Efficient Implementation of Multi-Moment Methods in Gas-Liquid Two-Phase Flows and Heat Transfer

A thesis submitted in partial fulfilment

of the requirement for the degree of Doctor of Philosophy

Mohammed Bakir Mohsen Al-Mosallam

March 2018

Cardiff University
School of Engineering

TO MY BELOVED IRAQ
with wishes of prosperity.
TO MY PARENTS
who taught me the art of gratitude.
TO WIFE, SON, AND DAUGHTER
for their patience and support.

Abstract

Numerical simulations are a vital tool for understanding gas-liquid two-phase flows, and robust numerical methods are essential for this purpose. In this regard, a code library was developed using C++ for the numerical simulation of three-dimensional gas-liquid two-phase flows and heat transfer. The code is written based on a framework of numerical methods namely; Volume/Surface Integrated Average-Based Multi-Moment Method (VSIAM3) including Constrained Interpolation Profile-Conservative semi-Lagrangian (CIPCSL) methods, Coupled Level-Set and Volume-of-Fluid (CLSVOF) method, and density scaled CSF model. VSIAM3 is a numerical method for compressible and incompressible flows based on the multi-moment concept. VSIAM3 employs CIP-CSL schemes for solving the conservation equation. The CLSVOF is an interface capturing method that is well suited for two-phase flows with surface tension. The density scaled CSF model is used for the surface tension computation.

An efficient implementation of the numerical methods was investigated through the discretisation techniques of the conservation equation in VSIAM3. These techniques were studied through the lid-driven cavity, shock tube problems, two-dimensional explosion test, and droplet splashing on a superhydrophobic substrate. It has been found that the use of a less oscillatory CIP-CSL method is essential for robust numerical simulation of compressible and incompressible flows using VSIAM3 and that the numerical results are sensitive to the discretization techniques of the velocity divergence term in the conservation equation.

A parallel code library was also developed using Open MPI (the Message-Passing Interface) for the three-dimensional numerical simulation of gas-liquid two-phase flows and heat transfer. The parallel performance has been evaluated, and a good scalability was obtained. The code library was further validated through the numerical simulation of equilibrium drop, single rising bubble, Kelvin-Helmholtz instability, and turbulent channel flow. The numerical results were reasonable.

Validations of VSIAM3 for heat transfer problems were also conducted through single-phase and two-phase Rayleigh-Benard convection. We found that solving the diffusion term of the Navier-Stokes equation and the conduction term of the energy equation for all the moments in VSIAM3 is essential for robust numerical simulation of heat transfer problems using VSIAM3. In addition to that, using Time Evolution Converting (TEC) for computing the boundary values of the temperature in VSIAM3 as suggested in the literature influences the robustness of VSIAM3.

In conclusion, an efficient implementation of VSIAM3 for gas-liquid two-phase flows and heat transfer using VSIAM3 and CLSVOF was developed and validated through single-phase and gas-liquid two-phase flows and heat transfer problems. The established code library is suitable for the numerical simulation of gas-liquid two-phase flows and heat transfer.

Acknowledgements

I would like to extend my gratitude to all those who have helped me complete this work, both academically and personally.

First I would like to thank my supervisors, Dr. Kensuke Yokoi for his guidance, discussions and much patience during my study, and Prof. Phil Bowen for his advice and support. I would like to express my sincere gratitude for their support throughout the duration of my PhD study.

I would like to thank my family for their continuous unlimited support.

I would also like to thank the research office staff and the team of ARCCA system at Cardiff University for their sincere help.

I would like to acknowledge the financial support of this study provided by Grant from the Iraqi Ministry of Higher Education and Scientific Research and my employer University of Basrah.

Contents

List of Publications

The work introduced in this thesis is based on the following publications.

- Mohammed Al-Mosallam and Kensuke Yokoi, Efficient Implementation of Volume/Surface Integrated Average-Based Multi-Moment Method, Int. J. Comp. Methods, Vol. 14, No. 2, 2017.
- Mohammed Al-Mosallam and Kensuke Yokoi, Efficient implementation of volume/surface integrated average based multi-moment method. Presented at: 24th Conference of the Association for Computational Mechanics in Engineering, Cardiff, UK, 31 March - 1 April 2016.
- Mohammed Al-Mosallam and Kensuke Yokoi, Discretisation strategies of the conservation equation. Presented at: Toyo University-Cardiff University Workshop, Cardiff University, UK 15 February 2016.

List of Figures

List of Tables

List of Acronyms

ALE Arbitrary Lagrangian Eulerian

BFC Boundary-Fitted Coordinates

CIP Constrained Interpolation Profile Method

CIP-CSL Constrained Interpolation Profile Conservative Semi-Lagrangian Method

CIP-CSL2 Constrained Interpolation Profile Conservative Semi-Lagrangian Method Based on Quadratic Interpolation Function

CIP-CSL3 Constrained Interpolation Profile Conservative Semi-Lagrangian Method Based on Cubic Interpolation Function

CIP-CSLR Constrained Interpolation Profile Conservative Semi-Lagrangian Method Based on Rational Interpolation Function

CFD Computational Fluid Dynamics

CLSVOF Coupled Level Set and Volume of Fluid Method

CSF Continuum Surface Force Model

DNS Direct Numerical Simulation

ENO Essentially Non-Oscillatory

FDM Finite Difference Method

FVM Finite Volume Method

FEM Finite Element Method

LSM Level Set Method

LWR Light Water Reactor

MAC Marker and Cell Method

PDE Partial Differential Equation

PLIC Piecewise Linear Interface Calculation Method

THINC Tangent Hyperbolic Interface Capturing Method

SLIC Simple Line Interface Calculation Method

TEC Time Evolution Converting

VOF Volume of Fluid Method

VSIAM3 Volume/Surface Integrated Average Based Multi-Moment Method

WENO Weighted Essentially Non-Oscillatory

WLIC Weighted Line Interface Calculation Method

WRM Weighted Residual Methods

Chapter 1

Introduction

1.1 Motivation

Gas-liquid two-phase flows play an essential role in nature and industry. Many natural processes occur at a free surface. The most widely recognized illustration is the interface separating air and water; examples are wind blow over rivers and open channels, bubble formation, rain droplets, atmosphere-ocean interaction and various types of sea waves. Two-phase flows include multi-physics phenomena. They also include multi-length and multi-time scales. Examples of interest are sprays, evaporation, gas absorption and heat transfer accompanying turbulent wind-waves. Other cases that involve multi-length and multi-time scales are Kelvin-Helmholtz waves which occur at small-scale motions in the oceans and atmosphere [114, 129] and two-phase Rayleigh-Benard convection caused by hydrodynamic and thermal interactions of convective flows through the interface [118]. Other interesting examples are water jets that break into drops and porous media like water in oil reservoirs. Similarly, industrial processes that involve interfacial flows are countless. One can mention interfacial convection which is vital in many engineering applications [118] such as microfluidics, material processing, crystal growth [85], and emulsified liquid membrane separation employed in industrial waste-water treatment [121]. Another notable example is condensation on liquid films and its applications in the nuclear industry. The typical reference situation, in this case, is the refill stage after a loss-of-coolant accident in an LWR. In this situation, the emergency cooling water comes into contact with the steam generated in the overheated core. One can also mention boiling heat transfer which is the preferred mode of heat transfer to extract large amounts of energy in various industries like power generation and cooling in metallurgical industries. Two-phase flows are also essential in fuel combustion where atomization of the fuel and formation of droplets are essential for combustion to commence. One can mention many other examples such as heat and mass transfer enhancement in bubbly flows as in bubble columns [35], gas absorption processes in chemical plants such as in mixing type heat exchangers, degassers and seawater desalting by multiple distillations [87], and so on.

Figure 1.1: Droplet splash on a dry solid surface. [198]

Since free surface flows appear in such diverse applications, understanding them is of critical engineering and scientific importance; for instance, for predicting their behaviour in nature and applying their fundamentals in engineering applications and industrial processes. However, despite the extensive work in free surface flows, their behaviour far less well understood, particularly when the flows involve large interfacial deformation [11, 169]. This is because flows with interfaces are difficult to investigate and much of our knowledge have acquired by experimental work and dimensional analysis. The later only applicable for simple flow cases, while experimental measurements are difficult to near interfaces in many flows of practical applications, where the length and time scales are small [169]. One can mention many essential example flows; for instance, wind-driven turbulence and its direct impact on climate and weather change including extreme weather events like the build-up and decay of tropical cyclones [43, 58, 52, 113, 210]. The interfacial deformation in wind-driven turbulence significantly facilitates transport of momentum and scalar [89, 53, 76, 171]. Despite the extensive literature, the mechanisms controlling wind-driven interfacial flow are not fully understood [176, 140]. Experimental works of wind-driven air-water flow are characterized by the difficulty of the measurements near a turbulent interface [51, 91]. In this context, due to the difficulties involved in studying such complex flow, inconsistent conclusions have been reported for the heat transfer coefficients. Based on field observations, it has been stated that the heat transfer is enhanced by wind shear and that the latent/sensible heat transfer coefficients have a constant value (e.g. [151, 54, 50, 99, 34, 130, 209, 94]). On the other hand, experimental investigations using wind-wave tanks have indicated that the latent heat transfer coefficient is proportional to wind speed (e.g. [123, 208, 39, 91]). The previous example shows the significant difficulties of studying such complicated flows. Dispensable tool to study dynamics of two-phase flows with a deformable interface and their underlying mechanisms is the numerical simulation.

Figure 1.2: Breaking wave: highly deformable air-water interface. [120]

Numerical simulation of free surface flows is a difficult task even when the interface remains smooth. The governing equations are highly nonlinear, and the interface must

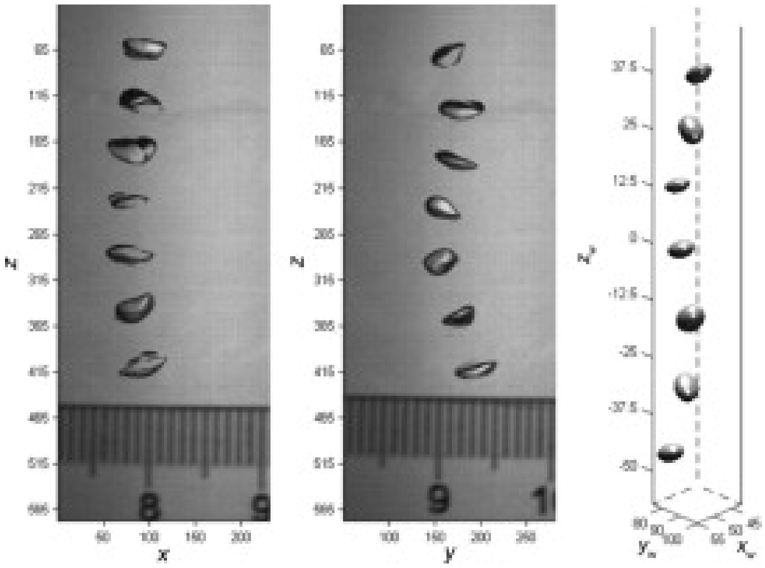


Figure 1.3: Single rising bubble in water [15]

be determined as part of the solution [169]. Therefore, employing efficient numerical schemes for obtaining the numerical solution is essential. As we can expect, numerous studies of interfacial flow have used different approximations of the governing equations or made assumptions about the nature of the interface to accomplish successful simulation [11]. Concerning these flows, various methodologies for tracking and capturing interface motion have been proposed and attempted; for instance, Front Tracking method, Volume-of-Fluid method, and Level-Set Method. Since each developed method has advantages and drawbacks, make use of the method with the best possible features is advantageous for robust numerical simulations. Various numerical schemes were applied to discretise the governing equations of the two-phase flow such as finite difference, finite volume and finite element schemes. Each method has its own features and drawbacks. An overview of the common strategies for the numerical simulation of two-phase flows and the most common spatial discretisation techniques is given in chapter 2.

In the present study, CLSVOF (Coupled Level Set and Volume of fluid) method on fixed grid has been employed [158, 204]. The THINC/WLIC scheme is used for interface capturing. WLIC method [201] is employed for interface reconstruction. THINC/WLIC method satisfies volume conservation, relatively easy to implement, manages interface motion, and is capable of handling highly deformable interfaces. The Level-Set method has been employed for the computations of curvature and hence surface tension force. The CLSVOF method integrates advantages of both VOF method and Level-Set Method. In the CLSVOF, using a VOF scheme conserves the volume fraction while maintaining sharp interface, and thus compensate a loss-of-mass disadvantage of the Level-Set method. A drawback of the VOF scheme is the difficulty in computing curvature from volume fractions due to the use of sharp volume fractions at the interface. This drawback is covered by the Level-Set method where interface curvature is computed from the level set function. The level set function in CLSVOF is computed from both the level set function and VOF function at the previous time step [158].

Accurate computations of the interface velocities are critically important for robust numerical simulation. In the present work, Volume/Surface Integrated Average-based Multi-Moment method (VSIAM3), Xiao et al.[188, 183, 184], has been employed as the solver for fluid flow and heat transfer. Volume/Surface Integrated Average-based Multi-Moment method (VSIAM3)[188, 183, 184] is a numerical method for compressible and incompressible flow based on a multi-moment concept. VSIAM3 employs conservative semi-Lagrangian (CIP-CSL) method to solve the conservative advection equation.

VSIAM3 has been used in the present work because it is based on the multi-moment concept. Multi-moment methods are numerical methods which use multiple integrated variables (moments) for a physical field and update these moments by utilising different formulations yet same conservation laws. VSIAM3 (including CIP-CSL method) is based of finite volume method. Thus, provides the finite-volume features such as conservation, computational efficiency, and flexibility in handling irregular geometries. Multi-moment methods possess attractive features that are well suited for multiphase flows. For instance, a compact stencil for spatial reconstruction and flexibility in treating complex geometries. VSIAM3 employs the accurate CIP-CSL advection schemes featuring modifiable interpolation for reconstruction. In VSIAM3, Cartesian coordinates are used to express the interface, thus, it does not require computational effort for continuous reconstruction of the computational grid even with extremely deformable interface [182, 188].

1.2 Research Objectives

- The first target of the present work is to develop an efficient C++ code library for the numerical simulation of gas-liquid two-phase flows and heat transfer based on the VSIAM3 and CIP-CSL schemes. CLSVOF scheme for interface capturing, where the method uses THINC/WLIC scheme as sort of VOF method.
- To study robustness issues in VSIAM3 and to investigate efficient implementation of VSIAM3 in incompressible and compressible flows.
- To develop a parallel implementation of the code for the numerical simulation of gas-liquid two-phase flows and heat transfer by using Open MPI (Massage Passing Interface) so that the numerical simulation can be implemented on a single node and supercomputers as well.
- To carry out further validation of the solver through three-dimensional numerical simulations of Kelvin-Helmholtz instability, single rising bubble, and turbulent channel flow.
- To study robustness issues in VSIAM3 heat transfer solver through the numerical simulation of Rayleigh-Benard convection.

1.3 Thesis Outline

The thesis consists of seven chapters:

- In chapter 1, the motivation, background, and objectives for the present work are given.

- In chapter 2, an overview of the literature on the numerical methods for two-phase flows and discretisation techniques for partial differential equations are given.

- Numerical methods are presented in chapter 3. VSIAM3 method along with the CIP-CSL conservation equation schemes (CIP-CSL2, CIP-CSLR, and CIP-CSL3) are explained in detail in order to simplify the multi-moment framework. The CLSVOF scheme for interface capturing is also explained.

- In chapter 4, an investigation for efficient implementation of VSIAM3 in single-phase and gas-liquid two-phase flows is carried out. The implementation was carried out through cavity flow, one-dimensional Sod and Lax problems, two-dimensional explosion test, and droplet splashing on dry surface.

- A description of the parallel implementation of the numerical methods is given in chapter 5. The written code library is further validated through different problems of single and two-phase gas-liquid flows.

- In chapter 6, We studied robustness issues of VSIAM3 in the numerical simulation of heat transfer problems through numerical simulation of Rayleigh-Binard convection.

- A summary and suggestions for further work are given in chapter 7.

Chapter 2

Literature Review

2.1 Introduction

The following literature review addresses the choice of CLSVOF [202, 204] as an interface capturing scheme among other numerical methods for free surface flows. It also considers the choice of the multi-moment VSIAM3 [188] in the present work for the spatial discretisation of the Navier-Stokes equations among various discretisation techniques. First the development of the interface capturing techniques is considered in section 2.2. An overview of the common spatial discretisation strategies for partial differential equations is secondly presented in section 2.3. The VSIAM3 is introduced in section 2.4 followed by a conclusion in section2.5.

2.2 Interface Capturing Techniques

In recent times critical advancements in numerical techniques and computing power have enabled fast evolution in numerical simulation of two-phase flows. These simulations of two-phase flows numerically solve the Navier-Stokes equations to predict fluid dynamics and physical processes in both phases and follow interface motion by treating an advection-type equation

$$\frac{\partial \psi}{\partial t} + u\frac{\partial \psi}{\partial x} + v\frac{\partial \psi}{\partial y} + w\frac{\partial \psi}{\partial z} = 0. \tag{2.1}$$

Where ψ represents the interface (e.g. volume fraction in VOF method, level set function in Level Set method). This equation states that ψ moves with the fluid [74].

Numerical methods for free surface flows can be categorised depending on the employed grid type into three groups; fixed grid (Eulerian) [143, 187], moving grid (Lagrangian) [42, 65, 79], and Arbitrary Lagrangian Eulerian grid (ALE) [73, 77]. Moving grid methods consider the interface as a boundary between two domains of meshes, and allows the mesh to move with the fluid, which results in accurate tracking of the interface motion. However, to track the interface, the interface motion requires continuous re-meshing, which in turn, requires substantial computational effort for interfacial flows subjected to high topological changes. In fixed grid methods, on the other hand, the interface motion is tracked on a non-moving grid and feature the ability to treat large interfacial deformations, relatively simple interface description, and more straightforward extension to three dimensions, which makes it more applicable to numerical simulations of complex interfacial flows [136]. ALE method was developed in an attempt to combine the advantages of the above grid types, while minimizing their respective disadvantages. The method features precise interface definition, however, in comparison to fixed grid it only capable of handling small topological changes in the interface and no inclusion of one phase into the other are assumed (e.g. in application of ALE with boundary-fitted coordinates (BFCs) on moving grids for wind-driven turbulence [92, 51, 97, 90, 170, 62, 161, 96]). Thus Eulerian methods are generally the most employed methods for complex two-phase flows [203, 204, 205], because they allow considering large interfaces deformation.

Eulerian methods for free surface flows can be classified depending on the type of interface representation into interface tracking methods and interface capturing methods. Front tracking method represents the interface explicitly (thus they have been described as interface tracking schemes) by marker-particles. A disadvantage of this method is the difficulty of handling topological changes like droplets merging and break-ups [178]. Examples of interface tracking methods are Marker-and-Cell (MAC) methods [66] and Front-Tracking method [78, 100, 57, 172] . The Marker-and-Cell (MAC) and the Front Tracking use marker-particles to identify the free surface. The original implementation involves only one fluid. Later was extended to include both phases and applied for various free surface flow problems [57, 149, 133, 166, 168, 41, 172, 163]. Inaccuracies characterise MAC method due to the use of marker-particles [66]. Front Tracking methods are known to be difficult to apply for interfaces with topological changes [172].

Interface capturing methods represent the interface implicitly (thus they have been described as interface capturing schemes). Interface capturing methods include Level Set methods [3, 127, 126, 159], Volume of Fluid (VOF) methods [74, 143, 145, 108], THINC methods [187, 186, 106], and Coupled Level Set and Volume of Fluid method (CLSVOF) [158, 180, 204].

In the following sections, an overview of the main approaches of interface capturing strategies is given. First Volume of Fluid method (VOF) is introduced in section (2.2.1) followed by the Level Set method in section (2.2.2) and Coupled Level Set and Volume of Fluid (CLSVOF) in section (2.2.3). A summary is given in section (2.2.4).

2.2.1 Volume of Fluid Method

The Volume of Fluid method was developed by Hirt and Nichols (1981) [74]. the method represents the interface implicitly by using Heaviside step function (also called characteristic function) $\xi(x,y)$:

$$\xi(x,y)=\begin{cases}1 & \text{for the liquid at point (x,y),}\\ 0 & \text{for the air at point (x,y).}\end{cases} \tag{2.2}$$

The Heaviside step function takes a value of 0 for the light fluid and 1 for the other fluid. Noting that $\xi(x,y)$ is defined at a point (x,y) and not on a computational grid. The value on the grid is called volume fraction or color function ϑ and is defined as the cell average of $\xi(x,y)$,

$$\vartheta_{i,j}=\frac{1}{\Delta x\Delta y}\iint_{\Omega_{i,j}}\xi dxdy, \tag{2.3}$$

where $\Omega_{i,j}$ is a grid cell. The volume fraction has the value ($0 \leq \vartheta \leq 1$).

Crucial advantage of the method is that the volume of the fluid is completely conserved when the interface advection equation (2.1) is discretised using conservative formulation.

In the numerical simulations, fluids physical properties are accounted for by the volume fraction. Moreover, the method allows straightforward extension to multi-dimensional problems. Thus it has become popular in commercial CFD software and has been applied for numerical studies of various fluid dynamics problems [9, 26, 37, 46, 82, 107]. On the other hand, major disadvantages of the volume of fluid method are manifested by the numerical errors caused by the discrete representation of the interface and the numerical solution of the advection equation. These numerical errors appears as diffusion errors as well as non-physical behaviour in the interface motion called spurious currents, which are parasitic velocities induced by the numerical method. Moreover, the Volume of Fluid method is characterised by the difficulty of surface tension force computation. The curvature calculation is complicated since it requires estimation of derivatives at the interface of sharp function (Heaviside unit step). Many methods are developed to improve the volume of fluid method are dedicated to reduce interface diffusion and to overcome the difficulty of curvature computation.

Most of the proposed developments to overcome difficulty of computing surface tension term tend to regularise the fluid volume fraction to facilitate the estimation of derivatives [18]. The smoothing of the fluid volume fraction using variety of kernel functions was proposed in [16, 19]. An investigation was carried out in [33] to study and compare different strategies to overcome the difficulty. These are the smoothing with kernel functions, using a height function to interpolate the interface and reconstruction of a distance function, similar to the level sets method. An accurate reconstruction of the interface is required to reduce the numerical diffusion and the spurious currents and to achieve a more accurate interface approximation. The main techniques for interface reconstruction are presented and discussed in the following paragraphs.

In the original algorithm of the Volume of Fluid method, called the donor-acceptor method by Hirt and Nicolas [74], the interface is reconstructed parallel to one spatial

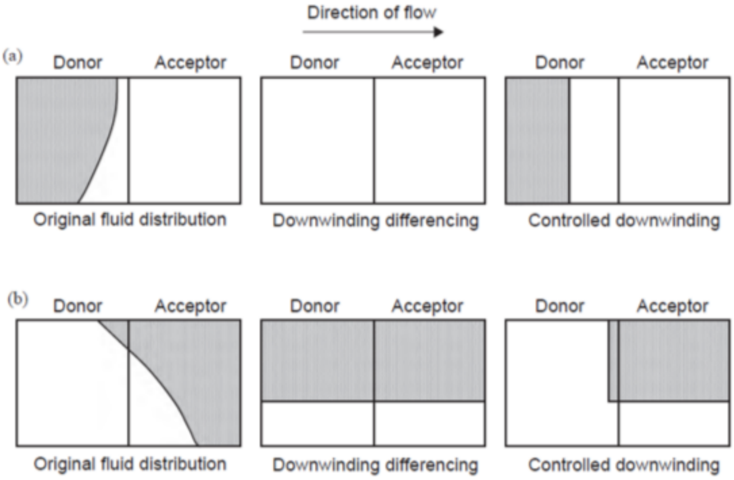


Figure 2.1: The donor-acceptor interface reconstruction [199]

direction using the volume fraction information of the adjacent cells in the direction of interface motion, where the interface is reconstructed either vertically or horizontally relevant to the coordinate system depending on the adjacent grid cells. The numerical flux parallel to the interface is estimated using an upwind technique. Other numerical fluxes are estimated using a both downwind and upwind techniques. Figure 2.1 depicts the donor-acceptor method. These techniques are first-order accurate. In order to reduce numerical diffusion of the interface, Noh and Woodward [122] proposed another geometrical interface reconstruction called the Simple Line Interface technique (SLIC). The interface reconstruction in the SLIC technique is achieved by introducing straight lines in the interface containing cells. Here the interface is also reconstructed parallel to one coordinate direction. Both donor-acceptor and SLIC techniques are considered as piecewise constant techniques because the interface is taken either vertical or horizontal. An advantage of both reconstruction techniques is their simplicity. However, they are only first-order accurate with respect to grid size [74, 122, 18]. Hence, both techniques produce inaccurate results for simulations of interfaces with high topological deformations.

Another reconstruction technique, flux line-segment model for advection and interface reconstruction (FLAIR) was proposed in [8]. FLAIR algorithm is based on approximating the interface by a set of line segments fitted at the boundary of every two adjacent

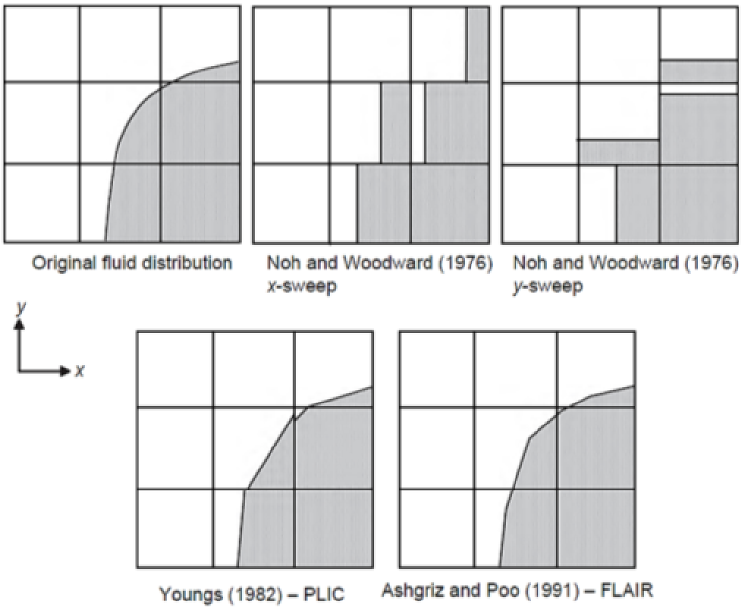


Figure 2.2: Comparison of SLIC, PLIC, and FLAIR interface reconstruction techniques [199].

cells. The orientation of the interface in a computational cell is found by inspecting the the cell volume fraction. The new volume fractions are obtained by integrating the area underneath the interface line-segment.

A more accurate geometric interface reconstruction technique is Piecewise Linear Interface Calculation (PLIC). It is attributed to Young [207] and further advanced by Rider and Kothe [142]. The PLIC technique reconstructs the interface using a line segment of a slope determined by the gradient of the volume fraction function. This technique is second-order accurate technique [142]. Figure 2.2 shows a comparison of the SLIC, PLIC and FLAIR interface reconstruction techniques. The PLIC is robust technique when the interface has a small or large curvature with respect to the grid size [18]. The key point in the technique is the determination of the direction of each segment of the reconstructed interface based on the interface normal vector which is also determined using volume fraction values in the adjacent cells [104, 61, 105, 18]. Other techniques were also proposed. Interface reconstruction technique based on the least square fit was suggested in [146]. The technique features interface continuity at the boundaries of adjacent grid cells.

Disadvantages of the PLIC geometrical reconstruction technique is the computational cost and the complexity of implementation in three-dimensional problems. Yokoi [201] proposed an efficient technique for interface reconstruction, namely Weighted Line Interface Calculation (WLIC). Similar to the SLIC and Hirt and Nicholas interface reconstruction techniques, the WLIC offers simple implementation while considering the information of the interface normal vector more effectively than the former techniques. This is achieved by weighting the interface along the coordinate directions rather than reconstructing it parallel to one spatial direction as in the former techniques. Meaning, in two dimensions for instance, the WLIC employs both the horizontal surface and the vertical surface for reconstructing the interface by using weights of both surfaces calculated form the interface normal as depicted in fig. 2.3. It has been reported that the results by the THINC/WLIC method are almost same with these by the VOF/PLIC method [202, 204]. The WLIC features a straightforward extension to three-dimensional problems.

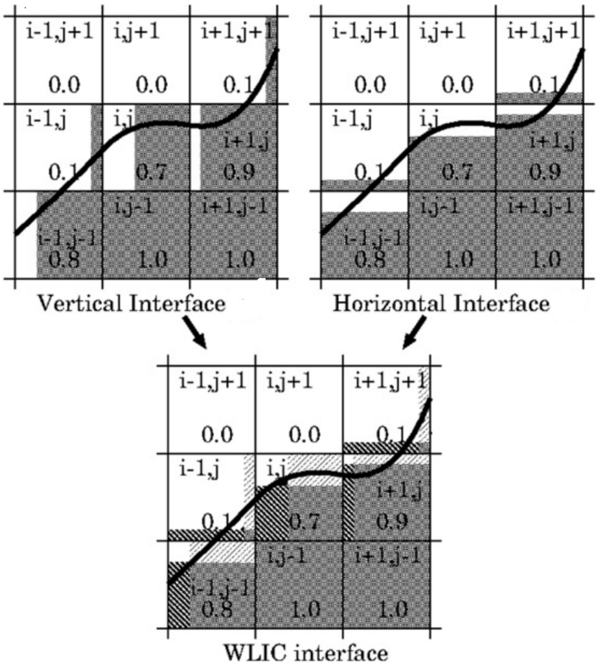


Figure 2.3: The WLIC technique [201]

The THINC method [187, 185] is conceptually a Volume of Fluid method with the exception that a smoothed Heaviside function (a one-dimensional piece-wise modified hyperbolic tangent function) which can be written as

$$\xi_{x,i} = \frac{1}{2}\left(1 + \alpha_x \textbf{tanh}\left(\beta\left(\frac{x - x_{i-1/2}}{\Delta x} - \tilde{x}_i\right)\right)\right), \tag{2.4}$$

is used by the THINC method. Where $\tilde{x}_i \Delta x$ corresponds to the distance between grid point $x_{i-1/2}$ and the interface. The parameters α_x and β are important in determining the quality of the numerical solution. Thus the method satisfy volume conservation. The method solves the following advection equation

$$\frac{\partial \xi}{\partial t} + \nabla \cdot (\mathbf{u}\xi) - \xi \nabla \cdot \mathbf{u} = \mathbf{0}. \tag{2.5}$$

The cell-integrated average of the smoothed Heaviside functions is the fluid volume fraction. The idea of using a smoothed Heaviside function was first introduced [185]. In the THINC algorithm, the calculation of the interface normal vector benefits from the feature of using smoothed volume fraction function and it plays an important role in preventing flotsam and jetsam [201, 202]. This feature allows employing simple interface reconstruction technique, the WLIC. The characteristic function of the THINC method is a piecewise hyperbolic tangent function and the flux is calculated based on dimensional splitting approach.

To conclude, great advances have been accomplished since the presentation of the volume of fluid method taking the advantage of mass conservation. Most of the developments in this field were committed to simplify the implementation and improve the accuracy of the numerical method. The VOF method requires interface reconstruction for more accurate simulations of interface phenomena. Several geometrical interface reconstruction techniques have been proposed some of which are characterised by accurate representation of the interface, expensive computational cost and difficulty of three-dimensional implementations as in the VOF/PLIC technique. However other techniques provide accurate interface representation, simple implementation and straightforward extension to three-dimensional problems like the THINC/WLIC technique.

2.2.2 Level Set Method

The level set method was first proposed by Osher and Sethian [127] (1988). It is a numerical method for capturing moving interfaces. The method employs implicit representation of the interface which is given by zero level set function ($\psi = 0$) and can handle interface deformation without special treatment since the interface is embedded in a higher dimensional function. This function enables robust calculation of geometric information. Application of the method to simulate multiphase flows was first reported by Sussman et al. [159]. The level set function is defined as a smooth signed distance function where it take a positive and negative values for the heavy and light fluids, respectively. The interface separating the two fluids is thus represented by the set of points at which the level set function equals zero ($\psi = 0$). A level set advection equation (2.1) ($\partial\psi/\partial t + \mathbf{u} \cdot \nabla\psi = \mathbf{0}$) advects the level set function (ψ) by the flow velocity field ($\mathbf{u}$) and enables tracking the interface motion. Furthermore, the signed distance function offers simple and straightforward way for computing the interface normal vector ($\mathbf{n}$),

$$\mathbf{n} = \frac{\nabla\psi}{|\nabla\psi|}, \tag{2.6}$$

and the curvature (κ),

$$\kappa = \nabla \cdot \frac{\nabla\psi}{|\nabla\psi|}, \tag{2.7}$$

therefore, the surface tension term can be computed simply using the continuum surface force model [16]. Therefore, it has been widely applied to solve various problems [127, 1, 2, 64, 25, 150, 60].

On the other hand, the level set method has the drawback that the discretisation of the level set advection equation experiences more numerical error (significant numerical dissipation) than Front Tracking or VOF methods when the interface subjects to strong stretching or tearing (areas of high curvature) [141]. This numerical dissipation causes issues with mass conservation. Although the level set function can be conserved by a conservative formulation of the discrete advection equation, the mass enclosed by the zero level set is not conserved [158]. Moreover, the degeneration of the level set function; the signed distance function from the interface does not preserved by the solution of the advection equation; necessitates rescaling the function at each time step. The issue with the unpreserved level sets function requires updating the distance function by solving a non-linear hyperbolic type equation to steady state. This step is called the reinitialisation of the distance function. Improvements to the accuracy and robustness of the level set method have been carried out. These improvements target the reinitialisation step to ensure mass conservation, investigating alternative higher-order discretisation techniques or even using hybrid approaches [111].

Sussman et al. [157] proposed an enhancement to the reinitialisation of the distance function by fixing the number of iterations required to solve the reinitialisation equation to steady state. The enhancement guarantees that the level set ensures correct distance to the interface. It is known that the reinitialisation step affects the position of the interface (the position of the zero level set) which leads to the so called reinitialisation error. The reinitialisation error increases at each time step by accumulation. Figure 2.4 depicts the reinitialisation error. Yokoi [200] proposed a development to the level set method in a Cartesian fixed grid. The proposed improvement aims at preventing the issue of accumulation of reinitialisation error. In his method, a second level set function is defined and used to capture the interface. The original level set function is then reconstructed from the zero level set of the defined function. This treatment prevents the accumulation of the error because the time evolution is calculated using the new function.

Several studies have been conducted to enhance the accuracy of the level set method. Numerical diffusion error is typical of using standard differencing schemes to discretise the level set equation which also leads to mass conservation issues or distortion of the signed distance function. It is therefore important to adopt a non-diffusive differencing

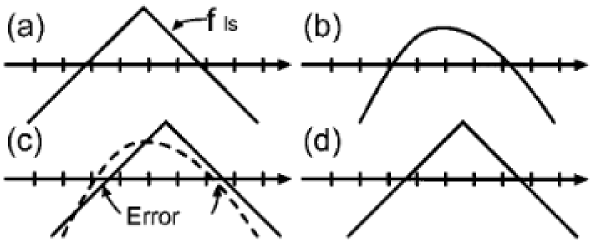


Figure 2.4: Schematic figure of the reinitialisation error. (a) the original interface, (b) the advected interface, (c) reinitialisation error, (d) Error accumulation. (f_{ls} is the level set function) [200].

scheme such as the third-order accurate ENO (essentially non-oscillatory schemes) [67] to discretize the level set advection equation as was shown in [157]. Due to the possible smearing of the interface that could occur if the grid is not sufficiently fine, it is even desirable to employ higher-order schemes such as the fifth-order accurate WENO (Weighted essentially non-oscillatory schemes) [110] in order to better capture the sharp interface. The Total Variation Diminishing Runge-Kutta scheme was also adopted in [148] as a remedy to the numerical diffusion problem. In simulations with large topological deformation, [156] proposed combining the level set with adaptive projection schemes to obtain higher resolution accuracy with low additional computational cost. An adaptive meshing techniques were also adopted for free surface flows [112]. Another technique, the Refined Level Set Grid method, was proposed in [69] to enhance the accuracy of the level set method and to overcome the issue with mass conservation. The method uses a refined mesh in the interface area in addition to the original computational mesh, where the level set equation and the reinitialisation equation are solved on the refined grid. The method however does not totally eliminate that error. Other strategies to enhance the accuracy of the level set method were proposed, for example mixed methods such as the particle level set method [44] and the coupled level set and volume of fluid method [158], and adaptive meshing techniques for free surface flows [112].

In conclusion, the level set method provides desirable features for the numerical simulation of two-phase flows such as its ability to handle topological changes of interfaces without the need to special treatments as well as simple computation of the interface normal and curvature. A disadvantage, however, is the issue with mass conservation caused by the discretisation of the level set equation which reveal itself as significant dissipation error. Several remedies has been suggested to overcome the issue such as using higher-order numerical schemes, improving the implementation of the reinitialisation step and proposing mixing strategies to combine the advantages of different numerical schemes. One of the promising and common strategies in the field is the coupled level set and volume of fluid method.

2.2.3 Coupled Level Set and Volume of Fluid Method

The mixing of VOF and level set methods is advantageous to combine the desirable properties of both methods, see for example [158, 173, 202]. It allows mass conservation and keeps a smooth approximation around the interface, but needs a strategy to let both the level set function and the volume fraction function work together efficiently. Several strategies have been adopted. In [173], a signed distance function is reconstructed from the advected volume fraction function and the interface jump conditions are satisfied using a method similar to the ghost fluid method [45] for incompressible flow. The resulted scheme is named Mass-Conserving Level-Set (MCLS) and applied to simulate bubbly flows [174].

Motivated by simulating microscale jetting devices, Sussman and Puckett introduced the Coupled Level Set and Volume of Fluid method (CLSVOF) [158] to simulate three-dimensional and axisymmetric incompressible two-phase flows. The method is superior to both methods since it overcome the loss of mass issue of the level set method by the advection of the volume fraction function and the interface sharpness is kept by using the level set method to compute the normal vector and curvature. In [158], a piecewise linear interface is used to initialize a new signed distance function at each time step. Then, the advection algorithm simultaneously solves both the level set equation and the volume fraction advection equation to advance the free surface on a uniform grid. The values of the signed distance function are employed to accurately compute a gradient and normal for use in the piecewise linear interface reconstruction. Noting that in the computational cells which are away from the interface (the signed distance function is larger than a grid resolution) , no interface is assumed and the volume fraction function is assigned a value of 0 or 1 regardless of the computed value of it, thus reducing the undesirable appearance of jetsam and flotsam. The interface is given a specified thickness using a smoothed Heaviside function. The uniformity of the interface thickness is maintained since the level set function represents a signed distance to the interface. Furthermore, the level set function can be used to compute geometric information such as the curvature more efficiently than using the volume fraction function. Although the presented method produce better results over using either the level set method only and the volume of fluid method only, the reconstructed interface appears noisy and lacks time coherence [115]. Moreover, the only way to remove unsightly and inaccurate flotsam and jetsam is to delete it from the calculation non-physically removing mass [111]. The researchers in [158] have shown that the developed CLSVOF is superior to the level set method for problems in which the interface develops corners, or there is interfacial merging or pinching since the mass is conserved to a fraction of a percent using CLSVOF while the loss of mass is much larger in the level set method. Moreover, the CLSVOF is superior to volume of fluid methods in problems with surface tension.

A coupling of the level set and volume of fluid which only solves the volume fraction advection equation (VOSET) to advance the interface was presented in [40] for computing incompressible two-phase flows. The aim of this approach is to reduce the computational cost and simplify the implementation of the CLSVOF. The initial value of the level set function is computed from the volume fraction function after advection. The level set function is then reinitialized within a region of three mesh cells on each side of the interface. The numerical results conserve mass. Advancing the interface by the solution of the volume fraction advection equation only in CLSVOF was also implemented in OpenFOAM in [95]. The level set function is computed from the volume fraction function and the interface is represented by the 0.5-contour. The interface is reconstructed to appropriately simulate the contact line evaporation for boiling heat transfer studies.

An adaptive approach is employed to the CLSVOF in [197] for interface capturing on unstructured triangular grids. The level set equation is solved by a discontinuous Galarkin finite element method while the advection of the volume fraction is implemented by a Lagrangian-Eulerian formulation. The method is coupled to a finite element based Stokes solver. The interface normal is calculated the level set function while the line constant is reconstructed by a VOF formulation. The researchers show that the method maintain mass conservation accurately. Moreover, the method is able to treat topological changes efficiently due to the adaptive grid algorithm.

