Implementing a Multiphase Flow Modelling Capability targeting Boiling in the High-Fidelity Software CHAPSim2

Bo Liu[1], Wei Wang[1], Charles Moulinec[1], Marco Colombo[2]

[1] Scientific Computing, Science and Technology Facilities Council, Daresbury WA4 4AD
[2] Department of Mechanical Engineering, The University of Sheffield, Sheffield S1 3JD

ABSTRACT

This work focuses on developing multiphase flow simulation capabilities in the Direct Numerical Simulation software CHAPSim2. The Volume of Fluid framework in conjunction with a high accurate algebraic interface capturing algorithm has been implemented. In this method, the interface is identified and represented using a continuous hyperbolic tangent function. Unlike the traditional Volume of Fluid methods that rely on geometrical interface reconstruction, this approach reconstructs the interface in an algebraic way and ensures the moving interface to be captured with geometrical faithfulness and compact thickness. The solver is systematically validated through a series of benchmark tests, including, for example, the classic Zalesak's rotation problem and a more practical case of rising bubble in quiescent fluid. In these cases, CHAPSim2 produces predictions in excellent agreement with reference/benchmarking data in capturing the moving interfaces and their deformation.

KEY WORDS: Direct Numerical Simulation, multiphase flow, Volume of Fluid, algebraic interfacing capturing

1. INTRODUCTION

Boiling heat transfer is a highly effective mechanism for heat removal from a heated surface. In the nuclear industry, boiling heat transfer is widely used, particularly in light water reactors, steam generators and heat exchangers. In water-cooled reactors, an upper limit on the allowed operating power is imposed by a crucial parameter, the Critical Heat Flux (CHF) [1]. CHF represents the threshold beyond which heat transfer becomes ineffective, increasing the risk of overheating and nuclear fuel meltdown. To ensure safety, it is necessary to prevent CHF from being reached under both normal operation conditions and postulated loss-of-coolant accidents. However, due to the complex physics that governs boiling, the prediction of the CHF still heavily relies on empiricism [2]. As a result, significant engineering design margins are implemented to guarantee reactor safety, which, in turn limits the maximum power output and efficiency of the reactors.

In the recent decades, with the rapid advancement in High Performance Computing (HPC), high fidelity simulations of multiphase boiling flows become increasingly feasible. The produced data can be used to support the development of more accurate engineering models for predicting CHF, thereby reducing uncertainties and minimising conservatism in future design of nuclear reactors.

This work aims to address the need for better Computational Fluid Dynamics (CFD) tools by developing a multiphase module within a high-order Direct Numerical Simulation (DNS) CFD software, CHAPSim2, based on the Volume of Fluid (VOF) method. To ensure high numerical accuracy, an advanced algebraic interface capturing algorithm known as Multi-dimensional Tangent of Hyperbola Interface Capturing (MTHINC) is employed, and a directional-splitting scheme is applied to update the volume fraction [3].

2. METHODOLOGY

Compared to conventional VOF approaches, the MTHINC method eliminates the need for explicit geometric reconstruction of the interface. As a result, smoothing or convolution operations that are typically required when computing the numerical fluxes are no longer necessary. This makes the MTHINC method more efficient and accurate in capturing sharp interfaces in multiphase flow simulations. The key idea of the MTHINC method is to approximate the phase indicator function $H(\mathbf{x}, t)$ using a hyperbolic tangent function (Eq.(1)), and the nterface is reconstructed based on this.

$$H(\mathbf{X}) \approx \tilde{H}(\mathbf{X}) = \frac{1}{2}\left(1 + \tanh(\beta P(\mathbf{X}))\right) \tag{1}$$

where β is an interface sharpness factor, and $P(\mathbf{X})$ is a polynomial of arbitrary order. The order of the polynomial $P(\mathbf{X})$ directly determines the accuracy of the interface reconstruction. A higher-order polynomial provides a more accurate approximation of the interface, while a lower-order polynomial may lead to a simpler but less precise representation. For instance, When $P(\mathbf{X})$ is selected as first order, the reconstruction corresponds to a piece-wise linear approximation of the interface, similar to conventional VOF methods. Conversely, when $P(\mathbf{X})$ is selected as second order, the reconstruction becomes quadratic, allowing for a more accurate and smoother representation of the interface curvature.

