the identified and transferred Lagrangian drops, and the remaining tracked interface geometry after identification and removal. Tables 3 and 4 quantify the corresponding identified and transferred drops. Considering cases d1g1 and d1g2 (the difference being only the employed G-grid resolution) presented in detail in Table 3, out of the initially seeded 982 separated structures, only 967, respectively 971 are identified and removed. In the d1g1 case, 953 out of 965 single sphere structures, 13 out of 16 dual-sphere structures, and the single triple-sphere structure are removed. In addition 9 dual-sphere structures are not removed, which are shown in the top right of Fig. 9. What is the reason that not all seeded single and dual-sphere structures are identified and removed?

The reason lies in the way fixed grid methods, like the level set method, handle topology changes. In the level set method, as soon as two interfaces enter into the same cell, the topology is changed automatically due to the fact that the level set scalar is single valued. While this is a desired behavior in most applications, it is a limiting discrepancy in some. To avoid it, multiple level set scalars would have to be defined. In the d1 distribution, a total of 6 sphere pairs have surfaces less than the g1 grid resolution apart. Thus, although theoretically these result in 12 separate spheres, in the numerical g1 level set

Table 1
Initial sphere distributions for geometry d1, with N the total number of seeded spheres, N_i the resulting number of structures consisting of i spheres, and M the resulting total number of separated structures.

	N	N_1	N_2	N_3	$N_{>3}$	M
d1	1000	965	16	1	0	982

Table 2
Number of active G-grid nodes in millions for each case.

	Δx_G	d1	d2	d3	d4
g1	1/1024	34	155	495	410
g2	1/2048	162	763	2809	2736

Table 3
Drop identification verification results: number of structures removed (m) and seeded (M), number of single sphere structures removed (n_1) and seeded (N_1), number of multi-sphere structures removed $(n_{>1})$ and seeded $(N_{>1})$, and number of disqualified structures n_{DQ}.

Case	m	M	n_1	N_1	n_2	N_2	n_3	N_3	n_{DQ}
d1g1	967	982	953	965	13	16	1	1	9
d1g2	971	982	957	965	13	16	1	1	7

Table 4
Drop identification results: total number of removed (m) and seeded (M) structures, minimum (R_{min}) and maximum (R_{max}) equivalent volume sphere radius of removed structures, and wall time $t_{n_p=256}$ used to identify and remove the liquid structures using 256 processors.

Case	m	M	$R_{min} \times 10^3$	$R_{max} \times 10^3$	$t_{n_p=256}$ (s)
d1g1	967	982	7.7965	11.027	0.0796
d1g2	971	982	7.8091	11.033	0.3624
d2g1	4437	4627	7.7965	11.138	0.4171
d2g2	4469	4627	7.8091	11.140	1.8653
d3g1	12753	17459	7.7947	13.046	1.9265
d3g2	13305	17459	7.8090	13.203	11.6015
d4g1	753	2433	1.9641	10.105	0.8696
d4g2	760	2433	1.9598	10.032	5.7023

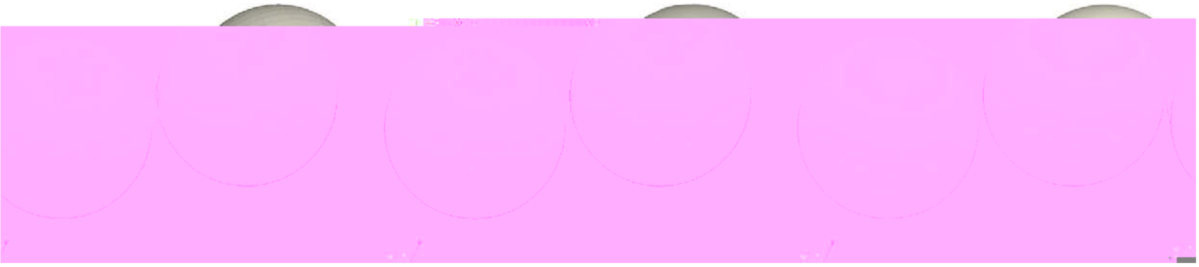


Fig. 10. Impact of grid resolution on interface topology of two close spheres: dumbbell structure on coarse grid for d1g1 (left), separated spheres on fine grid for d1g2 (center), and transferred Lagrangian drops for d1g2 (right).

