ABSTRACT

FANG, JUN. Development of Advanced Analysis Toolkit for Turbulent Bubbly Flow Simulations. (Under the direction of Dr. Igor A. Bolotnov).

An understanding of turbulent two-phase flow is important due to the prevalence of this phenomenon in natural and engineering systems. In nuclear engineering, some examples include the coolant in light water reactor (LWR) fuel rod bundles, and the flow of water vapor mixture in heat exchangers. The rapid development of high-performance computing (HPC) is allowing for the increasingly large-scale high-fidelity simulations. Thanks to this favorable historical trend, direct numerical simulation (DNS), integrated with interface tracking methods (ITM), has been emerging as a valuable tool to complement and expand our ability to understand the two-phase flow phenomenon. However, in order to make best use of the large-scale two-phase simulations, advanced data analysis techniques are highly desired, as they enable the extraction of detailed parameters about individual bubble behavior. Some examples of those parameters include bubble location, velocity, deformation level, and local liquid conditions. Part of the proposed flow analysis is to investigate the correlations among those parameters to improve the understanding of bubbly flow behavior.

The advanced analysis toolkit would allow efficient processing of very large-scale bubbly flow simulations in complex geometries. More valuable information can be extracted as compared to the basic averaging approach. To make this analysis possible, a novel bubble-tracking methodology has been developed for the level set based two-phase flow simulations. To validate the proposed approach of collecting the bubble behavior parameters, several separate simulation techniques have been developed. These include: (1) estimation of interfacial forces; (2) in-situ visualization; (3) simulation steering and others. The turbulent bubbly flows in a single PWR subchannel have been studied at various Reynolds numbers with

both conventional and advanced analysis approaches. The major objective of the present work is to develop new data analysis techniques in order to study representative turbulent two-phase flow problems in nuclear engineering using a DNS/ITM approach. Taking into account DNS prediction capability (based on first-principles calculations) and rapid advancement of HPC, DNS/ITM approach will hopefully serve as ‘virtual experiments’ to help fill the knowledge gap between current understanding of turbulent two-phase flows and that required for future engineering applications. From this perspective, the bubble tracking methodology is believed to pave a new way towards the fundamental understanding of complex turbulent two-phase flows of interest.

Development of Advanced Analysis Toolkit for Turbulent Bubbly Flow Simulations

by
Jun Fang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Nuclear Engineering

Raleigh, North Carolina

2016

APPROVED BY:

Prof. Igor Bolotnov
Committee Chair

Prof. Nam Dinh

Prof. Joseph Doster

Prof. Tiegang Fang

Prof. Paul Turinsky

Prof. Gretar Tryggvason

DEDICATION

This dissertation is dedicated to my parents.

BIOGRAPHY

The author, Jun Fang, was born in the Tenth day of the Seventh month in Geng-Wu year (1990 A.D.), a year of horse in Chinese calendar. The given name ‘Jun’ literally means a spirited horse. The family name ‘Fang’ means direction or solution which can date back to at least the invention of Chinese written language. He is the only child of his father Jishan Fang and his mother Wenping Shi. His hometown is a beautiful valley among mountains along the south bank of Yangtze River, where his clan has settled there for centuries.

Jun Fang graduated from University of Science and Technology of China (USTC) in 2012 with the Bachelor’s degree in Nuclear Engineering. During the undergraduate study, he had an internship at Institute of Modern Physics, Chinese Academy of Sciences, supervised by Dr. Yuan He in the summer of 2011. Soon after his graduation from USTC, he came to North Carolina State University (NCSU) and started the Ph.D. study. He has been working on the direct numerical simulations of turbulent two-phase flows under the guidance of Dr. Igor Bolotnov since then.

There is an ancient Chinese saying, ‘read 10,000 books, and travel 10,000 miles’ (in order to become a wise man). Reading and Traveling are both his favorite hobbies. He is open-minded to all books but especially interested in books of philosophy, history and psychology. Other favored types include Chinese poetry, British plays, French novels, and various popular science books. Reading is a very efficient way to broaden one’s horizon. Although human society and the environment are ever changing, history will always repeat itself explicitly or implicitly because the human nature is constant. When possible, he will do the road trip with no hesitation. He thinks a car is the best symbol of freedom in United States. In the past three years, he has driven to Boston, Buffalo, Detroit, Chicago, Cleveland, New York, Pittsburgh, West Lafayette, Princeton, Philadelphia, Cincinnati, DC, Richmond, Charlotte, Knoxville, Asheville, Wilmington, Myrtle Beach, Atlanta, Charleston, Orlando, Miami and Key West. A road trip to California is still on the to-do list. The world is so broad and magnificent, and there is no time for worrying, self-pity or self-obsession. It is well understood by him that 'eat well, sleep well, work hard, and the Ph.D. is just a beginning'.

ACKNOWLEDGMENTS

I would like to thank God for guiding me to U.S. and helping me so much during my Ph.D. study abroad. I want to give my most appreciation to my parents for they are always my role models. My mother is the most diligent person I have known, and she also teaches me the kindness and patience. My father lets me know the importance of wisdom in life. I feel so lucky and proud to be their son. I also want to thank my girlfriend for her continuous support and understanding for the past years.

I want to express my sincerest thanks to Dr. Bolotnov for all his guidance and help. He is one of my favorite teachers in the department of nuclear engineering as well as a scientist I admire. Dr. Bolotnov not only helps me lay a solid foundation in the thermal-hydraulics research of nuclear reactor designs, but also set me a great example to follow as a responsible researcher. It is my privilege to collaborate with two experienced researchers, Dr. Michel Rasquin and Dr. Hong Yi. Special thanks are given to Dr. Rasquin for his tutoring regarding the large-scale simulations on state-of-the-art HPC facilities. Without their help many of presented results would not be obtained in time. I am also grateful to Dr. Dinh, Dr. Doster, Dr. Tryggvason, Dr. Turinsky and Dr. Fang to help in my committee, and gave me valuable advice to improve my Ph.D. research.

I want to thank my current and former colleagues for the productive cooperation we achieved. Their assistances are highly appreciated. Mr. Anand Mishra helped with the creation of initial Parasolid model of subchannel geometry as well as a XYZTS probe generation code. Bubble control study is a cooperative work with Mr. Aaron Thomas (Matt) and Mr. Jinyong

Feng. Mr. Konor Frick helped to write a bubble generation code for box domains, which I debugged and modified to support the subchannel geometry. The improved coalescence control is based on Mr. Matthew Talley’s coalescence control model and I integrated it with bubble tracking capability. Mr. Shrey Satpathy assisted me in creating Parasolid model of the subchannel geometry with spacer grid and mixing vanes. Mr. Joseph Cambareri originally created a nozzle injection case, which I used later to test the compatibility of bubble tracking algorithm with bubble break-up events.

I want to thank my dear friend, Yuwei Zhu, who is also my roommate and undergraduate classmate. We explored the new world together, and he is just like a brother to me. I also want to thank folks in CASL office, Matt, Bassam, Anand and many others. Thank you all for the inspiring talks, funny jokes and most importantly the companion. I would like to give my special thanks to Mrs. Hermine Kabbendjian for all her kindly help and advice during my study at NCSU.

Finally, I would like to acknowledge the support from Consortium for Advanced Simulation of Light Water Reactors (CASL) (http://www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy. The solution presented herein made use of the Acusim linear algebra solution library provided by Altair Engineering Inc. and meshing and geometric modeling libraries by Simmetrix Inc. This computational resources are provided by the Argonne Leadership Computing Facility at Argonne National Lab.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Overview and motivation

Increasing computer power has paved a new path for the advancement of science and engineering along with experiments, that is, by performing high-resolution numerical simulations to study the problems of interest. One of the most important and complicated phenomena in light water reactors (LWR) under normal operation and accident conditions is two-phase turbulent flow. A reliable prediction of two-phase flows in fuel rod bundles is crucial for both reactor safety evaluation and thermal-hydraulics analysis because of the importance of coolant in reactor cores. Therefore, nuclear engineers have been working for decades experimentally and computationally to obtain a better understanding of two-phase flow behavior.

Considering the extreme conditions and complex supporting structures inside fuel rod bundles, it is very challenging (if not impossible) to study two-phase flows under realistic LWR conditions with high-fidelity experiments. Instead, validated computational approaches are often chosen as a practical way to predict two-phase flow behavior for nuclear reactor design and safety margin evaluation. The nuclear community has developed several thermal-hydraulics codes, such as TRACE [1], RELAP5 [2] and GOTHIC [3], to analyze various reactor transients and loss-of-coolant accidents (LOCA). In practical applications, these codes can provide numerical predictions of important thermal-hydraulics parameters based on 1-D finite volume approach, which involves various single- and/or two-phase closure relations. As a result, the prediction from thermal-hydraulics codes is highly dependent on the applicability of certain closure models. For instance, in the calculation of two-phase flows, interfacial area concentration (IAC) is one of the essential quantities directly related to the exchange of mass, momentum and energy across the interface. IAC is commonly being determined by flow regime maps. Most of those maps are based primarily on finite number of experimental setups and limited to certain geometries (e.g. pipe or rectangular channel) and the fluid conditions near those of the measurements [4]. Thus, the maps might not be directly applicable for some specific LWR calculations (e.g. the flow regime transient in LWR subchannel geometry). Another example of better closures desired is the interfacial force correlations of two-phase flows. The conventional correlations of bubble drag/lift forces are usually derived from experiments of single gas bubbles in stagnant liquid [5-7], which does not fully represent the realistic reactor conditions when bubbles move within a turbulent flow interacting with the walls, structures and each other. Although system thermal-hydraulics codes consists mainly of 1-D and 0-D components, the multi-dimensional capability has also been explored in many newer versions, such as RELAP7 [8] and latest TRACE.

To improve reactor economics and ensure nuclear safety, higher accuracy prediction of key thermal-hydraulic parameters is demanded. This motivation calls for more advanced modeling for two-flow turbulent flow. Although the reactor turbulent flow study could date back to the early days of nuclear energy, researchers are still working to find more suitable closure laws for two-phase flow under specific flow conditions and geometries. Meanwhile, the tremendous growth of computer power in recent years has led to a renewed interest in studying two-phase turbulent flows using direct numerical simulation (DNS) approach. Equipped with a sufficiently fine mesh, DNS can resolve all turbulence structures based on first principles calculations. This involves no turbulence models and can be coupled with interface tracking methods (ITM) to create a promising methodology for the advanced modeling of two-phase turbulent flows [9, 10]. The development of new closure laws for multiphase computational fluid dynamics (M-CFD) and subchannel analysis can utilize the detailed information provided by the high-fidelity interface tracking simulations (ITS) of bubbly flows with DNS of liquid turbulence.

As will be discussed in Section 2.1.2, the level set ITM [11-13] is utilized in our DNS flow solver, PHASTA [14]. The level set method makes use of a signed distance field, and the gas–liquid interface is modeled by the zero level set. The level set method can be readily used to distinguish different phases based on the sign of corresponding level set value (e.g. the sign is positive in the liquid phase while negative in the gas phase.). However, the traditional level set approach is not able to collect calculated values and associate with specific bubbles when multiple (even thousands of) bubbles present in the simulations of two-phase flow. This drawback hinders the collection of useful bubble information, which can give us valuable insights about bubbly flows. For example, how the different local fluid conditions could affect bubble interfacial forces, bubble deformation level, and eventually the bubble distribution throughout the whole domain.

Due to the excellent scalability of PHASTA, large-scale parallel DNS (up to 131,072 computer cores in the presented study) can be performed on state-of-the-art HPC facilities, such as the world's top supercomputers Cray XK7 "Titan" at Oak Ridge National Laboratory (#3) and IBM BlueGene/Q "Mira" at Argonne National Laboratory (#6)[1]. A very large amount of numerical data can be produced from this type of simulations, which imposes a big challenge on data analysis, interpretation and understanding. An important question to ask is that how one can make the most advantage of such a big data and contribute to the understanding and modeling of two-phase turbulent flows. As a response to the need of new data analysis techniques, the presented research has been focused on developing an advanced analysis methodology for ITS of bubbly flows. The development of this new methodology consists of two major parts: (a) the development of ***bubble tracking capability*** and associated data extraction algorithms for level set ITM; (b) the development of statistical analysis tools to process data obtained from bubble tracking simulations. The new advanced analysis methodology can significantly improve the data analysis capabilities of level set based simulations.

In addition to the development of advanced analysis methodology, an interactive simulation workflow was also developed in this dissertation. Since large-scale parallel DNS are usually carried out on HPC facilities (built and maintained by national labs), there is usually a long queue time and launching time (e.g. to request the usage of 128x1024 computing cores, typical queue time can vary between 2 days and 2 weeks). In order to accelerate the DNS

[1] The ranking of HPC facilities is based on the statistics provided by TOP500 (http://www.top500.org), and the most recent TOP500 report was released at 06/20/2016.

discovery process, the interactive workflow allows users checking the instantaneous simulation solutions via ***in-situ visualization*** and even tuning simulation parameters with ***simulation steering*** while the submitted large scale simulation job is on-going. This allows to greatly accelerate the simulation debugging process at large scale, since many issues cannot be re-produced at small scale.

The reliable predictive capability of PHASTA has been extensively demonstrated for various two-phase flow problems, such as bubbly flow with spherical and/or deformed bubbles [10, 15], Taylor bubble flow in a vertical pipe or inclined channel [16], and even the annular flow with resolved entrained droplets [17]. Recent investigation of drag and lift coefficients [18] provided additional evidence that PHASTA can properly capture the physics behind two-phase flow phenomena. These individual validation and verification studies have helped build up our confidence in PHASTA's performance in accurately representing interface behavior. In the future, more verification and validation efforts will be devoted to expand the range of two-phase flow conditions where PHASTA results are credible.

1.2 Literature review

In the past decades, the thermal-hydraulics analysis of nuclear engineering relied heavily on 1-D models with numerous empirical correlations. Meanwhile, the developments in other fields, such as mechanical engineering and computer science, have offered opportunities of using high-fidelity simulations to guide nuclear reactor designs. In this literature review, we will review the legacy and recent research effort related to two-phase flow phenomena in reactor designs. To put current work into a proper perspective, both experimental and simulation research is covered. The literature review is structured as follows: (1) experimental studies of single- and two-phase turbulent flows; (2) two-phase turbulent flow structure and basics of flow regime transition; (3) interfacial forces that bubbles experience in turbulent flows; (4) with the background of flow regimes and bubble interfacial forces, one will see the discussion of two-phase flow modeling; (5) direct numerical simulation is introduced; (6) the concluding section is an overview of popular interface tracking methods. This literature review depicts the general background of presented research, which doesn't claim to cover every detail of selected topics.

1.2.1 Experimental studies of turbulent flow relevant to LWR

In light water reactor cores, the nuclear fuel rods are arranged together as rod bundles with a triangular or square pattern. The fuel rod bundles are supported by the spacer grids with mixing vanes to enhance the heat removal efficiency from fuel rod surface to coolant (as shown in Figure 1). A considerable amount of literature has been published on experiments of turbulent flow in rod bundles. These studies usually measured the distributions of axial velocity, turbulence kinetic energy, and Reynolds stress with various aspect ratios (pitch to diameter ratio, *P/D*) and Reynolds numbers. Trupp and Azad [19] measured the spatial distributions of mean velocity, Reynolds stresses and other important quantities as functions of Reynolds number and tube spacing for fully developed flows. The measurement was made with three different spacing ratios (i.e. *P/D* are 1.20, 1.35 and 1.50), and the Reynolds number ranges from 12,000 to 84,000. The experiments done by Trupp also suggest the existence of secondary flow motion in a rod bundle which is rather different from flow behavior in a typical pipe/channel. Heat transfer calculations form an important part in the design of nuclear fuel elements, which can only be carried out if information of the accurate velocity information is available. Carajilescov and Todreas [20] performed experimental measurements of the distributions of the axial velocity, turbulence kinetic energy, and Reynolds stresses using a laser Doppler anemometer (LDA) in a simulated interior subchannel of a triangular rod array with P/D = 1.123 and L/D_h = 77. Neti et al. and Vonka measured the secondary lateral/horizontal velocity profiles within subchannels by using 2D LDA and they confirmed that the magnitudes were about 1% of the bulk/axial velocity [21, 22].

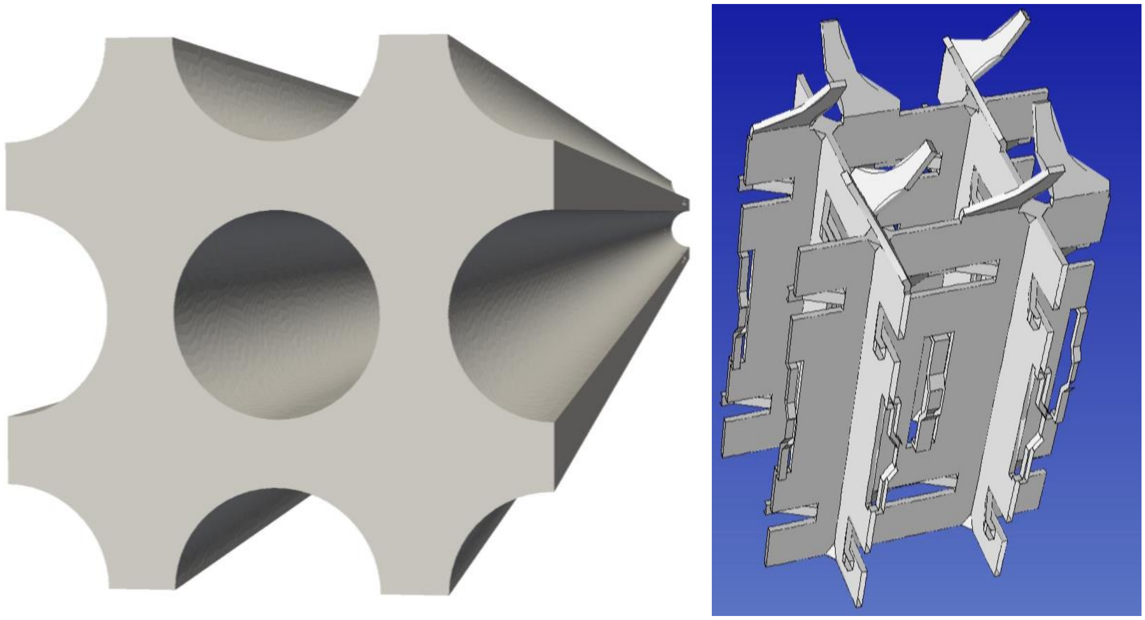


Figure 1: A 2×2 subchannel geometry enclosed by a 3×3 PWR fuel rod bundle (left) and the spacer grid and mixing vanes inside it (right).

An accurate prediction of the hydraulic losses in multirod fuel bundles is necessary for reactor design calculations. Rehme et al. [23-26] proposed various correlations of pressure drop of rod bundles on the basis of numerous experimental investigations. The dependence of the pressure drop on the pitch-to-diameter or wall-to-diameter ratios has been examined. In addition, correlations for the calculation of pressure loss due to spacer grids are presented and compared with experimental data. Engel et al. [27] performed the experimental investigation of parallel flow pressure drop across wire-wrapped hexagonal rod bundles positioned inside a duct using water, sodium and air. The flow conditions range from laminar, transition, to fully turbulent regimes. Bubelis and Schikorr [28] presented a comprehensive review about aforementioned and many other existing correlations of rod bundle pressure drop/friction factor. They also proposed a best-fit correlation by qualitatively evaluating the correlations available at that time.

The momentum and heat transport process between reactor core subchannels is an important phenomenon to be considered when designing fuel rod bundles. According to the early studies by Todreas et al. [29, 30], there are two distinguish mixing patterns: forced mixing effects and natural mixing effects. Forced mixing effects are caused by the presence of spacer grids or other geometrical disturbances (e.g. mixing vanes). Natural mixing results from radial pressure gradients between adjacent subchannels, especially important in the entrance region where the inlet mass flow rate is redistributed among the subchannels or if the heat-flux distribution across a rod bundle is non-uniform. The influence of the spacer grids on the turbulent mixing within square subchannel geometry was studied by Yang and Chung [31].

The authors analyzed the generation and decay of the turbulent energy from their LDV measurement results. In order to understand the mechanism of natural mixing between subchannels, Rehme performed measurements using hot wire anemometry (HWA) [26]. He concluded that the natural mixing is mostly influenced by the almost periodic macroscale flow pulsations caused by the flow instability. The secondary flow motion in subchannels does not contribute significantly to the mixing process.

The fundamental inter-subchannel exchange mechanisms also include the void drift when it comes to two-phase flows, besides the turbulent mixing and cross-flow convection one can see in single-phase cases. Based on experimental data from an inter-connected, two-subchannel test section in the slug-to-churn turbulent flow regime, Shirai and Ninokata's study [32] indicates that void drift can be regarded as a flow redistribution phenomenon. This void drift, when explained as a flow redistribution behavior, can achieve the minimum possible pressure gradient under the hydrodynamic equilibrium condition.

Thanks to state-of-the-art experimental techniques, researchers can now obtain more reliable data-base with higher spatial and temporal resolution. For example, Dominguez-Ontiveros and Hassan [33] have recently done a non-intrusive experimental investigation of flow behavior inside a transparent 5x5 rod bundle with spacer grids using dynamic particle image velocimetry (DPIV). Another example can be found in two-phase flow experiments. Gas volume fraction (or void fraction) is a fundamental quantity for characterizing the structure of two-phase flows. Murai et al. developed advanced techniques for the three-dimensional measurement of void fraction in a bubble plume using statistic stereoscopic image processing, which can obtain the instantaneous 3-D void fraction distribution even for high void faction bubbly flows [34].

1.2.2 Two-phase flow structure and regime transition

Two-phase turbulent flows are commonly observed under the operational conditions of boiling water reactor (BWR) and pressurized water reactor (PWR), which is a more complicated problem to understand compared to single-phase flows. BWR operate through a wide range of flow regimes, while PWR are typically limited to subcooled boiling two-phase flow. There are many legacy experiments dealing with two-phase turbulent flows. Under room temperature and atmospheric pressure, Serizawa et al. performed one of the early experimental studies to investigate the microstructure as well as heat and bubble transport in turbulent air-water bubbly flows [35-37]. The flow patterns in Serizawa's experiments range from bubble flow to slug flow with varying gas superficial velocities. The most important discoveries include: (a) the void fraction distribution is observed to change from wall-peaked in bubble flow to center-peaked in slug flow; (b) turbulent flow quantities, like bubble velocities and water velocities, have a rather flat radial profile (excluding wall region) in fully developed flow; (c) the augmentation, suppression, and further augmentation of turbulence intensity is observed with increasing gas flow rate.

To understand the flow structure of subcooled boiling water flow in a subchannel, Yun et al. [38] measured local void fraction, interfacial area concentration, interfacial velocity, Sauter mean diameter, and liquid velocity using a conductivity probe and a Pitot tube in 20 locations inside one of the subchannels. Wheeler et al. [39] performed the experimental study to investigate the effect of spacer grid on the structures of air-water two-phase flows. They presented measurements of local time-averaged two-phase flow parameters acquired upstream and downstream of the spacer grid with the conductivity probe in a representative subchannel of a 1x3 rod bundle for eight flow conditions.

The safety margin evaluation of both PWR and BWR involves the consideration of flow regime/pattern transition. Depending on flow rate, fluid properties and geometry, the gas-liquid flow can present various flow patterns, typically from bubble flow, to slug flow, to churn flow and eventually annular flow. Heat and mass transfer rates, momentum loss, rates of back mixing and residence time distributions all vary significantly with different flow patterns. Void fraction and interfacial area concentration (IAC) are two most important characteristic parameters for flow regime transition phenomena. As void fraction quantifies the probability of gas phase presenting at a certain location in the domain, IAC is used in two-phase models to determine the exchange rate of mass, momentum and energy across the interface. Since IAC is a critical parameter in two-phase flow modeling and simulations (M&S), several approaches have been proposed to predict IAC under different flow conditions. The conventional approach is based on the so-called "flow regime maps", and empirical correlations of IAC are then developed for specific flow regimes. Taitel et al. [4] experimentally investigated the flow regime transition criteria in pipes. The study answered under which set of operating conditions a certain flow pattern can exist as well as the gas-liquid flow rate pair at which transition between flow patterns will take place. Improved flow regime maps were then proposed for pipes of different diameters.

However, the application of flow regime maps is reported to have many limitations. Ishii and co-workers have raised the following concerns: (a) the conventional flow regime transition criteria cannot fully reflect the true dynamic nature of changes in the interfacial structure; (b) the method based on flow regime transition criteria may introduce compound errors from transition criteria and interfacial area correlation; (c) the transition criteria and flow-regime-dependent interfacial correlations are valid in limited parameter ranges for certain specific operational conditions and geometries [40, 41]. These limitations motivate the development of interfacial area transport equations (IATE). IATE is aimed at replacing the static flow regime maps and predicting the dynamic evolution of interfacial area concentration. The most challenging task in the development of IATE is how to accurately model the IAC source and sink terms resulting from bubble coalescence, bubble breakup or phase change.

1.2.3 Interfacial force study

As one of the two-phase flow regimes, bubbly flow is of great importance in both BWR and PWR. For example, the bubbly flow in upper core of PWR caused by subcooled boiling can improve the heat removal efficiency due to the bubble-driven turbulence and convection [42]. In general, the ability to predict thermohydraulic phenomena of two-phase flow depends on the availability of mathematical models and experimental correlations [43]. The overall distribution of bubbles is directly related to the interfacial forces acting on individual bubbles. Therefore, the development of better interfacial force correlations is demanded in order to make more precise prediction regarding bubble behavior and phase distribution in two-phase flows. The bubble interfacial force can be further decomposed into various components.

Generally, important interfacial force components include drag force [5, 6], lift force [44, 45], virtual mass force [46] and turbulent dispersion force [47]. Many correlations of interfacial forces have been proposed based on experimental database, serving as an important foundation of two-phase flow modeling.

Drag force

Drag force is one of the dominant interfacial forces, observed when there is a non-zero relative velocity between dispersed phase (e.g. steam or gas bubble) and continuous phase (e.g. liquid). A lot of theoretical or experimental correlations can be found in the literature regarding bubble drag coefficient or terminal velocity [5, 6]. In order to predict the average volumetric concentration in two-phase flow systems, pioneers like Zuber and Findlay [48] placed an emphasis on the effect of local relative velocity (i.e. drag effect). Ishii and Zuber [5] developed more general correlations of drag coefficient and relative motion for dispersed two-phase flows of bubbles, drops, and particles using simple similarity criteria and a mixture viscosity model. The proposed model obtained satisfactory agreements at wide ranges of the particle concentration and Reynolds number. Drag coefficient strongly depends on fluid properties, bubble equivalent diameter, gravity and the contamination on the gas-liquid interface [49]. Based on experiments of single bubbles in stagnant liquid, Tomiyama et al. [6] proposed simple yet reliable correlations for drag coefficient of single bubbles under a wide range of fluid properties, bubble diameter and acceleration of gravity. They employed a force balance strategy and available empirical correlations of terminal rising velocities of single bubbles to derive and validate their drag coefficient correlations.

In specific engineering applications, bubbles sometimes experience significant size change during their lifetime. This is expected when the pressure inside the bubble varies rapidly because of the variations of the hydrostatic pressure in the surrounding liquid or when the bubble moves in a superheated or subcooled liquid and tends to grow or to collapse. Magnaudet and Legendre [50] analytically derived the expression of the drag force experienced by a spherical bubble having a variable radius and moving in a viscous incompressible liquid.

Besides experimental and analytical research on drag effect/phenomena, computational approach is also an important tool. Based on front tracking method (Section 1.2.6) and direct numerical simulation (Section 1.2.5) techniques, Roghair et al. [51] developed a closure correlation for the drag force acting on a bubble in a mono-disperse swarm accounting for the presence of neighboring bubbles, referred to as the swarm effect, in particular as a function of the gas void fraction.

Lift force

Lift force is perpendicular to the relative velocity and acts in the direction of the liquid velocity gradient, which directly influences the radial/transverse void fraction distribution. The lateral movement and distribution of bubbles in the channel also affect turbulence generation [52, 53] and heat flux from the fuel rods to the coolant. Pioneering work has made several assumptions in order to derive an analytical form of the lift force which is valid for a limited number of low shear flows [54, 55]. This and other analytical lift approximations are too basic for use in complex flows, where bubble shape [56, 57], location with respect to a wall [58], turbulence [59, 60], and many other factors all influence the shear-induced lift phenomenon.

One of the most widely used lift force correlations [7] was experimentally developed for high viscosity systems, where the transverse force due to a bubble’s slanted wake as separate from the shear induced lift force (but same function form) was also correlated; the sum of the shear induced lift force and the slanted wake induced lift force equals the net transverse lift force. However, this experimentally developed correlation may not be applicable to low viscosity systems, such as steam and water.

One of the most recognized characteristics of the lift force in an up-flow condition is that small bubbles tend to migrate toward a channel wall, whereas larger bubbles tend to migrate toward a channel center [61-64], or more generally, the bubble deformability greatly affects the direction of the lift force. Low surface tension can allow for bubble deformation and result in a sign flip in the lift force which was shown experimentally [7, 57], and was in agreement with numerical work [65]. A numerical study suggested that the interaction between the bubble’s wake, shear flow regime, and internal gas flow of the bubble strongly affect the lateral migration [66]; it was also found that the Eötvös and Morton numbers play an essential role in the direction and magnitude of the lateral migration.

Virtual mass force

Virtual mass force is also referred as added mass force, which is an unsteady force due to a change of the relative velocity of a submerged body in a fluid. More specifically, a sphere accelerating through a quiescent fluid experiences a resistance force proportional to its acceleration. The magnitude of virtual mass force of a rigid sphere is equal to one-half the mass of the displaced fluid times the sphere’s acceleration [67]. Drew et al. [68] presented the derivation of virtual mass acceleration as well as the appropriate verification experiments. In the follow-up work of Drew and Lahey [45], virtual mass force and lift force are treated together as nondrag force combination.

Turbulent dispersion force

Turbulent dispersion force is the result of the turbulent fluctuations of liquid velocity. As has been pointed out by Lahey et al. [47], the motivation to develop a turbulent dispersion force comes from the fact that the averaged two-phase continuity equations do not allow for a phasic diffusion term. Using an analogy with the thermal diffusion of air molecules in the atmosphere, they derived a turbulent dispersion force model. A more generalized non-uniform turbulent dispersion coefficient on basis of homogeneous turbulence was proposed by de Bertodano [69]. Similar expressions of turbulent dispersion force can be found in the works of Drew [70] and several other researchers.

1.2.4 LWR two-phase flow modeling

Experiments remain an undisputed reference for model validation nowadays [15]. However, it is usually prohibitive to conduct high-fidelity realistic pressure/temperature conditions experiments for reactor coolant flows because of the complex nature and extreme conditions of realistic LWR environment. Taking AP1000 as an example, the design system pressure is 2250 psia (i.e. 153.1 atm) and average core temperature is 578.1 F (i.e. 303.4 C) [71]. Therefore, in practical applications, researchers have to adopt validated computational codes to predict flow behavior under LWR conditions [72]. For instance, the advanced thermal-hydraulic code COBRA-TF [73] is being used worldwide for best-estimation evaluations of nuclear reactor safety margins.

Both subchannel and computational fluid dynamics (CFD) methods are being improved as the nuclear industry advances to Generation III+ and Generation IV reactor technology. Avramova [32] recently improved the theoretical models and numerics of COBRA-TF. In the latest version of COBRA-TF, a revised spacer grids model was implemented to account for the turbulent mixing enhancement due to spacers and the lateral flow patterns created by specific configurations of the spacers' structural elements [74]. Conner et al. [75] presented the Westinghouse CFD methodology to model single-phase, steady-state conditions in PWR fuel assemblies as well as benchmark testing.

As has been observed from numerous experiments, the flow conditions inside fuel rod bundles are quite different from those typical pipe flows. One of most significant differences observed is that the coolant actually undergoes a spiral pattern through the bundle. This secondary flow is caused by the near wall turbulence anisotropy in non-circular ducts. The regular RANS models in fact assume isotropic eddy diffusivity in modeling the Reynolds stress tensors, and therefore anisotropic effects are not accounted for. Baglietto et al. [76] have developed an isotropic turbulence model which can reproduce the secondary spiral motion of coolant in fuel rod bundles.

A number of studies have been concentrated on multiphase flow simulations. Historically, the modeling of two-phase turbulent flow evolved from homogeneous mixture [77, 78] (1930's and 1940's), mixture models involving slip correlations [79] (1950's and 1960's), and then into two-fluid modeling [80] (1970’s – present). In 1992, Lahey and Drew [81] derived a three dimensional two-fluid model of vapor/liquid two-phase flows using ensemble averaging. As admitted by the authors, the key to accurate two-fluid modeling is the interfacial and wall closure laws. Later on, Lahey and Drew presented a mechanistically-based, four field, two-fluid model for two-phase flow and heat transfer that can accurately predict the distribution of the continuous vapor, continuous liquid, dispersed vapor, and dispersed liquid fields [82]. A “research roadmap” was also proposed by Lahey for the development of flow-regime-specific CMFD models using DNS and experimental results [83]. Although the multi-field conservation equations seem to be a direct extension of those governing single-phase flows, it turns out that the averaging procedure introduces several constraints on the formulation of individual models. Podowski [84] provided insightful discussion about the consistent formulation of ensemble-averaged conservation equations, and the associated models of interfacial phenomena between the continuous and disperse fields.

Many two-phase flow models have been developed for nuclear fuel rod bundle designs. The CFD-BWR model was developed by Ustinenko et al. [85] which can allow the detailed analysis of the two-phase coolant flow and heat transfer phenomena in a BWR fuel bundle. This model was further combined with the commercial code STAR-CD to calculate location of vapor generation onset, axial temperature profile, and axial and radial void distributions. In the field of nuclear power research, one of the major technological issues is the Departure from Nucleate Boiling (DNB) condition in the fuel assembly of a nuclear reactor core [86].

Due to the absorption of heat from heated fuel rod surface, liquid coolant may undergo the phase change (evaporation), such as the subcooled boiling in PWR. The early stage of boiling phenomenon is usually referred as “nucleate boiling”, during which one could expect higher heat transfer rate because the bubble motions (bubble-driven turbulence and convection) can lead to rapid mixing of the coolant. However, this boiling heat transfer is limited by Critical Heat Flux (CHF), beyond which the burnout of fuel rod surface may take place due to the insufficient heat transfer of the gas film formed between heated surface and liquid coolant. Since CHF is a characteristic parameter for DNB phenomena, CHF mechanisms for subcooled flow boiling were reviewed by Lee and Mudawwar [87]. Considering the dependence of bubble diameter on local flow conditions, Končar et al. did the multidimensional modeling of vertical upward subcooled boiling flow. The modeling was based on a two-fluid approach and local two-phase flow parameters (e.g. bubble size) were calculated [88]. Krepper et al. [89] also presented their work on CFD modeling of subcooled boiling. Although subcooled boiling represents only a preliminary state towards the critical heat flux occurrence, essential parameters like swirl, cross flow between adjacent channels and concentration regions of bubbles can be determined. By calculating the temperature of the rod surface the critical regions can be identified which may later on lead to departure from nucleate boiling and possible damage of the fuel pin [89].

1.2.5 Direct numerical simulations

In engineering applications, which usually involve complex geometries and high Reynolds number flows, a full-scale DNS calculation may not be a practical solution yet. Although the computational cost is and will be the major bottleneck in the near future, DNS approach has started to attract the engineering community’s attention. It may serve as a promising tool in studying challenging turbulence problems, even those related to nuclear reactor coolant. The optimism with DNS is supported by following several rational reasons: (1) DNS could provide high-fidelity fundamental insights to complex fluid phenomena, such as the turbulence anisotropy; (2) this approach allows to carefully study the separate effects of various parameters; (3) the tremendous growth of HPC is making expensive simulations more and more affordable.

In DNS of turbulence, the equations of fluid motion (i.e. the Navier-Stokes equations) are solved, without any turbulence closure assumptions (unlike classic CFD approach, or even more empirical subchannel analysis). With sufficient temporal and spatial resolution, DNS can represent all the scales of turbulence down to the Kolmogorov scales [90, 91]. Thanks to this prediction capability based on first principles calculations, DNS is usually regarded as a reliable data source for model development and validation along with experiments. However, in order to fully resolve the turbulence of interest, DNS requires very fine meshes, and the mesh size grows exponentially as Reynolds number goes up.

The history of DNS is a vivid example of how scientific research could benefit from the rapid growth of computing power. In 1980’s, Moser and co-workers [92] have applied DNS in turbulence studies of fully developed channel flow at low Reynolds number. A large number of turbulence statistics are computed and compared with the existing experimental data at comparable Reynolds numbers. The maximum Reynolds number (Re_τ) investigated at that time was 180 based on wall shear velocity/friction velocity (u_τ). Eqs (1) and (2) provide the definitions of friction Reynolds number and bulk Reynolds number, where δ is half width of flat channel and ν is fluid viscosity. u_τ and U_b are friction velocity (defined by Eq. (50)) and mean bulk velocity respectively. More than ten years later, as a continuation of the previous work, Moser et al. [93] performed the channel simulations with two higher Reynolds numbers (one at $Re_\tau \approx 395$, and the other at $Re_\tau \approx 590$). Most recently, Lee and Moser [94] have conducted a DNS of incompressible channel flow at friction Reynolds number of 5,186 (Re_{bulk} of 125,000 [94]). The flow exhibits a number of the characteristics of high-Reynolds-number wall-bounded turbulent flows. Based on the definition given by Eq. (45), the hydraulic diameter of a flat channel is twice its channel width (i.e. 4δ). This results in a Reynolds number (Re_h) of 500,000 taking hydraulic diameter as the characteristic length as shown in Eq. (3). It's worthwhile to mention that Re_h of 500,000 is higher than that of turbulent flow conditions (Re_h of 452,500) in a realistic PWR environment [95].

Friction Reynolds number:
$$Re_\tau = \frac{u_\tau \delta}{\nu} \tag{1}$$

Bulk Reynolds number
$$Re_{bulk} = \frac{U_b \delta}{\nu} \tag{2}$$

Reynolds number
$$Re_h = \frac{U_b D_h}{\nu} \tag{3}$$

Due to the tremendous growth in computing power over past decades, DNS capability has already overlapped with problems of practical engineering interest. This field is yet to be fully explored, and more effort should be devoted into the development of DNS capability as well as associated advanced analysis techniques. The recent effort of DNS application in nuclear related problems can be found in the literature. For example, Ninokata et al. have applied DNS to a fully-developed single phase turbulent flow analysis for triangular pin bundles, but the Reynolds numbers resolved in their DNS are relatively low (up to Re_h of 24,300) [96].

1.2.6 Interface tracking methods

DNS of multiphase flows has also been studied previously and provided unprecedented insights into complex flow phenomena. Lu and Tryggvason [97] studied a turbulent bubbly upflow in a vertical channel using front tracking method, and it was observed that the void fraction profile highly depends on the deformability of the simulated bubbles. The extended study done by Dabiri et al. has revealed that a regime transition of vertical channel bubbly upflow takes place when Eötvös number is between 2.0 and 3.0 [98]. Bolotnov et al. studied the turbulent bubbly flows in flat channels with DNS to investigate the bubble distribution and bubbles' influence on the turbulence field [10, 15]. Fang et al. have implemented a proportional-integral-derivative (PID) controller within level set interface tracking simulations to evaluate the drag and lift forces a single bubble experiencing in laminar uniform shear flows [99], based on which Thomas et al. have introduced higher order terms in the controlling model and studied the interfacial forces of single bubbles in laminar medium, laminar high and turbulent high shear flows [18]. Demonstration cases show that the drag coefficients extracted at low shear with various bubble Reynolds numbers can achieve an excellent agreement with experimentally based correlations proposed by Tomiyama et al. [6].

As discussed by Tryggvason et al. [100], the extension of single phase simulation capability to two phase flow simulations usually relies on a marker (i.e. phase indicator) function that can be advected by the flow. The phase indicator function plays two important roles: (1) it is used to determine interface location and represent interface topology change, and the related two-phase simulation techniques are referred as interface tracking methods; (2) it is used to determine the material properties of different phases, such as density and viscosity. Here we will introduce the three mainstream interface tracking methods, volume of fluid method [101], front tracking method [102] and level set method [103]. The solution techniques of these three approaches are all based on the "one-fluid" formulation. The one-fluid formulation allows multiphase flow with interfaces to be treated as a single fluid which has different properties on each side of the interface. This way the Navier-Stokes equations for both phases can be solved in single system of equations using variable material properties determined by phase indicator function.

Volume of fluid method

The volume-of-fluid (VOF) method [101, 104, 105] is the oldest and continues to be widely used after many improvements and innovations. To implement the VOF method, one can define a function F whose value is unity at any point occupied by fluid and zero otherwise. The average value of F in a cell would then represent the fractional volume of the cell occupied by fluid. In particular, a unit value of F would correspond to a cell full of fluid, while a zero value would indicate that the cell contained no fluid. Cells with F values between zero and one must then contain a free surface. This way VOF method could provide the coarse interface information. In order to track the interface motion, the evolution of the *F* field is then computed based on Eq. (4). As the next step, the interface is reconstructed from the marker function by generating either a horizontal or a vertical interface within each interface cell.

$$\frac{\partial F}{\partial t} + \boldsymbol{u} \cdot \nabla F = 0 \tag{4}$$

One of the most unfavorable features of VOF method is its relatively poor representation of interface topology. This drawback becomes especially significant when one needs to consider surface tension forces because surface tension implementation requires accurate information of interface curvature. As a consensus, the key to improving the behavior of the VOF advection scheme is the reconstruction of the interface in each cell. Various interface reconstruction techniques have been developed so far, such as simple line interface calculation (SLIC) [104], Hirt-Nichols reconstruction [101], and piecewise linear interface reconstruction (PLIC) [106]. VOF method still enjoy considerable popularity due to the fact that it is relatively simple to implement and reasonably accurate. However, for higher accuracy at the cost of considerably more complexity, it is necessary to consider front tracking or level set method.

Front tracking method

Front tracking method is another major interface tracking approach. Instead of advecting the marker function directly, the boundary between the different fluids is explicitly tracked in front tracking method. This special treatment significantly reduces the resolution needed to keep the front sharp, and eliminate numerical diffusion altogether. The interface is not kept completely sharp but is rather given a finite thickness of the order of the mesh size to provide stability and smoothness. A justification for spreading the sharp interface onto the grid can be found in Tryggvason’s work [99]. The interface is represented using connected marker elements and dynamically advected with the flow [102, 107]. A marker function/phase indicator function is then reconstructed from the location of interface. Unlike VOF method, front tracking method can provide accurate representation of interfacial quantities, such as interfacial normal and curvature information. However, front tracking method has difficulties in handling the interaction of a front with another front. The original computational procedure does not recognize more than one front in each cell of the stationary grid, and double interfaces have to be merged into one or eliminated. In order to overcome this drawback, several sophisticated algorithms have been developed to better model bubble coalescence or breakup over the years [108]. Although front tracking method has achieved considerable success in two-phase bubbly flows in simple geometries with structured grids [9, 109, 110], more effort is needed for its application in two-phase flow problems within complicated geometries of engineering interest (where the unstructured mesh is generally involved for precise representation of complex engineering geometries).

Level set method

The specific interface tracking method used in the present research is the level set method, which was first introduced by Osher and Sethian [111] and further developed by Sussman [11, 103, 112]. The level set method utilizes a signed distance field to represent the phases separated by an interface, and the interface is modeled by zero level set. More details regarding level set implementation can be found in Section 2.1.2. Since the initial introduction of level set method into multiphase simulations [103], it has been widely used as one of the major interface tracking methods. Level set method is selected for presented research due to following three desirable features: (a) level set method can provide accurate representation of interfacial quantities, such as interfacial normal and curvature; (b) level set method makes no assumptions about the connectivity of the interface, which can allow topological transition (e.g. bubble coalescence or breakup) to occur naturally without user intervention or extra coding; (c) level set method can be easily coupled with finite element method and unstructured mesh to provide simulations of two phase flow in very complicated geometries, such as 2x2 PWR structure with spacer grid and mixing vanes [95].

The traditional level set ITM is certainly capable to distinguish the different phases (e.g. gas phase and liquid phase). However, in the multi-bubble simulations it couldn’t support the collection of detailed information regarding the individual bubble behavior, such as bubble velocity, volume, deformation level, and even the local liquid velocity and shear. These detailed bubble parameters are vital in developing more precise correlations among various aspects of bubble behavior important for closure models. As a result, in this dissertation the bubble tracking methodology was developed to improve the data collection capability for level set based simulations.