The VOF method is then coupled with the two-phase momentum equation by incorporating the interface physical properties and an additional force to account for the surface tension. The non-dimensional two-phase momentum equation is expressed as follows,

$$\frac{\partial u_i^*}{\partial t^*} + \frac{\partial (u_i^* u_j^*)}{\partial x_j^*} = -\frac{1}{\rho^*}\frac{\partial p^*}{\partial x_i^*} + \frac{1}{\rho^* \mathrm{Re}}\frac{\partial}{\partial x_j^*}\left[\mu^*\left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*}\right)\right] + \frac{1}{Fr^2} + \frac{1}{We}\frac{\kappa^*}{\rho^*}\frac{\partial \phi}{\partial x_i^*} \tag{2}$$

where,

$$\mathrm{Re} = \frac{\rho_0 U_0 L_0}{\mu_0},\ Fr = \frac{U_0}{\sqrt{L_0 g_i}},\ We = \frac{\rho_0 U_0^{\,2} L_0}{\sigma} \tag{3}$$

u_i^* denotes the non-dimensional velocity component, p^* the non-dimensional pressure, and ρ^* and μ^* the non-dimensional density and viscosity, respectively, normalised by the lighter phase density (ρ_0) and viscosity (μ_0). κ^* represents the non-dimensional interface curvature, and ϕ is the volume fraction. The key non-dimensional numbers include the Reynolds number (Re), the Froude number (Fr), and the Weber number (We), all defined based on the reference length scale (L_0), velocity scale (U_0), density scale (ρ_0), and viscosity scale (μ_0). Additionally, g_i represents the gravitational acceleration, and σ is the surface tension coefficient.

The equation is numerically solved in CHAPSim2 using finite different methods with an optional second or fourth order accuracy, embedded within a third order Runge-Kutta time advancing scheme.

3. NUMERICAL IMPLEMENTATION

The CHAPSim2 implementation differs from the original MTHINC method, primarily in how the unit normal vector and the Cartesian curvature of the interface are calculated. This requires evaluating the first and second order derivatives of the volume fraction field. In CHAPSim2, several finite difference schemes are available to compute these derivatives, including the second and fourth order central difference schemes as well as the fourth and sixth order compact schemes. The use of the high order schemes is expected to enhance the overall accuracy of the method.

Figure 1 shows a flowchart illustrating the implementation of the VOF method in CHAPSim2.

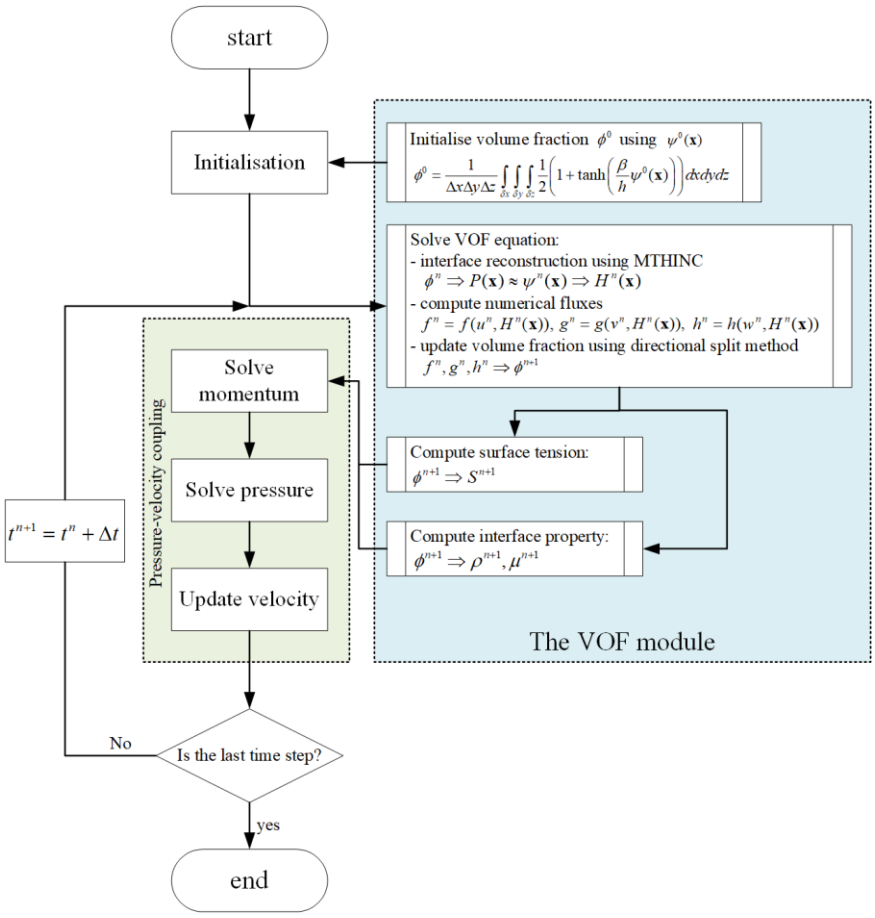


Figure 1 Implementation of the VOF method in CHAPSim2

In CHAPSim2, a 2-D domain decomposition strategy is employed to enable efficient parallelisation on CPU distributed memory machines using the library 2decomp&FFT [4], which relies on Message Passing Interface (MPI) for communication between CPU processors. This strategy involves dividing the domain in two directions, resulting in what is known as an 'pencil-like' subdomains, as shown in Figure 2.

