VOF WITH CENTER OF MASS AND LAGRANGIAN PARTICLES (VCLP) - A SURFACE TRACKING AND ADVECTION METHOD FOR FLUIDS

A Dissertation

by

RICHARDS CHIZHUTHANICKEL SUNNY

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Juan J. Horrillo
Committee Members,	Marc Perlin
	Hamn-Ching Chen
	Paul M. Koola
Head of Department,	Sharath Girimaji

December 2021

Major Subject: Ocean Engineering

ABSTRACT

This dissertation presents a novel surface tracking, and advection algorithm for incompressible fluid flows in two and three dimensions. This method based on the volume-of-fluid (VOF) method, is named VOF-with-center-of-mass-and-Lagrangian-particles (VCLP), and it uses spatially and temporally localized Lagrangian particles (LPs) inside a finite volume framework. The fluid surface is recaptured and reconstructed piecewise using the mean slope, mean curvature, and fluid estimated using new methods from the local spatial distribution of the volume fluid fraction values. The reconstructed surfaces are either a finite plane or part of a spherical surface, in 3D and line segments or circular arcs, in 2D. The fluid mass inside each cell is discretized spatially by LPs and distributed as blue noise. LPs are then advected cell by cell with a choice of two different advection schemes in time using interpolated velocity and approximated acceleration fields. VCLP continuously tracks the center of mass of the fluid parcels in the Lagrangian way and this helps to reduce the errors due to numerical acceleration resulting from lack of information to reconstruct the interface accurately. LPs enable VCLP to work with structured and unstructured grids in two and three dimensions and might work for Courant–Friedrichs–Lewy numbers larger than one. LPs exist only inside a single fluid cell at a given time-step, allowing it to work without constraints on domain size and storage memory, unlike standard Lagrangian methods. LPs make it easy to adjust computational accuracy vs. speed by only changing the number of LPs. VCLP's performance is evaluated using standard benchmark tests such as translation, rotation, single vortex, deformation, and Zalesk's tests from the literature. VCLP is applied to TSUNAMI2D, a 2D Navier-Stokes model to simulate the dam-break problem and breaking waves.

DEDICATION

To my Lord Jesus Christ, Mother Mary, St. Joseph, my mother, father, wife, son, brother, teachers, and good friends.

ACKNOWLEDGMENTS

I want to express my sincerest thanks to my Ph.D. advisor Dr. Juan Horrillo for his constant guidance, support, and encouragement. It has been a privilege for me to be his student. I am very grateful to professors Marc Perlin, Hamn-Ching Chen, Paul Koola for serving on my committee, and professor Hermann Fritz for the help with the NSF project. I want to thank them for being very kind and supportive and for their constructive reviews, comments.

I am very grateful to all my teachers, friends, and family who gave me the strength and guidance for this journey that I started twenty-five years back. I thank my mother, who has been there every single time I needed her prayers and support. I want to thank my beloved wife and baby boy for being my greatest support and encouragement. I cannot thank my Dad and brother enough for their sacrifices and support. Many thanks to my friends, Johnson, Chinnu, and Emil, and my wife's parents for their love and support.

I want to express my profound gratitude to all my Professors, Sharath Girimaji, Rodriguez Ignacio Iturbe, Robert Randall, P. Krishnankutty, G. Idichandy, K. Murali, S. Surendran, for their time and guidance. Last but not least, I want to thank you NSF for the funding for my Ph.D. studies and Texas A&M for everything. Many thanks to Professor Hamn-Ching Chen, my MS advisor, and my friend Wei Cheng. I am also immensely grateful to Mrs. Maya Varyani and IITM alumni for the scholarships.

Above all, I thank my Lord Jesus Christ, Mother Mary, and St. Joseph for all the blessings.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

Presented work was completed under the guidance of the student's advisor, the chair of his Ph.D. thesis committee, Dr. Juan J. Horrillo. The other committee members, Dr. Marc Perlin, Dr. Paul Koola, and Dr. Hamn-Ching Chen of the Ocean engineering department, gave advice and corrections on the research work and thesis. The CFD code used for research, TSUNAMI2D, and TSUNAMI3D, was developed by Dr. Horrillo, and the numerical modeling of the submarine volcanic tsunami experiments using TSUNAMI3D was done with his help. The discussions with Dr. Horrillo on the research have provided valuable ideas that were explored and realized. He reviewed and corrected this thesis. The physical laboratory experiment on submarine volcanic tsunami was conducted by the research group led by Dr. Hermann Fritz and Dr. Yibin Liu. All the presented physical laboratory experimental results belong to them. Rest all work for the dissertation was completed by the student.

Funding Sources

Graduate research assistant position supported the doctoral research and studies under the supervision of Dr. Horrillo, funded by the National Science Foundation (NSF) project with contract CMMI-1563220.

NOMENCLATURE

2D	Two dimensional
ALE	Arbitrary Lagrangian-Eulerian method
ANSYSFluent	Commercial computational fluid dynamics software
BEM	Boundary element method
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy condition/number
CONVERGE	Commercial computational fluid dynamics software
CLSVOF	Coupled level-set VOF method
CM	center of mass
DES	Detached eddy simulation
DNS	Direct numerical simulation
FDM	Finite difference method
FLOW-3D	Commercial computational fluid dynamics software
FNFP-BEM	fully non-linear potential flow theory and a higher-order boundary element method
FVM	Finite volume method
H&N	Hirt and Nichols method
LBM	Lattice Boltzmann method
LES	Large-eddy-simulation
LHS	Left hand side
LP	Lagrangian particle
LPs	Lagrangian particles

LSM	Level-set method
LVOF	Lagrangian volume of fluid
MAC	Marker and cell method
MIC	Mean interface curvature
MoF	Moment of fluid method
NSF	National Science Foundation
N-S	Navier-Stokes
OpenFOAM	Commercial computational fluid dynamics software
PCIC	Piecewise-Circular-Interface-Calculation method
PIC	Particle in cell method
PLIC	Piecewise-Linear-Interface-Calculation
PPM	Piecewise Parabolic Method
RANS	Reynold's averaged Navier-Stokes model
R&K	Rider and Kothe method
RHS	Right hand side
SCO	Skip core optimization method
SPH	Smoothed particle hydrodynamics
SSC	Sorted surface constant method
STAR-CCM+	Commercial computational fluid dynamics software
TJSM	Proposed numerical advection scheme- Tracking journey in spiral method
TSUNAMI2D	a two-dimensional Navier-Stokes numerical model
TSUNAMI3D	a three-dimensional Navier-Stokes numerical model
VCLP	VOF with center of mass and Lagrangian particles method
VOF	Volume of fluid method
VTG	Volcanic tsunami generator

F	a function defined at every point in the computational domain. It is 1 when fluid is present, else 0
f	Fractional volume of fluid
$\mathbf{\Omega}$	Computational domain
ρ_w	density of water/heavier fluid
ρ_a	density of air/lighter fluid
ρ	density of the fluid in a single phase flow
μ_w	viscosity of water/heavier fluid
μ_a	viscosity of air/lighter fluid
μ	viscosity of the fluid in a single phase flow
t	time
$\boldsymbol{u}$	velocity vector
$\boldsymbol{a}$	acceleration vector
X	X axis in the Cartesian coordinate system
Y	Y axis in the Cartesian coordinate system
Z	Z axis in the Cartesian coordinate system
x	position vector component in X direction
y	position vector component in Y direction
z	position vector component in Z direction
u_x	velocity vector component in X direction
u_y	velocity vector component in Y direction
u_z	velocity vector component in Z direction
a_x	acceleration vector component in X direction
a_y	acceleration vector component in Y direction
a_z	acceleration vector component in Z direction
p	pressure

$\boldsymbol{g}$	body acceleration vector
$\boldsymbol{x}$	position vector
ν	kinematic viscosity
τ	stress tensor
I	Interface
u_n	normal component of the velocity vector
u_t	tangential component of the velocity vector
Θ	a scalar field function representing fraction volume of a cell open to fluid flow
$\boldsymbol{t}$	tangential vector
p_0	ambient pressure
σ	surface tension
κ	curvature
n_X	number of cells in the X direction
n_Y	number of cells in the Y direction
n_Z	number of cells in the Z direction
t_m	time at time step
m	m^{th} time step
Δt_m	duration of the m^{th} time step
i	cell index in X direction
j	cell index in Y direction
k	cell index in Z direction
s	space occupied by a cell
δx	cell dimension along X axis
δy	cell dimension along Y axis
δz	cell dimension along Z axis

v	volume of the fluid inside a cell
$\boldsymbol{V_m}$	Total volume of fluid in the domain at a time
$\boldsymbol{c}$	the center of mass vector
c_x	center of mass vector component in X direction
c_y	center of mass vector component in Y direction
c_z	center of mass component in Z direction
$\boldsymbol{U_{X,m}}$	Matrix of size $n_X \times n_Y \times n_Z$ with u_x values of cells at the m^{th} time step.
$\boldsymbol{U_{Y,m}}$	Matrix of size $n_X \times n_Y \times n_Z$ with u_y values of cells at the m^{th} time step.
$\boldsymbol{U_{Z,m}}$	Matrix of size $n_X \times n_Y \times n_Z$ with u_z values of cells at the m^{th} time step.
$\boldsymbol{U_m}$	denotes the three matrices $\boldsymbol{U_{X,m}}$, $\boldsymbol{U_{Y,m}}$, and $\boldsymbol{U_{Z,m}}$ of size $n_X \times n_Y \times n_Z$
$\boldsymbol{f_m}$	Matrix of size $n_X \times n_Y \times n_Z$ with f values of cells at the m^{th} time step.
$\boldsymbol{c_m}$	denotes the three matrices with the CM location components
M_w	mass of water in the cell
E	Empty Cell
S	Surface Cell
N_s	Inner Surface Cell
$\boldsymbol{N}$	Inner Surface Cell
N_m	Inner Mantle Cell
N_c	Inner Core Cell
C_T	Cell type value for classification
q	Ceiling function of Courant–Friedrichs–Lewy number
C_T*	Temporary Cell type value for classification
N	Number of Lagrangian Particles

$\boldsymbol{L}$	Reconstructed Planar interface with Piecewise-Linear-Interface-Calculation
$\boldsymbol{C}$	Reconstructed spherical interface with Piecewise-Circular-Interface-Calculation construction
$\boldsymbol{n}$	Interface normal
n_x	Interface normal vector component in X direction
n_y	Interface normal vector component in Y direction
n_z	Interface normal component in Z direction
n_z	Interface normal component in Z direction
∇_{nX}	Numerical gradient in X direction
S_x, S_y	interface normal estimation parameters
H_x, H_y	interface normal estimation parameters
F_x, F_y	interface normal estimation parameters
$\bar{\kappa}$	Mean interface curvature
a	Planar and surface constant
$\mathbf{\Lambda}$	The location coordinates of Lagrangian particles representing the fluid region within a cell
$\boldsymbol{c_r}$	Center of rotation
Ψ	Stream function
E_{geo}	Geometric Error
h_0	Water depth
H_0	Solitary wave height
η	Water elevation
g	Acceleration due to gravity
psi	Pounds per square inch
C_m	Crest height from numerical model
T_m	Trough height from numerical model

C_e	Crest height from the laboratory experiment
T_e	Trough height from the laboratory experiment
T_E	Percentage error in the estimation of creTrough height from the laboratory experiment

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Fluid flows

The states of matter, such as liquid, gas, and plasma, are examples of fluid, defined as a material with zero shear modulus and undergoes continuous deformations under applied shear stress. Two fluids can have the same state of matter and different phases as phase is defined as the region with uniform physical and chemical properties. This definition would mean that water and water vapor are different phases, and so is the case of water and oil, even though both are liquids. A phase is 'continuous' if the fluid exists within a continually connected region and 'disperse' if it occupies a disconnected region. Fluid flows consisting of more than one phase are called multiphase flows, and natural processes such as sea waves and sediment transport are examples of multiphase flows. Even though all fluid flows are compressible, they are classified into two compressible and incompressible flows for practical purposes. Flows with significant changes in fluid density are treated as compressible, and if the Mach number (which is the ratio of flow speed to the speed of sound) greater than 0.3, it is considered compressible as the density change would be more than 5% [8]. A flow in which the material density is nearly constant within an infinitesimal fluid volume moving with the flow velocity (fluid parcel) is referred as an incompressible or an isochoric flow. In most scenarios, liquid flows can be considered incompressible flows, while gas flows with high velocities, as in jet engines or rocket motors, need to be treated as compressible flows. There are miscible and immiscible flows within multiphase flows, and two phases are miscible if they mix in all proportions forming a solution. Immiscible phases in a multiphase flow would have an interface or a boundary separating them.

1.2 Multiphase flows

The presence of multiple phases makes it challenging to describe and quantify the nature of flow due to phase flows affecting each other. The unknown velocities of each phase at a point in space makes it difficult to determine the velocity distribution.

The forces involved in a multiphase flow can be classified into five: pressure, inertia, viscous, buoyancy, and surface tension. Pressure forces accelerate the fluid in the opposite direction of the pressure gradient acting on surface elements. The viscous forces come into play when there is a velocity gradient, and it attempts to make the flow uniform by reducing the differences in velocities. Inertia force acting on the volume retains the direction and magnitude of the fluid motion. Buoyancy force represents the net gravitational force in the presence of a non-uniform density. Surface tension is a localized force that tries to minimize the surface area of the interface and acts on a line, and its magnitude is inversely proportional to the local radius of curvature of the interface.

While modeling two-phase flows with a significant difference in molecular viscosity and density like sea waves where the density of the seawater phase is around 837 times more than that of the air phase, the dynamics of the heavier phase is independent of the presence of the lighter phase. In this situation, the presence of the lighter phase can be neglected, and then the interface is called a free surface or surface. This assumption is valid for modeling many liquid-gas flows such as ocean waves because of the significant difference in the magnitude of the pressure, inertia, viscous, and buoyancy forces due to the liquid and gas. This assumption is attractive to numerical modelers and widely used as this would significantly increase the computational speed. However, it also depends on the purpose of simulation in the first place. For example, to study the mixing of CO_2 in the sea through breaking waves is impossible without taking the dispersed gaseous phase, i.e., the air bubbles, into account; other examples can be found in [9, 10].

The Lagrangian and Eulerian specifications are two classical perspectives to look at fluid flows. In the Lagrangian approach, as an individual fluid parcel moves through space and time, the observer follows it [11, 12] and the position plot of the parcel gives the pathline. In the Eulerian approach, the focus is on spatial locations through which fluid flows with time, and the observer is watching the change happening inside the specific location over time. In computational fluid dynamics (CFD), Eulerian(Euler) simulations are carried out inside a fixed mesh, while Lagrangian(Lagrange) simulations are mesh-free and have nodes following the velocity field. Euler-

Lagrange and Euler-Euler are the two most chosen ways to model multiphase flows containing dispersed and continuous phases. Both of these approaches assume the immiscibility of the phases.

In the Euler-Lagrange method, pressure and velocity fields of the fluid phase are solved using the Navier-Stokes equation by considering it as a continuum, and the dynamics of the dispersed phase such as solid particles, bubbles, or droplets are solved by tracking them in the Lagrangian way [13]. In the Euler-Euler two-phase immiscible flow, both dispersed and continuous phases are treated as fluids and solved using the Navier-Stokes equations. This approach is signified by the volume-averaged mass conservation equation for each phase and requires tracking of interfaces [14]. The concept of volume fraction enables the interface tracking in this approach, and the simplified version of this approach is to treat each phase independently utilizing the homogeneous flow model concept. The discontinuity in density and viscosity at the interface of the phases are generally present for the immiscible liquid-liquid or liquid-gas flows, and the interface is critical as it is the gate of momentum and energy transfer. The interface leads the phase flows, and hence its shape and location are crucial components of the solution, making interface tracking a vital aspect in modeling multiphase fluid flows.

Figure 1.1 shows the different types of two-phase flows, which are transient, separated, and dispersed. In the transient flow, both phases act as a continuous and dispersed phase. In the separated flow, both phases are continuous, whereas in dispersed flow, one phase is continuous, and the other phase is dispersed. In the case of liquid-gas flows, if gas is the dispersed phase, it is called bubbles, else droplets. In general, the Eulerian approach would be more efficient for the separated flow. For the dispersed flow, bubbles and droplets could be tracked using the Lagrangian approach while continuous phase using the Eulerian approach. The transient two-phase flow is complicated and might need a coupled Eulerian-Lagrangian approach to model in the most efficient way. Two-phase immiscible flows are also called interfacial flow, and it is seen in many natural and industrial processes such as sloshing waves, oil extraction, and liquid atomization. In many applications, the numerical modeling of these flows is essential to understand their mechanics and dynamics.

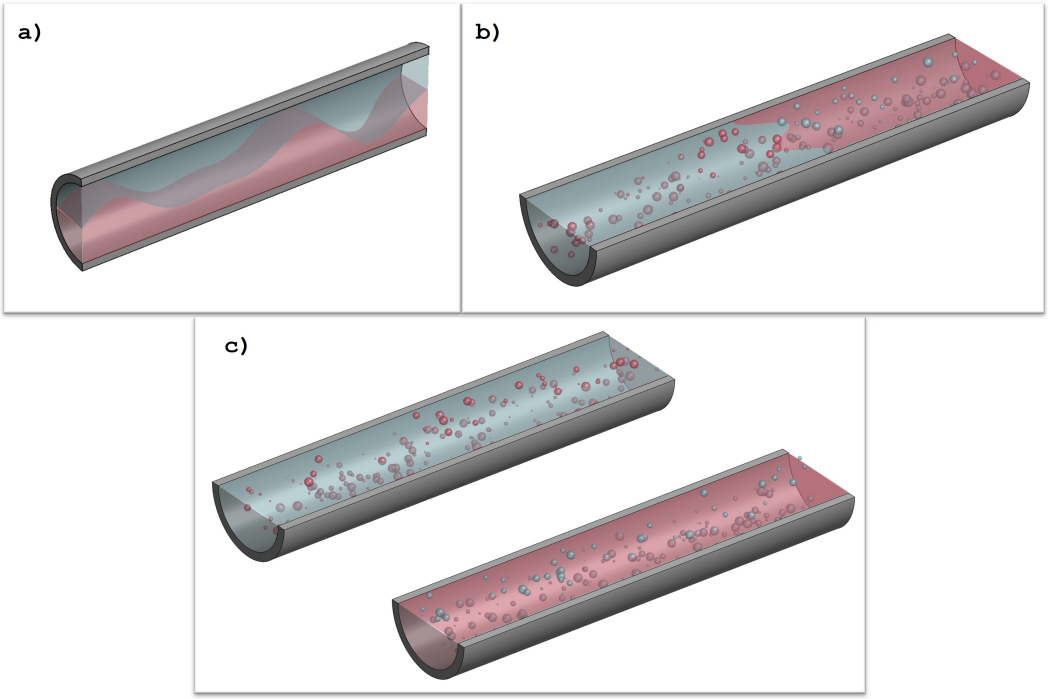


Figure 1.1: Different types of two-phase flows a) Separated flow b) Transient flow c) Dispersed flow - bubbles and droplets. Blue is the gas and red is the liquid

1.3 Computational Fluid Dynamics

To model and predict the fluid flows with complex geometry or complex flows is challenging without the help of Computational Fluid Dynamics (CFD). CFD was developed as a branch of fluid dynamics to understand better and predict fluid flows using numerical analysis and data structures. It uses high-speed computers to perform calculations based on the equations such as Navier-Stokes and continuity equations that govern the fluid flow to simulate the time-dependent free-stream flow of the fluid and its interactions with different kinds of structures or objects. The conservation laws of mass, linear momentum, and energy are the most critical equations in CFD. CFD has vast applications in engineering and research problems in aerodynamics, ocean engineering, weather simulation, combustion analysis, and environmental engineering. In general, CFD solutions have three stages: preprocessing, simulation, and post-processing. The preprocessing phase begins by defining the geometry and fluid distributions. The next step is constructing meshes through which

the fluid domain is discretized into cells or control volumes. The meshes can be structured, unstructured, hybrid, orthogonal, and non-orthogonal. The governing physics equations with applicable assumptions in the discretized form are modeled as the next step. The final step in pre-processing stage takes initial conditions and boundary conditions for the problem. Running the simulation is the second stage during which many equations are solved iteratively and march forward in time. The final stage is post-processing, where the resulting solution is analyzed and visualized. One of the most distinguishing features of CFD models is how numerically model the physics of the fluid flow. Popular discretization approaches in CFD are the finite volume method (FVM), finite element method (FEM), Finite difference method (FDM), and Boundary element method (BEM). In FVM, the computational domain is partitioned into fixed cells, and the partial differential equations representing the conservation laws are integrated within each cell representing control volumes to obtain a set of balance equations. The balance equations undergo discretization, resulting in discretized linear or non-linear equations that depend on a finite set of unknowns. If these systems of equations are linear, direct or iterative solvers are used, else fixed type methods are used. FVM has lower memory requirement, better speed, and works well with source-dominated and high Reynolds number flows [15]. FEM application areas of interest include heat transfer, structural analysis, fluid flow, and electromagnetic potential. Variation methods such as the Galerkin method characterize FEM. In FEM, the computational domain is divided into smaller overlapping parts called finite elements, over which local functions, generally polynomials, approximate the functions. FEM needs more computation time, and memory than FVM methods [16]. In FDM, derivatives in the partial differential equations are approximated by joining function values in the nearby locations using weights. The approximated equations defined at discretized points are solved using matrix algebra techniques. In BEM, partial differential equations are solved by formulating integral equations. BEM requires less computational resources than other methods when the surface to volume ratio is small [17]. One important area in CFD is the modeling turbulence in fluid flows and models like Reynold's-averaged-Navier-Stokes (RANS), Large-eddy-simulation (LES), Detached-eddy-simulation (DES), and Direct-numerical-simulation (DNS) are some of the most

successful ones.

1.4 Computational Grid

A computational grid is a CFD model's necessity, and the choice depends on the strengths and weaknesses of the model, requirements of the solution, and characteristics of the specific problem. The computational grid can generally be classified based on three criteria: the grid structure, its relation to boundary, and its degree of freedom. The grid or mesh divides a geometrical volume or area into many elements or cells, and they are used to construct control volumes. Corners of the cells are called nodes, faces are the boundaries of a cell, and edges are the boundaries of faces. Grids are classified into three based on the structure: structured, unstructured, and hybrid. Regular connectivity among cells identifies a structured grid, and its cells are hexahedra in 3D and quadrilateral in 2D. The storage arrangement determines the neighborhood relationships of a structured grid, and hence it has high space efficiency. The structured grid also has higher resolution, and better convergence [18]. An unstructured grid typically employs tetrahedral in 3D and triangles in 2D and is identified by irregular connectivity. An unstructured grid needs explicit storage for neighborhood relationships and can be highly space inefficient. A hybrid grid is made up of portions of a structured and an unstructured grid in an optimized manner. While regions around regular geometry may use a structured grid, the complex geometry regions would have an unstructured grid. Structured grids can be Cartesian or curvilinear. The Cartesian grid edges are parallel to the coordinate axes, while in the curvilinear grid system, coordinate surfaces are body fitted or curved to fit the boundaries [19]. If the fluid domain boundary cuts the computational cell, then the grid is an example for non-boundary fitted grid [20, 21, 22, 23]. Structured grids are also classified as orthogonal and non-orthogonal grids depending on the angle at which grid lines cross. Block-structured grids are another variation in which the domain is composed of a small number of regions, and the Chimera grid is an example [24]. As the flow gets complicated, as in the case of three dimensions, multiphase, dispersed or transient flow, generating boundary fitted grids is computationally expensive than the Cartesian grid. In contrast, if an optimized number of finer resolution grids are required near complex flow patterns or geometry, it is more difficult in

the structured than the unstructured grid. The Cartesian grid system can have varying or stretched grid size along the x, y, or z-axis, but this allows only to create a rectangular cuboid region with the desired resolution, and as a consequence, the computational time is wasted [25]. Discretization of equations in the structured grids is lucid than the unstructured grid, locally refined according to the geometry and flow. When it comes to the free-surface flow or the interface flow, structured grid topology is more convenient given that the geometrical observations are the basis of most numerical models that have the interface advection capacity. However, methods like the front-tracking method use an unstructured grid for interface tracking, and other examples may include [26].

1.4.1 Interface Tracking and Advection

Interface in a multiphase flow is the boundary separating two immiscible phases. While the interfaces around the bubbles and droplets in the transient and dispersed phases are small in radius with surface tension being the dominating force, the interfaces of the separated flows have a larger radius of curvature, and phases are continuous without droplets or bubbles. The sizes of bubbles and droplets are generally smaller than the size of the grid cell, and hence for FVMs where only the integrated properties are used, it is not possible to model the interfaces around the bubbles and droplets present in the transient and dispersed flows, without extra efforts. When a small fluid element containing the interface in a separated flow is considered, the interface has one dimension less than the fluid element. For example, the interface of a small 3D fluid element is a 2D surface, and that of a 2D fluid element is a line segment. In numerical modeling of interfacial flows, a grid cell would represent the fluid element, and mathematical surfaces or curves approximate the interface inside the element. These interfaces are usually captured and reconstructed in the Eulerian methods and tracked in the Lagrangian method, which means that the information about the interface is not explicit in the Eulerian approach and is extracted or captured from the knowledge of the spatial distribution of the phases and reconstructed in piecewise manner inside grid cells containing the interface. In the Lagrangian approach, trackers or body-fitted mesh are distributed throughout the interface and follow it in time, and hence the shape and location of the interface are explicitly known or tracked.

The reconstructed or tracked interface is advected by the interface advection methods that calculate the movement and deformation of the interface according to the velocity field during the time step. The Lagrangian interface advection methods are tracking methods in which the individual particles' trajectory, momentum, and collision are calculated while the observer travels along with the particle. Thus, the interface tracking methods that use the Lagrangian approach have either particles or fitted mesh moving with the interface [27, 28, 29]. Eulerian interface advection methods are associated with calculating the changes happening to a collection of particles or fluid mass inside a fixed control volume. Figure 1.2 illustrate the two approaches. A cubical fluid element located at the origin with unit dimension in positive axes is placed in a velocity field given by Eqn. 1.1. (u_x, u_y, u_z) is the velocity components in X, Y, and Z directions, and the time step is $0.1s$. Now, since we are interested in the incompressible fluid flows, the divergence of the velocity field should be zero, and this can be verified as shown in Eqn. 1.2. Vorticity, defined as a pseudovector field that describes the rotational tendency near a point, as would be seen by an observer traveling along with the flow. Vorticity $\boldsymbol{\omega}$ is calculated by taking the curl of the velocity as given by Eqn. 1.3.

$$\begin{aligned} u_x &= -\,y + 2.0 \ \ m/s \\ u_y &= -\,z - 2.0 \ \ m/s \\ u_z &= 2x - 2.4 \ \ m/s \end{aligned} \tag{1.1}$$

$$\nabla \cdot \boldsymbol{U} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \tag{1.2}$$

$$\nabla \times \boldsymbol{U} = \left(\frac{\partial u_z}{\partial y} - \frac{\partial u_y}{\partial z}\right)\hat{i} + \left(\frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial x}\right)\hat{j} + \left(\frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}\right)\hat{k} \tag{1.3}$$

$$\boldsymbol{\omega} = \ 1\hat{i} - 2\hat{j} + 1\hat{k} \ \ Hz$$

The Eulerian control volume is fixed with no shape or size change, and fluid may continuously

enter and exit the control volume, whereas, in the Lagrangian approach, the fluid mass inside the control volume remains fixed, which means that the control volume move and changes in shape and size and move along with the fluid flow. It can be seen in the Fig. 1.2 that the control volume undergoes deformation and is rotated while maintaining the constant volume.

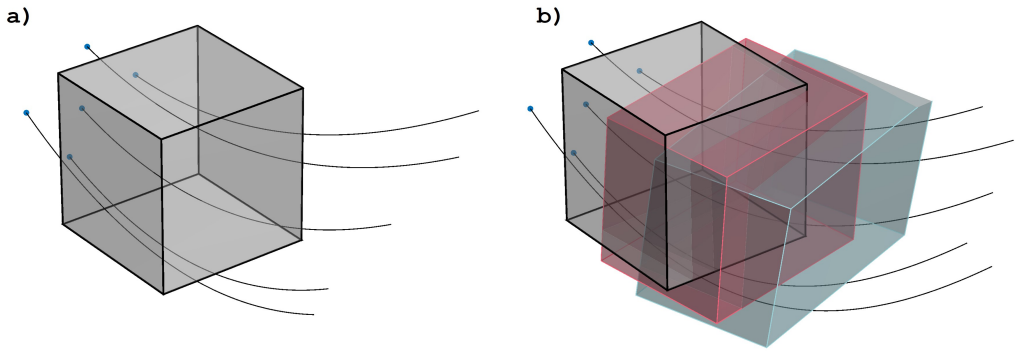


Figure 1.2: Control volume in the a) Eulerian and the b) Lagrangian approach in a accelerating non-divergent rotational velocity field given by eqn. 1.1. The curved lines are the stream lines and red color is the control volume after one time step and the blue after two time steps of advection.

The early work with the Lagrangian approach for interface tracking is found in the Marker and Cell method (MAC) [30]. In the MAC method, the velocity field models the fluid flow, while the markers represent the fluid in space, and fluid flow is simulated by moving the marker particles through space as per the interpolated velocities from the velocity field. Peskin [31] used the immersed boundary method to analyze the blood flow in the heart where the moving heart walls were treated as inner boundaries. Other Lagrangian methods use a deformable mesh that conforms to the interface at all times as in the front-tracking method [32] and Arbitrary Lagrangian Eulerian (ALE) methods [33, 34, 35]. In the front-tracking method, the flow field is discretized on a stationary grid using conservative finite difference approximation, while the interface is identified using an unstructured grid that can move through the stationary grid along with it. Now, since the interface undergoes deformation, as the calculations proceed, the unstructured grid needs remeshing. In the immersed boundary methods, the structure is represented on a Lagrangian system,

whereas the fluid is represented on an Eulerian grid system. In the Lagrangian approach, the markers, interface, or structure are parameterized to be defined completely and easily [36, 37, 38, 39]. The Lagrangian approach defines interfaces as internal boundaries, and boundary conditions are prescribed on them. This approach may not be suitable for fluid flows with large deformations, fragmenting interfaces, and fluid-structure interactions [40] because of the computational cost for frequent re-meshing and interpolation errors introduced while re-meshing. However, a Lagrangian grid is mostly two-dimensional, and it is possible to remove, add or reconnect the particles in case of large deformations [1], and it is used by Chen et al. [41] where fluid fronts can be converged using the sub-grid information. Figure 1.3 shows an example of Lagrangian tracking of the interface inside the Euler mesh.

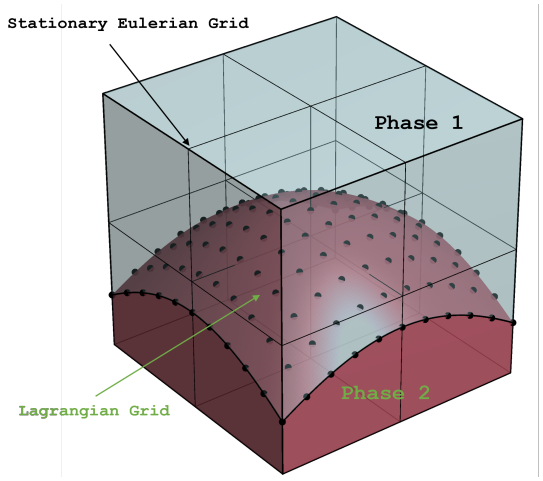


Figure 1.3: Control volume in the Lagrangian and Eulerian approach.

Smoothed-particle-hydrodynamics (SPH) is a mesh-free Lagrangian method introduced by Gingold and Monaghan [42], and Lucy [43], initially for astrophysical problems. Its increasing use in fluid simulation is concentrated in real-time animation and games where accuracy is not as critical as interactivity. Lack of mesh makes SPH ideally suited to simulate problems dominated by complex boundary dynamics like free surface flows and simplifies1 the model implementation and its parallelization, even for many-core architectures [44, 45]. The expense of grid-based simu-

lations per number of cells is substantially higher than the price of SPH simulations per number of particles. However, this is not advantageous for fluids like water due to the high-density [46] requiring a very high number of particles. SPH's main limitation includes the difficulty in setting up the boundary conditions and the need for many particles to produce simulations of equivalent resolution compared to Eulerian methods. Another Lagrangian method called Particle-in-cell (PIC) method, have discrete mass points representing the fluids, and it have been used since the 1960's [47]. Differential equations of mass, momentum characterize flow and energy in this method, and mass points move by weighted velocities. PIC method can follow interfaces and advect particles accurately because of its Lagrangian nature [48].

Lattice Boltzmann methods (LBM) use a very different approach from rest that, instead of solving the Navier-Stokes equation directly, it uses streaming and collision processes to simulate fluid density on a lattice. Even though this modern method can deal with complex multiphase flows, the number of iterations required to satisfy the stability conditions are many times more than the conventional finite volume methods, and not recommended for incompressible fluids.

The interface capturing methods are mostly used in the Eulerian framework, where the interface is not tracked explicitly, but it is hidden in the Eulerian data from a previous time step and can be captured and reconstructed. Color function is a capturing method where the interface may have a width of the order of the cell size. In the case of a two-phase flow, each phase is represented by a different color. If the grid cell is filled with one phase alone, the color function value for that cell is the color representing that phase. If there is an interface inside the cell, both phases will be present and the color function's value will be the resultant color got by mixing the two colors, according to the proportion of the volume filled by the corresponding phases. Since the relation that the total derivative of the color function with time is zero, it is discretized to solve for the advection. This approach is called the continuum advection method. The major drawback of this method is that the interface is diffused along with the time steps, increasing the interface width to multiple times the cell size. This problem could be solved by introducing a color function that varies smoothly in the fluid domain, and this is the idea behind the level-set method (LSM) [49, 50].

LSM uses -sets to analyze the surfaces and shapes and allows easier numerical computations with curves and surfaces on a fixed Cartesian grid without parameterization [51]. LSM is also efficient in tracking topology changes. In this method for any given point in domain, the level-set function provide its shortest distance from the interface. The interface with zero width at any time would be easily defined by setting the level-set function to zero. The normal direction to the interface is calculated from the gradient of the level-set function. Even though the advection using the level-set function is smooth and well defined, the original level-set method LSM cannot ensure volume conservation and shape and size conservation in an advection field. Attempts have been made to overcome this shortcoming by coupling with volume-of-fluid method (VOF) like the Coupled-Level-Set-VOF (CLSVOF), and others [52]. Eulerian and Lagrangian frameworks have been coupled previously to avoid shortcomings and to complement each other. Semi-implicit surface tension formulation with a Lagrangian surface mesh on an Eulerian simulation grid by Schroeder [53] and enhancement of pressure calculation by the incorporation of background mesh scheme by Wang [54] are such cases. A hybrid particle-level-set method proposed by Enright [55] called the Lagrangian volume of fluid (LVOF) improved the level-set method's mass conservation properties using the Lagrangian marker particles. When an interface is passively advected in a flow field, LVOF rebuilds the level-sets in the underresolved regions using marker particle, thereby producing better results [56, 57, 58]. LVOF couples Lagrangian level set [59] and the marker and cell method [60] with the continuum body force method [61] to compute surface tension. For interface reconstruction, an extension of the VOF method was introduced by Dyadechko & Shashkov [62] called the moment of fluid method (MoF) that keeps track of the cell-wise material centroids and uses this information to reconstruct interface by minimizing the defect of the first moment along with volume conservation constraint. This method allows the reconstructing of the interface with no information on neighboring cell fluid volume. Very recent work by Wenzel and Garrick [63] introduced the point-mass particle method, which is an Eulerian-Lagrangian approach where Navier-Stokes equations are solved in the Eulerian grid while Lagrangian particles discretize the fluid mass and phase information. Further review on interface tracking methods can be found

in [64, 65, 66, 67].

1.5 Volume of Fluid Method (VOF)

Among the interface capturing methods, the volume-of-fluid method is the most popular. Hirt and Nichols introduced the VOF method in 1979, a widely used FVM for tracking incompressible fluid interfaces [68, 69]. VOF's computational efficiency, straightforward treatment of changes in interface topology, and mass conservation make it a success [3]. VOF is characterized by a static or moving mesh and a discontinuous scalar function that gives the percentage of fluid present inside each mesh cell. VOF is faster and has lower memory requirements than other methods, and is used in the popular commercial CFD codes such as FLOW-3D, ANSYS Fluent, STAR-CCM+, OpenFOAM, and CONVERGE. VOF method introduces a new function 'F' that can vary in space and time. If the fluid occupies any given point in space at a given time, then the function 'F' takes the value of one, else zero. So theoretically, 'F' is analogous to marker particle-filled in the entire space. However, it is customary in the VOF method to use only one value per cell to define the fluid state. By the definition of the 'F' function, if we take the average value of 'F' inside a grid cell, represented as f, its value would always satisfy the relation $0 < f < 1$ and the average value would then represent the fractional volume of the cell that is filled with a specific fluid or phase. The value of one would mean that the cell is completely filled with the fluid, and zero would mean that the corresponding fluid is not present. A two-phase flow with air and water, 'f_w' value of 0.6 for a cell means 60% of the cell volume is filled with water and '40%' with air. Even though the F function carries the exact information about the interface, we know only f, which is the average value. Still, VOF can reconstruct the interface from the f values. The mean normal direction of the interface could be found out from the spatial gradient of the f function. Once this is known, a plane in three dimensions or a line in two dimensions is constructed such that the f value and normal direction are satisfied, which is the approximate boundary or interface. The evolution of the f function is given by $Df/Dt = 0$, which states that the F moves with the fluid. One of the main advantages of the VOF method is that it stores the minimum number of variables and follows the region instead of an interface, and avoids the problems associated with the intersecting and splitting surfaces.

In the original VOF advection, the piecewise constant scheme was used, i.e., the interface inside each cell is parallel to one of the axes. An improvement to this method can be seen in Nichols et. al. [70], Hirt and Nichols [69], Torrey et al. [71] and Duff [72]. Improvement is facilitated via an additional degree of freedom to form stair shaped profile within a cell, according to the local spatial distribution of fluid mass. Piecewise-linear schemes are an improvement in terms of accuracy to piecewise-constant schemes, even though these are more complex and computationally expensive. Since the fluid volume is conserved, the interface is located uniquely from interface normal alone. Interface normal can be estimated using many algorithms in the context of VOF, but all of them rely on the material volume data from the neighboring cells for interface reconstruction. Some notable piecewise linear schemes include Rider and Kothe [3], Harvie and Fletcher [73, 74], Geuyffier et al. [75], and Scardovelli and Zaleski [76, 4].

1.6 Outline

This dissertation presents a new interface tracking and advection method for incompressible and immiscible fluid flows through four subsequent chapters. The proposed method named VOF with Center of Mass and Lagrangian Particles (VCLP) couples Eulerian and Lagrangian approaches by using localized Lagrangian particles (LPs) inside the Volume of Fluid (VOF) framework. Chapter one provides the background for the interface tracking and advection methods and the review of the literature. The mathematical basis for the development of the proposed numerical method is presented in the second chapter. This chapter contains the continuity equation, the Navier-Stokes equation, interface equations, and boundary conditions. The mathematical framework of the VCLP method can be seen in the third chapter. It begins with the formulation of the problem and algorithm and then explains the different steps inside VCLP. These steps include the classification of grid cells, interface reconstruction, Lagrangian particles, the center of mass tracking, fluid advection, updated fluid distribution, and filters and optimization. The interface reconstruction section includes methods to find the interface normal, interface curvature, and locate the interface in two and three dimensions. The Lagrangian particles section explains its production, storage, and how it is used to represent the fluid flow in a localized spatial and temporal manner. It

also shows how the center of mass of the fluid is being tracked to improve accuracy. The advection of fluid section explains methods to interpolate the velocity and figure out the acceleration field. A new mathematical advection scheme used in VCLP is also presented in this section. Chapter four contains the numerical test performed to evaluate the performance of the proposed method. The numerical tests include translation, rotation, single vortex, deformation, and Zalesak's test in 2D, and 3D deformation test in 3D. The computational accuracy vs. speed of the method is presented in the next section. The final section covers the application of the VCLP method with TSUNAMI2D, a 2D Navier-Stokes model to simulate the dam break and breaking wave problems. The breaking wave simulation results are compared with the literature.

The fifth chapter presents the numerical modeling of volcanic tsunami experiments conducted using TSUNAMI3D, a fully 3D Navier-Stokes model. The final chapter summarizes and concludes the dissertation.

2. GOVERNING EQUATIONS

2.1 Introduction

In the mid-1800s, the French engineer and physicist Claude-Louis Navier and Anglo-Irish physicist and mathematician George Gabriel Stokes independently derived the most prominent and influential set of equations for fluid flows based on Euler's equation following Newton's second law, called by the name Navier-Stokes(N-S) equations. Euler's equation derived in the 17^{th} century, which is the backbone of N-S equations dealt with non-viscous fluids and subsequently are non-realistic for vast regimes of unsteady flows. N-S equations included the viscosity effects based on Newton's law of viscosity and opened the possibility to solve and realize the complicated phenomena such as vorticity and turbulence. Mass conservation equations and N-S equations, which are momentum conservation equations, and the energy conservation equations, form the foundation of fluid dynamic simulations. Even after two centuries, immense advancements in mathematics and physics, and many practical applications, N-S equations remain as one of the most difficult problems. However, scientists and engineers have been using it by reducing it into simpler forms, adding constraints, and using empirical relationships. With the exponential rise of the computational power, numerical calculations became faster and cheaper than ever. They hence opened the possibility to solve millions of unknown variables in discretized forms of N-S equations that would converge to the real solution as the resolution gets finer and closer to the Kolmogorov microscales [77]. CFD uses different methods such as FVM, finite difference method (FDM), finite element method (FEM), and spectral methods to discretize the fluid flow problem into smaller domains in terms of time, space, and frequency. CFD has tackled the N-S equations very efficiently with the computational resources as far as a single fluid is present. However, the multiphase fluid flows and single-phase flow with free-surface are challenging to solve efficiently. Complexity increases as a multiphase flow goes from a separated flow to dispersed flow, and from a dispersed flow to transient flow. The interface tracking and capturing techniques are used in these scenarios,

which help to apply boundary conditions and estimate the momentum and energy transfer between the flow phases. An introduction to interface tracking and advection method is given in the first chapter. The proposed surface tracking and advection method VCLP is based on the VOF method, an interface tracking and advection method. The interfaces are captured and reconstructed from the fluid distribution. The fluid volume bounded by the interfaces and boundaries are relocated to match the center of mass (CM) of the cells that is tracked via Lagrangian approach. The fluid cells are then advected using LPs representing the fluid mass.

2.2 Fluid Dynamics Equations

The governing equations and boundary conditions of an incompressible and immiscible fluid flow with a free surface are presented in this section. The corresponding equations are modified for stationary and solid moving objects. The fluid flows considered in this dissertation have the following conditions.

- The fluids are incompressible, and the bounded domain is given by $\mathbf{\Omega}$.
- The density and viscosity of the heavier phase and water is represented as ρ_w and μ_w respectively and are constants.
- The density and viscosity of the lighter phase and the air is represented as ρ_a and μ_a, respectively.
- When a single-phase flow is described, the density and viscosity of the fluid is represented as ρ and μ, respectively.

2.2.1 Equation of Continuity

In the absence of mass-energy conversion, the law stating "energy can neither be created nor be destroyed" is equivalent of "mass is neither created nor destroyed in a system." Mass conservation is the basis of the continuity equation in fluid dynamics which states that the rate at which mass enters a system is equal to the rate at which the mass leaves the system plus accumulation of mass within the system [78, 79]. Now, if we consider an infinitesimally small control volume within the

fluid flow, it satisfies the equation of continuity in the differential form represented as Eqn. 2.1 where t is the time and $\boldsymbol{u}$ is the velocity vector [78].

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0$$

or

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_x}{\partial x} + \frac{\partial \rho u_y}{\partial y} + \frac{\partial \rho u_z}{\partial z} = 0 \tag{2.1}$$

The accumulation or loss of fluid mass in time can be understood from the time derivative part, and this is zero throughout the space for a typical flow domain. The divergence term calculates the net inflow and outflow of fluid mass in the system. In general, liquids have very low compressibility and the variation in volume is minimal even for a high pressure difference. Liquid are generally considered incompressible for this reason. With the incompressibility assumption of the fluid, the continuity equation reduces to a volume conservation equation. Eqn. 2.2 gives the volume continuity equation stating that the divergence of velocity is zero throughout space for an incompressible fluid flow.

$$\nabla \cdot \boldsymbol{u} = 0$$

$$\frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \tag{2.2}$$

2.2.2 Navier-Stokes Equation

Newton's second law of motion states that the rate of change of momentum of an object is directly proportional to the force applied, and in the absence of an external force, the momentum is conserved. Navier-Stokes (N-S) equations are a set of non-linear partial differential equations that represents Newton's second law for fluid flows with the forces included. Eqn. 2.3 shows N-S equations in the convective form, which assumes constant dynamic and second viscosity value for the fluid.

$$\rho \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \right) = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \frac{1}{3}\mu \nabla (\nabla \cdot \boldsymbol{u}) + \rho \boldsymbol{g} \tag{2.3}$$

This equation is applicable for both compressible and incompressible flows, and with the assumption of incompressibility, it reduces to Eqn. 2.4.