representation, they are 6 joined dumbbells, see Fig. 10. The number of existing single sphere structures on the g1 grid is thus not 965, but instead $965 - 12 = 953$, the exact number of identified and removed single sphere structures. The correct number of dual-sphere structures on the g1 grid is then 16 + 6 = 22, again the exact number of removed and disqualified dual-sphere structures (13 + 9). A consequence of this is that the number of removed single sphere structures is grid dependent. Indeed, the finer g2 grid does not merge 2 of the pairs, see Fig. 10 for an example, and thus results in 4 more single sphere structures and 2 less disqualified dual-sphere structures. It should be pointed out that it is in fact impossible to achieve grid independency in the general case, since the distance between surfaces belonging to different liquid structures can be infinitesimal small, requiring infinite grid resolution.

Next, let us consider the disqualified structures. They are disqualified not because their volume exceeds the threshold volume, since it would take at least 16 partially overlapping spheres to disqualify a structure based on this criterium. Instead, they are disqualified because they exceed the essentricity measure threshold. To explain this, consider two spheres that are just touching. The joined structure will have an essentricity measure of $e = 2/2^{1/3} = 1.587$ exceeding the threshold value of $e_{cut} = 1.5$. Indeed, as can be seen from Figs. 9 and 10, all 9 non-transferred dumbbell structures fall into this category. Note that these 9 structures include the aforementioned 6 artificially joined dumbbells for the g1 grid. In the g2 grid case, two of these are no longer joined, they are thus no longer disqualified and are instead transferred as 4 single sphere structures, see Fig. 10 for an example.

The above results are thus consistent with the prescribed initial seeding and thus verify the implementation of the proposed algorithm.

Table 4 summarizes the results for all eight cases, listing the number of removed and seeded structures, m and M, respectively, the minimum and maximum radius of the resulting Lagrangian drops, and the wall time taken to identify and remove the liquid structures using 256 processors. Again, due to the above outlined arguments, the total number of identified and removed structures can be significant less than the number of initially seeded structures. In the d2 and d3 cases, this is for