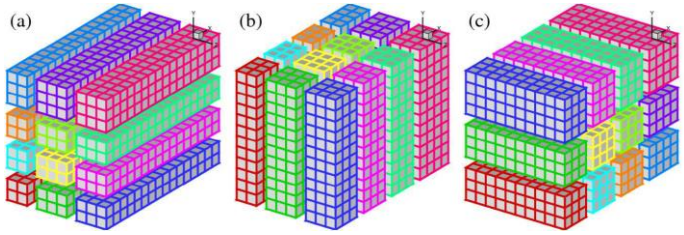


Figure 2 2-D decomposition using a 4 × 3 processor grid [4]

For example, when the decomposition is performed on the y-z plane, an x-pencil decomposition is created. In each of the x-pencil subdomains, a single processor has access to all the mesh information along x-direction, which remains non-distributed. Consequently, calculations involving derivatives that rely only on x-direction points can be performed without the need for communication between processors. When dealing with another mesh direction, such as the y-direction, the relevant data within the x-pencil subdomains are transposed to the y-pencil decomposition. This ensures that no additional MPI communication is required for the calculation of derivatives in the y-direction. Such a parallelisation

strategy is inherently compatible with the directional-splitting scheme used in the integration of the VOF equation, as each sub-step of the directional-splitting scheme involves the numerical fluxes only in one direction, making the approach efficient and well-suited for parallel execution.

4. 3. TEST CASES AND RESULTS

4.1 Zalesak's Rotation Problem

The Zalesak's rotation problem [5] is a well-documented benchmark for evaluating the accuracy of interface capturing algorithms in tracking moving interfaces under a given velocity field. The problem is defined in a [1 × 1] square domain, and the phase indicator function at $t = 0$ for a slotted disk is defined as follows,

$$H(\mathbf{x},0)=\begin{cases}1 & (x-0.5)^2+(y-0.75)^2 \le 0.15^2 \cap (|x-0.5| \ge 0.025 \cup y \ge 0.85) \\ 0 & \text{elsewhere}\end{cases} \tag{4}$$

A rotational velocity field $\mathbf{u} = (0.5 - y, x - 0.5)$ is imposed. Since CHAPSim2 does not support a 2-D mesh simulation, a 100×100 mesh is used for the x-y plane of the computational domain, and a 5-layered mesh is added in the z-direction, in which the domain is made periodic. The 5-layered mesh is the minimum in the z-direction to avoid numerical issues when using the sixth order compact scheme which relies on 5-cell stencils. A time step size of $\Delta t = \pi/200$ is used, corresponding to a maximum CFL number of 0.628 to ensure numerical stability. Results are analysed after a full revolution, corresponding to $t = 2\pi$.

Figure 3 shows the effects of several key numerical parameters by comparing the shape of the disk after a full revolution with its initial state (the reference). Figure 3(a) shows that employing linear reconstruction (i.e., using the first order of $P(\mathbf{X})$) leads to noticeable lagging in the rotation direction at the outer sharp corners of the disk. However, this effect is significantly reduced when quadratic reconstruction (i.e., using the second order of $P(\mathbf{X})$) is used, where the geometry closely follows the reference and remains symmetric. Given the considerable increase in computational complexity associated with using orders higher than the second order of $P(\mathbf{X})$, only the first and second orders are considered in this work. Figure 3(b) shows the effect of the numerical schemes used in gradient calculations. The effect is overall very small, differences only appearing near the sharp corners of the disk where the higher order schemes produce slightly better predictions.

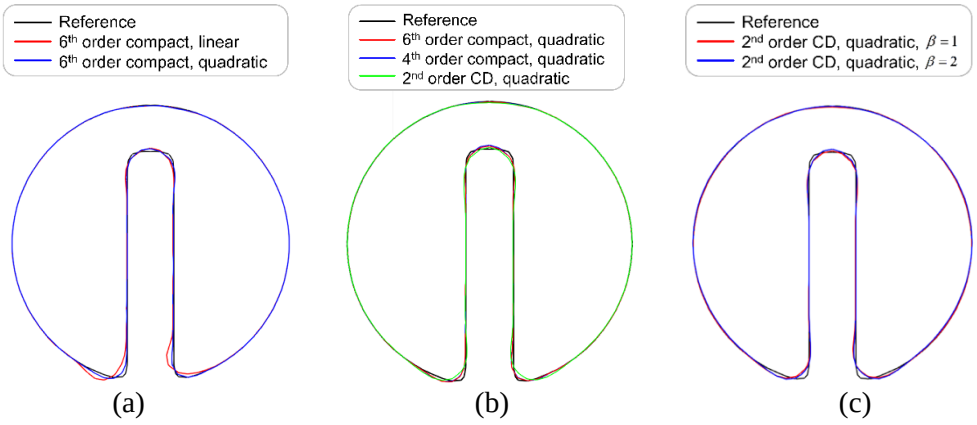


Figure 3 Effect of numerical parameters on the interface shape (represented by $\phi = 0.5$) at $t = 2\pi$ of the 2-D Zalesak's rotation problem. Investigated parameters include: (a) the order of $P(\mathbf{X})$, (b) the discretisation scheme, and (c) the interface sharpness factor

