ORNL/TM-2021/2318

Simulation of Transport Phenomena in Bubble Column Reactors

Ramanan Sankaran
Vimal Ramanuj
Luka Malenica
Leonardo Spanu
Guoqiang Yang

September 2022

CRADA final report for
CRADA number NFE-19-07532

Approved for public release.
Distribution is unlimited.

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2021/2318

High Performance Computing for Materials (HPC4Mtls)

Simulation of Transport Phenomena in Bubble Column Reactors

Ramanan Sankaran, Vimal Ramanuj, and Luka Malenica
Oak Ridge National Laboratory

Leonardo Spanu and Guoqiang Yang
Shell International Exploration and Production

Date Published: Sep 2022

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-Battelle, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This research was supported by the High-Performance Computing for Materials Program (HPC4Mtls), managed by the U.S. Department of Energy, Hydrogen and Fuel Cell Technologies Office (HFTO) within the Energy Efficiency and Renewable Energy (EERE) Office. This research used resources of the Oak Ridge Leadership Computing Facility (OLCF), which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

ABSTRACT

The production of carbon and hydrogen from methane is a viable pathway for monetizing the vast reserves of US natural gas. Liquid (metals and/or salts) bubble column reactors operating at temperatures greater than $1000^{\circ}C$ allow the conversion of methane into hydrogen without CO_2 emission. The low-density carbon floats on top of the liquid surface, allowing an easier removal, compared to other reactor concepts for methane pyrolysis. In this current report, we present results from simulations of transport phenomena in a liquid bubble column reactor. The goal is to investigate and understand parameters that are crucial for bubble column reactor design and scale up. The report presents the formulation and numerical methods used in the multiphase direct numerical simulation software, Quilt. This software was validated for the bubbly flow using numerical and physics based test problems. Quilt was then used to perform simulations of the bubble column reactor as a parametric study, varying the bubble injection parameters. The simulations were visualized for qualitative analysis and followed up with a quantitative analysis. The effect of injection parameters on various quantitative measures of the bubble residence time, interfacial area and motion through the column are presented.

1. Introduction

The proven world gas reserves show an increase in the gas reserves, while environmental concerns associated with the carbon dioxide emissions are also growing. Therefore, a large incentive exists to develop technology for converting natural gas into lower emission fuels and chemicals. The production of carbon and hydrogen from methane is a viable pathway for monetizing the vast reserves of US natural gas. Hydrogen is a central molecule for the chemical and refinery industry and has the potential to play a role as an energy carrier for future energy systems and mobility. The United States produces approximately ten million metric tons of hydrogen annually and more than 95% of the hydrogen is produced by steam-methane reforming (SMR) of natural gas [1]. SMR is the state of the art for the production of hydrogen from natural gas and is a highly endothermic process that requires temperatures of $750°C$ to $900°C$ and pressure above 30 bar on a Ni-based catalysts [2]. SMR is attractive because of its high hydrogen yield, but it also converts the carbon to carbon dioxide (CO_2). Methane pyrolysis, the non-oxidative thermal decomposition of methane to carbon and hydrogen, is an alternative to SMR and produces CO_2-free hydrogen. Solid carbon, with the appropriate morphology, could be utilized in materials for numerous applications, ranging from rubbers, lightweight composites, to soil enhancer. The produced carbon can be sold as a co-product, thus providing an economic credit that reduces the delivered net cost of hydrogen.

Liquid (metals and/or salts) bubble column reactors [3] operating at T>$1000°C$ allows the conversion of methane into hydrogen without CO_2 emission. The low-density carbon floats on top of the liquid surface, allowing an easier removal, compared to other reactor concepts for methane pyrolysis. In-situ characterization of a bubble molten media (salt or metal) reactor is extremely challenging, due to the high temperature involved and the corrosive environment. Key chemical reaction engineering parameters such as molecular diffusion, dispersion and flow hydrodynamics are extremely challenging, and virtually impossible, to measure with experimental techniques (e.g. magnetic resonance imaging). A very limited amount of data is available on the molten media reactor from open literature and even those available are mostly covering methane conversion and kinetics parameters.

Numerical simulations remain therefore the ideal tool for studying flow, heat, and transport properties within the molten phase reactor at high temperatures. While both molten metals and salts have been used in the pyrolysis columns, molten salts (e.g. chlorides) have cost and energy advantages compared to molten metals. The simulation of a molten salts bubble column is an intrinsic multiscale and multiphase problem, posing numerous challenges even without the implicit inclusion of chemical reactions. Gas (injected methane and produced hydrogen) bubbles through the liquid in the presence of a carbon solid phase that eventually moves to the top of the liquid surface, with several time scales involved: bubble residence time, methane reactions in the bubble, carbon diffusion, etc. Understanding how carbon particles of different sizes distribute across the reactor and collect at the free surface is crucial information for designing possible salt-carbon separation schemes. Unfortunately, the partial data on carbon distribution could be gathered only ex-situ, after the reactor has stopped to operate and the salts have re-solidified. Such an ex-situ approach is costly, time consuming, and does not really allow a detailed understanding of transport and floating mechanisms. On the other hand, efforts to build a reactor with transparent materials to monitor carbon flotation are challenging due to materials compatibility with the salts. Simulations remain the faster approach to study carbon transport and its accumulation at the top of the liquid interface.

The optimization of a bubble column reactor for the pyrolysis of natural gas to carbon and hydrogen needs

to consider several key factors, such as

- *Gas residence time.* A higher residence time of the gas within the liquid column is necessary to increase the conversion rate of the feed gas, considering the kinetics of the pyrolysis process.

- *Gas holdup and interfacial area.* A larger gas holdup is essential to achieve a good volumetric effectiveness within the column. However, a large holdup can also lead to bubble coalescence and the formation of slugs that destroys the gas-liquid interfacial surface area, thereby reducing the conversion. Therefore, the holdup needs to be retained within the bubbly flow regime.

- *Circulation of liquid.* The superficial velocity of the liquid phase and the net circulation in the column affects the reactor efficiency. The azimuthal component of the velocity in the column induces a corresponding motion in the gas bubble and increases the effective residence time of the bubble. However, the vertical circulation will transport the solid carbon from the surface of the column towards the depths of the column, saturate the liquid with carbon, and adversely affect the equilibrium of the pyrolysis reaction.

- *Bubble and particle dynamics at the surface.* The gas bubble is covered by a liquid film as it leaves the free surface and collapses. The dynamics of bubble evolution at the free surface and extent of splashing determine the amount of salt that is deposited on the carbon surface as a contaminant. The ability to limit the bubble size and momentum as it breaks through the free surface is important in minimizing contamination and loss of thermal energy.

Three dimensional simulations enabled by high performance computing (HPC) are able to provide the insights on these key transport factors that allow the optimization of reactor designs and operating parameters.

The numerical simulation of bubble column in a molten medium is a multi-physics problem, involving computational fluid dynamics (CFD) and chemical kinetics. The accuracy and predictivity of simulated results is strongly dependent on the choice of the numerical models and their fidelity in representing the turbulent flow characteristics and bubble dynamics. Lower fidelity approaches that do not account for bubble coalescence, breakup or the effect of turbulent flow on the bubble motion are not sufficiently predictive [4]. The effort spent in improving these models using air-water column experiments do not necessarily improve their predictivity of the systems involving complex flow physics and reactivity, where the experimental data are scarce. Interface resolved direct numerical simulation (DNS) will fully resolve the turbulent flow in the column and the bubble dynamics including deformation, breakup and coalescence. It will provide the predictive models needed to study the various reactor geometries, and gas injection parameters. ORNL has developed a DNS solver for reacting multiphase flows, called Quilt. Quilt is a structured mesh incompressible flow solver for DNS of complex flows, scalable to the graphics processing unit (GPU) accelerated HPC systems, such as Summit. Quilt uses an immersed boundary method for the representation of complex geometries on a structured mesh without requiring time consuming body fitted mesh generation. Quilt is coupled to Cantera, a chemical kinetics library, for the evaluation of thermochemical and transport properties, in addition to gas phase and interfacial reaction kinetics. Here, we apply Quilt for the simulation of the multiphase flow in the bubble column reactor.

2. Impact

The United States produces approximately ten million metric tons of hydrogen annually and more than 95% of the hydrogen is produced by SMR of natural gas [1], followed by water-gas shift reaction. High capital costs and high energy consumptions characterize the SMR technologies. In addition, compared to SMR technologies, the pyrolytic route allows hydrogen production with virtually no CO_2 emission, while producing high purity carbon. With the proper control of the morphology (via a combination of reactor design and catalyst selections, etc.) the carbon could be used for manufacturing composite lightweight materials, steel, and even for soil remediation. Technoeconomic analysis have shown that molten reactors could produce H2 at a cost competitive against SMR. The absence of oxygenated contaminants makes the hydrogen manufactured via pyrolysis suitable for fuel cell application, usually sensitive to even very low CO levels. The successful deployment of methane cracking technology based on molten metal reactors would have a large impact in numerous sectors, ranging from chemical industry, refinery, and mobility. Given the construction cost and material compatibility issues of molten metal reactors, simulations are critical to reduce the developing cost and time. With the inclusion of the right physics and chemistry, a numerical model could help down-select the number of specific geometries/configurations, and identify possible bottle necks in the scaling up phase. The simulation results will guide the design of an experimental pyrolysis reactor for continuum operations with carbon removal. A similar reactor is essential to collect data on the process to derisk future investments on a larger demonstration unit.

3. Formulation and Numerical Method

In this section, the formulation and numerical methods used for the solution of the time-dependent Navier-Stokes equations is given. The one-fluid formulation is used, where a single set of equations is solved for all fluid phases, allowing for the material properties, such as density and viscosity, to change discontinuously. This approach allows usage of methods developed for single-phase flows; however, the different fluids (i.e. their interfaces) needs to be calculated as part of the solution. In this approach, the calculation of the surface tension on the interface is critical to accuracy, as will be explained later in this section.

The one fluid formulation of the Navier-Stokes equations is given by:

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho\nabla\cdot(\mathbf{vv}) = -\nabla p + \nabla\cdot\left\{\mu[\nabla\mathbf{v} + (\nabla\mathbf{v})^T]\right\} + \mathbf{f} \tag{1}$$

$$\nabla\cdot\mathbf{v} = 0 \tag{2}$$

where the non-conservative form of the momentum equation (1) is usually preferred for multiphase flow simulations [5]. In Eq. 1, ρ is the density, $\mathbf{v}$ is the velocity vector, p is the pressure, μ is the dynamic viscosity, and $\mathbf{f}$ is the body force. There are different possibilities for the solution of the time-dependent Navier-Stokes equations. In the following we described two approaches implemented in Quilt. The first one is based on fractional-step method and uses fully explicit time integration, while the second one is based on fully implicit SIMPLE algorithm. Both approaches are well established for single flow problems. As already mentioned, in addition to solution of one-fluid equations, the modeling of multiphase flows problems requires solutions for interface motion and surface tension calculation. The level-set approach is

used for modeling of interface motion, while surface tension is included via continuum surface force (CSF) model or pressure boundary method (PBM), as will be described in Section 3.3.

3.1 Unsteady Flow Solver

The equation (1) is rewritten symbolically as:

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho\mathbf{C} = -\nabla p + \mathbf{D} + \mathbf{f} \tag{3}$$

where $\mathbf{C} = \nabla \cdot (\mathbf{vv})$ is convection term, $\mathbf{D} = \nabla \cdot \left\{\mu[\nabla \mathbf{v} + (\nabla \mathbf{v})^T]\right\}$ is diffusion term, and $\mathbf{f}$ denotes all other force terms, such as gravity and surface tension.

3.1.1 Explicit Fractional-Step Method (EFSM)

The fractional step method (usually referred to as projection method [6]) is one of the most common solution approaches for the unsteady incompressible flow simulations. The main idea is to first calculate temporary velocity field by solving momentum equation (usually by ignoring pressure gradient). Then, in the second step, the velocity field is corrected by adding appropriate pressure gradient to give a divergence-free velocity field at the end of the time step. No iterations are performed between momentum and pressure equations.

There are many variants of fractional step methods. The most notable differences among them are related to time discretization and treatment of the pressure term in the momentum equation. The convective term is usually treated explicitly to avoid non-linearity introduced by implicit discretization. However, the diffusion is treated implicitly to avoid time step restrictions which can be very stringent for explicit schemes, especially when the mesh needs to be locally refined. Considering the pressure term in the momentum equation, there are three most common used approaches [7]. The first one, denoted as P1, sets the pressure term to be zero in the momentum equation during predictor step. The second approach P2 sets the pressure equal to the value from the previous time step, and the pressure Poisson equation is then solved to find a pressure correction; while in the P3 approach, the pressure is extrapolated using previous time-step values. While these approaches differ accuracy, all of them still introduce splitting error. This error generally can be eliminated only by using some kind of iterative procedure (similar to the SIMPLE algorithm described in the next section).

The algorithm presented here is fully explicit and uses first order forward Euler time integration. The momentum equation (3) is split into two parts. The first one, denoted as predictor, ignores the pressure term (P1 approach) and calculates temporary velocity field ($\mathbf{v}^*$) by:

$$\frac{\mathbf{v}^* - \mathbf{v}^\mathbf{n}}{\Delta t} = -\mathbf{C}^\mathbf{n} + \frac{1}{\rho^n}\left(\mathbf{D}^\mathbf{n} + \mathbf{f}^\mathbf{n}\right) \tag{4}$$

In the second part, the projection (or corrector) step, the pressure gradient is used to calculate final velocity at the end of the time step:

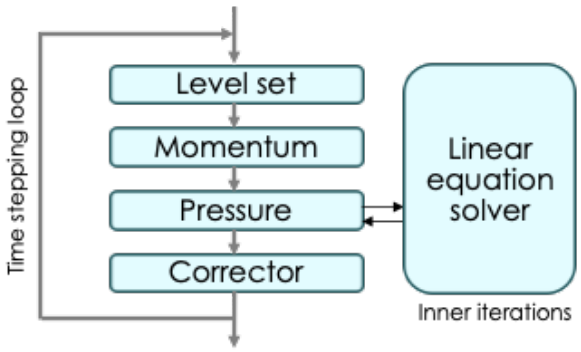


Figure 1. Flow chart of fractional-step method.

$$\frac{\mathbf{v^{n+1}} - \mathbf{v}^*}{\Delta t} = -\frac{1}{\rho^n}\left(\nabla p\right) \tag{5}$$

The pressure field is calculated by expressing $\mathbf{v^{n+1}}$ from equation (5) and combining with continuity constraint:

$$\nabla \cdot \mathbf{v^{n+1}} = 0 \tag{6}$$

obtaining final Poisson-type equation for pressure:

$$\nabla \cdot \left(\frac{1}{\rho^n}\nabla p\right) = \frac{1}{\Delta t} \cdot \mathbf{v}^* \tag{7}$$

The spatial discretization for the equations (4), (5), and (7) can be any of the standard methods. In this work, the conservative finite difference method on the the staggered grid arrangement (Section 3.1.3) is used. Further details on EFSM can be found in [5].

The flow chart of the EFSM algorithm is show in Figure 1. Usually, the level set field is staggered in time and advected from $t^{n-\frac{1}{2}}$ to $t^{n+\frac{1}{2}}$. Then using fluid properties defined at time $t^{n+\frac{1}{2}}$, the new flow field (at t^{n+1}) is calculated using described procedure.

During simulation runs it was recognized that there are two main computational limitations for the large scale simulations when using EFSM. The first one was solution of the linear system of the equations arising from pressure Poisson equation (7), while the second was time step restriction due to fully explicit temporal discretization. Due to the former limitation, the geometric multigrid solver (Section 3.2) was implemented in Quilt. In the next section, a fully implicit algorithm is presented to ease the time step size limitation.

3.1.2 Semi-Implicit Method for Pressure Linked Equations (SIMPLE)

The implicit methods are preferred for steady or slow-transient problems because much larger time steps can be used. This is particularly important when fine spatial discretization is needed and where explicit methods would require much smaller time steps then is actually needed for accuracy [8]. In the following,

the main steps of the finite volume (FV) discretization using co-located grid (Section 3.1.3) are described, while the more detailed description of the SIMPLE-type algorithms can be found in classical CFD textbooks, e.g. [8, 9].

The SIMPLE algorithm is fully implicit approach, meaning that all terms are always evaluated at the end of the time step. Applying second order backward differentiation formula (BDF) for temporal discretization, the semi-discretized form of momentum equation is obtained:

$$\rho^{n+1}\frac{3\mathbf{v}^{n+1}-4\mathbf{v}^{n}+\mathbf{v}^{n-1}}{2\Delta t}+\rho^{n+1}\mathbf{C}(\mathbf{v}^{n+1})=-\nabla p^{n+1}+\mathbf{D}(\mathbf{v}^{n+1})+\mathbf{f}(\mathbf{v}^{n+1}) \tag{8}$$

The equation (8) is non-linear because the convective term is evaluated at the end of the time step. The standard Picard method is applied to linearized convective term $\nabla\cdot(\mathbf{v^{n+1}v^{n+1}})\approx\nabla\cdot(\mathbf{v^{m,n+1}v^{m+1,n+1}})$, where m and $m+1$ are used to denote previous and current outer (non-linear) iteration values. Moreover, since standard FV linear interpolation (equivalent to the central-difference finite difference discretization) of the convective term is only conditionally stable, as well as it can lead to matrix system that is not diagonally dominant (causing some iterative linear solvers fail to converge), the convective term is discretized using deferred-correction approach [8].

The matrix form of the discretized equation for the i-th grid point of the x velocity component can be written as:

$$A^{m}u_{i}^{m+1}=Q_{i}^{m}-G_{i}(p^{m}) \tag{9}$$

where A denotes matrix elements for row i, Q_i is the source term containing all the terms that can be explicitly computed (using values from previous time step or previous outer iteration), and G_i is shorthand notation for the discretized gradient operator. The time step superscripts were dropped out due to the fact that in the SIMPLE algorithm all the terms (except for the velocity part from the discretized temporal derivative) are evaluated at the end of the time step $n+1$, and only superscript for outer iterations is of importance here. The final coefficients for x component of discretized momentum equations (using second-order finite volume method with deferred-correction approach for the convective term) are given by:

$$\begin{aligned}
a_E^u &= \rho_P^*\cdot\min(q_e^*,0)-\frac{\mu_e^* S_e}{x_E-x_P}, & a_W^u &= \rho_P^*\cdot\min(q_w^*,0)-\frac{\mu_w^* S_w}{x_P-x_W},\\
a_N^u &= \rho_P^*\cdot\min(q_n^*,0)-\frac{\mu_n^* S_n}{y_N-y_P}, & a_S^u &= \rho_P^*\cdot\min(q_s^*,0)-\frac{\mu_s^* S_s}{y_P-y_S},\\
a_T^u &= \rho_P^*\cdot\min(q_t^*,0)-\frac{\mu_t^* S_t}{z_T-z_P}, & a_B^u &= \rho_P^*\cdot\min(q_b^*,0)-\frac{\mu_b^* S_s}{z_P-z_B},\\
a_P^u &= \rho_P^*\cdot\frac{3}{2}\cdot\frac{\Omega_P}{\Delta t}-(a_E^u+a_W^u+a_N^u+a_S^u+a_T^u+a_B^u).
\end{aligned} \tag{10}$$

The coefficients are denoted with superscript u representing x component of momentum equation; however, because the co-located grid arrangement is used, and the part of the diffusion term treated explicitly (in terms of previous outer iteration), the coefficients are the same for all velocity components.

Similarly to the EFSM, the pressure Poisson equation (PPE) for SIMPLE algorithm can be derived by expressing the velocity at the end of the time step $\mathbf{v^{n+1}}$ from equation (8) and combining it with continuity equation (2). The most important difference between EFSM and SIMPLE is that the influences of the convective and viscous terms are now included in the PPE. Since PPE is derived by incorporating momentum equation into continuity constraint, it is important that discretization of the same terms is consistent in the momentum and the PPE equation. This is usually achieved by using a discretized form of momentum equation (9) during derivation of PPE.

Moreover, the PPE equation for SIMPLE is usually derived using pressure and velocity corrections (p' and $\mathbf{v}'$), rather then for actual values:

$$p^* = p^m + p' \ , \ u_i^{**} = u_i^* + u_i' \tag{11}$$

where superscripts $*$ and $**$ denote new corrected values at the end of the current outer iteration. Now, using equation (11) and satisfying a simplified version of the equation (9), the relationship between velocity and pressure corrections can be defined [8]:

$$A_D u_i' + A_{OD} u_i' = -G_i(p') \tag{12}$$

where matrix A was split into a diagonal A_D and off-diagonal part A_{OD}. In the original SIMPLE algorithm, the second term on the left hand side is neglected. This is justified by the fact that (12) is a correction equation, where all corrections are expected to converge to zero during the outer iterations. However, these simplifications generally affects the convergence rate and is the main reason why under-relaxation is generally needed [8]. The implicit under-relaxation is generally applied for the velocity components (and additional scalar fields, like level-set field in this case), while explicit under-relaxation is applied during pressure calculation. It is worth mentioning that the pressure-corrections are not explicitly under-relaxed when correcting velocity components to assure that velocity field at the end of the outer iteration is divergence free (to accuracy of the PPE solution).