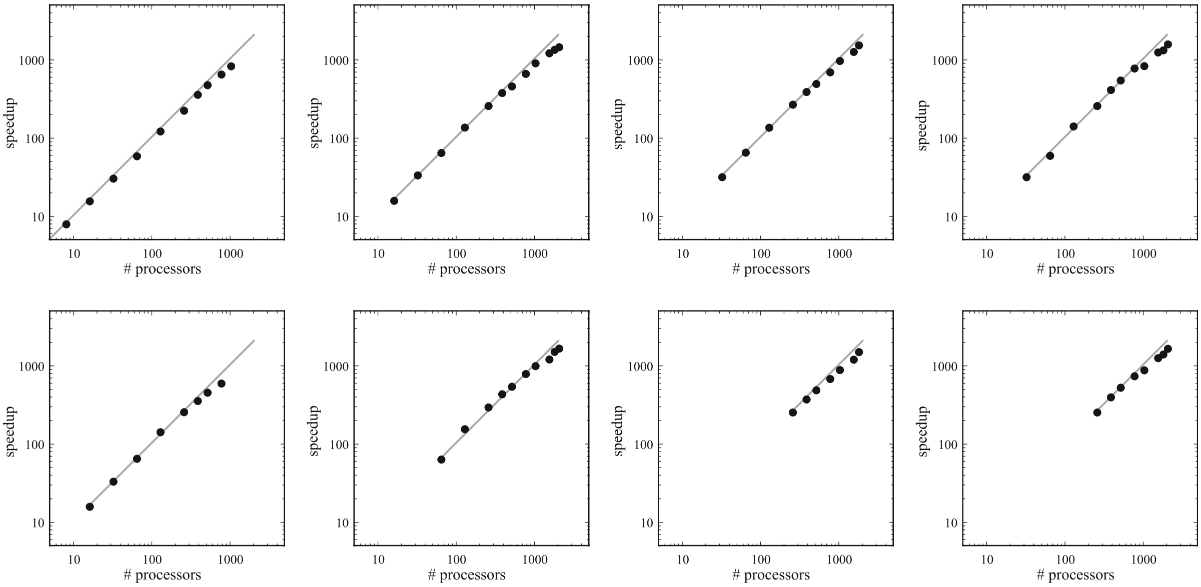


Fig. 11. Observed parallel speedup for grid g1 (top row) and grid g2 (bottom row) and distribution d1, d2, d3, and d4 (from left to right) with respect to the lowest number of processors the individual case would run on. Line denotes optimum scaling.

the most part due to the large number of structures violating the eccentricity measure constraint. In the d4 distribution, structures are mostly disqualified since they violate the volume constraint. In fact, only 1387 out of the initially seeded 5000 spheres of the d4 distribution have a radius less than R_{cut} and would be candidates for transfer.

Due to the employed second order accurate volume calculation algorithm [33,25] the obtained R_{min} values are not equal to the seeded minimum sphere radii, but rather approach it with second order convergence. The maximum radius R_{max} in the d4 distribution cases, exceeds the threshold value of R_{cut} slightly for the reasons discussed in the previous section. The overshoot is grid dependent and converges in the shown manner under grid refinement.

Finally, Fig. 11 shows the observed parallel speedup for each of the eight cases. Results within each case are scaled assuming optimum speedup for the smallest number of cores the case would run on. Results were obtained using Arizona State University's HPCI cluster Saguaro2, employing Intel EM64T Xeon E54xx (Harpertown) quad-core processors running at 2830 MHz using Infiband interconnects and 2 GB of memory per core. Actual wall clock execution times running on 256 cores are listed in Table 4. Excellent scalability is observed up to the maximum number of cores, 2048, that were available

Table 5
Operating conditions.

Gas density ρ_g	25 kg/m^3
Gas viscosity μ_g	1.78×10^{-5} kg/ms
Liquid density ρ_l	850 kg/m^3
Liquid viscosity μ_l	1.7×10^{-3} kg/ms
Surface tension coefficient σ	0.05 N/m
Injector diameter D	100 μm
Mean injection velocity U	100 m/s
Injection Reynolds number Re	5000
Injection Weber number We	17000
Computational domain	$21D \times 4D \times 2\pi$
Minimum Δx flow solver	$D/100 = 1$ μm
Δx_G RLSG solver	$D/256 = 0.39$ μm
Flow solver grid size	24 million
RLSG max active nodes	403 million

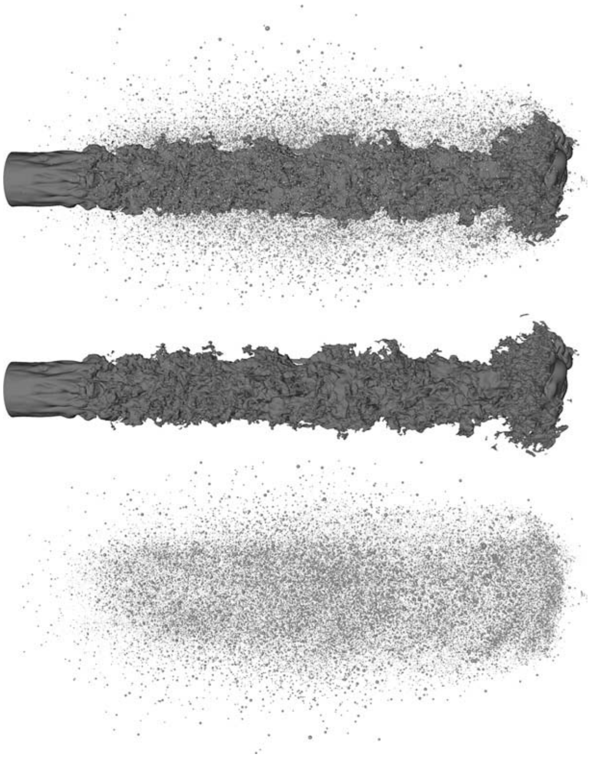


Fig. 12. Atomization of turbulent liquid jet 10 μs after start of injection. Liquid core and spray (top), tracked phase interface of liquid core (center), Lagrangian tracked spray (bottom).