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{g} \tag{2.4}$$

This equation assumes that the Cauchy stress tensor is Galilean invariant which means that it only depends on the spatial derivatives of the flow velocity and not directly on the fluid flow velocity. Moreover, the fluid is assumed to be isotropic. The divergence of the deviatoric stress is given by the second term in the RHS of the Eqn. 2.4. In the Eqn. 2.4, $\boldsymbol{u}(\boldsymbol{x}, t)$ is the velocity vector and is the function of position vector $\boldsymbol{x}$ and time. The scalar components of the velocity vector in the Cartesian coordinate system is represented as $\boldsymbol{u} = (u_x, u_y, u_z)$ and that of the position vector is given by $\boldsymbol{x} = (x, y, z)$. p is the pressure and a scalar. ν is the kinematic viscosity which is equal to $\frac{\mu}{\rho}$ where μ is the dynamic viscosity of the fluid. $\boldsymbol{g}$ is the body forcing or an acceleration vector. For example, in case of gravity $g_x = 0, g_y = 0$, and $g_z = -g$, where g is the acceleration due to gravity. In comparison with the Cauchy momentum equation, N-S equations can be understood as follows. The first term on LHS is the variation, the second term is the convective acceleration, and the combination is the total derivative of the velocity. Hence often, LHS is written using the material derivative as below.

$$\frac{D\boldsymbol{u}}{Dt} = \frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$$

The pressure term represents an internal source, and the acceleration term represents an external source or forcing. The viscosity term is the diffusive term. The equation 2.4 is in the non-conservative form and the second term on the LHS, $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ can be represented in the conservative form as below Eqn. 2.5

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) - \boldsymbol{u}(\nabla \cdot \boldsymbol{u}) = \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) \tag{2.5}$$

Applying continuity equation into Eqn. 2.5, the resulting term $\nabla \cdot (\boldsymbol{u}\boldsymbol{u})$ is the conservative form. The conservation argument holds as long as there are no external outflows and inflows in the fluid domain Ω. Nevertheless, there are instances where an external inflow or outflow exists, in which case, Eqn. 2.5 needs to be modified as below Eqn. 2.6

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) - \phi \boldsymbol{u} \tag{2.6}$$

Here, ϕ is an internal function, and it represents the volume that is created by a source object or taken away by a sink object in the control volume domain. It is more convenient to use the viscous term in the following manner given in Eqn. 2.7 since we are dealing with the Newtonian fluid. Here τ is the stress tensor, and $\nabla \boldsymbol{u}$ is the rate of strain.

$$\nu \nabla^2 \boldsymbol{u} = \frac{1}{\rho} \nabla \cdot (\mu \nabla \boldsymbol{u}) = \frac{1}{\rho} \nabla \cdot \tau \tag{2.7}$$

2.2.3 Equations of the Interface

Figure 2.1 shows a schematic representation of the computational domain for a two-phase flow with stationary and moving objects. The seafloor is treated as a rigid stationary object, and the ellipsoid represents a rigid moving object. The sea surface or the air-water interface can be seen. Because of the large density and viscosity of water compared to air, it is common to only simulate water by neglecting air, in which case the air-water interface is a free surface. The computational domain is a bounded box, and it will be divided into smaller control volumes in a typical FVM procedure. Every point on the interface moves along with the fluid, which would mean that if we ride a particle at the interface, its material derivative is zero. So, if the interface at anytime t is represented as scalar function $I(\boldsymbol{x}, t)$, then its material derivative is zero as given in Eqn.2.8.

$$\frac{DI}{Dt} = \frac{\partial I}{\partial t} + (\boldsymbol{u} \cdot \nabla) I = 0 \tag{2.8}$$

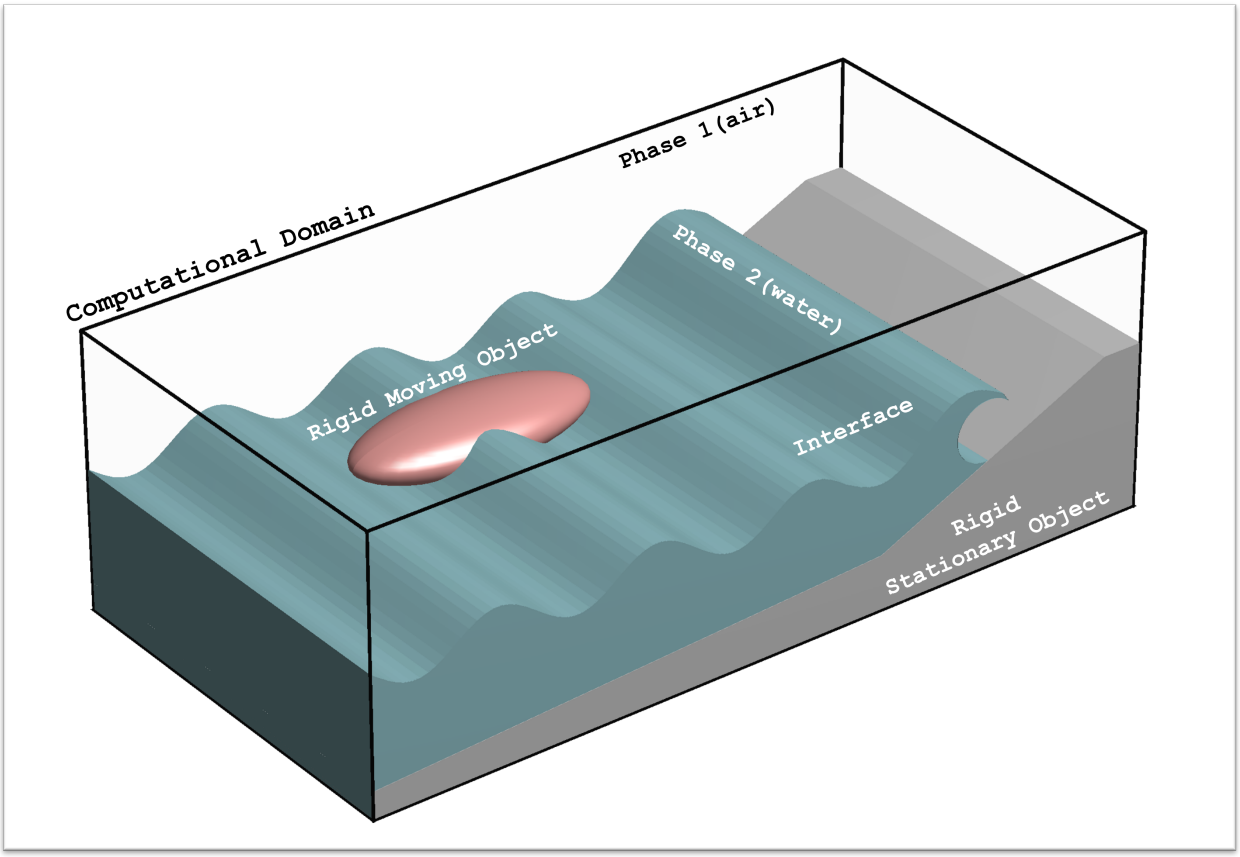


Figure 2.1: A schematic representation of a computational domain Ω containing a two-phase flow with rigid stationary and moving objects.

This equation describes the evolution of the interface in space and time, and at the same time, the scalar function I also satisfies the continuity equation since the flow is incompressible. The velocity field would advect the interface I along with it. In Eulerian methods, the interface's position and shape are unknown at the beginning of each time-step, and it needs to be captured from the spatial distribution of the fluid. Interface I is solved only in the fluid domain, and it is more consistent with the convection term presented in the non-conservative form [80]. Grid cells or computational elements that are part of an internal object and wholly immersed in the fluid are called object elements. An interesting way to consider the internal objects is as a fluid of zero velocity and infinite density, and hence no force is enough to move or deform it and remains stationary behaves like a solid object. Alternative approach to model the internal objects is by blocking out the entire object

elements, but this leads to objects with step-wise boundaries. To achieve a finer geometric representation of the boundaries of the curved objects in the computational domain, a technique called Fractional Area Volume Obstacle Representation (FAVOR) [70, 81, 82, 80] is used. In FAVOR method, the stationary or moving rigid bodies may have any arbitrary shape as geometry is defined only by the fractional face area and fractional volume that allows the fluid flow. A scalar field function $\Theta(\boldsymbol{x})$ takes the value zero if it is entirely an object element, whereas if some percentage of the volume of the object element is open for the fluid flow, then $\Theta(\boldsymbol{x})$ takes that fractional value. So in the case of internal objects, the continuity, Navier-Stokes, and free-surface advection equations can be written as follows.

$$\begin{aligned} \nabla \cdot (\Theta \boldsymbol{u}) &= 0 \\ \Theta \frac{\partial \boldsymbol{u}}{\partial t} + (\Theta \boldsymbol{u} \cdot \nabla)\boldsymbol{u} &= -\frac{\Theta}{\rho}\nabla p + \Theta \nu \nabla^2 \boldsymbol{u} + \Theta \boldsymbol{g} \\ \frac{\partial \Theta f}{\partial t} + (\boldsymbol{u} \cdot \nabla)\Theta f &= 0 \end{aligned} \tag{2.9}$$

When there are moving objects, the momentum and continuity equations need to be changed to account for occupying new grid cells or retrieving from old cells, imparting a force on the fluid and changing the boundaries of the fluid domain. Figure 2.2 illustrates a three-dimensional control volume with a moving object. V_o is the volume of object inside the control volume, and V_F the fluid volume inside. S_o and S_F represent the object and fluid surfaces, while S_{of} is the interface of the object and fluid. If more object volume is entering than leaving the control volume, that means the fluid volume decrease and vice-versa. Therefore, the conservation mass inside the control volume V is given by Eqn. 2.2.3 and 2.2.3 where $\dfrac{dV_o}{dt}$ is the rate of change of object volume inside the control volume.

$$\int_S (u \cdot n)\, dS = \int_{S_F} (u \cdot n_F)\, dS_F + \int_{S_o} (u_o \cdot n_o)\, dS_o = 0 \tag{2.10}$$

and,

$$\int_{S_{oF}} (u_o \cdot n_o)\, dS_{oF} = \frac{dV_o}{dt} \tag{2.11}$$

Taking this into account the new continuity equation inside the control volume becomes Eqn. 2.2.3

$$\Delta \cdot u = \frac{1}{V}(\frac{dV_o}{dt}) = \phi(\boldsymbol{x}, t), \tag{2.12}$$

where, the function $\phi(\boldsymbol{x}, t)$, ensures a zero divergence inside the control volume. The momentum equations for the control volume with moving object is attained by substituting Eqn. 2.2.3 into 2.4 to get Eqn. 2.13.

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \phi u = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{g} \tag{2.13}$$

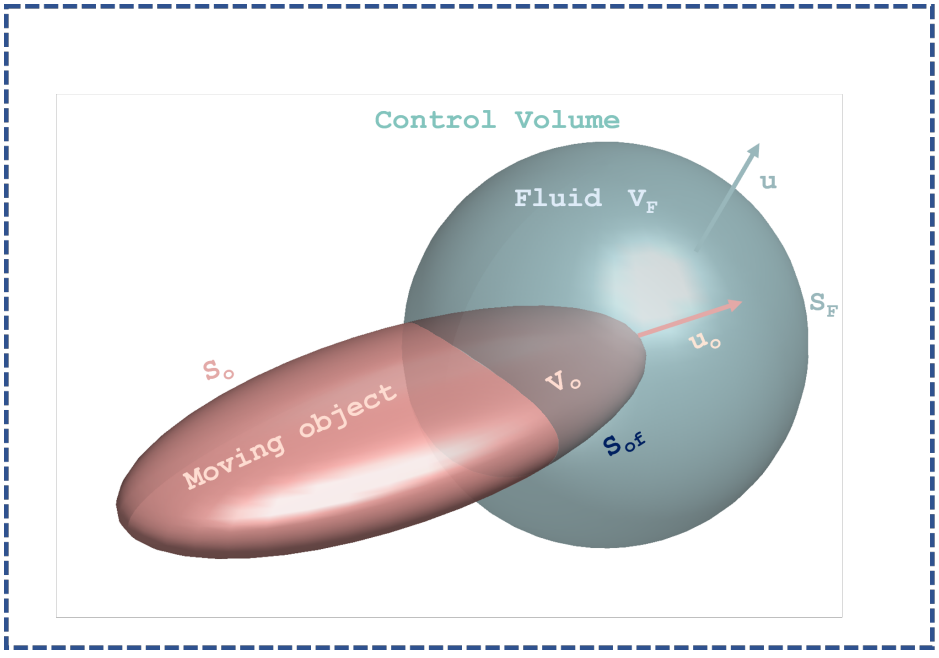


Figure 2.2: An illustration of a three-dimensional control volume with a moving object.

In general, the interface tracking methods use an Eulerian or Lagrangian approach. In the Lagrangian approach, either a set of marker particles distributed throughout the interface continuously track the interface, or an interface fitted mesh moves over a stationary Eulerian mesh. However, both strategies are not apt in case of large interface deformations or the breaking and merging interfaces, in such case, frequent remeshing would be required and would result in significantly higher computational cost. The Eulerian approach is generally more efficient and straightforward in the presence of highly deforming, merging, and breaking interfaces. It is also better than the Lagrangian methods in updating the interface location since the fluid flowing through the mesh is computed using the finite-difference method. Nevertheless, fluid flow properties get averaged in Eulerian methods while Eqn. 2.8 is solved numerically because of the integration of functional values over the control volume or grid cells. It would also spread the interface and reduce its curvatures, along with smoothing the fluid flow properties. Hence, extracting more information about the interface position or shape in the sub-grid resolution is not easy. One solution is to recreate an interface with the highest mathematical likelihood from the known averaged values. Interface capturing methods attempt to do this with better accuracy and lesser computational effort. Different types of interface capturing and advection methods are discussed in the first chapter. These discussions led to the development of a new surface tracking algorithm that can overcome some of the limitations by combining the strengths of the Lagrangian and Eulerian approaches. The literature review guided the author to use the volume-of-fluid (VOF) method as a starting point. VOF method developed by Hirt and Nichols in 1979 is a popular method that is currently being used in multiple commercial software with newer modifications. VOF method is outlined in the first chapter, and it will be explained in detail later in this chapter. The presented surface tracking and advection method is based on the VOF method and is named VOF-with-Center-of-Mass-and-Lagrangian-Particles (VCLP). The mathematical formulation of VCLP is presented in detail in the next chapter.

2.3 Boundary Conditions

The interfaces in the fluid flow and internal objects, both stationary and moving, requires boundary conditions. It is the initial conditions and the boundary conditions that introduce the dynamics and decide the path of solution. Moreover, if surface tension needs to be considered, the boundary condition at the contact line is required, which is the intersection of the surface of the solid object with the free surface. A typical computational domain related to ocean engineering problems features a sea bottom floor,seawall, structures, beach, wavemakers, landslides, dam gates, and floating objects as the internal objects, with some moving and some stationary. The solid boundaries have, in general, the no-slip boundary condition if fluid is viscous. The no-slip boundary condition is a Dirichlet and Neumann condition prescribed at the flow domain boundary, which states that the boundary is impermeable and fluid very close to the solid wall would have the same velocity or a zero velocity relative to the wall. So, fluid velocity at the point of contact will be zero for seafloor and other stationary objects. In contrast, the free-slip boundary condition neglects shear stress at the solid boundary, which means that the presence of the wall does not affect the tangential flow. Eqn. 2.3 shows the free-slip boundary condition.

$$
\begin{aligned}
u_n &= 0, \\
\frac{\partial u_t}{\partial n} &= 0
\end{aligned}
$$

The dot product of velocity and normal wall vector gives u_n, and the dot product of velocity and tangential vector gives u_t. u_n is the normal component of the fluid velocity at the solid wall, and u_t is the tangential velocity. Figure 2.3 shows the difference between the velocity profiles near the solid boundary wall with no-slip and free-slip boundary conditions. The physical boundary conditions at the interface between air and water play a major role in determining the behavior of the water waves. In the surface capturing methods, the interface is only known in the initial time $t = 0$. The kinematic and dynamic boundary conditions need to be applied at the free surface. While the kinematic boundary condition describes the evolution of the interface or free surface, the

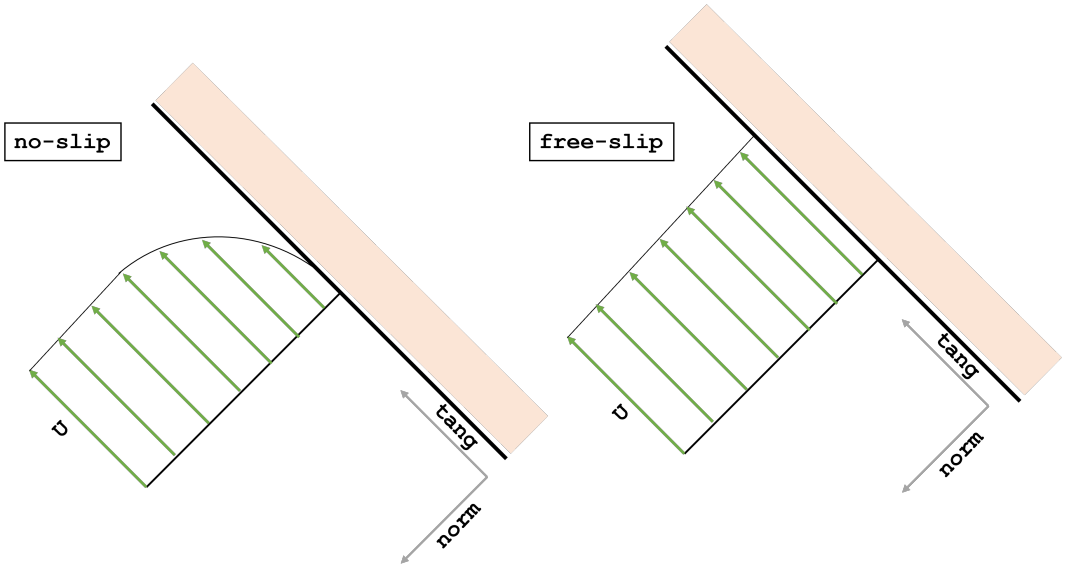


Figure 2.3: Solid boundary velocity profiles for no-slip and free-slip boundary conditions.

stresses acting on the interfaces are described by the dynamic boundary conditions. The tangential free-surface dynamic boundary condition can be obtained from the force balance at the boundary, and Eqn. 2.14 shows it below.

$$\mu \left(\frac{\partial u_{\boldsymbol{n}}}{\partial \boldsymbol{t}} + \frac{\partial u_{\boldsymbol{t}}}{\partial \boldsymbol{n}} \right) = 0 \tag{2.14}$$

The normal free-surface dynamic boundary condition got from the force balance at the boundary is given by Eqn. 2.15.

$$-p + 2\mu \frac{\partial u_t}{\partial \boldsymbol{n}} = -p_0 + \sigma\kappa \tag{2.15}$$

Here p_0 is the ambient air pressure, σ is the surface tension, and κ is the interface curvature. The mean curvature of an interface is given by the divergence of normal $\boldsymbol{n}$ as shown in Eqn. 2.16.

$$\kappa = \nabla \cdot \boldsymbol{n} = \nabla \cdot \left(\frac{\nabla I}{|\nabla I|} \right) \tag{2.16}$$

It is assumed in the dynamic boundary condition that the ambient pressure is constant or zero. Because of this approximation, the air and water phases in ocean waves are decoupled. If surface tension and viscosity effects on the boundary are neglected, it leads to $p = p_0$. Kinematic bound-

ary conditions describe the evolution and movement of the free surface. Many surface tracking methods exist because even though kinematic conditions are well defined in both Eulerian and Lagrangian approaches, we can derive different forms of kinematic boundary conditions. In order to balance the forces, the interface has to satisfy the boundary conditions and be in agreement with the stationary and moving objects. Since the exact interface is only known at the initial time $t = 0$, the Eulerian approach requires the reconstruction of interfaces each time step, and its accuracy is crucial to the solution.

3. VOF WITH THE CENTER OF MASS AND LAGRANGIAN PARTICLES (VCLP) METHOD

3.1 Problem formulation

Consider a laboratory wave basin with a two-dimensional piston wavemaker creating waves that travel towards the increasing slope and the beach. The wave propagates in the X-direction, and the computational domain is divided using a Cartesian grid system with non-uniform cell sizes. The control volumes are fixed in space and time. Figure 3.1 shows a schematic representation of the 2D cross-sectional view of the computational domain. The cell dimensions have a higher resolution near the wave-breaking zone, and the resolution in each direction varies in a quadratic manner.

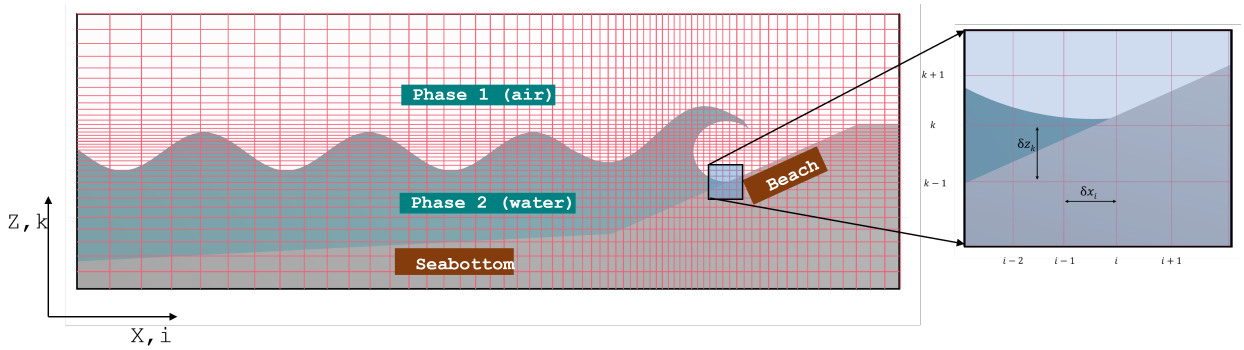


Figure 3.1: Schematic representation of the 2D cross-sectional view of a computational domain for a typical nearshore two-phase ocean flow with slopped bottom and beach. The resolution of Cartesian cells varies in a quadratic manner along the X and Z directions. Zoomed in view shows the numbering of cells.

The known quantities at the start of a time step in the computational domain are the geometry, boundary conditions, grid locations, fluid distribution, pressure, and velocity field from the previous time step. Now, the equation of continuity and N-S equations in the discretized form are applied to the control volumes, and by solving this large set of simultaneous equations corre-

sponding to every cells, the pressure and velocity field for the given time step is solved. Once the pressure and velocity field is known, fluid is advected according to the velocity field. The changes to the control volume quantities such as mass and momentum are updated and given as an input to solve pressure and velocity for the next time step.

Now, when an interfaces is present inside a cell, it needs to be located and reconstructed before advection, and during advection, different methods are implemented to ensure accuracy and a non-divergent flow. The increase in grid resolution is one way to improve accuracy, but by the expense of computational time as this would lead to an increase in the number of pressure-velocity coupled simultaneous equations to be solved implicitly. The proposed method, VCLP, attempts to improve surface tracking and advection accuracy of the SOLA-VOF [70] method, without a significant increase in computational intensity compared to the increasing resolution. It uses the solved velocity field, fluid mass and CM distribution from a given time step, reconstructs interfaces, advects, and redistributes the fluid for the next time step.

To present the proposed method in the simplest way, consider the earlier mentioned laboratory wave basin with a two-dimensional waves traveling towards the beach. We assume air is void, making it a single-phase flow with a free surface or surface. Now, the computational domain $\mathbf{\Omega}$ represent the 3D space occupied by the wave basin, and $\mathbf{\Omega}$ is discretized into n_X, n_Y, n_Z Cartesian grid cells or control volumes in positive X, Y, Z directions. The wave simulation time T_s is discretized into n_t time steps where m^{th} time step t_m have a duration of Δt_m units. The space occupied by any cell inside $\mathbf{\Omega}$ is denoted by 's' and is located by a 3-tuple (i, j, k), which are cell's position numbers along X, Y, and Z axes. $\delta x_i, \delta y_j, \delta z_k$ are corresponding cell's dimensions along X, Y and Z axes. So, $\mathbf{\Omega}$ is the sum of all the control volumes or cell spaces as given in Eqn. 3.1.

$$\sum_{k=1}^{k=n_Z} \sum_{j=1}^{j=n_Y} \sum_{i=1}^{i=n_X} s(i, j, k) = \mathbf{\Omega} \tag{3.1}$$

If the volume of water inside the cell $s(i, j, k)$ at time t_m is $v(i, j, k, m)$, then total volume of water

in the wave basin during time t_m is represented as $\boldsymbol{V_m}$ given by the Eqn. 3.2.

$$\sum_{k=1}^{k=n_Z} \sum_{j=1}^{j=n_Y} \sum_{i=1}^{i=n_X} v(i,j,k,m) = \boldsymbol{V_m} \tag{3.2}$$

Now, consider a partially filled surface cell in $\boldsymbol{\Omega}$, $s(i,j,k)$ at time t_m. The volume of water inside this cell is denoted by $v(i,j,k,m)$ and the cell is shown in Fig. 3.2. The point (x_i, y_j, z_k) is located at top diagonal corner of the cell and velocity components satisfying the zero divergence condition are shown defined at the the cell centers. The dimensions of the cell are $\delta x_i, \delta y_j$, and δz_k). The

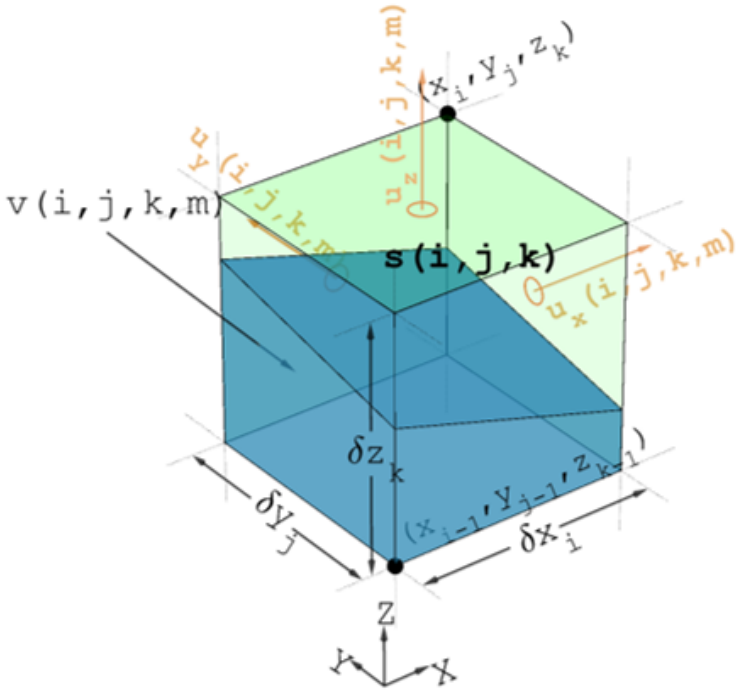


Figure 3.2: Cell space $s(i,j,k)$ at time step t_m with fluid volume $v(i,j,k,m)$, dimension $(\delta x_i, \delta y_j, \delta z_k)$. Exact location of (x_i, y_j, z_k) and $(x_{i-1}, y_{j-1}, z_{k-1})$ is shown along with the three components of the advective velocity $\boldsymbol{u}(i,j,k,m)$ defined at the cell face centers.

volume occupied by this cell is calculated from its dimensions using Eqn. (3.3). Here the modulus sign represents the volume of the cell space $s(i,j,k)$.

$$|s(i,j,k)| = \delta x_i \delta y_j \delta z_k \tag{3.3}$$

The center of mass (CM) denoted by '$\boldsymbol{c}$' of the water inside the cell is given by Eqn. (3.4), where ρ_w is water's density, M_w is mass, and $\boldsymbol{x} = (x, y, z)$ is the position vector. ρ_w is taken outside the integral on the assumption that density is uniform throughout the cell volume. If the fluid is represented by infinitely many uniformly distributed particles with equal mass, then the average of their positions would be equal to the location of the CM.

$$\boldsymbol{c}(i,j,k,m) = (c_x, c_y, c_z) = \frac{\rho_w}{M_w} \iiint\limits_{v(i,j,k,m)} \boldsymbol{x} \; dv = \frac{\iiint\limits_{v(i,j,k,m)} \boldsymbol{x} \; dv}{\iiint\limits_{v(i,j,k,m)} dv} \tag{3.4}$$

The volume occupied by the water in any cell in $\boldsymbol{\Omega}$ always satisfies the inequality condition given by Eqn. (3.5) which simply states volume is a positive quantity and its maximum value is the volume of cell space itself.

$$0 \leq v(i,j,k,m) \leq |s(i,j,k)| \tag{3.5}$$

Now, a binary value function $F(\boldsymbol{x}, m)$ with outputs zero and one, based on the VOF method, is defined at every point in computational domain $\boldsymbol{\Omega}$ and through time $[0, T_s]$. F can be considered a marker particle of zero sizes that takes a value of one if water is present, else zero as given in Eqn. 3.6. This means that an empty cell have zeros and completely filled fluid cells have ones throughout the cell space.

$$F(\boldsymbol{x}, m) = \begin{cases} 1, & \text{if water is present at location } \boldsymbol{x} \text{ at time } t_m \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

For the cell space $s(i,j,k,m)$ where F is defined everywhere, the average of value of F function gives an another function f known as fractional fluid volume is given by Eqn. 3.7. f is a scalar

value defined for every cells in the domain.

$$f(i,j,k,m) = \frac{1}{G}\sum_{g=1}^{g=G} F(x_g, y_g, z_g, t_m) \begin{cases} \text{where} & x_{i-1} < x_g \leq x_i \\ & y_{j-1} < y_g \leq y_j \\ & z_{k-1} < z_g \leq z_k \\ G \to \infty & \end{cases} \tag{3.7}$$

Since, the marker particles are filled uniformly, the value of $f(i,j,k,m)$ would be exactly same as the fractional volume of water in the cell $s(i,j,k,m)$ which gives the Eqn. 3.8. This means that f value is is the ratio of fluid volume estimated from the geometry of the fluid boundaries inside cell including the interface to the cell space volume.

$$f(i,j,k,m) = \frac{v(i,j,k,m)}{|s(i,j,k)|} \tag{3.8}$$

If Eqn. 3.5 is applied to Eqn. 3.8, it gives that the value of f always lies between zero and one, as shown in Eqn. 3.9. Even though this is a strict condition in theory, numerical methods may cause it go out of the limit ($f > 1.0$ or $f < 0$) because of non-zero divergence error.

$$0 \leq f(i,j,k,m) \leq 1 \tag{3.9}$$

$\boldsymbol{f_m}$ is a matrix of dimension $n_X \times n_Y \times n_Z$ filled with $f(i,j,k,m)$ values of all cells. Now, consider three matrices of same dimensions containing the velocities $\boldsymbol{u}(i,j,k,m)$ corresponding to every cells in $\boldsymbol{\Omega}$, represented as $\boldsymbol{U_{X,m}}$, $\boldsymbol{U_{Y,m}}$, and $\boldsymbol{U_{Z,m}}$. Together these three matrices is denoted as $\boldsymbol{U_m}$. Similarly the CM matrices is denoted as $\boldsymbol{c_m}$. Then the entire purpose of VCLP can be conveyed by the simple relation between input and output parameters given by Eqn. 3.10. Thus VCLP needs velocity field, fluid distribution, and the CM location of the given time step to provide the fluid distribution, and the CM for the next time step. So it takes seven/four scalar

input/output in 3D and five/three in 2D.

$$\text{VCLP(input parameters)} = \text{(output parameters)}$$

$$\boxed{\textit{VCLP}(\boldsymbol{f_m}, \boldsymbol{c_m}, \boldsymbol{U_m}) = (\boldsymbol{f_{m+1}}, \boldsymbol{c_{m+1}})} \tag{3.10}$$

3.2 Classification of Cells

The purpose of classification of cells is to identify if a cell have fluid and interface as only cells with fluid needs advection and only cells with interface require interface reconstruction. Therefore, all the cells in $\boldsymbol{\Omega}$ are classified into empty (E), surface (S), and interior cells $\boldsymbol{N}$. VCLP further classifies the interior cells into three more to increase the computational performance by skipping cells that satisfy certain conditions using a new method called Skip Core Optimization (SCO), which is explained later in the chapter. The three subdivisions of the interior cells are Inner Surface (N_s), Inner Mantle (N_m), and Inner Core (N_c). The classification is done on the basis on the value of cell-type denoted by C_T. Table 3.1 gives the classifications and the corresponding C_T values.

Table 3.1: Classification of cells in the computational domain $\boldsymbol{\Omega}$.

Class	Notation	C_T
Empty	E	0
Surface	S	1
Inner Surface	N_s	2
Inner Mantle	N_m	3
Inner Core	N_c	4
Interior	$\boldsymbol{N}$	2,3,4

The C_T values for the classification of cells are calculated in the following steps 1 to 6, where q is the ceiling function value of the Courant–Friedrichs–Lewy (CFL) number ($q = \lceil \textit{CFL} \rceil$) and

the CFL number is given by $CFL = \dfrac{u_{max}\Delta t}{\delta x}$. C_T^* is the temporary C_T value which gets updated along the steps.

1. $C_T^*(i,j,k,m) = \lfloor f(i,j,k,m) \rfloor$
2. $C_{T1}(i,j,k,m) = \sum_{k=k-q}^{k=k+q} \sum_{j=j-q}^{j=j+q} \sum_{i=i-q}^{i=i+q} C_T^*(i,j,k,m)$
3. $C_T(i,j,k,m) = \begin{cases} 2 & \text{where } C_{T1} = (2q+1)^3 \text{ and } f(i,j,k,m) > 0.5 \\ C_T^*(i,j,k,m) & \text{otherwise} \end{cases}$
4. $C_{T2}(i,j,k,m) = \sum_{k=k-q}^{k=k+q} \sum_{j=j-q}^{j=j+q} \sum_{i=i-q}^{i=i+q} C_T^*(i,j,k,m)$
5. $C_T^*(i,j,k,m) = \begin{cases} 3 & \text{where } C_{T2} = 2(2q+1)^3 \\ C_T^*(i,j,k,m) & \text{otherwise} \end{cases}$
6. $C_T(i,j,k,m) = \begin{cases} 4 & \text{where } C_{T3} = 3(2q+1)^3 \\ C_T^*(i,j,k,m) & \text{otherwise} \end{cases}$

Figure 3.3 shows the classification of cells in 2D and 3D. The left figure shows a surface cell C and its twenty-six neighbors and some of their classifications: which are empty(E), surface(S), and interior($\boldsymbol{N}$) cells based on the C_T value. Here a planar fluid surface cuts through the cell C. The figure in the middle shows a part of a circular 2D fluid body inside a rectangular domain with square grid cells. The radius is 4.8 times the cell length. Figure in the right shows how the 2D fluid body is classified according to the C_T values, and transformed into the f values, and the CM locations.

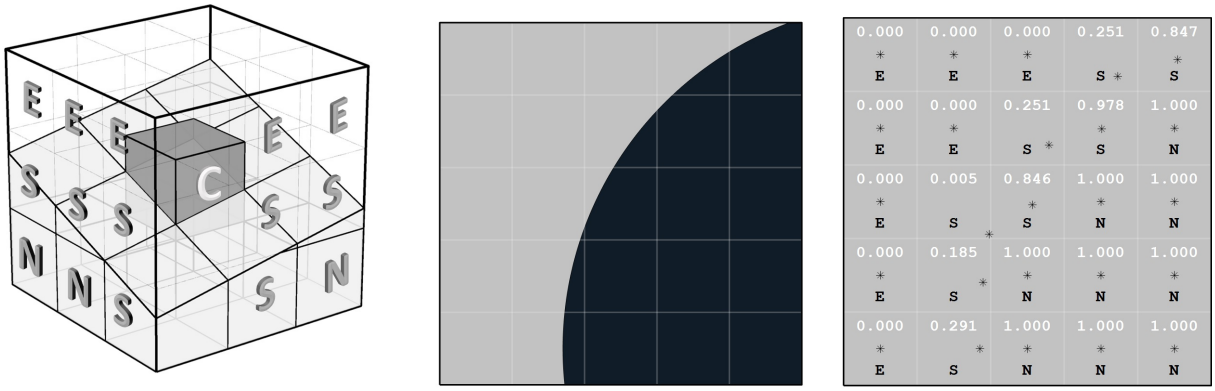


Figure 3.3: *Left:* 3D cell classification into empty ('E'), surface ('S') and interior ('N'). Cell 'C' is a surface cell. Slanted plane is the fluid surface. *Middle:* A part of a 2D circular fluid body and grid cells drawn on top. *Right:* Equivalent f-value, cell classifications and the CM locations represented by '*'.

The calculation of C_T can be explained as follows. Cells without any fluid are an empty cell (E) with $C_T = 0$, else non-empty . If two cell have at least a face, edge, or vertex in common, they are called neighbor cells. In 2D Cartesian grid system, there are eight neighbor cells and twenty-six in 3D. Step 1 label all non-empty cells as surface cells ($C_T = 1$). Steps 2 and 3 identifies surface cells with all neighbors also being surface cells and classify them as inner surface cells ($C_T = 2$). Steps 3 and 4 then identifies the inner surface cells with all neighbors also being inner surface cells and classify them as inner mantle cells ($C_T = 3$). Finally Steps 5 and 6 identify the inner mantle cells with all neighbors also being inner mantle cells and classify them as inner core cells ($C_T = 4$).

3.3 Interface Reconstruction

Interface reconstruction is done after the classification for all the surface cells (S) in $\mathbf{\Omega}$ identified by f value between less than one and greater than zero. In VCLP, there are two choices for interface reconstruction calculation; Piecewise-Linear-Interface-Calculation (PLIC) method and the Piecewise-Circular-Interface-Calculation method (PCIC). PLIC is used by many models and it creates a line segment representing the interface in 2D and a plane in 3D, whereas PCIC creates a circular arc for the interface in 2D and a spherical surface in 3D. The following sections look into

two methods closely.

3.3.1 Interface Normal for PLIC

The objective of an analytically recreated PLIC interface is to match closely to the actual fluid interface in terms of two parameters which are the f value and mean slope. Even though it may not be possible to predict the interface exactly from the averaged Eulerian information, it is possible to predict the most likely interface. Now, we go back to the surface cell $s(i, j, k, m)$ mentioned in Fig. (3.2). The PLIC construction is done using the f values of the cell and its neighbors. $\boldsymbol{L}(i, j, k, m)$ denotes the PLIC construction of the cell given by Eqn. 3.11.

$$\boldsymbol{L}(i, j, k, m) : n_x x + n_y y + n_z z + a = 0 \tag{3.11}$$

Where direction of the planar surface is given by its normal $\boldsymbol{n} = (n_x, n_y, n_z)$ and a is the planar constant. In theory, the planar normal is determined using the Eqn. 3.12 where ∇F is the gradient of the F. However, only average of F in the cell, f is known and not F. As the size of cells gets smaller, f converges towards F and hence, the approximate Eqn. 3.13 is written where ∇f is given by Eqn. 3.14.

$$\boldsymbol{n} = \frac{\nabla F}{|\nabla F|} \tag{3.12}$$

$$\frac{\nabla f}{|\nabla f|} \approx \frac{\nabla F}{|\nabla F|} \tag{3.13}$$

$$\nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k} \tag{3.14}$$

f function is discretized in space, and to find the gradient along the X axis the Eqn. 3.15 is used. Similarly, the other two components along Y and Z axes of the gradient function can be estimated. $\nabla_n f$ denotes the interface normal estimated this where subscript n stands for numerical.

$$\nabla_{n_X} f = \frac{\partial f}{\partial x}_{(i,j,k)} = \frac{f_{(x)}(i+1) - f_{(x)}(i)}{\delta x_{i+1} + \delta x_i} + \frac{f_{(x)}(i) - f_{(x)}(i-1)}{\delta x_{i-1} + \delta x_i} \tag{3.15}$$

where,

$$f_{(x)}(i) = \frac{\sum_{k=k-1}^{k=k+1} \sum_{j=j-1}^{j=j+1} f(i,j,k)\delta y_j \delta z_k}{\sum_{k=k-1}^{k=k+1} \sum_{j=j-1}^{j=j+1} \delta y_j \delta z_k}$$

In the literature, the two common ways to estimate the interface normal are by finding the $\nabla_n f$ and by using a method dealing minimum slope with respect to the axes. We may call these numerical gradient method and the minimum slope error method. Descriptions for these methods are given below, after which a new proposed method called T- Slope method, is presented.

3.3.1.1 Numerical Gradient Method

This method is same as finding $\nabla_n f$, however Eqn. 3.16 shows the calculation of normal using matrices in 2D. The linear system of equations can form a matrix and can be represented as $\boldsymbol{A}\tilde{\boldsymbol{n}} = \boldsymbol{b}$ and this can be solved by multiplying with the transpose of $\boldsymbol{A}$ matrix which gives $\tilde{\boldsymbol{n}} = \boldsymbol{A^T b}$ and is solved by Gaussian elimination method. The 'normal' is normalized by $\boldsymbol{n} = \dfrac{\tilde{\boldsymbol{n}}}{|\tilde{\boldsymbol{n}}|}$

$$\begin{pmatrix} 0.5(\delta x_{i+1} + \delta x_i) & 0 \\ 0.5(\delta x_{i+1} + \delta x_i) & 0.5(\delta y_{j+1} + \delta y_j) \\ 0 & 0.5(\delta y_{j+1} + \delta y_j) \\ -0.5(\delta x_{i-1} + \delta x_i) & 0.5(\delta y_{j+1} + \delta y_j) \\ -0.5(\delta x_{i-1} + \delta x_i) & 0 \\ -0.5(\delta x_{i-1} + \delta x_i) & -0.5(\delta y_{j-1} + \delta y_j) \\ 0 & -0.5(\delta y_{j-1} + \delta y_j) \\ 0.5(\delta x_{i+1} + \delta x_i) & -0.5(\delta y_{j-1} + \delta y_j) \end{pmatrix} \begin{pmatrix} \tilde{n_x} \\ \tilde{n_y} \end{pmatrix} = \begin{pmatrix} f_{i+1,j+0} - f_{i,j} \\ f_{i+1,j+1} - f_{i,j} \\ f_{i+0,j+1} - f_{i,j} \\ f_{i-1,j+1} - f_{i,j} \\ f_{i-1,j+0} - f_{i,j} \\ f_{i-1,j-1} - f_{i,j} \\ f_{i+0,j-1} - f_{i,j} \\ f_{i+1,j-1} - f_{i,j} \end{pmatrix} \tag{3.16}$$

3.3.1.2 Minimum Slope method

This method is described by the developers of the VOF method [69], checks if the surface is represented more error-free as $Y(x)$ or $X(y)$. 'This is determined by calculating minimum of $\left|\dfrac{dX}{dy}\right|$ and $\left|\dfrac{dY}{dx}\right|$. If $\left|\dfrac{dX}{dy}\right|$ is smaller, then surface is represented as $X(y)$, else $Y(x)$. The equations

used for this are shown in Eqns. 3.17 to 3.20.

$$Y_i = Y(x_i) = f_{i,j-1}\delta y_{j-1} + f_{i,j}\delta y_j + f_{i,j+1}\delta y_{j+1} \tag{3.17}$$

$$X_j = X(y_j) = f_{i-1,j}\delta x_{i-1} + f_{i,j}\delta x_i + f_{i,j+1}\delta x_{i+1} \tag{3.18}$$

$$\left(\frac{dY}{dx}\right)_i = \frac{2(Y_{i+1} - Y_{i-1})}{\delta x_{i+1} + 2\delta x_i + \delta x_{i-1}} \tag{3.19}$$

$$\left(\frac{dX}{dy}\right)_j = \frac{2(X_{j+1} - X_{j-1})}{\delta y_{j+1} + 2\delta y_j + \delta y_{j-1}} \tag{3.20}$$

3.3.1.3 T-Slope method

T-Slope method is a new proposed method to estimate the interface normal. It uses some of the ideas behind both, the numerical gradient method and the minimum slope method. The T-slope method is described by Eqns. 3.21 to 3.26 in order to find the interface normal based on the intermediate parameters $S_x, S_y, H_x, H_y, F_x, F_y, \alpha$ and θ. The first two parameters S_x and S_y represent the change is volume of with change in distance, with units of height. They are calculated with Eqn. 3.21.

$$\begin{aligned} S_x &= \sum_{j=j-1}^{j=j+1} \delta y_j \left(\frac{f_{i+1,j}\delta x_{i+1} - f_{i,j}\delta x_i}{\delta x_i + \delta x_{i+1}} + \frac{f_{i,j}\delta x_i - f_{i-1,j}\delta x_{i-1}}{\delta x_i + \delta x_{i+1}} \right) \\ S_y &= \sum_{i=i-1}^{i=i+1} \delta x_i \left(\frac{f_{i,j-1}\delta y_{j-1} - f_{i,j}\delta y_j}{\delta y_j + \delta y_{j-1}} + \frac{f_{i,j}\delta y_j - f_{i,j+1}\delta y_{j+1}}{\delta y_j + \delta x_{j+1}} \right) \end{aligned} \tag{3.21}$$

Then, Eqn. 3.22 is used to estimate the two slopes parameters H_x and H_y which are non-dimensional.

$$\begin{aligned} H_x &= \sum_{j=j-1}^{j=j+1} \delta y_j \left(\frac{f_{i+1,j} - f_{i,j}}{\delta x_i + \delta x_{i+1}} + \frac{f_{i,j} - f_{i-1,j}}{\delta x_i + \delta x_{i+1}} \right) \\ H_y &= \sum_{i=i-1}^{i=i+1} \delta x_i \left(\frac{f_{i,j-1} - f_{i,j}}{\delta y_j + \delta y_{j-1}} + \frac{f_{i,j} - f_{i,j+1}}{\delta y_j + \delta x_{j+1}} \right) \end{aligned} \tag{3.22}$$

Eqn. 3.23 estimates two non-dimensional parameters which are independent of the cell dimen-

sions.

$$
\begin{aligned}
F_x &= \frac{\sum_{j=j-1}^{j=j+1} f_{i+1,j} - \sum_{j=j-1}^{j=j+1} f_{i-1,j}}{\sum_{j=j-1}^{j=j+1} f_{i,j}} \\
F_y &= \frac{\sum_{i=i-1}^{i=i+1} f_{i,j-1} - \sum_{i=i-1}^{i=i+1} f_{i,j+1}}{\sum_{i=i-1}^{i=i+1} f_{i,j}}
\end{aligned}
\tag{3.23}
$$

The magnitudes of these two parameters F_x and F_y are used to estimate an another parameter α as in Eqn. 3.24.

$$
\alpha = \begin{cases} 1 & \text{if } |F_x| < |F_y| \\ 0 & \text{else} \end{cases}
\tag{3.24}
$$

The calculated parameters $S_x, S_y, H_x, H_y, F_x, F_y$ and α are used as in Eqn. 3.3.1.3 to estimate θ which is the angle of rotation of interface with respect to the X-axis.

$$
\theta = \begin{cases} \tan^{-1}(H_y) + \pi & \text{if } \alpha = 0 \;\&\; S_x \geq 0 \\ -\tan^{-1}(H_y) & \text{if } \alpha = 0 \;\&\; S_x < 0 \\ \tan^{-1}(H_x) + \dfrac{\pi}{2} & \text{if } \alpha = 1 \;\&\; S_y \leq 0 \\ -\tan^{-1}(H_x) - \dfrac{\pi}{2} & \text{if } \alpha = 1 \;\&\; S_y > 0 \end{cases}
\tag{3.25}
$$

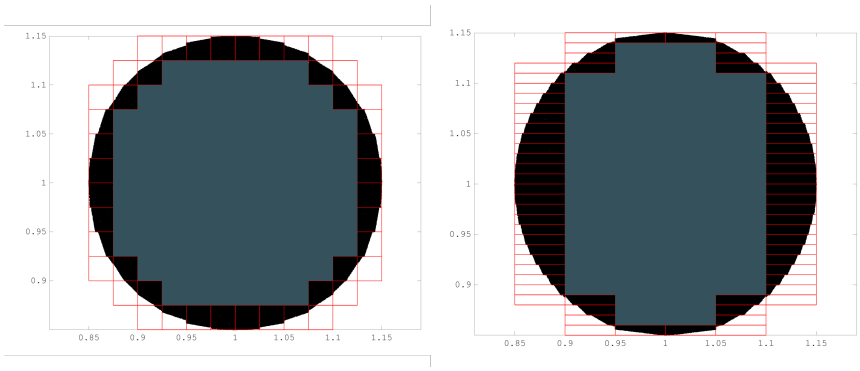
Finally, the components on the interface normal are estimated using Eqn. 3.26.

$$
n_x = \cos\theta, \;\; n_y = \sin\theta
\tag{3.26}
$$

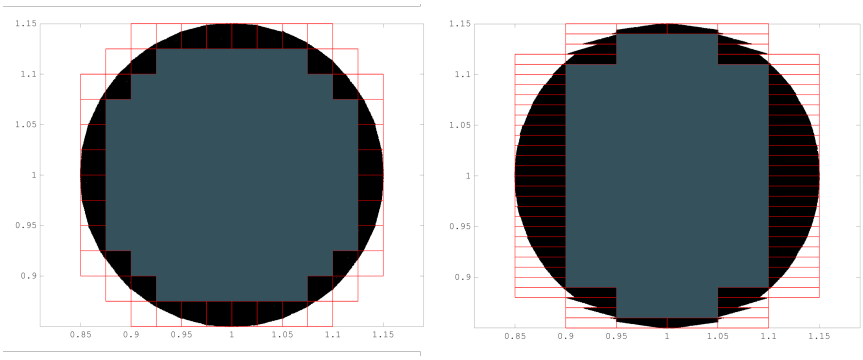
3.3.1.4 Comparison of methods to find the interface normal

A two-dimensional circular fluid domain is reconstructed from the actual f values, i.e., from the intersection area of the circle and cell boundaries using PLIC with the three methods described

in previous sections. The circular domain is centered at (1,1) with a radius of 0.15 units. The grid resolution for two cases are 1) $\delta x = 0.025$, $\delta y = 0.025$, with aspect ratio 1:1, and 2) $\delta x = 0.05$ and $\delta y = 0.01$ with aspect ratio 5:1. The qualitative results are shown in Fig. 3.4 and 3.5. The numerical gradient method interfaces are not smooth and have significant discontinuities in both cases with different aspect ratios. The minimum slope method works well with 1:1 aspect ratio, however for the larger or smaller aspect ratios it is biased and have larger errors. It can be seen that the estimated slope near the top and bottom parts of the circular fluid are more horizontal than the true slope for the aspect ratio of 5:1. The introduced T-slope method seems to resolve this issue, producing good result for both scenarios as can be seen from Fig. 3.5.