Final coefficients and right hand side for discretized pressure-correction equation are given by:

$$\begin{aligned}
a_E^{P'} &= -\frac{S_e}{2} \cdot \left[\left(\frac{1}{a_P^u} \right)_P + \left(\frac{1}{a_P^u} \right)_E \right], \qquad a_W^{P'} = -\frac{S_w}{2} \cdot \left[\left(\frac{1}{a_P^u} \right)_P + \left(\frac{1}{a_P^u} \right)_W \right], \\
a_N^{P'} &= -\frac{S_n}{2} \cdot \left[\left(\frac{1}{a_P^v} \right)_P + \left(\frac{1}{a_P^v} \right)_N \right], \qquad a_S^{P'} = -\frac{S_s}{2} \cdot \left[\left(\frac{1}{a_P^v} \right)_P + \left(\frac{1}{a_P^v} \right)_S \right], \\
a_T^{P'} &= -\frac{S_t}{2} \cdot \left[\left(\frac{1}{a_P^w} \right)_P + \left(\frac{1}{a_P^w} \right)_T \right], \qquad a_B^{P'} = -\frac{S_b}{2} \cdot \left[\left(\frac{1}{a_P^w} \right)_P + \left(\frac{1}{a_P^w} \right)_B \right], \\
a_P^{P'} &= -(a_E^{P'} + a_W^{P'} + a_N^{P'} + a_S^{P'} + a_T^{P'} + a_B^{P'}), \\
b_P^{P'} &= -(q_e^* + q_e^* + q_w^* + q_n^* + q_s^* + q_t^* + q_b^*).
\end{aligned} \tag{13}$$

The SIMPLE algorithm in the present work uses a segregated approach for solution of the coupled system of the nonlinear equations during each time step. Thus, during the discretization of each equation (velocity component, pressure corrections, or any additional scalar field like level-set in this case) all other variables

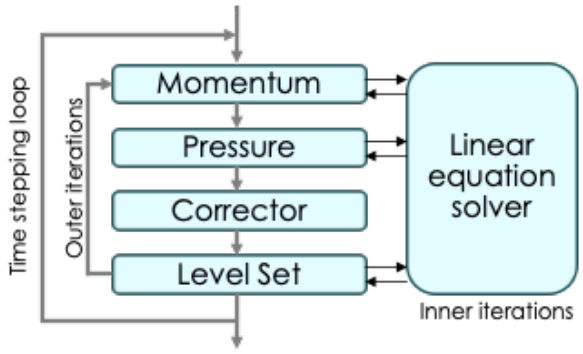


Figure 2. Flow chart of SIMPLE-type method.

(except the variable being solved for) are treated as a known and assigned last known solution value. Thus, after solution of the linearized equations for each velocity component, the pressure corrections needs to be calculated by solving the Poisson-type equation. Using pressure corrections, both pressure and velocity fields are corrected to obtained divergence-free velocity field at the end of the time step. However, this field does not satisfy conservation of the momentum, as well as interface defined by the level-set field now needs to be updated using latest velocity field, which is done in the next outer iteration. The iterations are stopped and the calculation for the next time step can begin when the residuals for each particular equation are below prescribed tolerance.

For efficiency reasons, the solution of each linearized system of the equation (for velocity component, pressure-corrections, or level-set) doesn't need to be accurately solved, because its matrix coefficients and right hand side will be updated in the next outer iteration. Usually, reducing residuals one order of the magnitude in the each outer iteration is sufficient and solving for a tighter convergence will not reduce the number of the outer iterations (thus only increasing overall computational cost). Moreover, the accurate extrapolation of the initial guess for the first outer iteration (instead using previous time step solution) can reduce the number of the outer iterations. In this work, second order extrapolation using solutions from two previous time steps is used. Finally, the flow chart of the SIMPLE algorithm is show in Figure 2.

3.1.3 Variable Arrangement on the Grid

In a co-located arrangement of variables on the grid shown in Figure 3, all the variables are stored at the same set of points. Because many of the terms in each of the equations are essentially identical, the number of coefficients that must be computed and stored is minimized, and the programming is simplified by this choice. Furthermore, when multigrid procedures are used, the same restriction and prolongation operators for transfer of information between the various grids can be used for all variables [8].

The co-located arrangement was out of favor for a long time for incompressible flow computation due to the difficulties with pressure-velocity coupling and the occurrence of oscillations in the pressure. The staggered grid was introduced in the 1960s [10] and used almost exclusively until the 1980s when stabilization techniques for pressure oscillations (most notably Rhie-Chow interpolation) were introduced. The staggered grid uses different variable arrangements for different variables (Figure 3), and its biggest advantage is the strong coupling between velocity and pressure. It also conserves kinetic energy. Its main

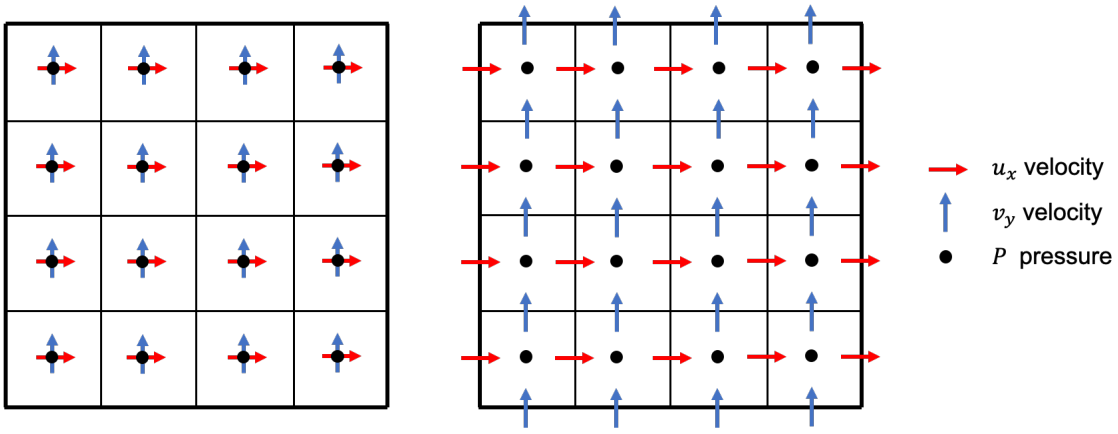


Figure 3. Co-located (left) and staggered (right) variable arrangement for velocity components and pressure.

Table 1. Summary of the differences between EFSM and SIMPLE algorithms implemented in the code

	EFSM	**SIMPLE**
Grid	staggered	co-located
Flow solver spatial discretization (order)	FDM (2^{nd})	FVM (2^{nd})
Flow solver temporal discretization (order)	explicit (1^{st})	implicit (2^{nd})
Level-set spatial discretization (operator, order)	FDM (HOUC3, 3^{rd})	FVM (QUICK, 3^{rd}))
Level-set temporal discretization (operator, order)	explicit (4^{th})	implicit (2^{nd})

drawback, beside the somewhat more complicated programming and memory requirements, is that the staggering procedure becomes very complicated for non-Cartesian and, in particular, for unstructured grids.

Both co-located and staggered variable storage are used in the Quilt, depending on the flow solver used. The staggered grid procedure is used within previously developed EFSM algorithm mostly due to its robustness for the multiphase flow simulations, and particularly because the Quilt uses structured grids relaying on the level-set/immersed-boundaries approaches for modeling of static or dynamic interfaces or boundaries. During development of the fully implicit SIMPLE algorithm, the co-located grid was chosen due to complications encountered during implementation of the geometric multigrid using the staggered grid. Also, different prolongation/restriction operators for different variables are needed. Thus, SIMPLE algorithm implemented in the Quilt uses co-located grid and relies on the Rhie-Chow stabilization [11] for suppressing pressure oscillations.

The main differences between the EFSM and SIMPLE algorithms for Finite Difference Method (FDM) and Finite Volume Method (FVM) implemented in Quilt are summarized in the Table 1.

3.2 Geometric multigrid solver

As shown in previous sections, the solution of incompressible flow equations requires solving system of equations during each time step. This is the case even when fully explicit algorithm (see Figure 1) is used,

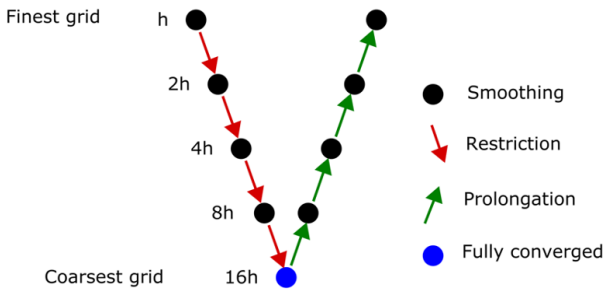


Figure 4. Schematic presentation of geometric multigrid scheme using V-cycles.

since continuity constraint translates in pressure Poisson equation, which is of the elliptic type. Thus, contrary to compressible flow problems, the incompressible flow problem always requires the solution of linear system of equations.

In the case of EFSM algorithm, the pressure Poisson equations has to be solved only once per time step, while for SIMPLE algorithm the momentum, pressure-correction, and level-set equations require solutions of multiple linear systems of equations during each outer iteration. However, the solutions of the momentum and level set equations, which are of parabolic and hyperbolic type, are generally easier then solution of the pressure-correction equation. Thus, PPE usually represents main computational cost of most incompressible flow solvers. Moreover, because the SIMPLE type of the algorithm solves for series of linearized problem in segregate manner (Figure 2), the solution of system of equations doesn't need to be solved to a very tight tolerance (like PPE in EFSM), because coefficients are always updated during each outer iteration. Usually, reducing residuals one order of magnitude in each iteration is sufficient and performing more inner iterations for linear system of equations would not reduce the number of outer iterations, and thus only can increase overall computational cost.

There are different approaches for the solution of the linear system of equations [12]. The most simplest ones are basic iterative methods (often denoted also as relaxation methods) such as Jacobi, Gauss-Seidel, and successive over-relaxation. These methods are simple to understand and easy to parallelize, which makes them attractive for GPU-accelerated codes such as Quilt. However, the main drawback of such methods for large scale simulations is that their rate of convergence deteriorates as the size of the system of the equations increases. This is because the information needs to travel several times back and forth across the domain for the solution to converge.

Geometric multigrid [13, 14] is one of the most powerful approaches for improving the convergence rate of basic iterative methods. The main idea is to use the sequence of coarser grid problems to find global corrections on the finest level, while its main strength is that the required number of iterations is independent of the number of grid points (i.e. size of the system of equations). In this work, the basic under-relaxed Jacobi method is used as a smoother, while direct injection and linear interpolation are used as restriction and the prolongation inter-grid operators. The efficiency of the implemented parallel geometric multigrid solver is demonstrated by an example.

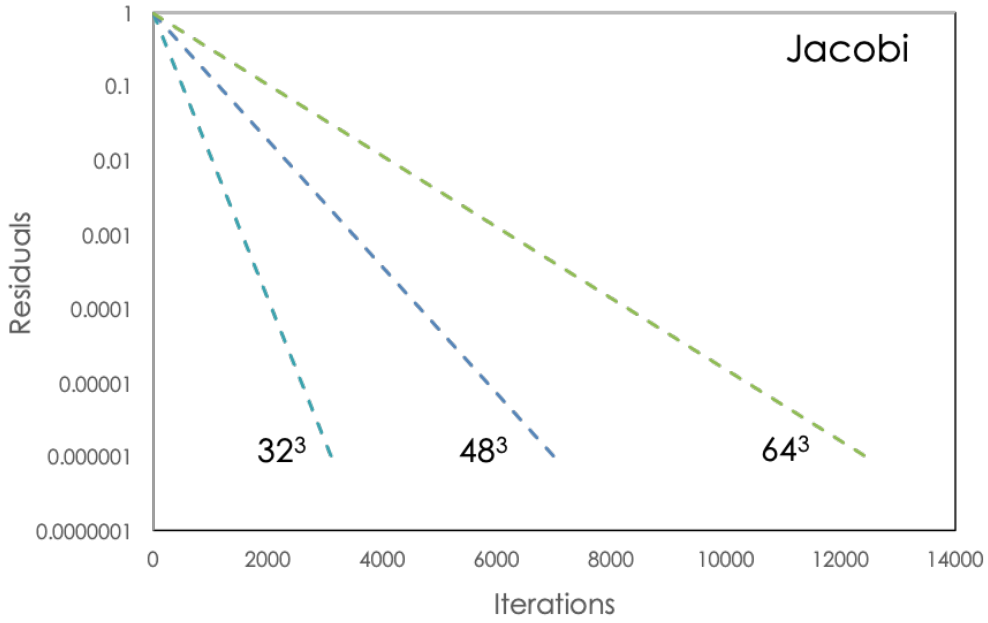


Figure 5. Residual as the function of the number of the iteration for Jacobi method and three different discretization levels.

3.2.1 Multigrid demonstration

To demonstrate efficiency of the described geometric multigrid, here we present solution of the Poisson equation with available analytical solution. The governing equation is give by:

$$\nabla^2\phi = -3\pi \ \sin(\pi x)\sin(\pi y)\sin(\pi z) \tag{14}$$

with (periodic) boundary conditions:

$$\phi(0,y,z) = \phi(1,y,z) = \phi(x,0,z) = \phi(x,1,z) = \phi(x,y,0) = \phi(x,y,1) = 0 \tag{15}$$

and the analytical solution:

$$\phi(x,y,z) = \sin(\pi x)\sin(\pi y)\sin(\pi z) \tag{16}$$

First, Figure 5 demonstrates classical behavior of the most relaxation methods. For the same problem, the number of iterations needed for reaching the same residual error rapidly increases when using finer grids.

Figure 6 shows comparison of the residual convergence between the Jacobi method and the MG method based on the number of performed iterations. For the Multigrid method, the number of the iterations is sum of all performed iterations on the all levels.

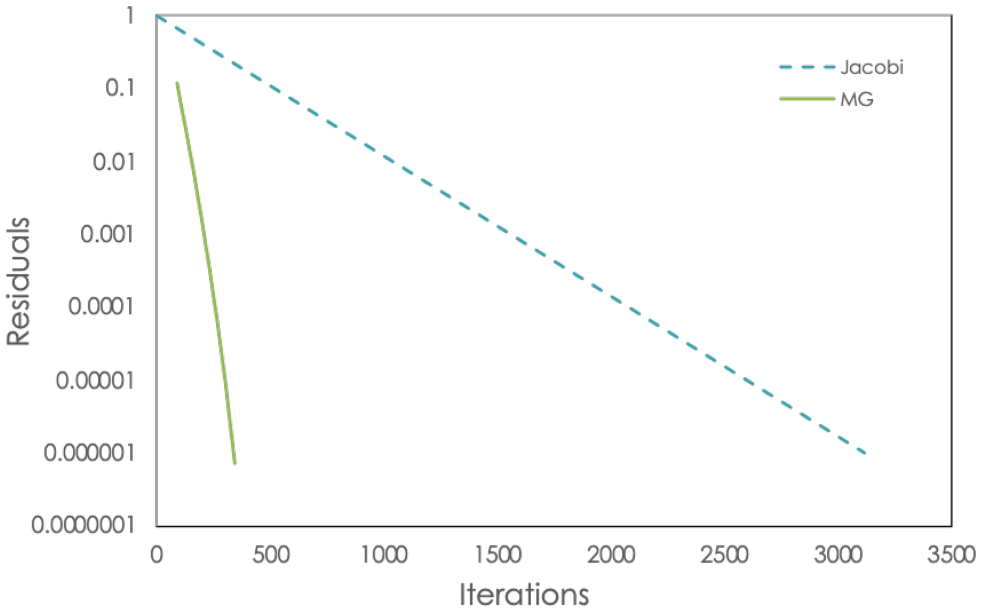


Figure 6. Residual as the function of the number of the iteration for Jacobi and Multigrid (MG) method for number of grid points $N = 32^3$.

Table 2. Performance comparison between Jacobi method and geometric multigrid for the solution of the problem (14)-(15).

	Iterations (V-cycles)		CPU [s]		
Grid	**Jacobi**	**MG**	**Jacobi**	**MG**	**Speedup**
32^3	3113	5	9.42	1.03	9.14
48^3	7008	5	295.05	2,18	135.53
64^3	12461	5	1386.43	4.38	316.83
96^3	28042	5	7559.08	15.67	482.27
128^3	49854	5	38419.85	53.85	719.78
192^3	112176	5	993967.8	218.06	4558.23
256^3	-	5	-	526.48	-

Finally, Table 2 compares the performance of the geometric multigrid solver using the Jacobi method as a smoother with regular Jacobi solver without multigrid acceleration. The number of the total iteration for basic Jacobi solver is drastically rising when increasing grid resolution for the same problem, while we see that this is not case when considering the number of the V-cycles for the multigrid. The efficiency of the multigrid solver is clearly demonstrated by measuring the the time-to-solution time. It is visible that the efficient linear solvers, like presented geometric multigrid, become crucial tools for enabling the large-scale simulations.

3.3 Interface Tracking Method

Capturing and tracking of the time varying gas/liquid interface is a critical aspect of multiphase flow problems. Various methods for accomplishing this can be categorized into Lagrangian [15] and Eulerian approaches. In front tracking approaches, the interface is represented with a collection of mass-less particles (markers). Geometric properties of the interface such as normal vector and curvature used in modeling interfacial phenomena are computed using polynomial construction. In problems where the interface often undergoes topological changes, maintaining a constant particle surface density is a challenge and involves the addition or removal of markers to ensure consistent accuracy. In the context of Eulerian approaches, phase field [16], Volume of fluid (VOF) [17] and level set [18] are most commonly employed. While phase field offers advantages in terms of mathematical representation of interfacial phenomena, it does not capture a sharp interface. The VOF method represents the interface using a conservative function which is the fraction of mass of a particular phase in a given control volume. The actual definition of the interface depends on additional construction algorithms [19, 20]. The level set is a signed function whose magnitude represents the shortest distance at any grid point from the interface. It implicitly captures the location and orientation of the interface as the zero isocontour. Both VOF and level set functions capture the interface implicitly in time through transport equations for the respective marker functions. Two major differences between these approaches are: (1) VOF formulation ensures mass conservation through the transport equation. On the other hand, to ensure the distance property of the level set function is retained, it needs to be reinitialized after advection, resulting in artificially movement of the interface and consequently, conservation errors; and (2) Level set, being a distance function, enables computation of interface geometric properties through difference operators. VOF needs an additional interface construction prior to evaluation of these properties. Such construction algorithms are mathematically complex [19, 20] and their application in large scale simulations remains to be explored. Scalability and accurate computation of interface geometry using the level set method motivates its use in the current problem. The conservative high order level set implementation has been briefly outlined here with a detailed description available in Ref. [21].

Level set function is defined as:

$$\phi = \begin{cases} -d, & \text{Gas} \\ 0, & \text{Interface} \\ d, & \text{Liquid} \end{cases}, \tag{17}$$

where d is the shortest distance of a point from the interface, schematically shown in Figure 7. The gas and liquid phases are characterized by negative and positive values of the function respectively. The interface, given by the zero isocontour, is captured implicitly. Being a distance function, it can be understood that the

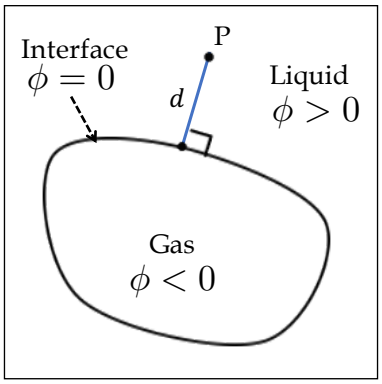


Figure 7. Schematic representation of the level set function distribution in a computational domain.

local normal and curvature are given by

$$\hat{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad \text{and} \tag{18}$$

$$\kappa = \nabla\cdot\hat{n} \ . \tag{19}$$

Since the level set contours are parallel (imposed by reinitialization described later), normal vector on the interface can be approximated as the local normal. Interface curvature needs high order interpolation based on local values in the vicinity of the zero contour.