1.3 Dissertation overview

This dissertation is structured as follows. In Chapter 2, we provide the introduction of the flow solver, basics of level set ITM, mesh design, an efficient bubble initialization approach as well as the conventional data analysis technique [10] utilized in our research before the development of bubble tracking capability. The preliminary results and development of the bubble tracking methodology has been reported in several previous papers [99, 113, 114]. Based on these results a comprehensive description of the capability is provided in Chapter 3. In addition to the bubble tracking capability, several novel PHASTA techniques are introduced in Chapter 4, which would lead to a more efficient and reliable simulation workflow: improved bubble coalescence control can prevent premature coalescence events on unstructured meshes; bubble control capability can be used to estimate the lift and drag force a bubble experiences in uniform shear flows [18, 99]; in-situ visualization and simulation steering allow interactive navigation of the large-scale parallel simulations [115, 116]. In Chapter 5, the DNS of single- and two-phase turbulent flows are presented for a single PWR subchannel geometry. Post-processing analysis adopts both the basic (static virtual probes) and advanced (bubble tracking) analysis techniques. It will be shown that the bubble tracking capability greatly enriches the amount of information obtained from the large scale DNS for turbulent bubbly flows. These new insights can be used to develop new and improve existing multiphase flow closure models. To conclude the dissertation, the completed research work is summarized in Chapter 6 and future work recommendations are presented in Chapter 7 as well.

CHAPTER 2. NUMERICAL TOOLS AND METHODS

The flow solver used in the present work is PHASTA (a finite element code described in Section 2.1); the visualization is carried out using ParaView [117], open-source software developed by Kitware Inc. Meshing tools for PHASTA are provided by Simmetrix Inc. and linear solver libraries are provided by Altair Eng. Inc. via academic licenses. The current analysis includes conventional statistical averaging for flow evolution data from static virtual probes, and advanced analysis techniques for bubble tracking data that will be discussed in Chapter 3.

2.1 PHASTA overview

PHASTA is a **p**arallel, **h**ierarchic, higher-order accurate (from the 2nd to the 5th order accuracy, depending on function choice), **a**daptive, **s**tabilized (finite element) **t**ransient **a**nalysis flow solver (both incompressible and compressible). This approach has been shown by Jansen [118] and Whiting [119] to be an effective tool for bridging a broad range of length scales in turbulent (RANS, large-eddy simulation (LES), detached eddy simulation (DES), DNS) flows. PHASTA (and its predecessor, ENSA) was the first unstructured grid LES code [120] and has been applied to turbulent flows ranging from validation benchmarks (channel flow, decay of isotropic turbulence) to complex flows (airfoils at maximum lift, flow over a cavity, near lip jet engine flows and fin-tube heat exchangers). The PHASTA code uses advanced anisotropic adaptive algorithms [121] and the most advanced LES/DES models

[122]. Note that DES, LES, and DNS are computationally intensive even for single phase flows because a very fine mesh is required to resolve the turbulence scales of interest. This capability has been recently [14] extended to two phase flows where we use the level set method to track the boundary between two immiscible fluids (either compressible - where we captured new instabilities in sonoluminescence, or incompressible – to study bubble coalescence and two-phase turbulence [10]).

In addition, PHASTA supports unstructured grid, which makes it feasible for simulations of turbulent flows in complex geometries, such as the 2x2 PWR structure with spacer grids and mixing vanes. Together with the highly scalable performance on massively parallel computers, PHASTA is believed to be a promising tool for advanced modeling of turbulent bubbly flows, which is currently an active topic in the research of nuclear community. The excellent scalability of PHASTA has already been demonstrated [123], and the code has shown good scaling up to 768×1024 processors on the IBM Blue Gene/Q Mira system (ANL, #6 in top500 as of June 2016).

PHASTA is an open source code. However, in the current setup, it uses commercial linear solver libraries from Altair Engineering, Inc. A possible switch to open-source solvers is considered (e.g. Trilinos, PETSc). Meshing capabilities utilize tools from Simmetrix, Inc. Creating mesh converters from open-source tools to PHASTA format is also a possibility. PHASTA works with hexahedral, tetrahedral and mixed finite element meshes and has adaptive mesh refinement (AMR) and coarsening capabilities.

2.1.1 Governing equations

PHASTA solves the Incompressible Navier-Stokes (INS) equations directly in three dimensions using a stabilized FEM. The spatial and temporal discretization of INS equations within PHASTA has been discussed previously by Whiting and Jansen [119] and Nagrath et al. [124]. The fluid is assumed to be isothermal in presented research. The strong form of INS is given by

Continuity:
$$\frac{\partial u_i}{\partial x_i} = 0 \tag{5}$$

Momentum:
$$\rho \frac{\partial u_i}{\partial t} + \rho u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + f_i \tag{6}$$

where u_i is the velocity in the x_i-direction, ρ denotes the density of the fluid, p the static pressure and τ_{ij} the viscous stress tensor. f_i represents the component of the body force along the x_i-direction. For the incompressible flow of a Newtonian fluid, the viscous stress tensor is related to the fluid viscosity μ and the strain rate tensor, S_{ij}, as:

$$\tau_{ij} = 2\mu S_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{7}$$

As discussed in Section 1.2.6, level set method has a superior capability in capturing interface quantities. Based on the level set distance field (φ), the curvature of gas-liquid interface can be computed as

$$\kappa(\varphi) = -\nabla \cdot \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) \tag{8}$$

The surface tension force is then represented by

$$\kappa\sigma\left(\frac{\nabla\varphi}{|\nabla\varphi|}\right) \tag{9}$$

where σ is surface tension coefficient. Adopting the Continuum Surface Tension (CST) model proposed by Brackbill et al. [125], the surface tension force in PHASTA is computed as a local interfacial force density, which is included in f_i.

2.1.2 Level set method

Level set method [11-13] utilizes a signed distance field from the interface and models the interface as the zero-level set of a smooth function, φ, where φ is the signed distance from the interface, often called the first scalar. Hence, the interface is defined by $\varphi = 0$. The scalar, φ, is convected within a moving fluid based on

$$\frac{D\varphi}{Dt} = \frac{\partial\varphi}{\partial t} + \boldsymbol{u}\cdot\nabla\varphi = 0 \tag{10}$$

where $\boldsymbol{u}$ is the local fluid velocity vector. In two-phase simulations, phase-1, the liquid phase, is indicated by a positive level set, $\varphi > 0$, and phase-2, the gas, by a negative level set, $\varphi < 0$. Since evaluating the jump in physical properties using a step change across the interface leads to poor computational results, the properties near an interface were defined using a smoothed Heaviside kernel function, H_ε [12], given by:

$$H_\varepsilon(\varphi) = \begin{cases} 0 & \text{for } \varphi < -\varepsilon \\ \frac{1}{2}\left[1 + \frac{\varphi}{\varepsilon} + \frac{1}{\pi}\sin\left(\frac{\pi\varphi}{\varepsilon}\right)\right] & \text{for } |\varphi| < \varepsilon \\ 1 & \text{for } \quad \varphi > \varepsilon \end{cases} \tag{11}$$

where $[-\varepsilon, \varepsilon]$ represents the interface transient region in level set distance field. As exhibited in Figure 2, Heaviside function could bridge two discrete properties (i.e. step change) over the interface with a smoother transition profile.

The ‘one-fluid’ approach is implemented in PHASTA such that multiphase flow can be simulated by solving a single set of INS equations for the whole flow filed. Consequently, it is necessary to account for differences in the material properties of different fluids. With the property transition over interface described by the smoothed Heaviside function, the expressions of fluid properties are then provided by:

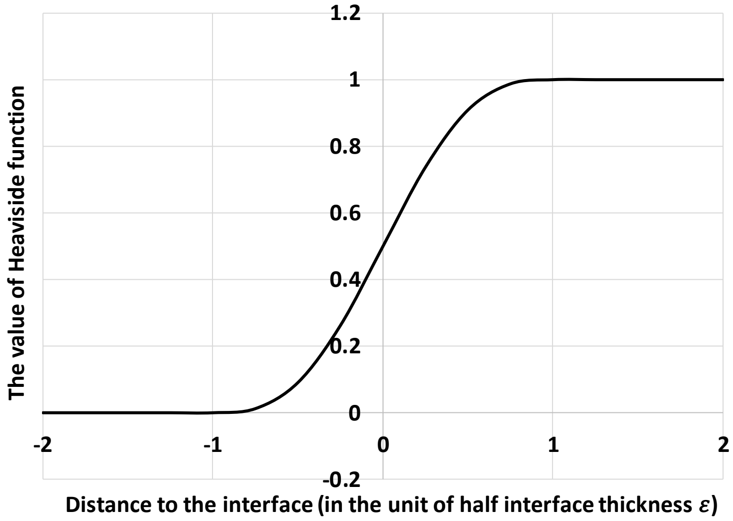


Figure 2: The transition profile of Heavisible function.

$$\rho(\varphi) = \rho_1 H_\varepsilon(\varphi) + \rho_2(1 - H_\varepsilon(\varphi)) \tag{12}$$

$$\mu(\varphi) = \mu_1 H_\varepsilon(\varphi) + \mu_2(1 - H_\varepsilon(\varphi)) \tag{13}$$

Although the solution may be reasonably good in the immediate vicinity of the interface, the distance field may not be correct in the whole domain. The varying local fluid velocities throughout the flow field distort the level set contours (especially in fully resolved turbulent flow simulations). Thus, the level set is corrected with a re-distancing operation [126] by solving the following PDE:

$$\frac{\partial d}{\partial \tau} = S(\varphi)[1 - |\nabla d|] \tag{14}$$

where d is a scalar that represents the corrected distance field and τ is the pseudo time over which the PDE is solved to steady-state. This may be alternately expressed as the following transport equation:

$$\frac{\partial d}{\partial \tau} + \boldsymbol{w} \cdot \nabla d = S(\varphi) \tag{15}$$

The so-called second scalar, d, is originally assigned the level set field, φ, and is convected with a pseudo velocity, $\boldsymbol{w}$, where,

$$\boldsymbol{w} = S(\varphi)\frac{\nabla d}{|\nabla d|} \tag{16}$$

and $S(\varphi)$ is defined as:

$$S(\varphi) = \begin{cases} -1 & \text{for } \varphi < -\varepsilon_d \\ \left[\frac{\varphi}{\varepsilon_d} + \frac{1}{\pi}\sin\left(\frac{\pi\varphi}{\varepsilon_d}\right)\right] & \text{for } |\varphi| < \varepsilon_d \\ 1 & \text{for } \quad \varphi > \varepsilon_d \end{cases} \tag{17}$$

where ε_d is the distance field interface half-thickness which, in general, may be different from ε used in Eq. (11). Note that the zeroth level set, or interface, $\varphi = 0$, does not move since its convecting velocity, $\boldsymbol{w}$, is zero. Solving the second scalar to steady-state restores the distance field to $|\nabla d| = 1$ but does not alter the location of the interface. The first scalar, φ, is then updated using the steady solution of the second scalar, d.

The workflow of level set algorithm can be summarized as follows:

Step 1: Initialize $\varphi(\boldsymbol{x},0)$ to be singed normal distance to bubble interface.

Step 2: Solve the velocity field and advection of level set field

$$\frac{\partial \boldsymbol{u}}{\partial t} = f_\rho(\boldsymbol{u}) \qquad\qquad \frac{\partial \varphi}{\partial t} + \boldsymbol{u}\cdot\nabla\varphi = 0 \tag{18}$$

for one time step with $\rho(\varphi)$ given by Eq. (12) and $\mu(\varphi)$ given by Eq. (13), and the surface tension force. Let's denote φ by $\varphi^{(n+1/2)}$, and the updated $\boldsymbol{u}$ by $\boldsymbol{u}^{(n+1)}$.

Step 3: Construct a new distance function by solving Eq. (15) with $\varphi(\boldsymbol{x},0) = \varphi^{n+1/2}(\boldsymbol{x})$ to steady state. The steady state solution is taken as $\varphi^{(n+1)}$.

Step 4: The simulation has been advanced for one time step. The zero level set of $\varphi^{(n+1)}$ gives the new interface position while $\varphi^{(n+1)}$ is the updated distance function at new time step. Repeat Steps 2 and 3.

2.2 DNS mesh design

PHASTA can be used to perform the DNS by satisfying the following requirements to ensure an accurate representation of all relevant scales: (1) The computational domain must be sufficiently large to contain the largest turbulent eddies in the modeled flow, and (2) the grid spacing must be sufficiently fine in order to capture the smaller scales of interest (e.g. Kolmogorov turbulent length scale). The first requirement is met if the two-point correlations in the stream-wise and span-wise directions vanish within one-half of the computational domain [91]. In the meantime, the number of mesh points in physical domain must be chosen to resolve the finest scale of appreciable excitation. The mesh size grows exponentially with resolved Reynolds number (usually with power of 9/4). More detailed discussions of DNS mesh resolution requirements can be found in [93]. The first plane of grid points off the walls was at a normalized distance (y^+) of 1.0. Also, the pressure gradient is chosen to balance the force exerted by the fluid on the conduit walls. A Maple worksheet was created to evaluate all related parameters; this workflow is discussed step by step in Appendix A.

Once the mesh configuration parameters (e.g. bulk resolution, boundary layer settings) are determined, a parallel unstructured meshing infrastructure (PUMI) [127] is utilized to create the computational mesh based on the model topology as produced by CAD software[2]. The meshing tool used is called Chef (a PUMI based derivative). Chef is designed to support the representation of, and operations on, unstructured meshes as needed for the execution of mesh-based simulations on massively parallel computers. Besides the capability of high level model definition, Chef also support a specific set of services to better facilitate parallel unstructured mesh simulations, such as mesh uniform refinement, solution migration, and adaptive mesh partitioning.

In an efficient large scale PHASTA simulation workflow, large velocity fluctuations/turbulent eddies are usually computed on relatively coarse mesh without resolving smaller turbulent scales, and then mesh refinement and solution migration brings the obtained turbulence velocity field onto a finer mesh. Smaller turbulence scales are then allowed to

[2] The CAD software used in present research is SolidWorks, which is a solid modeling computer-aided design and computer-aided engineering computer program published by Dassault Systèmes.

develop on the finer mesh in a time-efficient manner (it is expensive to allow the development of the largest eddies on the finest mesh since this takes a lot of computational time, and thus consumes the expensive core-hours). This process will be continued until all necessary turbulence scales can be properly resolved, namely, a DNS representation of turbulent flow is achieved. In order to fully resolve liquid turbulence, the DNS mesh generated usually consists of very large number of elements (millions or even billions), especially the ones for intended simulations of high Reynolds number flows of engineering interest. In parallel simulations, the mesh is to be partitioned into smaller pieces that can be distributed to specific compute cores/ranks. To ensure reliability and efficiency, chef is also equipped with the adaptive mesh partitioning techniques to limit the imbalance among mesh partitions within a satisfactory level [128]. Proper mesh balance (e.g. relative difference of number of elements and nodes on each partition) is important to achieve good code scalability.

2.3 Bubble initialization

For the simulations of two-phase bubbly flows, bubbles are introduced into the domain by defining the initial level set distance field using the following equations:

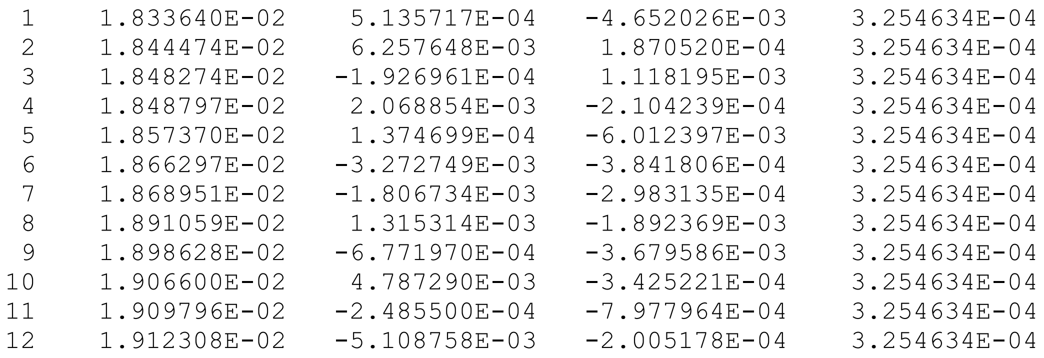
$$\varphi = \min(\varphi_i) \tag{19}$$

$$\varphi_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} - R_i \tag{20}$$

where x_i, y_i and z_i are the coordinates of *i^th^* bubble center; R_i is the corresponding radius. The conventional way of bubble initialization is to come up with a complex minimum function supplied to the graphical user interface (GUI) of Simmodeler[3] (TM) with bubble centers' coordinates and corresponding radii. It can result in a correct analytical expression of the distance field for multiple bubbles. However, this approach is inefficient and very time consuming when a large number of bubbles (more than 100) are to be modeled on large meshes (larger than 100M elements). The main reason is that the serial pre-processing code which assigns the initial condition has to apply the *min*() function which includes all the bubbles for each mesh node in the domain. Based on our experience, the code would run for over 3 days to initialize more than 100 bubbles for meshes larger than 100M elements given the correct expression. Moreover, it is usually difficult to determine location of a large number of bubbles in various domains manually in such a way that they are distributed evenly without overlap. To address these issues, a novel bubble initialization code has been developed; some example algorithms in this code are provided in Appendix C.

Given void fraction and domain size, the pre-processing code is used to generate a practical initial bubble distribution. The intersections/overlaps among bubbles or between bubbles and walls are avoided by using Monte Carlo rejection sampling method [129]. The code can produce bubble centers' coordinates and radii, as well as bubble ID's which will be used in initializing the ID field for bubble tracking capability (as shown in Figure 3). In order to preserve the periodicity of level set field, imaginary/ghost bubbles are generated for the ones close to periodic boundaries.

[3] The GUI used to specify boundary and initial conditions for PHASTA simulations provided by Simmetrix. Inc.



```
 1     1.833640E-02     5.135717E-04    -4.652026E-03     3.254634E-04
 2     1.844474E-02     6.257648E-03     1.870520E-04     3.254634E-04
 3     1.848274E-02    -1.926961E-04     1.118195E-03     3.254634E-04
 4     1.848797E-02     2.068854E-03    -2.104239E-04     3.254634E-04
 5     1.857370E-02     1.374699E-04    -6.012397E-03     3.254634E-04
 6     1.866297E-02    -3.272749E-03    -3.841806E-04     3.254634E-04
 7     1.868951E-02    -1.806734E-03    -2.983135E-04     3.254634E-04
 8     1.891059E-02     1.315314E-03    -1.892369E-03     3.254634E-04
 9     1.898628E-02    -6.771970E-04    -3.679586E-03     3.254634E-04
10     1.906600E-02     4.787290E-03    -3.425221E-04     3.254634E-04
11     1.909796E-02    -2.485500E-04    -7.977964E-04     3.254634E-04
12     1.912308E-02    -5.108758E-03    -2.005178E-04     3.254634E-04
```

Figure 3: A screenshot of bubble information file (bubble IDs, x_1, x_2, x_3 coordinates and radii in columns from left to right).

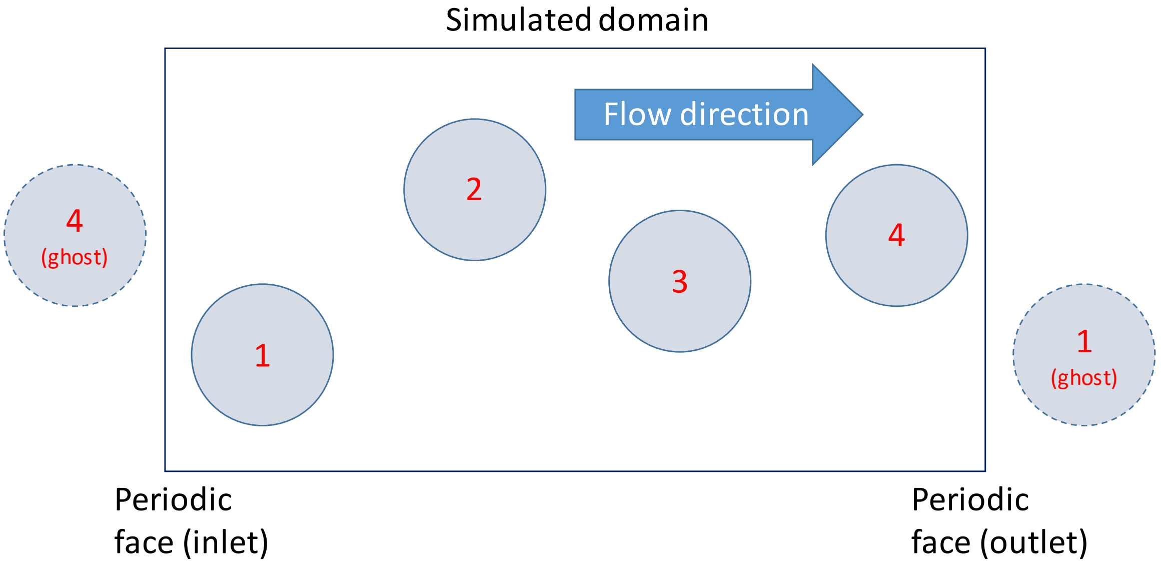


Figure 4: The schematic of ghost bubbles for a one-way periodic domain.

As illustrated in Figure 4, the level set field close to inlet and outlet is determined by both bubble 1 and bubble 4. The influence of bubble 4 on level set distance field at inlet region is preserved by the ghost bubble 4 which doesn’t physically exist in the domain. The periodicity in the distance field is very important for ensuring the initial convergence of the level set method. The code can work for both rectangular channels and single or multiple subchannel geometries. The level set field with periodicity in both x direction and y direction is shown in Figure 5 from a 17 bubble case, in which the void fraction is 1%. Figure 6 shows the initial distribution for a 262 bubble subchannel case with void fraction of 1%, in which the bubbles are colored by bubble ID's.

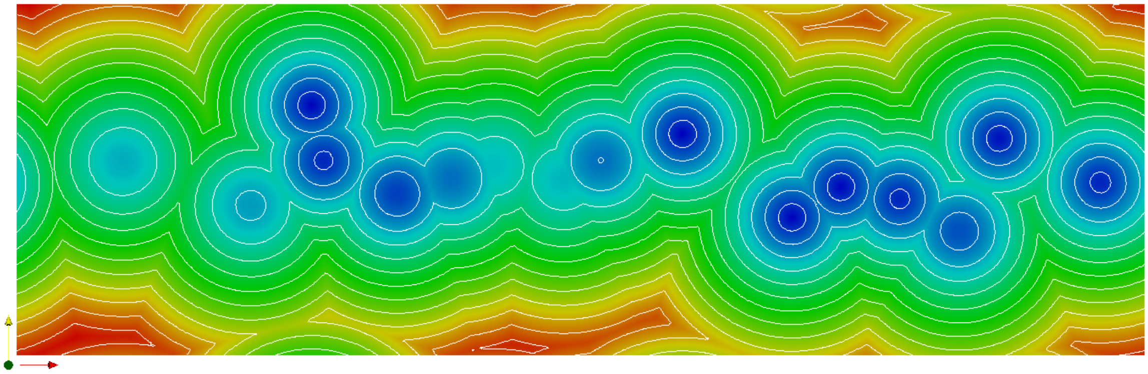


Figure 5: Initial level set distance field for a 17 bubble case.

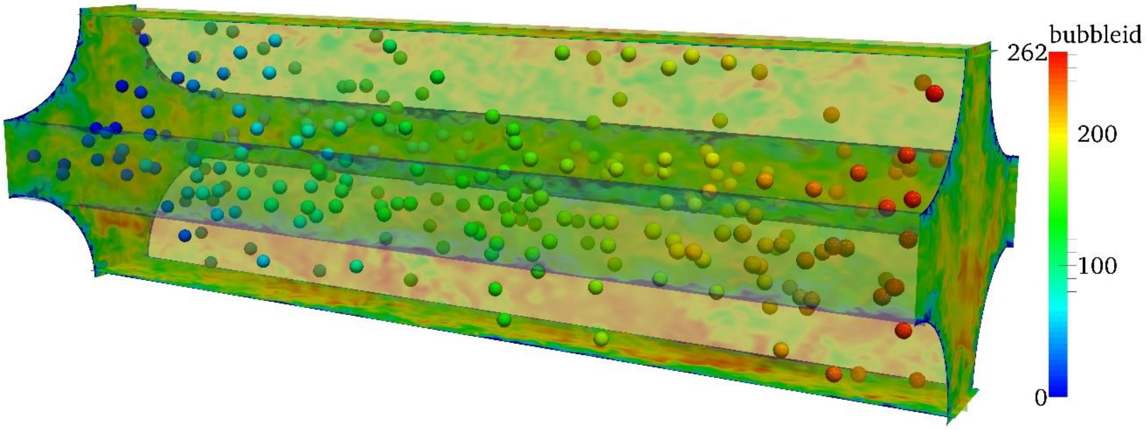


Figure 6: Initial conditions for 262 bubble case with turbulent velocity field.

2.4 Conventional analysis approach

As the major analysis technique before bubble tracking capability, the static virtual probe averaging has helped us gain a lot of valuable insights regarding single- and two-phase turbulent flow behaviors [10, 15, 130]. Specifically, to extract flow statistics, a set of virtual probes are generated and placed in the domain in a flat plane arrangement as shown in Figure 7. These probes are located at several different layers. Each layer has a fixed distance from the wall and a certain number of points in “homogeneous” direction where we expect the same statistical behavior. The probe plane is perpendicular to the mean flow direction. The probe distribution density is designed to be higher in the boundary layer region in order to collect detailed information to study the near wall behavior of the turbulence which is commonly referred as the law of the wall [131].

The quality of statistics is of great importance in numerical simulations. In order to obtain high quality statistical results, the most straightforward approach is to increase the sample size as much as possible. The associated uncertainty can be significantly reduced by averaging quantities of interest over a larger time range for steady state problems. However, for statistically transient cases (e.g. flow development, flow regime transition), time averaging will not provide meaningful statistics. Thus a series of ensemble simulations may be needed to gain better insights in accordance with specific conditions. Each of the simulations in the ensemble has same statistical behavior, but different instantaneous parameters, such as individual bubble locations. Averaging among the ensemble allows for higher quality results with smaller simulation time.

In order to obtain the flow statistics, the data at all probes with a constant distance from the wall is averaged among this set of points and over a certain time window. An example of instantaneous recorded data as well as time averaged quantity are shown in Figure 8. Once the data is obtained at the probe locations, the post-processing step is performed to analyze it. Basic turbulence parameters, such as for mean velocity components and turbulent kinetic energy, can be estimated using the following set of equations:

$$U_i(t) = \frac{1}{N_e}\sum_{m=1}^{N_e}\left(\frac{1}{N_w}\sum_{j=1}^{N_w} u_m^i(t+t_j))\right) \tag{21}$$

$$k(t) = \frac{1}{N_e}\sum_{m=1}^{N_e}\left(\frac{1}{N_w}\sum_{j=1}^{N_w}\sum_{i=1}^{3}\frac{1}{2}(u_m'^i\left(t+t_j\right)^2)\right) \tag{22}$$

where, $u_m'^i\left(t+t_j\right) = u_m^i\left(t+t_j\right) - U_i(t)$is the fluctuation of velocity component-i computed during the ensemble run m at the time instant $t+t_j$; N_e is the number of ensemble runs; N_w is the number of velocity samples in each window, t is the current time, $t_j = (j - N_w/2)\Delta t$ is the local window time, and Δt is the time step. For two-phase flows additional parameters, such as void fraction and phasic velocities are also determined using this basic statistical analysis method. In two-phase flow simulation, the time averaged void fraction is computed using Eq. (23) where X_k is the phase indicator function of phase-*k*.

$$\alpha_k(t) = \frac{1}{N_e}\sum_{m=1}^{N_e}\left(\frac{1}{N_w}\sum_{j=1}^{N_w} X_k(t+t_j))\right) \tag{23}$$

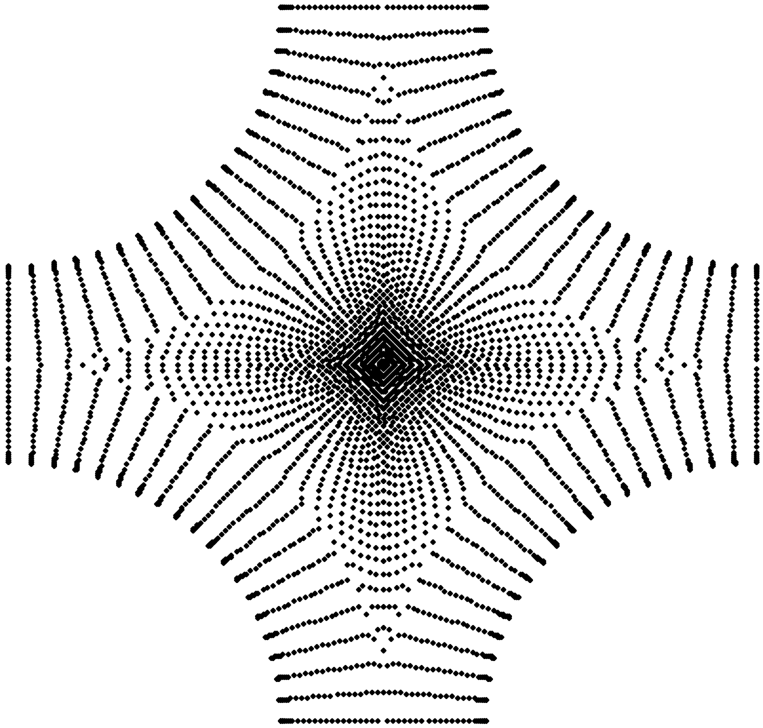


Figure 7: Probe locations at inflow plane of a subchannel domain.

This static virtual probe approach has made indispensable contribution in previous studies [10, 15, 130], especially the simulations of single-phase turbulent flows. However, when it comes to two-phase flow simulations, it becomes increasingly challenging for static probes to collect enough statistic information. According to the current computation capability, the number of bubbles resolved in two-phase simulation is still "not enough" for the desired statistics. During each flow through time, only tens of bubbles pass though the probe plane. In our most recent and large scale two-phase DNS, only 262 bubbles are resolved in the simulated subchannel domain (using about 1B computational elements). The limiting factors include the smaller CFL numbers and more re-initialization loops, in order to ensure the interface separating phases propagates properly. Moreover, if each bubble is allowed to fully deform and evolve, it requires at least 20 points resolution across the bubble's diameter.

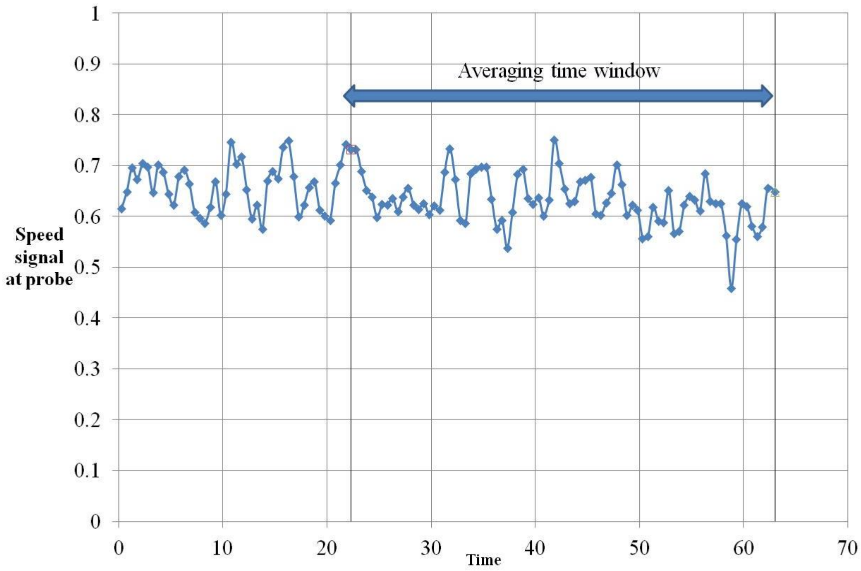


Figure 8: Example of instantaneous velocity signal at a probe locations and time window over which data is averaged to get mean statistics.

Averaging procedure always loses some information, such as the detailed information of individual and instantaneous bubble behavior. This approach is found not very feasible in studying the dynamic transient problems of bubbly flow. For instance, how bubbles interact with geometrical constraints and lead to the instant thermal-hydraulic responses. Another disadvantage of the static virtual probe averaging approach is the lack of capability to extract detailed bubble information. Thus it is impossible to investigate the correlation/dependency between bubble behavior and local bubble/fluid conditions. All these drawbacks suggest the necessity to develop more advanced analysis techniques for two-phase flow simulations. As a result, the objective of current study is to develop the bubble tracking methodology. The detailed discussion regarding bubble tracking capability is provided in the following Chapter.

CHAPTER 3. DEVELOPMENT OF BUBBLE TRACKING METHODOLOGY

3.1 Method development

As discussed in Section 2.1.2, the traditional level set ITM utilizes a signed distance field to represent the phases separated by an interface, and is capable to distinguish the different phases (e.g. gas phase and liquid phase). However, it couldn’t distinguish and track the individual bubbles in the multi-bubble simulations. Historically, the ITM approach could handle only a few bubbles due to high computational cost. In recent years the simulations are increasing in scale, which brings the issue of extracting useful information from hundreds of bubbles efficiently. Tracking the bubbles would allow to collect the detailed information regarding the individual bubble behavior, such as bubble velocity, volume, deformation level, and even the local liquid velocity and shear rate. Such type of bubble information is very valuable for the development of better closure laws that can lead to more accurate predictions of turbulent bubbly flow. As an example, the DNS of bubbly flow can help guide the development of more accurate correlations of interfacial forces [18]. Therefore, a bubble tracking capability has to be developed to enhance the amount of meaningful numerical data extracted from ITS. It will help take the best advantage of computationally expensive large-scale simulations. The implementation of bubble tracking capability is developed here for level set based two-phase simulations, but it is also applicable to other alternative interface tracking approaches, such as VOF method and front tracking method.

In order to identify and track different bubbles, a marker/ID field is generated in the PHASTA source code along the level set distance field and every node in the domain has its own ID. The nodes inside the regions of interest are colored by the corresponding bubble ID while the ID value of a point outside the regions of interest is set to be zero. As shown in Figure 9, the ***region of interest*** usually consists of the bubble region (to collect bubble information) and a near interface liquid shell (to collect local liquid information). The pre-processing bubble initialization algorithm (Section 2.3) produces the bubble center coordinates and the associated ID's, which PHASTA solver is able to read to correctly initialize the ID field.

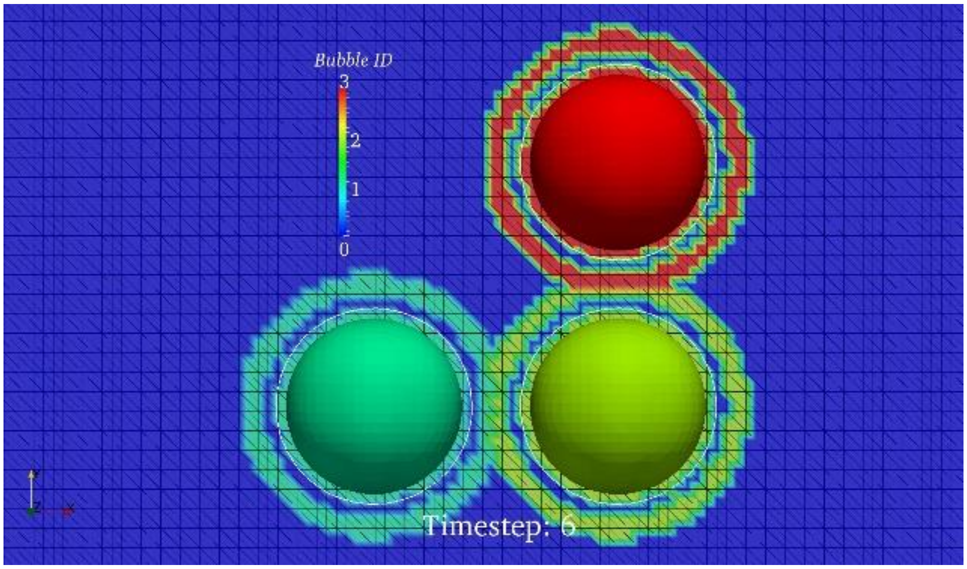


Figure 9: A slice of the domain in a three bubble case with marker field shown (zero value indicates liquid).

Considering the computational efficiency and simplicity, the marker field is designed to get updated at every time step based on the level set value and the marker field at the previous time step (vs. clean re-generation at every step). By checking the level set value, the code evaluates the relative distance of any nodal point from the bubble interface. Illustrated in Figure 10 is the workflow of bubble tracking capability. If the point is outside the regions of interest, the corresponding marker value will get reset to be zero. For the nodes located in the regions of interest, they could fall into either bubble region or local liquid shell. When a point is detected to be inside the bubbles, the code will keep its marker if the old value is non-zero, otherwise, the code assigns the maximum marker from the neighborhood to this point (i.e. the maximum maker in a tetrahedral or hexahedral element). And when the point is detected in the local liquid shell, the code will find out the closest bubble centroid and assign the associated bubble ID. The updating algorithm of ID field is appended in Appendix D.

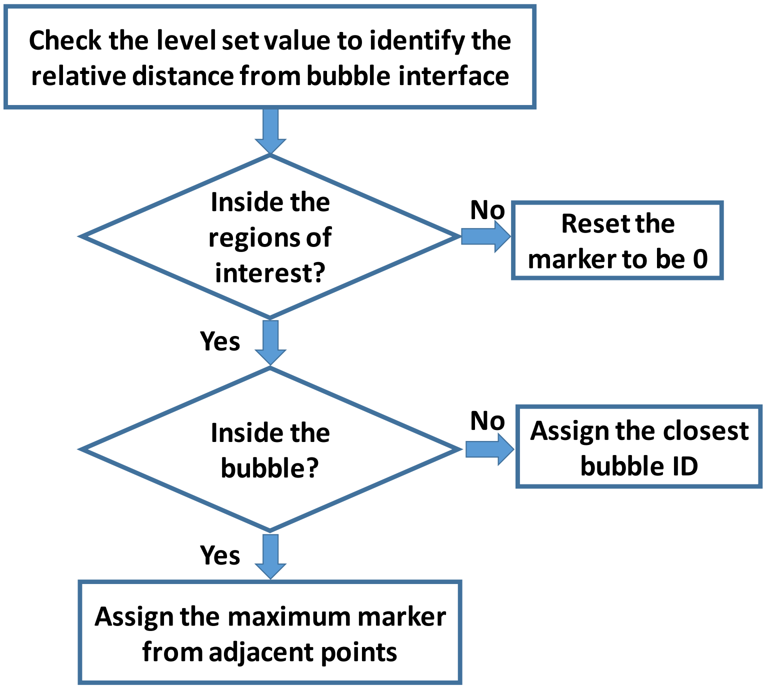

Check the level set value to identify the relative distance from bubble interface
Inside the regions of interest?
No
Reset the marker to be 0
Yes
Inside the bubble?
No
Assign the closest bubble ID
Yes
Assign the maximum marker from adjacent points

Figure 10: The schematic diagram for bubble tracking workflow.

This simple approach works very well especially when the local Courant number in our two-phase simulations is less than 1.0 (which is usually the case in order to properly resolve bubble deformation and interaction). Since PHASTA is a massively parallel code with good scaling performance, we have to ensure that the marker field feature could maintain the scaling capabilities. It has been tested on large simulations (with mesh size up to 1.11 B elements on 65,536 computing cores) that the computational penalty for the marker field tracking did not exceed 7.0%.

In the two-phase flow simulations where multiple (sometimes even hundreds or thousands of) bubbles present, one would expect the bubble coalescence and/or breakup events to happen. Bubble tracking capability is directly compatible with bubble coalescence (no extra coding is necessary to handle those events). When a coalescence event takes place, two separate bubbles will merge into each other to form a new larger bubble. During this process, the larger ID from two separate bubbles will spread onto all elements associated with the new larger bubble (as shown in Figure 11). A practical example of ID field evolution during bubble coalescence will be shown in Figure 23. However, when it comes to bubble breakup, it becomes not that straightforward how one can assign different bubble ID's to new bubbles generated from the breakup event. In order to deal with this challenge, the so-called "bubble ID seeding" is introduced in bubble tracking algorithm.

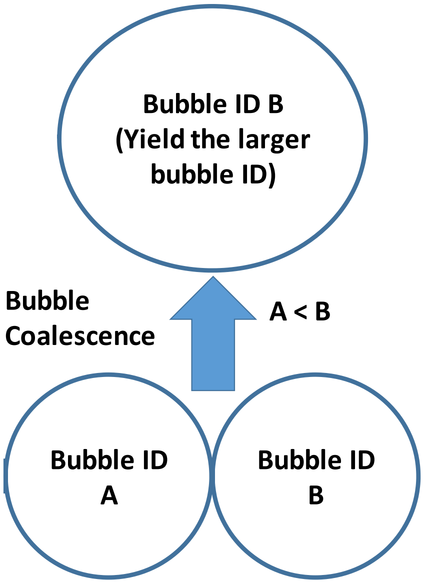


Figure 11: Bubble ID change during bubble coalescence.

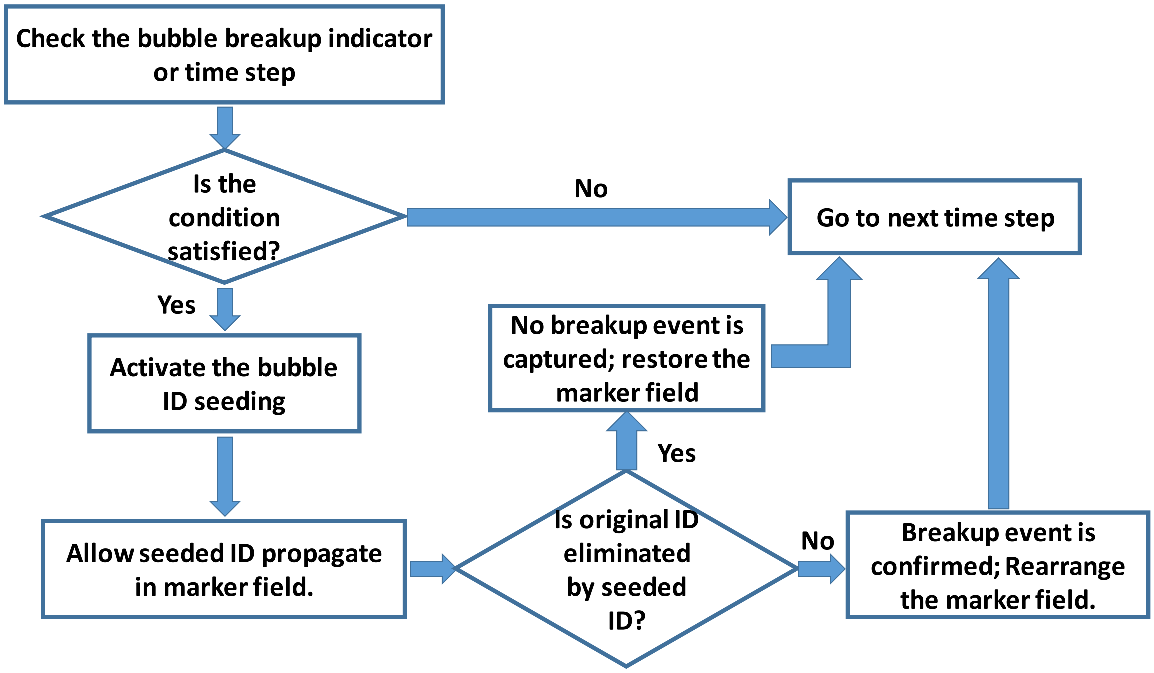


Figure 12: The schematic diagram for bubble breakup tracking.

As an important step in break-up tracking algorithm, the ID seeding can be initialized either when a sudden bubble information change is observed or at every certain time steps (i.e. brute force approach). What this ID seeding does is that a larger ID/marker will be seeded inside the region colored by a certain bubble ID. Please keep in mind that the seeding of a certain new ID is performed for only one point in the domain. Due to the dominating effect of the larger ID value, it will quickly spread within an individual gas/bubble entity. If a breakup event doesn't occur, all the points with same bubble ID should belong to a single entity, and thus the larger ID could reach all of them. However, if a breakup event indeed happens, then a single entity will split into two or more parts, and the larger ID will not be able to recolor all the related points. By comparing the volume occupied by seeded ID (i.e. the larger one) with that by original ID, one could determine whether a breakup event has happened or not. The whole process is illustrated by Figure 12. Together with the correction algorithms of marker field, the current breakup tracking routines are made highly tolerant with ‘fake’ seeding. In other words, the bubble ID field is quite resilient and can recover itself to the state before ID seeding if a suspected bubble break-up event doesn’t occur. Appendix E contains the details of the method implementation.