(a) Normal Gradient Method.



(b) Minimum Slope Method.

Figure 3.4: Circular fluid domain reconstructed from the f values using PLIC with interface normal calculated using a) Normal Gradient Method b) Minimum Slope Method.

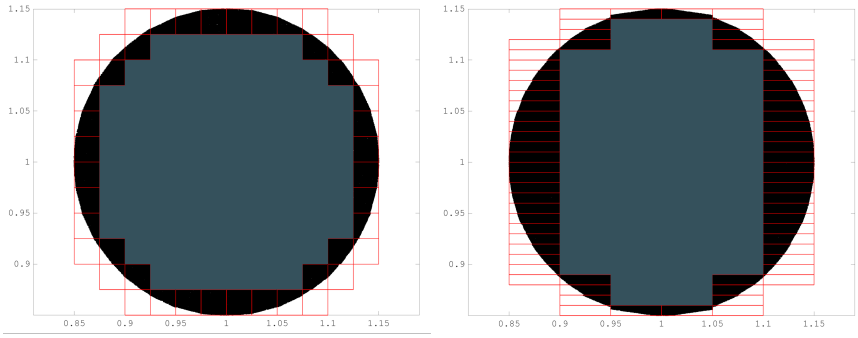


Figure 3.5: Circular fluid domain reconstructed from the f values using PLIC with interface normal calculated using the T-Slope Method.

3.3.2 Interface Curvature for PCIC

The Piecewise-Circular-Interface-Calculation (PCIC) is an alternative interface reconstruction method to PLIC, but with a higher order. It constructs a circular arc (2D) or a spherical surface (3D) for all the surface cells in with f values less than one. A PLIC construction can be seen as an equivalent PCIC construction withe radius of the of the circular arc approaching infinity. As PCIC have higher order than PLIC, improved accuracy is expected in the reconstructed interface, and consequently in the simulated flow. Since the curvature information can also be used to estimate the surface tension force, it is expected to improve accuracy in related scenarios also. PCIC build on top of PLIC meaning PLIC calculation are required for the PCIC. The PCIC construction inside the cell $s(i,j,k,m)$ is denoted as $\boldsymbol{C}(i,j,k,m)$ and in 3D, it is given by Eqn. 3.27,

$$\boldsymbol{C}(i,j,k,m) : (x-x_0)^2 + (y-y_0)^2 + (z-z_0)^2 - R^2 = 0 \begin{cases} x_{i-1} < x \leq x_i \\ y_{j-1} < y \leq y_j \\ z_{k-1} < z \leq z_k \end{cases} \tag{3.27}$$

where (x_0, y_0, z_0) is the sphere's center and R its radius. The extra information required for PCIC than PLIC is radius of the sphere which PCIC construction is part of or equivalently the mean

interface curvature (MIC) which is the reciprocal of R, i.e. κ.

3.3.2.1 MIC in 2D

VCLP use a new method named, average neighbor curvature method in 2D to estimate the mean-interface -curvature (MIC) or $\bar{\kappa}$. Here, MIC is found by averaging four curvatures calculated from the four nearest neighbor surface cells. Figure 3.6 shows the normals drawn at the four neighbors and their intersection with the normal of the surface cell in consideration providing the neighbor radius'. MIC is then calculated using Eqn. (3.28) where R_m is the mean radius of curvature and r_k is the neighbor radius. Normals are drawn at the midpoints of PLICs, which are shown as dots.

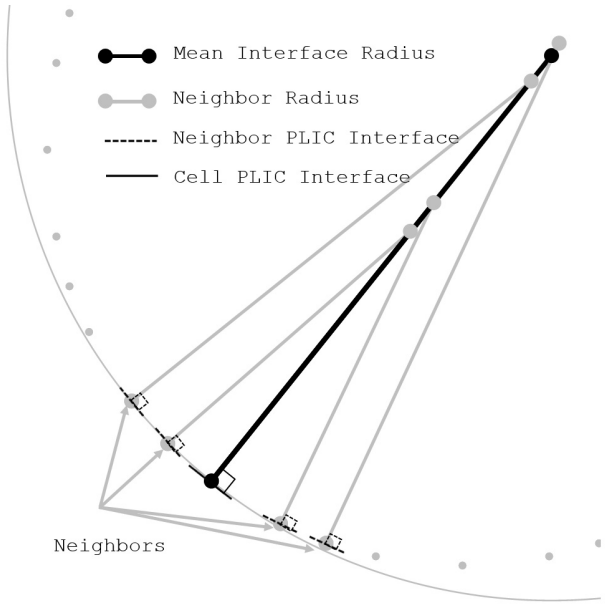


Figure 3.6: Estimation of the mean interface curvature in 2D for the PCIC construction. Normals are drawn from the mid point of nearest four neighboring PLICs to the normal of the surface cell in consideration passing through its PLIC construction's midpoint. Neighbor curvatures are averaged to get the mean interface curvature (MIC).

$$R_m = \frac{4}{\sum_{k=1}^{k=4} \frac{1}{r_k}} \tag{3.28}$$

Neighbor radius r_k is found using Eqn. (3.29) where (x_0, y_0) is the mid point of the PLIC construction and (n_x, n_y) is its normal passing through its midpoint, (m_x, m_y). (n_{ex}, n_{ey}) is the normal of

the neighbor's PLIC construction.

$$r_k = \frac{|n_{ex}(y_0 - m_y) - n_{ey}(x_0 - m_x)|}{\sqrt{n_x n_{ey} - n_y n_{ex}}} \tag{3.29}$$

3.3.2.2 MIC in 3D

To estimate MIC in 3D, a new method called differential interpolation (DI) is used. DI uses the surface normals located at the centroids of $\boldsymbol{L}$ constructions in the cell and in its nearest three neighboring surface cells. The centroid of the cell of consideration is $\boldsymbol{e_0}$ and its three neighbors are $\boldsymbol{e_1}, \boldsymbol{e_2}$ and $\boldsymbol{e_3}$. Similarly four normals at the centroids are $\boldsymbol{n_0}$ to $\boldsymbol{n_3}$. The mean radius of curvature or the inverse of MIC ($\bar{\kappa}$) is given by Eqn. (3.30) where $\nabla \cdot \boldsymbol{n}$ is the divergence of normal at the $\boldsymbol{L}$ centroid.

$$R = \frac{1}{\bar{\kappa}} = -\frac{2}{\nabla \cdot \boldsymbol{n}}, \;\; \nabla \cdot \boldsymbol{n} = \frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y} + \frac{\partial n_z}{\partial z} \tag{3.30}$$

The three components of $\nabla \cdot \boldsymbol{n}$ are found using Eqn. (3.31) to (3.33). Note that here $\boldsymbol{n_0} - \boldsymbol{n_k} = (X_k, Y_k, Z_k)$ and $\boldsymbol{e_0} - \boldsymbol{e_k} = (x_k, y_k, z_k)$ where k =1, 2, and 3 representing the three nearest neighboring surface cells.

$$\frac{\partial n_x}{\partial x} = \frac{X_1(y_2 z_3 - y_3 z_2) + X_2(y_3 z_1 - y_1 z_3) + X_3(y_1 z_2 - y_2 z_1)}{D} \tag{3.31}$$

$$\frac{\partial n_y}{\partial y} = \frac{Y_1(x_3 z_2 - x_2 z_3) + Y_2(x_1 z_3 - x_3 z_1) + Y_3(x_2 z_1 - x_1 z_2)}{D} \tag{3.32}$$

$$\frac{\partial n_z}{\partial z} = \frac{Z_1(x_2 y_3 - x_3 y_2) + Z_2(x_3 y_1 - x_1 y_3) + Z_3(x_1 y_2 - x_2 y_1)}{D} \tag{3.33}$$

where, $D = x_1(y_2 z_3 - y_3 z_2) + x_2(y_3 z_1 - y_1 z_3) + x_3(y_1 z_2 - y_2 z_1)$

3.3.2.3 Locating the PLIC and PCIC interfaces

The previous section showed the estimation of the interface normal $\boldsymbol{n}$ for PLIC and MIC for PCIC. The remaining variable in the exact definition of the interface is the planar constant 'a' in PLIC and the center of the sphere (x_0, y_0, z_0) for PCIC. In the PLIC construction if the interface is

accurately located, then the fractional volume of of the cell that satisfies the condition $n_x x + n_y y + n_z z > -a$ will be equal to the f value. Similarly for PCIC, if the interface is accurately located, the space (volume) inside the cell that satisfies the relation $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 < R^2$ or $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 > R^2$ (depending on the sign of radius of curvature) would be exactly equal to the fluid volume present.

Two common approaches to locate the interface are the analytical and the numerical iterative methods (bisection method for example). Analytical solution is accurate but computationally intensive, especially when it is in 3D. Bisection method is a numerical iterative method and its iteration is stopped when the error is below a predefined limit. A new numerical approach is proposed to locate the interface named the Sorted-Surface-Constant method.

3.3.2.4 Bisection Method

In the bisection method, the initial upper and lower bounds for finding the solution is obtained by estimating the planar constant at every vertices of the cell, which is four in 2D and eight in 3D. If the upper bound of the planar constant is a_u and lower bound is a_l then, the intermediate value is the average of these two. The interface is constructed using the average planar constant obtaining a new signed error. Then either a_l or a_r is set to the intermediate value based on the sign of the error. These steps are repeated until absolute error is less than a predefined limit. Even though this method is iterative, it takes lesser computational effort than analytical solution in 3D and convergence is ensured. Therefore this method has been successfully used.

3.3.2.5 Sorted Surface Constant Method for PLIC

Given the interface normal vector, planar constant a is estimated using a new non-iterative numerical method called the Sorted Surface Constant (SSC). Figure 3.7 shows PLIC construction using this method. For illustration uniformly distributed points inside the cell are colored according to their planar constant values. SSC method works for structured, unstructured grids in 2D and 3D with the same efficiency given boundaries are well defined to determine if a point is inside or outside the fluid volume. This method is developed based on the principle that planar constants in

normal direction vary linearly.

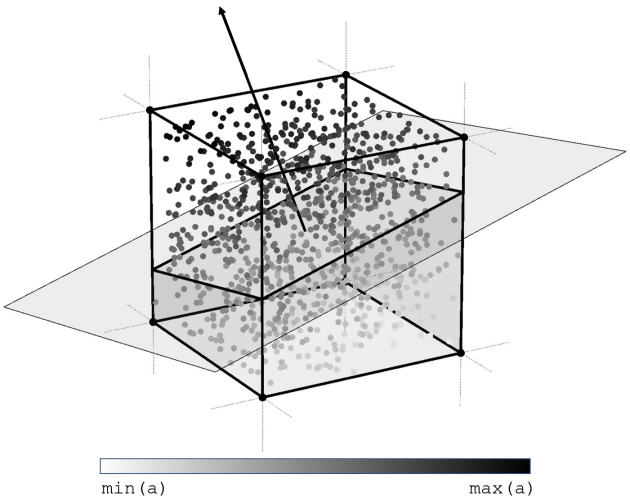


Figure 3.7: Estimation of planar constant a using the Sorted Surface Constant (SSC) method for the surface cell given the interface normal and the f-value.

The algorithm to apply SSC method for the PLIC in 2D and 3D follows.

1. Fill the surface cell uniformly with N points with a blue noise distribution.

2. Calculate planar constants for all N points by simply substituting coordinates of each points in Eqn. (3.34).

3. Sort the planar constants in ascending order and find fN^{th} planar constant.

If fN is not an integer, find weighted average of $\lfloor fN \rfloor^{th}$ and $\{\lfloor fN \rfloor + 1\}^{th}$ planar constants. Here '$\lfloor\rfloor$' is the floor function which gives the largest integer smaller than the next integer.

$$a = -(n_x x_i + n_y y_j + n_z z_k) \tag{3.34}$$

3.3.2.6 Sorted Surface Constant Method for PCIC

The sorted surface constant (SSC) method for PCIC is depicted in the Fig. 3.8. The first step is to fill the cell with random points of blue noise distribution. Then all points are mapped to the

PLIC normal line passing through the PLIC mid point. For a given point, (example: 1 or 2 in Fig. 3.8) the mapped point on the normal line is located at distance R from it. The mapped points are then sorted based on the surface-constant 'a' which is the signed distance from the midpoint to the mapped point to locate the interface. Now while mapping, there are two possible location along the normal. If R is negative, the point with minimum surface constant value is chosen else the maximum. After sorting, we find the fN^{th} (surface constant), which gives the sphere's center. Thus, the final equation of $\boldsymbol{C}$ (Eqn. 3.27) is achieved.

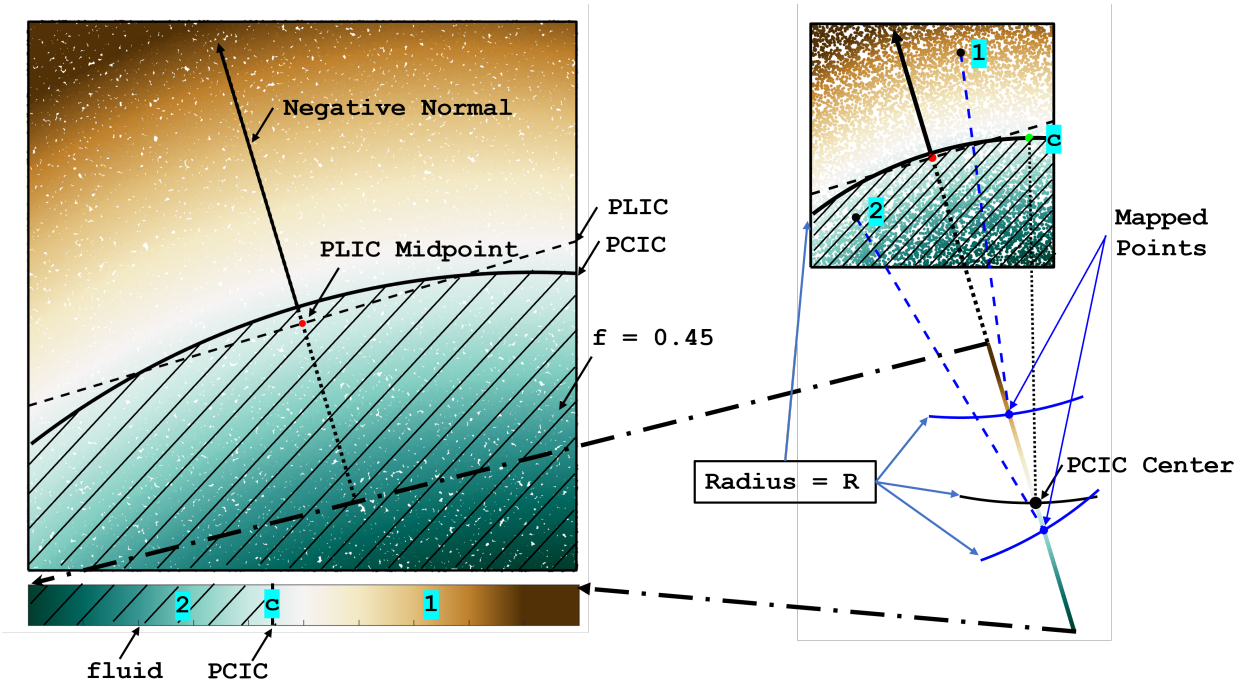


Figure 3.8: Sorted Surface Constant (SSC) method for PCIC. Each N points inside cell is mapped to the normal through the PLIC midpoint such that intersecting point is at distance R. Points are then sorted (color) according to the signed distance(surface constant value) from PLIC midpoint to the intersecting point (example: points 1, 2). fN^{th} surface constant provides the center of circle 'c', making fraction of number of points inside PCIC to be f.

There is an alternative approach for the SSC, which is SSC*. The steps for SSC* is the following.

1. Make N uniformly distributed random points inside a unit square with blue noise distribution but sorted in positive y direction. Random points need to fill a larger volume or area than the cell.For example: a square or cube with a side length equal to the length of the cell's largest diagonal.

2. Rotate and orient points such that sorted direction align with the normal direction.

3. Remove points outside the cell ending up with M points.

4. Find $(fM)^{th}$ point's planar constant.

If fM is not an integer, find weighted average of $\lfloor fM \rfloor^{th}$ and $\{\lfloor fM \rfloor + 1\}^{th}$ planar constants. Figure 3.9 shows a $\boldsymbol{C}$ construction with spherical surface. Parts of $\boldsymbol{C}$ surface would lie on both sides of equivalent $\boldsymbol{L}$ as both have same f value in comparison with Fig. 3.7.

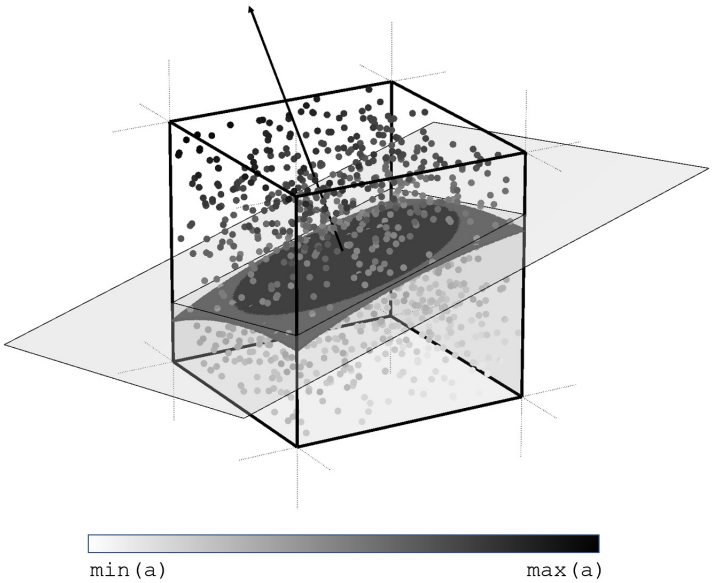


Figure 3.9: Piecewise Circular Interface Calculation (PCIC) in 3D. The rectangular plane is the PLIC interface, and the spherical surface has some portion above and below the plane to have the same f-value. For each point in the cell, 'a' is the surface-constant, signed distance from the centroid of $\boldsymbol{L}$ to the intersection of normal through centroid and of a sphere of signed radius R (mean radius of curvature) centered at that point.

The computational speed of SSC is depended on the speed of sorting the array (surface constant 'a'). This is a one step process. It does not require to solve any equations as in the analytical case, which usually gets complicated in 3D and for unstructured and curvilinear systems. SSC is flexible

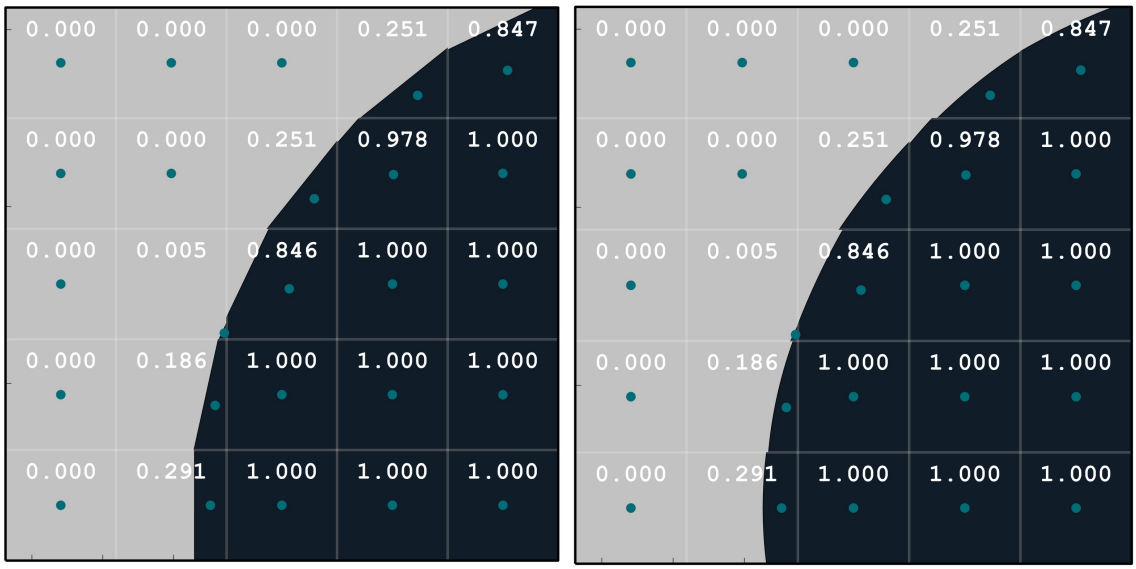
because speed vs. accuracy is dependent only on number of points N, which can be customized. While using large number of points, SSC* may be more efficient than SSC as sorting is required in the latter.

3.3.3 PLIC and PCIC

Figure 3.10a depicts the reconstructed fluid domain along the interfaces using PLIC and Fig. 3.10b shows the fluid domain constructed with PCIC. It can be seen that the PCIC interface constructions are more accurate and the radius of curvatures are close to the true value (circle of radius is 0.15 m on cell size of 1/32 $\times$ 1/32 m).

3.4 Lagrangian Particles and Center of Mass Tracking

Fluid is advected after the interface reconstruction. To advect in VCLP, the fluid inside the cell is represented using Lagrangian particles (LPs) as concentrated mass points. A template of fluid mass points generated with blue noise distribution is used to represent the fluid with uniform density. Here, LPs representing the fluid are advected cell by cell, and after advection of one cell, the same LPs template is used for the next cell advection. This is an important difference to the common Lagrangian approaches, where the entire fluid domain or interface are represented simultaneously by LPs and all LPs are tracking through the simulation time, requiring very large number of LPs proportional to the domain size for the same resolution. VCLP uses only one blue noise template of LPs with custom number of particles, and hence only a total of 100 to 1000 LPs are required for the entire simulation irrespective of the domain size in 2D, and 1000 to 8000 LPs in 3D.



(a) PLIC (b) PCIC

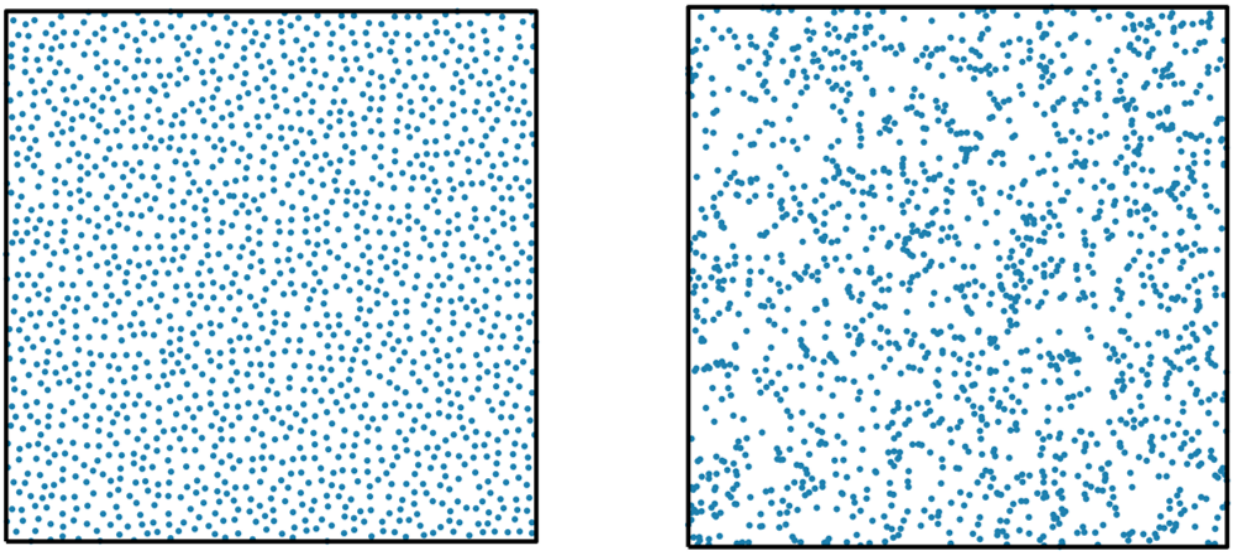
Figure 3.10: (a) Reconstructed PLIC interfaces and (b) Reconstructed Piecewise Circular Interface Calculation (PCIC), using f values. Center of mass locations are shown as dots. Fluid body is circular with a radius of 0.15 inside a unit square domain with a resolution of 32×32.

3.4.1 Lagrangian Particle Creation

The way in which the LPs are distributed is important, because if they are not uniformly distributed or not isotropic, advection error will increase. As mass particles represent the fluid with uniform density, a perfectly uniform distribution without any spatial bias is the best scenario. One possibility is divide the cell into rectangles or hexagons and place the particles at the vertices. However, they have directional bias. Other possibility is to generate uniformly distributed random numbers as a white noise for the coordinates of the points inside the cell. Nevertheless, this also doesn't ensure uniform distribution locally and so, not an ideal representation for the fluid with uniform density leading to an another possibility to use blue noise or Poisson disk sampling [83]. The difference between these sampling techniques can be seen in Fig. 3.11. The blue noise have no clusters and every small regions looks similar but they feature tightly packed random points with a minimum allowed distance between them.

A blue noise template of LPs is created once and reused in each cell to advect the fluid. If a

finer resolution is opted the a template with larger number of LPs or density is generated. It is expected that LPs distributed as blue noise is more uniform than LPs as white noise. Figure 3.11 shows blue noise generated by a proposed method named J-method with $N = 40^2$ on the left side, and white noise on right. It can be seen that blue noise representation of the fluid with uniform density is free from directional bias, which is a concern while using white noise template. The template of LPs is a square area (2D) or cubic volume (2D) with unit dimension located at the origin. The generation procedure for white noise and blue noise is given below.



(a) Blue Noise.

(b) White Noise.

Figure 3.11: Comparison of blue noise template generated by J-method and white noise using uniform random numbers.

3.4.1.1 White Noise

White noise template of N LPs along each dimension inside a unit square or cubic cell is done as follows.

- generate a 2-tuple of N^2 uniformly distributed random numbers in 2D or a 3-tuple N^3 uniformly distributed random numbers in 3D , which are the position coordinates of LP.

3.4.1.2 Blue Noise

Blue noise template with N particles along each dimension using the J-method is as follows.

- Estimate the radius r, where $r = \dfrac{1}{\sqrt{2}N}$ in 2D and $r = \dfrac{\sqrt{3}}{2N}$ in 3D.
- Divide the cell equally with $\delta x = \delta y = \dfrac{1}{N}$ creating N^2 square cell division in 2D and into N^3 cell divisions in 3D with $\delta x = \delta y = \delta z = \dfrac{1}{N}$
- Generate 2-tuple of N^2 uniformly distributed random numbers multiplied by δx in 2D and 3-tuple of N^3 uniformly distributed random numbers multiplied by δx and place one LP inside each cell divisioin using the random numbers.
- Check through all cell divisions if shortest distance between the LPs inside the neighbor cell is less than r, in which case delete the LP in the cell division (not the LP in neighbor cell).
- Consider the cell division from which LP was deleted in the previous step. Fill those cell divisions with 30 to 100 random particle in 2D and 200 to 1000 in 3D and remove newly generated LPs that are within distance r from the neighbors. Select just a single LP from the remaining set of newly generated LPs that makes sure every cell division have one LP and shortest distance between any two LPs are larger than r.

3.4.1.3 Modifying the Template for Surface Cells

After blue noise LP template with N LPs along each dimension is created, the template is used to populate the fluid region in every cells.

Template is used to fill the cell space completely or partially after which LPs outside the PCIC construction $\boldsymbol{C}$ are removed representing the fluid region. Figure 3.12 shows the modification of blue noise template on the left side into a representation of fluid in a surface cell with a PCIC interface. The blue noise LPs 2D template have $N = 25$ per dimension.

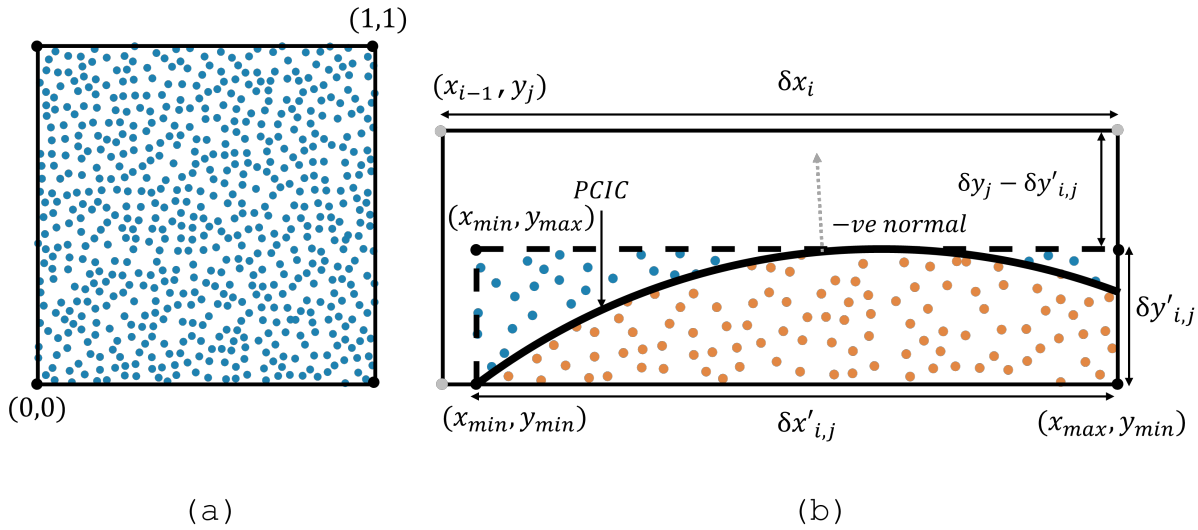


Figure 3.12: *Modification of the LPs template*: (a) Initial Lagrangian particles unit square template with $N = 25$ particles per cell dimension uniformly distributed (b) Region of particles from the template is scaled in to rectangle that fills fluid region with no change in the aspect ratio. The orange colored LPs represent the fluid inside PCIC reconstruction. The blue colored LPs outside the interface are removed.

The blue noise LPs template fills only the part of the cell with fluid. A limiting cell subdomain is constructed to improve the accuracy and efficiency in the calculations pertaining to advection. The subdomain is the smallest rectangle that can accommodate the fluid region. The subdomain boundary is shown by the dashed line in the Fig.3.12. The subdomain keeps the same aspect ratio. Therefore most of the LPs from the template, could be used to represent the fluid even in case of a surface cell with a small f value.

The scaling of template by keeping the aspect ratio is possible since blue noise spatial distribution properties is not lost, however if there is a change in the aspect ratio as in the case of Fig.3.12, the following (2D) Eqns. 3.35 and 3.36 are used to maintain the blue noise spatial distribution properties with maximum possible LP representing the fluid. T_x and T_y are the array of coordinates of LP in the template and $x_{min}, x_{max}, y_{min}$, and y_{max} are the bounds of the subdomain, the smallest rectangle that fluid can fit in. R_x and R_y are final location coordinates of LPs after the

transformation.

$$R_x(i,j) = \begin{cases} T_x \delta x'_{i,j} + x_{min}(i,j) & \textit{if}\, \beta \geq 0 \quad \text{where,}\ \ 0 \leq T_y \leq \dfrac{\delta y'_{i,j}}{\delta x'_{i,j}} \\ T_x \delta x'_{i,j} + x_{min}(i,j) & \textit{if}\, \beta < 0 \quad \text{where,}\ \ 0 \leq T_x \leq \dfrac{\delta x'_{i,j}}{\delta y'_{i,j}} \end{cases} \tag{3.35}$$

$$R_y(i,j) = \begin{cases} T_y \delta y'_{i,j} + y_{min}(i,j) & \textit{if}\, \beta \geq 0 \ \text{where,}\ \ 0 \leq T_y \leq \dfrac{\delta y'_{i,j}}{\delta x'_{i,j}} \\ T_y \delta y'_{i,j} + y_{min}(i,j) & \textit{if}\, \beta < 0 \ \text{where,}\ \ 0 \leq T_x \leq \dfrac{\delta x'_{i,j}}{\delta y'_{i,j}} \end{cases} \tag{3.36}$$

where,

$$\beta = \delta x'_{i,j} - \delta y'_{i,j},$$
$$\delta x'_i = x_{max}(i,j) - x_{min}(i,j)$$
$$\delta y'_j = y_{max}(i,j) - y_{min}(i,j)$$

Using Eqn. 3.27 of $\boldsymbol{C}$ and normal direction, point masses representing fluid (orange) are separated from other LPs in the subdomain and retained. A function called elimination function $E(\boldsymbol{p}, \boldsymbol{n})$ as defined in Eqn. (3.37) is used to determine whether a LPs are retained or not. LPs are eliminated if the elimination function value is negative. Here sgn stands for signum function.

$$E(\boldsymbol{p}, \boldsymbol{n}) = \begin{cases} \text{sgn}\left\{n_x p_x + n_y p_y + n_z p_z + a\right\} & : \text{for } \boldsymbol{L} \\ \text{sgn}\left\{(p_x - x_0)^2 + (p_y - y_0)^2 + (p_z - z_0)^2 - R^2\right\}\ \text{sgn}\left\{\nabla \cdot \boldsymbol{n}\right\} & : \text{for } \boldsymbol{C} \end{cases} \tag{3.37}$$

After elimination, the LPs representing the fluid inside the cell space $s(i,j,k)$ at time t_m before the CM correction is given by $\tilde{\boldsymbol{\Lambda}}(i,j,k,m)$. The X, Y, and Z coordinates of LPs are $\tilde{\Lambda}_x(i,j,k,m)$, $\tilde{\Lambda}_y(i,j,k,m)$, and $\tilde{\Lambda}_z(i,j,k,m)$. The LPs after CM correction is denoted by $\boldsymbol{\Lambda}$ and its calculation is provided later in the next section.

Figure 3.13 shows an overlapped image of transformed LPs using blue noise template with PLIC construction of a part of circular fluid domain. It can be seen that the number of LPs are

good enough even when f value is small, even it mostly have higher resolution than cells with larger f value, but depends also on the aspect ratio of the smallest rectangle the fluid can fit it.

3.4.2 Center of Mass Tracking

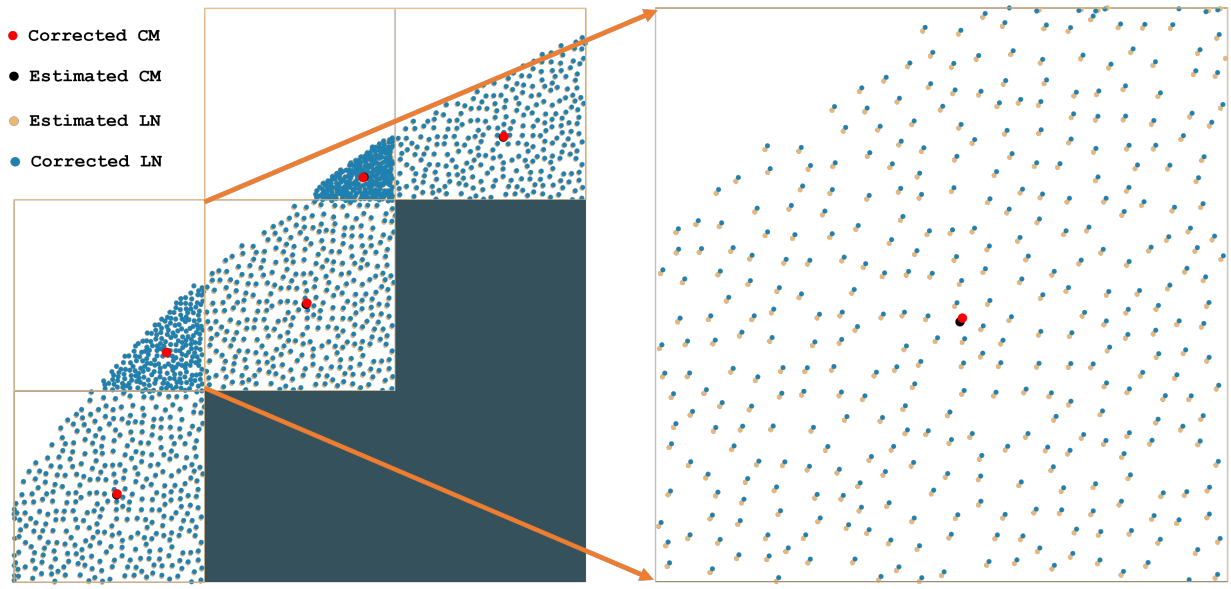


Figure 3.13: *Lagrangian Particles*: The fluid region inside the surface cells represented by uniformly and randomly distributed LPs as a blue noise per cell. The effect of subdomain approach to increase LPs resolution inside cells with smaller f values are shown. The locations of LPs (LN) before the CM correction are orange colored LPs and after correction is the blue colored dots as seen in the zoomed view, the right panel.

The accuracy of an interface reconstruction is increased if we could use extra information about the fluid in the cell. Tracking of the center of mass (CM) is such an idea following the Lagrangian approach. In theory if the grid cell size is infinitesimal, this converges to pure Lagrangian method. So for control volumes, tracking CM in Lagrangian way is essentially tracking the entire fluid whole time, but with minimal number of points, one per cell. This is the one of the main idea behind the VCLP method. During an interface reconstruction CM mostly will not be match exactly due to different reasons, mainly due to the simplification of the interface into PLIC or PCIC, that

too constructed from the mean slope and curvature that itself is an approximation. These errors make the fluid move slower or faster in the wrong direction, depending on the errors and velocity-field. Figure 3.13 shows the CM of cells after reconstruction (black) and the red dot shows the corrected CM. All the LPs are corrected with this change as seen in the zoomed in image. Even though the change is very small, results shows that it is not negligible over hundreds or thousands of time steps which is the usual case for a simulation. The importance of CM correction could be visualized from the following illustration Fig. 3.14, showing a scenario where a fluid region inside a cell with a vertical velocity u_y not advected correctly.

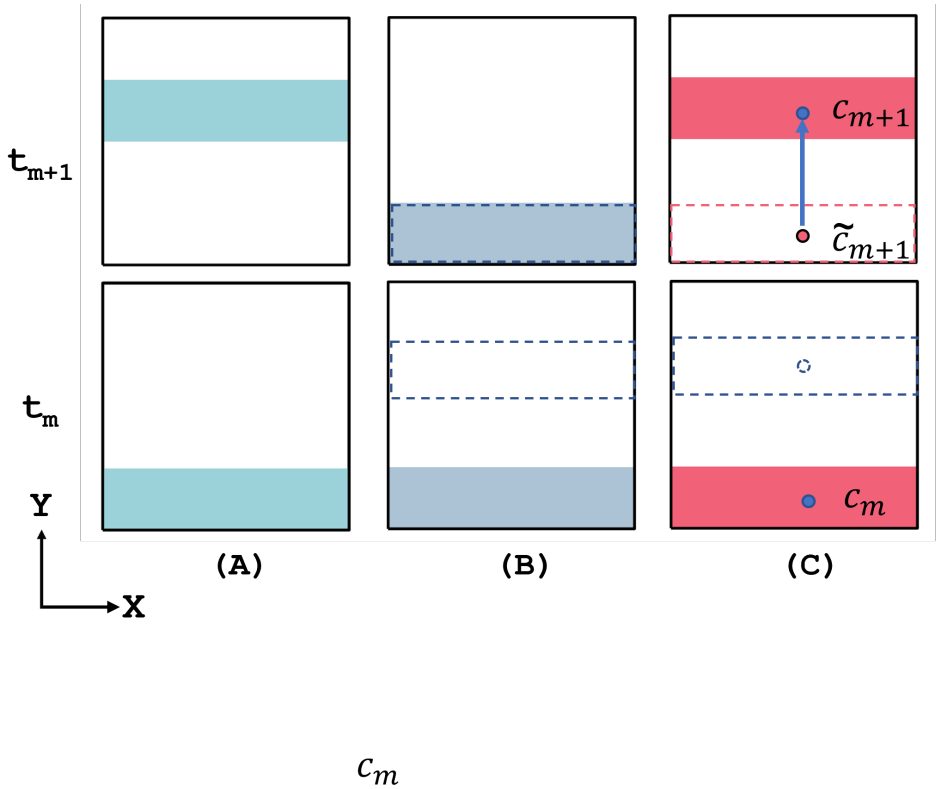


Figure 3.14: *Center of mass (CM) tracking*: A rectangular fluid body (A t_m) is inside a square cell of 1 m dimension with a velocity field u_y= 0.5 m/s. The ideal state of fluid body at the start (lower panel) and after one second (upper panel) is shown in (A). (B) the numerical simulation of fluid flow for a CFL value of 0.5 without CM tracking. (C) the numerical simulation of fluid flow for a CFL value of 0.5 with CM tracking. Blue dot is the true CM ($\boldsymbol{c}$) which is tracked in the Lagrangian way, while red dot, the estimated CM ($\tilde{\boldsymbol{c}}_{m+1}$). The reconstructed fluid region is translated to match the true CM shown by the arrow.

Consider a rectangular fluid body inside a square cell of 1 m dimension with a velocity field u_y= 0.5 m/s as depicted in Fig. 3.14 (A, lower panel). The ideal state of fluid body after one second is shown in (A, (top panel). Case (B) and (C) show the surface tracking simulation results with and without CM tracking respectively for a CFL value of 0.5, i.e., for a time step interval of 1 s. For cases (B) and (C), lower panel shows the fluid body's initial condition (m^{th} time step). The dashed line is the location of the fluid region inside cell immediately after the advection being in the same time step. Note that for case (C), the CM location after advection is also known (dashed dot). In the next time step, after 1 s, the available information of case (B) are the f value (0.25), the cell dimensions, velocity field (u_y= 0.5 m/s), time step duration (1 s), and the interface normal (estimated). For case (C), the CM location is also known additionally. The reconstructed fluid body at t_{m+1} with available information for case (B) is shown in the top panel, and it is same as the lower panel, meaning that the fluid advection is not correct. For case (C) at t_{m+1}, the initial reconstructed fluid body (red dashed line) is incorrect ans same as case (B). However, from the difference of the estimated CM ($\tilde{\boldsymbol{c}}_{\boldsymbol{m+1}}$, red dot) of the fluid cell and the true CM of the new time step, t_{m+1} (blue dot), the fluid region inside the cell is translated so as to match CM. Thus, the numerical result is matched with the ideal case in (A). This shows the advantages of using CM tracking in the VCLP method.

3.4.2.1 Estimation of the Center of Mass

During the time step t_m, there are two CMs for the cell space $s(i,j,k)$, the tracked CM known at t_{m-1} denoted as $\boldsymbol{c}(i,j,k,m)$, and the estimated CM after the reconstruction given by $\tilde{\boldsymbol{c}}(i,j,k,m)$. The LPs representing the fluid inside the cell space $s(i,j,k)$ at time t_m before the CM correction is given by $\tilde{\boldsymbol{\Lambda}}(i,j,k,m)$. The X,Y, and Z coordinates of LPs are $\tilde{\Lambda}_x(i,j,k,m)$, $\tilde{\Lambda}_y(i,j,k,m)$, and $\tilde{\Lambda}_z(i,j,k,m)$. The LPs after CM correction is denoted by $\boldsymbol{\Lambda}$. Estimated CM location ($\tilde{\boldsymbol{c}}$) is found using the Eqn. 3.38, where $\overline{\boldsymbol{\Lambda}}$ is the average of the positions all LPs inside the cell space $s(i,j,k,m)$.

$$\tilde{\boldsymbol{c}}(i,j,k,m) = \overline{\boldsymbol{\Lambda}}(i,j,k,m) \tag{3.38}$$

Once $\tilde{\boldsymbol{c}}$ is known, the positions of the LPs representing the fluid are corrected using the following Eqn. 3.39.

$$\boxed{\boldsymbol{\Lambda}(i,j,k,m) = \tilde{\boldsymbol{\Lambda}}(i,j,k,m) + \boldsymbol{c}(i,j,k,m) - \tilde{\boldsymbol{c}}(i,j,k,m)} \tag{3.39}$$

3.4.2.2 Mass Distribution

When the fluid inside the cell is represented by LPs, each LP would represent a fraction of the total mass of that fluid within the cell. That mass denoted by m_p is estimated using Eqn. 3.40, where $N_{\boldsymbol{\Lambda}(i,j,k,m)}$ is the number of LPs in $\boldsymbol{\Lambda}(i,j,k,m)$.

$$m_p(i,j,k,m) = \rho \frac{|s(i,j,k,m)| f(i,j,k,m)}{N_{\boldsymbol{\Lambda}(i,j,k,m)}} \tag{3.40}$$

3.5 The Advection of Fluid

After the interface reconstruction as explained in section 3.3 and the correction using CM, the next step is to advect the LPs representing the fluid according to the velocity field. VCLP use one of two numerical advection schemes, Taylor's second-order numerical advection scheme or a new proposed scheme named Tracking-Journey-in-Spiral (TJSM) method to advect the fluid.

3.5.1 Numerical Advection Schemes

In this section the numerical advection schemes that can be used to advect LPs are explained. Consider a single LP whose position at time step t_m is denoted by $\boldsymbol{p}_m = p_x(m)\hat{i} + p_y(m)\hat{j} + p_z(m)\hat{k}$. The velocity and acceleration is denoted by $\boldsymbol{u}(\boldsymbol{p}_m, t_m)$ and $\boldsymbol{a}(\boldsymbol{p}_m, t_m)$, respectively with time step $\Delta t_m = t_{m+1} - t_m$.

3.5.1.1 Taylor's First Order and Second Order Methods

The simplest numerical advection scheme used in CFD codes is the first order advection given by Eqn. (3.41). It is also known as Euler's method or Taylor's first order method. The first order advection scheme is implemented when the velocity field is known, but not the acceleration field. This is mostly the case as the acceleration field is not available from the solution of N-S equation.

$$\boldsymbol{p}_{m+1} = \boldsymbol{p}_m + \boldsymbol{u}(\boldsymbol{p}_m, t_m)\Delta t_m \tag{3.41}$$

To bring the acceleration into Eqn. 3.41,Taylor series is applied and the resultant Taylor's second-order method is Eqn. (3.42).

$$\boldsymbol{p}_{m+1} = \boldsymbol{p}_m + \boldsymbol{u}(\boldsymbol{p}_m, t_m)\Delta t_m + \frac{1}{2}\boldsymbol{a}(\boldsymbol{p}_m, t_m)\Delta t_m^2 \tag{3.42}$$

VCLP uses Taylor's second-order method as this can advect LPs more precisely.

3.5.2 Tracking Journey In Spiral Method (TJSM)

Tracking Journey In Spiral Method (TJSM) is a proposed numerical advection scheme to integrate the equation of motion. This scheme tries to capture rotational effects inside a velocity field which may be more intuitive for fluid flows where vorticity is present. Instead of a linear motion, TJSM assumes that the particle undergoes a circular motion with a constant angular velocity during the time step intervals. In the higher-order TJSM, thus particle could move following a spiral path, hence the name. The derivation of TJSM is given below.