One of the limitations of the level set method is its lack of conservation. Advection of the level set function does not guarantee distance property. Periodic reinitialization is needed to ensure Eq. 17 is satisfied by solving the Eikonal equation given by

$$|\nabla\phi| = 1 \tag{20}$$

However, reinitialization is known to introduce artificial motion of the interface resulting in unphysical mass gain/loss. Recent studies have focused on addressing this limitation through various formulations. These include modification to the transport equation [22, 23] and high order accurate reinitialization schemes [21]. A recent review on these approaches highlights their advantages and limitations. These methods commonly involve transport of the quantity $|\nabla\phi| - 1$ in pseudo time with a speed equal to the sign function. The sign function is smoothed over a finite thickness to ensure stability. Theoretically, it can be shown that steady state solution of this transport equation is equivalent to the solution of Eq. 20. However, such an approach is known to introduce mass conservation errors. Another approach for reinitialization is to obtain a direct solution of the Eikonal equation. Fast marching [24] is a scheme that propagates the distance information outwards from the interface by algebraically solving for the distance function in the discretized form of Eq. 20. The method, however, relies on the known distance function in the vicinity of the interface which acts as boundary condition. In practical applications, these values are not available and need to be approximated. Moreover, such methods are not scalable for large scale applications. An alternative to the fast marching, called fast sweeping method is scalable and also offers a convenient formulation to include algebraic anchoring to supply the necessary the boundary conditions. A detailed description of the algorithm can be found in Ref. [21]. The high order anchoring coupled with a consistent fast sweeping scheme enables mass conservation improvements, and also accurate computation of interface normal and curvature.

Using a schematic representation in Figure 8 (taken from Ref. [21]), we briefly outline the anchoring and reinitializtion equations. The anchor point shown by filled symbols are assigned a level set function by solving for ϕ_B in

$$\phi = \phi_B + \delta n \frac{\partial \phi}{\partial n} + \frac{1}{2}\delta n^2 \frac{\partial^2 \phi}{\partial n^2} + \ldots \ , \tag{21}$$

which is basically the high order expansion of ϕ at the interface in the neighborhood of an anchor point 'B.' n is the signed distance in the normal direction, $\hat{n}$. In the above equation, $\phi = 0$ since the interface is defined as the zero isocontour. These anchor points provide the boundary condition for the fast sweeping solution of Eq. 20. The equation at an interior point, 'I,' shown by open symbols can be discretized as

$$\left(\frac{\phi_I - \phi_{nb,x}}{h}\right)^2 + \left(\frac{\phi_I - \phi_{nb,y}}{h}\right)^2 + \left(\frac{\phi_I - \phi_{nb,z}}{h}\right)^2 = 1 \ , \tag{22}$$

where ϕ_{nb} denotes the distance function at the immediate neighbor of 'I' in each dimension, closest to the interface, i.e. having a smaller magnitude compared to 'I'. These can also be high order approximations. Note that this is an algebraic equation. In certain cases, the above equation involves coupled points, shown by open triangles. The existance of coupled points implies that ϕ_{nb} cannot be explicitly calculated. An underrelaxed iterative solution is sought, which has been observed to converge within less than ten iterations (particularly, since a smooth initial approximation is already available from the advection equation).

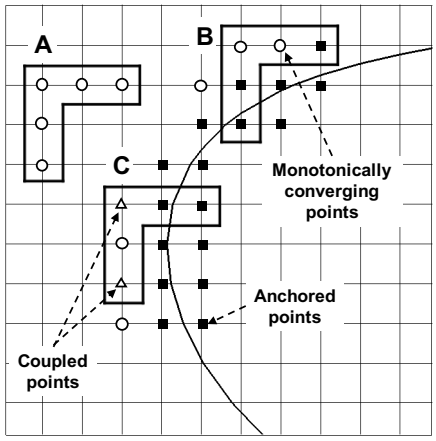


Figure 8. Schematic representation of the anchoring and fast sweeping procedure for reinitialization.

3.4 Model Verification

3.4.1 Taylor-Green Vortex

Taylor-Green vortex is the fundamental flow problem outlined by the High-Order CFD Workshop [25] and is used to study vortex dynamics, transition to turbulence and energy dissipation process. It is a standard benchmark test case used for the validation of both compressible and incompressible flow solvers due to its

simple initial and boundary conditions, as well as the available reference DNS solution [26]. The domain is a triply-periodic box $[\pi, \pi, \pi]$ and initial conditions are specified by:

$$u = V_0 \sin(x) \cos(y) \cos(z) \tag{23}$$

$$v = -V_0 \cos(x) \sin(y) \cos(z) \tag{24}$$

$$w = 0 \tag{25}$$

$$p = p_0 + \frac{\rho_0 V_0^2}{16} + (\cos(2x) + \cos(2y))(\cos(2z) + 2)\,. \tag{26}$$

Figure 9 shows such initial condition flow fields in 2D plane. This problem is used for verification of implicit (SIMPLE-type) flow solver described in Section 3.

The vortex stretching will occur only in 3D configurations. However, the two-dimensional case of the same problem is useful since it has closed analytical solution. Thus, both 2D and 3D simulations are performed to test the numerical method.

First, the 2D problem is used to check grid convergence order. To do so, the temporal solution error is made negligible by using a small enough time step ($\Delta t = 10^{-5}$), and performing only ten time steps to avoid accumulation of time integration error. All non-dimensional parameters are set to unity. As explained earlier, the second order spatial discretization of convective term is implemented via a deferred-correction scheme and can be blended with low order discretization (if needed due stability reasons). Thus, Figure 9 shows grid convergence of L_2 error norm. Both first order (upwind) and second order (linear) interpolation profiles recover expected convergence order for both pressure and velocity fields. The same behavior is observed in L_∞ norm.

Next, the 3D case of Taylor-Green breakdown is considered. The physical time frame is from 0 to 20, density ρ, and initial velocity V_0 are set to unity and viscosity is chosen so that the Reynolds number is 1600. This configuration is used in many studies using both compressible and incompressible solvers due to availability of spectral DNS solution [26], and set as standard reference data during High-Order CFD workshop (HOCFDW).

Figure 11 shows evolution of vorticity field in time, while Figure 12 demonstrates effects of grid resolution before breakdown.

The temporal evolution of kinetic energy integrated over the domain:

$$E_k = \int_\Omega \rho \frac{\mathbf{v} \cdot \mathbf{v}}{2} d\Omega \tag{27}$$

and temporal evolution of the kinetic energy dissipation rate: $\epsilon = -\frac{dE_k}{dt}$ are calculated and compared with reference data (HOCFDW) using different numerical parameters as shown on Figures 13, 14 and 15.

Finally, comparison of numerical results obtained for different Reynolds numbers is shown in Figure 16.

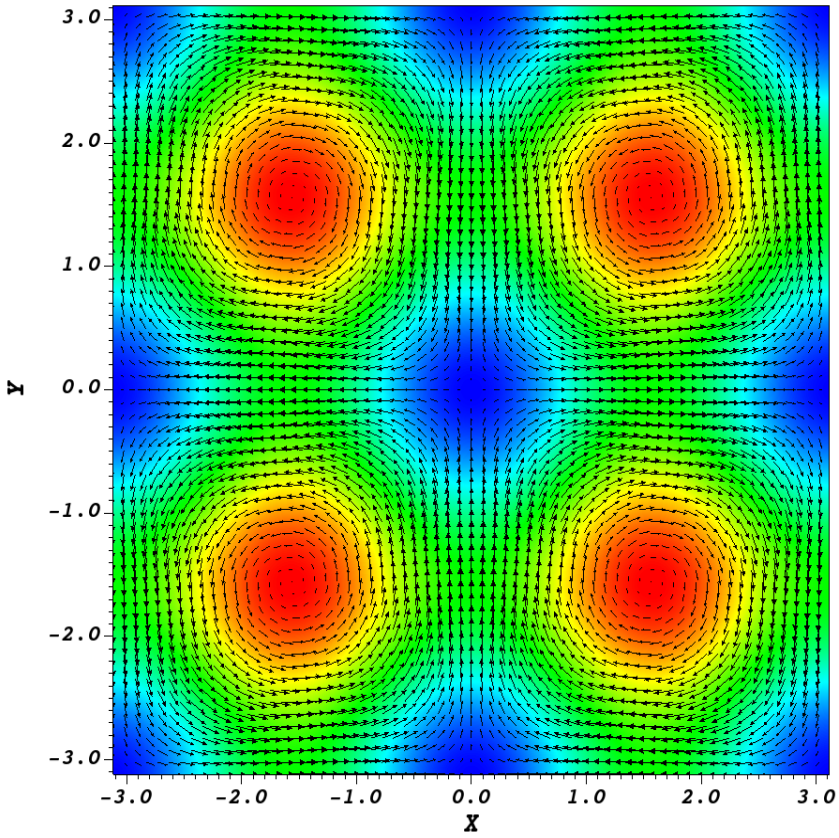


Figure 9. Initial conditions (pressure and velocity fields) in 2D plane for Taylor-Green vortex.

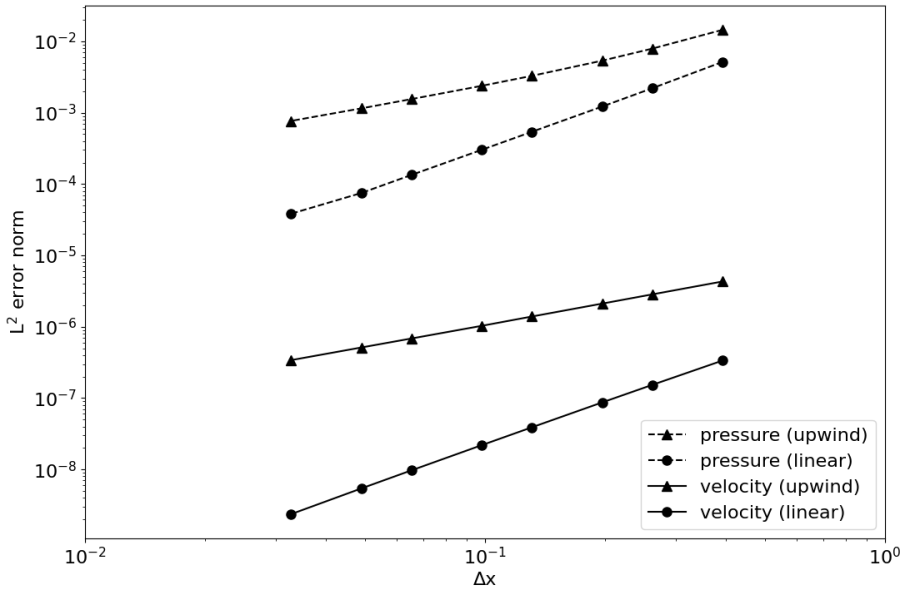


Figure 10. Convergence order for 2D Taylor-Green problem. First order upwind and second order linear interpolations are presented. Linear interpolation is implemented using deferred-correction approach.

3.4.2 Static drop test

Next, we proceed to validate the model for the multiphase interface using an established test case for surface tension modeling [27, 28]. A circular liquid droplet with a zero initial velocity is positioned at the center of a 2D computational domain of 8.0x8.0 with a radius $R = 2.0$. Liquid and gas density are given by $\rho_l = 1.0, \rho_g = 10^{-3}$, while viscosity is $\mu_l = 10^{-2}$ for liquid, and $\mu_g = 10^{-3}$ for gas phase. The gravity is set to zero and the surface tension is $\sigma = 73.0$. The 2D domain is discretized using 40x40 grid points and time step size $\Delta t = 10^{-6}$. The exact pressure jump across the interface is given by:

$$\Delta P_{exact} = \sigma\kappa \tag{28}$$

where exact curvature in 2D is $\kappa_{exact} = 1/R$, which for this particular case means that the exact pressure difference is 36.5.

In this section we compare results between two presented algorithms: EFSM and SIMPLE. The Figures 17 and 18 show pressure fields, while Figures 19 and 20 show velocity fields after the first and last time step for both presented methods.

For this example, level set field is not advected or reinitialized, meaning that spurious current do not deform interface which is always circular. However, the effect of spurious currents is still visible due to nonphysical velocity and pressure oscillations, which is a classical problem with basically all surface tension models. Further, in the Figure 21, the relative pressure jump error is shown for both approaches as

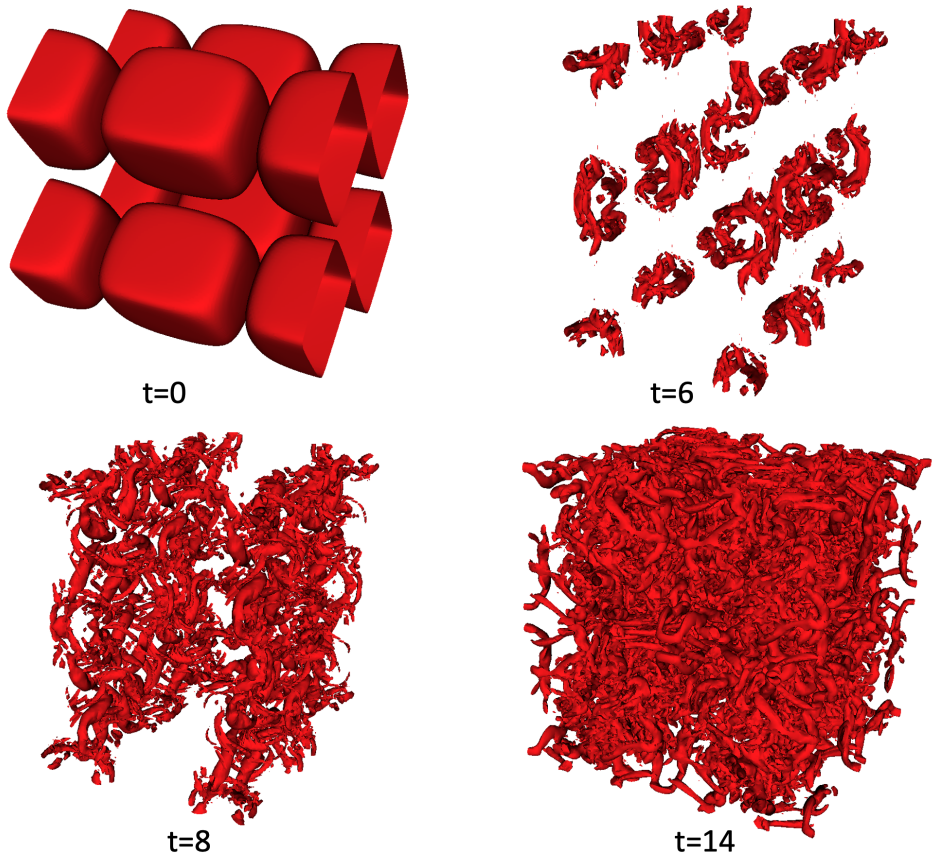


Figure 11. Contours of the Q-criterion for the 3D Taylor-Green vortex at four different times. Case with 128^3 grid resolution and second order BDF2 time integration ($\Delta t = 0.005$) is used.

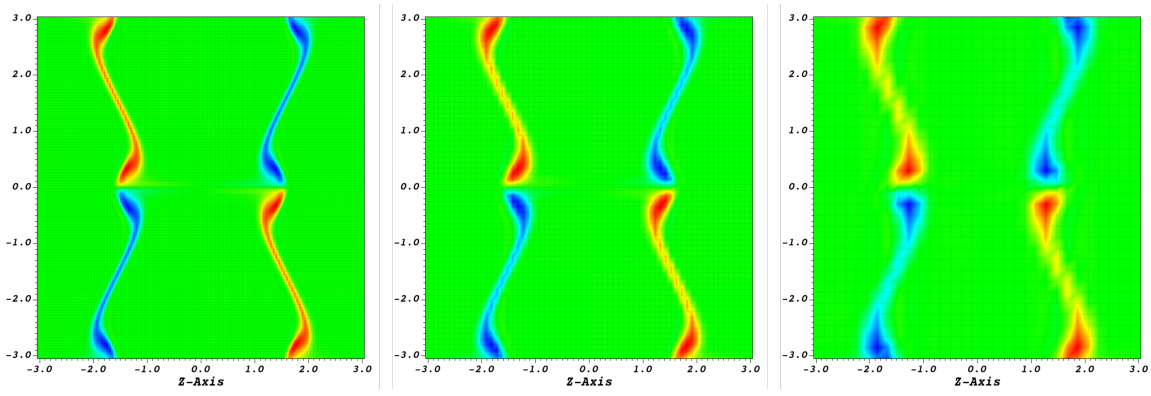


Figure 12. Vorticity in x-plane at $t = 3.75$. Three different discretizations are shown: 128^3, 64^3, and 32^3 .

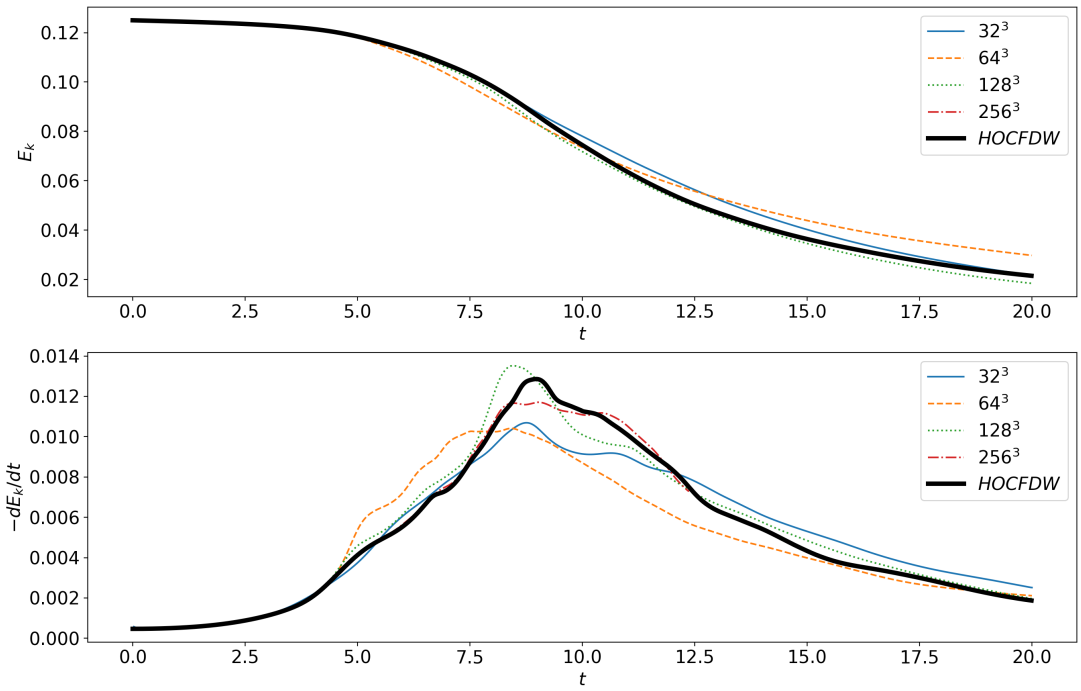


Figure 13. Kinetic energy E_k (top) and kinetic energy dissipation rate $-dE_k/dt$ (bottom) as function of time for 3D Taylor-Green vortex. Four different discretization levels using second order BDF2 time integration and the same time step ($\Delta t = 0.025$) are compared to reference data (HOCFDW).

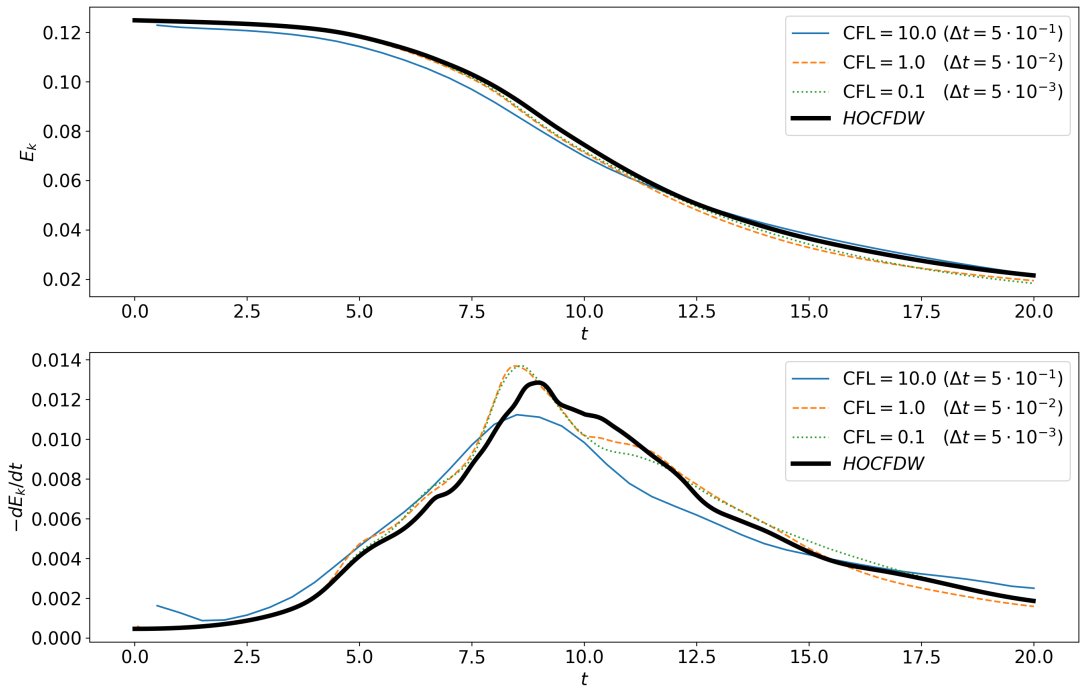


Figure 14. Kinetic energy E_k (top) and kinetic energy dissipation rate $-dE_k/dt$ (bottom) as function of time for 3D Taylor-Green vortex. 128^3 grid resolution and 2^{nd} order BDF2 time integration results are obtained for three different time-steps and compared to reference data (HOCFDW).