Besides bubble coalescence and break-up, bubble tracking capability can also automatically handle bubble volume growth and shrinking due to phase-change phenomena. We have now presented a complete bubble tracking capability. The proposed schemes can not only track dispersed bubbles in a continuous fluid, but also handle the bubble ID rearrangement when coalescence or break-up takes place. As readers may notice, the whole bubble tracking process doesn’t involve the calculations of partial differential equations (PDE). Thanks to this feature, the computational overhead associated with bubble tracking is low. Based on the bubble tracking schemes developed for level set based simulations, it is also possible to develop a droplet tracking capability or bubble tracking for VOF or front tracking if necessary. Most challenges with bubble tracking development come from the coding practices. For an efficient parallel flow solver like PHASTA, the possible difficulties in building bubble tracking capability may include (1) setting up a proper communication interface among message-passing-interface (MPI) ranks for advecting bubble ID field; (2) periodicity of bubble ID field when needed; and (3) saving bubble ID field to and reading it from restart files. All these tasks are supposed to be accomplished in a cost-efficient manner for a successful implementation of bubble tracking capability.

3.2 Data extraction

With the newly developed bubble tracking capability, PHASTA is capable to distinguish and track all the bubbles in the flow and extract various local parameters, including bubble’s volume, position, velocity and level of deformation. The bubble volume can be directly obtained by integrating the volumes of all computational finite elements contained within the bubble:

$$\mathrm{V_{bub}} = \sum_{ielem} V_{ielem} \tag{24}$$

where $ielem$ is the index of all the elements belonging to a same bubble. The position and velocity of each bubble are the volumetric average quantities within the region occupied by a bubble.

$$x_i = \frac{\sum_{ielem} x_i(ielem) \cdot V_{ielem}}{\sum_{ielem} V_{ielem}} \tag{25}$$

$$u_i = \frac{\sum_{ielem} u_i(ielem) \cdot V_{ielem}}{\sum_{ielem} V_{ielem}} \tag{26}$$

where $i = 1, 2, 3$ for 3 orthogonal directions in a Cartesian coordinate system.

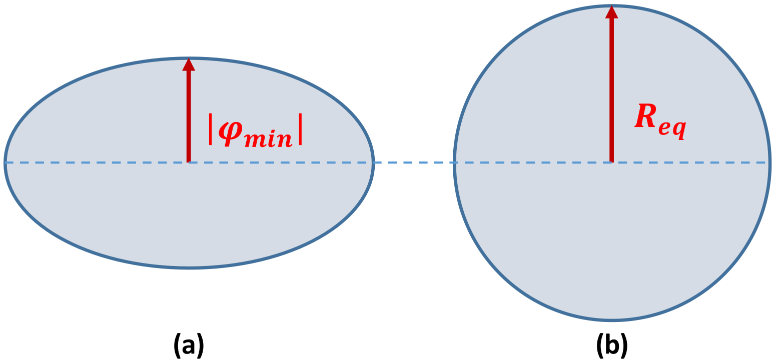


Figure 13: (a) The deformed bubble with minimum level set value inside the bubble region; (b) the spherical bubble with equivalent radius.

To quantify how strongly the bubble is deformed, a deformability factor is introduced as a ratio of the minimum value of level set distance field inside a bubble over the equivalent radius of a sphere that has the same volume as the bubble under consideration. Both minimum level set value inside a bubble and the equivalent radius are illustrated in Figure 13. When a bubble is spherical, the factor is 1.0 and it’s less than 1.0 for deformed bubbles.

$$D_f = \frac{|\phi_{min}|}{R_{eq}} \tag{27}$$

By expanding the regions of interest to cover the local liquid region (shell) near the bubble interface, the code is also capable to collect local liquid parameters, such as local liquid velocity and local liquid shear rate. Average local liquid velocity is computed in the same fashion as average bubble velocity as shown in Eq. (26). As for the local liquid shear rate, a wall distance information (d_{wall}) is used. A wall distance (or distance to the wall) is the closest distance from a domain point to the walls. With this wall distance available, the local shear rate is the velocity gradient between furthest point and nearest point to the wall in local liquid shell (as illustrated in Figure 14).

$$Sr = \frac{u_f - u_n}{\Delta d_{wall}} \tag{28}$$

PHASTA parallel computation is based on a 3-layer hierarchic strategy; the whole simulated domain (1st layer) is first divided into certain number of mesh partitions (2nd layer). In parallel simulations, each compute core/unit is responsible of calculations on one mesh partition, which usually contains thousands of elements (the recommended range is between 3,000 and 100,000 elements per each partition). In order to manage the simulation in an organized manner, smaller mesh unit, called mesh block (3rd layer), is defined (which typically has 64 elements). These blocks are necessary to take advantage of the cache memory in each processor and to reduce the core/shared memory communication. In bubble tracking simulations, the data extracted at the block level will be transferred and assembled at the partition level. The MPI (Message Passing Interface) operations will then summarize the bubble information based on the feedback from all compute cores/ranks. The data extraction algorithms at all three layers are attached in Appendix F.

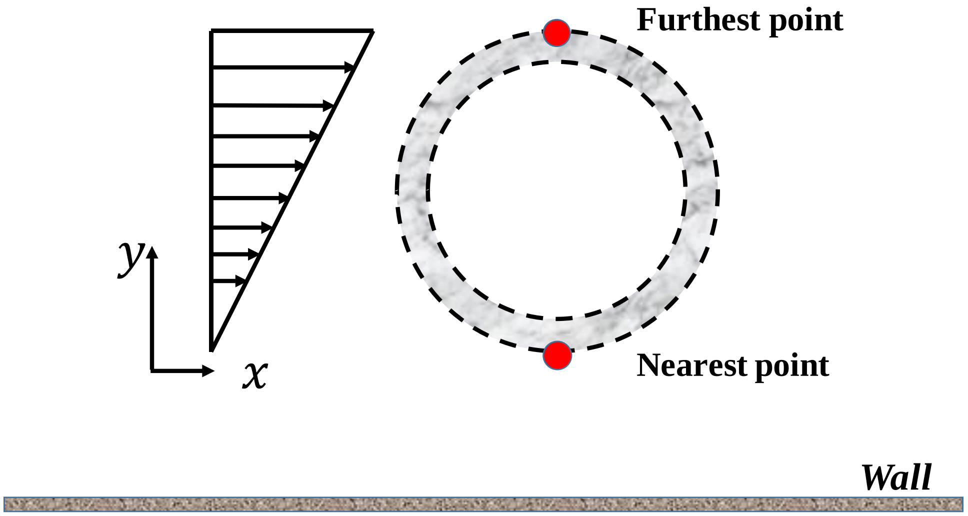


Figure 14: Schematic of local liquid shell and furthest and nearest point with respect to the wall.

Experimental data has always been crucial in the development and validation of multiphase flow models. However, it is very challenging (if not impossible) to measure some quantities in experiments, for instance, the local shear rate in the turbulent bubbly flow experiments. These quantities are very important and can give us valuable insights regarding bubbles' behaviors under different conditions. The bubble tracking capability can be used in large-scale DNS coupled with ITM to collect detailed information regarding the individual bubble behavior and correlate it with bubble parameters, which will help develop more accurate closure laws for multiphase computational fluid dynamics and lead to a higher quality prediction of two-phase turbulent flows in current and future generations of nuclear reactor designs.

3.3 Method verification

The bubble tracking capability is developed to improve the data extraction efficiency and produce a high-fidelity database regarding detailed bubble behavior. For this purpose, there are two basic prerequisites: (1) the reliable performance of bubble information collection, and (2) the compatibility of bubble coalescence and breakup. Both of them are tested in this section.

3.3.1 Data collection performance

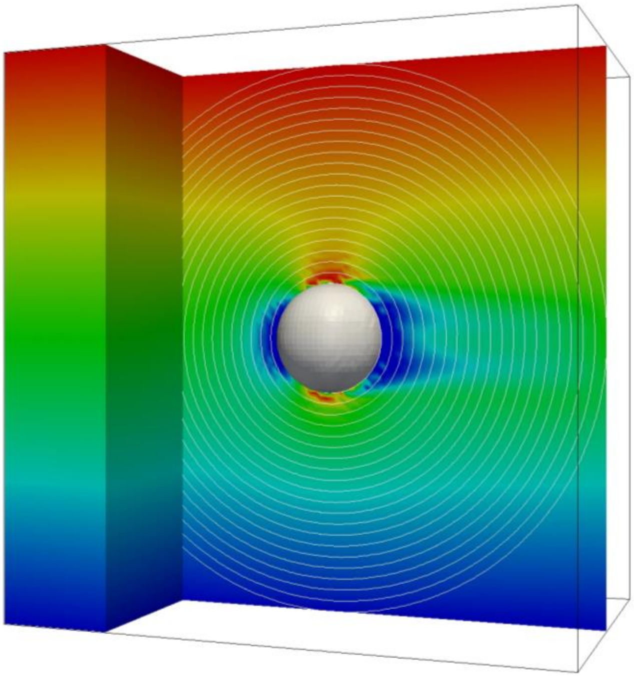


Figure 15: A controlled bubble in laminar uniform shear rate flow.

In order to evaluate how well our local analysis approach obtains the parameters of interest, it has been decided to test it on two well-controlled cases (as shown in Figure 15). This way we have prior knowledge about the parameters of interest (e.g. relative velocity, shear rate, bubble position) since we utilize small domain and control the bubble.

Two demonstration cases were chosen and run with both PID controller (which will be discussed thoroughly in Section 4.2) and bubble tracking capability [99, 132]. A bubble is placed at the domain center in the uniform shear laminar flow with prescribed shear rate of 2.0 s^{-1} and 10.0 s^{-1}. The velocity profiles are defined in such a way that the centerline liquid velocity is 0.05 *m/s* in both cases. The thickness of near interface liquid shell for bubble tracking capability is equal to the bubble radius. As expected from our previous experience [18], the bubble can be successfully controlled in both uniform shear flows (Figure 16 and Figure 17). Under the steady state conditions, the magnitude of lift and drag forces is estimated based on the control forces applied to keep bubbles in place. Meanwhile the bubble tracking capability can record bubble volume, relative velocity as well as local shear rate. With these comprehensive parameters, one would be able to calculate the lift and drag coefficients.

As the bubble position is controlled, the relative velocity is obtained by averaging the velocity of all liquid elements in the near interface shell, while the local shear rate is estimated by calculating the velocity gradient between top and bottom of liquid shell in *y* direction. These two quantities are measured in both cases and their evolution is shown in Figure 18 and Figure 19. Provided the bubble is perfectly controlled and stays stationary, the relative velocity between bubble and the liquid around it is expected to be 0.05 m/s which is close to the local liquid velocity extracted (2.32% difference in shear 2.0 s^{-1} and 7.28% difference in shear 10.0 s^{-1}). In the meanwhile, the local shear rates are slightly smaller than the prescribed values.

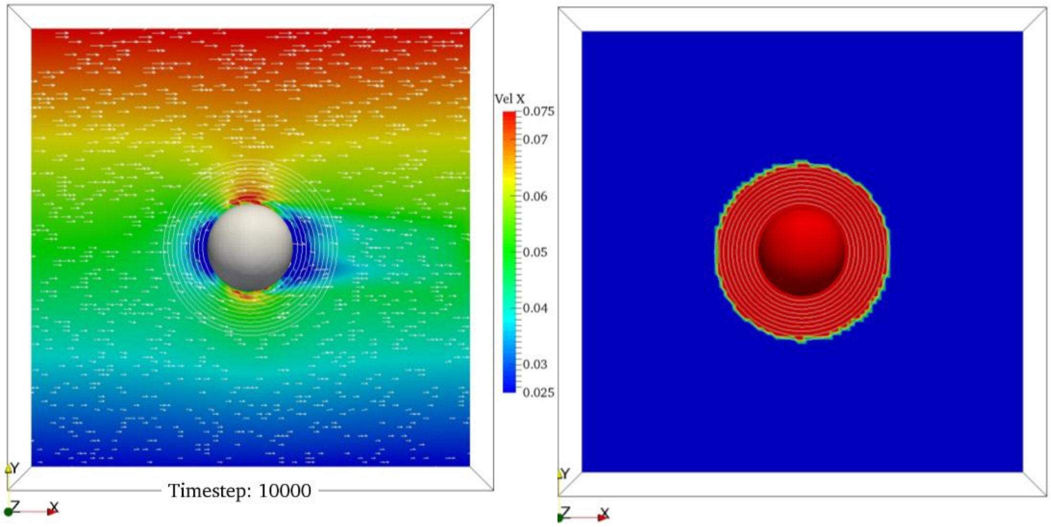


Figure 16: The steady state velocity profile and marker field of well-controlled bubble within shear rate 2.0 s^{-1}.

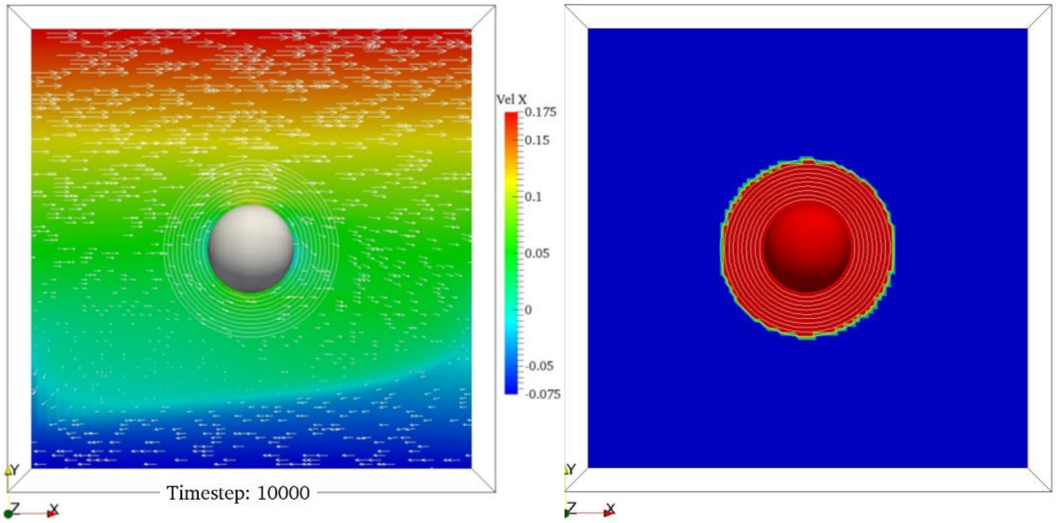


Figure 17: The steady state velocity profile and marker field of well-controlled bubble within shear rate 10.0 s^{-1}.

The lift and drag coefficients calculated in both cases are listed in Table 1. An experimentally based correlation for drag coefficient was developed in [7]. It shows how the drag coefficient depends on bubble Reynolds number (defined using bubble diameter as the length scale) as well as Eötvös number. This correlation predicts the drag coefficient to be 0.16 for our cases. The case with shear of 2.0 s^{-1} gives a closer estimation compared with that from shear of 10.0 s^{-1}, which may be related to the fact that the case with low shear can sustain a more stable velocity profile while the case with higher shear experience some shear induced eddy in the domain (as shown in left parts of Figure 16 and Figure 17).

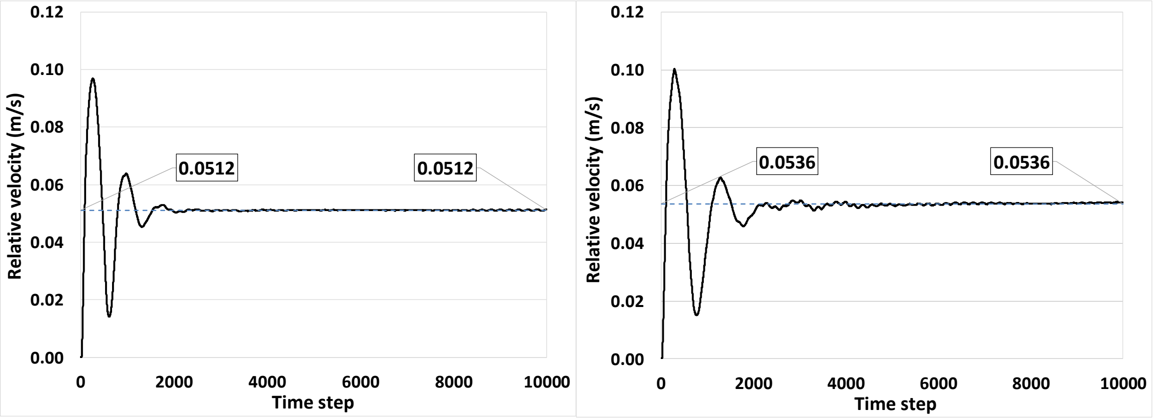


Figure 18: The bubble relative velocity measured in the flow with shear rate of 2.0 s^{-1} (left) and 10.0 s^{-1}(right).

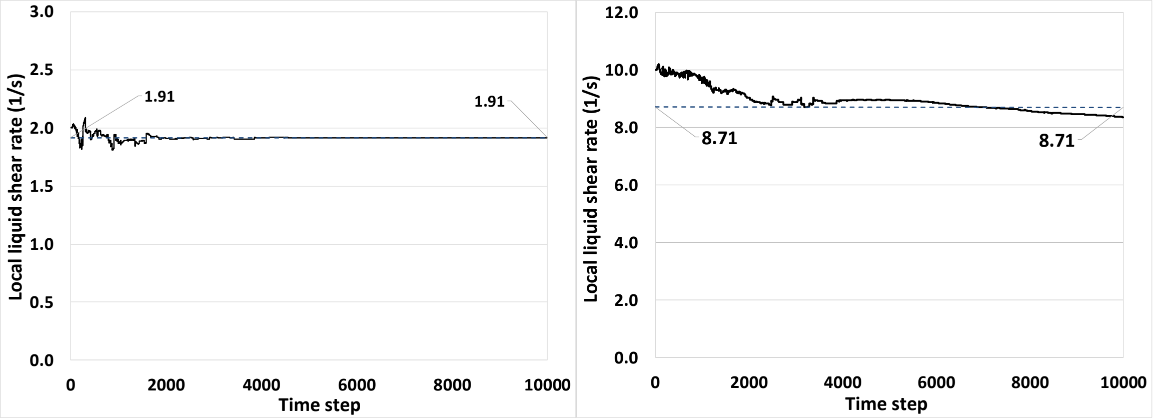


Figure 19: The local shear rate measured in the flow with shear rate of 2.0 s^{-1} (left) and 10.0 s^{-1} (right).

Table 1: The drag and lift coefficients estimated in uniform shear flows

Shear rate	C_D	C_L
2.0 s^{-1}	0.18	0.37
10.0 s^{-1}	0.20	0.37

In order to verify the estimation method proposed for bubble deformability factor, a multi-bubble test case is carried out. There are seven bubbles marked with different colors shown in Figure 20, and they have different radii spanning from 5 mm to 1.1 cm. These bubbles are expected to display different levels of deformation, and smaller bubbles are less deformable while larger bubbles are more deformable with the same surface tension coefficient.

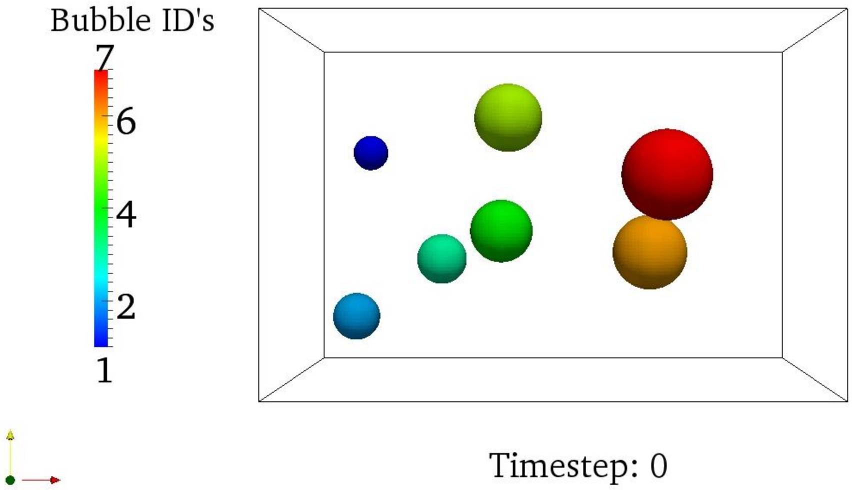
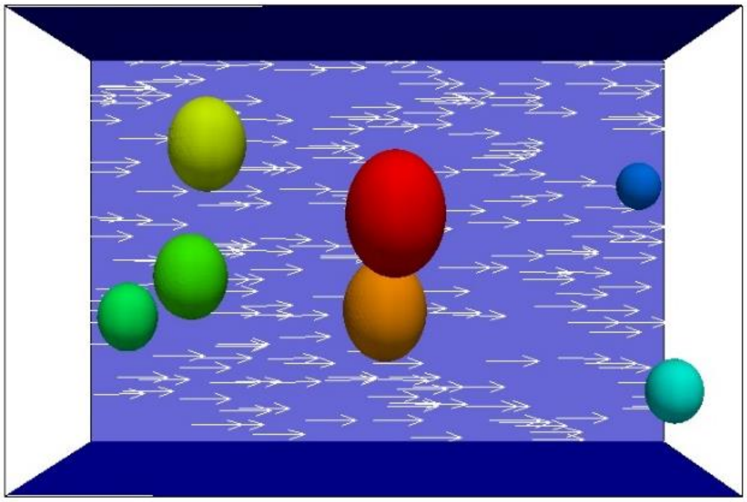


Figure 20: The initial bubble distribution of the demonstration case.

It is observed in Figure 21 that larger bubbles experience more deformation compared to smaller bubbles, as expected. The deformability evolution over time for all bubbles is shown in Figure 22. Initially the bubbles are spherical and the theoretical values of deformability factor are expected to be 1.0. However, the actual deformability factor extracted is always lower than the ideal value due to the finite resolution. The maximum error is about 6% for smallest bubble resolved by only 10 elements across the diameter. Ideally the measured factor can approach 1.0 asymptotically for spherical bubbles as long as mesh resolution is fine enough. A large bubble usually gets resolved by more elements such that its initial deformability factor is closer to ideal factor (1.0). As the simulation proceeds, the bubbles start to undergo different levels of deformation. Although small bubbles' deformability factors are smaller at the beginning, their shape are more stable, in other words, small bubbles can sustain their shapes over the simulation while large bubbles deform. The consistency between observation and expectation further confirms the reliability of data extraction from bubble tracking simulations.



D_f
🟦 0.9395
🟦 0.9404
🟩 0.9244
🟩 0.9077
🟨 0.8841
🟫 0.8645
🟥 0.8525

Figure 21: Deformability factors for bubbles at various deformation levels (the color match is only available in electronic version).

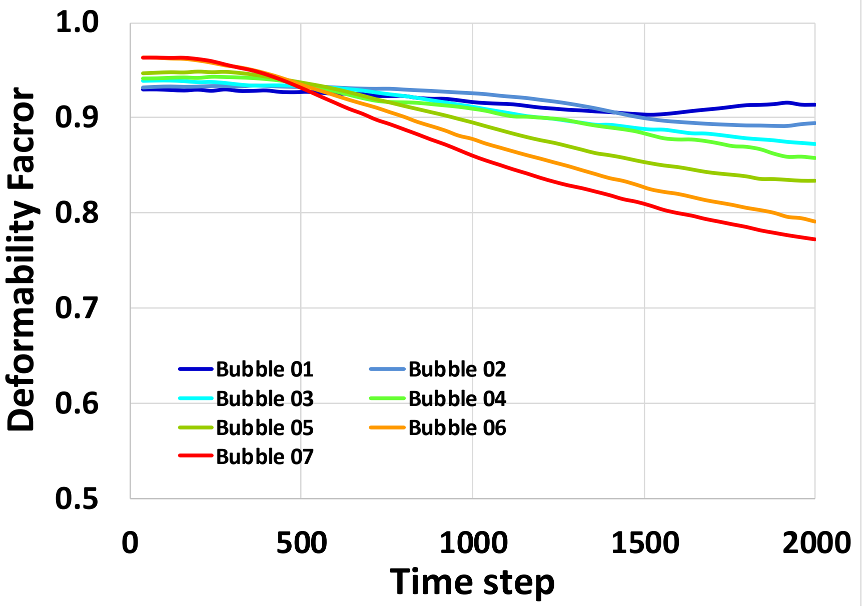


Figure 22: The evolution of deformability factor of all simulated bubbles.

3.3.2 Compatibility with interface topology change

Bubble coalescence and breakup are both major mechanisms involved in interface topology change of two-phase flows [46]. We want to ensure that bubble tracking capability can work properly when bubble coalescence or break-up happens. In order to demonstrate the compatibility with bubble coalescence, a test case is selected where two bubbles are rising in standing water. The smaller bubble is placed below but very close to the larger bubble. As one can see in Figure 23, there are a larger bubble marked by ID 1 (on the top) and a smaller bubble marked by ID 2 (on the bottom). Due to the wake entrainment effect, bubble 2 will catch up and merge with bubble 1. Based on the schematic workflow illustrated by Figure 11, the resultant bubble (i.e. the cap bubble in frame (d)) will be colored by the largest ID among its precedent bubbles, that is, 2 in this case.

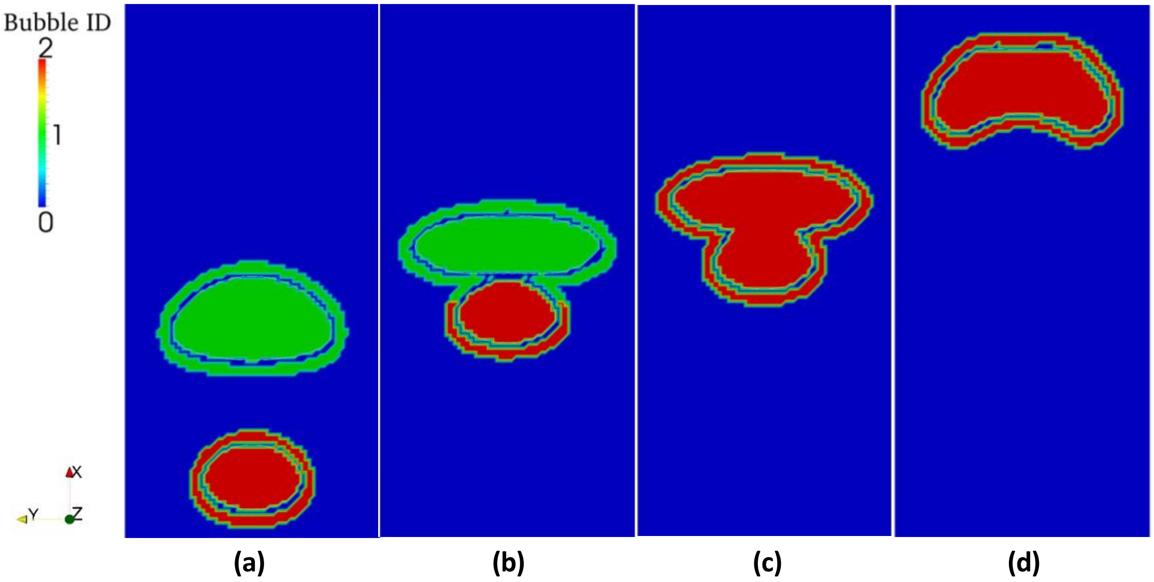


Figure 23: Marker field evolution when bubbles coalesce.

Another test case is carried out to demonstrate the compatibility of bubble tracking capability with bubble breakup event. In the case setup, a gas nozzle is placed on the bottom of a water tank. Gas is injected from the nozzle, and thus the gas entity around the nozzle will grow periodically [133]. Once the gas region becomes large enough, due to the interface instability, new bubbles will be released and depart from the bottom. As shown in Figure 24, a series of bubbles are released from the nozzle, and each of them gets assigned a unique ID, represented by a separate color in marker field.

Bubble tracking capability has been demonstrated to be capable to work with both bubble coalescence and bubble break-up. Together with the reliable data extraction capability, all these wonderful features will be indispensable in the future studies of interesting two-phase flow problems, such as nucleate boiling and flow regime transition. In the next part, we will introduce the post-processing techniques developed so far and how data recorded from bubble tracking simulations could be translated into useful knowledge.

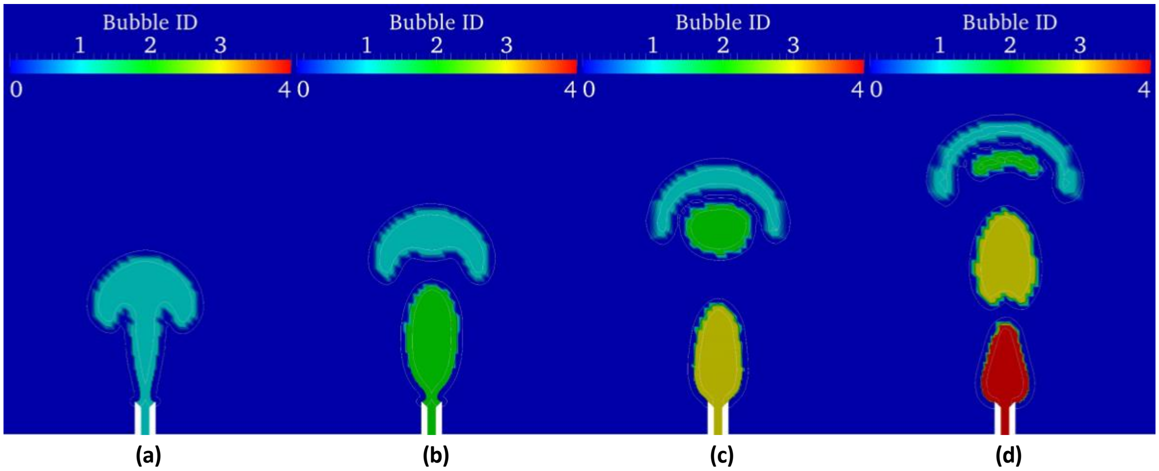


Figure 24: Marker field evolution when bubbles are injected from a bottom nozzle.

3.4 Post-processing analysis

From bubble tracking simulations, detailed bubble parameters can be recorded at every time step and saved into corresponding bubble ID file. By accessing the recorded data, one can have a full history regarding the behavior of all simulated bubbles, such as bubble trajectories, evolution of bubble shape, velocity and so forth. The fundamental numerical data obtained from bubble tracking DNS of two-phase flow will hopefully address a lot of two-phase flow research challenges of engineering interest. The analysis of recorded bubble information is a big topic to be fully explored in the future according to specific research goals. In this section, we will discuss two analysis examples. From these examples, one could appreciate the superior data collection capability of bubble tracking methodology compared to conventional simulation/experiment measurement techniques.

For the purpose of post-processing, a series of analysis codes were developed on the platform of MATLAB[4]. The first example is the reconstruction of instantaneous 3-D void fraction distribution of two-phase flows. Based on bubble movement, the second example shows the estimation of bubble interfacial forces, particularly, the drag force in stream-wise direction and lift force in transverse direction. Combined with suitable bubble partitioning criterion, the bubbles can be classified into several categories and these bubble categories can be studied separately.

3.4.1 Void fraction distribution

Based on the bubble center position and volume, the analysis code can reconstruct the instantaneous 3-D void fraction distribution with respect to different distances from the walls. When a certain distance is selected, a surface (or contour in 2-D) is then defined with all the points having the same wall distance. Sample points/probes are checked uniformly on the surface/contour to determine the probability of gas presence at this wall distance. Specifically, the code will compare the distance between the sample point and all bubble centers with bubble radii. If a point is found to be inside a bubble, then gas point population grows by one. After looping this process over all sample points, the ratio of gas point count over the entire

[4] MATLAB is a multi-paradigm numerical computing environment and fourth-generation programming language, developed by MathWorks.

population of sample points is taken as an estimation of void fraction at designated wall distance. As bubbles migrate in turbulent flows, the dynamic evolution of void fraction distribution can be accurately reproduced by bubble tracking capability.

Spherical shape assumption is good enough when simulated bubbles are generally small. Whereas the deformability factor can be used instead in the void fraction estimation when large deformable bubbles are simulated. Please note that it's much more expensive to obtain the similar quality of void fraction distribution from the conventional static probe approach (detailed discussion is provided in Section 2.4). The same task is as challenging with conventional simulation analysis approach as that in experiments. The void fraction information obtained from experiments is either measured as an average over a long time interval or a cross-section area. A 3-D measurement of void fraction distribution has been developed for sparse bubbly flow (i.e. there is not too much overlap among bubbles within a side view of the flow). Some related experimental effort is mentioned at the end of Section 1.2.1.

3.4.2 Bubble interfacial forces

Bubble interfacial forces are very important due to its critical role in determining the void fraction distribution. The bubble tracking methodology offers a novel way to study the bubble interfacial forces. With the full history of bubble velocity, one can easily get the time derivatives of bubble velocity, namely the acceleration of bubble. The bubble density is prescribed while the bubble size is also available from bubble tracking simulations. Newton second law will inform us the net force each bubble experiences. The introduction of important interfacial forces has been given in Section 1.2.6. Here we will focus on the two dominant interfacial forces, i.e. the drag force in stream-wise direction and lift force in transverse direction. As an application of bubble tracking method, a vertical PWR subchannel geometry is to be studied. More discussion will be presented in Chapter 5. In a vertical conduit like a subchannel, the buoyance force is expected to balance off drag force when the relative velocity between bubble and liquid reaches steady state. Given relative velocity, bubble size and liquid density, one could estimate the drag coefficient for any individual bubble.

In the transverse direction, bubble acceleration is a synthetic effect of both lift force and virtual mass force if average contribution of turbulence dispersion force can be assumed to be negligible. As the virtual mass coefficient for spherical object is 0.5, the virtual mass force can also be estimated because the simulated bubbles in present research are small enough to be spherical. For future simulations with large deformable bubbles, virtual mass coefficient of a certain bubble can be evaluated based on the deformability factor. With bubble acceleration, relative velocity and local liquid shear rate, one can also estimate the lift coefficient of any bubbles. An idea of “bubble averaging time” is introduced to mitigate the noise effect of measured bubble and liquid parameters due to random oscillation of turbulent flow. The bubble averaging time is a short time interval during which the bubble moves one diameter length.

To study the dependencies of drag and lift on distance to the walls, we can apply the idea of “bubble grouping”. Based on the bubble trajectory information, the distance to wall can be computed from all the simulated bubbles. The bubbles can be classified into several groups, for example, near (wall) group, middle group, and far (from wall) group. This treatment allows us to study the drag and/or lift effect on bubbles with respect to different distance to the wall. It has been observed that both interfacial forces exhibit different behavior in different bubble groups (Chapter 5). In the future studies, we can specify more classification bins and propose new correlations for drag or lift forces by considering the dependency of wall distance, or other parameters. Experiments will remain an important cornerstone for the development of new numerical models, but the simulations are becoming increasingly important in model development, especially the DNS of which the prediction is based on first principles calculations. In addition, simulation techniques have better data collection efficiency compared to experimental measurement

CHAPTER 4. DEVELOPMENT OF AUXILIARY TECHNIQUES FOR ADVANCED WORKFLOW

The discovery process of large scale parallel simulations relies on the efficient and flexible simulation workflow. In this Chapter several auxiliary simulation techniques developed in the framework for a better DNS workflow are presented. These techniques include: (1) improved bubble coalescence control based on bubble tracking capability, (2) bubble control technique of a single bubble, and (3) in-situ visualization and simulation steering.

Improved bubble coalescence control now utilizes the bubble tracking information to locate and prevent (or slow down) possible coalescence events, while the previous version developed by Talley [134] used the curvature information of level set contour. New coalescence control approach is more time-efficient, and can achieve superior performance on unstructured meshes.

Bubble control technique is not only useful in the verification of bubble tracking method (Section 3.3.1) but also able to estimate the interfacial force on single bubbles. Thomas et al. [18] have introduced the single-bubble control approach and reported the simulation results of lift coefficient for single bubbles in laminar and turbulent shear flows. The additional study of drag and lift coefficients is documented here as the supplement to our previous work. In the near future, the integration of bubble tracking and bubble control will offer a flexible investigation tool for the interesting underlying mechanisms of the bubble interactions.

In-situ visualization and simulation steering is another cooperative research effort in order to minimize debugging time and the associated computational costs. Since the speed of raw compute power for supercomputer clusters far outpaces disk storage and access speed, solution visualization becomes increasingly expensive for large-scale numerical simulations due to prohibitively huge size of raw data. In-situ visualization offers a new alternative to generate solution pictures without saving raw data to the disk. Moreover, simulation steering would allow users to interactively tune simulation parameters within a single run. This kind of flexibilities will help save a lot of queue time and job launching time, especially when working on state-of-the-art HPC platforms.

4.1 Bubble coalescence control with bubble tracking capability

Physically, when two bubbles approach, a thin liquid film forms between the bubbles. If this film has sufficient time to drain, then the coalescence event will occur. Otherwise, the bubbles will bounce off one another. However, the standard level set method will cause coalescence of any bubbles that approach close to one another, since the level set method uses a smoothed Heaviside function to transition between phase properties. This causes the coalescence process to begin sooner than experimentally observed since the thin liquid film represented by the method has some mixed gas/liquid properties. In order to simulate the coalescence process more accurately, Talley [134] has developed the coalescence control capability within PHASTA based on the curvature information collected from simulations of two-phase bubbly flow. In the simulation of multiple bubbles, the control process involves two important steps: coalescence location identification and control force application.

First, the code detects the locations of suspicious bubble coalescence events. In particular, the code locates all the points that have high curvature value at fixed interface distance (typically 5-6 interface half-thicknesses), and calculates the averaged center among these points. The distance vectors between this center and points under consideration are also computed. The maximum distance vector is the one which has the longest length, and the angle between this maximum distance vector and any other distance vector will be estimated as well. Based on the angles and vector distance criteria, the code will sort and tag the event coordinates. Then an averaging process for individual coordinates can provide the center position of a specific coalescence event. The location identification is repeated iteratively until all coordinates tagged or the maximum estimated coalescence event is exceeded.

After the coalescence event centers are determined in step one, as the second step, a control force is applied in a spherical region centered at coalescence event centers with a radius of five times the level set interface half thickness. This control force is realized by asymmetrically increasing the surface tension of local bubble interface, which acts on each bubble to prevent the coalescence event or slow the process.

The coalescence control model aforementioned has been demonstrated in Talley's master thesis study [134]. It successfully prevented all the coalescence events in simulations of multiple bubble flows up to 32 bubbles in a flat channel with structured mesh (Figure 25). However, this coalescence control failed to deliver satisfactory performance in many latest simulations with unstructured mesh, such as the 262-bubble two-phase simulation within a PWR subchannel geometry and the 895-bubble simulation through a pipe domain. Even in the subchannel case with only 17 bubbles, the coalescence control was not able to prevent most of the coalescence events.

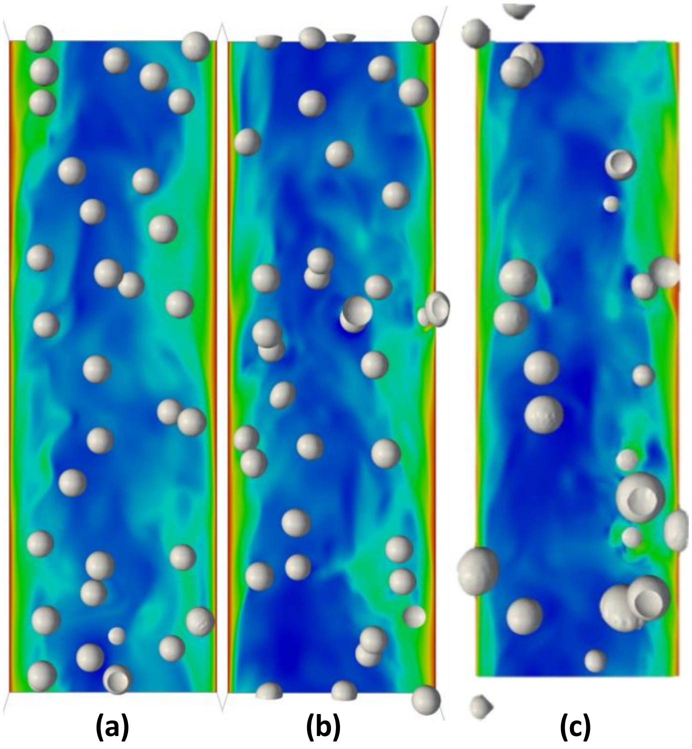


Figure 25: Demonstration case on structured mesh: (a) initial bubble distribution; (b) final bubble distribution with coalescence control; (c) final bubble distribution without coalescence control. [134]

Through careful scrutiny, it was found that the location identification process proposed by Talley has some flaws. These flaws impede the successful realization of coalescence control when many bubbles are simulated in complex geometries or liquid turbulence is very energetic (i.e. high Re two-phase flow). In a dense bubbly flow, a bubble can be easily blocked by other bubbles if one places the virtual camera at the averaged center of all coalescence events. If that is the case, Talley's control scheme has a high chance to miss some ongoing coalescence events.

Moreover, Talley’s coalescence control depends on accurate curvature information of level set contours in liquid phase. However, the accuracy of curvature is not guaranteed in PHASTA simulations for regions far away from the interface because the level set re-initialization process is normally performed only to clean up the near-interface regions. In addition, a prior “maximum number of possible coalescence events” has to be specified in Talley’s control, which makes this scheme less reliable. It should be noted that Talley’s algorithm has been developed for structured grid case, and did performed well in those domains. The transition to reactor geometries has been difficult and the algorithm could not handle the unstructured meshes.

In order to have a more reliable coalescence control capability, we recently further developed the coalescence control by integrating it with bubble tracking capability. In the new control scheme, the old location identification approach is replaced by a simpler yet more reliable approach. With the bubble tracking, the code knows instantaneous positions of all simulated bubbles. If the distance between two bubbles is found to be too close (i.e. smaller than the given tolerance), a suspicious coalescence event is then identified. The associated event center can be approximated by the middle-point of the two bubble centers under consideration. The step 2 of Talley’s control is kept because this force method allows a very cost efficient computation despite of some drawbacks discussed in Talley’s master thesis [134]. Also users don’t have to guess the “maximum number of possible coalescence events” for the new coalescence control. As shown in Figure 26, the new coalescence control has been working very well with the 17-bubble subchannel case in which the former coalescence control didn't work[5].

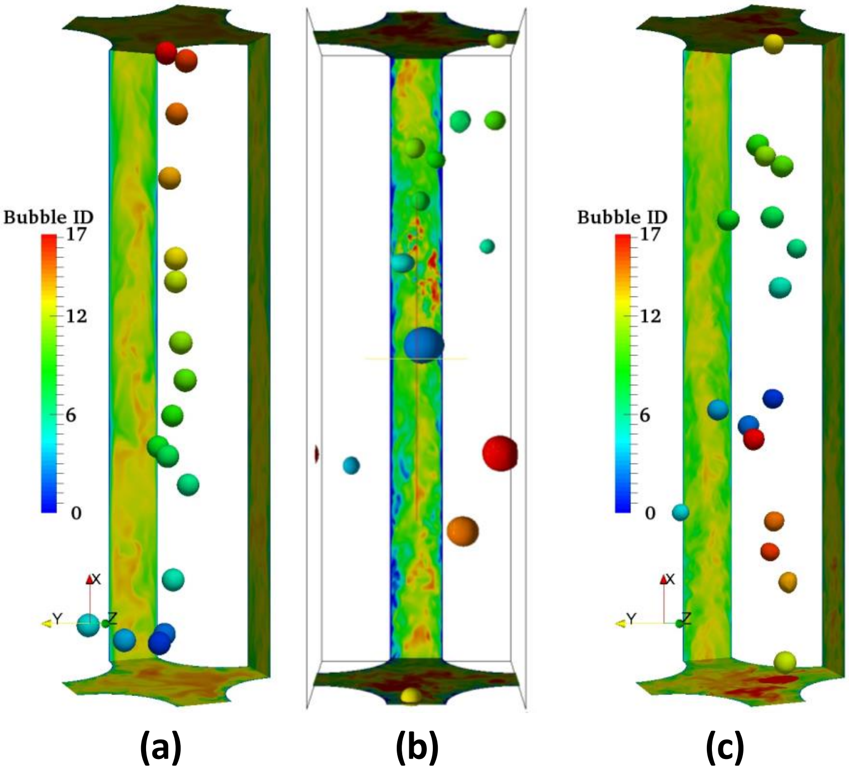


Figure 26: Demonstration case on unstructured mesh: (a) initial bubble distribution and bubbles are colored by their IDs; (b) final bubble distribution with Talley's coalescence control; (c) final bubble distribution with new coalescence control.

[5] In the figure, plot (a) and (c) use the same velocity scale, and plot (b) uses a narrower velocity scale that leads to a larger contrast in velocity magnitude field.

4.2 Bubble interfacial forces estimation

The interfacial forces play an important role in turbulent bubbly flows, which is one of key mechanisms to understand in two-phase reactor coolant flow [18]. Various interfacial forces have been identified and studied both experimentally and computationally, which include standard drag force [135], virtual mass force [48, 136], lift force [44, 45], wall lift force [137] and turbulent dispersion force [47]. The interface force that a single bubble experiences in laminar shear flow can be divided into two major components: drag force in the stream-wise direction and transverse lift force normal to the relative velocity. In order to improve the current interfacial forces database, a novel estimation method was developed by implementing a simple proportional-integral-derivative (PID) controller into our two-phase flow solver, PHASTA (Section 2.1).