for benchmarking. In the most demanding test case d3g2 involving more than 2.8 billion active *G*-nodes, wall clock execution time on 2048 cores is less than 2 s, demonstrating the efficiency of the proposed algorithm and its suitability for large scale simulations.

3.2. Detailed simulation of the atomization of a turbulent liquid jet

To demonstrate the applicability and performance of the proposed methodology to relevant engineering applications, this section presents results of a detailed simulation of the primary atomization of a turbulent liquid jet injected into still compressed air under conditions relevant to diesel engine injection systems [18]. The operating conditions, computational domain, and grid resolution for this case are reported in Table 5.

Drop identification, removal from the level set tracked phase interface representation, and injection into the Lagrangian spray model is performed at every time step. Details about the employed flow solver NGA can be found in [34], whereas the employed RLSG level set solver and the required coupling procedures between the two solvers are presented in [25]. The motion of the Lagrangian spray drops is described taking two-way momentum coupling between the discrete and continuous phase into account by including a standard solid sphere drag force model [35], see [36] for further details. Neither volumetric effects nor enhancements to the Lagrangian framework for drops larger than the local flow solver cells are taken into account. Also, secondary atomization of the Lagrangian drops is not considered.

Figs. 12–14 show the atomizing liquid jet, the transferred Lagrangian drops, and the remaining tracked phase interface geometry at 10, 15, and 20 μs after injection start. At 20 μs after injection, roughly 403,000 drops have been transferred into the Lagrangian spray model. Drop diameters of these transferred drops range from the sub-micron size up to 10.8 μm. This is significantly below the employed cut-off diameter, due to the fact that large drops are characterized by relatively large Weber numbers and thus are typically highly non-spherical. The eccentricity threshold thus prevents their transfer and they instead continue to atomize in the tracked representation until surface tension forces are sufficient to pull the resulting drops into near spherical shapes.

It should be pointed out that the use of the eccentricity measure threshold is critical in order to capture certain atomization mechanisms. In the present simulation, atomization occurs to a large extend via the formation of ligaments stretching