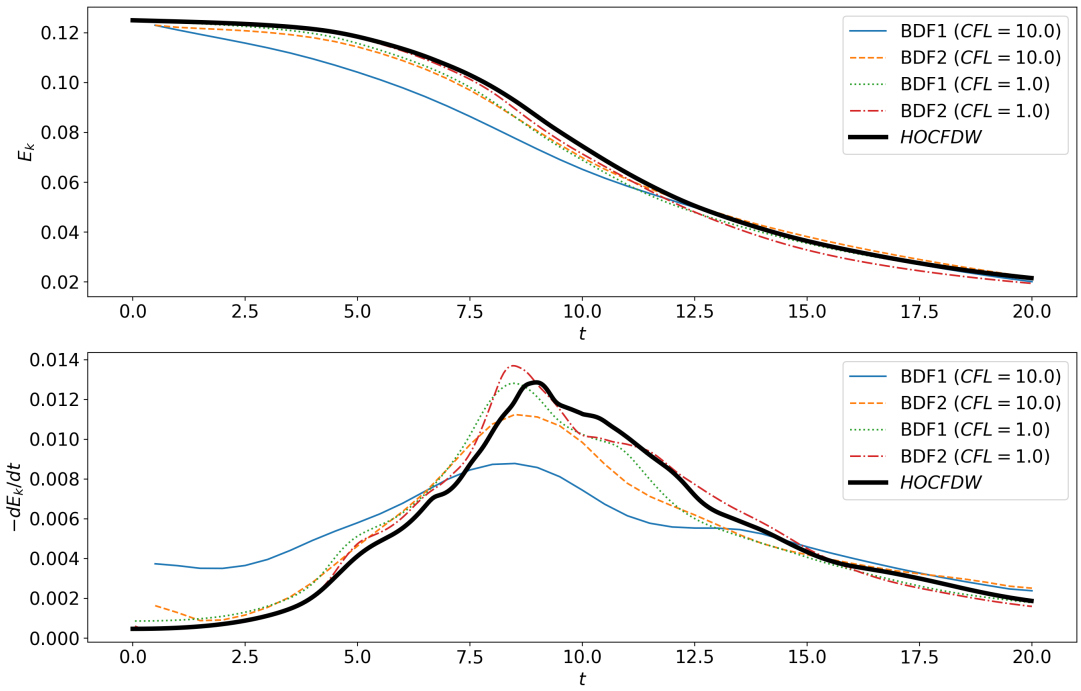


Figure 15. Kinetic energy E_k (top) and kinetic energy dissipation rate $-dE_k/dt$ (bottom) as function of time for 3D Taylor-Green vortex. 128^3 resolution is used, and 1^{st} order BDF1 and 2^{nd} order BDF2 schemes are compared for two different time-step sizes.

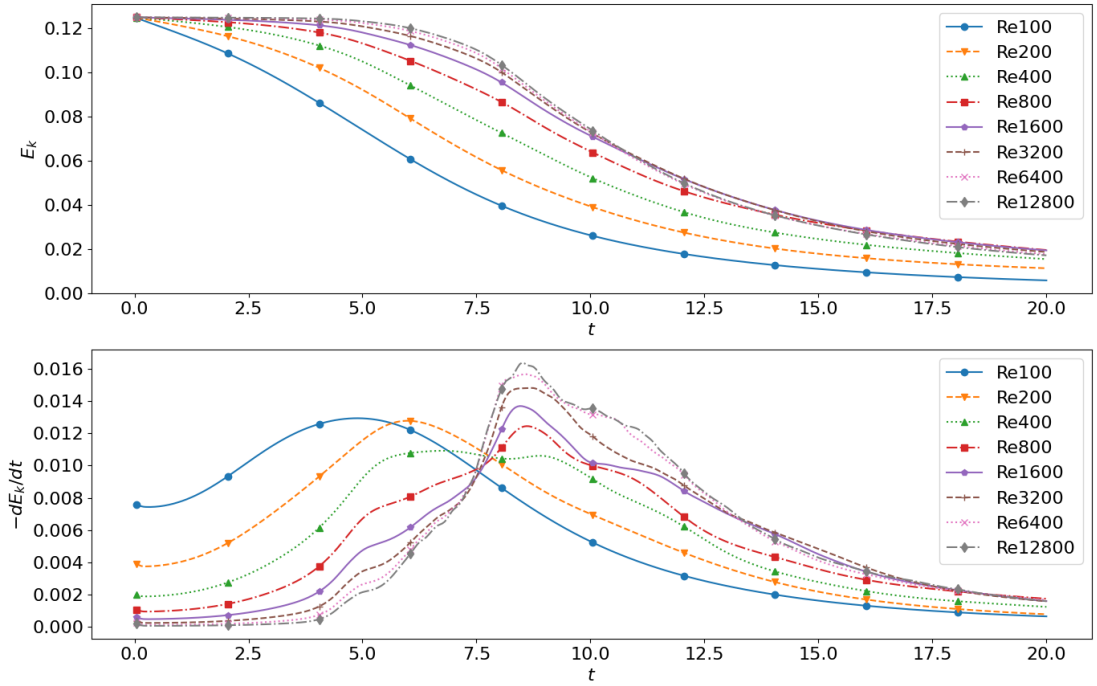
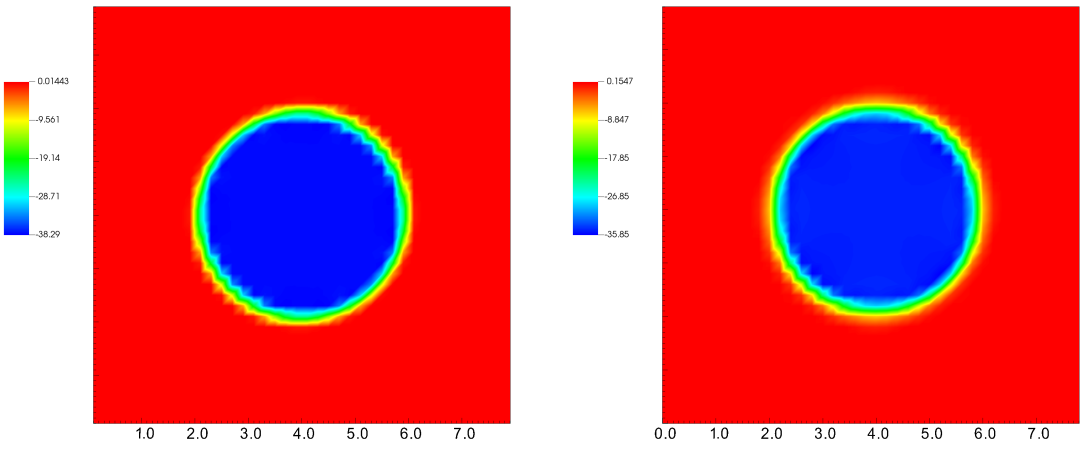
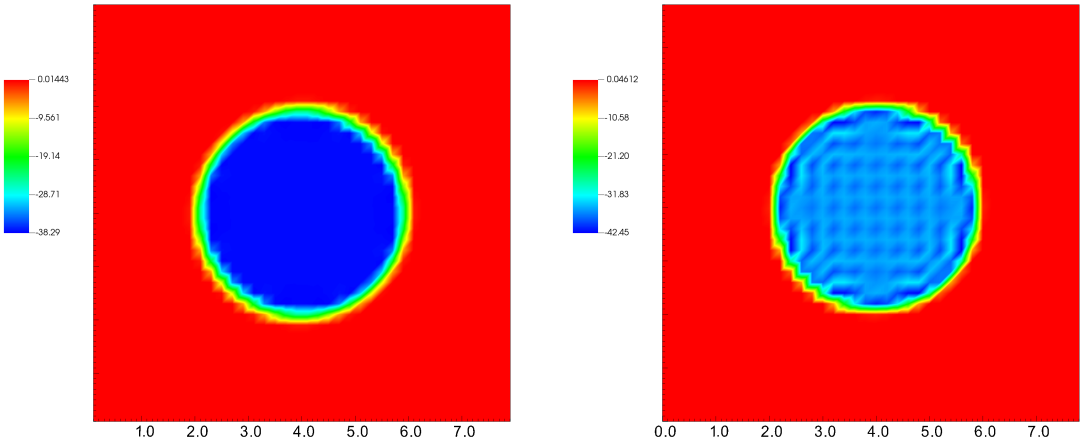


Figure 16. Kinetic energy E_k (top) and kinetic energy dissipation rate $-dE_k/dt$ (bottom) as function of time for 3D Taylor-Green vortex. Comparison of numerical results for different Reynolds numbers. 128^3 grid resolution with 2^{nd} order BDF2 and $\Delta t = 0.05$ are used for all results.



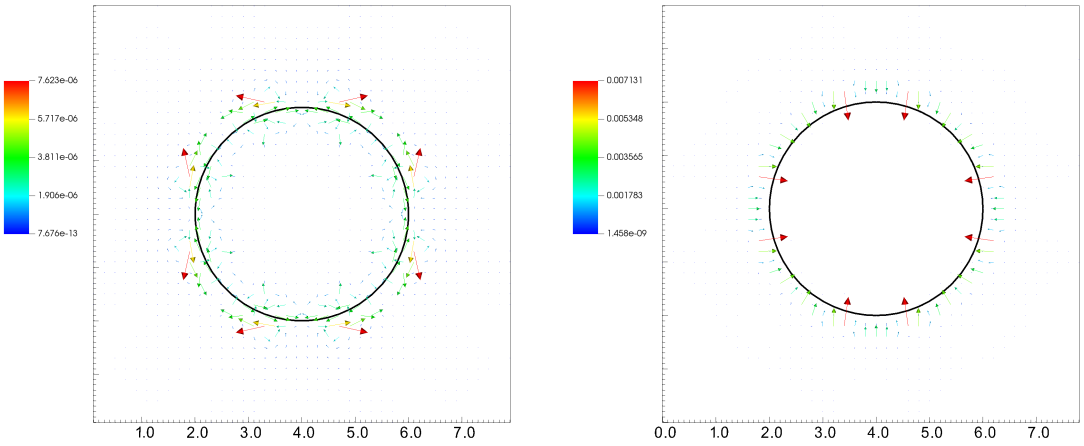
(a) EFSM **(b) SIMPLE**

Figure 17. Pressure field for static drop test after 1 time step.



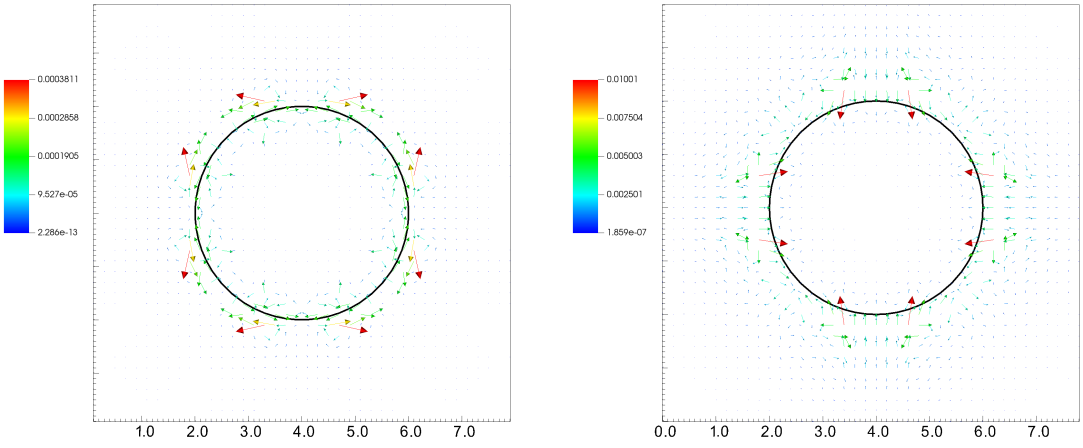
(a) EFSM **(b) SIMPLE**

Figure 18. Pressure field for static drop test after 1 time step.



(a) EFSM **(b) SIMPLE**

Figure 19. Velocity field for static drop test after 1 time step.



(a) EFSM **(b) SIMPLE**

Figure 20. Velocity field for static drop test after 50 time step.

function of the time step. The relative pressure jump error is calculated as:

$$E(\Delta P) = \frac{|\Delta P_{total} - \Delta P_{exact}|}{\Delta P_{exact}} \tag{29}$$

where $\Delta P_{total} = P_{in} - P_{out}$, where subscripts "in" and "out" denotes averaged values inside and outside the drop, respectively. Moreover, Figure 22 shows maximum velocity magnitude resulting from unphysical spurious currents, while Figure 23 shows the ratio of the maximum velocity obtained with two methods.

As demonstrated with presented results, it is clear that the SIMPLE algorithm produces larger spurious currents and higher pressure oscillation near interface. The main reason for this behaviour is believed to be due to the grid variable arrangement differences (staggered vs co-located, see Section 3.1.3) and not due to the different pressure velocity coupling (FSM vs SIMPLE). It is well known that co-located grids are more unstable, and they were out of favor for a long time for incompressible flow computation due to the difficulties with pressure-velocity coupling and the occurrence of oscillations in the pressure. The staggered grid, introduced in the mid-1960s was almost exclusively used until the mid-1980s when the Rhie-Chow stabilization for co-located grid was introduced. Even though we use Rhie-Chow interpolation in this work, still large body forces near interface produces significant problems for co-located grids. Thus, additional work is needed to avoid such pressure oscillation in the case of the strong body forces due surface tension, and some of the possible solutions could be balance-force algorithm [27], or modified Rhie-Chow interpolation [29].

3.4.3 Oscillating drop test

The oscillations of the stretched liquid drop due to surface tension are captured by the kinetic energy variations in time. The oscillations are damped owing to the viscosity of the drop, and it ultimately approaches a state of equilibrium and attains a steady circular shape.

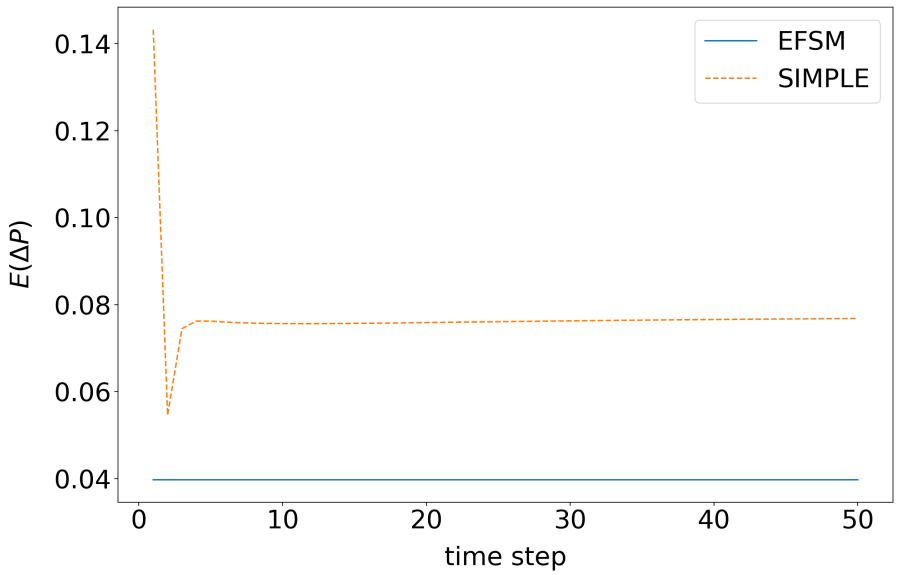


Figure 21. The relative pressure jump error as function of time step.

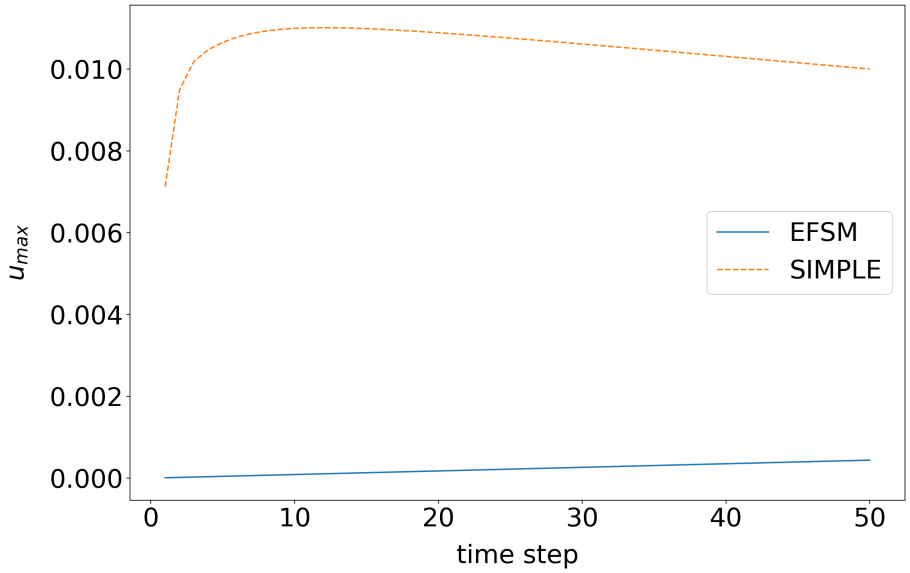


Figure 22. Maximum velocity magnitude as function of time step.