3.5.2.1 Derivation of TJSM

In a 3D domain, the position, velocity and acceleration of the particle at time step t_m within Δt_m are $\boldsymbol{p}_m$, $\boldsymbol{u}_m$, and $\boldsymbol{a}_m$. This is written as Eqn. 3.43.

$$\begin{aligned}
\boldsymbol{p}_m =& x(t_m)\hat{i} + y(t_m)\hat{j} + z(t_m)\hat{k} \\
\boldsymbol{u}_m =& u_x(\boldsymbol{p}_m, t_m)\hat{i} + u_y(\boldsymbol{p}_m, t_m)\hat{j} + u_z(\boldsymbol{p}_m, t_m)\hat{k} \\
\boldsymbol{a}_m =& a_x(\boldsymbol{p}_m, t_m)\hat{i} + a_y(\boldsymbol{p}_m, t_m)\hat{j} + a_z(\boldsymbol{p}_m, t_m)\hat{k}
\end{aligned} \tag{3.43}$$

Now impose a small time duration much smaller than the time step Δt_m, δ_t for which the particle is advected in a straight line. The position and velocity of the advected particle after δ_t are given by $\boldsymbol{p}_{m'}$ and $\boldsymbol{u}_{m'}$ (Eqn. 3.44). From the Taylor series expansion, their approximate values can be

evaluated as shown in .

$$\begin{aligned} \boldsymbol{p}_{m'} &= \boldsymbol{p}_m + \boldsymbol{u}_m \delta_t \\ \boldsymbol{u}_{m'} &= \boldsymbol{u}_m + \boldsymbol{a}_m \delta_t \end{aligned} \tag{3.44}$$

Now we assume that during δ_t particle is undergoing a uniform circular planar motion which means then there would be a center of rotation ($\boldsymbol{c}_r$), a plane of rotation, a radius (r), and an angular velocity ($\boldsymbol{\omega}$). The center of rotation should pass through the plane which is parallel to the planes that can carry both $\boldsymbol{u}_{m'}$ and $\boldsymbol{u}_m$ and passes through the point $\boldsymbol{p}_m$. We name this as plane $P1$. The $\boldsymbol{c_r}$ would also pass through a plane with normal as $\boldsymbol{u}_m$ and passing through point $\boldsymbol{p}_m$, plane $P2$. Finally $\boldsymbol{c_r}$ passes also through a plane with normal as $\boldsymbol{u}_{m'}$ and through point $\boldsymbol{p}_{m'}$, plane $P3$. The intersection of $P1$, $P2$, and $P3$ uniquely identify the center of rotation, $\boldsymbol{c_r}$. The equations for the three planes are given by Eqn. 3.45 to 3.47.

$$P1 : (\boldsymbol{u}_m \times \boldsymbol{u}_{m'}) \cdot (\boldsymbol{x} - \boldsymbol{p}_m) = 0 \tag{3.45}$$

$$P2 : \boldsymbol{u}_m \cdot (\boldsymbol{x} - \boldsymbol{p}_m) = 0 \tag{3.46}$$

$$P3 : \boldsymbol{u}_{m'} \cdot (\boldsymbol{x} - \boldsymbol{p}_{m'}) = 0 \tag{3.47}$$

Substituting Eqn. 3.44 and Eqn. 3.43 in Eqn. 3.45, we get Eqn. 3.48

$$\begin{aligned} P1 : (\boldsymbol{u}_m \times (\boldsymbol{u}_m + \boldsymbol{a}_m \delta_t)) \cdot (\boldsymbol{x} - \boldsymbol{p}_m) &= 0 \\ (\boldsymbol{u}_m \times \boldsymbol{a}_m) \cdot (\boldsymbol{x} - \boldsymbol{p}_m) &= 0 \end{aligned} \tag{3.48}$$

Substituting Eqn. 3.44 Eqn. 3.46 and in Eqn. 3.45, Eqn. 3.49 is obtained.

$$\begin{aligned} P3 : (\boldsymbol{u}_m + \boldsymbol{a}_m \delta_t) \cdot (\boldsymbol{x} - (\boldsymbol{p}_m + \boldsymbol{u}_m \delta_t)) &= 0 \\ (\boldsymbol{a}_m \cdot \boldsymbol{x}) - (\boldsymbol{a}_m \cdot \boldsymbol{p}_m) - (\boldsymbol{u}_m \cdot \boldsymbol{u}_m) - \delta_t (\boldsymbol{a}_m \cdot \boldsymbol{u}_m) &= 0 \\ (\boldsymbol{a}_m \cdot \boldsymbol{x}) - (\boldsymbol{a}_m \cdot \boldsymbol{p}_m) - \|\boldsymbol{u}_m\|^2 &= 0 \end{aligned} \tag{3.49}$$

The term multiplied by δ_t is neglected as it tends to zero. The three equations Eqn. 3.46, 3.48 and 3.49 where '$\|\|$' is the Euclidean norm is solved for $\boldsymbol{c_r}$. After simplification using identities we get Eqn. 3.50 for $\boldsymbol{c_r}$.

$$\boldsymbol{c}_{r,m} = \boldsymbol{p}_m + \{(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\} \frac{\|\boldsymbol{u}_m\|^2}{\|(\boldsymbol{u}_m \times \boldsymbol{a}_m)\|^2} \tag{3.50}$$

From Eqn. 3.50, Eqn. 3.51 for radius vector is obtained as indicated below.

$$\boldsymbol{r}_m = \boldsymbol{p}_m - \boldsymbol{c}_{r,m} = -\{(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\} \frac{\|\boldsymbol{u}_m\|^2}{\|(\boldsymbol{u}_m \times \boldsymbol{a}_m)\|^2} \tag{3.51}$$

Once radius vector and velocity vector is known, angular velocity vector $\boldsymbol{\omega}$ can be estimated from 3.55, by using Eqns. 3.52 and 3.53 indicated below.

$$\boldsymbol{\omega}_m = \frac{(\boldsymbol{r}_m \times \boldsymbol{u}_m)}{\|\boldsymbol{r}_m\|^2} \tag{3.52}$$

$$\boldsymbol{\omega}_m = \frac{-[\{(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\} \times \boldsymbol{u}_m] \frac{\|\boldsymbol{u}_m\|^2}{\|(\boldsymbol{u}_m \times \boldsymbol{a}_m)\|^2}}{\|(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\|^2 \frac{\|\boldsymbol{u}_m\|^4}{\|(\boldsymbol{u}_m \times \boldsymbol{a}_m)\|^4}} \tag{3.53}$$

$$\tag{3.54}$$

After simplification using identities, the angular velocity vector is obtained as

$$\boldsymbol{\omega}_m = -\frac{(\boldsymbol{u}_m \times \boldsymbol{a}_m) \|(\boldsymbol{u}_m \times \boldsymbol{a}_m\|^4}{\|\boldsymbol{u}_m\|^2 \|(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\|^2} \tag{3.55}$$

Now, the angular rotation θ during the time step and normalized vector $\boldsymbol{\omega}$ is given by Eqn. 3.56 and Eqn. 3.57

$$\theta_m = \|\boldsymbol{\omega}_m\| \Delta t_m = \frac{\Delta t_m \|\boldsymbol{u}_m \times \boldsymbol{a}_m\|^5}{\|\boldsymbol{u}_m\|^2 \|(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\|^2} \tag{3.56}$$

$$\hat{\boldsymbol{\omega}}_m = -\frac{\boldsymbol{u}_m \times \boldsymbol{a}_m}{\|\boldsymbol{u}_m \times \boldsymbol{a}_m\|} \tag{3.57}$$

Using Rodrigues' formula given by Eqn. 3.58 [84], after simplification using vector identities, we get the final advected location of the particle as given by Eqn. 3.59.

$$\boldsymbol{p}_{m+1} = \boldsymbol{c}_{r,m} + \boldsymbol{r}_m \cos\theta_m + (\hat{\boldsymbol{\omega}}_m \times \boldsymbol{r}_m)\sin\theta_m + \hat{\boldsymbol{\omega}}_m(\hat{\boldsymbol{\omega}}_m \cdot \boldsymbol{r}_m)(1-\cos\theta_m) \tag{3.58}$$

$$\boldsymbol{p}_{m+1} = \boldsymbol{p}_m + \frac{\{\langle(\boldsymbol{u}_m \times \boldsymbol{a}_m) \times \boldsymbol{u}_m\rangle\,(1-\cos\theta_m) - \boldsymbol{u}_m\|\boldsymbol{u}_m \times \boldsymbol{a}_m\|\sin\theta_m\}\,\|\boldsymbol{u}_m\|^2}{\|(\boldsymbol{u}_m \times \boldsymbol{a}_m)\|^2} \tag{3.59}$$

If $\boldsymbol{u}_m = 0$, or $\boldsymbol{u}_m \times \boldsymbol{a}_m = 0$, then Taylor's second order advection method is to be used to advect LPs (Eqn. 3.43).

3.5.2.2 TJSM in Scalar Form

The final TJSM equation advection Eqn. 3.59 is not easy to apply directly to advect LPs because of use of operators like the curl and the norm. So, it is simplified into multiple steps and presented here. Subscript 'm' representing time step is left out until the final calculation step.

- The position ($\boldsymbol{p}$), velocity ($\boldsymbol{u}$), and acceleration ($\boldsymbol{a}$) vector components are substituted into Eqns. 3.60 to 3.65, to find the intermediate parameters. $\boldsymbol{\alpha}, \beta, \gamma$ and ϵ where $\boldsymbol{\alpha}$ is a vector, and the rest are scalars obtained as

$$\alpha_x = u_y a_z - u_z a_y \tag{3.60}$$

$$\alpha_y = u_z a_x - u_x a_z \tag{3.61}$$

$$\alpha_z = u_x a_y - u_y a_x \tag{3.62}$$

$$\beta = u_x^2 + u_y^2 + u_z^2 + a_x p_x + a_y p_y + a_z p_z \tag{3.63}$$

$$\gamma = u_x p_x + u_y p_y + u_z p_z \tag{3.64}$$

$$\epsilon = \alpha_z^2 + \alpha_x^2 + \alpha_y^2 \tag{3.65}$$

- The position ($\boldsymbol{p}$), acceleration ($\boldsymbol{a}$), $\boldsymbol{\alpha}, \beta$, and γ are substituted into Eqns. 3.66 to 3.68, to get

an intermediate vector parameter, $\boldsymbol{\delta}$.

$$\delta_x = \alpha_x(p_z\alpha_z + p_y\alpha_y + p_x\alpha_x) + \gamma(a_y\alpha_z - a_z\alpha_y) + \beta(u_z\alpha_y - u_y\alpha_z) \tag{3.66}$$

$$\delta_y = \alpha_y(p_x\alpha_x + p_z\alpha_z + p_y\alpha_y) + \gamma(a_z\alpha_x - a_x\alpha_z) + \beta(u_x\alpha_z - u_z\alpha_x) \tag{3.67}$$

$$\delta_z = \alpha_z(y_0\alpha_y + x_0\alpha_x + p_z\alpha_z) + \gamma(a_x\alpha_y - a_y\alpha_x) + \beta(u_y\alpha_x - u_x\alpha_y) \tag{3.68}$$

- The position ($\boldsymbol{p}$), $\boldsymbol{\delta}$, and ϵ are substituted into Eqn. 3.69, to get $\boldsymbol{\zeta}$, a parameter related to the center of rotation.

$$\boldsymbol{\zeta} = \left(p_x - \frac{\delta_x}{\epsilon}, p_y - \frac{\delta_y}{\epsilon}, p_z - \frac{\delta_z}{\epsilon}\right) \tag{3.69}$$

- In the final step, $\boldsymbol{\delta}$, $\boldsymbol{\zeta}$, and the velocity $\boldsymbol{u}_m$ are substituted into Eqn. 3.70, to get the advected position $\boldsymbol{p}_{m+1}$ of the particle, where m is the current time step.

$$\boldsymbol{p}_{m+1} = \boldsymbol{\delta}_m + \boldsymbol{\zeta}_m \cos\left(\frac{\|\boldsymbol{u}_m\|\Delta t_m}{\|\boldsymbol{\zeta}_m\|}\right) + \hat{\boldsymbol{u}}_{\boldsymbol{m}} \sin\left(\frac{\|\boldsymbol{u}_m\|\Delta t_m}{\|\boldsymbol{\zeta}_m\|}\right) \|\boldsymbol{\zeta}_m\| \tag{3.70}$$

3.5.3 Interpolation of Velocity Field

In VCLP, the velocity of the LPs is the function of their position within the cell, however the velocity is known only at specific discretized locations relative to the cell (example: at the cell vertices). In this case, velocity at the position of LPs are interpolated using bilinear (2D) and trilinear (3D) methods. Figure 3.15 shows a LP at position (p_x, p_y, p_z) within the cell $s(i, j, k)$. Velocities at the eight vertices of cell are u_1 to u_8. V_1 to V_8 are the eight weight factors (proportional to the volumes of the blocks) used in the trilinear interpolation. Weights are estimated based on the cell dimensions and the position of the LP inside the cell.

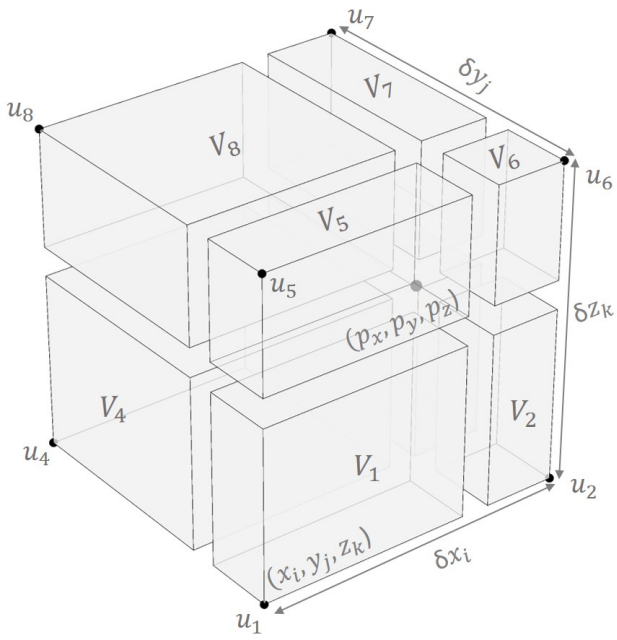


Figure 3.15: *Trilinear Interpolation of velocity*: Trilinear interpolation with partial volumes as weights to interpolate advection velocity for a LP at (p_x, p_y, p_z) using known velocities at the vertices.

The velocity of the LP located at position $\boldsymbol{p}$ is determined using Eqn. 3.71, thus

$$\boldsymbol{u}(\boldsymbol{p}) = \frac{1}{\sum_{k=1}^{k=8} V_k} \left(V_1\boldsymbol{u}_7 + V_4\boldsymbol{u}_6 + V_5\boldsymbol{u}_3 + V_8\boldsymbol{u}_2 + V_2\boldsymbol{u}_8 + V_3\boldsymbol{u}_5 + V_6\boldsymbol{u}_4 + V_7\boldsymbol{u}_1\right) \tag{3.71}$$

3.5.4 Finding the Acceleration for Advection

For second-order advection methods described in Eqn. 3.42 and Eqn. 3.59, acceleration term is also required to advect the particle. Equations to find an approximate value for acceleration is derived as follows.

The acceleration of a particle in the compact Eulerian form is written in Eqn. 3.72.

$$\boldsymbol{a} = \frac{\partial \boldsymbol{u}}{dt} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \tag{3.72}$$

when Eqn. 3.72 is expanded at each directions of the domain axes, it becomes

$$\begin{aligned} a_x = \frac{du_x}{dt} &= \frac{\partial u_x}{\partial t} + u_x\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_x}{\partial y} + u_z\frac{\partial u_x}{\partial z} \\ a_y = \frac{du_y}{dt} &= \frac{\partial u_y}{\partial t} + u_x\frac{\partial u_y}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_y}{\partial z} \\ a_z = \frac{du_z}{dt} &= \frac{\partial u_z}{\partial t} + u_x\frac{\partial u_z}{\partial x} + u_y\frac{\partial u_z}{\partial y} + u_z\frac{\partial u_z}{\partial z} \end{aligned} \tag{3.73}$$

The partial derivative terms of u_x, $\frac{\partial u_x}{\partial x}$, $\frac{\partial u_x}{\partial y}$, and $\frac{\partial u_x}{\partial z}$ in Eqn. 3.73 are derived using Eqn. 3.74.

$$\begin{aligned} -\frac{\partial u_x}{\partial x} =& \frac{(u_{x1}-u_{x2})}{\delta x_i} + \frac{(p_y-y_j)}{\delta x_i \delta y_j}E_{1x} + \frac{(p_z-z_k)}{\delta x_i \delta z_k}E_{2x} + \frac{(p_y-y_j)(p_z-z_k)}{\delta x_i \delta y_j \delta z_k})E_{0x} \\ -\frac{\partial u_x}{\partial y} =& \frac{(u_{x1}-u_{x4})}{\delta y_j} + \frac{(p_x-x_i)}{\delta x_i \delta y_j}E_{1x} + \frac{(p_y-y_j)}{\delta y_j \delta z_k}E_{3x} + \frac{(p_z-z_k)(p_x-x_i)}{\delta x_i \delta y_j \delta z_k})E_{0x} \\ -\frac{\partial u_x}{\partial z} =& \frac{(u_{x1}-u_{x5})}{\delta z_k} + \frac{(p_x-x_i)}{\delta x_i \delta z_k}E_{2x} + \frac{(p_y-y_j)}{\delta y_j \delta z_k}E_{3x} + \frac{(p_y-y_j)(p_x-x_i)}{\delta x_i \delta y_j \delta z_k})E_{0x} \end{aligned} \tag{3.74}$$

The partial derivative terms of the Y component of velocity, u_y $\frac{\partial u_y}{\partial x}$, $\frac{\partial u_y}{\partial y}$, and $\frac{\partial u_z}{\partial x}$ are given by Eqns. 3.75.

$$\begin{aligned} -\frac{\partial u_y}{\partial x} =& \frac{(u_{y1}-u_{y2})}{\delta x_i} + \frac{(p_y-y_j)}{\delta x_i \delta y_j}E_{1y} + \frac{(p_z-z_k)}{\delta x_i \delta z_k}E_{2y} + \frac{(p_y-y_j)(p_z-z_k)}{\delta x_i \delta y_j \delta z_k})E_{0y} \\ -\frac{\partial u_y}{\partial y} =& \frac{(u_{y1}-u_{y4})}{\delta y_j} + \frac{(p_x-x_i)}{\delta x_i \delta y_j}E_{1y} + \frac{(p_z-z_k)}{\delta y_j \delta z_k}E_{3y} + \frac{(p_y-y_j)(p_z-z_k)}{\delta x_i \delta y_j \delta z_k})E_{0y} \\ -\frac{\partial u_y}{\partial z} =& \frac{(u_{y1}-u_{y5})}{\delta z_k} + \frac{(p_x-x_i)}{\delta x_i \delta z_k}E_{2y} + \frac{(p_y-y_j)}{\delta y_j \delta z_k}E_{3y} + \frac{(p_y-y_j)(p_x-x_i)}{\delta x_i \delta y_j \delta z_k})E_{0y} \end{aligned} \tag{3.75}$$

The partial derivatives of the Z component of velocity u_z, $\frac{\partial u_z}{\partial x}$, $\frac{\partial u_z}{\partial x}$, and $\frac{\partial u_z}{\partial x}$are found using

Eqns. 3.76.

$$
\begin{aligned}
-\frac{\partial u_z}{\partial x} =& \frac{(u_{z1}-u_{z2})}{\delta x_i} + \frac{(p_y-y_j)}{\delta x_i \delta y_j} E_{1z} + \frac{(p_z-z_k)}{\delta x_i \delta z_k} E_{2z} + \frac{(p_y-y_j)(p_z-z_k)}{\delta x_i \delta y_j \delta z_k}) E_{0z} \\
-\frac{\partial u_z}{\partial y} =& \frac{(u_{z1}-u_{z4})}{\delta y_j} + \frac{(p_x-x_i)}{\delta x_i \delta y_j} E_{1z} + \frac{(p_z-z_k)}{\delta y_j \delta z_k} E_{3z} + \frac{(p_x-x_i)(p_z-z_k)}{\delta x_i \delta y_j \delta z_k}) E_{0z} \\
-\frac{\partial u_z}{\partial z} =& \frac{(u_{z1}-u_{z5})}{\delta z_k} + \frac{(p_x-x_i)}{\delta x_i \delta z_k} E_{2z} + \frac{(p_y-y_j)}{\delta y_j \delta z_k} E_{3z} + \frac{(p_x-x_i)(p_y-y_j)}{\delta x_i \delta y_j \delta z_k}) E_{0z}
\end{aligned}
\tag{3.76}
$$

The terms $E_0 i$ to $E_3 i$ where i takes x, y and z depending on the component axis, seen in the above equations are calculated using the set of Eqns. 3.77 given below.

$$
\begin{aligned}
E_{0i} =& (u_{i1} - u_{i2} + u_{i3} - u_{i4} - u_{i5} + u_{i6} - u_{i7} + u_{i8}) \\
E_{1i} =& (-u_{i1} + u_{i2} - u_{i3} + u_{i4}) \\
E_{2i} =& (-u_{i1} + u_{i2} + u_{i5} - u_{i6}) \\
E_{3i} =& (-u_{i1} + u_{i4} + u_{i5} - u_{i8})
\end{aligned}
\tag{3.77}
$$

Finally the partial derivative of acceleration with respect to time (Eqn. 3.73), is determined by Eqn. 3.78. This term ($\frac{\partial \boldsymbol{a}}{\partial t}$) requires the velocities from the previous time step.

$$
\frac{\partial \boldsymbol{a}}{\partial t_m} = \frac{\boldsymbol{u}_m - \boldsymbol{u}_{m-1}}{t_m - t_{m-1}} \tag{3.78}
$$

Having obtained all terms described in the Eqns. 3.73, the approximate acceleration for all LPs are obtained. Thus, using the interpolated velocity and approximate acceleration LPs are advected by implementing the Taylor's second order (Eqn. 3.42) or the TJSM (Eqn. 3.59) numerical advection schemes.

3.5.5 Comparison of the Numerical Advection Schemes in 2D

The performance of the proposed numerical advection scheme, the TJSM method is tested using one of the deformation benchmark test for the interface tracking. In this test, a circular fluid body of radius 0.15 centered at (0.5,0.75) inside a square domain of unit size with its left bottom

corner at the origin (0,0). The fluid body is inside a solenoidal velocity field that deforms the circular fluid body. Deformation test induces radical deformation and topology change of fluid bodies, which is more challenging than single vortex test (discussed later) [3]. There are sixteen vortices in the domain, and flow fields are time-reversed. Time reversed flow field means that for half of the simulation time, the fluid body undergoes deformation and then for the next half the velocity field is time-reversed causing the circular fluid to go back to its original shape and location. For the numerical case where time is discretized, final state of the fluid body is not exactly same as the starting state. Thus the similarity of the final state to the initial state is the measure of goodness for the advection method. The stream function for this velocity field is given by Rider et al. [7] as shown in Eqn. (3.79).

$$\Psi = \frac{1}{4\pi} \sin\left\{4\pi\left(x+\frac{1}{2}\right)\right\} \cos\left\{4\pi\left(y+\frac{1}{2}\right)\right\} \tag{3.79}$$

The velocity field is given by Eqns. 3.80, and Fig. 3.16 shows the velocity field at time $t = 0$. The time-varying field have its maximum magnitude of velocity at $t = 0$ and $t = T$ and zero at $t = T/2$ where it reverses.

$$\begin{aligned} u_x &= \sin(4\pi y)\sin(4\pi x)\cos\left(\frac{\pi t}{T}\right) \\ u_y &= \cos(4\pi x)\cos(4\pi y)\cos\left(\frac{\pi t}{T}\right) \end{aligned} \tag{3.80}$$

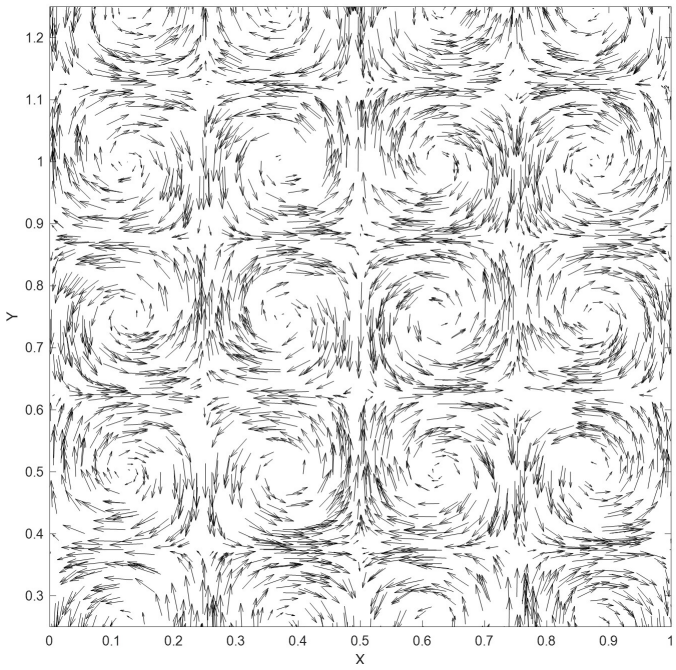
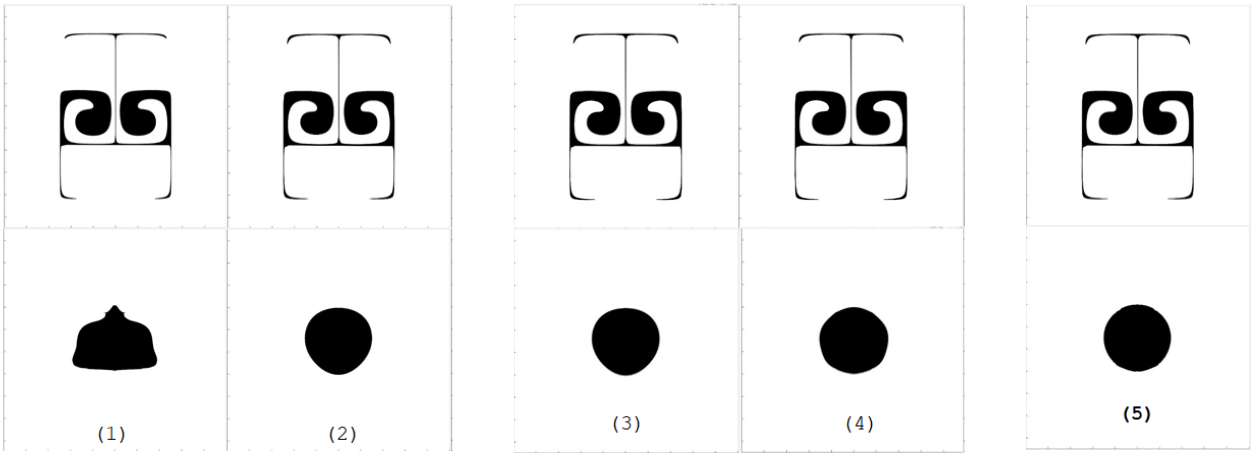


Figure 3.16: The velocity field at $t = 0$ for the deformation test of a circular fluid body of radius 0.15 centered at (0.5,0.75). There are sixteen time-varying vortices inside the unit square domain that deforms a fluid body inside. The velocity field is defined by Eqn. 3.79

.

The 2D test is conducted by putting marker particles on the circle perimeter, and each particle is advected in time using several numerical advection schemes. The numerical advection schemes tested are 1) Euler's Method, 2) Taylor 2^{nd} order, 3) Taylor 3^{rd} order, and 4) Runge-Kutta 4^{th} order, and 5) TJSM 2^{nd} order. The number of time steps is 256 for a simulation time of $T = 2$ s.

The deformation test results are depicted in the Fig. 3.17 where marker particles bound the fluid area to improve the contrast. The 2D deformation test results for the numerical advection schemes show that at $t = T/2$ difference between methods are not easily noticeable except for the T-shaped tip of the top part in the Taylor's first order method, which is shorter than other methods (see top panels of Fig. 3.17). At $t = T$, the final shape is ideally the initial circle itself, however all methods show some deviations from it. Taylor's first order have large deviation from circular shape, while other cases are closer to the initial circle. Taylor's second and third order have an



(a) (b) (c)

Figure 3.17: Deformation test results $t = 1 = T/2$ (Top), $t = T = 2$ s (Bottom). a) Taylor's first order (left) and Taylor's Second order (right) b) Taylor's third order (left) and Runge-Kutta fourth order (right) order c) TJSM second-order (error Δt^3).

elliptical shape at the bottom and the third order have slightly closer shape to the initial circle than the second order. The Runge-Kutta fourth order have an overall circular and symmetric shape, however edges are flatter over the bottom half of fluid body and is similar to a polygon like shape. The TJSM result looks very close to the initial circle and very slight deformation can be noticed at the top and bottom part. While comparing all methods qualitatively, the closeness to the circular shape is in the order 1) TJSM second order, 1) Runge-Kutta fourth order, 3) Taylor's third order, 4) Taylor's second order, and 5) Taylor's first order. So, as per the test results, the proposed advection method, TJSM performs well.

3.5.5.1 Comparison of the Numerical Advection Schemes in 3D

The deformation test in 3D is carried out in the similar way as in 2D, where the spherical fluid body of radius 0.15 centered at (0.5,0.5,0.5) is considered. The fluid body is placed inside a solenoidal time-reversing velocity field. The three numerical advection schemes compared are 1) Taylor's first order, 2) Taylor's second order, and the TJSM second order. The velocity field for

this test is given by given by Eqns. 3.81 to 3.83 for velocity components in X, Y, and Z axes.

$$u_x = 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos\left(\frac{\pi t}{T}\right) \tag{3.81}$$

$$u_y = -\sin^2(\pi y)\sin(2\pi x)\sin(2\pi z)\left(\frac{\pi t}{T}\right) \tag{3.82}$$

$$u_z = -\sin^2(\pi z)\sin(2\pi x)\sin(2\pi z)\left(\frac{\pi t}{T}\right) \tag{3.83}$$

The simulation time of $T = 12$ s is completed in 32 time steps. Figure 3.18 shows the deformation test results of the fluid body. At $t = T/2$, the fluid deformation (pointed peaks) of Taylor's first order method (a) Left panel) is shorter than the other two methods. At $t = T$, the first order have a polyhedron shape with faces bending inwards, while the second order method (panel b) have the polyhedron shape too, but more spherical. TJSM results (panel c) resembles sphere quite well, but there are some visible perturbation over the fluid body surface. Qualitatively, the resemblance of the final results to the spherical shape are in the order 1) TJSM second order, 2) Taylor's second order, and 3) Taylor's first order. So, as per the test results in 3D, the proposed numerical advection scheme, TJSM seems to perform well in comparison with the two standard methods.

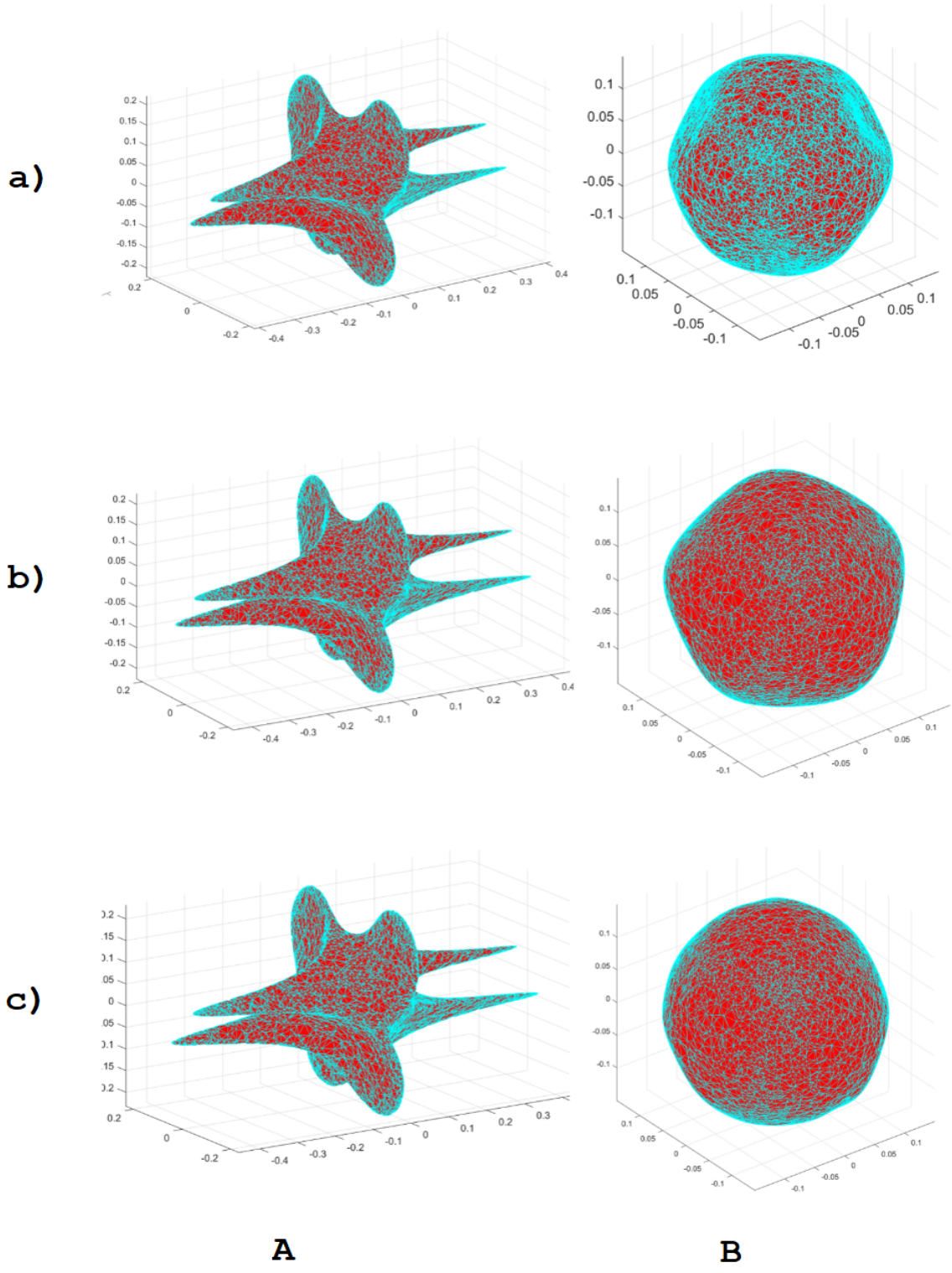


Figure 3.18: Deformation test results for three numerical advection schemes a) Euler's method b) Taylor's second order c) TJSM second order. Column *A*: $t = \frac{T}{2} = 6s$, *B*: $t = T = 12s$.

3.6 Updating Mass and Center of Mass

After the surface reconstructions, and advection of LPs are completed in a cell, next step is to update the f values and the CM locations of the cells involved in the advection process. The updating of f value and the CM is depicted in Fig. 3.19. An new copy of the computational domain Ω is created to receive the fluid mass advected, denoted by Ω^*_m. This new domain receives the advected LPs from cell-by cell and new values for the fluid mass and CM keeps updating. '*' denotes that the functions and parameters belongs to Ω^*. After advection of all the fluid in the cells, this updated domain copy, Ω^*_m becomes the next time step value, Ω_{m+1}. A typical interface

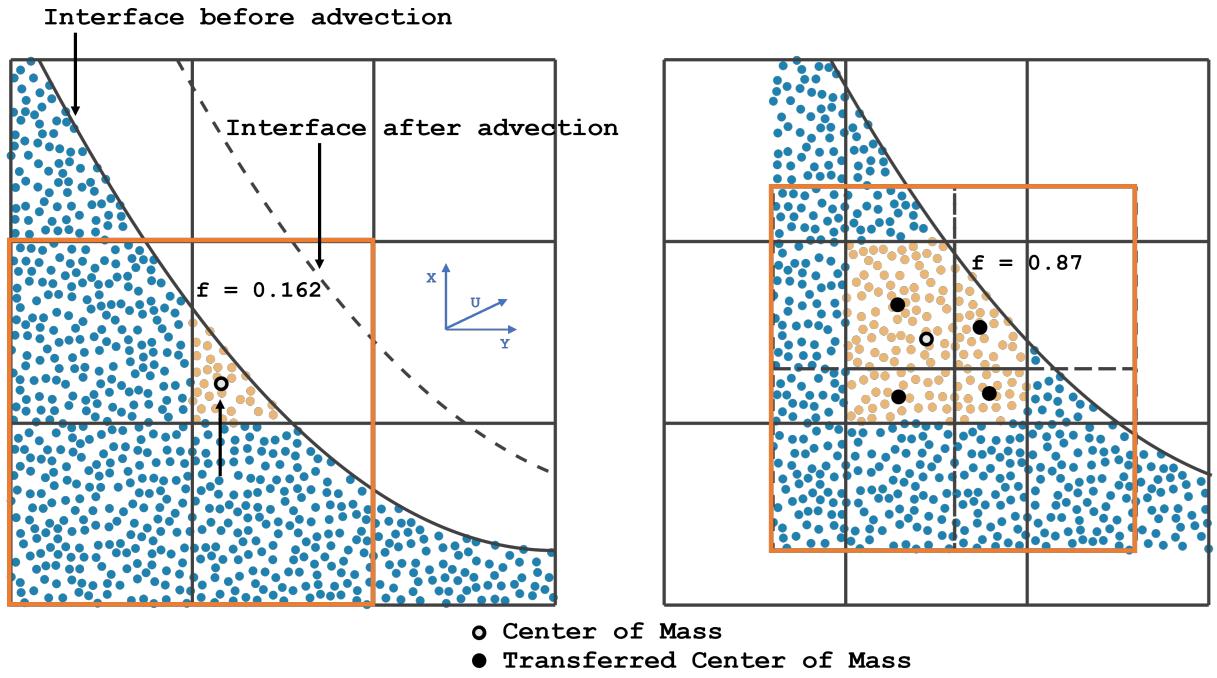


Figure 3.19: *Mass transfer and CM Update*: Left: LPs representing the fluid before advection. f- value and center of mass is shown for the surface cell(middle cell). Right: advected LPs from the middle and neighbor cells into the middle cell (orange color) and their transferred CMs. The updated f- value and CM is also shown. Note cells are advected one by one and this figure is an combined representation of the process.

before and after advection along with the LPs are shown on the left and right panels, respectively.

Transferred mass and transferred CM from three neighbor cells into the surface cell in the middle are shown on right panel of Fig. 3.19, along with the final CM location and the f value (right panel). If a cell space $s(i, j, k)$ receives n LPs from a cell with mass per point m_p as indicated in Eqn. (3.40), then the updated f value is estimated by Eqn. (3.84) as

$$f^*_{up}(i, j) = f^*(i, j) + \frac{nm_p}{\rho \delta x_i \delta y_j} \tag{3.84}$$

Finally, Eqn.3.85 provides the updated CM, where ($\boldsymbol{c}_{tr}$) is the transferred CM.

$$\boldsymbol{c}^*_{up} = \frac{\boldsymbol{c}^*_e f^* \delta x_i \delta y_j \rho + \boldsymbol{c}_{tr} n m_p}{n m_p + \delta x_i \delta y_j \rho} \tag{3.85}$$

3.7 Filters and Optimization for VCLP

The advection using LPs might result in cells with f value greater than one, which is mainly due to the fact that velocity field is discretized. When the velocity field is interpolated to advect LPs the divergence of the interpolated velocity field at certain location become non-zero. Hence, it can be said the f value larger than one at times is the result of the non-zero divergence resulting from the necessity to interpolate the discretized velocity field. As this is unavoidable, a filter is used for cells having $f > 1$ and takes the excess and redistributes mass to non-empty neighboring cells uniformly. After this, excess mass from cells with $f > 1$ is redistributed uniformly to interior cells with $f < 1$ and repeated until all f value becomes less than or equal to one, and typically it takes one or two iterations. To account for the non-conservative character arising from the interpolation of the velocity field, VCLP uses a scaling technique. If an interior cell has f value differ from one, the area or volume of the advected cell from its advected boundaries are estimated, and LPs' locations are scaled about the center of mass such that the advected cell volume or area is the same as the cell volume or area. VCLP method proposes an optimization technique named skip core optimization (SCO) to increase computational efficiency. This technique allows us to skip the computation for a portion of cells which are far from the interface.

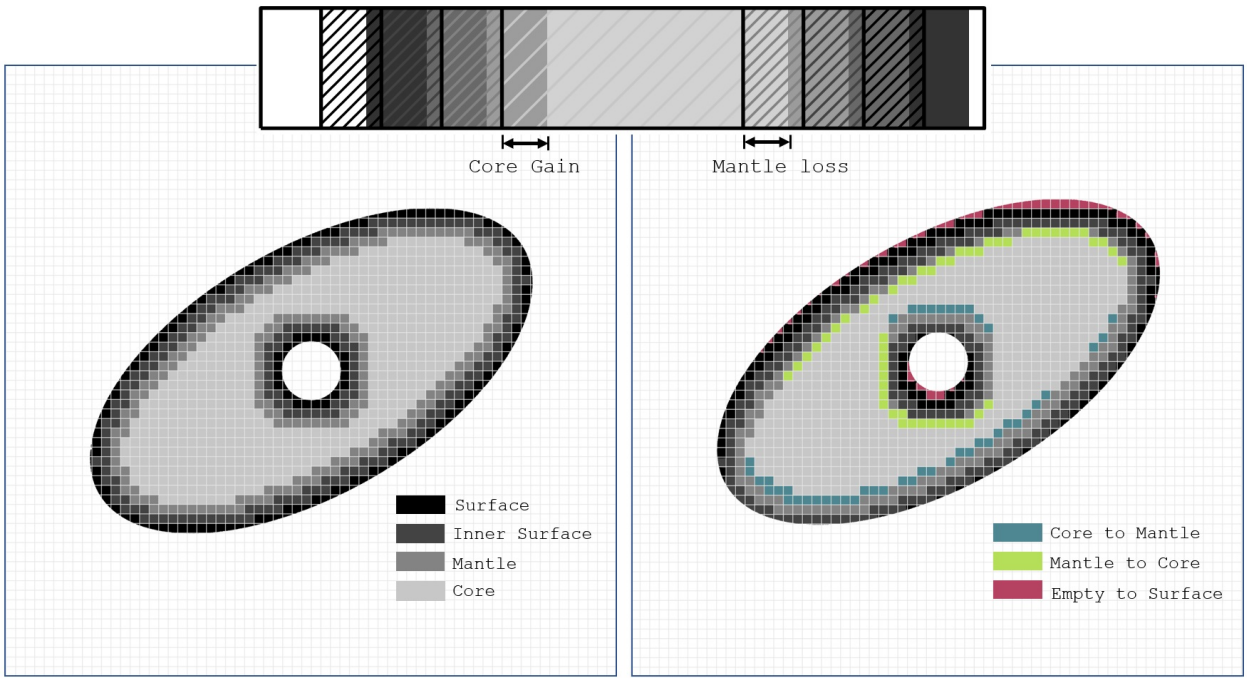


Figure 3.20: *Skip Core Optimization*: Top: Rectangular block is a simplified 1-D translation diagram with CFL = 0.75 showing if core is not advected the core gain equals mantle loss. Hatched is before advection and filled is after. Left: an ellipsoidal fluid body with a circular hole inside inside a rectangular computational domain with a grid resolution of 64×64 before advection. Right: fluid body after advection with CFL = 1 with a uniform velocity field of $\cos(3\pi/4)\hat{i} + \sin(3\pi/4)\hat{j}$. Left: Classification of cells before advection. Right: cells with changes in classification after advection and the corresponding change.

3.7.0.1 Skip Core Optimization

Figure 3.20 shows an ellipsoidal fluid body with a circular hole in it inside a square computational domain with the grid resolution of 64×64. The classification of cells can be seen in the left panel and the criteria for the classification of cells and definition of q is given in section 3.2. The inner surface and inner mantle thickness depends on q, which is closely related to the maximum CFL number. If q is one, then there needs to be only one layer of inner surface and inner mantle cells, and if it is two, there should be two layers of inner surface cells and inner mantle cells. It can be seen that the percentage of each cell type depends on the surface area to volume or surface perimeter to area ratio for 3D and 2D respectively. Figure 3.20 left panel depicts the fluid domain before advection and right panel after advection with CFL = 1, where entire fluid domain

is translated δx along direction $\cos(3\pi/4)\hat{i} + \sin(3\pi/4)\hat{j}$. q value of one means that the LPs from the surface cell cannot create any effect on the inner mantle and inner core cells. Similarly, inner surface cells cannot affect inner core cells. Vice versa is also true, which means it is enough to advect only the surface, inner surface, and inner mantle. So, to conserve the fluid mass inside the inner core cells, the mass received from the inner mantle cells should be exactly equal to the mass advected from the inner core cells. This means than if the fluid inside all cells except the inner core cells are advected, then it is enough to put back exactly the mass received from the inner mantle cells while no change happening to the fluid inside the inner core cells. This means that f value of inner mantle and inner core should remain as one after advection if the entire fluid were advected. However, if only fluid inside the surface cells are advected, inner surface, and inner mantle, those core cells will receive fluid from the mantle, and the mantle will lose some of its fluid to the interior cells. Loss and gain can be further understood from colored cells in Fig. 3.20 which shows that the inner core cells that became inner mantle cells are almost the same as the number of inner mantle cells that became the inner core cells. Core cells can interact only with the mantle cell because of the CFL limit. Since the core is filled with fluid before and after advection; the only possibility is that the net fluid received from the inner mantle by the inner core should be the same as net fluid that is meant to be given by the inner core to the mantle cells to satisfy the mass conservation, i.e., if the fluid inside surface cells, inner surface cells, and inner mantle cells alone are advected, then the extra fluid volume received by inner core cells (core gain) and the net fluid volume lost by the inner mantle cell (mantle loss) should be same. The f value of inner mantle cells and inner core cells are set to one, and their CM location is not updated for next time step. However, there will be a tiny difference in the core gain and mantle loss that results from the non-zero divergence error.

The application of filters is the final step of VCLP producing the final outputs the new f value matrix ($\boldsymbol{f_{m+1}}$) and the new CM matrices ($\boldsymbol{c_{m+1}}$). A simplification technique is proposed to minimize the computer memory storage and making VCLP code a module with similar input and output parameters as in VOF method. That is, $\boldsymbol{f_{c,m}}$ and $\boldsymbol{U_m}$ as the input parameters and $\boldsymbol{f_{c,m+1}}$ as the single output, where $\boldsymbol{f_{c,m+1}}$ is made by merging $\boldsymbol{f_{m+1}}$ matrix and $\boldsymbol{c_{m+1}}$ without creating extra

variables for CM. This is achieved via following steps.

The CM locations are normalized with respect to the cell by the Eqn. 3.86.

$$c_x^n = \frac{c_x - x_i}{\delta x_i} \quad c_y^n = \frac{c_y - y_j}{\delta y_j} \quad c_z^n = \frac{c_z - z_k}{\delta z_k} \tag{3.86}$$

The normalized CM locations $\boldsymbol{c}^n$ are then combined with f value to get f_c using the Eqn. (3.87).

$$f_c c = f + 10\lfloor 10^4 c_x^n \rfloor + 10^5 \lfloor 10^4 c_y^n \rfloor + 10^9 \lfloor 10^4 c_z^n \rfloor \tag{3.87}$$

To retrieve the normalized CM locations and f values, Eqns. 3.88 are used.

$$\begin{aligned} c_z^n &= 10^{-4} \lfloor 10^{-9} f_c \rfloor \\ c_y^n &= 10^{-4} \lfloor 10^{-5} (f_c - 10^{13} c_z^n) \rfloor \\ c_x^n &= 10^{-4} \lfloor 10^{-1} (f_c - 10^{13} c_z^n - 10^9 c_y^n) \rfloor \\ f &= f_c - 10^{13} c_z^n - 10^9 c_y^n - 10^5 c_x^n \end{aligned} \tag{3.88}$$

Subsequently, (c_x, c_y, c_z) are retrieved from Eqn. 3.86. This procedure compresses four scalar values into one, thus saving computational memory storage, but with customizable round of errors.

4. VCLP BENCHMARK TESTS AND RESULTS

To evaluate the accuracy of VCLP method for interface reconstruction and advection, five benchmark tests in 2D and one in 3D are conducted. The 2D tests are translation, rotation, single vortex, deformation, and Zalesak's test. In each of these tests, a well defined 2D fluid body inside a computational domain with specified grid resolution and dimensions is allowed to flow according to a specified solenoidal (non-divergent) velocity field. The flow is then simulated till the given time T, using the proposed method with the specified time intervals. The ideal final state of the fluid body at time T is known and the performance of the proposed method is evaluated on the basis of its qualitative and quantitative similarity with the simulation result. The translation test have a uniform velocity field, while the rotation test have a circular velocity field for a rigid body rotation. The single vortex and deformation test velocity fields can be thought as mixing, causing large variations such as stretching, tearing, and folding in the interfacial area through vorticity [7]. The single vortex test spins fluid into a filament spiraling toward the vortex center and it was introduced by Bell [85]. The deformation test introduced by Smolarkiewicz [86] has sixteen vortices inside the domain that causes fluid elements to undergo large topology changes [7]. Single vortex and deformation tests are time-reversible T-periodic tests following Leveque [87] in which flow returns ideally to the initial condition. This is achieved by multiplying velocity field by $\cos\left(\frac{\pi t}{T}\right)$. This reversal makes a quantitative comparison and evaluation of these tests possible.