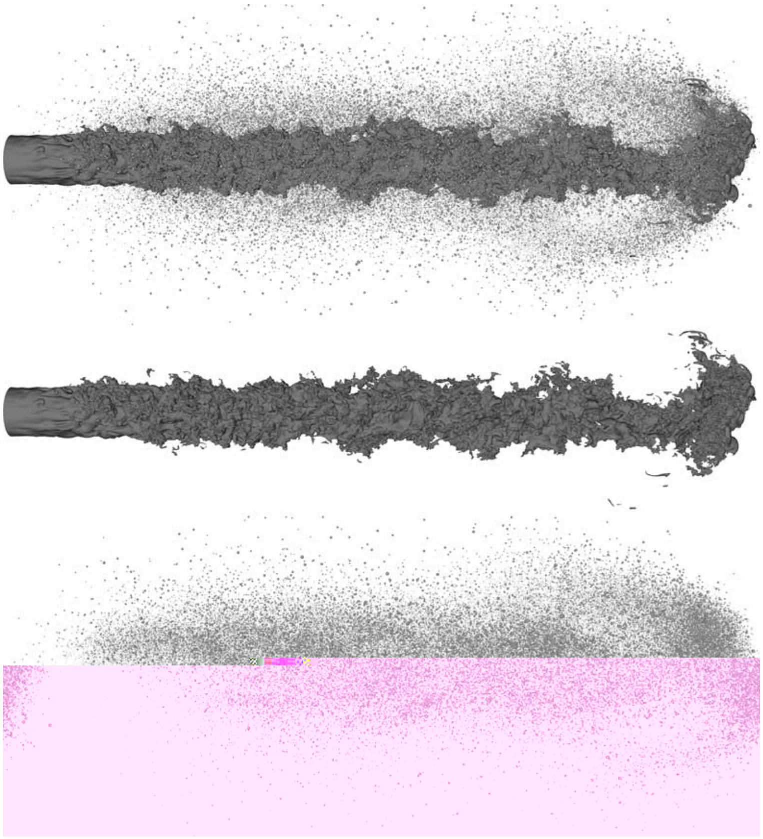


Fig. 13. Atomization of turbulent liquid jet 15 μs after start of injection. Liquid core and spray (top), tracked phase interface of liquid core (center), Lagrangian tracked spray (bottom).

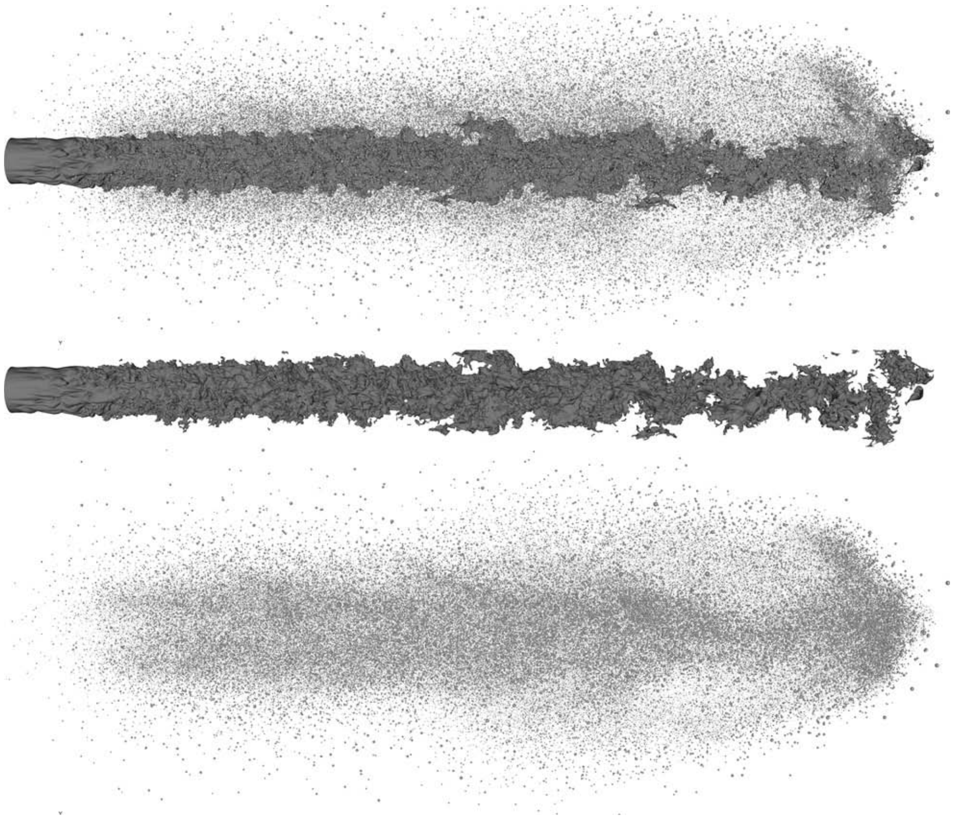


Fig. 14. Atomization of turbulent liquid jet 20 μs after start of injection. Liquid core and spray (top), tracked phase interface of liquid core (center), Lagrangian tracked spray (bottom).

from the liquid surface, see Figs. 12–14. These typically separate from the liquid core and continue to undergo atomization reminiscent of the Rayleigh mechanism. Since the volume contained in some of the separated liquid ligaments prior to their continued breakup is small, they would be transferred into the Lagrangian description as a sphere were it not for the eccentricity threshold measure that prevents their transfer. If transferred, the conjectured Rayleigh breakup would have to be fully modeled. Instead, breakup can occur in the resolved description and does not require additional modeling.