External control forces are applied in both lateral direction and stream-wise direction to balance lift force and drag force. Control forces can update themselves based on a proportional-integral-derivative (PID) controller until the steady state is achieved. Under steady state condition, the bubble is fully controlled and kept at a stable position. By collecting control forces applied, lift and drag forces are then estimated based on the force balance relation. The formulation of a typical PID controlling model for low shear rate cases is given below:

$$CF_i^{(n+1)} = c_1\overline{CF}_i + c_2\left[CF_i^{(n)} + c_3\left(x_i^{(n)} - x_{i0}\right) + c_4 u_i^{(n)} + c_5\left(u_i^{(n)} - u_i^{(n-1)}\right)\right] \qquad (29)$$

where CF stands for control force and i denotes the axial direction (e.g. 1 is for x direction or stream-wise direction while 2 is for y direction or transverse direction); $\overline{CF_i}$ is the historical average of the control force; x_i and u_i are volumetric average bubble position and velocity component in i direction; n is the time step index. The control constants used for lift and drag control forces' expressions are listed in Table 2. The constants actually do not influence the force measurements as long as they allow achieving steady-state result. As presented in [18], the PID control force model may involve higher order terms if one wants to control the bubble under more complex flow conditions, such as laminar or turbulent flows with higher shear rates. The control forces have the unit of gravity (N/kg), which will produce buoyance forces in x and y directions respectively to balance off the drag and lift forces. Shown in Figure 27 is the flow profile where a bubble is successfully controlled at the steady state position.

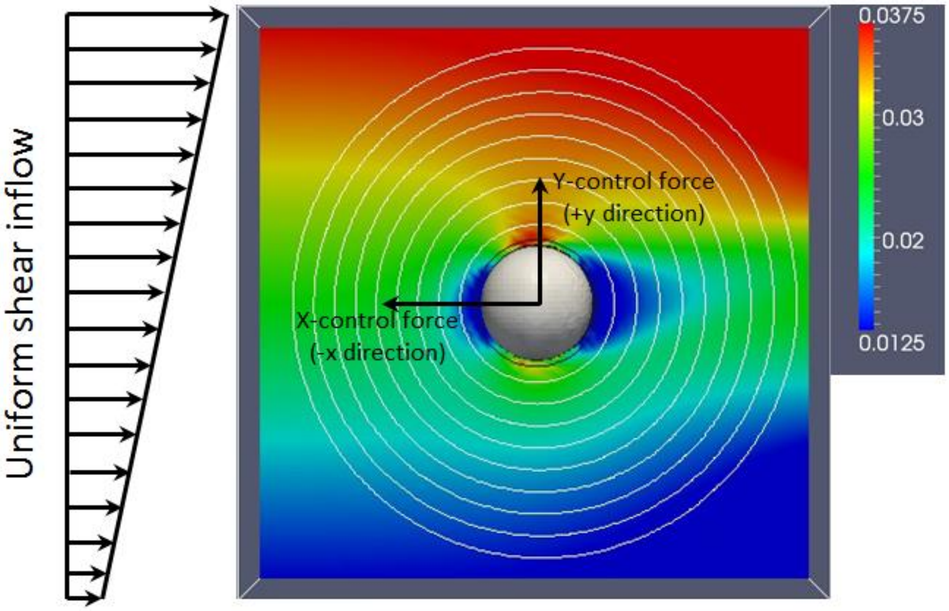


Figure 27: The velocity and level set distance field profile of the uniform shear flow case.

Table 2: The constants used in control force expression

Model Constant	$\boldsymbol{c_1}$	$\boldsymbol{c_2}$	$\boldsymbol{c_3}$	$\boldsymbol{c_4}$	$\boldsymbol{c_5}$
Drag/x control force formulation	0.1	0.9	20.0	4.0	0.0
Lift/y control force formulation	0.2	0.8	20.0	6.0	6.0

During steady state conditions, the lift coefficient can be estimated based on the magnitude of transverse control force:

$$\boldsymbol{F_L} = -\boldsymbol{F_{yc}} = -C_L \rho_L V_b |\boldsymbol{v_r}| \cdot Sr \tag{30}$$

here the relative velocity and shear rate are defined as

$$|\boldsymbol{v_r}| = |\boldsymbol{v_G} - \boldsymbol{v_L}| \tag{31}$$

$$Sr = \left|\frac{dv}{dy}\right| \tag{32}$$

Since the average bubble velocity can be assumed to be zero under steady state, the bubble relative velocity is essentially equal to the prescribed liquid velocity at the centerline. Meanwhile, the shear rate is also a prescribed parameter as part of the case setup. Similarly, the drag coefficient is estimated by solving

$$\boldsymbol{F_D} = -(\boldsymbol{F_{BF}} + \boldsymbol{F_{xc}}) = \frac{1}{2}\, C_D \rho_L v_r^2 A \tag{33}$$

Compared to the experiments, one of the most significant advantages of computational study is the flexibility in parameter control. By choosing a certain control variable, one can easily investigate the dependency of the quantity of interest (e.g. drag/lift coefficient) on the control variable. In the present work, a set of single bubble control cases were simulated to study the dependency of drag and lift coefficients on the bubble Reynolds number (or relative velocity). The definitions of bubble Reynolds number and Eötvös number are given by Eq. (34) and (35). The drag and lift coefficients estimated at shear rate of 1.0 *s*$^{-1}$ are listed in Table 3.

$$Re_b = \frac{\rho_L v_r d_b}{\mu_L} \tag{34}$$

$$Eo_b = \frac{g(\rho_L - \rho_G)d_b^2}{\sigma} \tag{35}$$

Table 3: The drag and lift coefficients at different bubble Reynolds numbers

Case	**R12.5**	**R17.5**	**R25**	**R40**	**R50**	**R60**	**R70**	**R80**	**R90**	**R100**
Relative velocity	0.0125	0.0175	0.025	0.04	0.05	0.06	0.07	0.08	0.09	0.10
$\boldsymbol{Re_b}$	72.93	102.11	145.86	233.38	291.73	350.07	408.42	466.77	525.11	583.46
$\boldsymbol{C_D}$	0.6805	0.4493	0.3172	0.2075	0.1722	0.152	0.1372	0.1266	0.1198	0.1142
$\boldsymbol{C_L}$	0.3596	0.3775	0.3807	0.4086	0.4264	0.4223	0.4142	0.4177	0.397	0.3925

Employing a balance of forces acting on a bubble and available theoretical and empirical correlations of terminal rising velocity, Tomiyama et al. have developed a simple but reliable correlation for the drag coefficient of single bubbles under a wide range of fluid properties [6]. Taking into account the effects of fluid properties, gravity, bubble diameter and the degree of contamination, the C_D correlation for a pure system (e.g. fully distilled water and air) is proposed as

$$C_D = \max\left\{\min\left[\frac{16}{Re_b}(1 + 0.15Re^{0.687}), \frac{48}{Re_b}\right], \frac{8}{3}\frac{Eo_b}{Eo_b + 4}\right\} \tag{36}$$

Figure 28 shows the drag coefficient dependence on bubble Reynolds number compared with this experimentally based correlation. At low bubble Reynolds numbers, the estimated drag coefficients show excellent agreement with the experimentally based drag correlation. The deviation between the correlation and the estimated coefficient is observed for higher bubble Reynolds numbers, which is expected because the controlled bubble starts to deform due to increased relative velocity and relatively large drag force. The bubble shape and level set profile at a Reynolds number of 583.46 is shown in Figure 29 below. Estimation of the drag coefficient for a slightly deformed bubble will over predict a correlation developed for a spherical bubble.

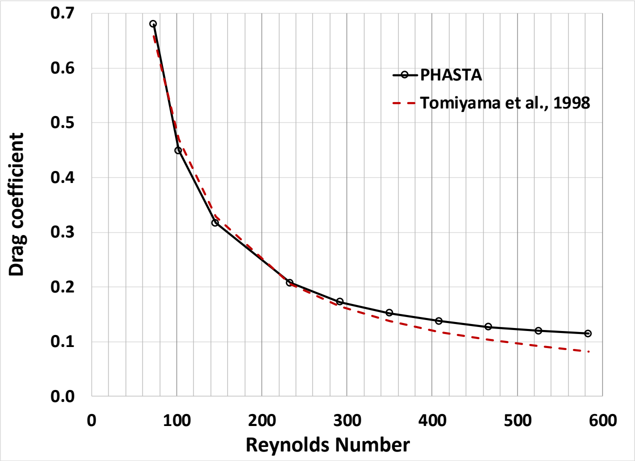


Figure 28: Drag coefficient dependence on Reynolds number.

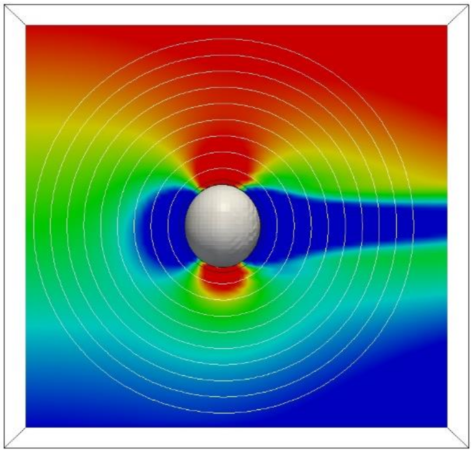


Figure 29: Slightly deformed bubble at Re=583.46.

Besides the drag coefficient, the transverse force-based lift coefficient is also estimated and plotted in Figure 30. It is observed that the lift coefficient doesn't show as a strong dependence as the drag coefficient on bubble Reynolds number. Also, the simulation results are compared with experimental correlation developed by Tomiyama et al. [7]. It was confirmed in Tomiyama's experiments that C_L for small bubbles is a function of the bubble Reynolds number Re_b, whereas C_L for larger bubbles is well correlated with the Eötvös number Eo_b. The correlation of lift force is shown in Eqs. (37) and (38), and compared with the estimated lift coefficient from PHASTA. One should note that the experimentally developed correlation is for high viscosity systems and may not be directly applicable to low viscosity systems, such as air and water. Nevertheless, the similar trend is observed for both Tomiyama's correlation and PHASTA estimation that the lift coefficient stay relatively stable between Re_b of 100 and 600. A similar convergent behavior of lift coefficient is also observed in the results presented by Legendre and Magnaudet [138] for $Re_b > 50$.

$$C_{LF} = \begin{cases} \min[0.288\tanh(0.121Re_b), f(Eo_b)] & \text{for } Eo_b < 4 \\ f(Eo_b) & \text{for } 4 \le Eo_b \le 1.7 \end{cases} \tag{37}$$

where

$$f(Eo_b) = 0.00105Eo_b^3 - 0.0159Eo_b^2 - 0.0204Eo_b + 0.474 \tag{38}$$

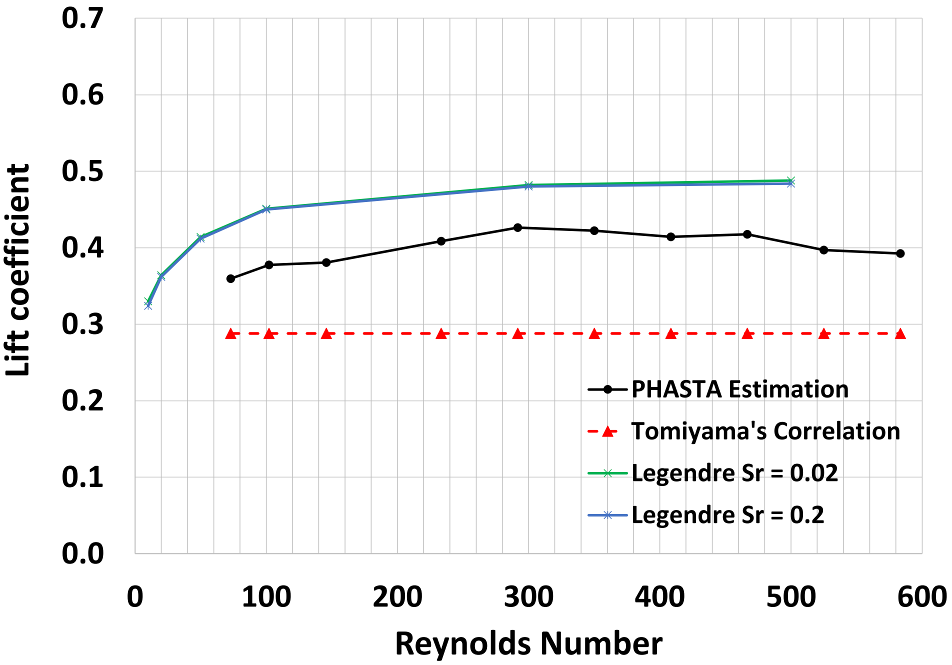


Figure 30: Lift coefficients estimated with different bubble's Reynolds number (where *Sr = 1.0 s*$^{-1}$).

As a supplemented study, a set of ‘near the wall’ studies were successfully completed in order to assess how the lift force acts on single bubbles flowing in regions near the wall of a channel. All ‘near the wall’ low shear laminar flow cases simulated the same shear velocity profile of 2.0 s^{-1}; only the bubble location was altered in the near wall region close to the top and bottom boundaries of the domain. Figure 31 shows the typical setup for three ‘near the wall’ low shear laminar flow cases. The stream-wise velocity spans from 0.025 *m/s* at the bottom wall to 0.075 *m/s* at the top wall. In all ‘near the wall’ cases presented here, the domain height and length are 25 *mm*, the domain width is 12.5 *mm*, and the bubble diameter is 5 *mm*. In Figure 31, the color scale shows pressure, and the rise of pressure inside bubbles is due to the surface tension effect.

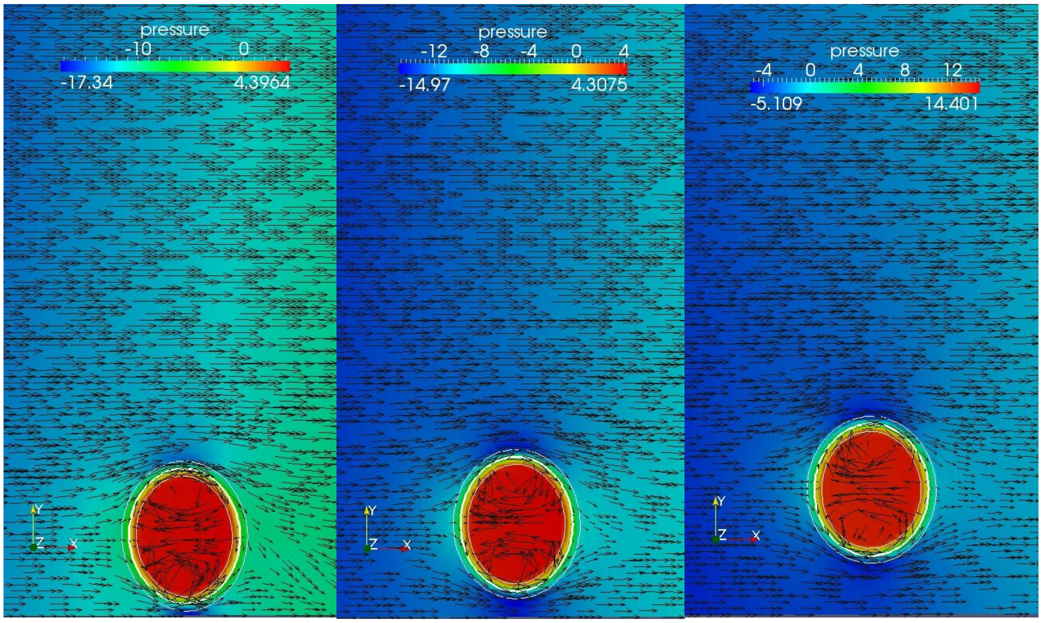


Figure 31: Near the wall lift studies from left to right: (a) 0.25 bubble radius from bottom wall, (b) 0.5 bubble radius from bottom wall, (c) 1 bubble radius from bottom wall.

It has been demonstrated that the bubble control capabilities for resolving the shear induced lift force produce promising results. Table 4 below presents the results obtained from simulating a single bubble in the ‘near wall region’ of a low shear laminar flow regime. The wall distance is measure from the nearby wall to the closest bubble interface.

Table 4: ‘Near the Wall’ lift coefficients for a shear rate of 2.0 s-1

Wall distance (# bubble radii)	0.25	0.5	1.0	4.0	1.0	0.5	0.25
Wall	Bottom	Bottom	Bottom	Center	Top	Top	Top
Relative Velocity (m/s)	0.06875	0.0675	0.065	0.05	0.065	0.0675	0.06875
Lift Coefficient	-0.3297	0.0384	0.3875	0.7049	0.6032	0.9775	1.5466

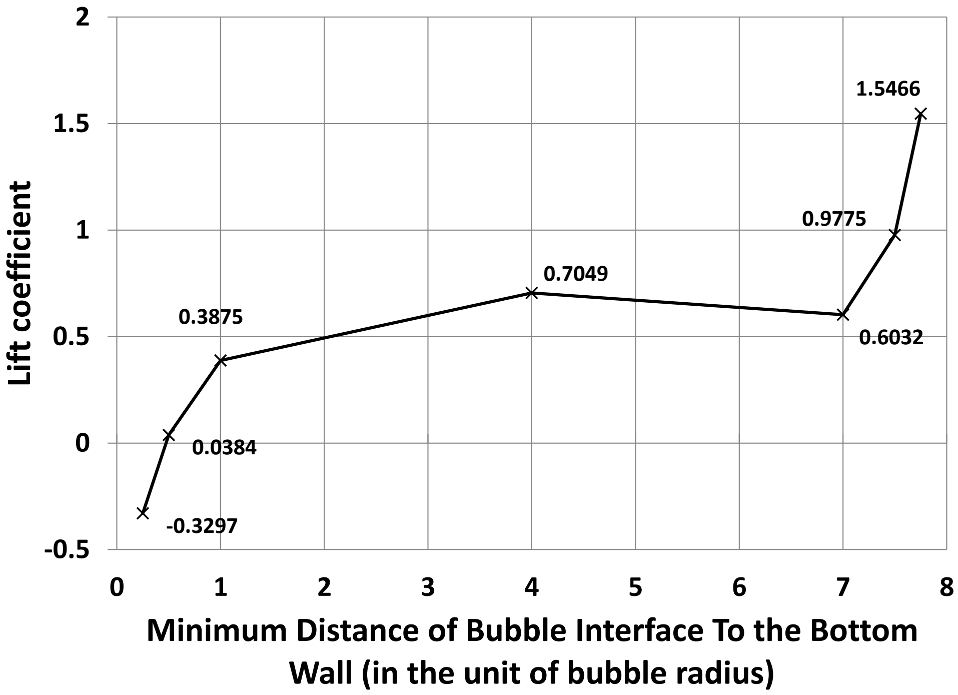


Figure 32: Lift coefficient dependency on distance from the wall (estimated in bubble radii from the closest point of the interface to the wall).

A series of near wall bubble control cases allows one to understand the effect of the wall dependence. From Figure 32 it is observed that the lift coefficient becomes negative as the distance from the bottom wall goes from ½ of a bubble radius to ¼ of a bubble radius. The lift coefficient sign flip in this region of the near wall can be explained by the wall force overpowering the lift force. It is especially interesting to know that there is a certain region near the walls where a small bubble experiences little lateral force, and this mechanism may be related to the aggregation phenomena of small bubbles near the walls observed in both experiments [139] and simulations [10].

4.3 In-situ visualization and simulation steering

4.3.1 Overview

For large scale simulations, there is a remaining challenge about how to efficiently process vast amount of raw simulation data. Over the past decades, the growing speed of raw compute power for supercomputer clusters far outpaces disk storage and access speed. As a consequence, the time spent writing data to and reading data from disk storage is beginning to dominate the time spent in the solver, the data analysis and the visualization. The conventional approach that records these data to disk storage for further post-processing and visualization is no longer a viable way due to prohibitive cost of disk access. To take full advantage of those high fidelity simulations, we have introduced the in-situ processing and visualization capabilities into our solver [115]. In-situ processing and visualization enables researchers to study the full extent of the data generated by their simulations and to monitor and steer the simulation as it progresses [140, 141]. In-situ visualization of a simulation is a technique to couple a simulation with post-processing so visualization occurs while the simulation is running, shown in Figure 33.

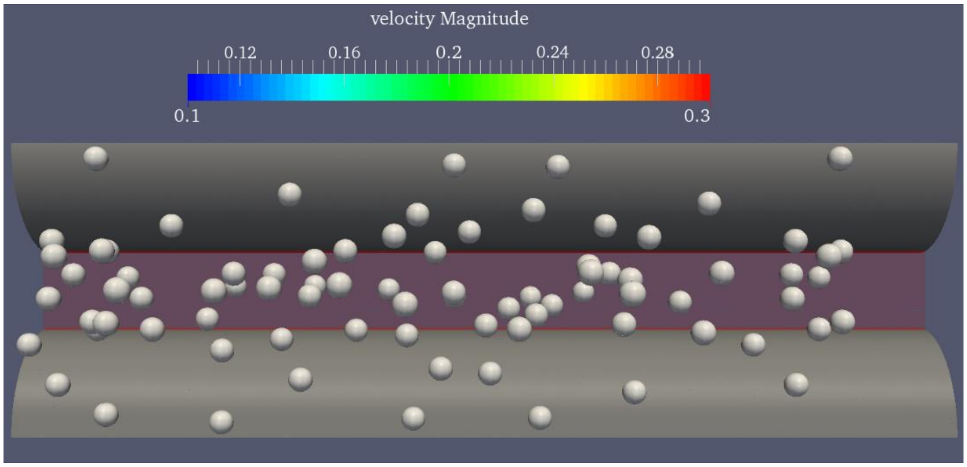


Figure 33: Image produced by in-situ co-processing which shows the real-time bubble distribution.

ParaView Catalyst [142, 143] is used to produce images directly during DNS of coolant flow through mixing vanes using PHASTA. The ParaView Catalyst is embedded within a full ParaView [117] build on both ORNL’s “Titan” and ANL’s “Mira” for the purpose of in-situ visualization. ParaView Catalyst is an open-source C++ library with an external application programming interface (API) to C, FORTRAN, and Python, and is designed to be easily integrated directly into large-scale simulation code for scalable in-situ visualization. It leverages the Visualization Toolkit (VTK) [144] for scalable data structures and algorithms at its core and uses ParaView as the control structure for constructing, configuring, and processing visual analysis pipeline for in-situ co-processing. A typical workflow for using

ParaView Catalyst for in-situ processing includes the implementation of an adaptor code to interface the simulation with the Catalyst libraries, the instrumentation of the simulation code, and the setup of a visual analysis pipeline for in-situ processing (shown in Figure 34).

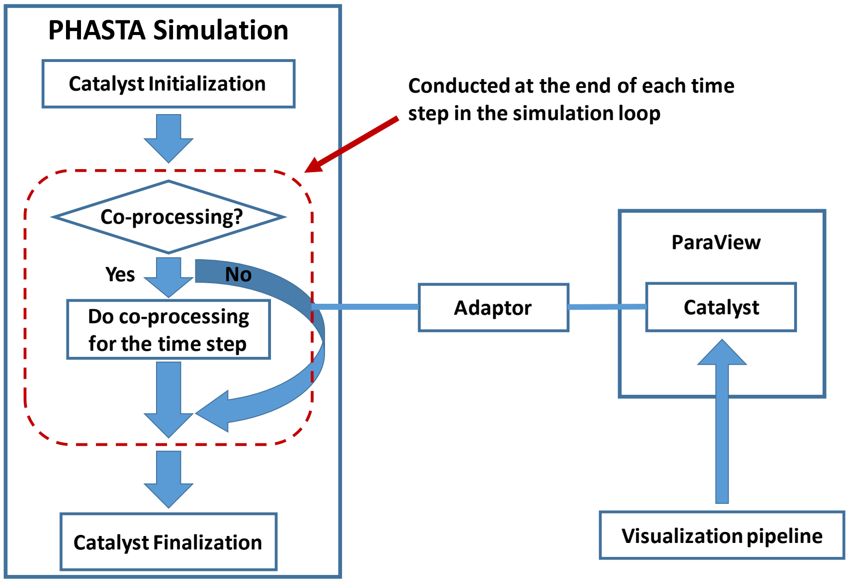


Figure 34: Workflow diagram of using ParaView Catalyst for in-situ visualization.

In addition, live monitoring and simulation steering process can be introduced to the workflow of co-processing and visualizing simulation data in-situ. This simulation steering process allows recreating a fully simulated environment by modifying the right parameters during simulation, hence removing the need to re-launch expensive simulation, which conserves CPU cycles, saves time and money, and accelerate the discovery process. Specifically, the simulation code PHASTA is modified to check at a user-specified time step interval an input file (i.e. numstart.dat file) that PHASTA already uses to retrieve the start time step for simulation. When a change is detected, the updated parameters will be re-read and utilized in the next time step during the on-going simulation. This important feature allows the adjustment of any input simulation parameters to ensure simulation convergence (Figure 35). For large simulations, typical queue wait time can be significant due to the large core count requested (which can reach weeks when more than 50% of the whole machine is requested). Thus, the routine simulation debugging approaches which require job re-submission become quickly prohibitive for large cases. Note that some bugs only appear on a large scale job, and small job debugging may not be an option. On the other hand, the steering allows users not only to adjust the parameters, but also to step-back the simulation to a previous saved state without interrupting the code execution. By using this together with in-situ visualization, the simulations of complex geometries running with large core count, where it is difficult to determine all parameters in advance, become a feasible task.

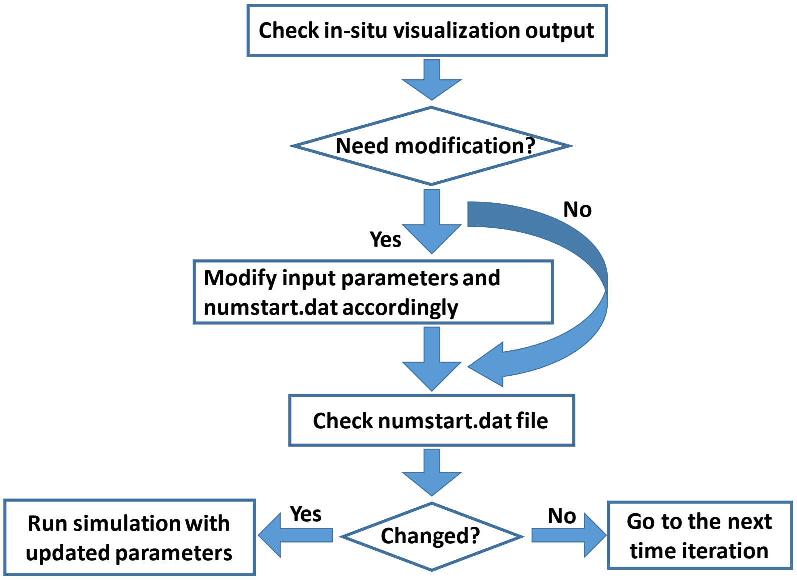

Check in-situ visualization output
Need modification?
No
Yes
Modify input parameters and numstart.dat accordingly
Check numstart.dat file
Yes
No
Run simulation with updated parameters
Changed?
Go to the next time iteration

Figure 35: Workflow diagram that illustrates the simulation steering approach.

4.3.2 Demonstration

Advanced meshing capabilities used in PHASTA code can handle very complex geometries, such as spacer grids and mixing vanes. To test and demonstrate these capabilities, we have performed the direct numerical simulation for 2x2 realistic geometry with spacer grids and mixing vanes. Figure 36 shows the details of the domain. There are 4 subchannels represented by 9 fuel rods. One rod is removed for the clear view of the structure in left figure while all rods are removed in the right one[6]. The flow is driven by pressure gradient to move through mixing vanes, and turbulence is generated as the result. Figure 37 shows the details of a coarser mesh (4M elements, not 167M elements used for the simulations) to demonstrate both complex geometry meshing and boundary layer meshing capabilities.

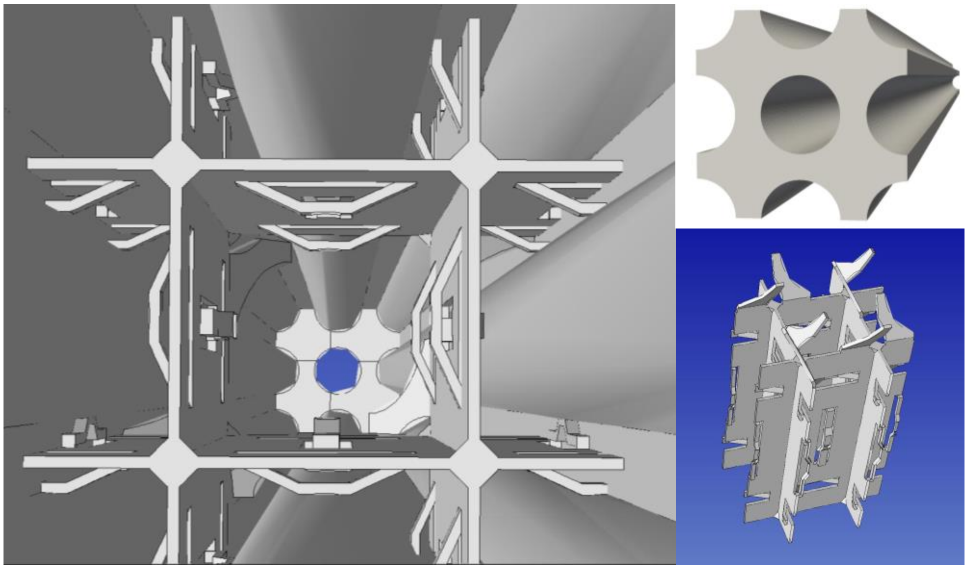


Figure 36: Realistic reactor spacer grids and mixing vanes used for turbulent flow simulations.

[6] This geometry has been provided through CASL collaboration by Prof. Emilio Baglietto (MIT)

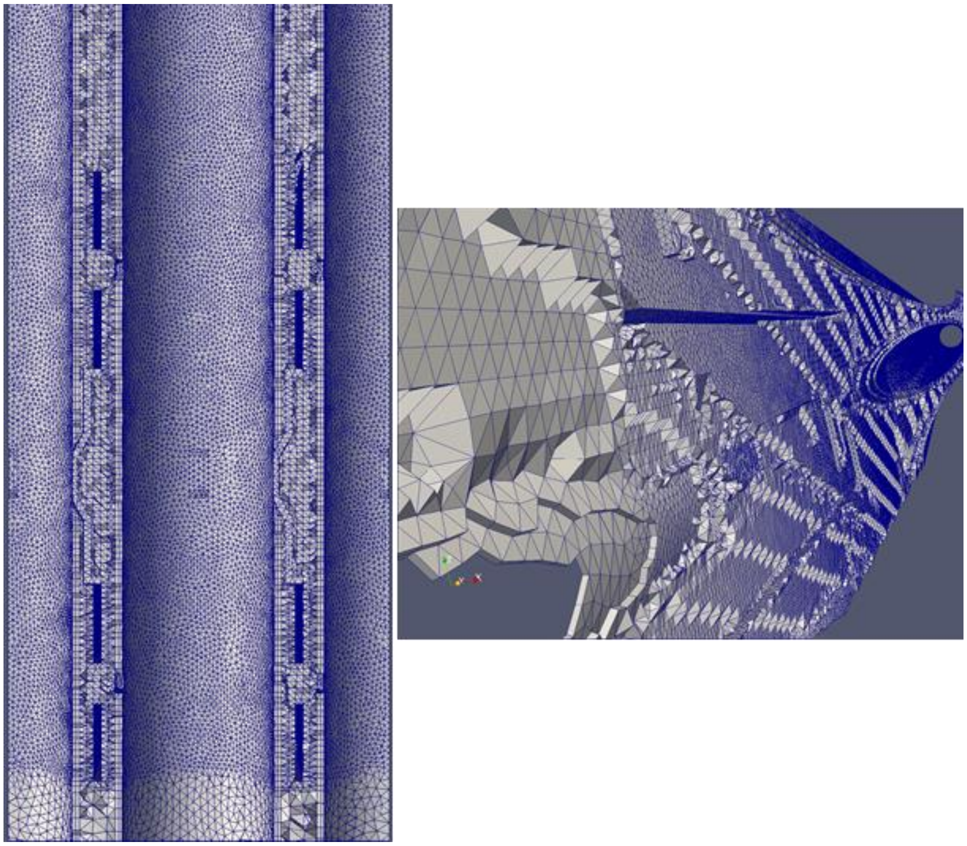


Figure 37: Coarse mesh with boundary layers in spacer grid 2×2 geometry.

The in-situ co-processing and visualization is used in the simulations for 2×2 geometry with spacer grids and mixing vanes, and the close view of turbulence generation at mixing vanes is shown at Figure 38 in the form of vorticity (red and blue structures). From the 3 screenshots of Figure 38, it is clearly observed the propagation of velocity fluctuations. Moreover, the Figure 39 shows the turbulence generation process at mixing vanes from a larger view.

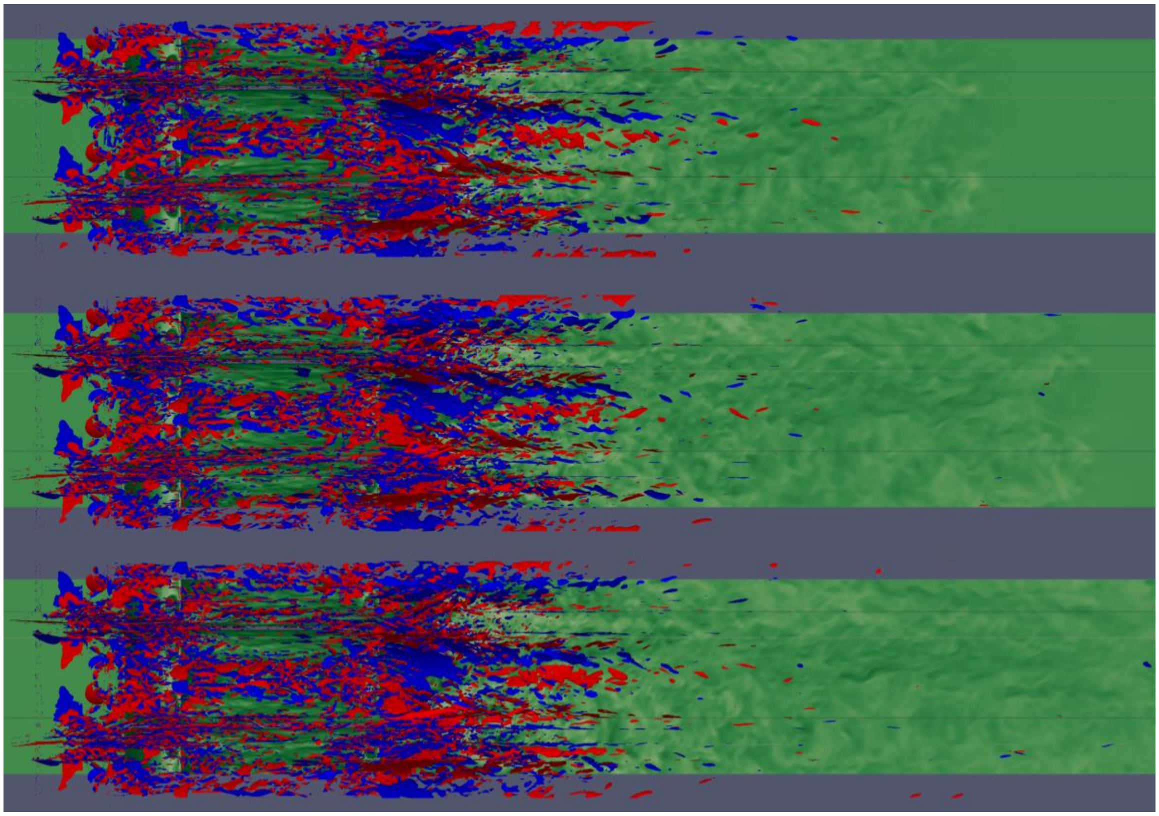


Figure 38: The turbulence generated when flowing through mixing vanes.

If adequate computational resources become available, we plan to perform bubbly flow simulations on the mixing vane 2x2 geometry. According to preliminary mesh estimates, in order to fully resolve the bubbles traveling through the mixing vane in this domain (estimated Reynolds number is about 80,000) is about 16B elements. We have tested the latest state-of-the-art parallel meshing tools to create those meshes, and the largest mesh ever generated in our group is the 2x2 geometry with 16B elements (Figure 40 right).

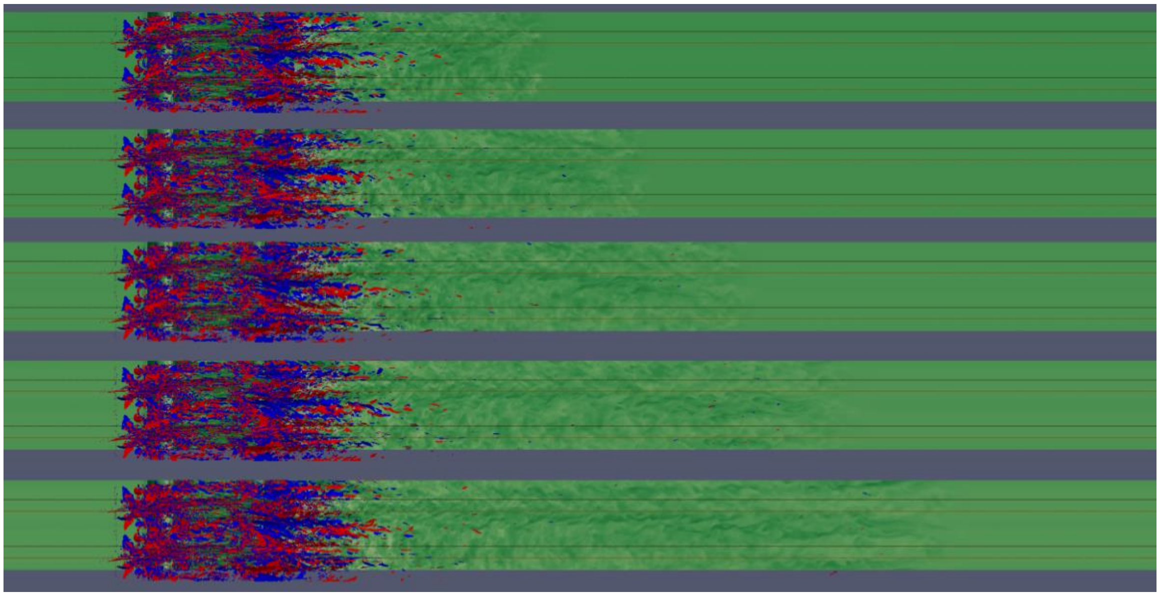


Figure 39: The turbulence generation downstream from the mixing vanes.

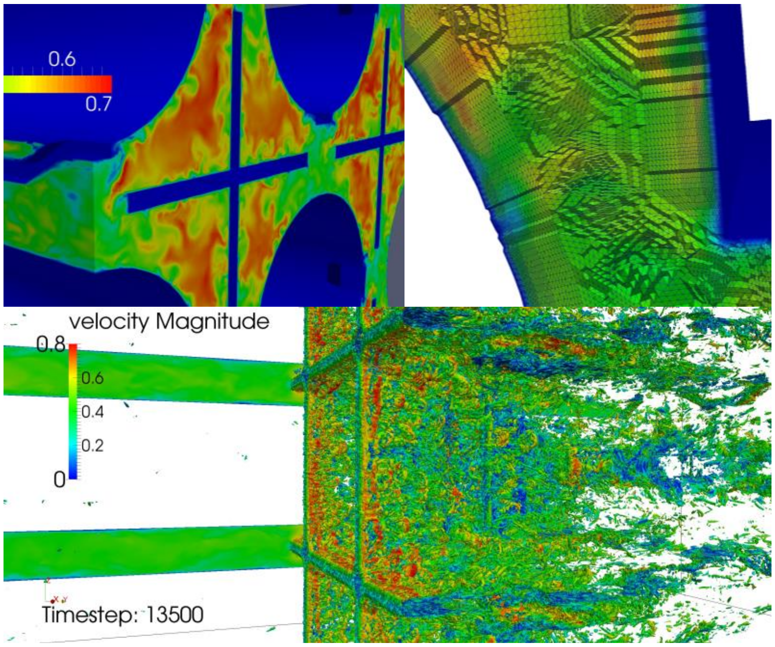
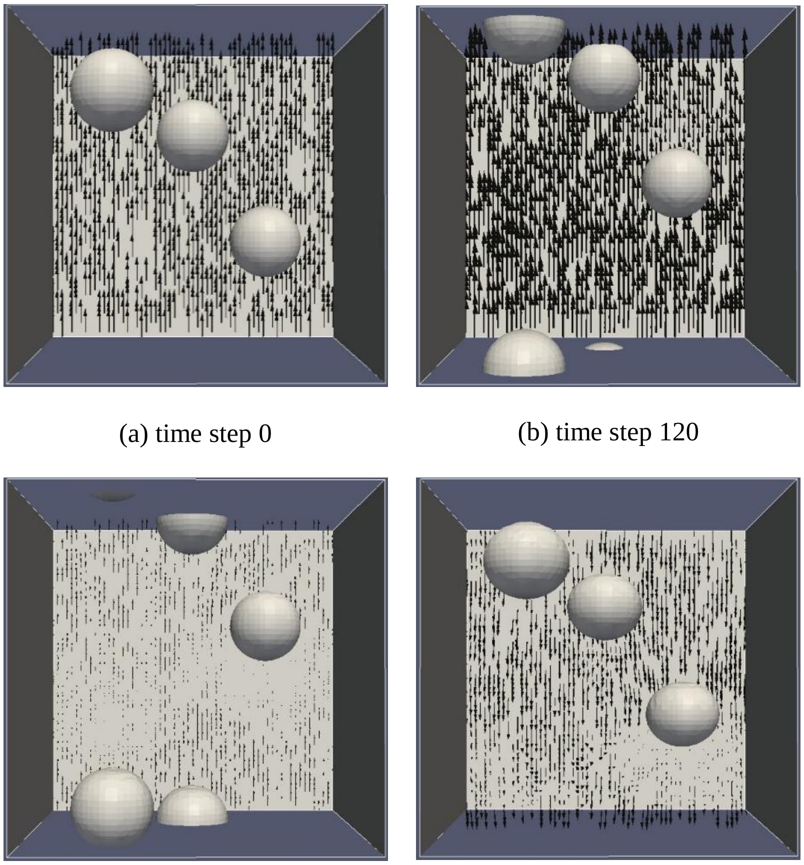


Figure 40: The visualization of the 2×2 geometry (2 B elements) with turbulence and a portion in the 16 B element mesh with refined boundary layers.



(a) time step 0

(b) time step 120

(c) time step 250

(d) time step 400

Figure 41: Demonstration of simulation steering where the value of pressure gradient was flipped at time step 120 from 100 to -200.

As we mentioned, the simulation steering enabled by in-situ visualization allows scientists to modify the simulation parameters as the simulation progresses to steer the simulation in the desired direction. Figure 41 shows the simulation steering example conducted with the in-situ visualization to flip the input value of pressure gradient at time step 120 from +100 to -200 (Figure 41.b), which resulted in a deceleration of the flow which was initially accelerated in the positive Z direction (Figure 41.a), and finally an inversion of the flow direction as shown in Figure 41.c and Figure 41.d where the negative pressure gradient now drives the bubbles in the negative Z direction. The simulation steering capability enabled by in-situ visualization allowed us to adjust the pressure gradient while the simulation was ongoing to reach a desired flow rate and steer bubbles toward the desired direction as needed.

CHAPTER 5. APPLICATION OF BUBBLE TRACKING METHODOLOGY

In the previous Chapters we have covered the details about development and implementation of the bubble tracking capability. This Chapter will address the following questions: how well bubble tracking capability could work with large scale DNS of turbulent two-phase flow; what kind of useful insights only bubble tracking can capture. In order to answer these questions, we will introduce the demonstration effort we have invested for bubble tracking methodology. Due to its relevancy to nuclear engineering, a single PWR subchannel geometry is selected as the computational domain. In the PWR core, fuel rods are arranged together in a bundle using spacer grids. Additional structures like mixing vanes are installed to help generate turbulence in coolant flow to enhance heat removal efficiency. In the presented research, both single- and two-phase turbulence are simulated within a 'simple' subchannel geometry[7] without geometrical barriers like spacer grid or mixing vanes. The Reynolds numbers (Re_h) investigated are 29,079 and 80,774 (based on the hydraulic diameter and mean velocity). In addition, a supplemental subchannel case is carried out with spacer grid and mixing vanes with bubbles moving through the geometrical constraints. The additional case is used to demonstrate bubble tracking performance in studying the geometrical influence on bubble behavior.

[7] For clarification, the simple subchannel case will be referred as 'subchannel' case in this manuscript while the supplemental subchannel case will be called 'subchannel with geometrical barriers'.