Yokoi [204, 202] proposed a practical CLSVOF for the numerical simulations of complex free surface flows with surface tension force such as droplet splashing. The method employs THINC/WLIC scheme as a volume of fluid method, where the WLIC interface reconstruction technique has been shown to produce as accurate results as those produced by the PLIC. The WLIC offers simple implementation and straightforward extension to three-dimensional problems. The method employs a multi-moment (CIP-CSL and VSIAM3) and the level set curvature interpolation technique. The algorithm is based on advancing the interface by solving the volume fraction advection equation using THINC/WLIC on a uniform Cartesian grid. The position of the interface (zero level set) is computed from the volume fraction function using linear interpolation between adjacent grids, then, the signed distance function is constructed within a grid spacing by the fast marching method [147]. In the computational cells which are away from the interface, the signed distance function is calculated by the iterative reinitialisation step as suggested by Sussman et al. [159] while the values of the signed distance function at the interface cells are fixed. Noting that the number of iterations in the reinitialisation step is less than 15 times (typically few iterations). The formulation accurately conserves mass, where the author reported that the maximum volume error was less than 10^{-10}. Noting that the volume of fluid methods satisfy conservation of mass accurately when the divergence-free condition is precisely satisfied, where a tolerance of the pressure Poisson equation of 10^{-10} is used.

In conclusion, the coupling approach of the level set and the volume of fluid is employed to integrate the advantages of the mass conservation of the later and to maintain the interface sharpness and the simplicity of computing interfaces geometrical information of the former. The coupling of the two methods was achieved using different implementations which vary in the degree of complexity and the accuracy of the resulted numerical simulations. Generally most of the implementations proceed by solving the advection equation of both methods and require geometrical interface reconstruction. Other implementations advance the interface by only solving the volume fraction advection equation. The numerical results of the CLSVOF has been shown to be superior to those of the standalone implementation of both level set methods and volume of fluid methods.

2.2.4 Summary

Interface capturing methods include Level Set methods [3, 127, 126, 159], Volume of Fluid (VOF) methods [74, 143, 145, 108], THINC methods [187, 186, 106], and Coupled Level Set and Volume of Fluid method (CLSVOF) [158, 180, 204]. Volume of fluid methods use Heaviside step function to represent the interface. These methods are characterised by volume conservation and relatively simple to implement, thus, have become popular in interfacial flow simulations and have been employed in commercial simulation software. Interface reconstruction is necessary to reduce diffusion caused by advection of VOF function. Various techniques have been used for interface reconstruction such as SLIC (simple line interface calculation) method [122] and the PLIC (piecewise line interface calculation) method [104, 61, 105]. Although the PLIC method is considered more accurate than the SLIC method, constructing the method in three dimensions is significantly difficult [204]. In the Level-Set method, the interface is represented by a smooth signed distance function, which is the distance between the interface and the grid points. A distinctive feature of this method is the relatively simple calculation of curvature using this function and is relatively easy to implement as compared to the front tracking methods and VOF methods. Thus, the level set methods have also been widely used. However, the main drawback here is that Level Set method does not conserve volume. The THINC method is a type of

VOF method. THINC method is characterised by non-diffusive VOF function because it makes use of a smoothed Heaviside function. THINC method employs simple and easy to implement interface reconstruction method, namely WLIC [201]. It has been reported that the results by the THINC/WLIC method are almost same with these by the VOF/PLIC method [204]. The CLSVOF method couples both the Level-Set method and VOF method. Thus can take advantage of both schemes, namely, the conservation of mass fraction of the latter, and sharp interface and an easier technique for computing interface curvature of the former.

2.3 Spatial Discretisation Techniques

Many competing numerical methods are used to discretise partial differential equations. The Discretisation is the process of converting the governing partial differential equations to a system of algebraic equations. The most common discretisation methods which are most suitable for computational fluid dynamics software are finite difference method, finite volume method, and finite element method. These methods are related and can be considered as part of a unified framework [49]. Technically, time derivatives are discretised by finite difference method while spatial derivatives are descretised by any of the methods. These spatial discretisation methods can be regarded as single-moment methods because they customarily represent problem variables by one moment either point value such as in finite difference method or cell-integrated average as in finite volume method and obtain the intermediate values between two variables by interpolation. On the other hand, in multi-moment method this interpolation is not necessary because the problem fields are represented by both cell-integrated averages and boundary values. In the following sections, an overview of finite difference method is given in section 2.3.1, followed by an overview of finite volume method in section 2.3.2. Finite element method is presented in section 2.3.3. A summary is given in section 2.3.4.

2.3.1 Finite Difference Method

Finite difference method was first introduced by Courant, Friedrichs and Lewy (1928) [30] in there research on the solution of physical problems using finite differences. They used a finite difference approximation for the wave equation, and the CFL stability condition was shown to be necessary for convergence [165].

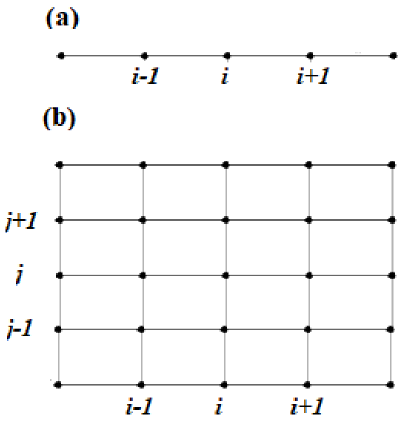


Figure 2.5: A representation of a 1D (a) and 2D (b) Cartesian grid for Finite Difference methods.

In finite difference method [6, 165, 49] the grid is usually locally structured, this means each grid node may be considered the origin of a local coordinate system, whose axes coincide with grid lines [47]. Figure 2.5 depicts Cartesian grids for finite difference method. The method represents solution variables of the governing equations on a set of discrete grid points. Finite difference discretisation is achieved by replacing the derivatives at a grid point by equivalent finite difference approximations. Finite difference approximation of a derivative can simply be obtained via the mathematical definition of a derivative [93] as follows

$$\left(\frac{\partial T}{\partial x}\right)_{x_i} = \lim_{\Delta x \to 0} \frac{T(x_i + \Delta x) - T(x_i)}{\Delta x} \tag{2.8}$$

The first derivative of a variable $T(x)$, $(\partial T/\partial x)_{x_i}$ at a point x_i is the slope of the tangent to the curve of the variable T(x) at that point. Figure (2.6) depicts the meaning of a first derivative and also shows various possible approximations of the derivative at a point. The derivative of $T(x)$ at x_i can be approximated by a straight line passing through two neighbouring points on the curve of $T(x)$. There are three typical and distinct finite difference approximations of a derivative, namely backward difference, forward difference and central difference. These three approximations are shown in fig.(2.6). The former first and second approximations are shown by the dashed line and the dotted line, respectively. Backward difference approximates the derivative $(\partial T/\partial x)_{x_i}$ by the slope of a straight line passing through the point x_i and the point before it $x_i - \Delta x$, hence the term backward. Forward difference, on the other hand, can be obtained from the slope of a straight line passing between x_i and the point next to it $x_i + \Delta x$. Central finite difference approximates the derivative by the slope of a straight line passing between the adjacent points of x_i, i.e., $x_i - \Delta x$ and $x_i + \Delta x$. Central difference is depicted by the line referred to by 'central difference' in fig.(2.6). There are systematic methods to approximate the first-order derivative [47]. Discretisation can be achieved by replacing the derivative at a grid point by equivalent finite difference approximations using Taylor series expansion.

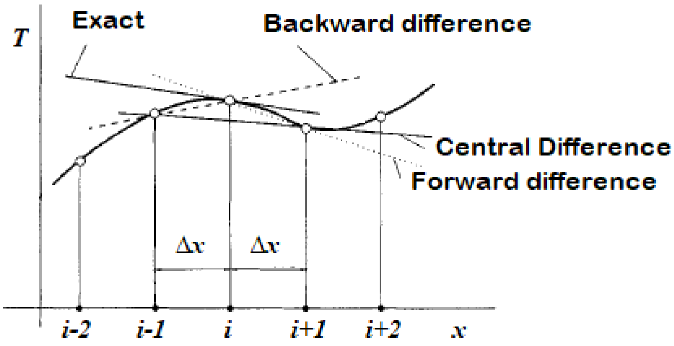


Figure 2.6: Geometric representation of the first-order derivative approximations.

A continuous differentiable function can be expressed as a Taylor series. For instance the function $T(x)$ can be expanded in the neighbourhood of the point x_i by a Taylor series of the following form

$$T(x) = T(x_i) + (x - x_i)\left(\frac{\partial T}{\partial x}\right)_i + \frac{(x - x_i)^2}{2!}\left(\frac{\partial^2 T}{\partial x^2}\right)_i + \ldots, \tag{2.9}$$

where x is any point on the computational grid.

Many possible expressions for the first-order derivative can be obtained from (2.9). For example, derivative approximations which is shown in fig.2.6 can be achieved by substituting the points x_{i-1} and x_{i+1} into equation (2.9) for the backward difference and forward difference, repectively, and ignoring the higher order derivatives in the Taylor series expansion. This can be written as follows

$$\left(\frac{\partial T}{\partial x}\right)_i \approx \frac{T_i - T_{i-1}}{\Delta x}. \tag{2.10}$$

$$\left(\frac{\partial T}{\partial x}\right)_i \approx \frac{T_{i+1} - T_i}{\Delta x}. \tag{2.11}$$

The ignored terms in (2.9) are termed as the truncation error; they specify the approximation accuracy and show the rate at which the error reduces as the grid spacing between points decreases. The truncation error is dominated by the next term in the expansion when $\Delta x \ll 1$ in (2.9) and is typically expressed by $O(\Delta x^m)$, where m is the order of accuracy. It is evident that the error magnitude reduces as the mesh size Δx decreases [49]. Hence, backward difference (2.10) and forward difference (2.11) expressions are both first order accurate $O(\Delta x)$. Similarly, a central finite difference approximation can be obtained by expanding (2.9) at both points x_{i-1} and x_{i+1} yields the following expression,

$$\left(\frac{\partial T}{\partial x}\right)_i = \frac{T_{i+1} - T_{i-1}}{2\Delta x} - \frac{\Delta x^2}{3!}\left(\frac{\partial^3 T}{\partial x^3}\right)_i + \ldots, \tag{2.12}$$

a centred finite difference expression that is second order accurate $O(\Delta x^2)$ is therefore can be given as

$$\left(\frac{\partial T}{\partial x}\right)_i \approx \frac{T_{i+1} - T_{i-1}}{2\Delta x}. \tag{2.13}$$

Finite difference approximations of derivatives can also be achieved using aternative methods in addition to the Taylor series expansion such as polynomial and spline fitting, integral method and control volume approach [6, 49, 47]. A higher-order accurate finite difference approximations can be obtained using any of these methods, for instance, polynomial curves by fitting the function to an interpolation polynomial and differentiate the resulting curve.

Using high-order accurate finite difference expression with sufficiently fine grid generally produces more accurate solution than when using coarse grid. Formulae which involve more than two grid points are used to enhance the accuracy of a finite difference expression such as approximations which include three, five or higher number of grid points. Using longer grid stencils to enhance accuracy is dispensable in single moment methods such as the FD method. Expanding the stencil [20, 144, 6, 47] results in the undesirable properties of creating larger algebraic equations, the numerical treatment complexity of boundary conditions and increasing communication requirements for implementation on parallel computer architectures [154].

Using longer stencils is also not advantageous when the exact solution is not smooth such as in inviscid supersonic flows where discontinuous solution may arise due to the presence of shock waves [183]. Higher-order approximations are not appreciably more accurate than a low-order approximation in the presence of discontinuities. On the other hand, in viscous flows at high Reynolds number where the inertial forces are dominant, discontinuities cannot occur but severe gradients in the flow fields present and hence the exact solution is also not smooth. In the latter case, sufficiently fine grid is necessary for higher-order approximations to produce accurate solution [155]. When severe gradients present, higher-order derivatives are much larger in magnitude than low-order derivatives. Higher-order terms in the Taylor series expansion,i.e., in the truncation error expression do not diminish at such rapid rate as when the flow fields are smooth [49]. Therefore, unless the grid is sufficiently fine the magnitude of the high derivative in the leading term of the truncation error may be so large for the higher-order discretisation that the overall error is comparable to that of a low-order discretisation.

2.3.2 Finite Volume Method

The finite volume method [177, 47] is a common choice for numerical simulations in computational fluid dynamics (CFD), particularly where conservation is important, since most partial differential equations are mathematical statements of underlying conservation laws. Therefore the finite volume method has raised as the preferred strategy in many commercial and open source codes such as FLUENT and OpenFOAM [131]. It was introduced by Patankar in 1980 [128] who developed it for handling fluid flow and heat transfer problems. At present, the finite volume method is powerful in the field of computational fluid dynamics (CFD).

One of the advantages of the method is that local and global conservation is built into the method. This conservativeness of the numerical method is of great significance from physical viewpoint and can be accomplished by expressing fluxes of the conserved quantity through the cell faces of neighbouring computational cells. The direct connection between the physical conservation concepts and the numerical strategy constitutes one of the desirable features of finite volume method. Writing a balance equation for a physical quantity within a control volume makes finite volume method easier to grasp than other numerical strategies. The method also offers the flexibility of application to various mesh structures and geometries. The formulation of the method fits natural execution of boundary conditions, even where the boundaries and related boundary conditions are complicated. The computational grid itself may be either structured or unstructured. The finite volume method works proficiently with either.

In finite volume method the boundaries of the computational cells define the grid instead of the computational points in finite difference grids. The computational variables are represented as cell-integrated averages and located at the cells (control volumes) centres. Figure 2.7 shows a Cartesian finite volume grid. Discretisation in finite volume method, i.e. to replace the integral conservation equations by algebraic equations for control volumes, is achieved by approximating the surface and volume integrals, such as the convection and diffusion terms, using various finite difference formulae. Depending on the employed approximations, the obtained algebraic equations may or

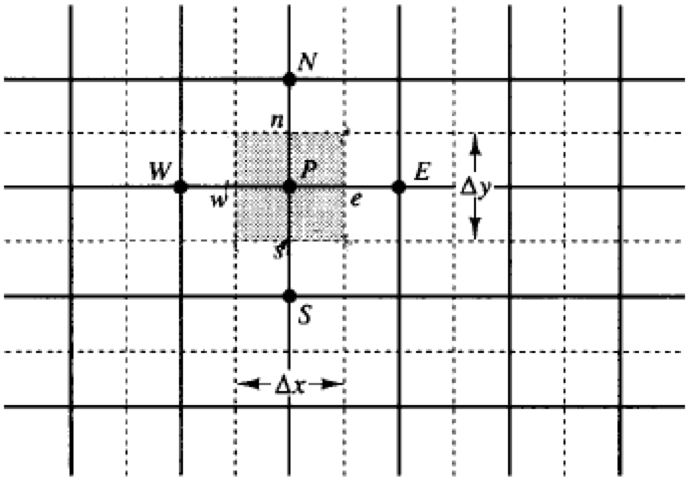


Figure 2.7: A part of 2D finite volume grid [177]. Cells centres are marked by capital letters. centres of cell boundaries are marked by small letters.

may not be similar to those resulted from finite difference method [47].

The approximation of surface integrals including the convective and diffusive terms in the governing equations can be achieved by calculating the net flux across the control volume boundary. It can be obtained by summing the fluxes (convective or diffusive) across all the faces of each control volume in the computational domain. The evaluation of convective and diffusive fluxes over cell boundaries requires the determination of field values and their gradients on computational cells faces. Therefore approximations are required using interpolation between adjacent control volumes since field variables are located at centres of control volumes. A variety of interpolation techniques have been used to calculate field variables at cell boundaries. Each of these techniques has advantages and disadvantages; the most commonly used are the upwind interpolation, linear interpolation and quadratic upwind interpolation.

Also called upwind differencing scheme, the upwind interpolation approximated a field variables at cell boundaries by their values at upwind cell depending on the field direction normal to the cell face. This technique avoid producing oscillatory solutions meaning it is the only technique that assure the boundedness of the solution among other approximations. However, it produces numerical diffusion because it is first-order accurate [47]. Advantage of this approximation is its simplicity, however, a serious disadvantage of this technique is that the numerical diffusion increases in multidimensional simulations if the flow is not normal to the computational cell boundaries. Very dense computational meshes are required to achieve accurate solutions.

Another simple approximation technique is the linear interpolation between the two nearest nodes to the cell face. It is second-order accurate as can be shown by using the Taylor series expansion. This technique is widely used in finite volume method because because it is second-order accurate and its simplicity. It is similar to the central difference approximation in finite difference methods. A disadvantage of this technique is the possibility of oscillatory solutions as in all higher-order techniques. Another variant of the linear interpolation technique is the linear upwind technique which employ extrapolation from two upstream cells. This technique is a second-order accurate and is more complex than the former technique and is more susceptible to numerical oscillation [128].

The other technique that has been used to approximate field variables and their derivatives at cells faces is the quadratic upstream interpolation also known as Quick [103]. This technique uses parabola to interpolate between field values in two adjacent cells; thus requires three points, the third point is chosen in the upwind cell. Quick is a third-order accurate approximation and is more complex than the linear interpolation method particularly in non-uniform computational meshes. In comparison to the linear interpolation technique, Quick is slightly more accurate, however, both approximation techniques show second-order accuracy and the differences are rarely large [47].

Other interpolation techniques such as higher-order techniques and hybrid techniques [72] are also proposed to approximate field variables and their derivatives at cells faces. Higher-order techniques such as the fourth-order central difference scheme [20, 154] which uses higher-order polynomials or splines are constrained by complexity of application and high computational cost, particularly in implicit formulation. For instance, using fourth-order polynomial would result in the requirement of fifteen cells to approximate each cell-face flux which leads to a very expensive equation system in terms of coding and computation. From an accuracy viewpoint, higher-order interpolation techniques do not necessarily result in more accurate solutions because the solution accuracy depends on the computational mesh resolution [177]. Increasing the number of computational cells results in the solution accuracy increase. A highly-accurate solution can be accomplished when the computational mesh is sufficiently fine to capture the details of the phenomenon under consideration. The mesh resolution can be specified by mesh refinement studies.

Another way to approximate the fluxes is the skew upstream differencing techniques [139] which employ upwind extrapolation along streamlines rather than mesh lines. Similar to the previous techniques, a first-order accurate upwind technique and a second-order accurate linear upwind technique have been proposed. However these techniques have higher accuracy than the previous mentioned techniques, they are characterised by rather complex formulations and the high number of necessary extrapolations due to the many possibilities of flow directions. Therefore, these methods are not popular in the computational fluid dynamics field because in addition to the coding difficulties, they also may result in unbounded solutions when the computational mesh is not sufficiently fine.

Hybrid techniques which incorporate two or more techniques to approximate the fluxes were also proposed. The most popular hybrid technique was proposed by Spalding [153] which switches between the upwind technique and the linear interpolation technique based on the local Peclet number. Other hybrid techniques which switch between lower-order and higher-order techniques were also proposed to overcome numerical oscillations in compressible flow applications that involve shock waves [86, 71].

2.3.3 Finite Element Method

Finite element method in engineering was initially developed for structural analysis [13, 17, 28, 29, 179, 212]. Afterwards, it has been applied for various fields of numerical simulations such as fluid flow and heat transfer [164, 10, 14, 4]. The finite element method is part of a class of spatial discretisation techniques called weighted residual methods which also includes spectral methods in their approach to spatial discretisation.

Weighted residual methods [48] differ conceptually from finite difference and finite volume methods in assuming the ability to represent the local change in the solution variables analytically. The concept is achieved by expressing the solution variables by a trail solution using a set of trail piecewise functions with constants which are chosen to give the best solution to the governing equations. For example, a trail solution for the simple heat diffusion equation can take the following form

$$T = \sum_{i=1}^{N} c_i T_i, \tag{2.14}$$

where c_i are unknown constants (or functions) and T_i are specified analytic functions. The trail functions could be polynomials or trigonometric functions while the unknown constants are to be determined by solving a system of equations generated from the governing equations [49]. Noting that the substitution of the trail solution in the governing equations of the problem under consideration yields residual, which is the approximation error. In the weighted residual methods, the constants of the trail functions are chosen in such a way that the residual is forced to equal zero by setting weighted integrals of the residual to zero. The Weighted residual methods uses different formulations of the weighted functions in the weighted integrals such as the subdomain method, the collocation method, the least square method, and the Galerkin method [31]. The later is considered as one of the best formulations for the weighted functions and hence most fluid dynamics and heat transfer problems are formulated by the Galerkin formulation of the finite element method [164].

Spatial discretisation of the physical domain in a finite element model involves subdividing it into a number of geometrically simple domains, called finite elements connected together at their vertices, called nodes. Hence, the computational grid in finite element method is formed by contiguous collection of finite elements connected together by their nodes. In comparison to finite difference methods where the solution variables are represented only at grid nodes (point values), the interpolation is built in the finite element method. The finite element method writes the trail solutions (e.g.

2.14) at the nodes connecting the finite elements. The expression of the trail functions in terms of the nodal unknowns (e.g. 2.14) can be seen as an interpolation between the point solution T_j at each node. Therefore, the trail solution are called interpolation functions or shape function [212]. In other words, spatial discretisation in finite element method can be divided into two steps. Firstly, the physical domain is subdivided into a number of finite elements over which a piecewise interpolation is brought to connect the solution variables to the nodes. Secondly, the solution variables at the finite element nodes are expressed in algebraic equations using the weighted residual formulation.

Several types of interpolation functions are used in finite element method. The main interpolation functions are linear interpolation functions and quadratic interpolation functions. In the linear interpolation, the trail solution vary linearly between the nodes connecting the finite elements. Typically, domain discretisation introduces errors in any spatial discretisation technique. The error introduced by the linear interpolation reduces by increasing the number of elements. In order to enhance the interpolation error introduced by the linear interpolation, quadratic interpolation functions are used. The later is more accurate than the former with the same grid size. Using quadratic interpolation, grid refinement leads to faster reduction in the discretisation error as compared to the linear interpolation. Higher order interpolation functions are more accurate. However, they are seldom used in practise because of the higher computational cost in comparison to the linear and quadratic functions. The computational cost increases in multi-dimensional problems [10].

2.3.4 Summary

Two-phase flow problems are described by partial differential equations (PDE). Many numerical methods exist for the discretization of PDE. Three classical mesh based methods are finite difference methods, finite volume methods and finite element methods. In the following summary, a comparison of the advantages and drawbacks of the spatial discretisation techniques is given.

Finite differences have the advantages of their derivation from a Taylor series expansion and their straightforward implementation for simple problem geometries, however, extension of their application to complex geometries is difficult. The finite volume method employs control volume formulation where the conservation principles for mass, momentum and energy are applied to discrete control volumes by expressing fluxes of the conserved quantities around every mesh cell. The finite volume and finite element methods are well suited for unstructured meshes allowing flexibility in handling complicated geometries and making them popular for fluid dynamics computations.

An important property of discretization methods is their order of convergence, which measures how fast the discretization error decreases when the mesh resolution is increased. First and second order accurate finite difference and finite volume are common but their extension to higher orders is cumbersome to implement. For these methods a higher order approximation usually requires larger interpolation stencils. This not only increases the band-size of the matrix but also makes it difficult to handle boundary conditions. In the finite element method, on the other hand, higher order can be achieved locally inside a compact element, however the computational cost increases when higher-order accuracy is required. An advantage of multi-moment methods is the ability to be extended to higher orders of accuracy by increasing the number of moments in each computational cell. Thus, higher order approximation may not require larger interpolation stencil.

In comparison to finite difference and finite volume methods, the solution error in finite element method is not only related to the type of interpolation functions used on a specific computational grid, it is also affected by an additional error due to the nodal point solution not coinciding with the analytic solution. Generally, the accuracy of using linear interpolation functions and quadratic interpolation functions is equivalent to a second-order and third-order finite difference, respectively [49]. Obtaining solutions of up to second-order accuracy is straightforward in both finite element and finite volume methods on unstructured grids. Obtaining higher-order accurate solutions is easier in finite element method on unstructured grids but with the expense of high computational cost. Finite volume method handles non-conforming meshes more easily and robustly

[6].

Both finite volume and finite element methods feature the ability to handle complex geometries, flexibility in re-meshing the computational domain and the strong mathematical foundation. Both methods uses integral form of the governing equations. In the finite volume method, the physical conservation principles are specified in terms of fluxes in the surface integrals. The finite element method, on the other hand, uses integral equations of the weak form and hence is not exactly conservative, therefore affected by stability issues in physical problems which involve discontinuous processes. In other words, spatial discretisation in methods other than finite volume method, such as the finite difference and finite element methods, does not necessarily leads to the conservation of the transport quantities. Therefore, the numerical solutions may manifest non-physical behaviour. The finite volume method, on the other hand, guarantees the conservation of these quantities, meaning that the strategy is in agreement with the underlying physical laws at the discretisation level [128].

2.4 VSIAM3

Volume/Surface Integrated Average-based Multi-Moment method (VSIAM3), Xiao et al. [188, 183, 184] is a numerical method for compressible and incompressible flow. VSIAM3 employs conservative semi-Lagrangian (CIP-CSL) method [162, 195, 190, 192] to solve the conservative advection equation. VSIAM3 (including CIP-CSL method) is a multi-moment finite-volume method for solving the Navier-Stokes equations. Thus, provides the finite-volume features such as conservation, computational efficiency, and flexibility in handling irregular geometries. VSIAM3 has been applied to various fluid dynamics simulations [188, 182, 183, 184] including complex two-phase flows such as droplet splashing [203, 204, 205] and simulation of a sanitary ware [81].

VSIAM3 including CIP-CSL uses multiple integrated variables (moments) for a physical field. In a three dimensional context, these are the volume integrated average (VIA) and the surface integrated averages (SIA), which are considered as the computational

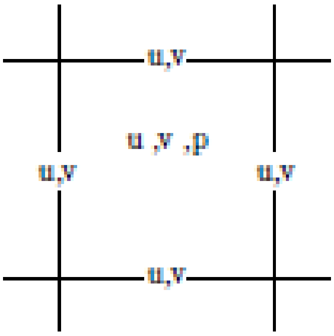


Figure 2.8: A Multi-moment concept. Representation of flow field in a computational cell in two dimensions.

variables and are updated simultaneously in time. For example, the temperature is represented by a VIA located at cell centre and three SIA variables located at the centre of faces of a three-dimensional computational cell. Similarly, in two dimensions, a physical quantity is represented by a cell-average value and two line-integrated averages. In a one-dimensional computations, the multi-moments reduces to a cell-average located at cell centre and a point value on the cell boundary. The computational variables are arranged on a multi-moment fixed grid (M-grid). Figure 2.8 shows the arrangement of flow field in a computational cell in two dimensions. In the M-grid, the arrangement of the boundary values (SIA) of the velocity in VSIAM3 and the cell-integrated average (VIA) pressure form staggered Cartesian grid for the solution of the pressure Poisson equation. Using both (VIA) and (SIA) as the model variables provides a convenient framework with a desirable features such as a flexibility in spatial discretization, a compact grid stencil and high robustness and efficiency in simulating incompressible and compressible two-phase flows. VSIAM3 remarkably makes use of a CIPCSL method [162, 195, 190, 192] as the continuity equation (the advection equation in conservation form) solver because it efficiently deploys multi-moments ((VIA) and (SIA)). CIP-CSL features conservation of the advected quantity, high accuracy, and reduction of the numerical dispersion for advection [195]. The multiple integrated variables (moments) employed by multi-moment methods for a physical field are updated by different formulations yet same conservation laws. The cell-integrated average values are updated by finite volume formulation, that is by calculating the net flux across the cell boundaries. The boundary values, on the other hand, are updated by finite difference formulation (i.e., two different formulations to predict the computational variables).

As the name suggests, VSIAM3 and CIPCSL multi-moment methods can be essentially distinguished from the conventional single-moment finite volume method, as well as other single moment methods such as the finite difference method. Being the other model variables, the boundary values are also stored and updated at each step rather than interpolated from the cell average values. The CIP (Constrained Interpolation Profile) method [193, 194, 196], IDO (interpolated differential operator) method [7] and MCV (multi-moment constrained finite volume) [80] can likewise be categorized in multi-moment methods. On the other hand, the vast majority of numerical strategies in computational fluid dynamics depend on predicting single-moment of the physical quantity for each mesh cell, i.e., either cell-integrated average value or point value as in finite volume method and finite difference method, respectively, such as MUSCL (monotonic upwind-centred scheme for conservation laws) [175], ENO (essentially non-oscillatory) [68] and WENO (weighted ENO) [110, 83].

The CIP-CSL is a conservation equation solver which is employed as a part of VSIAM3. Several CIP-CSL schemes have been proposed for fluid transport, for example, CIP-CSL2 (CIP-CSL with quadratic interpolation function) [195], CIP-CSL3 (CIP-CSL with third order interpolation function) [190] and CIP-CSLR (CSL with rational interpolation function) [192]. These CIP-CSL methods depend on a semi-Lagrangian approach in which a piece-wise interpolation function is fundamental in determining the reconstructed profile and the numerical solution. The CIP-CSL methods formulate the piece-wise interpolation functions using only moments (variables) over one cell and increase the order of accuracy by increasing the number of moments in each cell, while single-moment methods increase the order of accuracy by expanding the number of cells which are used in their spatial discretisation. In the CIP-CSL2 scheme, second-order piece-wise polynomial interpolation function is utilised as the interpolation function, and two boundary values (SIA) and a cell-integrated average (VIA) in the upwind cell were employed as the constraints. In CIP-CSL3, third-order piece-wise polynomial interpolation function is utilised, and two boundary values and a cell-integrated average in the upwind cell are utilised as the constraints, and a gradient in the upwind cell is likewise utilised as a control parameter. CIP-CSL3 is commonly used for compressible flow simulations and the control parameter is utilised as a limiter. CIP-CSLR is a less oscillatory CIP-CSL formulation based on rational interpolation functions.

It has been reported that VSIAM3 is a highly robust and efficient numerical strategy [188, 203, 204, 205]. However, most investigators who attempted to construct the code could not accomplish robust numerical simulations [124]. This is because the multi-moment structure of CIPCSL and VSIAM3 has risen some execution difficulties, and the full detail of the execution of VSIAM3 has not been explained in the previous works [5]. The issue on the robustness in VSIAM3 has also been implied in [102] and a treatment based on the Constrained Interpolation Profile (CIP) interpolation between the computational variables has been proposed. In the present work, an attempt will be carried out to identify the possible reasons of the issue of robustness in VSIAM3 and CIPCSL and to provide the full details of robust implementation of VSIAM3.

2.5 Conclusions

An overview is presented for the main approaches of interface capturing strategies (Volume of Fluid, Level Set and Coupled Level Set and volume of Fluid) as well as the most common spatial discretisation techniques for partial differential equations (FDM, FVM and FEM). The VSIAM3 multi-moment method is also introduced. The advantages and drawbacks of each strategy are explained with the intention of addressing the choice of the Coupled Level Set and THINC/WLIC as an interface capturing scheme and the finite volume based VSIAM3 as the fluid flow and heat transfer solver in the present work.

As it was mentioned, the coupling approach of the level set and the THINC/WLIC is employed to integrate the various advantages of both schemes such as the mass conservation and maintaining the interface sharpness and the simplicity of computing interfaces geometrical information. Moreover, THINC/WLIC with its smoothed Heavisided function (a piecewise hyperbolic tangent function) is a VOF type scheme. The smoothed Heavisided function plays an important role in preventing flotsam and jetsam. This feature allows employing simple interface reconstruction technique, the WLIC. Using THINC/WLIC provides attractive features such as the relative simple implementation and the straightforward extension to three dimensions. As it is mentioned before, it was reported that the results by THINC/WLIC are comparable to VOF/PLIC [204].

VSIAM3 is a promising numerical framework and it require further investigation, development and numerical experimentations. This is because multi-moment methods (e.g. VSIAM3 and CIPCSL) offers numerous possibilities for the discretisation of the governing equations and have the ability to be extended to higher-order of accuracy by using compact stencils. Furthermore, as a finite volume formulation it can naturally employ unstructured grids and handle complex geometries. VSIAM3 is highly efficient and robust numerical framework [188, 203, 204, 205], however the multi-moment structure of VSIAM3 and CIPCSL schemes has risen implementation difficulties, and the full detail of the execution of VSIAM3 has not been explained in the previous works [5].

In the present study, an investigation will be carried out to identify the possible reasons of the issue of robustness in VSIAM3 and CIPCSL and to provide the full details of robust and efficient implementation of VSIAM3 and robust code library will be constructed. One of the main topics in the present work is the velocity divergence term in the conservation equation. This term results from the solution of the conservation equation by CIP-CSL schemes. The velocity divergence term is important and it has been rarely mentioned in the literature of CIP-CSL schemes and VSIAM3 [188, 183], and despite VSIAM3 uses multi-moments of a physical variable no attempt has been done to use various moments in the discretisation of the term. In the present study various discretisation techniques for the velocity divergence term in the conservation equation will be proposed and studied. Furthermore, In the published works of VSIAM3 [188, 182, 189, 81, 183, 184, 80, 203, 204, 205, 5], the numerical framework has not been applied to heat transfer problems, particularly heat transfer problems with interfaces. In the present study, the energy equation will be discretised using VSIAM3 and a robust implementation of VSIAM3 in heat transfer problems will be investigated.

Chapter 3

Numerical Methods

3.1 Introduction

The present work employs a numerical framework based on a one-fluid formulation for the coupled two-phase flows. The numerical framework is based on the following numerical methods, VSIAM3 (Volume/Surface Integrated Average-based Multi-Moment Method) , CIP-CSL (Conservative Semi-Lagrangian method), CLSVOF (Coupled Level Set and Volume of Fluid method), THINC (Tangent of Hyperbola for Interface Capturing), and WLIC (Weighted Line Interface Calculation) method.

In this chapter, The details of the numerical framework are explained. First, the VSIAM3 method for incompressible flows is elucidated in detail in section 3.2. VSIAM3 for compressible flow is given in section 3.3. Then the interface capturing method for the simulation of two-phase flows is described in section 3.4 followed by a surface tension model in section 3.5. The summary comes in Section 4.4.

3.2 VSIAM3 for Incompressible Flows

3.2.1 Equations of Fluid Flow and Heat Transfer

The following governing equations cast in an integral form are used for incompressible flow of Newtonian fluid with constant physical properties

$$\int_{\Gamma} \mathbf{u} \cdot \mathbf{n} dS = 0, \tag{3.1}$$

$$\rho \left(\frac{\partial}{\partial t} \int_{\Omega} \mathbf{u} dV + \int_{\Gamma} \mathbf{u}(\mathbf{u} \cdot \mathbf{n}) dS \right) = - \int_{\Gamma} p \mathbf{n} dS + \int_{\Gamma} (2\mu \mathbf{D}) \cdot \mathbf{n} dS + \int_{\Omega} \mathbf{f} dV + \mathbf{F}_{sf}, \tag{3.2}$$

and for heat transfer

$$\frac{\partial}{\partial t} \int_{\Omega} T dV + \int_{\Gamma} T(\mathbf{u} \cdot \mathbf{n}) dS = \frac{1}{\rho c_p} \int_{\Gamma} q^{''} \mathbf{n} dS \tag{3.3}$$

Equation (3.1) expresses the incompressibility constraint states that the volume of any incompressible fluid element cannot be changed, where $\mathbf{u}$ is the velocity, $\mathbf{n}$ the outgoing normal for the control volume Ω with its surface denoted by Γ (see Fig. 3.2). Equation (3.2) represents the momentum-conservation principle, stating that the rate of change of fluid momentum in the fixed control volume Ω is the difference in momentum flux across the volume boundary Γ plus the net forces acting on the volume. The first and second terms on the left hand side of equation (3.2) are the time rate of change of fluid momentum in the volume Ω and the net momentum flux across the control surface Γ, respectively. The first and second terms on the right hand side of equation (3.2) are the pressure stress and the viscous stresses acting on the control surface, where Where ρ the density, p the pressure, $\mathbf{D}$ the deformation tensor ($\mathbf{D} = 0.5(\nabla u + (\nabla u)^T)$). The third term is the total body force on Ω, where, $\mathbf{f}$ is the body force per unit volume $\mathbf{F_{bf}}$ and is taken as the gravitational force ($\mathbf{f} = \rho \mathbf{g}$) in the present work.

and $\mathbf{F_{sf}}$ are body forces and surface forces, respectively, integrated on the control volume Ω. T is the temperature, c_p is the specific thermal heat capacity, $q^{''}$ is the thermal heat flux ($q^{''} = \kappa \nabla T$). κ is the thermal conductivity. A fractional step approach [202] is used to solve the governing equations. Equations (3.1) and (3.2) are split as follows:

$$\mathbf{u}^{t+\Delta t} = f^{NA4}(f^{NA3}(f^{NA2}(f^{NA1}(f^{A}(\mathbf{u}^{t}))))). \tag{3.4}$$

1. Advection term (f^A):

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}dV + \int_\Gamma \mathbf{u}(\mathbf{u}\cdot\mathbf{n})dS = 0, \tag{3.5}$$

2. Viscous term (non-advection part 1 f^{NA1}):

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}dV = \frac{1}{\rho}\int_\Gamma (2\mu\mathbf{D})\cdot\mathbf{n}dS, \tag{3.6}$$

3. Body forces term (non-advection part 2 f^{NA2}):

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}dV = \frac{\mathbf{F}_{bf}}{\rho}, \tag{3.7}$$

4. Surface tension force (non-advection part 3 f^{NA3}):

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}dV = \frac{\mathbf{F}_{sf}}{\rho}, \tag{3.8}$$

5. Divergence free and pressure gradient (projection step) (non-advection part 4 f^{NA4}):

$$\int_\Gamma \mathbf{u}\cdot\mathbf{n}dS = 0, \tag{3.9}$$

$$\frac{\partial}{\partial t}\int_\Omega \mathbf{u}dV = -\frac{1}{\rho}\int_\Gamma p\mathbf{n}dS. \tag{3.10}$$

These equations are solved by VSIAM3, in which the advection part is solved by a CIP-CSL method. The CIP-CSL is a directional splitting method.

3.2.2 Grid for VSIAM3 (M-Grid)

VSIAM3 utilizes a grid called multi-moment grid (M-grid) [189]. Figure 3.2 depicts the grid in two-dimensions, in which all the components of the velocity field are defined using two different moments, one moment at cell centre $u_{i,j}, v_{i,j}$, and the other on the cell boundaries $u_{i-1/2,j}$, $u_{i,j-1/2}$,$v_{i-1/2,j}$,$v_{i,j-1/2}$ as the cell-integrated average and the line-integrated average, respectively. The temperature field is likewise defined using two different moments, i.e., at cell centre $T_{i,j}$ as the cell average and on all

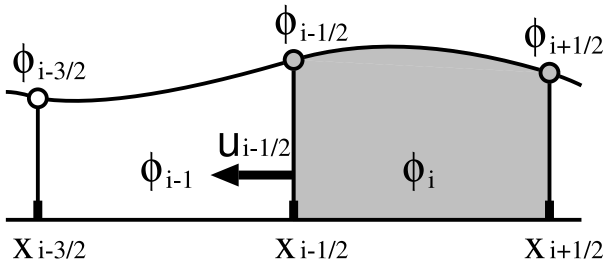


Figure 3.1: Schematic figure of the CIP-CSL2 method. $u_{i-1/2} < 0$ is assumed. The moments which are indicated by gray color ($\phi_{i-1/2}$, ϕ_i and $\phi_{i+1/2}$) are used to construct the quadratic interpolation function.

cell boundaries $T_{i-1/2,j}$, $T_{i,j-1/2}$ as boundary averages. The rest of the computational variables such as the pressure $P_{i,j}$, volume of fluid (VOF) function $\vartheta_{i,j}$ and level set function $\psi_{i,j}$, and the thermophysical properties of fluids such as density $\rho_{i,j}$, viscosity $\mu_{i,j}$, thermal conductivity $\kappa_{i,j}$ are defined only as cell-integrated averages at the cell centre.

The multi-moment grid system has a vital advantage over numerous other grid systems. The degrees of flexibility of every velocity component, in VSIAM3, likewise the temperature are three times in 2D and four times in 3D those in single-moment methods. This is true because these variables have been defined not only at the cell centre but also on cell boundaries as shown in Fig 3.2. The extra degrees of flexibility enhances the computation of the convection term in the conservation laws. VSIAM3, as previously mentioned, utilizes CIP-CSL schemes for solving the advection term.

3.2.3 Definition of Moments in 2D

In two-dimensional domain, a cell-integrated average and boundary-integrated averages are defined as

$$u_{i,j} = \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} u(x,y)dxdy, \tag{3.11}$$

and

$$u_{i-1/2,j} = \frac{1}{\Delta y}\int_{y_{j-1/2}}^{y_{j+1/2}} u(x_{i-1/2}, y)dy, \tag{3.12}$$

$$u_{i,j-1/2} = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} u(x, y_{j-1/2})dx. \tag{3.13}$$

respectively.