Near the end of the simulation, when the tracked phase interface reaches its maximum extend, the total compute time per time step needed to identify and remove separated structures is less than 0.9% of the total compute time of a single time step. At earlier times, the required compute time is significantly less, since the tracked phase interface area is significantly smaller. This demonstrates the high efficiency of the proposed method and its applicability to realistic problems.

4. Conclusions

In this paper we presented an efficient parallel multi-scale coupling procedure between a Eulerian level set method tracking an interface and a standard Lagrangian description of small scale, separated structures. The proposed method shows excellent scalability, even to thousands of processors. It is applicable to large scale simulations involving billions of active level set grid points, as demonstrated by the presented test cases. The proposed method has been applied successfully to the detailed simulation of the primary atomization of a turbulent liquid jet under Diesel engine conditions, incurring only a minimal overhead, but enabling the detailed simulation of the phase interface dynamics during primary atomization of turbulent liquid jets generating hundreds of thousands of spray drops.

Acknowledgments

This work was supported in part by CASCADE Technologies Inc. under the NavAir SBIR N07-046.

References

[1] M. Gorokhovski, M. Herrmann, Modeling primary atomization, Ann. Rev. Fluid Mech. 40 (1) (2008) 343–366, arXiv: <http://arjournals.annualreviews.org/doi/pdf/10.1146/annurev.fluid.40.111406.102200>, doi: 10.1146/annurev.fluid.40.111406.102200, URL: <http://arjournals.annualreviews.org/doi/abs/10.1146/annurev.fluid.40.111406.102200>.

[2] S.V. Apte, M. Gorokhovski, P. Moin, Les of atomizing spray with stochastic modeling of secondary breakup, Int. J. Multiphase Flow 29 (9) (2003) 1503–1522.

[3] M. Gorokhovski, The stochastic sub-grid-scale approach for spray atomization, Atom. Sprays 11 (2001) 505–519.

[4] P.J. O'Rourke, Collective drop effects on vaporizing liquid sprays, Ph.D. Thesis, Princeton University, 1532-T, 1981.

[5] P.J. O'Rourke, A.A. Amsden, The TAB method for numerical calculation of spray droplet breakup, Tech. Rep. 872089, Los Alamos National Lab, 1987.

[6] R.D. Reitz, Modeling atomization processes in high-pressure vaporizing sprays, Atom. Spray Tech. 3 (1987) 309–337.

[7] R.D. Reitz, R. Diwakar, Structure of high pressure fuel sprays, Tech. Rep. 870598, SAE Technical Paper, 1987.

[8] F.X. Tanner, Liquid jet atomization and droplet breakup modeling of non-evaporating diesel fuel sprays, SAE Trans.: J. Engines 106 (1998) 127–140.

[9] F.A. Williams, Spray combustion and atomization, Phys. Fluids 1 (1958) 541–545.

[10] F. Laurent, M. Massot, P. Villedieu, Eulerian multi-fluid modeling for the numerical simulation of coalescence in polydisperse dense liquid sprays, J. Comput. Phys. 194 (2) (2004) 505–543. URL: <http://www.sciencedirect.com/science/article/B6WHY-4B0P7GB-2/2/4da42338c1947bba379b506155-fb91c6>.

[11] R.O. Fox, F. Laurent, M. Massot, Numerical simulation of spray coalescence in an Eulerian framework: direct quadrature method of moments and multi-fluid method, J. Comput. Phys. 227 (6) (2008) 3058–3088. URL: <http://www.sciencedirect.com/science/article/B6WHY-4R6JP3B-1/2/1a00fbfb7dc144-f7b8621d5dfd0d13d2>.

[12] J.C. Beale, R.D. Reitz, Modeling spray atomization with the Kelvin–Helmholtz/Rayleigh–Taylor hybrid model, Atom. Sprays 9 (1999) 623–650.

[13] C. Habchi, T. Baritaud, Modeling atomization and break-up in high pressure diesel sprays, SAE Technical Paper, 9700881.