In the following sections, VCLP with PLIC is written as 'VCLP-' and with PCIC as 'VCLP+.' 'VCLP+' with Skip Core Optimization is given as 'VCLP+ SC' and 'VCLP+' while using Eqn. 3.42 for advection it is shown as 'VCLP+ T2'.

4.1 Translation Test

For translation tests, a circular and a square fluid body are placed inside two different velocity fields. The fluid bodies are transported in 1000 time steps over a distance approximately five times the diameter of the fluid. The number of grid cells along the diameter or the size of the

fluid body is equal to 15. The two velocity fields are $\boldsymbol{u} = \hat{j}$ and $\boldsymbol{u} = 2\hat{i} + \hat{j}$. For this case, the VCLP results (PLIC- column f, PCIC colum-g) are compared with two well known methods, Hirt and Nichols with height function (column c) and without (column b) and Young's method with height function (column e) and without (column d). The test results are shown in Fig. 4.1.

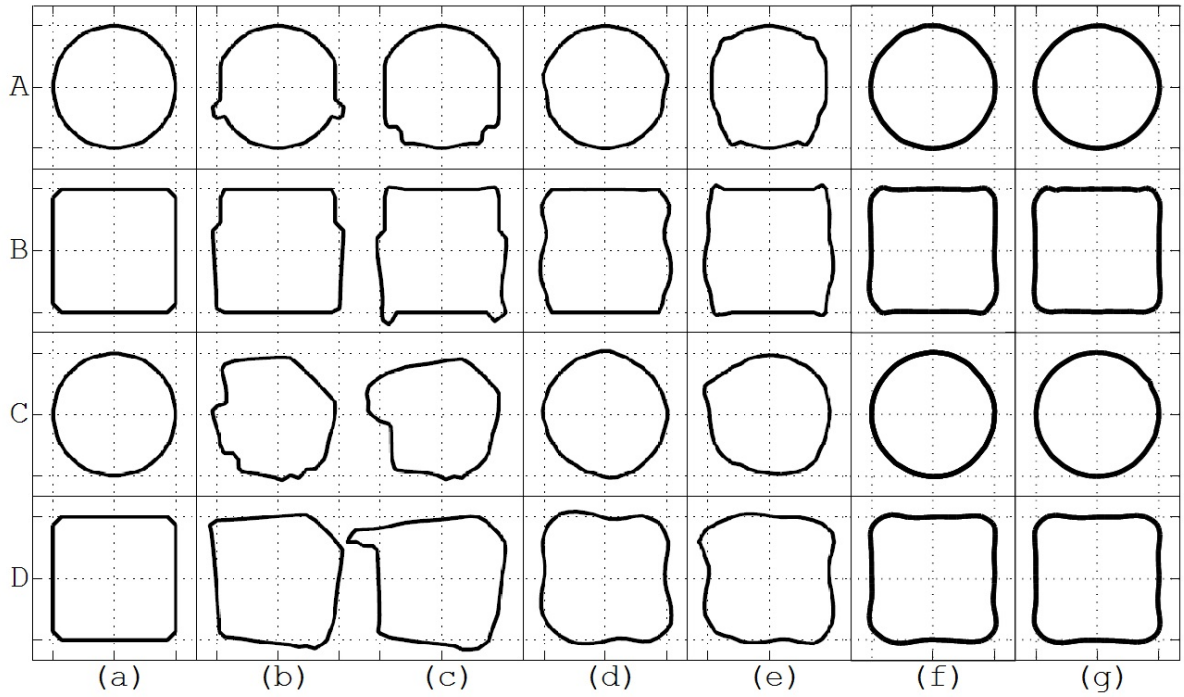


Figure 4.1: Translation of a circle (rows A and C) and a square (rows B and D) in two different velocity fields $\boldsymbol{u} = \hat{j}$ (rows A and B) and $\boldsymbol{u} = 2\hat{i} + \hat{j}$ (rows C and D). (a) Initial condition, (b) Hirt and Nichols without local height function, (c) Hirt and Nichols with local height function, (d) Young's without local height function, (e) Young's with local height function, (f) VCLP- with PLIC and (g) VCLP+ with PCIC Note: Results from column (a) to (e) are obtained from [1].

For translation velocity field $\boldsymbol{u} = \hat{j}$, for circular fluid body (panel A), VCLP- and VCLP+ match the initial circular shape without perturbation as in the other methods. Young's results without the height function is also comparable. VCLP has a very slight expansion along the horizontal axis, which is normal to the velocity field. VCLP is more symmetric across both velocity field direction and its normal, than the other methods. For square,(panel B) the corners are smoothed

in VCLP, edges are almost straight, and maintains square form. Original Hirt and Nichols (H&N) and Young's method have straight top and bottom edges, but left and right edges has deformed. Height functions causes little perturbations at the corners. For the velocity field $\boldsymbol{u} = 2\hat{i} + \hat{j}$, for circle (panel C), Young's maintains the shape and size, VCLP has maintained more circular shape. H&N method seems to lose some fluid volume. For square (panel D), VCLP results are symmetric, edges are almost straight and maintains right angle with edges. Corners are more smoothed than the vertical translational test. H&N method have straight edges, but the angles have changed and a new edge is formed at the leading region. Young's method have smoothened the edges and edges are no longer straight. The difference between VCLP- and VCLP+ is not noticeable.

4.2 Rotation Test

In rotation test, a circular fluid body of radius 0.15 m and a square shaped fluid body of side length 0.30 m are placed inside a 1 m square computational domain where a circular-anticlockwise velocity field is present. The fluid bodies undergo a rigid body rotation of 2π radian (one complete circle) during 600 time-steps. The distance between the fluid's CM and the center of rotation is 0.6 m with a grid resolution of 50×50. Ideally, after one complete cycle of rigid body rotation, the expected final state of the fluid body would be exactly same as the initial state. Thus, the accuracy of a surface tracking and advection methods can be evaluated based on the resemblance of the final and initial fluid body states. The rotation test results of the VCLP method (column f with PLIC and column g with PCIC) are presented in Fig. 4.2. The rotation test results under an ideal case (column a), H&N method without the height function (column b), H&N with the height function (column c), Young's method without the height function (column d), and Young's method with the height function (column e) are also presented.

Rigid body rotation test results with the circular fluid body (panel A) show, that the VCLP results maintain the circular shape. H&N method results have rough surface with perturbations and fails to maintain the circular shape, while Young's method have maintained circular shape except for the right side, which is the trailing edge in the rotation test. For the square shaped fluid body (panel B), VCLP results conserve fluid mass, and maintain right angles between the edges,

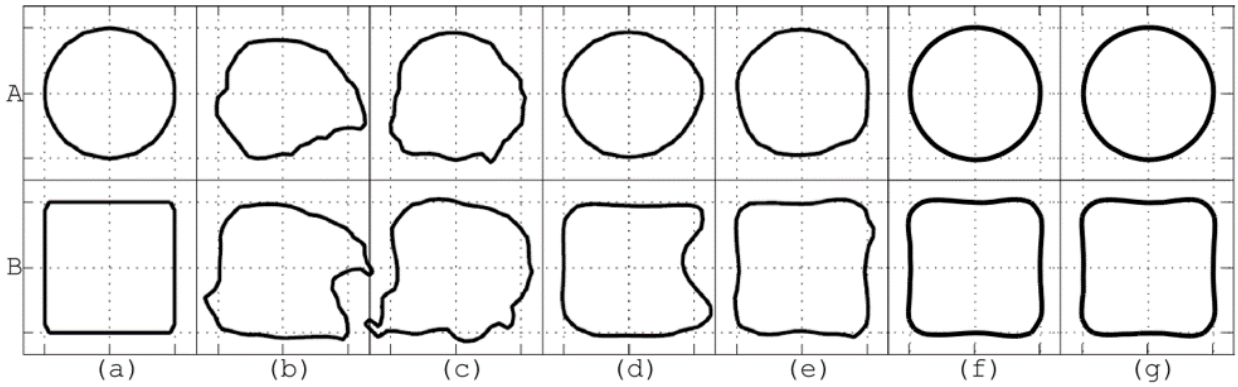


Figure 4.2: Rigid body rotation test results of a circular (panel A) and a square shaped fluid body (panel B). (a) Ideal case (b) Hirt and Nichols without local height function, (c) Hirt and Nichols with local height function, (d) Young's without local height function, and (e) Young's with local height function,(f) VCLP with PLIC and (g) VCLP with PCIC. Note: Results from column (a) to (e) is obtained from [1]

.

even though the edges are slightly curved. Results of Young's methods maintain the shape only partially, however they are closer to the ideal shape than H&N methods. The use of local height function seems to improve Young's method results, but not the H&N results. There is no noticeable difference between VCLP- and VCLP+ results. Overall, the VCLP methods produce good results for the rigid body rotation tests, for both, circular and square shaped fluid bodies.

4.3 Single Vortex

The single vortex test for evaluating the accuracy of a surface tracking and advection method was proposed by Rider and Kothe [7] and the solenoidal flow velocity field was suggested by Bell et al. [85]. For this test, a circular fluid body of radius 0.15 m, centered at (0.5,0.75) is placed inside a solenoidal velocity field whose stream function (Ψ) is given by Eqn. 4.1 as following.

$$\Psi = \frac{1}{\pi} sin^2(\pi x) sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right) \tag{4.1}$$

The velocity field is related to the stream function as $(u_x, u_y) = \left(\frac{-d\Psi}{dy}, \frac{d\Psi}{dx}\right)$ and defined as follows.

$$u_x = -sin^2(\pi x)\sin(2\pi y)\cos\left(\frac{\pi t}{T}\right) \tag{4.2}$$

$$u_y = sin^2(\pi y)\sin(2\pi x)\cos\left(\frac{\pi t}{T}\right) \tag{4.3}$$

where the T is the time period of the single vortex and t is the time. The $\cos(\frac{\pi t}{T})$ term makes the velocity field reversible. At time $t = 0$, velocity field have maximum magnitude and it keeps reducing reaching zero at time $t = \frac{T}{2}$. From time $t = \frac{T}{2}$ to $t = T$, the velocity field reverses, and keeps increasing in magnitude. Thus, in an ideal case the initial circular fluid body in a solenoidal velocity field gets stretched into a filament spiraling around the vortex center till time $t = \frac{T}{2}$, and gets back into the initial circular shape at $t = T$. Similarity of the final state to the initial state is the measure of accuracy for the surface tracking and advection method. Figure 4.3 depicts the single vortex test results of the VCLP methods along with that of H&N and Young's methods. Column (b) and (c) are the results of H&N method, without and with local height function respectively, whereas column (d) and (e) are the results of Young's method, without and with local height function respectively. Results of VCLP with PLIC is shown in column (f), whereas VCLP with PCIC results are portrayed in column (g). All tests are conducted for a CFL value of one. Two grid resolutions of 60×60 and 100×100 are used for the tests. While panels A to D use 60×60, panels E and F uses 100×100 resolution. Panels A and B show results of a single test (with $T = 2$ s) at times $t = \frac{T}{2} = 1\ s$ and $t = T = 2\ s$, respectively, while panels C and D depict results of a single test (with $T = 8$ s) at times $t = \frac{T}{2} = 4\ s$ and $t = T = 8\ s$, respectively. Bottom panels E and F are the results of a single test (with $T = 8$ s and 100×100) at times $t = \frac{T}{2} = 4\ s$ and $t = T = 8\ s$, respectively. The results in column (b) to (e) are obtained from [1].

Test results from all methods, in panel A ($t = \frac{T}{2} = 1s$ with grid resolution 60×60), are very similar except for some small perturbation with the H&N results. Circular shape is maintained in all test results from panel B ($t = T = 2s$ with grid resolution 60×60), except for the H&N

(column b and c) results, where the right half of both circular fluid bodies are deformed and have more noticeable perturbations. Circular shapes obtained from VCLP methods (column f and g) are almost perfect, while that from Young's methods (column d and e) have slight deviations from the circular shape. Test results from panel C ($t = \frac{T}{2} = 4s$ with grid resolution 60×60) have significant differences among the different methods. The vortex filament features discontinuity when H&N method is used (column b and c). The filament remains unbroken or continuous for half length with Young's results (column d and e). The filament in VCLP results (column f and g) are the longest and remains unbroken for most part. Distance between the discontinuous filaments are also minimum for VCLP results. Results at time $t = T = 8$ s for the grid resolution of 60×60 are shown in panel D, where H&N method result (column b) shows completely broken and spread out fluid body. H&N with height function (column c) and Young's results (column d and e) are able to concentrate most part of fluid near the initial fluid state, but fail completely in maintaining the circular shape. Results from VCLP methods (column f and g) have good circular shape, except for the top part which is flatter. The finer grid resolution results in panel E ($t = T = 8\ s$ with grid resolution 100×100) show significantly better results than panel C with grid resolution 60×60. The filaments in Young's and VCLP results (column d to g) are continuous and smooth for most parts. The final time $t = T = 8\ s$ results for the grid resolution of 100×100 are shown in panel F. Results show significant improvement than panel D for column c to g, where the fluid body is more circular.

VCLP results (column f and g) are significantly more close to perfect circular shape than the results from H&N with height function and Young's method (column c to e).

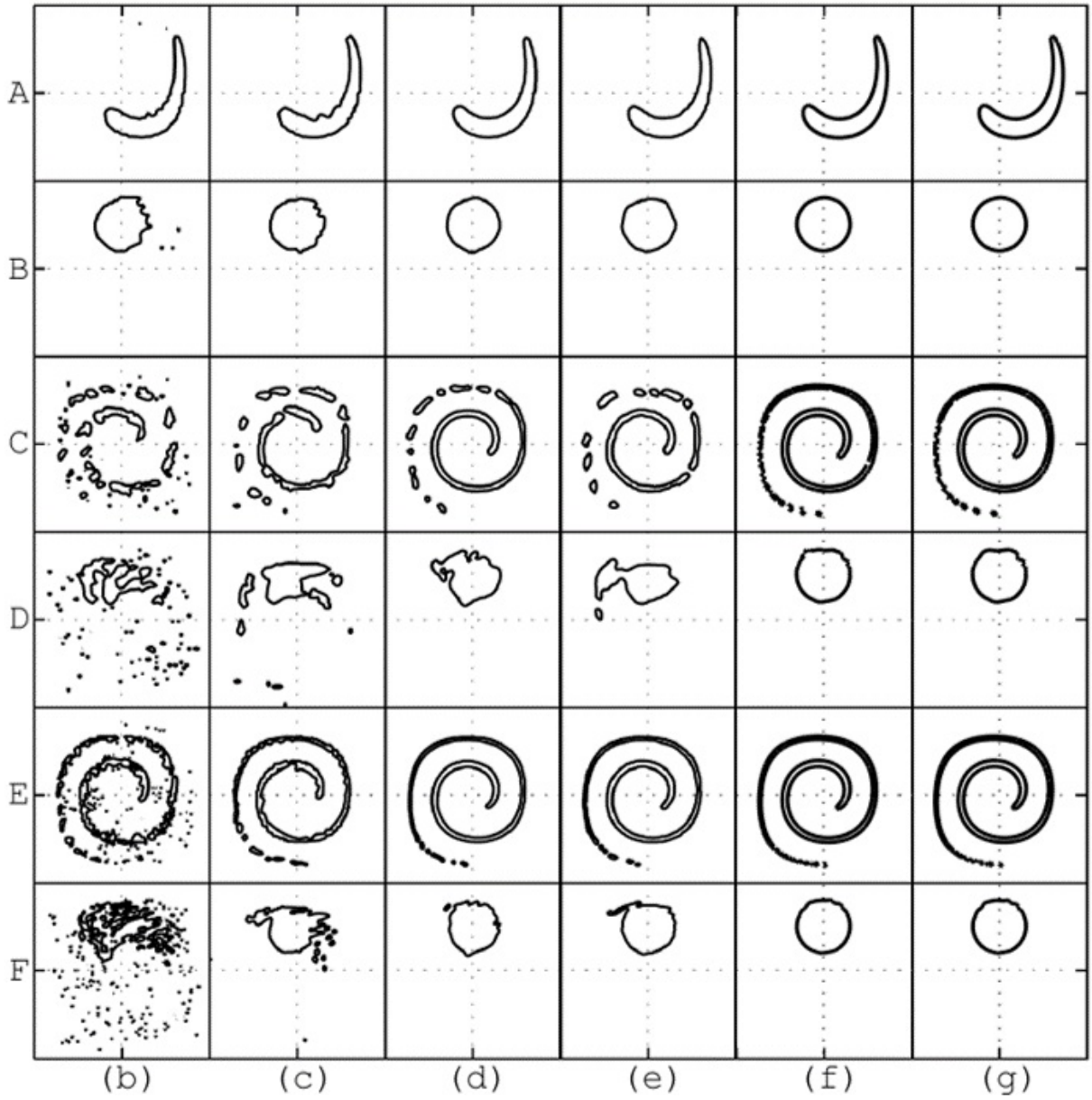


Figure 4.3: Single vortex test result: Advection of a circular body of fluid placed in a single-vortex field. The maximum simulation time is equal to $T = 8$ except for rows A and B, where $T = 2$. Snapshot are shown at time $T/2$ (rows A, C, and E) and at time T (rows B, D, and F). The computational grid consists of 60 x 60 cells (rows A,B,C and D) or 100 x 100 cells (rows E and F). (a) Fluid body at time t = 0 or initial state, (b) Hirt and Nichols results without local height function, (c) Hirt and Nichols with local height function, (d) Young's without local height function, and (e) Young's with local height function, (f) VCLP with PLIC and (g) VCLP with PCIC. Note: Results from column (a) to (e) are obtained from [1]

Overall, VCLP results succeed more than than H&N and Young's method in stretching the circular fluid body into a continuous vortex filament and then bringing it back by the reversal of the solenoidal velocity field. In comparison, even VCLP results using a lower resolution (60×60 resolution, panel D, column f and g) are comparable with results from H&N and Young's method using a finer resolution of 100×100 (panel F column c, d, and e).

Single vortex tests are conducted for CFL = 0.5 and $T = 8\ s$ with two grid resolutions of 64×64 and 128×128 to compare VCLP results with two newer methods, moment of fluid (MOF) and coupled level set volume of fluid (CLSVOF). Figure 4.4 shows the test results at time $t = T$ for the grid resolution of 64×64 where the circle drawn in black color is the ideal case, whereas the dashed red lines are the test results. Single vortex test results a and b are obtained from [2]. The lower half of the test results from MOF (case a) matches well with circle, but the upper half does not. CLSVOF result (case b) is not circular for this resolution. VCLP results have circular shape, however some perturbations exist on topside.

Results of grid resolution 128×128 are presented in Fig. 4.5. All methods show significantly improved result than 64×64. Single vortex test results a and b are obtained from [2]. Qualitatively, the VCLP results are closest to the circular shape, followed by CLSVOF, and then MOF. Perturbations are present at the top part of the fluid body in all cases, however VCLP results have the least perturbations and deviation from the circular shape. Overall, qualitatively, VCLP single vortex test results for CFL$= 0.5$ and the two grid resolutions are good in comparison with results of MOF and CLSVOF.

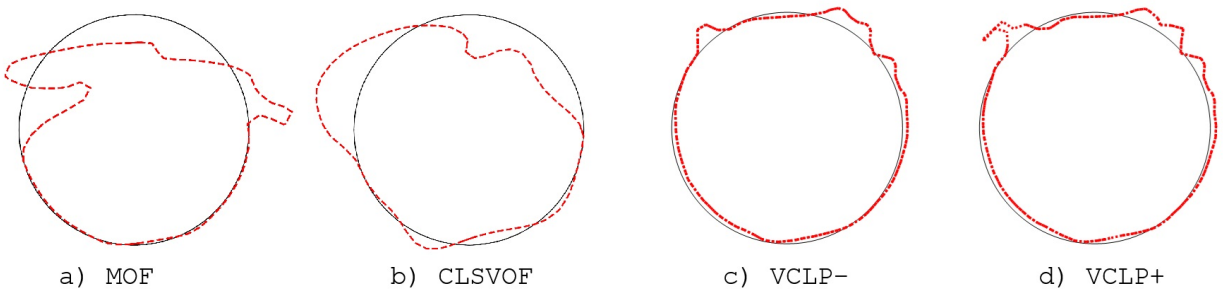


Figure 4.4: Time reversed single vortex test results for grid resolution 64×64 and CFL = 0.5 for $T = 8\ s$. Results at $t = T = 8\ s$ for a) Moment of Fluid (MOF) and b) Coupled Level Set Volume of Fluid (CLSVOF) c) VCLP- and d) VCLP+. Results a) and b) are obtained from [2].

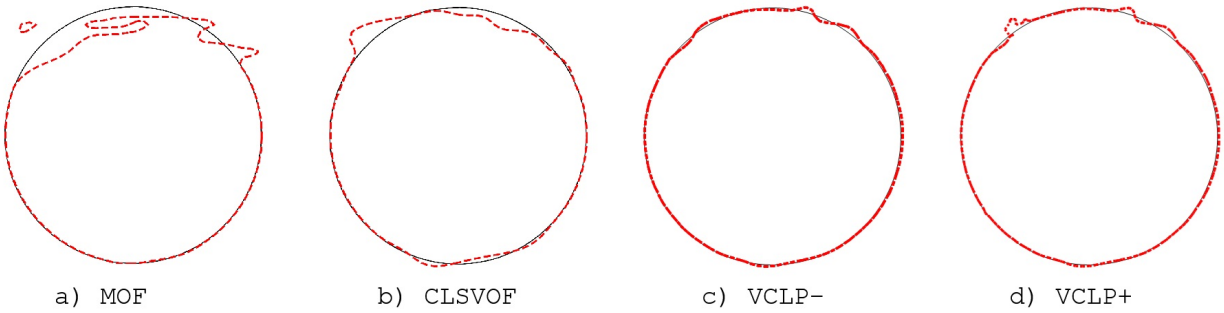


Figure 4.5: Time reversed single vortex test results for grid resolution 128×128 and CFL = 0.5 for $T = 8\ s$. Results at $t = T = 8\ s$ for a) Moment of Fluid (MOF) and b) Coupled Level Set Volume of Fluid (CLSVOF) c) VCLP- and d) VCLP+. Results a) and b) are obtained from [2].

4.3.1 Quantitative Analysis of Single Vortex Tests

Geometric error is used to quantify advection tests' accuracy by comparing the initial and final fluid distribution or the f-matrix. The f-matrix error is integrated over the computational domain, to account for the error in geometry of the fluid body. Geometric error in 2D is estimated using the Eqn. 4.4.

$$E_{geo} = \sum_{i=1}^{i=n_x} \sum_{j=1}^{j=n_y} |f(i,j,T) - f(i,j,0)| dx_i dy_j \tag{4.4}$$

where $t = T$ is final time. Table 4.1 shows the single vortex test results of VCLP and five other surface tracking and advection methods. The tests are conducted for three grid resolutions: 32 x 32, 64 x 64 and 128 x 128. For each grid resolution, two cases of $T = 2$ and $T = 8$ are performed. For convenient representation, H&N and Young's method with '+' sign denotes the application of local height function and '-' denotes without, while for VCLP, '+' means PCIC is used for interface construction and '-' means PLIC is used. The Taylor second order advection scheme is denoted as 'T2' and TJSM as 'TJ'. 'SC' denotes the use of skip core optimization.

From table 4.1, the error ratio of the best result of VCLP vs. the best of H&N for $T = 8\ s$ are 10.3, 30.7, and 48.5 for grid resolutions of 32×32, 64×64, and 128×128, respectively. Young's method error ratios are 7.3, 10.2, and 12.3, whereas Rider & Kothe (R&K) error ratios are 5.9, 3.5, and 2.2 respectively. For $T = 2$, the error ratio of the best result of VCLP vs. the best of H&N are 12.2, 26.9, and 42.4 for grid resolutions of 32×32, 64×64, and 128×128, respectively. Young's method error ratios are 6.9, 15.5, and 20.9, whereas Rider & Kothe (R&K) error ratios are 2.7, 3.1, and 1.7 respectively.

As per results, VCLP performed better than the other methods, especially for larger T, which is the most demanding scenario. Geometric errors of VCLP- and VCLP+ are very close to each other. There is no considerable increase in error with the skip core optimization, which implies the optimization technique is successful. Taylor's second-order advection shows slightly better results than TJSM. Overall, geometric error results show that both VCLP- and VCLP+ do well compared to other methods for the single vortex test.

Single vortex tests for time periods $T = 2,\ 4$ and 6 are conducted for a grid resolution of 64×64 and CFL$= 1$ to compare with results from five other methods, obtained from [7]. The five methods are 1) 1^{st} Order Upwind, 2) Piecewise Parabolic Method (PPM), 3) Level Sets, 4) SLIC, and 5) PLIC.

Table 4.1: Geometric errors E_{geo} for single vortex time reversible tests with time periods $T = 2\ s$ (upper panel) and $T = 8\ s$ (lower panel). H&N and Young's method with '+' sign denotes the application of local height function and '-' denotes without, while for VCLP, '+' means PCIC is used for interface construction and '-' means PLIC is used. The Taylor second order advection scheme is denoted as 'T2' and TJSM as 'TJ'. 'SC' denotes the use of skip core optimization. Results for all tests except VCLP are obtained from literature [1, 3].

Method ($T = 2\ s$)	32×32	64×64	128×128
H&N -	1.90×10^{-2}	1.02×10^{-2}	7.90×10^{-3}
H&N +	1.07×10^{-2}	5.13×10^{-3}	3.31×10^{-3}
Young's -	6.04×10^{-3}	2.97×10^{-3}	1.93×10^{-3}
Young's +	7.23×10^{-3}	3.19×10^{-3}	1.63×10^{-3}
Rider& Kothe	2.36×10^{-3}	5.85×10^{-4}	1.31×10^{-4}
VCLP - TJ	1.39×10^{-3}	5.18×10^{-4}	2.48×10^{-4}
VCLP - T2	8.74×10^{-4}	2.01×10^{-4}	7.87×10^{-5}
VCLP + TJ	1.35×10^{-3}	5.19×10^{-4}	2.49×10^{-4}
VCLP + T2	9.11×10^{-4}	1.93×10^{-4}	7.80×10^{-5}
VCLP+ T2 SC	9.16×10^{-4}	1.91×10^{-4}	1.11×10^{-4}
Method ($T = 8\ s$)	32×32	64×64	128×128
H&N -	1.01×10^{-1}	7.96×10^{-2}	6.76×10^{-2}
H&N +	8.39×10^{-2}	6.05×10^{-2}	3.17×10^{-2}
Young's -	5.95×10^{-2}	2.00×10^{-2}	8.25×10^{-3}
Young's +	1.10×10^{-1}	4.49×10^{-2}	8.02×10^{-3}
Rider& Kothe	4.78×10^{-2}	6.96×10^{-3}	1.44×10^{-3}
VCLP - TJ	9.03×10^{-3}	2.10×10^{-3}	7.25×10^{-4}
VCLP - T2	8.14×10^{-3}	1.97×10^{-3}	6.90×10^{-4}
VCLP + TJ	9.49×10^{-3}	2.03×10^{-3}	7.93×10^{-4}
VCLP + T2	1.02×10^{-2}	2.24×10^{-3}	7.10×10^{-4}
VCLP + T2 SC	8.32×10^{-3}	2.43×10^{-3}	6.53×10^{-4}

The test results are presented in table 4.2. For $T = 2,\ 4$ and 6, the error ratio of 1st Order Upwind results to VCLP results (least error case) are 68.2, 149.3, and 88.4 respectively. For PPM, the error ratios are 4.0, 19.5, and 26.5, whereas for level set method, they are 5.1, 20.9, and 19.0. The error ratios for SLIC are 19.5, 54.4, and 54.7, while for PLIC, they are 0.8, 2.4, and 3.12. Thus, the geometric error of VCLP results are smaller compared with other methods except for one case, suggesting a good performance. Errors associated with VCLP with skip core optimization tests are close to VCLP+ TJ, suggesting. Within VCLP methods, Taylor second order advection scheme results have lesser error than the results with the proposed TJSM advection scheme.

Table 4.2: Geometric errors E_{geo} for single vortex time reversible tests with a grid resolution of 64 x 64 and CFL = 1. VCLP+ denotes VCLP with PCIC and VCLP- denotes VCLP with PLIC.'T2' is Taylor second order advection and 'TJ' is the TJSM method while 'SC' is the Skip Core Optimization. Results for all tests except VCLP methods are taken from literature obtained from [7]

Method	Vor T = 2	Vor T = 4	Vor T = 6
1^{st} Order Upwind	5.96×10^{-2}	9.66×10^{-2}	1.14×10^{-1}
Piecewise Parabolic Method (PPM)	3.48×10^{-3}	1.26×10^{-2}	3.42×10^{-2}
Level Sets	4.42×10^{-3}	1.35×10^{-2}	2.45×10^{-2}
SLIC	1.70×10^{-2}	3.52×10^{-2}	7.05×10^{-2}
PLIC	7.00×10^{-4}	1.55×10^{-3}	4.03×10^{-3}
VCLP - TJ	1.39×10^{-3}	8.59×10^{-4}	1.40×10^{-3}
VCLP - T2	8.74×10^{-4}	6.47×10^{-4}	1.29×10^{-3}
VCLP + TJ	1.35×10^{-3}	8.63×10^{-4}	1.53×10^{-3}
VCLP + T2	9.11×10^{-4}	6.76×10^{-4}	1.42×10^{-3}
VCLP + T2 SC	9.16×10^{-4}	6.79×10^{-4}	1.36×10^{-3}

Table 4.3 gives the geometric errors for the single vortex tests for the three grid resolutions, 32×32, 64×64 and 128×128 with CFL=0.5. The ratio of MOF error to VCLP error for coarser

to finer resolutions are 5.36, 5.77, and 7.55 respectively. The error ratios for CLSVOF to VCLP are 8.47, 6.20 and 4.64. Thus, quantitatively, VCLP single vortex test results are significantly more accurate than newer methods MOF and CLSVOF.

Table 4.3: Geometric errors E_{geo} for single vortex time reversible tests with CFL = 0.5 and $T = 8\ s$. VCLP+ denotes VCLP with PCIC and VCLP- denotes VCLP with PLIC.'T2' is Taylor second order advection and 'TJ' is the TJSM method while 'SC' is the Skip Core Optimization. VCLP test results are compared against MOF & CLSVOF results obtained from [2]

Method	32×32	64×64	128×128
MOF	4.06×10^{-2}	1.61×10^{-2}	5.62×10^{-3}
CLSVOF	6.46×10^{-2}	1.73×10^{-2}	3.45×10^{-3}
VCLP - T2	7.58×10^{-3}	2.93×10^{-3}	7.44×10^{-4}
VCLP + T2 SC	7.63×10^{-3}	2.79×10^{-3}	8.69×10^{-4}

Like geometric error, the relative position error is a quantitative measure used in literature [88] to evaluate the accuracy of advection methods. The relative position error estimates the positional change of fluid parcels in the obtained result to the ideal result and it is defined by Eqn. 4.5. The relative position error of VCLP - T2 for resolution of 64×64 is 2.79%, whereas the new VOF method proposed by Wang et al. [88] have a relative position error of 40.95%.

$$\textit{Relative position Error} = \frac{\sum_{i=1}^{i=n_x} \sum_{j=1}^{j=n_y} |f(i,j,0) - f(i,j,T)|}{\sum_{i=1}^{i=n_x} \sum_{j=1}^{j=n_y} |f(i,j,0)|} \tag{4.5}$$

The relative mass losses found out using Eqn. 4.6 during the single vortex tests using VCLP are of the order 10^{-15}, with nearly perfect mass conservation. In contrast, the relative mass loss using CLSVOF method are 1.42×10^{-5} by Zhao et al. [89], 4.0×10^{-4} by Wang et al. [88], and 1×10^{-4} Menard et al. [90].

$$\textit{Relative mass loss} = \frac{\sum_{i=1}^{i=n_x}\sum_{j=1}^{j=n_y} f(i,j,0)\delta x_i \delta y_j - \sum_{i=1}^{i=n_x}\sum_{j=1}^{j=n_y} f(i,j,T)\delta x_i \delta y_j}{\sum_{i=1}^{i=n_x}\sum_{j=1}^{j=n_y} f(i,j,0)\delta x_i \delta y_j} \tag{4.6}$$

In conclusion, according to the single vortex test results, VCLP method have lesser geometric and relative position error along with negligible mass loss. Hence by quantitative analysis, VCLP performs and better in comparison with other presented methods.

4.4 Deformation Test

Deformation field test is more stringent than the single vortex test, where sixteen vortices are present in the computational domain inducing a radical deformation and topology change of fluid bodies. This test was proposed by Rider et al. [7] and it is a time-reversed flow field defined by the stream function given in Eq. (4.7). The velocity field at time $t = 0$ can is illustrated in Fig. 3.16 inside which a circular fluid body of radius 0.15 m, centered at (0.5,0.75) is placed.

$$\Psi = \frac{1}{4\pi}\sin\left\{4\pi\left(x+\frac{1}{2}\right)\right\}\cos\left\{4\pi\left(y+\frac{1}{2}\right)\right\}\cos\left(\frac{\pi t}{T}\right) \tag{4.7}$$

Deformation tests were conducted using VCLP with PCIC (VCLP+) for two time periods, $T = 2$ and $T = 4$ with CFL = 1 and grid resolutions of 32×32, 64×64 and 128×128. Test results at final time $t = T$ are depicted in Fig. 4.6 (column b and c). Results using R&K method for $T = 4$ (column a) are obtained from [3]. For $T = 4$, grid resolution of 32×32 seems not fine enough for both methods (panel C, column a and b). For 64×64, VCLP has a more circular and symmetric shape. For 128×128, both methods produce better results. While VCLP is more symmetric than R&K at the top and bottom, R&K results are more circular. For $T = 2$, only VCLP results are shown; it is nearly a circle for 128×128. Overall, qualitatively, VCLP results and R&K's results are comparable.

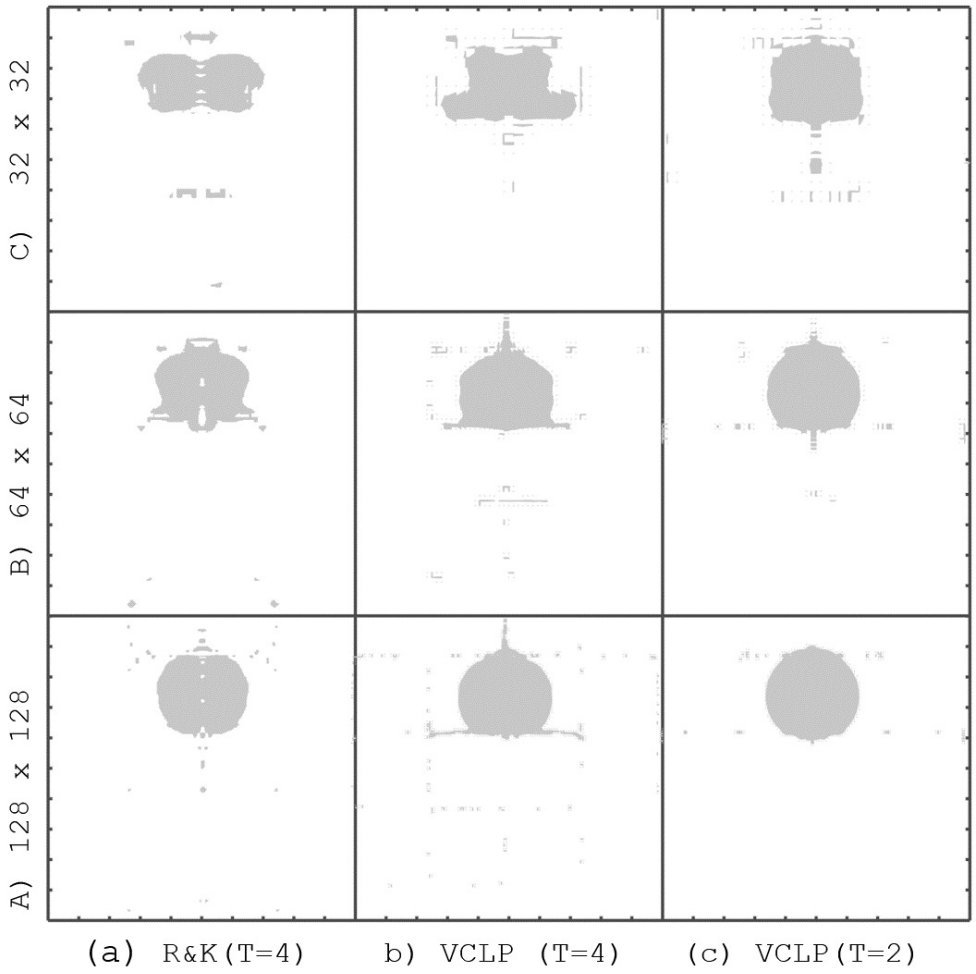


Figure 4.6: Deformation test result at $t = T$ for CFL = 1 and grid resolutions A) 32×32, B) 64×64 and C) 128×128. Column (a) results are for $T = 4$ using R&K method obtained from [3] , Column (b) and (c) use VCLP with PCIC for $T = 4$ and $T = 2$ respectively.

4.4.1 Quantitative Analysis of 2D Deformation Test

Table 4.4 gives geometric errors of the deformation test results for two time periods, $T = 2$ and $T = 4$ with CFL = 1 and grid resolutions of 32×32, 64×64 and 128×128. For $T = 2$, best of VCLP results have 1.61, 2.26 and 3.52 times smaller error than R&K respectively. For $T = 4$, ratio of errors are 1.33, 0.99 and 8.33. So quantitatively, VCLP results have smaller errors than R&K. So overall, VCLP performs well in this challenging deformation tests.

Table 4.4: Geometric error for time reversed deformation test for $T = 2$ and $T = 4$ for three grid resolutions, 32 x 32, 64 x 64 and 128 x 128 and CFL = 1. VCLP+ denotes VCLP with PCIC and VCLP- denotes VCLP with PLIC.'T2' is Taylor second order advection and 'TJ' is the TJSM method while 'SC' is the Skip Core Optimization. Results for all tests except VCLP are from literature [7]

Method	T(s)	32×32	64×64	128×128
R& K	2.0	1.96×10^{-2}	1.12×10^{-2}	5.95×10^{-3}
VCLP - TJ	2.0	1.43×10^{-2}	6.14×10^{-3}	2.00×10^{-3}
VCLP - T2	2.0	1.35×10^{-2}	5.04×10^{-3}	1.70×10^{-3}
VCLP + TJ	2.0	1.22×10^{-2}	5.90×10^{-3}	2.01×10^{-3}
VCLP + T2	2.0	1.54×10^{-2}	4.96×10^{-3}	1.80×10^{-3}
VCLP + T2 SC	2.0	1.61×10^{-2}	5.62×10^{-3}	1.69×10^{-3}
R& K	4.0	4.68×10^{-2}	1.63×10^{-2}	9.08×10^{-2}
VCLP - TJ	4.0	3.51×10^{-2}	1.65×10^{-2}	1.09×10^{-2}
VCLP - T2	4.0	3.90×10^{-2}	1.80×10^{-2}	1.12×10^{-2}
VCLP + TJ	4.0	3.54×10^{-2}	1.68×10^{-2}	1.09×10^{-2}
VCLP + T2	4.0	4.29×10^{-2}	1.87×10^{-2}	1.16×10^{-2}
VCLP + T2 SC	4.0	3.96×10^{-2}	2.00×10^{-2}	1.14×10^{-2}

4.5 Zalesak's disk Test

Zalesak's test problem (Zalesak et al. [91]) is a modified version of the rigid body rotation test presented earlier where instead of a circular fluid body a slotted disk (Zalesak's disk) is used. The slotted disk centered at (0.5, 0.75) have a radius r = 0.15 m. The slot in the disk is defined such that it satisfies the relations, $|x - 0.5| \leq 0.025$ and $y \leq 0.85$. The slotted fluid is placed inside the

velocity field for rigid body rotation given by Eqns. 4.8.

$$\begin{aligned} u_x &= -2\pi(y - 0.5) \\ u_y &= \ 2\pi(x - 0.5) \end{aligned} \tag{4.8}$$

The slotted disk undergoes a rigid body rotation of 2π radian (one complete circle) during 628 time-steps. This test is carried out for a resolution of 200×200 and Fig. 4.7 shows the initial and final states from the test results using VCLP.

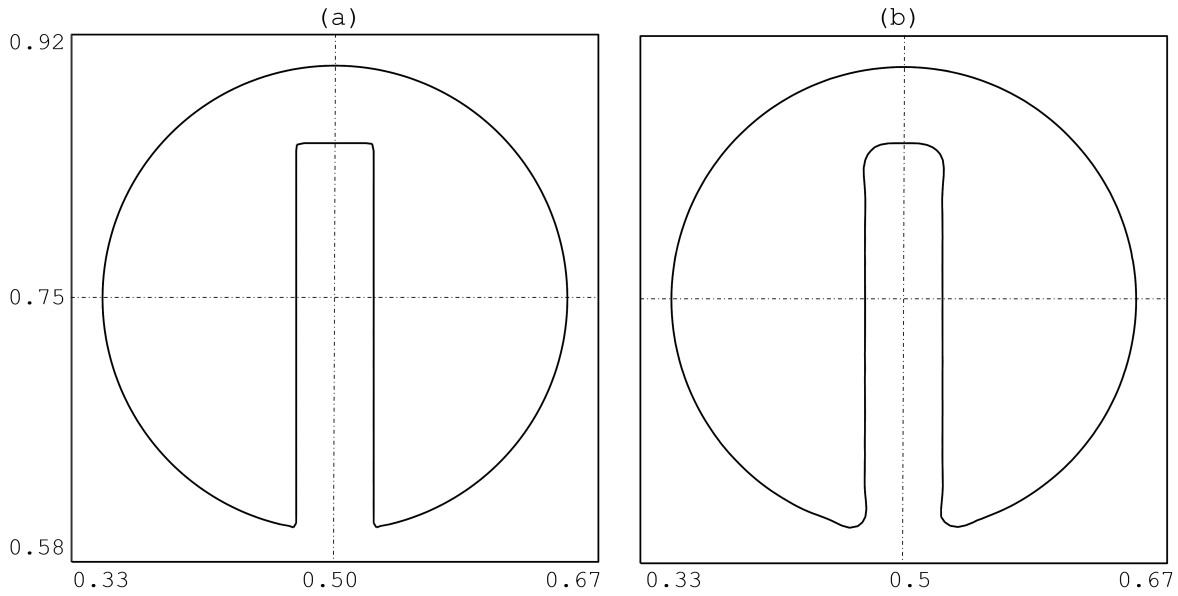


Figure 4.7: Zalesak's rigid body rotation test result for 200×200 resolution and 628 time steps using VCLP.

Figure 4.8 compare the results for 100×100 resolution with the work of Scardovelli et al.[4] using ELVIRA reconstruction method and its variations. The inner edge of the slot has become rounded in all three cases as well as the tip of the slot as observed in left, middle and right panels. The slot edges is not completely parallel with the ELVIRA method, but edges are maintained well with VCLP and the continuity algorithm. The sharpness of the slot tip is maintained in the

decreasing order of 1) VCLP, 2) quadratic fit with the continuity algorithm, 3) quadratic fit, and 4) ELVIRA reconstruction. Overall, VCLP method results are as good as the compared methods from [4].

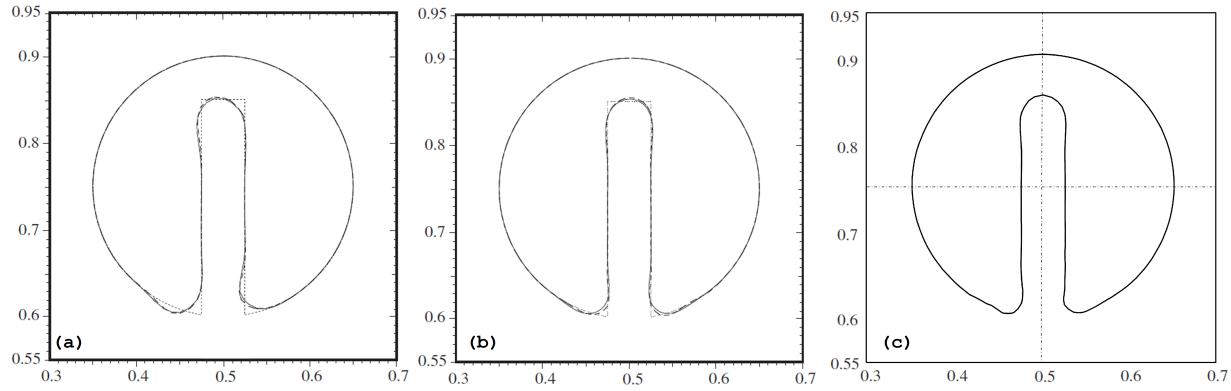


Figure 4.8: Comparison of Zalesak's rigid body rotation test results for 100×100 resolution obtained from [4] with VCLP result. a) ELVIRA reconstruction (solid line) and the linear best fit (dashed line) b) with the quadratic fit alone (solid line) and with the continuity algorithm (dashed line) c) VCLP. Dotted line is the ideal result.

4.5.1 Quantitative Analysis of Zalesak's disk Tests

Table 4.5 gives the relative error of Zalesak's disk test results for the 200×200 grid resolution. The relative errors of 10 other methods are presented from literature. VCLP methods have the least error, even lesser than purely Lagrangian methods. TJSM advection scheme have slightly lesser error than Taylor's second-order scheme. PCIC reconstruction performs better than PLIC with lesser error. So, overall VCLP methods performs very well in the Zalesak's disk test.

Table 4.5: Relative error for Zalesak's disk rigid body rotation test for resolution of 200×200.

Reconstruction/Advection Methods	Relative Error
Higher-order bounded convection/Eulerian [92]	5.72×10^{-3}
Youngs/Stream (unsplit) [4]	1.07×10^{-2}
Puckett/Stream /Eulerian [4]	1.00×10^{-2}
ELVIRA/Lagrangian (split) [4]	1.00×10^{-2}
Linear fit/Lagrangian [4]	9.42×10^{-3}
gVoFoam/Eulerian [93]	1.36×10^{-2}
interFoam/Eulerian [93]	6.61×10^{-2}
OpenFoam®/Eulerian [94]	7.03×10^{-2}
Quadratic fit/Lagrangian [4]	5.47×10^{-3}
Quadratic fit+continuity/Lagrangian [4]	4.16×10^{-3}
VCLP - TJ/ Eulerian-Lagrangian	4.21×10^{-3}
VCLP - T2/ Eulerian-Lagrangian	4.36×10^{-3}
VCLP + TJ/ Eulerian-Lagrangian	3.79×10^{-3}
VCLP + T2/ Eulerian-Lagrangian	3.96×10^{-3}
VCLP + T2 SC/ Eulerian-Lagrangian	4.05×10^{-3}

4.6 3D Deformation Test

The 3D deformation test is used in literature to test the accuracy of surface tracking and advection methods in 3D. In this test a spherical fluid body centered at $(0.35, 0.35, 0.35)$ with a radius of 0.15 m is placed inside a time reversing solenoidal velocity field with vortices. Equations 4.9 gives the velocity field for the deformation test proposed by LeVeque [87].

$$
\begin{aligned}
u_x(x,y,z,t) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T) \\
u_y(x,y,z,t) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/T) \\
u_z(x,y,z,t) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/T)
\end{aligned} \tag{4.9}
$$

The time period for the test is, $T = 3\ s$ with CFL$= 0.5$, which is long enough to stretch the sphere into a thin curved sheet by $t = \frac{T}{2} = 1.5$ s and reverse to get back the initial sphere at $t = T = 3$ s. The 3D deformation tests for VCLP uses 20 LPs per dimension and these are conducted for two grid resolutions, $32 \times 32 \times 32$ and $64 \times 64 \times 64$.