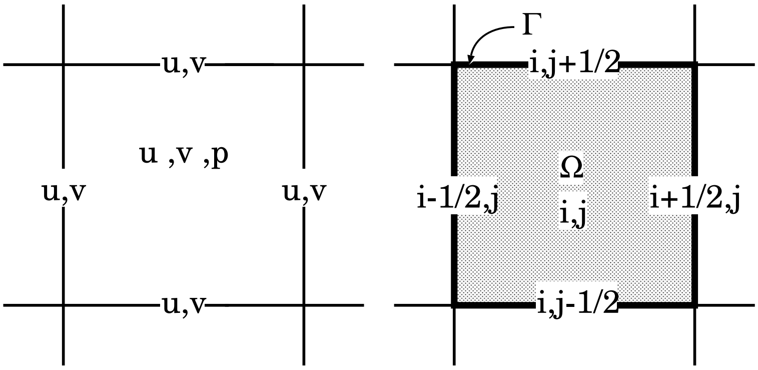


Figure 3.2: Schematic figure of the grid in two dimensional case. $u_{i,j}$ **is the cell average and** $u_{i-1/2,j}$**,** $u_{i+1/2,j}$**,** $v_{i,j-1/2}$ **and** $v_{i,j+1/2}$ **are the boundary values.**

3.2.4 Definition of Moments in 3D

In three-dimensional computations, a cell average and boundary values become volume integrated average (VIA) and surface integrated averages (SIA), respectively. On a three-dimensional grid system, these variables are defined as

$$u_{i,j,k} = \frac{1}{\Delta x \Delta y \Delta z}\int_{x_{i-1/2}}^{x_{i+1/2}}\int_{y_{j-1/2}}^{y_{j+1/2}}\int_{z_{k-1/2}}^{z_{k+1/2}} u(x,y,z)dxdydz, \tag{3.14}$$

and

$$u_{i-1/2,j,k} = \frac{1}{\Delta y \Delta z} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{z_{k-1/2}}^{z_{k+1/2}} u(x_{i-1/2}, y, z) dydz, \tag{3.15}$$

$$u_{i,j-1/2,k} = \frac{1}{\Delta x \Delta z} \int_{x_{j-1/2}}^{x_{j+1/2}} \int_{z_{k-1/2}}^{z_{k+1/2}} u(x, y_{j-1/2}, z) dxdz. \tag{3.16}$$

$$u_{i,j,k-1/2} = \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} u(x, y, z_{k-1/2}) dxdy, \tag{3.17}$$

respectively.

3.2.5 Advection Part (f^A)

The CIP-CSL methods are used to solve the scalar conservation equation

$$\frac{\partial}{\partial t} \int_{\Omega} \phi dV + \int_{\Gamma} \phi(\mathbf{u} \cdot \mathbf{n}) dS = 0, \tag{3.18}$$

here ϕ is a scalar transported quantity. In the following subsections 3.2.5.1 and 3.2.5.3, the CIP-CSL2 method [195] and the CIP-CSLR method [192] are explained, respectively.

3.2.5.1 CIP-CSL2

In the CIP-CSL2 method [195], given the cell average (ϕ_i), and the boundary values ($\phi_{i-1/2}$) and ($\phi_{i+1/2}$) at the nth time step, a quadratic piece-wise interpolation function $\Phi_i(x)$

$$\Phi_i(x) = a_i(x - x_{i-1/2})^2 + b_i(x - x_{i-1/2}) + \phi_{i-1/2}, \tag{3.19}$$

is used to interpolate between $x_{i-1/2}$ and $x_{i+1/2}$ as shown in fig. 3.1. The interpolation function $\Phi_i(x)$ is important in determining the reconstructed profile and the numerical solution. By applying the following constraints

$$\Phi_i(x_{i+1/2}) = \phi_{i+1/2}, \tag{3.20}$$

$$\phi_i = \int_{x_{i-1/2}}^{x_{i+1/2}} \Phi_i(x)dx/\Delta x, \tag{3.21}$$

the coefficients of the interpolation function, a_i and b_i, are determined as follows

$$a_i = \frac{1}{\Delta x^2}(-6\phi_i + 3\phi_{i-1/2} + 3\phi_{i+1/2}), \tag{3.22}$$

$$b_i = \frac{1}{\Delta x}(6\phi_i - 4\phi_{i-1/2} - 2\phi_{i+1/2}). \tag{3.23}$$

By using the interpolation function $\Phi_i(x)$, the boundary value $\phi_{i-1/2}$ can be updated by the conservation equation of a differential form

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = -\phi\frac{\partial u}{\partial x}. \tag{3.24}$$

A splitting approach is used to solve equation (3.24) as follows

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0, \tag{3.25}$$

$$\frac{\partial \phi}{\partial t} = -\phi\frac{\partial u}{\partial x}. \tag{3.26}$$

The numerical solution of the advection equation (3.25) is obtained by using a semi-Lagrangian approach

$$\phi^*_{i-1/2} = \begin{cases} \Phi_{i-1}(x_{i-1/2} - u_{i-1/2}\Delta t) & \text{if} \quad u_{i-1/2} \geq 0 \\ \Phi_i(x_{i-1/2} - u_{i-1/2}\Delta t) & \text{if} \quad u_{i-1/2} < 0. \end{cases} \tag{3.27}$$

(3.26) represents a correction due to the divergence term of the velocity and is solved by a finite difference formulation. The divergence term is one of the main topics in this work as discussed in Section 4.2. The cell average ϕ_i is updated by a finite volume formulation

$$\phi_i^{n+1} = \phi_i^n - \frac{1}{\Delta x}(F_{i+1/2} - F_{i-1/2}), \tag{3.28}$$

here $F_{i-1/2}$ is the flux

$$F_{i-1/2} = \begin{cases} -\int_{x_{i-1/2}}^{x_{i-1/2}-u_{i-1/2}\Delta t} \Phi_{i-1}(x)dx & \text{if} \quad u_{i-1/2} \geq 0 \\ -\int_{x_{i-1/2}}^{x_{i-1/2}-u_{i-1/2}\Delta t} \Phi_i(x)dx & \text{if} \quad u_{i-1/2} < 0. \end{cases} \tag{3.29}$$

3.2.5.2 CIP-CSLR

The CIP-CSLR [192, 191] method is characterized by less numerical oscillations than the other CIP-CSL schemes. In comparison to the CIP-CSL2, a rational interpolation function is employed in the CIP-CSLR method instead of equation (3.19). Although two CIP-CSLR formulations (CIP-CSLR0 and CIP-CSLR1) have been proposed [192], we explain only CIP-CSLR0 here (hereafter CIP-CSLR0 is referred to as CIP-CSLR in this work).

Given the cell average (ϕ_i), and the boundary values $(\phi_{i-1/2})$ and $(\phi_{i+1/2})$ at the nth time step, the reconstruction profile is approximated by the following rational piecewise interpolation function

$$\Phi_i(x) = \frac{\alpha_i \beta_i (x - x_{i-1/2})^2 + 2\alpha_i (x - x_{i-1/2}) + \phi_{i-1/2}}{\left(1 + \beta_i (x - x_{i-1/2})\right)^2}, \tag{3.30}$$

which is used to interpolate between $x_{i-1/2}$ and $x_{i+1/2}$ as shown in fig.3.1. by using the constraints (3.20) and (3.21), the rational function cab be determined and the coefficients α_i and β_i are determined as follows

$$\alpha_i = \beta_i \phi_i + (\phi_i - \phi_{i-1/2})/\Delta x, \tag{3.31}$$

$$\beta_i = \frac{1}{\Delta x}\left(\frac{|(\phi_{i-1/2} - \phi_i)| + \epsilon}{|(\phi_i - \phi_{i+1/2})| + \epsilon} + 1\right). \tag{3.32}$$

Here ϵ is an infinitesimal number to avoid zero division. We used $\epsilon = 10^{-16}$ for all results in this work. All other procedures are the same with these in CIP-CSL2.

3.2.5.3 Complex Wave Advection Problem

To show characteristics of CIP-CSL2 and CIPCSLR, a one-dimensional Jiang-Shu complex wave advection problem was carried out using a constant advection velocity of $u = 1$. The initial condition [83] is given by

$$\phi(x) = \begin{cases} \frac{1}{6}\left(G(x,\beta,\zeta-\delta)+G(x,\beta,\zeta+\delta)+4G(x,\beta,\zeta)\right) & (-0.8 \leq x \leq -0.6), \\ 1 & (-0.4 \leq x \leq -0.2), \\ 1-|10(x-0.1)| & (0.0 \leq x \leq 0.2), \\ \frac{1}{6}\left(F(x,\alpha,a-\delta)+F(x,\alpha,a+\delta)+4F(x,\alpha,a)\right) & (0.4 \leq x \leq 0.6), \\ 0 & \text{otherwise}, \end{cases} \tag{3.33}$$

where $G(x,\beta,\zeta) = e^{-\beta(x-\zeta)^2}$, $F(x,\alpha,a) = \sqrt{max(1-\alpha^2(x-a)^2, 0)}$, $a = 0.5$, $\zeta = -0.7$, $\delta = 0.005$, $\alpha = 10$, and $\beta = \frac{\log 2}{36\delta^2}$. Periodic boundary conditions are used in the test. The results of the 1D advection using CIP-CSL2 and CIP-CSLR are presented in figures 3.3 and 3.4, respectively. Using CIP-CSL2, results in the generation of numerical oscillation around discontinuities as shown in fig.3.3. Non-oscillatory result is obtained using CIP-CSLR. However, CIP-CSL2 result shows higher accuracy than CIP-CSLR result.

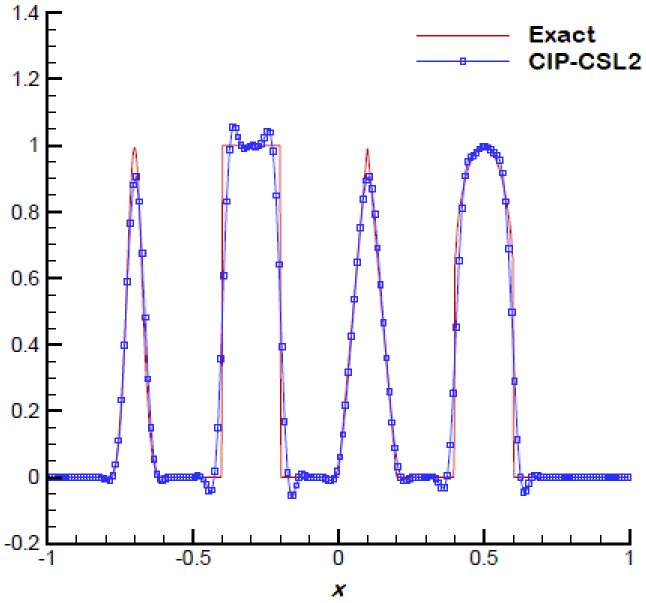


Figure 3.3: Numerical result of complex wave advection problem using CIP-CSL2. The advected wave $\phi(x)$ is plotted vs. the x-axis.

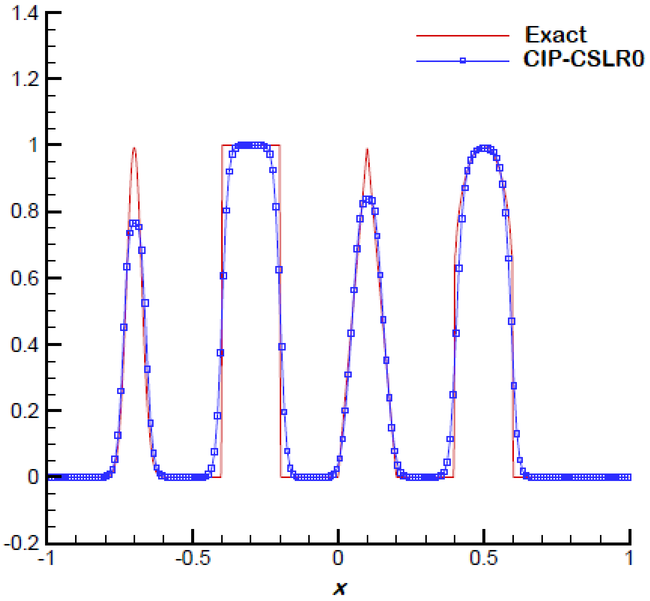


Figure 3.4: Numerical result of complex wave advection problem using CIP-CSLR. The advected wave $\phi(x)$ is plotted vs. the x-axis.

3.2.5.4 Multi-Dimensional Cases

For multi-dimensional cases, a dimensional splitting method [188] based on a one dimensional CIP-CSL scheme is used. In a two-dimensional case, the advection solver can be constructed as follows

1. **For x-direction**, $\phi^*_{i,j}$ and $\phi^*_{i-1/2,j}$ are firstly computed from $\phi^n_{i,j}$ and $\phi^n_{i-1/2,j}$ by using 1D CIP-CSL solver. However $\phi^n_{i,j-1/2}$ cannot be updated by using 1D CIP-CSL solver. Therefore $\phi^n_{i,j-1/2}$ is updated by TEC (Time Evolution Converting) as follows:

$$\phi^*_{i,j-1/2} = \phi^n_{i,j-1/2} + \frac{1}{2}(\phi^*_{i,j} - \phi^n_{i,j} + \phi^*_{i,j-1} - \phi^n_{i,j-1}). \tag{3.34}$$

2. **For y-direction** a similar approach is used . $\phi^{n+1}_{i,j}$ and $\phi^{n+1}_{i,j-1/2}$ are computed from $\phi^*_{i,j}$ and $\phi^*_{i-1/2,j}$ by using a 1D CIP-CSL method. $\phi^*_{i-1/2,j}$ is updated by

TEC as follows:

$$\phi^{n+1}_{i-1/2,j} = \phi^{*}_{i-1/2,j} + \frac{1}{2}(\phi^{n+1}_{i,j} - \phi^{*}_{i,j} + \phi^{n+1}_{i-1,j} - \phi^{*}_{i-1,j}). \quad (3.35)$$

The TEC formula is an approximation for the time derivative by using linear interpolation.

Similar approach is used for solving the advection equation in three dimensions as follows

1. **For x-direction**, $\phi^{*}_{i,j,k}$ and $\phi^{*}_{i-1/2,j,k}$ are firstly computed from $\phi^{n}_{i,j,k}$ and $\phi^{n}_{i-1/2,j,k}$ by using 1D CIP-CSL solver. Next $\phi^{n}_{i,j-1/2,k}$ and $\phi^{n}_{i,j,k-1/2}$ are updated by TEC (Time Evolution Converting) as follows:

$$\phi^{*}_{i,j-1/2,k} = \phi^{n}_{i,j-1/2,k} + \frac{1}{2}(\phi^{*}_{i,j,k} - \phi^{n}_{i,j,k} + \phi^{*}_{i,j-1,k} - \phi^{n}_{i,j-1,k}). \quad (3.36)$$

and

$$\phi^{*}_{i,j,k-1/2} = \phi^{n}_{i,j,k-1/2} + \frac{1}{2}(\phi^{*}_{i,j,k} - \phi^{n}_{i,j,k} + \phi^{*}_{i,j,k-1} - \phi^{n}_{i,j,k-1}). \quad (3.37)$$

2. **For y-direction**, $\phi^{**}_{i,j,k}$ and $\phi^{**}_{i,j-1/2,k}$ are computed from $\phi^{*}_{i,j,k}$ and $\phi^{*}_{i,j-1/2,k}$ by the one dimensional CIP-CSL method. $\phi_{i-1/2,j,k}$ and $\phi_{i,j,k-1/2}$ are updated by TEC as follows:

$$\phi^{**}_{i-1/2,j,k} = \phi^{*}_{i-1/2,j,k} + \frac{1}{2}(\phi^{**}_{i,j,k} - \phi^{*}_{i,j,k} + \phi^{**}_{i-1,j,k} - \phi^{*}_{i-1,j,k}) \quad (3.38)$$

and

$$\phi^{**}_{i,j,k-1/2} = \phi^{*}_{i,j,k-1/2} + \frac{1}{2}(\phi^{**}_{i,j,k} - \phi^{*}_{i,j,k} + \phi^{**}_{i,j,k-1} - \phi^{*}_{i,j,k-1}). \quad (3.39)$$

3. **For z-direction**, $\phi^{n+1}_{i,j,k}$ and $\phi^{n+1}_{i,j,k-1/2}$ are computed from $\phi^{**}_{i,j,k}$ and $\phi^{**}_{i,j,k-1/2}$ by the one dimensional CIP-CSL method. $\phi_{i-1/2,j,k}$ and $\phi_{i,j-1/2,k}$ are updated by TEC as follows:

$$\phi^{n+1}_{i-1/2,j,k} = \phi^{**}_{i-1/2,j,k} + \frac{1}{2}(\phi^{n+1}_{i,j,k} - \phi^{**}_{i,j,k} + \phi^{n+1}_{i-1,j,k} - \phi^{**}_{i-1,j,k}) \quad (3.40)$$

and

$$\phi^{n+1}_{i,j-1/2,k} = \phi^{*}_{i,j-1/2,k} + \frac{1}{2}(\phi^{n+1}_{i,j,k} - \phi^{**}_{i,j,k} + \phi^{n+1}_{i,j-1,k} - \phi^{**}_{i,j-1,k}). \quad (3.41)$$

3.2.6 Stability Criterion

The stability criterion for the advection equation is discussed in the literature e.g. [47] and [155]. The fractional step method described above is performed for each time step. The stability of the scheme is governed by the CFL condition as a criterion. In the numerical code, adaptive time stepping is normally used. The time step is then automatically adjusted using the CFL-condition as a criterion. Adjusting the time step during simulation gives efficient computations and robust code. The time step size is adjusted by computing the maximum velocity in each grid cell in each direction

$$\Delta t_x = \frac{\Delta x}{u^{max}}, \tag{3.42}$$

$$\Delta t_y = \frac{\Delta y}{v^{max}}, \tag{3.43}$$

$$\Delta t_z = \frac{\Delta z}{w^{max}}, \tag{3.44}$$

and considering the minimum in the computational domain

$$\Delta t = cmin\left(\Delta t_x, \Delta t_y, \Delta t_z\right), \tag{3.45}$$

where u^{max}, v^{max}, w^{max} is the maximum velocity in the x, y, and z, respectively. Δx, Δy, Δz is the corresponding cell size where the maximum velocity occurs in the x, y, and z, respectively. Δt is the time step size. c is the CFL number. Our numerical experiments indicate that the scheme is stable when using $c = 0.2$.

3.2.7 Viscous Term (Non-Advection Part 1 f^{NA1})

The viscous term is computed by a standard finite volume formulation for cell averages. For constant fluid viscosity, $\left(\int_\Gamma (2\mu\mathbf{D}) \cdot \mathbf{ndS}\right)$ is simplified to $\left(\int_\Gamma \tau \cdot \mathbf{ndS}\right)$.

$$\frac{1}{\rho}\int_{\Gamma}\tau\cdot\mathbf{n}dS=\frac{1}{\rho_{i,j,k}}\left(\frac{\tau_{i+1/2,j,k}-\tau_{i-1/2,j,k}}{\Delta x}+\frac{\tau_{i,j+1/2,k}-\tau_{i,j-1/2,k}}{\Delta y}+\frac{\tau_{i,j,k+1/2}-\tau_{i,j,k-1/2}}{\Delta z}\right)=\mathbf{D}. \tag{3.46}$$

For the general case, by substituting for the viscous stress, the viscous term of equation (3.46) in the x-direction, D_x, becomes

$$\begin{aligned} D_{x,i,j,k}=\frac{1}{\rho_{i,j,k}}\Bigg(&\frac{\mu_{i+1/2,j,k}\left(\frac{\partial u}{\partial x}\right)_{i+1/2,j,k}-\mu_{i-1/2,j,k}\left(\frac{\partial u}{\partial x}\right)_{i-1/2,j,k}}{\Delta x}\\ &+\frac{\mu_{i,j+1/2,k}\left(\frac{\partial u}{\partial y}\right)_{i,j+1/2,k}-\mu_{i,j-1/2,k}\left(\frac{\partial u}{\partial y}\right)_{i,j-1/2,k}}{\Delta y}\\ &+\frac{\mu_{i,j,k+1/2}\left(\frac{\partial u}{\partial z}\right)_{i,j,k+1/2}-\mu_{i,j,k-1/2}\left(\frac{\partial u}{\partial z}\right)_{i,j,k-1/2}}{\Delta z}\Bigg), \end{aligned} \tag{3.47}$$

substituting for the shear strain, equation (3.47) can be written as

$$\begin{aligned} D_{x,i,j,k}=\frac{1}{\rho_{i,j,k}}\Bigg(&\frac{\mu_{i+1/2,j,k}\left(u_{i+1,j,k}-u_{i,j,k}\right)-\mu_{i-1/2,j,k}\left(u_{i,j,k}-u_{i-1,j,k}\right)}{(\Delta x)^2}\\ &+\frac{\mu_{i,j+1/2,k}\left(u_{i,j+1,k}-u_{i,j,k}\right)-\mu_{i,j-1/2,k}\left(u_{i,j,k}-u_{i,j-1,k}\right)}{(\Delta y)^2}\\ &+\frac{\mu_{i,j,k+1/2}\left(u_{i,j,k+1}-u_{i,j,k}\right)-\mu_{i,j,k-1/2}\left(u_{i,j,k}-u_{i,j,k-1}\right)}{(\Delta z)^2}\Bigg), \end{aligned} \tag{3.48}$$

where the boundary values of the viscosity are the average of the cell-centre values, for example, $\mu_{i+1/2,j,k}=(\mu_{i,j,k}+\mu_{i+1,j,k})/2$. Similarly, in the y-direction we have

$$\begin{aligned} D_{y,i,j,k}=\frac{1}{\rho_{i,j,k}}\Bigg(&\frac{\mu_{i+1/2,j,k}\left(v_{i+1,j,k}-v_{i,j,k}\right)-\mu_{i-1/2,j,k}\left(v_{i,j,k}-v_{i-1,j,k}\right)}{(\Delta x)^2}\\ &+\frac{\mu_{i,j+1/2,k}\left(v_{i,j+1,k}-v_{i,j,k}\right)-\mu_{i,j-1/2,k}\left(v_{i,j,k}-v_{i,j-1,k}\right)}{(\Delta y)^2}\\ &+\frac{\mu_{i,j,k+1/2}\left(v_{i,j,k+1}-v_{i,j,k}\right)-\mu_{i,j,k-1/2}\left(v_{i,j,k}-v_{i,j,k-1}\right)}{(\Delta z)^2}\Bigg), \end{aligned} \tag{3.49}$$

and in the z-direction we have

$$
\begin{aligned}
D_{z,i,j,k} = \frac{1}{\rho_{i,j,k}} \Bigg(& \frac{\mu_{i+1/2,j,k}\left(w_{i+1,j,k} - w_{i,j,k}\right) - \mu_{i-1/2,j,k}\left(w_{i,j,k} - w_{i-1,j,k}\right)}{(\Delta x)^2} \\
& + \frac{\mu_{i,j+1/2,k}\left(w_{i,j+1,k} - w_{i,j,k}\right) - \mu_{i,j-1/2,k}\left(w_{i,j,k} - w_{i,j-1,k}\right)}{(\Delta y)^2} \\
& + \frac{\mu_{i,j,k+1/2}\left(w_{i,j,k+1} - w_{i,j,k}\right) - \mu_{i,j,k-1/2}\left(w_{i,j,k} - w_{i,j,k-1}\right)}{(\Delta z)^2} \Bigg),
\end{aligned} \tag{3.50}
$$

The boundary values $u_{i-1/2,j,k}$, $u_{i,j-1/2,k}$,$u_{i,j,k-1/2}$, $v_{i-1/2,j,k}$, $v_{i,j-1/2,k}$, $v_{i,j,k-1/2}$, $w_{i-1/2,j,k}$, $w_{i,j-1/2,k}$ and $w_{i,j,k-1/2}$ are updated by TEC as explained in the previous subsection. The body force $\mathbf{F}_{bf}$ and the surface forces $\mathbf{F}_{sf}$ are also calculated similarly.

3.2.8 Divergence Free and Pressure Gradient (Projection Step) (Non-Advection Part 4 f^{NA4})

By using the divergence of equation (3.10) and $\int_\Gamma \mathbf{u}^{n+1} \cdot \mathbf{n} dS = 0$, the following Poisson equation

$$
\int_\Gamma \frac{\nabla p^{n+1}}{\rho} \cdot \mathbf{n} dS = \frac{1}{\Delta t} \int_\Gamma \mathbf{u}^* \cdot \mathbf{n} dS, \tag{3.51}
$$

is obtained, where $\mathbf{u}^*$ is the velocity after non-advection $part1$. (3.51) is discretized as

$$
\begin{aligned}
& \frac{(\frac{1}{\rho^{n+1}_{i+1/2,j,k}} \partial_x p^{n+1})_{i+1/2,j,k} - (\frac{1}{\rho^{n+1}_{i-1/2,j}} \partial_x p^{n+1})_{i-1/2,j}}{\Delta x} + \\
& \frac{(\frac{1}{\rho^{n+1}_{i,j+1/2,k}} \partial_y p^{n+1})_{i,j+1/2,k} - (\frac{1}{\rho^{n+1}_{i,j-1/2,k}} \partial_x p^{n+1})_{i,j-1/2,k}}{\Delta y} + \\
& \frac{(\frac{1}{\rho^{n+1}_{i,j,k+1/2}} \partial_y p^{n+1})_{i,j,k+1/2} - (\frac{1}{\rho^{n+1}_{i,j,k-1/2}} \partial_x p^{n+1})_{i,j,k-1/2}}{\Delta y} = \\
& \frac{1}{\Delta t} (\frac{u^*_{i+1/2,j,k} - u^*_{i-1/2,j,k}}{\Delta x} + \frac{v^*_{i,j+1/2,k} - v^*_{i,j-1/2,k}}{\Delta y} + \frac{w^*_{i,j,k+1/2} - w^*_{i,j,k-1/2}}{\Delta z}),
\end{aligned} \tag{3.52}
$$

here

$$
(\frac{1}{\rho^{n+1}_{i-1/2,j,k}} \partial_x p^{n+1})_{i-1/2,j,k} \equiv \frac{2}{\rho^{n+1}_{i,j,k} + \rho^{n+1}_{i-1,j,k}} \frac{p^{n+1}_{i,j,k} - p^{n+1}_{i-1,j,k}}{\Delta x}, \tag{3.53}
$$

$$(\frac{1}{\rho^{n+1}_{i,j-1/2,k}}\partial_y p^{n+1})_{i,j-1/2,k} \equiv \frac{2}{\rho^{n+1}_{i,j,k}+\rho^{n+1}_{i,j-1,k}}\frac{p^{n+1}_{i,j,k}-p^{n+1}_{i,j-1,k}}{\Delta y}, \tag{3.54}$$

and

$$(\frac{1}{\rho^{n+1}_{i,j,k-1/2}}\partial_z p^{n+1})_{i,j,k-1/2} \equiv \frac{2}{\rho^{n+1}_{i,j,k}+\rho^{n+1}_{i,j,k-1}}\frac{p^{n+1}_{i,j,k}-p^{n+1}_{i,j,k-1}}{\Delta z}. \tag{3.55}$$

A preconditioned conjugate gradient (CG) method [12, 70] is used for the pressure Poisson equation. The convergence tolerance of the pressure Poisson equation $\epsilon_p = 10^{-10}$ is used in this work. By using p^{n+1}, the boundary values of the velocity ($u_{i-1/2,j,k}$, $v_{i,j-1/2,k}$, and $w_{i,j,k-1/2}$) are updated as follows

$$u^{n+1}_{i-1/2,j,k} = u^*_{i-1/2,j,k} - \frac{\Delta t}{\rho_{i-1/2,j,k}}\left(\partial_x p^{n+1}\right)_{i-1/2,j,k}, \tag{3.56}$$

$$v^{n+1}_{i,j-1/2,k} = v^*_{i,j-1/2,k} - \frac{\Delta t}{\rho_{i,j-1/2,k}}\left(\partial_y p^{n+1}\right)_{i,j-1/2,k}, \tag{3.57}$$

$$w^{n+1}_{i,j,k-1/2} = w^*_{i,j,k-1/2} - \frac{\Delta t}{\rho_{i,j,k-1/2}}\left(\partial_z p^{n+1}\right)_{i,j,k-1/2}. \tag{3.58}$$

Other velocity components ($u_{i,j,k}$, $v_{i,j,k}$, $w_{i,j,k}$, $u_{i,j-1/2,k}$, $u_{i,j,k-1/2}$, $v_{i-1/2,j,k}$, $v_{i,j,k-1/2}$, $w_{i-1/2,j,k}$, $w_{i,j-1/2,k}$,) are updated by the TEC formula.

3.2.9 The Energy Equation

In a similar manner to equation (3.2), the energy equation (3.3) is split as follows:

$$T^{t+\Delta t} = (f^{NA}(f^{A}(T^t))), \tag{3.59}$$

1. Convection term (f^A):

$$\frac{\partial}{\partial t}\int_\Omega T dV + \int_\Gamma T(\mathbf{u}\cdot\mathbf{n})dS = 0, \tag{3.60}$$

2. Conduction term (non-advection part f^{NA}):

$$\frac{\partial}{\partial t}\int_\Omega T dV = \frac{1}{\rho c_p}\int_\Gamma q^{''}\mathbf{n}dS. \tag{3.61}$$

The convection term (f^A) is solved by using the CIP-CSL methods as explained in section 3.2.5. The conduction term is discritized by a standard finite volume formulation for the cell average values. Letting $\left(\frac{1}{\rho c_p}\int_\Gamma q^{''}\mathbf{n}dS = C\right)$, the conduction term for the cell average can be written as

$$C_{i,j,k} = \frac{1}{(\rho c_p)_{i,j,k}}\left(\frac{q^{''}_{i+1/2,j,k} - q^{''}_{i-1/2,j,k}}{\Delta x} + \frac{q^{''}_{i,j+1/2,k} - q^{''}_{i,j-1/2,k}}{\Delta y} + \frac{q^{''}_{i,j,k+1/2} - q^{''}_{i,j,k-1/2}}{\Delta z}\right). \tag{3.62}$$

Substituting for the heat flux ($q^{''}$), the conduction term (3.62) becomes

$$\begin{aligned} C_{i,j,k} = \frac{1}{(\rho c_p)_{i,j,k}} & \left(\frac{\kappa_{i+1/2,j,k}\left(\frac{\partial T}{\partial x}\right)_{i+1/2,j,k} - \kappa_{i-1/2,j,k}\left(\frac{\partial T}{\partial x}\right)_{i-1/2,j,k}}{\Delta x}\right. \\ & + \frac{\kappa_{i,j+1/2,k}\left(\frac{\partial T}{\partial y}\right)_{i,j+1/2,k} - \kappa_{i,j-1/2,k}\left(\frac{\partial T}{\partial y}\right)_{i,j-1/2,k}}{\Delta y} \\ & \left. + \frac{\kappa_{i,j,k+1/2}\left(\frac{\partial T}{\partial z}\right)_{i,j,k+1/2} - \kappa_{i,j,k-1/2}\left(\frac{\partial T}{\partial z}\right)_{i,j,k-1/2}}{\Delta z}\right). \end{aligned} \tag{3.63}$$

Substituting for the temperature derivatives, equation (3.63) can be written as

$$\begin{aligned} C_{i,j,k} = \frac{1}{(\rho c_p)_{i,j,k}} & \left(\frac{\kappa_{i+1/2,j,k}\left(T_{i+1,j,k} - T_{i,j,k}\right) - \kappa_{i-1/2,j,k}\left(T_{i,j,k} - T_{i-1,j,k}\right)}{(\Delta x)^2}\right. \\ & + \frac{\kappa_{i,j+1/2,k}\left(T_{i,j+1,k} - T_{i,j,k}\right) - \kappa_{i,j-1/2,k}\left(T_{i,j,k} - T_{i,j-1,k}\right)}{(\Delta y)^2} \\ & \left. + \frac{\kappa_{i,j,k+1/2}\left(T_{i,j,k+1} - T_{i,j,k}\right) - \kappa_{i,j,k-1/2}\left(T_{i,j,k} - T_{i,j,k-1}\right)}{(\Delta z)^2}\right), \end{aligned} \tag{3.64}$$

here the boundary values of the conductivity are the average of the cell-centre values, for instance, $\kappa_{i+1/2,j,k} = (\kappa_{i,j,k} + \kappa_{i+1,j,k})/2$.

The boundary values $T_{i-1/2,j,k}$, $T_{i,j-1/2,k}$,$T_{i,j,k-1/2}$ are updated by TEC.

Our numerical experiments show numerical oscillations when using the TEC formulation to update the boundary values of the temperature. These oscillations affect first the boundary values of the temperature and then affect the cell averages at later time of computations. To resolve this issue, we have suggested that the boundary values of the temperature ($T_{i-1/2,j,k}$, $T_{i,j-1/2,k}$, $T_{i,j,k-1/2}$) are also updated using a finite volume formulation in a similar manner to equation (3.62). For example, integrating $\left(\frac{1}{\rho c_p}\int_\Gamma q^{''}\mathbf{n}dS = C\right)$ around a control volume containing ($T_{i-1/2,j,k}$) gives

$$\begin{aligned} C_{i-1/2,j,k} = \frac{1}{(\rho c_p)_{i-1/2,j,k}} \Bigg(& \frac{q^{''}_{i,j,k} - q^{''}_{i-1,j,k}}{\Delta x} + \frac{q^{''}_{i-1/2,j+1/2,k} - q^{''}_{i-1/2,j-1/2,k}}{\Delta y} \\ & + \frac{q^{''}_{i-1/2,j,k+1/2} - q^{''}_{i-1/2,j,k-1/2}}{\Delta z} \Bigg), \end{aligned} \tag{3.65}$$

Substituting for the heat flux ($q^{''}$) into (3.65) gives

$$\begin{aligned} C_{i-1/2,j,k} = \frac{1}{(\rho c_p)_{i-1/2,j,k}} \Bigg(& \frac{\kappa_{i,j,k}\left(\frac{\partial T}{\partial x}\right)_{i,j,k} - \kappa_{i-1,j,k}\left(\frac{\partial T}{\partial x}\right)_{i-1,j,k}}{\Delta x} \\ & + \frac{\kappa_{i-1/2,j+1/2,k}\left(\frac{\partial T}{\partial y}\right)_{i-1/2,j+1/2,k} - \kappa_{i-1/2,j-1/2,k}\left(\frac{\partial T}{\partial y}\right)_{i-1/2,j-1/2,k}}{\Delta y} \\ & + \frac{\kappa_{i-1/2,j,k+1/2}\left(\frac{\partial T}{\partial z}\right)_{i-1/2,j,k+1/2} - \kappa_{i-1/2,j,k-1/2}\left(\frac{\partial T}{\partial z}\right)_{i-1/2,j,k-1/2}}{\Delta z} \Bigg). \end{aligned} \tag{3.66}$$

Substituting for the temperature derivatives, (3.63) can be written as

$$\begin{aligned} C_{i-1/2,j,k} = \frac{1}{(\rho c_p)_{i-1/2,j,k}} \Bigg(& \frac{\kappa_{i,j,k}\left(T_{i+1/2,j,k} - T_{i-1/2,j,k}\right) - \kappa_{i-1,j,k}\left(T_{i-1/2,j,k} - T_{i-3/2,j,k}\right)}{(\Delta x)^2} \\ & + \frac{\kappa_{i-1/2,j+1/2,k}\left(T_{i-1/2,j+1,k} - T_{i-1/2,j,k}\right) - \kappa_{i-1/2,j-1/2,k}\left(T_{i-1/2,j,k} - T_{i-1/2,j-1,k}\right)}{(\Delta y)^2} \\ & + \frac{\kappa_{i-1/2,j,k+1/2}\left(T_{i-1/2,j,k+1} - T_{i-1/2,j,k}\right) - \kappa_{i-1/2,j,k-1/2}\left(T_{i-1/2,j,k} - T_{i-1/2,j,k-1}\right)}{(\Delta z)^2} \Bigg). \end{aligned} \tag{3.67}$$

In a similar manner, the integration of $\left(\frac{1}{\rho c_p}\int_\Gamma q^{''}\mathbf{n}dS = C\right)$ around a control volume containing ($T_{i,j-1/2,k}$) gives

$$\begin{aligned}C_{i,j-1/2,k} = \frac{1}{(\rho c_p)_{i,j-1/2,k}} \Bigg(& \frac{q''_{i+1/2,j-1/2,k} - q''_{i-1/2,j-1/2,k}}{\Delta x} + \frac{q''_{i,j,k} - q''_{i,j-1,k}}{\Delta y} \\ & + \frac{q''_{i,j-1/2,k+1/2} - q''_{i,j-1/2,k-1/2}}{\Delta z} \Bigg),\end{aligned} \tag{3.68}$$

Substituting for the heat flux into equation (3.68) gives

$$\begin{aligned}C_{i,j-1/2,k} = \frac{1}{(\rho c_p)_{i,j-1/2,k}} \Bigg(& \\ & \frac{\kappa_{i+1/2,j-1/2,k}\left(T_{i+1,j-1/2,k} - T_{i,j-1/2,k}\right) - \kappa_{i-1/2,j-1/2,k}\left(T_{i,j-1/2,k} - T_{i-1,j-1/2,k}\right)}{(\Delta x)^2} \\ & + \frac{\kappa_{i,j,k}\left(T_{i,j,k} - T_{i,j-1,k}\right) - \kappa_{i,j-1,k}\left(T_{i,j,k} - T_{i,j-1,k}\right)}{(\Delta y)^2} \\ & + \frac{\kappa_{i,j-1/2,k+1/2}\left(T_{i,j-1/2,k+1} - T_{i,j-1/2,k}\right) - \kappa_{i,j-1/2,k-1/2}\left(T_{i,j-1/2,k} - T_{i,j-1/2,k-1}\right)}{(\Delta z)^2} \Bigg),\end{aligned} \tag{3.69}$$

Similarly, the integration of $\left(\frac{1}{\rho c_p}\int_\Gamma q''\mathbf{n}dS = C\right)$ around a control volume containing ($T_{i,j,k-1/2}$) gives

$$\begin{aligned}C_{i,j,k-1/2} = \frac{1}{(\rho c_p)_{i,j,k-1/2}} \Bigg(& \frac{q''_{i+1/2,j,k-1/2} - q''_{i-1/2,j,k-1/2}}{\Delta x} \\ & + \frac{q''_{i,j+1/2,k-1/2} - q''_{i,j-1/2,k-1/2}}{\Delta y} + \frac{q''_{i,j,k} - q''_{i,j,k-1}}{\Delta z} \Bigg).\end{aligned} \tag{3.70}$$

$$\begin{aligned}C_{i,j,k-1/2} = \frac{1}{(\rho c_p)_{i,j,k-1/2}} \Bigg(& \frac{q''_{i+1/2,j,k-1/2} - q''_{i-1/2,j,k-1/2}}{\Delta x} + \frac{q''_{i,j+1/2,k-1/2} - q''_{i,j-1/2,k-1/2}}{\Delta y} \\ & + \frac{q''_{i,j,k} - q''_{i,j,k-1}}{\Delta z} \Bigg).\end{aligned} \tag{3.71}$$

Substituting for the heat flux, equation (3.71) can be written as

$$
\begin{aligned}
C_{i,j,k-1/2} = \frac{1}{(\rho c_p)_{i-1/2,j,k-1/2}} \Bigg(& \\
& \frac{\kappa_{i+1/2,j,k-1/2}\left(T_{i+1,j,k-1/2} - T_{i,j,k-1/2}\right) - \kappa_{i-1/2,j,k-1/2}\left(T_{i,j,k-1/2} - T_{i-1,j,k-1/2}\right)}{(\Delta x)^2} \\
& + \frac{\kappa_{i,j+1/2,k-1/2}\left(T_{i,j+1,k-1/2} - T_{i,j,k-1/2}\right) - \kappa_{i,j-1/2,k-1/2}\left(T_{i,j,k-1/2} - T_{i,j-1,k-1/2}\right)}{(\Delta y)^2} \\
& + \frac{\kappa_{i,j,k}\left(T_{i,j,k+1/2} - T_{i,j,k-1/2}\right) - \kappa_{i,j,k-1}\left(T_{i,j,k-1/2} - T_{i,j,k-3/2}\right)}{(\Delta z)^2} \Bigg).
\end{aligned}
\tag{3.72}
$$

3.3 VSIAM3 for Inviscid Compressible Flows

Here the application of VSIAM3 to compressible flows is explained.

3.3.1 Governing Equations

The Euler equations describe the dynamics of inviscid compressible flows and are written as

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0, \tag{3.73}$$

where

$$U = \left\{ \begin{array}{c} \rho \\ m \\ E \end{array} \right\}, \tag{3.74}$$

$$F(U) = \left\{ \begin{array}{c} m \\ um + p \\ Eu + pu \end{array} \right\}, \tag{3.75}$$

where m is the momentum per unit volume ($m = \rho u$) and E the total energy per unit volume. The equations are completed by the equation of state

$$p = \left(E - \frac{\rho u^2}{2} \right) (\gamma - 1), \tag{3.76}$$

where γ is the specific heat ratio. By using the VSIAM3 formulation [183], (3.75) is split into two parts, advection part and non-advection part as follows

$$F(U) = F^{I}(U) + F^{II}(U) = \left\{ \begin{array}{c} m \\ um \\ Eu \end{array} \right\} + \left\{ \begin{array}{c} 0 \\ p \\ pu \end{array} \right\}. \tag{3.77}$$

A fractional step approach is used to solve (3.73), in which the advection part

$$\frac{\partial U}{\partial t} + \frac{\partial F^{I}(U)}{\partial x} = 0, \tag{3.78}$$

is solved by CIP-CSL3 method [190]. The non-advection part

$$\frac{\partial U}{\partial t} + \frac{\partial F^{II}(U)}{\partial x} = 0, \tag{3.79}$$

is solved by finite volume/difference formulations.

