

# Coalescence Prevention Algorithm for Level Set Method

**Matthew L. Talley**

Nuclear Engineering Department,  
North Carolina State University,  
3140 Burlington Engineering Labs,  
2500 Stinson Drive,  
Raleigh, NC 27695  
e-mail: mltalle2@ncsu.edu

**Matthew D. Zimmer**

Nuclear Engineering Department,  
North Carolina State University,  
3140 Burlington Engineering Labs,  
2500 Stinson Drive,  
Raleigh, NC 27695  
e-mail: mdzimmer@ncsu.edu

**Igor A. Bolotnov**

Nuclear Engineering Department,  
North Carolina State University,  
3140 Burlington Engineering Labs,  
2500 Stinson Drive,  
Raleigh, NC 27695  
e-mail: Igor\_bolotnov@ncsu.edu

*An algorithm to prevent or delay bubble coalescence for the level set (LS) method is presented. This novel algorithm uses the LS method field to detect when bubbles are in close proximity, indicating a potential coalescence event, and applies a repellent force to simulate the unresolved liquid drainage force. The model is introduced by locally modifying the surface tension force near the liquid film drainage area. The algorithm can also simulate the liquid drainage time of the thin film by controlling the length of time the increased surface tension has been applied. Thus, a new method of modeling bubble coalescence has been developed. Several test cases were designed to demonstrate the capabilities of the algorithm. The simulations, including a mesh study, confirmed the abilities to identify and prevent coalescence as well as implement the time tracking portion, with an additional 10–25% computational cost. Ongoing tests aim to verify the algorithm's functionality for simulations with different flow conditions, a ranging number of bubbles, and both structured and unstructured computational mesh types. Specifically, a bubble rising toward a free surface provides a test of performance and demonstrates the ability to consistently prevent coalescence. In addition, a two bubble case and a seven bubble case provide a more complex demonstration of how the algorithm performs for larger simulations. These cases are compared to much more expensive simulations capable of resolving the liquid film drainage (through very high local mesh resolution) to investigate how the algorithm replicates the liquid film drainage process.*

[DOI: 10.1115/1.4036246]

## Introduction

Many engineering problems involve multiphase bubbly flows, and the recent growth in high-performance computing capabilities enables the evaluation of a wider range of problems using high fidelity simulation approach. The interface tracking approach with direct numerical simulation (DNS) of turbulence may not yet allow the prediction of large systems, but it represents a valuable tool in developing closure laws for multiphase computational fluid dynamics (M-CFD) models. Massively parallel finite element based code, PHASTA [1], is an advanced flow solver that can utilize the level set (LS) method to model fully resolved bubbly flows. However, the standard LS approach [2] lacks the capability to represent the physics of bubble coalescence.

In the presented research, we develop an algorithm for the LS approach, implemented in PHASTA, that identifies and controls bubble coalescence events. We perform a comprehensive set of three-dimensional (3D) simulations with the algorithm implemented in order to verify the prevention or slowing of coalescence events.

Up until recently, the study of bubble coalescence in two-phase flow could be characterized by two major categories: surfactant/impurity effects on coalescence and the mechanics behind the coalescence process [3]. However, with the continuing development of computing power, a third category could be added. This third category consists of applying the analytical models of the first two categories to numerical simulations of multiphase bubbly flows with coalescence events.

Direct numerical simulations use numerical techniques to solve the time-dependent Navier–Stokes equations in three dimensions [4]. For a review of multiphase DNS simulations, one is referred to Crowe et al. [5]. For DNS simulations, many different methods have been proposed to track the bubble interfaces including the front-tracking (FT) method, the volume-of-fluid (VOF) method,

the Lattice Boltzmann (LBM) method, and the LS method. The majority of the mentioned methods inherently incorporate coalescence of bubbles without any issue. The FT method, however, uses two separate grids for the two different phases and is unable to represent coalescence without a subgrid model. The VOF method can simulate coalescence but it is difficult to calculate the curvature from the front using volume fractions. van Sint Annaland et al. [6] and Sussman et al. [7] provide a more comprehensive review of each of the above-mentioned interface tracking methods. For more specific review of the FT, VOF, and LBM, one is referred to the following: Unverdi and Tryggvason [8], van Sint Annaland et al. [6], and Dabiri et al. [4] for FT; van Sint Annaland et al. [9] and Passandideh-Fard et al. [10] for VOF; and Takada et al. [11] and Inamuro et al. [12] for LBM.

The LS method uses a distance field to track the bubbles. The bubble–fluid interface is identified by a distance field value of zero and all other points in the domain are marked with the distance to the nearest interface. The method also uses a smoothed Heaviside function to shift between gaseous and liquid properties across the interface. At the zero LS value, the properties will be halfway between liquid and gas, and the transition to full gas/liquid properties happens across the corresponding interface half-thickness. The interface half-thickness is determined by the user and is based on the element size, typically about 1.5 elements. For more information on the LS method, one is referred to two papers by Sussman et al. [7,13] and by Osher and Fedkiw [14]. Sanada et al. [3] further tested and validated the LS method using experimental data from Kirkpatrick and Lockett. A bubble and free surface were modeled in a two-dimensional (2D) simulation with a grid resolution of  $150 \times 150$  or 40 points per diameter. In order to prevent coalescence, the two interfaces were modeled using two independent distance functions. The bubble bounced with the free surface in the simulation just as seen in the experiment. It was also determined that the bouncing We caused is constant and that the coalescence time increases with increasing We. Yang et al. [15] continued development with the LS method by combining it with the VOF method to make a hybrid called the adaptive

Contributed by the Fluids Engineering Division of ASME for publication in the JOURNAL OF FLUIDS ENGINEERING. Manuscript received August 1, 2016; final manuscript received February 21, 2017; published online May 18, 2017. Assoc. Editor: Alfredo Soldati.

coupled level set/volume-of-fluid (ACLSVOF) method. They found that ACLSVOF took advantage of the strengths of both the LS and VOF methods by making the surface tension calculations easier and more accurate while also keeping the mass conserved accurately. Yu and Fan [16] also studied the accuracy of the LS method. They performed simulations for a bubble rising in an infinite liquid by means of the buoyancy force in a 3D environment. They investigated the bubble shapes based on the  $Re$  and found good agreement between their simulations and experimental results. They also study the shape of bubbles during coalescence with the LS method and noted a deviation in the shape of the second bubble compared with a single bubble case. They also mentioned that no successful method has been developed to portray bouncing and coalescence based on the dynamic conditions of when they collide and that whether the bubbles will coalesce or bounce must be decided before the simulation is run. The algorithm developed in this work aims to add this coalescence portrayal capability to the LS method.

## Numerical Approach

Based on the previous work of Sanada et al. [3], it was possible to simulate the thin liquid film viscosity effects in a 2D simulation without any grid dependencies. The work showed that the film did not have time to drain before the bubble was repelled/bounced as observed by Kirkpatrick and Lockett [17]. This simulation used a resolution of  $150 \times 150$  elements which meant that there were 20 elements per bubble radius. The bubble diameter ranged from 1.6 to 2.0 mm. The simulation domain used in computation also used symmetry so that the calculations were only performed on half of the domain. If the largest bubble diameter were used and the same resolution was applied to a small 3D domain of  $20 \text{ mm} \times 10 \text{ mm} \times 10 \text{ mm}$  used in PHASTA simulations, the resolution would be  $400 \times 200 \times 200$  (16M elements) which is much finer than the finest used in PHASTA of  $160 \times 80 \times 80$  (1M elements) for a 5 mm diameter bubble. This much finer mesh for the small 3D domain is feasible but this domain contains only two bubbles.

The goal is to prevent coalescence or slow the coalescence process for hundreds of bubbles. If the two bubble domains are taken as a standard it would need to be 50 times larger to allow for 100 bubbles, resulting in a  $1000 \text{ mm} \times 10 \text{ mm} \times 10 \text{ mm}$  domain. Using the resolution from Sanada et al., this would create a resolution of  $20,000 \times 200 \times 200$  (800M elements) for 100 5 mm diameter bubbles. The computational cost would be too high for a simulation of this size. This means another method is required to simulate the physics of film drainage during bubble coalescence.

**Repulsive Force.** The method chosen to allow the rougher resolution applies a force acting in opposite directions to each bubble interface in order to prevent the coalescence event or slow the process. This force represents the liquid film drainage process between the approaching bubbles which cannot be modeled directly due to the high resolution requirements. Since multiple factors affect the liquid film drainage (e.g., velocity and bubble approach angle), it is necessary that this force will allow for coalescence events under certain flow conditions.