Figure 4.9 displays the result for resolution $64 \times 64 \times 64$. The spherical fluid body on the left panel is the initial state at $t = 0$, whereas the middle panel is the maximum stretched shape at $t = T = 1.5$ s. The final result at $t = T = 3$ s, on the right panel shows the final result at $t = T = 3$ s. The fluid body have a spherical shape, even though there are some perturbations on the surface. Overall, qualitatively, VCLP performs satisfactorily in the 3D deformation test.

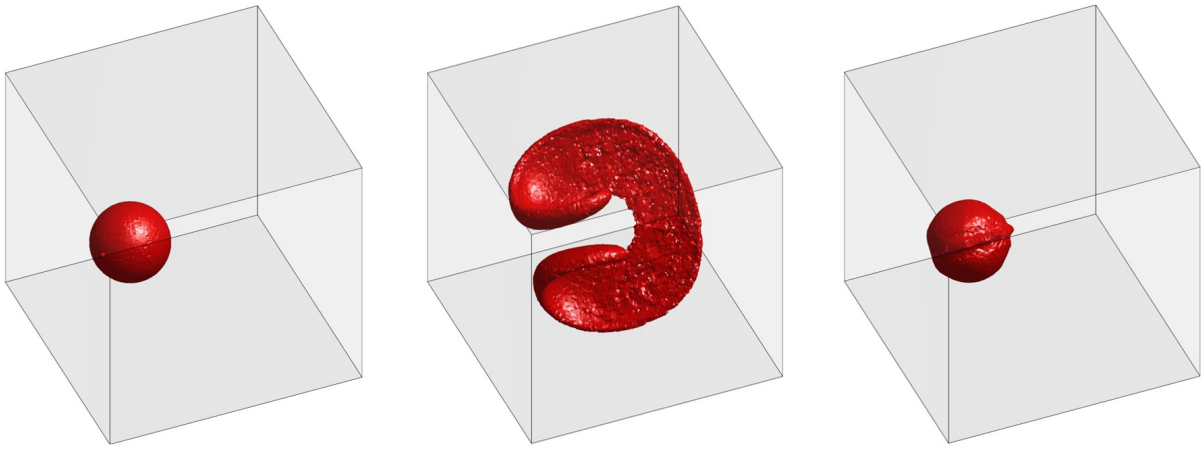


Figure 4.9: 3D Deformation test results of VCLP with $T = 3$ s and CFL = 0.5 for a grid resolution of $64 \times 64 \times 64$. Left panel: initial state at $t = 0$, middle panel: maximum deformed state at $t = \frac{T}{2} = 1.5\ s$, Right panel: final results for $t = T = 3$ s.

4.6.1 Quantitative Analysis of 3D Deformation Test

Table 4.6 provide the geometric errors of the 3D deformation test results using VCLP and other methods from literature. VCLP results are compared with the Moment of fluid method (MOF) [95] and CLSVOF [52] obtained from [2].

Table 4.6: Geometric errors E_{geo} and order of convergence for 3D deformation tests with CFL = 0.5 and $T = 3$ s. MOF and CLSVOF test results are obtained from [2].

Method	$32 \times 32 \times 32$	Order	$64 \times 64 \times 64$
MOF	5.62×10^{-3}	1.26	2.35×10^{-3}
CLSVOF	7.77×10^{-3}	1.15	3.50×10^{-3}
VCLP-	4.40×10^{-3}	1.34	1.73×10^{-3}
VCLP+	4.35×10^{-3}	1.33	1.73×10^{-3}

From results with grid resolution of $32 \times 32 \times 32$, VCLP has 1.29 and 1.78 times lesser errors than MOF and CLSVOF, respectively. For $64 \times 64 \times 64$, VCLP has 1.35 and 2.02 times less geometric error than MOF and CLSVOF, respectively. VCLP also shows a slightly better order of convergence (a measure of how quickly solution converge with increase in resolution) with a value of 1.33. The total fluid volume loss in VCLP is 2.7×10^{-17} m^3, which means nearly a perfect mass conservation. Therefore, based on the quantitative error analysis, VCLP performs well in comparison with the MOF and CLSVOF methods.

4.7 Computational Performance of VCLP

A parametric study was conducted for LPs per dimension distributed as blue noise to evaluate the optimum number of LPs. The single vortex test with resolution of 64×64 and $T = 8\ s$ was chosen for the study. Absolute errors and relative computational times were obtained as indicated in Fig. 4.10. The total computational time for time-steps is normalized with the case using 89 LPs.

It can be seen that error decreases rapidly with an increase in the number of LPs till 25 LPs. After that, error is nearly constant passing 40 LPs. Computational time seems reduce initially till 12 LPs and then increases somewhat linearly, suggesting that a number around 33 LPs is optimal for these tests. The initial decreasing trend from 7 to 12 LPs, is because of the increase in accuracy and reduction in the number of surface cells arising from the broken vortex filament.

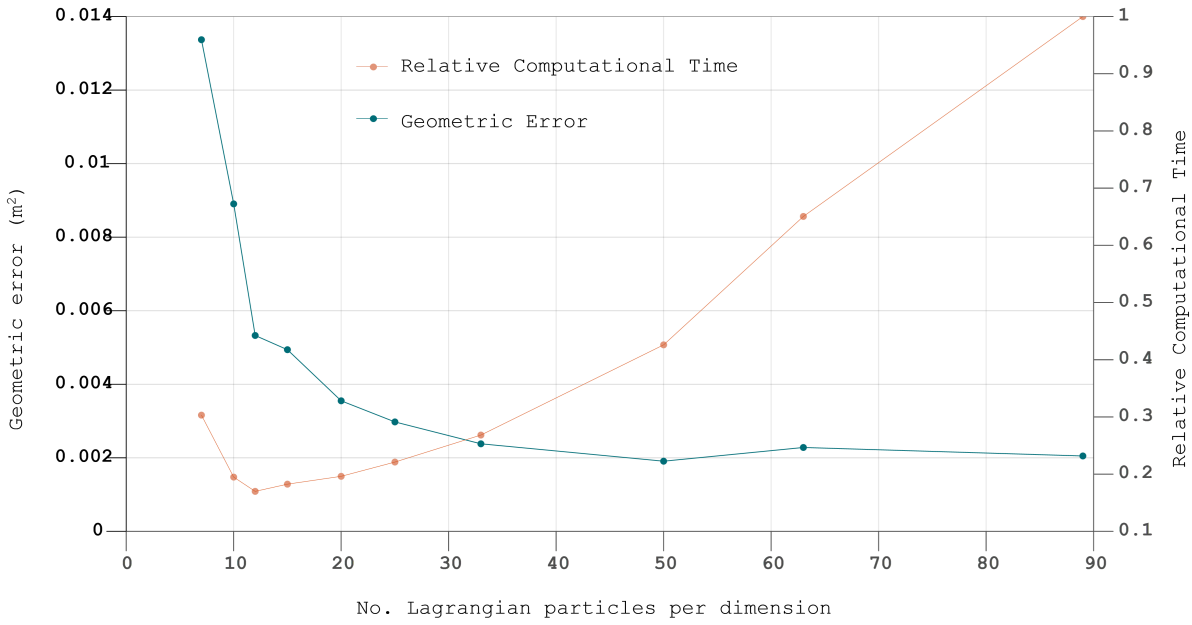


Figure 4.10: Geometric error and relative computational time for VCLP-PCIC with skip core optimization with respect to number of Lagrangian particles for the single Vortex test with $T = 8\ s$ in a 64×64 grid.

A blue noise template has advantages over a white noise template because the distribution of LPs is more uniform in the former. For the same accuracy, the number of LPs required in a blue noise template is considerably lesser that in a white noise template and hence blue noise template is used in VCLP. The following table 4.7 compares the computational time and accuracy for the single vortex test as defined in the section 4.2. The resolution of the VCLP tests is 60×60 and are conducted for 10 and 33 LPs per dimension. In the table 4.7, non-advective means the advection

is entirely skipped from calculation to account for the computational time all methods have in common. CM means with the center of mass tracking, T2 stands for Taylor's second-order method with acceleration, TJ stands for the advection using second-order TJSM method with acceleration. The results show that without the advection (skipping all steps of VCLP; classification, interface reconstruction, advection and updating), the computational time is around 7.4 s. The VCLP method with 10 LPs, Taylor 2^{nd} order advection, and CM tracking takes 19 % more time than VOF, but the error reduced by 1289 %, whereas the VCLP method with 10 LPs, TJSM advection, and CM tracking takes 45 % more time than VOF, but the error reduced by 1350 %.

The case with the least error is VCLP T2 CM 33, followed by VCLP TJ CM 33. The computational time increases somewhat linearly with the number of LPs per dimension in 2D, with an exception for the TJSM method for which it is of a higher power than one. The CM tracking seems to reduce the error by 268 to 600 % and less than five percent more computational time. For a resolution of 120×120, VOF takes 105.5 s with an error of 0.07924, which is 629 % more computational time and 1200 % times more error. This supports the claim that using VCLP at lower resolution obtains better results than SOLA-VOF with lesser computational time. Skip core optimization is not tested since it depends on the ratio of surface perimeter to the area and varies for each tests. With skip core optimization, there will be further reduction in computational time with very small increase in error as evident from the benchmark tests. The advection algorithm typically takes a small fraction of the total computational time and its major portion is consumed by the N-S solver. Therefore, the computational time for advection tests accounts only for the minor portion since N-S equations are not solved.

Notice that some of geometrical errors are more than the area of the circular fluid body (0.07068 m^2). The upper limit of geometrical error is two times the area of the fluid body(0.14137 m^2), which occurs when the fluid bodies at the initial and final times have zero overlap.

Table 4.7: Computational time and error compared of VCLP method compared with original VOF method for Single Vortex T= 8, $\delta x = \delta y = 1/60$ with the circle.

No.	Method	Computational Time	Geometric Error m^2
1	No Advection	7.359	-
2	VOF	14.078	0.0851
3	VCLP T2 10	16.594	0.0396
4	VCLP T2 33	38.484	0.0169
5	VCLP TJ 10	19.875	0.0393
6	VCLP TJ 33	79.250	0.0168
7	VCLP T2 CM 10	16.765	0.0066
8	VCLP T2 CM 33	40.141	0.0042
9	VCLP TJ CM 10	20.438	0.0063
10	VCLP TJ CM 33	83.219	0.0045

4.8 VCLP Applications

To test the proposed method, two practical CFD problems are presented; the dam break problem, and the breaking wave problem. The CFD code used for these simulations is TSUNAMI2D by Horrillo [80] with the VCLP advection module. These two problems are explained in detail below. Additional information on the TSUNAM2D model can be found in [80].

4.8.1 Dam Break Problem

The computational domain is 10 m in length and 15 m in height. The free surface at time $t = 0$ has a width of 3 m and a height of 8 m. There is a square obstacle placed on the floor at position 5 m. Figure 4.11 shows a qualitative comparison of VCLP and VOF using TSUNAMI2D code. The grid cells are squares with a resolution of 0.10 m, and there are 150 x 100 cells. The time step $\Delta t = 0.003$ s.

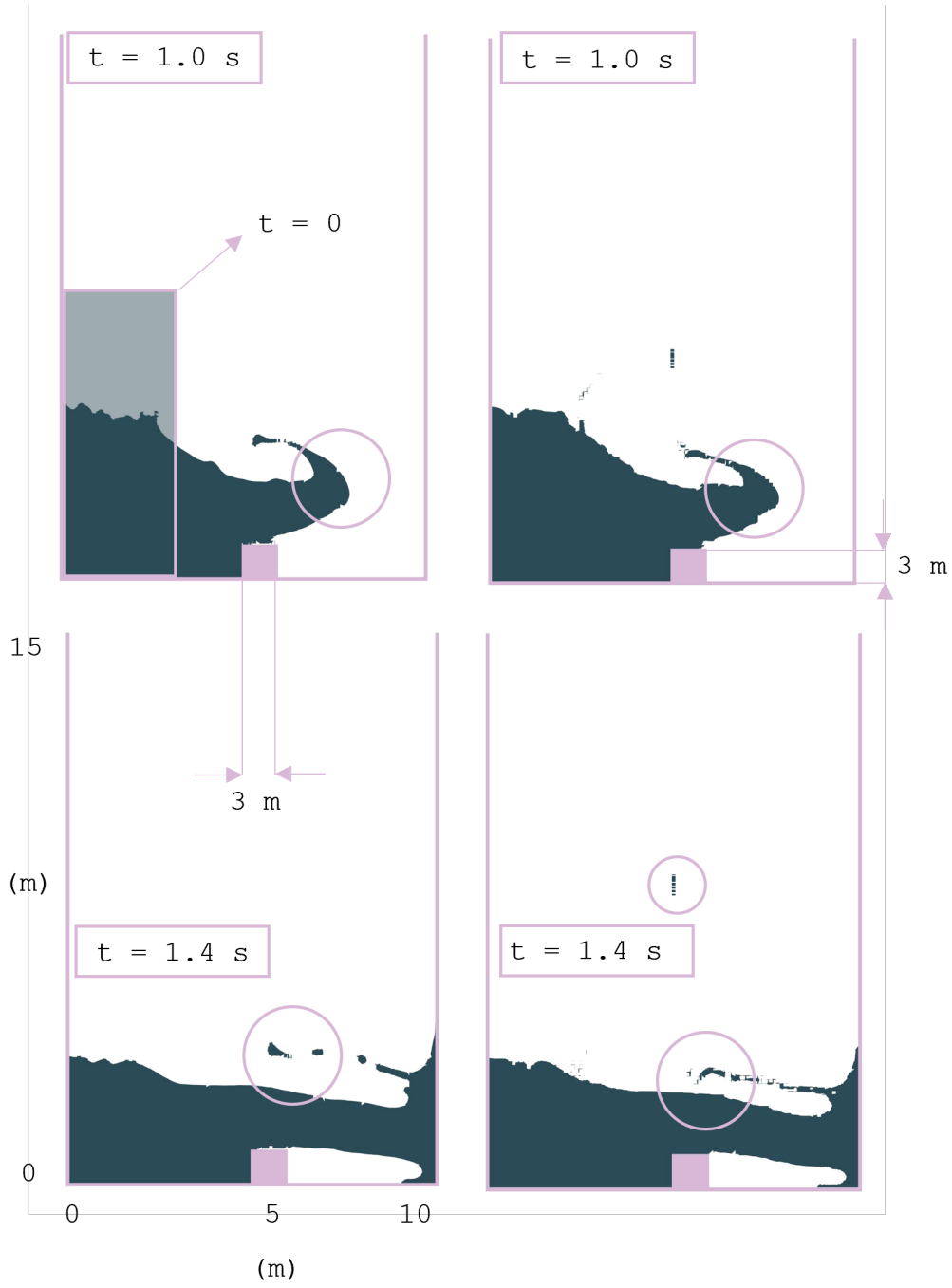


Figure 4.11: Dam break VCLP (left) SOLA-VOF (right) $t = 1.0s$ and $t = 1.4s$

There are subtle changes in the location and shape of the fluid at time $t = 1$ s. Compared to the SOLA-VOF results, the surfaces in the VCLP result are smoother, and the tip of the projected fluid is higher (top panels). At time $t = 1.4$ s (lower panels), the fluid hits the wall, and some minute

differences are noticed, example: the projected fluid reflecting from the wall is moving in a slightly upward direction in the VCLP and an unrealistic upward movement of a fluid parcel is visible in SOLA-VOF result. Overall, there is not any considerable difference in the flow pattern and VCLP result is devoid of some minor unrealistic errors such as surface roughness and upward moving fluid parcel. Since VCLP has more surface features and capability than the SOLA-VOF, it can be understood from this simulation, that VCLP is able to produce very similar result to SOLA-VOF, with some improvement.

4.8.2 Breaking Wave Problem

The breaking wave is a challenging transient multiphase problem to numerically simulate and solve for converging breaker jet free surfaces. However, the problem may be simplified by assuming air is absent (void), making it as a single-phase flow. The simplified version is also difficult to model and has undergone extensive research. Grilli et al. [5] have numerically modeled shoaling and breaking of 2D solitary waves generated by a moving lateral boundary and traveling over a plane slope. The model used for these simulations is based on the fully non-linear potential flow theory and a higher-order boundary element method (FNFP-BEM). One of the drawback of this method is that a singularity is reached at the wave breaking points and at the converging breaker jets. Later, researchers have simulated beyond the breaking point by coupling the FNFP-BEM with the NS-VOF method (Lachaume et al. [6]). They implement two types of coupling principles, weak and strong coupling. In the weak coupling the wave is generated and propagated in the BEM domain, up to near the breaking point and then NS-VOF is used in the breaking zone to simulate breaking and post-breaking. In strong coupling, a moving vertical boundary exchange boundary conditions between models where the fluid regions overlap.

The experiment conducted on this study using TSUNAMI2D-VCLP is same as the one by Grilli et. al [5], where a solitary wave of height $H_0 = 0.45$ m in a 1 m deep wave tank shoals and breaks over a slope of 1:15. Figure 4.12 shows the 2D numerical wave tank for the simulation of the breaking wave. The numerical wave tank is 36 m long and 2 m high. The slope is 1:15 starting at x = 20 m and ending at x = 34 m. This specific breaking wave problem is taken from Grilli et al.

[5], where a solitary wave is generated using a piston wavemaker. As the wave travels towards the slope in the positive X direction, the wave shoals and eventually breaks. The water depth (h_0) is 1 m, and the solitary wave is 0.45 m high. The domain resolution in the Y direction has 400 cells with $\delta_y = 0.5$ cm and in the X direction, δ_x reduces from 10 at x = 0 to 0.5 cm at x = 25 m till the end of the numerical wave tank. Number of cells in X direction is 3944 for a total of (400×3944) 1,577,600 cells.

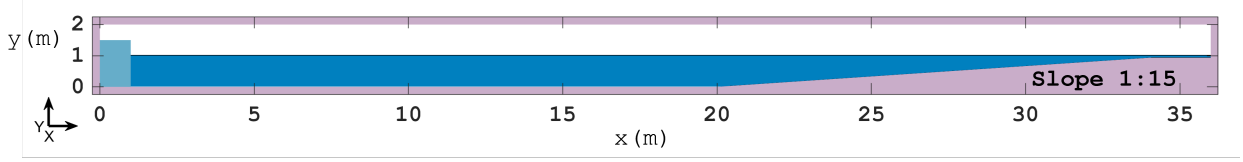


Figure 4.12: Numerical wave tank for the simulation of a solitary wave that shoals and breaks

Equation 4.10 defines the movement of the piston wavemaker to generate the solitary waves. Here H_0 is the solitary wave height in constant depth h_0, and c is the celerity of the solitary wave (analytically for constant depth, $c = \sqrt{g(h_0 + H_0)}$). The motion of the piston wavemaker corresponds to a wave profile that is cut at λ distance on both sides of the wave crest. More detailed information about the piston wavemaker can be found in [5].

$$x_p(t) = \frac{H_0}{\kappa}\left[\tanh\frac{\kappa}{h_0}(ct - x_p(t) - \lambda + \tanh\frac{\kappa\lambda}{h_0})\right] \tag{4.10}$$

where $\kappa = \frac{1}{2}\sqrt{3H_0'}$

The evolution of the simulated breaking solitary wave using TSUNAMI2D-VCLP is given in Fig. 4.13 from time 1 to 10 s. It takes around 3 seconds to create the solitary wave of length 2λ. Speed of the solitary wave in the numerical experiment (3.75 m/s) confirms the theoretical speed of 3.77 m/s estimated from the celerity equation. The shoaling and the breaking can be seen during time from 10.5 to 11.4 s in Fig. 4.13. The solitary wave breaks approximately at 34 m. The hollow region formed at the breaking can be seen clearly and eventually splashing upwards after breaking.

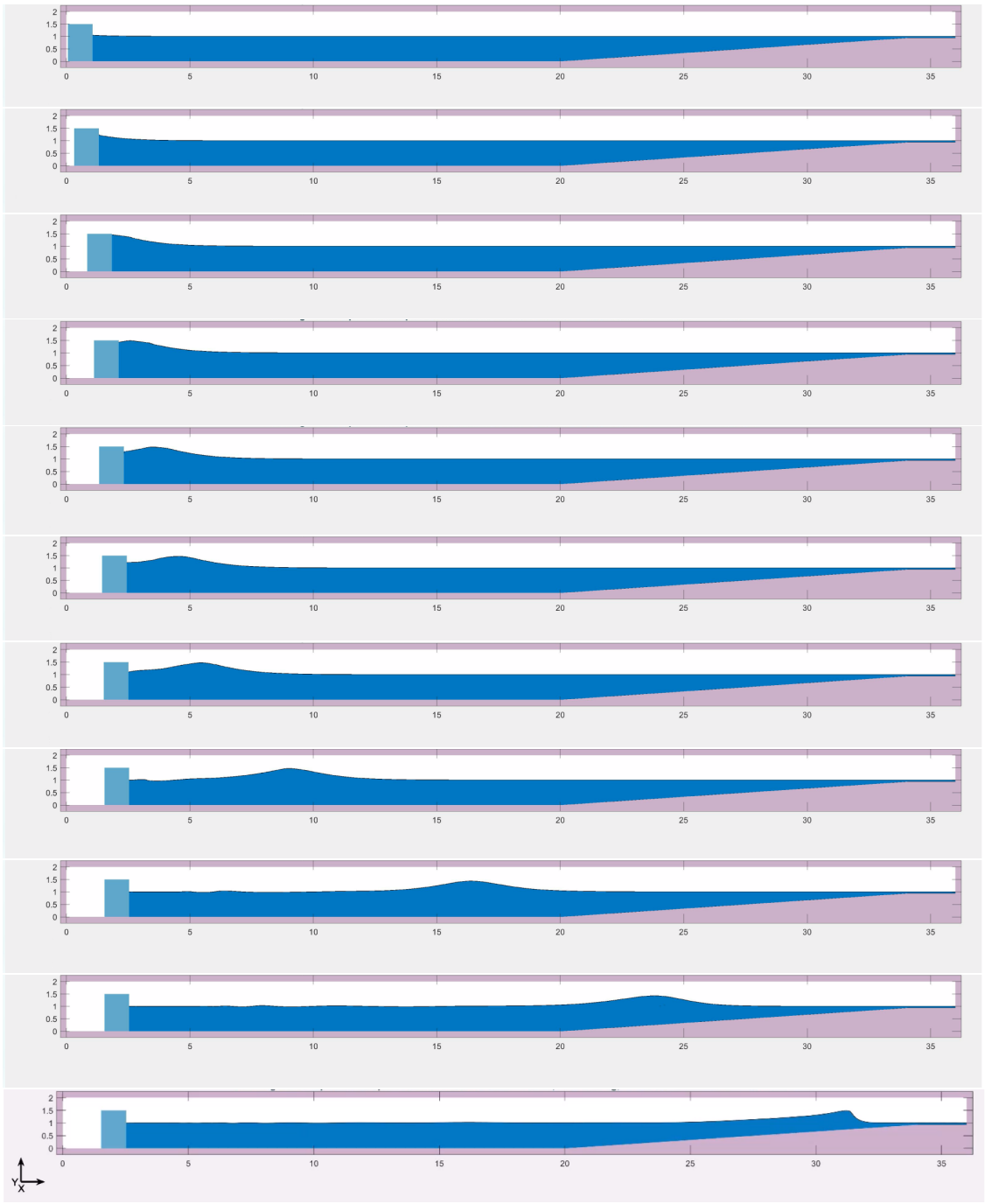


Figure 4.13: Breaking wave problem: Evolution of the solitary wave in time from 1 to 10s. Plots are in in the order of 1, 1.5, 2, 2.25, 2.5, 2.75, 3, 4, 6, 8, 10 seconds.

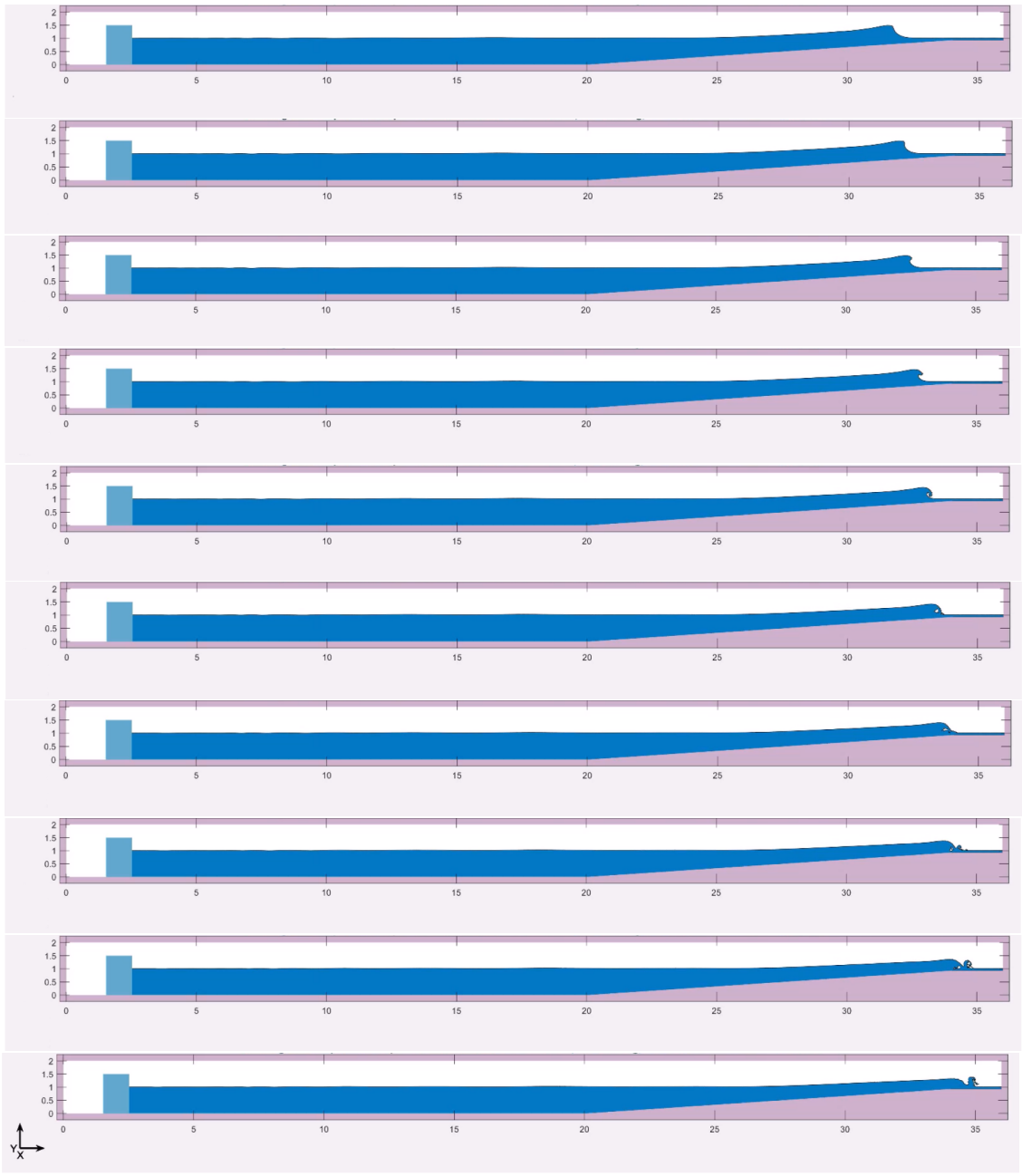


Figure 4.14: Breaking wave problem: Evolution of the solitary wave of height 0.45 m in time during 10.1 to 11 s. Plots are in the order of 0.1 seconds apart.

The TSUNAMI2D-VCLP results till the breaking point is compared with the FNFP-BEM results by Grilli et al. [5] obtained from [6] and are given in Fig. 4.15. The dotted line is FNFP-BEM, while the solid line represents the TSUNAMI2D-VCLP results. The solitary wave at locations 'a'

to ‘f’ matches well in terms of height position and shape.

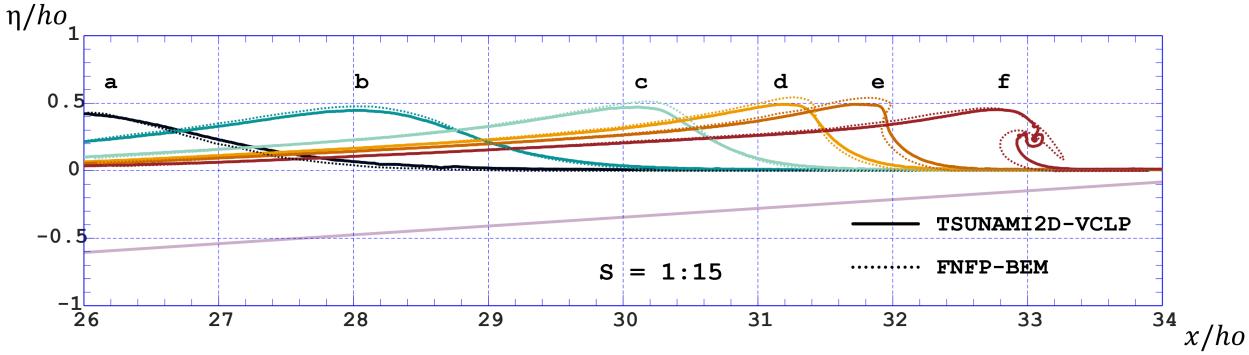


Figure 4.15: Comparison of the evolution of a solitary wave model using TSUNAMI2D-VCLP and FNPF-BEM model by Grilli et al. [5] with Ho/ ho = 0.45, over a slope of 1:15. The solid lines are FNPF-BEM model results and dotted lines are TSUNAMI2D-VCLP.

TSUNAMI2D-VCLP simulation result after breaking is compared with the numerical modeling results by Lachaume et al. [6] that uses strong-coupled FNPF-BEM/NS-VOF model in Fig. 4.16.

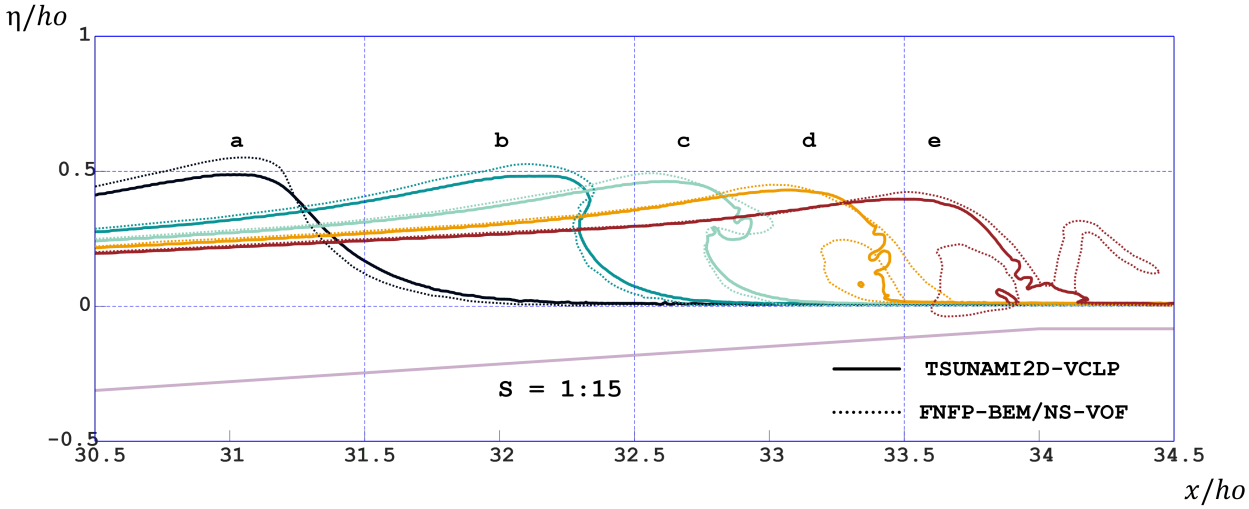


Figure 4.16: Comparison of the shoaling and breaking of the solitary wave model using TSUNAMI2D-VCLP and strong-coupled FNPF-BEM/NS-VOF(SL-VOF) model by Lachaume [6] et al. with $H_o/h_o = 0.45$, over a slope of 1:15. The dotted lines are strong coupled FNPF-BEM/NS-VOF(SL-VOF) model results and solid lines are TSUNAMI2D-VCLP.

The NS-VOF model developed by Guignard et al. [96] uses the SL-VOF and PLIC reconstruction for interface tracking and a N-S solver by De Jouette [97]. The N-S solver of [97] use a 5^{th} order Runge-Kutta scheme, while TSUNAMI2D-VCLP uses second-order Van Leer scheme. Both VCLP and the SL-VOF models use PLIC construction. It can be seen that the model results 'a' to 'e' match, before and after breaking in terms of height position and shape. So, overall TSUNAMI2D-VCLP performs well in simulating the breaking wave problem.

5. NUMERICAL MODELING OF VOLCANIC TSUNAMI EXPERIMENTS

5.1 Introduction

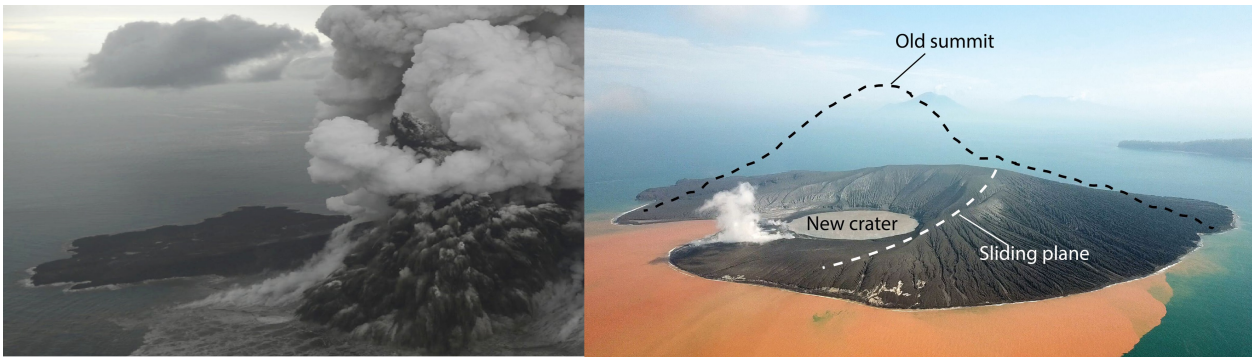
Tsunami is a sequence of waves in a water body, generally in an ocean or a large lake, caused by the displacement of a huge volume of water. On December 26, 2004, more than two hundred thousand lives were lost by the Indian Ocean tsunami. While an underwater earthquake caused this catastrophic tsunami, tsunamis can also be originated from submarine and subaerial landslides, volcanic activities, meteorological causes, and asteroid impacts. A tsunami is generated when enormous energy from one or more above-mentioned sources is transferred to a water body by either displacement, or deformation. Once generated, it can propagate through the ocean with a large velocity, and on approaching shore, it undergoes shoaling, resulting in the slowing down of tsunami waves and decrease in its wavelength. The lost kinetic energy is then transformed into potential energy with an increase in wave heights. While tsunami in open water is minuscule, it is of great size and power while traveling towards land and boats [98, 99]. The great sized waves then become tsunami runups, bore formation, overland flow, and inundation that can devastate the coastal regions on their way.

Volcanic tsunami makes up to five percent of all tsunamis, and among volcanic fatalities, about 16.9 percent are caused by tsunamis [100]. A volcanic tsunami can be formed from submarine volcanic activities like explosions, caldera collapses, flank failures, pyroclastic, and tectonic plate movements. Ninety-plus events have been identified in the last 250 years, and it is estimated that volcanic-tsunami fatalities make up to 25% of all lives lost due to volcanic eruptions. The disastrous tsunamis can reach up to 25 km, more extensive than other volcanic effects.

It is believed that the explosion/collapse of the volcano of Santorin in the Aegean Sea destroyed the Minoan civilization in Greece in 1490 B.C. The volcanic activity of Mount Unzen situated in the Shimabara Peninsula, Japan, in 1792 caused a mega volcanic tsunami [101] of 165-foot height, killing 15,000 people. The massive lateral collapse of Ritta Island on March 13, 1888, caused the

second-largest volcanic tsunami making waves tens of meters high [102]. Even when they arrived at the parts of New Guinea Island after traveling hundreds of kilometers, the waves were still 8 m tall according to witnesses [102]. The most disastrous and most enormous volcanic tsunami was the Krakatoa eruption on August 27, 1883, and there were more than 40 m waves and 36000 fatalities [100, 103]. Shortly after the eruption, the magma chamber collapsed, seawater rushed into it, and a second explosion generated a 10 m tsunami. The third-largest volcanic tsunami ever and disastrous one taking 437 lives is only three years old, as Anak Krakatau volcano, Indonesia, laterally collapsed on December 22, 2018 [104]. The eruption and the lateral collapse of the Anak Krakatau volcano generated a tsunami with runups up to 13 m causing at least 426 fatalities. In Fig. 5.1, the left side shows the erupting mount Anak Krakatau after the tsunami, making waves. The right side shows the drone image after the flank collapse of Anak Krakatau with a missing 320 meters tall old summit. In comparison with earthquake-generated tsunamis, volcanic tsunamis are short-period

Figure 5.1: a) Erupting mount Anak Krakatau. b) Anak Krakatau after collapsed flank 320 meters tall old summit is missing. *(Source: Helmholtz Association of German Research Centres*)



waves with larger dispersion and limited far-field effects. They have intermediate water depth wavelengths and produce large wave heights locally and near the source, [105, 106, 107, 108]. Twenty percent of volcanic tsunamis result from earthquakes [105, 109]. Only volcano-tectonic earthquakes with high frequency can deform ground, large enough to produce a tsunami. Volcanic-

tectonic earthquakes with magnitudes typically less than six are seismic swarms at shallow depth and generate only tsunamis with small magnitude. However, these tsunamis often occur before volcanic eruptions [110, 111]. Now, earthquakes of any origin could cause slope instability flank failures and thus, indirectly generate tsunamis. Even though these tsunamis from volcano flank failures make up one percent of all tsunamis in the past four centuries, on a local scale, they are potentially high magnitude hazards [112]. Volume, origin, and dynamics determine the characteristics of a landslide tsunami from slope instability [113]. In terms of energy, these may match earthquake induced tsunamis, but radial spreading limits their propagation [114]. Tsunamis produced by landslides with high acceleration or deceleration produce short wavelengths, and dispersion restricts its far-field propagation [115]. Pyroclastic flows made up of hot gas and particles can also generate tsunamis. It is shown that the dense, basal debris component of pyroclastic flow is required for energetic and coherent wave formation [116]. Pyroclastic flows are suggested for the cause of the large waves of 15-30 m formed in Sunda Strait during the 1883 tsunami [117, 118]. Another source for a volcanic tsunami could be shock wave. Volcanic explosion pressure could be high enough to cause a meteorological tsunami; a rare phenomenon [119, 120]. This might explain the worldwide tsunami after Krakatao explosion [103, 121]. Caldera collapse resulting from large volcanic explosions and eruptions can displace large water masses, producing tsunamis. Geometry and duration of collapse determine the amplitude of water subsidence that initiates propagation of the tsunami's leading trough. A tsunami can also result from an underwater volcanic explosion, and depending on the water depth and energy of the explosion, a water crater development may be initiated. Two successive bore waves are produced when an expansion rise to follow this, and gravitational collapse of the crater occurs [114]. Many slight undulations follow the bore waves that propagate radially outwards from the crater. Studies show explosion energy determines the initial water crater depth [122]. Underwater explosions typically produce small amplitude and short period waves, and the dispersion effects restrict far-field effects. However, for the underwater explosion in shallow water, the dispersion rate is lower, and the result is tall runups locally, as in the case of Karymsky Lake in 1996 with 19 m runup [90, 109, 110]. The physics of underwater

explosions and magma-water interactions are complicated as it is the function of the geometry of the vent and magma-water interface, water depth, thermal energy transfer, mixing processes of water and magma, metastability of the superheated water, and the amount of gas inside magma that is moving up [111, 112, 113]. Compared to the Surtseyan explosion, underwater violent steam explosions like the Taal volcano in 1965 forming maars and tuff rings more likely produce a tsunami. The physical experiments are modeled based on the active volcano in Grenada, Kick'em Jenny. Figure 5.2 shows the bathymetry, location, and eruption in the action of this volcano with an 'Orange Alert' issued in 2015. Disastrous tsunamis in the past two decades, such as Papua

Figure 5.2: Kick'em Jenny Volcano elevation, eruption and location. (*Sources: Left, Middle: Seismic Research Centre at the University of the West Indies. Right: 2013 R/V Nautilus data set in the main map and GEBCO in the inset*)

New Guinea (1998), the Indian Ocean (2004), Japan (2011), and Indonesia (2018), stirred the research interests in earthquake and submarine landslide tsunamis. Although relatively infrequent, the fatalities and magnitude of the recent events due to volcanic tsunamis show the relevance of modeling and predicting volcanic tsunami. The prediction and monitoring of volcanic tsunamis are complex, and the available time is just a few minutes to issue an alarm [123, 117, 124, 125]. In the case of volcanic tsunamis, distant tide gauges and post-tsunami field surveys are the only available field observations. Experimental or numerical modeling studies are required to understand different source mechanisms and tsunami wave generation. Numerical models can be used for validation, and physical modeling experiments can provide data to validate. The increase in computational power and memory in past decades allows numerical modeling of large water domains

with thousands of kilometers in area, with a few-meter resolution even while solving a highly non-linear Navier-Stokes equation. Even though Direct Numerical Simulation (DNS) of such a large domain is not even closely achieved yet, methods such as Reynolds Averaged Navier-Stokes Simulation (RANS), and Large Eddy Simulation (LES) are up-and-coming in terms of non-linearity and complexity it can handle, and accuracy. Thus, numerical modeling is a feasible and effective way to learn and predict tsunamis, even large scale. Since the past two eventful decades, there have been considerable developments and better tools in understanding, modeling, and studying tsunami source mechanisms, and geographical distribution [104]. Landslide tsunami from lateral collapses has been widely modeled, including large landslides on ocean islands [126, 127, 128]. However, validation of these models is incomplete due to the lack of historical examples [129, 130, 131]. Numerical models are sensitive to various parameters, grid sizes, boundary conditions, approximations and the equations used, and thus errors could be coupled and cumulative. In this scenario, one of the best ways to validate a numerical model would be to validate with the laboratory experiments on tsunamis in wave basins even though they are costly, time-consuming, and limited in different scenarios that can be constructed for experiments. The NSF-funded volcanic tsunami physical experiments were conducted inside the three-dimensional NHERI O. H. Hinsdale Wave Research Laboratory at Oregon State University [132], by deploying a novel Volcanic Tsunami Generator (VTG) on the basin floor. Measurements from four wave gauges are compared to validate its potential in modeling submarine volcanic tsunami experiments. This model successfully predicts highly non-linear volcanic tsunami waves with intricate wave patterns. Thus, in the future, validated models can conduct experiments that are not feasible due to laboratory or time constraints, which would help assess volcanic tsunami hazards more accurately, improve tsunami warning and mitigating systems, and save lives.

