Numerical Study of Multiphase Flows using an Adaptive Filament Moment-of-Fluid Method

Philippe Hergibo

School of Engineering

Cardiff University

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2024

I would like to dedicate this thesis to my parents and siblings, all doctors.

Declaration

I hereby declare that the content of the work submitted is original. Appropriate credit is given where specific reference is made to the work of others.

This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Philippe Hergibo

December 2024

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Zhihua Xie and Prof. Timothy Phillips, for their invaluable guidance and support throughout the entire journey of my Ph.D. Their expertise and encouragement have been instrumental in shaping this research.

I am also thankful for the insightful feedback and suggestions provided by Prof. Qiuhua Liang during our collaboration. His constructive criticism and thoughtful, kind comments greatly contributed to the refinement of my work.

Although this research primarily represents an independent endeavor, I am grateful for the occasional discussions and assistance from colleagues and friends at university or during conferences. Their diverse perspectives and shared experiences enriched my understanding and enhanced the quality of this thesis.

I extend a special acknowledgment to my dear friend, Will, for sharing this academic journey with me. Your camaraderie, intellectual discussions, and shared experiences have added a meaningful dimension to the process of completing this Ph.D. thesis.

Special thanks go to my family and partner for their unwavering support, understanding, and encouragement during the ups and downs of this academic pursuit. Their love and patience have been my pillars of strength.

Lastly, I want to acknowledge the financial support provided by EPSRC. This support played a crucial role in enabling the successful completion of this research.

In conclusion, while this journey has largely been solitary, the collective contributions of those mentioned above have made it a truly enriching and fulfilling experience

Abstract

Multiphase flows are present in many industrial and engineering applications as well as in some physical phenomena. Simulations of multiphase flows serve as a crucial tool in understanding the complex fluid dynamics present in various natural and engineering phenomena.

The main objective of this study is to develop and advance numerical methods for simulating multiphase flows with a novel approach that can reconstruct under-resolved structures. The second objective is to evaluate its performance and extend the complexity using adaptivity.

The research introduces and refines the moment-of-fluid (MOF) method, particularly in addressing challenges related to interface capturing and fine-scale structure resolution. The first part of the study presents a symmetric approach for reconstructing thin structures, or filaments, using a new efficient approach. The MOF method is extended to an adaptive mesh refinement (AMR) framework, employing a quadtree structure to efficiently resolve thinner structures without a substantial increase in computational cost.

The effectiveness of the MOF method and its adpative extension are demonstrated through benchmark problems, showcasing computational efficiency and accuracy. Furthermore, the thesis introduces a novel adaptive dual grid technique, enabling simultaneous adaptive resolution of fine interfacial details and solution of the Navier-Stokes equations on a coarse grid. Validation against benchmark cases demonstrates the capability of the method in accurately capturing complex interface dynamics while maintaining computational efficiency.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

A	Displacement amplitude
a	Domain dimension units
a_P	Coefficients in algebraic equations
A_f	Face area vector
At	Atwood ratio
b_P	Source term in algebraic equations
B_D	Fixed value boundary
B_N	Fixed gradient boundary
V_{CV}	Control volume
d	Domain dimension units
D_f	Diffusive coefficients at face values
div_max	Maximum refinement level
$\mathbf{dS}$	Outward pointing surface vector
dx	Horizontal length scale of cell
dy	Vertical length scale of cell
E	Total centroid defect
E_{L_1}	L_1 error
E_r	Relative error
E_{sym}	Symmetric difference error

$\mathbf{f}$	External forces
F	Volume fraction
f	Objective function
F_{act}	Reconstructed volume fraction
F_{ADG}	Subcell volume fraction on the secondary grid
$\mathbf{F_C}$	Convective term
$\mathbf{F_D}$	Diffusive term
F_f	Convective coefficients at face values
F_{NS}	Volume fraction for the Navier-Stokes solver
f'	First derivative of the objective function for a convex cell
F_{ref}	Reference volume fraction
f_{sym}	Objective function for symmetric reconstruction
$f(t)$	Amplitude function
$\mathbf{g}$	User-specified boundary function
g	Gravitational acceleration
H	Height of the domain
h	Still water level
H_c	Height of the water column
h_f	Height of the rotating filament
h_Z	Maximum height of the Zalesak disc
i	Cell index in the horizontal direction
is	Subcell index in the horizontal direction
j	Cell index in the vertical direction
js	Subcell index in the vertical direction
k	Wave number
L	Length of the domain

$L0$	Level 0
$L1$	Level 1
$L2$	Level 2
lev	Level of refinement
L_∞	Maximum error
m	Number of material within a cell
M_s	Maximum subcell index
$\mathbf{n}$	Outward pointing normal
N	Number of mixed cells on a uniform mesh
n_{it}	Number of iterations
n^{max}	Maximum number of grid cells
N_{mix}	Number of mixed cells at a certain level
n^x	Horizontal components of the gradient
n^y	Vertical components of the gradient
OC	Pseudo-order of convergence
P	Wave period
p	Pressure
Q_S	Source term
r	Radius of circles/discs
Re	Reynolds number
T	Nondimensional time
t	Time
t_ε	Transition period
T_{max}	Nondimensional time at maximum deformation
$\mathbf{u}$	Fluid velocity
U	Horizontal velocity at cell face

u	Horizontal fluid velocity
$\mathbf{u}_{exc}$	Excitation velocity
u_n	Horizontal node velocity
uv	Multi-directional node velocity
v_n	Vertical node velocity
V	Vertical velocity at cell face
v	Vertical fluid velocity
w_f	Width of the rotating filament
w_Z	Slotted rectangle width for the Zalesak disc
$\mathbf{x}$	Spatial coordinates
X	Horizontal surface single-valued function
x	Streamwise direction
x_A	Location of probe A
x_{act}	Reconstructed centroid
x'_{act}	First derivative of x_{act}
x_B	Location of probe B
x_C	Location of probe C
x_c	Centroid
x_{ref}	Reference centroid
x^*	Reference centroid
Y	Vertical surface single-valued function
y	Vertical direction
z	Spanwise direction

Greek Symbols

α_p	Under-relaxation factor
β	Angle of the orientation of the interface

Δ	Spatial displacement
δ	Global redistribution variable
δ_{low_lev}	Lower level redistribution variable
Δm	Mass variation
Δt	Time step
Δx	Horizontal spatial displacement
Δy	Vertical spatial displacement
$\partial\Lambda$	Domain boundaries
∂V	Control volume boundary
Γ	Interface segment
Γ^{ℓ}	Reconstructed interface
Γ^{star}	Reference interface
Λ	Computational domain
μ	Fluid dynamic viscosity
μ_1	Material 1
μ_2	Material 2
μ_{rem}	Remaining material
Ω	Arbitrary convex cell
Φ	Full body rotation angle
ϕ	Flow quantity
ϕ_a	Angle of the orientation of the interface
Φ_{lev}	Refinement subset
ρ	Fluid density
ε	Cross-product tolerance
φ	Arbitrary boundary value
ω	Excitation frequency

ω_{cp}	Convex polygon
ω_0	Natural frequency
ω^{ℓ}	Reconstructed subset
$\omega^{\star}$	Reference subset

Superscripts

0	Value at previous iteration
a	Value characterising fluid A
act	Reconstructed state
b	Value characterising fluid B
$\prime$	Corrected value in pressure-velocity coupling
$n+1$	Value at new time step
ref	Reference state
$*$	Guessed value in pressure-velocity coupling
T	Matrix transpose

Subscripts

A	Heavy fluid index
act	Reconstructed state
B	Light fluid index
E	Value at the east of point P
e	Value for the east face of control volume P
f	Centre of the cell face
$final$	Final state
$initial$	Initial state
N	Value at the north of point P
n	Value for the north face of control volume P
nb	Neighbouring control volumes

P	Value at the point considered
ref	Reference state
S	Value at the south of point P
s	Value for the south face of control volume P
W	Value at the west of point P
w	Value for the west face of control volume P

Other Symbols

+	Reference centroid
□	Vertices advected with RK2
△	Vertices interpolated
o	Reconstructed centroid

Acronyms / Abbreviations

ADG	Adaptive Dual Grid
ALE	Arbitrary Lagrangian-Eulerian
AMR	Adaptive Mesh Refinement
BM	Base Mesh
CCALE	Cell-Centered Arbitrary Lagrangian–Eulerian
CC	Cell count
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CICSAM	Compressive Interface Capturing Scheme for Arbitrary Meshes
CLSMOF	Coupled Level Set Moment of Fluid
CLSVOF	Coupled Level Set Volume of Fluid
EPSRC	Engineering and Physical Sciences Research Council
FCT	Flux Corrected Transport
FIL	Filament

FP	Filament procedure count
LSM	Level Set Method
MOF	Moment-of-Fluid
NS	Navier-Stokes
PIMPLE	PISO-SIMPLE
PISO	Pressure-Implicit with Splitting of Operators
PLIC	Piecewise Linear Interface Calculation
PPIC	Piecewise Parabolic Interface Calculation
Re-ALE	Reconnection-based Arbitrary Lagrangian Eulerian
RK2	2^{nd}-order Runge-Kutta scheme
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
SLIC	Simple Line Interface Calculation
SPH	Smoothed Particle Hydrodynamics
STD	Standard
THINC	Tangent of Hyperbola for INterface Capturing
TVD	Total Variation Diminishing
VOF	Volume-of-Fluid

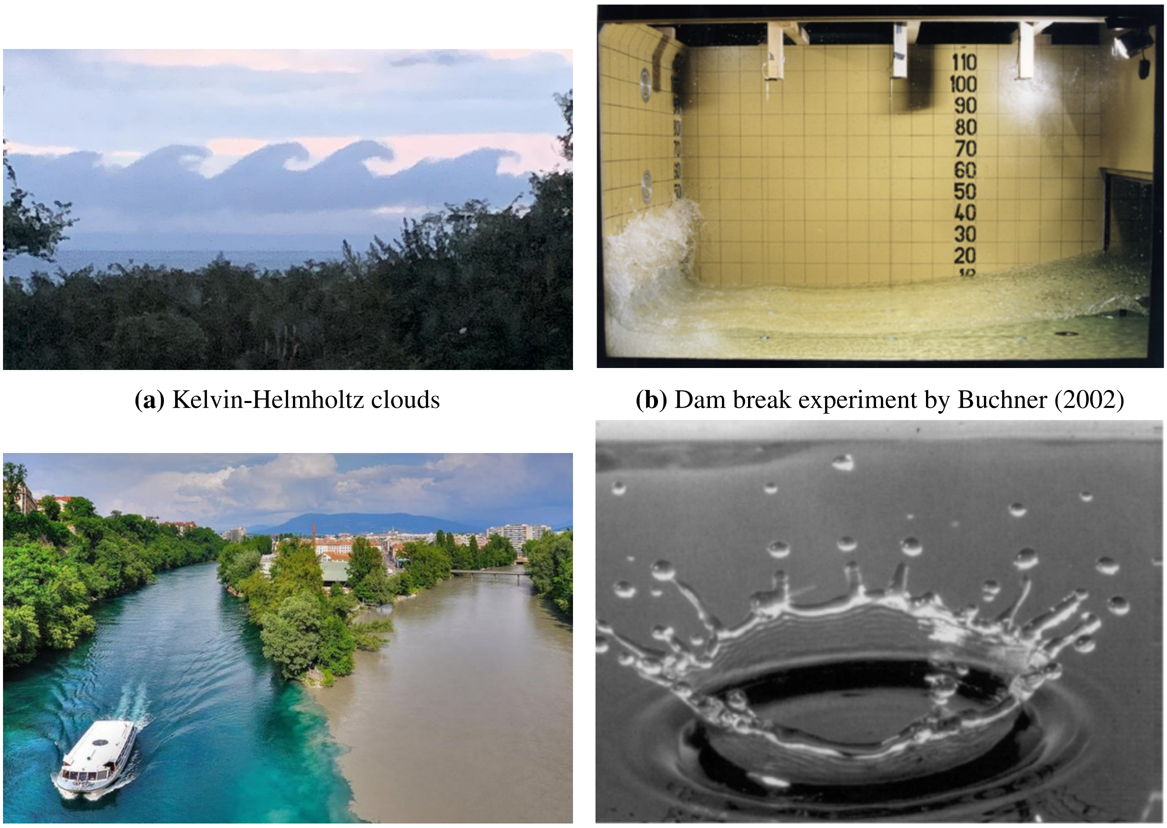
Chapter 1

Introduction

1.1 Background and context of the research

Multiphase flows are characterised by the simultaneous presence and coexistence of two or more discernible phases, encompassing liquids, gases, and solids, or by the presence of multiple fluids, also called materials. They play a pivotal role in diverse fields of study, ranging from engineering applications, natural phenomena to physical processes. The complex interface between these phases has profound implications for the design and operation of numerous engineering systems, such as oil transportation or gas extraction, inkjet printing or coastal protection. Applications include electronics cooling and chemical reactions in bubble columns and fluidised beds. Some applications are relevant to developing a deep understanding of natural occurrences like river sediment transport, wave breaking, ocean currents, volcanic eruptions, cloud formation, droplet behaviour or bubble dynamics. Exploring and understanding these phenomena is crucial for the development of novel methods aimed at optimising industrial processes, enhancing energy efficiency and safety, and safeguarding environmental sustainability.

Fig. 1.1 illustrates various instances of multiphase flow. The first example (a) depicted is the Kelvin-Helmholtz instability, illustrated by cloud movements as a result of wind shearing clouds in opposite directions. The second example (b) shows a dam break experiment which underscores the significance of complex dynamics of the water surface and its impact, essential for predicting flood propagation, inundation patterns and forces on walls or structures. The third example (c) illustrates a river confluence, where two streams converge, which underscores the challenges in predicting the interface behaviour between distinct water bodies, crucial for understanding sediment transport dynamics and ecological implications. The last example (d) highlights the interface dynamics phenomenon of droplet crown formation. This phenomenon is observed when a liquid droplet impacts on a thin liquid surface. This causes the surrounding liquid to disperse radially outward, exhibiting intricate patterns and morphological changes as the liquid redistributes and retracts due to surface tension forces, typically forming a very thin bridge of water.



(a) Kelvin-Helmholtz clouds

(b) Dam break experiment by Buchner (2002)

(c) Interface at the confluence of two rivers

(d) Droplet impact by Cossali et al. (1997)

Fig. 1.1 Examples of multiphase flow applications seen in nature or in the laboratory.

Experimental studies are indispensable in scientific research (Buchner, 2002; Cossali et al., 1997). They are frequently subject to constraints that may prevent their effectiveness when compared to numerical approaches. One prominent challenge is the inability to control all variables in a real-world setting. This may potentially result in factors that obscure the correlation between variables of interest. In addition, experimental studies are typically constrained by practical considerations such as cost, time, and simplicity which may restrict the scope and scale of investigations.

On the contrary, numerical work offers flexibility, repeatability and scalability. The advantage of control over model parameters and boundary conditions is significant. Furthermore, numerical simulations can study phenomena that may be impractical or impossible to replicate in a laboratory setting. Despite these limitations, experimental studies remain indispensable for validating numerical models and providing empirical grounding for theoretical frameworks.

Within the realm of physics, the intricacies of multiphase flows not only test our understanding of fluid dynamics but also pose formidable numerical challenges, pushing the boundaries of scientific research. In numerical multiphase flows, representing the interface with accuracy is essential for several critical reasons. First and foremost, the accurate depiction of interfaces is integral to providing reliable predictions of complex topological structures. Yet, this accuracy is particularly challenging due to the technical complexities associated with representing the interface between two or more materials numerically. Without a precise representation, the simulation may fail to capture the intricacies of multiphase flow dynamics, leading to inaccurate predictions and preventing our ability to understand and optimise various engineering processes.

Moreover, the evolving interfaces in multiphase flows necessitate considerable computational resources for accurate prediction. A precise representation becomes crucial in managing these computational demands effectively. It ensures that mass, momentum, and energy transfer between phases are robustly and accurately simulated. In addition, increasing the level of detail numerically also requires substantial resources, which may be currently insufficient.

Tackling problems with different characteristic sizes may present numerical complexities, particularly concerning the dynamic nature of the interfaces involved. These challenges arise from the necessity to capture interfaces that may exist on scales smaller than those of the overall flow field, often referred as "subgrid". As a result, using fixed grids necessitates a fine uniform grid throughout the entire computational domain to accurately resolve intricate interfacial details, leading to a substantial increase in computational effort. In addition, increasing the global resolution to directly resolve these interfaces using conventional numerical methods can be impractical and highly expensive in large-scale simulations.

Therefore, developing advanced numerical methods and computational techniques capable of capturing the intricate dynamics of multiphase interfaces is imperative for extending the boundaries of multiphase flow simulations and addressing real-world engineering challenges effectively. The use of subgrid modelling techniques to accommodate unresolved structures is crucial to depict accurately their dynamics and interactions within the flow. In essence, a precise characterisation of the interface is fundamental for reliable and robust predictions of multiphase flows.

1.2 Previous studies

1.2.1 Interface techniques

Over the years, several techniques have emerged to effectively represent interfaces in computational simulations. These fall into two broad categories: interface tracking and interface capturing methods.

On one hand, interface tracking techniques are designed to follow and monitor a defined set of points that represent the interface. The tracking is often aimed to track points along the velocity field associated with the fluid flow. By continuously updating the positions of points defining the interface, interface tracking methods provide detailed characteristics of the interface over time. These techniques are particularly useful in situations where the interface is sharp and well-defined (Leung and Zhao, 2009; Li, 2013; Sato and Ničeno, 2013; Tryggvason et al., 2001).

On the other hand, interface capturing techniques are aimed at approximating the interface without explicitly tracking individual points. Instead, these methods try to characterise the interface to distinguish between different phases or materials (Scardovelli and Zaleski, 1999). While interface capturing methods are generally less computationally intensive compared to interface tracking, they may struggle to accurately represent sharp interfaces or capture intricate interface dynamics.

Both approaches have their advantages and limitations. Some numerical techniques might be easier to implement, some might have better mass conservation properties, and some might resolve complex interfaces in a superior way. Overall, all these techniques have been adopted widely in the literature for interface calculation.

Methods for predicting the behaviour of fluids with complex interfaces include the marker-and-cell method (Harlow and Welch, 1965), front tracking method (Unverdi and Tryggvason, 1992), diffuse interface method (Anderson et al., 1998), level-set method (Osher and Sethian, 1988; Sethian and Smereka, 2003), volume-of-fluid (VOF) method (Hirt and Nichols, 1981; Scardovelli and Zaleski, 1999), and some meshless methods such as smoothed particle hydrodynamics (SPH) (Monaghan, 1992). This section will discuss some of them in more detail.

1.2.1.1 Interface tracking

Front tracking method The front tracking technique represents a traditional method that combines interface tracking features on a fixed Eulerian grid (Tryggvason et al., 2001). In a structured grid approach, the interface, or front, is explicitly tracked by a network of connected marker points. The motion of the front within the velocity field is achieved by tracking the advection of these marker points. Accuracy and stability are key advantages offered by this explicit approach. Unlike approaches reliant on the advection of a marker function, front tracking minimises numerical inaccuracies associated with scalar function advection and surface tension representation. However, its complex implementation and computational efficiency present significant challenges. In addition, simulating the breakup and coalescence of material may be difficult. Fig. 1.2 depicts a visual schematic of the front tracking approach, inspired by Tryggvason et al. (2001).

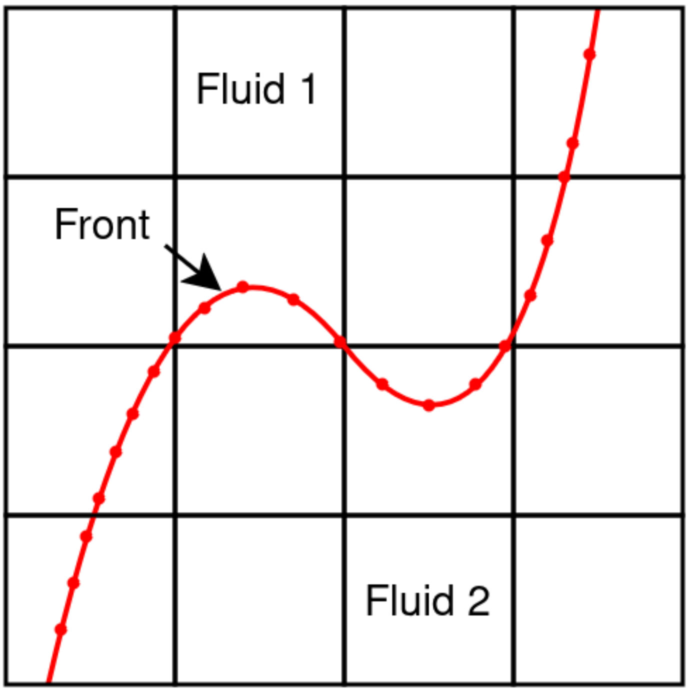


Fig. 1.2 Schematic of the front tracking technique. Marker points are linked to each other forming the front/interface.

Smoothed Particle Hydrodynamics Smoothed Particle Hydrodynamics (SPH) was originally designed to simulate compressible flows in astrophysics rather than to tackle incompressible fluid problems. The SPH method is a mesh-free Lagrangian method that has gained popularity for simulating complex fluid dynamics in multiphase flows. In SPH, the fluid domain is discretised into particles, and physical quantities are represented by smoothing kernels. This approach excels in handling large deformations and complex interfaces. Specifically in multiphase flows, SPH has proven effective in capturing dynamic interfaces, such as the breakup of droplets or the merging of liquid streams (Shao and Lo, 2003). However, some limitations and difficulties arise due to the computational cost associated with this method. Fig. 1.3 highlights an example of an SPH simulation outside the context of multiphase flows, shedding light on the dynamic processes underlying Moon formation. Without relying on a fixed grid, SPH enables the interactions of particles and gravitational forces to be simulated and offers insights into the origins of the Moon.

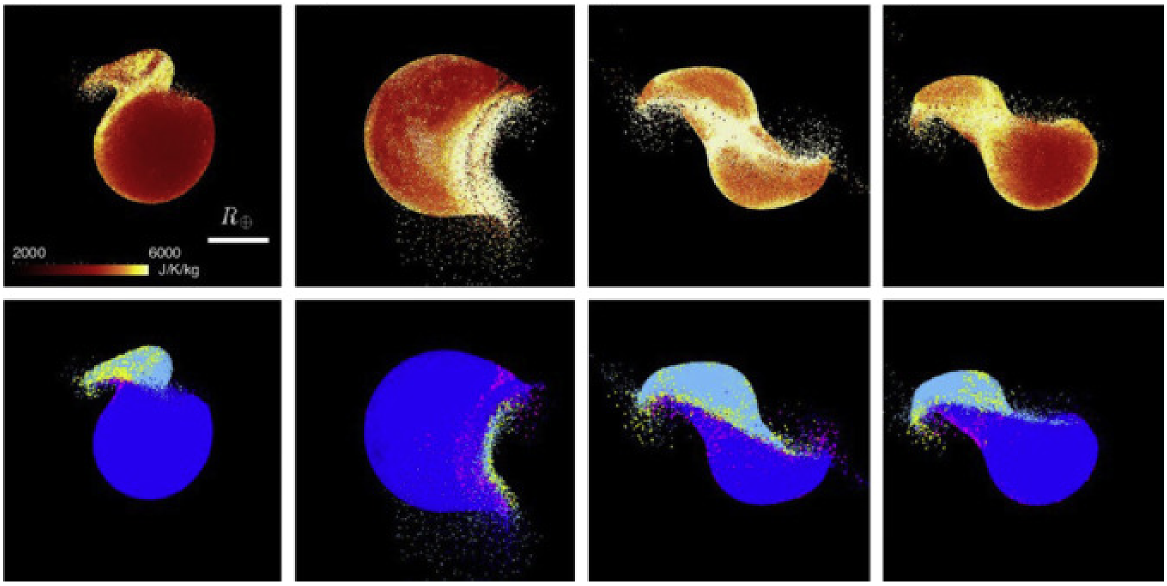


Fig. 1.3 Formation of the Moon simulated using SPH (Nakajima and Stevenson, 2014). Four scenarios of collision are studied. Top row shows the energy of particles, bottom rows shows the origins of particles.

Particle tracking method Similar to the SPH method, particle tracking methods involve the simulation of individual particles representing fluid elements, enabling the tracking of their trajectories over time. This Lagrangian approach proves advantageous for modelling particle transport and intricate interactions within a flow (Koshizuka and Oka, 1996). This approach, also called MPS method, is particularly effective in capturing phenomena like particle dispersion, sedimentation, and the dynamic movement of individual fluid elements. The MPS method employs kernel functions to calculate particle interactions. Particle detection, resolution and computational cost are some of the drawbacks of the method. Indeed, to ensure robust calculations, neighbouring particles need to be identified and detected, which can be a particularly expensive computational process.

Moving mesh method Moving mesh techniques dynamically adjust the computational grid to accommodate changes in the flow field and have proved to be particularly useful in scenarios with evolving interfaces. The integration of particle tracking with a moving mesh strategy provides a powerful approach for simulating multiphase flows with both accuracy and efficiency (Zwart et al., 1999). Moving mesh techniques contribute to a comprehensive understanding of multiphase flow phenomena across diverse applications. Whilst this method enables a sharp interface to be maintained, it is not suited for predicting the breakup and coalescence of materials.

1.2.1.2 Interface capturing

The trade-off between sharp and diffuse interface representations directly impacts the quality of the interface and its numerical stability. Sharp interfaces enable precise calculation of the normal of the interface. Conversely, diffuse methods enhance stability but can degrade accuracy in the estimation of the normal, leading to errors in the dynamics of the interface. Balancing these factors is critical for robust and accurate multiphase flow simulations.