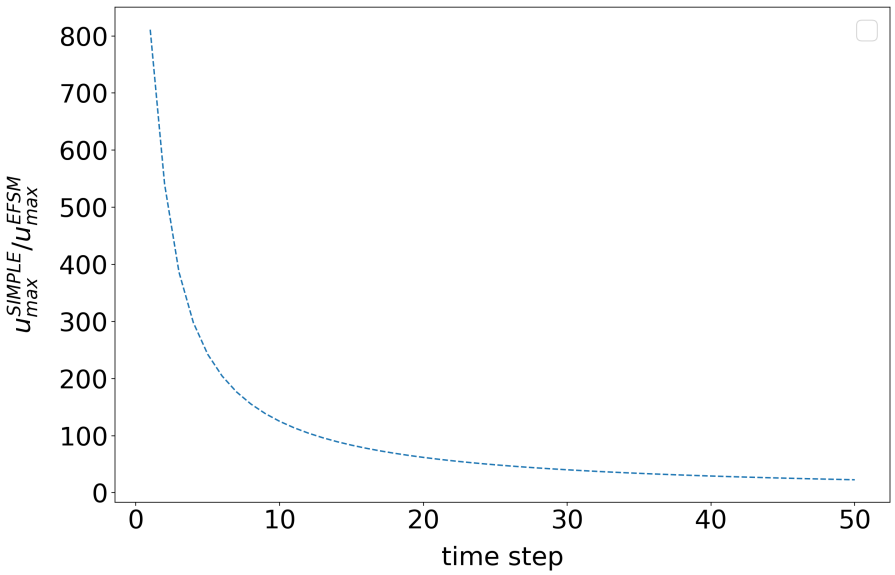
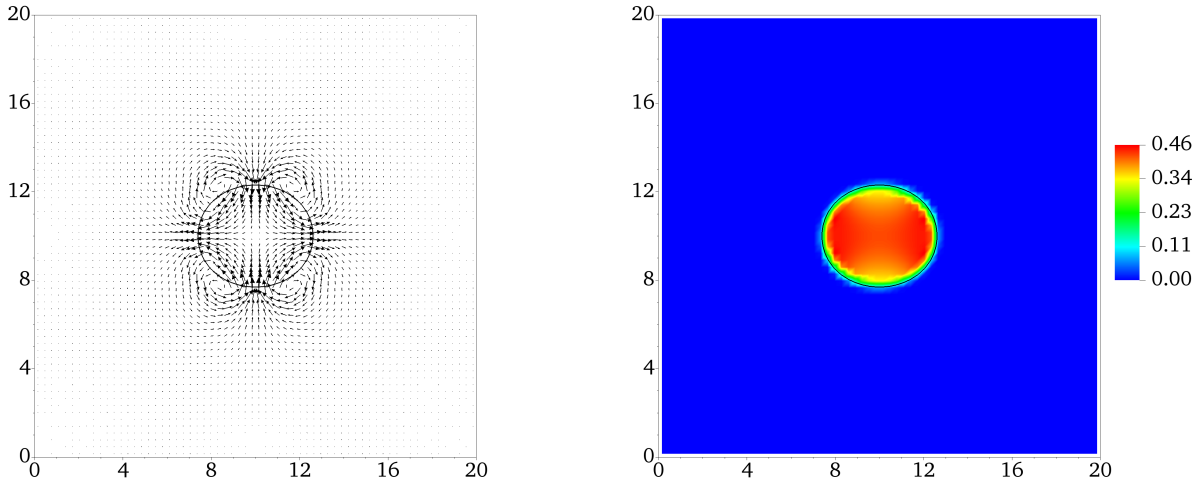
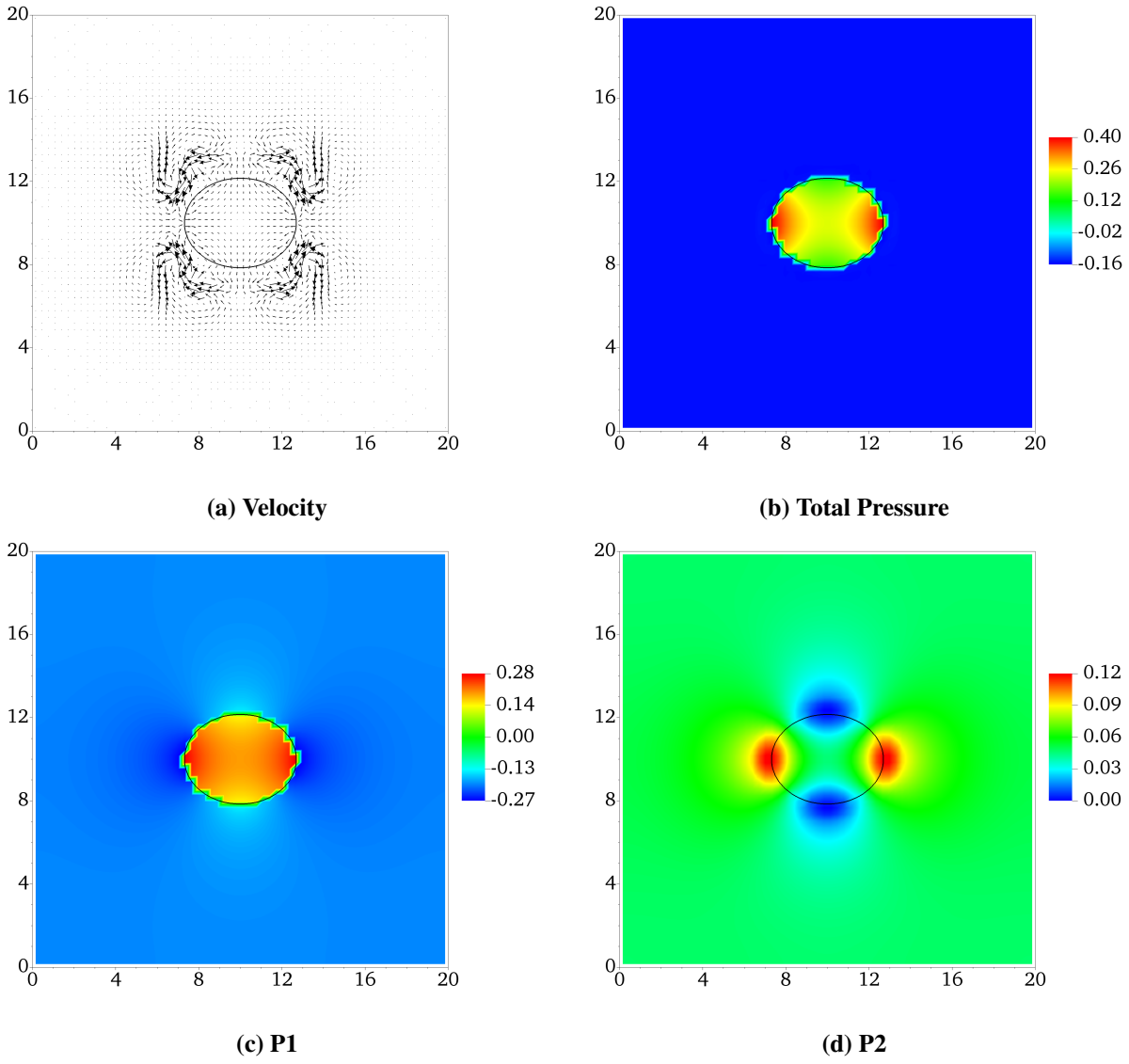


Figure 23. Maximum velocity ratio between two methods as function of time step.



(a) Velocity **(b) Pressure**

Figure 24. Flow field and pressure distribution at $t = 30$ using the CSF method for surface tension.



(a) Velocity **(b) Total Pressure**

(c) P1 **(d) P2**

Figure 25. Flow field and pressure distribution at $t = 30$ using the PBM method for surface tension.

Both the CSF and PBM methods can efficiently capture the oscillations of the drop and the kinetic energy variations are in good agreement with previous works [28, 27, 30, 5], who have used different numerical approaches. Tryggvason and Torres use a front tracking method with marker points distributed on the surface. Accurate curvature can be calculated by high order polynomial construction from the marker points. Francois and Wang use variations of the volume of fluid method where a height function is constructed using the VOF field. Both, front tracking and volume of fluid methods have shown the existence of parasitic (spurious) currents resulting from multiple numerical treatments. Firstly, calculation of interface curvature is a critical aspect in any surface tension implementation. Although the level set function can theoretically result in very good curvature estimates, its accuracy relies on the reinitialization of the function. We have shown the correlation of curvature accuracy with that of reinitialization in [21]. Classical reinitialization methods find it challenging to produce similar results and often leads to conservation errors. On the other hand, the VOF techniques ensures conservation but needs additional interface construction and height functions which limits its scalability.

Secondly, it has also been shown that the projection method often leads to a generation of spurious currents as it does not satisfy the sinusoidal velocity field. It is theoretically true in case of a continuum approach where the surface forces are transformed into body forces distributed over a finite interface thickness. Wang [28] have shown that treating the surface tension as a truly interfacial phenomena can significantly reduce the spurious currents. We implemented the PBM and applied it to the oscillating drop problem. As seen in Figure 26, the CSF method results in some residual kinetic energy which comes from the spurious currents. The PBM method significantly reduces this effect and the kinetic energy is found to be more than one order of magnitude smaller. Torres [30] points to a similar observation using a front tracking approach and shows a modified projection algorithm in 2D (3D implementation is not trivial) to reduce the residual kinetic energy arising from spurious currents. Using a level set method which provides accurate interface curvature, we have been able to achieve similar results using the standard projection algorithm. We ran the simulations till $t = 450$ which is three times longer than what is shown in literature to capture the deviation from steady stare as a result of spurious currents. Within the inset in Figure 26, we show the kinetic energy oscillations beyond $t = 150$. We see that the deviation from steady state happens at well after $t = 300$, i.e., more than twice the characteristic time scale of viscous damping. The standard projection scheme shown by Torres and Brackbill [30] exhibits this behaviour even prior to achieving steady state.

3.4.4 Interface Merging Test

The previous oscillating drop test has been modified to capture the merging of two interfaces. Instead of one single stretched fluid element, we place two identical elements next to each other as shown in Figure 27. During the first oscillation, the interfaces will run into each other and induce merging. Subsequent surface tension effects leads to further oscillations similar to the previous problem. To make the problem more challenging, the densities of the gas and liquid phases are chosen to be 10^{-3} and 1, respectively.

The position of the interfaces of the two bubbles and their merging behaviour is shown in Figure 28. The simulations show that the CSF method with level set approach is able to capture the complex dynamics of bubble coalescence. We also performed the same test with the fluid properties interchanged to capture the merging of two stretched liquid drops. While the steady state shape in both cases is the same, merging of two liquid drops exhibits high amplitude oscillations because it can easily displace the surrounding low density fluid. On the other hand, deformations of bubbles in liquid are damped faster. The observation of

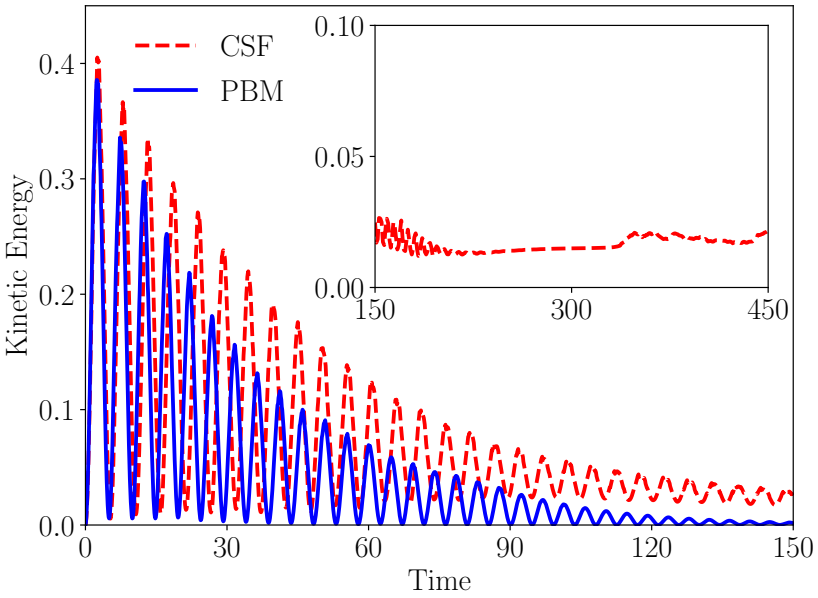


Figure 26. Kinetic energy variations using the CSF and PBM methods.

kinetic energy variations in time is shown in Figs. 30 and 31 indicate this effect quantitatively. As the frequency of oscillations is primarily governed by the surface tension, it is approximately the same between both of these tests.

It needs to be noted that although the pressure boundary method outperforms the CSF method in terms of spurious currents, it is challenging to apply for cases involving topological changes of the interface such as the present test. We tested it for the merging of two drops with a density ratio of 100. However, once the interfaces come in close proximity of each other, the level set function becomes non-differentiable and ultimately results in a singularity for curvature. While the same problem persists in the CSF method too, it is rather insignificant. The reason is that because the surface tension is transformed into a distributed body force, when two interfaces are close to each other, they have a natural tendency to merge. The same effect is produced by a high interface curvature calculated numerically. On the contrary, the PBM method requires accurate interface curvature in the application of boundary condition, which makes the scheme unstable near discontinuities. [31, 32] have shown ways to calculate local curvature using the level set function in such situations. The PBM for surface tension treatment is an active research area and there exists an opportunity to employ better curvature calculation algorithms. One of the other challenges of the PBM method is that it requires the solution of three elliptic partial differential equations. It is expected to benefit greatly from the multigrid implementation shown earlier. Currently, the PBM method in Quilt is under development to address both of these challenges.

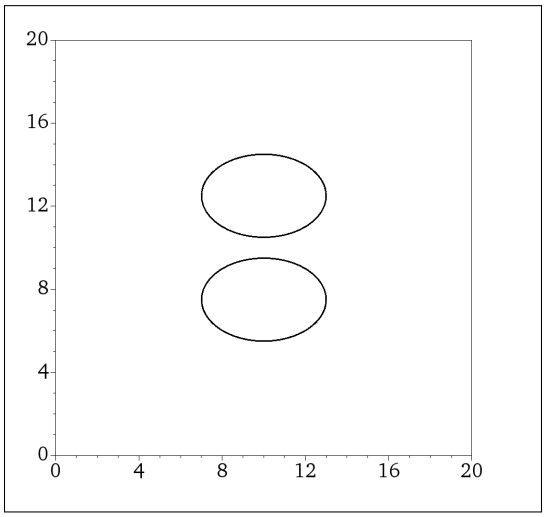
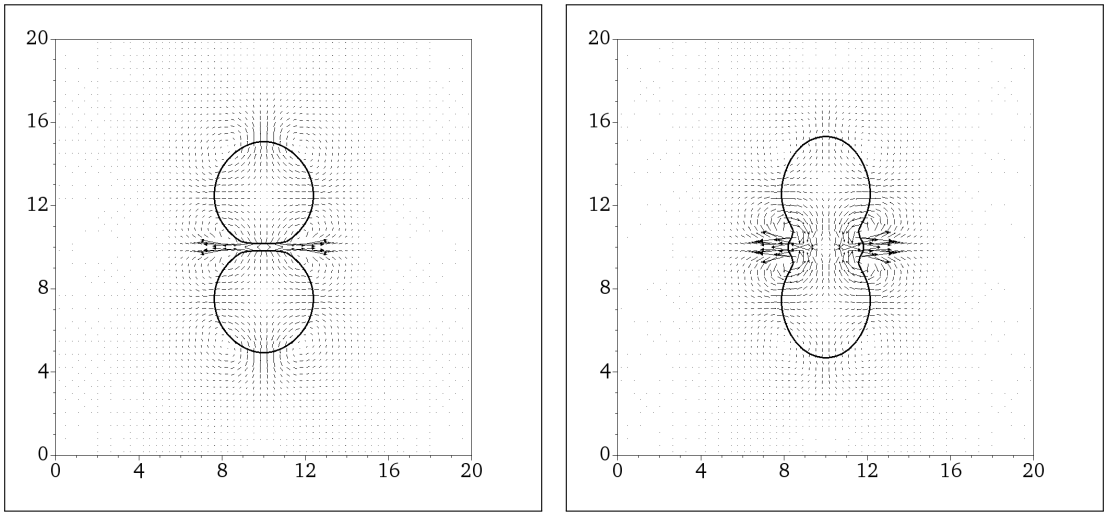
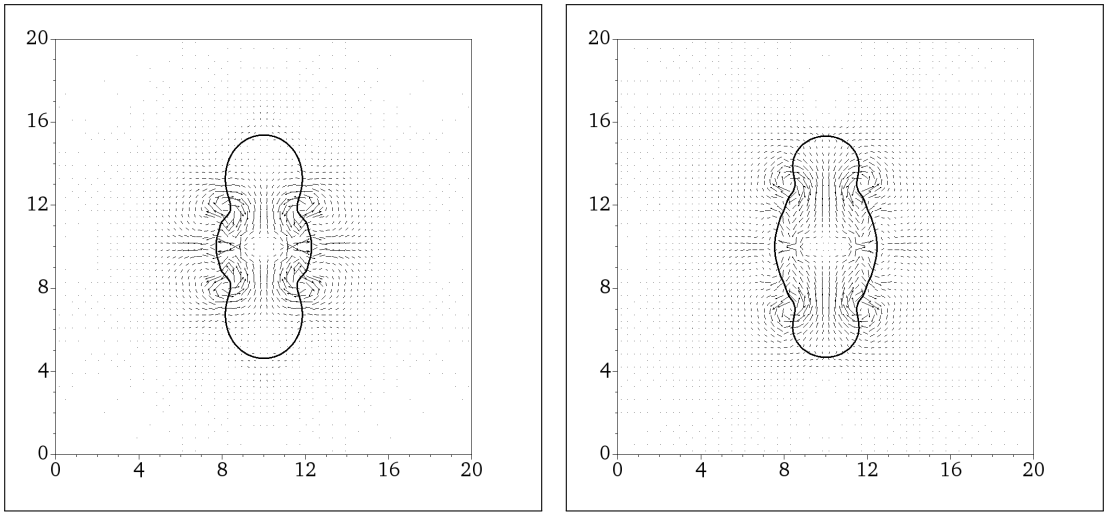


Figure 27. Initial position of the interface (bubbles and drops) for merging test.

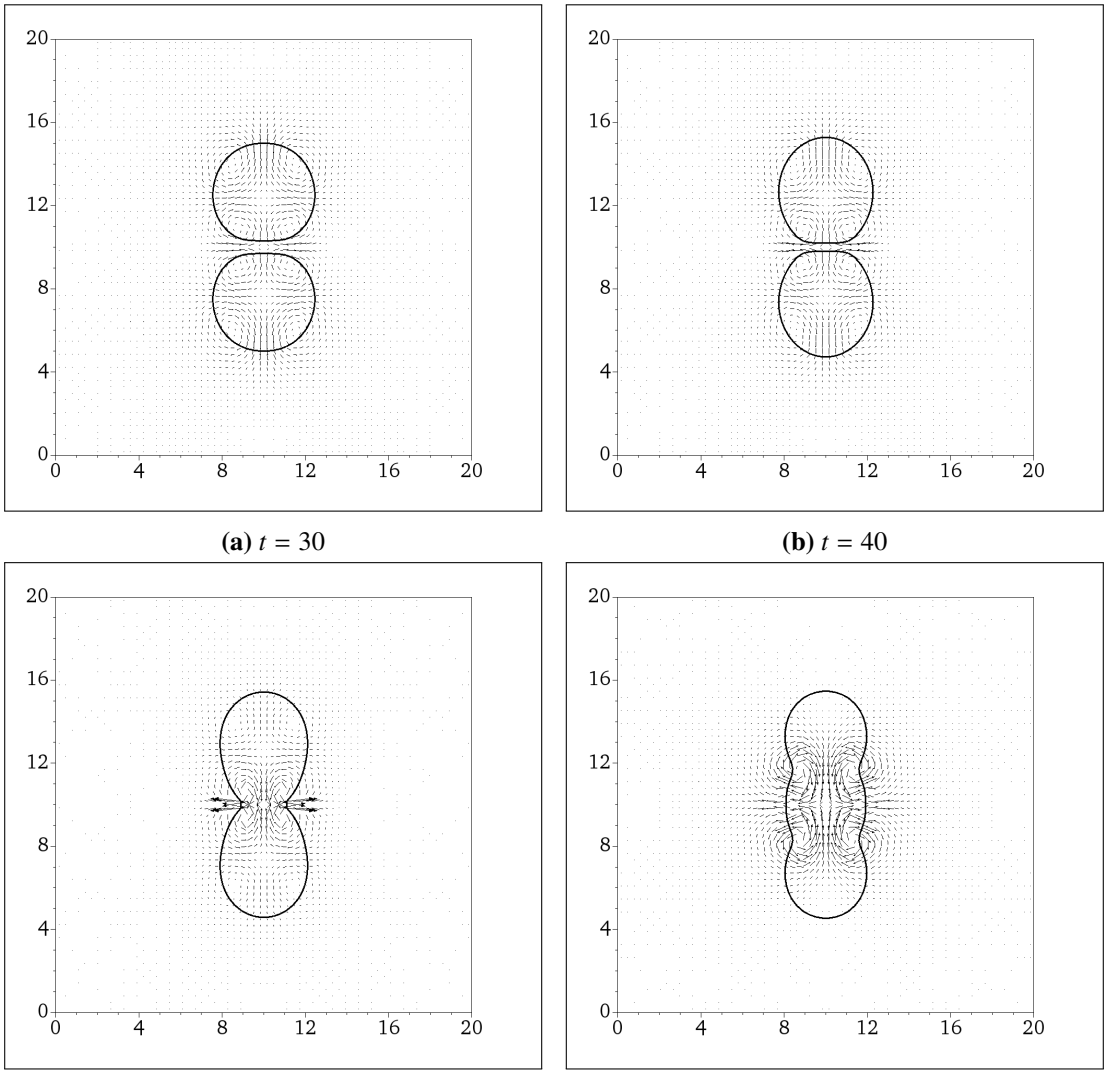


(a) $t = 30$ **(b)** $t = 40$



(c) $t = 50$ **(d)** $t = 60$

Figure 28. Flow field and interface locations for merging of two stretched drops in gas.



(a) $t = 30$ **(b)** $t = 40$

(c) $t = 50$ **(d)** $t = 60$

Figure 29. Flow field and interface locations for merging of two stretched bubbles in liquid.

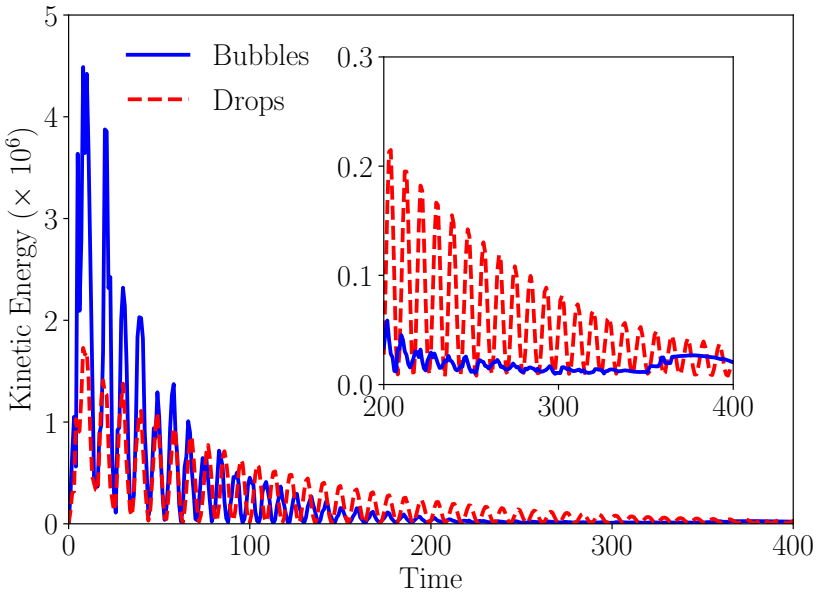


Figure 30. Kinetic energy variations for the merging test using the CSF method for surface tension.