5.2 Physical modeling of Volcanic Tsunamis

Physical modelling of volcanic tsunamis can be classified into surface impact models and bottom uplift models. Related to the atomic tests in 1946 Marshall islands, Johnson dropped steel plates on the water surface to model the underwater explosions and experimentally determined

that the ratio of wave height to the water depth is inversely proportional to the radius [101]. According to Kranzer's mathematical model, power of radius should be -1/2, and it matches with the study of Law's experiments with sliding blocks of different mass and aspect ratios into a channel [133, 134]. The fact that attenuation from viscosity and radial propagation are both proportional to power of -1/2 supports this study. Van Dorn arrived at a result on the tsunami amplitude attenuation with the power of radius being -5/6 [135]. Méhauté and Wang compared the wave profiles with non-linear wave theory [136]. The result shows the first two waves of similar and highest amplitudes were followed by a series of smaller waves. In an experiment to model submarine landslides, Watts used sliding triangular blocks at an angle of 45° and found out that around only 3-7% of the kinetic energy was transformed into wave energy [137, 138]. Fritz generated waves from granular impulse and found the kinetic to wave energy conversion was between 2 to 30% [139, 139]. McFall modeled subaerial landslide tsunamis using a pneumatic landslide generator and provided empirical equations to predict the wave amplitude of the first two largest waves [140]. Kim numerically modeled these experiments using TSUNAMI3D to validate [141]. Grilli used sliding solid block with Gaussian cross-section to study subaerial landslides [142]. Wave runups of subaerial landslides were found to be larger than submarine landslides based on the video analysis. A bottom uplift model experiment conducted by by Hammack used the upward motion of a rectangular piston at the bottom of the tank and adjacent to the wall to generate solitary waves and dispersive trailing waves [143]. Jamin used a solid flat circular piston under a circular region covered by a stretchable elastic sheet [144]. A minimal water depth of 25 mm with a stroke of 5 mm was used, and hence scalability is questionable due to the presence of capillary and viscous effects in this scale. In 2019, Hermann Fritz and team conducted physical modeling experiments to understand better how underwater eruptions generate tsunami waves and to characterize volcanic tsunamis, inside Oregon State University's Hinsdale wave basin laboratory, an NSF-funded NHERI experimental facility, by deploying a novel Volcanic Tsunami Generator (VTG) on the basin floor [132]. VTG is the wavemaker, a pneumatically controlled and vertically expandable cylindrical telescopic eruptive column tank of 1.2 m diameter and 73 cm to 103 cm of height (expanded), which can push

the water column 0.3 m up through the water surface in less than a second, causing a significant displacement, which can model the source mechanism of a submarine volcanic eruption and subsequent tsunami waves. Over 300 experiments simulated explosions at different water depths and different energy and kinematics, to isolate the source mechanism of submarine volcanic eruptions, using sensors measuring 3D velocities, surface elevation around the volcano, and the wave runup on the shoreline. Eight synchronized pneumatic pistons of diameter 80 mm each with a stroke length of 30 cm accelerates and expands the VTG. Underwater snapshots of the VTG at the initial and final stages are shown in Fig. 5.3. Fig 5.3 a) shows the retracted stage or the initial stage where water is still, and VTG is 73 cm in height. b) shows the fully expanded VTG height of 103 cm with a radial wave around VTG. The controlled acceleration capability of VTG allows vari-

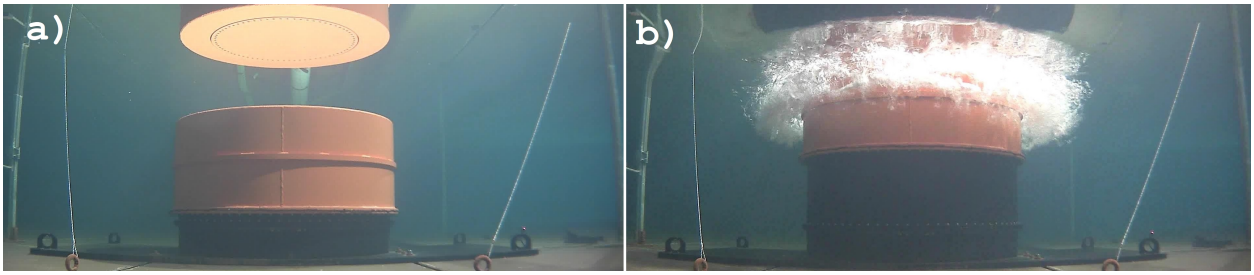


Figure 5.3: Volcanic Tsunami Generator (VTG) under water: a) retracted VTG. b) Fully expanded VTG.

able eruption velocities that mimic different kinds of underwater volcanic activities ranging from slower mud volcanoes to violent underwater explosions. Eruptive energy is from the pressurized air inside the pistons. The VTG displacement as a function of time is obtained from the internal linear potentiometer. Volcanic tsunami physical experiments satisfy generalized Froude similarity given by the equation 5.1 where 'M' is the model, and 'P' is the prototype.

$$\frac{V_M}{\sqrt{gL_M}} = \frac{V_P}{\sqrt{gL_P}} \tag{5.1}$$

VTG is deployed 10.08 m away from the wavemaker panels along the centerline of the wave basin of dimension $44.2\,m \times 26.5\,m$. The layout of the O. H. Hinsdale wave research lab basin and locations of VTG and the gauges are shown in Fig. 5.4 Surface elevation measurements

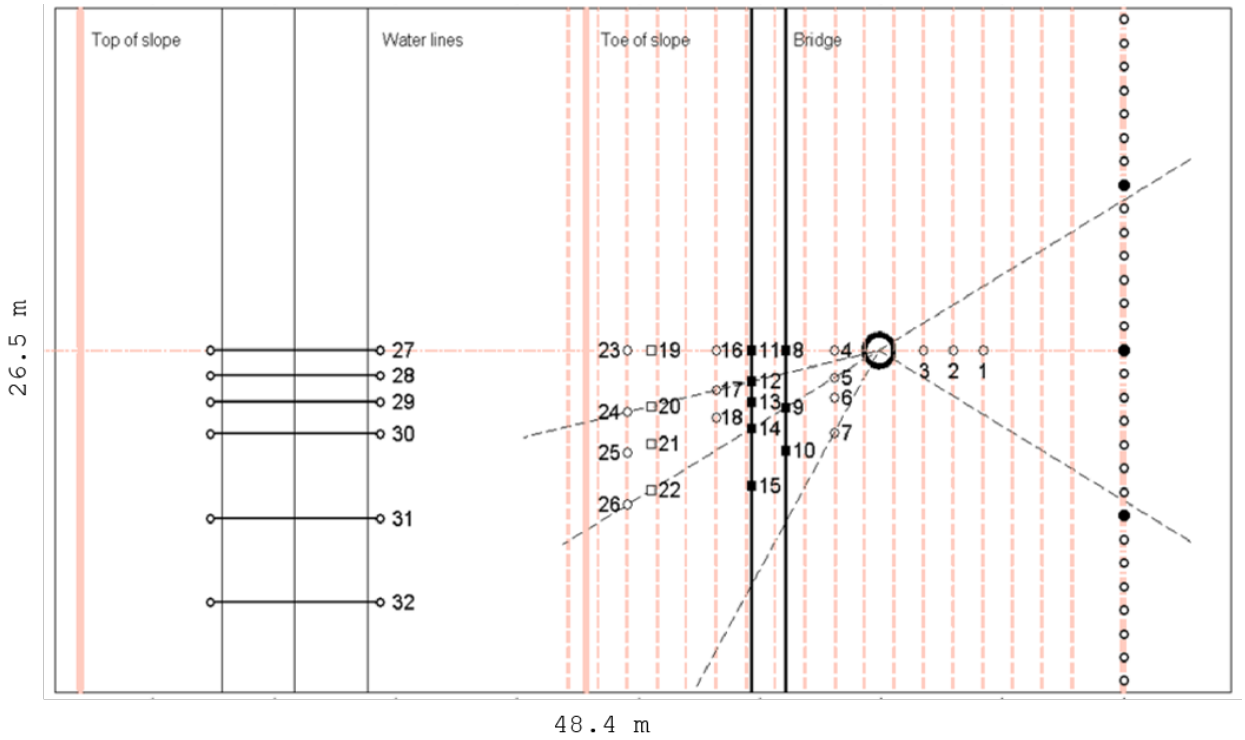


Figure 5.4: O. H. Hinsdale wave research basin layout and locations of volcanic tsunami generator and gauges.

were done using 26 resistance wave gauges and six runup gauges, as shown in the Fig. 5.4. The experimental team selected gauges' locations to measure radial propagation, dispersion, and decay of the generated radial waves. Four are chosen among the 26 resistance gauges (4, 8, 16, and 20) to validate the numerical model. The wave basin of inner dimension $44.2\,m \times 26.5\,m$ has a slope from halfway, i.e., 22.1 m that rises to 2.1 m at 44.2 m. Figure 5.5 shows the numerical wave basin, retracted VTG, and the four gauges with a water depth of 1.2 m. The sampling rate for the gauges is 200 Hz. The experimental procedure begins with filling or removing water from the wave basin to achieve the required water depth and then waits for 15 min until we have almost

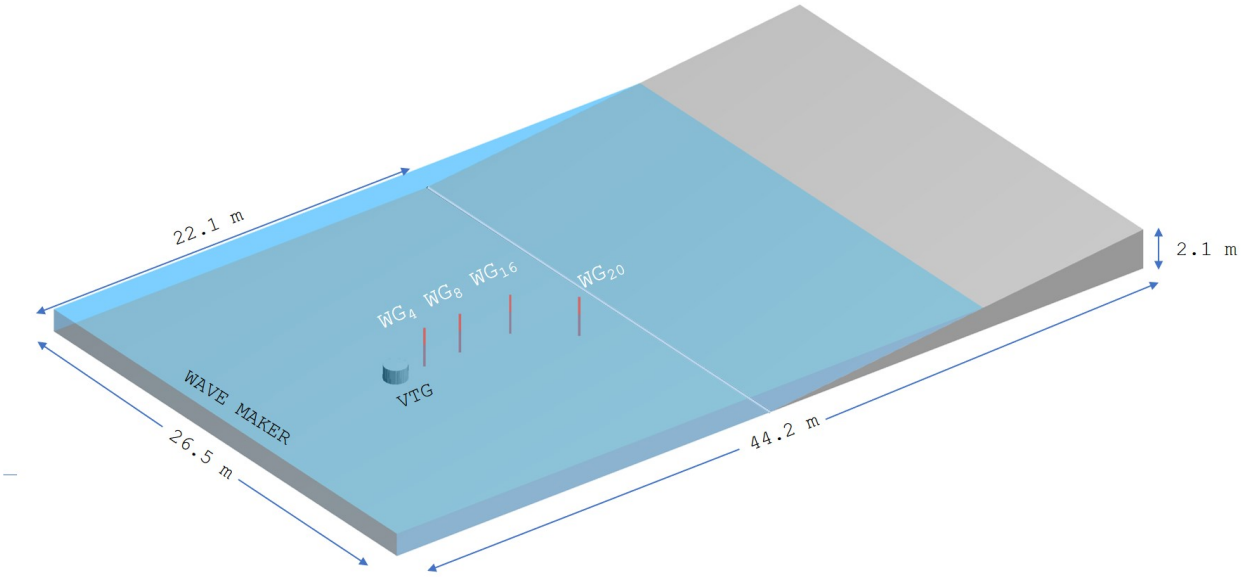


Figure 5.5: Numerical wave basin layout and locations of volcanic tsunami generator and gauges.

still water. The pressure chamber is then set to the required air pressure. Finally, the activation switch releases the pressurized air to eight pistons inside VTG, expanding it vertically and pushing the water column above to produce the wave. VTG has an initial height of 73 cm and a stroke length of 30 cm. Eight of the physical experiments are numerical modeled, in which water depth and tank pressure vary. Water depth ranges from 90 m to 150 m, and tank pressure varies from 40 psi to 145 psi. The piston's peak velocity during the stroke ranges from 0.68 m/s to 2.05 m/s. The water depth, tank pressure, and peak velocities of the ten experiments are shown in Table 5.1. The velocity-time series of VTG is obtained from the linear potentiometer inside. It can be seen that for experiment number one to four, VTG pierces water surface by 13 cm, and for experiments VTG is completely submerged during expansion and create non-piercing waves. The velocity-time series of VTG for ten experiments are shown in Fig. 5.6. As expected, peak velocity reduces as water depth increases or tank pressure decreases. For all cases, after attaining peak velocity, there are local peaks that get smaller exponentially. For all cases, peak velocity is achieved between 0.05 and 0.09 seconds. The second peak occurs around 0.3 seconds, with a magnitude half of the first peak. The displacement profile is shown in Fig. 5.7. The left side of Fig. 5.8 shows underwater

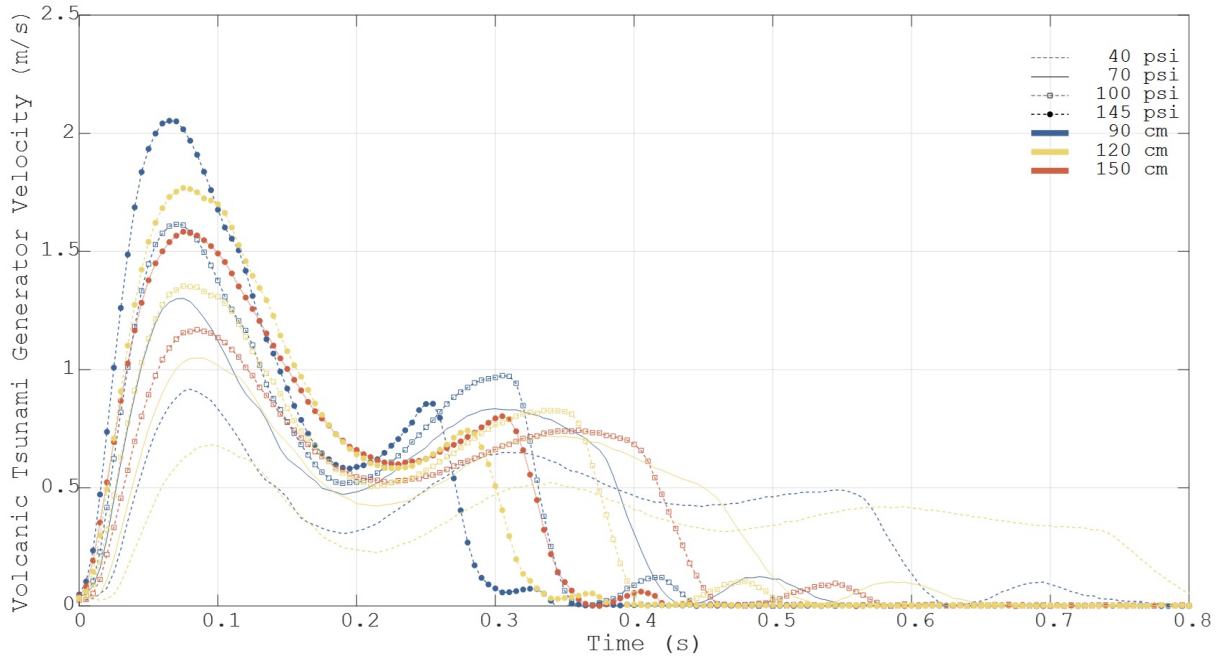


Figure 5.6: Velocity profile for the ten volcanic tsunami generator experiments with water depth and tank pressure.

Table 5.1: Physical experiment parameters for the eight cases that is numerically modeled.

Exp. N	Water Depth (cm)	Tank Pressure(psi)	Peak Velocity (m/s)
1	90	40	0.92
2	90	70	1.30
3	90	100	1.62
4	90	145	2.05
5	120	40	0.68
6	120	70	1.05
7	120	100	1.35
8	120	145	1.77

video time-series of volcanic tsunami generator erupting through the water surface with a depth of 0.97 m and launch pressure of 145 psi. The right side shows the video time-series (top to bottom) of the surface spike above the submerged VTG with a water depth of 1.17 m and a launch pressure of 130 psi.

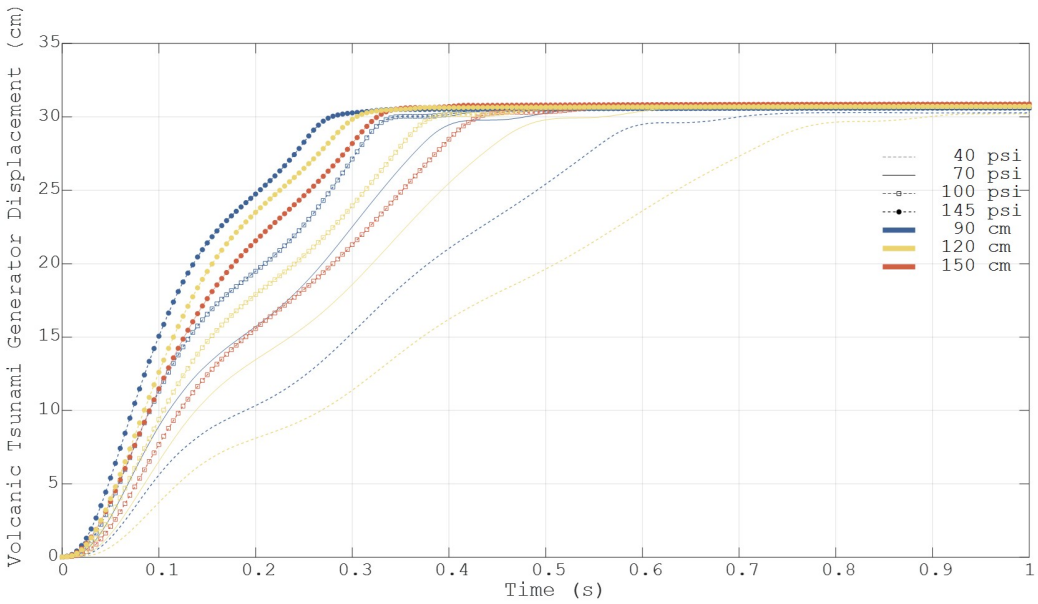


Figure 5.7: Displacement profile for the ten volcanic tsunami generator experiments with water depth and tank pressure.

5.3 Numerical Modeling of Volcanic Tsunami Experiments

The major objective of this study is to validate a fully 3D Navier-Stokes(N-S) numerical model, TSUNAMI3D, in modeling submarine volcanic tsunamis, using the set of physical laboratory experiments described in the earlier section. The N-S equations and continuity equations represented using the finite difference method are solved in TSUNAMI3D, and the transient water surfaces are tracked using the VOF method deveoped by [69] at the Los Alamos National Laboratory. VOF method is explained in chapter one 1.5. The Fractional Area Volume Obstacle Representation (FAVOR) technique defines internal obstacles such as walls, floor, and topography using partial and completely blocked grid cells. For the submarine volcanic tsunami experiments, the VTG and its motion are numerically modeled using this technique. The governing equations used in TSUNAMI3D are N-S equations and the continuity equations that are explained in chapter two 2.2.2.

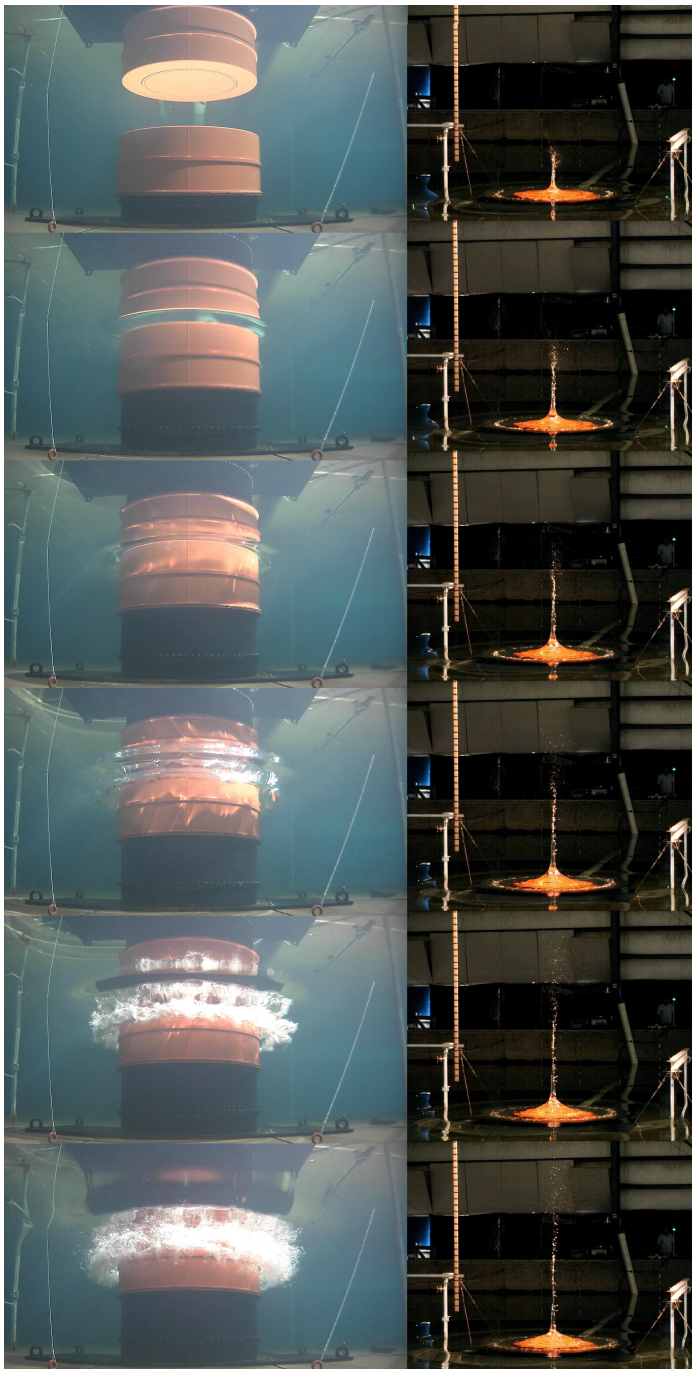


Figure 5.8: Left: Underwater video time-series of volcanic tsunami generator erupting through the water surface, Right: video time-series (top to bottom) of surface spike above the submerged VTG. (*Source: Yibin Liu, Georgia Tech*).

5.3.1 Numerical Model Input parameters

The computational domain represents the Hinsdale wave basin which is 48.4 m long and 26.5 m wide. The VTG in the retracted position is considered as the part of bathymetry as a cylindrical rise with a diameter of 1.2 m, centered at (9.9,13.25) with a height of 73 cm. The geometry and bathymetry of the wave basin, water depth, and the information about the location, shape, and motion of the VTG are the input parameters to the numerical model. VTG motion is modeled as a piston pushing from below. The walls of the computational domain are considered as reflective boundaries. The simulation time is 16 seconds. The initial wave state evolution close to the wave producing region is difficult to model. Very fine resolution is need at this location to capture highly non-linear characteristics of the wave [141]. However, for the tsunamis we are more concerned about the first few waves approaching the shore, and hence the grid resolution was chosen with this in mind. Cartesian grid system with uniform grid cell size is used for the simulations. Resolution along length and width is 5 cm and along depth is 1.25 cm. So, a total of 44 million cells are used for the simulation. The time step used for the simulation is 0.005 s. TSUNAMI3D takes around 24 hrs to simulate the 16 seconds on a Cray CX_1 computer with Intel Xeon CPUs having a total of 8 core utilizing two processors at 2.13 GHz.

5.3.2 Numerical Model Results

Numerical results are compared with the experimental water elevation time series measurements from the four gauges. The location of the gauges are given in the Table 5.2. The first three gauges are along the center line, and gauge 3 is at an angle of 12.58°. Gauge 0 is the closest to the VTG and gauge 3 is the farthest. The evolution of volcanic tsunami wave generated by the VTG for case 7 is shown through Fig, 5.9, 5.10 and 5.11. For this case the water depth is 120 cm and the tank pressure is 100 psi. The peak velocity is 1.35 m/s. The water level elevation with respect to the mean level is magnified 10 times for visualization. The color bar gives the elevations. Gauge 0 results is top left, gauge 1 is top right, gauge 2 is bottom left, and the gauge 3 is bottom right. The physical experimental measurements are shown as blue plots and the simulation results are

Table 5.2: Water level gauge location and numbers

Gauge No.	x (m)	y (m)	r (m)	θ (^{o})
0	11.81	13.22	1.91	0
1	13.93	13.25	4.02	0
2	16.88	13.24	6.98	0
3	19.32	11.12	19.44	12.58

red plots. The plot on the left bottom corner shows the VTG velocity with time. The deformation of the water surface during the motion of VTG is captured in Fig. 5.9.

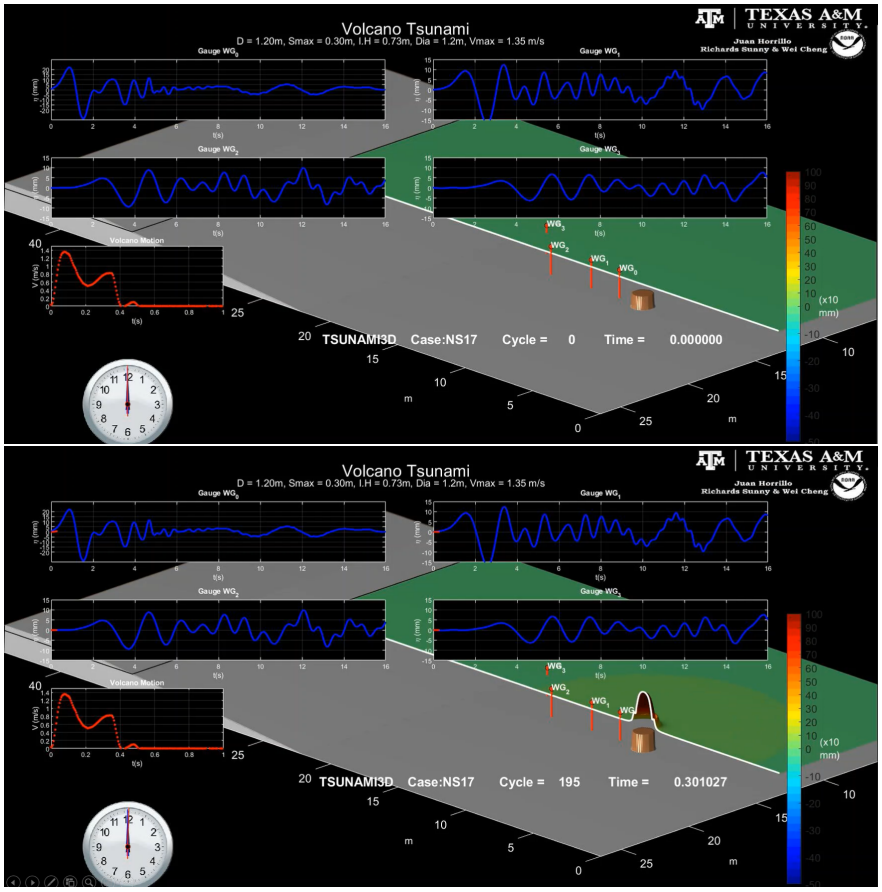


Figure 5.9: Numerical simulation results of case 7 with a water depth of 120 cm and the tank pressure of 100 psi. Results shows the time 0 to 0.3 s of the initial wave generation.

The second stage of the wave generation, the wave energy is dispersed in to the closer regions around VTG creating the first few largest waves. This can be seen from the Fig. 5.10.

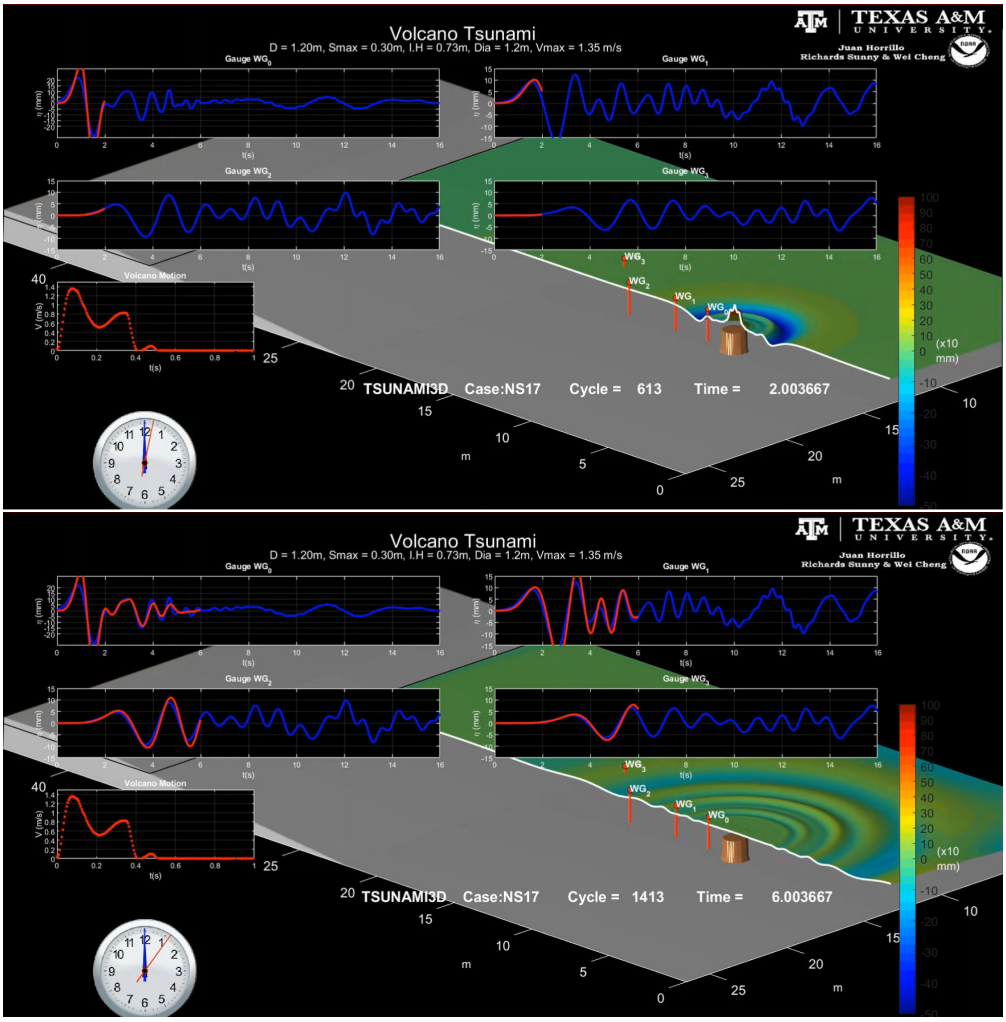


Figure 5.10: Numerical simulation results of case 7 with a water depth of 120 cm and the tank pressure of 100 psi. Results shows the time 2 to 6 s the initial large waves.

The third stage shows the reflected and superimposed waves. Some of the waves are propagating towards the beach and the circular waves are being transformed in to parallel waves. The wave elevation is increasing while moving through the region with slope toward the beach. These characteristics can be seen from the Fig. 5.11 where it shows snapshots from 10 to 16 seconds.

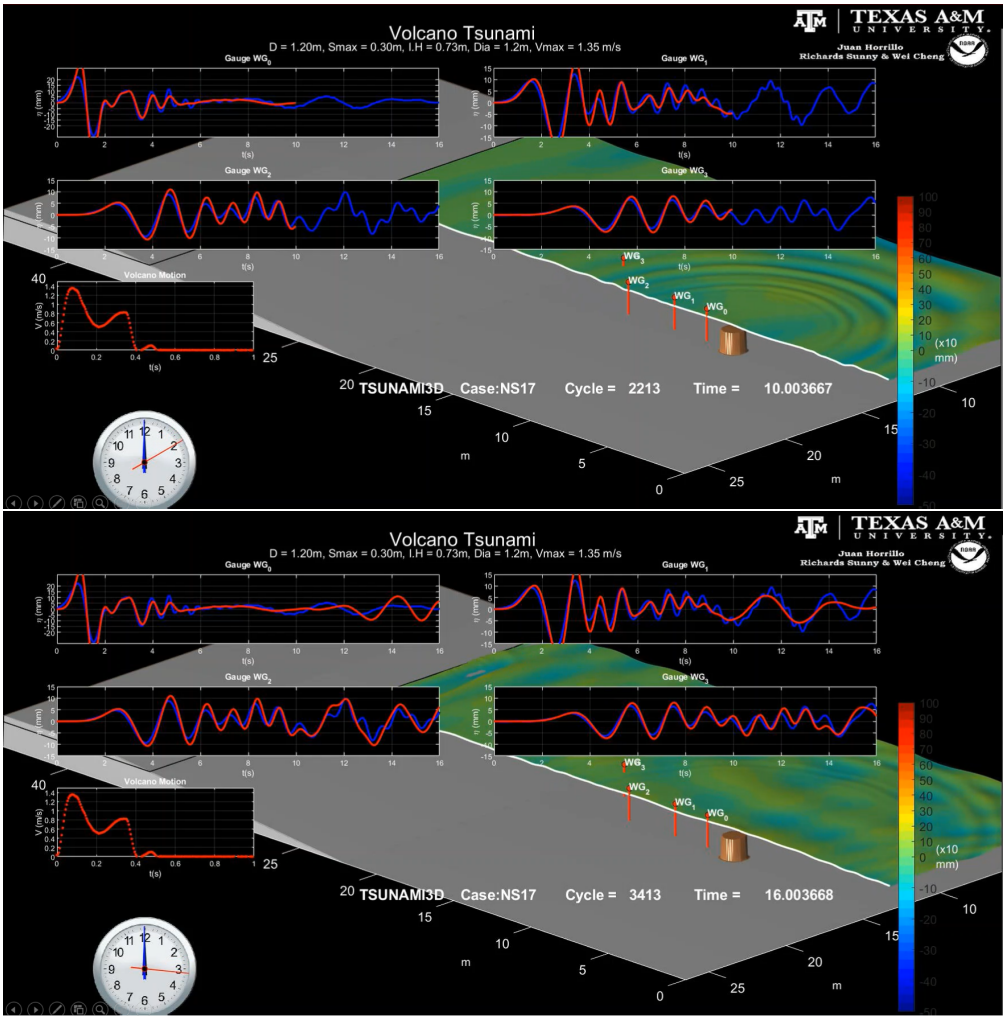


Figure 5.11: Numerical simulation results of case 7 with a water depth of 120 cm and the tank pressure of 100 psi. Results shows the time 10 to 16 s the initial large waves.

5.3.3 Comparison with the Wave Gauge Results

The comparison of the numerical results with the measurement from the submarine volcanic tsunami experiments for the cases 5 (table 5.1) is presented in this section. The results from the other cases can be found in the Appendix A. The wave gauge locations and number denoting each are provided in table 5.2. The time series from the wave gauge measurements and numerical model are plotted for the eight cases. The measurement of the first two crests and troughs from each experiments are compared with the numerical result and error percentage with respect to the

maximum wave height is also estimated. The gauges are denoted as g_i for convenience, where subscript 'i' is the gauge number.

Case 5 produces the smallest waves as it have the lowest pressure of 40 psi and higher water depth of 1.2 m and its results are shown in Fig. A.5. It is a non- piecing case with the lowest peak velocity of 0.68 m/s. The table A.5 shows the experimental measurement and the numerical results of the first two crests and troughs from case 5. The average absolute error percentage for gauges 0 to 3 in order are 9.8, 9.3, 8.4, and 7.8%.

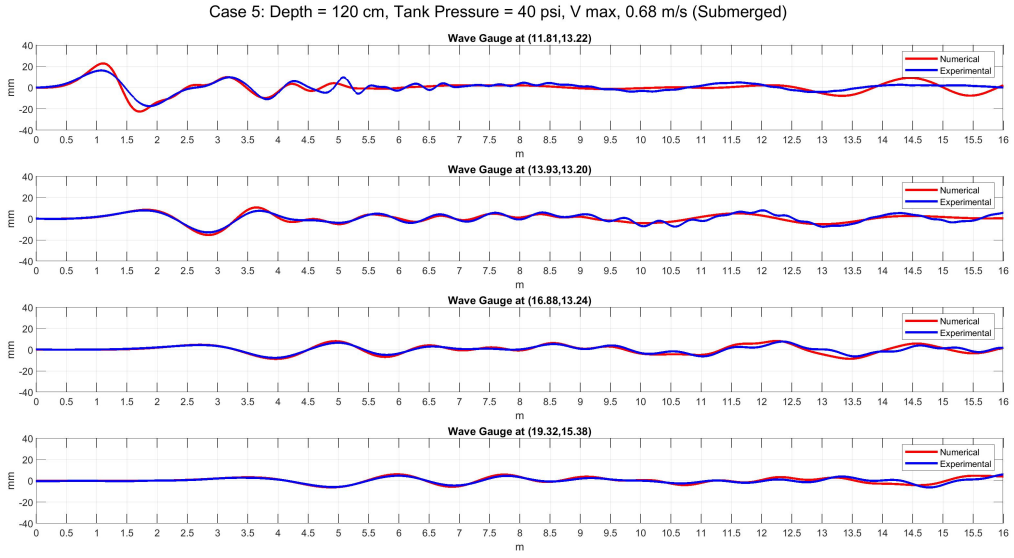


Figure 5.12: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 5 with water depth of 120 cm and tank pressure of 40 psi.

Table 5.3: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 5 with water depth of 120 cm and tank pressure of 40 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	22.79	16.13	19.9	-22.51	-17.39	-15.3	9.92	9.67	0.7	-9.74	-10.87	3.4
1	8.50	7.85	3.1	-15.38	-12.87	-12.1	10.72	7.45	15.8	-5.08	-3.81	-6.1
2	4.68	4.34	2.4	-8.70	-7.72	-6.9	8.07	6.53	10.8	-6.85	-4.92	-13.5
3	3.35	2.84	4.8	-6.17	-5.86	-3.0	6.13	4.78	12.7	-5.59	-4.45	-10.7

5.4 Conclusion

A submarine volcanic tsunami is a complex natural phenomenon. In this study, eight laboratory experiments of volcanic tsunami generation have been compared with results from the 3D numerical simulation with same initial conditions. The laboratory experiments were carried out at Hinsdale laboratory at the Oregon State University (OSU). The presented work focuses on the numerical validation of a 3D numerical model, TSUNAMI3D, in modeling 3D submarine volcanic tsunamis. Results attained from experiments and numerical simulations shows that submarine volcanic eruptions can create highly non-linear waves posing a challenging task to modelers for validation purpose. From the results, in general, TSUNAMI3D models have successfully simulated the waves generated by the volcanic tsunami generator (VTG). The wave gauges in the nearby regions that experience high non-linearity shows more error than the ones in the far-field regions. For case 5, the average percentage absolute error(two crests and troughs) with respect to the experimental wave height for the wave gauges are 9.85%, 9.27 %, 7.41 %, 7.8 %. For the water surface piercing cases, the wave gauges close to VTG predict a higher crest and trough. This can be understood from the fact that the energy lost by the breaking of water is not accounted in the numerical model.

The overall results shows an average error around 10 percent. Thus, it can be concluded that the TSUNAMI3D is able to model submarine volcanic tsunami experiments satisfactorily. The future work could include the simulation with different parameters such as the initial VTG height, expansion distance, and diameter of the piston. Subsequently, the actual volcano (Kick'em Jenny) bathymetry can be used, along with the scaled numerical VTG according to Froude number, and the inundation maps and wave run-up results from the simulation could be useful to prepare for a submarine volcanic tsunami event.

6. SUMMARY AND CONCLUSION

6.1 Summary

This dissertation proposes a new surface tracking, and advection method for incompressible and immiscible fluid flows, named the VOF-with-center-of-mass-and-Lagrangian-particles (VCLP) method. The VCLP is based on the volume of fluid (VOF) method ans couples Eulerian and Lagrangian approaches to localized Lagrangian particles (LPs) inside a finite volume framework. For a given time step, the computational domain input parameters are 1) spatially discretized fluid distribution (f matrix), 2) the discretized velocity field, and 3) the fluid center of mass (CM) location within the grid cells. The final output parameters are the 1) new spatially discretized fluid distribution (new f) and 2) the new fluid CM location within the grid cells. The major steps in the VCLP method to produce these outputs are the classification of grid cells, interface reconstruction representing fluid regions using LPs, correcting the location of LPs using fluid CM tracking, fluid advection via LPs, updating fluid distribution and center of mass, and filtering to achieve perfect mass conservation. VCLP completes all steps mentioned above for a single fluid cell and then moves to the next cell. This facilitates the VCLP method to be computationally parallelized. Cells are classified to identify if a cell has fluid and interface, as only cells with fluid need advection, and cells with interface require its reconstruction. Therefore, all cells in the computational domain are classified into empty, surface, and interior fluid cells. VCLP method further classifies the interior cells into three more kinds to increase the computational performance by skipping one kind of interior cells that satisfy certain conditions using the proposed skip core optimization technique in this study. After the classification, interfaces are reconstructed for all cells classified as surface cells. In the VCLP method, there are two choices for the calculation of interface reconstruction; the Piecewise-Linear-Interface-Calculation (PLIC) method and the Piecewise-Circular-Interface-Calculation method (PCIC). Several existing surface tracking methods use PLIC, and it creates a line segment representing the interface in 2D and a plane in 3D, whereas PCIC, which is a pro-

posed approach, creates a circular arc for the interface in 2D and a spherical surface in 3D. A PLIC construction can be seen as an equivalent PCIC construction withe radius of the of the circular arc approaching infinity. Along with improving the accuracy of interface reconstruction, the radius of curvature from the PCIC could also be used to estimate the surface tension force. In the interface reconstruction, VCLP uses new methods to estimate the interface normal (T-Slope method), find interface curvatures, and locate the interface (Sorted Surface Constant method). After the interface reconstruction, the fluid regions bounded by the interface are advected. Before advection, the fluid within the cells is represented using LPs as concentrated mass points. A template of LPs is generated with blue noise distribution (using a proposed technique, J-method) to represent the fluid. LPs are advected cell by cell, and after advection of each cell, the same LPs template is used for the next cell advection. This is an essential difference to the common Lagrangian approaches, where the entire fluid domain or interface is represented simultaneously by LPs, and all LPs used are tracked through the simulation time, requiring a vast number of LPs proportional to the domain size. VCLP uses only one blue noise template with a customizable number of LPs. Only 10 to 1000 LPs are required for the entire simulation irrespective of the domain size in 2D and 1000 to 8000 LPs in 3D. This enables large computational domains feasible with meager memory allocation. Because of the customization, VCLP gives control to the user over computational speed vs. accuracy by changing the number of LPs used for the given grid resolution. After the fluid region is represented using LPs, a correction is applied to the location of LPs to improve the accuracy of the interface reconstruction. This correction is based on tracking the fluid center of mass (CM), which follows the Lagrangian approach. In theory, if the grid cells or the control volumes are infinitesimal in size, then tracking of the CM in a Lagrangian approach is essentially tracking the entire fluid. So, it can be said that VCLP tracks fluid in the Lagrangian way, but with a minimal number of points, one per cell. The interface reconstruction introduces inaccuracies or errors due to the simplification of the interface into PLIC or PCIC, constructed from the mean slope and curvature, which is also an approximation. Even though the correction due to CM tracking is minimal, results show that it is not negligible over thousands of time steps which is the case for typical numerical

simulations. Moreover, the correction also tackles the error associated with the numerical acceleration or deceleration of fluid due to the interface reconstruction, which reduces the cumulative error of pressures and velocities. After correcting LPs locations using CM, the next step is to advect the LPs representing the fluid according to the velocity field. While advecting LPs, cell size is not a constraint, and hence VCLP method could work for CFL numbers larger than one. VCLP method uses one of the two numerical advection schemes, Taylor's second-order numerical advection scheme or a new scheme named Tracking-Journey-in-Spiral (TJSM) method. TJSM method is a proposed numerical advection scheme to integrate the equation of motion numerically. This scheme tries to capture rotational effects inside a velocity field which may be more intuitive for fluid flows where vorticity is present. TJSM assumes that the particle undergoes a circular motion with constant angular velocity during a time-step interval instead of a linear motion. In the VCLP method, the velocity of the LPs is a function of their position within the cell. However, for most N-S finite volume solvers, velocity is known only at specific discretized locations within the cell. In this case, individual LP's velocity is interpolated using bilinear (2D) and trilinear (3D) methods. The advection of fluid using LPs enables the VCLP method applicable to structured and unstructured grid domains. As the fluid mass inside a cell is precisely equal to the sum of masses of LPs representing it, perfect mass conservation is ensured if all advected LPs are received by the cells in the domain after advection. Because of this, the VCLP method can conserve mass perfectly. The numerical advection schemes used in VCLP contain both velocity and acceleration terms. Since the acceleration is not obtained from the N-S solver, the VCLP method uses newly derived equations (for the Cartesian grid) to find approximate acceleration to advect each LP within in the cell. The fluid mass distribution (new f) and CM location within the cells are updated after the advection. The advection of LPs might result in cells not satisfying the volume constraints due to the non-zero velocity divergence in certain regions arising from the interpolation of the velocity field. These non-divergent velocity fields are difficult to avoid, therefore a filter function is used to redistribute excess of fluid mass that satisfies the volume constraints.

6.2 Conclusion

The accuracy of the VCLP method for interface reconstruction and advection is evaluated based on five benchmark tests in 2D and 3D. The 2D tests are translation, rotation, single vortex, deformation, and Zalesak's disk test and the 3D test is deformation. In these a tests, a well-defined 2D fluid body inside a computational domain with specified grid resolution and dimensions is placed in a solenoidal (non-divergent) velocity field. The flow is then simulated for a given time T with specified time intervals. The ideal final state of the fluid body at time T is known and the performance of the method is evaluated based on its similarity to the test result's final state. The results of 2D translation, rotation, deformation and 3D deformation tests show that VCLP performs qualitatively well compared to conventional methods such as Hirt and Nichols (H&N), Young's, and newer methods such as the moment of fluid (MoF) and Coupled Level Set Volume of Fluid methods (CLSVOF). Later, the quantitative error comparisons support this observation.

From the solenoidal 2D single vortex field test results, VCLP has lesser geometric error than H&N, Young's, 1^{st} order upwind, PPM, Level sets, SLIC, and PLIC. For example, the 2D single vortex test with the grid resolution of 64×64 and a maximum simulation time of $T = 8$ s, the geometric error of the VCLP-PCIC (VCLP + T2) is 0.00197, while the errors from other methods, as H&N without and with height function, Young's method without and with height function, and Rider & Kothe methods are 40.4, 30.7, 10.1, 22.8, and 3.53 times 0.00197, respectively. Quantitative results of the deformation and Zalesak's disk test using VCLP also show less or similar error than the other methods in 2D and 3D. From the Zalesak's test result for grid resolution of 200×200, VCLP errors are smaller than successful methods from literature such as Youngs, Puckett, ELVIRA, gVoFoam, interFoam, OpenFoam, Quadratic fit, and Quadratic fit with continuity. VCLP-PCIC errors are even smaller than the completely Lagrangian advection methods such as the Quadratic fit, and Quadratic fit with continuity. For deformation and Zalesak's disk tests, TJSM performs better than Taylor's second order advection scheme. Also, VCLP-PCIC have lesser error than VCLP-PLIC for Zalesak's disk tests. This proves that the proposed advection scheme TJSM and the reconstruction method, PCIC are successful.

The computational performance study of the VCLP method for the single 2D vortex test for resolution of 60×60 shows that the VCLP method with 10 LPs, Taylor 2^{nd} order advection, and CM tracking takes 19 % more time than SOLA-VOF, but the error reduced by 1289 %. The CM tracking seems to reduce the error from 2.68 to 6 times with less than five percent more of computational time. For a resolution of 120×120, SOLA-VOF takes 105.5 s with an error of 0.07924, which is 629 % more computational time and 1200 % times more error than VCLP with 10 LPs per dimension. This supports the claim that using VCLP at lower resolution obtains better results than SOLA-VOF with lesser computational time. Overall, from the computational performance analysis in 2D, the VCLP method shows considerable improvement with a slightly more computation time.

From the dam break comparison between TSUNAMI2D-VCLP and TSUNAMI2D-(SOLA-VOF) methods, the surfaces in the VCLP result are smoother and devoid of some numerical errors that causes unrealistic flows regions. So, it was understood from the simulation supported by the evidence from the benchmark tests, that VCLP is able to produce very similar result to SOLA-VOF, with some improvement.

The challenging breaking wave problem was simulated using TSUNAMI2D-VCLP and results were compared with the FNFP-BEM and the strong coupled FNPF-BEM/NS-VOF methods from literature. The analysis showed that even with higher-order solvers, the results are very comparable. Thus, the applications of the VCLP method in the TSUNAMI2D solver show satisfactory results.

REFERENCES

[1] J. Gerrits and A. Veldman, "Dynamics of liquid-filled spacecraft," *Journal of Engineering Mathematics*, vol. 45, no. 1, pp. 21–38, 2003.

[2] A. A. Mukundan, T. Ménard, J. C. B. de Motta, and A. Berlemont, "A 3d moment of fluid method for simulating complex turbulent multiphase flows," *Computers & Fluids*, vol. 198, p. 104364, 2020.

[3] W. J. Rider and D. B. Kothe, "Reconstructing volume tracking," *Journal of computational physics*, vol. 141, no. 2, pp. 112–152, 1998.

[4] R. Scardovelli and S. Zaleski, "Interface reconstruction with least-square fit and split eulerian–lagrangian advection," *International Journal for Numerical Methods in Fluids*, vol. 41, no. 3, pp. 251–274, 2003.

[5] S. T. Grilli and R. Subramanya, "Numerical modeling of wave breaking induced by fixed or moving boundaries," *Computational Mechanics*, vol. 17, no. 6, pp. 374–391, 1996.

[6] C. Lachaume, B. Biausser, P. Fraunié, S. T. Grilli, and S. Guignard, "Modeling of breaking and post-breakingwaves on slopes by coupling of bem and vof methods," in *The Thirteenth International Offshore and Polar Engineering Conference*, OnePetro, 2003.

[7] W. Rider and D. Kothe, "Stretching and tearing interface tracking methods," in *12th Computational Fluid Dynamics Conference*, p. 1717, 1995.

[8] J. D. Anderson Jr, *Fundamentals of aerodynamics*. Tata McGraw-Hill Education, 2010.

[9] D. Peregrine and L. Thais, "The effect of entrained air in violent water wave impacts," *Journal of Fluid Mechanics*, vol. 325, pp. 377–397, 1996.

[10] D. J. Wood, D. H. Peregrine, and T. Bruce, "Wave impact on a wall using pressure-impulse theory. i: trapped air," *Journal of waterway, port, coastal, and ocean engineering*, vol. 126, no. 4, pp. 182–190, 2000.

[11] G. Badin and F. Crisciani, "Variational formulation of fluid and geophysical fluid dynamics," *Mech. Symmetries Conservation Laws*, 2018.

[12] C. K. Batchelor and G. Batchelor, *An introduction to fluid dynamics*. Cambridge university press, 2000.

[13] C. T. Crowe, *Multiphase flow handbook*. CRC press, 2005.

[14] M. Wörner, *A compact introduction to the numerical modeling of multiphase flows*, vol. 6932. FZKA, 2003.

[15] S. V. Patankar, *Numerical heat transfer and fluid flow*. CRC press, 2018.

[16] A. Barrett, "The finite element method for engineers–third edition. kh huebner, ea thornton and eg byrom. john wiley & sons, baffins lane, chichester, west sussex po19 1ud. 1995. 627pp. illustrated.£ 53.95.," *The Aeronautical Journal*, vol. 99, no. 989, pp. 412–412, 1995.

[17] J. T. Katsikadelis, *Boundary elements: theory and applications*. Elsevier, 2002.

[18] J. E. Castillo, *Mathematical aspects of numerical grid generation*. SIAM, 1991.

[19] P. He, M. Salcudean, I. Gartshore, and P. Nowak, "Multigrid calculation of fluid flows in complex 3d geometries using curvilinear grids," *Computers & fluids*, vol. 25, no. 4, pp. 395–419, 1996.

[20] D. K. Clarke, M. Salas, and H. Hassan, "Euler calculations for multielement airfoils using cartesian grids," *AIAA journal*, vol. 24, no. 3, pp. 353–358, 1986.

[21] K. Morinishi, "A finite difference solution of the euler equations on non-body-fitted cartesian grids," *Computers & fluids*, vol. 21, no. 3, pp. 331–344, 1992.

[22] H. S. Udaykumar, H.-C. Kan, W. Shyy, and R. Tran-Son-Tay, "Multiphase dynamics in arbitrary geometries on fixed cartesian grids," *Journal of Computational Physics*, vol. 137, no. 2, pp. 366–405, 1997.

[23] R. Verstappen and A. Veldman, *Numerical computation of a viscous flow around a circular cylinder on a Cartesian grid*. Citeseer, 2000.

[24] J. Benek, J. Steger, F. Dougherty, and P. Buning, "Chimera. a grid-embedding technique," tech. rep., ARNOLD ENGINEERING DEVELOPMENT CENTER ARNOLD AFB TN, 1986.

[25] D. De Zeeuw and K. P.-A. A.-R. Cartesian, "Mesh solver for the euler equations," tech. rep., AIAA-91-1542, 1991.