The force is applied by locally increasing the surface tension within a designated area of each bubble. The surface tension force is calculated from

$$F(j, i) = -\sigma(j)\kappa(j, i) \quad (1)$$

where  $\sigma(j)$  is the surface tension in an individual cell and  $\kappa(j, i)$  is the curvature of the bubble interface in a certain cell for a certain direction ( $x$ ,  $y$ , or  $z$ ). Equation (1) demonstrates how the force can be locally applied since a surface tension value is assigned at each cell in the domain. The increase in surface tension was chosen because it was the most computationally affordable method to prevent coalescence compared to the other options (e.g.,

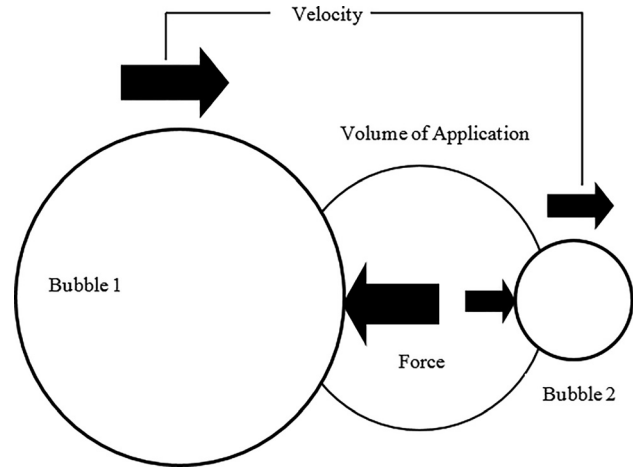


Fig. 1 Schematic of how the coalescence control is implemented and restricts coalescence

artificially increasing viscosity between the bubbles which requires much larger mesh resolution). Since the surface tension is not adjusted uniformly along the interface, a net force acting on each bubble in the direction opposite to the coalescence path arises (see Figs. 1 and 2). The magnitude of the increased surface tension was chosen to be a factor of two stronger. This allows the force to be strong enough to consistently prevent coalescence while minimizing energy added to the bubbles.

The surface tension was changed within a sphere centered at the event location, found by an identification process which is described later. The sphere uses a radius of five times the LS interface half-thickness. The nature of the spherical force application region automatically changes the magnitude of the force based on the distance between the bubbles. This scales the force acting on the bubbles based on the surface area that is subtended by the sphere, which grows by  $\sim d^2$ . This seems to leave the least amount of impact on the accuracy of the simulation since the property values are only changed in a local area around the event. Also, the change only lasts as long as the bubbles are close to one another.

This force method allows for more cost efficient computation but comes with other drawbacks. One issue is that the force may not accurately represent the distance in which coalescence event takes place. The initial thin liquid film distance is estimated to be 0.1 mm by Kirkpatrick and Lockett [17] and the final film thickness is estimated to be  $1 \times 10^{-5}$  mm by Kim and Lee [18]. In comparison to the estimated initial thickness, five times the interface half-thickness depends on the resolution and is only half the film distance. For a moderate resolution and 5 mm diameter bubble, it is equal to 4.5 mm which is much larger than 0.1 mm. This

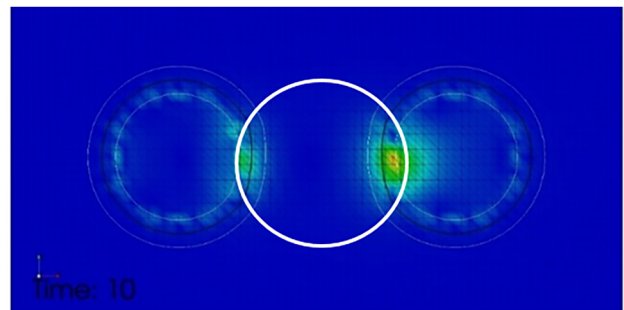
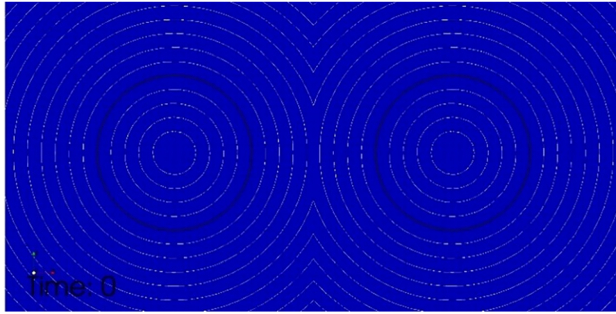
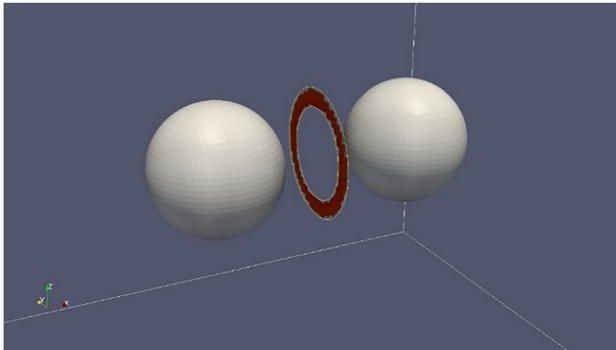


Fig. 2 A later time step after surface tension changed with the coalescence control application volume shown as the white circle and the black circles as the bubble interfaces



**Fig. 3** The distance field of the bubbles shown as the white contour lines with the black circles representing the bubble interface

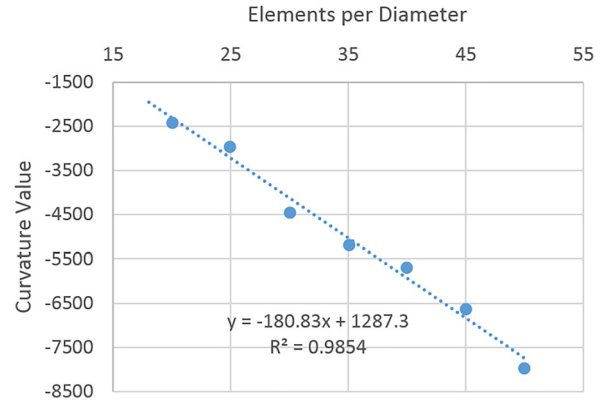


**Fig. 4** Location of high curvature where coordinates are used to generate average coalescence event coordinates. In this image, the annular shape is shown because only the fifth and sixth distance field contours are intersecting so that only those curvature points are detected.

means that the slowing down of a coalescence event would begin sooner than the physical phenomena that are being modeled. Another problem with the model is that the algorithm adjusts the shape of the bubbles. When the surface tension is increased in the sphere, the bubble interfaces within the sphere begin to flatten. This may be physical when the liquid film is very small; however, this begins to occur once the bubbles enter the force application volume. This can be an issue if it is assumed that the bubbles in the simulation are perfectly spherical and the purpose of the simulation is to observe the effects of spherical bubbles on the multiphase flow.

**Location Identification.** One of the important aspects of this algorithm is identifying the location of a coalescence event. It is necessary to know the event location to be able to prevent or slow the coalescence properly. One simple method to find the location is to use the distance field contours generated by the LS method in the PHASTA code. A single distance field contour consists of all the points within the domain that are within the same distance to an interface. In this work, the distance field contours are incremented by the interface half-thickness. By using the distance field contours, it makes it possible to identify not just one coalescence event, but multiple events simultaneously. However, it is beneficial to initially consider one coalescence event because it explains some of the basic principles necessary to detect multiple coalescence events at one time.

*Single Event.* Since PHASTA uses the LS method to identify the bubble interface, there are LS contours throughout the whole domain. In the case of more than one bubble, multiple sets of



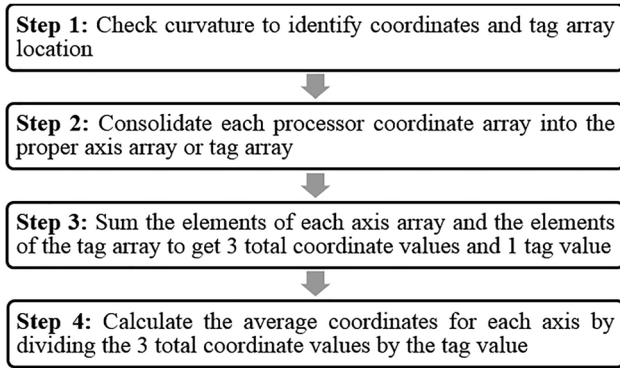
**Fig. 5** Plot of chosen curvature values plotted against the number of elements across a 5 mm bubble diameter. A linear fit was applied with the resulting equation and square of the correlation coefficient.

contours are spread throughout the whole domain and they will intersect with one another (see Fig. 3). At these intersections, the curvature of the contours drastically changes in magnitude and sign. By choosing some of the different contours close to the bubble interface and limiting curvature values, it is possible to record the coordinates of the intersection points in an array. They are also tagged to identify the element positions in the array and that a set of coordinates has been found. These coordinates roughly fill the area between two concentric circles centered in between the approaching bubbles (see Fig. 4).