Diffuse interface method The diffuse interface method emerges from the idea of a diffused interface between two fluids (Anderson et al., 1998). Instead of a distinct sharp separation between two phases, this method describes a thin interface layer where fluid mixing occurs gradually and smoothly, similar to the concept of shock waves. To simulate the behaviour within this region, a function called the phase field is initialised. This phase distribution represents the total free energy within the diffuse region. However, the method is prone to numerical issues including the choice of thickness parameter and acceptable computational costs.

Level set method Within interface capturing methods, the level set method stands as a prominent technique. The fundamental component of this method is a signed-distance function that is used to differentiate several phases/materials and which therefore defines the interface between them (Osher and Sethian, 1988). The different phases are determined based on the positive and negative values of the level-set function. By defining the interface as the zero level set of this function, the method implicitly tracks the evolution and behaviour of the interface over time. Traditionally, numerical errors are introduced in the calculation of the smooth function which may lead to numerical instability and loss of accuracy. Some improvements have been developed in order to rectify these by using a reinitialisation equation (Sussman et al., 1994). Therefore, the enhanced accuracy and inherent robustness make it valuable in simulating many complex multiphase flows. Whether dealing with phenomena such as fluid mixing, phase separation, or surface tension effects, the method has enjoyed great success in capturing intricate interface dynamics. Yet, despite its strengths, in most engineering problems, the lack of mass conservation of the traditional technique poses a significant challenge and makes it undesirable for accurate predictions. Fig. 1.4 shows a visual representation of the traditional level set method and how the interface is represented.

To address this limitation, notable improvements have been made on this issue by developing the conservative level set method (Chiodi and Desjardins, 2017; Olsson and Kreiss,

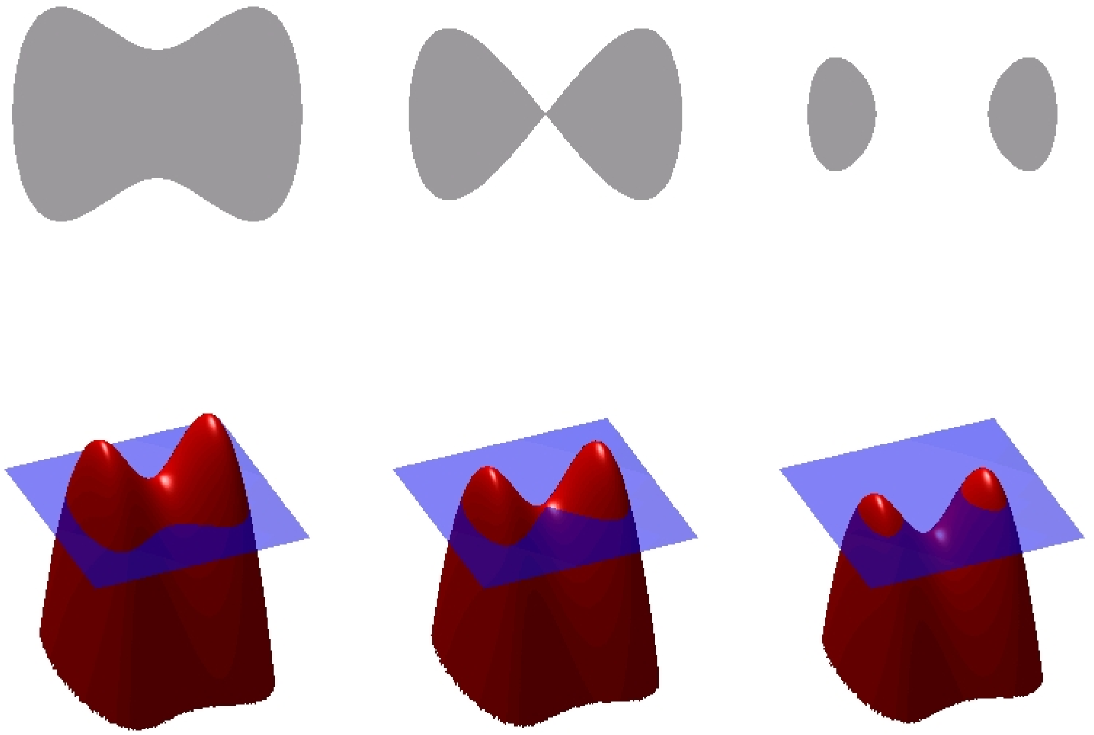


Fig. 1.4 Illustration of the level set method in 2D. The smooth function is in red and the zero level set representing the interface is highlighted with the blue plane. The projection of the boundary from above forms the grey shape (Alexandrov and Santosa, 2005).

2005). By incorporating conservation principles into the formulation, the conservative level set method strives to rectify the mass conservation issue, thereby enhancing its suitability for a wider range of multiphase flow problems. Recent advances have extended the applicability of the conservative level set method to encompass non-Newtonian multiphase flows, enabling the solution of complex rheological problems involving viscoelastic fluids (Doherty et al., 2023).

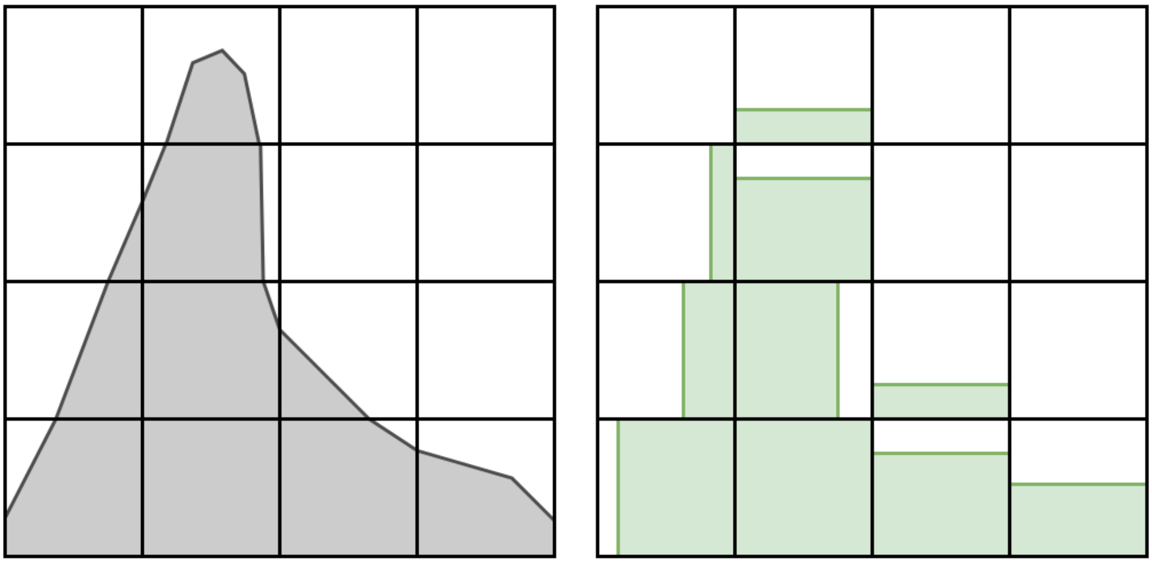
Volume-of-fluid method Another widely recognised interface capturing technique is the volume-of-fluid (VOF) method (Rider and Kothe, 1998; Scardovelli and Zaleski, 1999). The VOF method is a prominent numerical technique employed in the simulation of fluid interfaces. The method has gained popularity thanks to advantageous features such as mass conservation, computational efficiency and ease of implementation. This method delineates the interface between different fluid phases by quantifying the volume fraction of each phase within grid cells. Broadly speaking, there are two categories of algorithms for solving the transport equation: algebraic and geometric computation (Rider and Kothe, 1998).

Algebraic VOF The algebraic VOF method employs an algebraic equation to compute directly the volume fraction through the transport equation, eliminating the need for interface reconstruction. However, numerical schemes may exhibit excessive diffusion, where the volume fraction values F fall outside of the interval $[0,1]$. This excessive diffusion can be mitigated using limiter schemes. Additionally, this diffusion issue poses challenges to maintaining mass conservation during the computational process.

Drawing on the principles of flux-corrected transport (FCT), a FCT-VOF algorithm was introduced (Rudman, 1997; Zalesak, 1979). Initially, an intermediate F value is computed using a monotonic low-order flux along cell boundaries. Subsequently, a corrective flux is established to address the numerical diffusion arising from the low-order scheme. Finally, correction factors are computed to guarantee that the application of anti-diffusive fluxes does not introduce any new extremes into the solution in order to update the F value at the new time step. Oscillations may arise with high-order schemes, which need mitigating. Efforts to suppress these oscillations have been made using TVD or compressive schemes (Zhang et al., 2015). A compressive scheme CICSAM (Compressive Interface Capturing Scheme for Arbitrary Meshes) has been developed to maintain sharp interfaces by Ubbink (1997). Similarly, the THINC (Tangent of Hyperbola for Interface Capturing) method uses hyperbolic tangent functions to compute fluxes for the volume fraction (Xiao et al., 2005).

Geometric VOF On the other hand, the geometric VOF method relies on capturing the fluid interface geometry explicitly. It employs numerical techniques to reconstruct the interface, providing a geometric profile and location of the fluid phases. Changes in volume fraction are determined subsequently by integrating volume fluxes along cell boundaries. Regarding interface reconstruction, geometric VOF methods can be categorised into two classes:

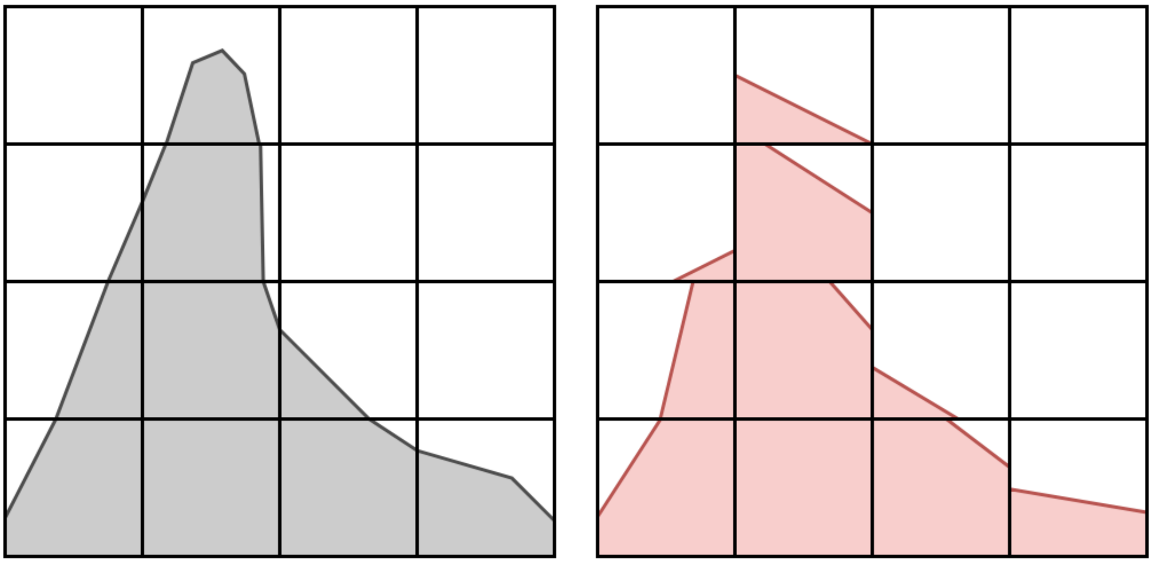
- Simple Line Interface Calculation (SLIC): Initially developed by Hirt and Nichols (1981), the interface is defined as being either horizontal or vertical within a cell and is evaluated using the volume fraction of surrounding grid cells. A piecewise constant method assumes that the interface within a cell forms a line aligned with one of the grid axes. This approach is straightforward and easy to implement but is characterised by low accuracy (see Fig. 1.5). In this instance, the orientation of the interface is determined by approximating the gradient around the desired cell. If the vertical gradient is smaller than the horizontal gradient, then the interface is horizontal and vice versa.



(a) Actual interface configuration **(b)** Hirt & Nichols SLIC interface reconstruction

Fig. 1.5 Interface reconstruction of the configuration compared to the SLIC method presented by Hirt and Nichols (1981), inspired by Rudman (1997).

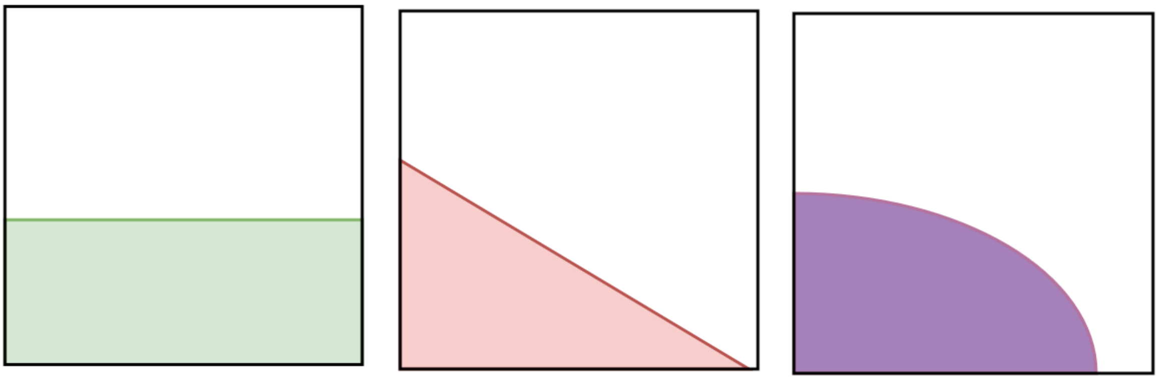
- Piecewise Linear Interface Calculation (PLIC): In a subsequent development, Youngs method introduced an improved orientation to the interface (Youngs, 1982). The interface representation within a cell is depicted as a sloped line in 2D. The slope of the line is determined by an interface normal, which corresponds to the gradient of the volume fraction, namely the volume fraction of neighbouring cells (see Fig. 1.6). PLIC methods offer increased accuracy compared to piecewise constant methods as they provide a more realistic topology of the interface (Pilliod Jr and Puckett, 2004; Rider and Kothe, 1998). In the case of the Youngs PLIC method, the orientation of the interface, β, creates an angle with the horizontal. It is evaluated using neighbouring volume fractions.



(a) Actual interface configuration **(b)** Youngs PLIC interface reconstruction

Fig. 1.6 Interface reconstruction of the configuration compared to the Youngs PLIC method presented by Youngs (1982), inspired by Rudman (1997).

Despite these advancements, traditional geometric VOF methods are not without their shortcomings. Both geometric VOF methods are subject to natural diffusion and artificial surface tension. These drawbacks may lead to rounder sharp corners causing the separation of forming filaments and exhibit large errors in reconstructing interfaces with accuracy. Moreover, these methods are limited by the constraint that only one interface can be reconstructed within a single cell, which means that structures thinner than a cell size cannot be resolved. To address these challenges, more sophisticated VOF approaches have been developed to enhance accuracy and mitigate errors related to material diffusion using a parabolic reconstruction within a cell (Evrard et al., 2023; López et al., 2004; Remmerswaal and Veldman, 2022). Often these methods are referred as piecewise parabolic interface reconstruction (PPIC). Fig. 1.7 highlights three different types of reconstruction: SLIC, PLIC and PPIC. Others have aimed to resolve filaments more effectively, particularly in scenarios where such thin structures play a crucial role (Han et al., 2024; López et al., 2005).



(a) SLIC reconstruction **(b)** PLIC reconstruction **(c)** PPIC reconstruction

Fig. 1.7 Differences in reconstruction shape between SLIC, PLIC and PPIC methods (not scaled).

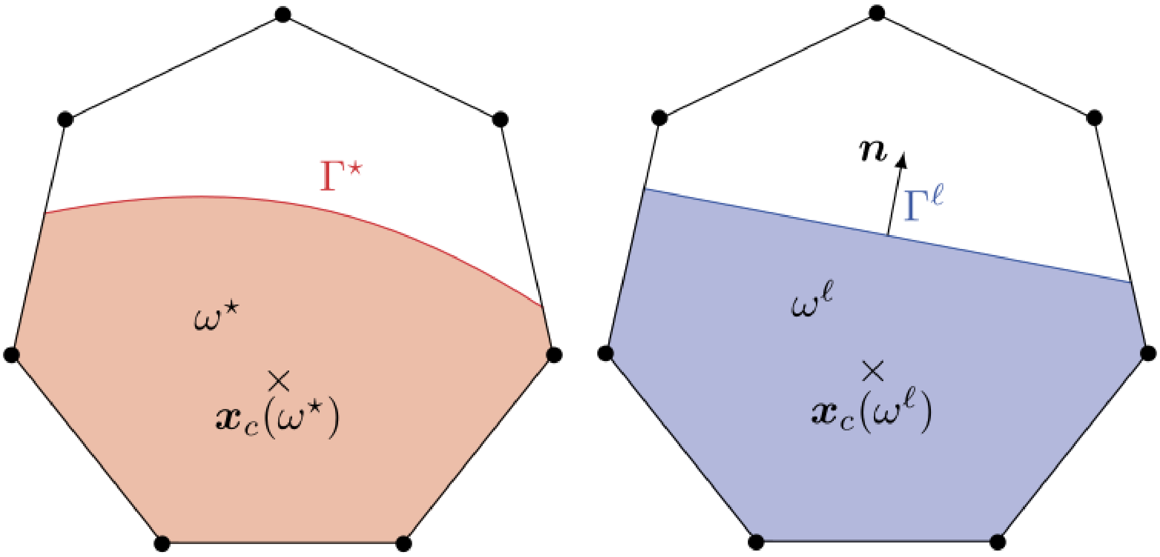
Furthermore, to leverage the strengths of both VOF and level set methods, hybrid approaches have been developed. These methods aim to combine the advantages of each technique while mitigating their respective limitations. For instance, a coupled level set and VOF (CLSVOF) method has been proposed to achieve a more robust and accurate interface representation (Sussman and Puckett, 2000).

Moment-of-fluid method The latest advancement in the evolution of the VOF method is manifested in the form of the moment-of-fluid (MOF) method. Distinguished by its innovative approach, the MOF method, initially introduced in 2005, uses both the volume fraction (zeroth moment) and the centroid (first moment) of the reference interface to accurately reconstruct the interface within a cell (Dyadechko and Shashkov, 2005). The method significantly enhances the accuracy in interface orientation. While conserving mass in each cell, the best approximation is obtained by finding the normal to the interface that minimises the distance between the centroid of the reference interface and the centroid of the reconstructed interface. Compared to a standard VOF method, the MOF method improves the precision of interface reconstruction, mitigating errors inherent in traditional approaches. Moreover, the MOF method exhibits superior mesh convergence properties.

A standout feature of the MOF method is its ability to reconstruct a piecewise linear interface without requiring the exchange of information from neighbouring cells. In addition, the MOF method can be implemented with ease for general polyhedral cells. Indeed, the initial approach encompasses both convex and non-convex polyhedral configurations. This versatility underlines its utility in diverse computational domains, where the complexity of cell shapes may vary significantly. By accommodating both convex and non-convex polyhedral cells, the approach demonstrates robustness and adaptability.

Nevertheless, it is essential to acknowledge that the MOF method contains some challenges. The time-consuming nature of the minimisation procedure is seen as the computational bottleneck. Indeed, an optimisation algorithm is inherently required in order to reconstruct the interface within each cell. In addition, it is worth noting that the proposed optimisation process relies on an initial guess, which does not necessarily ensure the optimal solution. Fig. 1.8 highlights the key concepts for the MOF approach for a general polyhedral cell with the reference (actual) interface $\Gamma^{\star}$, in which the reference material subset is $\omega^{\star}$ and its reconstructed polygon is ω^{ℓ}.

Following this advancement, Lemoine et al. (2017) made a notable breakthrough by discovering an analytical solution that removes the necessity to employ an optimisation algorithm. However, it is important to note that this solution is limited to rectangular cells,



(a) Reference interface $\Gamma^\star$ and reference centroid $\mathbf{x}_c(\omega^\star)$

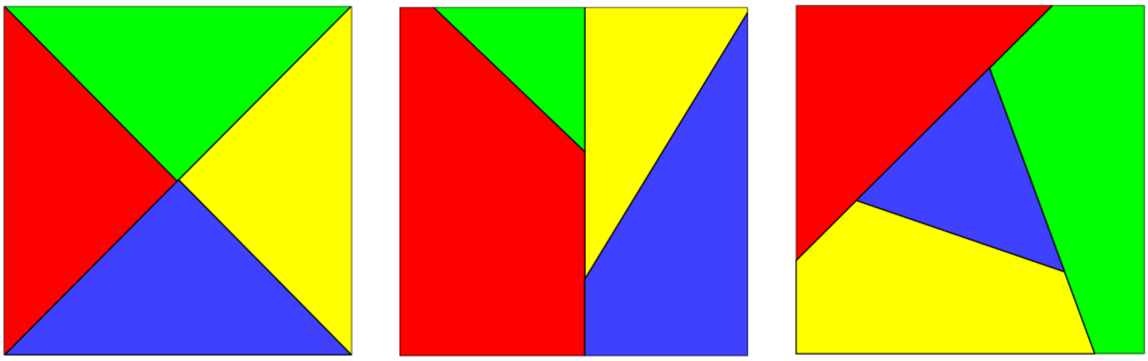
(b) Reconstructed interface Γ^ℓ and reconstructed centroid $\mathbf{x}_c(\omega^\ell)$

Fig. 1.8 Initial MOF development using volume fraction and centroid for reconstruction. The normal to the interface is represented by the outward pointing normal **n**. Image from Lemoine et al. (2017).

which means it cannot be implemented for more complex geometries. Their work has been extended to 3D rectangular hexahedral cells (Milcent and Lemoine, 2020).

Building on the contribution of Lemoine et al. (2017), recent efforts have focused on increasing the computational efficiency of the MOF method and reducing its considerable computational runtime. One notable advancement involves the utilisation of pre-computed values to accelerate the process of finding the reconstructed centroid (Cutforth et al., 2021). By leveraging pre-computed data, the computational runtime associated with interface reconstruction is significantly reduced. Moreover, additional innovative methodologies have further enhanced the computational efficiency of the MOF method. Ye et al. (2021) introduced a machine learning-based approach that results in drastic reductions in runtime. However, subsequent enhancements have exclusively targeted Cartesian cells, highlighting the limitation in applicability to other cell geometries or shape of grids.

One notable extension of the MOF method lies in the area of multi-material reconstruction, where the MOF method outshines traditional VOF techniques. In scenarios involving three or more distinct materials, multi-material interfaces are considered to be situations in which more than one interface delineate materials within a cell. This complex problem has been addressed by Dyadechko and Shashkov (2008) for up to six materials. Filaments, T-junctions and more complex interfaces are discussed using serial and nested dissections. When more than four materials are used, groups are formed and dissection is carried out, also called B-tree dissection. Fig. 1.9 shows the representation of multi-material reconstruction for four materials scenarios. This includes the X-junction, the N-junction and the Δ-junction. Ahn and Shashkov (2007) have addressed similar complex problems for general polyhedral cells, either convex or non-convex. The capability of reconstruction of multiple materials within a cell opens avenues for simulating complex multiphase systems with unprecedented accuracy and fidelity.



(a) X-junction **(b)** N-junction **(c)** Δ-junction

Fig. 1.9 Four material reconstruction for complex scenarios using a B-tree dissection multi-material MOF method.

Furthermore, efforts have been directed towards dealing with the ability of the MOF method to effectively capture under-resolved filaments, also called subgrid structures. A filament is a structure (material) containing two interfaces within a cell, which makes it challenging to detect and resolve. Indeed, understanding the topology is paramount and requires to perform multiple reconstructions to identify the solution that best matches the reference topology. The reconstruction of these structures is facilitated using the multi-material reconstruction framework (Jemison et al., 2015). First, by detecting which materials may or may not be adjacent to each other, the conglomeration algorithm enables one to identify which type of reconstruction is needed. Consequently, a fictitious material is introduced in order to proceed with a multi-material reconstruction. After reconstruction, the fictitious material is reassigned to its original phase and therefore filaments are created. However, despite great advances, this work has been combined with an adaptive grid approach and has not been applied to fixed grids. By overcoming this limitation, the MOF method enhances its applicability to a wide range of fluid dynamics phenomena, where the presence of intricate filamentary features plays a pivotal role.

Another noteworthy development is the pursuit of symmetric reconstruction techniques within the MOF framework (Hill and Shashkov, 2013). In most MOF methods, the optimisation is carried out considering the centroid of one material in the distance minimisation procedure. However, by incorporating considerations of all material information, the aim of this research is to achieve more accurate and balanced representations of complex interfaces. In a symmetric technique, the centroid of the remaining material is also considered in the objective function. Therefore, the sum of distances between both centroids and their respective reference centroid is minimised at the same time. This additional feature of the MOF method has thereby enhanced the overall robustness, reliability and versatility of the method.