3.3.2 Advection Part: CIP-CSL3

The CIP-CSL3 method is an extension of the CIP-CSL2 method. In this method, third-order polynomial function is employed as the interpolation function instead of the quadratic function of CIP-CSL2. Then CIP-CSL3 needs one more constraint to determine all coefficients of the interpolation function. Hence, a control parameter (gradient at the cell center) is introduced in CIP-CSL3 as the additional constraint. The control parameter can be used as slope limiter to eliminate numerical oscillation [183]. The third-order polynomial interpolation function between $x_{i-1/2}$ and $x_{i+1/2}$ is written as

$$\Phi_i(x) = a_i(x - x_{i-1/2})^3 + b_i(x - x_{i-1/2})^2 + c_i(x - x_{i-1/2}) + \phi_{i-1/2}. \tag{3.80}$$

In addition to the constraints (3.20) and (3.21), the following constraint

$$\frac{d\Phi_i(x_i)}{dx} = d_i, \tag{3.81}$$

is used to determine the coefficients of equation (3.80) as follows

$$a_i = \frac{4\left(\phi_{i+1/2} - \phi_{i-1/2} - \Delta x d_i\right)}{\Delta x^3}, \tag{3.82}$$

$$b_i = \frac{3\left(-2\phi_i - \phi_{i+1/2} + 3\phi_{i-1/2} + 2\Delta x d_i\right)}{\Delta x^2}, \tag{3.83}$$

$$c_i = \frac{2\left(3\phi_i - 3\phi_{i-1/2} - \Delta x d_i\right)}{\Delta x}. \tag{3.84}$$

The derivative d_i is given as

$$d_i = \beta_i \tilde{d}_i, \tag{3.85}$$

$$\tilde{d}_i = \text{minmod}\left(\frac{S_{i+1/2} + S_{i-1/2}}{2}, 2S_{i+1/2}, 2S_{i-1/2}\right), \tag{3.86}$$

here

$$\text{minmod}(A,B,C) = \begin{cases} \text{m}(A,B,C) & \text{if} \qquad \text{sgn}(A) = \text{sgn}(B) = \text{sgn}(C) \\ 0 & \text{otherwise,} \end{cases} \tag{3.87}$$

$$m(A,B,C) = \begin{cases} A & \text{if} \quad \min(|A|,|B|,|C|) = |A| \\ B & \text{else if} \quad \min(|A|,|B|,|C|) = |B| \\ C & \text{else} \quad \min(|A|,|B|,|C|) = |C|, \end{cases} \tag{3.88}$$

and

$$S_{i-1/2} = \frac{\hat{\phi}_i - \hat{\phi}_{i-1}}{\Delta x}, \tag{3.89}$$

where

$$\hat{\phi}_i = \frac{3}{2}\phi_i - \frac{1}{4}(\phi_{i+1/2} + \phi_{i-1/2}), \tag{3.90}$$

and

$$\beta_i = \begin{cases} 0.0125 & \text{if} \qquad (u^n_{i-1/2} - u^n_{i+1/2}) < 0.02\Delta x \\ 1.2 & \text{otherwise.} \end{cases} \tag{3.91}$$

Given at time step n, the cell averages, ρ^n_i, u^n_i, p^n_i, m^n_i, E^n_i and the cell boundary values $\rho^n_{i-1/2}$, $u^n_{i-1/2}$, $m^n_{i-1/2}$, $E^n_{i-1/2}$, the CIP-CSL3 method is used to obtain the corresponding density at the next time step n+1 (i.e. ρ^{n+1}_i and $\rho^{n+1}_{i-1/2}$) and the provisional values of the momentum and energy (i.e. m^*_i, $m^*_{i-1/2}$, E^*_i and $E^*_{i-1/2}$).

3.3.3 The Non-Advection Part

A simple explicit equation [183] is used to advance the pressure

$$p_i^{n+1} = C_i^2 \Delta t \left(u_i^* \frac{\rho_{i+1/2}^{n+1} - \rho_{i-1/2}^{n+1}}{\Delta x} + \frac{\rho_i^{n+1}}{\gamma \Delta t} - \frac{m_{i+1/2}^* - m_{i-1/2}^*}{\Delta x} \right), \tag{3.92}$$

here $u_i^* = \frac{m_i^*}{\rho_i^{n+1}}$, and, $C_i^2 = \frac{\gamma p_i^*}{\rho_i^{n+1}}$. The boundary values of the momentum and total energy are updated as follows

$$m_{i-1/2}^{n+1} = m_{i-1/2}^n - \frac{\Delta t}{\Delta x} \left(p_i^{n+1} - p_{i-1}^{n+1} \right), \tag{3.93}$$

$$E_{i-1/2}^{n+1} = E_{i-1/2}^n - \frac{\Delta t}{\Delta x} \left(u_i^{n+1} p_i^{n+1} - u_{i-1}^{n+1} p_{i-1}^{n+1} \right). \tag{3.94}$$

The cell averages of the momentum and total energy can be obtained via TEC formula as follows

$$m_i^{n+1} = m_i^n + \frac{1}{2} \left(m_{i+1/2}^{n+1} - m_{i+1/2}^n + m_{i-1/2}^{n+1} - m_{i-1/2}^n \right), \tag{3.95}$$

$$E_i^{n+1} = E_i^n + \frac{1}{2} \left(E_{i+1/2}^{n+1} - E_{i+1/2}^n + E_{i-1/2}^{n+1} - E_{i-1/2}^n \right). \tag{3.96}$$

For numerical simulations of compressible flows, CIP-CSL3 should be used. Although CIP-CSLR is also a less oscillatory CIP-CSL formulation, CIP-CSLR does not include a slope limiter so that CIP-CSLR cannot prevent oscillation around shock in VSIAM3 (see Appendix for a numerical results by CIP-CSLR).

3.4 Numerical Methods for Free Surface Flows

In this work, we employ the CLSVOF scheme [158, 200] with the THINC/WLIC scheme [202] as the VOF type scheme for the interface advection and reconstruction. The CLSVOF interface capturing scheme has been widely used in a variety of applications including free surface flow simulations with surface tension force [202, 206, 203, 204]. In the following sections, the CLSVOF scheme [200, 202, 206, 203, 204] is reviewed.

3.4.1 Interface Capturing Using Coupled Level Set and THINC/WLIC

In the CLSVOF scheme used in our simulations, the interface is tracked by a VOF function which is updated by the THINC/WLIC [203, 204, 205] scheme. The level set function is built by using the interface indicated by the VOF function.

3.4.2 The THINC/WLIC Scheme

The THINC/WLIC [203, 204, 205] scheme is a kind of VOF scheme. The VOF function is evolved according to the following advection equation

$$\frac{\partial \xi}{\partial t} + \nabla \cdot (\mathbf{u}\xi) - \xi \nabla \cdot \mathbf{u} = \mathbf{0}. \tag{3.97}$$

here $\mathbf{u}$ is the velocity, ξ is the characteristic function which has the value of 0 or 1, and the cell average of ξ is the VOF function $\vartheta_{i,j}$ which has a value of $0 \leqslant \vartheta_{i,j} \leqslant 1$. In a two dimensional case

$$\vartheta_{i,j} = \frac{1}{\Delta x \Delta y} \iint_{\Omega_{i,j}} \xi dx dy, \tag{3.98}$$

given $\vartheta^n_{i,j}$ at the nth time step, $\vartheta_{i,j}$ is updated by a flux form formulation using a dimensional splitting approach as following

$$\vartheta^*_{i,j} = \vartheta^n_{i,j} - \frac{F^n_{x,i+1/2,j} - F^n_{x,i-1/2,j}}{\Delta x} + \vartheta^n_{i,j} \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \Delta t, \tag{3.99}$$

$$\vartheta^{n+1}_{i,j} = \vartheta^*_{i,j} - \frac{F^*_{y,i,j+1/2} - F^*_{y,i,j-1/2}}{\Delta y} + \vartheta^n_{i,j} \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \Delta t, \tag{3.100}$$

with

$$F_{x,i+1/2,j} = - \int_{y_{i,j-1/2}}^{y_{i,j+1/2}} \int_{x_{i+1/2,j}}^{x_{i+1/2,j} - u_{i+1/2,j} \Delta t} \xi_{is,j}(x, y) dx dy, \tag{3.101}$$

$$F_{y,i,j+1/2} = -\int_{y_{i,j+1/2}}^{y_{i,j+1/2}-v_{i,j+1/2}\Delta t} \int_{x_{i-1/2,j}}^{x_{i+1/2,j}} \xi_{i,js}(x,y)dxdy, \tag{3.102}$$

where $F_{x,i+1/2,j}$ and $F_{y,i,j+1/2}$ are the advection fluxes for x direction and y direction, respectively. is and js are given as follows

$$is = \begin{cases} i & \text{if} \quad (u_{x,i+1/2,j}) \geqslant 0, \\ i+1 & \text{if} \quad (u_{x,i+1/2,j}) < 0, \end{cases} \tag{3.103}$$

$$js = \begin{cases} j & \text{if} \quad (u_{y,i,j+1/2}) \geqslant 0, \\ j+1 & \text{if} \quad (u_{y,i,j+1/2}) < 0. \end{cases} \tag{3.104}$$

The flux is calculated based on a one dimensional THINC scheme. A piece-wise modified hyperbolic tangent function is used as a characteristic function of the THINC scheme

$$\xi_{x,i} = \frac{1}{2}\left(1 + \alpha_x \textbf{tanh}\left(\beta\left(\frac{x - x_{i-1/2}}{\Delta x} - \tilde{x}_i\right)\right)\right). \tag{3.105}$$

where the parameters α_x and β are important in determining the quality of the numerical solution. α_x is determined by

$$\alpha_x = \begin{cases} 1 & \text{if} \quad (n_{x,i}) \geqslant 0, \\ -1 & \text{if} \quad (n_{x,i}) < 0, \end{cases} \tag{3.106}$$

where $n_{x,i} = \vartheta_{i+1} - \vartheta_{i-1}$ is used for determining (3.106) and $\beta = 3.5$. $\tilde{x}_i\Delta x$ corresponds to the distance between $x_{i-1/2}$ and the interface.Using the cell average of ξ_i, $\tilde{x}_i$ can be calculated as follows

$$\vartheta_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \xi_i(x, \tilde{x}_i)dx. \tag{3.107}$$

The WLIC is a simple method for the interface reconstruction which takes into account the information of the surface normal more effectively than the SLIC method and maintain simpler implementation than the PLIC method. The WLIC method employs the information of the interface along the x-coordinate as well as the interface along the y-coordinate to reconstruct the interface using weights calculated from the surface normal $\mathbf{n}$:

$$\xi_{i,j}(x,y) = \omega_{x,i,j}(\mathbf{n_{i,j}})\xi_{\mathbf{x,i,j}}(\mathbf{x},\mathbf{y}) + \omega_{\mathbf{y,i,j}}(\mathbf{n_{i,j}})\xi_{\mathbf{y,i,j}}(\mathbf{x},\mathbf{y}), \tag{3.108}$$

where ω_x and ω_y are the weights, ξ_x and ξ_y are the characteristic functions of the interface along the x-coordinate and the interface along the y-coordinate, respectively. The weights ω_x and ω_y and the characteristic functions ξ_x and ξ_y must satisfy the following constraints

$$\omega_{x,i,j} + \omega_{y,i,j} = 1, \tag{3.109}$$

and

$$\vartheta_{i,j} = \frac{1}{\Delta x \Delta y}\iint_{\Omega_{i,j}} \xi_{x,i,j}dxdy = \frac{1}{\Delta x \Delta y}\iint_{\Omega_{i,j}} \xi_{y,i,j}dxdy, \tag{3.110}$$

using the following simple weights

$$\omega_{x,i,j} = \frac{|n_{x,i,j}|}{|n_{x,i,j}| + |n_{y,i,j}|}, \tag{3.111}$$

$$\omega_{y,i,j} = \frac{|n_{y,i,j}|}{|n_{x,i,j}| + |n_{y,i,j}|}, \tag{3.112}$$

where n_x and n_y are the Cartesian components of the surface normal $\mathbf{n}$, respectively. These x-component and the y-component of the surface normal are computed using 3×3 grid stencil as follows

$$n_{x,i,j} = \frac{1}{4}(n_{x,i+1/2,j+1/2} + n_{x,i-1/2,j+1/2} + n_{x,i+1/2,j-1/2} + n_{x,i-1/2,j-1/2}) \tag{3.113}$$

$$n_{y,i,j} = \frac{1}{4}(n_{y,i+1/2,j+1/2} + n_{y,i-1/2,j+1/2} + n_{y,i+1/2,j-1/2} + n_{y,i-1/2,j-1/2}) \tag{3.114}$$

where

$$n_{x,i+1/2,j+1/2} = \frac{1}{2\Delta x}(\vartheta_{i+1,j} - \vartheta_{i,j} + \vartheta_{i+1,j+1} - \vartheta_{i,j+1}), \tag{3.115}$$

$$n_{y,i+1/2,j+1/2} = \frac{1}{2\Delta y}(\vartheta_{i,j+1} - \vartheta_{i,j} + \vartheta_{i+1,j+1} - \vartheta_{i+1,j}), \tag{3.116}$$

In a similar manner, a three-dimensional implementation is straightforward by using the dimensional splitting approach.

3.4.3 The Level Set Scheme (CLSVOF)

The level set scheme [159] is an interface capturing scheme, in which the interface is implicitly represented by the level set function $\psi = 0$ (zero level set). The function is also called the signed distance function that corresponds to the shortest distance from points on the grid to the interface. The level set scheme features simplicity particularly in calculating the curvature of the interface.

The CLSVOF procedure [200] includes calculating the position of the 0.5-contour of the VOF function ϑ (zero level set) by a linear interpolation between cells. Then the signed distance function ψ within Δh from the interface, where Δh is the grid resolution, is calculated by the fast marching procedure [147, 3], solving the Eikonal equation:

$$|\nabla\psi| = 1. \tag{3.117}$$

Other ψ values further from the interface is computed by an iterative re-initialization procedure [159] by solving the following equation to a steady state:

$$\frac{\partial\psi}{\partial\tau_0} = F(\psi)\left(1 - |\nabla\psi|\right), \tag{3.118}$$

where τ_0 is an artificial time and $F(\psi)$

$$F(\psi) = \begin{cases} 0 & \text{for grid cells updated by the fast marching} \\ \frac{\psi}{\sqrt{\psi^2+\epsilon^2}} & \text{else,} \end{cases} \tag{3.119}$$

while ψ calculated by the fast marching procedure is fixed. $\epsilon = \Delta h$ is used in equation (3.119). The the number of iterations of the solution of equation (3.118) equals $\frac{\alpha}{\Delta\tau_0}$ [204]. Where 2α is the interface thickness. $\alpha = 1.5\Delta h$ and $\Delta\tau 0 = 0.1\Delta h$ were used in the simulations carried out in this thesis. A smoothed Heaviside function is used as the color function ϕ which is used to define the physical properties of the light fluid and the heavy fluid like the density and viscosity

$$\phi = H_\alpha(\psi), \tag{3.120}$$

$$H_\alpha(\psi) = \begin{cases} 0 & \text{if} \quad \psi < -\alpha, \\ \frac{1}{2}\left(1 + \frac{\psi}{\alpha} + \frac{1}{\pi} sin\left(\frac{\pi\psi}{\alpha}\right)\right) & \text{if} \quad |\psi| \leqslant \alpha, \\ 1 & \text{if} \quad \psi > \alpha. \end{cases} \tag{3.121}$$

Equation 3.121 sets the color function $\phi = 1$ for the heavy fluid and $\phi = 0$ for the light fluid. The physical properties such as the density ρ and viscosity μ are calculated as follows

$$\rho = \rho_h\phi + \rho_l(1-\phi), \tag{3.122}$$

$$\mu = \mu_h\phi + \mu_l(1-\phi), \tag{3.123}$$

where ρ_h and μ_h are the density and the viscosity of the heavy fluid, and ρ_l and μ_l are the density and the viscosity of the light fluid.

3.5 Model of Surface Tension Force

The continuum surface force (CSF) model [16] is well known for computing surface tension force in various numerical schemes of free surface flows such as VOF schemes [74, 145] and level set schemes [160, 159, 138]. In the (CSF) procedure [16], the surface tension effect on a fluid motion is modeled as a body force associated by a smooth delta function δ_α and the resulting force corresponds to the product of the interface gradient and curvature. The surface tension effects are represented in the simulation model by an external force $\mathbf{F_{sf}}$ in the momentum equation.

$$\mathbf{F_{sf}} = \sigma\kappa\delta_\alpha\mathbf{n_s}, \tag{3.124}$$

Where σ is the surface tension coefficient, κ is the local mean curvature and $\mathbf{n_s}$ is the unit vector normal to the interface.

In the level set formulation [159], the standard CSF model is formulated using the level set function ψ

$$\mathbf{F_{sf}} = \sigma\kappa\delta_\alpha(\psi)\mathbf{n_{ls}}, \tag{3.125}$$

where the interface normal $\mathbf{n_{ls}}$ is calculated by the gradient of the level set function as

$$\mathbf{n_{ls}} = \frac{\nabla\psi}{|\nabla\psi|}, \tag{3.126}$$

the surface curvature κ is approximated by

$$\kappa = \nabla \cdot \frac{\nabla\psi}{|\nabla\psi|}, \tag{3.127}$$

and the smoothed delta function $\delta_\alpha(\psi)$ can be estimated using the level set function as

$$\delta_\alpha(\psi) = \begin{cases} \frac{1}{2\alpha}\left(1 + cos\left(\frac{\pi\psi}{\alpha}\right)\right) & \text{if} |\psi| < \alpha \\ 0 & \text{else,} \end{cases} \tag{3.128}$$

Noting that the corresponding smoothed Heaviside function of equation (3.128) which is used in the CLSVOF scheme is given in equation (3.121) and that (2α) is the interface thickness.

In this work, a modified level set based standard CSF model is employed within the CLSVOF formulation, namely the level set based density-scaled balanced CSF model [204, 205], in which the following balanced force formulation

$$\mathbf{F_{sf}} = \sigma\kappa\nabla\phi^{\mathbf{scaling}}, \tag{3.129}$$

is used. Equation (3.129) is mathematically equivalent to equation (3.125), yet a different discretization method is used. In the balanced force formulation, it is vital to use the same discretization for $\nabla\phi^{scaling}$ in 3.129 and the pressure gradient term in the projection step. For instance, since the x-component of the acceleration due to the pressure gradient is discretized as

$$\left(\frac{\partial u}{\partial t}\right)_{i-1/2,j,k} = -\frac{1}{\rho_{i-1/2,j,k}} \frac{P^{n+1}_{i,j,k} - P^{n+1}_{i-1,j,k}}{\Delta x}, \tag{3.130}$$

hence, the x-component of the acceleration due to equation (3.129) must be discretized in a similar manner as follows

$$\left(\frac{F_{sf}}{\rho}\right)_{i-1/2,j,k} = \frac{\sigma\kappa_{i-1/2,j,k}}{\rho_{i-1/2,j,k}} \frac{\phi^{scaling}_{i,j,k} - \phi^{scaling}_{i-1,j,k}}{\Delta x}, \tag{3.131}$$

here $\rho_{i-1/2,j,k} = \frac{1}{2}(\rho_{i,j,k} + \rho_{i-1,j,k})$, and $\phi^{scaling}_{i,j,k} = H^{scaling}_\alpha(\psi)$, where $H^{scaling}_\alpha(\psi)$ is an asymmetrical smoothed Heaviside function utilised in this formulation

$$H_{\alpha}^{scaling}(\psi) = \begin{cases} 0 & \text{if} \quad \psi < -\alpha, \\ \frac{1}{2}\left[\frac{1}{2} + \frac{\psi}{\alpha} + \frac{\psi^2}{2\alpha^2} - \frac{1}{4\pi^2}\left(cos\left(\frac{2\pi\psi}{\alpha}\right) - 1\right) + \frac{\alpha+\psi}{\alpha\pi} sin\left(\frac{\pi\psi}{\alpha}\right)\right] & \text{if} \quad |\psi| \leqslant \alpha, \\ 1 & \text{if} \quad \psi > \alpha. \end{cases} \tag{3.132}$$

The level set based density-scaled balanced CSF formulation reduces spurious currents and improves the stability of the CSF procedure [205], particularly in the case of two-phase flows of high-density ratio.

3.6 Summary

A detailed review and explanation of the VSIAM3 numerical framework for solving the Navier-Stokes equations have been presented in this chapter. VSIAM3 for incompressible and compressible fluid flow and heat transfer has first been introduced. A comprehensive explanation of the implementation of CIP-CSL schemes (CIP-CSL2, CIP-CSLR, and CIP-CSL3) was also given. A description of the execution of various terms of the Navier-Stokes equations and the energy equation in VSIAM3 has been accounted for including the projection step and the solution of pressure Poisson equation on a multi-moment M-grid.

The numerical model is extended based on the one-fluid formulation for gas-liquid two-phase flows. A review of the coupling of the Level-Set method with the THINC/WLIC methods has also been presented, where the implementation of the combination of CLSVOF advection method with the equations of incompressible gas-liquid two-phase flows is explained. The THINC/WLIC surface capturing scheme has first been reviewed. Followed by a presentation of the level set method. Finally, the model of the surface tension used in the present work is given.

In conclusion, It is worth noting that a detailed construction of the VSIAM3 numerical framework for gas-liquid two-phase flows and a comprehensive explanation of the implementation complexities which arise from the multi-moment concept has not been given in a published work.

Chapter 4

Efficient Implementation of Multi-Moment Method

4.1 Introduction

Despite that VSIAM3 has been presented as a robust and computationally efficient numerical scheme [203, 204, 205], most of the analysts who attempted to build up the code could not complete robust fluid simulations [124]. This is on account of a multi-moment structure of VSIAM3 (including the CIP-CSL method). The multi-moment structure has risen complications in the execution of VSIAM3. The issue on the robustness in VSIAM3 has likewise been suggested in [102] and a conceivable arrangement utilizing the simple CIP interpolation for the problem has been proposed. The CIP interpolation [196] requires incorporating and updating derivatives of computational variables in the numerical computations. In this chapter, we study discretization strategies of the conservative advection equation and their effect on the robustness of VSIAM3.

As we mentioned in the details of the CIPCSL scheme in (section 3.2.5), the boundary value $\phi_{i-1/2}$ is updated by the conservation equation of differential form in the VSIAM3 framework

$$\frac{\partial \phi}{\partial t} + \frac{\partial u\phi}{\partial x} = 0. \tag{4.1}$$

(4.1) is equivalent to

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = -\phi\frac{\partial u}{\partial x}. \tag{4.2}$$

Using a splitting approach to solve (4.2) gives

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0, \tag{4.3}$$

and

$$\frac{\partial \phi}{\partial t} = -\phi\frac{\partial u}{\partial x}. \tag{4.4}$$

Equation (4.3) indicates the advection of ϕ therefore, is solved as an interpolation problem. Equation (4.4) represents a correction of the convected scalar due to the divergence term of the velocity, i.e., amounts to the change of ϕ due to the compression or expansion and is solved by a finite difference formulation.

As mentioned in section 3.2.5.1, the divergence term of the velocity is one of the main topics in this work. Despite the importance of the divergence term discretization on the robustness of VSIAM3 method as shown in this chapter, it has rarely been mentioned in the published works of VSIAM3 and CIP-CSL advection schemes. Additionally, although VSIAM3 is a multi-moment method, no work has been carried out to examine the divergence term discretization by using various moments and stencils. In this chapter, we carry out a study of the discretization strategies of the conservation equation in VSIAM3. An investigation into these strategies is conducted through the lid-driven cavity flow, shock tube problems, two-dimensional explosion test, and droplet splashing problem. We found that the use of the (CIP-CSLR) method (section 3.2.5.3) as the conservation equation solver is critically essential for the robustness of incompressible flow simulations using VSIAM3 and that the use of (CIP-CSL2) in these simulations affects the robustness of VSIAm3 due to the resulted numerical oscillations. It is also found that numerical results are sensitive to discretization formulations of the velocity divergence term (3.26) in the conservation equation and various approximations have been proposed.

In the following section 4.2, different formulations of the divergence term are presented. Then a study of the divergence term formulations in Fourier Analysis is given in section 4.2.1. In section 4.3, the numerical results are presented. Followed by a summary in section 4.4.

4.2 Formulations of the Divergence Term

In this section, we explain possible discretizations of the velocity divergence term (4.4) of the one-dimensional conservation equation.

Simple Central Difference based on Boundary Value (CDb)

$$\phi\frac{\partial u}{\partial x} = \phi^*_{i-1/2}\frac{u^n_{i+1/2} - u^n_{i-3/2}}{2\Delta x}. \tag{4.5}$$

In the first works of the CIP-CSL methods [195, 190], the basic central difference approximation using boundary values has been recommended.

Up-winding approximation with a time average (Upw-Avg1)

$$\phi\frac{\partial u}{\partial x} = \begin{cases} (\phi^*_{i-1/2} + \phi^n_{i-1/2})(\frac{u^*_{i-1/2} - u^*_{i-3/2} + u^n_{i-1/2} - u^n_{i-3/2}}{4\Delta x}) & \text{if} \quad u_{i-1/2} > 0 \\ (\phi^*_{i-1/2} + \phi^n_{i-1/2})(\frac{u^*_{i+1/2} - u^*_{i-1/2} + u^n_{i+1/2} - u^n_{i-1/2}}{4\Delta x}) & \text{if} \quad u_{i-1/2} \le 0, \end{cases} \tag{4.6}$$

This approximation has been suggested in [188, 182] for incompressible flow situations, where u^* is the provisional value of the velocity right after the semi-Lagrangian solution in fluid computation.

Upwind approximation with a time average for compressible flow (UPW-Avg2)

$$\phi\frac{\partial u}{\partial x} = \begin{cases} \phi^*_{i-1/2}(\frac{u^*_{i-1/2} - u^*_{i-3/2} + u^n_{i-1/2} - u^n_{i-3/2}}{2\Delta x}) & \text{if} \quad u_{i-1/2} > 0 \\ \phi^*_{i-1/2}(\frac{u^*_{i+1/2} - u^*_{i-1/2} + u^n_{i+1/2} - u^n_{i-1/2}}{2\Delta x}) & \text{if} \quad u_{i-1/2} \le 0. \end{cases} \tag{4.7}$$

This approximation has been suggested in [183] for inviscid compressible flow.

It could be noticed that the first approximation uses relatively large stencil while the last two estimates are not simple regarding implementation and they require storage for the provisional value of the velocity.

In this chapter, we propose the following approximations of the velocity divergence term.

Upwind based on Boundary Value (UPW)

$$\phi\frac{\partial u}{\partial x} = \begin{cases} \phi^*_{i-1/2}(\frac{u^n_{i-1/2}-u^n_{i-3/2}}{\Delta x}) & \text{if} \quad u_{i-1/2} > 0 \\ \phi^*_{i-1/2}(\frac{u^n_{i+1/2}-u^n_{i-1/2}}{\Delta x}) & \text{if} \quad u_{i-1/2} \leq 0. \end{cases} \tag{4.8}$$

This is a simple upwind approximation based on the boundary values.

Central Difference based on Cell Integrated Average (CDca)

$$\phi\frac{\partial u}{\partial x} = \phi^*_{i-1/2}\frac{u^n_i - u^n_{i-1}}{\Delta x}. \tag{4.9}$$

This is a central difference approximation based on the cell integrated averages. In comparison to CDb (4.5), the proposed central difference formulation uses the cell integrated average in place of the boundary values which results in a shorter stencil than that of CDb.

Central Difference based Cell Centre Value (CDcc)

$$\phi\frac{\partial u}{\partial x} = \phi^*_{i-1/2}\frac{\hat{u}^n_i - \hat{u}^n_{i-1}}{\Delta x} \tag{4.10}$$

This is another central difference approximation based on the cell center values ($\hat{u}_i$), where $\hat{u}_i$ is the velocity calculated at cell centre [190]. By using the quadratic function (3.19), $\hat{u}_i$ can be obtained

$$\hat{u}_i = \frac{3}{2}u_i - \frac{1}{4}\left(u_{i+1/2} + u_{i-1/2}\right). \tag{4.11}$$

Central Difference based on a 4th-Order Polynomial Function (CDbca)

$$\phi\frac{\partial u}{\partial x} = \phi^*_{i-1/2}\left(2\frac{u^n_i - u^n_{i-1}}{\Delta x} - \frac{u^n_{i+1/2} - u^n_{i-3/2}}{2\Delta x}\right). \tag{4.12}$$

This formula (4.12) can be derived from a fourth-order central interpolation function using $u_{i-3/2}$, u_{i-1}, $u_{i-1/2}$, u_i and $u_{i+1/2}$ [125].

Mixed Formulation of the Upwind and a Central Difference (UPW-CDcc)

$$\phi\frac{\partial u}{\partial x} = \begin{cases} \mathsf{D}_{UPW} & \text{if} & \mathsf{D}_{UPW} \cdot \mathsf{D}_{CDcc} < 0 \\ \mathsf{D}_{UPW} & \text{else if} & |\mathsf{D}_{UPW}| < |\mathsf{D}_{CDcc}| \\ \mathsf{D}_{CDcc} & \text{else,} & \end{cases} \tag{4.13}$$

Here D$_{UPW}$ and D$_{CDcc}$ represent $\phi\dfrac{\partial u}{\partial x}$ and are calculated by (4.8) and (4.10), respectively. The mixed formulation is introduced to take advantages of both upwind and central difference approximations. The formulation employs the upwind formula (4.8) when the sign of derivatives of UPW and CDcc are different (D$_{UPW}$ $\cdot$ D$_{CDcc}$ < 0) or |D$_{CDcc}$| is larger than that of |D$_{UPW}$|. Otherwise the central difference formula (4.10) is used. Although we combined UPW with CDcc , it can be combined with any other central difference formulations. In this work, we also combined UPW with CDbca (UPW-CDbca).

Interpolation at Characteristic Departure Point (DP)

$$\phi\frac{\partial u}{\partial x} = \begin{cases} \phi^{*}_{i-1/2}\dfrac{\partial \Phi_{i-1}}{\partial x}(x_{i-1/2} - u_{i-1/2}\Delta t) & \text{if} \quad u_{i-1/2} \geq 0 \\ \phi^{*}_{i-1/2}\dfrac{\partial \Phi_{i}}{\partial x}(x_{i-1/2} - u_{i-1/2}\Delta t) & \text{if} \quad u_{i-1/2} < 0. \end{cases} \tag{4.14}$$

This formulation evaluates the divergence at the characteristic departure point using a CIP-CSL interpolation function.

4.2.1 Formulations of the Divergence Term in Fourier Analysis

In this section, we carry out Fourier analysis of the introduced divergence term formulations. Fourier analysis presents the resolution of spatial derivatives in the wavenumber domain. The spatial profile of the velocity $U(x)$ is defined over the domain [0, L] with a uniform grid spacing Δx is decomposed into Fourier series

$$U(x) = \sum_{\kappa} U(\kappa) e^{j\omega x/\Delta x}, \tag{4.15}$$

where $j = \sqrt{-1}$ and $\omega = 2\pi\kappa\Delta x/L$ is the scaled wavenumber. The point value at $x_{i-1/2}$ is also decomposed as

$$u_{i-1/2} = \sum_{\kappa} U(\kappa) e^{j\omega x_{i-1/2}/\Delta x}. \tag{4.16}$$

Using (4.16), the point value at $x_{i-1/2+m}$ is decomposed as

$$u_{i-1/2+m} = u_{i-1/2} e^{j\omega m}. \tag{4.17}$$

The cell integrated average u_i is also decomposed as

$$u_i = \frac{1}{\Delta x}\int_0^{\Delta x} U(x_{i-1/2}+x)dx = u_{i-1/2}\frac{e^{j\omega}-1}{j\omega} \tag{4.18}$$

Since equation (4.18) indicates the relation between the point value and the cell integrated average, the accuracy of the suggested approximations of the velocity divergence term can be analysed by using (4.17) and (4.18). The formulations of the velocity divergence term in the wavenumber domain can be acquired as follows

$$U_{x,CDb}(\omega) = j(\sin(\omega)), \tag{4.19}$$

$$U_{x,UPW}(\omega) = (\cos(\omega)-1)+j(\sin(\omega)), \tag{4.20}$$

$$U_{x,CDca}(\omega) = j(\sin(\omega)), \tag{4.21}$$

$$U_{x,CDcc}(\omega) = j(\frac{6\sin^2(\omega/2)}{\omega}-\frac{\sin(\omega)}{2}), \tag{4.22}$$

$$U_{x,CDbca}(\omega) = j(-\sin(\omega)+\frac{8\sin^2(\omega/2)}{\omega}). \tag{4.23}$$

$U_{x,DP}$ depends on the interpolation function of a CIP-CSL method. However, in this work, CIP-CSLR and CIP-CSL3 are mainly used and both CIP-CSL methods are employing nonlinear interpolation function. Therefore we cannot analyse DP formulation.

Figure 4.1 depicts the resolution of various formulations of the velocity divergence term in Fourier domain. All central difference methods have no error in real part (no diffusion error) and only UPW has diffusion error, as shown in fig. 4.1 (b). Figure 4.1 (a) has shown that CDbca is the closest to the exact solution. CDcc is second closest, CDca third, CDca fourth, and CDb and UPW are the most inaccurate.

4.3 Numerical Results

4.3.1 Lid-Driven Cavity Flow

The lid-driven incompressible flow in a square cavity has been used as a model problem for assessing numerical methods [56]. This test problem was used to investigate

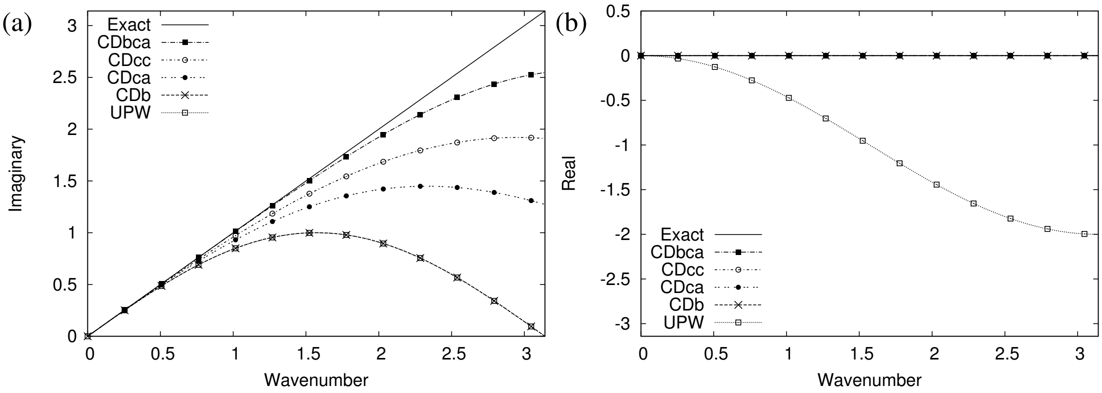


Figure 4.1: The formulations of the divergence term in Fourier analysis, (a) imaginary part and (b) real part.

the discretisation strategies of the conservation equation in multi-moment method. The tests were conducted at Reynolds number Re$= 1000$ and Re$= 5000$. The numerical results were compared with the solution by Ghia et. al [56]. Ghia et. al employed the vorticity-stream function formulation of the incompressible Navier-Stokes equations to solve the shear-driven flow in a square cavity on a uniform fine grid. They employed multigrid method in the solution of the Navier-Stokes equations to enhance the solution convergence. Furthermore, second-order accurate central finite-difference approximations were used for second-order derivatives in the stream function and vorticity equations. First-order accurate upwind difference formulation was used for the convective terms. The later formulation results in second-order accuracy at convergence through a correction scheme. Additionally, they used a grid of (129×129) and (257×257) for Re$= 1000$ and Re$= 5000$, respectively.

The numerical result of Re$= 1000$ using the CIP-CSL2 scheme with the upwind formulation (UPW) is shown in fig.4.2a. The figure depicts the horizontal velocity component along the vertical line through the centre of the cavity. The result shows reasonable agreement with the solution by Ghia [56], however, the calculation was not stable after attaining the depicted solution. Although we also examined CIP-CSL2 scheme using central difference approximations, these were not stable and did not attain the steady-state solution. Figure 4.2b shows the numerical solution when the velocity divergence term was set to zero. In this case although the result was inaccurate, the computations were free of numerical oscillations. These results suggest that the use of the CIP-CSL2 scheme in VSIAM3 deteriorates the robustness and the velocity divergence term is pertinent to the robustness of the numerical method.

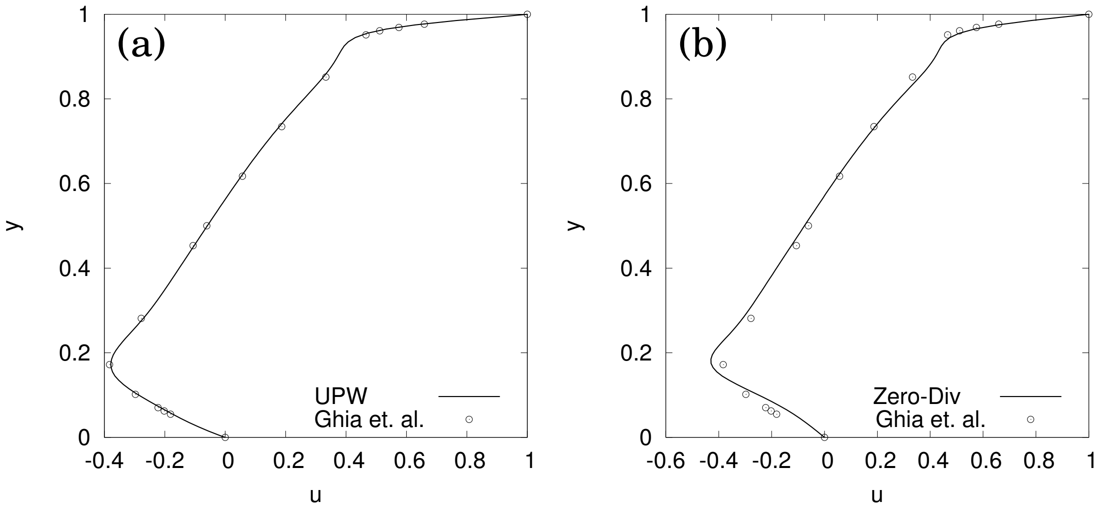


Figure 4.2: Numerical results of lid-driven cavity flow problem. Re = 1000. CIP-CSL2 with UPW was used for (a). (b) is the result by CIP-CSL2 when the divergence term was ignored. The line and dot represent the numerical result and the solution by Ghia [56], respectively. A Cartesian grid of 100×100 **was used.**

We also inspected the use of the CIP-CSLR scheme in VSIAM3. The numerical results of the CIP-CSLR scheme with various divergence term formulations are shown in figures 4.4-4.10 with grid refinement studies where three grid resolutions (50×50, 100×100 and 200×200) were used. Presented in these figures are the horizontal velocity component along the vertical line through the centre of the cavity and the vertical velocity component along the horizontal line through the centre of the cavity in parts a and b, respectively, of each figure. The numerical results indicate reasonable agreements with the Ghia solution [56] and reasonable convergences as well. All the numerical computations using CIP-CSLR scheme (with any velocity divergence term formulation) were oscillation-free in this test problem. These results by CIP-CSL2 and CIP-CSLR suggest that the use of CIP-CSLR scheme enhances the robustness of VSIAM3. As detailed in section 3.2.5.3 and [192], CIP-CSLR is a less oscillatory formulation and CIP-CSL2 is not free of numerical oscillations. Therefore it can be considered that the numerical oscillations generated by CIP-CSL2 affect the robustness of VSIAM3 through the velocity divergence term and that the use of a less oscillatory advection scheme is critically essential for the robust execution of VSIAM3.

The numerical result by UPW are shown in fig. 4.3. The result is almost equivalent to that of the upwind with a time average approximation given in [188]. Additionally, implementing the proposed UPW approximation is simpler than the latter.