To assess the accuracy of the CHAPSim2 implementation of the method, simulations are conducted over a series of mesh resolutions, ranging from 32×32 to 256×256. The numerical errors are evaluated using the L_1-error norm, which is calculated using the following expression,

$$e = \frac{\sum_{i=1,j=1}^{N} \delta V_{i,j} \cdot \left| \phi_{i,j} - \phi_{i,j}^{0} \right|}{\sum_{i=1,j=1}^{N} \delta V_{i,j}} \tag{5}$$

where $\phi_{i,j}$ and $\phi^{0}_{i,j}$ represent the numerical result of the volume fraction at $t = 2\pi$ and the initial volume fraction of cell (i, j), respectively. Figure 4 shows the calculated L_1-error norm and its mesh convergence rate. As the mesh density increases, the convergence rate of the L_1-error norm falls between first and second order. This is consistent with the observations in Ii and co-authors' study [3]. Furthermore, using a higher order scheme (e.g. the sixth order compact scheme), reduces the absolute magnitude of the error but does not alter its convergence rate.

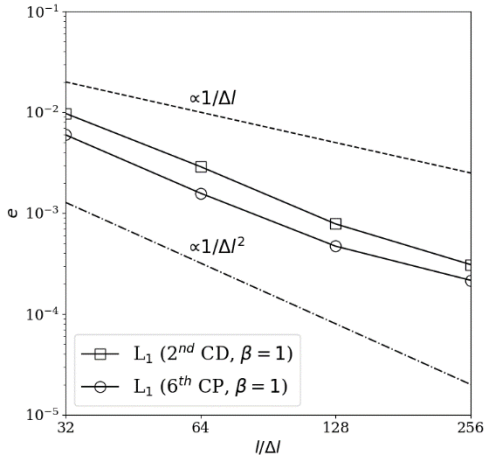
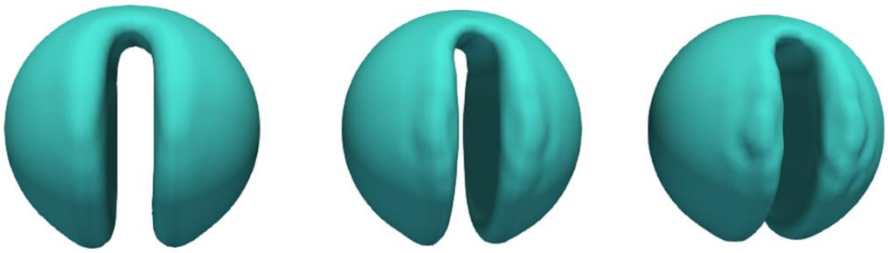


Figure 4 L_1-error norms for the second order central difference and sixth order compact schemes in the 2-D Zalesak's rotation problem

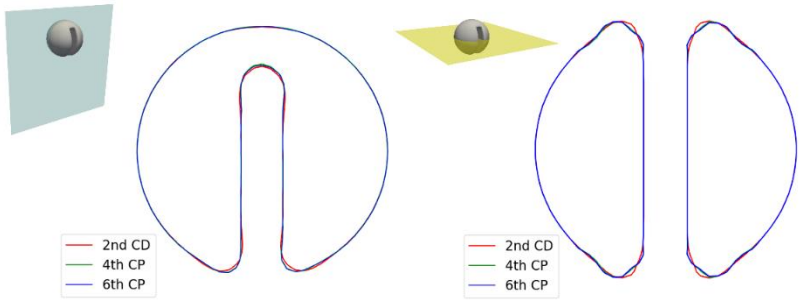
To conduct a more comprehensive validation of the full implementation of the method in CHAPSim2, the 3-D version of the Zalesak's rotation problem is used. In this case, a depth of non-dimensional length of 0.5 is used in the z-direction, and the computational domain becomes a cuboid [1 × 1 × 0.5] domain. At t = 0, a slotted sphere is placed near the upper boundary of the domain with the phase indicator function given by,

$$H(\mathbf{x},0) = \begin{cases} 1 & (x-0.5)^2 + (y-0.75)^2 + (z-0.25)^2 \le 0.15^2 \cap (|x-0.5| \ge 0.025 \cup y \ge 0.85) \\ 0 & \text{elsewhere} \end{cases} \tag{6}$$

The rotation velocity imposed is the same as that in the 2-D version of the case. The numerical parameters used remain unchanged where possible. For all the test simulations, a 100×100×50 mesh is used. Results after one full revolution of the slotted sphere are presented in Figure 5. In contrast to the 2-D case, the use of higher order schemes introduces unexpected distortions to the captured interface near the sharp corners (see Figure 5(a)), which may compromise the overall quality of the simulation. This can be better exhibited in Figure 5(b2), where horizontal slices of the sphere are presented for different numerical schemes. To improve the predictions, a smoother or limiter may be introduced for the high order schemes in the future.