The current algorithm uses jumps in curvature between the first and the sixth distance field contour. The limiting curvature values used to determine the coordinates have been determined empirically. A set of simple two bubble simulations, similar to the setup above, was used (see Fig. 4). Each simulation in the set used a different resolution with 5 mm bubbles. The curvature values between the first and the sixth distance field contour were recorded. A curvature value close to each jump in curvature was then taken and plotted against the number of elements across the bubble diameter (see Fig. 5). A linear fit was then used to obtain an equation to calculate the limiting curvature value. As a buffer, this limiting curvature value was also multiplied by 1.45 to make sure that intersecting contour coordinates were the only points tagged. The curvature values at the distance field contour intersections are independent of bubble size so the radius of the bubbles used in this process does not contribute to the result.

Since it is possible to run PHASTA in parallel, multiple processors will contain an array with the recorded coordinates. These coordinate arrays from each processor are consolidated into three separate arrays distinguished by axis direction. The tag array from each processor is also consolidated into a single array. Each element of the coordinate arrays is summed and averaged by using the summation of the elements in the tag array. The average coordinates are the center coordinates for the two concentric circles generated by the intersecting contours. Since it is assumed that there is only one coalescence event, the center coordinates identify the location of the event and can be used as the midpoint between the two approaching bubbles. A summary of the process can be seen in the following block diagram (see Fig. 6).

*Multiple Events.* The previously described method works in the case of just one coalescence event but fails to identify multiple events occurring simultaneously. To identify more than one location at a time, an extension of the previously described method is required. A summary of the following process can be seen in the block diagram of Fig. 7. To determine if there are multiple events occurring simultaneously, the average vector distance from the average coordinates to the coalescence events coordinates is

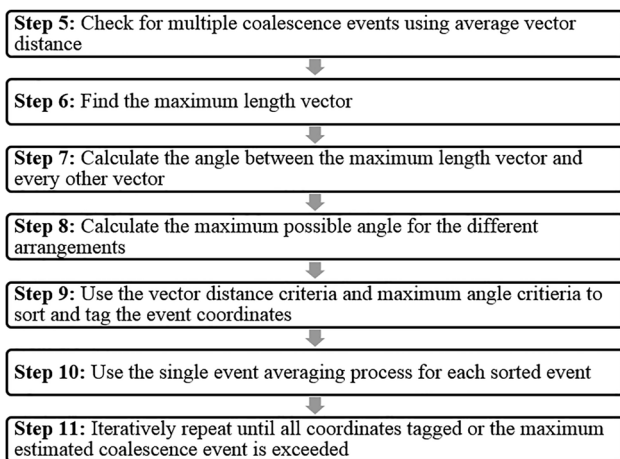


**Fig. 6 Summary of process to identify a coalescence event and calculate the average coordinates for the event**

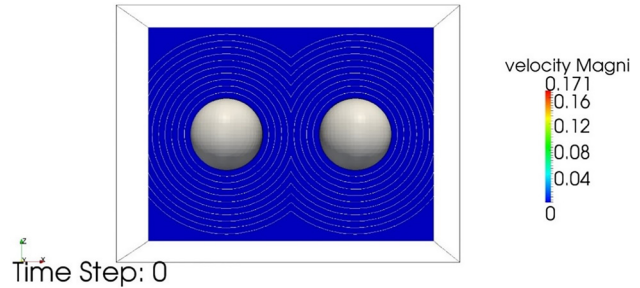
calculated. If this distance is larger than the contour diameter distance, then it signifies that there are multiple coalescence events in the domain.

If it has been determined that multiple events are occurring simultaneously, it is necessary to sort and tag the coalescence events specific to each event. For multiple coalescence events, the average coordinates found from all of the intersecting contour coordinates are located at a point between all of the different coalescence events. This location is not necessarily centered between all the events because the averaging process used in steps 3 and 4 does not use any weights based on the number of coordinates recorded from each event. By knowing this location and the coordinates from each of the coalescence events, it is possible to generate vectors that start at this averaged location and extend to each coalescence event coordinate.

To identify each coalescence event separately, two different calculated criteria are used to consolidate the coordinates to their proper coalescence event. The first criterion used is the distance of each vector originating from the averaged location and the coalescence event coordinates, which was previously calculated. For each coalescence event, the vector with the largest magnitude is found. For a coordinate to be considered part of the specific event belonging to the maximum length vector, it must be located within a distance equal to or less than the diameter of the farthest chosen LS contour for event identification. The distance between an event coordinate and the coordinate of the maximum length vector can be calculated by using vector addition and then



**Fig. 7 Summary of process to find the average coordinates for each specific coalescence event during multiple coalescence events**



**Fig. 8 Initial setup of the simulation used in the mesh study. The domain contains two 5 mm bubble.**

calculating the magnitude of the resulting vector. If the resulting vector is less than or equal to the diameter of the farthest chosen contour for the event identification, then there is a high probability that the coordinate belongs to the specific coalescence event.

Since it is not possible to assuredly determine that a coordinate belongs to a specific coalescence event using the vector comparison, the angle between each vector and the maximum length vector for a specific coalescence event is used as the second criterion. By using the farthest chosen contour diameter from which coordinates are tagged, it is possible to calculate the maximum angle that would exist between the longest vector for a specific coalescence event and another vector, resulting in a triangle with the contour diameter and maximum length vector. Once each of the values for the criteria has been calculated, they can be used to sort and tag the event coordinates specific to each coalescence event. This process will occur iteratively until all event coordinates have been tagged or the maximum number of estimated coalescence events has been exceeded. Once all the iterations have completed, the same averaging process used for a single coalescence event is used for each tag separately. This results in an array of center coordinates for each set of concentric circles generated by different coalescence events and therefore identifies each event separately.

**Drainage Time.** Once the coalescence event has been identified and the force is implemented, it is necessary to track the amount of time the force has been active. The tracked time can then be compared to a coalescence time model [19] that estimates the total coalescence time. Once the tracked time has exceeded the estimated total coalescence time, the force is removed, signifying that the liquid film has had sufficient time to drain, and the bubbles are allowed to coalesce.

The drainage time uses a model [19] that is based on the equivalent bubble radii ( $r_{ij}$ ), the density of the liquid ( $\rho_l$ ), the surface tension ( $\sigma$ ), and the initial and final film thickness ( $h_o, h_f$ )

$$t_{ij} = \sqrt{\frac{r_{ij}^3 \rho_l}{16\sigma}} \ln\left(\frac{h_o}{h_f}\right) \quad (2)$$

The equivalent radius is found by the following equation:

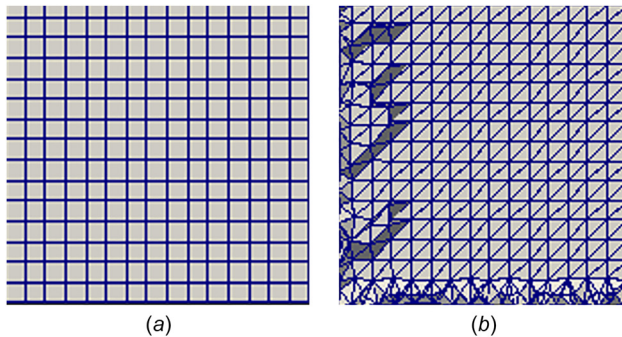
$$r_{ij} = 2\left(\frac{1}{r_{bi}} + \frac{1}{r_{bj}}\right)^{-1} \quad (3)$$

where the indices  $i, j$  represent different bubbles. The initial film thickness is estimated as  $1 \times 10^{-4}$  m by Kirkpatrick and Lockett [17] and the final film thickness is usually taken to be  $1 \times 10^{-8}$  m [18]. The equivalent radius is specified by the user and remains constant even if the simulation has multiple bubbles with different sizes. This is a natural limitation of the model.