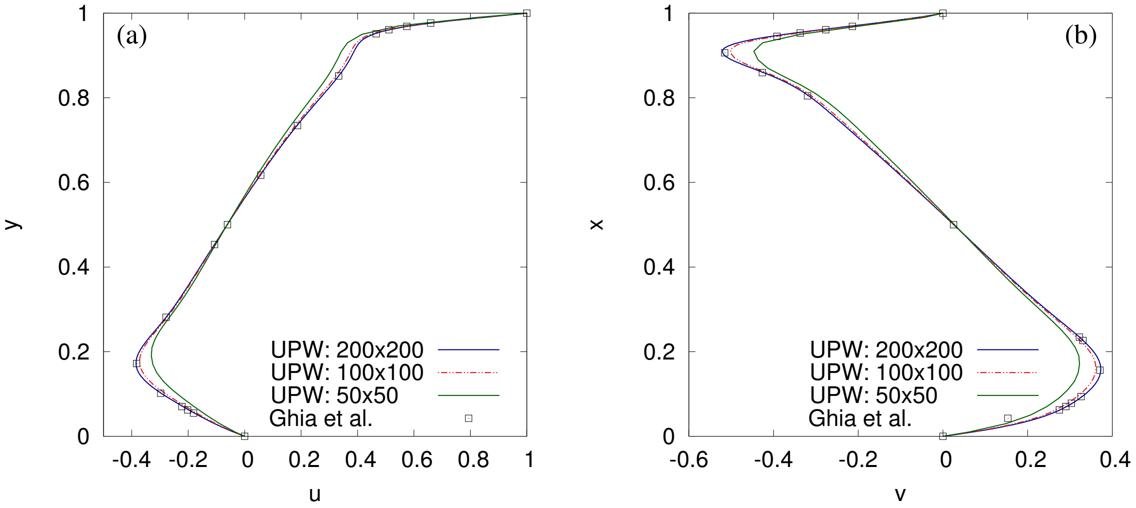


Figure 4.3: Numerical results of lid-driven cavity flow problem using CIP-CSLR with UPW formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

The numerical results of the central difference formulations are presented in figures 4.4, 4.5, 4.6, and 4.7. These results indicate that central difference formulations are superior to UPW in this test problem. Although all central difference approximations give similar results, CDbca and CDcc are slightly better than CDb and CDca as shown in fig. 4.11 (enlarged figure). The difference between the results by CDbca and CDcc is barely seen. Furthermore, CDb is superior to CDca in this test problem. The result by mixed formulations (UPW-CDcc) and (UPW-CDbca) are shown in figures 4.8 and 4.9, respectively.

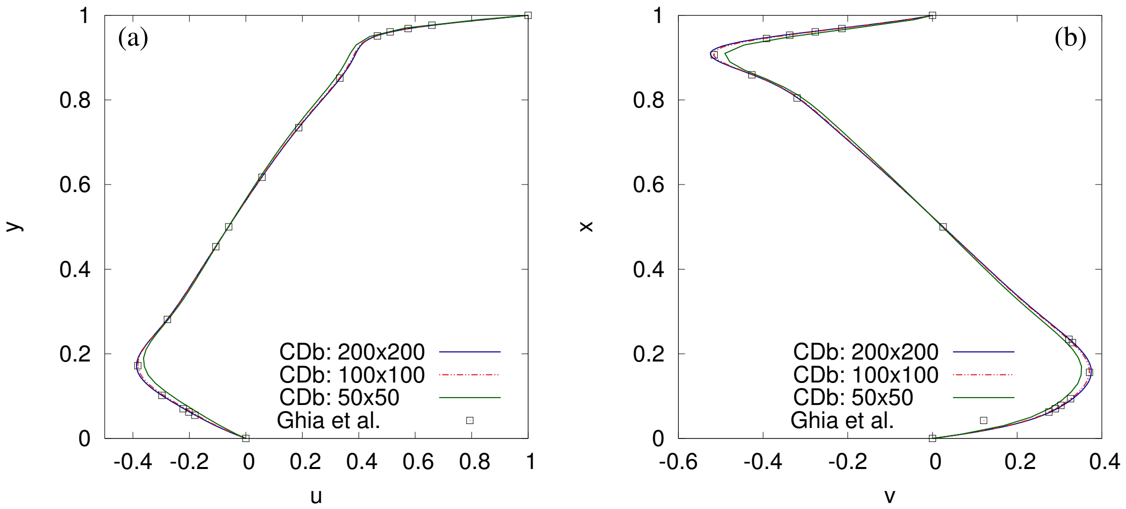


Figure 4.4: Numerical results of lid-driven cavity flow problem using CIP-CSLR with CDb formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

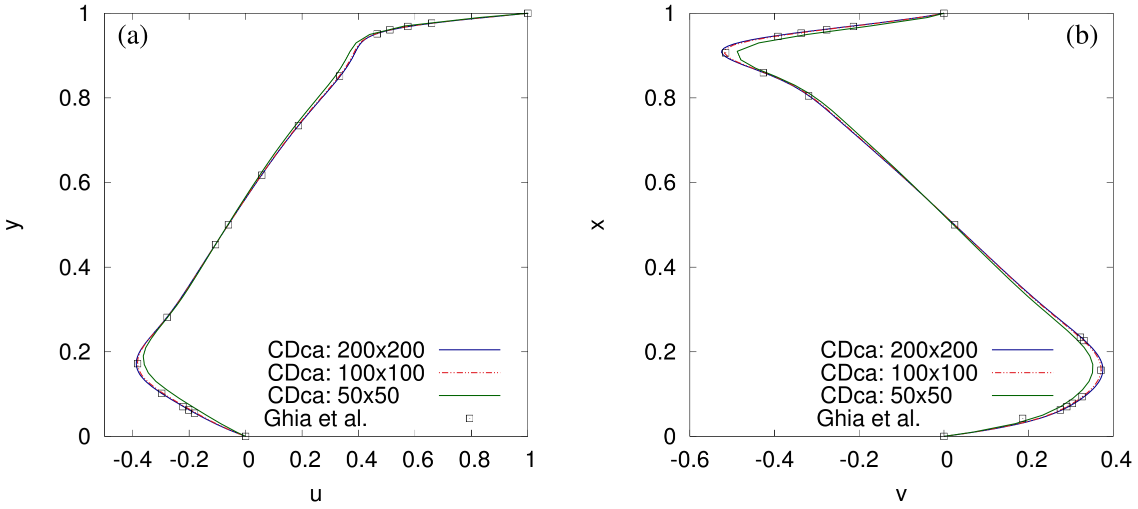


Figure 4.5: Numerical results of lid-driven cavity flow problem using CIP-CSLR with CDca formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

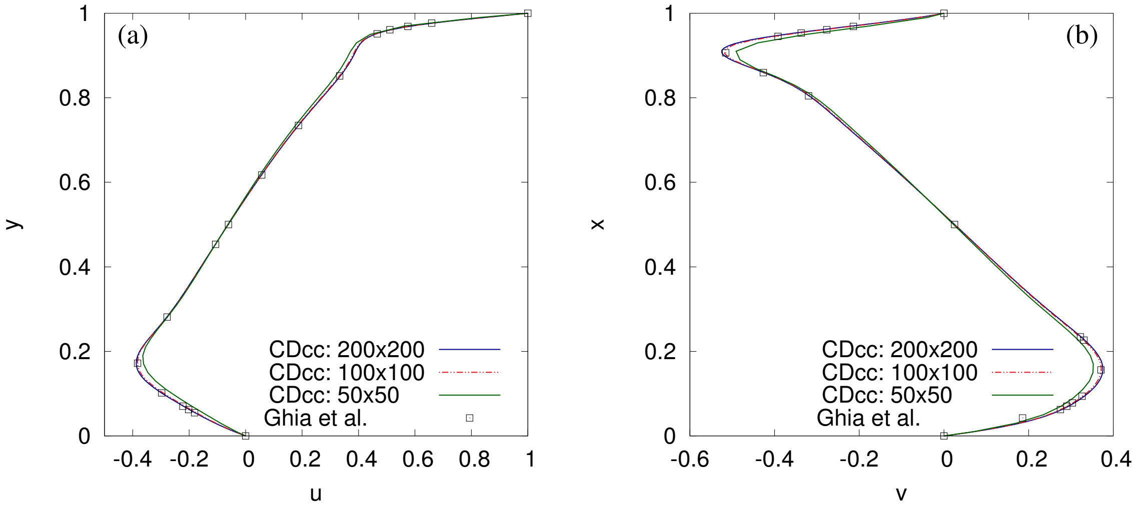


Figure 4.6: Numerical results of lid-driven cavity flow problem using CIP-CSLR with CDbcc formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

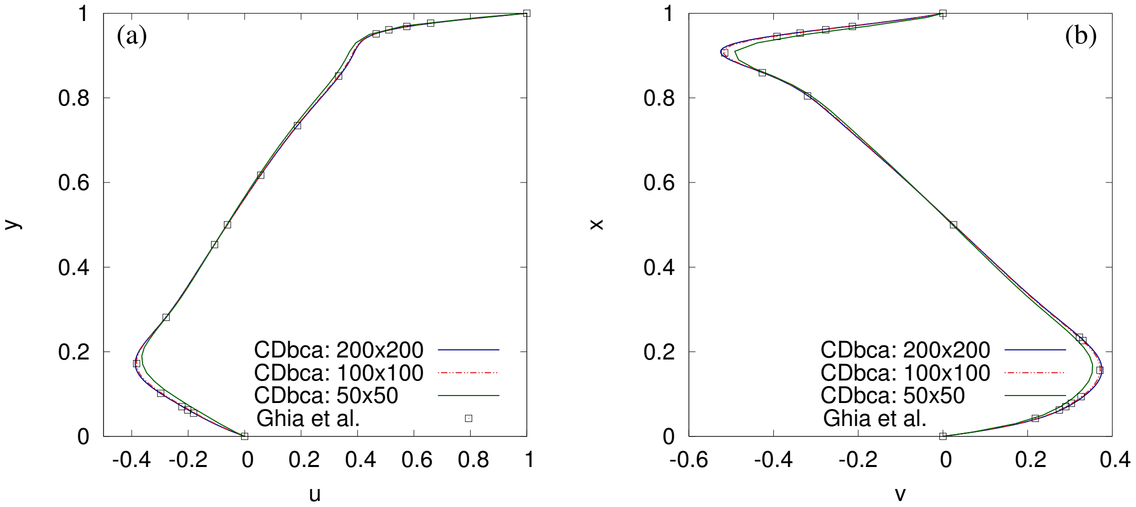


Figure 4.7: Numerical results of lid-driven cavity flow problem using CIP-CSLR with CDbca formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

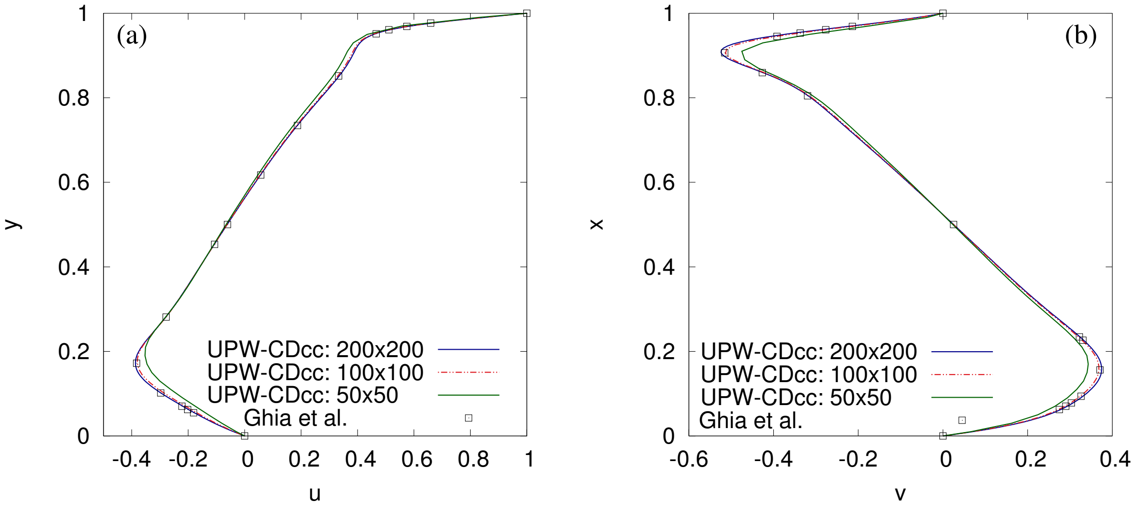


Figure 4.8: Numerical results of lid-driven cavity flow problem using CIP-CSLR with UPW-CDbcc formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

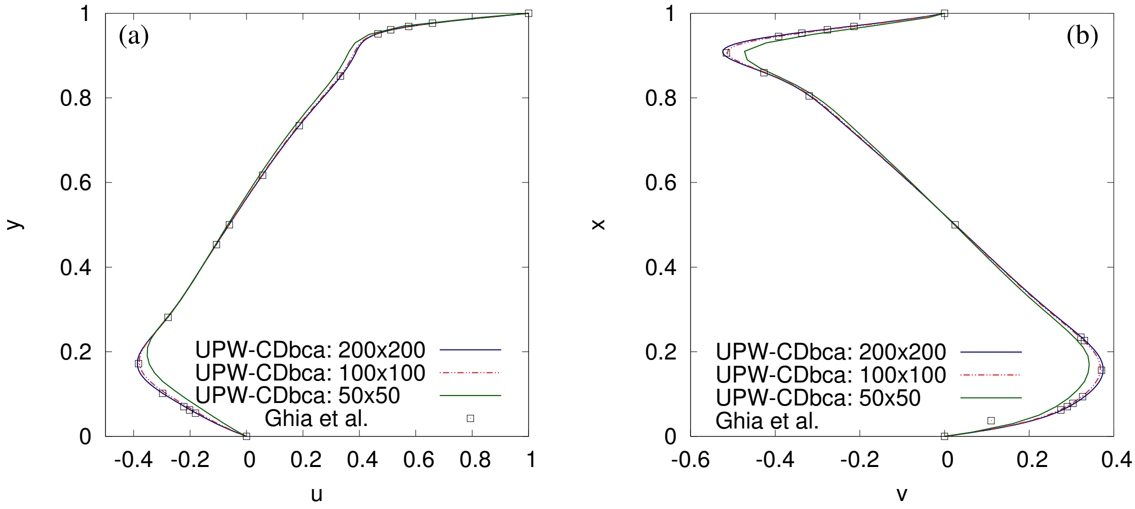


Figure 4.9: Numerical results of lid-driven cavity flow problem using CIP-CSLR with UPW-CDbca formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

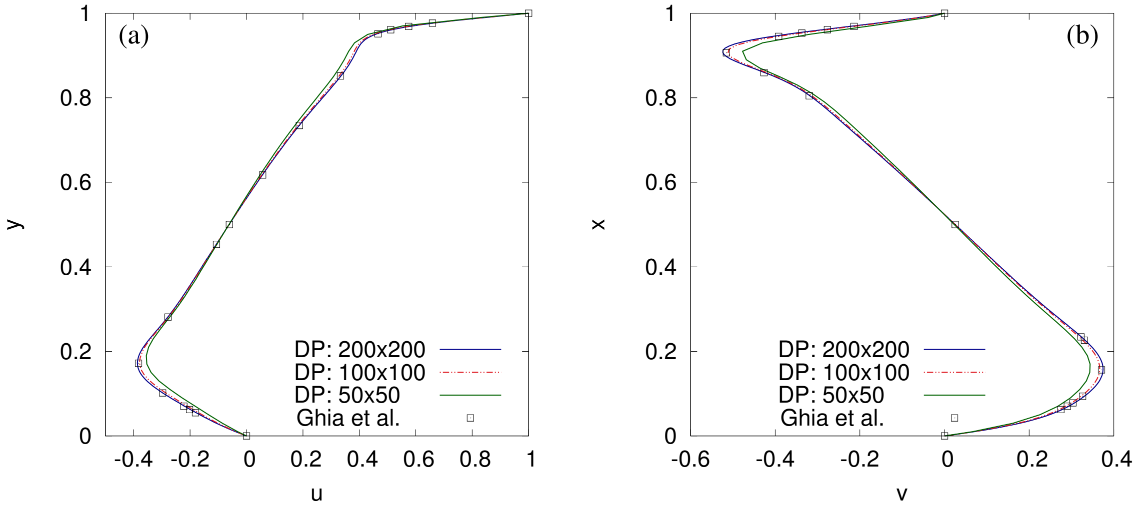


Figure 4.10: Numerical results of lid-driven cavity flow problem using CIP-CSLR with DP formulation for the divergence term. Re = 1000. Three different grid sizes (50×50, 100×100 and 200×200) were used.

Figure 4.12 gives a comparison among UPW, CDcc and UPW-CDcc. The result by UPW-CDcc is closer to the Ghia solution than that by UPW. The results by UPW-CDbca are almost identical with these by UPW-CDcc. In this test problem, the mixed formulations have no advantage over central difference approximations. Nonetheless, as shown in 4.3.3, the mixed formulation plays an essential part in complex simulations such as droplet splashing. The result by DP is better than that by UPW but worse than these by central diference approximations in this test as shown in figures4.10 and 4.11.

Figure 4.13 gives numerical results of Re$= 5000$. The pattern of the results is almost same with the results of $Re = 1000$.

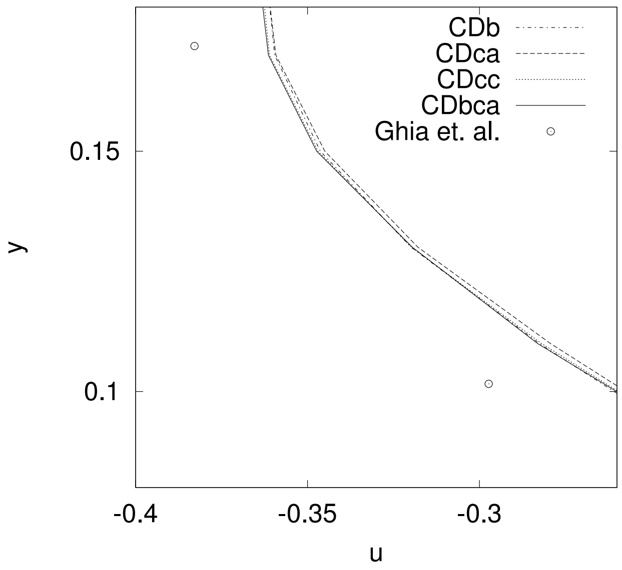


Figure 4.11: Comparison among numerical results by CSLR-CDb, CSLR-CDca, CSLR-CDcc, CSLR-CDbca and CSLR-DP. A Cartesian grid of 50×50 was used.

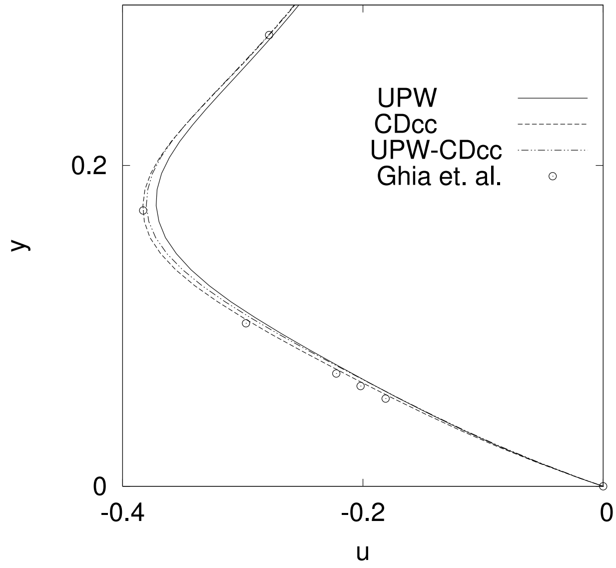


Figure 4.12: Comparison among numerical results by CSLR-UPW, CSLR-CDcc and CSLR-UPW-CDcc. A Cartesian grid of 100×100 was used.

4.3.2 Compressible Flows (Sod's and Lax's Problems, and 2D Explosion Test)

We demonstrate the effects of the proposed discretization techniques for the velocity divergence term through benchmark problems incompressible flows; Sod's problem [152] and Lax's problem [101]. The initial condition of Sod's problem is

$$(\rho, u, p) = \begin{cases} (1.0, 0, 1.0) & \text{if} \quad x \leq 0 \\ (0.125, 0, 0.1) & \text{if} \quad x > 0. \end{cases} \tag{4.24}$$

The initial condition of Lax's problem is

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528) & \text{if} \quad x \leq 0 \\ (0.5, 0, 0.571) & \text{if} \quad x > 0. \end{cases} \tag{4.25}$$

All computations were performed on a 400-point uniform grid. Figures 4.14 and 4.15 show numerical results of Sod's and Lax's problems, respectively. Table 4.2 gives L_1 errors in Sod's and Lax's problems.

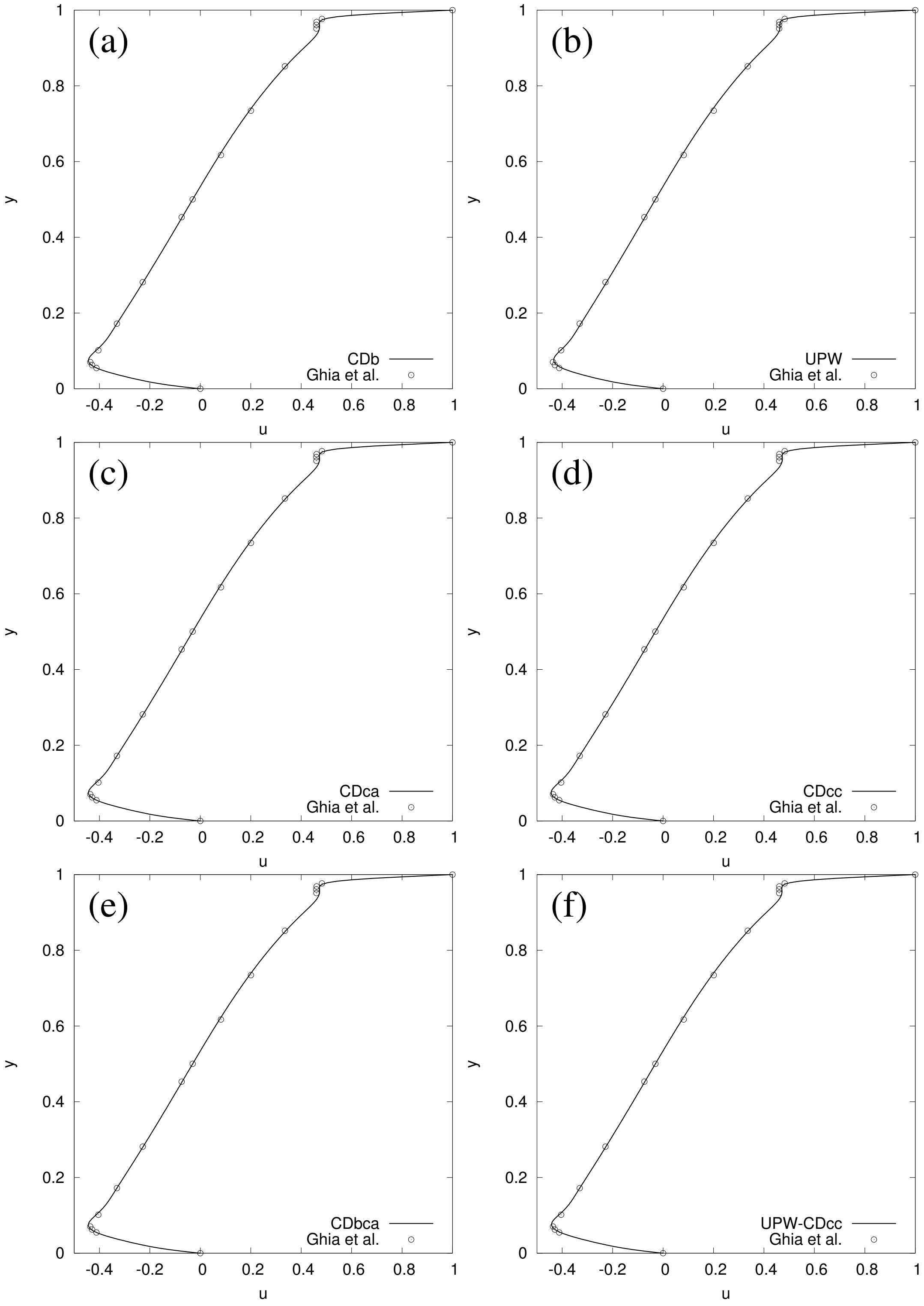


Figure 4.13: Numerical results of lid-driven cavity flow using six different formulations for the divergence term. Re = 5000. A Cartesian grid of 256×256 was used.

Numerical results of Sod's problem by various divergence term formulations excluding DP are almost similar as can be seen in fig. 4.14. Regarding L_1 errors, CDbca is the most accurate in Sod's problem, and UPW is the worst as indicated in table 4.2. The other central difference approximations (CDb, CDca and CDcc) fundamentally results in similar errors ($2.24 \times 10^{-3} - 2.25 \times 10^{-3}$). Intermediate errors between those of UPW and CDcc (or CDbca) are shown by mixed formulations (UPW-CDcc and UPW-CDbca). DP formulation could not simulate Sod's problem. Computations using DP was not stable for this test problem. The unstable calculations using DP formula could be due to the discontinuity in the velocity immediately after the start of the numerical simulation. DP formulation tends to create a substantial value around discontinuity compared to other approximations, and this may cause unstable computations. However if we simulated first 10 time steps using UPW, it was able to simulate this problem. This means that DP is not so good at the strong discontinuity which appears at the beginning of this numerical simulation.

Regarding Lax's shock tube problem, some differences in the numerical results can be observed particularly around the contact discontinuity as shown in fig. 4.15. It is evident from fig. 4.15 and table 4.2 that the result by CDb is the worst. This is because Lax's problem includes a discontinuity in the initial velocity condition and the simple central difference approximation (CDb) which requires a longer stencil than the other approximations cannot manage such discontinuity well. Although DP succeeds in simulating Lax's problem, it was the second worst. This will be because of the sharp discontinuity in the initial condition. Although CDbca was the most accurate in Sod's problem (which does not involve the discontinuity in the initial velocity condition), it is the second worst in central difference methods. This will also be because CDbca uses a longer stencil like that of CDb. At the same time, CDcc is the most accurate and CDca second best in Lax's problem. This will be because CDcc and CDca approximations use a shorter stencil than those of CDb and CDbca. UPW is less accurate than CDcc and CDca but more accurate than CDb and CDbca, and has a small numerical oscillation around the shock (the similar oscillation has also been observed in the previous work by Xiao [183]). When using the mixed formulation, the small oscillation which appears in the numerical result by UPW disappears and numerical diffusion immediately before/after discontinuities, which is observed in numerical results by all central difference approximations is also reduced. Although the mixed formulation presents some enhancements, the results by CDcc and CDca are still more accurate in terms of L_1 error because the mixed formulation is slightly more diffusive on the discontinuities.

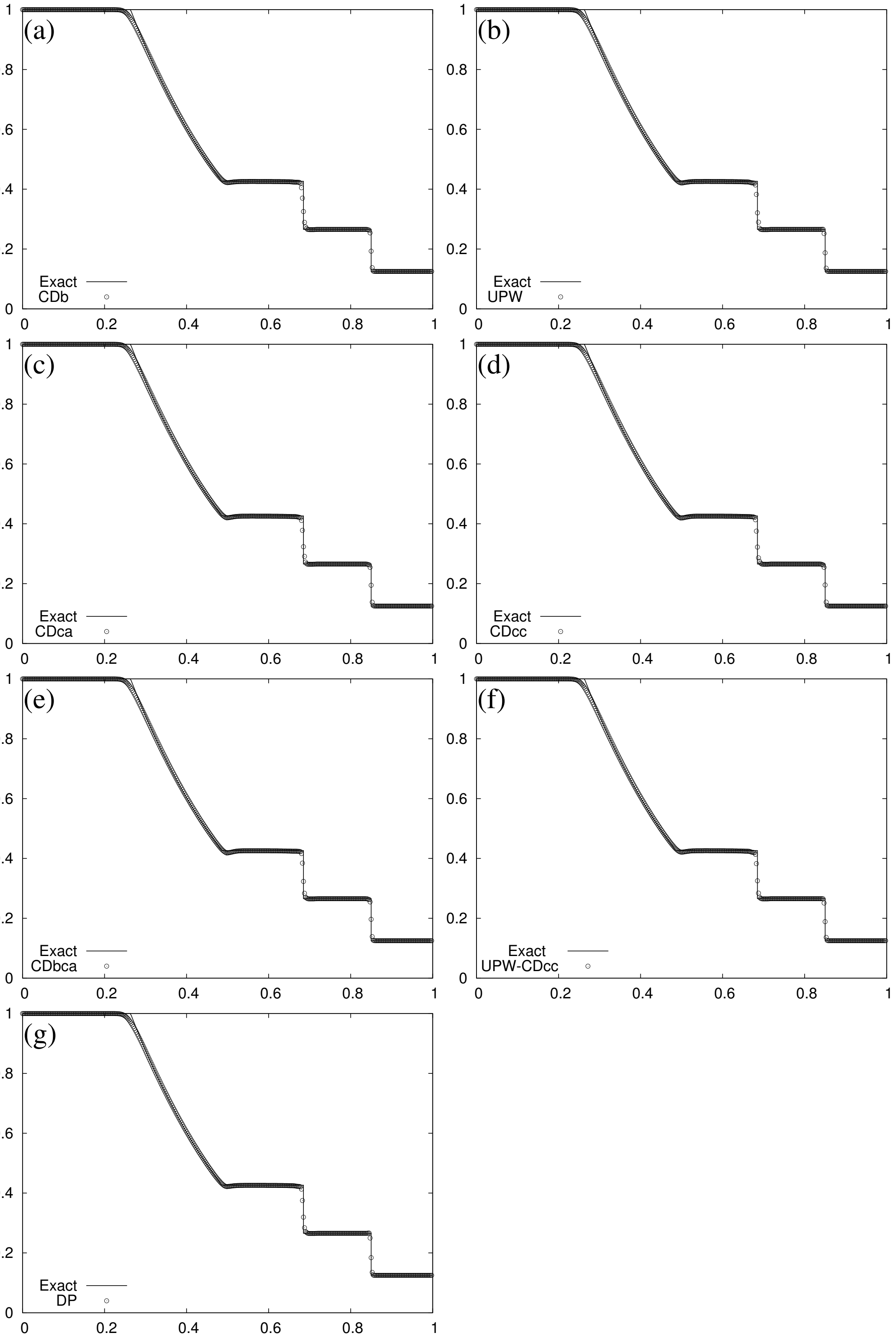


Figure 4.14: Numerical results of Sod's Problem. Plotted are density profiles vs. axial distance. The dots show the density profile of numerical results. The line shows the exact solution.

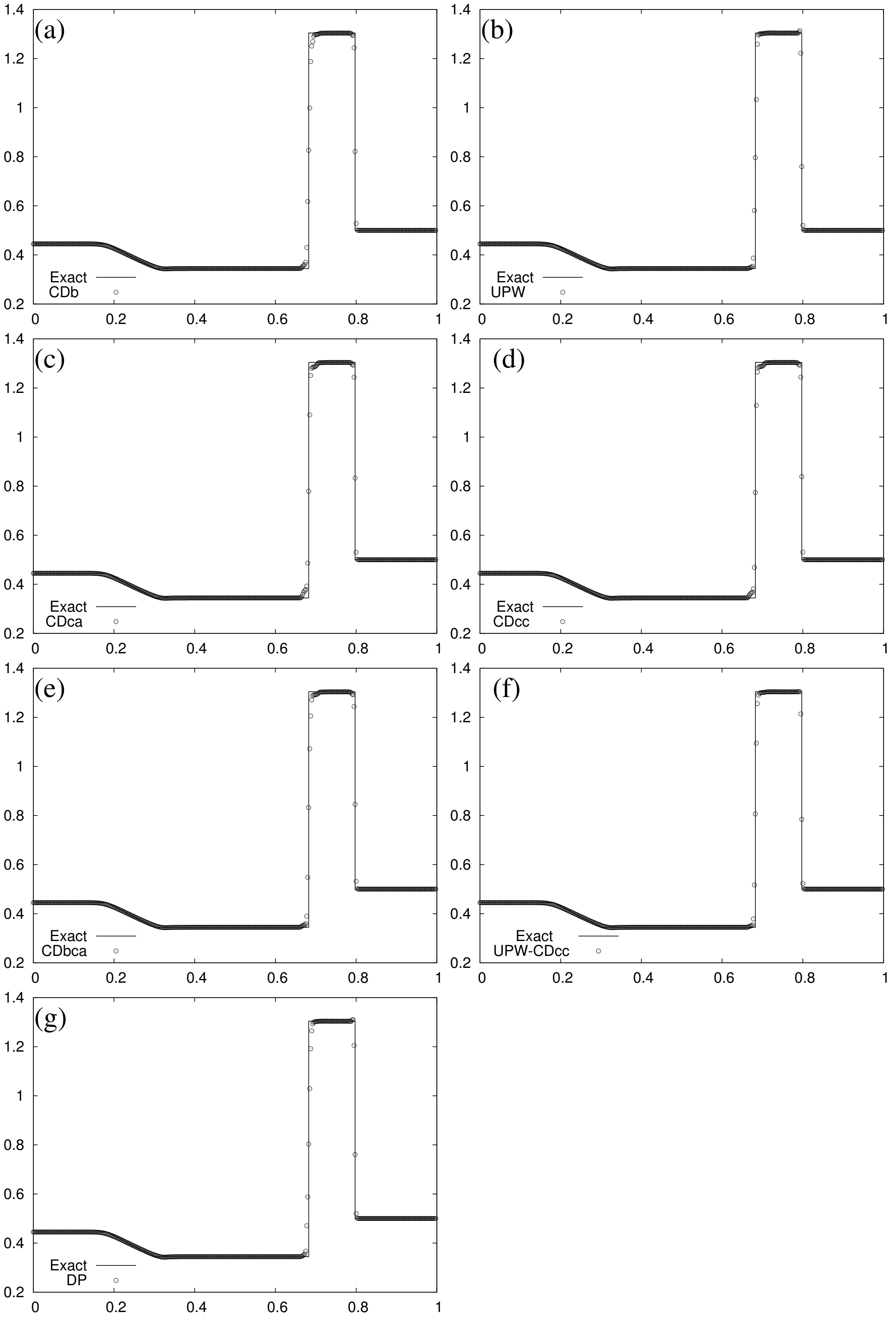


Figure 4.15: Numerical results of Lax's Problem. Plotted are density profiles vs. axial distance. The dots show the density profile of numerical results. The line shows the exact solution.

We also carried out numerical simulations of two-dimensional compressible flow (2D exploion test) [167] on the domain $[-1,1] \times [-1,1]$. The initial condition of the 2D explosion problem is

$$\begin{array}{lllll} \rho(x,y,0)=1; & u(x,y,0)=0; & v(x,y,0)=0; & p(x,y,0)=1; & \text{if } r<0.5 \\ \rho(x,y,0)=0.125; & u(x,y,0)=0; & v(x,y,0)=0; & p(x,y,0)=0.1; & \text{otherwise,} \end{array} \tag{4.26}$$

where $r = \sqrt{x^2+y^2}$. The inviscid Euler conservation laws are solved on a 200×200 Cartesian grid. The perspective view of the density at t=0.25 is presented in fig. 4.16. Those formulations of the velocity divergence term can straightforwardly be applied to multi-dimensional fluid problems, and the pattern is almost same with that discussed in Sod's problem.

Nonetheless, as indicated in 4.3.3 , the mixed formulation plays an essential part in complex flow simulations.

4.3.3 Divergence Term Formulations in Complex Free Surface Flows

Numerical simulations of the effects of the divergence term formulations in VSIAM3 in a complex gas-liquid two-phase problem like droplet splashing on a superhydrophobic substrate has been carried out. Parameters of the simulations are given in table 4.1. A uniform Cartesian grid of $192 \times 48 \times 192$ is used. The details of the simulations are given in [5]. The results are shown in fig. 4.17. The results indicate that VSIAM3 with CIP-CSL2-UPW was not stable after a short time of the start of the computations as shown in fig.4.17 a. Moreover, VSIAM3 using CIP-CSL2 with any central difference formula was not stable for this test problem. On the other hand, VSIAM3 with CIP-CSLR is stable when UPW was employed for the velocity divergence term as shown in fig.4.17 b. However, when using any central difference formulation for the velocity divergence term, VSIAM3 with CIP-CSLR was also unstable. Furthermore, utilizing a mixed formulation (UPW-CDcc), VSIAM3 with CIP-CSLR could accomplish stable numerical simulation of droplet splashing and capture droplet splashing well as indicated

Table 4.1: Quantitative parameters of the droplet splashing simulations. ρ is density, μ is viscosity, D initial droplet diameter, σ is surface tension, v is impact speed, and θ the equilibrium contact angle.

Phase	$\rho(kg/m3)$	$\mu(Pa.s)$	$D(mm)$	$\sigma(N/m^2)$	v (m/s)	$\theta(^\circ)$
Liquid	1000	1×10^{-3}	7.2$\times10^{-2}$	1.86	2.98	163
Air	1.25	1.82×10^{-5}	–	–	–	–

in fig.4.17 c.

VSIAM3 with CSL2-UPW could not capture droplet splashing well as shown in Fig. 4.17a. CSL2-UPW also caused relatively large amount of flotsam and was not stable after around 1.1 ms. VSIAM3 with CSL2 with any central difference formulation was not stable for this problem. VSIAM3 with CSLR is stable when UPW was used for the divergence term. The formulation could capture droplet splashing well as shown in Fig. 4.17b. However if we use any central difference formulation for the divergence term, VSIAM3 with CSLR was also unstable. If we use UPW-CDcc (mixed formulation), VSIAM3 with CSLR could conduct stable numerical simulation of droplet splashing and capture droplet splashing well as shown in Fig. 4.17c.

Hereafter CIP-CSLR with UPW-CDbca is employed in the conducted numerical simulations.

4.4 Summary

We studied discretization techniques of the advection transport equation for efficient implementation of VSIAM3 through the lid-driven cavity flow, shock tube problems, and droplet splashing.

We investigated the usage of the CIP-CSL2 scheme in addition to the CIP-CSLR scheme in VSIAM3 through the lid-driven cavity flow and droplet splashing. The nu-

Table 4.2: L_1 errors in shock tube problems.

	Sod	Lax
CDb	2.25×10^{-3}	5.67×10^{-3}
UPW	2.30×10^{-3}	4.94×10^{-3}
CDca	2.24×10^{-3}	4.78×10^{-3}
CDcc	2.24×10^{-3}	4.47×10^{-3}
CDbca	2.21×10^{-3}	5.05×10^{-3}
UPW-CDcc	2.28×10^{-3}	4.61×10^{-3}
UPW-CDbca	2.27×10^{-3}	4.67×10^{-3}
DP	N/A	5.47×10^{-3}

Table 4.3: Summary of numerical results of incompressible flows. In the cavity flow problem, result by CSLR with central difference was slightly better than that by CSLR with mixed formulation.

	Cavity flow	Droplet splashing
CSL2 with upwind	Fairly precise and not robust	Barely capture the phenomenon and not robust
CSL2 with central difference	Not robust	Not robust
CSLR with upwind	Fairly precise and robust	Capture the phenomenon and robust
CSLR with central difference	Precise and robust	Not robust
CSLR with mixed formulation	Precise and robust	Capture the phenomenon and robust

merical results indicated that VSIAM3 using CIP-CSL2 is not robust enough and that VSIAM3 using CIP-CSLR is highly robust (when an appropriate approximation is used for the velocity divergence term). These results show that the use of a less oscillatory formulation (i.e., CIP-CSLR) is a key for robust incompressible flow simulations.

We also found that the numerical results are sensitive depending on discretization approximations of the velocity divergence term in the advection transport equation. The numerical results of the lid-driven cavity flow showed that CIP-CSLR with central difference formulations is superior to the simple upwind formulation in this test. However, both results are reasonably precise. On the other hand, the numerical results of droplet splashing showed that VSIAM3 with any central difference formulation is not robust even though CIP-CSLR is used, while VSIAM3 with the simple upwind formulation was highly robust and captures the droplet splashing well. These results indicate that the use of the upwind formulation is suitable for robust numerical simulations, especially for complex gas-liquid two-phase flows like droplet splashing. Although the central difference formulations are precise for simple flow problems such as cavity flow, will not be robust enough for complicated flow problems.

Based on the numerical results, we also proposed the mixed formulation using both a central difference and the simple upwind formulation for the velocity divergence term. The mixed formulation can simulate the lid-driven cavity well (better than UPW and slightly worse than CDcc) and also simulate droplet splashing like the result using the simple upwind. The mixed formulation can take advantages of both central difference and upwind formulations. We summarize the results of incompressible results in Table 4.3.

We also tested formulations for the divergence term through the inviscid compressible flow problems (Sod's and Lax's problems and 2D explosion test). In Sod's problem which does not involve discontinuity in the initial velocity condition, we could not observe much difference in numerical results by all divergence formulations excluding DP and DP was not stable. In Lax's problem which involves the discontinuity in the initial velocity condition, we could observe some differences, especially around the contact discontinuity. In this test, CDb and CDbca, which use a wider stencil were less accurate than CDcc and CDca, and CDcc and CDca, which use a shorter stencil were more accurate than CDb and CDbca . The mixed formulation shows some improvements compared to numerical results by any central difference or UPW. However, regarding L_1 error, CDcc and CDca are still better than UPW-CDcc.

In conclusion, employing a less oscillatory CIP-CSL scheme (i.e., CIP-CSLR, CIP-CSL3, etc.) with an appropriate formulation for the velocity divergence term is critically important for robust implementation of VSIAM3. As an optimal divergence term formulation, we would suggest the mixed formulation because it is highly robust, more accurate than UPW and works well for both incompressible and compressible flows problems.