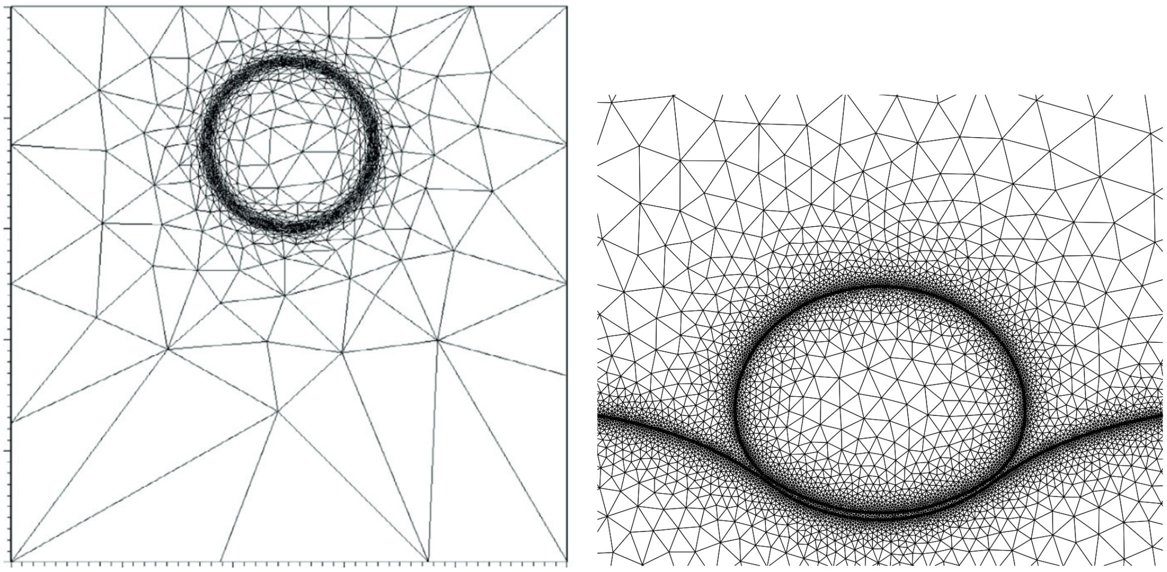
Moreover, the integration of adaptive mesh refinement (AMR) strategies into the MOF method has garnered significant attention (Ahn and Shashkov, 2009). By dynamically adjusting the mesh resolution, AMR improves computational efficiency and ensures optimal grid resolution in regions of interest. The refinement is triggered based on the error in reconstruction, which also depends on the value of the objective function. In this work however, only standard reconstruction has been considered. When large deformations of the interface occur, standard MOF techniques are not precise enough to maintain a *smooth* interface and breakup occurs similar to VOF methods. As discussed earlier, the work of Jemison et al. (2015) has later tackled the issues of adaptivity and material breakup.

Attempts have also been made to couple the level set method with the MOF method (Jemison et al., 2013; Mukundan et al., 2022). Some research has been conducted on Arbitrary Lagrangian-Eulerian (ALE) MOF methods, which aims to improve the quality of interface reconstruction by combining the advantages of Lagrangian and Eulerian approaches (Breil et al., 2013; Galera et al., 2011; Kucharik et al., 2010).

Collectively, these advancements have highlighted the evolution of the MOF method into a sophisticated and versatile tool. More recently, even more elaborate and powerful MOF methods have been developed in order to broaden the applicability of the MOF method. In 2020, a quadratic MOF was developed which enables parabolic interface reconstruction within a cell using second moments (Spainhour, 2020). Shashkov and Kikinzon (2023) and Shashkov (2023) have developed the MOF^2 which is able to reconstruct an interface with two planes within a cell. This method has been extended to circular interfaces as well . In the light of large and expensive optimisation processes, Milcent and Lemoine (2024) extended an analytical solution from Cartesian to tetrahedral meshes.

1.2.2 Adaptive schemes

Adaptive methods represent a crucial advancement in the field of computational modelling, effectively enhancing accuracy of the solution and substancially improving computational efficiency. Indeed, fixed meshes require a uniformly fine grid across the entire computational domain to precisely capture complex flow features, significantly increasing computational demand. These methods have been applied in many complex and large-scale engineering modelling applications, including fluid dynamics, climate modelling and structural analysis. Examples include adaptive unstructured mesh (Xie et al., 2016; Zheng et al., 2005), adaptive polynomial degree (Cantwell et al., 2015) and adaptive mesh refinement (Berger and Oliger, 1984; Ceniceros et al., 2010; Chen and Yang, 2014; Greaves, 2004; Hartmann et al., 2011; Liang, 2012; Mirzadeh et al., 2016; Popinet, 2009; Sussman et al., 1999) methods. Fig. 1.10 shows two examples of adaptive unstructured meshes for interface reconstruction for a deformed circle and a droplet impact case.



(a) Refinement for a circle in a square domain (Xie et al., 2016)

(b) Droplet refinement before impacting a surface (Zheng et al., 2005)

Fig. 1.10 Examples of adaptive unstructured meshes for interface reconstruction.

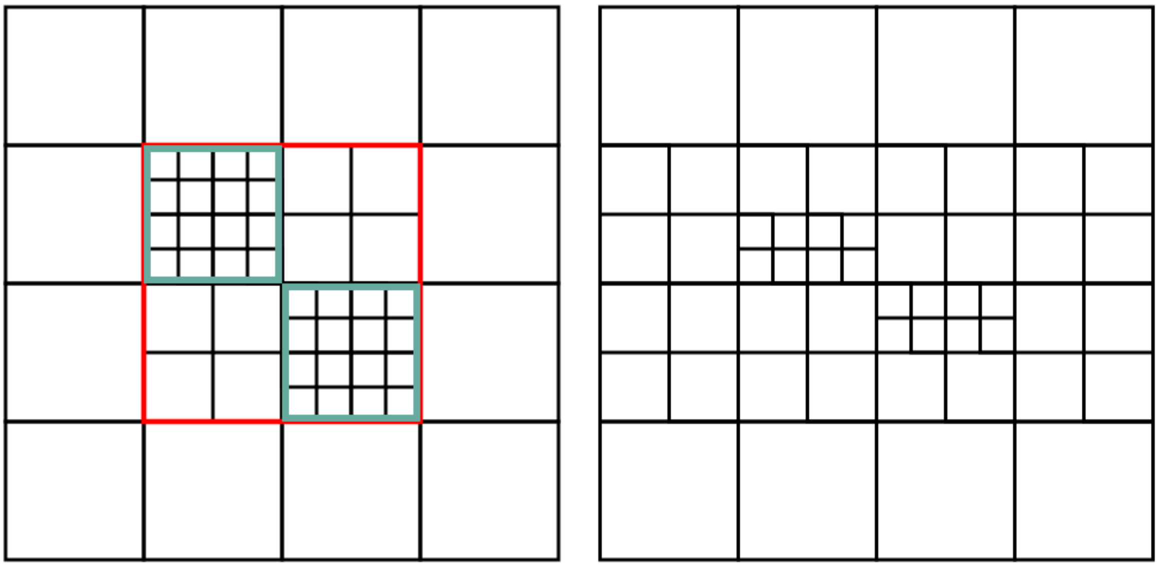
The purpose of adaptive mesh methods is to adjust dynamically the resolution of a grid in regions of interest or rapid change whilst maintaining coarse grid resolution in the regions where the solution "stagnates". AMR focuses on locally refining regions of interest in the grid, rather than uniformly refining the entire domain. Grid adaptation is triggered using a specified refinement criterion. Typically, in VOF or level set methods, the mesh is refined when either the volume fraction or level set function falls within a specific range, or the estimated curvature gradient reaches a particular threshold value. Through "optimising" the number of grid cells used in the computation, such a grid adaptation strategy may effectively enhance computational efficiency while maintaining overall solution accuracy (Liang, 2012).

In essence, AMR methods represent a cornerstone in the pursuit of efficient and accurate engineering modelling, intelligently assigning computational resources where necessary in order to tackle potentially large challenges posed by complex real-world phenomena.

In the realm of AMR, several techniques have been developed and refined over the years in order to capture interfaces effectively and handle spatial refinement (Berger and Oliger, 1984). These approaches may be broadly classified into two categories: patch-based AMR or quadtree(octree)-based AMR, each offering distinct advantages and drawbacks.

Patch-based AMR involves dividing the computational domain into a set of refinement patches. This allows for local control over the mesh resolution, and the patches can be refined or coarsened dynamically based on the evolution of the numerical solution being computed. The main advantage of patch-based AMR is its flexibility. However, it can lead to increased complexity in the maintenance and management of the grid being created, potentially increasing the computational runtime (Zeng et al., 2023).

On the other hand, quadtree(octree)-based AMR uses a tree-based hierarchical data structure. The computational domain is recursively divided into four subcells (in 2D), or eight for octree (in 3D) when the refinement criteria are met. This AMR approach typically refines regions with steep gradients or complex flow features whilst it coarsens areas with little variation. This approach has the advantage of being computationally efficient and easier to implement due to having a well-defined data structure (Liang, 2012). This method is typically implemented on Cartesian grids, and also used to support some finite element simulations on tetrahedral meshes (Antepara et al., 2021). Additionally, this method finds utility in particle methods (Fu et al., 2017). Fig. 1.11 shows a schematic of the difference between patch-based AMR grid and a quadtree AMR grid. For the patch-based grid, the first level of refinement is surrounded by a red box; the second level of refinement is surrounded by a thin green box. The quadtree grid has no color code for the refinement.



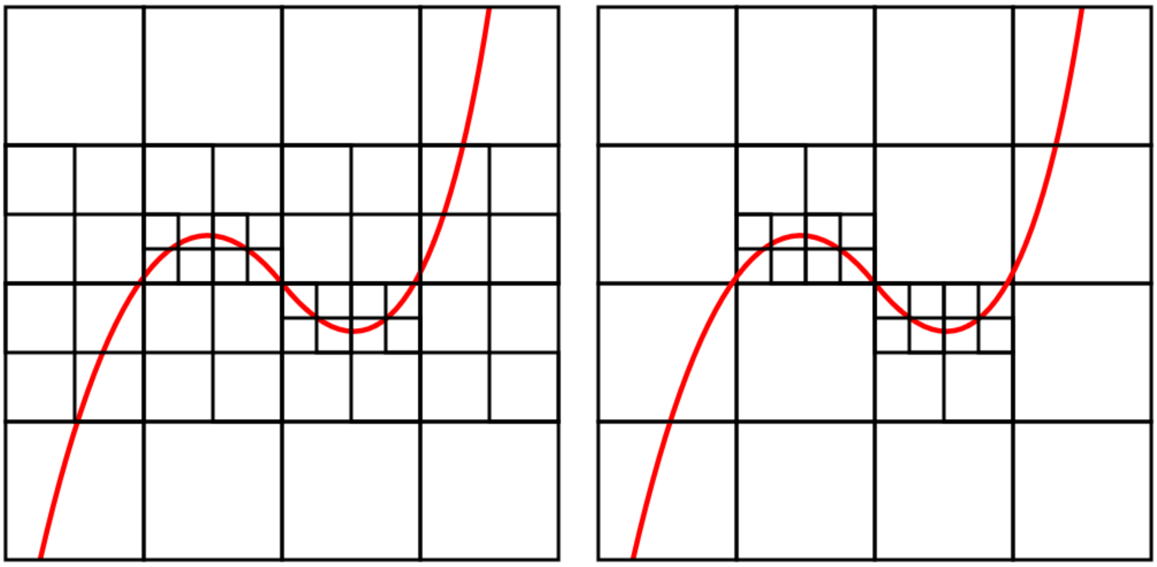
(a) Patch-based AMR grid

(b) Quadtree AMR grid

Fig. 1.11 Schematic highlighting the difference between patch-based AMR and quadtree AMR grids. On the patch-based grid, the red box depicts the level 1 patch, the green box depicts the level 2 patch.

In quadtree-based AMR, most methods use a 2 : 1 balance constraint, which refers to a condition where surrounding cells need to maintain a certain level of refinement based on the finest subcell, ensuring balance across faces and corners in 2D. This approach aims to ease mesh-based operations. On the other hand, unconstrained AMR involves dynamically adjusting the mesh resolution without allocating a constraint on refinement levels onto neighbouring cells. This approach aims to minimise computational costs. Fig. 1.12 shows the differences in grid adaptation for these two approaches for the same interface. The unconstrained approach exhibits notably fewer subcells.

In the past, AMR has emerged as a powerful technique to simulate interfacial flows using volume-of-fluid (Chen and Yang, 2014; Greaves, 2004; Popinet, 2009) and level-set (Mirzadeh et al., 2016; Sussman et al., 1999) methods. Local grid refinement has been confirmed to be effective in significantly reducing the computational cost compared to refining the entire grid while maintaining solution accuracy (Ginzburg and Wittum, 2001; Greaves, 2004; Malik et al., 2007). One of the key advantages of local grid refinement



(a) Constraint 2 : 1 balanced AMR grid

(b) Unconstrained AMR grid

Fig. 1.12 Example showing the difference between (a) constrained and (b) unconstrained adaptive mesh refinement grid for interface reconstruction.

techniques lies in its ability to mitigate unphysical material breakup. The estimated curvature (Wang et al., 2004) and interface gradient have also been used as refinement criteria.

Despite the potential gain in computational cost, using adaptive mesh refinement in the context of MOF has not been sufficiently explored. Undoubtedly, the associated complexity and natural computational cost of the MOF method itself is the reason why AMR has been limited in this context. In MOF situations, where zeroth and first moments are computed for interface reconstruction, the latter is used as a refinement criterion. Indeed, the centroid approximation is an estimate of the quality of the interface reconstruction. The accuracy of the centroid approximation directly influences the quality of the interface reconstruction, making it a suitable parameter for guiding grid refinement.

This adaptive mesh refinement method combined with the original MOF method was first developed in 2009 by Ahn and Shashkov (2009). Their approach involved the use of an unconstrained quadtree structure capable of up to five levels of refinement. In addition, the refinement criterion was set to be unique for all levels and to a value smaller than machine precision. Later, Jemison et al. (2015) proposed for the first time a filament MOF approach in a patch-based AMR framework. In the advection (dynamic) process, their innovative approach reached up to two levels of refinement and their refinement criterion included a tolerance taking into account the relative subcell size. More recently, a standard MOF-AMR using a patch-based grid has been used in the flow simulation of droplets (Islam et al., 2022). However, no attempt has been reported to combine the quadtree-based AMR and filament MOF approaches to explore their advantages in improving model performance.

1.2.3 Application to multiphase flow solver

1.2.3.1 Single grid methods

In the realm of computational fluid dynamics (CFD), numerous methods have been used to couple interface capturing/tracking methods to a fluid flow solver based on the Navier-Stokes equations. A diverse array of techniques, including finite element methods, level set methods, diffuse interface methods, and edge-tracking methods, have been employed to effectively capture and model interfaces in fluid flows (Chirco and Zaleski, 2023; Ding et al., 2007; Tryggvason, 1988; Xie et al., 2014; Xie and Stoesser, 2020). However, there has only been a limited number of attempts at performing this coupling with a MOF method. The first contribution coupled the MOF method with a finite element method (Ahn et al., 2009). Other contributions have employed a coupled level set moment-of-fluid (CLSMOF) method with a Navier-Stokes solver on a Cartesian grid but using a split advection method for several 2D and 3D benchmark problems (Jemison et al., 2013; Mukundan et al., 2022). Moreover, there are also some contributions that have used MOF in engineering applications (Banerjee et al., 2022; Islam et al., 2022; Li et al., 2013; Schofield et al., 2010). The dynamics of interfacial flows can often produce highly complex and intricate topological changes. Therefore, based on fixed meshes, a very fine mesh across the entire computational domain is required to capture the intricate interfacial details accurately, resulting in significantly increased computational effort. As such, the development and integration of efficient strategies, such as subgrid reconstruction, represent a promising avenue for mitigating computational costs while maintaining flow features and accuracy in modelling complex interfacial flows.

1.2.3.2 Dual grid methods

Within the CFD community, the dual grid method has emerged as a powerful tool to enhance the accuracy and efficiency of the computation of complex flow phenomena by combining solvers on separate grids. In this context, the aim is to solve the Navier-Stokes and transport equations on separate grids. This method involves the use of two distinct grids: a primary grid that captures the global flow variables, including velocity and pressure fields, and a secondary or dual grid that focuses on specific regions of interest, for interface reconstruction, scalar or mass transport, often where high gradients or intricate flow structures exist (Ding et al., 2007; Pozzetti et al., 2019).

Generally, the convergence of the flow might have been achieved well before the interface attains its grid independence. Therefore, maintaining identical grid resolutions for both the Navier-Stokes and transport equations, despite achieving convergence in the flow, could prolong computational time excessively and unnecessarily (Gada and Sharma, 2011; Rudman, 1998). This observation has led to the development of methodologies within the dual grid framework that enable AMR to be employed selectively, thereby using computational resources intelligently and efficiently. One prominent approach within the dual grid framework is the overset grid method, in which multiple grids with different resolutions are overlaid on one another, allowing for localised refinement where necessary (Ma et al., 2018; Wang and Stern, 2022). The overset grid method offers flexibility in grid generation, especially in complex geometries and facilitates dynamic simulations where grid movement is required. However, the method tends to focus on solid rigid boundaries, whereas the primary focus in this thesis is the dynamic deformation of interfaces. Indeed, in this context, the resolution of interfaces is improved by adaptivity.

Adaptive mesh refinement (AMR), particularly the quadtree approach, has gained significant traction in CFD due to its ability to dynamically adjust grid resolution to capture intricate flow features with high precision. The quadtree method subdivides the computational domain into quadrants, refining areas with high gradients or intricate flow details or coarsening regions with minimal variation. This ensures computational efficiency without compromising accuracy. The benefits of AMR include reduced computational cost, enhanced resolution in critical regions, and the ability to capture transient phenomena with high fidelity (Biswas et al., 1993; Hay and Visonneau, 2005). However, several challenges persist, such as ensuring solution continuity across refined boundaries, increased algorithmic complexity, and potential difficulties in parallelisation. (Berger and Oliger, 1984; Chen and Yang, 2014; Hartmann et al., 2011; Liang, 2012; Mirzadeh et al., 2016; Sussman et al., 1999). Indeed, achieving refinement on a single AMR grid may induce large computation error. The efficiency on a single grid would be compromised with the novel unconstrained AMR framework for MOF methods (Hergibo et al., 2024).

Combining a dual grid technique or a single AMR grid technique with interface capturing methods has been a subject of increasing interest in the numerical multiphase flow community. Oftentimes, the interface capturing/tracking methods have been either Volume-of-Fluid (VOF), level-set (LSM) or even recently moment-of-fluid (MOF) methods. In these, both Navier-Stokes and transport equations are solved on the same grid. In these instances, some limitations appear such as the increased complexity in combining these numerical methods (Banerjee et al., 2022; Ceniceros et al., 2010; Popinet, 2009; Xie et al., 2016). Efforts to overcome these limitations have led to the development of dual grid techniques. Dual grid techniques combined with VOF or LSM have been developed but typically employ fixed grids to facilitate the interface capturing process (Lakdawala et al., 2016; Patel and Lakdawala, 2018; Rudman, 1998). However, despite recent advances in MOF methods, the integration of dual grid techniques with fixed grids or with adaptive mesh refinement, has not been achieved.

Fig. 1.13 provides a timeline of the history and progression of the MOF method throughout the years, from its origins in 2005 to its latest advancements. Over this period of time, the MOF method has undergone considerable development in many directions, including analytical solution, filament structures, adaptive mesh refinement, and multi-material applications. In addition, advancements such as efficient algorithms, advanced MOF methodologies using second moment data, hybrid ALE techniques and coupling with level set or Navier-Stokes equations have further extended the capabilities of the MOF method. On the other hand, Fig. 1.14 provides an overview of the prominent contributions at the forefront of the innovations within the MOF context.

1.3 Research objectives and significance

The primary objective of this research in numerical multiphase flows is to advance the current state of the art by developing novel methodologies characterised by their high degree of complexity that address critical challenges in simulating multiphase flow phenomena. One key focus is on achieving a sharp interface representation, aiming to capture accurately the distinct boundaries between different phases. This involves refining numerical techniques to overcome inherent difficulties related to interfacial dynamics, ensuring that the simulated interfaces closely replicate real-world behaviour.

By pushing the boundaries of methodological intricacy, this research aspires to enhance understanding and capability across a diverse array of applications. Another goal is to ensure the correct topology of multiphase flows, emphasising the accurate representation of intricate topological changes that often occur during dynamic interactions between different fluids.

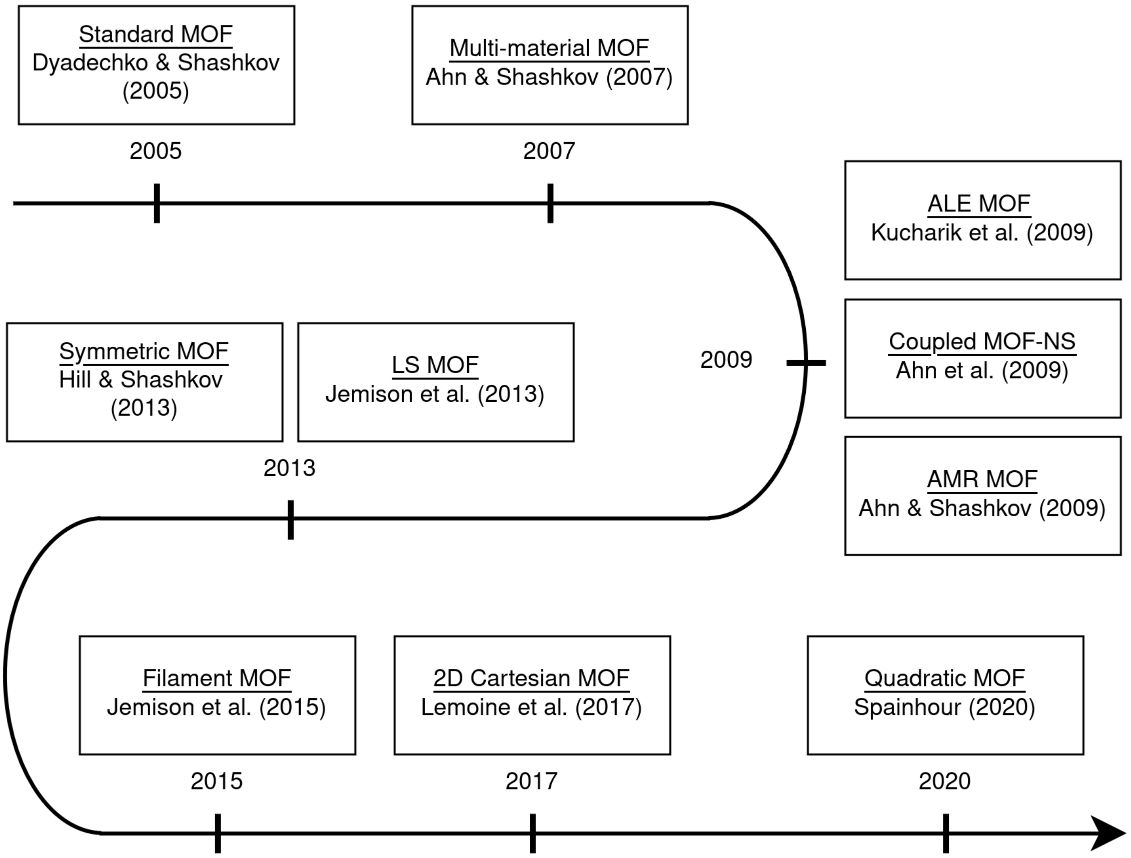


Fig. 1.13 Timeline of the MOF method.

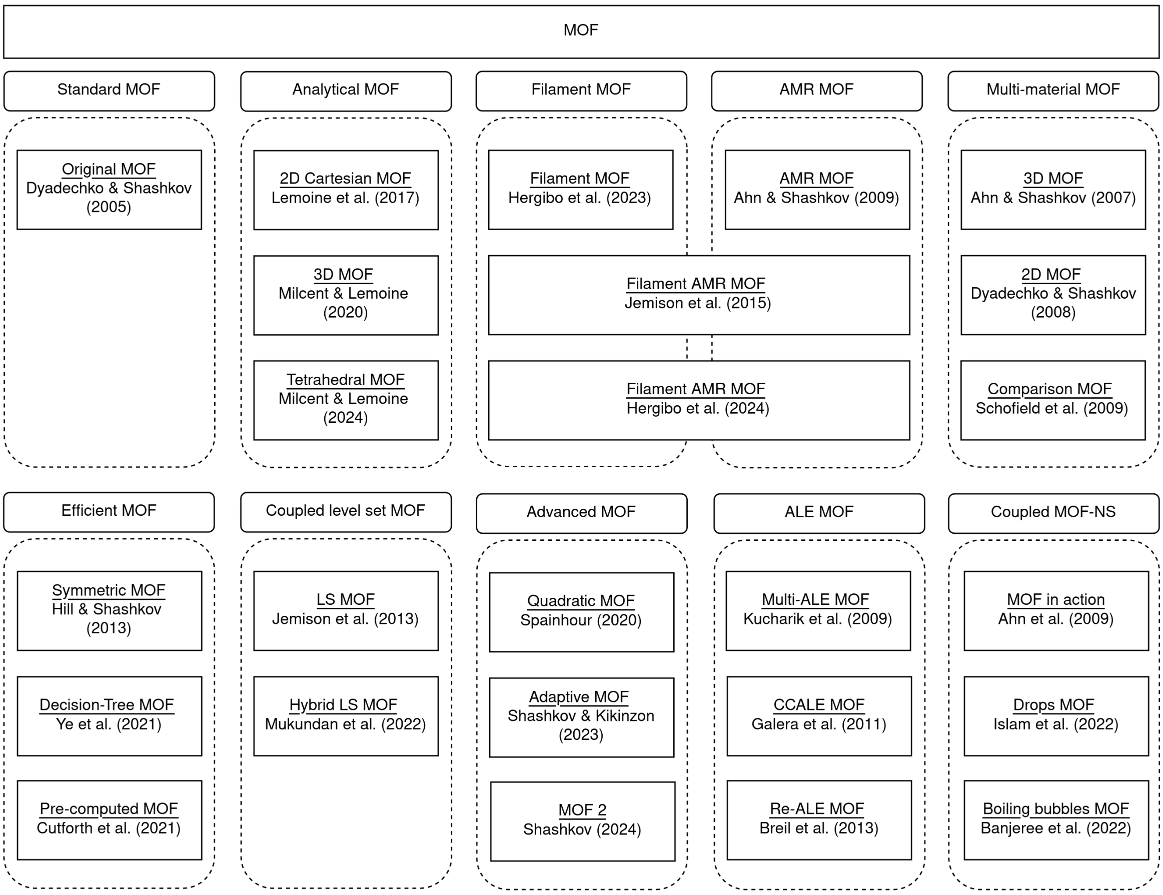


Fig. 1.14 History of the MOF method.