To track the coalescence events, it is necessary to develop several logic loops in order to identify new coalescence events, keep track of the continuing events that have and have not exceeded the

**Table 1 Overview of simulation parameters for both mesh types**

Parameter	
Liquid	Water
Gas	Air
Pressure, Pa	0
Temperature, °C	N/A (0)
Liquid density, kg/m <sup>3</sup>	997.17
Gas density, kg/m <sup>3</sup>	1.161
Liquid viscosity, kg/m s	$2.67 \times 10^{-3}$
Gas viscosity, kg/m s	$1.86 \times 10^{-2}$
Body force, m/s <sup>2</sup>	0.5
Body force pressure gradient, Pa/m	498.585



**Fig. 9 Visualization of a (a) structured mesh and a (b) unstructured mesh in Cartesian domain**

model time limit, and determine when events have ended either through coalescence or the bubbles moving away from (bouncing off) one another. Since each event is not assigned the same event number after each iteration, the distance between the average coordinates from the previous iteration and the current iteration is used to identify which events are the same. The distance between the average coordinates will always be quite small. The default values for the average coordinates are used to help identify when any new and ending events occur.

## Results

In order to test the functionality, capability, and cost of the algorithm, three different types of simulations were used and performed. Each of these simulations was performed using the PHASTA code with the coalescence control algorithm added in as a functional capability.

**Mesh Study.** To test whether the algorithm was calculating the average coordinates correctly irrespective of the mesh, a

**Table 3 The error between the reported coordinates and the exact geometric coordinates normalized by the distance between bubble centers**

Percent error normalized by distance between bubble centers			
	20 pts.	30 pts.	40 pts.
X-coord	1.01	0.41	0.45
Y-coord	0.082	9.11	7.86
Z-coord	0.087	7.27	1.34

simple domain of 20 mm × 15 mm × 15 mm was created with two 5 mm diameter bubbles (see Fig. 8). Only gravity in the negative *x*-direction was used to generate a buoyancy force to move the bubbles. The distance between the bubble centers was small enough to activate the coalescence control algorithm. Several different cases were run where the mesh resolution was increased from 20 elements per diameter (288,000 discrete elements and 1.8 million parasolid elements) to 40 elements across diameter (2.3 million discrete elements and 13.3 million parasolid elements). The distance between the bubble centers was kept constant. The value of the interface half-thickness input into PHASTA was also unchanged with the increasing resolution. Two separate simulations were created for each case: one mesh contained discrete hexagonal elements, while the other contained adaptive tetrahedral elements. The same parameters were used for both simulations (see Table 1).

The reason to use two different mesh types comes from the fact that each mesh type is used for different situations. The discrete mesh uses hexagonal elements within a structured grid (see Fig. 9(a)). This grid is simple and efficient but unable to represent shapes other than a cube/prism. On the other hand, a parasolid mesh uses tetrahedral elements in an unstructured mesh (see Fig. 9(b)). This allows noncube/prism shapes to be simulated but may result in more complex and less efficient grid. The coalescence control algorithm will be used in both meshes, and therefore, a mesh study must be performed using both meshes.

Additionally, to test for grid convergence, the magnitude of a bubble's bounce away from a free surface (see Fig. 14) was analyzed using the metric laid out in Ref. [20]. The grids for this study were unstructured and had a resolution of 20 ( $2.7 \times 10^6$  elements), 33 ( $3.7 \times 10^6$  elements), and 40 ( $6.0 \times 10^6$ ) points across the bubble diameter. The mesh was only refined along the path of the bubble, not the whole domain, to decrease computational cost. The value of the interface half-thickness was unchanged for each simulation. As gravity drives the bubble to the surface, it will bounce away as the algorithm is triggered. The vertical, *y*-direction, displacement the bubble experiences,  $\Delta H$ , is the characteristic value that will be used to produce this grid convergence index (GCI). An advanced analysis tool that collects individual bubble information was used to track the center of the bubble [21]. Using this tool,  $\Delta H$  was determined to be the vertical distance traveled in the first bounce by the center of the bubble. Since

**Table 2 The average coordinates for different discrete domain resolutions**

Reported center points for coalescence events in discrete simulation					
Third iteration coordinates					
	20 pts.	30 pts.	40 pts.	Exact	Bubble center distance
X-coord	$1.01 \times 10^{-2}$	$1.00 \times 10^{-2}$	$1.00 \times 10^{-2}$	$1.00 \times 10^{-2}$	$9.00 \times 10^{-3}$
Y-coord	$7.49 \times 10^{-3}$	$8.32 \times 10^{-3}$	$8.21 \times 10^{-3}$	$7.50 \times 10^{-3}$	$9.00 \times 10^{-3}$
Z-coord	$7.49 \times 10^{-3}$	$6.85 \times 10^{-3}$	$7.38 \times 10^{-3}$	$7.50 \times 10^{-3}$	$9.00 \times 10^{-3}$

Note: It also contains the exact geometric coordinates and distance between bubble centers.

**Table 4 The average coordinates for different parasolid domain resolutions**

Reported center points for coalescence events (30 pts. eps.)					
Third iteration coordinates					
	20 pts.	30 pts.	40 pts.	Exact	Bubble center distance
X-coord	$1.01 \times 10^{-2}$	$1.00 \times 10^{-2}$	$1.01 \times 10^{-2}$	$1.00 \times 10^{-2}$	$7.20 \times 10^{-3}$
Y-coord	$5.87 \times 10^{-5}$	$4.50 \times 10^{-5}$	$6.15 \times 10^{-4}$	$0.00 \times 10$	$7.20 \times 10^{-3}$
Z-coord	$1.18 \times 10^{-5}$	$-4.81 \times 10^{-5}$	$1.21 \times 10^{-3}$	$0.00 \times 10$	$7.20 \times 10^{-3}$

Note: It also contains the exact geometric coordinates and distance between bubble centers.

bounce direction parallel to the surface is a random process that varies between real bubbles with similar initial conditions, the  $x$  and  $z$  directional changes are not considered relevant across each grid resolution.

*Structured Mesh.* For this test, the interface half-thickness was kept at  $4.5 \times 10^{-4}$  m which is equal to 1.8 times a 20 point per diameter resolution. The bubbles in the diameter were set up such that the midpoint between them was set at  $(1.0 \times 10^{-2}, 7.5 \times 10^{-3}, 7.5 \times 10^{-3})$ . Each simulation was run for five iterations and the coalescence event coordinates reported were taken from the third iteration.

The third iteration was chosen to allow the algorithm to perform a couple of iterations while not waiting too long such that the bubbles moved sufficiently to drastically move the coalescence event location. These coalescence event coordinates can be seen in Table 2. Each set of coordinates was compared to the exact midpoint value and normalized based on the distance between the center points of the bubble (see Table 3).

Among all the cases, the highest error was seen to be about 9%. The largest errors were also only seen for the  $y$  and  $z$  coordinates. A possible explanation for this result can be found when considering that the intersecting contours of the two bubbles are found in the  $y-z$  plane. It is possible that the concentration of points found around the intersecting contours was not evenly distributed. This would cause a skew in the average  $y$  and  $z$  coordinates without affecting the  $x$  coordinate.

To determine whether the errors affect the outcome in later iterations, 20 elements and 40 elements across the 5 mm bubble diameter simulations were run longer. These simulations were performed for 500 iterations each. The 250th iteration was chosen for the 20 element resolution since the simulation matched the 500th iteration of the 40 element resolution. As expected for the 20 elements per diameter resolution, the coalescence event was prevented. The coalescence event was prevented in the 40 element resolution as well. This shows that the errors in the higher resolution are insufficient to cause the algorithm to work improperly.

*Unstructured Mesh.* For this test, the interface half-thickness was kept at  $3.0 \times 10^{-4}$  m, which is equal to 1.8 times a 30

point per diameter resolution. The bubbles in the diameter were set up such that the midpoint between them was set at  $(1.0 \times 10^{-2}, 0, 0)$ . Again, each simulation was run for five iterations and the coalescence event coordinates reported were taken from the third iteration. These coalescence event coordinates can be seen in Table 4. Each set of coordinates was compared to the exact midpoint value and normalized based on the bubble center distance (see Table 5).

Among all the cases, the highest error was seen to be about 16%. The largest errors were only seen in the  $y$  and  $z$  coordinates. This result shows high accuracy with the expected results. The higher error in the  $y$  and  $z$  coordinates can be explained by the same reasoning in the discrete domain. The  $y$  and  $z$  average coordinates are generated from intersecting contours in the  $y-z$  plane. It is possible that the concentration of points found around the intersecting contours was not evenly distributed. This would cause a skew in the average  $y$  and  $z$  coordinates without affecting the  $x$  coordinate.