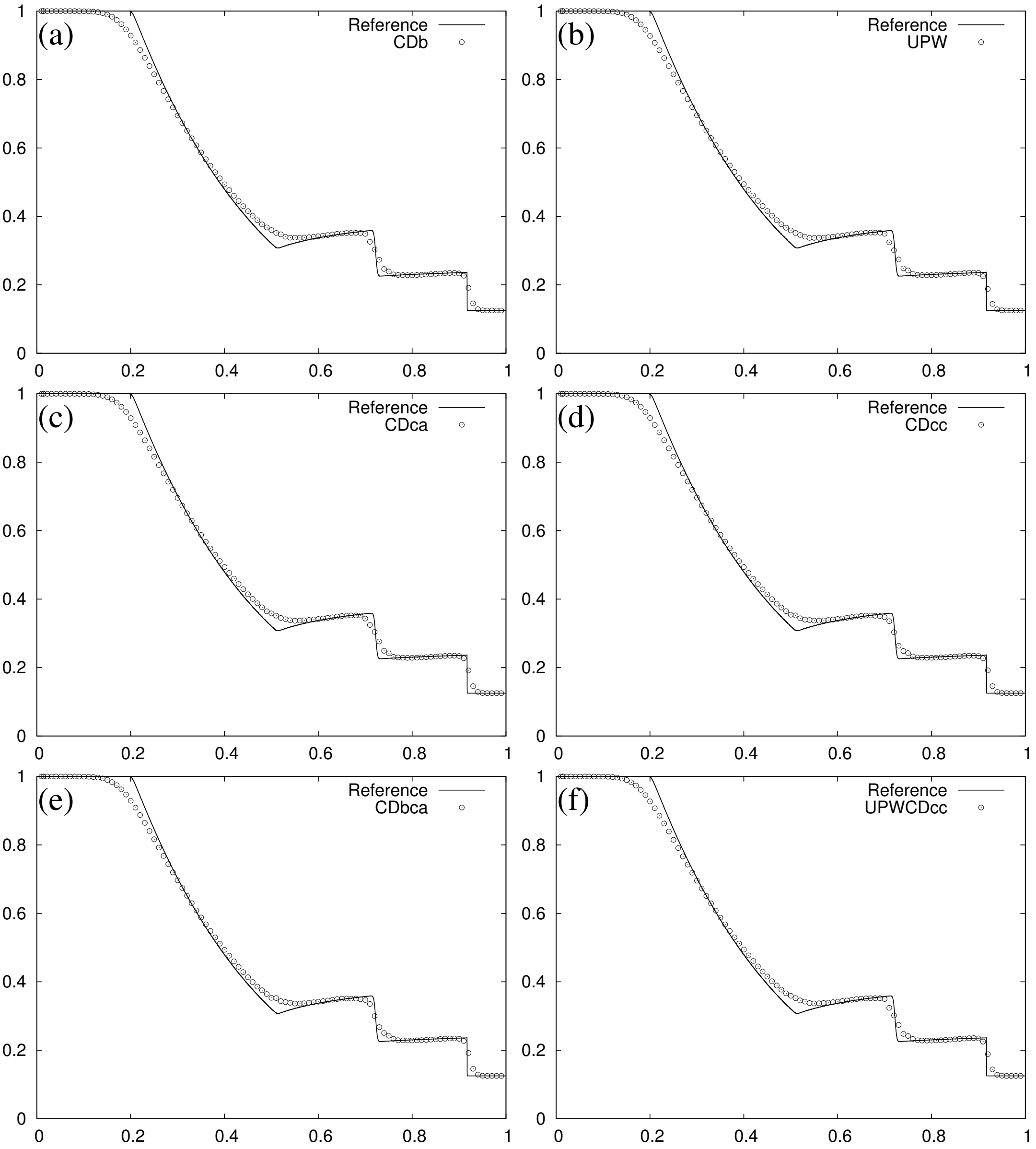
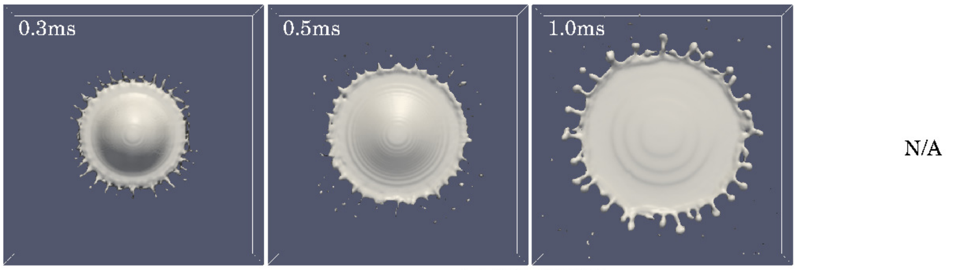
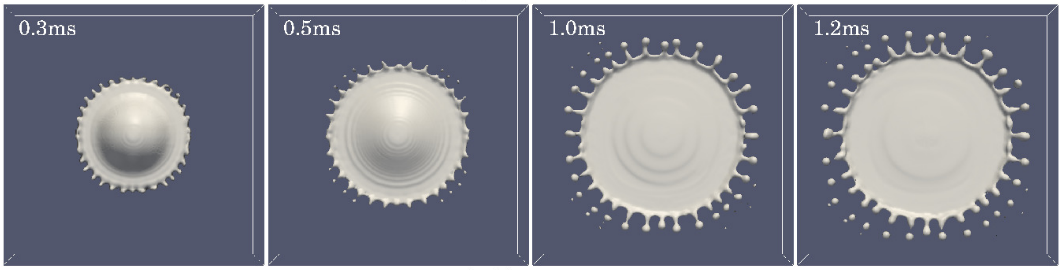


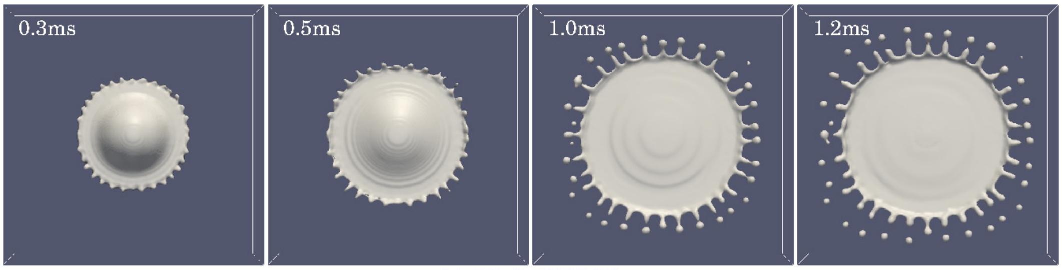
Figure 4.16: The density profiles of the 2-d explosion test at t=0.25 along the line of y = 0 . The dots represent numerical results by using six different formulations for the divergence term. The line represents the reference solution.



(a) CSL2-UPW



(b) CSLR-UPW



(c) CSLR-UPW-CDcc

Figure 4.17: Numerical results of droplet splashing by CSL2-UPW (a), CSLR-UPW (b) and CSLR-UPW-CDcc (c). VSIAM3 with CSL2-UPW was not stable after around 1.1ms.

Chapter 5

Parallel Computation

A parallel implementation of the fluid dynamics and heat transfer solver has been carried out using C++ and Open MPI [59]. The importance of this implementation and its evaluation was presented in this chapter. The solver was further validated by three-dimensional problems; equilibrium drop, single rising bubble problem, Kelvin-Helmholtz instability, and fully developed turbulent channel flow.

5.1 The Necessity of the Parallel Implementation

Numerical simulation of fluid flow and heat transfer in three dimensions obviously involves a higher number of grids than that of two-dimensional simulations and demands longer execution time. Additionally, particular physical phenomena, like Rayleigh Bénard Convection for example, where the driving temperature difference produces slow evolution processes of the velocity and temperature fields, and thus, imposes long computational time with small time steps to attain the steady-state solution. Therefore it is more efficient to write the code for execution on more than single node and high performance computing in general. Numerical simulation of three-dimensional turbulent flows, for instance, requires fine spatial mesh with small time steps to resolve the full range of length and time scales that span the inertial and the dissipative scales. This requirement increases with the Reynolds number (Re). Kolmogorov hypotheses can be used to quantify the required mesh resolution. The Kolmogorov length scale (η) is related to the Reynolds number [12]

$$\frac{\eta}{L} = \frac{uL}{\nu}^{\frac{-3}{4}} = Re^{\frac{-3}{4}}, \tag{5.1}$$

where L, u, and ν are the integral length scale, a typical velocity of the flow and the kinematic viscosity, respectively. The grid size of $\Delta\ \eta$ is required to clarify these length scales, The demanding complexity of DNS increases as a result of the three-dimensional nature of turbulent flows. For instance, the simulations of wind-driven turbulent gas-liquid flows under breaking-waves conditions are performed with fixed domain lengths and relatively high inertial range. This means high gas-phase free stream velocity which results in the reduction of the dissipative scale that is the size of the smallest structures. Considering a domain of dimensions (L^3), the number of grid points needed to resolve the flow

$$N = \left(\frac{L}{\Delta}\right)^3 = Re^{\frac{9}{4}}, \tag{5.2}$$

furthermore, the time step (Δt) considerably decreases with the reduction of the mesh size which adds to the computational cost. Since the ratio of the largest to smallest time scales shows

$$\frac{T}{\Delta t} = Re^{\frac{1}{2}}, \tag{5.3}$$

where the term ($T/\Delta t$) represents the number of time steps necessary for the largest eddies to pass the flow domain. In practice, more substantial amount of time steps are required to allow for the large eddies to pass several times through the flow domain to obtain meaningful turbulence statistics [177]. In addition to the demanding nature of the three-dimensional simulations, the combination of the demanding requirements of the mesh resolution and the time step limitation obviously show the need for parallel computing.

5.2 Open MPI and Domain Decomposition

Domain decomposition was used to split the computational domain into the required number of local subdomains for parallel computing. In the present implementation, the problem domain can be separated into local parts by one-dimensional partitioning to three-dimensional partitioning depending on the spatial mesh size and the available number of processors (5.1). This results in the suitable size of the data being communicated among various processors in a distributed memory architecture. The Message Passing Interface (Open MPI) implementation [59] was employed for data communication among processors to be used on distributed memory computers. The physical variables are defined as local arrays for each core. The parallel implementation deploys the single instruction multi-data manner. This means the local data is assigned for different processors and the sequence of computations is alike throughout various processors, which leads to better computational load balancing.

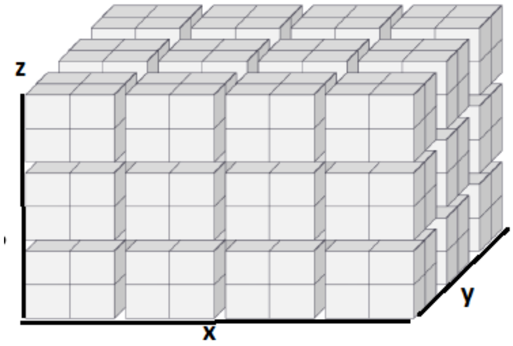


Figure 5.1: Three-dimensional domain decomposition

5.3 Evaluation of the Parallel Performance

The performance tests were run on the supercomputer Raven Advanced Research Computing at Cardiff University (ARCCA System-Raven). Raven system is a Bullx b510, a Linux based cluster with 2048 cores (16 cores per node) based on Intel Xeon e5-2670 2.6GHz. For the scaling tests, the three-dimensional gas-liquid two-phase flows parallel code was run with a grid size of (288^3; i.e., about 23887872 cells). Considering the availability of the system, the tests were carried out on 1, 32, 64, and 128 cores only. We attempted to run the test on 256 processors, however, after a queue

Table 5.1: The performance of the parallel implementation

Nodes	Cores	Time	S	E	PR$_{eff}$
1	1	3982.63	-	-	-
2	32	185.9	21.42	66.98	98.41
4	64	74.12	53.73	83.96	99.70
8	128	40.48	98.38	76.86	99.76

waiting time of more than two months, the code never started and was terminated by ARCCA staff. Denoting the execution time with n processor by t_n, the speedup (S) due to the parallel computation

$$S = \frac{t_{n1}}{t_{n2}}, \tag{5.4}$$

where $n1 < n2$.

The parallel efficiency (E) of $n2$ processors relative to $n1$ processors is

$$E = S\frac{n1}{n2}. \tag{5.5}$$

The effective parallelization ratio PR_{eff} is then given by

$$PR_{eff} = \frac{t_{n1} - t_{n2}}{\frac{n2-1}{n2}t_{n1} - \frac{n1-1}{n1}t_{n1}}. \tag{5.6}$$

The results of the parallel performance are given in table 5.1. An acceptable performance has been obtained. The speedup curve relative to the execution time using one processor is presented in fig.5.2. Figure 5.3 shows the speedup curve relative to the execution time on the 32 cores test.

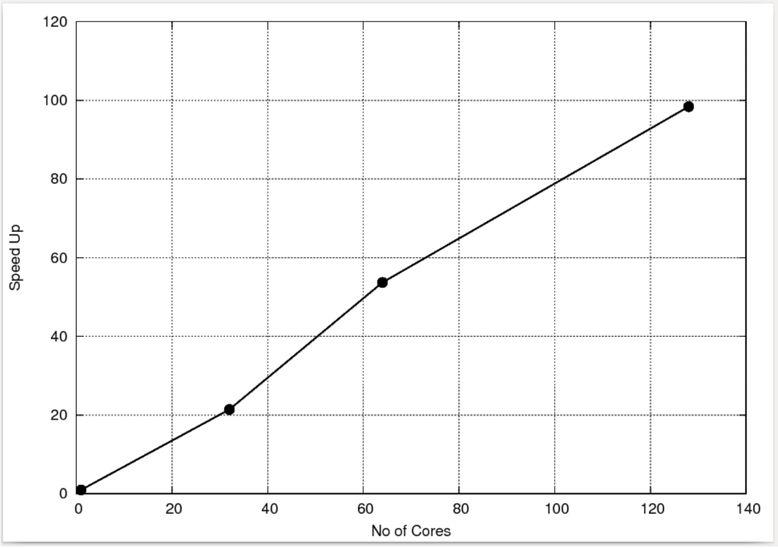


Figure 5.2: The speedup curve relative to the execution time one processor.

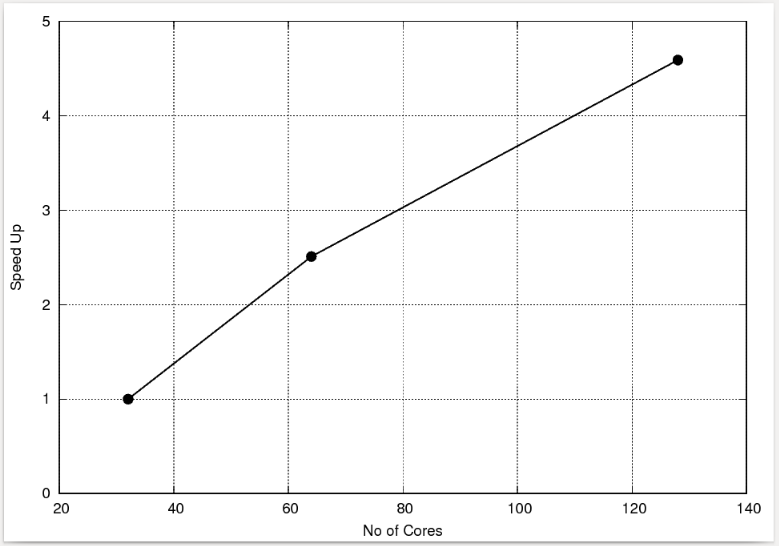


Figure 5.3: The speedup curve relative to the execution time on the 2 nodes (32 cores) test.

Table 5.2: The quantitative parameters used in the numerical simulation of the static drop test .

Droplet radius cm	Liquid density g/cm^3	Air density g/cm^3	Surface tension $dyne/cm$	Gravitational acceleration cm/s^2
1	1	0.001	1	0

5.4 Validations

We further validate the proposed VSIAM3 for gas-liquid two-phase flows including the parallel implementation through equilibrium drop test, single rising bubble problem, Kelvin-Helmholtz instability problem, turbulent channel flow, and simulation of wind wave.

5.5 Equilibrium Drop

To further validate the proposed VSIAM3 including the parallel implementation, and to inspect the performance of the surface force model and curvature calculation, as well as, the pressure Poisson solver, a numerical simulation of static drop problem [16] was carried out. In the test problem, the liquid is inviscid. The other quantitative parameters of the test problem; the radius of the drop 1 cm, liquid density 1 g/cm^3, air density 0.001 $dyne/cm^3$, and the surface tension coefficient 1 $dyne/cm$. The gravitational acceleration 0 cm/s^2. The quantative parameters are given in table 5.2. A regular Cartesian mesh of $40 \times 40 \times 40$ cell is employed for the benchmark problem. The velocity field of the numerical results is shown in fig.5.4. Figure 5.5 shows the pressure solution of the static drop, where the analytic value of the pressure, P, can be obtained by the force balance ($P\pi R^2 = 2\pi R\sigma$), and, ($P = 2\sigma/R$), where σ is the surface tension coefficient and R is the drop radius.

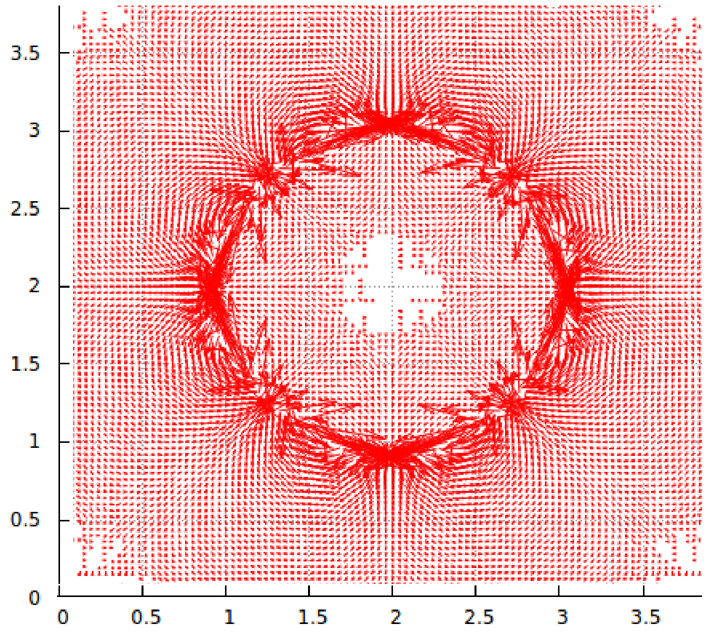


Figure 5.4: Spurious currents in the numerical simulation of the equilibrium drop.

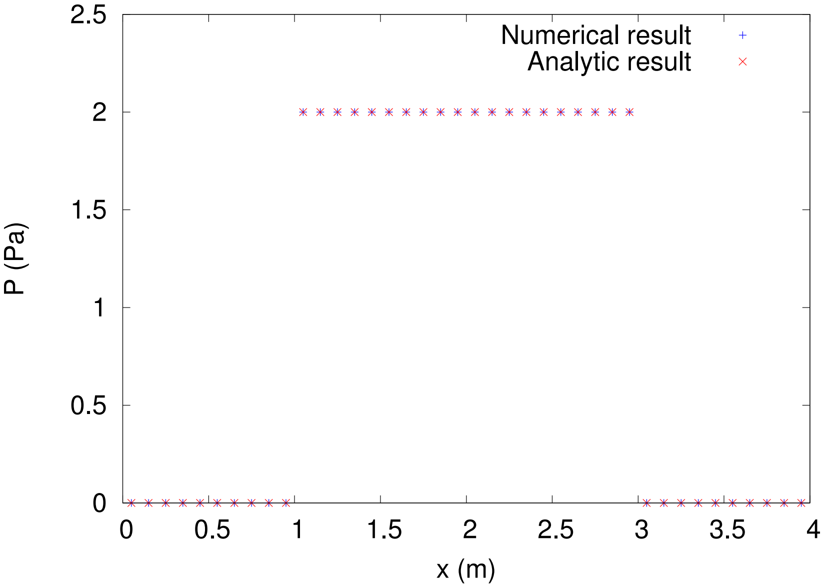


Figure 5.5: The pressure of the numerical result of the equilibrium drop.

Table 5.3: The quantitative parameters used in the numerical simulation of the rising bubble .

Volume mL	Density kg/m^3	Viscosity $Pa.s$	Surface tension N/m	Rise velocity m/s	Re_r -
0.94	875.5	$0.118e^{-5}$	0.032	0.215	9.8

5.6 Single Rising Bubble

To further validate the fluid flow solver, a three-dimensional numerical simulation of single rising bubble has been carried out. This test problem has been used for validating many two-phase flow codes (e.g. [158, 162, 180, 106, 36]). We compare the numerical results with the experimental data of the spherical cap bubbles presented in [75] for the case of $Re_r = 9.8$, where Re_r is Reynolds number based on bubble radius. In the test problem, we used a Cartesian grid of 64×64×256 cells, and the domain dimensions are 3D×3D×12D, where D is the initial bubble diameter. The bubble volume and the physical properties are given in table 5.3. Figure 5.6 displays snapshots of the numerical result. While using high density and viscosity ratios, a comparable terminal rise velocity of ($Re_r = 9.7$) was obtained. A comparison between the numerical result and the laboratory measured terminal rise velocity is plotted in fig.5.7.

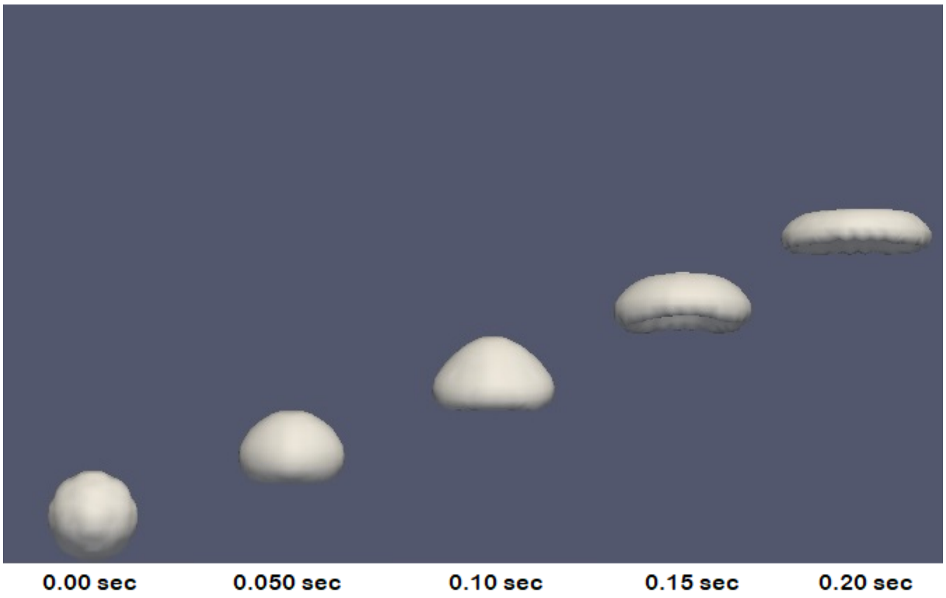


Figure 5.6: Snapshots of the numerical simulation of a single rising bubble.

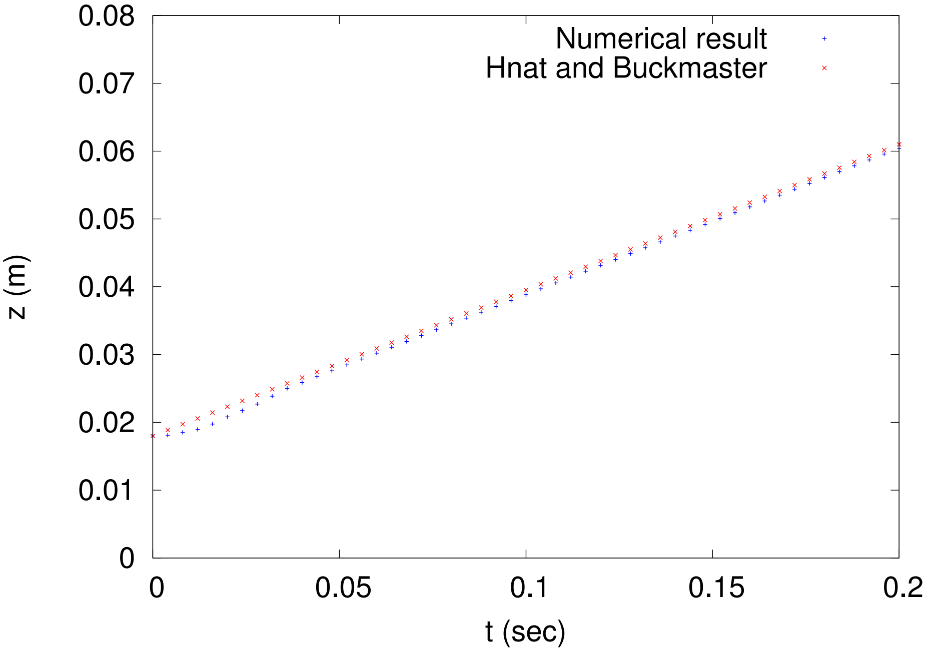


Figure 5.7: A comparison between the numerical result of the bubble rising velocity and the experimental result (0.215 m/s) [75].

5.7 Kelvin-Helmholtz Instability

The Kelvin-Helmholtz instability problem [21, 129, 137] was used to validate the fluid flow solver further and also to validate the application of periodic boundary conditions applied to the simulation of turbulent flows. Helmholtz in (1868) first recognized the fundamental flow problem, but the mathematical formulation of instability was first proposed and worked out by Kelvin in (1871); thus the name is the Kelvin-Helmholtz instability [38]. Kelvin–Helmholtz instability can occur when velocity shear is present within a continuous fluid or when there is sufficient velocity difference across the interface between two fluids. We consider a problem of two fluids flow counter-currently in a domain of size $1m \times 1m \times 0.5m$ in the x, y, and z directions, respectively. The interface was given by a sinusoidal wave in the diagonal direction as shown in fig.5.8,

$$z = a \sin\left(12\pi\sqrt{\frac{5.0}{3.0}}\left(\frac{x+y}{\sqrt{l_x^2+l_y^2}}\right)\right), \tag{5.7}$$

with a$= 0.01m$ the wave amplitude, l_x and l_y are the longitudinal and the transverse dimensions of the flow domain, respectively. The initial velocity field is then given by

$$u = U\frac{l_x}{\sqrt{l_x^2+l_y^2}}, \tag{5.8}$$

$$v = U\frac{l_y}{\sqrt{l_x^2+l_y^2}}, \tag{5.9}$$

$$w = 0.0, \tag{5.10}$$

where U $= 3.0$ and -3.0 m/s for the lower and upper phases, respectively. Periodic boundary conditions were applied in the longitudinal and transverse directions, while the top and bottom of the flow domain were subjected to a slip boundary condition. The number of mesh points is $100 \times 100 \times 50$ in the x, y, and z direction, respectively. Snapshots of the numerical result of the instability at various times are shown in figures

5.9 and 5.10. Figure 5.11 shows top-view of the instability at time$= 0.06$ sec. As can be seen, these figures cleanly depicts the evolution of the instability and validate the periodic boundary conditions as can be seen in the corners of figures (5.9 - 5.11).

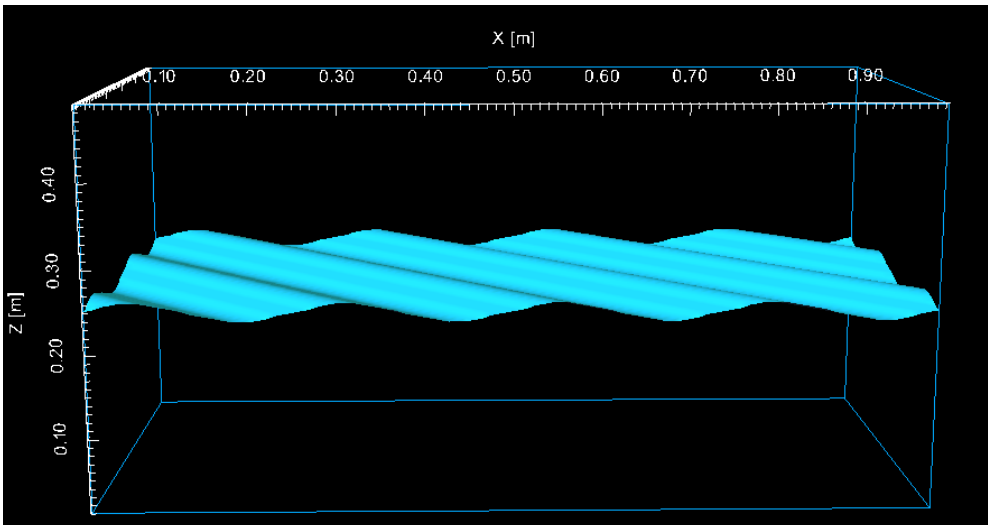


Figure 5.8: Initial configuration of the Kelvin-Helmholtz instability problem.

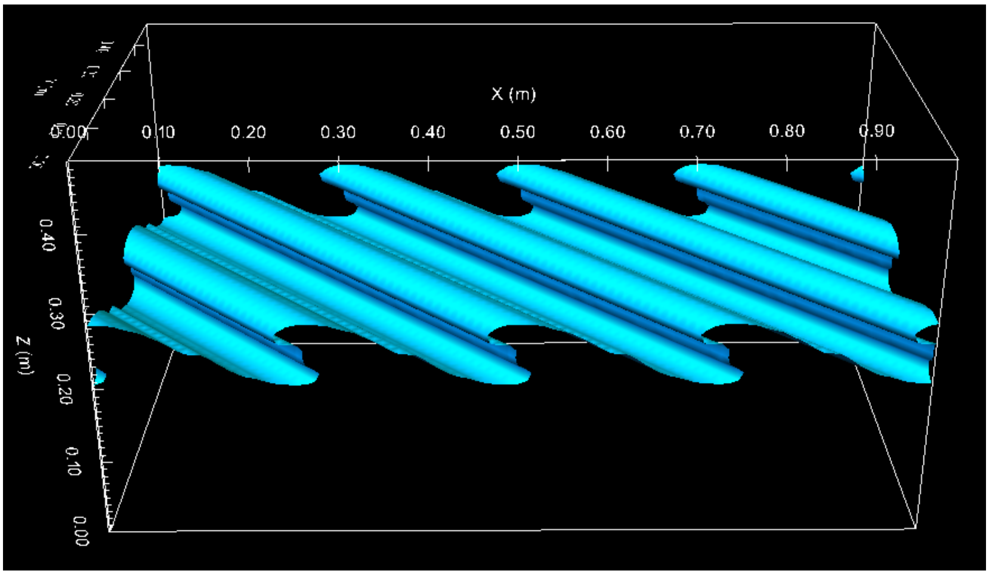


Figure 5.9: Snapshot of the numerical result of the Kelvin-Helmholtz instability at time$= 0.04$ **sec.**

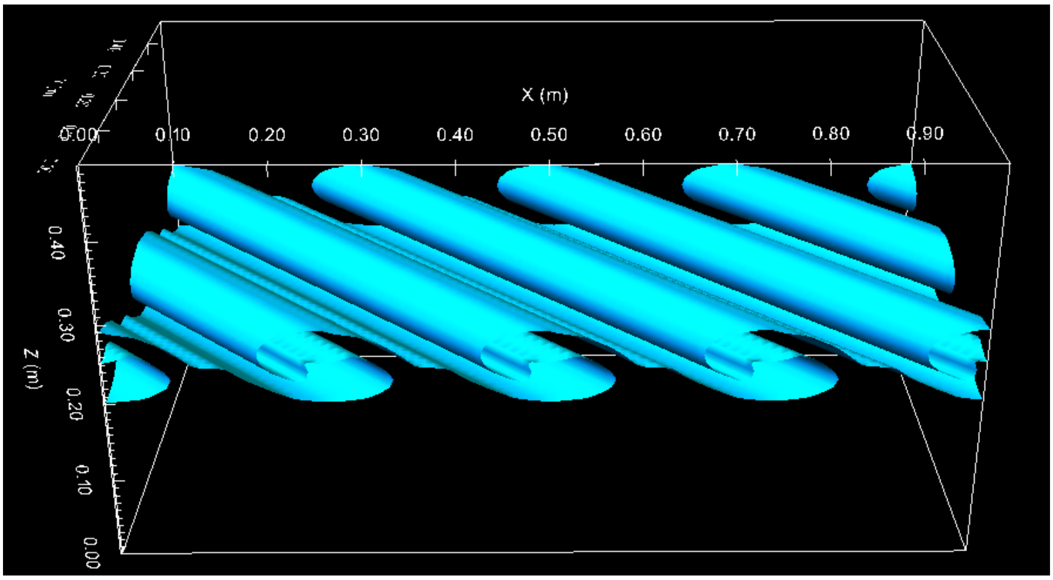


Figure 5.10: Snapshot of the numerical result of the Kelvin-Helmholtz instability at time$=0.06$ **sec.**

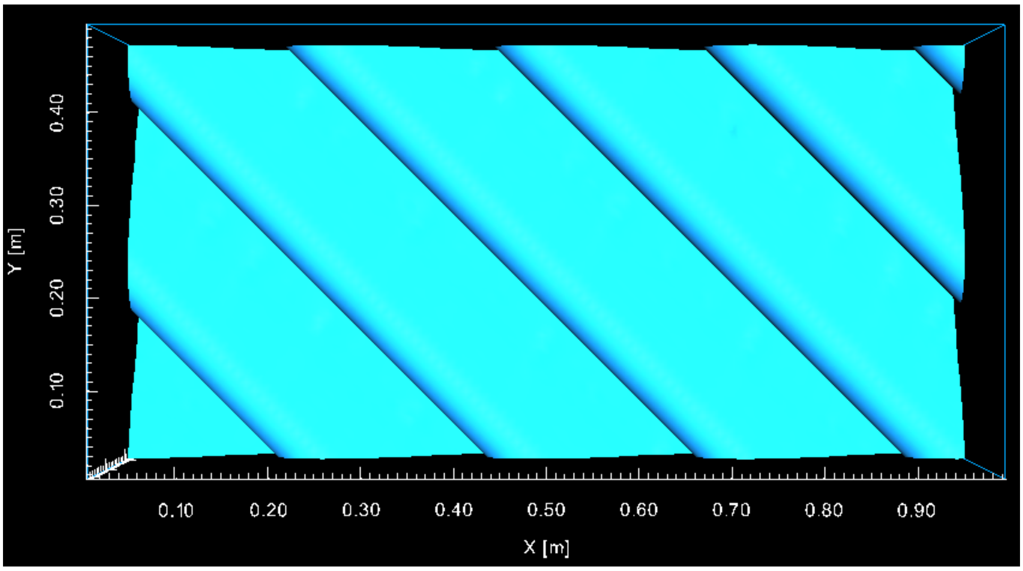


Figure 5.11: Snapshot of the top-view of the instability at time$=0.06$ **sec**

5.8 Numerical Simulation of Turbulent Channel Flow

Turbulent channel flow case is considered to validate the proposed VSIAM3 solver further and to attain a fully developed turbulent channel flow which can be used to initiate various moments for the simulation of turbulent wind wave. A schematic representation of the flow domain is shown in fig. 5.12. A fully developed flow driven by constant pressure gradient was considered. Assuming constant fluid properties and constant pressure gradient in the flow direction ($F_i = (F_1, F_2, F_3) = (1, 0, 0)$), the dimensionless governing equations [88] can be written as

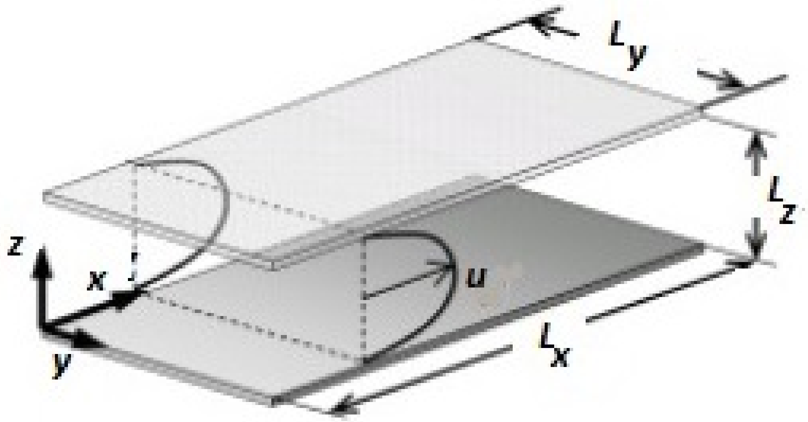


Figure 5.12: Schematic figure of the turbulent channel flow

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{5.11}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{1}{Re_\tau}\frac{\partial^2 u_i}{\partial x_i x_j} + F_i. \tag{5.12}$$

The governing equations are scaled by the domain half height δ as the representative length and the turbulent frictional velocity (u_τ) as the representative velocity. Where the velocity u_i is scaled by u_τ, P is the pressure normalized by ρu_τ^2. $Re_\tau = \frac{u_\tau \delta}{\nu}$ is the Reynolds number based on the turbulent frictional velocity and the channel half height. The fully developed turbulent channel flow is homogeneous in the streamwise and spanwise directions [116], and periodic boundary conditions are used in these directions. The top and bottom boundaries ($z/\delta = \pm 1$) are subjected to non-slip and non-penetration boundary conditions. The initial condition is given by a uniform flow with disturbances generated by random number.

A case of ($Re_\tau = 380$) using a grid of ($200 \times 96 \times 240$) in the streamise (x), spanwise (y), and vertical (z) directions, respectively, is considered. Table 5.4 gives the numerical configurations for the case. The mesh sizes were chosen according to the mesh sizes in the wall-bonded turbulent flows [132] (e.g., the nearest point to the wall is located within $z^+ < 1$ in wall units (zu_τ/ν), and there are at least 10 points within $z^+ < 10$. Uniform meshes were used in the streamwise and spanwise directions while non-uniform meshes are used in the vertical direction with

$$z_k = \delta \frac{tanh(\theta_k)}{tanh(SF)} + \delta, \tag{5.13}$$

for θ_k

$$\theta_k = \frac{SF(-1 + 2((k-1) - 1)}{N_z}, \tag{5.14}$$

where $k = 1, 2, \ldots, N_z$. Here N_z is the number of grid points in the normal direction. SF is a stretch factor. Grid resolutions in wall units are presented in table 5.5, where the superscript $(+)$ represents wall units, i.e., scaling by the ratio of the friction velocity and the kinematic viscosity ($\frac{u_\tau}{\nu}$). Here, a value of $SF = 2.17$ is considered and results in the first mesh point away from the wall at $z^+ = 3.6 \times 10^{-1}$ wall units, and the maximum spacing (at the middle of the channel) is 6.95 wall units. Figure 5.13 shows the resulted variable grid resolution in the normal direction. The numerical results are presented in figures (5.14-5.18). The velocity and Reynolds stress results are normalized by (u_τ) and (u_τ^2), respectively. The numerical results are compared with direct numerical

Table 5.4: Simulation parameters for the channel numerical simulation

Re_τ	u_τ (m/s)	δ (m)	ν (m^2/s)	L_x/δ	L_y/δ	L_z/δ	N_x	N_y	N_z
380	0.2432	0.025	1.6×10^{-5}	4	1.96	2	200	96	240
395 [116]	-	-	-	6	3	2	256	192	193

simulation data described in Kim et. al [116, 117] where they used spectral method. The parameters of these data are also described in Tables 5.4 and 5.5.

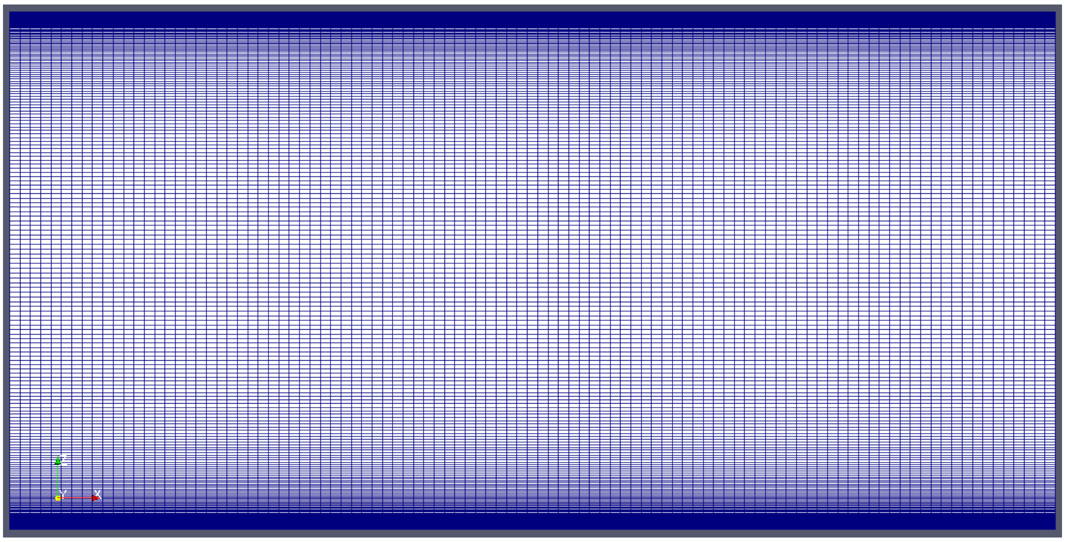


Figure 5.13: Non-uniform grid resolution in the vertical direction of the computational domain.