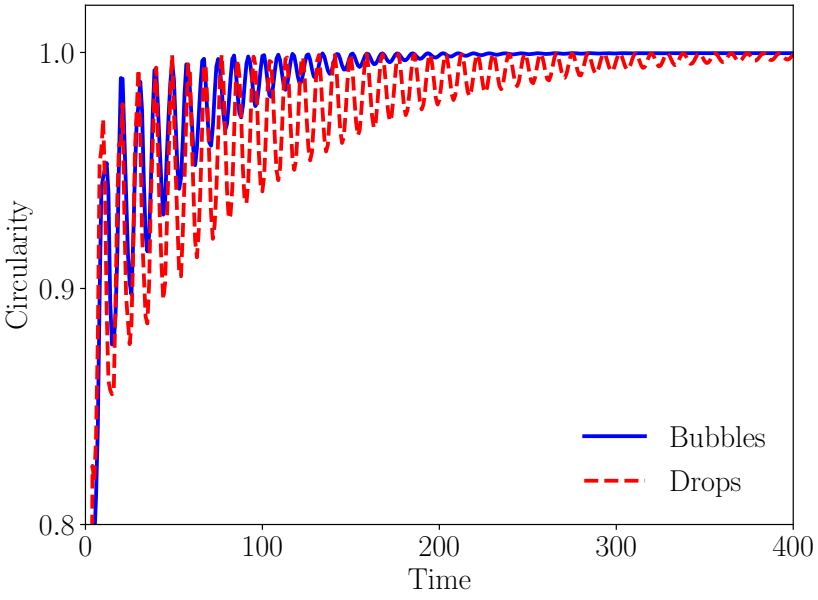


Figure 31. Circularity of the fluid elements undergoing merging and subsequent oscillations. CSF method is used to model surface tension forces.

4. Results and Discussion

In this section, we present the numerical simulations, comparison experimental data, and discuss the results in the context of identifying the critical parameters for transport in bubble column reactors. First, we perform simulations of a single gas bubble rising in a liquid column at various conditions that allows direct validation against published experimental results. Then we use the validated model to simulate the properties of bubble rise in molten salts and compare against bubbles in water. After the single bubble simulations, we progress to a bubble column with a continuous injection of bubbles at the bottom of a column and with a free surface at the top. We analyze the bubble column simulation results to provide quantitative measures of the influence of the various parameters such as bubble size and injection frequency.

4.1 Single rising bubble simulations

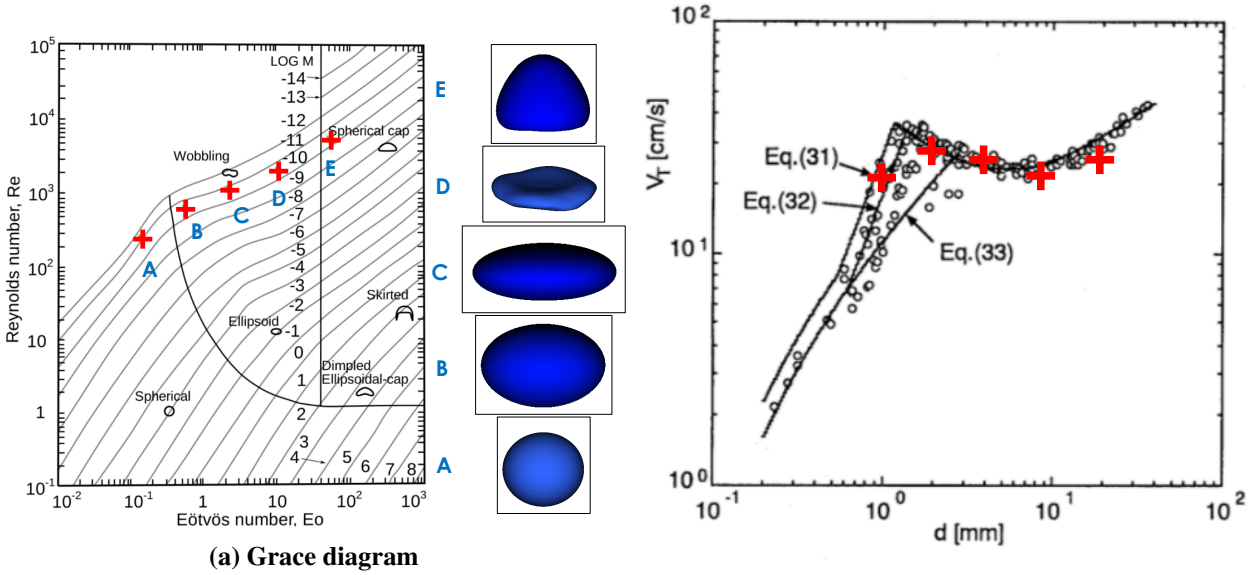
A single bubble rising in an infinitely tall column provides a reliable configuration for providing several fundamental properties of the system that serves multiple purposes. Firstly, it provides a good validation for the simulation against experimental results. Secondly, simulation of the single bubble system provides properties that will serve as a baseline reference for comparison against the multi-bubble systems with complex turbulent flow.

Tomiyama *et al.* [33] have presented results for the terminal velocities and drag coefficients of single bubbles under varying conditions, along with empirical correlations. Here, we report on the numerical simulations of a single bubble performed using Quilt and its validation against the experimental results reported in Ref. [33]. The fundamental quantities and non-dimensional numbers that affect the terminal velocity and shape of a gas bubble have been explained by Grace [34] with the help of the Grace diagram. The terminal properties of a rising bubble are governed by the fluid properties, namely viscosity, density, surface tension, and the forces due to buoyancy and drag. The drag force is especially a strong function of the shape of the bubble. Hence there is a relationship between the terminal velocity and the shape of the bubble. The governing non-dimensional numbers are the Reynolds number (Re), the Eotvos number (Eo), and the Morton number (M). Single bubble simulations were performed using Quilt by varying the bubble sizes and then measuring its terminal velocity. The bubble sizes and the terminal velocities are listed in Table 3. The simulation results are also shown by placing them in the context of the Grace diagram and compared with experimental results in Figure 32. Figure 32 shows that the validation covers bubble ranging from spherical (A) and wobbling (B, C, and D) to a spherical cap (E). The measured velocities compare well with the experimental results also shown in Figure 32. This provides us the validation needed to then obtain the results for systems other than air-water by using respective properties of the gas and liquids.

The terminal shape of the bubble also provides a measure of the surface area of the interface between the gas and liquid. The pyrolysis chemical conversion is primarily occurring on the bubble interface and therefore, there is every reason to believe that increasing this surface area will also lead to higher chemical reaction rates. A spherical shape has the lowest surface area for a given volume. Any deviation from sphericity, either steady or unsteady, such as due to wobbling motion, indicates a larger interfacial surface area. The sphericity of the bubble is defined as the ratio of the surface area of a sphere of the same volume to the actual surface area of the bubble. We study the effect of liquid properties on the sphericity of a single rising bubble. Figure 33 shows the time varying sphericity and the terminal velocity of the bubble using

Table 3. Rise velocity and dimensionless numbers for various bubble sizes. Mo for all cases is 1.58×10^{-11}

Diameter (m)	Rise vel. (m/s)	Re	Eo
1.0×10^{-3}	0.216	243	0.134
2.0×10^{-3}	0.281	632	0.537
4.0×10^{-3}	0.257	1157	2.15
8.8×10^{-3}	0.216	2138	10.4
2.0×10^{-2}	0.257	5775	53.7



(a) Grace diagram

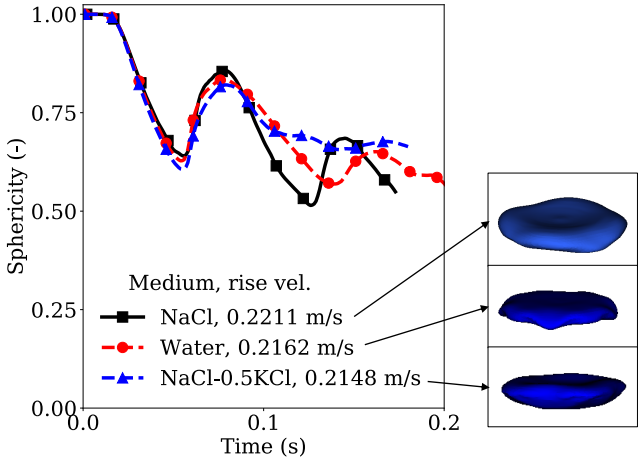
(b) Bubble rise velocity

Figure 32. Validation of simulation results for a single bubble rising in water column. Dimensionless correlations shown on the Grace diagram (a), and rise velocity (b). Corresponding bubble shapes are also shown.

different mixtures including water, molten sodium chloride (NaCl) and a mixture of molten sodium and potassium chlorides (NaCl-0.5KCl). It is seen from Figure 33 that the bubble dynamics are different in molten salt compared to water. The NaCl-0.5KCl mixture has the least unsteadiness and surface area generated. In comparison, a bubble in molten NaCl has the largest unsteadiness that leads to more surface area. The bubble rise velocities in all cases are close to each other. While these single bubble results point to the ability of different molten medium in increasing the interfacial area, the effective area in a multi bubble column also depends on the flow patterns and the bubble interaction.

4.2 Experimental measurements of holdup

The gas holdup in a bubble column was measured experimentally as described below. The holdup is defined as the volume of injected gas that is held within the liquid column, as a percentage of the original



Medium	Re	Eo	M
NaCl	3.7×10^{3}	11.1	2.2×10^{-12}
Water	2.1×10^{3}	10.4	1.6×10^{-11}
NaCl-0.5KCl	3.6×10^{3}	12.8	3.4×10^{-12}

Figure 33. Sphericity of single bubble rising in various molten media. In addition to rise velocity, dynamics of bubble deformation in molten salts (NaCl and NaCl-0.5KCl) are similar to that in water column. The non-dimensional numbers (and surface tension to density ratio) governing the bubble dynamics in all these media are almost the same.

volume of the liquid column. A higher holdup of gas within the column is desired since that is one of the many factors that can increase the conversion rate. The apparatus used to experimentally measure the gas holdup consisted of a tube with 30mm outer diameter and 25mm inner diameter. The tube was filled with water to a height of 24 inches. Nitrogen gas was injected at the base of the liquid column using a much smaller diameter tubing that was inserted into the column. The flow rate of nitrogen was varied and at each flow rate the height of the column was measured. The increase in height of the liquid column while gas was bubbled through it gives the holdup within the column at that injection rate. Two sets of measurements were made with the the gas injection close to the center of the outer tube or with the injection near the outer wall.

Figure 34 shows the holdup as a function of the nitrogen flow rate for both these cases. The notable effects found in the experiments were as follows. (i) Compared to the hold up at 10 and 30 liters per hour (lph), the holdup at 15 lph was slightly lower. Visual observations points to the possibility that the holdup is adversely affected at that flow rate by bubble phenomena including coalescence. (ii) The holdup is influenced by the location of injection, either centered in the tube or close to the wall. It is suspected that the circulation of the fluid within the column affects the holdup and the circulation is influenced by the injection location. (iii) The holdup is higher for the centered injection except at very high gas flow rates.

Next we perform DNS of bubbly flow in a liquid column to understand the multiphase physical processes and identify the parameters that are crucial for reactor design and scaleup.

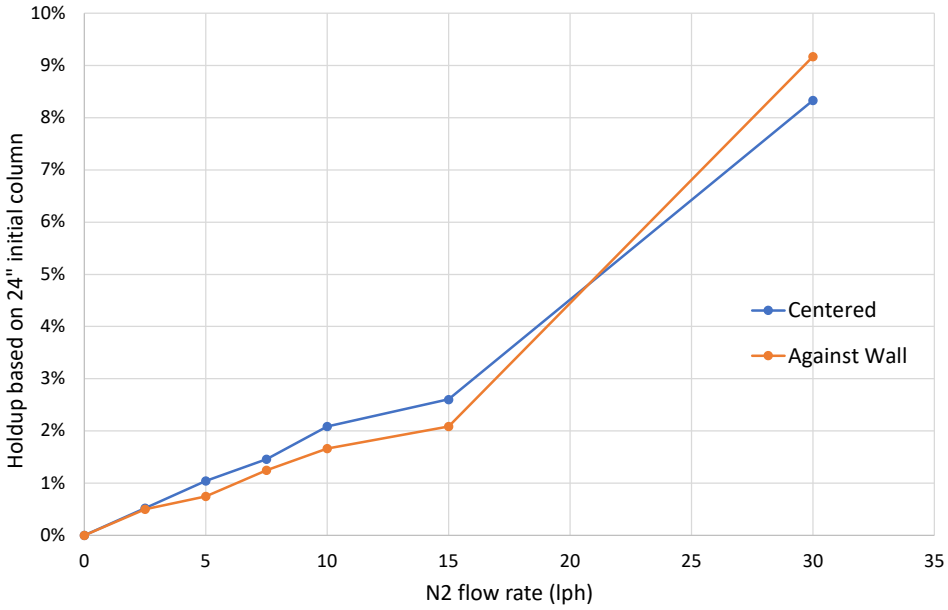


Figure 34. Experimentally measured gas holdup as a function of gas flow rate.

Table 4. Bubble column DNS case descriptions

Case name	Bubble diameter (mm)	Injection frequency (ms)	Gas flow rate (lph)
AS1	4.4	10	16.1
AS2	4.4	20	8.0
AS3	4.4	30	5.4
AS6	4.4	60	2.7
AM3	6.6	30	18.1
AM6	6.6	60	9.0
AL6	8.8	60	21.4

4.3 DNS of bubble column

4.3.1 Simulation Configuration

Direct numerical simulations of the bubble column were performed using the Quilt software. The simulations were performed using the GPU-accelerated OLCF Summit system. The simulation domain was discretized using an equi-spaced cartesian mesh with a resolution of 0.25mm. The mesh size was 128 cells in the transverse directions, and 512 cells in the vertical column direction. The resulting domain size was 32mm in the transverse directions, and 128mm in the vertical column direction. The domain was decomposed using 256 MPI ranks for parallel processing. A parametric study was performed by varying the injected bubble diameter and injection frequency. The bubble diameters and the corresponding case nomenclature are: 4.4mm (small, AS), 6.6mm (medium, AM) and 8.8mm (large, AL). Also, the bubble injection frequencies were every 1000, 3000, and 6000 timesteps. A fixed timestep size of 10us had been chosen based on the stability criterion. Based on the timestep size, the bubbles were injected every 10, 20, 30, and 60ms and the corresponding cases have the numbers 1, 2, 3, and 6 appended to the case names. For example, case AM6 has the medium 6.6mm bubble diameter and the injection frequency is every 60ms. Seven cases were simulated, namely AS1, AS2, AS3, AS6, AM3, AM6, and AL6. Other cases such as AL3 and AM2 were attempted. However, they did not provide meaningful interface results due to very high gas injection rate and therefore not reported. Table 4 shows the names of the cases and the corresponding injection parameters.

The DNS was initialized with a small height of air above the liquid column, separated by a free surface. The free surface is located 110mm from the bottom of the column, leaving a 18mm headroom of gas above the column. The bubble interface and the top free surface were both captured with the same level set function. This allows the eruption of the gas bubble through the free surface and the resultant fluid motion to be captured in the DNS. The boundary conditions are periodic in the two transverse directions. A free slip wall boundary condition is used in the top and bottom boundaries in the vertical column direction. The bubbles are injected 19mm above the bottom boundary of the column. The injection location in the two transverse directions was randomized to be between 40 – 60% of the transverse dimension, such that the average injection was at the middle of the domain. Based on the estimated bubble rise velocity (0.3 m/s) and height of the column between the point of injection and free surface, the flow through time for the bubble was estimated at 0.3s. The DNS was performed for a physical time of 1.5s, or 5 flow through times. The first two flow through times were ignored from statistical analysis to allow for the transient development from the initial condition. All statistical analysis and averages are performed for the final

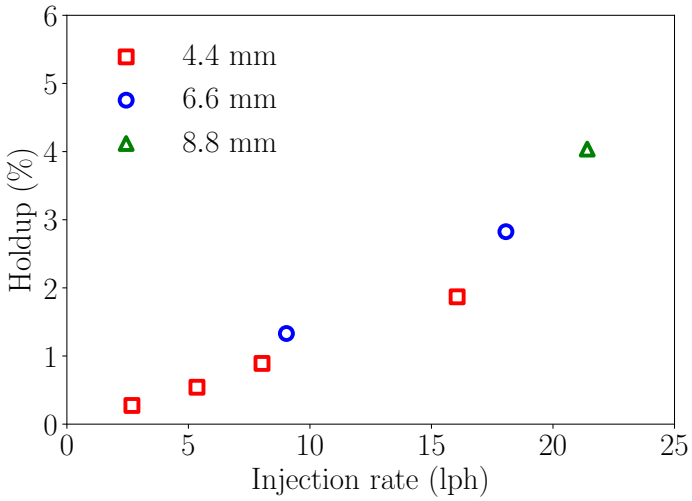


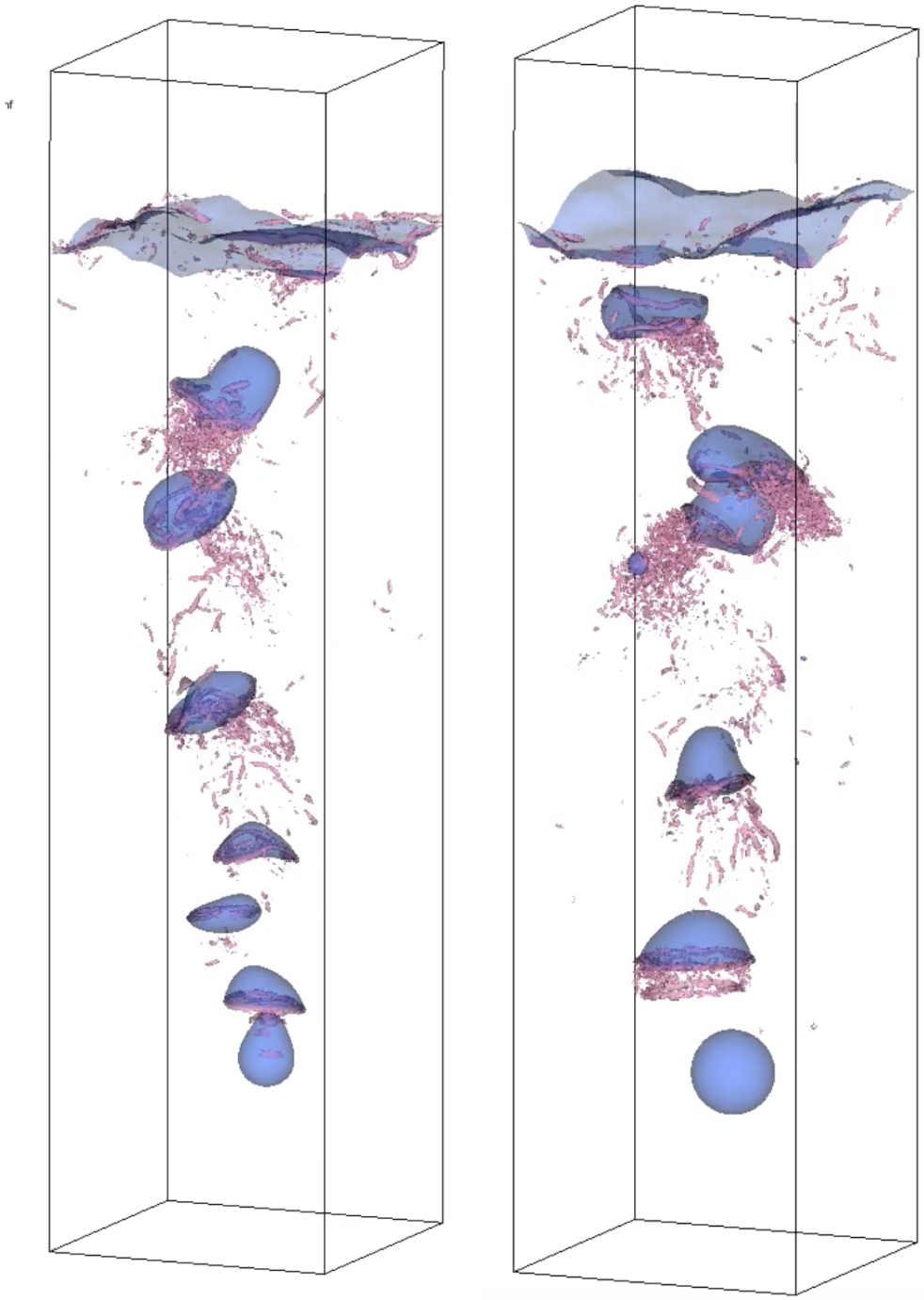
Figure 35. Gas holdup computed using simulation results. Hold up is calculated as the average fractional gas volume within the liquid column over the last three flow through times.

three flow through times. Firstly, the gas holdup is analyzed for all cases and reported in Figure 35. Gas holdup is calculated as the fraction of average gas volume within the liquid column over the final three flow through times. A good agreement is seen between the simulation and experimental findings. Interestingly, the hold up appears to increase much faster with respect to injection rate beyond 15 lph. This was also observed by experiments as outlined earlier, potentially due to larger bubble sizes resulting from coalescence. Large bubbles experience more drag and rise slowly through the liquid medium.