To determine whether the errors affect the outcome in later iterations, the same process was used as with the discrete mesh. The two simulations of 20 elements and 40 elements across the 5 mm bubble diameter were performed for 500 iterations each. Visualization of each case was chosen based on matching time within the simulation. For these two simulations, the interface half-thickness was not a proper match. However, for both simulations, the coalescence event was still prevented. This shows that the errors due to a mismatched interface half-thickness are insufficient to make the algorithm work improperly.

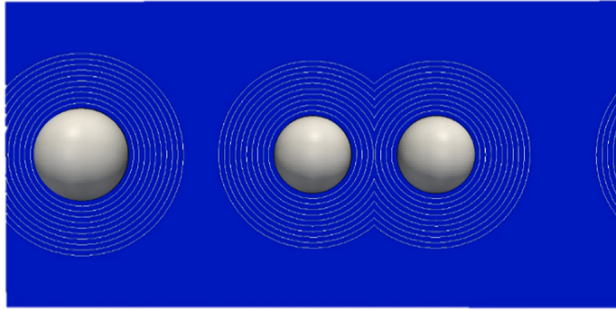
*Grid Convergence.* For this test, the interface half-thickness was kept at  $4.5 \times 10^{-4}$  m, which is equal to 1.8 times a 20 point per diameter resolution. Each simulation was run until the bubble began to move toward the surface after bouncing away the first time. At this point the distance traveled by bubble, the peak  $y$ -direction displacement, was extracted. Table 6 summarizes the analysis. This analysis shows that the algorithm works

**Table 5 The error between the reported coordinates and the exact geometric coordinates normalized by the distance between bubble centers**

Percent error normalized by distance between bubble centers			
	20 pts.	30 pts.	40 pts.
X-coord	1.34	0.55	0.84
Y-coord	0.82	0.63	8.55
Z-coord	0.16	0.67	16.77

**Table 6 Summary of the discretization error analysis**

Parameter	
Points per diameter	20, 33, 40
$r_{21}$	1.2
$r_{32}$	1.67
$\Delta H_1$ (mm)	4.863
$\Delta H_2$ (mm)	4.820
$\Delta H_3$ (mm)	4.242
$P$	3.95
$\Delta H_{ext}^{21}$	5.46
$e_a^{21}$	0.88%
$e_{ext}^{21}$	10.9%
$GCI_{fine}^{21}$	1.04%



**Fig. 10** Initial setup to test the algorithm for a single coalescence event identification and prevention

**Table 7** Overview of simulation parameters for the single coalescence prevention simulation

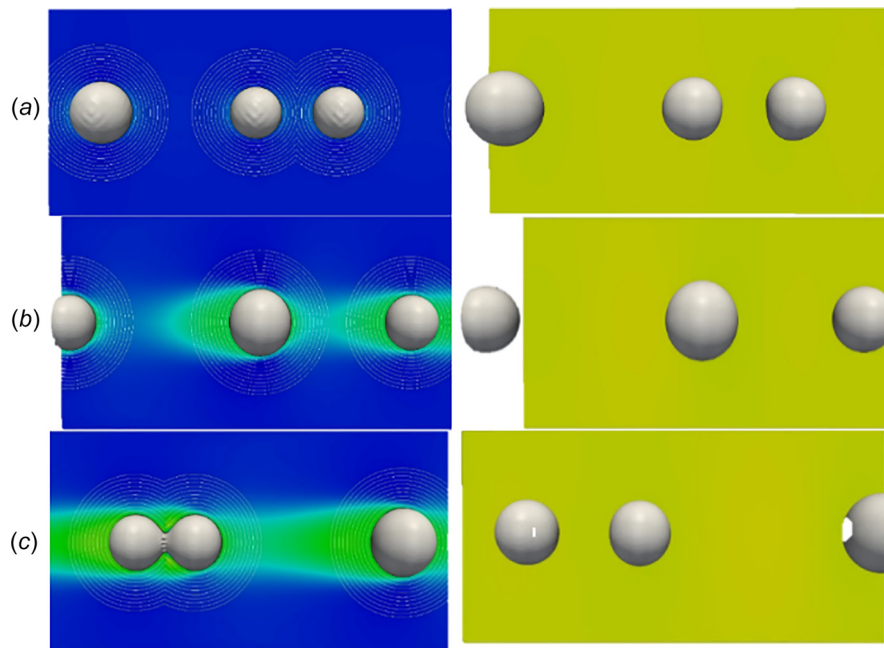
Parameter	
Liquid	Water
Gas	Air
Pressure, Pa	0
Temperature, °C	N/A (0)
Liquid density, kg/m <sup>3</sup>	997.17
Gas density, kg/m <sup>3</sup>	1.161
Liquid viscosity, kg/m s	$2.67 \times 10^{-3}$
Gas viscosity, kg/m s	$1.86 \times 10^{-2}$
Body force, m/s <sup>2</sup>	0.5
Body force pressure gradient, Pa/m	498.585

consistently across varying mesh sizes as the discretization error was found to be 1.04%, relatively small.

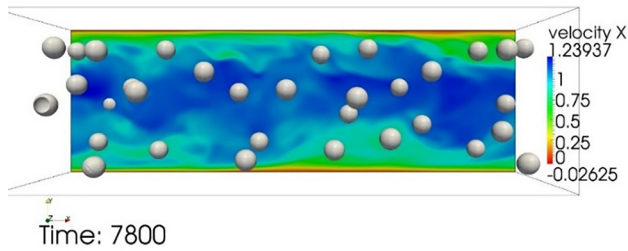
**Two Bubble Coalescence Prevention.** The first simulation consists of how the algorithm handles only one coalescence event occurring in the domain. The time tracking portion of the algorithm was not tested in this simulation. The domain size was  $40 \text{ mm} \times 20 \text{ mm} \times 20 \text{ mm}$  with the  $x$ - $y$  planes acting as walls. There were three bubbles placed in the domain: (i) and (ii) two bubbles with a 5 mm diameter and (iii) a third with a 6 mm diameter. The bubbles were placed at varied distances in the  $x$ -direction along the center line between two parallel plates. Gravity was used to provide a buoyancy force on the bubbles to cause them to flow while a pressure gradient was applied to the liquid to keep it stationary within the domain. A finite element mesh containing 2M hexahedral elements was generated. This means that the 5 mm diameter bubbles were resolved with at least 25 elements across its diameter. For more information on the domain and simulation parameters, see Fig. 10 and Table 7.

The simulation was used for two tests. The first test was performed without using the coalescence control algorithm. The second test was performed with the coalescence control active. To analyze how the algorithm handles a single event, only the interaction between the two 5 mm diameter bubbles is considered.

In the first test, a coalescence event was observed in iteration 870 between the two 5 mm diameter bubbles (see Fig. 11—left). The third larger bubble in the simulation provides a wake the smaller bubbles can become trapped together in. When the second test was performed, the coalescence control algorithm identified the event and locally adjusted the surface tension around the center of the event coordinates. Comparing the visualization of the same simulation time as in the first test, the coalescence of the two 5 mm bubbles was prevented (see Fig. 11—right).

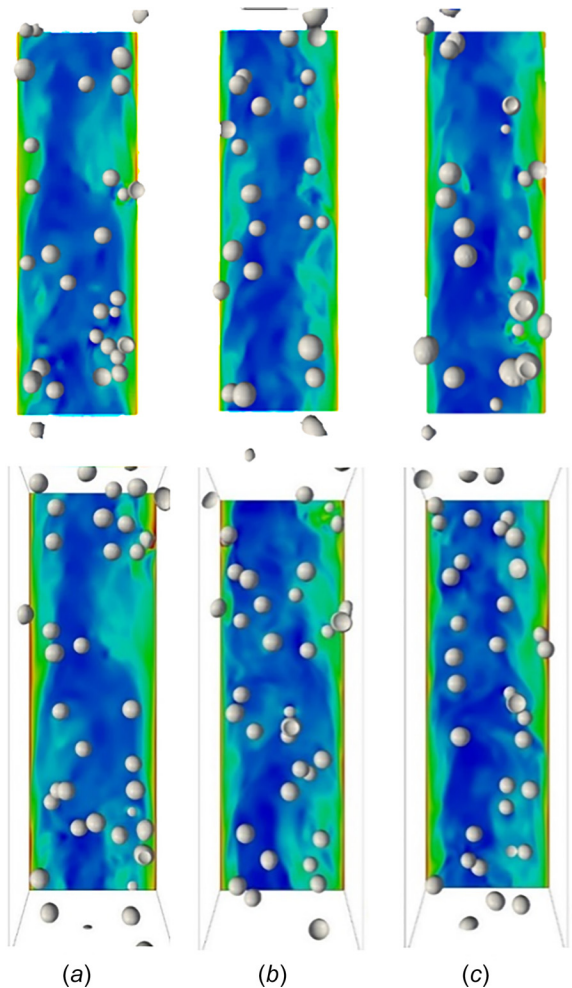


**Fig. 11** Visualization of the simulations for the single coalescence events at iterations: (a) 20, (b) 400, and (c) 870. Left: The simulation performed without the coalescence control algorithm. In iteration 870, the 5 mm bubbles have begun to coalesce (shaded by velocity magnitude). Right: The simulation performed with the coalescence control algorithm. In iteration 870, the coalescence event has been prevented.