[26] O. Ubbink and R. Issa, "A method for capturing sharp fluid interfaces on arbitrary meshes," *Journal of computational physics*, vol. 153, no. 1, pp. 26–50, 1999.

[27] D. E. Fyfe, E. S. Oran, and M. Fritts, "Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh," *Journal of Computational Physics*, vol. 76, no. 2, pp. 349–384, 1988.

[28] P. Hansbo, "Lagrangian incompressible flow computations in three dimensions by use of space-time finite elements," *International journal for numerical methods in fluids*, vol. 20, no. 8-9, pp. 989–1001, 1995.

[29] T. Nakayama and K. Washizu, "Nonlinear analysis of liquid motion in a container subjected to forced pitching oscillation," *International Journal for Numerical Methods in Engineering*, vol. 15, no. 8, pp. 1207–1220, 1980.

[30] F. H. Harlow, "Mac numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *Phys. Fluid*, vol. 8, p. 12, 1965.

[31] C. S. Peskin, "Numerical analysis of blood flow in the heart," *Journal of computational physics*, vol. 25, no. 3, pp. 220–252, 1977.

[32] 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 computations of multiphase flow," *Journal of computational physics*, vol. 169, no. 2, pp. 708–759, 2001.

[33] C. Hirt, A. Amsden, and J. Cook, "An arbitrary lagrangian–eulerian computing method for all flow speeds," *Journal of computational physics*, vol. 135, no. 2, pp. 203–216, 1997.

[34] J. Baiges, R. Codina, A. Pont, and E. Castillo, "An adaptive fixed-mesh ale method for free surface flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 313, pp. 159–188, 2017.

[35] P. Nithiarasu, "An arbitrary lagrangian eulerian (ale) formulation for free surface flows using the characteristic-based split (cbs) scheme," *International Journal for Numerical Methods in Fluids*, vol. 48, no. 12, pp. 1415–1428, 2005.

[36] S. O. Unverdi and G. Tryggvason, "A front-tracking method for viscous, incompressible, multi-fluid flows," *Journal of computational physics*, vol. 100, no. 1, pp. 25–37, 1992.

[37] A. M. Roma, C. S. Peskin, and M. J. Berger, "An adaptive version of the immersed boundary method," *Journal of computational physics*, vol. 153, no. 2, pp. 509–534, 1999.

[38] C. S. Peskin and B. F. Printz, "Improved volume conservation in the computation of flows with immersed elastic boundaries," *Journal of computational physics*, vol. 105, no. 1, pp. 33–46, 1993.

[39] D. Juric and G. Tryggvason, "A front-tracking method for dendritic solidification," *Journal of computational physics*, vol. 123, no. 1, pp. 127–148, 1996.

[40] E. Marchandise, J.-F. Remacle, and N. Chevaugeon, "A quadrature-free discontinuous galerkin method for the level set equation," *Journal of Computational Physics*, vol. 212, no. 1, pp. 338–357, 2006.

[41] S. Chen, D. B. Johnson, P. E. Raad, and D. Fadda, "The surface marker and micro cell method," *International Journal for Numerical Methods in Fluids*, vol. 25, no. 7, pp. 749–778, 1997.

[42] R. A. Gingold and J. J. Monaghan, "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly notices of the royal astronomical society*, vol. 181, no. 3, pp. 375–389, 1977.

[43] L. B. Lucy, "A numerical approach to the testing of the fission hypothesis," *The astronomical journal*, vol. 82, pp. 1013–1024, 1977.

[44] T. Harada, S. Koshizuka, and Y. Kawaguchi, "Improvement in the boundary conditions of smoothed particle hydrodynamics," *Computer Graphics & Geometry*, vol. 9, no. 3, pp. 2–15, 2007.

[45] A. C. Crespo, J. M. Dominguez, A. Barreiro, M. Gómez-Gesteira, and B. D. Rogers, "Gpus, a new tool of acceleration in cfd: efficiency and reliability on smoothed particle hydrodynamics methods," *PloS one*, vol. 6, no. 6, p. e20685, 2011.

[46] D. J. Price, "Smoothed particle hydrodynamics: things i wish my mother taught me," *arXiv preprint arXiv:1111.1259*, 2011.

[47] F. H. Harlow, "The particle-in-cell computing method for fluid dynamics," *Methods Comput. Phys.*, vol. 3, pp. 319–343, 1964.

[48] D. Snider, "An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows," *Journal of computational physics*, vol. 170, no. 2, pp. 523–549, 2001.

[49] M. Sussman, E. Fatemi, P. Smereka, and S. Osher, "An improved level set method for incompressible two-phase flows," *Computers & Fluids*, vol. 27, no. 5-6, pp. 663–680, 1998.

[50] M. Sussman and P. Smereka, "Axisymmetric free boundary problems," *Journal of Fluid Mechanics*, vol. 341, pp. 269–294, 1997.

[51] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of computational physics*, vol. 79, no. 1, pp. 12–49, 1988.

[52] M. Sussman and E. G. Puckett, "A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows," *Journal of computational physics*, vol. 162, no. 2, pp. 301–337, 2000.

[53] C. Schroeder, W. Zheng, and R. Fedkiw, "Semi-implicit surface tension formulation with a lagrangian surface mesh on an eulerian simulation grid," *Journal of Computational Physics*, vol. 231, no. 4, pp. 2092–2115, 2012.

[54] L. Wang, A. Khayyer, H. Gotoh, Q. Jiang, and C. Zhang, "Enhancement of pressure calculation in projection-based particle methods by incorporation of background mesh scheme," *Applied Ocean Research*, vol. 86, pp. 320–339, 2019.

[55] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell, "A hybrid particle level set method for improved interface capturing," *Journal of Computational physics*, vol. 183, no. 1, pp. 83–116, 2002.

[56] D. Enright, F. Losasso, and R. Fedkiw, "A fast and accurate semi-lagrangian particle level set method," *Computers & structures*, vol. 83, no. 6-7, pp. 479–490, 2005.

[57] Z. Li, F. A. Jaberi, and T. I.-P. Shih, "A hybrid lagrangian–eulerian particle-level set method for numerical simulations of two-fluid turbulent flows," *International journal for numerical methods in fluids*, vol. 56, no. 12, pp. 2271–2300, 2008.

[58] R. Bermejo and J. L. Prieto, "A semi-lagrangian particle level set finite element method for interface problems," *SIAM Journal on Scientific Computing*, vol. 35, no. 4, pp. A1815–A1846, 2013.

[59] S. E. Hieber and P. Koumoutsakos, "A lagrangian particle level set method," *Journal of Computational Physics*, vol. 210, no. 1, pp. 342–367, 2005.

[60] F. H. Harlow and J. E. Welch, "Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface," *The physics of fluids*, vol. 8, no. 12, pp. 2182–2189, 1965.

[61] J. P. Morris, "Simulating surface tension with smoothed particle hydrodynamics," *International journal for numerical methods in fluids*, vol. 33, no. 3, pp. 333–353, 2000.

[62] V. Dyadechko and M. Shashkov, "Reconstruction of multi-material interfaces from moment data," *Journal of Computational Physics*, vol. 227, no. 11, pp. 5361–5384, 2008.

[63] E. Wenzel and S. Garrick, "A point-mass particle method for the simulation of immiscible multiphase flows on an eulerian grid," *Journal of Computational Physics*, vol. 397, p. 108835, 2019.

[64] D. B. Kothe, "Perspective on eulerian finite volume methods for incompressible interfacial flows," in *Free surface flows*, pp. 267–331, Springer, 1998.

[65] D. Kothe, W. Rider, S. Mosso, J. Brock, and J. Hochstein, "Volume tracking of interfaces having surface tension in two and three dimensions," in *34th aerospace sciences meeting and exhibit*, p. 859, 1996.

[66] M. Rudman, "A volume-tracking method for incompressible multifluid flows with large density variations," *International Journal for numerical methods in fluids*, vol. 28, no. 2, pp. 357–378, 1998.

[67] R. Scardovelli and S. Zaleski, "Direct numerical simulation of free-surface and interfacial flow," *Annual review of fluid mechanics*, vol. 31, no. 1, pp. 567–603, 1999.

[68] E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski, "A geometrical area-preserving volume-of-fluid advection method," *Journal of Computational Physics*, vol. 192, no. 1, pp. 355–364, 2003.

[69] C. W. Hirt and B. D. Nichols, "Volume of fluid (vof) method for the dynamics of free boundaries," *Journal of computational physics*, vol. 39, no. 1, pp. 201–225, 1981.

[70] B. Nichols, C. Hirt, and R. Hotchkiss, "Sola-vof: A solution algorithm for transient fluid flow with multiple free boundaries," tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1980.

[71] M. D. Torrey, L. D. Cloutman, R. C. Mjolsness, and C. Hirt, "Nasa-vof2d: a computer program for incompressible flows with free surfaces," *NASA STI/Recon Technical Report N*, vol. 86, 1985.

[72] S. Elliot, *Fluid flow aspects of solidification modeling: Simulation of low pressure die casting*. PhD thesis, Ph. D. Thesis, Department of Mining & Metallurgical Engineering, University ..., 1999.

[73] D. J. Harvie and D. F. Fletcher, "A new volume of fluid advection algorithm: the defined donating region scheme," *International Journal for Numerical Methods in Fluids*, vol. 35, no. 2, pp. 151–172, 2001.

[74] D. J. Harvie and D. F. Fletcher, "A new volume of fluid advection algorithm: the stream scheme," *Journal of Computational Physics*, vol. 162, no. 1, pp. 1–32, 2000.

[75] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, "Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows," *Journal of Computational physics*, vol. 152, no. 2, pp. 423–456, 1999.

[76] R. Scardovelli and S. Zaleski, "Analytical relations connecting linear interfaces and volume fractions in rectangular grids," *Journal of Computational Physics*, vol. 164, no. 1, pp. 228–237, 2000.

[77] L.-P. Wang, S. Chen, J. G. Brasseur, and J. C. Wyngaard, "Examination of hypotheses in the kolmogorov refined turbulence theory through high-resolution simulations. part 1. velocity field," *Journal of Fluid Mechanics*, vol. 309, pp. 113–156, 1996.

[78] J. Pedlosky *et al.*, *Geophysical fluid dynamics*, vol. 710. Springer, 1987.

[79] L. J. Clancy, *Aerodynamics*. John Wiley & Sons, 1975.

[80] J. J. Horrillo, *Numerical method for tsunami calculation using full Navier-Stokes equations and the volume of fluid method*. University of Alaska Fairbanks, 2006.

[81] J. Sicilian and C. Hirt, "An efficient computation scheme for tracking contaminant concentrations in fluid flows," *Journal of Computational Physics*, vol. 56, no. 3, pp. 428–447, 1984.

[82] R. A. Gentry, R. E. Martin, and B. J. Daly, "An eulerian differencing method for unsteady compressible flow problems," *Journal of computational Physics*, vol. 1, no. 1, pp. 87–118, 1966.

[83] R. L. Cook, "Stochastic sampling in computer graphics," *ACM Transactions on Graphics (TOG)*, vol. 5, no. 1, pp. 51–72, 1986.

[84] J. E. Mebius, "Derivation of the euler-rodrigues formula for three-dimensional rotations from the general formula for four-dimensional rotations," *arXiv preprint math/0701759*, 2007.

[85] J. B. Bell, P. Colella, and H. M. Glaz, "A second-order projection method for the incompressible navier-stokes equations," *Journal of Computational Physics*, vol. 85, no. 2, pp. 257–283, 1989.

[86] P. K. Smolarkiewicz, "The multi-dimensional crowley advection scheme," *Monthly Weather Review*, vol. 110, no. 12, pp. 1968–1983, 1982.

[87] R. J. LeVeque, "High-resolution conservative algorithms for advection in incompressible flow," *SIAM Journal on Numerical Analysis*, vol. 33, no. 2, pp. 627–665, 1996.

[88] Z. Wang, J. Yang, and F. Stern, "A new volume-of-fluid method with a constructed distance function on general structured grids," *Journal of Computational Physics*, vol. 231, no. 9, pp. 3703–3722, 2012.

[89] Y. Zhao and H.-C. Chen, "Violent free surface flow simulations by a coupled level-set and volume-of-fluid method in overset grid systems," *International Journal of Offshore and Polar Engineering*, vol. 24, no. 02, pp. 114–121, 2014.

[90] T. Ménard, S. Tanguy, and A. Berlemont, "Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet," *International Journal of Multiphase Flow*, vol. 33, no. 5, pp. 510–524, 2007.

[91] S. T. Zalesak, "Fully multidimensional flux-corrected transport algorithms for fluids," *Journal of computational physics*, vol. 31, no. 3, pp. 335–362, 1979.

[92] F. Garoosi and T.-F. Mahdi, "Presenting a novel higher-order bounded convection scheme for simulation of multiphase flows and convection heat transfer," *International Journal of Heat and Mass Transfer*, vol. 172, p. 121163, 2021.

[93] P. Cifani, W. Michalek, G. Priems, J. G. Kuerten, C. van der Geld, and B. J. Geurts, "A comparison between the surface compression method and an interface reconstruction method for the vof approach," *Computers & Fluids*, vol. 136, pp. 421–435, 2016.

[94] M.-J. Li, "Interaction between free surface flow and moving bodies with a dynamic mesh and interface geometric reconstruction approach," *Computers & Mathematics with Applications*, vol. 81, pp. 649–663, 2021.

[95] V. Dyadechko and M. Shashkov, "Moment-of-fluid interface reconstruction," *Los Alamos Report LA-UR-05-7571*, 2005.

[96] S. Guignard, R. Marcer, V. Rey, C. Kharif, and P. Fraunié, "Solitary wave breaking on sloping beaches: 2-d two phase flow numerical simulation by sl-vof method," *European Journal of Mechanics-B/Fluids*, vol. 20, no. 1, pp. 57–74, 2001.

[97] C. De Jouëtte, H. Viviand, S. Wornom, and J. Le Gouez, "Pseudo compressibility method for incompressible flow calculation," in *Proc 4th International Symposium on Computational Fluid Dynamics, University of California, Davis*, 1991.

[98] S. Thorarinsson, "On the damage caused by volcanic eruptions with special reference to tephra and gases," in *Volcanica activity and human ecology*, pp. 125–59, Elsevier, 1979.

[99] J. Latter, "Tsunamis of volcanic origin: summary of causes, with particular reference to krakatoa, 1883," *Bulletin volcanologique*, vol. 44, no. 3, pp. 467–490, 1981.

[100] J.-C. Tanguy, C. Ribière, A. Scarth, and W. Tjetjep, "Victims from volcanic eruptions: a revised database," *Bulletin of volcanology*, vol. 60, no. 2, pp. 137–144, 1998.

[101] J. W. Johnson and K. Bermel, "Impulsive waves in shallow water as generated by falling weights," *Eos, Transactions American Geophysical Union*, vol. 30, no. 2, pp. 223–230, 1949.

[102] S. N. Ward and S. Day, "Ritter island volcano—lateral collapse and the tsunami of 1888," *Geophysical Journal International*, vol. 154, no. 3, pp. 891–902, 2003.

[103] B. H. Choi, E. Pelinovsky, K. Kim, and J. Lee, "Simulation of the trans-oceanic tsunami propagation due to the 1883 krakatau volcanic eruption," *Natural Hazards and Earth System Sciences*, vol. 3, no. 5, pp. 321–332, 2003.

[104] S. T. Grilli, D. R. Tappin, S. Carey, S. F. Watt, S. N. Ward, A. R. Grilli, S. L. Engwell, C. Zhang, J. T. Kirby, L. Schambach, *et al.*, "Modelling of the tsunami from the december 22, 2018 lateral collapse of anak krakatau volcano in the sunda straits, indonesia," *Scientific reports*, vol. 9, no. 1, pp. 1–13, 2019.

[105] H. Sigurdsson, B. Houghton, S. McNutt, H. Rymer, and J. Stix, *The encyclopedia of volcanoes*. Elsevier, 2015.

[106] E. A. Okal and C. E. Synolakis, "Source discriminants for near-field tsunamis," *Geophysical Journal International*, vol. 158, no. 3, pp. 899–912, 2004.

[107] C. B. Harbitz, F. Løvholt, G. Pedersen, and D. G. Masson, "Mechanisms of tsunami generation by submarine landslides: a short review.," *Norwegian Journal of Geology/Norsk Geologisk Forening*, vol. 86, no. 3, 2006.

[108] M. Di Risio, P. De Girolamo, and G. M. Beltrami, *Forecasting landslide generated tsunamis: a review*. IntechOpen, 2011.

[109] J. Kienle, Z. Kowalik, and T. Murty, "Tsunamis generated by eruptions from mount st. augustine volcano, alaska," *Science*, vol. 236, no. 4807, pp. 1442–1447, 1987.

[110] J. C. Lahr, B. A. Chouet, C. D. Stephens, J. A. Power, and R. A. Page, "Earthquake classification, location, and error analysis in a volcanic environment: Implications for the magmatic system of the 1989–1990 eruptions at redoubt volcano, alaska," *Journal of Volcanology and Geothermal Research*, vol. 62, no. 1-4, pp. 137–151, 1994.

[111] D. C. Roman and K. V. Cashman, "The origin of volcano-tectonic earthquake swarms," *Geology*, vol. 34, no. 6, pp. 457–460, 2006.

[112] B. H. Keating and W. McGuire, "Island edifice failures and associated tsunami hazards," *Pure and Applied Geophysics*, vol. 157, no. 6, pp. 899–955, 2000.

[113] F. Løvholt, G. Pedersen, C. B. Harbitz, S. Glimsdal, and J. Kim, "On the characteristics of landslide tsunamis," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2053, p. 20140376, 2015.

[114] R. Paris, "Source mechanisms of volcanic tsunamis," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2053, p. 20140380, 2015.

[115] K. B. Haugen, F. Løvholt, and C. B. Harbitz, "Fundamental mechanisms for tsunami generation by submarine mass flows in idealised geometries," *Marine and Petroleum Geology*, vol. 22, no. 1-2, pp. 209–217, 2005.

[116] P. Watts and C. Waythomas, "Theoretical analysis of tsunami generation by pyroclastic flows," *Journal of Geophysical Research: Solid Earth*, vol. 108, no. B12, 2003.

[117] F. Maeno and F. Imamura, "Tsunami generation by a rapid entrance of pyroclastic flow into the sea during the 1883 krakatau eruption, indonesia," *Journal of Geophysical Research: Solid Earth*, vol. 116, no. B9, 2011.

[118] S. Carey, H. Sigurdsson, C. Mandeville, and S. Bronto, "Pyroclastic flows and surges over water: an example from the 1883 krakatau eruption," *Bulletin of Volcanology*, vol. 57, no. 7, pp. 493–511, 1996.

[119] M. Ewing and F. Press, "Further study of atmospheric pressure fluctuations recorded on seismographs," *Eos, Transactions American Geophysical Union*, vol. 34, no. 1, pp. 95–100, 1953.

[120] A. Rabinovich and S. Monserrat, "Meteorological tsunamis near the balearic and kuril islands: Descriptive and statistical analysis," *Natural Hazards*, vol. 13, no. 1, pp. 55–90, 1996.

[121] I. Yokoyama, "A scenario of the 1883 krakatau tsunami," *Journal of Volcanology and Geothermal Research*, vol. 34, no. 1-2, pp. 123–132, 1987.

[122] B. Le Méhauté, "Theory of explosion-generated water waves," *Advances in hydroscience*, vol. 7, pp. 1–79, 2013.

[123] G. Bellotti, M. D. Risio, and P. D. Girolamo, "Feasibility of tsunami early warning systems for small volcanic islands," *Natural Hazards and Earth System Sciences*, vol. 9, no. 6, pp. 1911–1919, 2009.

[124] M. Ulvrová, R. Paris, K. Kelfoun, and P. Nomikou, "Numerical simulations of tsunamis generated by underwater volcanic explosions at karymskoye lake (kamchatka, russia) and kolumbo volcano (aegean sea, greece)," *Natural Hazards and Earth System Sciences*, vol. 14, no. 2, pp. 401–412, 2014.

[125] T. Giachetti, R. Paris, K. Kelfoun, and B. Ontowirjo, "Tsunami hazard related to a flank collapse of anak krakatau volcano, sunda strait, indonesia," *Geological Society, London, Special Publications*, vol. 361, no. 1, pp. 79–90, 2012.

[126] S. N. Ward and S. Day, "Cumbre vieja volcano—potential collapse and tsunami at la palma, canary islands," *Geophysical Research Letters*, vol. 28, no. 17, pp. 3397–3400, 2001.

[127] F. Løvholt, G. Pedersen, and G. Gisler, "Oceanic propagation of a potential tsunami from the la palma island," *Journal of Geophysical Research: Oceans*, vol. 113, no. C9, 2008.

[128] S. Abadie, J. Harris, S. Grilli, and R. Fabre, "Numerical modeling of tsunami waves generated by the flank collapse of the cumbre vieja volcano (la palma, canary islands): Tsunami source and near field effects," *Journal of Geophysical Research: Oceans*, vol. 117, no. C5, 2012.

[129] J. E. Hunt, R. B. Wynn, D. G. Masson, P. J. Talling, and D. A. Teagle, "Sedimentological and geochemical evidence for multistage failure of volcanic island landslides: A case study from icod landslide on north tenerife, canary islands," *Geochemistry, Geophysics, Geosystems*, vol. 12, no. 12, 2011.

[130] S. F. Watt, J. Karstens, A. Micallef, C. Berndt, M. Urlaub, M. Ray, A. Desai, M. Sammartini, I. Klaucke, C. Böttner, *et al.*, "From catastrophic collapse to multi-phase deposition:

flow transformation, seafloor interaction and triggered eruption following a volcanic-island landslide," *Earth and Planetary Science Letters*, vol. 517, pp. 135–147, 2019.

[131] S. Watt, P. Talling, M. Vardy, V. Heller, V. Hühnerbach, M. Urlaub, S. Sarkar, D. Masson, T. Henstock, T. Minshull, *et al.*, "Combinations of volcanic-flank and seafloor-sediment failure offshore montserrat, and their implications for tsunami generation," *Earth and Planetary Science Letters*, vol. 319, pp. 228–240, 2012.

[132] Y. Liu, H. M. Fritz, and Z. Zhang, "Physical modeling of tsunamis generated by submarine volcanic eruptions.," *Geophysical Research Abstracts*, vol. 21, 2019.

[133] H. C. Kranzer and J. B. Keller, "Water waves produced by explosions," *Journal of applied physics*, vol. 30, no. 3, pp. 398–407, 1959.

[134] L. Law and A. Brebner, "On water waves generated by landslides," in *Proceedings of the Third Australasian Conference on Hydraulics and Fluid Mechanics*, pp. 155–159, Inst. of Eng. Sydney, NSW, Australia, 1968.

[135] W. G. Van Dorn, "Some characteristics of surface gravity waves in the sea produced by nuclear explosions," *Journal of Geophysical Research*, vol. 66, no. 11, pp. 3845–3862, 1961.

[136] B. Le Méhauté and S. Wang, *Water waves generated by underwater explosion*, vol. 10. World Scientific, 1996.

[137] P. Watts, *Water waves generated by underwater landslides*. California Institute of Technology, 1997.

[138] P. Watts, "Tsunami features of solid block underwater landslides," *Journal of waterway, port, coastal, and ocean engineering*, vol. 126, no. 3, pp. 144–152, 2000.

[139] H. M. Fritz, *Initial phase of landslide generated impulse waves*. PhD thesis, ETH Zurich, 2002.

[140] B. C. McFall and H. M. Fritz, "Runup of granular landslide-generated tsunamis on planar coasts and conical islands," *Journal of Geophysical Research: Oceans*, vol. 122, no. 8, pp. 6901–6922, 2017.

[141] G.-B. Kim, W. Cheng, R. C. Sunny, J. J. Horrillo, B. C. McFall, F. Mohammed, H. M. Fritz, J. Beget, and Z. Kowalik, "Three dimensional landslide generated tsunamis: Numerical and physical model comparisons," *Landslides*, vol. 17, no. 5, pp. 1145–1161, 2020.

[142] S. T. Grilli and P. Watts, "Tsunami generation by submarine mass failure. i: Modeling, experimental validation, and sensitivity analyses," *Journal of waterway, port, coastal, and ocean engineering*, vol. 131, no. 6, pp. 283–297, 2005.

[143] J. L. Hammack and H. Segur, "The korteweg-de vries equation and water waves. part 2. comparison with experiments," *Journal of Fluid mechanics*, vol. 65, no. 2, pp. 289–314, 1974.

[144] T. Jamin, L. Gordillo, G. Ruiz-Chavarría, M. Berhanu, and E. Falcon, "Experiments on generation of surface waves by an underwater moving bottom," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2178, p. 20150069, 2015.

APPENDIX A

FIRST APPENDIX

A.1 Comparison of the numerical modeling and experimental results

The maximum crest height measured by the four laboratory gauges are 11.75, 5.75, 2.67, and 1.97 mm respectively and maximum trough depths are 13.26, 6.25, 3.79, and 3 mm. So the wave amplitude seems to reduce in amplitude as the distance is more. From the errors it can be seen that numerical model over predicts the crest and crest and trough. The average absolute error percentage for gauges 0 to 3 in order are 31.2, 9.4, 2.1, and 4.2%. So the numerical results are matching well except for g_0.

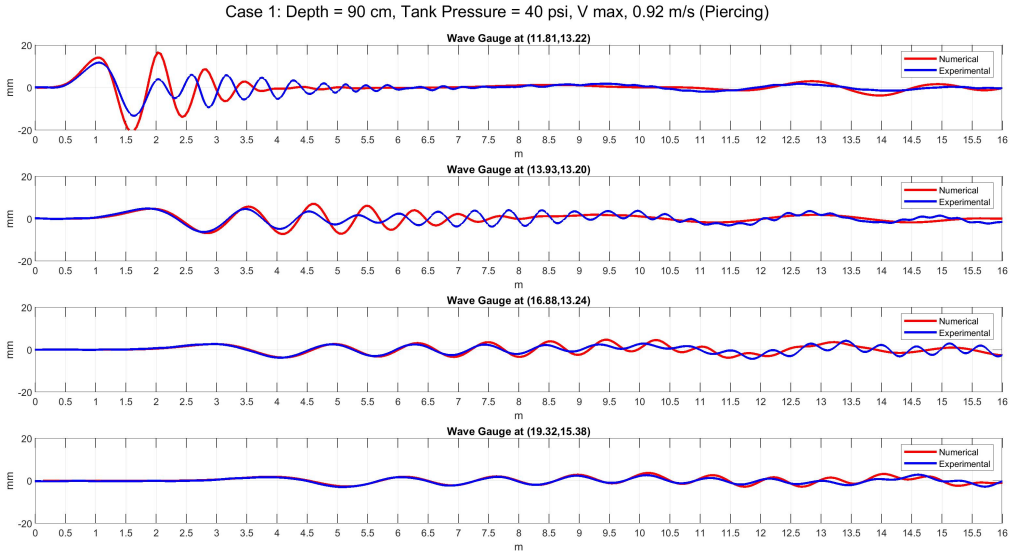


Figure A.1: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 1 with water depth of 90 cm and tank pressure of 40 psi.

Table A.1: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 1 with water depth of 90 cm and tank pressure of 40 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	14.14	11.75	9.6	-20.65	-13.26	-29.6	16.54	3.93	50.4	-13.80	-4.96	-35.3
1	4.76	4.81	-0.4	-6.79	-6.25	-4.9	5.75	4.61	10.4	-7.21	-4.80	-21.8
2	2.66	2.65	0.2	-3.67	-3.79	1.9	2.67	2.51	2.5	-3.25	-3.00	-3.9
3	1.96	1.90	1.2	-2.69	-3.00	5.8	1.97	2.07	-2.0	-2.18	-2.58	7.7

Case 2 results shown in Fig. A.2 is a surface piercing case with a water depth of 90 cm and medium-low tank pressure of 70 psi and have a 40% higher peak velocity than case 1. The table A.2 shows the experimental measurement and the numerical results of the first two crests and troughs from case 2. The average absolute error percentage for gauges 0 to 3 in order are 27, 4.5, 2.5, and 4.2%.

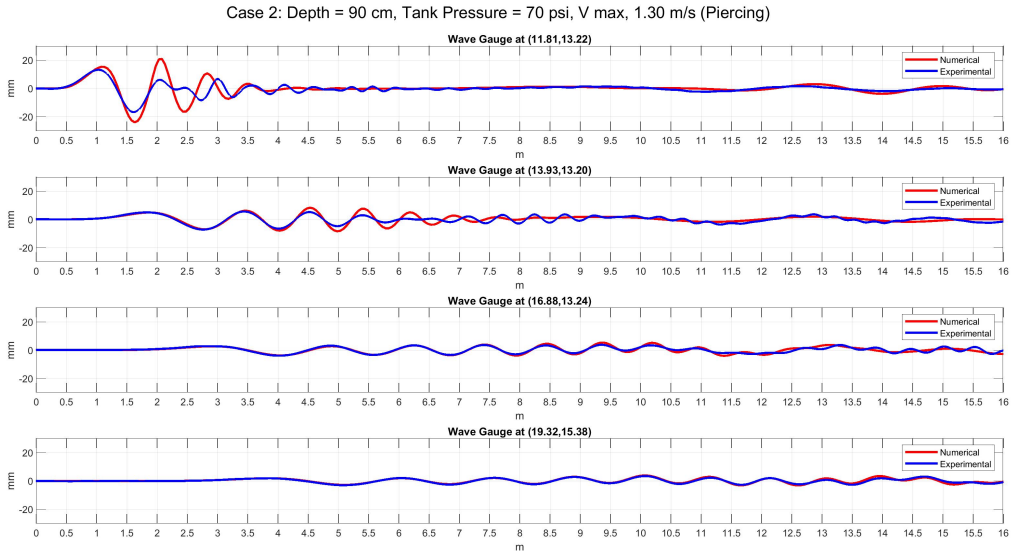


Figure A.2: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 2 with water depth of 90 cm and tank pressure of 70 psi.

Table A.2: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 2 with water depth of 90 cm and tank pressure of 70 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	15.33	13.18	7.2	-23.87	-16.73	-23.9	20.87	6.08	49.5	-16.43	-8.25	-27.4
1	4.88	4.93	-0.4	-7.03	-7.26	1.8	6.15	5.53	4.8	-7.99	-6.59	-10.9
2	2.70	2.76	-0.7	-3.75	-3.97	3.1	2.75	3.12	-5.0	-3.36	-3.46	1.3
3	1.96	1.90	1.2	-2.69	-3.00	5.8	1.97	2.07	-2.0	-2.18	-2.58	7.7

Case 3 results shown in Fig. A.3 is a surface piercing case with a water depth of 90 cm and medium-high tank pressure of 100 psi with a peak velocity 76% higher than case 1. The table A.3 shows the experimental measurement and the numerical results of the first two crests and troughs from case 3. The average absolute error percentage for gauges 0 to 3 in order are 26.3, 3.9, 5.6, and 8.8%.

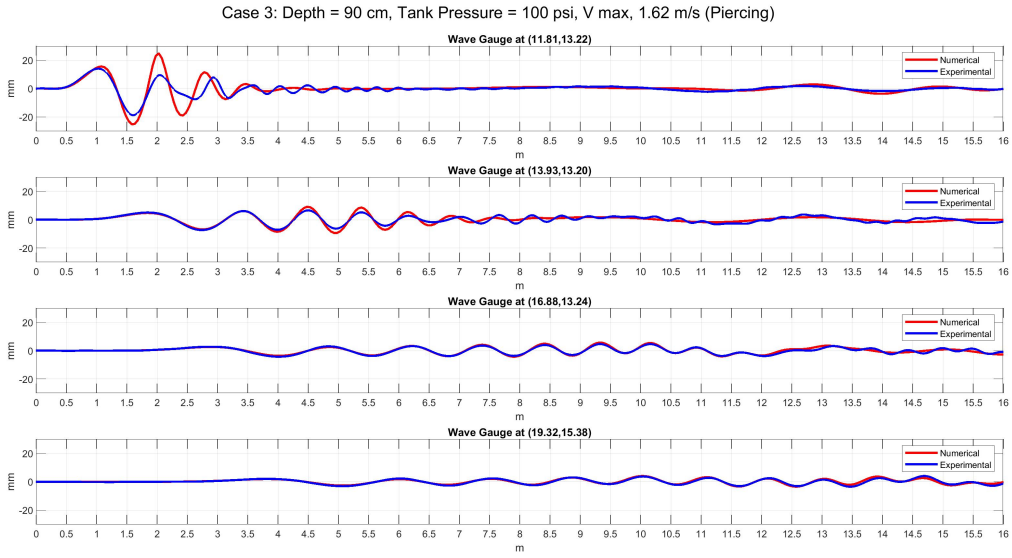


Figure A.3: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 3 with water depth of 90 cm and tank pressure of 100 psi.

Table A.3: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 3 with water depth of 90 cm and tank pressure of 100 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	15.66	14.10	4.8	-25.06	-18.76	-19.2	24.79	9.53	46.5	-18.93	-7.51	-34.8
1	4.81	5.10	-2.0	-6.81	-7.32	3.6	6.18	6.14	0.3	-8.48	-7.12	-9.7
2	2.67	2.77	-1.4	-3.61	-4.27	8.9	2.58	3.06	-6.4	-3.31	-3.74	5.7
3	1.94	2.28	-6.3	-2.60	-3.10	9.0	1.83	2.31	-8.7	-2.09	-2.70	11.2

Case 4 have the highest tank pressure and the largest peak velocity of 2.05 m/s and its results are shown in Fig. A.4. The table A.4 shows the experimental measurement and the numerical results of the first two crests and troughs from case 4. The average absolute error percentage for gauges 0 to 3 in order are 26, 6.2, 8.8, and 10.2%.

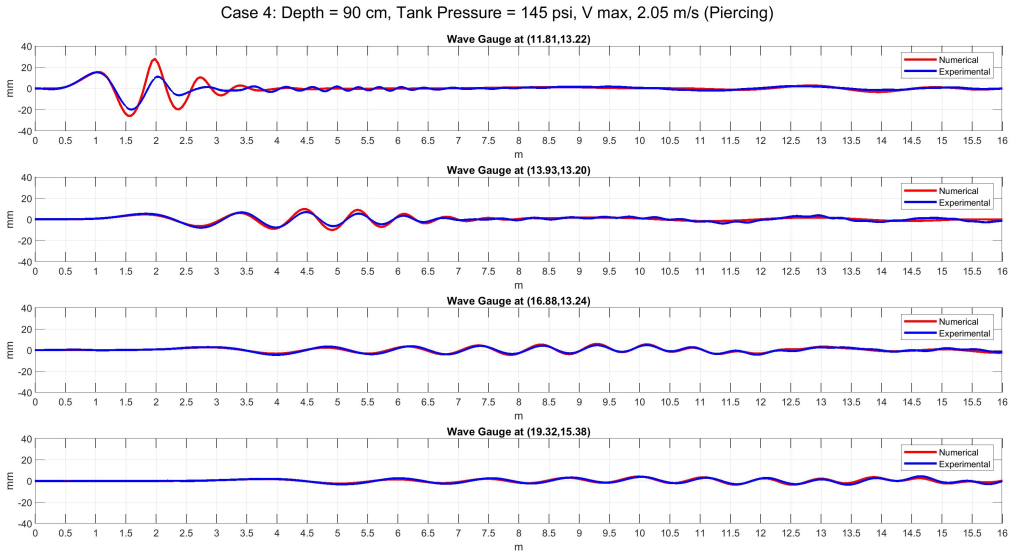


Figure A.4: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 4 with water depth of 90 cm and tank pressure of 145 psi.

Table A.4: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 4 with water depth of 90 cm and tank pressure of 145 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	15.52	15.17	1.0	-25.89	-19.87	-17.2	27.85	11.05	47.9	-19.62	-6.39	-37.7
1	4.66	5.26	-4.1	-6.42	-7.85	9.8	6.12	6.54	-2.8	-8.93	-7.73	-8.2
2	2.58	2.78	-2.4	-3.34	-4.51	14.2	2.35	3.29	-11.5	-3.27	-3.84	7.0
3	1.88	1.97	-1.5	-2.41	-3.11	12.4	1.62	2.53	-16.0	-1.98	-2.60	10.9

Case 5 produces the smallest waves as it have the lowest pressure of 40 psi and higher water depth of 1.2 m and its results are shown in Fig. A.5. It is a non- piecing case with the lowest peak velocity of 0.68 m/s. The table A.5 shows the experimental measurement and the numerical results of the first two crests and troughs from case 5. The average absolute error percentage for gauges 0 to 3 in order are 9.8, 9.3, 8.4, and 7.8%.

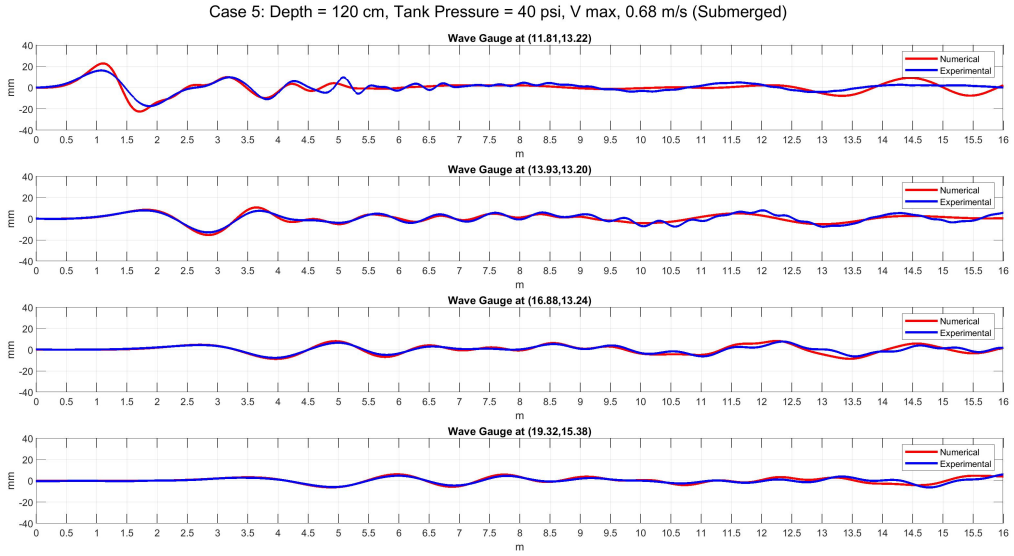


Figure A.5: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 5 with water depth of 120 cm and tank pressure of 40 psi.

Table A.5: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 5 with water depth of 120 cm and tank pressure of 40 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	22.79	16.13	19.9	-22.51	-17.39	-15.3	9.92	9.67	0.7	-9.74	-10.87	3.4
1	8.50	7.85	3.1	-15.38	-12.87	-12.1	10.72	7.45	15.8	-5.08	-3.81	-6.1
2	4.68	4.34	2.4	-8.70	-7.72	-6.9	8.07	6.53	10.8	-6.85	-4.92	-13.5
3	3.35	2.84	4.8	-6.17	-5.86	-3.0	6.13	4.78	12.7	-5.59	-4.45	-10.7

Case 6 and Case 7 results shown in Fig. A.6 and A.7. Both are non-piercing cases with a water depth of 120 cm and tank pressures 70 and 100 psi, respectively. The table A.2 shows the experimental measurement and the numerical results of the first two crests and troughs from case 6. For the case 6, the average absolute error percentage for gauges 0 to 3 in order are 13.1, 10.7, 9.8, and 7.8%. For the case 7, the average absolute error percentage for gauges 0 to 3 in order are 14, 10.7, 9.4, and 8.2%.

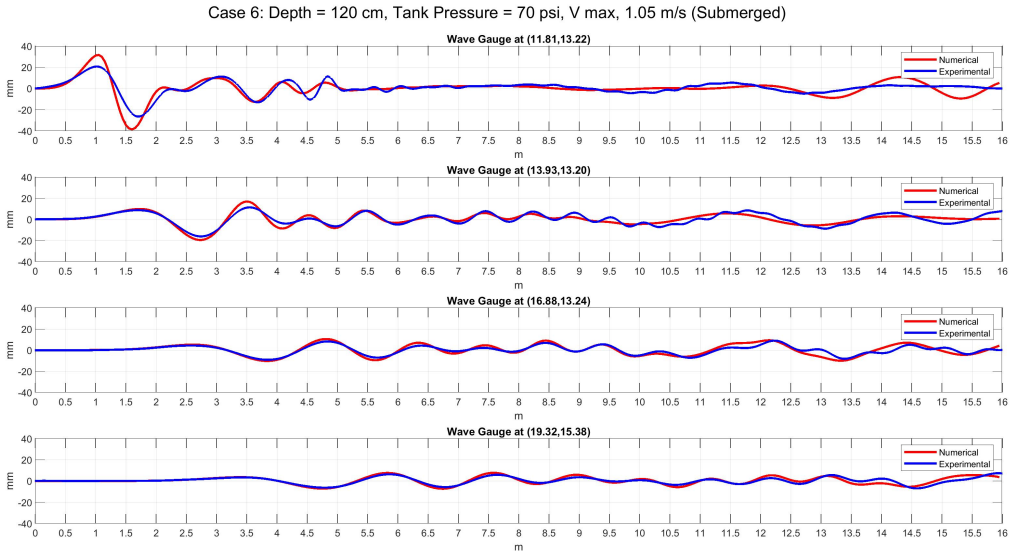


Figure A.6: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 6 with water depth of 120 cm and tank pressure of 70 psi.

Table A.6: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 6 with water depth of 120 cm and tank pressure of 70 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	31.61	20.63	23.3	-38.52	-26.48	-25.6	9.81	11.15	-2.9	-12.67	-13.06	0.8
1	9.84	8.67	4.3	-19.49	-16.16	-12.1	16.94	11.32	20.4	-8.22	-6.56	-6.0
2	5.20	4.29	5.3	-10.29	-9.16	-6.5	10.51	8.14	13.6	-9.52	-7.13	-13.8
3	3.66	3.36	2.3	-7.13	-6.31	-6.5	7.58	6.29	10.3	-7.29	-5.79	-11.9

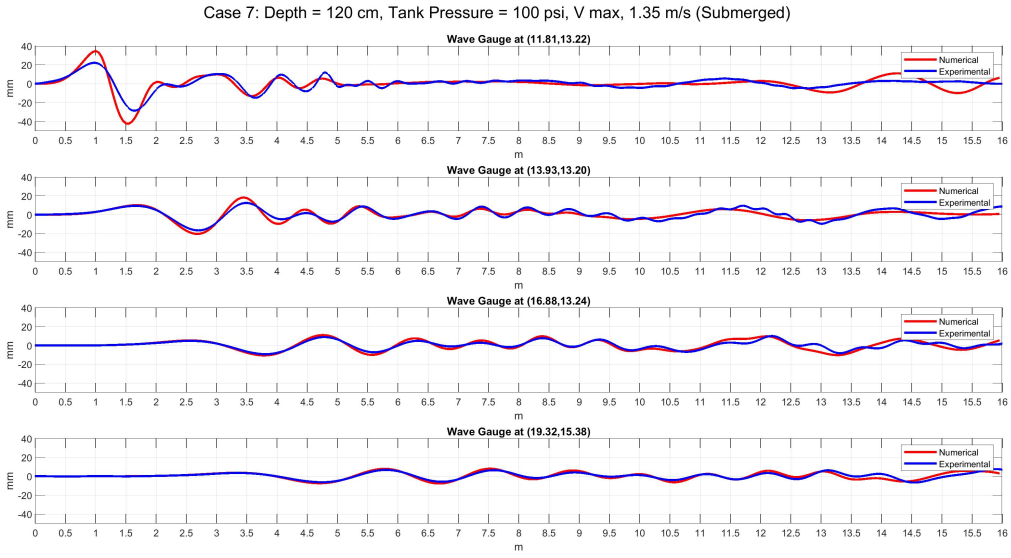


Figure A.7: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 7 with water depth of 120 cm and tank pressure of 100 psi.

Table A.7: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 7 with water depth of 120 cm and tank pressure of 100 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	034.74	22.15	24.8	-42.26	-28.54	-27.1	9.99	10.34	-0.7	-13.08	-14.88	3.6
1	10.18	9.31	3.0	-20.45	-16.76	-12.7	18.31	12.43	20.2	-9.55	-7.49	-7.1
2	5.32	4.79	3.0	-10.65	-9.34	-7.2	11.03	8.83	12.1	-10.07	-7.31	-15.2
7	3.73	3.49	1.8	-7.35	-6.32	-7.9	7.89	6.70	9.1	-7.62	-5.82	-13.8

The final one, case 8 have highest tank pressure same as case 4 of 145 psi, but lesser peak velocity of 1.77 m/s because of the larger water depth of 120 cm. The results are shown in Fig. A.8. The table A.2 shows the experimental measurement and the numerical results of the first two crests and troughs from case 8. The average absolute error percentage for gauges 0 to 3 in order are 14.8, 10.3, 10.1, and 7.1%.

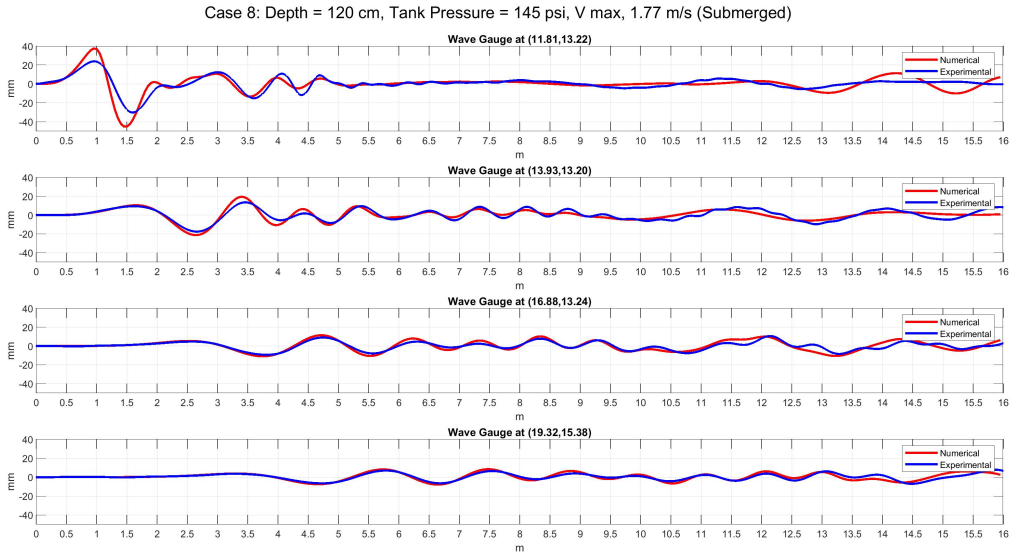


Figure A.8: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment. Results are for the experiment no. 8 with water depth of 120 cm and tank pressure of 145 psi.

Table A.8: Comparison of the numerical results with the measurements from four laboratory wave gauges for the submarine volcanic tsunami experiment according to the crest and trough heights. Results are for the experiment no. 8 with water depth of 120 cm and tank pressure of 145 psi. C_{m1} and C_{m2} are the first and second crest heights in from the numerical model, while T_{m1} and T_{m2}, the troughs. 'm' denotes the numerical results, and 'e', the experimental measurements.'E' stands for the error with respect to maximum experimental wave height. All measurements are in millimeter and errors in percentage.

No.	C_{m1}	C_{e1}	C_{1E}	T_{m1}	T_{e1}	T_{1E}	C_{m2}	C_{e2}	C_{2E}	T_{m2}	T_{e2}	T_{2E}
0	37.56	23.86	25.3	-45.10	-30.24	-27.5	10.69	12.32	-3.0	-13.43	-15.19	3.2
1	10.45	9.32	3.6	-21.30	-17.62	-11.8	19.48	13.45	19.4	-10.44	-8.48	-6.3
2	5.41	4.67	4.0	-10.94	-9.39	-8.4	11.44	8.95	13.6	-10.53	-7.92	-14.3
3	3.78	3.47	2.4	-7.51	-6.61	-6.6	8.16	6.89	9.4	-7.93	-6.58	-10.0