The instantaneous solutions were visualized, animated, and studied. Figure 36 shows two sample instantaneous visualizations of the bubble column and flow field. The two cases shown, AM3 and AL6, have comparable gas flow rates. However, AL6 has a larger diameter and a smaller injection frequency. It was seen that case AL6 is more turbulent and causes a larger disruption/stirring of the free surface compared to AM3. Also, AM3 bubbles are less likely to coalesce compared to AL6. The visualization and qualitative analysis of the time evolving flow field of the bubble column provides great physics insights. However, the greatest strength of the DNS comes from performing statistical analysis and comparing the mean flow properties among the various cases. Such a statistical analysis provides quantitative results to inform model development and to identify optimization opportunities. In the next section we report and discuss the statistical averages computed from the final three flow through times of the DNS for all seven cases.

4.3.2 Flow statistics in bubble column

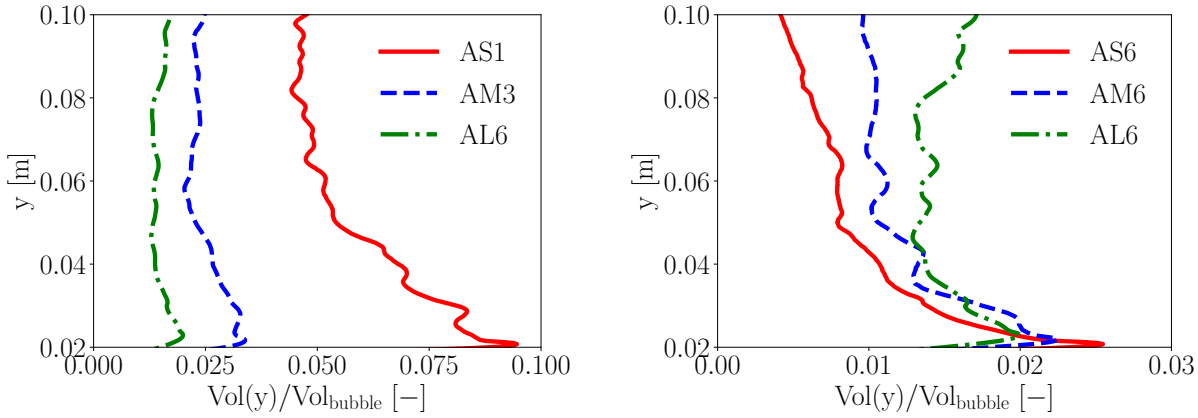
Figure 37 shows the gas volume distribution along the column height. The volume is normalized by the volume of a single bubble that is injected at the bottom of the column. In Figure 37(a), we see that the smaller bubbles are injected more frequently to attain a Similar volume injection rate. The corresponding time averaged gas volume in the liquid column is linearly proportional to the injection frequency. However,



(a) Case AM3

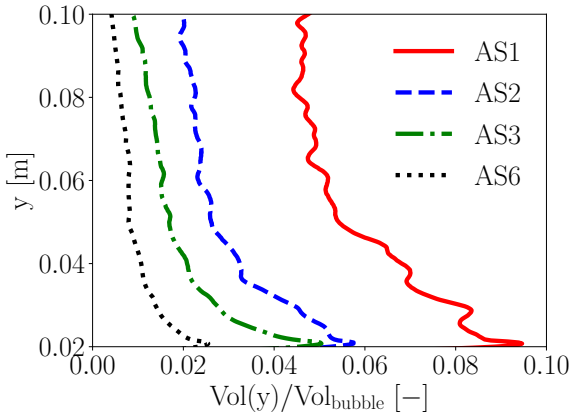
(b) Case AL6

Figure 36. Instantaneous snapshot from the DNS of the bubble column showing the gas liquid interface (blue). Also, the vortical flow structures are visualized in pink using the Q-criterion for vorticity.



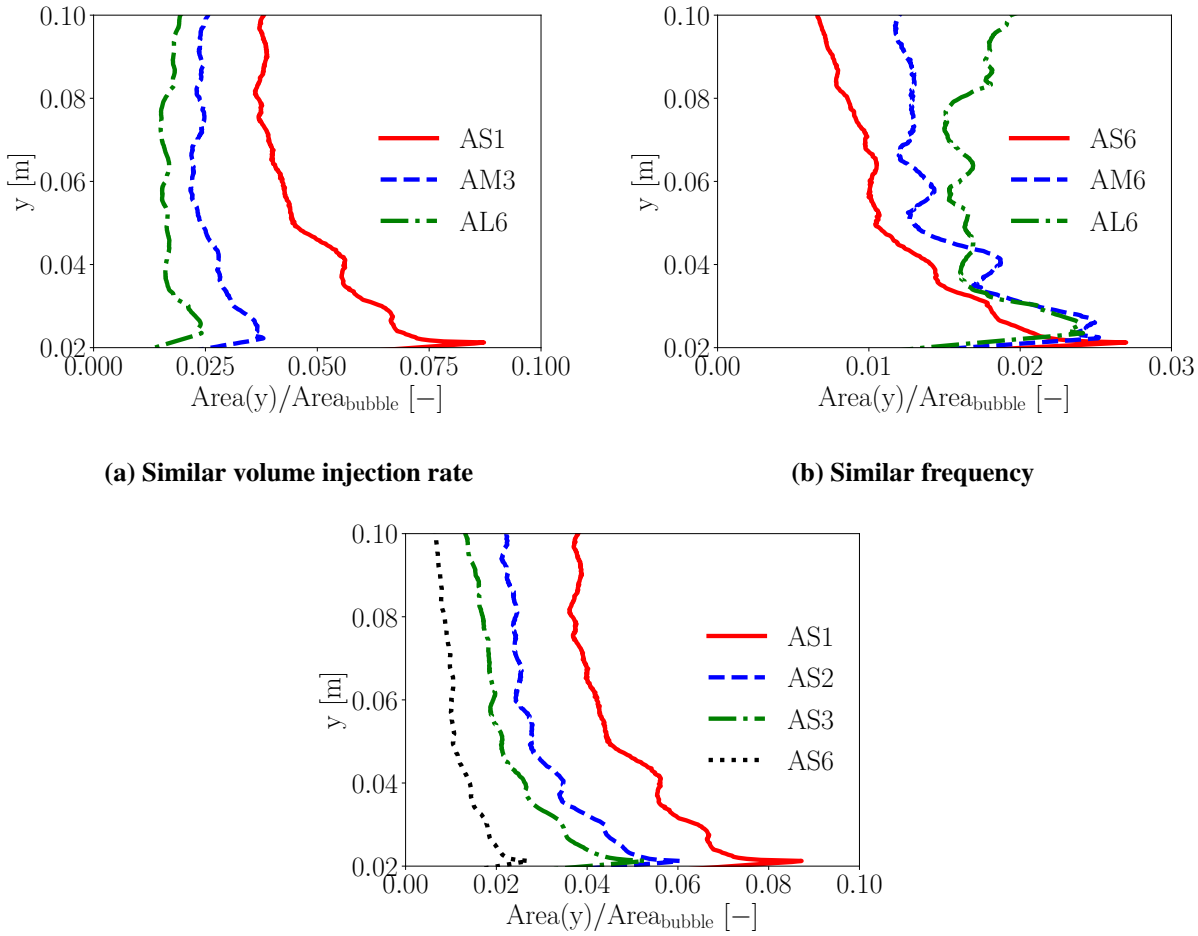
(a) Similar volume injection rate

(b) Similar frequency



(c) Similar bubble size

Figure 37. Mean gas volume distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).



(a) Similar volume injection rate

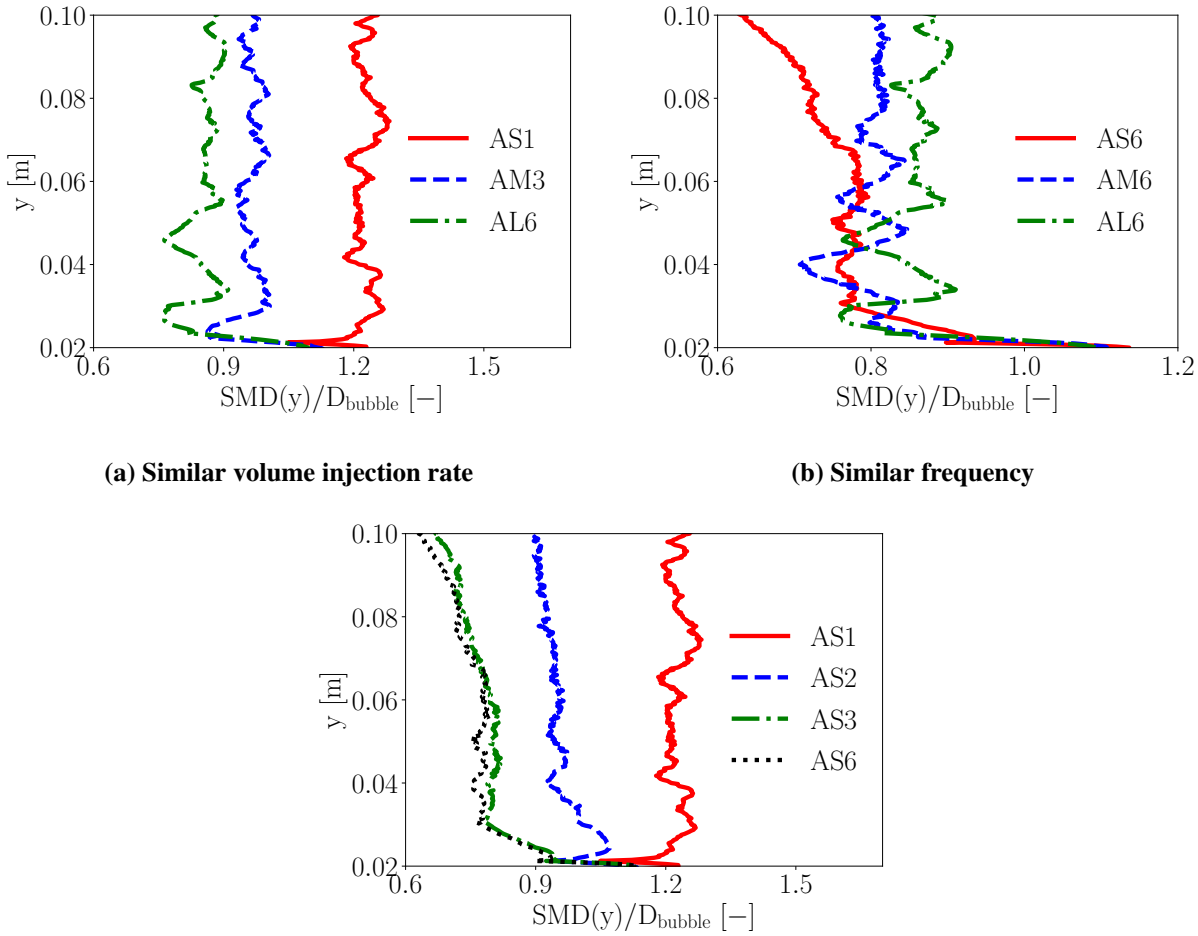
(b) Similar frequency

(c) Similar bubble size

Figure 38. Mean surface area distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).

in the case of the smallest bubbles, we see that the bubbles merge near the injection point and there is a significantly larger gas volume in this zone. In Figure 37(b), the injection frequency is kept constant while the bubble sizes are varied. The injection frequency is low enough to ensure that merging of the bubbles is not observed near the injection location. It is observed that as the bubbles rise in the column, larger bubbles exhibit more deformation, particularly just before bursting on the free surface. The effect is qualitatively observed in the visualization snapshots. The 6.6 mm and 4.4 mm diameter bubbles show almost constant volume throughout the column. In Figure 37(c), the volume distribution is shown for constant bubble size (4.4 mm diameter). The volume distributions are directly proportional to the injection frequencies.

Figure 38 shows the time averaged distribution of interface surface areas along the column height. The surface area is normalized by the surface area of the initial spherical bubble injected at the bottom of the column. It is seen that the area distribution is almost uniform throughout the column for 8.8 and 6.6 mm bubbles, while the smaller 4.4 mm bubbles show significantly high surface areas near the injection point.



(a) Similar volume injection rate

(b) Similar frequency

(c) Similar bubble size

Figure 39. Sauter mean diameter distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).

We believe this is due to the very high injection frequency which leads to merging of bubbles early on. If the injection frequency is kept constant, it is seen that the smaller bubbles show very little deformations while rising in the column. On the contrary, the larger bubbles clearly demonstrate deformations as evident from the oscillations in the surface area as a function of location in the column. If we compare the surface areas of the smallest bubbles at different injection frequencies, we see that they are proportional to the injection frequency with almost no oscillations.

The Sauter Mean Diameter (SMD) of bubbles is a quantitative measure of the deformation of a bubble or its deviation from the spherical shape. It is defined as

$$\text{SMD} = \frac{6 \times \text{Volume}}{\text{Area}}. \tag{30}$$

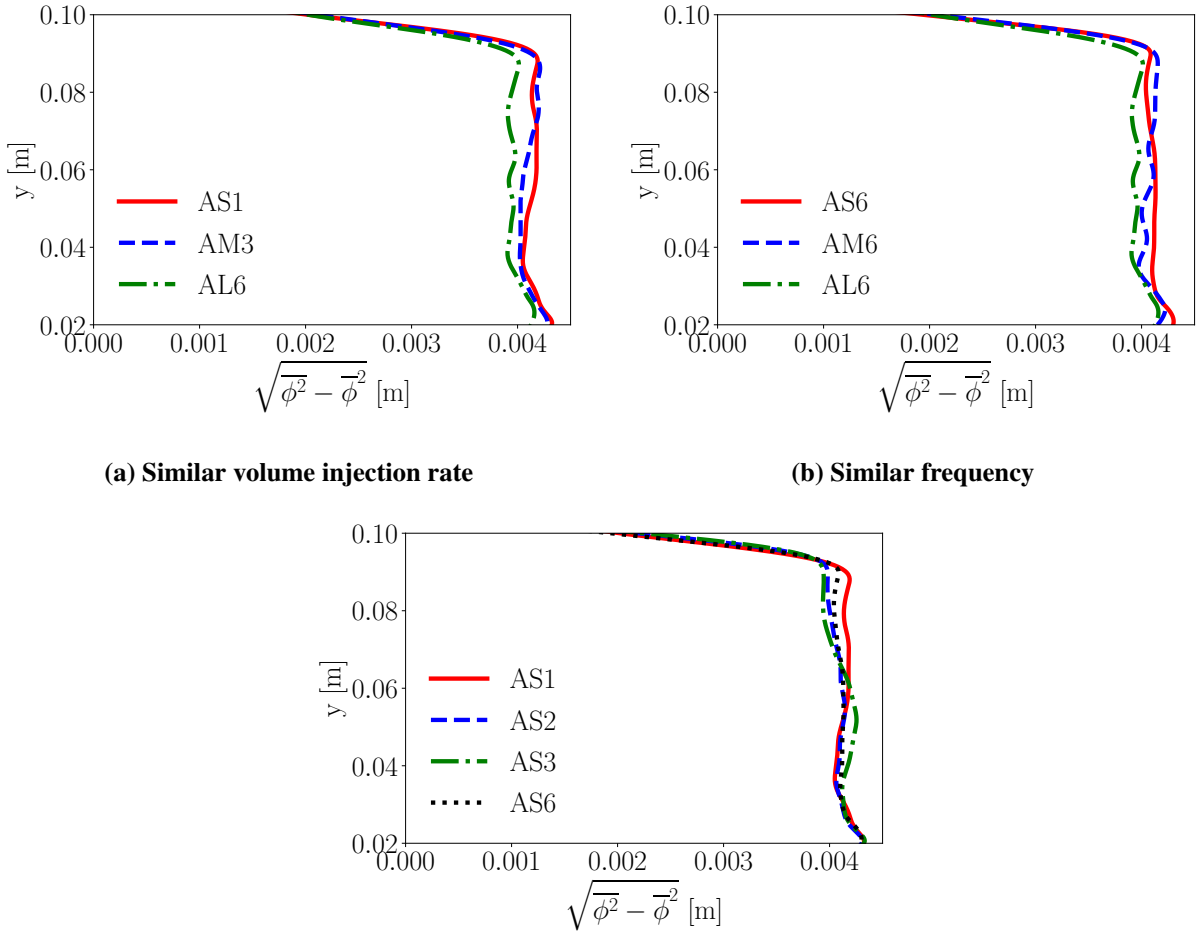
In Figure 39, the SMD is normalized with the diameter of the injected spherical bubble. In Figure 39(a), it can be seen that the normalized SMD is less than unity for 8.8 and 6.6 mm bubbles. The observation

indicates significant deformation of the bubbles as they rise in the column. The deformation is also higher for larger bubbles. At the same time, it is approximately 1.2 for the 4.4 mm bubble. Values greater than unity indicate that surface area was lost due to merging of bubbles. As noted from the previous two results, the smallest bubble has the highest injection frequency to keep the volume injection rate similar. The consequence is that subsequent bubbles are injected very close to each other and lead to merging as soon as they are injected. Another important observation is that the larger bubbles show the highest oscillation amplitudes which is also qualitatively captured in the visualization snapshots. Similar observation is made from Figure 39(b) where the SMD is compared for different bubble sizes keeping the injection frequency constant. The frequency is small enough that merging of bubbles near the injection point does not occur. All the cases have the normalized SMD less than unity, indicating severe deformation in the bubble shape while rising in the column. Oscillations in the case of the two larger bubbles are prominent, especially early on. In Figure 39(c) we compare the SMD for small bubbles at various injection frequencies. Except for the largest injection frequency, all other cases have normalized SMD less than unity and is constant throughout the column. The bubbles assume a steady state shape (which is not spherical) very quickly.

Figure 41 shows the time averaged rise velocity as function of the vertical location in the liquid column. As noted in earlier discussions, the rise velocity is strongly dependent on the bubble sizes. In Figure 41(a), it can be seen that the rise velocity reduces marginally as the bubble size increases. This is because a greater drag is experienced by larger bubbles due to a higher surface area and also more severe deformation. For small bubbles, the rise velocity is found to increase marginally with increase in the injection frequency. To analyze this effect, we look at the motion of the bubbles in the column as well as the flow fields of the surrounding liquid.

Qualitatively, the snapshots and experimental observations indicate that the bubbles follow a helical path while rising in the column. To quantitatively capture the effect, we analyze the tangential gas velocity distribution in the column. In all cases, it can be seen that the magnitude of the tangential velocity component is comparable to the rise velocity, indicating that the helical motion will have significant contribution to the residence time of the bubbles in the liquid medium. Moreover, we see from Figures 41 (a) and (b) that the tangential velocity magnitude for 4.4 mm bubble is around 0.2 m/s while that of 8.8 mm bubble is around 0.25 m/s. Comparing these values to the rise velocity in Figure 41 shows that larger bubbles have a greater lateral motion compared to smaller bubbles. The effect translates into higher residence time within the column.

It can be understood that the residence time, t_{res} of a bubble within the column can be given as L/V_y where L is the height of the column, and V_y is the rise velocity. Using this residence time, the total distance travelled by the bubble becomes $V \times t_{\text{res}}$. Thus, the ratio V/V_y can be used to evaluate the ratio of total distance to the rise of the bubble. Figure 43 shows the ratio of V/V_y. For all the cases, it is in the range of $1 - 1.5$, indicating that all the bubbles experience helical motion. It is also observed from Figures 43 (a) and (b) that the larger bubbles have a slightly higher value compared to the smaller ones. However, it can also be seen from (c) that the injection frequency has minimal effect on this quantity.