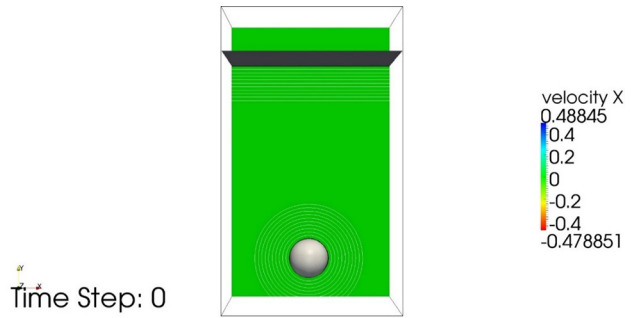


**Fig. 12** Initial setup at iteration 7800 of the bubbly flow in turbulent conditions. There are 32 bubbles randomly placed throughout the domain (shaded by velocity magnitude).

As for computational cost, the simulation without the coalescence control finished 710 iterations in about 2 h. When the coalescence control was activated, the simulation finished 710 iterations in about 2.2 h. This shows a 10% increase in cost of running with the coalescence control. Even though the coalescence control increases the computational cost, this increase in cost is manageable and a willing tradeoff to the resolution required for coalescence viscosity effects to be observed.



**Fig. 13** Visualization of both 32 bubble simulation at multiple iterations. Top: no coalescence control, iterations: (a) 15,600, (b) 23,800, and (c) 27,800. Bottom: coalescence control active, iterations: (a) 15,000, (b) 22,600, and (c) 26,800 (shaded by velocity magnitude).



**Fig. 14** Initial setup for the bubble rising toward a free surface. The simulation was used to test the time tracking portion of the algorithm (shaded by  $X$  velocity where gravity acts in the  $Y$  direction).

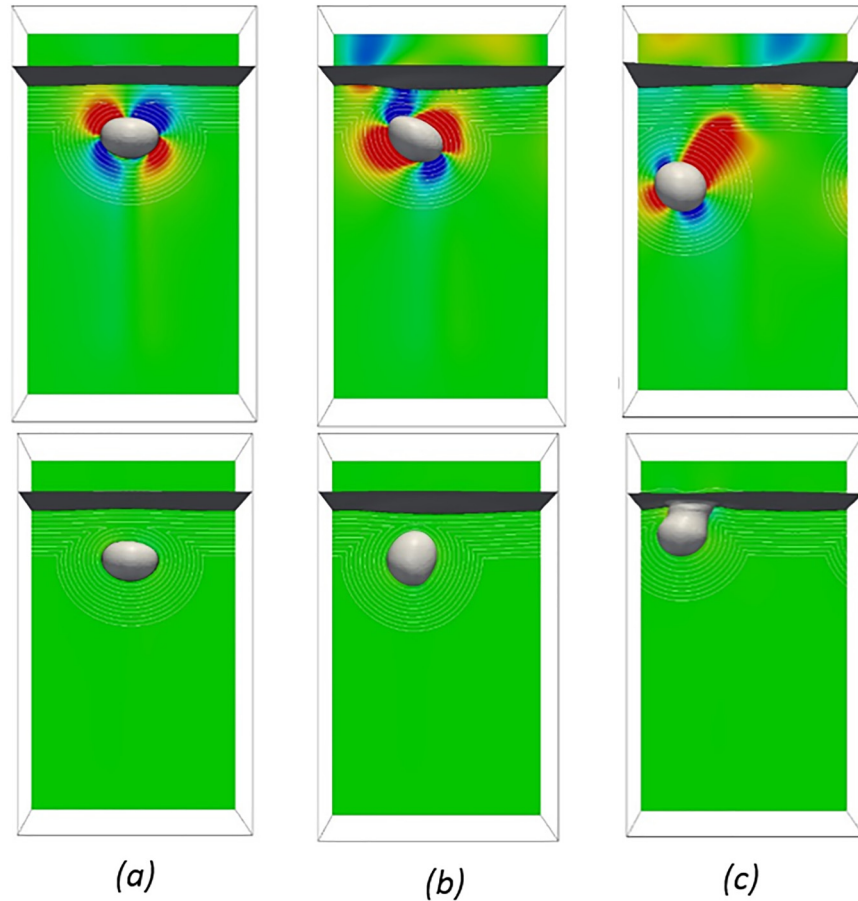
**Multibubble Coalescence Prevention.** In order to test the coalescence control algorithm's ability to handle multiple events as well as turbulence, a large simulation was created. This simulation is based on one reported on by Bolotnov and Podowski [22]. The simulation consists of a bubbly gas flow in turbulent flow conditions with an equivalent  $Re \approx 12,000$ . The flow is between two parallel plates and a total of 32 bubbles were placed randomly throughout the domain. The mesh consisted of about  $10 \times 10^6$  hexahedral elements resulting in approximately 18 elements across the diameter of the bubbles.

The described simulation was tested twice: once without using the coalescence control algorithm and once with the coalescence control algorithm active. The starting point for these tests is iteration 7800 (see Fig. 12). To determine if bubble coalescence events have occurred, later iterations were chosen and the number of bubbles was counted. The iterations that were chosen were matched based on time within the simulation. The times chosen represent two, four, and five full flow-throughs of the domain where a single flow-through means a bubble has moved through the domain to return to its initial position on the  $x$ -axis. For the test when the coalescence control was not used, only 25 bubbles remained the domain at iteration 15,600. Then, in iteration 23,800, only 22 bubbles remained, and in iteration 27,800, only 19 bubbles remained (see Fig. 13—top). This signifies that there were multiple coalescence events after only two flow-throughs. More events continued to follow in subsequent flow-throughs. For the second test using the coalescence control algorithm, in iterations 15,000, 22,600, and 26,800, 32 bubbles remained in the domain instead of 25, 22, or 19 bubbles (see Fig. 13—bottom). This shows that 13 coalescence events that occurred in the first test were prevented by the coalescence control algorithm. This

**Table 8** Overview of simulation parameters for the time tracking tests

Parameter	
Liquid	Water
Gas	Air
Pressure, Pa	0
Temperature, °C	N/A (0)
Liquid density, $kg/m^3$	997.17
Gas density, $kg/m^3$	1.161
Liquid viscosity, $kg/m \cdot s$	$2.67 \times 10^{-3}$
Gas viscosity, $kg/m \cdot s$	$1.86 \times 10^{-2}$
Body force, $m/s^2$	1
Body force pressure gradient, Pa/m	997.17





**Fig. 15 Visualization of both simulations at iterations (a) 800, (b) 920, and (c) 1150. Top: Simulation run without using the time tracking portion of the algorithm. Bottom: Simulation that uses the time tracking portion of the algorithm (shaded by  $X$  velocity where gravity acts in the  $Y$  direction).**

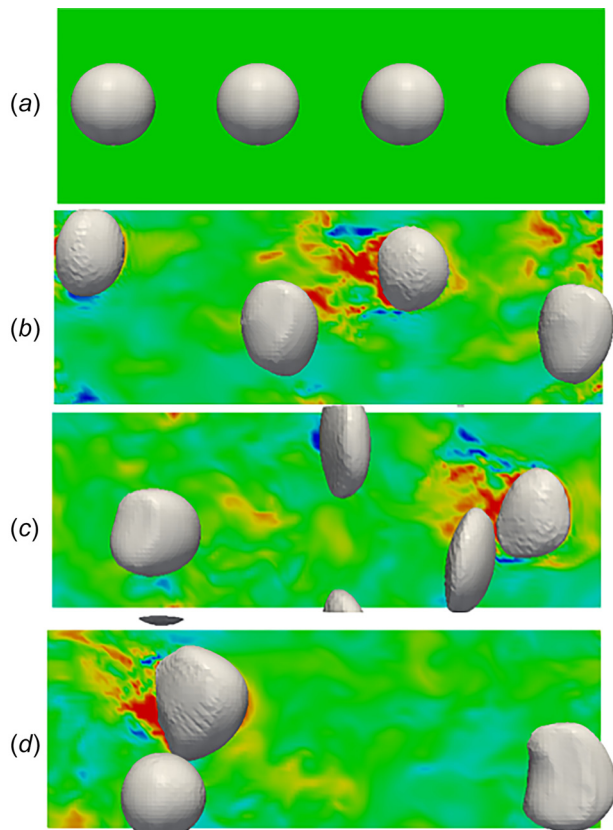
means that the algorithm is performing properly for turbulent conditions with multiple bubbles.