For instance, the research aims to model phenomena such as droplet breakup, coalescence, and phase separation with high fidelity.

Moreover, this research aims to incorporate a multi-scale perspective, recognising the inherent challenges of numerically modelling multiphase flow systems and the associated computational cost. By integrating multiple scales, the simulations aim to effectively capture large scale and sub-scale phenomena in a single numerical framework, contributing to a more comprehensive understanding of complex flow behaviour.

Additionally, a crucial aspect of this research is to develop methodologies that maintain computational efficiency. Balancing accuracy with computational cost is vital for the practical applicability of numerical methods, especially in large-scale simulations. Techniques such as adaptive mesh refinement and parallel computing will be explored to optimise computational resources while maintaining accuracy.

Overall, the objective is to contribute novel insights and computational tools that not only enhance the fidelity of multiphase flow simulations but also make them more accessible for real-world applications across engineering, environmental, and industrial domains.

Central to this research is the development of a robust and novel moment-of-fluid method, which specifically enables thin interfacial structures to be represented seamlessly. Additionally, an adaptive scheme extends the scope of this method with increased efficiency and simplicity. The application of these advanced methods to real-world scenarios marks a significant milestone. With the subsequent integration of a dual grid framework, the method paves the way for consistent applicability across various scientific and engineering contexts.

Critical to the credibility of this research, each step of the development has been thoroughly and rigorously validated using a variety of benchmark problems from the literature, including some experimental references. The validation process ensures the reliability and accuracy of the proposed methodologies.

1.4 Outline of the thesis

This thesis consists of seven chapters, with the current one serving as the introduction chapter. The reminder of the thesis is organised as follows.

Chapter 2 lays the theoretical foundation by delving into the fundamental equations governing multiphase flows. This includes an in-depth exploration of the challenges inherently associated with solving these equations numerically. Detailed discussions are provided on the Navier–Stokes equations, as well as the transport equation used for capturing interfaces. Additionally, a comprehensive explanation of the finite volume discretisation applied to the governing equations including the coupling of pressure and velocity are provided in this chapter. Furthermore, the formulation of initial and boundary conditions within this numerical framework is described.

In Chapter 3, the moment-of-fluid (MOF) method is introduced, providing a comprehensive overview of its principles and computational intricacies. Special attention is given to the novelty associated with subgrid structures and its limitation in capturing multiphase interfaces. This chapter establishes a baseline for the subsequent development and enhancement of numerical methods in the following chapters.

Chapter 4 introduces novel enhancements in the form of adaptivity within the MOF method including a novel dual grid framework. These advancements address the extension of the MOF method capability by offering improvements in both accuracy and efficiency. The chapter outlines the criteria and methodologies surrounding the novel data structure for refinement. Details are given on the implementation of the novel dual grid framework for capturing complex multiphase flow phenomena.

In Chapter 5, attention is directed towards the practical application of the MOF method on fixed grids, with a specific focus on resolving thin structures in multiphase flow scenarios. This chapter presents simulation results and validates the filament MOF approach against existing references in the literature. Discussions include an analysis of the physics associated with the coupling between a flow solver and the novel MOF method, emphasising the accuracy and reliability of the proposed model in capturing intricate and filamentary features of multiphase flows.

In Chapter 6, the outcomes of applying the adaptive MOF and dual grid methods are introduced. Through a series of simulations and validation exercises, this chapter evaluates the performance of these enhancements in various interface reconstruction scenarios. The application and performance of the proposed dual grid method to real-world cases are assessed with reference to predictions of existing methods and experimental data where available. The chapter includes a complex scenario highlighting the practical implications of the dual grid method for future developments in multiphase flow simulations.

Finally, Chapter 7 presents a summary of the primary findings and contributions. Additionally, it discusses potential avenues for future work, highlighting areas where further exploration and refinement could contribute to advancements in the field of multiphase flow simulations, in particular interface capturing techniques.

Chapter 2

Mathematical model and discretisation

2.1 Introduction

The primary objective of this thesis is the development of a numerical methodology capable of predicting the flow characteristics and features of two immiscible fluids, separated by a well defined interface. A mathematical model, describing the flow of the two fluids and the movement of the free surface or interface separating them is the subject of this chapter.

A mathematical model, which uses the continuum mechanics approach is used in this thesis. The fluids are modelled as a continuum with a jump in the fluid properties at the interface. As the purpose of the current work is to develop and test a methodology for interface capturing and since the test cases involve flows dominated by inertial and pressure forces rather than by turbulence effects, the issue of turbulence modelling and simulation will not be considered in any depth.

The discretisation framework introduced in this thesis lays the foundation for numerical investigations of fluid flow dynamics. Grounded in the principles of finite volume methods, the framework offers a rigorous and systematic approach to translating the continuum-based mathematical model, specifically the Navier-Stokes equations, into a discretised form for the purpose of computational implementation.

2.2 General transport equation

The fluid flow is described mathematically by three conservation laws, namely, the conservation of mass, momentum and energy. These laws completely determine the physical behaviour of the fluid and are totally independent of the nature of the fluid. The fluid characteristics itself are defined by physical properties such as density, viscosity, surface tension, conductivity and compressibility. The general form of the conservation equation for a flow quantity ϕ within a control volume V_{CV} with boundary ∂V is described as follows:

$$\frac{d}{dt}\int_{V_{CV}} \phi \, dV + \oint_{\partial V} \mathbf{F_C} \cdot \mathbf{dS} + \oint_{\partial V} \mathbf{F_D} \cdot \mathbf{dS} = \int_{V_{CV}} Q_S \, dV \ , \tag{2.1}$$

where t is the time, $\mathbf{F_C}$ is the convective flux vector field over the boundary, often written as $\mathbf{u}\ \phi$, where $\mathbf{u}$ is the fluid velocity, $\mathbf{F_D}$ is the diffusive flux vector field over the boundary, Q_S is the source term and $\mathbf{dS}$ the outward pointing surface vector to ∂V. Fig. 2.1 highlights the presence of each term defined above.

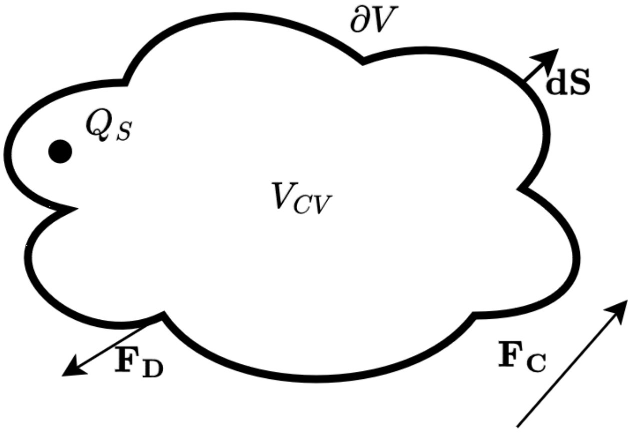


Fig. 2.1 Schematic of a control volume highlighting each term described in the general form of the transport equation.

In continuum mechanics, Gauss' theorem, also known as the divergence theorem, can be used. It relates the flux of a vector field across a closed surface and the divergence of the field within the enclosed volume. This means that the convective and diffusive terms may be expressed in the following alternative form:

$$\frac{d}{dt}\int_{V_{CV}} \phi \, dV + \int_{V_{CV}} \nabla \cdot \mathbf{F_C} \, dV + \int_{V_{CV}} \nabla \cdot \mathbf{F_D} \, dV = \int_{V_{CV}} Q_S \, dV \tag{2.2}$$

The general form of the transport equation for a flow quantity is valid for a scalar, vector or tensor field. If the control volume is taken as a single point, the general equation can be derived as:

$$\underbrace{\frac{\partial \phi}{\partial t}}_{\text{unsteady term}} + \underbrace{\nabla \cdot (\mathbf{u}\, \phi)}_{\text{convective term}} + \underbrace{\nabla \cdot F_D}_{\text{diffusive term}} = \underbrace{Q_S}_{\text{source term}} \tag{2.3}$$

2.3 Navier-Stokes equations

The governing equations for fluid flows are known as the Navier-Stokes equations. These equations are derived from the general form of the transport equation. Several assumptions are made in order to derive these three conservation laws. In this thesis, temperature is ignored, therefore the energy equation is omitted. The other two conservation laws, namely the conservation of mass and the conservation of momentum are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.4}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)\right] + \mathbf{f}\,, \tag{2.5}$$

where **u** is the velocity vector, p is the pressure, **f** represents the external body forces and t is time. The superscript T denotes the transpose. The fluid parameters, ρ and μ are the density and dynamic viscosity, respectively. These equations can be described in 2D or 3D in component form with respect to the Cartesian coordinate system.

In the case where incompressibility of the flow is assumed, the conservation of mass, or continuity is simplified as the overall change in material density is neglected,

$$\nabla \cdot \mathbf{u} = 0 \tag{2.6}$$

In terms of physical properties in multiphase flow, the flow solver is constituted of relations for the density and dynamic viscosity of different fluids. These properties are defined by:

$$\rho = F\rho^a + (1-F)\rho^b \ , \tag{2.7}$$

$$\mu = F\mu^a + (1-F)\mu^b \ , \tag{2.8}$$

respectively, where the two fluids are labelled '*a*' and '*b*' and F denotes the volume fraction of a fluid 'a' within a control volume, explicitly the share of a single fluid or another within a defined control volume. In a continous level, the volume fraction is defined by a step function. On a discrete level, the volume fraction can be defined by:

$$F = \begin{cases} 1 & \text{the control volume is full of fluid } \textit{'a'} \\ 0 < F < 1 & \text{the control volume contains both fluids} \\ 0 & \text{the control volume is full of fluid } \textit{'b'} \end{cases} \tag{2.9}$$

Fig. 2.2 showcases a schematic of the configuration of two-phase flows with respective densities and viscosities.

The air-water interface is then defined within the control volumes where $0 < F < 1$. The transport of F with the fluid is analogous to particles moving on a surface. It accounts for the change in phase fraction and transport of phase. In Lagrangian terms, F has a zero material derivative in the continuum approach,

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0 \tag{2.10}$$

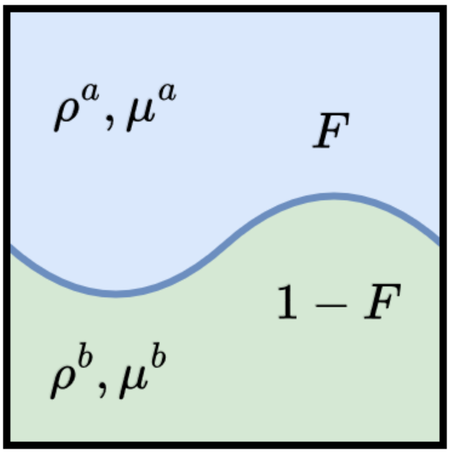


Fig. 2.2 Physical properties within a control volume highlighting the density and viscosity of fluid '*a*' and '*b*'.

Together with Eq. (2.6) and Eq. (2.5), this equation completes the mathematical description of multiphase flows. The aforementioned equations are general and provide a full description of the dynamics of the two fluids and the interface separating them. Equation (2.10) will be further discussed in the next chapter. This set of equations only possesses analytical solutions in limited cases which are often very simplified cases under specific initial and boundary conditions. Therefore, numerical solutions are typically required in order to transition from a continuous to a discrete level.

2.4 Discretisation of the Navier-Stokes equations

Discretisation allows the conversion of these continuous equations into discrete forms that can be solved numerically using discrete control volumes, such as grids or meshes. This facilitates the use of computational methods like finite difference, finite element, or finite volume methods to approximate solutions by dividing the domain into smaller elements. In the finite volume method, these smaller elements are called control volumes or cells; a control volume is bounded by a series of faces, allowing the transformation of Eq. (2.2) into a sum of integrals over these faces. For instance, the divergence term in the transport equation (second term of Eq. (2.2) can be discretised as follows:

$$\int_{V_{CV}} \nabla \cdot \phi \, dV = \oint_{\partial V} \phi \, \mathbf{dS} \approx \sum_{f=1}^{n} A_f \cdot \phi_f \ , \tag{2.11}$$

where f is the index labelling the centre of each cell face, A_f is the face area vector and n is the number of faces of a control volume. The same transformation can be applied for other terms in the integral or divergence form of the Navier-Stokes equations.

2.4.1 Finite volume method

The governing equations are discretised on a staggered mesh using the finite volume method and solved using a Cartesian grid-based fluid flow solver, which is chosen for its suitability in discretising partial differential equations. In this part, the application of the finite volume approach to the Navier-Stokes equations is discussed using Cartesian cells or control volumes to capture the dynamics of multiphase flows. This approach involves dividing the computational domain into a structured Cartesian grid, where the fluid flow solver utilises staggered grid arrangements for the flow variables. This Cartesian grid approach can also deal with either fixed or moving complex geometries using a cut-cell method (Xie, 2022). Fig. 2.3 highlights the arrangement of velocity and pressure unknowns on the staggered grid. The staggered grid arrangement removes the spurious chequerboard pressure mode that is present on a collocated grid.

The finite volume method, applied to this staggered grid configuration, ensures the conservation of mass and momentum by integrating the governing equations over control volumes. This combined strategy not only simplifies the representation of complex geometries but also enhances numerical stability, making it a widely employed and effective approach in CFD.

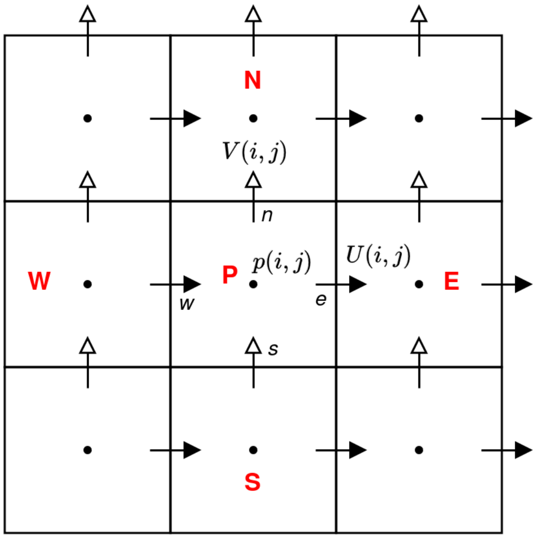


Fig. 2.3 Staggered grid arrangement on a 3 × 3 grid. The black circle represent the cell-centered pressure $p(i,j)$. The plain arrows represent face-centered horizontal velocities $U(i,j)$ and the hollow arrows face-centered vertical velocities $V(i,j)$. The red letters correspond to the present cell P and its neighbouring cells N, S, W, E as well as cell faces in black *e*,*w*,*n*,*s*.

Using a similar discretisation as in Eq. (2.11) and substituting each term into the integral form of Eq. (2.3), the resulting integrated momentum equation for a control volume with centre P is:

$$\begin{gathered}\frac{\rho V_{CV}}{\Delta t}\left(\phi_P-\phi_P^0\right)+\left(F_e\phi_e-F_w\phi_w\right)+\left(F_n\phi_n-F_s\phi_s\right)=\\ D_e\left(\phi_E-\phi_P\right)-D_w\left(\phi_P-\phi_W\right)+D_n\left(\phi_N-\phi_P\right)-D_s\left(\phi_P-\phi_S\right)+Q_S V_{CV}\ ,\end{gathered} \tag{2.12}$$

where F_f and D_f represent the convective and diffusive coefficients of the variable ϕ at the cell face f with surface area A_f and Q_S the source term. Additionally, Δt and ϕ_P^0 represent the time step and value of ϕ_P at the previous iteration, respectively. Note that the pressure term and any other body force terms are included in the source term. Uppercase letters $\{P, E, W, N, S\}$ denote the values of ϕ at the centre of the present and neighbouring control volumes, lowercase letters denote the values at cell faces $f = \{e, w, n, s\}$. The coefficients F_f

and D_f can be expressed as:

$$F_f = (\rho u A)_f \qquad\qquad D_f = \left(\mu \frac{A}{\Delta}\right)_f , \tag{2.13}$$

where μ is the viscosity and Δ is the cell size in the x- and y-directions.

Collecting terms involving ϕ_P, Eq. (2.12) can be rearranged to yield,

$$a_P \phi_P + \sum_{nb} a_{nb} \phi_{nb} = b_P , \tag{2.14}$$

(see Ferziger and Peric (2002)). The subscript *nb* refers to the neighbouring control volumes. The process of integrating the momentum equation across all control volumes throughout the entire domain results in an algebraic system of equations for the unknown values of the variable ϕ. This integration is performed separately for the momentum equations in each direction. Regarding the system of equations and its right-hand side, the source term encompasses the pressure gradient in their respective directions and the source term itself. The source term b_P in the horizontal component of the momentum equation is expressed as:

$$b_P = -\frac{p_{nb} - p_P}{\Delta} V_{CV} + Q_S V_{CV} , \tag{2.15}$$

where V_{CV} denotes the area/volume of the cell and p_{nb} the cell-centered pressure of neighbouring control volume and Δ represents the spatial displacement.

Because of the nonlinear nature and interdependence of the underlying differential equations, it is not feasible to solve Eq. (2.14) directly since the coefficients of the equation and the source term depend on the unknown solution ϕ_P. An iterative solution method is used in this approach.

2.4.2 Pressure-velocity coupling

In the incompressible Navier–Stokes equations, the pressure and velocity are decoupled because the pressure term does not appear in the continuity equation. However, this thesis employs a staggered grid, known for its effective coupling of pressure and velocity, introduced through the discretisation process. On a 2D Cartesian grid, as illustrated in Fig. 2.3, velocities are situated at the face centre of the control volume, while the pressure is stored at the cell centre. In addition, the volume fraction F is stored at the same location as the pressure.

The nonlinear system of equations representing the momentum equation can be discretised implicitly for the velocities at the subsequent time step, where the pressure term is separated from the source term, as follows:

$$a_P^{\mathbf{u}}\mathbf{u}_P^{n+1} + \sum a_{nb}^{\mathbf{u}}\mathbf{u}_{nb}^{n+1} = b_P^{\mathbf{u}} - \frac{(p_{nb} - p_P)}{\Delta} \,, \tag{2.16}$$

where $a^{\mathbf{u}}$ is the coefficient for the momentum equation and the subscripts P and $nb = E,W,N,S$ denote the variables in the present and neighbouring cells (shown in Fig. 2.3), respectively, and $b_P^{\mathbf{u}}$ is the source term, which in this case does not include the pressure term.

A guessed pressure, denoted as p^*, is necessary to facilitate the solution of velocities, represented by $\mathbf{u}^*$. This is expressed as:

$$a_P\mathbf{u}_P^* + \sum_{nb} a_{nb}\mathbf{u}_{nb}^* = -\frac{p_{nb}^* - p_P^*}{\Delta}V_{CV} + Q_S V_{CV} \tag{2.17}$$

Since the pressure in this equation corresponds to the previous time iteration, the velocities obtained from the momentum equations only fulfill the momentum requirements, hence there is no assurance regarding continuity. In this context, suppose the correct pressure is obtained by adding a pressure correction p' and similarly the correct velocity by adding a velocity correction $\mathbf{u}'$.

$$p_P = \underbrace{p_P^*}_{\text{guessed}} + \underbrace{p_P'}_{\text{corrected}} \qquad\qquad \mathbf{u}_P = \underbrace{\mathbf{u}_P^*}_{\text{guessed}} + \underbrace{\mathbf{u}_P'}_{\text{corrected}} \tag{2.18}$$

By substituting these expressions into the corresponding algebraic systems of equations Eq. (2.16), assuming that the $\mathbf{u}^*$ values satisfy each equation, then the following system of equations is derived for the corrected velocities,

$$a_P \mathbf{u}_P' + \sum_{nb} a_{nb} \mathbf{u}_{nb}' = -\frac{p_{nb}' - p_P'}{\Delta} V_{CV} \tag{2.19}$$

Ultimately, the pressure correction values p' are determined through the continuity equation, which has to be satisfied, using the initially guessed velocities $\mathbf{u}^*$ along with the corrections $\mathbf{u}'$. In this instance, the continuity equation (2.6), using the general discretisation form, can be expressed as:

$$\rho u_e A_e - \rho u_w A_w + \rho v_n A_n - \rho v_s A_s = 0\ , \tag{2.20}$$

where u_e and u_w are the horizontal velocities and v_n and v_s the vertical velocities within the control volume.

In order to obtain the resulting pressure correction, the discretised form of the equation is:

$$a_P^p p_P' + \sum_{nb} a_{nb}^p p_{nb}' = b_P' \tag{2.21}$$

The term b_P', referred to as the mass residual, in the pressure correction equation corresponds to the left-hand side of the discretised continuity equation Eq. (2.20). A zero value of b_P' indicates that the velocity field satisfies the continuity equation (2.6).

Once the pressure correction equation is solved, the pressure can be determined using the initially guessed and corrected pressure terms. Since this iterative scheme converges slowly, an under-relaxation factor α_p is used in the discretised momentum equation to accelerate the iterative process towards the solution (Ferziger and Peric, 2002). Typically, $\alpha_p = 0.3$ is used.

The SIMPLE (Semi-Implicit Method for Pressure-Linked Equations), PISO (Pressure-Implicit with Splitting of Operators), and PIMPLE (PISO-SIMPLE) algorithms are particular examples of pressure-correction methods used to solve the coupled system of equations described above. The SIMPLE algorithm solver approximates the velocity correction term by ignoring their neighbouring contributions (Caretto et al., 1973; Patankar, 2018). The PISO algorithm uses a second additional correction step (Issa, 1986). PIMPLE is a hybrid algorithm that combines features of the SIMPLE and PISO schemes (Ferziger and Peric, 2002). The first two methods are in widespread use in CFD. The SIMPLE method demonstrates robustness in handling steady problems, while the PISO method proves to be more efficient when addressing transient problems.

In this work, the SIMPLE algorithm will be used to solve the implicit pressure-velocity coupling described above. It is widely used to ensure stability and accuracy in CFD simulations. The SIMPLE algorithm is summarised in Algorithm 1.

2.5 Initial and boundary conditions

In order to fully describe the mathematical model, it is critical to establish boundary conditions that play a role in defining fluid flow behaviour at the boundaries of a finite computational domain. They capture accurately the interaction between the fluid and its surroundings. Let us define a domain Λ, the surroundings of the domain, its boundaries $\partial\Lambda$ and the unit outward pointing surface normal vector $\mathbf{n}$. In mathematical terms, there are five types of boundary conditions, namely Dirichlet, Neumann, Robin, mixed and Cauchy. However, the focus will be on the first two in this thesis:

Algorithm 1 SIMPLE algorithm

```
**Start** calculation at the new time $t^{n+1}$ using $\mathbf{u}^n$ and $p^n$
**Set** variables of the momentum equation $\mathbf{u}^* = \mathbf{u}^0$ and $p^* = p^0$
**Solve** equations to obtain $\mathbf{u}^*$
    $a_P\mathbf{u}_P^* + \sum_{nb} a_{nb}\mathbf{u}_{nb}^* = -\frac{p_{nb}^* - p_P^*}{\Delta} V_{CV} + Q_S V_{CV}$
**Evaluate** coefficients of the continuity equation
**Solve** the pressure-correction equation to obtain $p'$
**Correct** velocity and pressure to obtain $\mathbf{u}$ and $p$
    $\mathbf{u} = \mathbf{u}^* + (1 - \alpha_p)\mathbf{u}'$
    $p = p^* + \alpha_p \, p'$
**Repeat** until convergence is achieved
**Set** $\mathbf{u}^{n+1} = \mathbf{u}$ and $p^{n+1} = p$
**Advance** to next time iteration
**End**
```

- Fixed boundary value, also called the Dirichlet boundary condition. An arbitrary variable φ has a unique value defined at the boundary. This can be expressed as:

$$\varphi|_{\partial\Lambda} = B_D \; , \tag{2.22}$$

  where B_D is the unique value

- Fixed boundary gradient, also called Neumann boundary condition. An arbitrary variable ϕ has its gradient defined at the boundary. This can be expressed as:

$$\left.\frac{\partial\varphi}{\partial n}\right|_{\partial\Lambda} = B_N \; , \tag{2.23}$$

  where B_N is the unique value

It is important to be reminded that no boundary conditions are imposed on the interface between two fluids as the Navier-Stokes equations and the advection equation for the volume fraction yield the interface.

2.5.1 Boundary conditions

In the realm of multiphase flows, boundary conditions play a crucial role in defining the interactions between different phases and fluid characteristics within the computational domain. In the context of multiphase flows, the various boundary conditions are often referred as inflow, outflow, periodic and wall conditions. These can be classified as follows:

- **Inflow Boundary Condition**: This condition specifies fluid characteristics upon entering the domain. It applies at boundaries where one or more phases enter the computational domain. For instance, the inflow boundary may specify the properties of the incoming velocity. Therefore, it may use the Dirichlet boundary condition where $\mathbf{u}(\mathbf{x},t) = \mathbf{g}(\mathbf{x},t)$, where $\mathbf{g}$ is a user-specified function describing the inflow properties, $\mathbf{x}$ denotes spatial coordinates, t the time.

- **Outflow Boundary Condition**: Analogous to the inflow condition, the outflow boundary is where one or more phases exit the computational domain. It involves specifying the properties of the outgoing phases, ensuring a realistic representation of the flow behaviour at the domain's exit.