[14] M.A. Patterson, R.D. Reitz, Modeling the effects of fuel characteristics on diesel engine combustion and emission, SAE Paper, 980131.

[15] Y. Yi, R.D. Reitz, Modeling the primary breakup of high-speed jets, Atom. Spray 14 (2004) 53–80.

[16] F.X. Tanner, Development and validation of a cascade atomization and drop breakup model for high-velocity dense sprays, Atom. Sprays 164 (2004) 211–242.

[17] K.Y. Huh, A.D. Gosman, A phenomenological model of diesel spray atomization, in: Proc. Int. Conf. Multiph. Flow, Tsukaba, Japan, 1991, pp. 515–518.

[18] M. Herrmann, On simulating primary atomization using the refined level set grid method, in: ILASS Americas 21st Annual Conference on Liquid Atomization and Spray Systems, Orlando, FL, 2008.

[19] G.M. Bianchi, P. Pelloni, S. Toninel, R. Scardovelli, A. Leboissetier, S. Zaleski, 3d large scale simulation of the high-speed liquid jet atomization, SAE Technical Paper, 2007-01-0244.

[20] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416. URL: <http://www.sciencedirect.com/science/article/B6WHY-4SRCJWK-1/2/a8c8127ed57af20851bac3a9f75076c9>.

[21] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.

[22] D. Gueyffier, J. Li, A. Nadim, S. Scardovelli, S. Zaleski, Volume of Fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.

[23] M. Sussman, P. Smereka, S. Osher, A level set method for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146.

[24] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part I: numerical method, Int. J. Heat Mass Trans. 47 (2004) 5451–5461.

[25] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, J. Comput. Phys. 227 (4) (2008) 2674–2706.

[26] Y. Renardy, M. Renardy, Prost: A parabolic reconstruction of surface tension for the Volume of Fluid method, J. Comput. Phys. 183 (2002) 400–421.

[27] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2) (2007) 469–505.

[28] M. Herrmann, M. Gorokhovski, An outline of a LES subgrid model for liquid/gas phase interface dynamics, in: Proceedings of the 2008 CTR Summer Program, Center for Turbulence Research, Stanford University, Stanford, CA, 2008, pp. 171–181.

[29] S.V. Apte, K. Mahesh, T. Lundgren, A Eulerian–Lagrangian model to simulate two-phase/particulate flows, in: Annual Research Briefs-2003, Center for Turbulence Research, Stanford University, 2003, pp. 161–171.

[30] T. Kajishima, S. Takiguchi, H. Hamasaki, Y. Miyake, Turbulence structure of particle-laden flow in a vertical plane channel due to vortex shedding, JSME Int. J. Ser. B 44 (4) (2001) 526–535.

[31] T. Kajishima, S. Takiguchi, Interaction between particle clusters and particle-induced turbulence, Int. J. Heat Fluid Flow 23 (5) (2002/10) 639–646, URL: <http://www.sciencedirect.com/science/article/B6V3G-45FH2GT-5/2/35dc5e9185c27bf66f6b28d4953075f7>.

[32] S.V. Apte, A multilevel formulation to simulate particulate flows, in: Annual Research Briefs-2004, Center for Turbulence Research, Stanford University, 2004, pp. 201–208.

[33] S.P. van der Pijl, A. Segal, C. Vuik, A mass-conserving level-set method for modelling of multi-phase flows, Int. J. Numer. Meth. Fluids 47 (2005) 339–361.

[34] O. Desjardins, G. Blanquart, G. Balarac, H. Pitsch, High order conservative finite difference scheme for variable density low Mach number turbulent flows, J. Comput. Phys. 227 (15) (2008) 7125–7159.

[35] C. Crowe, M. Sommerfield, Y. Tsuji, Multiphase Flows with Droplets and Particles, CRC Press, 1998.

[36] P. Moin, S.V. Apte, Large-eddy simulation of realistic gas turbine combustors, AIAA J. 44 (4) (2006) 698–708.