The turbulent flow of Reynolds number of 29,530 has been previously simulated in a flat channel [15] and will be compared with the case with Re$_h$ of 29,079 to investigate the influence of PWR geometry on the turbulent flow structures. Since the mesh size for DNS grows exponentially as Re$_h$ increases [91], the Reynolds number of 80,774 is chosen as the effort approaching to the simulations with realistic PWR conditions by considering the state-of-the-art computing resources (e.g. currently #5 supercomputer in the world, IBM BG/Q "Mira" at Argonne National Laboratory). Some preliminary results from the low Reynolds number case (29,079) have been presented in [113] from the limited statistical data available at that time, and since then much larger dataset has been collected to help us better understand the bubbly turbulence phenomena in the PWR subchannel. Both conventional and advanced analysis approaches are applied to the DNS of turbulent two-phase flow in subchannel geometry.

By processing the instantaneous data recorded by static virtual probes, statistical time-averaged results obtained include the mean gas and liquid velocity profiles, void fraction distribution and turbulent kinetic energy profiles. Based on bubble tracking data, the dynamic void fraction distribution and bubble interfacial force study are presented. The most novel aspect of current work is that DNS coupled with both interface tracking and bubble tracking method has been applied to the analysis of turbulent bubbly flows inside the PWR subchannel. The related research effort will help develop more accurate closure laws and ensure a higher quality prediction of single and two-phase turbulent flows for nuclear reactor designs.

5.1 Case design

To create a single PWR subchannel domain, the model is first built in CAD software (e.g. SolidWorks), which can be then utilized by meshing tools (provided by Simmetrix, Inc. in our case) to generate the corresponding unstructured mesh. As discussed in Section 2.2, certain number of boundary layers is specified near the fuel rod surface to capture the detailed information regarding the turbulence in the region very close to walls, governed by well-known law of the wall [145]. The mesh size is 53.8 million elements for the case of Re_h of 29,079. Recent progress in advanced parallel meshing tool allows us to generate much larger meshes to fully resolve the turbulence of higher Reynolds numbers, and for the case with Re_h of 80,774 the mesh created includes 1.11 billion elements partitioned into 131,072 parts. Both the domain overview and a zoom-in view of the boundary layers are shown in Figure 42 for the case of Re_h of 29,079. The length of the subchannel corresponds to about 3.14 hydraulic diameters (40.5 mm), which is calculated based on the cross section area and perimeter shown in Eq. (45). The direction of gravity is opposite to the mean flow direction, which represents an upward flow condition in a vertical subchannel. The cases of two Reynolds numbers are labeled with RE01 (for Re_h of 29,079) and RE02 (for Re_h of 80,774). More detailed discretization parameters are listed in Table 5, including domain sizes and resolutions.

Periodic boundary conditions are utilized to represent a much longer domain than computationally feasible in DNS approach and to be able to achieve statistically steady state flow conditions. The domain is periodic at inflow and outflow planes as well as the transverse faces, and no-slip wall conditions are applied to the fuel rod surface (Figure 43).

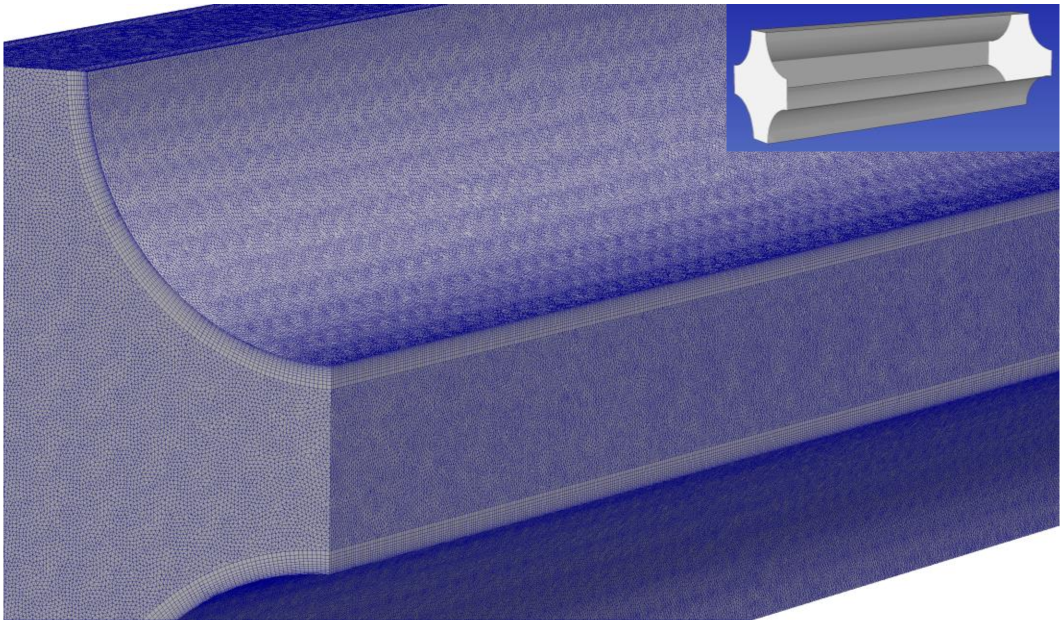


Figure 42: The simulated subchannel geometry and a typical unstructured mesh (of 54M elements) with boundary layers.

The DNS turbulent results for both single and two-phase flows are produced efficiently using a two-step approach. The single-phase turbulent velocity profile is first generated by placing a sphere blockage region at the domain center to create fluctuations (Figure 44). After large turbulence structures are observed the spherical barrier is removed and the flow can sustain turbulence (Figure 45). The statistical data is recorded at this point; the convergent behavior is observed as steady state is achieved as shown in Figure 53. When we ensured that the single phase turbulence has achieved statistically steady state flow conditions by comparing averaged velocity profiles over different time windows, the second step was performed to initialize the bubbles (representing a 1% bubble volume fraction). Since gas bubbles are lighter compared to the liquid, the pressure gradient is adjusted to compensate the density change of bulk flow and sustain a constant mean flow velocity. Bubbles' motion and deformation are resolved using level set interface tracking method. The detailed bubble initialization process has been descripted previously in [113].

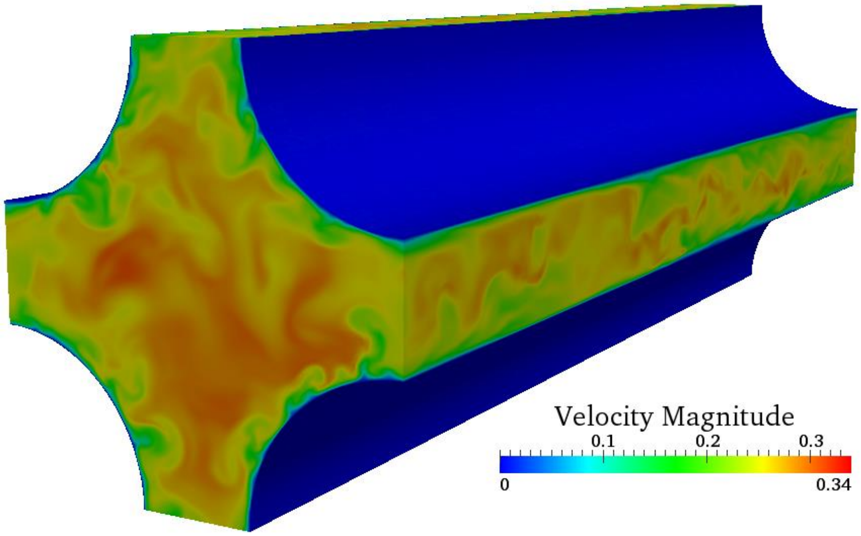


Figure 43: Wall condition in the subchannel simulations.

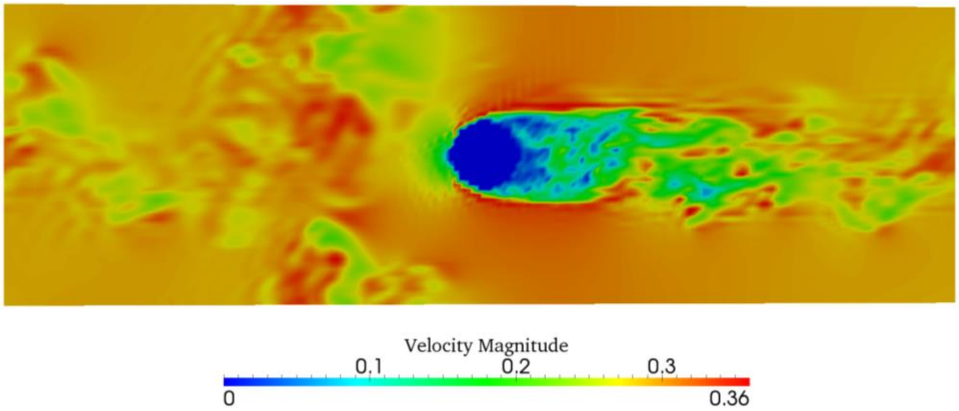


Figure 44: Turbulence is generated by the stationary blocking region.

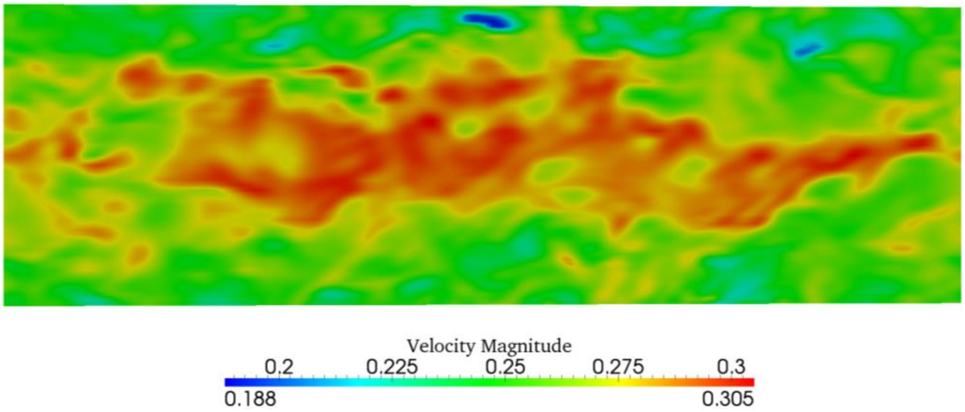


Figure 45: Sustained turbulence after removing the blocking region.

Table 5: Discretization parameters.

Case	RE01	RE02
Domain sizes (mm)	40.5x12.6x12.6	
Rod radius (mm)	4.57	
Aspect Ratio (*P/D*)	1.38	
Reynolds number resolved (Re_h)	29,079	80,774
Resolved bubble diameter (mm)	1.6210	0.6509
Bulk resolution (mm)	8.11×10^{-2}	3.25×10^{-2}
Bubble surface CFL number	0.21	0.39
Maximum CFL number in the domain	7.5	8.0
Thickness of first B. L. (y^+=1) (mm)	8.11×10^{-3}	3.25×10^{-3}
Number of boundary Layers	13	13
Number of points	9,249,506	186,825,949
Number of elements	53,837,248	1,111,168,768
Number of computing cores used	8,192	131,072
Element per core	6,572	8,478

When a statistically steady-state mean velocity is obtained for the coarse mesh, the mesh is refined using PHASTA meshing tools (e.g. “Chef”) and the solution from the coarse mesh is transferred onto the finer one. The finer mesh allows us to capture more flow physics by resolving smaller scale structures which are relatively quick to develop.

Considering both computational cost and quality of the results (based on previous resolution and validation studies) the resolution for bubbles is set to be 20 elements across diameter, which results in 17 bubbles for the 53.8 M mesh and 262 bubbles for the 1.11 B mesh at the 1% volumetric fraction. The scaling studies have shown that the most efficient approach is to use 4 message-passing-interface (MPI) partitions per physical core on Mira supercomputer and this results in computational cost of 26.8K CPU-hours for RE01 case and 730K CPU-hours for RE02 case to achieve one domain flow-through (about 3.14 L/D length). Higher resolution will result in the rapid increase of computational cost while lower resolution is not capable to capture enough details regarding bubbles’ behaviors to reach meaningful conclusions. As shown in Figure 7, a set of virtual probes are designed and placed near outflow plane to record instantaneous velocity fluctuations and bubble distribution across the domain. The bubble distribution and turbulence for 17 bubbles and 262 bubbles are shown in Figure 46 and Figure 47 (the direction of mean flow is upward in the opposite direction of gravity). Interface tracking simulations are run with the bubbles to allow the flow to fully develop and the bubbles to achieve their terminal velocities.

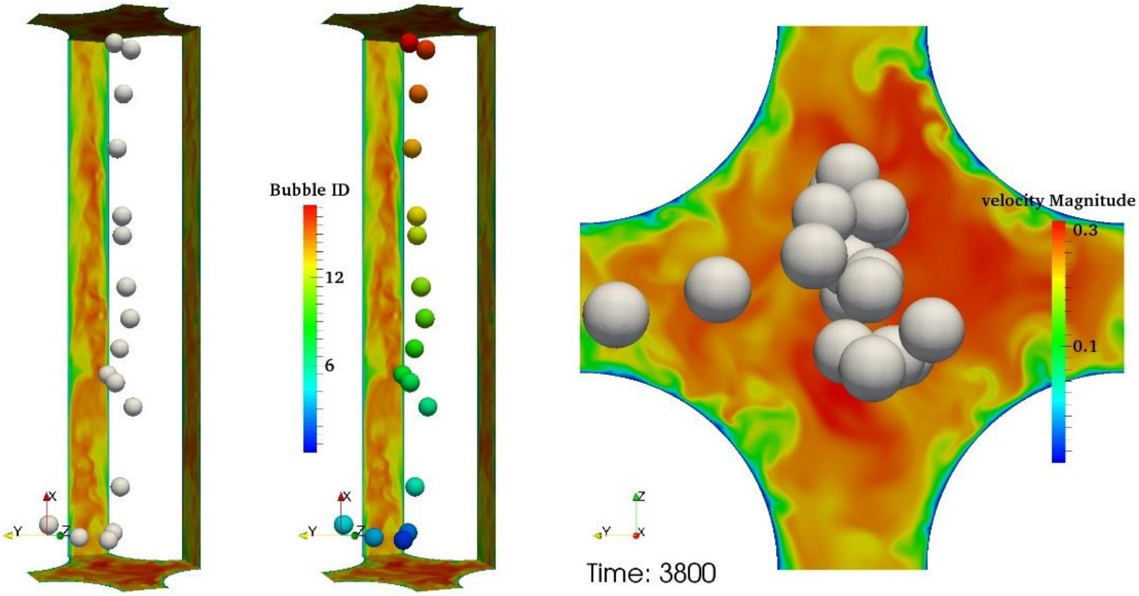


Figure 46: Initial distribution of 17 bubbles in the turbulent flow.

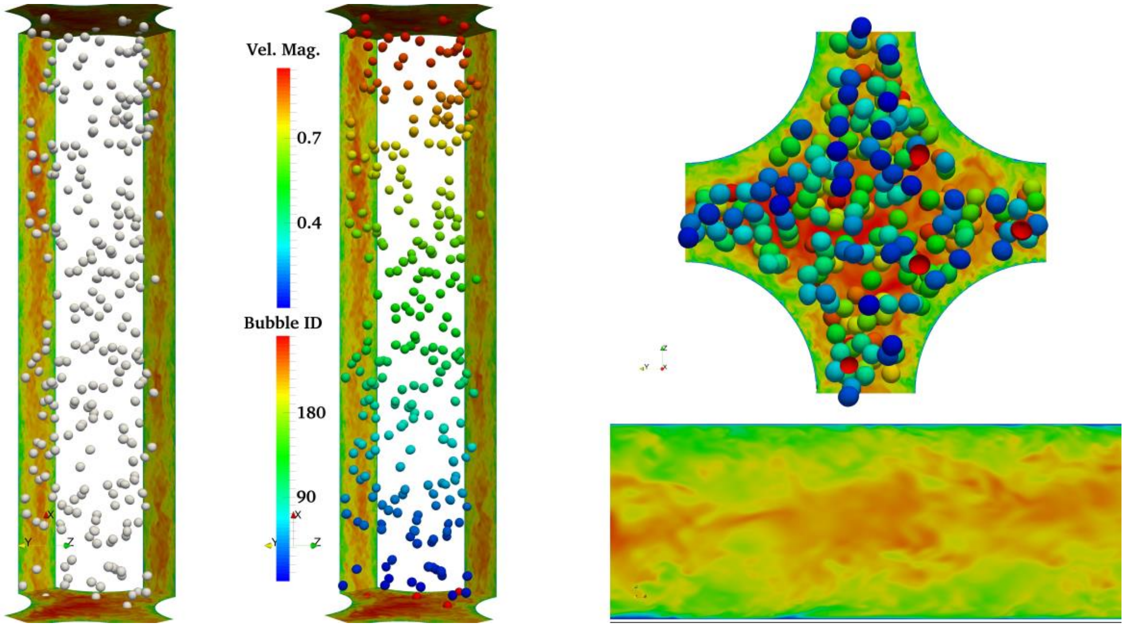


Figure 47: Initial distribution of 262 bubbles in the turbulent flow.

The key computational parameters and fluid properties are listed in Table 6. The viscosities[8] and densities of liquid/gas are determined by using the saturated properties of water and vapor at 300 °C. The estimation of realistic PWR conditions can be found in [146].

Table 6: Fluid properties used in the simulations

Case	RE01	RE02	Realistic PWR condition
Liquid/Gas Viscosities (Pa·s)	8.585×10^{-5}; 1.965×10^{-5}		
Liquid/Gas Densities (kg/m^3)	712.22; 46.17		
Mean velocity (m/s)	0.27	0.75	4.62
Reynolds number (Re_h)	29,079	80,774	452,500

To further demonstrate the advantages of the bubble tracking capabilities in unstructured geometries, a new simulation is being developed which incorporates reduced size PWR-like spacer grid and mixing vanes (Figure 48). The proposed geometry will serve two objectives. The first goal is to demonstrate the robustness of PHASTA in large-scale massively parallel simulations of two-phase turbulent flows with interface tracking, bubble tracking as well as the complex engineering geometry. The second one is to show the great potential of bubble tracking methodology in capturing the two-phase flow behavior and its interaction with geometrical constraints.

[8] The viscosity listed in this dissertation is dynamic viscosity μ.

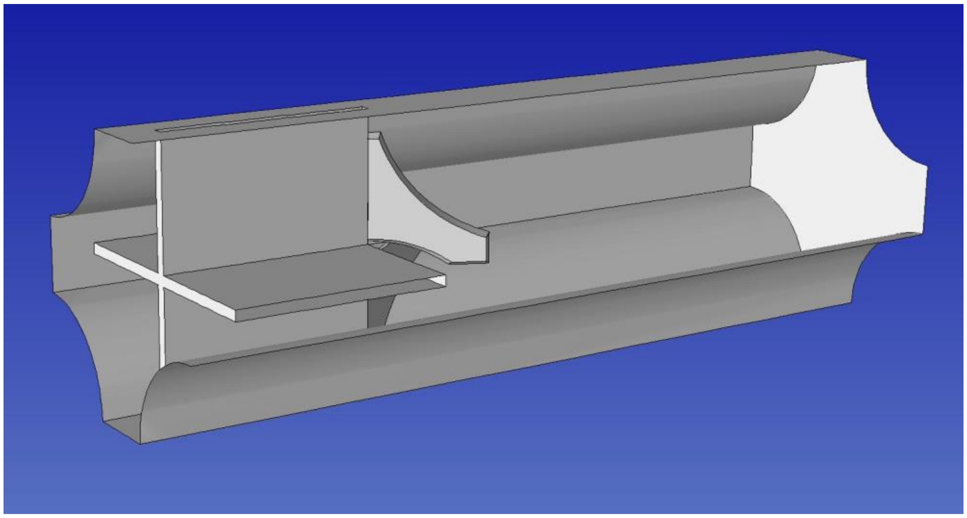


Figure 48: Single-subchannel domain with reduced size PWR-like spacer grid and two mixing vanes.

Following the same work flow as we discussed before, single-phase turbulence solution is produced before the introduction of bubbles, and the larger turbulent eddies are generated on a coarser mesh and then the solution is transferred to finer meshes. Unlike the aforementioned subchannel geometry (with no geometric barriers), the turbulence fluctuations are generated effectively by the spacer grid and mixing vanes in the new subchannel geometry (Figure 49). However, a very large mesh (2.13 B cells) is involved if one wants to fully resolve liquid turbulence and small bubble interface behavior. For example, according to our experience, one will need a resolution of at least 20 elements across bubble diameter in order to properly resolve the bubble interface behavior. A smaller case is carried out for the purpose of demonstration, while the planned case will be investigated in the near future. The important simulation parameters are listed and compared in Table 7.

Table 7: Simulation parameters of contingency and planned demonstration cases.

Parameters	Case 1	Case 2
Mesh size	265.64 M	2.13 B
Bubble resolution (# of elements across diameter)	16	20
Bubble diameter (mm)	1.0415	0.6509
Number of bubbles	91	262
Void fraction	1.46%	1%
Mean flow rate (m/s)	0.75	0.75

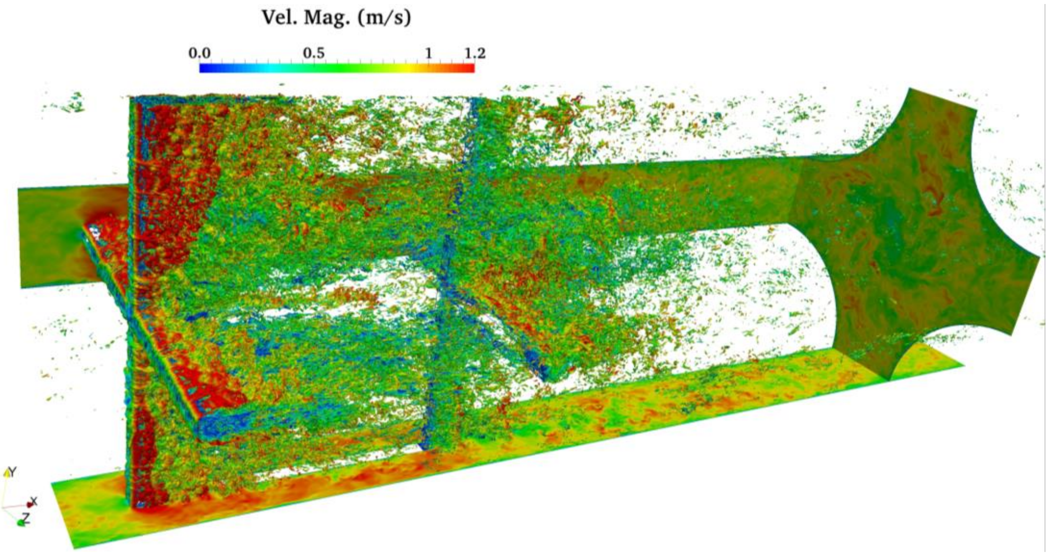


Figure 49: The single-phase turbulence vortices generated by spacer grid and mixing vanes.

In the under-resolved simulation case, the bubble resolution is about 16 elements across the diameter, and the actual mesh resolution is about one half of the intended resolution (based on linear element size) where liquid turbulence can be fully resolved (this way this represents an implicit LES simulations in the liquid turbulence vs. DNS in the planned case). As shown in Figure 50, ninety-one bubbles are initialized for the simulations of two-phase turbulent flows, and the bubbles are all placed in the upper region in order to investigate the bubble behavior when they move through the spacer grid and mixing vanes region.

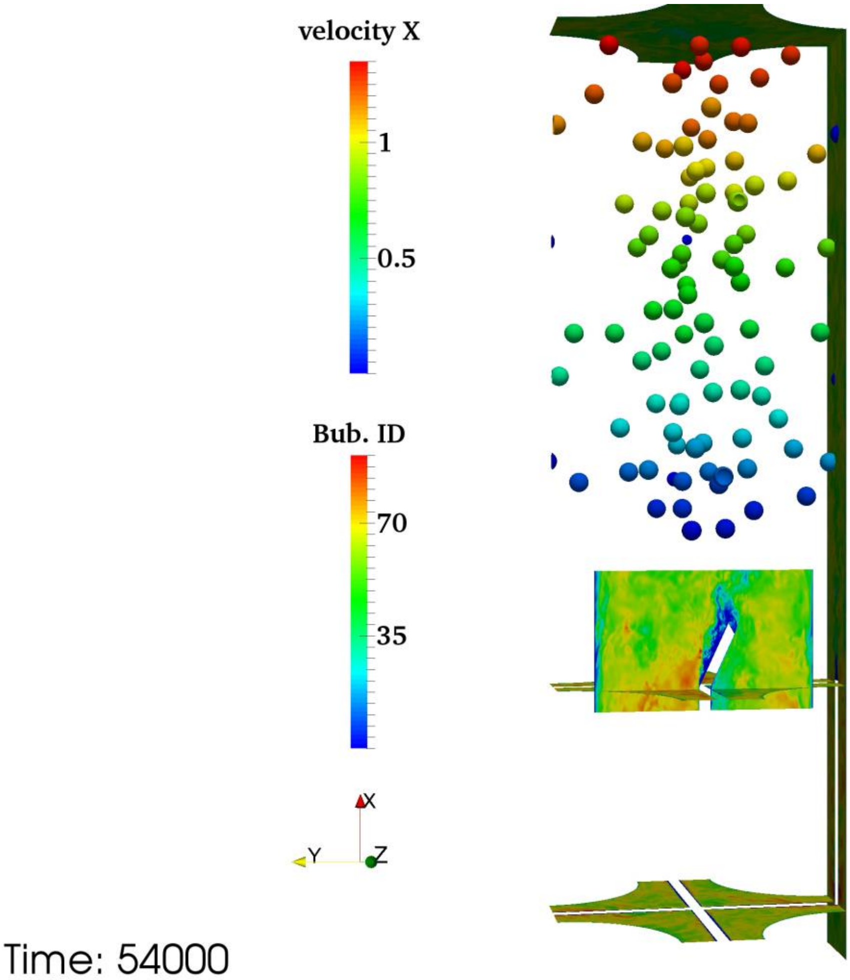


Figure 50: The initial profile of two-phase flow simulation in the subchannel geometry with spacer grid and mixing vanes (one mixing vane is visible on the slice).

5.2 Results and discussion

Both single- and two-phase subchannel simulations were performed on IBM BG/Q "Mira" at the Leadership Computing Facility (ALCF) located at the Argonne National Laboratory. The simulation results were visualized using the open-source software, ParaView. The scaling performance of the parallel DNS flow solver, PHASTA, was investigated for subchannel cases before the production runs. A set of standard parameters were adopted in all tests and the only difference is the number of mesh partitions.

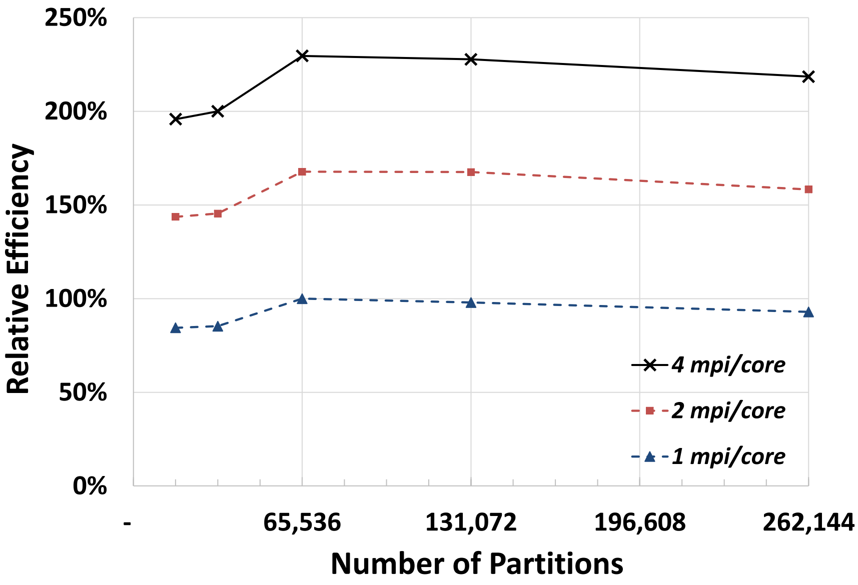


Figure 51: Scaling results of PHASTA on Mira BG/Q with a 1.11 B element mesh (subchannel geometry) normalized using 1 mpi/core run at 64K parts.

The results obtained confirmed our previous experience [123] that PHASTA scales very well for massively parallel computations of interest. The simulation efficiency is measured in core-hours consumed to conduct one full time step for each 1 billion elements. Each node on

“Mira” has 16 cores while each core is able to perform up to 4 Message Passing Interface (MPI) processes, and we observed that 4-mpi per core runs lead to the most efficient simulations, which means 4-mpi per core runs consume the least core-hours with the same mesh partitions. The computational efficiency in cases with smaller partition counts is not as good because the number of elements each core has to compute exceeds the limit that a core can efficiently handle due to memory cache limits. As expected, for partitions numbers higher than 65,536, the efficiency is slightly reduced due to increasing burden from inter-processor communications. Note that for 262,144 partitions we only have 4,238 finite elements per core and still demonstrate 92.9% efficiency compared to the 65,536 partitions.

5.2.1 Basic static probe analysis

Law of the wall profiles shown in Figure 52 and Figure 53 with dashed line results in the coefficients of $B = 6.7$ and $\kappa = 0.43$ observed in the single-phase RE01 simulations and $B = 6.7$ and $\kappa = 0.42$ for single-phase RE02 cases:

$$U^+ = \frac{1}{\kappa}\log y^+ + B \qquad (39)$$

These are expected constants for the turbulent law of the wall. We have previously observed the values of $B = 5.5$ and $\kappa = 0.4$ for a rectangular channel which were validated against available data and analytical correlations [130]. Note that in the literature, there is some variation in the values ascribed to the log-law constants, but they are generally within 5% of $B = 5.2$ and $\kappa = 0.41$ [145]. Fluctuations in law of the wall measured above are observed for large y$^+$ (200~400 for RE01 and 900~1300 range for RE02) which does not follow classic flat channel behavior. This behavior is related to the geometry of the subchannel: turbulent flow behavior at the center of subchannel is affected by all four rod walls and thus different from the law of the wall in the boundary layer/rectangular geometry.

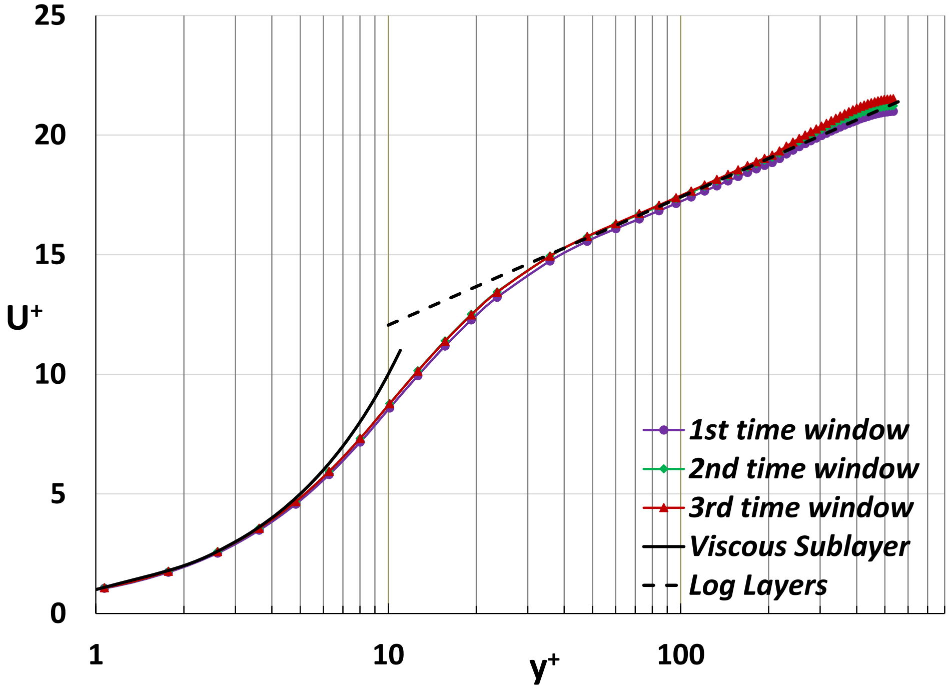


Figure 52: Law of the wall profile for single phase RE01 simulations.

The turbulent kinetic energy profile and dimensionless velocity profile are also captured by analyzing the DNS data statistically (Figure 55 and Figure 56). Interestingly, there is a flattened region on turbulent kinetic energy's decaying tail for both RE01 and RE02, and distance to the subchannel rod for this inflection is about 1.85 mm, which is very close to the half minimum distance between fuel rods (1.71 mm in our cases). As we can see in Figure 7, the probes at the same distance to fuel rod wall can experience different turbulent flow near the center of the subchannel compared to the boundaries. The statistical analysis tools we use are averaging the data from the probes located at a constant distance from the walls to produce each of the point in Figure 55 and Figure 56. At the larger distance from the wall, beyond the minimum half-distance between the fuel rods, the averaging occurs over smaller azimuthal region around each fuel rod. This causes the described behavior at the y^+ = 250-300 range shown in Figure 56 for cases RE01 and y^+ = 550-600 in case RE02.

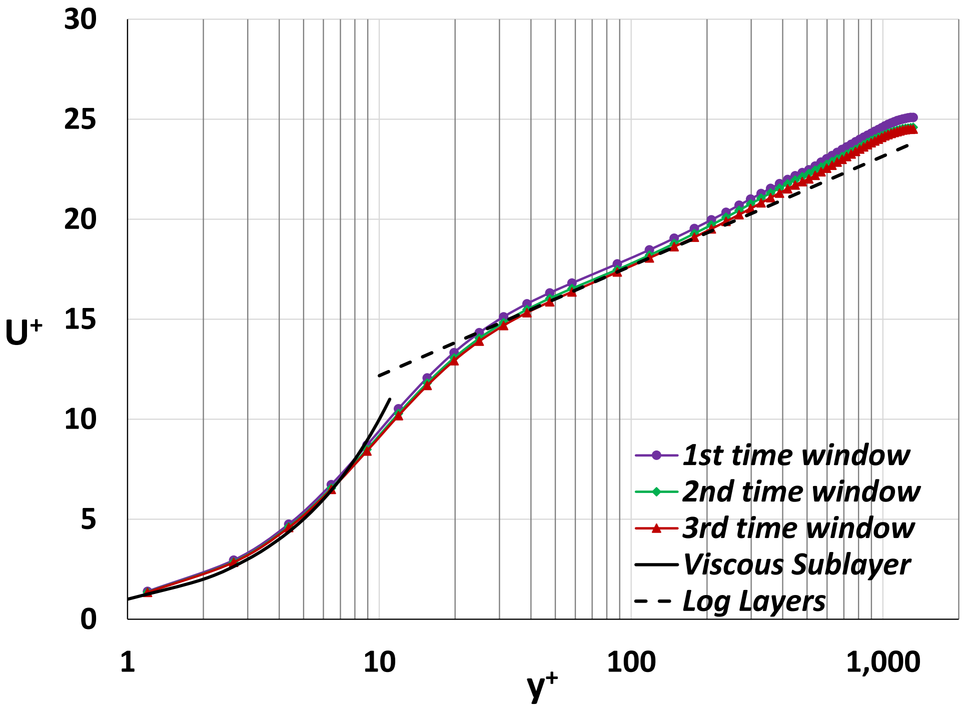


Figure 53: Law of the wall profile for single phase RE02 simulations.

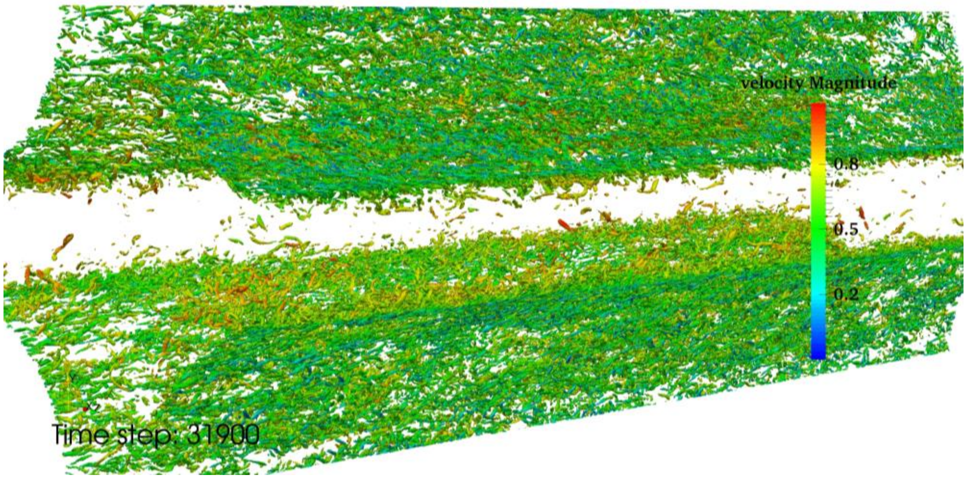


Figure 54: Turbulent vortices generated in single-phase RE02 case due to wall stress (Q-criterion value is 8×10^5).

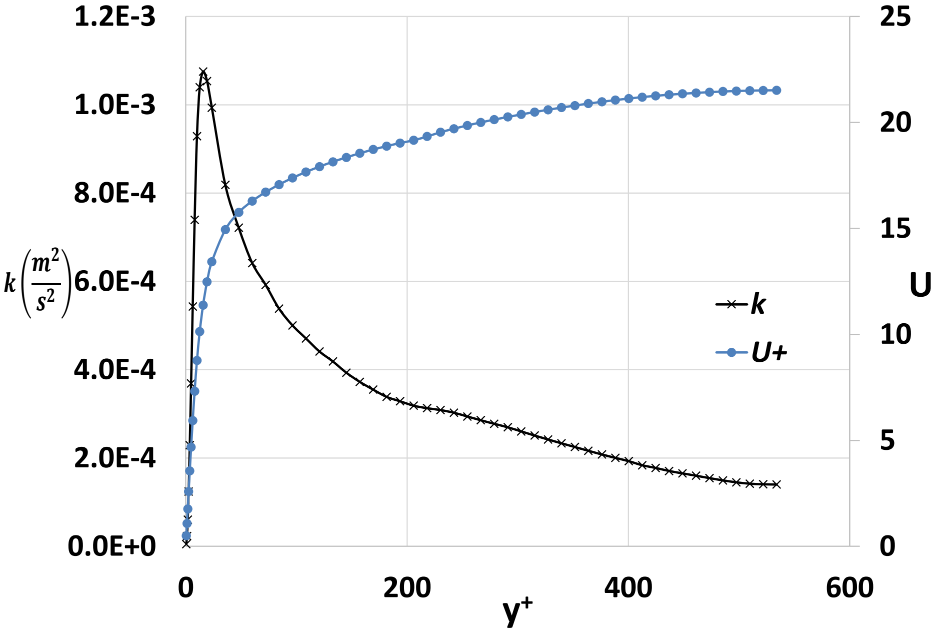


Figure 55: Turbulent kinetic energy and dimensionless velocity for single-phase RE01 simulations.

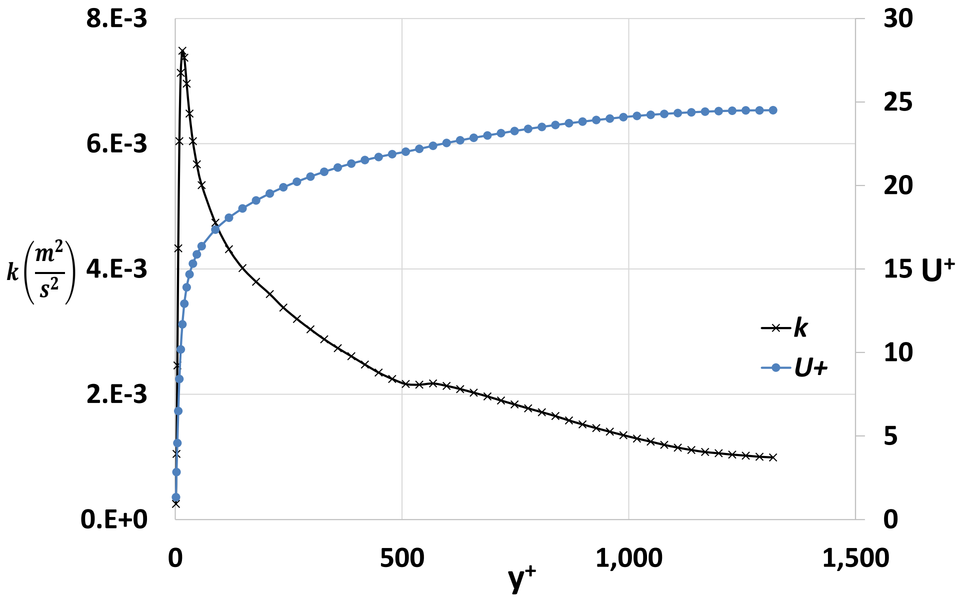


Figure 56: Turbulent kinetic energy and dimensionless velocity for single-phase RE02 simulations.

Figure 54 shows turbulent vortices generated at the walls in RE02 case, where Q-criterion [147] contour is used to visualize vortex structure and colored by fluid velocity magnitude. High Q-criterion value corresponds to more energetic turbulence. With a Q-criterion value of 8×10^5, most of the visible turbulence structures are attached to fuel rod surfaces. This is expected and consistent with the wall-peaked turbulent kinetic energy profiles extracted from DNS dataset.

Once statistically convergent flow is obtained for the single phase subchannel the bubbles are introduced in the domain through the level set method. Generally, two-phase simulations impose stricter requirements on the flow solver, such as smaller CFL number around the bubbles to properly resolve bubble deformation and advection and larger number of iterations at each time step. In addition, more simulation time is needed to accurately compute the bubble void fraction distribution for low void fraction flows due to much smaller data available for the gas phase compared to the liquid phase. The initial condition for the bubbles was specified as the distance field scalar. Seventeen bubbles were initialized in case RE01 and 262 bubbles in case RE02 to represent 1% gas volume fraction two-phase flow (as shown in Figure 46 and Figure 47). We intend to obtain statistically significant data in both cases to analyze the void fraction distribution, as well as gas and liquid mean velocity profiles.

The bubble coalescence is a natural phenomenon in bubbly flow due to random collisions among bubbles. However, the standard formulation of level set interface tracking method has difficulties in accurately representing the bubble coalescence process. Physically, when two bubbles approach, a thin liquid film develops between the bubbles. If this film has sufficient time to drain, then the bubbles will coalesce. Otherwise, the bubbles will bounce off one another. However, the standard level set method will cause coalescence of any bubbles that approach close to one another. Also, since the level set method uses a smoothed Heaviside function to transition between phase properties, this causes the coalescence process to begin sooner than experimentally observed since the thin liquid film represented by the method has somewhat mixed gas/liquid properties. In order to simulate the coalescence process more accurately, the coalescence control algorithm was developed to prevent or slow the coalescence process. This algorithm locally changes the surface tension on a portion of the bubble surface when it detects that two bubbles approach each other. This local change in surface tension creates a net force that repels the bubbles (Talley, 2014). The coalescence control algorithm has been recently upgraded by being incorporated with advanced bubble tracking capability (Section 4.1). In presented bubbly flow simulations (both 17 bubble and 262 bubble cases), this improved coalescence control is employed to better represent the physical behavior of turbulent bubbly flow in PWR subchannel geometry.

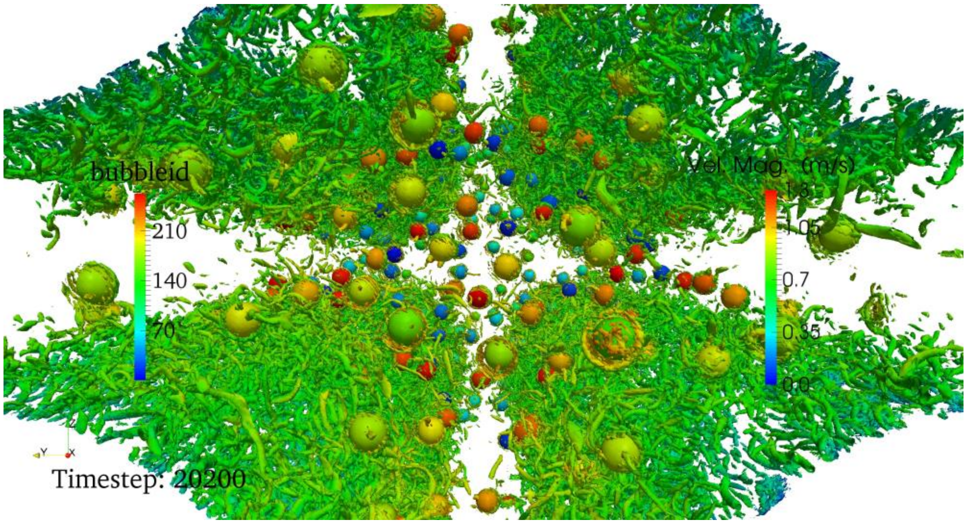


Figure 57: Turbulent vortices generated in two-phase RE02 case (Q-criterion values are ±8×10^5).