- **Periodic Boundary Condition**: In cases where the multiphase flow exhibits periodic behaviour, such as waves or repeating patterns, periodic boundary conditions are employed. These conditions enforce a periodic relationship between corresponding points on opposite boundaries, ensuring the simulation captures the periodicity of the flow. Mathematically, the imposed periodic condition may typically be written as $\mathbf{u}(\mathbf{x}+L) = \mathbf{u}(\mathbf{x})$, where L is the length of the domain.

- **Solid Boundary Condition**: This applies at boundaries representing solid surfaces. This condition accounts for the interaction between the fluid phases and the solid structure. There are two main types of solid boundaries, no-slip and slip conditions. The first one is commonly used in many applications, and assumes that the fluid at the solid boundary comes to a complete stop, adhering to the stationary solid surface. Mathematically, it can be defined as $\mathbf{u}(\mathbf{x},t) = 0$. In this context, it is important to highlight the both normal and tangential velocity are imposed to be zero. These are known as the no-penetration and no-slip conditions respectively.

Each of these boundary conditions plays a role in establishing a clearly defined computational domain that replicates the physical intricacies of multiphase flows, enabling precise and insightful simulations of intricate fluid interactions.

2.5.2 Initial condition

Initial conditions are described across the computational domain at the start of a simulation, at $t = 0$. Most of the time, users define the initial stage, however it does not necessarily mean that the setup is at rest. For computation, the velocity and pressure field are initialised. Additionally, for multiphase flows, the volume fraction is initialised.

The velocity can either be initialised at rest or with a prescribed velocity. The pressure is oftentimes initialised using hydrostatic pressure depending on the water depth in the domain. The initialisation of the volume fraction involves defining the volume fraction across all control volumes in the domain. It also involves defining the initial interface normal in each cell. Generally, cells can be either empty, full or contain an interface. Most of the time the interface can be horizontal, vertical or with a specified normal. Examples may be a flat surface, bubbles or droplets, or waves which often use an analytical solution as initial condition. In VOF methods, the normal of the interface is defined based on the neighbouring volume fraction. In the context of MOF methods, the initial normal of the interface is determined using knowledge of the centroid.

2.6 Closing remarks

In this chapter, the mathematical model and its numerical discretisation using the finite volume method are presented and the pressure-velocity coupling intricacies are described. While focusing on the Navier–Stokes equations, the extension of numerical methods to multiphase flows by incorporating an advection equation for the volume fraction is highlighted. The finite volume discretisation for two-fluid systems involved control volumes with arbitrary topology. The initial and boundary conditions have also been presented in this chapter. Additionally, the presentation of boundary conditions details the interactions between the computational domain and its surroundings, ensuring a well-posed mathematical formulation in order to obtain realistic representation of the physical constraints. The forthcoming chapter will present a comprehensive interface capturing technique, namely the moment-of-fluid method.

Chapter 3

Numerical interface calculation

3.1 Introduction

As technological advancements continue to enable increasingly sophisticated simulations, the accurate calculation of interfaces becomes essential for the successful representation of physical phenomena. Understanding and predicting the behaviour of interfaces has become an indispensable tool in order to study complex multiphase flow problems.

This chapter is concerned with numerical interface calculations, exploring its significance in the context of multiphase flow simulations. A comprehensive overview of numerical interface calculation methods employed in recent years is described earlier (see Section 1.2). In this chapter, the focus will be given on interface capturing methods, in particular the geometric Volume-of-Fluid (VOF) and moment-of-fluid (MOF) methods. The VOF method will be described initially, as a precursor to a recent extension of the technique, the MOF method. The standard MOF method will be examined, followed by the exploration of the filament MOF method. Finally, to establish a connection between interface capturing techniques and the underlying physics associated with fluid flow behaviour, the coupling between MOF methods and a flow solver and its integration will be discussed briefly.

3.2 The Volume-of-Fluid (VOF) method

In this section, the geometric VOF method is described. There are two variants of this explicit numerical technique employed in the simulation of fluid interfaces differentiated by the way that the interface is reconstructed: SLIC and PLIC. The first one captures the interface within a cell with a straight line, either horizontal or vertical. The second reconstructs interfaces with a sloped straight line.

3.2.1 SLIC reconstruction

The SLIC method is a piecewise constant method originated by Hirt and Nichols (1981), where the interface within a cell forms a line aligned to one of the grid axes (horizontal or vertical) (see Fig. 1.5). This approach is straightforward and easy to implement. In this case, the orientation of the interface is calculated by approximating the gradient of the cell. The vertical gradient can be expressed using neighbouring volume fractions as follows:

$$\left.\frac{dY}{dx}\right|_i = \frac{2(Y_{i+1} - Y_{i-1})}{dx_{i+1} + 2dx_i + dx_{i-1}} , \tag{3.1}$$

where dx_i represents the horizontal length scale of cell i, j, and Y_{i+1} and Y_{i-1} are determined using the expression which contains the volume fraction of $F(i, j)$ and their values either side of cell i, j.

$$Y_i = F(i, j-1)dy_{j-1} + F(i, j)dy_j + F(i, j+1)dy_{j+1} \tag{3.2}$$

Similarly, the horizontal gradient can be expressed as:

$$\left.\frac{dX}{dy}\right|_j = \frac{2(X_{j+1} - X_{j-1})}{dy_{j+1} + 2dy_j + dy_{j-1}} , \tag{3.3}$$

where on this occasion, dy_j represents the vertical length scale of cell i, j, and X_i is expressed as the sum of the volume fraction of $F(i, j)$ and its adjacent cells in the horizontal direction.

$$X_j = F(i-1, j)dx_{i-1} + F(i, j)dx_i + F(i+1, j)dx_{i+1} \tag{3.4}$$

If the vertical gradient is smaller than the horizontal gradient in magnitude, then the interface is horizontal. Contrariwise, if the magnitude of the horizontal gradient is smaller than the magnitude of the vertical gradient, the interface is represented vertically.

3.2.2 PLIC reconstruction

In the PLIC method used by Youngs (1982) and updated by Rudman (1997), the interface is represented with a slope within a cell (see Fig. 1.6). The slope of the line is calculated using the volume fractions of neighbouring cells to evaluate a normal to the interface. The neighbouring cells aid in evaluating the gradient of a desired cell. In the case of the Youngs PLIC method, the orientation of the interface, β, creates an angle with the horizontal. It is evaluated using neighbouring volume fractions as follows:

$$\beta = \tan^{-1}\left(-\frac{n^x_{i,j}}{n^y_{i,j}}\right) , \tag{3.5}$$

where $n^x_{i,j}$ and $n^y_{i,j}$ are the horizontal and vertical components of the gradient and can be expressed as follows:

$$\begin{aligned} n^x_{i,j} = \frac{1}{dx}\Big(&F(i+1, j+1) + 2F(i+1, j) + F(i+1, j-1) \\ &-F(i-1, j+1) - 2F(i-1, j) - F(i-1, j-1)\Big) \end{aligned} \tag{3.6}$$

$$\begin{aligned} n^y_{i,j} = \frac{1}{dy}\Big(&F(i+1, j+1) + 2F(i, j+1) + F(i-1, j+1) \\ &-F(i+1, j-1) - 2F(i, j-1) - F(i-1, j-1)\Big) , \end{aligned} \tag{3.7}$$

where F represents the volume fraction, i and j are the cell indices, dx and dy are the cell dimensions in Cartesian coordinates of a uniform mesh.

Once the normal is found, the interface can be reconstructed. In practice, four possible geometric scenarios exist in 2D to fill a cell. As the angle β is bounded up to 90°, the normal of the interface can be rotated depending on the volume fraction. Fig. 3.1 highlights the four possible option for filling a cell with a bounded angle.

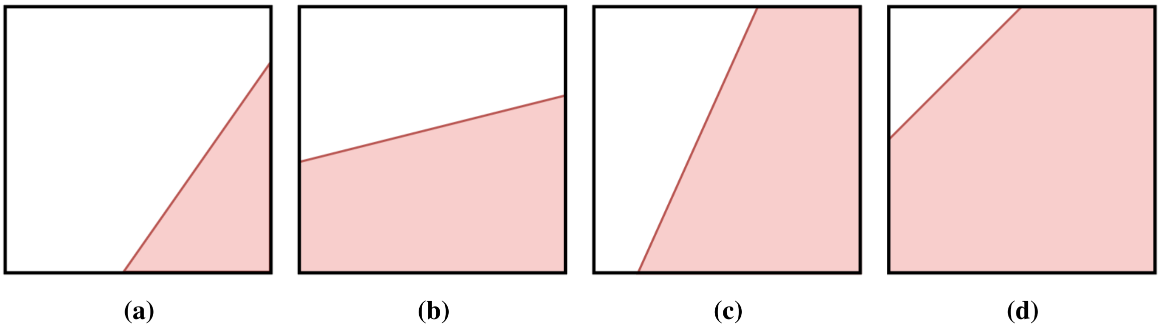


Fig. 3.1 Four possible interface reconstructions.

3.3 The moment-of-fluid (MOF) method: Standard

The moment-of-fluid (MOF) method is a numerical approach for simulating multiphase flows by representing fluid interfaces through moments. The MOF method is able to capture complex flow phenomena with enhanced computational efficiency and versatility (parallelisation, multi-material, filaments).

3.3.1 Problem definition

Let us define the problem posed by the MOF method in order to reconstruct an interface. Consider a convex polygon ω_{cp} that is defined by n vertices, $\mathbf{x}_1$,..., $\mathbf{x}_n$. The area of ω_{cp}, denoted $|\omega_{cp}|$, and the centroid (centre of mass), denoted $\mathbf{x}_c(\omega_{cp})$ can be computed as follows:

$$\left|\omega_{cp}\right| = \frac{1}{2}\sum_{i=1}^{n}\left[\mathbf{x}_i \times \mathbf{x}_{i+1}\right] \tag{3.8}$$

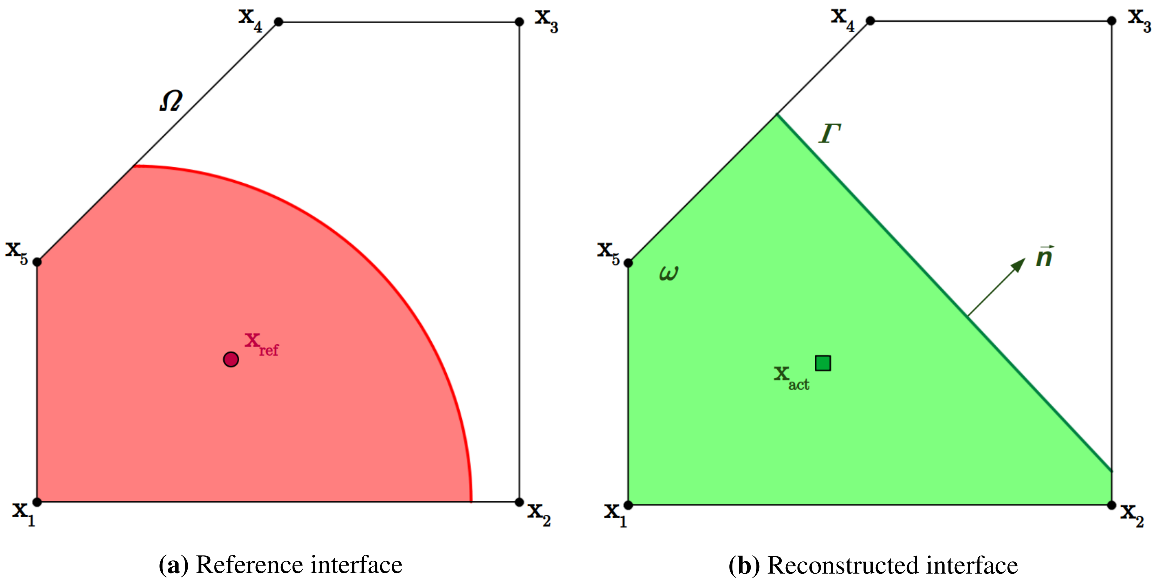
$$\mathbf{x}_c(\omega_{cp}) = \frac{1}{6\left|\omega_{cp}\right|}\sum_{i=1}^{n}\left[\mathbf{x}_i \times \mathbf{x}_{i+1}\right]\left(\mathbf{x}_i + \mathbf{x}_{i+1}\right)\ , \tag{3.9}$$

where [... × ...] is a 2D cross product, resulting in a scalar multiplying the following term, a vector. The sum of them forms the centroid. Alternative expression to find the centroid from the vertices of a polygon exists and can also be generalised in 3D. Note that $\mathbf{x}_{n+1} = \mathbf{x}_1$. Let Ω depict an arbitrary convex cell, hence not restricted to a Cartesian cell, filled with two different materials. Consider the first material μ_1 within Ω acting as a reference interface, its area relative to the area of the cell is denoted by $F_{ref}(\mu_1)$ which corresponds to the volume fraction. Similarly, $\mathbf{x}_{ref}(\mu_1)$ is defined to be the reference centroid of μ_1 within the cell.

The MOF reconstruction problem is formulated as an optimisation problem in which the distance between the reference centroid $\mathbf{x}_{ref}(\mu_1)$ and the centroid of the reconstructed polygon $\mathbf{x}_{act}(\mu_1)$ is minimised while keeping the volume fraction of the reconstructed polygon $F_{act}(\mu_1)$ equal to the volume fraction $F_{ref}(\mu_1)$. One can summarise the optimisation problem as follows:

$$\begin{cases} min\left|\mathbf{x}_{ref}(\mu_1) - \mathbf{x}_{act}(\mu_1)\right| \\ \text{subject to} \\ F_{act}(\mu_1) = F_{ref}(\mu_1) \end{cases} \tag{3.10}$$

As shown in Fig. 3.2, the reference interface may be curved, hence the minimised centroid distance will aim to give the *best* reconstruction. Although the solution of the optimisation problem is unique, local solutions and numerical round-off errors may cause some inconsistencies. If μ_1 already occupies a polygon with a piecewise linear interface, the MOF method aims to reconstruct the exact interface.



(a) Reference interface

(b) Reconstructed interface

Fig. 3.2 Reference vs. reconstructed interface with their respective centroids $\mathbf{x}_{ref}$ and $\mathbf{x}_{act}$ where $\mathbf{n}$ denotes the reconstructed normal to the interface and Γ denotes the length of the interface segment.

3.3.2 Reconstruction

The reconstructed normal to the interface within a polygon can be evaluated analytically but only for rectangular cells (Lemoine et al., 2017). However, for cells of any other geometrical shape, a minimisation algorithm is needed to evaluate the normal to the interface. The unit normal is defined to be $\mathbf{n} = [\cos(\phi_a), \sin(\phi_a)]$ where ϕ_a corresponds to the angle the interface makes with the horizontal. To cover all possible normal vectors, $\phi_a \in [0, 2\pi]$. The minimisation function, also known as the objective function, is recalled $f(\phi_a) = \left|\mathbf{x}_{act}(\phi_a) - \mathbf{x}_{ref}\right|$. In general, $f(\phi_a)$ may have multiple local minima. The first derivative of the objective function for a convex cell, initially given in Dyadechko and Shashkov (2005), is defined by

$$f'(\phi_a) = 2((\mathbf{x}_{act}(\phi_a) - \mathbf{x}_{ref}) \cdot \mathbf{x}'_{act}(\phi_a)) \ , \qquad (3.11)$$

where $\mathbf{x}'_{act}(\phi_a)$ is given by

$$\mathbf{x}'_{act}(\phi_a) = \frac{1}{12} \frac{|\Gamma(\phi_a)|^3}{|\Omega| \, F_{ref}(\mu_1)} [-\sin(\phi_a), \cos(\phi_a)] \ , \qquad (3.12)$$

and is evaluated using the length of the reconstructed interface segment $\Gamma(\phi_a)$.

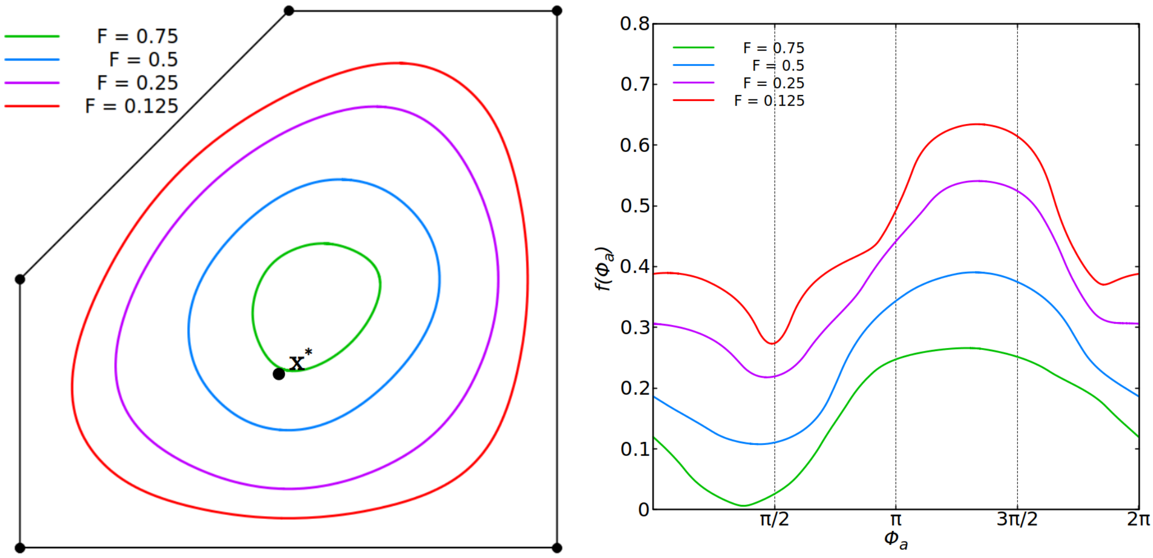
3.3.3 Bisection method

In this section, a new algorithm is presented to evaluate the normal to the interface. The algorithm used in this thesis to find the global minimum is a bisection method. Using four quadrants, explicitly $[0,\pi/2]$, $[\pi/2,\pi]$, $[\pi,3\pi/4]$ and $[3\pi/4,2\pi]$, the zeros of the first derivative of the objective function can be determined. The bisection method uses only a maximum of 10 iterations per quadrant to find the local minimum with a tolerance of 10^{-10}. When the value of the first derivative falls below the specified tolerance at the boundaries of a quadrant, the bisection method is terminated for that quadrant. Once the minimum for each quadrant is found, evaluating the objective function for all valid values will give the global minimum. The global minimum of $f(\phi_a)$ will result in the best approximation for the optimisation problem defined above. Fig. 3.3 shows the set of solutions as well as the objective function within the four quadrants. Knowledge of the normal enables one to flood the cell to reconstruct the interface with the minimum distance between the reference and reconstructed centroid, which is defined as the least centroid error (Lemoine et al., 2017).

This method has the advantage of not requiring any initial condition or fine parameter tuning to converge to the solution and is guaranteed to find the global minimum. However it may require a larger number of iterations to converge.

3.3.4 Advection

Dynamic tests involve advecting materials across multiple time iterations. Information from the previous time step is needed in order to reconstruct the material interface at the next time step. The most natural way to perform this reconstruction is to use a Lagrangian framework. Vertices of cells and polygons are tracked and displaced as particles. A pre-image captures



(a) Potential location of the constrained optimisation

(b) Objective function

Fig. 3.3 Set of solutions and objective function for various volume fractions $F = 0.125, 0.25, 0.5$ and 0.75 where $\mathbf{x}^*$ denotes the reference centroid.

the state of the interface at a given time. This framework uses the Lagrangian pre-image in order to compute the volume fraction and centroid of a material.

All vertices of a cell are advected backwards in time to capture the state of the interface at the previous time step using a 2^{nd}-order Runge-Kutta scheme (RK2) to form the backtrace cell as seen in Fig. 3.4(a) (Iserles, 2009). This is a pragmatic decision which balances accuracy and computational expense, in contrast to solely focusing on varying higher-order schemes. The backtrace cell may intersect several cells at the previous time level. The Sutherland-Hodgman polygon clipping algorithm is used in order to intersect each of these cells to gain information about volume fraction and centroid[1]. The advantage of using the Lagrangian approach is that there is no limitation on the Courant–Friedrichs–Lewy (CFL) number used in the model. Indeed, the CFL number is a dimensionless quantity used in numerical methods to represent the ratio of the physical time it takes for a particle to travel across a grid cell. In general, for explicit schemes, stability is maintained when the CFL number is less than or equal to unity. Moreover, the Lagrangian advection procedure is said

[1] https://rosettacode.org/wiki/Sutherland-Hodgman_polygon_clipping

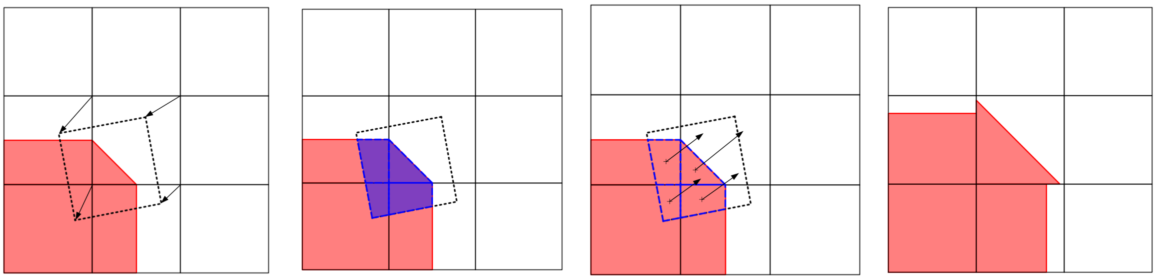
to be unsplit, which means it only requires one advection and reconstruction per cell. This differs from other split schemes (Mukundan et al., 2022).

3.3.4.1 Advection of volume fraction

To compute the volume fraction at the next time step, the sum of intersecting areas forms the new volume fraction of the cell as highlighted in purple in Fig. 3.4(b). However, in some cases, its value may depend on the backtrace cell area relative to the cell area. If the backtrace cell area is larger than the cell area, there is potential for the volume fraction to exceed unity. On the contrary, if the backtrace cell area is smaller than the cell area, there is potential for the volume fraction to be smaller than unity while being entirely filled with one material. These cases may occur when the backtrace cell intersects with only one material, making the new theoretical volume fraction equal to unity but the actual volume fraction is either greater than or less than unity. If this is the case, a post advection remapping procedure is introduced in order to ensure that the total material mass is consistent throughout the advection process. The difference between the actual volume fraction and unity is computed, then redistributed equally across all cells that can accept a gain or loss of mass/volume fraction. This is defined as a global redistribution (Shashkov and Wendroff, 2004). The modified mass in each cell is negligible so that the shape of the interface is not changed significantly, which is demonstrated later when the algorithm is validated. To limit computational expense, this procedure is only performed once per time step, which means there is a risk of not being able to sufficiently redistribute the total mass.

3.3.4.2 Advection of centroid

To compute the centroid at the next time step, the centroid of the intersection of the backtrace cell with a cell is computed, then advected using the same scheme as for the backtrace cell advection as shown in Fig. 3.4(c). All cell intersection centroids are advected forward in time individually. The new reference centroid is obtained by weighting the cell intersection centroids with their volume fraction. Since all centroids are framed within the backtrace cell at the previous time step, the new reference centroid is guaranteed to be within the cell after forward advection. Finally, Fig. 3.4(d) depicts the interface after the advection procedure. It is important to acknowledge that the interface between cells may not be continuous as the interface is resolved individually in each cell. The advection procedure is summarised in Algorithm 2



(a) Lagrangian backtrace **(b)** Volume intersection **(c)** Centroid advection **(d)** Interface reconstructed

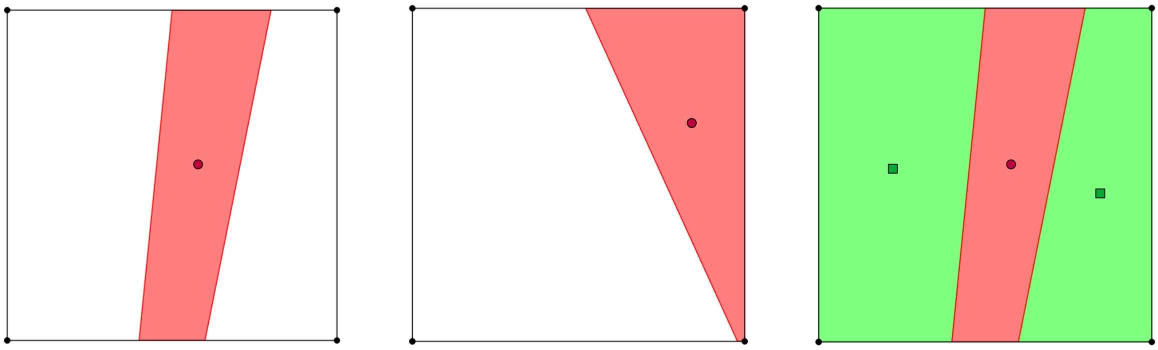
Fig. 3.4 Dynamic test: (a) advection of backtrace cell backwards, (b) intersection of volumes, (c) advection of centroids individually, (d) interface after advection.

Algorithm 2 Advection procedure

```
Before the calculation, identify which cells may contain an interface to reduce computational effort
    Advect vertices of a cell backwards (pre-image)
    Intersect the pre-image with the desired material
    Compute the volume fraction (sum of all polygons area) and centroid (weighted average of all polygons centroids)
    Advect the centroid forward in time
    Reconstruct the interface in a cell
End
Repeat for all cells identified across the domain
```

3.4 The moment-of-fluid (MOF) method: Filament

Filaments are defined as thin strands of material surrounded by another material within a cell. These are structures thinner than a cell size. A standard MOF reconstruction creates a linear interface splitting the cell into two, hence it does not reconstruct the topology of the interface correctly as shown in Fig. 3.5. When considering a filament, two linear interfaces emerge, one on each side of the structure, meaning that two reconstructions are needed to capture the topology perfectly. In filament reconstruction, the conglomeration algorithm is capable of detecting polygons of the same material that are not adjacent by using the numerical adjacency condition. A fictitious material is introduced to reconstruct one of the polygons surrounding the filament. Once reconstructed, the fictitious material is reassigned to its original material. A symmetric multi-material reconstruction is presented to generate a better topology.