The averaged numerical results (over time and space) were collected after attaining stationary steady state conditions of the flow, where it exhibits no more change in the mean axial velocity and the vertical Reynolds stress. In the following sections the numerical results are presented as compared to the results of Kim et. al [116].

Table 5.5: Grid resolutions in wall units for the channel numerical simulation.

Re_τ	SF	Δx^+	Δy^+	Δz^+_{min}	Δz^+_{max}
380	2.17	7.5	7.5	3.6×10^{-1}	6.95
395 [116]	-	10.0	6.5	-	6.5

5.8.1 Mean Velocity Profile

The profile of the normalized mean streamwise velocity is shown in figure 5.14. The mean value of the other velocity components is zero in the entire domain [132, 116]. Typical velocity distribution in wall-bonded turbulent flow is obtained in Fig.5.14, where the mean velocity profile ($\bar{u}$ vs. z/δ) is shown.

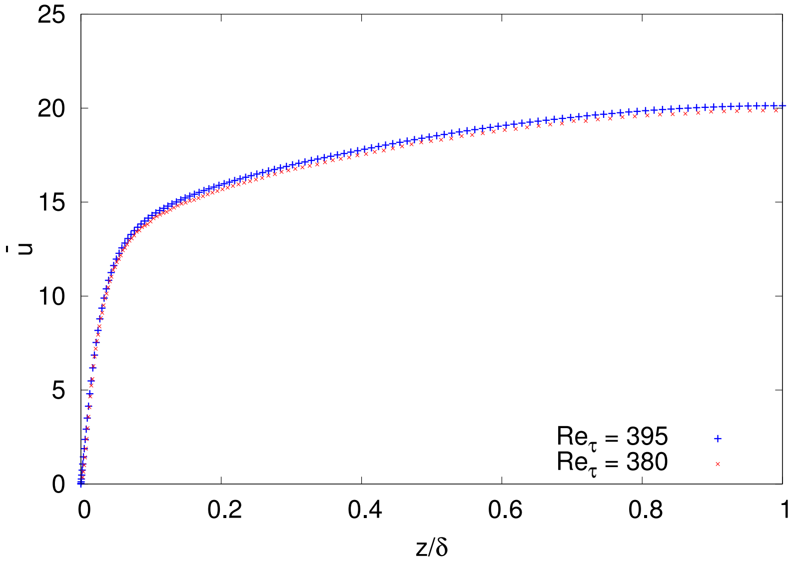


Figure 5.14: The mean of the normalized velocity profile in global coordinates..

5.8.2 Profile of RMS of Velocity

Another feature of the turbulent channel flow are the components of the Reynolds stress (turbulence intensities) describing the turbulent fluctuations. These are customarily presented in root mean square value $(u_{rms}, v_{rms}, w_{rms})$. Figures 5.15-5.17 show the distribution of the three components of the velocity plotted against z/δ. The peak value of (u_{rms}) coincides with that of the $(Re_\tau = 395)$ result. Away from the wall, the root mean square velocity distributions is expected to scale with the outer variables, rather than the wall units. As can be seen in figures 5.15-5.17.

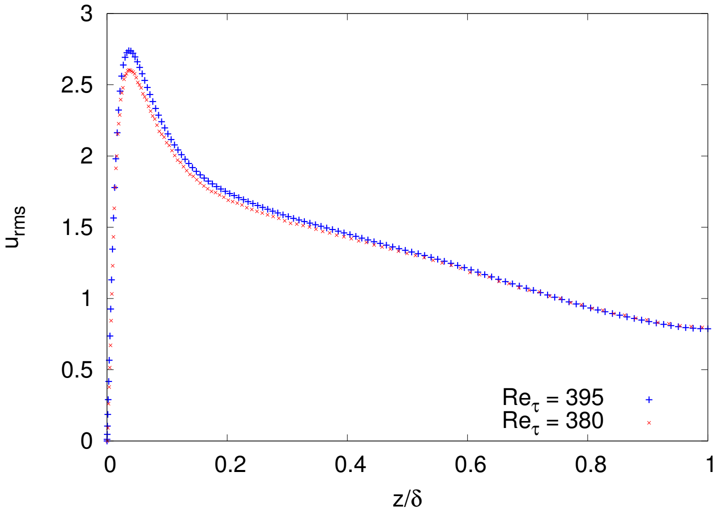


Figure 5.15: The root mean square of the normalized streamwise velocity, u_{rms}/u_τ.

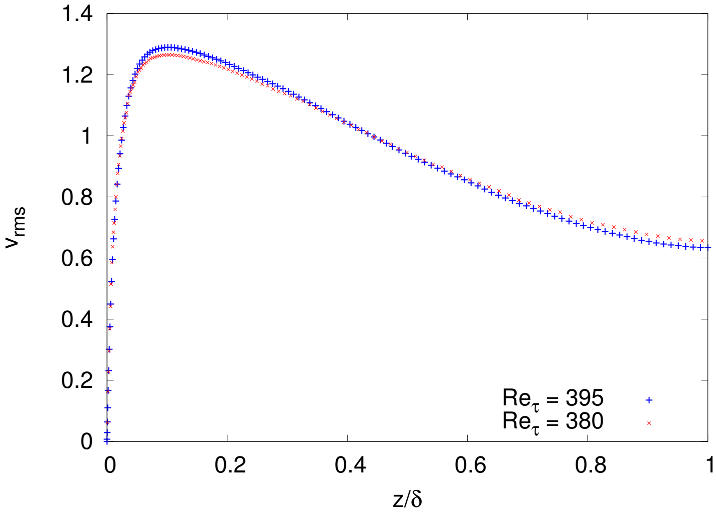


Figure 5.16: The root mean square of the normalized spanwise velocity, v_{rms}/u_τ.

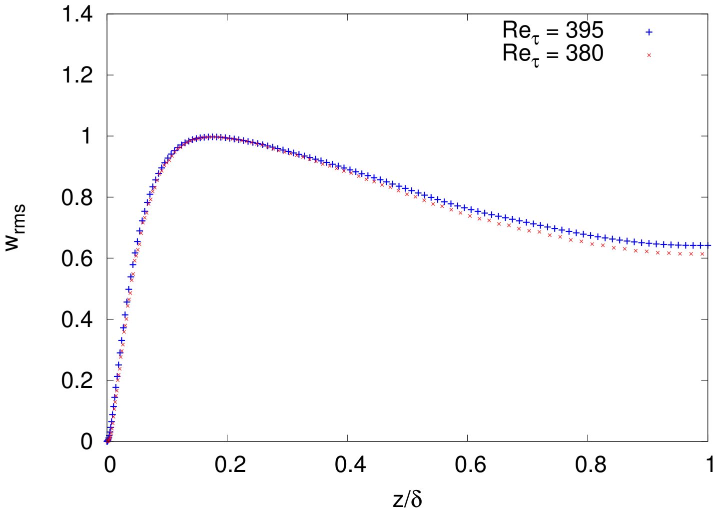


Figure 5.17: The root mean square of the normalized normal velocity, w_{rms}/u_τ.

5.8.3 Turbulent Shear Stress

The negative value of the xz component which is referred to as the turbulent shear stress is shown in fig.5.18. It is known that the total shear stress (τ_{tot}) (normalized by ρu_τ^2)

$$\tau_{tot} = -Re_{uv} + \frac{1}{Re_\tau}\frac{\partial u}{\partial z}, \tag{5.15}$$

varies linearly across the channel [132]. However, the viscous shear stress is only effective in the viscous sub-layer close to the wall. Therefore, the distribution of the turbulent shear stress can be anticipated to be linear in the outer region of the turbulent flow as can be seen in fig.5.18. This also shows that the numerical simulation has attained the statistically steady turbulent flow and the flow is fully developed [132]. The other components of the Reynolds stress tensor vary near zero.

In conclusion, in comparison to the results of Kim et. al [116], the presented numerical results are reasonable.

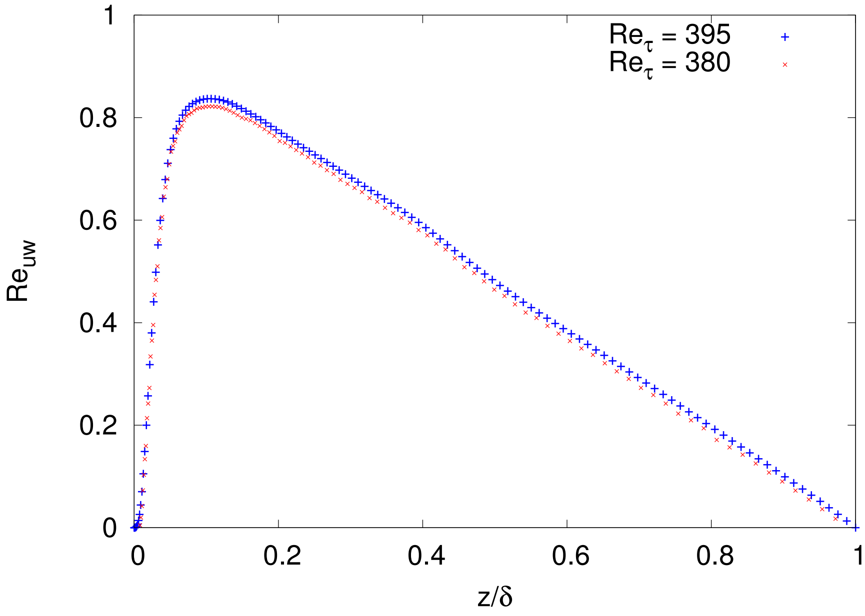


Figure 5.18: The profile of the normalized Reynolds stress, R_{uw}/u_τ^2

5.9 Summary

A parallel implementation of the three-dimensional code has been constructed using C++ and Open MPI. The three-dimensional solver of fluid flow and heat transfer can be executed on single node (serial) and on supercomputers of distributed memory architecture as well. This can be done by using the domain decomposition based on the required mesh size and the available number of nodes.

An evaluation of the parallel implementation was also conducted by large scale numerical simulations for two-phase flow on supercomputers with distributed memory architecture. A good speedup was obtained. The evaluation was carried out within the available resources of high performance computing up to 128 processors on ARCCA-Raven.

VSIAM3 was validated further through the equilibrium drop problem, single rising bubble problem, Kelvin-Helmholtz instability, and turbulent channel flow. Based on the conducted work the following remarks can be made

- In the equilibrium drop benchmark problem, the resulted velocity field shows very small spurious currents. The pressure solution is also acceptable in comparison with the exact solution.
- In comparison with the experimental observations, the numerical results of the single rising bubble are acceptable. This shows the ability of the VSIAM3 and CLSVOF numerical model in handling free surface motion up to high density and viscosity ratios in gas-liquid two-phase flows.
- The results of the numerical simulation of Kelvin-Helmholtz instability shows the ability of the numerical model for handling complex deformation of the gas-liquid interface. Therefore, it can be anticipated that the numerical model can be applied to complex gas-liquid flows that involve highly deformable interface.
- Based on the numerical results obtained for the turbulent channel flow, acceptable agreement with the results reported in the literature was obtained.

In conclusion, a robust numerical framework suitable for numerical simulation of two-phase gas-liquid flow has been constructed. The evident features of the numerical model of the CLSVOF which utilises multi-moments and CIP-CSL scheme has been shown. The later is one of the features of the multi-moment method, where an interpolation function of suitable accuracy can be constructed without the constraint of using longer stencil. This is a desirable feature in multi-phase numerical models, particularly, in complex flows that involve highly deformable interface such as Kelvin-Helmholtz instability. Furthermore, the extension of the numerical scheme of coupled Level-Set and THINC/WLIC to three dimensions is straightforward. The numerical schemes can deal with the complex free surface motion with surface tension. The employed surface tension model can handle surface tension computation well.

Chapter 6

VSIAM3 for Numerical Simulation of Heat Transfer Problems

6.1 Introduction

A two-phase Rayleigh-Bénard problem has been the subject of many theoretical, experimental, and numerical studies. One can mention the importance of interfacial convection in many important engineering applications such as microfluidics, material processing, and crystal growth. The effect of the natural convection with an interface is particularly significant under microgravity circumstances and on small scales [118]. Rayleigh-Benard Convection features different physical mechanisms and a variety of instabilities [55].

Deformation of the interface is particularly of great significance under the microgravity circumstances. The effect of a deformable interface is also essential in a rectangular cavity [24]. The computation of the finite-amplitude interfacial convection in the deformable interface situations is a challenging numerical problem because the interface layout is affected by the natural convective [119, 23].

When compared to the broadly examined single-layer cases, e.g., [32, 181], relatively few studies have been accounted for two-phase interfacial convection. Some numerical investigations of two-layer Rayleigh-Benard convection were carried out [[98, 211, 109, 134, 84]. However, the state of the interface was assumed to be rigid or flat in these studies.

In this chapter, we validate VSIAM3 for heat transfer problems further through single-phase and two-phase Rayleigh-Bénard convection.

6.2 Numerical Simulation of Rayleigh-Bénard Convection

Rayleigh-Bénard convection is a type of flow that is only driven by density gradient due to a temperature difference. The criterion for the Rayleigh-Bénard convection to occur in a volume of fluid subjected to temperature difference between its top and bottom boundaries is expressed in terms of the Rayleigh number, $Ra = \frac{\beta \Delta T g \delta^3}{\nu \alpha}$, which implies the ratio of the buoyancy and the viscous forces, Where ΔT is the temperature difference, β is the thermal expansion coefficient, g is the acceleration due to gravity, δ is the distance between the plates, ν is the kinematic viscosity, and α is the thermal diffusivity. Natural convection will commence due to density gradient, and the viscous forces oppose the resulted fluid motion.

There are many studies of this phenomenon. However, studies of two-phase Rayleigh-Bénard convection with a deformable interface are rare [22, 63]. The present simulation considers two horizontal layers of immiscible fluids confined between two isothermal walls kept at different temperatures. Figure 6.1 shows the computational domain. The fluids are assumed compressible with the Boussinesq approximation. Adiabatic boundary conditions are applied to the side walls. A non-slip velocity boundary conditions are applied to all boundaries. The domain has aspect ratio $A_H = \frac{L}{\delta} = 2.0$, and $A_W = \frac{W}{\delta} = \frac{1}{2}$ with L and W being the domain length and width, respectively. The important non-dimensional parameters of this problem defined based on the physical properties of the lower layer are the Rayleigh number, $Ra = \frac{\beta \Delta T g Pr}{\nu^2}$, the capillary number , $Ca = \frac{\rho \nu u_{ref}}{\sigma} = \frac{\rho \nu \sqrt{g \beta H \Delta T}}{\sigma}$, and $B = \beta \Delta T$, where $\Delta T = T_h - T_c$ is the temperature difference between the hot and the cold walls, and the Prandtl number is $Pr = \nu/\alpha$, where $\alpha = \kappa/\rho C_p$ is the thermal diffusivity.

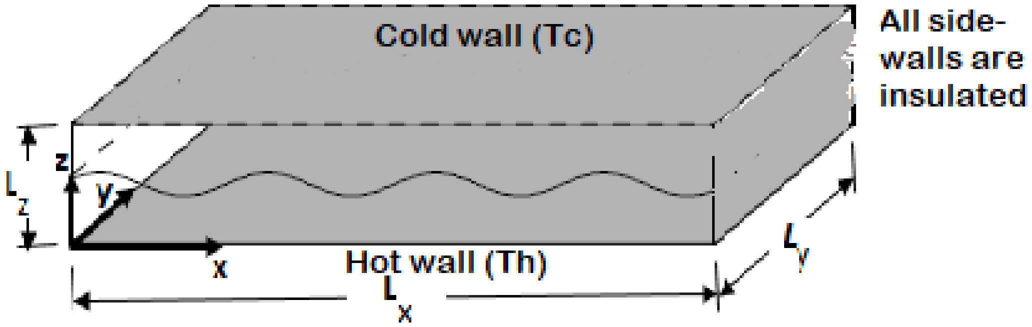


Figure 6.1: Schematic figure of Rayleigh Bénard Convection

The local Nusselt number is calculated as

$$Nu_x|_{z=z_0} = -\frac{\delta}{\Delta T}\frac{\partial T}{\partial z}, \tag{6.1}$$

and the average Nusselt number is defined as

$$Nu_{avg}|_{z=z_0} = -\frac{\delta}{L\Delta T}\int \frac{\partial T}{\partial z}dx. \tag{6.2}$$

6.3 Numerical Simulation of Single-Phase Rayleigh-Bénard Convection

In order to ensure that the heat transfer code is working properly in this test problem, a numerical simulation of single-phase Rayleigh-Bénard convection in a cavity is carried out and compared with the published literature [27, 135]. Here, we applied periodic boundary conditions for the velocity and temperature to the axial and transverse directions of the domain as in [27]. Non-slip velocity boundary conditions are applied to the top and bottom walls. Within the two-phase flow code, the physical properties of the fluids are set to result in a practically single phase flow problem. The dimensionless parameters are set as $B = 0.066$ and $Pr = 0.707$. A domain dimensions of $0.08.02 \times 0.04$ and grids of $160 \times 40 \times 80$ was used in the axial x, spanwise y and vertical z directions, respectively.

The initial temperature distribution is linear as depicted in Fig. 6.2 and is given by

$$T = T_c + \frac{\Delta T \,(1.0 - z)}{\delta}, \tag{6.3}$$

where T_c is the temperature of the top cold surface.

6.3.1 Numerical results

A comparison of the calculated average Nusselt numbers and those reported in the literature [27, 135] is given in table 6.1. A convergence study were conducted for the case of $Ra = 10000.0$. The calculated average Nusselt number for various mesh sizes are shown in table 6.2. Using finer mesh size results in maximum change of 1.6% in the calculated average Nusselt number.

6.3.2 TEC Formula in Heat Transfer Problems

Figure 6.3 shows the steady state temperature distribution ($Ra = 10000.0$). We observed that, using TEC formula for the computation of the boundary value variables causes numerical oscillation in this simulation problem. The numerical oscillations are first generated in the boundary values and then affect the cell average values. On the other hand, solving the diffusion and conduction terms for all the moments enhances the stability and accuracy of the energy equation solver in the multi-moment method as shown in Fig.6.4. However, in terms of numerical oscillation, the velocity field distribution does not show sensitiveness to the use of TEC formula as can be concluded from figs. 6.5 and 6.6.

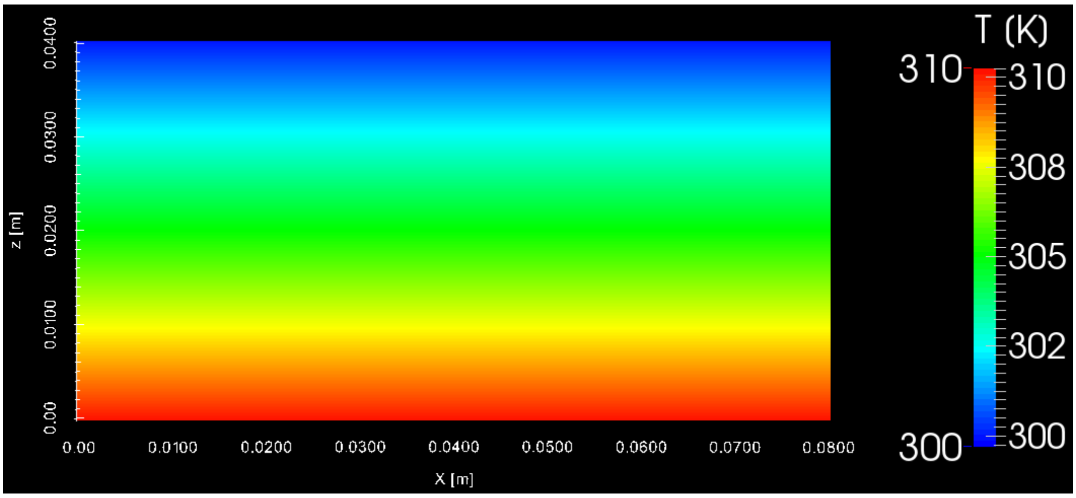


Figure 6.2: Rayleigh Bénard convection. Initial temperature distribution. Ra$= 10000$**, Pr**$= 0.707$**.**

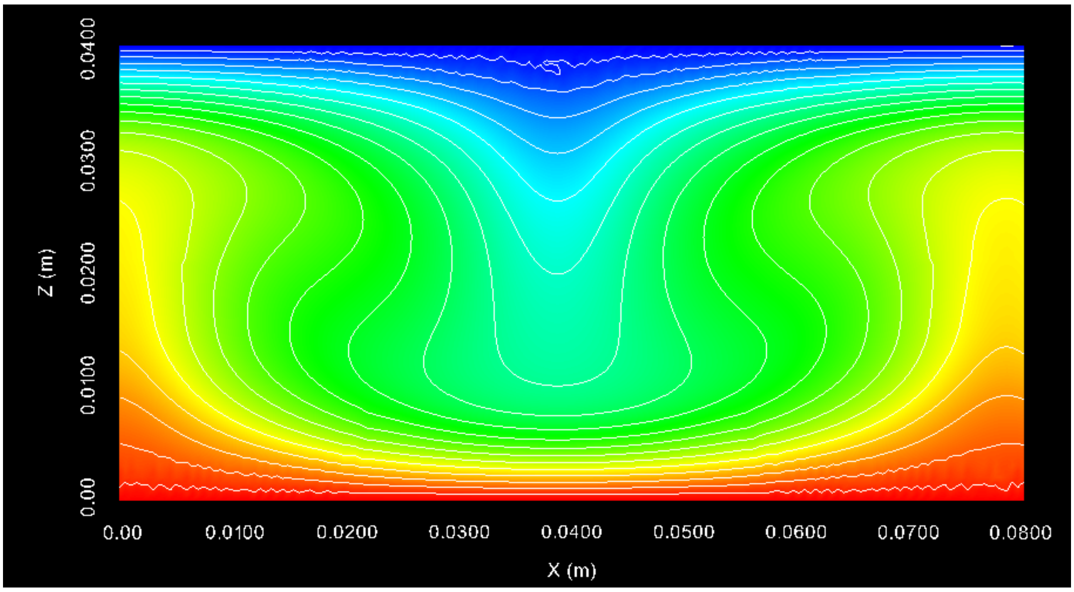


Figure 6.3: Temperature distribution for single-phase Rayleigh Bénard convection problem at t$= 60.0$ **sec. TEC is employed for the boundary values evolution. Ra**$= 10000$**, Pr**$= 0.707$**.**

Table 6.1: Comparison of calculated average Nusselt number with the literature..

Ra	10.0×10^3
	Nu_{av}
Clever et al. [27]	2.661
Prasianakis et al. [135]	2.644
Present simulation	2.656

Table 6.2: Convergence study of the average Nusselt number. Ra = 10000.0..

Mesh size	$80 \times 20 \times 40$	$160 \times 40 \times 80$	$200 \times 50 \times 100$
Nu_{av}	2.6864	2.6559	2.6432

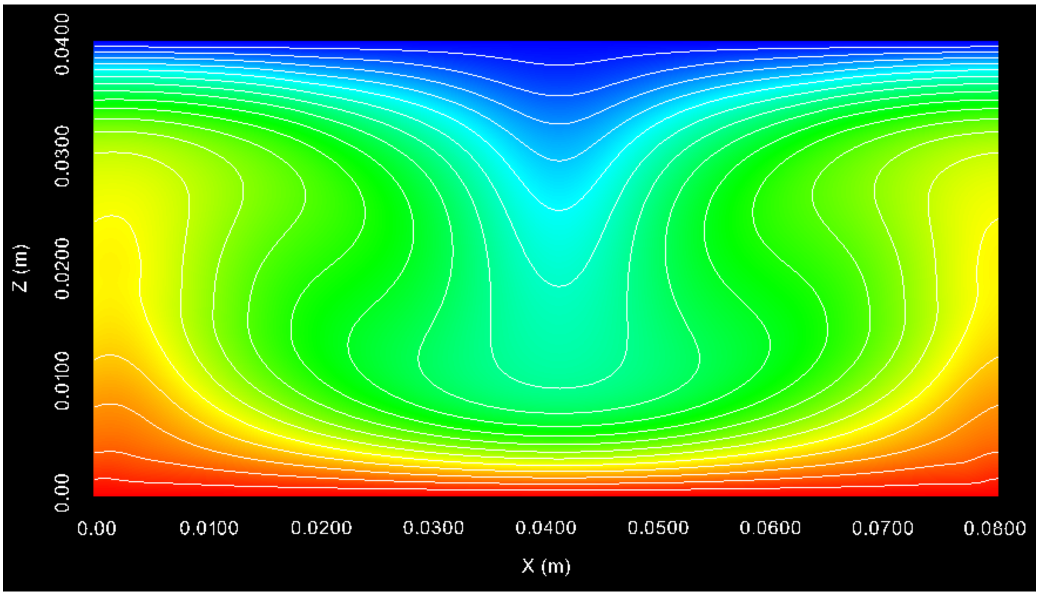


Figure 6.4: Temperature distribution for single-phase Rayleigh Bénard convection problem at t= 60.0 sec. TEC is abandoned for the boundary values evolution. Ra= 10000, Pr= 0.707.

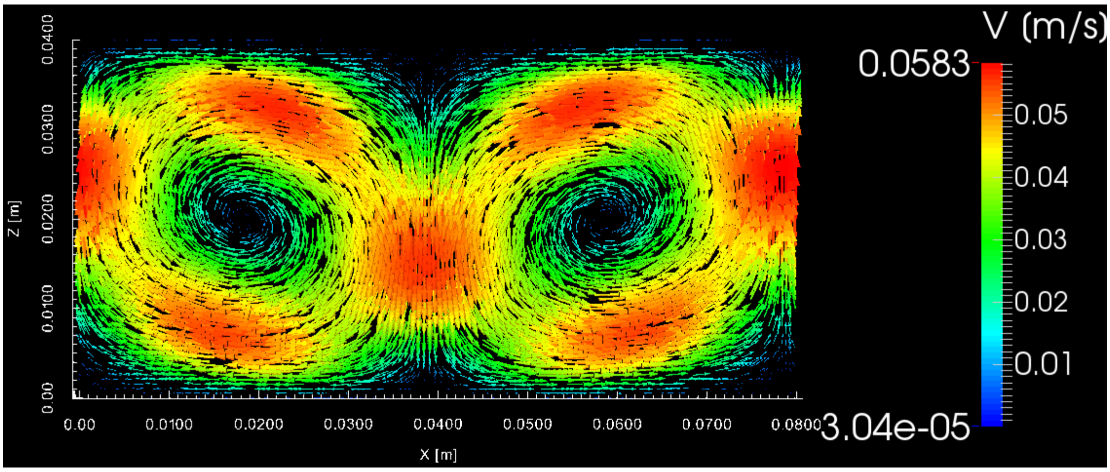


Figure 6.5: Velocity field for single-phase Rayleigh Bénard convection problem at t$=60.0$ **sec. TEC is employed for the boundary values evolution. Ra**$=10000$**, Pr**$=0.707$**.**

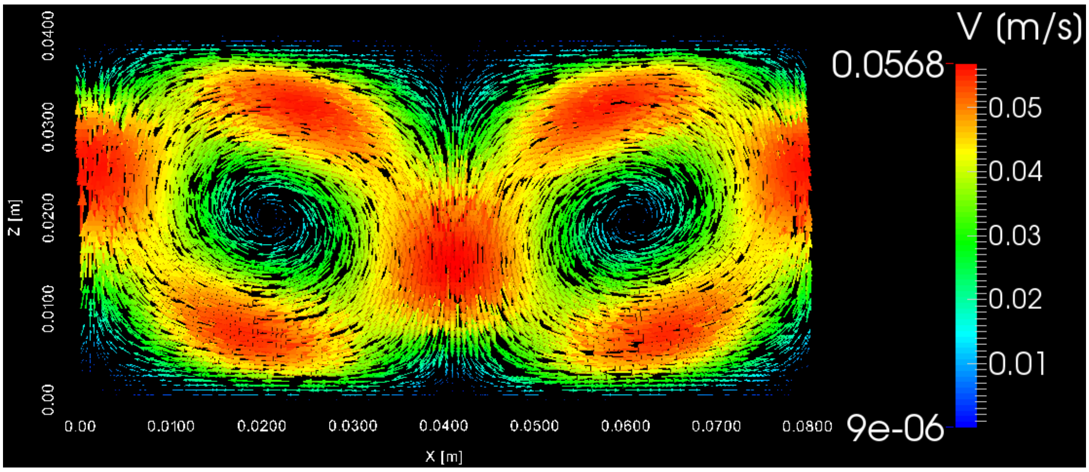


Figure 6.6: Velocity field for single-phase Rayleigh Bénard convection problem at t$=60.0$ **sec. TEC is abandoned for the boundary values evolution. Ra**$=10000$**, Pr**$=0.707$**.**

6.4 Simulation of Two-Phase Rayleigh-Bénard Convection

The non-dimensional parameters for the simulation of two-phase Rayleigh-Bénard convection are set as $Ra = 20000.0$, $Ca = 4 \times 10^{-4}$ $B = 0.066$, and $B = 0.1$. The ratio of the physical properties of the lighter to the heavier fluid are $Pr_r = 1.0$, $\rho_r = 0.33$, $\nu_r = 1.0$, $\beta_r = 2.0$, $\alpha_r = 1.0$. A domain dimensions of $0.04.01 \times 0.02$ and grids of $160 \times 40 \times 80$ was used in the axial x, spanwise y and vertical z directions, respectively. The initial temperature distribution is linear as shown in fig.6.7.

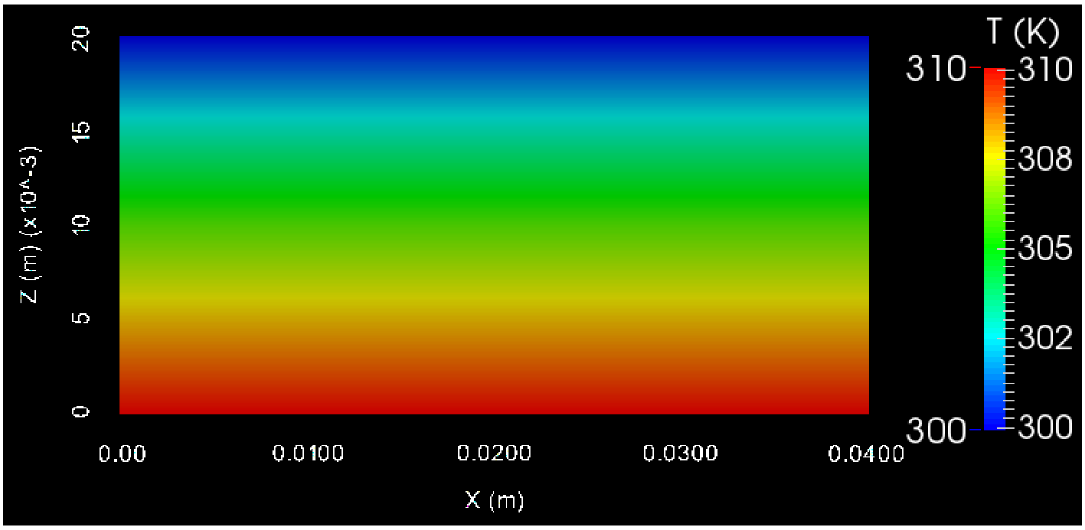


Figure 6.7: Two-phase Rayleigh Bénard convection. Initial temperature distribution. Ra$= 20000$**.**

6.4.1 Numerical results

The calculated average Nusselt number at the hot boundary and cold boundary are given in table 6.3 as compared to the results of Haghshenas et al. [63].

Figures 6.8 and 6.9 show the steady-state distribution of the temperature and velocity, respectively. A three-dimensional view of the numerical results of the flow and

Table 6.3: Comparison of calculated average Nusselt number with the literature..

Ra	20.0×10^3	
	$Nu_{av}^{hotwall}$	$Nu_{av}^{coldwall}$
Haghshenas et al. [63]	1.408	–
Present simulation	1.412	1.410

temperature fields are shown in figures 6.10 and 6.11. In fig. 6.9 the fluid motion in the lower layer is weaker than that in the upper layer. This is because the heavier fluid in the lower layer has higher density and lower thermal expansion coefficient than those of the lighter fluid, while the Prandtl number ratio and the viscosity ratio are one. The lower thermal expansion coefficient of the lower layer leads to a less buoyancy force and the higher density results in slower temperature change. Therefore weaker convection and fluid motion are expected. A slight upward bent can be noticed in the interface due to the use of low $B = 0.1$ value.

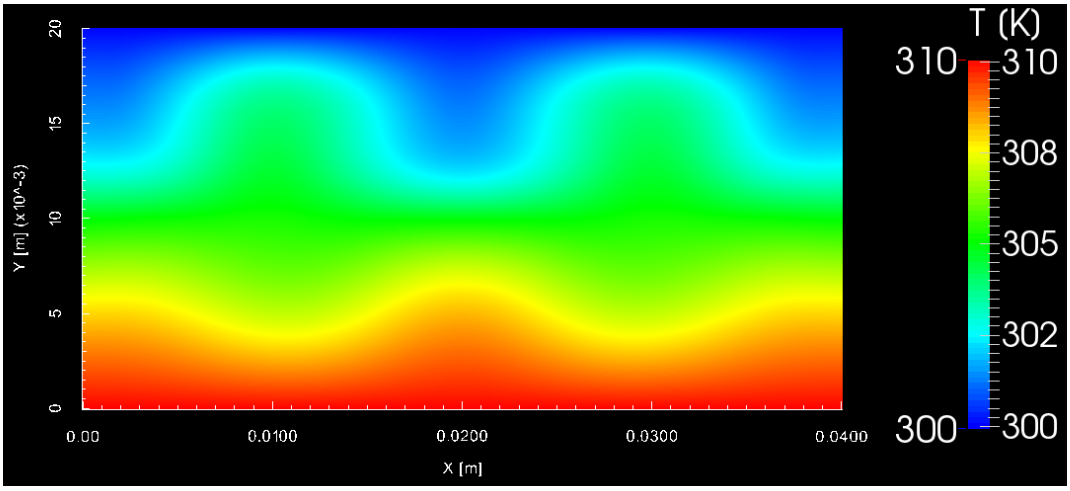


Figure 6.8: Two-phase Rayleigh Bénard convection. Steady state temperature distribution. Ra$= 20000$.

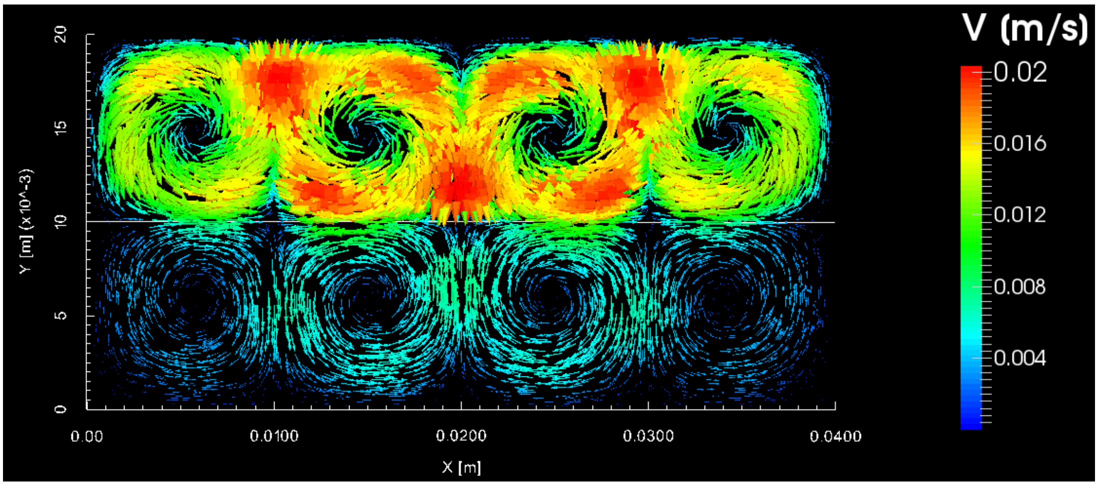


Figure 6.9: Two-phase Rayleigh Bénard convection. Steady state velocity distribution. Ra$=20000$**.**

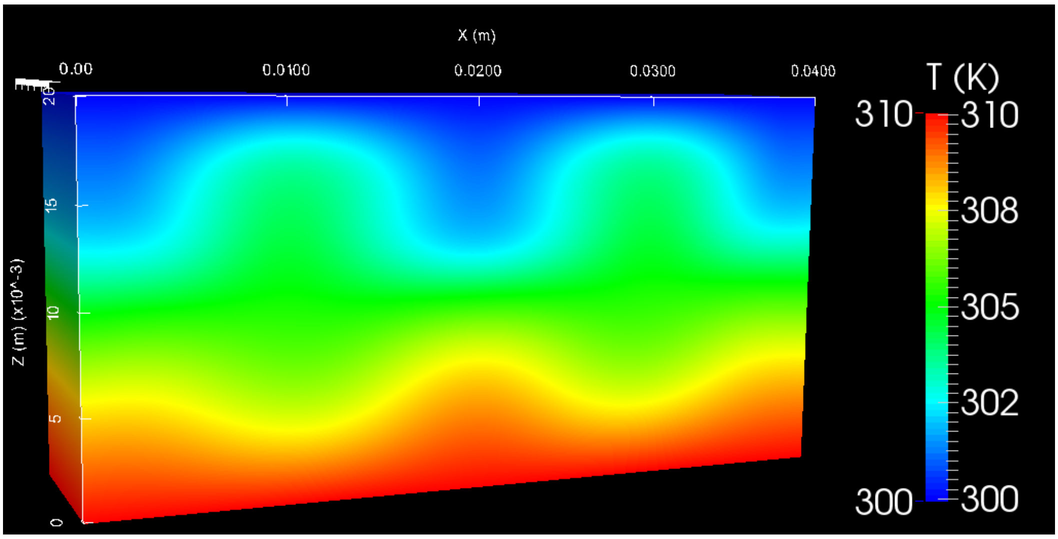


Figure 6.10: Rayleigh Bénard convection. 3D view of the steady state temperature distribution. Ra$=20000$**.**

6.5 Summary

We validated the VSIAM3 solver in the numerical simulation of heat transfer further through the Rayleigh-Bénard convection in single-phase and two-phase problems.

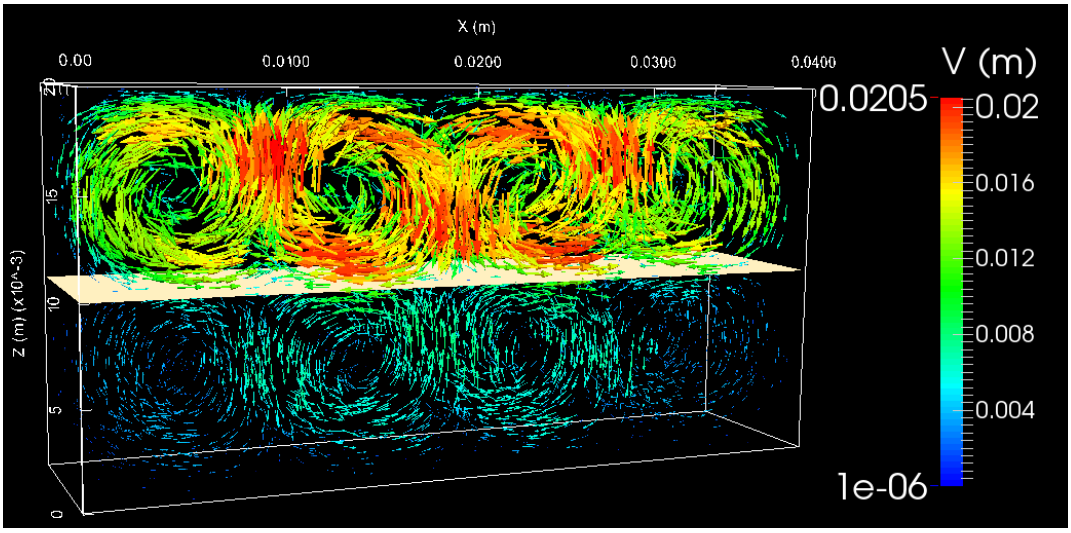


Figure 6.11: Rayleigh Bénard convection. 3D view of the steady state velocity distribution. Ra$= 20000$**.**

The fluids are assumed compressible with the Boussinesq approximation. First, a numerical simulation of single-phase Rayleigh Bénard convection was carried out to ensure the validity of the solver by comparing the obtained Nusselt number with the literature. Acceptable agreement was obtained. We observed that using TEC formula for the computation of the boundary values of the temperature affects the robustness of the energy equation solver and generates oscillations in the temperature field. These oscillations are generated first in the boundary values of the numerical model and then affect the cell average values. Therefore, we solve the diffusion/conduction terms for all the variables of the multi-moment method. Based on that, the resulted solver is robust and free of numerical oscillations. We also observed that solving the diffusion/conduction terms for all the variables does not affect the computational efficiency of the solver, and the computational time is almost the same as with using TEC formula.