The turbulence structures in two-phase RE02 simulations are shown in Figure 57 with bubbles colored their IDs. Both wall-induced and bubble-induced turbulent vortices can be observed. As illustrated in Figure 58, Figure 59 and Figure 60, we have statistically processed the recorded data from the two-phase RE01 and RE02 simulations. The turbulent bubbly flows have achieved 7.8 flow-throughs in RE01 case and 1.23 flow-throughs in RE02 cases. Law of the wall analysis shown in Figure 58 with dashed line results in the coefficients of B = 7.8 and κ = 0.5 observed in the two-phase RE01 simulations and B = 4.8 and κ = 0.35 for two-phase RE02 cases. Table 8 summaries the coefficients involved in our law of the wall study.

Compared with the results from RE02 simulations, RE01 two-phase cases exhibited more fluctuations and the law of the wall profile has been flattened. One should keep in mind that RE02 two-phase cases can resolve many more bubbles, so larger number of bubble instances will be detected by probes during one flow through. Thus, better statistics were obtained for 262 bubble simulations (RE02) vs. 17 bubble simulation (RE01).

Table 8: Summary of model constants in law of the wall study

Case	RE01(Subchannel)	RE02 (Subchannel)	Flat channel
Reynolds number (Re_h)	29,079	80,774	29,530
κ (single-phase)	0.43	0.42	0.40
B (single-phase)	6.7	6.7	5.5
κ (two-phase)	0.5	0.35	N/A
B (two-phase)	7.8	4.8	N/A

The TKE distributions of both single- and two-phase simulations are plotted and compared in Figure 59. As expected, the bubbles introduce additional energy to turbulence field. The magnitude of TKE in RE02 is notably higher than that in RE01 due to higher Reynolds number in the modeled flow. The distributions of gas and liquid velocity as well as the void fraction from the two-phase RE01 and RE02 simulations are shown in Figure 60. In the region with non-zero void fraction the gas velocity is observed to be larger than liquid velocity because the bubbles are accelerated by the buoyancy force in the subchannel. In both 17 bubble case and 262 bubble case, the void fraction peak shifts closer to the subchannel walls, which is expected because smaller bubbles will migrate in transverse directions due to the effect of the lift force. This bubble aggregation behavior towards fuel rod surface is more prominent in case RE02 in which a larger number of smaller bubbles present in the turbulent flow and better statistical data is available.

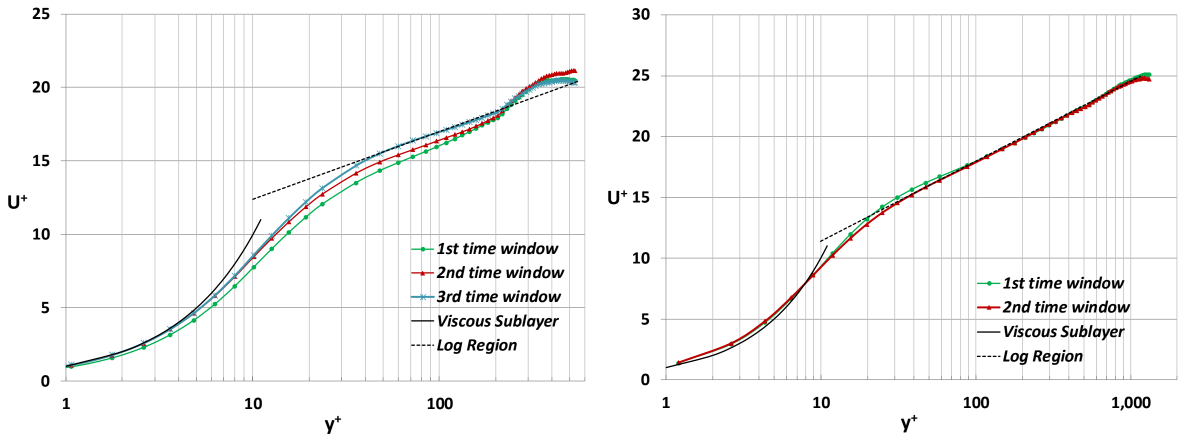


Figure 58: Law of the wall profile for two-phase simulations (left: RE01, right: RE02). The viscous sublayer curve shown is described by $\boldsymbol{u^+ = y^+}$ and the log law is described by Eq. (39).

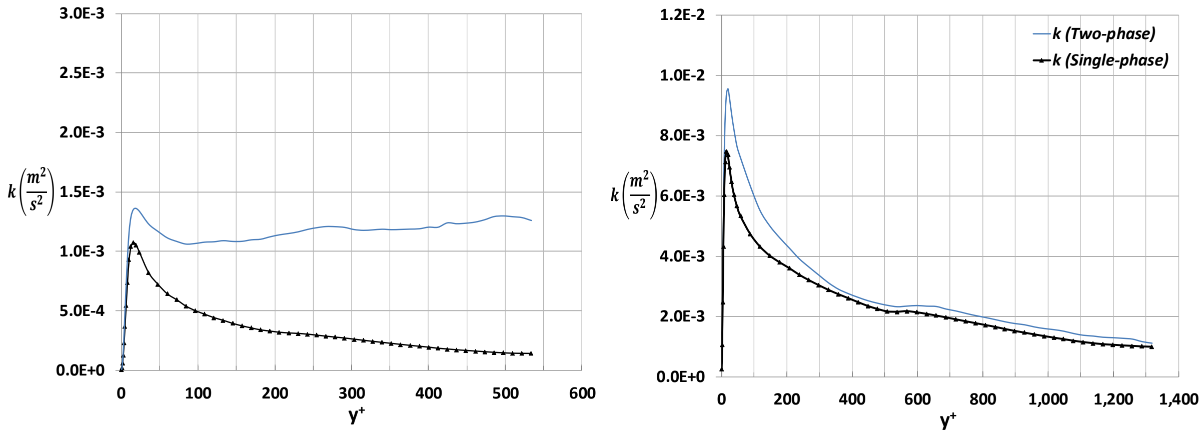


Figure 59: Turbulent kinetic energies for single- and two-phase cases (left: RE01, right: RE02).

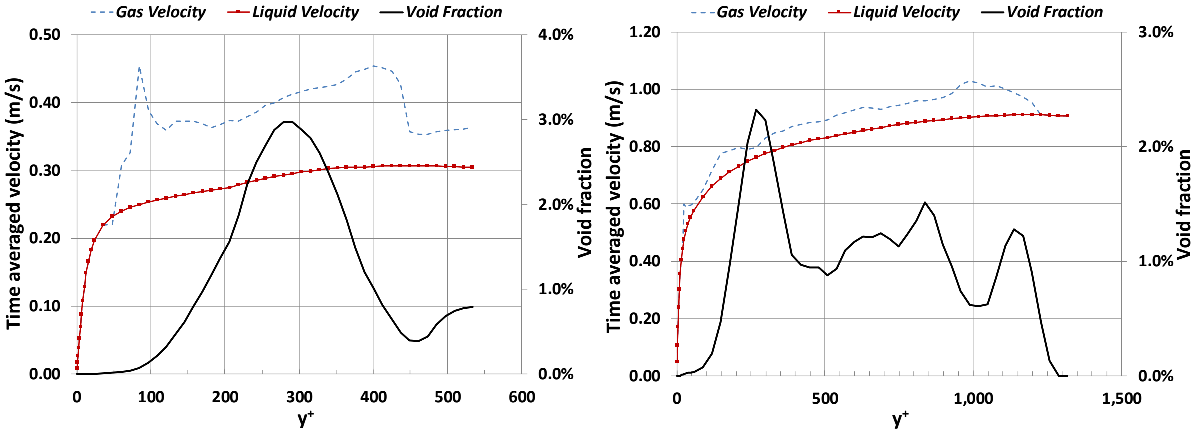


Figure 60: Void fraction and gas-liquid velocity profile from two-phase simulations (left: RE01, right: RE02).

In the two-phase flow simulations performed herein, bubbles are introduced when single-phase turbulence has reached the steady state conditions. The further development of turbulence field is primarily attributed to the interaction between bubbles and liquid turbulence. In experiments the channel hydraulic diameter is commonly used to quantify the length scale needed for fully developed single-phase turbulence. As an analogy, we propose to use average bubble diameter to quantify the length scale needed for a full fulfillment of bubbles' impact on steady-state single-phase turbulence field. From this perspective, smaller bubbles will result in more advection distance in terms of bubble diameter units compared to larger bubbles over the same period of time. As for our simulations in single PWR subchannel geometry, a single flow-through corresponds to about 62 bubble diameter units for RE02 262-bubble case. In RE02 case the bubbles have moved 76 bubble diameter length. As will be discussed in Section 5.2.2, the analysis of bubble tracking data has revealed that bubble relative velocity tends to converge after about 40 bubble diameter units. Therefore, seventy-six units would be probably enough for two-phase flow turbulence to further develop considering the prior single-phase steady-state turbulence. When two-phase flows achieve statistically steady state conditions, the drag coefficient can be estimated based on the bubbly buoyancy force and bubble terminal velocity. Assuming that the steady state conditions are reached, the bubble buoyancy force will balance out the drag force. By averaging the bubble relative velocities, the following equation can be used to estimate the drag coefficient (C_d) in RE02 case:

$$F_d = \frac{1}{2}\rho_l v_r^2 C_d A_{bubble} = (\rho_l - \rho_g) g V_{bubbble} \qquad (40)$$

where ρ_l and ρ_g are liquid and gas densities; v_r is averaged bubble relative velocity in streamwise direction; A_{bubble} and $V_{bubbble}$ are averaged cross-sectional area and volume of a bubble. We chose not to estimate the drag coefficient in the 17 bubble case due to unrealistically large bubbles. The resultant drag coefficient is 0.071 in RE02 two-phase case. According to Tomiyama drag correlation for **single** bubbles in **standing** liquid ([6]), the suggested drag coefficient is 0.1617 for case RE02. As mentioned earlier, our approach was used to re-produce the drag coefficient correlation by Tomiyama with less than 1% error for single spherical bubbles in the past ([18]). Using the same numerical tools, we have shown that the drag coefficient is about half of the single bubble in standing fluid correlation in realistic flows in the reactor subchannel. Multiple bubble interaction results in bubble clustering which allows higher relative velocities and effectively reducing the drag force experienced by any single bubble. Liquid turbulence also has an effect on bubble motion compared to the standing fluid assumption.

5.2.2 Advanced analysis with bubble tracking

Equipped with newly developed bubble tracking capability, PHASTA is now able to track all individual bubbles in turbulent flow. Detailed bubble information has been collected from bubble tracking simulations for both 17-bubble and 262-bubble cases. Taking bubble trajectories as an example, based on bubble tracking data, we can obtain the full picture of bubble relative positions with respect to the closest wall. In the RE02 two-phase case where 262 bubbles resolved, the wall distances are plotted in Figure 61 for 10 bubbles selected in the domain. The wall distance is in the unit of characteristic/average bubble radius while the simulation time is converted into ***bubble diameter time***. One unit of bubble diameter time is the time interval during which bubble moves one characteristic diameter length (discussed previously in Section 3.4). In this 10-bubble sample, despite the relatively uniform distribution at the beginning, most bubbles show an unambiguous tendency to migrate towards walls (i.e. fuel rod surface).

The bubbly flow simulation has achieved 7.8 flow-through for RE01 17-bubble case (54M elements) and 1.23 flow-through for RE02 262-bubble case (1.11B elements). Here one flow-through time is defined as the time simulated flow takes to move a distance of subchannel length (i.e. 3.14 hydraulic diameters). As for the migration of the entire simulated bubble population, the bubble distribution profiles are shown in Figure 62 and Figure 63 at various time steps for both cases. It is interesting to observe that the bubbles tend to aggregate near the fuel rod surfaces, especially in RE02 case where 262 smaller bubbles are simulated. The wall aggregation behavior is expected because the lift force of small/spherical bubbles is directed towards the closest wall in an upflow condition [98]. In addition, the bubble wall distance evolution shown in Figure 61 demonstrates a good consistency with bubble distributions in Figure 63.

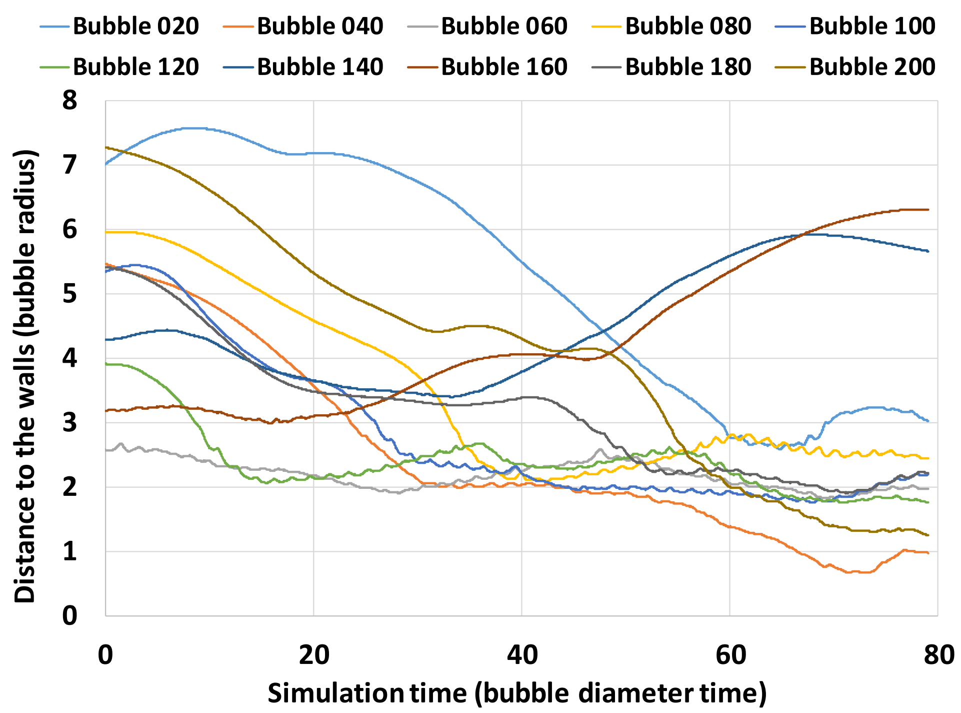


Figure 61: Evolution of bubble distance to the walls in case RE02.

In order to process the bubble information recorded with bubble tracking capability, a post-processing code was developed. The results obtained so far include (1) the evolution of various bubble parameters, (2) the bubble interfacial forces within different bubble groups, and (3) the dynamic void fraction distribution. As illustrated in Figure 64, the average bubble position gets stabilized after the initial developing regime for case RE01. This trend is due to the bubble aggregation phenomena usually observed in pipe or channel flow, especially when the average bubble size is small. Figure 65 exhibits the average bubble velocity and relative velocity between bubble and local liquid. The convergence to approximately steady state is further confirmed by the velocity evolution. The major role of case RE01 is only to serve as an affordable demonstration case where liquid velocity is slow and bubbles are too large considering the intended research goals. Therefore, further bubble tracking analysis will be focused on RE02 case where 262 bubbles are simulated in the subchannel geometry.

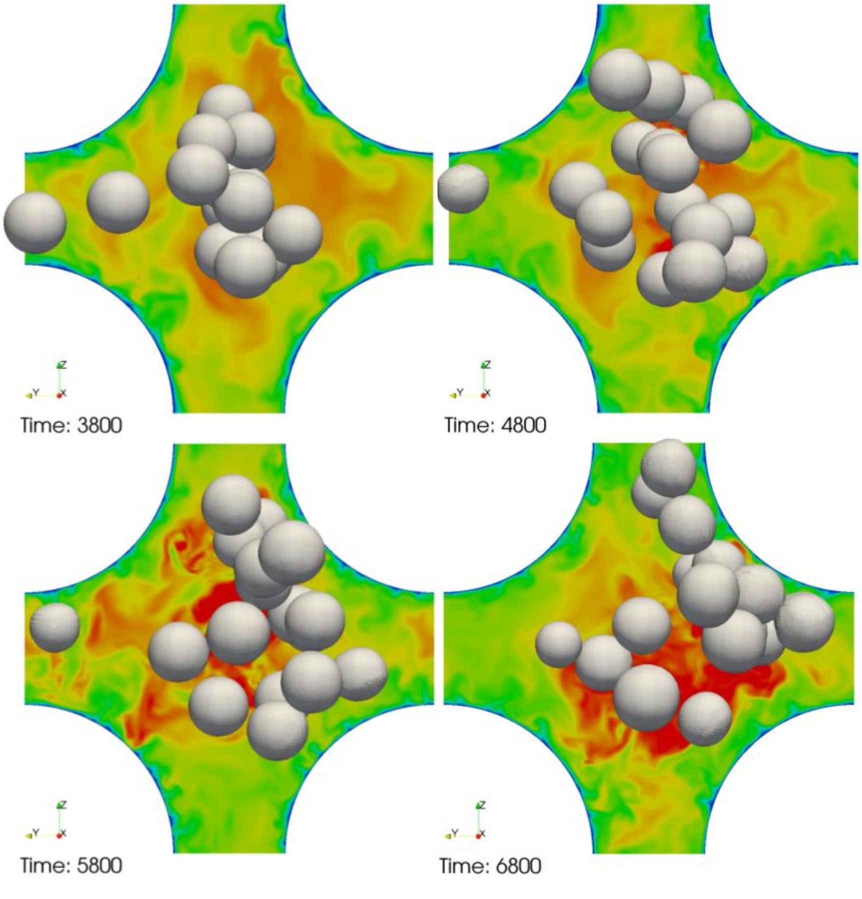


Figure 62: The 17-bubble distribution from the inflow direction at about 60, 120, 180, 240 bubble diameter times.

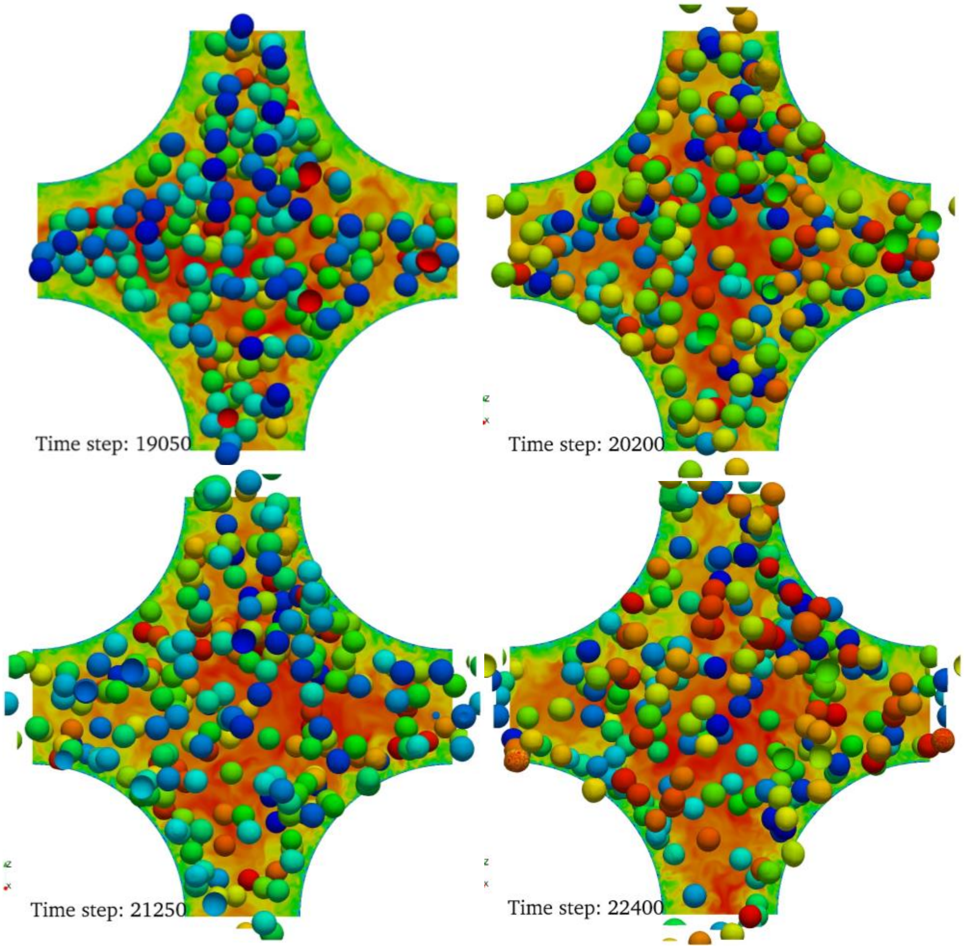


Figure 63: The 262-bubble distribution from the inflow direction at about 20, 40, 60, 80 bubble diameter times.

Because the computational cost of DNS grows exponentially with Reynolds number, case RE02 is much more computationally expensive compared to case RE01. Although more statistical data is needed, several interesting results have been obtained from the analysis of limited data already collected. Since bubble tracking capability can allow the recording of detailed bubble information, one can classify bubbles into several groups based on specific criterions and study their behavior separately.

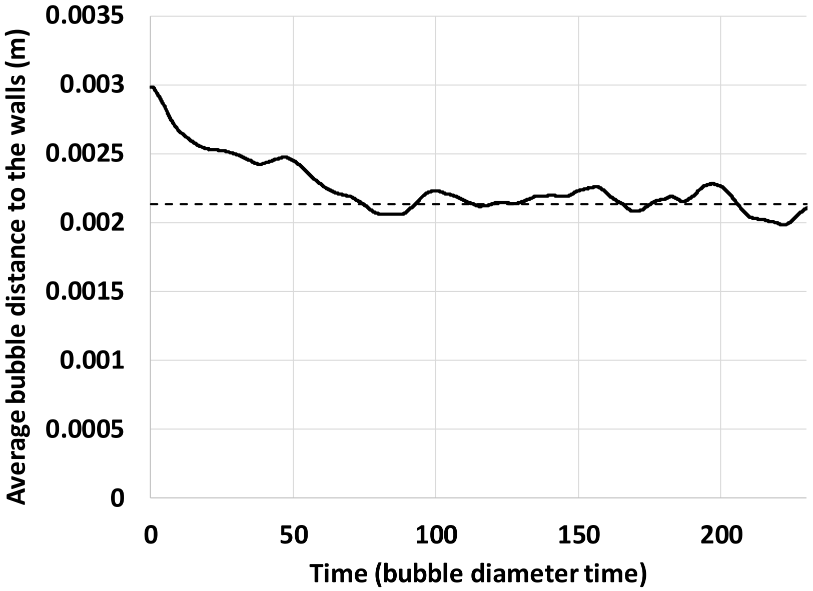


Figure 64: The average bubble distance to the wall in case RE01.

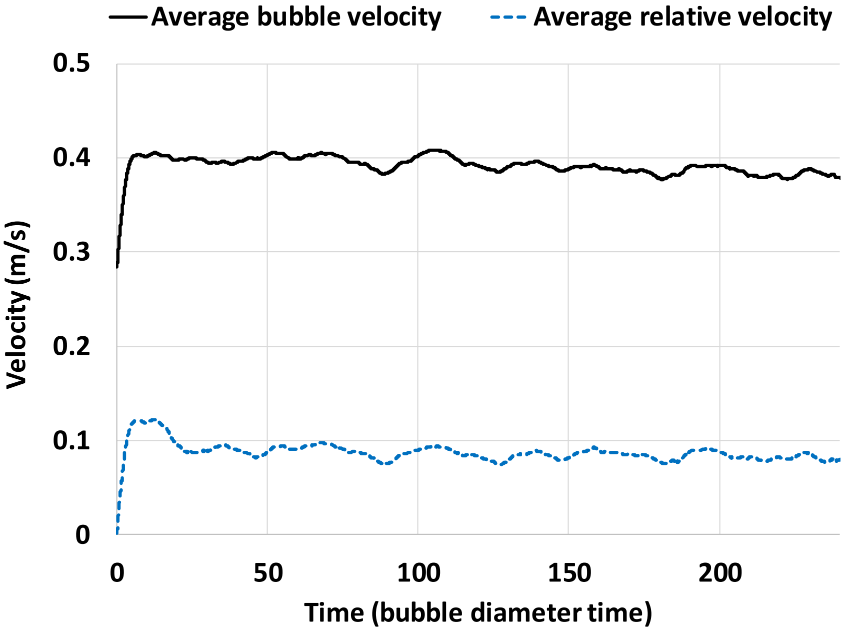


Figure 65: The average bubble and relative velocity in case RE01.

One classification criterion has been explored is bubble position with respect to fuel rod surfaces. In current subchannel geometry the domain centerline is the furthest position from the fuel rod surface (d_{wall} = 4.34 mm measured as distance to the wall). Three bubble groups are employed in the analysis (as depicted in Figure 66), including near (the wall) group ($0 < d_{wall} \leq 1.45\ mm$), middle group ($1.45 < d_{wall} \leq 2.90\ mm$) and far (from wall) group ($2.90 < d_{wall} \leq 4.13\ mm$). The evolution of bubble group population from RE02 simulations is shown in Figure 67. As consistent with the observation in conventional analysis, bubbles are driven towards walls due to lateral lift force, in other words, near group will see the growth in bubble population while other groups will have less bubbles. This is also consistent with bubble distribution evolution one can observe in Figure 63.

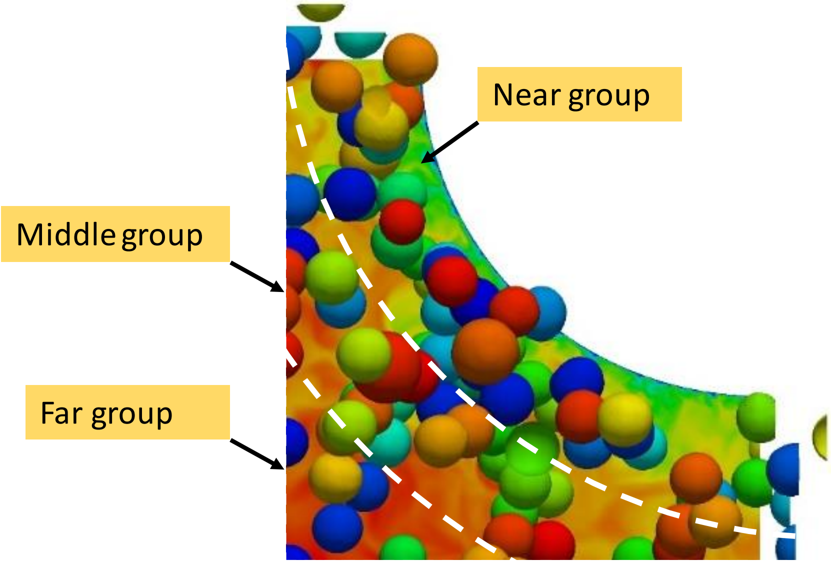


Figure 66: Bubble groups and associated regions around the fuel rod.

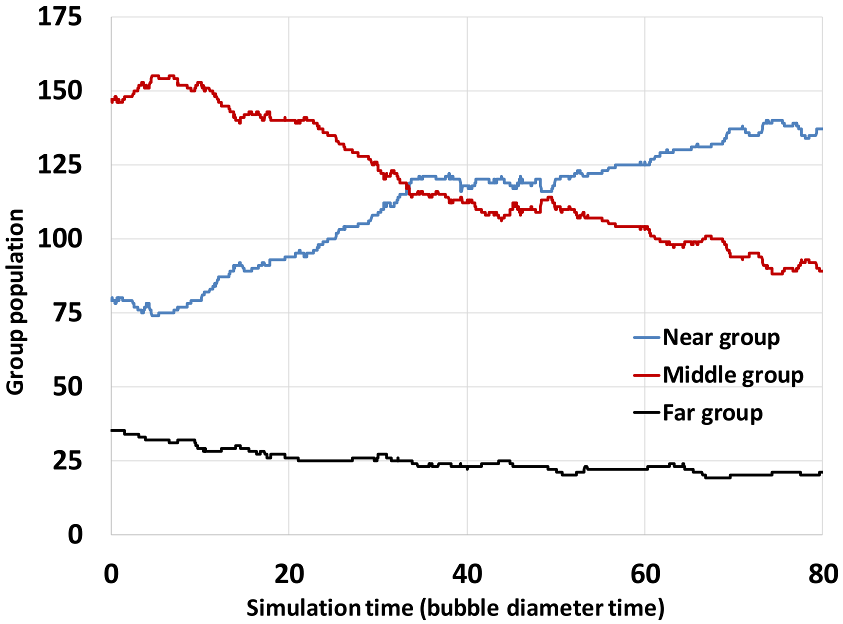


Figure 67: The evolution of bubble group population in case RE02.

By extending the regions of interest to include local liquid around the bubble, one can also collect local liquid information from bubble tracking simulations. The relative velocity between bubble and local liquid can be estimated by subtracting average local liquid velocity from average bubble velocity. Interestingly, it is noticed that bubbles in different groups experience quite different relative velocities. As shown in Figure 68, the bubbles in near group see the largest relative velocity while the far group experience the smallest. If one assumes bubble drag force is equal to buoyancy force at approximated steady state (when virtual mass force is negligible), drag coefficient will vary significantly due to different relative velocities. Although the bubbly flow is still in a developing regime (Figure 67), the bubble relative velocity has entered the stable state. Thus the assumption that buoyancy equals drag force should hold after 40 bubble diameter times (Figure 68). The difference in relative velocity suggests the necessity of the introduction of variable drag coefficient or additional dependencies into the classic interfacial model closure laws.

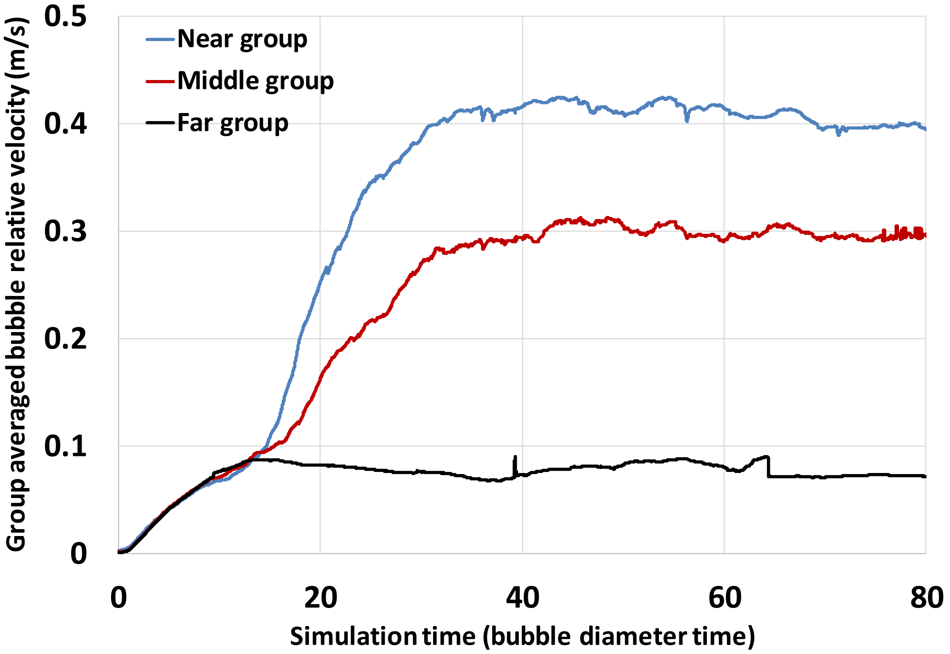


Figure 68: The average relative velocity from different bubble groups in stream-wise direction.

Based on the assumption that the buoyancy force is approximately equal to the drag force, the drag coefficient can be estimated based on the force balance as shown in Eqs (41) and (42). The estimated drag coefficients are shown in Figure 69 for all three bubble groups. The drag coefficient in near group has a range of 0.05 to 0.11; middle group has a range of 0.85 to 1.05; the far group has a range of 1.50 to 2.20. Overall drag coefficient presents an interesting tendency to decrease as bubbles are closer to the walls. Recall the definition of bubble

Reynolds number ($Re_b = U_r D/\nu$), and the bubble Reynolds number for all three groups are 2,211, 1,618 and 422 respectively. The general trend is consistent with experimental observation that bubble drag coefficient reduces as bubble Reynolds number grows (Figure 28). Tomiyama single-bubble drag model (the validated Re_b ranges from 10^{-3} to 10^{5}) [6] predicts a drag coefficient of 0.156 for all groups because the effect of Eo dominates that of Re_b given the investigated simulation conditions. The results of near group bubbles are close to the predicted coefficient while middle and far groups suggest larger drag coefficients. The difference may be related to the effect of liquid turbulence and bubble interaction. Although considerable effort has been devoted into two-phase bubbly flow experiments from both nuclear and mechanical engineering communities [148], the data available for direct comparison of presented results is still sparse. Either the geometries or material properties are very different. Despite this, the obtained results here are still very informative, and could be useful to guide the design of feature validation experiments.

Considering the smallest cross-sectional area (shown in Figure 66), the far group expects the least number of bubbles/samples, which explains the large fluctuations in the resultant drag coefficient. However, large fluctuation is also observed in near (wall) group, which is counter-intuitive at the first glimpse. Most bubbles in near group have very small drag coefficients, which makes the averaged coefficient of near group quite sensitive to large values. In the meantime, it’s still likely to observe small relative velocities for bubbles in near group, especially in turbulent bubbly flows. Based on Eq. (42), small relative velocity v_r directly lead to larger C_D, which explains the drag coefficient fluctuations in near (wall) group.

$$F_B = F_D = \frac{1}{2} C_d \rho_L A_b v_r^2 = g(\rho_L - \rho_G) V_b \quad (41)$$

$$C_D = \frac{2(\rho_L - \rho_G) g V_b}{\rho_L A_b v_r^2} \quad (42)$$

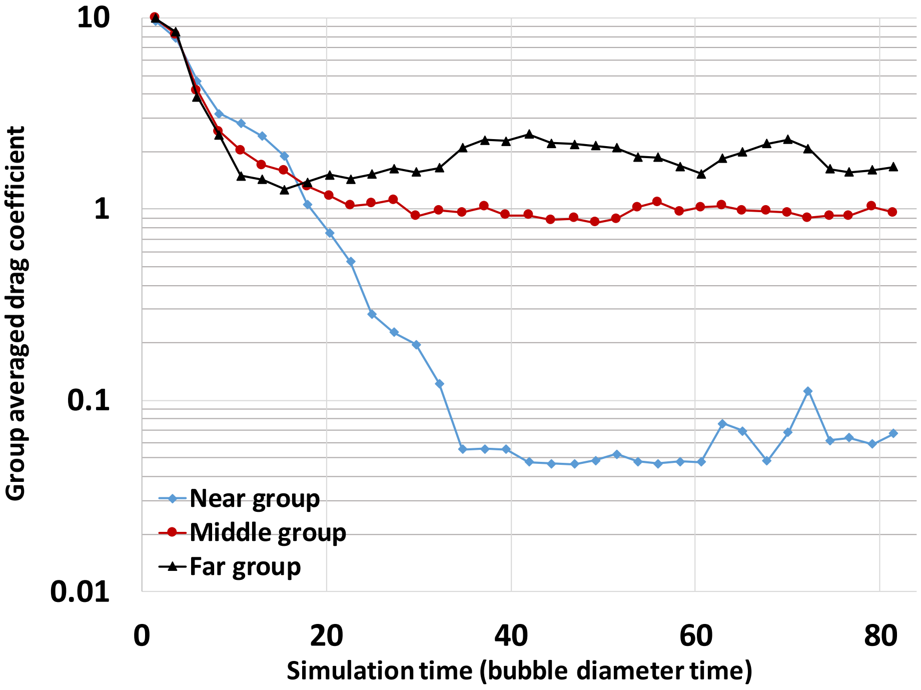


Figure 69: The group averaged drag coefficient by assuming buoyancy force is equal to drag force.

Besides the drag coefficient, the post-processing analysis also investigated the lift coefficient. As a transverse force, lift force is critical to determine radial void fraction distribution. To study the transverse lift force, it is essential to examine the transverse bubble velocity. Unlike stream-wise bubble velocity, transverse velocity carries more oscillations and more sensitive to virtual mass force and turbulence dispersion force. All these effects impose complexity on lift force analysis. A qualitative analysis is performed first to provide a survey about the direction of the bubble transverse motion. Based on bubble position and its transverse velocity, it is possible to check if a bubble is approaching or leaving walls/fuel rod surfaces. We assign a positive lift sign to bubbles approaching walls, and a negative sign to bubbles leaving the walls. Figure 70 exhibits the percentage of bubbles with positive lift sign over simulations, and bubbles shows a clear preference to migrate towards walls in near and middle groups. Larger fluctuations are noticed for bubbles in far group, and at some points during the simulation, more bubbles even tend to leave walls other than approaching walls. The far group behavior can be explained by the effect of "switching" the wall the far bubbles are associated with in the subchannel.

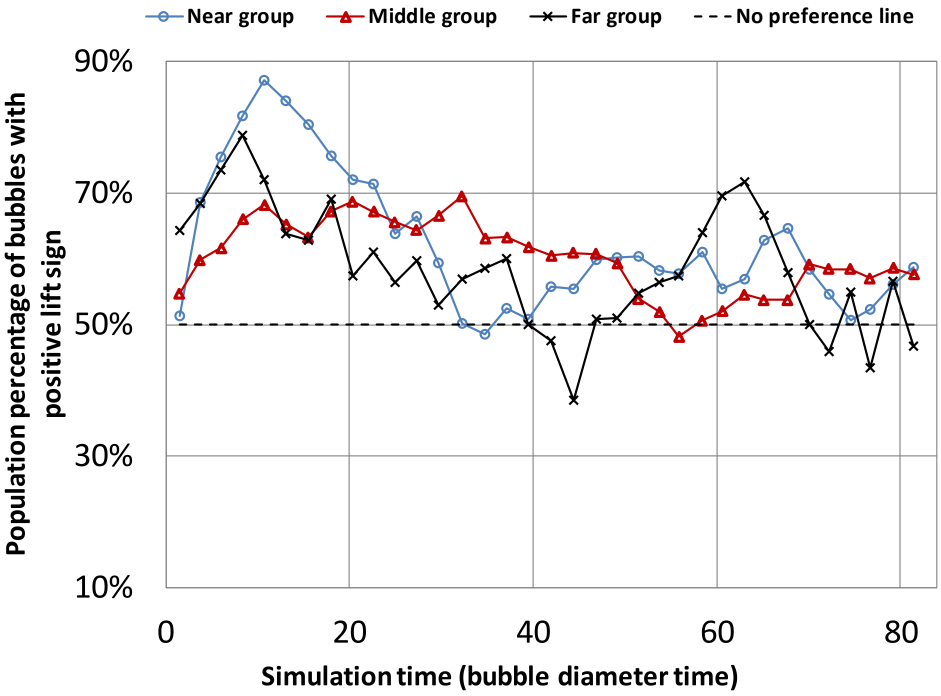


Figure 70: The percentage of bubbles with positive lift coefficient (i.e. a bubble is approaching the wall).

Several assumptions are involved in the quantitative analysis of the lift coefficient. At first bubble parameters are averaged over a short 'bubble diameter time' to reduce the random oscillation effect (due to turbulence). Those parameters include transverse bubble velocity, position, local liquid velocity and shear rate. Since the simulated bubbles are generally spherical, the virtual mass force coefficient is assumed to be 0.5 (this can be improved later to account for non-spherical bubble shape). Meanwhile, in a time interval like bubble diameter time, the effect of turbulence dispersion force is assumed to be negligible as the random turbulence fluctuations will be removed by the averaging. Bubble acceleration is calculated as the time derativative of the average bubble velocities of two adjacent bubble diameter times.

The derivation of the lift coefficient is shown in Eqs. (43) and (44). The group averaged lift coefficient is ploted in Figure 71 for all three groups. It can be noted that all groups display a positive lift coefficient in the initial developing stage, which is consistent with the observation that bubbles are 'scattered' towards fuel rod surfaces. Then after 40 bubble diameter times, bubbles in middle group result in a positive lift ($0 < C_L < 0.3$). This explains the continuous bubble migration from midde group to near group as one can see in Figure 67. Also, bubbles in near group show very small lift coefficient ($|C_L| < 0.02$) and they seem to be 'trapped' in the near wall region, corresponding to the wall aggregation phenomenon. Finally, the bubbles in far group experience an oscilated average lift coefficient which makes bubble not show a clear preference about approaching or leaving walls after initial rapid developing regime. As was already mentioned, this can be explained by constant switching of the associated wall for each bubble in the far group as it crosses the symmetry lines in the subchannel domain.

$$F_L = C_L \rho_L V_B |v_r| Sr = \left(\rho_B + \frac{1}{2}\rho_L\right) \cdot V_B \frac{d\boldsymbol{v}_{\boldsymbol{bub.trans}}}{dt} \cdot \boldsymbol{n}_{\boldsymbol{wall}} \tag{43}$$

$$C_L = \frac{\left(\rho_B + \frac{1}{2}\rho_L\right)\frac{d\boldsymbol{v}_{\boldsymbol{bub.trans}}}{dt} \cdot \boldsymbol{n}_{\boldsymbol{wall}}}{\rho_L |v_r| Sr} \tag{44}$$

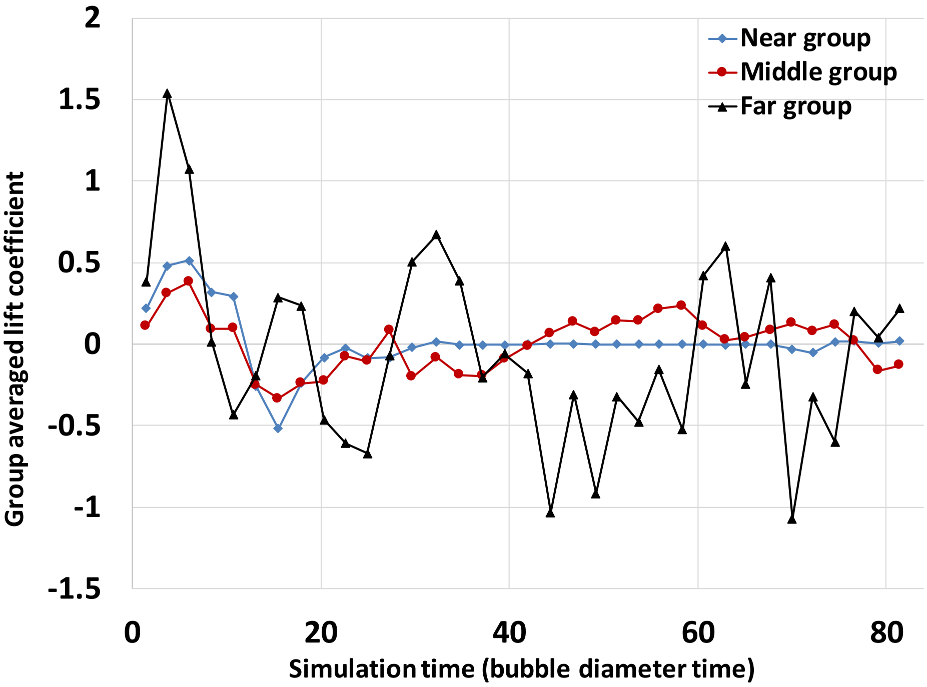


Figure 71: The group average lift coefficient.

Additional analysis includes the histogram of bubble deformation levels and dynamic void fraction distribution. As shown in Figure 72, the bubbles with different deformability levels are classified and distributed into different bins. Bin 1 to bin 10 represent the 10 equal-size subranges between 0.0 and 1.0. Bin 10 (i.e. deformability factor falls into 0.9 – 1.0) is for the bubbles nearly spherical. The lower the bin number is, the more deformable the corresponding bubbles are. At the initial time step when the bubbles are introduced in the domain (19,000), all the bubbles are spherical, thus the percentage of bin 10 is 100%. As the simulation progresses, the bubbles start to deform slightly due to various effects, such as coalescence, collisions with fuel rod surface and acceleration relative to the surrounding liquid. At time step 21,000, only 45% of all the bubbles are still in bin 10 and the rest shows different levels of deformation.

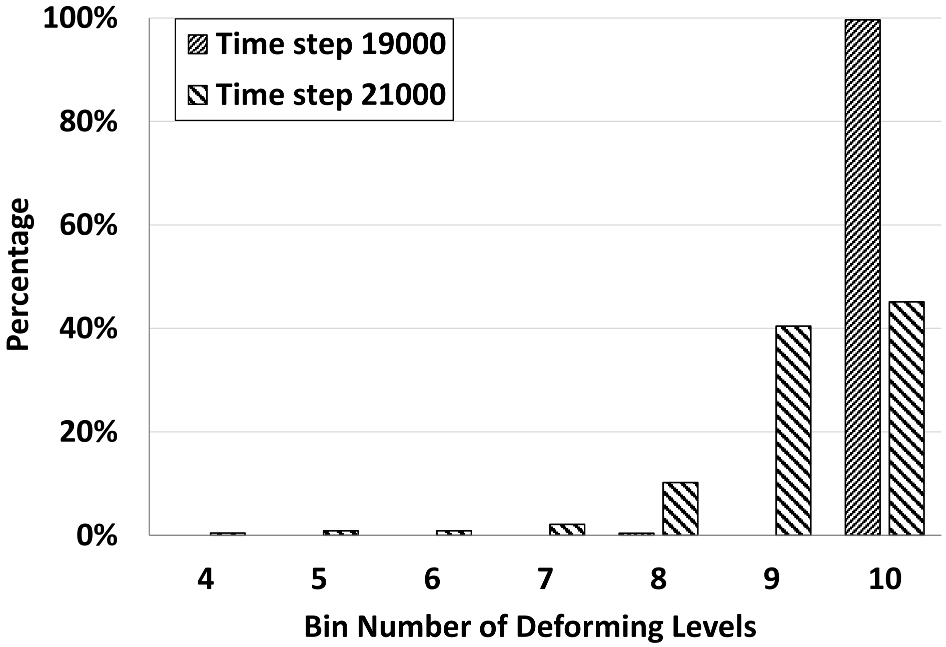


Figure 72: The deformability levels of the bubble population.