(a) Reference interface **(b)** Standard reconstruction **(c)** Filament reconstruction

Fig. 3.5 Schematic diagrams showing (a) a reference interface, (b) the standard MOF reconstruction and (c) filament MOF reconstruction.

3.4.1 Conglomeration

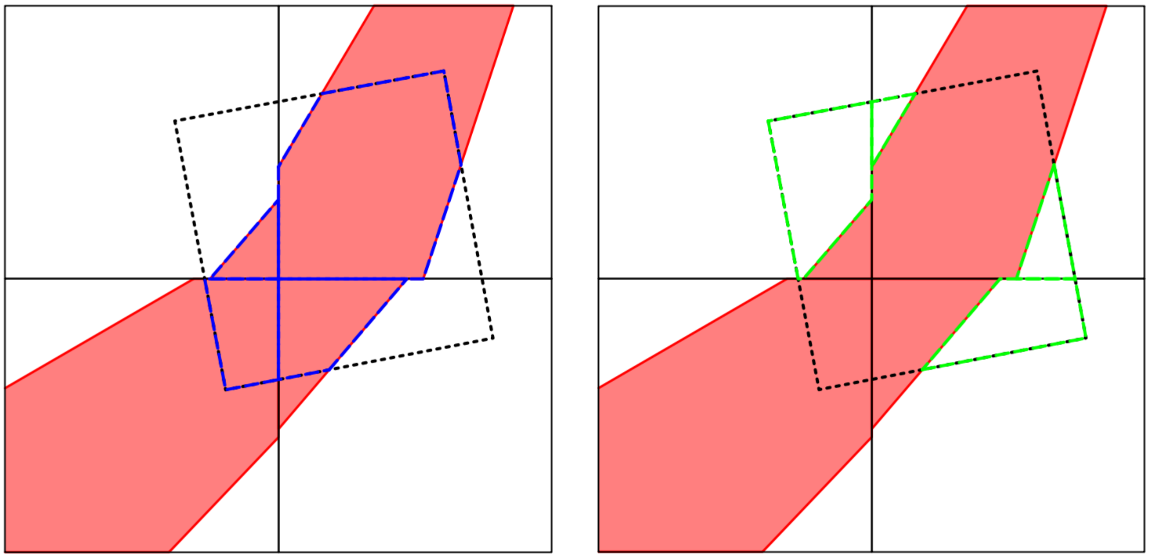
Filament reconstruction is performed when some adjacent polygons forming one material, called a conglomerate, are not adjacent to other conglomerates of the same material. The conglomeration algorithm allows the user to identify whether a cell needs a multi-material reconstruction or a standard reconstruction. It is possible to identify all polygons of one material intersecting with the backtrace cell as shown in Fig. 3.6. Once all of these polygons are identified, the conglomeration algorithm tests if each of these polygons is adjacent to each other. Conglomerates are considered even when they do not split a cell, i.e. being only adjacent to one cell edge. The green conglomerate in Fig. 3.8(b) is one of these. Flotsam are not discussed in this thesis, since in general on a coarse mesh they do not tend to exist. If more than one conglomerate is found, then one of these conglomerates is considered to be the fictitious material. The conglomeration algorithm is a tree-based structure testing adjacency of a list of polygons until the lowest level does not find any adjacent polygons. Algorithm 3 details the procedure to identify conglomerates. The reference volume fraction and reference centroid can easily be computed.

Algorithm 3 Conglomeration algorithm

```
Initialise list of polygons list_poly
while list_poly do
    new_group ← list_poly(1)
    while iter do
        for $k = 1, \mathrm{size}(list\_poly)$ do
            if is_adjacent(new_group(), list_poly(k)) then
                Remove list_poly(k)
                Add list_poly(k) to new_group()
                iter ← true
            end if
        end for
    end while
end while
```

3.4.2 Adjacency

The adjacency test is performed on all sides (segments) of a polygon with respect to another polygon. Some tolerance is accepted as sides may not be perfectly adjacent but can still be considered adjacent. For the purpose of numerical round-off errors, each segment is described by a vector and if the magnitude of the cross-product of two vectors meets the



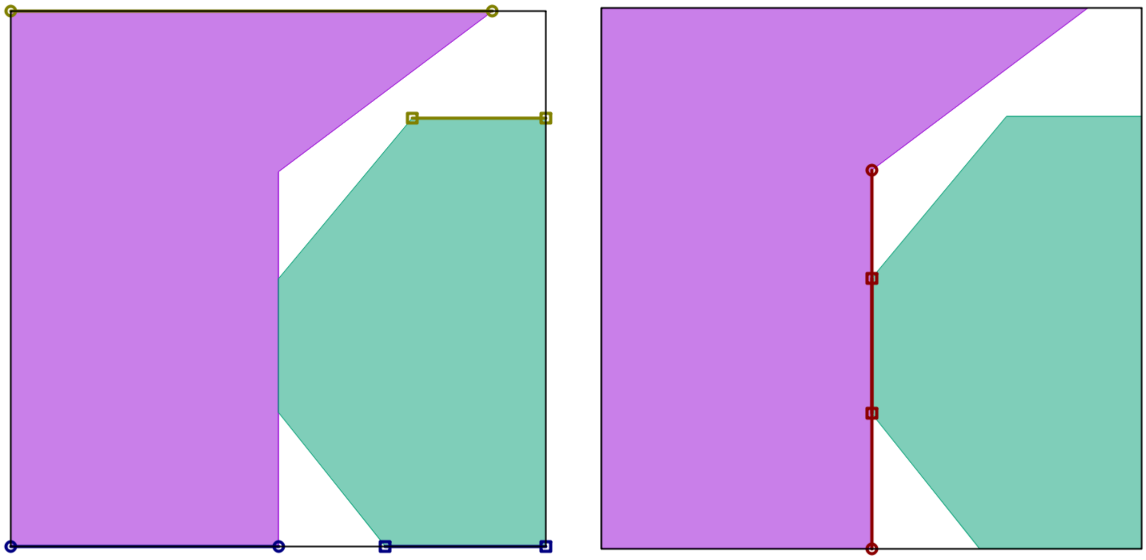
(a) Intersection and conglomeration of material μ_1 **(b)** Intersection and conglomeration of material μ_2

Fig. 3.6 Conglomeration of polygons within the backtrace cell (dashed black outline) leading to the creation of a fictitious material for a 3-material reconstruction. (a) Material μ_1 has 1 conglomerate (outline in blue); (b) Material μ_2 has 2 conglomerates (outline in green).

lower bound of a tolerance, here $\Delta x \Delta y \varepsilon$ with $\varepsilon = 10^{-3}$, then segments are considered parallel. Segments may be considered parallel, yet they also need to be adjacent. Hence, the endpoint of a segment is projected onto the line defined by the other segment. If the distance between the endpoint of the segment and its projection falls below the specified tolerance, here $\Delta x \varepsilon$, the projection of an endpoint also needs to fall between the bounds of the other segment. Both polygons are said to be adjacent and form a conglomerate only if all conditions are satisfied. Algorithm 4 summarises the conditional procedure to test if two polygons are adjacent with three nested conditions. Fig. 3.7 shows two polygons within a cell. Segments are highlighted in order to indicate the process of evaluating parallel and adjacent segments from two distinct polygons. Condition 1 is represented with gold segments. Condition 1 and 2 are represented with blue segments. All three conditions are represented with red segments.

Algorithm 4 Adjacency test

```
Initialise vector_poly1, vector_poly2 based on all vertex_l1, vertex_l2
for l2 = 1, size(vector_poly2) do
    for l1 = 1, size(vector_poly1) do
        % Condition 1
        if abs(cross_product(vector_poly2(l2),vector_poly1(l1))) $\leq \Delta x \cdot \Delta y \cdot \varepsilon$ then
            % Condition 2
            if distance(vertex_l2,projection_on_line(vertex_l2, line(vector_poly1)) $\leq \Delta x \cdot \varepsilon$ then
                % Condition 3
                if projection_on_segment(vertex_l2, segment_l1) $\leftarrow$ true then
                    is_adjacent $\leftarrow$ true
                end if
            end if
        end if
    end for
end for
```



(a) Parallel segments but not adjacent

(b) Adjacent segments

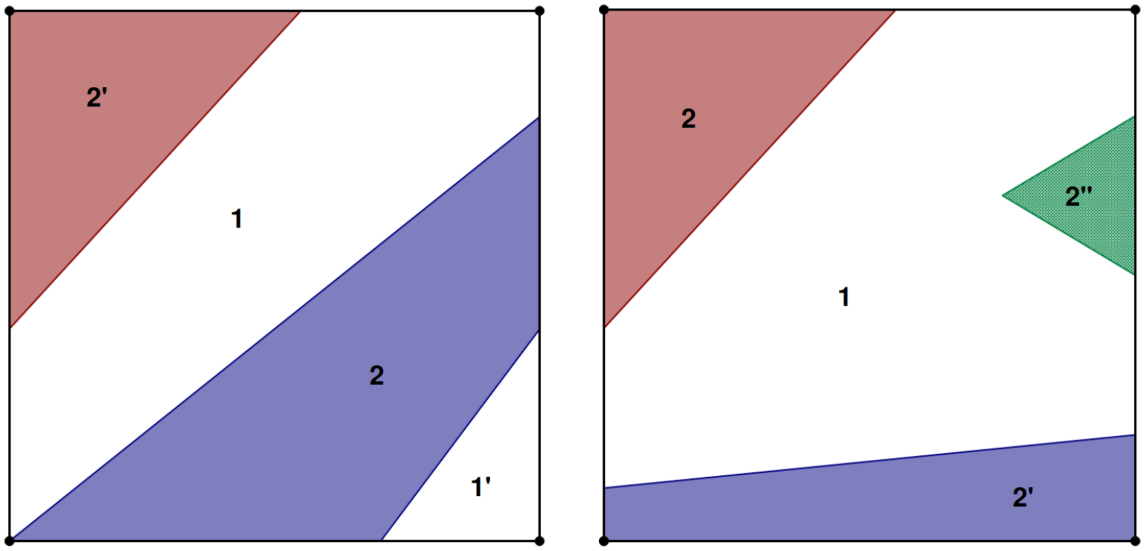
Fig. 3.7 Schematic diagrams to test adjacent segments with another polygon: (a) shows two configurations where segments are parallel. Projection of the endpoints does not fall within tolerance (highlighted in gold). One of the projections of the endpoints does not fall within the other segment (highlighted in blue); (b) shows two segments that are parallel and adjacent, the projection of the endpoints falls within tolerance and within the other segment.

3.4.3 Limitation to three materials

It may happen that more than three conglomerates form within the backtrace cell. In that case, a multi-material reconstruction can be considered. However, it can lead to expensive reconstruction when testing all the combinations for several cells per iteration. For this reason, the number of conglomerates is capped at three in the proposed model. Conglomerates are sorted by volume fraction.

If two conglomerates exist for each material, the following condition is tested. If the second conglomerate of one of the materials has a volume fraction smaller than 10^{-3}, then its volume fraction is added to the main (largest in volume fraction) conglomerate. If there still exists two conglomerates for each material, no conglomerates are considered and a standard reconstruction with the total volume fraction per material is performed. Fig. 3.8(a) highlights this scenario. Indeed, coloured conglomerates belong to Material 2, explicitly 2 and $2'$. Material 1 also has two conglomerates in white, explicitly 1 and $1'$. None of them are smaller than 10^{-3} in volume fraction.

In other cases, conglomerates with the smallest volume are "reattached" to the largest conglomerate of the same material in the cell, usually where one material has one conglomerate and the other has more than two conglomerates. Then, these smaller conglomerates have their volume fraction added to the largest conglomerates. Fig. 3.8(b) highlights this scenario. Three conglomerates (coloured) belong to Material 2, here explicitly 2, $2'$ and $2''$. Conglomerate $2''$ will be reattached to Conglomerate 2, while Conglomerate $2'$ will be considered to be the fictitious material for reconstruction. "Reattaching" to the nearest conglomerates based on the distance between their respective centroids may also be considered but does not affect the topology greatly as the volume fraction for these conglomerates is often very small.



(a) Two conglomerates exist of size larger than 10^{-3} for each material leading to them being reconstructed as a standard MOF

(b) Three conglomerates exist for Material 2 leading to reattachment of the green conglomerate to the red (largest in cell)

Fig. 3.8 Schematic showing two complex examples of sorting multiple conglomerates within the same cell. Let us assume all coloured polygons belong to Material 2.

3.4.4 Symmetric reconstruction of filaments

The reason to cap the number of materials at three is based on computational cost. Reconstructing more than three materials at once has a significantly higher cost than only three materials. Indeed, using the symmetric reconstruction in multi-material MOF schemes, the number of combinations for m materials is reduced from $m!$ order combinations to $m!/2$. Consequently, using four materials instead of three may lead to a computational cost reduction by a factor of four. In addition, using a symmetric reconstruction of filaments may provide a better topology in material reconstruction.

A standard reconstruction aims to reconstruct an interface based only on minimising the centroid error of one material regardless of the other material in cell reconstruction. In some cases, this can lead to a large error in the remaining material centroid. The symmetric reconstruction approach aims to minimise both centroids at the same time. The objective function $f_{sym}(\mathbf{n})$, combining both centroid defects, is given by

$$f_{sym}(\mathbf{n}) = \left|\mathbf{x}_{ref}(\mu_1) - \mathbf{x}_{act}(\mu_1)(\mathbf{n})\right| + \left|\mathbf{x}_{ref}(\mu_{rem}) - \mathbf{x}_{act}(\mu_{rem})(\mathbf{n})\right| \ , \tag{3.13}$$

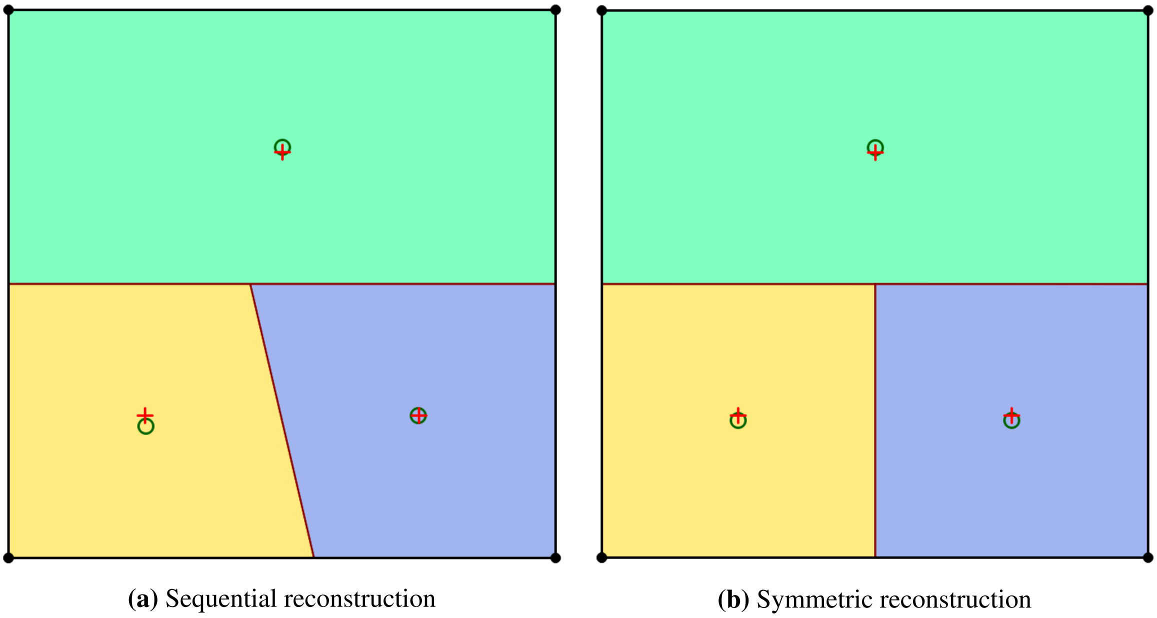
where $x_{ref}(\mu_{rem})$ denotes the reference centroid of the remaining material in the cell, and $x_{act}(\mu_{rem})$ is its reconstruction centroid.

When it comes to filament reconstruction or three material reconstruction, the standard approach is to test all ordering combinations and evaluate the topology that reduces the total centroid defect. This procedure is called a sequential reconstruction. The total centroid defect E can be expressed as the sum of the centroid errors for all materials μ_i.

$$E = \sqrt{\sum_{i}^{m} \left|\mathbf{x}_{ref}(\mu_i) - \mathbf{x}_{act}(\mu_i)\right|^2} \tag{3.14}$$

Consider three materials A, B and C, then six different configurations are possible. Explicitly, and in order of reconstruction, these are (ABC), (ACB), (BAC), (BCA), (CAB) and (CBA). A symmetric reconstruction reduces the number of combinations to only three, thereby reducing the computational effort. Considering the same materials, one of (ABC) and (ACB) combinations would be redundant as the first reconstruction minimises A and the grouping of B and C. Then, (BC) or (CB) will result in the same reconstruction as only symmetric reconstruction is considered. As seen in Fig. 3.9, a symmetric reconstruction provides a *better* topology.

When enabling the filament MOF reconstruction in a simulation, additional steps are required to ensure the advection process is complete. While the standard MOF can be performed only using the desired material as targeted intersection, the filament MOF needs to intersect all materials present in the domain in order to reproduce the exact topology. Moreover, the conglomeration algorithm, which also contains the adjacency test, is performed which enables the volume fraction and centroid of conglomerates to be calculated. Subsequently, the choice of reconstruction between standard MOF and filament MOF can be


(a) Sequential reconstruction

(b) Symmetric reconstruction

Fig. 3.9 Comparison of (a) sequential and (b) symmetric reconstruction when using three materials, where (+) denotes the reference centroids and (o) denotes the reconstructed centroids.

determined. Fig. 3.10 summarises the essential steps involved in the advection process for both standard and filament MOF.

3.5 Coupling MOF-flow solver

3.5.1 Coupling between flow solver and MOF method

While the MOF method presented by Hergibo et al. (2023) uses node velocities during the advection process, in both x- and y-directions, the finite volume Navier-Stokes flow solver uses face velocities (Xie, 2010). To accommodate this, an extra step is needed to interpolate the node velocity before MOF advection can be performed. The face velocity, representing the velocity at the face of each control volume, is known and readily available. However, to compute the node velocity, which characterises the velocity at the corner of each control volume, a simple linear interpolation technique is employed. Note that for each control

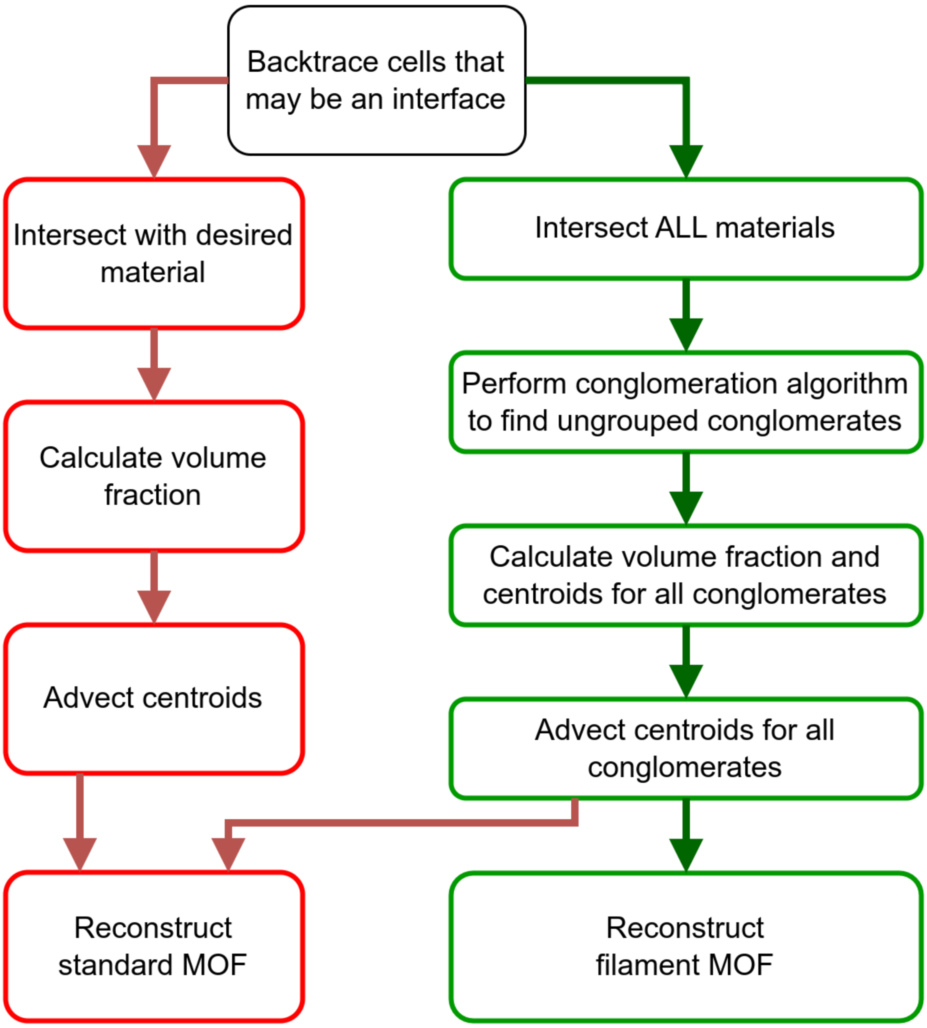

Backtrace cells that may be an interface
Intersect with desired material
Calculate volume fraction
Advect centroids
Reconstruct standard MOF
Intersect ALL materials
Perform conglomeration algorithm to find ungrouped conglomerates
Calculate volume fraction and centroids for all conglomerates
Advect centroids for all conglomerates
Reconstruct filament MOF

Fig. 3.10 Flowchart highlighting key steps to standard and filament MOF reconstruction and advection. Red denotes the use of standard MOF. Green denotes the use of filament MOF.

volume, the east face defines the x-velocity, the north face defines the y-velocity, whereas the bottom left node defines both x- and y-velocities simultaneously (see Fig. 3.11).

The case of a control volume located next to a wall is illustrated in Fig. 3.12. When dealing with boundary conditions, additional adjustments are required to ensure proper treatment of the velocity field near the boundaries. The interpolated node velocities near the boundaries are adjusted to adhere to the prescribed boundary conditions, maintaining consistency with the physics of the problem. For no-slip conditions, the interpolated velocities at the wall are zero. In this instance, the node velocity at the wall in the tangential direction

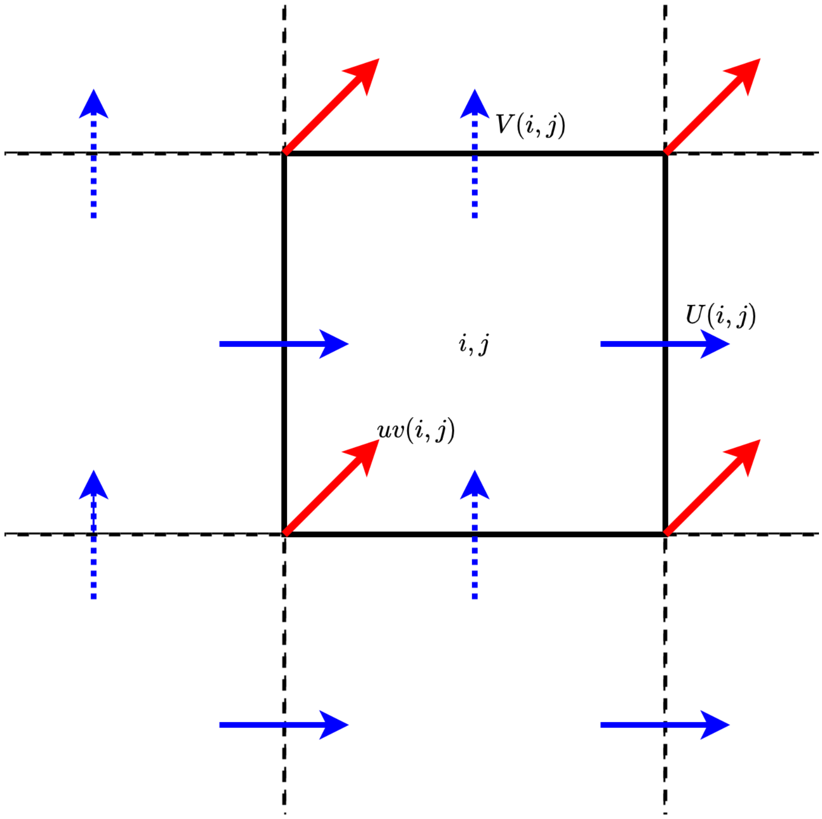


Fig. 3.11 Face velocities are defined at the face of the control volume. $U(i,j)$ denotes the horizontal velocity of cell i,j in full blue arrow. $V(i,j)$ denotes the vertical velocity of cell i,j in dashed blue arrow. Node velocity is interpolated from two neighbouring control volumes (red arrows). $uv(i,j)$ denotes both horizontal and vertical velocities of cell i,j at the node.

is forced to take the value at its nearest node, typically the first node velocity in its normal direction. It is interesting to note that this can be overcome by using a slip condition for the MOF method. Regarding periodic boundary conditions, node velocities remain unaffected by the domain boundaries. Nevertheless, the use of ghost cells presents a counter-intuitive aspect. Regardless of the direction of periodicity, the first row/column of cells corresponds to the last row/column of cells in the domain. In this instance, the row/column of ghost cells on one side corresponds to the penultimate row/column of cells of the domain of the other side and vice versa.