(a)



(b1) (b2)

Figure 5 Effect of the numerical schemes on the captured shape of the interface after one full revolution in the 3-D Zalesak's rotation problem: (a) 3-D shape of the interface (ϕ = 0.5), (b1) 2-D shape on the vertical symmetry plane, (b2) 2-D shape on the centre horizontal plane

Figure 6 shows scalability tests conducted on three HPC systems: Bessemer and Stanage at the Sheffield University, and ARCHER2. The simulations are run on a single node to ensure relatively fair comparisons, despite variations in CPU specifications across the machines. The scaling performance on Bessemer (20 cores per CPU, 2 CPUs per node) and Stanage (32 cores per CPU, 2 CPUs per node) demonstrates similar trends, wherein the speedup gradually deteriorates as the number of CPU cores increases. However, on ARCHER2 (64 cores per CPU, 2 CPUs per node), the scalability tends to improve when using more than 16 cores. This may be attributed to a reduction in average communication overhead if a majority of the cores used belong to the same CPU. In the future, investigations will be conducted over larger cases running on multiple nodes to provide a more accurate assessment of the scaling behaviour.

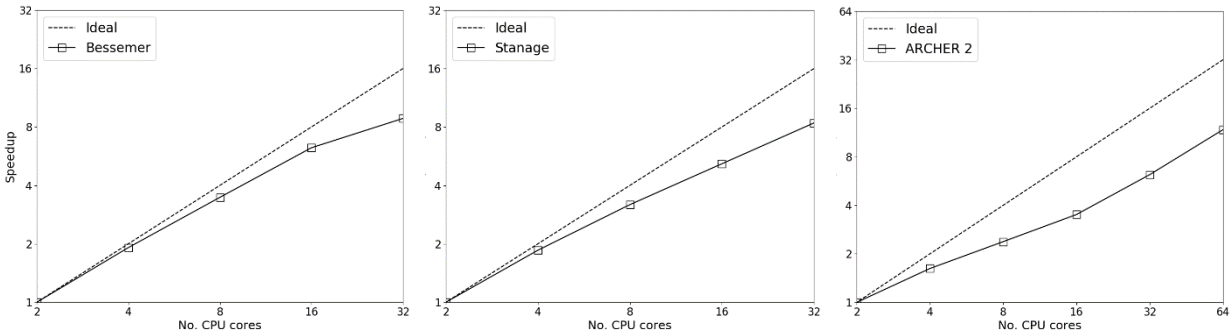


Figure 6 Scalability tests of the VOF module based on the 3-D Zalesak's rotation problem

A further parallel performance test of the CHAPSim2 implementation is conducted on ARCHER2. Table 1 shows the distribution of the CPU time used by the main functions/subroutines in the code. Notably,

approximately 60% of the CPU time is consumed by the MPI functions, with the MPI_ALLTOALLV operation being the primary contributor. This is, to some extent, an expected behaviour, considering that the MPI_ALLTOALLV function is heavily used in updating the volume fraction in the directional splitting scheme, as discussed earlier. To optimise the code in the future, efforts can be directed towards reducing the number of calls to the MPI_ALLTOALLV function. One potential optimisation strategy is to pack the coefficients of $P(\mathbf{X})$ into a single large array before being transposed between pencil states.

Table 1 Distribution of the CPU time used by the main functions/subroutines

Functions/subroutines	Time %
MPI	62.6
- MPI_ALLTOALLV	48.2
- MPI_FILE_OPEN	12.5
- MPI_BARRIER	1.3
USER (VOF functions)	17.4
ETC	13.2
MATH	6.4

4.2 Rising Bubble

The rising bubble problem is a well-documented benchmark used to evaluate interface-capturing or interface-tracking methods in multiphase flow simulations [6]. Its primary purpose is to evaluate how accurately a numerical method can capture the topological changes of a bubble rising under buoyancy in a quiescent fluid. The flow is governed by density difference between two immiscible fluids, with surface tension force playing a significant role. This makes it an ideal case to assess the VOF method implementation in CHAPSim2. The focus is on assessing the method's capability in conserving mass, preserving interface sharpness, and more importantly, accurately capture the evolution of the bubble shape over time.