Besides the deformation analysis, the post-processing code can also utilize the recorded bubble positions and sizes to reconstruct the void fraction distribution for the turbulent bubbly flow in a PWR subchannel. We already know that as the bubbly flow moving through the subchannel, the bubbles will migrate to the near wall regions in an upflow (especially when bubbles are small and nearly spherical). Illustrated in Figure 73, the x-axis represents the distance to fuel rod surface in turbulent wall units. As a result of bubble migration, the void fraction peak in low y^+ area is growing and reaches steady-state distribution after time step 21000 (this corresponds to 3.06×10^{-2} seconds of simulation time). As a comparison, the average void fraction distribution from time step 19900 to 21500 is also calculated with the static probe recording approach [10] and shown in dashed line. The results of the presented novel bubble tracking approach show a very good agreement with the static probe recording approach.

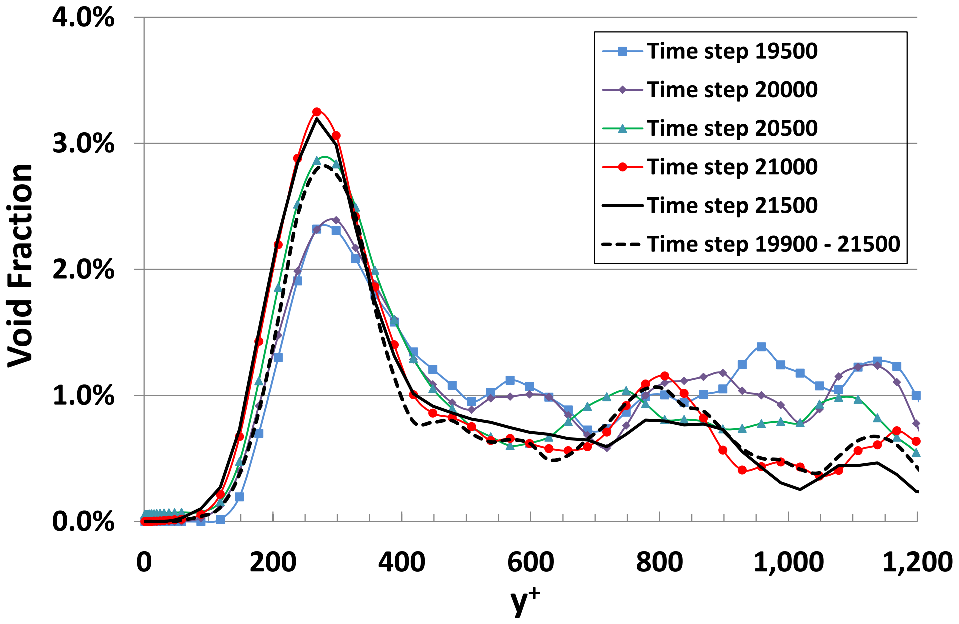


Figure 73: The void fraction distributions at different time steps from bubble tracking approach versus that from static probe recording (in dashed line).

In the demonstration case of more complicated subchannel geometry, the reduced size PWR-like spacer grid and mixing vanes are represented in the simulated domain. The bubbles will interact with these structures when passing through the spacer grid and mixing vanes region. Although both liquid turbulence and bubbles are under resolved in the contingency case, some expected phenomena, like bubble breakup and coalescence, are still observed in the simulations as shown in Figure 74. The robustness of PHASTA simulation has been successfully demonstrated even when the computation involves interface tracking, bubble tracking as well as complicated engineering geometries.

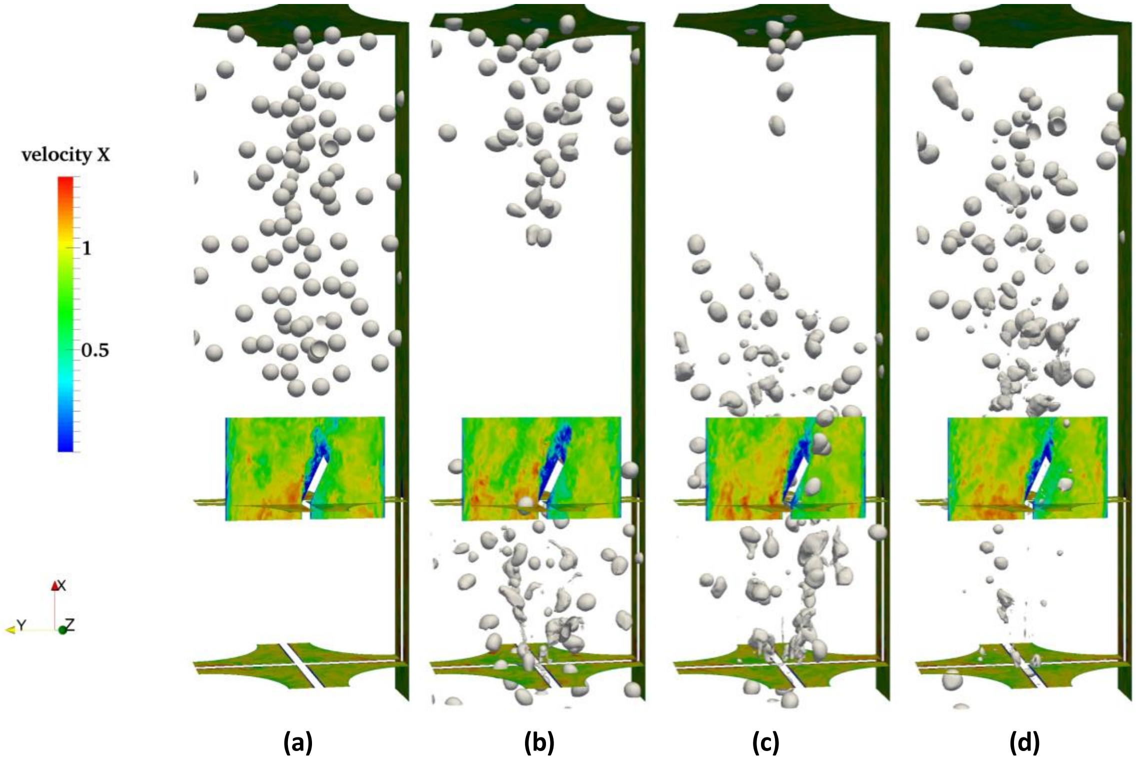


Figure 74: The simulation profiles of bubbly flow through a subchannel with spacer grid and mixing vanes (at *1.12×10^{-2} s*, *2.25×10^{-2} s*, *3.34×10^{-2} s*, *4.50×10^{-2} s*).

In previous simulations of subchannel geometry with no geometrical barriers, the bubbles show a clear tendency to approach fuel rod surfaces. This is expected because small/spherical bubbles experience a positive lift force in up flows which can drive them towards the walls. However, the similar behavior is not observed in the new case. Based on the recorded bubble tracking data, a first quick estimation is the average bubble wall distance of initial (Figure 74a) and final (Figure 74b) bubble distributions. The initial average bubble wall distance is 2.0 mm while this wall distance becomes 2.2 mm for the final bubble distribution. By selecting a sample window ranging from 2.0 cm to 4.0 cm in vertical direction, one can reconstruct the void fraction distribution with respect to the wall distance.

Figure 75 shows the window location of void fraction analysis in the channel as well as the reconstructed void fraction profile for both initial and final bubble distribution. It is interesting to observe that bubbles show a contradictory center-aggregation behavior compared to wall aggregation behavior observed in previous bubbly flow simulations. The existence of mixing vanes can lead to a momentum increase of two-phase flow in the transverse direction towards the walls (this results in the flow swirl and centrifugal effect on the flow mixture). Even though lift effect tends to drive small bubbles towards the walls in under-disturbed flows, the bubbles will be forcedly moved from boundary region to domain center because gas bubbles are much lighter compared to the liquid. The similar bubble center-aggregation behavior has also been observed in experiments [149] and multiphase Computational Fluid Dynamics (CFD) simulations [150].

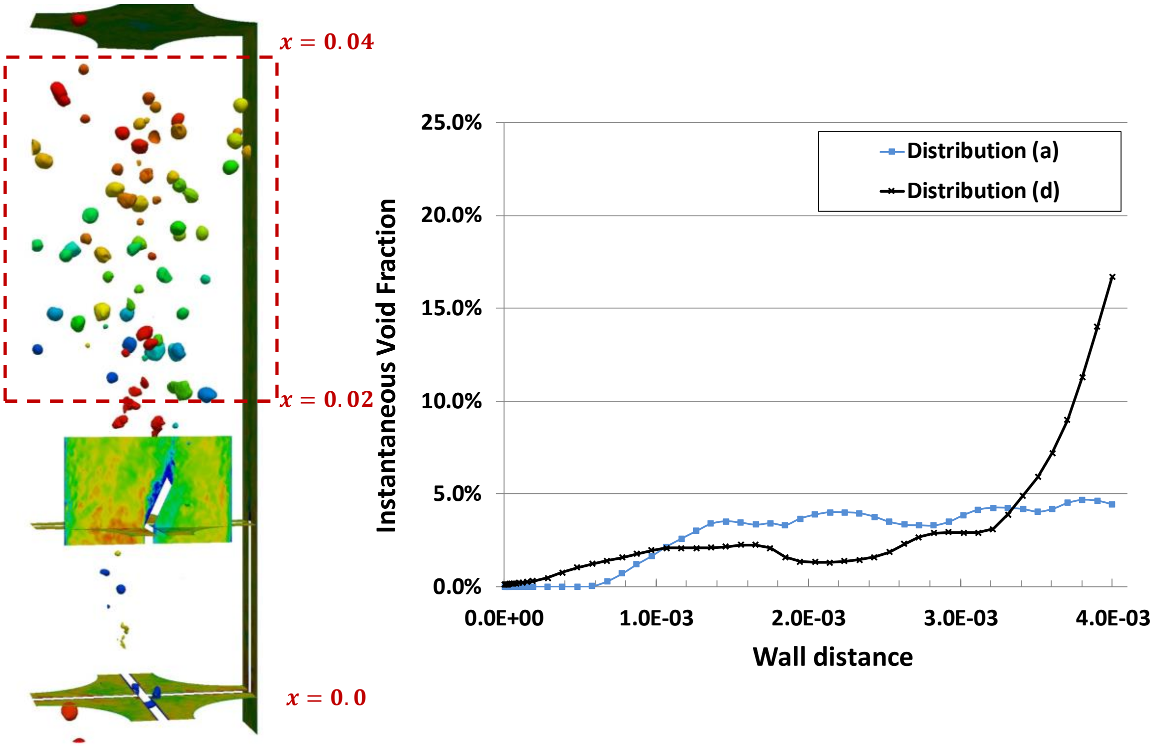


Figure 75: The region selected for bubble tracking analysis and void fraction profiles at initial (a) and final (d) time steps.

CHAPTER 6. CONCLUSIONS

A novel bubble tracking methodology has been developed and tested in this dissertation for level set based two-phase flow simulations. The advanced techniques developed will significantly improve the data collection and analysis capabilities of large-scale two-phase DNS. Experimental data has always been crucial in the development and validation of multiphase flow models. However, it is very challenging to measure some quantities in experiments, for example, the local shear rate around individual bubbles, especially when complex geometries are involved. These quantities are very important and can provide us additional information about bubbles' behavior under different conditions useful for advanced model development. Besides the bubble tracking capability, several advanced auxiliary techniques are also developed for the better large-scale simulation workflow. (1) Previous coalescence control model has been improved by integrating with bubble tracking techniques. (2) By introducing a PID controller, PHASTA was used to estimate the bubble interfacial forces in a controlled environment. The dependences of lift and drag coefficients of single bubbles in laminar low shear flows were investigated and the simulation results show a very good agreement with available experimentally based correlations. (3) In the meantime, to facilitate the discovery process of large scale DNS, in-situ visualization and simulation steering techniques were developed and demonstrated.

A single PWR subchannel geometry is selected as the computational domain to demonstrate the newly developed bubble tracking methodology. Both single- and two-phase bubbly turbulent flows are simulated in the subchannel geometry. The simulation results are then analyzed by both conventional approach and advanced analysis techniques. The conventional analysis can provide time-averaged phasic velocities, void fraction distribution as well as the turbulent kinetic energy profiles. This type of knowledge is important, but not enough. To better justify the large computational cost of large-scale DNS of two-phase flow, one really needs advanced analysis capability to extract more detailed and valuable information from the simulations. Bubble tracking methodology is developed for this purpose, which allows users to obtain in-depth insight of bubbly flow behavior. Based on the bubble tracking data, post-processing techniques can not only reconstruct dynamic 3-D void fraction distribution, but also estimate the bubble interfacial forces with certain classification criteria.

While the computational cost remains a big challenge for larger domain and high Reynolds number simulations, the presented studies clearly indicate the great potential of bubble tracking capability. Considering the continuous development of computing power, DNS coupled with interface tracking and bubble tracking will become a very promising tool to guide the development of M-CFD models as well as subchannel closure laws for nuclear reactor applications.

In conclusion, novel ways of studying and understanding multiphase bubbly flows have been developed, and the accomplished work includes:

a) Developing and demonstrating the novel bubble tracking methodology for nuclear engineering flows of interest;

b) Developing several advanced techniques for a more efficient simulation workflow.

c) Designing and performing large-scale DNS of PWR subchannel geometry for both single- and two-phase turbulent flows;

d) Exploring post-analysis approaches for bubble tracking data.

CHAPTER 7. FUTURE WORK RECOMMENDATIONS

The presented bubble tracking capability brings new level of useful information from large scale two-phase flow simulations. It changes the way we can approach the analysis of the data obtained from DNS. The research work presented in this dissertation will serve as a foundation for future studies. In order to help address the two-phase flow challenges in nuclear engineering, a proposed roadmap of bubble tracking based research is illustrated in Figure 76. Some work has been reported in previous CASL milestone reports; some are under investigation currently and the others will be explored in the near future. The recommended future work will be discussed in the following three directions.

Further development of post-processing techniques

Some advanced analysis examples have been discussed in previous chapters, such as reconstruction of instantaneous 3-D void fraction profile, interfacial force study and preliminary bubble grouping results. These exercises have clearly demonstrated the great potential of bubble tracking methodology. However, one should not be confined by the examples. With the detailed information collected with bubble tracking capability, a more advanced statistical analysis code can be developed to better translate the raw data into closure law-friendly correlations. For instance, in addition to the void fraction dependence with respect to the distance from the wall, one can also investigate distributions of bubble deformation levels or bubble sizes as a function of the distance from the wall. These collections are still quite basic, based on which the development of more sophisticated correlations is possible.

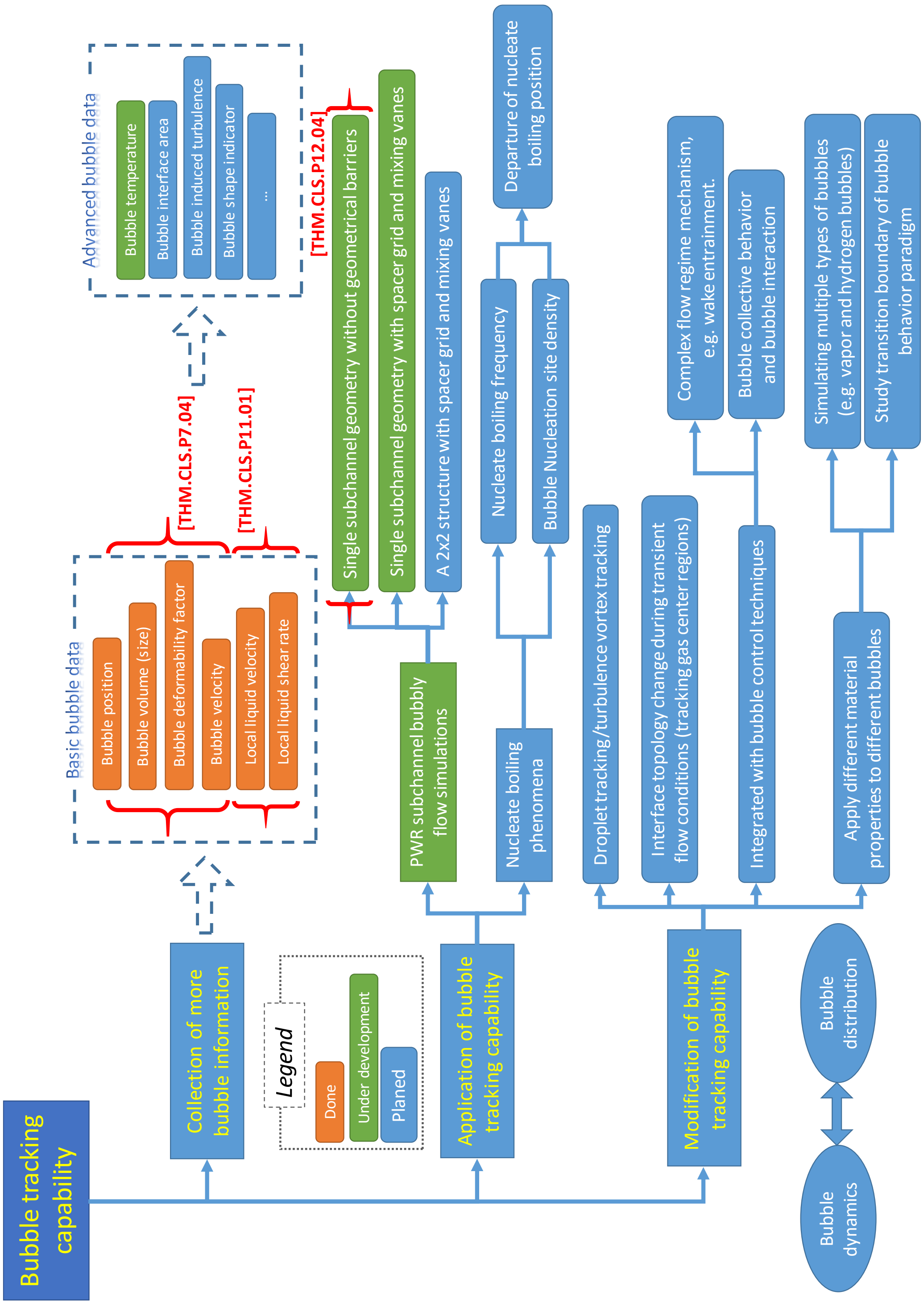

Bubble tracking capability
Collection of more bubble information
Application of bubble tracking capability
Modification of bubble tracking capability
Legend
Done
Under development
Planed
Basic bubble data
Bubble position
Bubble volume (size)
Bubble deformability factor
Bubble velocity
Local liquid velocity
Local liquid shear rate
[THM.CLS.P7.04]
[THM.CLS.P11.01]
Advanced bubble data
Bubble temperature
Bubble interface area
Bubble induced turbulence
Bubble shape indicator
...
[THM.CLS.P12.04]
PWR subchannel bubbly flow simulations
Single subchannel geometry without geometrical barriers
Single subchannel geometry with spacer grid and mixing vanes
A 2x2 structure with spacer grid and mixing vanes
Nucleate boiling phenomena
Nucleate boiling frequency
Bubble Nucleation site density
Departure of nucleate boiling position
Droplet tracking/turbulence vortex tracking
Interface topology change during transient flow conditions (tracking gas center regions)
Integrated with bubble control techniques
Complex flow regime mechanism, e.g. wake entrainment.
Bubble collective behavior and bubble interaction
Apply different material properties to different bubbles
Simulating multiple types of bubbles (e.g. vapor and hydrogen bubbles)
Study transition boundary of bubble behavior paradigm
Bubble dynamics
Bubble distribution

Figure 76: The roadmap of bubble tracking applications.

While only three bubble groups are adopted in current study, more bubble groups can be specified as more bubbles are resolved in future bubbly flow simulation. In other words, large sample/data size would allow higher fidelity studies of two-phase bubbly flows. Given enough bubble tracking data, one could study the correlations between lift/drag coefficient and wall distance, bubble size and/or even local bubble density. Taking into account all these related aspects, a comprehensive interfacial force model would significantly increase the prediction quality of two-phase flow simulations in nuclear reactor designs.

Expansion of bubble tracking capability

In the current bubble tracking methodology, the extracted bubble information includes bubble centroid position, bubble velocity, volume, mass, local liquid velocity and shear rate. In the meantime, it has been left the freedom to extract more useful bubble information according to specific research goals. There are still a lot of interesting bubble parameters one can extract from bubble tracking simulations. One of the examples could be the bubble shape indicator. The defined deformability factor (D_f) may be capable in describing shapes of elliptic (egg-shaped) bubbles. It has been noted that D_f alone is not enough to describe all possible bubble shapes, such as cap bubbles, Taylor bubbles or even slug bubbles. It will be very useful to develop bubble shape identification techniques, which can account for D_f and bubble volume and curvature to produce the corresponding bubble shape indicator. This shape identification and extraction capability would become especially important in flow regime transition studies. Another important example would be the bubble interfacial area. Since bubble tracking methodology is committed to a better understanding of two-phase flows, it would be very helpful if one could extract bubble interfacial area. The valuable data sets from first principles simulations would complement the existing experimental data to help better model source and sink terms in interfacial area transport equations.

Although simulated two-phase flows are assumed to be isothermal in current study, bubble tracking capability will also make a difference in the simulations of boiling problems. The development of boiling and condensation capabilities is an on-going research effort in our group. Once the boiling and condensation capabilities are matured, they will be coupled with bubble tracking for detailed analysis of the boiling flows of interest. Bubble tracking capability would provide the volume, average temperature, departure time and velocity of each bubble, which are all valuable details (in addition to available experimental data) to better understand nucleate boiling phenomenon.

In turbulent two-phase flows, it is of great importance to understand the interaction among bubbles or between bubbles and liquid turbulence. As introduced in Section 4.2, a PID control model has been developed for single bubble simulations. It is practical to incorporate bubble control capability with bubble tracking capability to control a bubble cluster instead of a single bubble. With this feature, one can investigate the interaction among bubbles, for example, in flow regime transition case how the presence of a Taylor bubble influences the lift or drag force the smaller bubbles experience in the wake. Meanwhile, the estimation of bubble induced turbulence is currently under development in our research group too. This capability could also be integrated with bubble tracking capability to investigate the interaction between bubble clusters and liquid turbulence. The addition information will present a clearer and comprehensive picture of turbulent two-phase bubbly flows in various geometries.

Extension of bubble tracking capability

Bubble tracking capability is only one example about how a tracking scheme can follow the evolution of a certain structure in the flow. As an analogy, the similar tracking scheme can also be developed to track the turbulence vortex for which Q-criterion [147] can be used as an indicator to advect the vortex ID field. This new insight can help us better understanding the propagation of liquid turbulence. Another example of bubble tracking transformation is the tracking of gas cores in high void fraction two-phase flows. A gas core here is referred to the region enclosed by a level set contour with a negative value far below zero. Note that the zero level set defines where the gas-liquid interface is. This capability will help clarify the mechanisms underneath flow regime transition phenomena.

In a traditional level set framework, the level set field is not only the phase indicator function, but also responsible to determine the local material properties, such as fluid density, viscosity and thermal conductivity. Now with bubble tracking capability, one can easily distinguish all the individual bubbles and assign different properties to specific bubbles. This flexibility allows simulating some interesting bubbly flows with level set method which are not possible before. For example, in a severe accident scenario where the chemical reaction of Zircaloy and water would lead to the release of hydrogen. If this is the case, both vapor bubbles and hydrogen bubbles could be present in the bubbly flow. Providing the bubble distinction of bubble tracking capability, one can simulate two (or more) types of bubbles in the simulations to represent a more realistic environment. Variable bubble property can also allow one to design computational ‘experiments’ which are not necessarily practical in physical experiments but instructive at the same time. One such example is that one can assign variable surface tension coefficients to simulated bubbles and study their behavior given the same initial conditions. This way it can be found out where a transition boundary is from one bubble behavior paradigm to another.

To conclude the future work recommendations, it is reminded that the bubble tracking methodology developed in this dissertation is just a beginning, but it paves a new way and offers numerous possibilities to better study and understand two-phase turbulent flows.

REFERENCES

[1] Odar, F., C. Murray, R. Shumway, M. Bolander, D. Barber and J. Mahaffy, (2003). "TRAC/RELAP Advanced Computational Engine (TRACE) V4. 0 User's Manual" *US NRC*.

[2] Allison, C. and E. Johnson, (1990). "RELAP5/MOD3 code manual" *US Nuclear Regulatory Commission Report NUREG/CR-5535*.

[3] George, T.L., (2001). "GOTHIC Containment Analysis Package Installation and Operations Manual " Numerical Applications, Inc.

[4] Taitel, Y., D. Bornea and A. Dukler, (1980). "Modelling flow pattern transitions for steady upward gas- liquid flow in vertical tubes" *AIChE J.*, 26(3), 345-354.

[5] Ishii, M. and N. Zuber, (1979). "Drag coefficient and relative velocity in bubbly, droplet or particulate flows" *AIChE J.*, 25(5), 843-855.

[6] Tomiyama, A., I. Kataoka, I. Zun and T. Sakaguchi, (1998). "Drag coefficients of single bubbles under normal and micro gravity conditions" *JSME international journal.Series B, fluids and thermal engineering*, 41(2), 472-479.

[7] Tomiyama, A., H. Tamai, I. Zun and S. Hosokawa, (2002). "Transverse migration of single bubbles in simple shear flows" *Chemical Engineering Science*, 57(11), 1849-1858.

[8] Berry, R., J. Peterson, H. Zhang, R. Martineau, H. Zhao, L. Zou and D. Andrs, (2014). "Relap-7 theory manual" *Idaho National Laboratory, Tech.Rep.INL/EXT-14-31366*.

[9] Tryggvason, G., A. Esmaeeli and N. Al-Rawahi, (2005). "Direct numerical simulations of flows with phase change" *Comput. Struct.*, 83(6), 445-453.

[10] Bolotnov, I.A., K.E. Jansen, D.A. Drew, A.A. Oberai and J.T. Lahey R., (2011). "Detached Direct Numerical Simulation of Turbulent Two-phase Bubbly Channel Flow" *Int. J. Multiphase Flow*, 37, 647-659.

[11] Sussman, M., E. Fatemi, P. Smereka and S. Osher, (1998). "An improved level set method for incompressible two-phase flows" *Comput. Fluids*, 27(5), 663-680.

[12] Sussman, M., A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell and M.L. Welcome, (1999). "An adaptive level set approach for incompressible two-phase flows" *Journal of Computational Physics*, 148(1), 81-124.

[13] Sethian, A.J., (1999) "Level Set Methods and Fast Marching Methods" Cambridge University Press.

[14] Nagrath, S., K. Jansen, R.T. Lahey Jr and I. Akhatov, (2006). "Hydrodynamic simulation of air bubble implosion using a level set approach" *Journal of Computational Physics*, 215(1), 98-132.

[15] Bolotnov, I.A., (2013). "Influence of Bubbles on the Turbulence Anisotropy" *Journal of Fluids Engineering*, 135(5), 051301.

[16] Behafarid, F., K. Jansen and M. Podowski, (2015). "A study on large bubble motion and liquid film in vertical pipes and inclined narrow channels" *Int. J. Multiphase Flow*, 75, 288-299.

[17] Rodriguez, J., O. Sahni, R. Lahey and K. Jansen, (2013). "A parallel adaptive mesh method for the numerical simulation" *Computers and Fluids*, 87, 115-131.

[18] Thomas, A.M., J. Fang, J. Feng and I.A. Bolotnov, (2015). "Estimation of shear-induced lift force in laminar and turbulent flows" *Nuclear Technology*, 109, 1-18.

[19] Trupp, A.C. and R. Azad, (1975). "The structure of turbulent flow in triangular array rod bundles" *Nucl. Eng. Des.*, 32(1), 47-84.

[20] Carajilescov, P. and N. Todreas, (1976). "Experimental and analytical study of axial turbulent flows in an interior subchannel of a bare rod bundle" *Journal of Heat Transfer*, 98(2), 262-268.

[21] Neti, S., R. Eichhorn and O. Hahn, (1983). "Laser doppler measurements of flow in a rod bundle" *Nucl. Eng. Des.*, 74(1), 105-116.

[22] Vonka, V., (1988). "Measurement of secondary flow vortices in a rod bundle" *Nucl. Eng. Des.*, 106(2), 191-207.

[23] Rehme, K., (1972). "Pressure drop performance of rod bundles in hexagonal arrangements" *Int. J. Heat Mass Transfer*, 15(12), 2499-2517.

[24] Rehme, K., (1973). "Pressure drop correlations for fuel element spacers" *Nucl Technol*, 17(1), 15-23.

[25] Rehme, K. and G. Trippe, (1980). "Pressure drop and velocity distribution in rod bundles with spacer grids" *Nucl. Eng. Des.*, 62(1), 349-359.

[26] Rehme, K., (1989). "Experimental observations of turbulent flow through subchannels of rod bundles" *Exp. Therm. Fluid Sci.*, 2(3), 341-349.

[27] Engel, F., R. Markley and A. Bishop, (1979). "Laminar, transition, and turbulent parallel flow pressure drop across wire-wrap-spaced rod bundles" *Nucl. Sci. Eng.*, 69(2), 290-296.

[28] Bubelis, E. and M. Schikorr, (2008). "Review and proposal for best fit of wire-wrapped fuel bundle friction factor and pressure drop predictions using various existing correlations" *Nucl. Eng. Des.*, 238(12), 3299-3320.

[29] Todreas, N. and L. Wilson, (1968). "Coolant mixing in sodium cooled fast reactor fuel bundles".

[30] Rogers, J. and N. Todreas, (1969). "Coolant interchannel mixing in reactor fuel rod bundles: single-phase coolants".

[31] Yang, S.K. and M.K. Chung, (1998). "Turbulent flow through spacer grids in rod bundles" *Journal of Fluids Engineering*, 120(4), 786-791.

[32] Shirai, H. and H. Ninokata, (2001). "Prediction of the equilibrium two-phase flow distributions in inter-connected subchannel systems" *J Nucl Sci Technol*, 38(6), 379-387.

[33] Dominguez-Ontiveros, E.E. and Y.A. Hassan, (2009). "Non-intrusive experimental investigation of flow behavior inside a 5× 5 rod bundle with spacer grids using PIV and MIR" *Nucl. Eng. Des.*, 239(5), 888-898.

[34] Murai, Y., Y. Matsumoto and F. Yamamoto, (2001). "Three-dimensional measurement of void fraction in a bubble plume using statistic stereoscopic image processing" *Exp. Fluids*, 30(1), 11-21.

[35] Serizawa, A., I. Kataoka and I. Michiyoshi, (1975). "Turbulence structure of air-water bubbly flow—I. Measuring techniques" *Int. J. Multiphase Flow*, 2(3), 221-233.

[36] Serizawa, A., I. Kataoka and I. Michiyoshi, (1975). "Turbulence structure of air-water bubbly flow—II. Local properties" *Int. J. Multiphase Flow*, 2(3), 235-246.

[37] Serizawa, A., I. Kataoka and I. Michiyoshi, (1975). "Turbulence structure of air-water bubbly flow—III. Transport properties" *Int. J. Multiphase Flow*, 2(3), 247-259.

[38] Yun, B., G. Park, J.E. Julia and T. Hibiki, (2008). "Flow structure of subcooled boiling water flow in a subchannel of 3× 3 rod bundles" *J Nucl Sci Technol*, 45(5), 402-422.

[39] Wheeler, J., T. Worosz and S. Kim, (2015). "Experiments on the Effects of a Spacer Grid in Air-Water Two-phase Flow" *Nucl Technol*, 190(3), 215-224.

[40] Hibiki, T. and M. Ishii, (2000). "One-group interfacial area transport of bubbly flows in vertical round tubes" *Int.Journal of Heat and Mass Transfer*, 43, 2711-2726.

[41] Hibiki, T. and M. Ishii, (2000). "Two-group interfacial area transport equations at bubbly-to-slug flow transition" *Nucl. Eng. Des.*, 202(1), 39-76.

[42] Kunugi, T., (2012). "Brief Review of Latest Direct Numerical Simulation on Pool and Film Boiling" *NUCLEAR ENGINEERING AND TECHNOLOGY*, 44(8), 847-854.

[43] Hibiki, T. and M. Ishii, (2007). "Lift force in bubbly flow systems" *Chemical Engineering Science*, 62(22), 6457-6474.

[44] Zun, I., (1980). "Transverse migration of bubbles influenced by walls in vertical bubbly flow" *Int. J. Multiphase Flow*, 6, 583-588.

[45] Drew, D. and R. Lahey, (1987). "The virtual mass and lift force on a sphere in rotating and straining inviscid flow" *Int. J. Multiphase Flow*, 13(1), 113-121.

[46] Ishii, M. and K. Mishima, (1984). "Two-fluid model and hydrodynamic constitutive relations" *Nucl. Eng. Des.*, 82(2), 107-126.

[47] Lahey, R., M.L. de Bertodano and O. Jones, (1993). "Phase distribution in complex geometry conduits" *Nucl. Eng. Des.*, 141(1), 177-201.

[48] Zuber, N. and J. Findlay A., (1965). "Average volumetric concentration in two-phase flow systems" *Journal of Heat Transfer*, 87(4), 453-468.

[49] Clift, R., J.R. Grace, M.E. Weber, (2005) "Bubbles, Drops, and Particles" Courier Corporation.

[50] Magnaudet, J. and D. Legendre, (1998). "The viscous drag force on a spherical bubble with a time-dependent radius" *Physics of Fluids (1994-present)*, 10(3), 550-554.

[51] Roghair, I., Y. Lau, N. Deen, H. Slagter, M. Baltussen, M.V.S. Annaland and J. Kuipers, (2011). "On the drag force of bubbles in bubble swarms at intermediate and high Reynolds numbers" *Chemical engineering science*, 66(14), 3204-3211.

[52] Mazzitelli M., I. and D. Lohse, (2003). "The effect of microbubbles on developed turbulence." *Phys. Fluids*, 15(1), L5-L8.

[53] Mazzitelli, I.M., D. Lohse and F. Toschi, (2003). "On the relevance of the lift force in bubbly turbulence" *J. Fluid Mech.*, 488, 283-313.

[54] Saffman G., P., (1965). "The lift on a small sphere in a slow shear flow" *J. Fluid Mech.*, 22(2), 385-400.

[55] Saffman G., P., (1968). "Corrigendum to "The lift on a small sphere in a slow shear flow"" *J. Fluid Mech.*, 31, 624.

[56] Ervin, E. and G. Tryggvason, (1997). "The rise of bubbles in a vertical shear flow" *Journal of Fluids Engineering*, 119(2), 443-449.

[57] Tomiyama, A., (2004). *"Drag, lift, and virtual mass forces acting on a single bubble." Proceeding of the Third International Symposium on Two-Phase Flow Modeling and Experimentation,*.

[58] Cherukat, P. and J. McLaughlin B., (1994). "The inertial lift on a rigid sphere in a linear shear flow field near a flat wall." *J. Fluid Mech.*, 263, 1-18.

[59] Sridhar, G. and J. Katz, (1995). "Drag and lift forces on microscopic bubbles entrained by a vortex" *Phys. Fluids*, 7, 389-399.

[60] Pascal, P. and B. Oesterle, (2000). "On the dispersion of discrete particles moving in a turbulent shear flow" *Int.Journal of Multiphase Flow*, 26(2), 293-325.

[61] Liu J., T., (1993). "Bubble size and entrance length effects on void development in a vertical channel" *Int.Journal of Multiphase Flow*, 19(1), 99-113.

[62] Hibiki, T. and M. Ishii, (1999). "Experimental study on interfacial area transport in bubbly two-phase flows." *Int. J. Heat Mass Transfer*, 42, 3019-3035.

[63] Hibiki, T., M. Ishii and Z. Xiao, (2001). "Axial interfacial area transport of vertical bubbly flows" *Int. J. Heat Mass Transfer*, 44, 1869-1888.

[64] Hibiki, T., R. Situ, Y. Mi and M. Ishii, (2003). "Local flow measurements of vertical upward bubbly flow in an annulus" *Int. J. Heat Mass Transfer*, 46, 1479-1496.

[65] Takagi, S. and Y. Matsumoto, (1995). *"Three dimensional calculation of a rising bubble." Proceedings of the Second International Conference on Multiphase Flow,.*

[66] Tomiyama, A., I. Zun, A. Sou and T. Sakaguchi, (1993). "Numerical analysis of bubble motion with the VOF method" *Nucl. Eng. Des.*, 141, 69-82.

[67] Milne-Thomson, L.M., (1968) "Theoretical Hydrodynamics" Courier Corporation.

[68] Drew, D., L. Cheng and R. Lahey, (1979). "The analysis of virtual mass effects in two-phase flow" *Int. J. Multiphase Flow*, 5(4), 233-242.

[69] de Bertodano, Martin A Lopez, (1998). "Two fluid model for two-phase turbulent jets" *Nucl. Eng. Des.*, 179(1), 65-74.

[70] Drew, D.A., (2001). "A turbulent dispersion model for particles or bubbles" *J Eng Math*, 41(2-3), 259-274.

[71] Winters, J., R. Vijuk and W. Cummins, (2004). "AP1000 Design Control Document" *Weistinghouse Electric Company LLC, Pittsbrugh, PA, US*.

[72] Turinsky, P.J. and D.B. Kothe, (2016). "Modeling and simulation challenges pursued by the Consortium for Advanced Simulation of Light Water Reactors (CASL)" *Journal of Computational Physics*, 313, 367-376.

[73] Thurgood, M.J., (1983) "COBRA/TRAC, a Thermal-Hydraulics Code for Transient Analysis of Nuclear Reactor Vessels and Primary Coolant Systems" The Commission.

[74] Avramova, M., (2007). "Development of an innovative spacer grid model utilizing computational fluid dynamics within a subchannel analysis tool".

[75] Conner, M.E., E. Baglietto and A.M. Elmahdi, (2010). "CFD methodology and validation for single-phase flow in PWR fuel assemblies" *Nucl. Eng. Des.*, 240(9), 2088-2095.

[76] Baglietto, E., H. Ninokata and T. Misawa, (2006). "CFD and DNS methodologies development for fuel bundle simulations" *Nucl. Eng. Des.*, 236(14), 1503-1510.

[77] Karman, T.V., (1937). "The fundamentals of the statistical theory of turbulence" *Journal of the Aeronautical Sciences*, 4(4), 131-138.

[78] Kirkwood, J.G., (1935). "Statistical mechanics of fluid mixtures" *J. Chem. Phys.*, 3(5), 300-313.

[79] Chisholm, D., (1967). "A theoretical basis for the Lockhart-Martinelli correlation for two-phase flow" *Int. J. Heat Mass Transfer*, 10(12), 1767-1778.

[80] Ishii, M., (1975) "Thermo-Fluid Dynamic Theory of Two-Phase Flow", Paris.

[81] Lahey Jr, R. and D. Drew, (1992). "On the development of multidimensional two-fluid models for vapor/liquid two-phase flows" *Chem. Eng. Commun.*, 118(1), 125-139.

[82] Lahey Jr, R.T. and D.A. Drew, (2000). *"An analysis of two-phase flow and heat transfer using a multidimensional, multi-field, two-fluid computational fluid dynamics (CFD) model" Japan/US Seminar on Two-Phase Flow Dynamics*, 5-8.

[83] Lahey Jr, R.T., (2005). "The simulation of multidimensional multiphase flows" *Nucl. Eng. Des.*, 235(10), 1043-1060.

[84] Podowski, M., (2009). "On the consistency of mechanistic multidimensional modeling of gas/liquid two-phase flows" *Nucl. Eng. Des.*, 239(5), 933-940.

[85] Ustinenko, V., M. Samigulin, A. Ioilev, S. Lo, A. Tentner, A. Lychagin, A. Razin, V. Girin and Y. Vanyukov, (2008). "Validation of CFD-BWR, a new two-phase computational fluid dynamics model for boiling water reactor analysis" *Nucl. Eng. Des.*, 238(3), 660-670.

[86] Anglart, H. and O. Nylund, (1996). "CFD application to prediction of void distribution in two-phase bubbly flows in rod bundles" *Nucl. Eng. Des.*, 163(1), 81-98.

[87] Lee, C. and I. Mudawwar, (1988). "A mechanistic critical heat flux model for subcooled flow boiling based on local bulk flow conditions" *Int. J. Multiphase Flow*, 14(6), 711-728.

[88] Končar, B., I. Kljenak and B. Mavko, (2004). "Modelling of local two-phase flow parameters in upward subcooled flow boiling at low pressure" *Int. J. Heat Mass Transfer*, 47(6), 1499-1513.

[89] Krepper, E., B. Končar and Y. Egorov, (2007). "CFD modelling of subcooled boiling—concept, validation and application to fuel assembly design" *Nucl. Eng. Des.*, 237(7), 716-731.

[90] Lee, M., N. Malaya and R.D. Moser, (2013). *"Petascale direct numerical simulation of turbulent channel flow on up to 786k cores" Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis*, 61.

[91] Trofimova, A.V., A.E. Tejada-Martínez, K.E. Jansen and R.T. Lahey Jr, (2009). "Direct numerical simulation of turbulent channel flows using a stabilized finite element method" *Comput. Fluids*, 38(4), 924-938.

[92] Kim, J., P. Moin and R. Moser, (1987). "Turbulence statistics in fully developed channel flow at low Reynolds number" *J. Fluid Mech.*, 177, 133-166.

[93] Moser, R.D., J. Kim and N.N. Mansour, (1999). "Direct numerical simulation of turbulent channel flow up to Re= 590" *Phys. Fluids*, 11(4), 943-945.

[94] Lee, M. and R.D. Moser, (2015). "Direct numerical simulation of turbulent channel flow up to Re_tau 5200" *J. Fluid Mech.*, 774, 395-415.

[95] Fang, J., M. Rasquin and I.A. Bolotnov, (2015). *"Interface Tracking Simulations of Bubbly Flows in the PWR Relevant Geometries" 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-16)*, 7115-7128, Chicago, IL.

[96] Ninokata, H., N. Atake, E. Baglietto, T. Misawa and T. Kano, (2004). "Direct numerical simulation of turbulent flows in a subchannel of tight lattice fuel pin bundles of nuclear reactors" *Annual Report of the Earth Simulator Center April*, 2005.

[97] Lu, J. and G. Tryggvason, (2008). "Effect of bubble deformability in turbulent bubbly upflow in a vertical channel" *Physics of Fluids (1994-present)*, 20(4), 040701.

[98] Dabiri, S., J. Lu and G. Tryggvason, (2013). "Transition between regimes of a vertical channel bubbly upflow due to bubble deformability" *Physics of Fluids (1994-present)*, 25(10), 102110.

[99] Fang, J., A.M. Thomas and I.A. Bolotnov, (2013). *"Development of Advanced Analysis Tools for Interface Tracking Simulations" Transactions of the 2013 ANS Winter Meeting*, 109, 1613-1615, Washington D.C.

[100] Prosperetti, A., G. Tryggvason, (2007) "Computational Methods for Multiphase Flow" Cambridge university press.

[101] Hirt, C.W. and B.D. Nichols, (1981). "Volume of fluid (VOF) method for the dynamics of free boundaries" *Journal of computational physics*, 39(1), 201-225.

[102] Unverdi, S.O. and G. Tryggvason, (1992). "A front-tracking method for viscous, incompressible, multi-fluid flows" *Journal of computational physics*, 100(1), 25-37.

[103] Sussman, M., P. Smereka and S. Osher, (1994). "A level set approach for computing solutions to incompressible two-phase flow" *Journal of Computational physics*, 114(1), 146-159.

[104] Noh, W.F. and P. Woodward, (1976). *"SLIC (simple line interface calculation)" Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, 330-340.

[105] Youngs, D.L., (1982). "Time-dependent multi-material flow with large fluid distortion" *Numerical methods for fluid dynamics*, 24, 273-285.

[106] Rider, W.J. and D.B. Kothe, (1995). "Stretching and tearing interface tracking methods" *AIAA paper*, 95, 1-11.

[107] Chern, I., J. Glimm, O. McBryan, B. Plohr and S. Yaniv, (1986). "Front tracking for gas dynamics" *Journal of Computational Physics*, 62(1), 83-110.