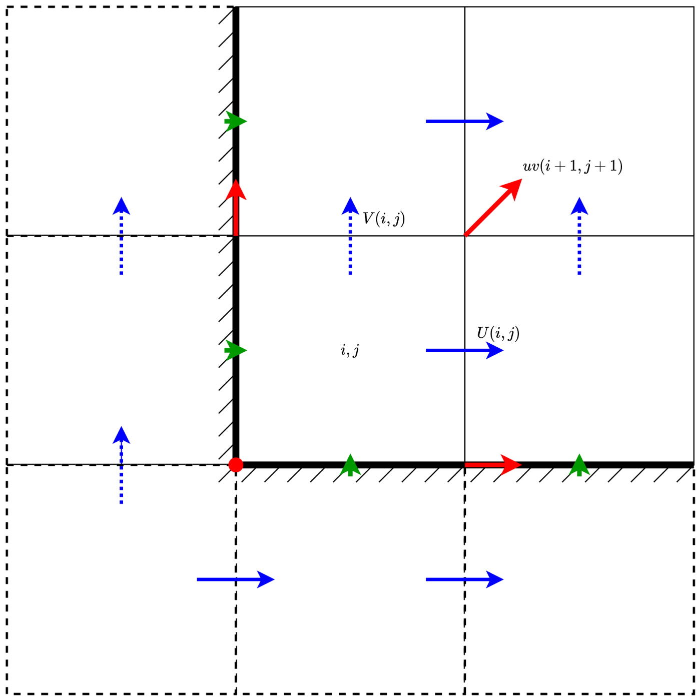


Fig. 3.12 Schematic diagrams showing the wall boundary conditions imposed in the domain for all velocities at the bottom left edge of the domain. Cells in dashed lines are ghost cells. Red arrows & dot represent the node velocities for MOF treatment. Green arrows are imposed boundary conditions for face velocities.

3.5.2 Boundary conditions

To fully specify the mathematical model, it is imperative to establish boundary conditions that play a critical role in defining fluid flow behaviour at the boundaries of a finite computational domain. They accurately capture the interaction between the fluid and its surroundings, classified into inflow, outflow, and solid boundaries. Inflow conditions specify fluid characteristics upon entering the domain, while outflow conditions prevent disturbances caused by fluid leaving the domain. Solid boundaries simulate fluid interactions with solid objects through various approaches. The accurate selection and implementation of boundary conditions is crucial for reliable results in numerical fluid simulations, requiring careful consideration of the specific problem and desired representation of fluid behaviour near the boundaries.

In most MOF methods, the treatment of the boundary condition is rarely discussed as the interfacial topology is inside the computational domain. However, the reduction of the number of neighbouring cells near the boundary means that MOF methods exhibit better accuracy when reconstructing interfaces in these regions. Therefore, special attention needs to be paid to the MOF method in addition to the boundary conditions for the Navier-Stokes solver.

Mass conservation is performed only near the interface. Indeed, in order to reduce computational expenses, only cells near the interface are advected as the CFL constraint guarantees that cells further away from the interface maintain their volume fraction. Therefore, material distribution is performed on "mixed" cells, i.e. cells that contain an interface in a global fashion. Over/under-filled cells are used to redistribute mass uniformly to mixed cells in the domain. The author is aware that this does not guarantee mass conservation to machine precision. However, the choice of time step may influence mass variation during the simulation, which is not discussed in the remainder of this thesis.

3.6 Closing remarks

In conclusion, this chapter has provided a comprehensive exploration of the moment-of-fluid (MOF) method, highlighting its theoretical foundations and computational implementation. Initially, the geometric VOF method has been described briefly in order to provide the framework for the development of the extension to MOF methods. Establishing a linkage between VOF and MOF methods facilitates the comprehension of their strengths and limitations. A fundamental distinction lies in their handling of neighbouring cells for interface reconstruction. The standard MOF and filament MOF methods have been detailed, highlighting their significance and relevance in the context of interface reconstruction. In Fig. 3.13, the key differences between standard and filament MOF methods and the additional steps required in the latter are summarised in a flowchart. Finally, the coupling between the MOF method and a fluid solver is described for a fixed grid.

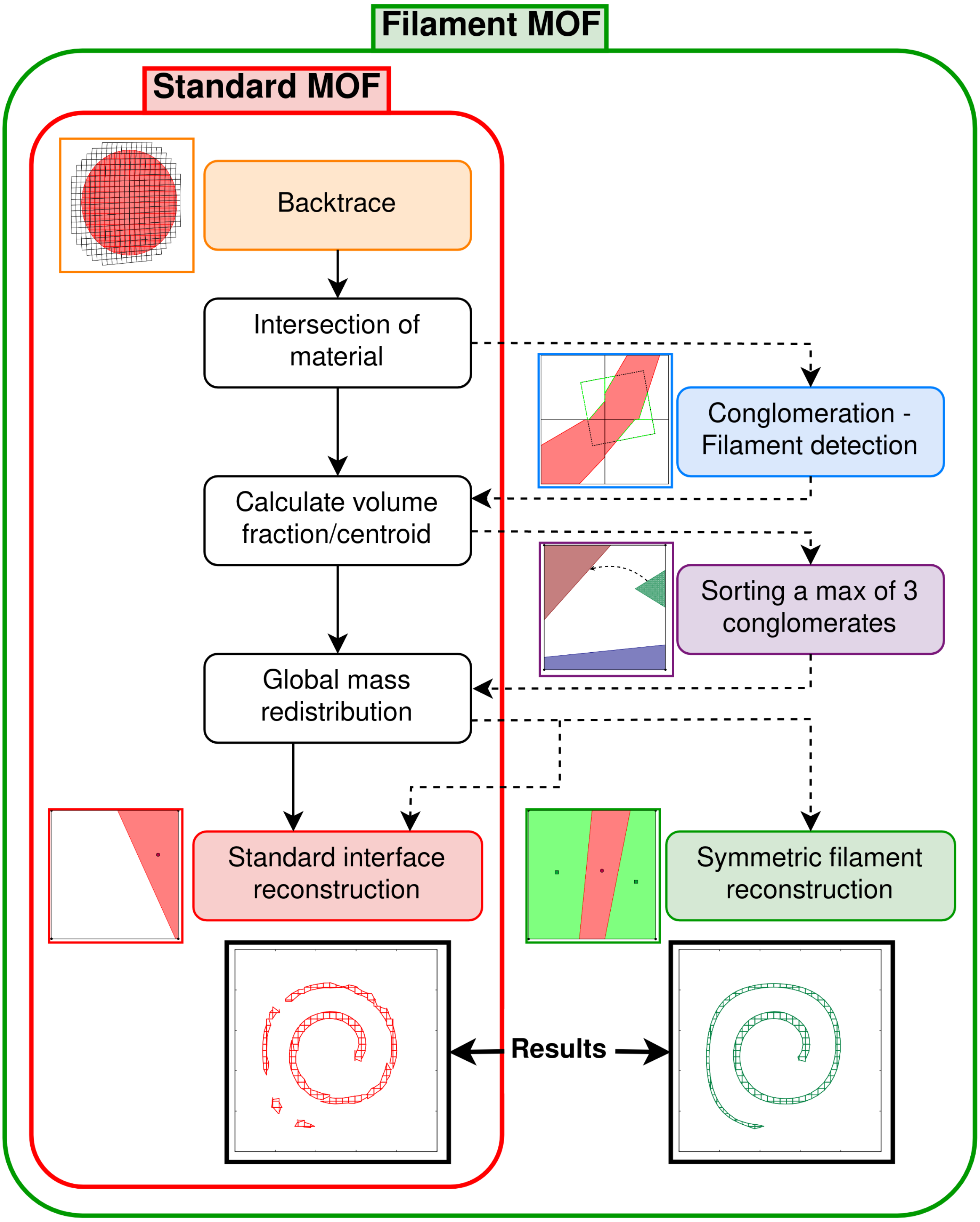

Filament MOF
Standard MOF
Backtrace
Intersection of material
Calculate volume fraction/centroid
Global mass redistribution
Standard interface reconstruction
Conglomeration - Filament detection
Sorting a max of 3 conglomerates
Symmetric filament reconstruction
Results

Fig. 3.13 Flowchart highlighting the key steps for a filament MOF method compared to a standard MOF method.

As the reader is embarked on the next chapter, the focus will shift towards the methodology of dynamic adaptivity in a numerical framework. The integration of adaptive techniques with the MOF method opens avenues for enhancing accuracy and reducing overall computational costs. The forthcoming chapter will explore how these combined methodologies contribute to advancing the state-of-the-art in multiphase flow simulations.

Chapter 4

Adaptive dual grid framework

4.1 Introduction

In the evolution of computational simulations, the accurate representation of interfaces is crucial for understanding the complexities of physical phenomena. The MOF method has shown that is a valuable and significant player in the field of interface capturing methods. Despite having a higher computational cost, the ease of interface representation in complex cases such as filament or multi-material scenarios has demonstrated that it is relevant and efficient.

The integration of adaptivity within the context of the moment-of-fluid (MOF) method signifies a notable advancement in numerical multiphase flow simulations. Adaptivity introduces a dynamic and responsive dimension to the MOF framework, allowing it to adjust intelligently the resolution, refining or coarsening a computational grid based on specific characteristics. This dynamic adaptability enhances the efficiency and accuracy of the MOF method using reduced computational resources. This not only addresses computational challenges associated with varying physical length scales but also contributes to a more effective representation of sharp interfaces.

This chapter explores the intricate details of adaptive mesh refinement (AMR) methods applied in the context of MOF methods. The unconstrained nature of this method is detailed. In addition, satisfaction of mass conservation to machine precision is accomplished. The last part of this chapter deals with the adaptive scheme used in a fluid flow solver. The focal point is the introduction and detailed examination of a dual grid framework, a pioneering approach designed to enhance the accuracy and efficiency of numerical interface calculations.

4.2 Adaptive mesh refinement

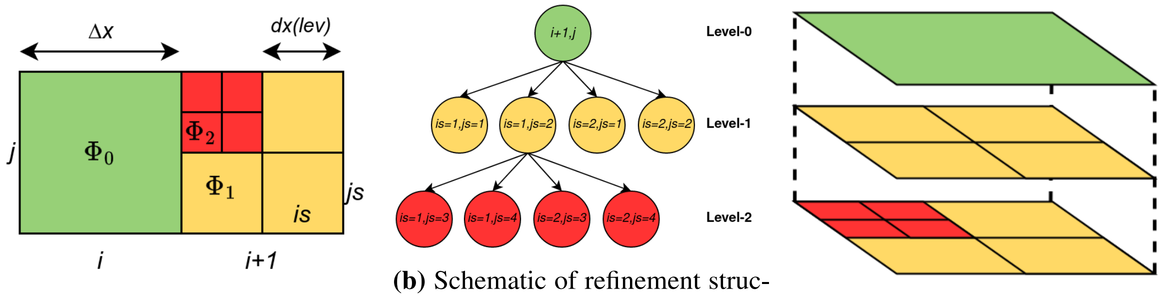
Several techniques have been used over the years to capture interfaces in the context of adaptive mesh refinement (AMR). These may be broadly categorised as either patch-based AMR or quadtree(octree)-based AMR (see Fig. 1.11). Patch-based AMR involves dividing the computational domain into a set of refinement patches. This allows for local control over mesh resolution, and the patches can be refined or coarsened dynamically based on the numerical solution being computed. The main advantage of patch-based AMR is its flexibility. However, it can lead to increased complexity in the maintenance of the grid being created. On the other hand, a quadtree(octree)-based AMR uses a tree-based hierarchical data structure. The quadtree approach is adopted in this thesis and this is described in this chapter.

The main motivation for using AMR is to balance the trade-off between solution accuracy and computational cost. High deformation regions are of interest for AMR in order to produce high-resolution prediction in these complex areas. The process of mesh refinement must be informed by an appropriate criterion and the reconstruction error is used to inform grid adaptation in the current MOF method (Ahn and Shashkov, 2009). Typically, the refinement process involves splitting each cell into 4 subcells in 2D, and 8 subcells in 3D; coarsening involves merging subcells into a larger (sub)cell when a specific condition is met. Generally, the refinement and coarsening processes are repeated until either a desired accuracy is achieved or a certain level of refinement is reached.

4.2.1 Data structure

This work adopts the quadtree-based AMR and the data structure is designed to store and manipulate the hierarchy of meshes with ease and efficiency, as well as allowing communication between levels (Liang, 2012). The quadtree data structure forms a tree where the root node represents the coarsest mesh (base mesh), and each additional level of refinement creates four children nodes to their parent node. This work introduces a simplified approach in which the data structure replicates a quadtree algorithm up to two levels of refinement such that children cells can be accessed from a parent cell at every level up to two. The other advantage is that no subroutines are needed to find or access neighbours. In addition, unlike many other AMR codes, the new approach does not constrain the refinement level of neighbouring cells or subcells.

An arbitrary cell on the quadtree mesh generated using the new approach is indexed as (i,j,is,js) where (i,j) represents the base mesh indices and $is = 1,...,M_s$ and $js = 1,...,M_s$ are the subcells indices, with $M_s = 2^{lev}$ and lev denoting the level of refinement starting at 0 for the base mesh. Subsequently, the size of the new subcells is defined by $dx(lev) = \Delta x/2^{lev}$ and $dy(lev) = \Delta y/2^{lev}$ with $lev = 0,1$ or 2. Naturally, at level 0, $dx(0) = \Delta x$. In addition, the cell-centre coordinates can be computed directly using the following relationships $x_c = x(i) + (is - \frac{1}{2}) \cdot dx(lev)$ in the x-direction and $y_c = y(j) + (js - \frac{1}{2}) \cdot dy(lev)$ in the y-direction, in which $x(i)$ and $y(j)$ are the bottom left coordinates of a level 0 cell. The cell area $|\Omega|(lev)$ is then defined by $|\Omega|(lev) = dx(lev)dy(lev)$, and for a uniform mesh, the cell area is simply defined as $|\Omega|$. Accessing children cells uses the logic from a parent cell's index parity is. Children subcell indices can be called using $(2is-1, 2is)$ when the parent index is is even, and $(is, is+1)$ when is is odd. This logic is valid up to level 2 and works in both horizontal and vertical directions. The data structure and index system are shown in Fig. 4.1.



(a) Quadtree structure grid

(b) Schematic of refinement structure using indices

(c) Schematic of level structure

Fig. 4.1 General idea of data structure used in an AMR framework.

In order to allow for the manipulation of variables, an additional index representing the level of refinement is used in the data structure. Because the data structure may contain several variables accounting for volume fraction, centroid or polygon representation at different levels of refinement, a specific variable is used in the code for identifying which level of refinement is reached. The logical variable *last_lev_refinement(i,j,is,js,lev)* allows one to enable or disable any values of unused level of refinement. A true value means that subcell (i, j, is, js) at level lev is the last refinement and contains a valid volume fraction to be intersected. A false value shows that the subcell (i, j, is, js) at level lev is not the last level of refinement and values are ignored. In general, when a higher level of refinement is triggered, the logical value of *last_lev_refinement* of the corresponding parent subcell at a lower level is set to false. The value of this logical variable is set to true for all children subcells. Mathematically, the set Φ_{lev}, including subsets Φ_0, Φ_1 and Φ_2, respectively, represents all cells at their finest refinement i.e. $\Phi_{lev} = \Phi_0 \cup \Phi_1 \cup \Phi_2$ with $\Phi_i \cap \Phi_j = \emptyset$ for $i \neq j$. This means, with reference to the colour scheme of Fig. 4.1, Φ_0 corresponds to green cells, Φ_1 to yellow subcells and Φ_2 to red subcells. Algorithm 5 details how to loop and access any variables in the code.

Algorithm 5 AMR data structure

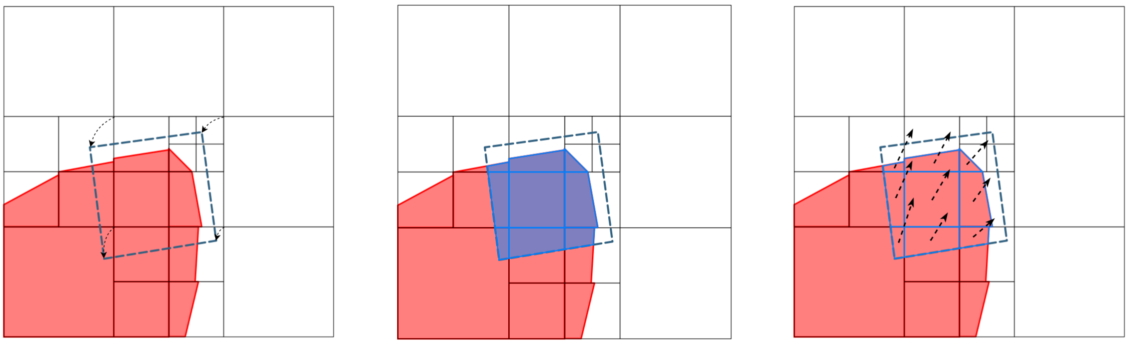
```
for lev = 0,2 do
  for i = 1, N_cell_x do
    for j = 1, N_cell_y do
      $M_s = 2^{lev}; dx(lev) = \Delta x/2^{lev}; dy(lev) = \Delta y/2^{lev}$
      for is = 1, Ms do
        for js = 1, Ms do
          $\%EXAMPLE$ : $accessing\ the\ volume\ fraction\ of\ a\ subcell$
          volume_fraction(i,j,is,js,lev)
          $\%EXAMPLE$ : $checking\ the\ last\ level\ of\ refinement\ of\ a\ subcell$
          last_lev_refinement(i,j,is,js,lev) $\leftarrow$ true
        end for
      end for
    end for
  end for
end for
```

4.2.2 Refinement criterion

In previous VOF or level set methods, refinement was triggered when the volume fraction or level set function is in a certain range or when the estimated curvature gradient reaches a certain value. In the adopted MOF method, the centroid error is used as the criterion for refinement. Indeed, when a standard MOF or a filament MOF procedure is used in a cell, the reconstruction error is a good indicator of the accuracy of the reconstruction. Eq. (3.14) includes material centroid error and fictitious material in a filament case. In this thesis, the refinement criterion is dependent on the cell size. Specifically, it is chosen to be $10^{-9}dx(lev)$, which is lower than that used by Jemison et al. (2015) and not set to a fixed tolerance, which is different from criteria used in previous MOF-AMR schemes (Ahn and Shashkov, 2009).

4.2.3 Advection procedure on a refined mesh

Similar to advection on a uniform mesh, the advection on a refined mesh entails the use of a Lagrangian pre-image (see Section 3.3.4). This requires intersecting the material at the previous time step without omitting the different levels of refinement. All levels of refinement need to be intersected. As per the uniform approach, the backtrace cell is advected backwards using RK2, and the area intersected in this pre-image relative to the subcell area corresponds to the volume fraction of the refined subcell. Eventually, the centroids of all polygons intersected forming the volume fraction are advected forward using the same scheme and the weighted average will define the new reference centroid. The respective reference volume fraction and centroid are used for reconstruction. Fig. 4.2 illustrates the advection procedure on a refined mesh, with particular emphasis placed on intersecting each subcell within the backtrace cell and advecting the centroid of all intersected subcells.



(a) Lagrangian backtracking pre-image

(b) Intersection between backtrace and material at all levels

(c) Forward advection of individual centroids

Fig. 4.2 Schematic showing advection of moments in an AMR framework.

Our approach differs from the AMR scheme of Ahn and Shashkov (2009) since filament MOF is enabled here. The ability to capture filaments has significant advantages over a standard MOF method and has been shown to generate high accuracy on a uniform mesh (Hergibo et al., 2023). The proposed work also differs from the work of Jemison et al. (2015) as filaments may be captured without the need to refine cells. For the sake of capturing filaments, the base mesh is used as the backtracking level, meaning that level 0 is advected first at all time. To determine which cells need advecting, neighbouring volume fractions are used at level 0 to evaluate its potential of containing an interface, similar to a uniform advection.

In the case of a cell being refined, the backtrace at a defined level of refinement is performed as follows. The central point common to all child subcells is advected using the usual RK2 subroutine. All other vertices are interpolated from the backtrace at level 0. This will guarantee exact material intersection with the level of refinement below, hence exact mass conservation. This is performed in a similar fashion for level 2.

(i) Backtrace the four vertices from a level 0 cell using RK2.

(ii) Advect the common node using RK2.

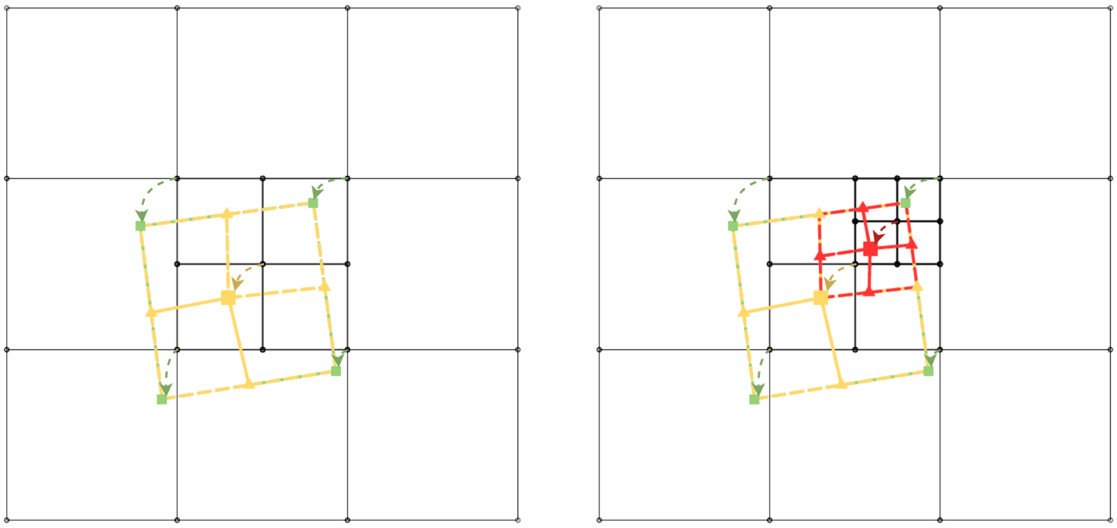
(iii) Interpolate the four mid-points from the level 0 backtrace cell

(iv) Create four new subcells

Fig. 4.3 shows the procedure in place for backtracking a subcell in this refinement framework. This ensures the intersection with a refined backtrace cell with the interface. This approach differs from the one introduced by Ahn and Shashkov (2009) where a simplified backtracking approach is used. As stated above, the proposed approach does not create gaps and overlaps, and therefore a simpler mass redistribution procedure can be implemented.

4.2.4 Time step on a refined mesh

In numerical simulations, the typical time step is determined according to the CFL condition. In this thesis, the CFL number is chosen to be unity unless stated otherwise. The Lagrangian approach enables an unrestricted choice of the CFL number (Hergibo et al., 2023). The Lagrangian approach provides a significant benefit enabling mesh refinement without introducing inconsistencies in time steps. Specifically, the time step is chosen with respect to the base mesh. When refining a mesh locally, the time step used for a refined cell is the same as the one adopted for the base mesh, and so the CFL number is 2 for level 1 and 4 for level 2 cells. Without this approach, the integration of AMR schemes with MOF methods could lead



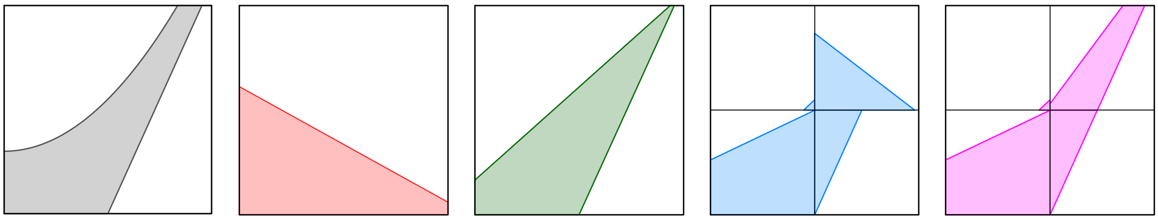
(a) Backtracking a subcell at level 1

(b) Backtracking a subcell at level 2

Fig. 4.3 Schematic showing backtracking of moments in an AMR framework. (□) symbols refer to vertices advected using RK2. (△) symbols refer to mid-point vertices being interpolated. Colour scheme shows green vertices for level 0; Yellow vertices and dashed lines for level 1 vertices and subcells; Level 2 vertices and subcells are in red.

to CFL issues, potentially due to instabilities or inaccuracies. Alternative time step strategies such as adaptive time stepping can be employed, but for simplicity these were not adopted here as no instability issues were encountered using the present approach.

In Fig. 4.4, the comprehensive integration of the concept of adaptive refinement is emphasised. A closer examination of the refinement strategy reveals a structure, incorporating one distinct level of refinement, in which one subcell contains a filament reconstruction. The combination of refinement and filament schemes ensures a precise representation of complex flow phenomena across varying levels of mesh resolution in order to enhance the accuracy and fidelity of the numerical results.



(a) Reference interface **(b)** Standard MOF **(c)** Filament MOF **(d)** Standard MOF-AMR **(e)** Filament MOF-AMR

Fig. 4.4 Example of complex reconstruction using the MOF-AMR method in which the aim is to reconstruct a curved filamentary interface: (a) reference interface, (b) fixed grid reconstruction using standard MOF which highlights the limitation of this method, (c) fixed grid reconstruction using a filament MOF in which the linear interface is reconstructed with a certain error, (d) reconstruction using standard MOF with one level of refinement which still shows the limitations of standard reconstruction, (e) reconstruction using a combination of filament MOF and MOF-AMR which is able to reconstruct a linear interface while mitigating the error of the curved interface.