The computational cost for these simulations was also analyzed. The simulation without the coalescence control finished 2150 iterations in 3 h. When the coalescence control was activated, the simulation finished 2150 iterations in 3.8 h. This shows a 25% increase in cost of running with the coalescence control. This computational cost is higher than the single coalescence event but is to be an expected increase. However, even this increase in cost is more desirable over the resolution required for coalescence viscosity effects to be observed.

**Coalescence Time Tracking.** The third test simulation was designed to test the coalescence control algorithm's time tracking model. This simulation design consists of a single bubble below a free surface of gas placed at the top of the domain (see Fig. 14). The domain size was 20 mm  $\times$  35 mm  $\times$  20 mm. The bubble diameter was 5 mm and the center point of the bubble was placed 20 mm below the free surface. Gravity was used to provide a buoyancy force to drive the bubble toward the free surface. A wall was placed on the top and bottom of the domain normal to the gravity and buoyancy force. The mesh was made of 0.9 M hexahedral elements resulting in approximately 20 elements across the diameter of the bubble. For more information on simulation parameters, see Table 8.

Two simulations were performed for this test (see Fig. 15). Both simulations were performed with the coalescence control portion of the code active. However, in only the second simulation, the time tracking portion of the coalescence control was active. It can be seen in iteration 800 of the first simulation that the force is implemented. Later, in iteration 1150, the simulation shows that the coalescence event has been prevented (see Fig. 15—top). For the test when the time tracking portion of the code was active, it can be seen that the force is implemented in the same iteration (800) as the first test. However, by comparing iteration 920 between the simulations, it can be seen that the force is removed in the second simulation. By iteration 1150, the bubble and free surface have coalesced in the second simulation (see Fig. 15—bottom). The second simulation demonstrates that the time tracking portion of the code is properly removing the application volume after the maximum coalescence time has been exceeded.

A new capability was used in these simulations so the computational cost changed. Without the time tracking algorithm being active, the simulation finished 1500 iterations in 2.33 h. When the tracking algorithm was active, the simulation finished 1500 iterations in 2.36 h. This signifies only a 1% increase in the running cost of adding the time tracking portion of the algorithm. This running cost increase is insignificant to the cost of running the rest of the coalescence control algorithm and is therefore an easy tradeoff.



**Fig. 16** Testing the time tracking capability with multiple coalescence events where (a) is the initial condition, (b) is at 0.1875, (c) is at 0.3375, and (d) is at 0.4375 s where one coalescence has occurred and three bubbles remain (shaded by X velocity)

To test the performance of the algorithm with the time tracking capability and multiple simultaneous events, a simple simulation was designed so that multiple coalescence events would be guaranteed to occur. The domain consists of a 28 mm  $\times$  10 mm  $\times$  10 mm periodic channel with four bubbles of 4 mm (20 grid points across) in diameter. The flow was driven by buoyancy in the  $x$ -direction. Figure 16(a) shows the initial condition. During the simulation, up to three coalescence events were simultaneously detected and prevented. Figures 16(b) and 16(c) show the simulation after 0.1875 and 0.3375 s, respectively, where it can be seen that four bubbles still exist. By 0.42075 s of simulation time, one event had persisted for greater than the drainage time and the prevention force was released, allowing the bubbles to coalesce.

**Table 9** Overview of simulation parameters

Parameter	
Liquid	Water
Gas	Air
Pressure, Pa	0
Temperature, °C	N/A (0)
Liquid density, kg/m <sup>3</sup>	997.17
Gas density, kg/m <sup>3</sup>	1.161
Liquid viscosity, kg/m s	$8.54 \times 10^{-4}$
Gas viscosity, kg/m s	$1.86 \times 10^{-4}$
Body force, m/s <sup>2</sup>	0.2
Body force pressure gradient, Pa/m	199.43

Figure 16(d) shows the simulation at 0.4375 s where only three bubbles exist.

**Liquid Film Simulation.** The bubble rising toward a free surface case, with some modified parameters (see Table 9) and no time tracking, was also used to resolve the thin liquid film layer just before the bubble reached the surface. The initial parasolid mesh had 20 points per diameter resolution, but as the bubble approached the surface the mesh was refined in order to resolve the liquid film. First, the mesh around the film was refined to 100 (0.05 mm mesh size), and then again to 285 (0.0175 mm mesh size), points per diameter resolution. With this highly resolved mesh, it was possible to capture the liquid film drainage (see Fig. 17—top) with five elements across the 0.1 mm film thickness estimated by Kirkpatrick and Lockett [17].

As can be seen in Fig. 17—top, the bubble begins to deform (Fig. 17(b)) as it approaches the surface and the film drains, indicated by the high  $X$  velocity in the film. The bubble becomes highly deformed, nearly flat, at the film just before the coalescence (Fig. 17(c)), which ends the event after about 0.3 s. In Fig. 17—bottom, the algorithm is implemented. Figure 17(a) represents the beginning of the event and Fig. 17(c) is the time step when the  $Y$  velocity of the bubble changes sign, i.e., begins to move opposite gravity. This process also lasts about 0.03 s, indicating that the algorithm behaves similar to PHASTA with regard to event timing. Also visible is the similar bubble deformation behavior between the two simulations. As mentioned before, the change in surface tension implemented by the algorithm flattens the bubble, which is observed to replicate the shape of a real bubble waiting for the liquid film to drain.

## Conclusions

An algorithm was developed for the LS method that represents the effect of liquid film drainage during a coalescence event. Instead of using a high resolution grid to simulate the viscous drainage of the film, it prevents or slows the coalescence process by applying a force to both bubbles. The force is generated by locally changing the surface tension on a portion of each interface. This method was tested for single and multiple coalescence events in both laminar and turbulent flow regimes. In both cases, the algorithm successfully identified coalescence events and prevented the bubbles from coalescing.

A portion of the developed algorithm also tracks the amount of time a coalescence event has been active. This tracking time is then compared to a coalescence time model developed by Prince and Blanch [19]. The time tracking capability of the code was tested using a bubble rising toward a free surface. The time tracking portion of the code successfully removed the locally modified surface tension, allowing the bubble and free surface to coalesce.

We have observed some noncritical issues with the algorithm. One of these issues pertains to the deformation of the bubble interface. When the surface tension is changed, that portion of the bubble begins to flatten. In some situations, this is a physical process (e.g., representing the effect of the liquid film and the other interface on the bubble). However, the algorithm applies the changed surface tension before the liquid film is sufficiently thin to cause the flattening of the surface. Another issue is related to the initial film thickness that develops between the bubbles. With a moderate resolution, the initial thickness generated by the coalescence control algorithm is equal to 4.5 mm. This is much larger than the estimated thickness of 0.1 mm by Kirkpatrick and Lockett [17]. For multiple bubble simulations, it is vastly more important to maintain the correct coalescence probability than to correctly resolve the thinnest film when the bubbles bounce off each other. Even with these minor issues, the presented coalescence control algorithm is the only feasible