[108] Ma, M., J. Lu and G. Tryggvason, (2014). "DNS and modeling of bubbly flows in vertical channels" *Bulletin of the American Physical Society*, 59.

[109] Tryggvason, G., B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas and Y. Jan, (2001). "A front-tracking method for the computations of multiphase flow" *Journal of Computational Physics*, 169(2), 708-759.

[110] Tryggvason, G., R. Scardovelli, S. Zaleski, (2011) "Direct Numerical Simulations of Gas–liquid Multiphase Flows" Cambridge University Press.

[111] Osher, S. and J.A. Sethian, (1988). "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations" *Journal of computational physics*, 79(1), 12-49.

[112] Sussman, M. and E. Fatemi, (1999). "An efficient, interface-preserving level set re-distancing algorithm and its application to interfacial incompressible fluid flow" *Siam J.on Scientific Computing*, 20(4), 1165-1191.

[113] Fang, J., A.V. Mishra and I.A. Bolotnov, (2014). *"Interface Tracking Simulation of Two-phase Bubbly Flow in A PWR Subchannel" International Embedded Topical Meeting on Advances in Thermal Hydraulics - 2014 (ATH '14),* Reno, NV.

[114] Fang, J. and I.A. Bolotnov, (2015). *"Application of Bubble Tracking Capability for Turbulent Two-phase Flow Simulations of a PWR Subchannel" Transactions of 2015 ANS winter meeting,* 113, 1549-1551, Washington, DC.

[115] Yi, H., M. Rasquin, A.V. Mishra, J. Fang and I.A. Bolotnov, (2014). *"In-Situ Processing and Visualization for Direct Numerical Simulation of Coolant Flow through Mixing Vanes" Proceedings of ICAPP 2014,* Charlotte, USA.

[116] Yi, H., M. Rasquin, J. Fang and I.A. Bolotnov, (2014). *"In-Situ Visualization and Computational Steering for Large-Scale Simulation of Turbulent Flows in Complex Geometries" 2014 IEEE International Conference on Big Data,* 567-572, Washington DC.

[117] Squillacote, A.H., (2007) "The ParaView Guide: A Parallel Visualization Application" Kitware.

[118] Jansen, K.E., (1999). "A stabilized finite element method for computing turbulence" *Comput. Methods Appl. Mech. Eng.*, 174(3), 299-317.

[119] Whiting, C.H. and K.E. Jansen, (2001). "A stabilized finite element method for the incompressible Navier-Stokes equations using a hierarchical basis" *Int. J. Numer. Methods Fluids*, 35(1), 93-116.

[120] Jansen, K.E., (1993). "Unstructured grid large eddy simulations of wall bounded flows" *Annual Research Briefs, Center for Turbulence Research, NASA Ames/Stanford University*, 151.

[121] Sahni, O., J. Müller, K. Jansen, M. Shephard and C. Taylor, (2006). "Efficient anisotropic adaptive discretization of the cardiovascular system" *Comput. Methods Appl. Mech. Eng.*, 195(41), 5634-5655.

[122] Tejada-Martínez, A.E. and K.E. Jansen, (2006). "A parameter-free dynamic subgrid-scale model for large-eddy simulation" *Comput. Methods Appl. Mech. Eng.*, 195(23), 2919-2938.

[123] Rasquin, M., C. Smith, K. Chitale, E.S. Seol, B.A. Matthews, J.L. Martin, O. Sahni, R.M. Loy, M.S. Shephard and K.E. Jansen, (2014). "Scalable Implicit Flow Solver for Realistic Wing Simulations with Flow Control" *Computing in Science & Engineering*, 16(6), 13-21.

[124] Nagrath, S., K.E. Jansen and R.T. Lahey, (2005). "Computation of incompressible bubble dynamics with a stabilized finite element level set method" *Comput. Methods Appl. Mech. Eng.*, 194(42), 4565-4587.

[125] Brackbill, J.U., D.B. Kothe and C. Zemach, (1992). "A continuum method for modeling surface tension" *Journal of computational physics*, 100(2), 335-354.

[126] Fatemi, E. and M. Sussman, (1995). "An efficient interface preserving Level-Set Re-distancing algorithm and its application to interfacial incompressible fluid flow" *SIAM J Sci Statist Comput*, 158(1), 36.

[127] Ibanez, D.A., E.S. Seol, C.W. Smith and M.S. Shephard, (2015). "PUMI: parallel unstructured mesh infrastructure" *ACM Transactions on Mathematical Software*.

[128] Smith, C.W., M. Rasquin, D. Ibanez, K.E. Jansen and M.S. Shephard, (2015). "Application Specific Mesh Partition Improvement" *SIAM Journal on Scientific Computing*.

[129] Rubinstein, R.Y., D.P. Kroese, (2011) "Simulation and the Monte Carlo Method" John Wiley & Sons.

[130] Mishra, A.V. and I.A. Bolotnov, (2015). "DNS of turbulent flow with hemispherical wall roughness" *Journal of Turbulence*, 16(3), 225-249.

[131] Van Driest, E.R., (2012). "On turbulent flow near a wall" *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, 23(11).

[132] Bolotnov, I.A. and J. Fang, (2015). *"Interface tracking simulations as an emerging tool for two-phase flow model development" 2015 Japan-U.S. seminar on Two-Phase flow Dynamics,* West Lafayette, Indiana, USA.

[133] Cambareri, J. and I.A. Bolotnov, (2016). *"Interface Tracking Simulations of Molten Glass for Melter Design" 2016 ANS Student Conference,* Madison Wisconsin.

[134] Talley, M.L., (2014). "Bubble coalescence control development for level set interface tracking method".

[135] Ishii, M. and T. Chawla, (1979). "Local drag laws in dispersed two-phase flow" Argonne National Lab., IL (USA).

[136] Mishima, K. and M. Ishii, (1984). "Flow regime transition criteria for upward two-phase flow in vertical tubes" *Int. J. Heat Mass Transfer*, 27(5), 723-737.

[137] Antal, S., R. Lahey and J. Flaherty, (1991). "Analysis of phase distribution in fully developed laminar bubbly two-phase flow" *Int. J. Multiphase Flow*, 17(5), 635-652.

[138] Legendre, D. and J. Magnaudet, (1998). "The lift force on a spherical bubble in a viscous linear shear flow." *J. Fluid Mech.*, 368, 81-126.

[139] Multiphase Flows Group, Group Interfaces, M. LANCE, J. MARIÉ, E. MOURSALI, J. BATAILLE, C. SUZANNE, V. ROIG, R. BEL FDHILA and L. MASBERNAT, (1996). "Experimental study of turbulent bubbly shear flows" *Chem. Eng. Commun.*, 141(1), 51-70.

[140] Ma, K., C. Wang, H. Yu and A. Tikhonova, (2007). *"In-situ processing and visualization for ultrascale simulations" Journal of Physics: Conference Series*, 78, 012043.

[141] Yu, H., C. Wang, R.W. Grout, J.H. Chen and K. Ma, (2010). "In situ visualization for large-scale combustion simulations." *IEEE Comput. Graphics Appl.*, 30(3), 45-57.

[142] Fabian, N., K. Moreland, D. Thompson, A.C. Bauer, P. Marion, B. Geveci, M. Rasquin and K.E. Jansen, (2011). *"The paraview coprocessing library: A scalable, general purpose in situ visualization library" Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, 89-96.

[143] Bauer, A.C., B. Geveci and W. Schroeder, (2013). "The ParaView Catalyst User's Guide".

[144] Schroeder, W., K. Martin and B. Lorensen, (2003). "The visualization toolkit: an object oriented approach to 3D graphics. New York: Kitware" *Inc.Publisher*, 156.

[145] Pope, S.B., (2000) "Turbulent Flows" Cambridge university press.

[146] Thomas, A.M., (2014). "Estimation of the Shear-Induced Lift Force on a Single Bubble in Laminar and Turbulent Shear Flows Using Interface Tracking Approach".

[147] Haller, G., (2005). "An objective definition of a vortex" *J. Fluid Mech.*, 525, 1-26.

[148] Wang K., S., S. Lee J., O. Jones C. and R. Lahey T., (1987). "3-D turbulence structure and phase distribution measurements in bubbly two-phase flows" *Int. J. Multiphase Flow*, 13(3), 327-343.

[149] Ylönen, A., (2013). "High-resolution flow structure measurements in a Rod bundle".

[150] Waite, B.M., D. Shaver and M.Z. Podowski, (2015). *"Mechanistic Modeling of Two-Phase Flow Around Spacer Grids with Mixing Vanes" 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-16)*, 7101-7114, Chicago, IL.

[151] Colebrook, C.F., (1939). "Turbulent Flow in Pipes, with particular reference to the Transition Region between the Smooth and Rough Pipe Laws." *Journal of the ICE*, 11(4), 133-156.

[152] Rider, W.J. and D.B. Kothe, (1998). "Reconstructing volume tracking" *Journal of computational physics*, 141(2), 112-152.

[153] Enright, D., R. Fedkiw, J. Ferziger and I. Mitchell, (2002). "A hybrid particle level set method for improved interface capturing" *Journal of Computational physics*, 183(1), 83-116.

APPENDICES

Appendix A

Maple[9] aided mesh design calculations

For simulated flat channel, pipe or PWR subchannel geometry, the hydraulic diameter (D_h) is calculated based on the cross section area (A) and wetted perimeter (P_w),

$$D_h = \frac{4A}{P_w} \tag{45}$$

and the Reynolds number is defined as

$$Re_{D_h} = \frac{U_m D_h}{\nu} \tag{46}$$

where U_m is the mean bulk velocity in the streamwise direction and ν is liquid viscosity.

Colebrook-White equation [151] is used to calculate the Darcy friction factor which is required in the estimation of wall shear stress,

$$\frac{1}{\sqrt{f_D}} = -2.0 \log_{10}\left(\frac{\epsilon}{3.7D_h} + \frac{2.51}{Re_{D_h} \cdot \sqrt{f_D}}\right) \tag{47}$$

where the wall roughness ϵ is assumed to be zero here.

The wall shear stress is then approximated by

$$\tau_w = f \cdot \left(\frac{1}{2}\rho U_m^2\right) = \frac{\rho U_m^2 \cdot f_D}{8} \tag{48}$$

Together with the wall surface area A_w, the wall shear stress can be used to estimate the desired pressure gradient

[9] Maple is a commercial computer algebra system developed and sold commercially by Maplesoft, a software company based in Waterloo, Ontario, Canada. The current major version is version 2016, which was released in March 2016.

$$\frac{dp}{dx} = \frac{\tau_w \cdot A_w}{A \cdot L} \tag{49}$$

where L is the domain length. The friction velocity is defined as (for two-phase flows liquid density is used since the wall is assumed to be always lubricated in the flows of interest):

$$u_\tau = \sqrt{\frac{\tau_w}{\rho_l}} \tag{50}$$

The length scale corresponds to the viscous sub-layer $y^+ = 1$ can then be approximated

$$\Delta y = \frac{\nu \cdot (y^+ = 1)}{u_\tau} \tag{51}$$

The mesh resolution is the finest in the near wall region to fully resolve large gradients, and the boundary layer mesh thickness grows from $\Delta y^+ = 1$ at the first layer off the wall to $\Delta y^+ = 10$ in the general bulk region. The choice of $\Delta y^+ = 10$ is comparable to the typical mesh resolution used in Moin and Moser's work [93]. Thirteen sublayers are created in the wall boundary layer region with an increase ratio of 1.2 (note that $1.2^{13} \cong 10$ to allow a smooth transient from the boundary layer mesh region to the bulk mesh region).

Appendix B

Mass conservation treatment for constant void fraction simulations

Although level set interface tracking method has been proved to be a promising alternative [103] to the volume of fluid (VOF) and front-tracking (FT) methods, it is infamous for mass conservation issues, especially for under-resolved flows [152]. The volume loss problem has been observed in some PHASTA simulations. We have introduced a fix (not a permanent solution) to ensure constant void fraction for adiabatic two-phase cases. The level set advection equation is given by Eq. (10), in which $\boldsymbol{u}$ is also referred as 'extension velocity' by Sethian [13]. As long as the expected void fraction is known, a simplified PID controller can be used to compare the current void fraction with the prescribed value, and then manipulate the extension velocity accordingly to preserve the volume fraction. Illustrated in Figure 77, a test case with prescribed volume fraction of 0.695% was simulated with mass conservation control, and the control can successfully work as expected.

There are more complicated approaches come up with to improve the mass conservation properties of level set method, for example, the Particle Level Set (PLS) method proposed by Enright et al. combines a level set function with Lagrangian maker particles and is able to considerably improve mass conservation in under-resolved flows while maintaining a smooth interface [153]. However, those approaches are computationally expensive for large scale simulations, and thus the appropriate improvements to the level set method is out of scope of the presented work.

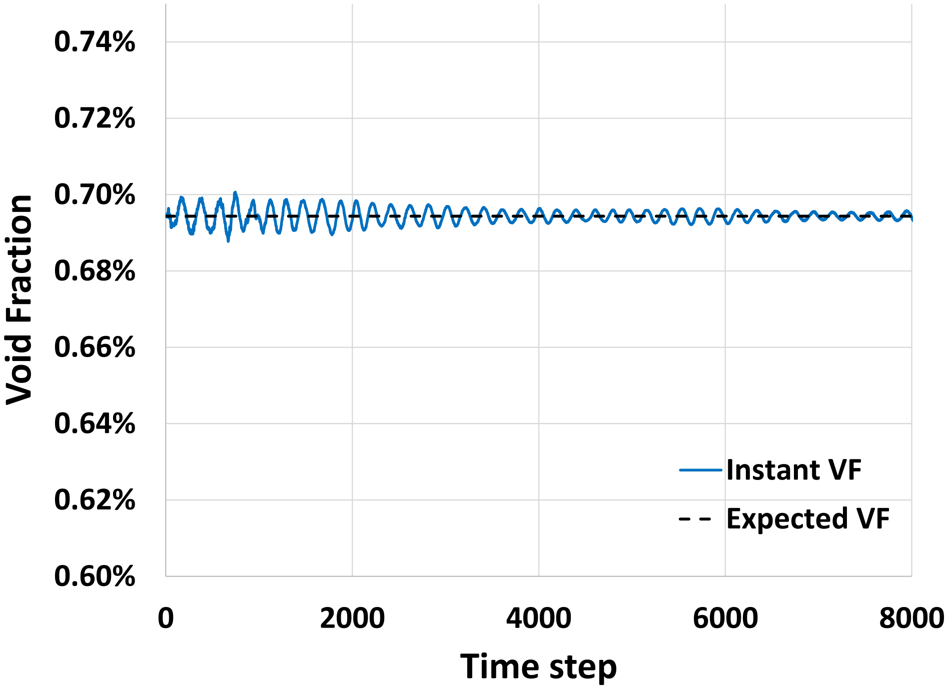


Figure 77: The void fraction evolution in a test case with mass conservation control

Appendix C

Here we present the selected algorithms in bubble generation code for single PWR subchannel geometry.

Case 1: the algorithm to find out the minimum distance from the walls or other bubbles.

```
!...Find the minimum distance to the walls
           distw(1)=sqrt((bubcoord(i,3))**2 + (bubcoord(i,2))**2)
           distw(2)=sqrt((bubcoord(i,3)-rbox2(3))**2 +
     &                  (bubcoord(i,2))**2)
           distw(3)=sqrt((bubcoord(i,3))**2 +
     &                  (bubcoord(i,2)-rbox2(2))**2)
           distw(4)=sqrt((bubcoord(i,3)-rbox2(3))**2 +
     &                  (bubcoord(i,2)-rbox2(2))**2)
           minw(i)=distw(1)
           DO j = 2, 4
              IF(minw(i).GE. distw(j)) minw(i)=distw(j)
           END DO
!...Find the minimum distance to other bubbles
           mindistmp = 1e9
           do j=1,i-1  !loop over the centers that have already been found
              distance=sqrt(((bubcoord(i,1)-bubcoord(j,1))**2)+
     &                      ((bubcoord(i,2)-bubcoord(j,2))**2)+
     &                      ((bubcoord(i,3)-bubcoord(j,3))**2))
              if(mindistmp.gt.distance) mindistmp = distance
           end do
```

Case 2: the algorithm to specify the ghost bubbles near periodic boundaries.

```
        nghost = 1
        DO i=1,4*nbub
           DO j=1,3
             if (j.eq.1) then
              ghostupplim = rbox1(2*j)   -ghost_area_ratio*outline(j)
             else
              ghostupplim = rbox1(2*j-1) -ghost_area_ratio*outline(j)
     &                    + outline(j)
             endif
              ghostlowlim = rbox1(2*j-1) +ghost_area_ratio*outline(j)
              IF (bubdupl(i,j) .GT. ghostupplim) THEN
                 bublist(4*nbub+nghost,2)=bubdupl(i,1)
                 bublist(4*nbub+nghost,3)=bubdupl(i,2)
                 bublist(4*nbub+nghost,4)=bubdupl(i,3)
                 bublist(4*nbub+nghost,j+1)=bubdupl(i,j)-outline(j)
                 WRITE(6,*) 'Bubble',i,'has the ghost bubble'
                 nghost=nghost+1
              ELSEIF (bubdupl(i,j) .LT. ghostlowlim) THEN
                 bublist(4*nbub+nghost,2)=bubdupl(i,1)
                 bublist(4*nbub+nghost,3)=bubdupl(i,2)
                 bublist(4*nbub+nghost,4)=bubdupl(i,3)
                 bublist(4*nbub+nghost,j+1)=bubdupl(i,j)+outline(j)
```

```
         WRITE(6,*) 'Bubble',i,'has the ghost bubble'
         nghost=nghost+1
      ENDIF
   ENDDO
ENDDO
```

Appendix D

The updating algorithm of bubble ID/marker field (called in ASIGMR.F).

```
        subroutine banmaUpdate(xl, yl, banma, ien, bml)
!----------------------------------------------------------------------
!       Called in asigmr.f
!       This subroutine will update the marker field in each time
!       iteration.
!       Banma is the Chinese pinyin translation of Zebra, an African
!       wild horse with black-and-white stripes (markers) and an
!       erect mane.
!       The code will first find out the max of banma values in each
!       element and update the rest based on their levelset value on a
!       certain processor, the markers are updated from one element to
!       another, and this update is conducting samutaniously on
!       different processors
!
!----------------------------------------------------------------------
        use bub_track   ! access to bubble information array
        include "common.h"

        dimension banma(nshg,1),              bml(npro,nshl,1),
     &            yl(npro,nshl,ndofl),        xl(npro,nenl,nsd)
        real*8  bmtmp, xltmp(3)
        real*8  bmmax, Mrk_dist, Bub_radi
        real*8  deltS           !the thickness of liquid shell in the
                                !unit of epsilon
        integer:: imrk

        deltS = 3.0d0
!**********************************************************************
        IF(lstep.eq.ts_hold) THEN
!c... localization of marker field
        call local (banma,  bml,     ien,     1,      'gather  ')
        do i = 1,npro
           bmmax = 0.0d0
           do n = 1,nshl
              if(bml(i,n,1).gt.bmmax) bmmax = bml(i,n,1)
           enddo
           do n = 1,nshl
!c... if the nodal point is in the liquid
              if(yl(i,n,6).gt.0.0d0) then
                bml(i,n,1) = 0.0d0
!c... for the nodal points in the gas and near interface region...
              else
                bml(i,n,1) = bmmax
              endif
           enddo
        enddo
!c.... assemble the marker field
        call local (banma,    bml,   ien,    1,  'globaliz')
!**********************************************************************
        ELSEIF(lstep.ne.ts_hold.and.iClrLiq.eq.1) THEN
!       Do not use this part in the very first timestep because the
!       bub_cent is needed
!       Localization of marker field
        call local (banma,  bml,     ien,     1,      'gather  ')
!       clean up ID field outside the bubble core region
```

```
        do i = 1,npro
           do n = 1,nshl
              if(yl(i,n,6).gt.0.0d0) bml(i,n,1) = 0.0d0
           enddo
        enddo

        do i = 1,npro
           bmmax = 0.0d0
           xltmp = zero
           do n = 1,nshl
              if(bml(i,n,1).gt.bmmax) bmmax = bml(i,n,1)
           enddo
!       Update the maker field differently in bubble elments and near
!       interface shell
           do n = 1,nshl
!       If the node is in the bubble region
              if(yl(i,n,6).lt.0.0d0) then
                bml(i,n,1) = bmmax
!       If the node is in the near interface liquid shell
              elseif(yl(i,n,6).lt.deltS*epsilonBT.and.
     &                 yl(i,n,6).gt.epsilonBT) then
                xltmp(1:3) = xl(i,n,1:3)
                call ReColor(xltmp,bmtmp)
                bml(i,n,1) = bmtmp

              endif

           enddo        !nshl
        enddo           !npro

c.... assemble the marker field
        call local (banma,    bml,   ien,    1,  'globaliz')
!***********************************************************************
        ELSEIF(lstep.ne.ts_hold.and.iClrLiq.eq.0) THEN
        call local (banma,  bml,     ien,     1,      'gather  ')
        do i = 1,npro
           bmmax = 0.0d0
           xltmp = zero
           do n = 1,nshl
              if(bml(i,n,1).gt.bmmax) bmmax = bml(i,n,1)
           enddo
!       Update the maker field differently in bubble elments and near
!       interface shell
           do n = 1,nshl
!       If the node is in the liquid (eg. levelset > epsilon)
              if(yl(i,n,6).gt.0.0d0) then
                bml(i,n,1) = 0.0d0
!       If the node is in the bubble region
              else
                bml(i,n,1) = bmmax
              endif

           enddo        !nshl
        enddo           !npro

c.... assemble the marker field
        call local (banma,    bml,   ien,    1,  'globaliz')

        ENDIF   !ts_hold
        end     !banmaUpdate ends
```

Appendix E

Here we present the subroutines used in bubble break-up tracking capability.

Case 1: this subroutine is to detect suspicious bubble break-up events and initiate the whole break-up tracking process.

```
        subroutine breakupDetector(ibreakupFlag)
!----------------------------------------------------------------------
!       This subroutine is used to detect the suspicious bubble breakup
!       events
!       This subroutine is only called by master core
!       breakupSeeder(ib,1)      : current bubble ID
!       breakupSeeder(ib,2)      : action flag
!       breakupSeeder(ib,3)      : one parallel rank that where the
!                                  bubble has elements on
!       breakupSeeder(ib,4)      : total number of bubble point when
!                                  seeding
!       breakupSeeder(ib,5)      : volume of region occupied by auxiliary
!                                  ID
!       breakupSeeder(ib,6)      : auxiliary bubble ID if flag is not 0
!
!----------------------------------------------------------------------
        use bub_track   ! access to bubble information array
        include "common.h"

        integer ib, iPossibleBreakup
        integer iactionFlag, iauxBubID, MaxAuxID
        integer ibreakupFlag, iseeding


        ibreakupFlag    = 0
        iPossibleBreakup= 0
        MaxAuxID        = 0

        do ib = 1,i_num_bubbles
           if(MaxAuxID.lt.int(breakupSeeder(ib,6)))
     &        MaxAuxID =  int(breakupSeeder(ib,6))
        enddo

!        write(*,*) 'i_num_bubbles in detector =', i_num_bubbles

        do ib = 1,i_num_bubbles
!           write(*,*) 'breakupSeeder in detector 1= ',
!     &                 breakupSeeder(ib,:)
           iseeding      = 0
           breakupSeeder(ib,1) = real(ib)
           iactionFlag  = int(breakupSeeder(ib,2))
           if(iactionFlag.eq.0.and.avg_info(ib,4).gt.0.0d0) then
           iauxBubID    = int(breakupSeeder(ib,1))
           if(mod(lstep,100).eq.0) iseeding     = 1
!       more sophisticated condition can be developed to determine when
!       setting iseeding to be 1.
           if(iseeding.eq.1) then
!           if(lstep+1.eq.17403) then
!              write(*,*) 'ID seeding is switched on manually'
```

```
              iPossibleBreakup = iPossibleBreakup + 1
              iactionFlag      = 1
              iauxBubID        = 3*MaxAuxID + iPossibleBreakup
           endif

           breakupSeeder(ib,2) = real(iactionFlag)
           breakupSeeder(ib,3) = avg_info(ib,14)
           breakupSeeder(ib,6) = real(iauxBubID)

           endif !iactionFlag = 0
!           write(*,*) 'breakupSeeder in detector = ',
!     &                 breakupSeeder(ib,:)

        enddo  !ib


!       The ibreakupFlag is used to determine if we should call breakup
!       confirmer, basically if there is one non-zero breakup index, the
!       flag will be 1.
        do ib = 1, i_num_bubbles
           if(breakupSeeder(ib,2).gt.0.0d0) then
              ibreakupFlag = 1
              exit
           endif
        enddo

        if(ibreakupFlag.eq.0) then
             write(*,*) 'There is no suspicious break-up event!'
        else
             write(*,*) 'Suspicious break-up event is detected!'
        endif

        end !breakupDetector ends
```

Case 2: this subroutine tracks the evolution of seeded bubble IDs and confirm whether a suspicious break-up event has happened or not.

```
        subroutine breakupConfirmer(y, banma)
!----------------------------------------------------------------------
!       This subroutine is used to double check the suspicious bubble
!       breakup events
!       This subroutine is called by all the parallel cores
!----------------------------------------------------------------------
        use bub_track   ! access to bubble information array
        include "common.h"
        include "mpif.h"
        include "auxmpi.h"

        real*8    y(nshg,ndof)
        real*8    Ratio
        dimension banma(nshg,1)
        integer   i, ib, i_mrk
        integer   iactionFlag, iauxBubID, irank

        do ib = 1,i_num_bubbles
           iactionFlag  = int(breakupSeeder(ib,2))
           irank        = int(breakupSeeder(ib,3))
           iauxBubID    = int(breakupSeeder(ib,6))
           Ratio        = 0.0d0

           if(iactionFlag.eq.1) then
              iactionFlag = 2

              if(myrank.ne.master .and. myrank.eq.irank) then
                 do i = 1, nshg
                 i_mrk = int(banma(i,1))
                 if(y(i,6).lt.0.0d0 .and. i_mrk.eq.ib) then
                 banma(i,1) = real(iauxBubID)
                 if(iactionFlag.eq.2) exit

                 endif
                 enddo
              endif

           elseif(iactionFlag.ge.2.and.iactionFlag.le.14) then
                  iactionFlag = iactionFlag + 1

           elseif(iactionFlag.eq.15) then
!       Confirm the exitance of new bubble and update the total number
!       of bubbles
               iactionFlag = 0
               if(breakupSeeder(ib,4).gt.0.0d0)
     &            Ratio = breakupSeeder(ib,5)/breakupSeeder(ib,4)
!               if(myrank.eq.master)write(*,*)'Ratio =',ib, Ratio
               if(Ratio.lt.0.1d0 .or. Ratio.gt.0.9d0 ) then
!       case 1: fake trigger and no breakup is happening
                  do i = 1, nshg
                     if(int(banma(i,1)).eq.iauxBubID)
     &               banma(i,1) = real(ib)
                  enddo
               else
!       case 2: breakup event is captured and recorded
                  i_num_bubbles = i_num_bubbles + 1
                  do i = 1, nshg
                     if(int(banma(i,1)).eq.iauxBubID)
     &               banma(i,1) = real(i_num_bubbles)
                  enddo
                  if(myrank.eq.master)
     &            write(*,'(A,I5,A)')' New Bubble',i_num_bubbles,
     &                                   ' is recognized!'
               endif

               breakupSeeder(ib,6) = breakupSeeder(ib,1)
           endif

           breakupSeeder(ib,2)   = real(iactionFlag)
        enddo

!        if(myrank.eq.master) then
!        write(*,*) 'breakupSeeder in confirmor = ',
!     &                  breakupSeeder(:,:)
!        endif

        end !breakupConfirmer ends
```

Appendix F

Here we present selected algorithms used in data extraction process.

Case 1: bubble information collection at the mesh block level.

```fortran
        subroutine BubCollect(u1,       u2,     u3,     Sclr, dist2w,
     &                        xx,       yl,     bml,    elemvol_local,
     &                        rho,      CurvInfo)
!----------------------------------------------------------------------
!       Called in e3ivar.f
!       This subroutine is dealing with the bubble information
!       collection at the very bottom level.
!       In bub_info(i_num_bubbls, 17)
!       bubble-wise: x,y,z coord, vel, elem vol, mass, levelset;
!       local liq  : x vel, y vel, z vel, d2wall;
!       BT field   : marker(bubble ID)
!       Interface curvature is weighted by volume of the interface
!       element.
!----------------------------------------------------------------------
        use  bub_track
        include "common.h"

        dimension u1(npro),     u2(npro),  u3(npro)
        dimension dist2w(npro), rho(npro), CurvInfo(npro)
        dimension yl(npro,nshl,ndof),      xx(npro,nsd),
     &            Sclr(npro)
        dimension bml(npro,nshl,1)
        real*8    elemvol_local(ibksiz)
        real*8    bmmax
        real*8    denswght

        rholiq=datmat(1,1,1)
        rhogas=datmat(1,1,2)

        bub_info = zero

        do i = 1, npro
!... collect LS value & markers for bubble and liquid shell
           if(Sclr(i) .le. 3.0d0*epsilonBT) then
              if(rholiq.eq.rhogas) then
                 denswght = 1.0d0
              else
                 denswght  =(rholiq-rho(i))/(rholiq-rhogas)
              endif
              bmmax     = 0.0
              do n = 1, nshl
                if(bml(i,n,1).gt.bmmax) bmmax = bml(i,n,1)
              enddo
              bub_info(i,11) = bmmax
!       Find out the minimum level set value
              do n = 1, nshl
                if(yl(i,n,6).lt.bub_info(i,10))
     &             bub_info(i,10) = yl(i,n,6)
              enddo
!       collect the local liquid velocity and the y coord of liquid
!       shell elments around the bubble
```

```fortran
              if(Sclr(i).gt.epsilonBT) then
                bub_info(i,6)  = u1(i)
                bub_info(i,7)  = u2(i)
                bub_info(i,8)  = u3(i)
                bub_info(i,9)  = dist2w(i)
                bub_info(i,17) = elemvol_local(i)
              elseif(Sclr(i).gt.0.0d0) then
                bub_info(i,5)  = elemvol_local(i)*denswght*rhogas
                bub_info(i,15) = CurvInfo(i)
              elseif(Sclr(i).gt.-epsilonBT) then
                bub_info(i,1)  = xx(i,1)
                bub_info(i,2)  = xx(i,2)
                bub_info(i,3)  = xx(i,3)
                bub_info(i,4)  = elemvol_local(i)
                bub_info(i,5)  = elemvol_local(i)*denswght*rhogas
                bub_info(i,12) = u1(i)*elemvol_local(i)
                bub_info(i,13) = u2(i)*elemvol_local(i)
                bub_info(i,14) = u3(i)*elemvol_local(i)
                bub_info(i,15) = CurvInfo(i)
              else
!... collect the bubble information in details
                bub_info(i,1)  = xx(i,1)
                bub_info(i,2)  = xx(i,2)
                bub_info(i,3)  = xx(i,3)
                bub_info(i,4)  = elemvol_local(i)
                bub_info(i,5)  = elemvol_local(i)*denswght*rhogas
                bub_info(i,12) = u1(i)*elemvol_local(i)
                bub_info(i,13) = u2(i)*elemvol_local(i)
                bub_info(i,14) = u3(i)*elemvol_local(i)
                bub_info(i,16) = real(myrank)
              endif ! Sclr(i)
           endif !Sclr(i) for the bubble region and liquid shell

        enddo !npro

        end     !BubCollect ends
```

Case 2: bubble information assembling at the mesh partition/compute processor level.

```fortran
        subroutine BubASSY()
!----------------------------------------------------------------------
!       Called in elmgmr.f
!       This subroutine is used to assembly bubbles' information after
!       the loop over blocks on each processor
!
!----------------------------------------------------------------------
        use bub_track   ! access to bubble information array
        include "common.h"
        include "mpif.h"
        include "auxmpi.h"

        integer i_mrk, ib

        do i = 1, npro
           i_mrk = INT(bub_info(i,11))
```

```fortran
!       The marker re-arrangement when auxillary index is seeded during
!       bubble breakup recognition
           if(iBK.eq.1 .and. i_mrk.gt.i_num_bubbles) then
              do ib = 1,i_num_bubbles
                 if(i_mrk.eq.int(breakupSeeder(ib,6)))then
                    i_mrk = int(breakupSeeder(ib,1))
!       j=15: phasic volume occupied by auxiliary ID
                    procs_dataset(i_mrk,15) = procs_dataset(i_mrk,15)
     &                                      + bub_info(i,4)
                    exit
                 endif
              enddo
           endif
!       Here starts the extraction part
           if (i_mrk .gt. 0) then
!       j=1-3: x coord, y coord, z coord (weighted by element volume)
              do j = 1, 3
              procs_dataset(i_mrk,j) = procs_dataset(i_mrk,j)
     &                               + bub_info(i,j)*bub_info(i,4)
              enddo
!       j=4-5: bubble element volume, gas mass
              do j = 4, 5
              procs_dataset(i_mrk,j) = procs_dataset(i_mrk,j)
     &                               + bub_info(i,j)
              enddo
!       j=6-8: local liquid velocity in x1, x2, x3 directions
              do j = 6, 8
              procs_dataset(i_mrk,j) = procs_dataset(i_mrk,j)
     &                               + bub_info(i,j)*bub_info(i,17)
              enddo
!       j=9: minimum level-set value in bubble region
              if(bub_info(i,10).lt.procs_dataset(i_mrk,9))
     &        procs_dataset(i_mrk,9) = bub_info(i,10)
!       j=10: the highest process rank the bubble elements exist
              if(bub_info(i,16).gt.procs_dataset(i_mrk,10))
     &           procs_dataset(i_mrk,10) = bub_info(i,16)
!       j=11: summation of element volume in local liquid shell
              procs_dataset(i_mrk,11) = procs_dataset(i_mrk,11)
     &                                + bub_info(i,17)
!       j=12-14: bubble velocity in x1, x2, x3 directions
              do j = 12, 14
              procs_dataset(i_mrk,j) = procs_dataset(i_mrk,j)
     &                               + bub_info(i,j)
              enddo
!       j=16: element count in bubble region
              if(bub_info(i,4).gt.0.0d0)
     &        procs_dataset(i_mrk,16) = procs_dataset(i_mrk,16) + 1.0d0

!       Find out the upper and lower bounds of element coordinates
!       inside each bubble
              IF(bub_info(i,4).ne.0.0d0) THEN   !Inside bubble
              if(bub_info(i,1).lt.procs_coordDn(i_mrk,1))
     &           procs_coordDn(i_mrk,1) = bub_info(i,1)
              if(bub_info(i,2).lt.procs_coordDn(i_mrk,2))
     &           procs_coordDn(i_mrk,2) = bub_info(i,2)
              if(bub_info(i,3).lt.procs_coordDn(i_mrk,3))
     &           procs_coordDn(i_mrk,3) = bub_info(i,3)
```

```

              if(bub_info(i,1).gt.procs_coordUp(i_mrk,1))
     &           procs_coordUp(i_mrk,1) = bub_info(i,1)
              if(bub_info(i,2).gt.procs_coordUp(i_mrk,2))
     &           procs_coordUp(i_mrk,2) = bub_info(i,2)
              if(bub_info(i,3).gt.procs_coordUp(i_mrk,3))
     &           procs_coordUp(i_mrk,3) = bub_info(i,3)
              ENDIF

!       Find out the points with min and max d2wall and the associated
!       velocities for each bubble
              if(bub_info(i,6).ne.0.0d0) then
                 if(bub_info(i,9).lt.Shear_NodeMin(i_mrk,1)) then
                    Shear_NodeMin(i_mrk,1) = bub_info(i,9)      !min d2w
                    Shear_NodeMin(i_mrk,2) = bub_info(i,6)      !x vel
                 endif
                 if(bub_info(i,9).gt.Shear_NodeMax(i_mrk,1)) then
                    Shear_NodeMax(i_mrk,1) = bub_info(i,9)      !max d2w
                    Shear_NodeMax(i_mrk,2) = bub_info(i,6)      !x vel
                 endif
              endif

           endif !i_mrk
        enddo !npro

        end     !BubASSY = bubble assembly
```

Case 3: bubble information processing at the entire domain level

```
        subroutine BubMPIprocess()
!-----------------------------------------------------------------------
!       Called in elmgmr.f
!       This subroutine is dealing with the MPI processing of bubble
!       information from different processors
!
!-----------------------------------------------------------------------
        use bub_track   ! access to bubble information array
        include "common.h"
        include "mpif.h"
        include "auxmpi.h"

        real*8 one_procs,       all_procs
        real*8 phi_tmp,         phi_min,        phi_max
        real*8 eq_rad
        real*8 Shear_dummy01(2), Shear_dummy02(2), Shear_dummy03
        real*8 diffinY

        allocate ( unive_dataset(i_num_bubbles, 16) )
        allocate ( unive_coordDn(i_num_bubbles, 3)  )
        allocate ( unive_coordUp(i_num_bubbles, 3)  )
        allocate ( bubbl_coordDf(i_num_bubbles, 3)  )
        allocate ( Shear(i_num_bubbles,4)           )

        unive_dataset   = zero
        unive_coordDn   = zero
        unive_coordUp   = zero
        bubbl_coordDf   = zero
        Shear           = zero

        if (numpe .gt. 1) then
           do i = 1, i_num_bubbles
              do j = 1, 16
                 if (j.eq.9) then
!       Find out the minimum value for each bubble
                 phi_tmp = procs_dataset(i,j)
                 call MPI_ALLREDUCE (phi_tmp, phi_min, 1,
     &                MPI_DOUBLE_PRECISION, MPI_MIN,
     &                MPI_COMM_WORLD, ierr)
                 unive_dataset(i,j) = phi_min
                 elseif(j.eq.10) then   !find max.
                 phi_tmp = procs_dataset(i,j)
                 call MPI_ALLREDUCE (phi_tmp, phi_max, 1,
     &                MPI_DOUBLE_PRECISION, MPI_MAX,
     &                MPI_COMM_WORLD, ierr)
                 unive_dataset(i,j) = phi_max
                 else
                 one_procs = procs_dataset(i,j)
                 call MPI_ALLREDUCE (one_procs, all_procs, 1,
     &                MPI_DOUBLE_PRECISION,MPI_SUM, MPI_COMM_WORLD,ierr)
                 unive_dataset(i,j) = all_procs
                 endif
              enddo !index j
!------------------------------------------------------------------------
!       Here, we determine the x bounds, y bounds, z bounds for each
!       bubble
!------------------------------------------------------------------------
              do j = 1, 3
                 one_procs = procs_coordDn(i,j)
                 call MPI_ALLREDUCE (one_procs, all_procs, 1,
     &                MPI_DOUBLE_PRECISION, MPI_MIN,
     &                MPI_COMM_WORLD, ierr)
                 unive_coordDn(i,j) = all_procs

                 one_procs = procs_coordUp(i,j)
                 call MPI_ALLREDUCE (one_procs, all_procs, 1,
     &                MPI_DOUBLE_PRECISION, MPI_MAX,
     &                MPI_COMM_WORLD, ierr)
                 unive_coordUp(i,j) = all_procs
!       This range of elments coordinates inside a bubble will be used
!       is crossing the periodic planes
                 bubbl_coordDf(i,j) = unive_coordUp(i,j) -
     &                                unive_coordDn(i,j)
              enddo
!------------------------------------------------------------------------
!       Find the two points to calculate local shear
!------------------------------------------------------------------------
!       The one with minimum d2wall
              Shear_dummy01(1) = Shear_NodeMin(i,1)
              Shear_dummy01(2) = myrank  ! myrank is to be coerced to a real*8

              call MPI_ALLREDUCE (Shear_dummy01, Shear_dummy02, 1,
     &                MPI_2DOUBLE_PRECISION, MPI_MINLOC,
     &                MPI_COMM_WORLD, ierr)

              Shear(i,1) = Shear_dummy02(1)
              Shear_dummy03 = Shear_NodeMin(i,2)
!       Broadcast the x vel (for local liq) associated with min d2wall
              call MPI_Bcast(Shear_dummy03,1,MPI_DOUBLE_PRECISION,
     &                       INT(Shear_dummy02(2)),MPI_COMM_WORLD,ierr)
              Shear(i,2) = Shear_dummy03
!       The one with maximum d2wall
              Shear_dummy01(1) = Shear_NodeMax(i,1)
              Shear_dummy01(2) = myrank  ! myrank is coerced to a real*8

              call MPI_ALLREDUCE (Shear_dummy01, Shear_dummy02, 1,
     &                MPI_2DOUBLE_PRECISION, MPI_MAXLOC,
     &                MPI_COMM_WORLD, ierr)

              Shear(i,3) = Shear_dummy02(1)
              Shear_dummy03 = Shear_NodeMax(i,2)
!       Broadcast the x vel (for local liq) associated with max d2wall
              call MPI_Bcast(Shear_dummy03,1,MPI_DOUBLE_PRECISION,
     &                       INT(Shear_dummy02(2)),MPI_COMM_WORLD,ierr)
              Shear(i,4) = Shear_dummy03

           enddo !i_num_bubbles
!------------------------------------------------------------------------
!       Calculate the average properties
!------------------------------------------------------------------------

        avg_info = zero
        if(myrank.eq.master) then

        do i = 1, i_num_bubbles
!       The gas entity smaller than 5x5x5 is not considered as
!       a valid bubble
           if(unive_dataset(i,16).lt.125) cycle
!       The average coordinates for each bubble
           if(unive_dataset(i,4).gt.0.0d0) then
           avg_info(i,1:3)  = unive_dataset(i,1:3)  /unive_dataset(i,4)
           avg_info(i,11:13)= unive_dataset(i,12:14)/unive_dataset(i,4)
           endif
           avg_info(i,14)   = unive_dataset(i,10)

           eq_rad = (3.0*unive_dataset(i,4)/(4.0*pi))**(1.0/3.0)
           avg_info(i,4)    = eq_rad
!       The bubble mass
           avg_info(i,5)    = unive_dataset(i,5)

!       The deformability factor is defined by the ratio b/w min level
!       set value with equivalent radius, which is 1.0 for spherical
!       bubbles ideally.
           if(eq_rad.ne.0.0d0)
     &     avg_info(i,6)    = abs(unive_dataset(i,9))/eq_rad

!       The local liquid velocity components near the bubble
           if(unive_dataset(i,11).gt.0.0d0)
     &     avg_info(i,7:9)  = unive_dataset(i,6:8)/unive_dataset(i,11)

!       The local liquid shear around the bubble
           diffinY = Shear(i,3) - Shear(i,1)
           if(diffinY.ne.0.0d0)
     &     avg_info(i,10)   = (Shear(i,4) - Shear(i,2))/diffinY

           if(iBK.eq.1) then
           breakupSeeder(i,4) = unive_dataset(i,4)
           breakupSeeder(i,5) = unive_dataset(i,15)
           endif
!       Adjust the bubble centers coordinates for those crossing
!       periodic planes (1% tolerance is allowed)
           if(abs(bubbl_coordDf(i,1)-XLEN).lt.0.01d0*XLEN) then
```

```
              if(avg_info(i,1) .ge. Xmid ) then
                 avg_info(i,1) = DomainSize(2) - eq_rad *
     &          (avg_info(i,1) - Xmid)/(0.5d0*XLEN-eq_rad)
              else
                 avg_info(i,1) = DomainSize(1) + eq_rad *
     &          (Xmid - avg_info(i,1))/(0.5d0*XLEN-eq_rad)
              endif
           endif

           if(abs(bubbl_coordDf(i,2)-YLEN).lt.0.01d0*YLEN) then
              if(avg_info(i,2) .ge. Ymid ) then
                 avg_info(i,2) = DomainSize(4) - eq_rad *
     &          (avg_info(i,2) - Ymid)/(0.5d0*YLEN-eq_rad)
              else
                 avg_info(i,2) = DomainSize(3) + eq_rad *
     &          (Ymid - avg_info(i,2))/(0.5d0*YLEN-eq_rad)
              endif
           endif

           if(abs(bubbl_coordDf(i,3)-ZLEN).lt.0.01d0*ZLEN) then
              if(avg_info(i,3) .ge. Zmid ) then
                 avg_info(i,3) = DomainSize(6) - eq_rad *
     &          (avg_info(i,3) - Zmid)/(0.5d0*ZLEN-eq_rad)
              else
                 avg_info(i,3) = DomainSize(5) + eq_rad *
     &          (Zmid - avg_info(i,3))/(0.5d0*ZLEN-eq_rad)
              endif
           endif

!        write(*,*)'breakupSeeder in bubMPI =', breakupSeeder(i,:)
        enddo  !i_num_bubbles
!-------------------------------------------------------------------------

        endif !myrank

        else
           write(*,*) 'Bubble tracking was not coded for 1-procs case!'
        endif

        deallocate ( unive_dataset )
        deallocate ( unive_coordDn )
        deallocate ( unive_coordUp )
        deallocate ( bubbl_coordDf )
        deallocate ( Shear         )


        end     !BubMPIprocess ends
```