The problem is defined as shown in Figure 7, at time zero, a circular bubble of diameter $d_0 = 0.5$ is placed at [x, y] = [0.5, 0.5] in a [1 × 2] rectangular domain. Driving by the buoyancy force resulting from density difference between the bubble and the surrounding fluid, the bubble rises and undergoes shape changes as it interacts with the induced flow field. The process can be characterised by several dimensionless groups, including the Reynolds number (Re), which represents the ratio of inertial to viscous forces; the Weber number (We), which compares the inertial to surface tension forces; the Froude number (Fr), relating inertial effects to buoyancy; density ratio $\rho_1/\ \rho_2$ and viscosity ratio $\mu_1/\ \mu_2$, which quantify the relative differences in density and viscosity between the two phases, respectively. Two cases with different density and viscosity ratios are considered. The first test case considers moderate density and viscosity ratios (both set to 10). The bubble undergoes moderate deformation, evolving from an initially circular shape into an ellipsoidal shape without breaking up. It serves as a fundamental benchmark due to its moderate complexity, enabling clear and quantitative assessments of numerical accuracy in predicting interface shapes, rise velocities, and bubble deformation. The second case, which is more challenging, involves a bubble with much lower density and viscosity relative to the surrounding fluid (density ratio of 1000 and viscosity ratio of 100, respectively). Significant deformation occurs as the bubble rises, and it eventually reaches a regime where breakup becomes possible, posing a particular challenge for interface-tracking methodologies. This work focuses on Case 1 with detailed comparisons of the results with benchmarks. For Case 2, only preliminary results are presented.

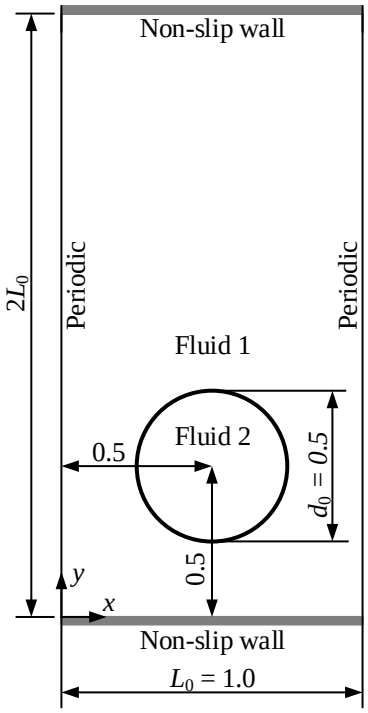


Figure 7 Setup of the initial and boundary conditions of the 2-D rising bubble benchmark

One of the key benchmark quantities is the bubble centroid velocity,

$$V_c(t) = \frac{\int_\Omega \phi v \mathrm{d}\Omega}{\int_\Omega \phi \mathrm{d}\Omega} \tag{7}$$

where ϕ is the volume fraction of bubble, and v is the vertical velocity component (i.e. the bubble rising direction). Figure 8 (left) shows time evolution of the bubble rising speed of Case 1. The bubble accelerates from rest due to buoyancy, reaching a peak rising speed at about $t = 0.9$, after which it gradually stabilizes at a near-constant terminal velocity. Quantitatively, the predicted maximum rising speed (~ 0.24) and its timing align very well with the reference data. Figure 8 (right) shows the bubble shape at $t = 3$, which has deformed into a stable ellipsoidal form without breakup. The CHAPSim2's prediction is in excellent agreement with the reference data, just overlapping with it. Additionally, data from the DNS code FluTAS [7] is used for cross-code comparison. Figure 8 (right) also shows that CHAPSim2's prediction is slightly more accurate that FluTAS in predicting the bubble shape evolution, further reinforcing the confidence in its ability for capturing interface dynamics.

In Case 2, the bubble has an extreme density ratio (1000:1) and viscosity ratio (100:1), leading to significant deformation of the bubble and the onset of breakup, as shown in Figure 9. It is important to emphasise that the Case 2 results are preliminary, and no trusted reference solutions are available for quantitative validation. Unlike Case 1, the extreme conditions in Case 2 causes different codes to produce noticeably different predictions with no exact solution to compare against. Thus, the observations in this case should be viewed as a qualitative assessment of CHAPSim2's VOF implementation. The key observations, such as bubble stretching, tail formation, and breakup, are physically reasonable and consistent with other high-density-ratio bubble studies, but further quantitative verification is needed.