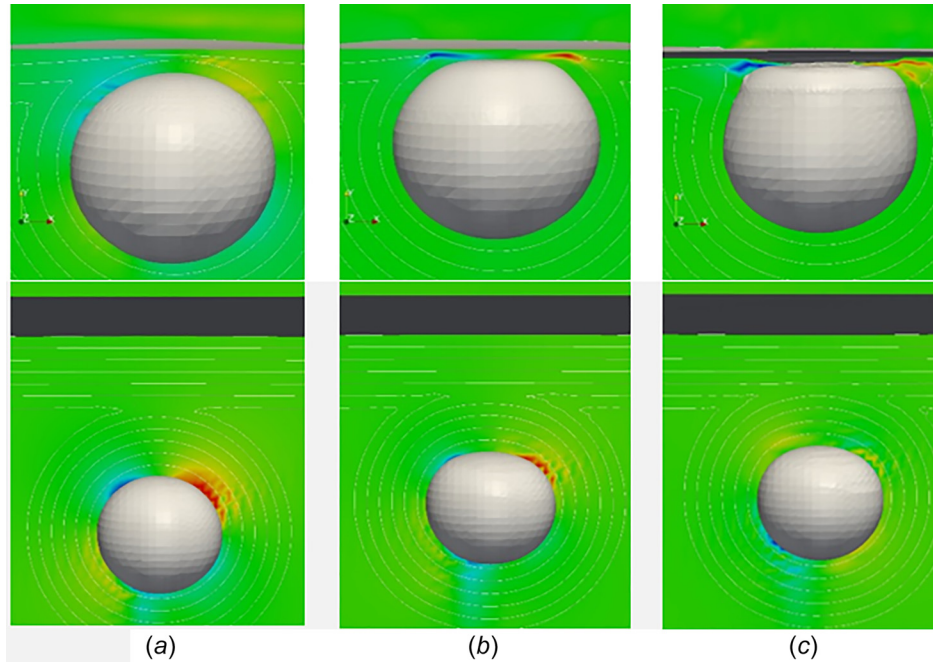


Fig. 17 Comparison of a highly refined mesh (top: at times (a) 0.4025, (b) 0.4212, and (c) 0.4365 s) and the algorithm (bottom: at times (a) 0.4025, (b) 0.4212, and (c) 0.4365 s) for a bubble approaching a free surface (shaded by  $X$  velocity where gravity acts in the  $Y$  direction)

way to simulate multiple bubble behavior using LS approach at large scale.

### Acknowledgment

Computational resources were provided by the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy (Grant No. DE-AC05-00OR22725). Finally, the solutions presented herein made use of the Acusim linear algebra solution library provided by Altair Engineering, Inc., along with the meshing and geometric modeling libraries developed by Simmetrix, Inc. This research was partially supported by the U.S. NRC faculty development program as well as the University Academic Center of Excellence (ACE) at NCSU funded by Idaho National Laboratory. The authors would also like to acknowledge the support from Consortium for Advanced Simulation of Light Water Reactors<sup>1</sup>, an Energy Innovation Hub<sup>2</sup> for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy (Grant No. DE-AC05-00OR22725).

### Nomenclature

$e_a^{21}$	= approximate relative error
$e_{ext}^{21}$	= extrapolated relative error
$h_o$	= initial liquid film thickness
$h_f$	= final liquid film thickness
$p$	= order of grid convergence
Re	= Reynolds number
$r_b$	= bubble radius
$r_{ij}$	= equivalent bubble radius
$r_{21}$	= ratio of finest and second finest mesh size
$r_{32}$	= ratio of second finest and coarsest mesh size
$t_{ij}$	= liquid film drainage time

<sup>1</sup><http://www.casl.gov>

<sup>2</sup><http://www.energy.gov/hubs>

We	= Weber Number
$\Delta H$	= bounce magnitude
$\Delta H_{ext}^{21}$	= extrapolated bounce magnitude
$\kappa$	= interface curvature
$\rho_l$	= liquid density
$\sigma$	= surface tension

### References

- [1] Bolotnov, I., Jansen, K., Drew, D., Oberai, A., Lahey, R. T., Jr., and Podowski, M. Z., 2011, "Detached Direct Numerical Simulation of Turbulent Two-Phase Bubbly Channel Flow," *Int. J. Multiphase Flow*, **37**(6), pp. 647–659.
- [2] Sethian, J. A., 1999, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, Cambridge, UK.
- [3] Sanada, T., Watanabe, M., and Fukano, T., 2005, "Effects of Viscosity on Coalescence of a Bubble Upon Impact With a Free Surface," *Chem. Eng. Sci.*, **60**(19), pp. 5372–5384.
- [4] Dabiri, S., Lu, J., and Tryggvason, G., 2013, "Transition Between Regies of a Vertical Channel Bubbly Upflow Due to Bubble Deformability," *Phys. Fluids*, **25**(10), p. 102110.
- [5] Crowe, C. T., Troutt, T. R., and Chung, J. N., 1996, "Numerical Models for Two-Phase Turbulent Flows," *Annu. Rev. Fluid. Mech.*, **28**(1), pp. 11–43.
- [6] van Sint Annaland, M., Dijkhuizen, W., Deen, N. G., and Kuipers, J. A. M., 2006, "Numerical Simulations of Behavior of Gas Bubbles Using a 3-D Front Tracking Method," *AIChE J.*, **52**(1), pp. 99–110.
- [7] Sussman, M., Almgren, S., Bell, J. B., Colella, P., Howell, L. H., and Welcome, M. L., 1999, "An Adaptive Level Set Approach for Incompressible Two-Phase Flows," *J. Comput. Phys.*, **148**(1), pp. 81–124.
- [8] Unverdi, S. O., and Tryggvason, G., 1992, "A Front-Tracking Method for Viscous, Incompressible, Multi-Fluid Flows," *J. Comput. Phys.*, **100**(1), pp. 25–37.
- [9] van Sint Annaland, M., Deen, N. G., and Kuipers, J. A. M., 2005, "Numerical Simulation of Gas Bubbles Behaviour Using a Three-Dimensional Volume of Fluid Method," *Chem. Eng. Sci.*, **60**(11), pp. 2999–3011.
- [10] Passandideh-Fard, M., and Farhangi, M. M., 2008, "A Numerical Study on Bubble Rise and Interaction in a Viscous Liquid," *Int. J. Comp. Fl. Dyn.*, **24**(1–2), pp. 13–28.
- [11] Takada, N., Misawa, M., Tomiyama, A., and Hosokawa, S., 2001, "Simulation of Bubble Motion Under Gravity by Lattice Boltzmann Method," *J. Nucl. Sci. Technol.*, **38**(5), pp. 330–341.
- [12] Inamuro, T., Ogata, T., Tajima, S., and Konishi, N., 2004, "A Lattice Boltzmann Method for Incompressible Two-Phase Flows With Large Density Differences," *J. Comput. Phys.*, **198**(2), pp. 628–644.
- [13] Sussman, M., Smereka, P., and Osher, S., 1994, "A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow," *J. Comput. Phys.*, **114**(1), pp. 146–159.

- [14] Osher, S., and Fedkiw, R. P., 2001, "Level Set Method: An Overview of Some Recent Results," *J. Comput. Phys.*, **169**(2), pp. 463–502.
- [15] Yang, X., James, A. J., Lowengrub, J., Zheng, X., and Cristini, V., 2006, "An Adaptive Coupled Level-Set/Volume-of-Fluid Interface Capturing Method for Unstructured Triangular Grids," *J. Comput. Phys.*, **217**(2), pp. 364–394.
- [16] Yu, Z., and Fan, L.-S., 2008, "Direct Simulation of the Bouyant Rise of Bubbles in Infinite Liquid Uisng Level Set Method," *Can. J. Chem. Eng.*, **86**(3), pp. 267–265.
- [17] Kirkpatrick, R. D., and Lockett, M. J., 1974, "The Influence of Approach Velocity on Bubble Coalescence," *Chem. Eng. Sci.*, **29**(12), pp. 2363–2373.
- [18] Kim, J. W., and Lee, W. K., 1987, "Coalescence Behavior of Two Bubbles in Stagnant Liquids," *J. Chem. Eng. Jpn.*, **20**(5), pp. 448–453.
- [19] Prince, M. J., and Blanch, H. W., 1990, "Bubble Coalescence and Break-Up in Air-Sparged Bubble Columns," *AIChE J.*, **36**(10), pp. 1485–1499.
- [20] Celik, I. B., Ghia, U., Roache, P. J., Freitas, C. J., and Coleman, H., 2008, "Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications," *ASME J. Fluids Eng.*, **130**(7), p. 078001.
- [21] Fang, J., and Bolotnov, I. A., 2016, "Bubble Tracking Simulations of Turbulent Two-Phase Flows," *ASME Paper No. FEDSM2016-1005*.
- [22] Bolotnov, I. A., and Podowski, M. Z., 2012, "Informing the Development of the Turbulence Models for Bubbly Gas/Liquid Flow Using Interface Tracking Simulations," *Advances in Thermal Hydraulics (ATH)*, San Diego, CA, Nov. 11–15.