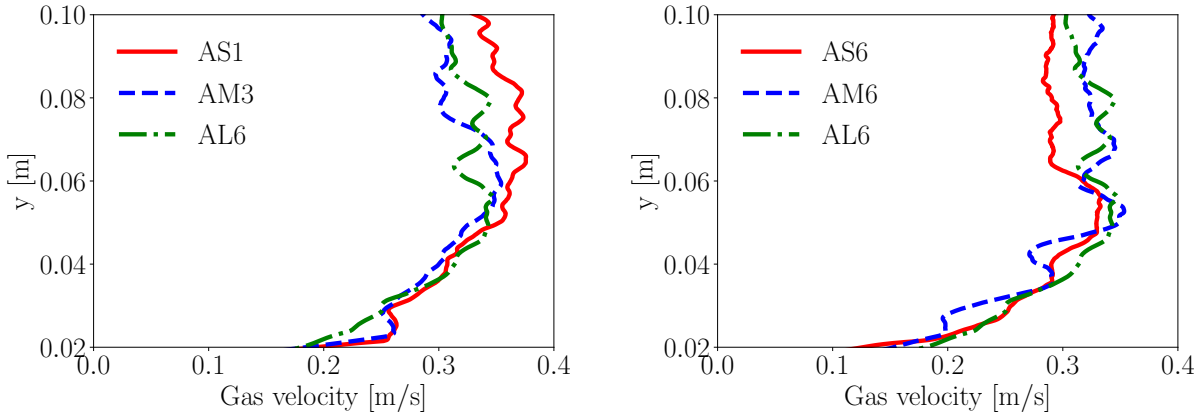


(a) Similar volume injection rate

(b) Similar frequency

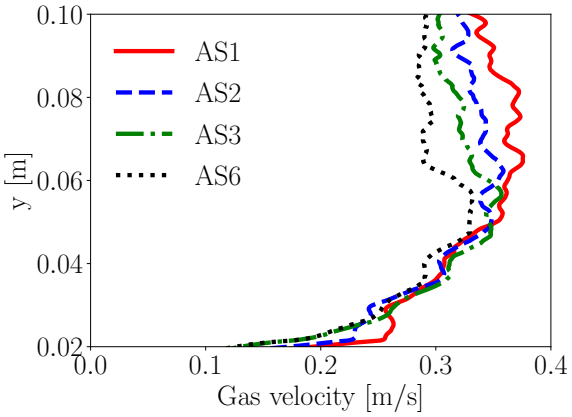
(c) Similar bubble size

Figure 40. Mean distance function distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).



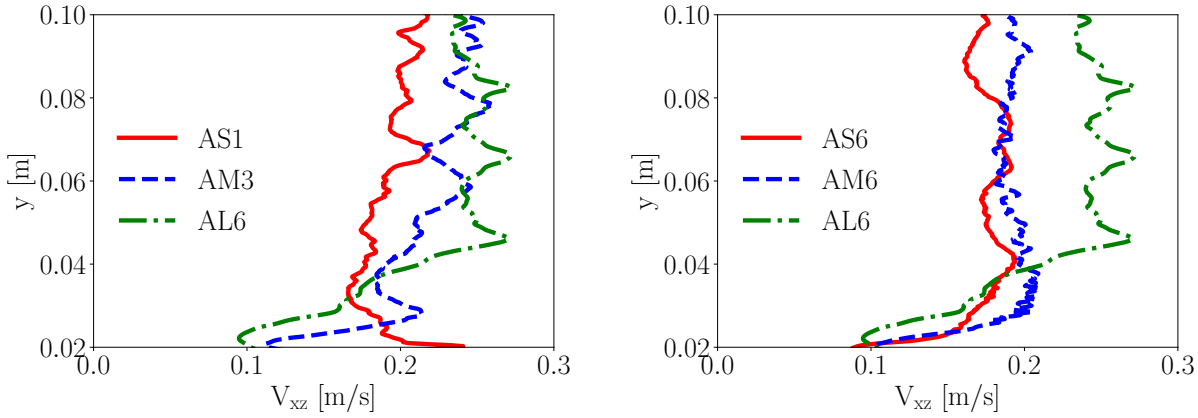
(a) Similar volume injection rate

(b) Similar frequency



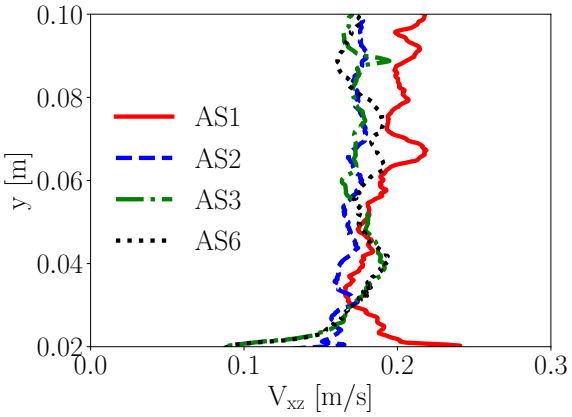
(c) Similar bubble size

Figure 41. Mean bubble rise velocity distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).



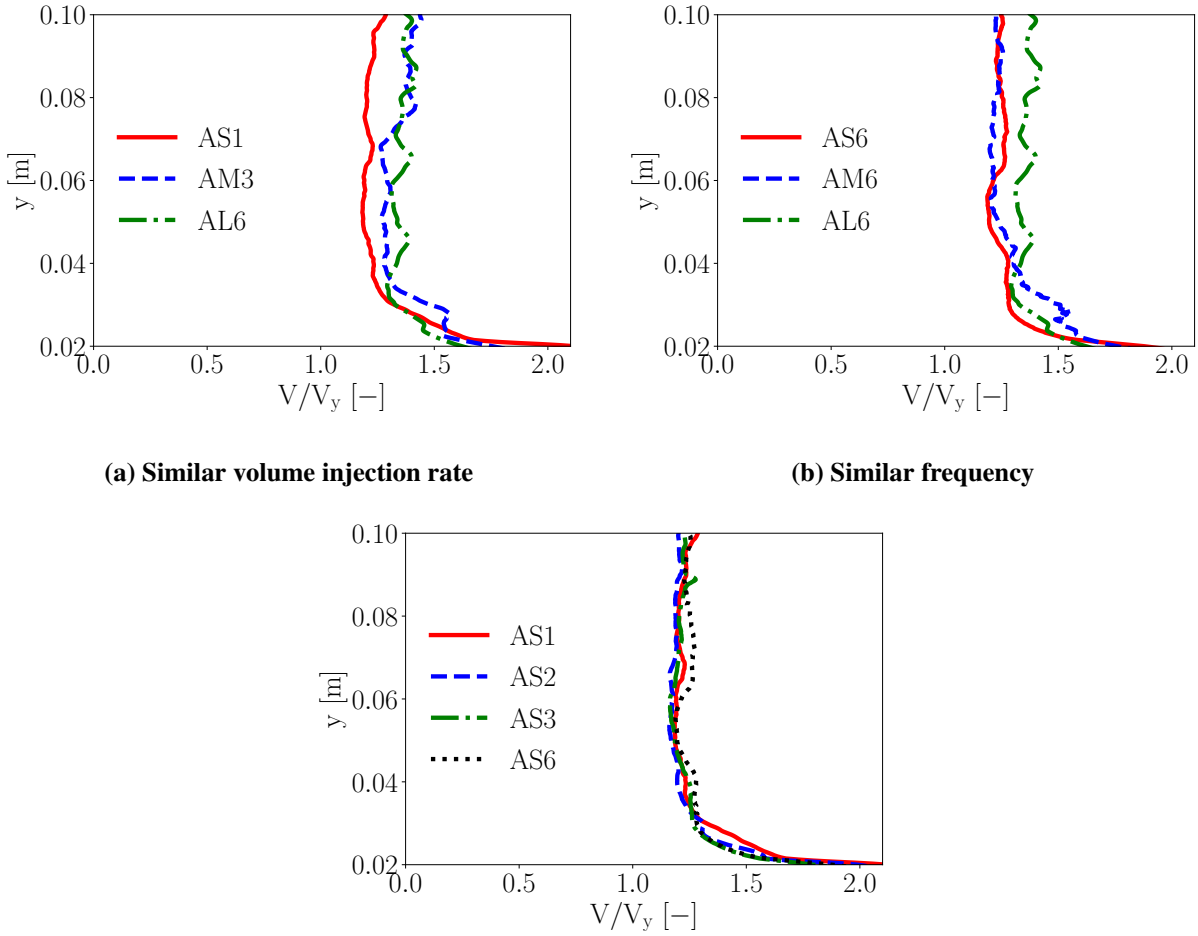
(a) Similar volume injection rate

(b) Similar frequency



(c) Similar bubble size

Figure 42. Mean gas phase tangential velocity distribution along the column height for similar volume injection rate (a), injection frequency (b) and bubble size (c).

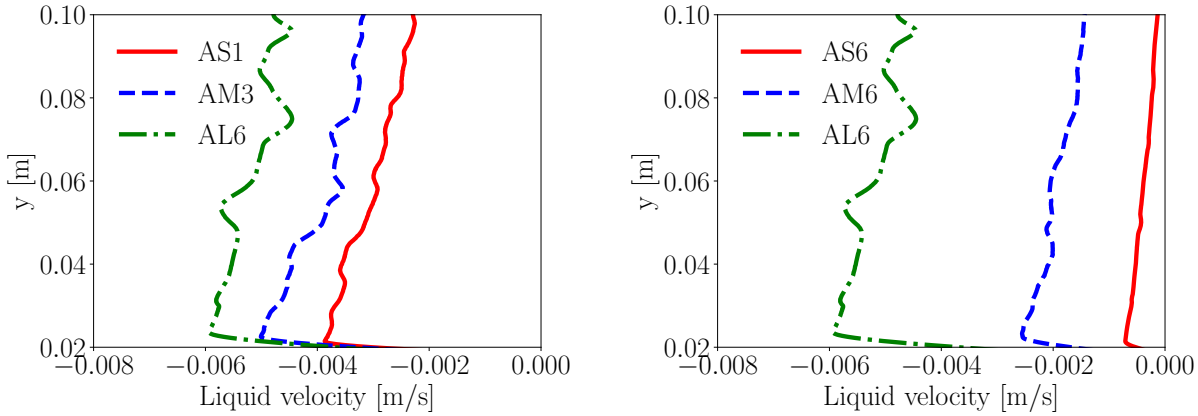


(a) Similar volume injection rate

(b) Similar frequency

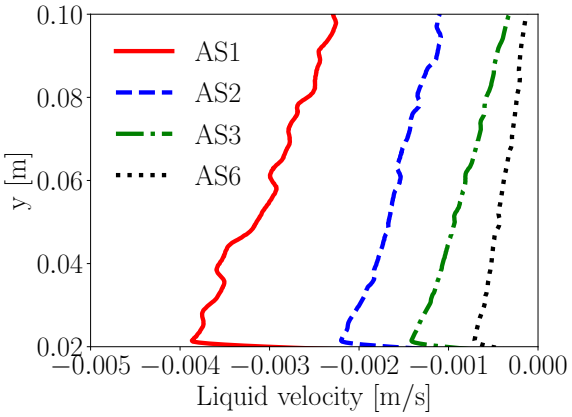
(c) Similar bubble size

Figure 43. Ratio of gas velocity magnitude to the rise velocity along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).



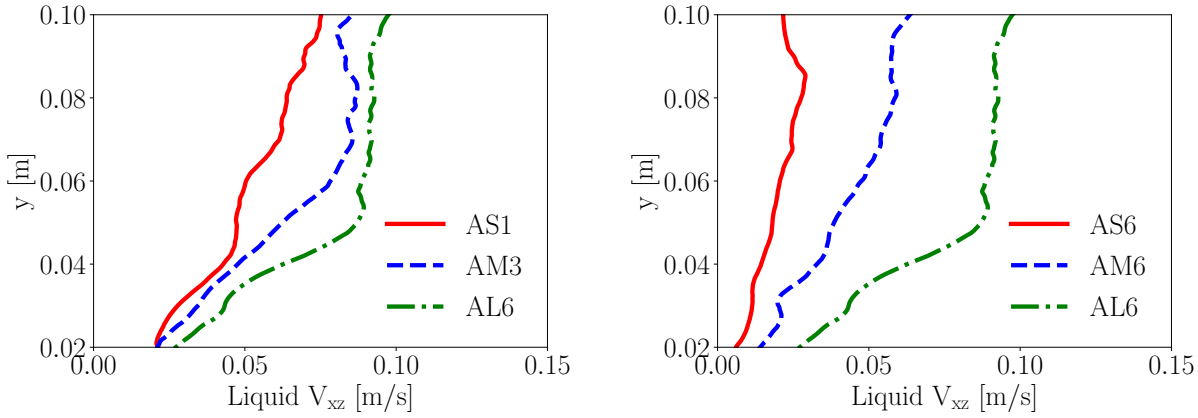
(a) Similar volume injection rate

(b) Similar frequency



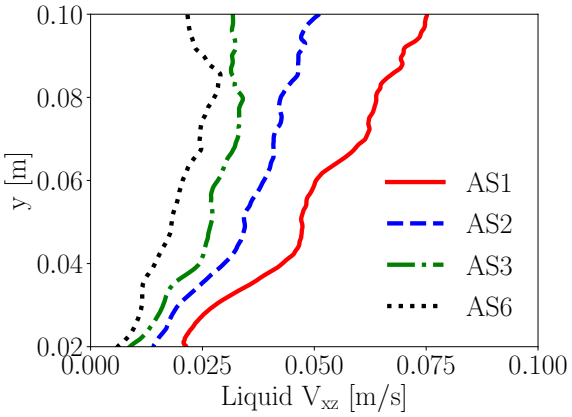
(c) Similar bubble size

Figure 44. Mean liquid velocity magnitude distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).



(a) Similar volume injection rate

(b) Similar frequency



(c) Similar bubble size

Figure 45. Mean liquid tangential velocity distribution along the column height for similar volume injection rate (a), injection frequency (b), and bubble size (c).

5. Conclusions and Future Work

Pyrolysis of natural gas to produce hydrogen and carbon provides considerable economical and emission advantages, compared to state of the art SMR. Molten salt bubble columns are a viable path to pyrolyis of natural gas. However, significant challenges remain in the design, optimization, and scale up, due to the lack of in-situ experimental measurements. High fidelity DNS, such as the work presented here, enabled by HPC provide unique scientific insights that cannot otherwise be obtained through empirical correlations and engineering scale numerical models.

This report presented the formulation and numerical methods used in Quilt for the DNS of multiphase flows. Quilt uses a level set method to capture the gas liquid interface and the associated phenomena, such as bubble deformation, merging, breakup, and interaction with free surface. The formulation and methods used for DNS were hardened for this application problem through carefully selected numerical benchmarks consisting of single phase and multiphase test problems. The software was then validated using single bubble simulations against experimental measurements and observations. The DNS methodology was then applied to simulate the bubble motion and liquid flow in a bubble column. A parametric study was performed to study the effects of varying the bubble sizes and injection frequencies. Statistical averages were obtained for key quantities of interest that measure the residence time, interfacial area and flow patterns. The results show that significant opportunities exist for improving the bubble column performance through control of gas injection characteristics.

Further work is required to improve the models, numerical method, and scale the simulations to more complex simulations of the bubble column reactor. The numerical methods need to be further improved and made robust to handle complex bubble merging scenarios, when the liquid layer between the bubbles gets thinned out. The simulation software and techniques needs to be scaled to large problem sizes to achieve simulations of much taller bubble columns within fixed wall clock times. Future work should also consider simulations of the bubble column reactor including gas kinetics in the bulk phase of the bubble and on the gas-liquid interface. Lagrangian particle tracking of the carbon particles will enable the simulation of the transport through the bubble column and behavior on the free surface.

References

[1] Robert A. Dagle, Vanessa Dagle, Mark D. Bearden, Jamelyn D. Holladay, Theodore R. Krause, and Shabbir Ahmed. An Overview of Natural Gas Conversion Technologies for Co-Production of Hydrogen and Value-Added Solid Carbon Products. Technical Report PNNL–26726, ANL–17/11, 1411934, November 2017.

[2] Stefan Schneider, Siegfried Bajohr, Frank Graf, and Thomas Kolb. State of the Art of Hydrogen Production via Pyrolysis of Natural Gas. *ChemBioEng Reviews*, 7(5):150–158, October 2020.

[3] T. Geißler, A. Abánades, A. Heinzel, K. Mehravaran, G. Müller, R.K. Rathnam, C. Rubbia, D. Salmieri, L. Stoppel, S. Stückrad, A. Weisenburger, H. Wenninger, and Th. Wetzel. Hydrogen production via methane pyrolysis in a liquid metal bubble column reactor with a packed bed. *Chemical Engineering Journal*, 299:192–200, September 2016.

[4] Xi-Bao Zhang, Wei-Cheng Yan, and Zheng-Hong Luo. CFD-PBM Simulation of Bubble Columns: Sensitivity Analysis of the Nondrag Forces. *Industrial & Engineering Chemistry Research*, 59(41):18674–18682, October 2020.

[5] Grétar Tryggvason, Ruben Scardovelli, and Stéphane Zaleski. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011.

[6] Alexandre Joel Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104):745–745, January 1968.

[7] S. Armfield and R. Street. An analysis and comparison of the time accuracy of fractional-step methods for the navier-stokes equations on staggered grids. *International Journal for Numerical Methods in Fluids*, 38(3):255–282, 2002.

[8] Joel H. Ferziger, Milovan Perić, and Robert L. Street. *Computational Methods for Fluid Dynamics*. Springer International Publishing, 2020.

[9] F. Moukalled, L. Mangani, and M. Darwish. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*, volume 113 of *Fluid Mechanics and Its Applications*. Springer International Publishing, 2016.

[10] Francis H. Harlow and J. Eddie Welch. Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. *Physics of Fluids*, 8(12):2182–2189, December 1965.

[11] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21(11):1525–1532, November 1983. Publisher: American Institute of Aeronautics and Astronautics.

[12] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, USA, 2nd edition, 2003.

[13] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A Multigrid Tutorial: Second Edition*. Society for Industrial and Applied Mathematics, USA, 2000.

[14] Pieter Wesseling. *An Introduction to Multigrid Methods*. R.T. Edwards, 2004.

[15] Damir Juric and Grétar Tryggvason. A Front-Tracking Method for Dendritic Solidification. *Journal of Computational Physics*, 123(1):127–148, January 1996.

[16] Ryo Kobayashi. Modeling and numerical simulations of dendritic crystal growth. *Physica D: Nonlinear Phenomena*, 63(3):410–423, March 1993.

[17] C. W Hirt and B. D Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, January 1981.

[18] Mark Sussman, Peter Smereka, and Stanley Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, September 1994.

[19] David Youngs. Time-Dependent Multi-material Flow with Large Fluid Distortion. In *Num. Method Fluid Dyn.*, volume 24, pages 273–285. January 1982.

[20] David J Benson. Volume of fluid interface reconstruction methods for multi-material problems. *Applied Mechanics Reviews*, 55(2):151–165, April 2002.

[21] Vimal Ramanuj and Ramanan Sankaran. High Order Anchoring and Reinitialization of Level Set Function for Simulating Interface Motion. *Journal of Scientific Computing*, 81(3):1963–1986, December 2019.

[22] Olivier Desjardins, Vincent Moureau, and Heinz Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416, September 2008.

[23] Elin Olsson and Gunilla Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1):225–246, November 2005.

[24] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, February 1996.

[25] International workshop on high-order cfd methods. `https://www1.grc.nasa.gov/research-and-engineering/hiocfd/`. Accessed: 2019-11-01.

[26] Wim M. van Rees, Anthony Leonard, D.I. Pullin, and Petros Koumoutsakos. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high reynolds numbers. *Journal of Computational Physics*, 230(8):2794–2805, 2011.

[27] Marianne M. Francois, Sharen J. Cummins, Edward D. Dendy, Douglas B. Kothe, James M. Sicilian, and Matthew W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, March 2006.

[28] Zhaoyuan Wang and Albert Y. Tong. A sharp surface tension modeling method for two-phase incompressible interfacial flows. *International Journal for Numerical Methods in Fluids*, 64(7):709–732, 2010.

[29] M. Stanic, M. Nordlund, E. M. A. Frederix, A. K. Kuczaj, and B. J. Geurts. Evaluation of oscillation-free fluid-porous interface treatments for segregated finite volume flow solvers. *Computers & Fluids*, 131:169–179, June 2016.

[30] D. J. Torres and J. U. Brackbill. The Point-Set Method: Front-Tracking without Connectivity. *Journal of Computational Physics*, 165(2):620–644, December 2000.

[31] Mathieu Coquerelle and Stéphane Glockner. A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces. *Journal of Computational Physics*, 305:838–876, January 2016.

[32] Antoine du Chene, Chohong Min, and Frederic Gibou. Second-Order Accurate Computation of Curvatures in a Level Set Framework Using Novel High-Order Reinitialization Schemes. *Journal of Scientific Computing*, 35(2-3):114–131, June 2008.

[33] Akio Tomiyama, Isao Kataoka, Iztok Zun, and Tadashi Sakaguchi. Drag Coefficients of Single Bubbles under Normal and Micro Gravity Conditions. *JSME International Journal Series B*, 41(2):472–479, 1998.

[34] J Grace. Shapes and velocities of bubbles rising in infinite liquid. *Transactions of the Institution of Chemical Engineers*, 51:116–120, 1973.