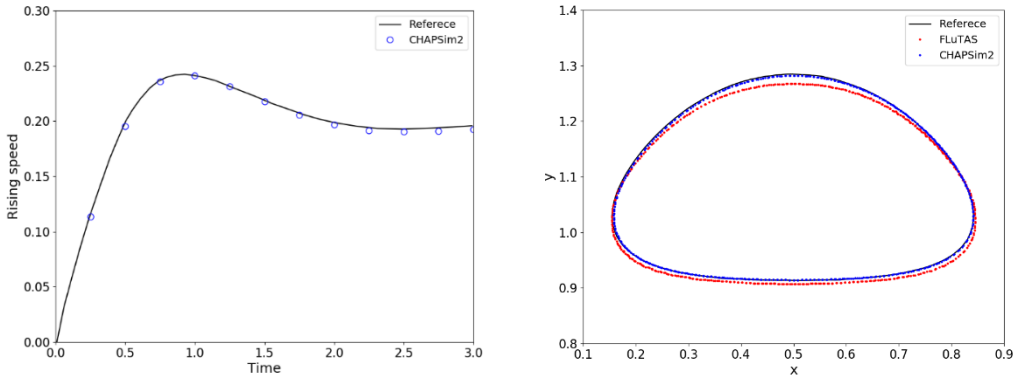
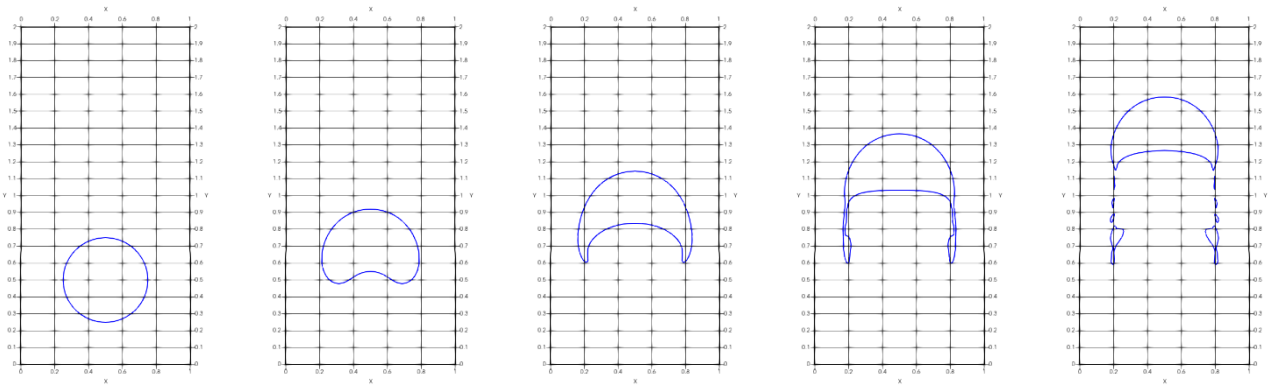


Figure 8 Simulation results of Case 1 (second order scheme is used for the calculation of the numerical fluxes, and the interface sharpness factor is set to 1). Left: Bubble rising speed calculated using Equation (7). Right: bubble shape at t = 3 (result of the DNS code FluTAS is provided for cross-code comparison)



t = 0 t = 1 t = 2 t = 3 t = 4

Figure 9 Bubble shape evolution in Case 2: snapshots at t = 0, 1, 2, 3 and 4

To further assess the VOF multiphase methodology in CHAPSim2, simulations are conducted for a fully 3-D version of the rising bubble problem. The computational domain is created by extending the 2-D domain in the z-direction, resulting in a [1 × 1 × 2] domain. A uniform 128×256×128 mesh is used. Only Case 1 is considered, with the density ratio, viscosity ratio, surface tension coefficient, gravitational acceleration, and reference velocity remaining identical to those specified in the 2-D case.

The bubble is initialised as a perfect sphere of diameter d_0 = 0.5, centrally located in the x and z directions (x_0 = 0.5, z_0 = 0.5), and placed at the lower half of the domain in the y-direction (y_0 = 0.5). Same as in the 2-D setup, gravity acts downward, and the density difference between the bubble and the surrounding fluid drives the bubble to rise. Figure 10 shows the isosurfaces of ϕ = 0.5 at various time instants (t = 0, 0.75, 1.5, 2.25, 3). As the bubble rises, the flow field around it induces deformations similar to the 2-D benchmark. However, in 3-D, additional flow structures can form in the wake of the bubble, slightly altering its deformation pattern. At t = 3, the bubble has ascended significantly and deforms into an ellipsoidal shape, maintaining overall integrity without breakup under the moderate density and viscosity ratios of Case 1.