A two-phase Rayleigh-Binard convection problem was then considered. Consulting the linear instability analysis of the problem [118], the solver produces the expected velocity fields in both domains, where the fluid motion is induced by the evolution of the temperature field. The resulted Nusselt number was compared with the result of a similar case in the literature [63]. The latter employed Lattice-Boltzmann method with a grid resolution about double the resolution used in the present simulation. Good agreement was obtained.

In conclusion, a robust VSIAM3 numerical model for heat transfer problems requires solving the diffusion/conduction terms for all the variables rather than employing TEC formula for the time evolution of the boundary values.

Chapter 7

Summary and Recommendations for Future Work

7.1 Summary

In summary, the following remarks can be pointed out :

- Gas-liquid two-phase flows are complex phenomena, and numerical simulation is an essential tool for studying their dynamics and underlying mechanisms. Efficient numerical schemes are indispensable for this purpose. In this regard, A C++ code library has been developed for the numerical simulation of three-dimensional gas-liquid two-phase flows and heat transfer. The code is written based on a framework of numerical methods namely; Volume/Surface Integrated Average-Based Multi-Moment Method (VSIAM3) including Constrained Interpolation Profile-Conservative semi-Lagrangian (CIPCSL) methods, Coupled-Level-Set-and-Volume-of-Fluid (CLSVOF), and density scaled CSF model. VSIAM3 is a numerical method for compressible and incompressible flows based on the multi-moment concept. VSIAM3 employs CIP-CSL schemes for solving the conservation equation. The CLSVOF is an interface capturing method that is well suited for two-phase flows with surface tension. The density scaled CSF model is used for the surface tension computation. The numerical schemes have been shown to be robust in simulating complex two-phase flows with surface tension.

- VSIAM3 and CIPCSL schemes are multi-moment methods. The multi-moment concept has risen difficulties in the implementation of the numerical methods. A comprehensive explanation of the numerical schemes has been presented in the present work which clarifies multi-dimensional execution of VSIAM3.
- Discretisation strategies of the conservation equation in VSIAM3 have been studied for an efficient implementation of VSIAM3 through the lid-driven cavity flow, shock tube problems, two-dimensional expulsion test, and droplet splashing. Discretisation techniques have been proposed for the velocity divergence term which result from the solution of the conservation equation using CIP-CSL schemes. It has been found that using a less oscillatory CIP-CSL scheme (i.e., CIP-CSLR, CIP-CSL3, etc.) with an appropriate formulation for the velocity divergence term is critically important for robust implementation of VSIAM3. As an optimal divergence term formulation, the mixed formulation has been suggested because it is highly robust for both incompressible and compressible flows problems.
- A parallel implementation of the code has been developed by using Open MPI for the three-dimensional numerical simulations of gas-liquid two-phase flows and heat transfer so that the code can be executed in serial and on supercomputers as well. The parallel performance was also evaluated, and good speedup has been obtained.
- VSIAM3 has been validated further through three-dimensional simulations such as equilibrium drop, single rising bubble, Kelvin-Helmholtz instability and turbulent channel flow. We obtained Reasonable results.
- The developed VSIAM3 solver has also been validated for heat transfer problems through Rayleigh-Benard convection. Two cases have been considered; single-phase and two-phase Rayleigh-Benard convection. It has been found that using TEC formula for computing the boundary values of VSIA3 affects the robustness of the numerical scheme. Therefore, a solution has been suggested by solving the diffusion/conduction terms for all the moments in VSIAM3 instead

of using TEC formula, which results in robust numerical simulations of single-phase and two-phase Rayleigh-Benard convection.

7.2 Recommendations for Future Work

Based on the conclusions drawn from the present work the following recommendations for future work can be suggested

- A more accurate CIP-CSL scheme can be constructed which takes into account the features of both the (CIP-CSL2) and the (CIP-CSLR), namely the accuracy of the former and the less oscillation featured by the latter while preserving the robustness of the numerical framework.
- The use of mixed moments (i.e., cell averages and boundary values) of VSIAM3 in the discretization of the diffusion and conduction terms for each moment can be investigated. We anticipate that this would more enhance the robustness and accuracy of the solver.
- The optimisation experiments for the parallel execution of the solver indicated that the pressure Poisson equation solver accounts for most of the computations time and exert limitation to further enhance the speed up of the incompressible flow solver. Thus, further numerical experiments are suggested, particularly, in pre-conditioners.
- In addition to the previous point, a significant enhancement is expected when using CUDA alongside OpenMPI, for both the pressure Poisson equation solver and the cross-nodes communications of various moments used in VSIAM3.
- Complex phenomena can be investigated using the solver such as heat transfer in wind-driven turbulence. An extensive study of Rayleigh-Benard convection with the deformable interface can also be carried out.

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. A unified level set approach to etching, deposition and lithography I: algorithms and two-dimensional simulation. *J. Comput. Phys.*, 120:128, 1995.

[2] D. Adalsteinsson and J. A. Sethian. A unified level set approach to etching, deposition and lithography III: complex simulations and multiple effects. *J. Comput. Phys.*, 138:193–223, 1997.

[3] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *J. Comp. Phys.*, 184(1):2–22, January 1999.

[4] A. W. Al-Khafaji and J. R. Tooley. *Numerical methods in engineering practice*. Holt, Rinehart & Winston, 1986.

[5] M. Al-Mosallam and K. Yokoi. Efficient implementation of volume/surface integrated average-based multi-moment method. *Int. J. of Comput. Methods*, 14(2):1–28, November 2017.

[6] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Corporation, 1984.

[7] T. Aoki. Interpolated differential operator (IDO) scheme for solving partial differential equations. *Computer Physics Communications*, 102(1-3):132–146, May 1997.

[8] N. Ashgriz and J. Y. Poo. FLAIR: Flux line-segment model for advection and interface reconstruction. *J. comput. phys.*, 93(2):449–468, 1991.

[9] C. Bailey, G. A. Taylor, M. Cross, and P. Chow. Discretisation procedures for multi-physics phenomena. *J. Comput. & App. Maths.*, 103:3–17, 1999.

[10] A. J. Baker. *Finite Element Computational Fluid Mechanics*. McGraw-Hill, New York, 1983.

[11] G. Baker. *Boundary Element Methods in Engineering and Sciences*, volume 4, chapter BIT for free surface flows. Imperial College Press, 2011.

[12] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. Petsc web page, 2016. http://www.mcs.anl.gov/petsc.

[13] E. A. Baskharone. *The finite element method with heat transfer and fluid mechanics applications*. Cambridge University Press, 2013.

[14] K.J. Bathe. *Finite Element Procedures*. K.J. Bathe and M.A. Watertown, second edition, 2014.

[15] Y. Bian, F. Dong, W.Zhang, H. Wang, and Z. Zhang C. Tan. 3D reconstruction of single rising bubble in water using digital image processing and characteristic matrix. *Particuology*, 11(2):170–183, 2013.

[16] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comp. Phys.*, 100(2):335–354, June 1992.

[17] J.C. Connor C. A. Brebbia. *Fundamentals of finite element techniques for structural engineers*. Butterworth, 1975.

[18] A. Caboussat. Numerical simulation of two-phase free surface flows. *Arch. Comput. Meth. Engng.*, 12(2):165–224, 2005.

[19] A. Caboussat. A numerical method for the simulation of free surface flows with surface tension. *Computers & Fluids*, 35(10):1205–1216, 2006.

[20] J. E. Castillo, J. M. Hyman, M. J. Shashkov, and S. Steinberg. The sensitivity and accuracy of fourth order finite-difference schemes on nonuniform grids in one dimension. *Computers and Mathematics with Applications*, 30(8):41–55, 1995.

[21] S. Chandrasekhar. *Hydrodynamic and Hydromagnetic Stability*. Dover Books on Physics. Dover Publications Inc., 1981.

[22] Q. Chang and J. I. D. Alexander. Application of the lattice Boltzmann method to two-phase Rayleigh-Benard convection with a deformable interface. *J. Comp. Phys.*, 212:473–489, 2006.

[23] C. F. Chen and J. M. Floryan. Numerical simulation of nonisothermal capillary interfaces. *J. Comp. Phys.*, 111:183–193, 1994.

[24] J.-C. Chen and F.-S Hwu. Oscillatory thermocapillary flow in a rectangular cavity. *Int. J. Heat Mass Transfer*, 36:3743–3749, 1993.

[25] S. Chen, B. Merriman, M. Kang, R. E. Caflisch, C. Ratsch, L.-T. Cheng, M. Gyure, R. P. Fedkiw, C. Anderson, and S. Osher. Level set method for thin film epitaxial growth. *J. Comput. Phys.*, 167:475–500, 2000.

[26] P. Chow, M. Cross, and K. Pericleous. A natural extension of the conventional finite volume method into polygonal unstructured meshes for CFD application. *Applied Mathematical Modelling*, 20:170–183, 1996.

[27] R. M. Clever and F. H. Busse. Transition to time-dependent convection. *J. Fluid Mech.*, 65(4):625–645, 1974.

[28] R. D. Cook. *Finite Element Modeling for Stress Analysis*. John Wiley & Sons, 1995.

[29] R. D. Cook. *Concepts and applications of finite element analysis*. Wiley, 2001.

[30] R. Courant, K. O. Friedrichs, and H. Lewy. Uber die partiellen di erenzengleichungen der mathematischen physik. *Math. Ann.*, 100:32–74, 1928. (English translation, with commentaries by P. B. Lax et al. in IBM J. Res. Develop. 11 (1967)).

[31] S. H. Crandall. *Engineering Analysis*. McGraw-Hill, New York, 1956.

[32] M.C. Cross and P.C. Hohenberg. Pattern formation outside of equilibrium. *Rev. Modern Phys.*, 65(3):851–1112, 1993.

[33] S. J. Cummins, M. M. Francois, and D. B. Kothe. Reconstructing distance functions from volume fractions: A better way to estimate interface topology? *Computers and Structures*, 38:425–434, 2005.

[34] J. DeCosmo, K. B. Katsaros, S. D. Smith, R. J. Anderson, W. A. Oost, K. Dumke, and H. Chadwick. Air-sea exchange of water vapor and sensible heat: the humidity exchange over the sea (HEXOS) results. *J. Geophys. Res. Oceans*, 101 (C5):12001–12016, 1996.

[35] N. G. Deen, M. V. S. Annaland, and J.A.M. Kuipers. Direct numerical simulation of heat transport in dispersed gas-liquid two-phase flow using front-tracking approach. Fifth Int. Conference on CFD in the Process Industries, 2006.

[36] N. G. Deen and J.A.M. Kuipers. Direct numerical simulation of wall-to-liquid heat transfer in dispersed gas-liquid rwo-phase flow using volume of fluid approach. *Chem. Eng. Sci.*, 102:268–282, 2013.

[37] H. Dong, Z. Qiao, S. Sun, and T. Tang. Adaptive moving grid methods for two-phase flow in porous media. *J. Comput. App. Maths.*, 265:139–150, 2014.

[38] P. G. Drazin. *Introduction to hydrodynamic stability*. Cambridge University Press, 2002.

[39] W. M. Drennan, J. A. Zhang, J. R. French, C. Mccormick, and P. G. Black. Turbulent fluxes in the hurricane boundary layer. part ii: Latent heat fluxes. *J. Atmos. Sci.*, 64(-):1103–1115, 2007.

[40] D.Sun and W. Tao. A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows. *Int.J. Heat and Mass Transfer*, 53(4):645–655, 2010.

[41] J. Du, B. Fix, J. Glimm, X. Jia, X. Li, Y. Li, and L. Wu. A simple package for front tracking. *J. Comp. Phys.*, 213:613–628, 2006.

[42] F. Durst, D. Milojevic, and B. SchSnung. Eulerian and Lagrangian predictions of particulate two-phase flows: a numerical study. *Appl. Math. Modelling*, 8:101–115, 1984.

[43] K. A. Emanuel. An air–sea interaction theory for tropical cyclones. part i: Steady-state maintenance. *J. Atmos. Sci.*, 43:585–604, • 1986.

[44] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *J. Comput. Phys.*, 183:83–116, 2002.

[45] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comp. Phys.*, 152(2):457–492, 1999.

[46] W. J. Ferguson and I. W. Turner. A control volume finite element numerical simulation of the drying of spruce. *J. Comput. Phys.*, 125:59–70, 1996.

[47] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer-Verlag, 2002.

[48] B. A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic, New York, 1972.

[49] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics*, volume 1. Springer-Verlag, 1991.

[50] C. A. Friehe and K. F. Schmitt. Parameterization of air-sea interface fluxes of sensible heat and moisture by the bulk aerodynamic formulas. *J. Phys. Oceanogr.*, 6(-):801–809, - 1976.

[51] M. Fulgosi, D. Lakehal, S. Banerjee, and G. Yadigaroglu. Direct numerical simulation of turbulence and interfacial dynamics in counter-current air-water flows. *Direct and Large-Eddy Simulation IV*, 8:443–452, - 2001.

[52] S. Gao and S. Zhai. Satellite air–sea enthalpy flux and intensity change of tropical cyclones over the western north pacific. *J. App. Meterio. Climat.*, 55:425–444, 2016.

[53] C. S. Garbe, B. Delille, C. W. Fairall, N. Gruber, J. Hare, D. T. Ho, M. T. Johnson, P. D. Nightingale, H. Pettersson, J. Piskozub, E. Sahlee, Wu ting Tsai, B. Ward, D. K. Woolf, and C. J. Zappa. *Transfer Across the Air-Sea Interface*, chapter 2. Springer Earth System Sciences. Ocean-Atmosphere Interactions of Gases and Particles. Springer, Berlin, Heidelberg, 2014.

[54] J. R. Garratt and P. Hyson. Vertical fluxes of momentum, sensible heat and water vapour during the air mass transformation experiment (AMTEX) 1974. *J. Met. Soc. Japan*, 53(-):149–160, - 1975.

[55] A . V. Getling. *Rayleigh Binard convection structures and dynamics*. World Scientific Publishing, 1998.

[56] U. Ghia, K. N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comp. Phys.*, 48(3):387–411, December 1982.

[57] J. Glimm, J. W. Grove, W. B. Linquist, O. McBryan, and G. Tryggvason. The bifurcation of tracked scalar waves. *SIAM J. Comput,*, 9:61–79, 1988.

[58] B. W. Green and F. Zhang. Impacts of air–sea flux parameterizations on the intensity and structure of tropical cyclones. *Mon. Weath. Rev.*, 141:3308–2324, 2013.

[59] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI Portable Parallel Programming with the Message-Passing Interface*. Scientific and Engineering Computation Series. Massachusetts Institute of Technology, 2014.

[60] H. Grzybowski and R. Mosdorf. Modelling of two-phase flow in a minichannel using level-set method. *Journal of Physics: Conference Series*, 530(1):012049, 2014.

[61] D. Gueyffie, J. Li, A Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *J. Comp. Phys.*, 152(2):423–456, July 1999.

[62] X. Guo and L. Shen. Numerical study of the effect of surface wave on turbulence underneath. part 2. Eulerian and Lagrangian properties of turbulence kinetic energy. *J. Fluid Mech.*, 744(-):250–272, - 2014.

[63] A. Haghshenas and M. H. Rahimian. Simulation of two-phase Rayleigh-Benard problem using lattice Boltzmann method. *IARJSET*, 2(10):1–6, 2015.

[64] H. Haj-Hariri, Q. Shi, and A. Borthan. Thermocapillary motion of deformable drops at finite Reynolds and Marangoni numbers. *Phys. Fluids*, 9(4):845–855, 1997.

[65] P. Hansbo. A free-Lagrange finite element method using space-time elements. *Comp. Meth. Appl. Mech. Engrg*, 188:347–361, 2000.

[66] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8(12):2182–2189, 1965.

[67] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *J. Comput. Phys.*, 71(2):231–303, 1987.

[68] A. Harten and S. Osher. Uniformly high order accurate non-oscillatory schemes. *SIAM J. Numer. Anal.*, 42(2):279–309, April 1987.

[69] M. Herrmann. Refined level set grid method for tracking interfaces. *CTR Ann. Res. Briefs*, pages 3–18, 2005.

[70] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(Research Paper 2379):1, December 1952.

[71] C. Hirsch. *Numerical computation of internal and external flows*, volume 1. Wiley New York, 1991.

[72] R. S. Hirsh. Higher order accurate difference solution of fluid mechanics problems by a compact differencing technique. *J. Comput. Phys.*, 9(1):90–109, 1975.

[73] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all speeds. *J. Comput. Phys.*, 14:227–253, 1974.

[74] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, 39(1):201–225, 1981.

[75] J. G. Hnat and J. D. Buckmaster. Spherical cap bubbles and skirt formation. *Phys. Fluids*, 19(2):182–194, 1976.

[76] R. X. Huang. *Ocean circulation wind-driven and thermohaline processes*. Cambridge University Press, 2010.

[77] T. J. R. Hugues, W. K. Liu, , and T. K. Zimmermann. Lagrangian Eulerian finite element formulation for incompressible viscous flows. *Comp. Meth. Appl. Mech. Engrg*, 29:329–349, 1981.

[78] J. M. Hyman. Numerical methods for tracking interfaces. *Physica D Nonlinear Phenomena*, pages 396–407, July 1984.

[79] S. R. Idelsohn, M. A. Storti, and E. Onate. Lagrangian formulations to solve free surface incompressible inviscid fluid flows. *Comput. Meth. Appl. Mech. Engrg*, 191:583–593, 2001.

[80] S. Ii and F. Xiao. High order multi-moment constrained finite volume method. part I: Basic formulation. *J. Comp. Phys.*, 228(10):3669–3707, June 2009.

[81] A. Ikebata, Y. Muraoka, and F. Xiao. Multiphase fluid simulations on a multiple GPGPU PC using unsplit time integration VSIAM3. *Progress in NUCLEAR SCIENCE and TECHNOLOGY*, 2:491–497, 2011.

[82] P.A. Jayantha and I.W. Turner. A second order control-volume finite-element least-squares strategy for simulating diffusion in strongly anisotropic media. *J. Comput. Maths*, 13(1):31–56, 2005.

[83] G-S. Jiang and C-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comp. Phys.*, 126(1):202–228, June 1996.

[84] Ch. Jing, T. Sato, and N. Imaishi. Rayleigh-Marangoni thermal instability in two-liquid layer system. *Microgravity Sci. Tech.*, 1, 1997.

[85] E. S. Johnson. Liquid encapsulated float zone melting for GaAs. *J. Crystal Growth*, 30:249–256, 1975.

[86] J. D. Anerson Jr. *Computational fluid dynamics the basics with applications*. McGraw-Hill Inc., 1995.

[87] A. D. Khawaji, I. K. Kutubkhanah, and J. M. Wie. sea water desalting by multiple distillations. *Desalination*, 221:47–69, 2008.

[88] J. Kim, R. Moin, and R. D. Moser. Turbulence statistics in fully developed channel flow at low Reynolds numbers. *J. Fluid Mech.*, 177:133–166, 1987.

[89] B. Kinsman. *Wind Waves: Their Generation and Propagation on the Ocean Surface*. Dover Earth Science, 1965.

[90] S. Komori, R. Kurose, K. Iwano, T. Ukai, and N. Suzuki. Direct numerical simulation of wind-driven turbulence and scalar transfer at sheared gas-liquid interfaces. *J. Turbul.*, 11(-):1–20, - 2010.

[91] S. Komori, R. Kurose, N. Takagaki, S. Ohtsubo, K. Iwano, K. Handa, and S. Shimada. *Sensible and latent heat transfer across the air-water interface wind-driven turbulence*, chapter Gas Transfer at Water Surfaces 2010, pages 78–89. Kyoto University Press, 2011.

[92] S. Komori, R. Nagaosa, Y. Murakami, S. Chiba, K. Ishii, and K. Kuwahara. Direct numerical simulation of three-dimensional open-channel flow with zero-shear gas-liquid interface. *Phys. Fluids*, 115(5):115–125, - 1993.

[93] E. Krevszig. *Advancved Engineering Mathematics*. ohn Wiley and Sons Ltd, 10th edition edition, 2011.

[94] V. Kudryavtsev, B. Chapron, and V. Makin. Impact of wind waves on the air-sea fluxes: A coupled model. *J. Geophys. Res. Oceans*, 119:1217–1236, 2014.

[95] C. Kunkelmann and P. Stephan. Modification and extension of a standard volume-of-fluid solver for simulating boiling heat transfer. V European Conference on Computational Fluid Dynamics, 2010.

[96] R. Kurose, N. Takagaki, A. Kimura, and S. Komori. Direct numerical simulation of turbulent heat transfer across a sheared wind-driven gas-liquid interface. *J. Fluid Mech.*, 804(-):646–687, - 2016.

[97] D. Lakehal, M. Fulgosi, G. Yadigaroglu, and S. Banerjee. Direct numerical simulation of turbulent heat transfer across a mobile, sheared gas-liquid interface. *J. Heat Transfer*, 125(6):1129–1139, - 2003.

[98] C.W. Lan, M.C. Liang, and M.K. Chen. Bifurcation and stability analyses for a two-phase Rayleigh-Benard problem in a cavity. *Phys. Fluids*, 10(6):1329–1343, 1998.

[99] G. W. Large and S. Bond. Sensible and latent heat flux measurements over the ocean. *J. Phys. Oceanogr.*, 12(-):464–481, - 1982.

[100] K. J. Laskey, E. S. Oran, and J. P. Boris. Approaches to resolving and tracking interfaces and discontinuties. Technical report, Naval Research Laboratory, July 1987.

[101] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Comm. Pure Appl. Math.*, 7(1):159–193, February 1954.

[102] S. Lee, I. K. Park, and J. J. Jeong. Modified CIP-CSL FV method for incompressible flows. *Computers & fluids*, 86(5):240–250, November 2013.

[103] B. P. Leonard. A stable and accurate convection modelling procedure based on quadratic upstream interpolation. *Copmput. Meth. Appl. Mech. Engrg.*, 19:59–98, 1979.

[104] J. Li. Calcul d'Interface Affine par Morceaux (piecewise linear interface calculation). *Acad. Sci. Paris*, 320:391, 1995.

[105] J. Li, Y. Renardy, and M. Renardy. Numerical simulation of breakup of a viscous drop in simple shear flow with a volume of fluid method. *Phys. Fluids*, 12:269–282, 2000.

[106] S. Li, B. Xie, and F. Xiao. An interface capturing method with a continuous function: The THINC method on unstructured triangular and tetrahedral meshes. *J. Comp. Phys*, 259:260–269, February 2014.

[107] F. Liu, I. W. Turner, and V. Anh. An unstructured mesh finite volume method for modelling saltwater intrusion into coastal aquifers. *Korean J. Comput. Maths*, 9(2):391–407, 2002.

[108] H. Liu, S. Krishnan, S. Marella, and H.S. Udaykumar. Sharp interface Cartesian grid method II: A technique for simulating droplet interactions with surfaces of arbitrary shape. *J. Comput. Phys.*, 210(1):32–54, November 2005.

[109] Q.-S. Liu, B.-H. Zhou, R. Liu, H. Nguyen-Thi, and B. Billia. Oscillatory instabilities of two-layer Rayleigh-Marangoni-Benard convection. *ActaAstronautica*, 59:40–45, 2006.

[110] X. D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200–212, November 1994.

[111] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers & Fluids*, 35(10):995–1010, 2006.

[112] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Trans. Graph.*, 23:457–462, 2004.

[113] Yimin Ma, N. E. Davidson, , Y. Xiao, and J-W. Bao. Revised parameterization of air–sea exchanges in high winds for operational numerical prediction: Impact on tropical cyclone track, intensity, and rapid intensification. *J. geophys. res.*, 122(16):8617–8630, 2017.

[114] T. Maxworthy and F. K. Browand. Experimernts in rotating and stratified flows with oceanographic application. *Ann. Rev. Fluid Mech.*, 7(1):273–305, 1975.

[115] V. Mihalef, D. Metaxas, and M. Sussman. Animation and control of breaking waves. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '04, pages 315–324, Goslar Germany, Germany, 2004. Eurographics Association.

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

[117] R. D. Moser, J. Kim, and N. N. Mansour. DNS data for turbulent channel flow, 1999. http://www.turbulence.ice.utexas.edu.

[118] A. Nepomnyashchy, J. C. Legros, and I. Simanovskii. *Interfacial convection in multilayer systems*. Springer Monographs in Mathematics. Springer, 2006.

[119] A. A. Nepomnyashchy and I. B. Simanovskii. Thermocapillary and thermogravitational convection in a two-layer system with a distorted interface. *Fluid Dynamics*, 19:494–499, 1984.

[120] NOAA. National oceanic and atmospheric administration web page, 2013. https://www.nesdis.noaa.gov.

[121] R. D. Noble and J. D. Way. *Liquid membrane technology: An overview*. ACS Symp. Series. American Chemical Society, 1987.

[122] W. L. Noh and P. Woodward. SLIC (Simple Line Interface Calculation). Lecture Notes in Physics, 1976.

[123] F. J. Ocampo-Torres, M. A. Donelan, N. Merzi, and F. Jia. Laboratory measurements of mass transfer of carbon dioxide and water vapour for smooth and rough flow conditions. *Tellus*, 46B(-):16–32, - 1994.

[124] N. Onodera. Private communication, 2015.

[125] N. Onodera, T. Aoki, and H. Kobayashi. Large-eddy simulation of turbulent channel flows with conservative IDO scheme. *J. Comput. Phys.*, 230(4):5787–5805, June 2011.

[126] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag New York, Inc., 2003.

[127] S. Osher and J. A. Sethian. Front propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulation. *J. Comput. Phys.*, 79(1):12–49, 1988.

[128] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, 1980.

[129] P. C. Patnaik, F. S. Sherman, and G. M. Corcos. A numerical simulation of Kelvin-Helmholtz waves of finite amplitude. *J. Fluid Mech.*, 73(2):215–240, 1976.

[130] R. Pedreros, G. Dardier, H. Dupuis, H. C. Graber, W. M. Drennan, A. Weill, and P. Nacass. Momentum and heat fluxes via the eddy correlation method on the R/V L'Atalante and an ASIS buoy. *J. Geophys. Res. Oceans*, 108 (C11)(-):3339, - 2003.

[131] D. Pepper, A. Kassab, and E. Divo. *Introduction to Finite Element, Boundary Element, and Meshless Methods*. ASME, 2014.

[132] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2015.

[133] S. Popinet and S. Zaleski. A front tracking algorithm for accurate representation of surface tension. *Int. J. Num. Meth. Fluids*, 30:775–793, 1999.

[134] A. Prakash and J. N. Koster. Steady Rayleigh-Benard convection in a two-layer system immiscible liquids. *J. Heat Transfer*, 118(2):366–373, 1996.

[135] N. I. Prasianakis and I. V. Karlin. Lattice Boltzmann method for simulation of compressible flows on standard lattices. *Physical Review E*, 78(1), 2008.

[136] A. Prosperetti and G. E Tryggvason, editors. *Computational methods for multuiphase flow*. Cambridge University Press, 2007.

[137] D. I. Pullin. Numerical studies of surface tension effects in nonlinear Kelvin-Helmholtz and Rayleigh-Taylor instability. *J. Fluid Mech.*, 119:507–532, 1982.

[138] M. Raessi, M. Bussmann, and J. Mostaghimi. A semi-implicit finite volume implementation of the CSF method for treating surface tension in interfacial flows. *Int. J. Numer. Methods Fluids*, 59(10):679–691, July 2008.

[139] G. D. Raithby. Skew upstream differencing schemes for problems involving fluid flow. *Copmput. Meth. Appl. Mech. Engrg.*, 9:153–164, 1976.

[140] D. H. Richter and D. P. Stern. Evidence of spray-mediated air–sea enthalpy flux within tropical cyclones. *Geophys. Res. Lett.*, 41(-):2997–3003, - 2014.

[141] W. J. Rider and D. B Kothe. Streching and tearing interface tracking methods. *AIAA*, 1717(95), 1995.

[142] W. J. Rider and D. B Kothe. Reconstructing volume tracking. *J. Comput. phys.*, 141(2):112–152, 1998.

[143] M. Rudman. Volume-tracking method for interfacial flow calculations. *Int. J. Numer. Methods Fluids*, 24(7):679–691, 1997.

[144] M. Sari, G. Gurarslan, and A. Zeytinoglu. High-order finite difference schemes for solving the advection-diffusion equation. *Mathematical and Computational Applications*, 15(3):449–460, 2010.

[145] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31(1):567–603, June 1999.

[146] R. Scardovelli and S. Zaleski. Interface reconstruction with least square fit and split Eulerian Lagrangian advection. *Int. J. Numerical Methods in Fluids*, 41(3):251–274, 2003.

[147] J. A. Sethian. *Level set methods and fast marching methods*. Cambridge University Press, 1999.

[148] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, 77(2):439–471, 1988.

[149] H. S. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Sminth. *Computational fluid dynamics with moving boundaries*. Taylor & Francis, 1996.

[150] P. Smereka and J. A. Sethian. Level set methods for fluid interfaces. *Ann. Rev. Fluid Mech.*, 35:341–372, 2003.

[151] S. D. Smith and E. G.Banke. Variation of the sea surface drag coefficient with wind speed. *Q. J. R. Meteorol. Soc.*, 101(429):665–673, 1975.

[152] G. A. Sod. A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws. *J. Comput. Phys.*, 27(1):1–31, April 1978.

[153] D. B. Spalding. A novel finite-difference formulation for differential expressions involving both first and second deeivatives. *Int. J. Numer. Methods Engrg*, 4:551–559, 1972.

[154] W. F. Spotz and G. F. Carey. High-order compact finite difference methods. In *Houston Journal of Mathematics*, pages 397–407. ICOSAHOM'95, University of Houston, 1996.

[155] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Siam, 2004.

[156] M. Sussman, A. S. Almgren, J. B. Bell, P P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flows. *J. Comput. Phys.*, 148(1):81–124, 1999.

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

[158] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162(2):301–337, 2000.

[159] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solution to incompressible two-phase flow. *J. Comput. Phys.*, 114(1):146–159, September 1994.

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

[161] N. Takagaki, R. Kurose, Y. Tsujimoto, S. Komori, and K. Takahashi. Effects of turbulent eddies and langmuir circulations on scalar transfer in a sheared wind-driven liquid flow. *Phys. Fluids*, 27(-):016603, - 2014.

[162] R. Tanaka, T. Nakamura, and T. Yabe. Constructing exactly conservative scheme in a non-conservative form. *Comput. Phys. Commun.*, 126(3):232–243, April 2000.

[163] H. Terashima and G. Tryggvason. A front-tracking/gohst-fluid method for fluid interfaces in compressible flows.

[164] F. Thomasset. *Implementation of Finite Element Methods for Navier-Stokes Equations*. Springer, Berlin, Heidelberg, 1981.

[165] V. Thomee. From finite difference to finite elements a short history of numerical analysis of partial differential equations. *J. Comput. App. Maths.*, 128(1):1–54, 2001.

[166] A. K. Tornberg. *Interface tracking with application to multiphase flows*. PhD thesis, Royal Institute of Technology, Stockholm, 2000.

[167] E. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[168] G. tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for the computation of multiphase flow. *J. Comp. Phys.*, 169(2):708–759, 2001.

[169] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011.

[170] W.-T. Tsai, S. M. Chen, G. H. Lu, and C. S. Garbe. Characteristics of interfacial signatures on a wind-driven gravity-capillary wave. *J. Geophys. Res. Oceans*, 118(-):1715–1735, - 2013.

[171] D. E. Turney. Coherent motions and time scales that control heat and mass transfer at wind-swept water surfaces. *J. Geophys. Res. Oceans*, 121:8731–8748, 2016.

[172] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100(1):25 – 37, May 1992.

[173] S.P. van der Pijl, A. Segal, and C. Vuik. A mass-conserving level-set (MCLS) method for modeling of multi-phase flows. Technical Report 03-03, Delft University of Technology, 2003.

[174] S.P. van der Pijl, A. Segal, C. Vuik, and P. Wesseling. A mass-conserving level-set method for modelling of multi-phase flow. *Int. J. Numer. Meth. Fluids*, 47:339–361, 2005.

[175] B. van Leer. Towards the ultimate conservative difference scheme. V. a second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(2):229–248, August 1997.

[176] F. Veron, W. K. Melville, and L. U. Clenian. Wave-coherent air-sea heat flux. *J. Phys. Oceanography*, 38:788–802, 2007.

[177] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics the finite volume method*. Longman Scientific and Technical, 1995.

[178] T. G. Vignesh and S. Bakshi. Noniterative interface reconstruction algorithms for volume of fluid methods. *Numerical methods in Fluids*, 73(1), 2013.

[179] P. R. Johnston W. Weaver. *Finite elements for structural analysis*. Prentice-Hall, 1984.

[180] Y. Wang, S. Simakhina, and M. Sussman. A hybrid level-set volume constrained method for incompressible two-phase flow. *J. Comp. Phys.*, 231(19):6438–6471, August 2012.

[181] T. Watanable. Flow pattern and heat transfer rate in Rayleigh-Benard convection. *Phys. Fluids*, 16:972, 2004.

[182] F. Xiao. Multi-fluid simulations using CIP finite volume method VSIAM3. *ECCOMAS*, pages 24–28, July 2004.

[183] F. Xiao. Unified formulation for compressible and incompressible flows by using multi-integrated moments I: One-dimensional inviscid compressible flow. *J. Comput. Phys.*, 195(2):629–654, April 2004.

[184] F. Xiao, R. Akoh, and S. Ii. Unified formulation for compressible and incompressible flows by using multi-integrated moments II: Multi-dimensional version for compressible and incompressible flows. *J. Comp. Phys.*, 213(1):3–56, March 2006.

[185] F. Xiao, Y. Honma, and T. Kono. A simple algebraic interface capturing scheme using hyperbolic tangent function. *Int. J. Numer. Meth. Fluid.*, 48(9):1023–1040, April 2005.

[186] F. Xiao, S. Ii, and C. Chen. Revisit to the THINC scheme: A simple algebraic VOF algorithm. *J. Comput. Phys.*, 230(19):7086–7092, 2011.

[187] F. Xiao and A. Ikebata. An effcient method for capturing free boundaries in multi-fluid simulations. *Int. J. Numer. Meth. Fluids*, 42:187–210, 2003.

[188] F. Xiao, A. Ikebata, and T. Hasegawa. Numerical simulations of free-interface fluids by a multi-integrated moment method. *Computers & Structures*, 83(6):409–423, February 2005.

[189] F. Xiao, X. D. Peng, and X. S. Shen. A finite-volume grid using multimoments for geostrophic adjustment. *Mon. Wea. Rev.*, 134(9):2515–2526, September 2006.

[190] F. Xiao and T. Yabe. Completely conservative and oscillationless semi-Lagrangian schemes for advection transportation. *J. Comput. Phys.*, 170(2):498–522, July 2001.

[191] F. Xiao, T. Yabe, and T. Ito. Constructing oscillation preventing scheme for advection equation by rational function. *Computer Physics Communications*, 93:1–12, 1996.

[192] F. Xiao, T. Yabe, X. Peng, and H. Kobayashi. Conservative and oscillation-less atmospheric transport schemes based on rational functions. *J. Geophys. Res.*, 107(D22):4609, November 2002.

[193] T. Yabe and T. Aoki. A universal solver for hyperbolic equations by cubic-polynomial interpolation I. One-dimensional solver. *Computer Physics Communications*, 66(2-3):219–232, 1991.

[194] T. Yabe, T. Ishikawa, P.Y. Wang, T. Akoi, Y. Kadota, and F. Ikeda. A universal solver for hyperbolic equations by cubic-polynomial interpolation ii. Two- and three-dimensional solvers. *Computer Physics Communications*, 66(2-3):233–242, September-October 1991.

[195] T. Yabe, R. Tanaka, T. Nakamura, and F. Xiao. An exactly conservative semi-Lagrangian scheme (CIP–CSL) in one dimension. *Mon. Wea. Rev.*, 129(2):332–344, February 2001.

[196] T. Yabe, F. Xiao, and T. Utsumi. Constrained interpolation profile method for multiphase analysis. *J. Comput. Phys.*, 169(2-3):556–593, 2001.

[197] X. Yang, A. J. James, J. Lowengrub, X. Zheng, and V. Cristini. An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *J. Comput.Phys.*, 217(2):364–394, 2006.

[198] A. L. Yarin. Drop impact dynamics: Splashing, spreading, receding, bouncing. *Annu. Rev. Fluid Mech.*, 38(4):159–192, 2006.

[199] G.-H. Yeoh and J. Tu. *Computational Techniques for Multiphase Flows*. Butterworth-Heinemann, 2009.

[200] K. Yokoi. Numerical method for complex moving boundary problems in a Cartesian fixed grid. *Phys. Rev. E.*, 65(5):055701(R), May 2002.

[201] K. Yokoi. Efficient implementation of THINC scheme: A simple and practical smoothed VOF algorithm. *J. Comp. Phys.*, 226(2):1985–2002, October 2007.

[202] K. Yokoi. A numerical method for free-surface flows and its application to droplet impact on a thin liquid layer. *J. Sci. Comput.*, 35(2):372–396, June 2008.

[203] K. Yokoi. Numerical studies of droplet splashing on a dry surface: triggering a splash with the dynamic contact angle. *Soft Matter*, 7(11):5120–5123, May 2011.

[204] K. Yokoi. A practical numerical framework for free surface flows based on CLS-VOF method, multi-moment methods and density-scaled CSF model: Numerical simulations of droplet splashing. *J. Comput. Phys.*, 232(1):252–271, January 2013.

[205] K. Yokoi. A density-scaled continuum surface force model within a balanced force formulation. *J. Comput. Phys.*, 278(1):221–228, December 2014.

[206] K. Yokoi, D. Vadillo, J. Hinch, and I. Hutchings. Numerical studies of the influence of the dynamic contact angle on a droplet impacting on a dry surface. *Phys. Fluids*, 21(7):072102, July 2009.

[207] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. *Numerical methods for fluid dynamics*, 1(1):41–51, 1982.

[208] C. J. Zappa, W. E. Asher, A. T. Jessup, J. Klinke, and S. R. Long. Microbreaking and the enhancement of air-water transfer velocity. *J. Geophys. Res.*, 109(C8):C08S16, - 2004.

[209] J. A. Zhang, P. G. Black, J. R. French, and W. M. Drennan. First direct measurements of enthalpy flux in the hurricane boundary layer: the CBLAST results. *Geophys. Res. Lett.*, 35(-):L14813, - 2008.

[210] J. A. Zhang, J. J. CIONE, E. A. Kalina, E. W. Uhlhorn, T. Hock, and J. A. Smith. Observations of infrared sea surface temperature and air–sea interaction in hurricane edouard (2014) using gps dropsondes. *Journal of Atmospheric and Oceanic Technology*, 34(6):1333–1349, 2017.

[211] B. Zhou, Q. Liu, and Z. Tang. Rayleigh-Marangoni-Benard instability in two-layer fluid system. *Acta Mechanica Sinica*, 20(4):366–373, 2004.

[212] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*, volume 1. Butterworth-Heinemann, fifth edition, 2000.

Appendices

Appendix

Sod's and Lax's Problems by the CIP-CSLR Method

Figure 1 shows numerical results of Sod and Lax problems by CIP-CSLR. CIP-CSLR does not include a slope limiter so that numerical oscillations around shock cannot be suppressed in VSIAM3.

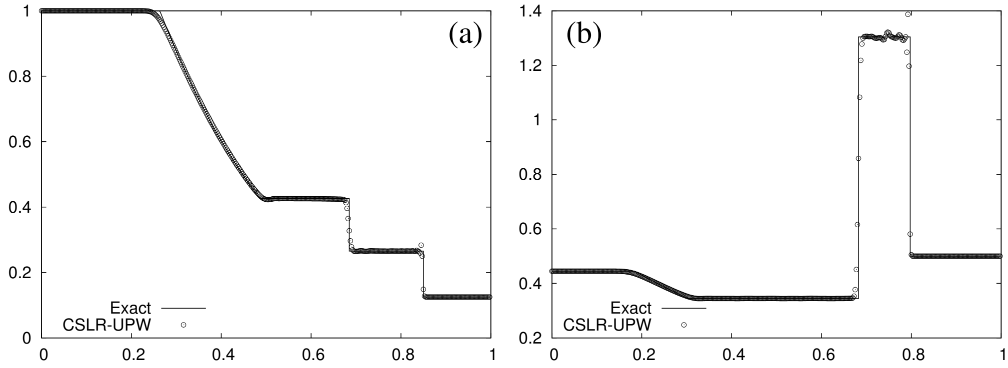


Figure 1: Numerical results of shock tube problems by CSLR-UPW. Plotted are density profiles vs distance, (a) Sod problem and (b) Lax problem. The dots show the density profile of numerical results. The line shows the exact solution..