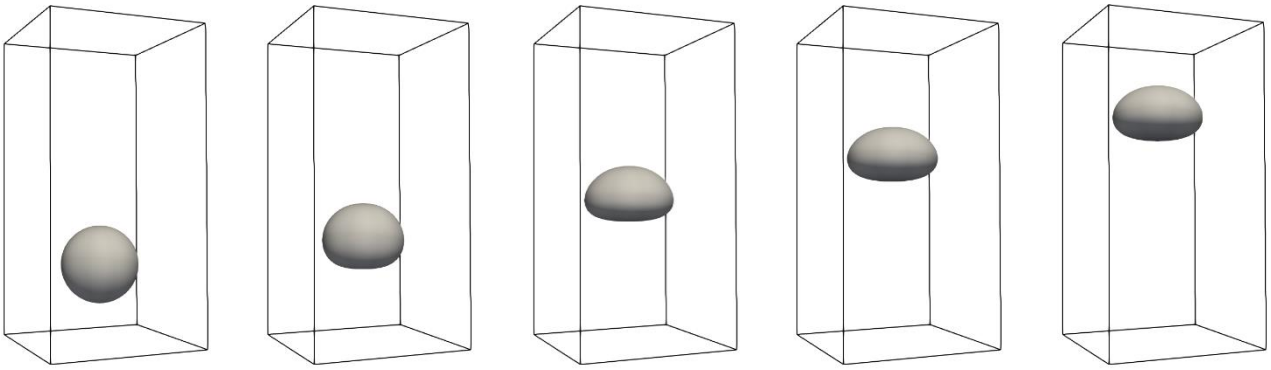


Figure 10 Isosurfaces of ϕ =0.5 at t = 0, 0.75, 1.5, 2.25 and 3

5. CONCLUDING REMARKS

This work aims at adding a Volume of Fluid (VOF) module to the high fidelity CFD software CHAPSim2 to enable simulations of multiphase flows. The MTHINC method proposed by Ii and co-authors [3] is selected for the implementation. In this method, an algebraic approach based on a hyperbolic tangent function is used to accurately reconstruct the geometry of the interface, ensuring a compact interface thickness.

High-order central differences and compact schemes are available in CHAPSim2 to compute the first and second-order derivatives of the volume fraction. These derivatives are used to determine the unit normal vector and the Cartesian curvature tensor of the interface, which are critical parameters to ensure the accuracy of the method. As demonstrated in numerical tests using the Zalesak's rotation problem [5], high-order schemes improve the overall accuracy, with a reduced absolute error observed, but the mesh convergence rate of the L_1-error norm remain unchanged. For the 3-D version of the Zalesak's rotation problem, high order schemes unexpectedly introduce distortions to the interface in regions with high curvature. This potentially compromises the accuracy of the method. Additionally, scalability tests of the CHAPSim2 implementation have been conducted across different HPC systems, showing that the MPI_ALLTOALLV operation for sub-domain transposition using the library 2decomp&FFT consumes a significant portion (> 50%) of the total CPU time. Further optimisation will focus on reducing the number of MPI_ALLTOALLV calls to improve the parallel efficiency of the code.

The complete multiphase isothermal solver in CHAPSim2, with coupling between the interface tracking and the momentum transport, is finally validated using a rising bubble benchmark [6]. CHAPSim2 demonstrates good capability in capturing the dynamics of a rising bubble. The solver accurately tracks the bubble surface for a moderate deformation case with relatively small density and viscosity ratios between the dense and light phases. The predicted bubble shapes, deformation patterns, as well as the bubble rising speed closely match those from established benchmark solutions. For a more extreme case with potential bubble breakage, reasonable predictions are observed and further quantitative validations are needed.

ACKNOWLEDGEMENT

This work is funded under the embedded CSE programme of the ARCHER2 UK National Supercomputing Service (http://www.archer2.ac.uk).

REFERENCES

1. W. M. Rohsenow & J. R. Hartnett, *Handbook of heat transfer*, 3rd ed (New York: McGraw-Hill, 1998). https://doi.org/10.5860/choice.36-3347.

2. X. Cheng & U. Müller, Review on Critical Heat Flux in Water Cooled Reactors. *Forschungszentrum Karlsruhe*, **6825** (2003).

3. S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi, & Y. Matsumoto, An interface capturing method with a continuous function : The THINC method with multi-dimensional reconstruction. *Journal of Computational Physics*, **231** (2012) 2328–2358. https://doi.org/10.1016/j.jcp.2011.11.038.

4. N. Li & S. Laizet, 2DECOMP & FFT-A Highly Scalable 2D Decomposition Library and FFT Interface. *Cray User Group 2010 conference*, (2010) 1–13.

5. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, **31** (1979) 335–362. https://doi.org/10.1016/0021-9991(79)90051-2.

6. S. Hysing, S. Turek, N. Kuzmin, E. Parolini, E. Burman, S. Ganesan, & L. Tobiska, Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids*, **60** (2009) 1259–1288. https://doi.org/10.1002/fld.

7. M. Crialesi-Esposito, N. Scapin, A. D. Demou, M. E. Rosti, P. Costa, F. Spiga, & L. Brandt, FluTAS: A GPU-accelerated finite difference code for multiphase flows. *Computer Physics Communications*, **284** (2022) 108602. https://doi.org/10.1016/j.cpc.2022.108602.