4.3 Mass conservation during advection

4.3.1 Uniform global mass redistribution

Mass conservation is difficult to enforce in a grid refinement procedure. Local redistribution in a refinement step can lead to a large deformation of the interface, hence global redistribution is used in this thesis. When using a global redistribution approach, over/under-filled cells are considered. These cells are formed by only one material, however their area intersected leads to a volume fraction being either less or greater than unity. These cells have their volume fraction set to unity and the difference to unity multiplied by the cell area $|\Omega|\,(lev)$ is added to a global variable. Let us call the global redistribution variable δ. In a uniform approach, δ is redistributed to N mixed cells, i.e. cells with an interface. In fact, a $\delta/(|\Omega|N)$ amount is redistributed to mixed cells. In the case that not all of the mass is redistributed, an iterative procedure is enacted to ensure all mass is redistributed. Indeed, lack of mass redistribution can penalise mass conservation at other levels.

At other levels of refinement, over/under-filled cells may also occur in the intersection process, meaning that mass needs to be redistributed at all levels. However, on a refined mesh, mass has been redistributed at a lower level with cells needing refinement. Therefore, another local variable *"distributed"* is considered in the redistribution process corresponding to the mass redistributed in each cell/subcell at a lower level. It allows the user to keep track of redistribution to cells that may trigger refinement, with the sum of them all being δ_{low_lev}. Indeed, mass may be redistributed to cells that will be refined, hence that amount needs to be shared at the next refinement level. All cells needing refinement have their mass redistributed at a lower level added to the δ of the refinement level. Then, the new amount of mass redistributed is $\delta + \delta_{low\_lev}$. Therefore, for each cell, the new volume fraction F_i is calculated using

$$F_i \leftarrow F_i + \frac{(\delta + \delta_{low\_lev})}{|\Omega|\,(lev)N_{mix}} \ , \tag{4.1}$$

where N_{mix} corresponds to the number of mixed cells in the domain at a certain level. The redistribution procedure is implemented as detailed in Algorithm 6. In this subroutine, the amount of mass that is not repaired *"not_repaired"* is taken into account because some "almost" full/empty cells may not be able to receive/give their contribution. In these instances, the redistribution subroutine is repeated until the amount of mass is close to machine precision, i.e. 10^{-15}. The amount redistributed is kept in the variable *"distributed"*.

Several approaches can be used when redistributing the mass globally. Two of them are presented in this section, these are termed the directly proportional and inversely proportional distribution approaches.

4.3.2 Directly proportional global mass redistribution

Amongst redistribution procedures, the directly proportional redistribution approach seems intuitive. The redistribution occurs in a similar fashion as the uniform case. However, the mass is redistributed proportional to the volume fraction in a cell. Hence, the total volume

Algorithm 6 Redistribution in a refined mesh

```
**Initialise** $\delta$, $\delta_{low\_lev}$
$repair \leftarrow \delta + \delta_{low\_lev}$
$not\_repaired \leftarrow repair$
% $Note\ \ \delta_{low\_lev} = 0$ at level 0
**while** $(not\_repaired > 10^{-13})$ **do**
    **if** $(mixed\_cell \leftarrow$ **true .AND.** $repair < 0)$ **then**
        $\%\ REPAIR\ IS\ NEGATIVE\ \ - REMOVE\ VOLUME\ FRACTION$
        **if** $(volfrac(i,j,is,js,lev) + repair/(|\Omega|\,(lev)N\_mix) < 0)$ **then**
            $\%\ CELL\ VOLUME\ FRACTION\ CANNOT\ BE\ NEGATIVE\ \ - RESET\ TO\ 0$
            $not\_repaired = not\_repaired + volfrac(i,j,is,js,lev) * (|\Omega|\,(lev)N\_mix)$
        **else**
            $volfrac(i,j,is,js,lev) = volfrac(i,j,is,js,lev) + repair/(((|\Omega|\,(lev)N\_mix)$
            $distributed(i,j,is,js,lev) = distributed(i,j,is,js,lev) + repair/N\_mix$
            $not\_repaired = not\_repaired - repair/N\_mix$
        **end if**
    **else if** $(mixed\_cell \leftarrow$ **true .AND.** $repair > 0)$ **then**
        $\%\ REPAIR\ IS\ POSITIVE\ \ - ADD\ VOLUME\ FRACTION$
        **if** $(volfrac(i,j,is,js,lev) + repair/(|\Omega|\,(lev)N\_mix) > 1)$ **then**
            $\%\ VOLUME\ FRACTION\ CANNOT\ MORE\ THAN\ UNITY\ \ - RESET\ TO\ 1$
            $not\_repaired = not\_repaired + (1 - volfrac(i,j,is,js,lev)) * (|\Omega|\,(lev)N\_mix)$
        **else**
            $volfrac(i,j,is,js,lev) = volfrac(i,j,is,js,lev) + repair/(((|\Omega|\,(lev)N\_mix)$
            $distributed(i,j,is,js,lev) = distributed(i,j,is,js,lev) + repair/N\_mix$
            $not\_repaired = not\_repaired - repair/N\_mix$
        **end if**
    **end if**
**end while**
$\%\ WHEN\ REDISTRIBUTION\ IS\ TRIGGERED\ AT\ NEW\ REFINEMENT\ LEVEL$
$\delta_{low\_lev} = \sum distributed(i,j,is,js,lev)$ **if**$(last\_lev\_refinement(i,j,is,js,lev) ==$ **false**$)$
```

fraction of all mixed cells is calculated. The repair is then performed using the redistribution process in which the new volume fraction is given by:

$$F_i \leftarrow F_i + \frac{(\delta + \delta_{low\_lev})}{|\Omega|\,(lev)} \frac{F_i}{\sum_{j=1}^{N_{mix}} F_j} \tag{4.2}$$

This approach may alter the shape of the interface to a lesser extent. This approach is discussed later in this thesis.

4.3.3 Inversely proportional global mass redistribution

Opposite to the previous concept, mass is redistributed inversely proportional to its volume fraction in this approach. Conceptually, a proportional approach may lead to several iterations of redistribution because a large mass is redistributed to an "almost" full/empty cell, and therefore the mass that is not repaired may be large. By using the inversely proportional approach, more mass is redistributed to those almost empty cells which intuitively would reduce the number of redistribution iterations, but may alter the shape of the interface more, as shown later in this thesis. The repair is redistributed as follows:

$$F_i \leftarrow F_i + \frac{(\delta + \delta_{low\_lev})}{|\Omega|\,(lev)} \frac{(1 - F_i)}{\sum_{j=1}^{N_{mix}} (1 - F_j)} \tag{4.3}$$

Fig. 4.5 summarises the major components of the procedure used in the MOF-AMR method framework. This process involves the advection of "mixed" cells and their intersection with the relevant materials within the backtrace cell. Mass is redistributed across the interface. Eventually, interfaces are reconstructed using either the standard or filament MOF method. Refinement is triggered if the error in reconstruction is unsatisfactory. Before looping back and adding a level of refinement, the data structure is updated for both children and parent cells.

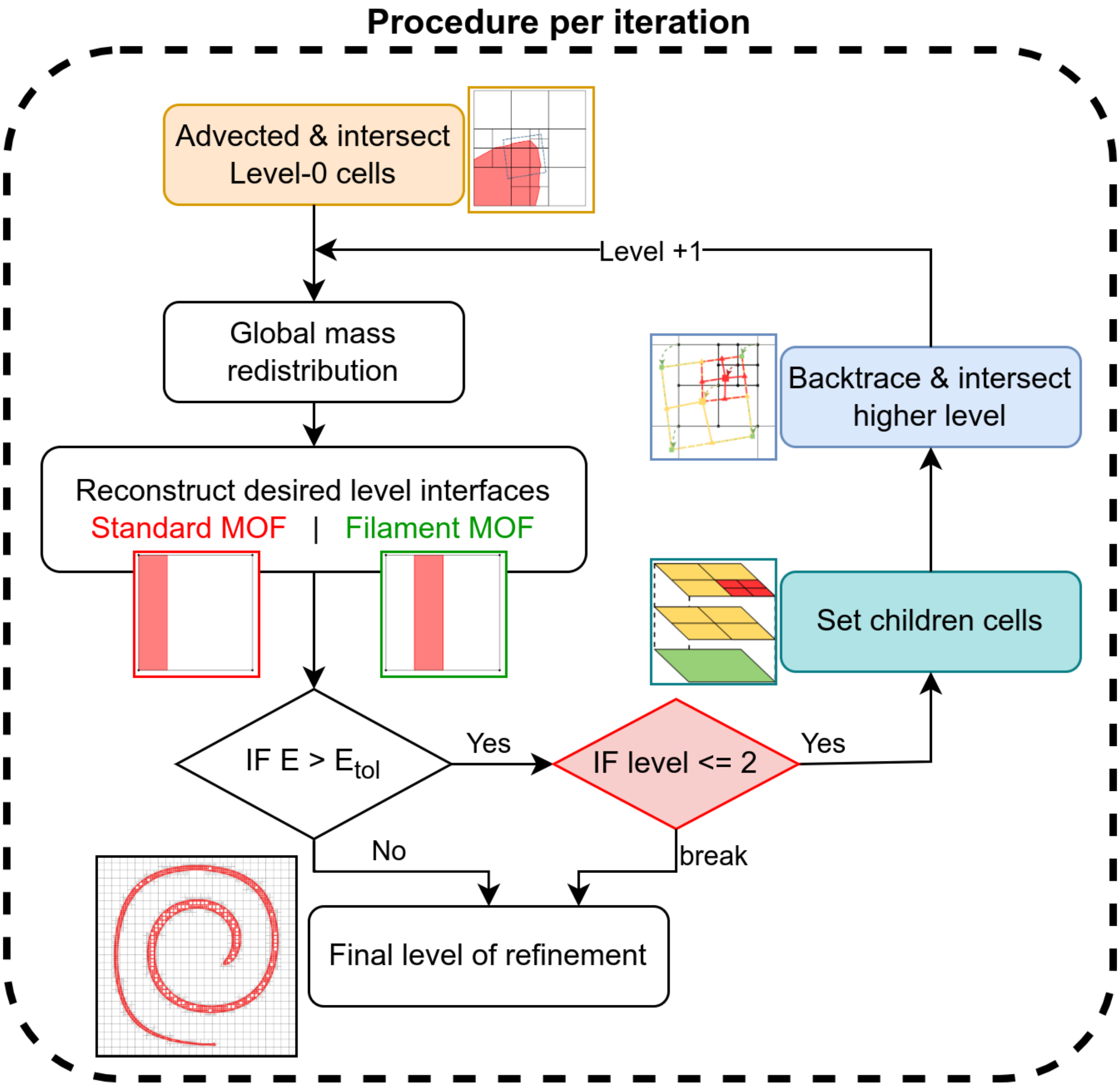
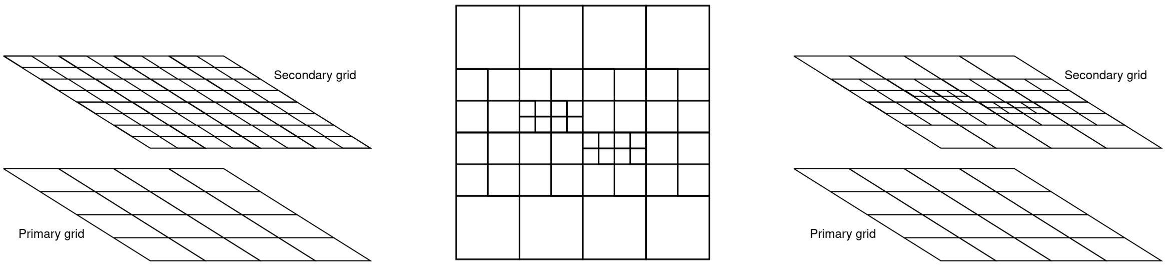

Procedure per iteration
Advected & intersect Level-0 cells
Level +1
Global mass redistribution
Backtrace & intersect higher level
Reconstruct desired level interfaces
Standard MOF | Filament MOF
Set children cells
IF E > E$_{tol}$
Yes
IF level <= 2
Yes
No
break
Final level of refinement

Fig. 4.5 Flowchart highlighting the key steps in MOF-AMR reconstruction.

4.4 Adaptive dual grid

Adaptive dual grid is an innovative approach in the field of fluid dynamics, designed to address the computational challenges associated with complex and evolving flow phenomena. This method uses the power of adaptivity to dynamically refine grids in response to changes in local flow characteristics, enabling higher resolution in regions of interest. By intelligently allocating computational resources on two distinct grids, the adaptive dual grid technique achieves a balance between accuracy and efficiency. One grid solves the velocity-pressure coupling on a fixed grid ensuring stability and consistency in the flow characteristics. Simultaneously, the second grid resolves the evolving interface by dynamically adjusting its resolution to effectively capture detailed features of the interface.

The dynamic allocation of computational resources allows for a comprehensive representation of interface phenomena, offering significant advantages in simulating a wide range of fluid dynamics problems over fixed dual grids or a single adaptive grid. Fig. 4.6 shows the configuration for the three different method, namely the fixed dual grid, the single adaptive grid and the adaptive dual grid. The selection of an optimal approach entails a consideration of various factors such as flexibility, efficiency and complexity. The fixed dual grid approach provides stability and accuracy in the solution, but lacks the flexible grid adjustment which makes it less efficient, and computationally more expensive (Fig. 4.6(a)). Similarly, although the single AMR grid approach offers adaptivity, it may encounter challenges in maintaining stability and consistency in flow calculations across diverse levels of grid resolution (Fig. 4.6(b)). Computationally, the grid may be refined due to unresolved features on the interface while having converge velocity-pressure. In contrast, the adaptive dual grid method emerges as a balance between adaptivity, stability and efficiency thanks to its flexibility at targeting regions of interest while maintaining low computational resources with efficiency and accuracy (Fig. 4.6(c)).



(a) Fixed dual grid **(b)** Single adaptive grid **(c)** Adaptive dual grid

Fig. 4.6 Schematic highlighting different refinement scenarios when coupling a flow solver to an interface capturing method.

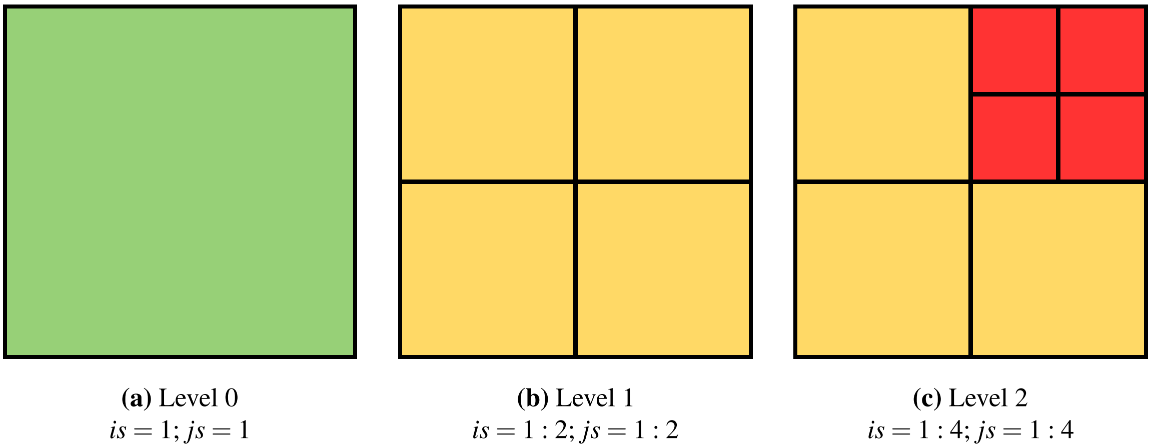
The aim of this work involves the use of adaptive dual grid techniques to maintain sharp and precise interfaces whilst using a coarse and fixed primary grid to keep a low computational cost. The secondary grid involves interfacial features, enabling adaptivity and subgrid structures in the interface reconstruction.

4.4.1 Adaptive scheme

The unconstrained quadtree adaptive mesh refinement method presented by Ahn and Shashkov (2009) is a sophisticated numerical technique employed to enhance accuracy and computational efficiency. A hierarchical data structure, referred to as a quadtree, is used where the domain is repeatedly divided into subcells based on a user-defined criterion. Unlike constrained methods, which follow the 2:1 refinement rule, the unconstrained approach allows for irregular grid adaptations, making it particularly well-suited for capturing complex flow features efficiently (see Fig. 1.12).

In the novel yet simplified quadtree mesh approach of Hergibo et al. (2024), an arbitrary cell is denoted by (i, j, is, js), where (i, j) signifies the base mesh coordinates. Subsequently, the subcell indices is and js are related to the desired level, lev, of refinement. To access children cells, a logic based on the parent cell's index parity is employed, where children subcell indices are given by $(2is-1, 2is)$ when the parent index is even and $(is, is+1)$ when the parent index is odd. This logical scheme is applicable up to level 2 and is applicable in both the horizontal and vertical directions. Fig. 4.7 highlights the refinement structure and indexing of the hierarchy. In this example, the green cells are level 0 (L0), the yellow subcells are level 1 (L1) and the red subcells are level 2 (L2). This example clearly demonstrates that the red subcells indices are $(3:4, 3:4)$.

The triggering of refinement in the context of the MOF method employed here uses the centroid error as the primary criterion. This approach considers the reconstruction error as an effective gauge of the accuracy of the reconstruction, which encompasses the material centroid error, including the fictitious material in filament scenarios. Notably, the refinement criterion in this study is normalised by the cell size.



(a) Level 0
$is = 1; js = 1$

(b) Level 1
$is = 1:2; js = 1:2$

(c) Level 2
$is = 1:4; js = 1:4$

Fig. 4.7 Structure and indexing of the novel quadtree approach. Green cell corresponds to level 0 (L0), yellow cells to level 1 (L1) and red cells to level 2 (L2).

4.4.2 Dual grid method

The dual grid method involves the addition of a secondary grid alongside the primary grid, which allows for a refined resolution of complex flow features. Conserving computational resources is critical in the context of MOF methods since the optimisation component of the method can be expensive. By adapting grid resolution locally, the adaptive dual grid method aims to enhance the precision and fidelity of numerical simulations in accurately predicting fluid dynamics.

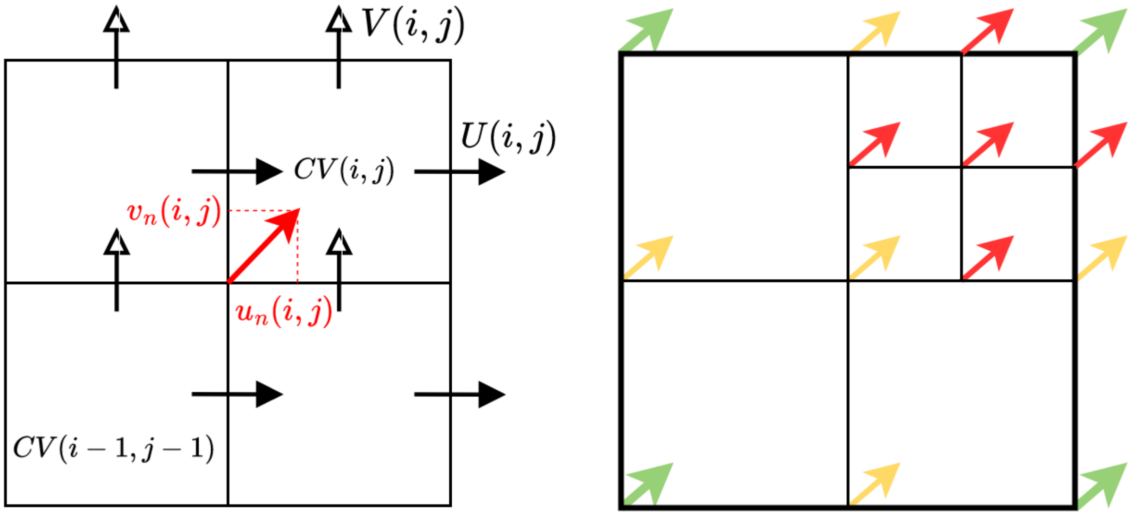
Due to the unconstrained nature of the proposed novel AMR structure, calculating fluxes through different levels of refinement on a single AMR grid may be necessary for velocity, pressure and volume fraction variables. Therefore, the dual grid aspect comes naturally. Indeed in numerical multiphase flow simulations, the velocity-pressure coupling and volume fraction are solved independently. The adaptive dual grid method stems from the difference in resolution needed for accurate interface representation. Reasons to use an adaptive dual grid approach rather than a single AMR grid include facilitating parallel computation, limiting the complexity of the flux calculations, utilising the convergence rates for the different components of the numerical scheme and the unconstrained nature of the this approach.

The velocity-pressure coupling on a fixed coarse grid facilitates the overall understanding of the flow. Concurrently, the interface dynamics are resolved effectively on an AMR grid. The exchange between the two grids is facilitated through smooth communication of vital information. The first part of the communication is from the fluid solver on a fixed coarse grid to the interface MOF-AMR grid, in which the node velocities needed to advect the interface are interpolated from the base grid for the Navier-Stokes solver. Specifically, the fluid solver generates face-defined velocities U and V, in the horizontal and vertical direction of a control volume, respectively. Linear interpolation is applied on a uniform Cartesian grid with i and j indices as follows:

$$\begin{aligned} u_n(i,j) &= \frac{U(i-1,j-1)+U(i-1,j)}{2} \\ v_n(i,j) &= \frac{V(i-1,j-1)+V(i,j-1)}{2} \end{aligned}, \tag{4.4}$$

where u_n and v_n are the horizontal and vertical velocities at the bottom left corner node of the control volume used in the MOF context. In the fluid solver, the pressure and volume fraction are cell-centered. Fig. 4.8(a) shows how node velocities are spatially interpolated for a single control volume, in addition to their directional components. Fig. 4.8(b) considers the node velocity interpolation when refinement is triggered in the dynamic procedure. These velocities are required during the material advection part.

The second part of the communication process exchanges information about volume fraction between the MOF-AMR grid and the fixed coarse grid used for the fluid solver in order to solve the system of discretised equations. The exchange is performed in the following way. Communication to the coarse grid is carried out using the equivalent volume fraction as follows



(a) Node velocity interpolated in a control volume denoted $CV(i,j)$. Plain black arrows are horizontal velocities U, hollow black arrows are vertical velocities V, both face-centered. The red arrow denotes the node velocity with components in horizontal and vertical directions u_n and v_n, respectively.

(b) Node velocity interpolation when refining grid for level 0 (green arrows), level 1 (yellow arrows) and level 2 (red arrows).

Fig. 4.8 Visual schematic of the interpolation of node velocity from the fluid solver on a fixed grid to the MOF-ADG grid. All other velocities when subsequently refining the grid are interpolated from the four fixed grid nodes.

$$F_{NS}(i,j) = \sum_{lev} \sum_{is,js \subset \{\Phi_{lev}\}} \frac{F_{ADG}(i,j,is,js,lev)\,|\Omega|\,(lev)}{|\Omega|\,(0)}\,, \tag{4.5}$$

where $F_{NS}(i,j)$ denotes the volume fraction used on the flow solver grid, Φ_{lev} is the set of cells/subcells and is, js and lev are indices in the AMR structure for each level of refinement and $|\Omega|\,(lev)$ corresponds to the subcell area. It is crucial to note that if a cell has not been refined then level 0 remains. In this case, the above relationship is simplified as follows:

$$F_{NS}(i,j) = F_{ADG}(i,j,1,1,0) \tag{4.6}$$

Fig. 4.9 serves as an illustrative snapshot of the effective communication and integration between the two grids. The depiction encapsulates the exchange of data and information that is fundamental to the success of the adaptive dual grid method. The coordination, as seen with the flowchart, is essential for maintaining accuracy and coherence in the numerical simulation.

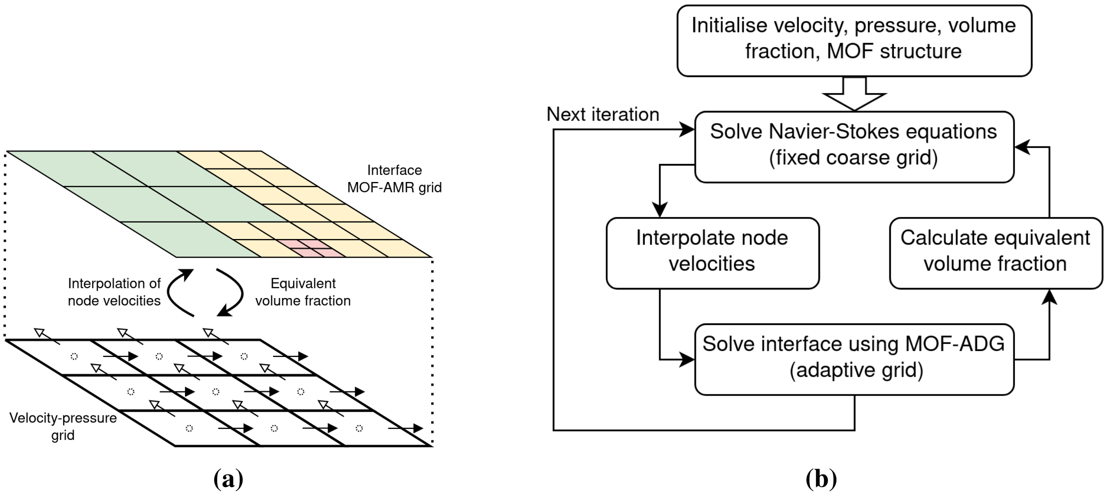

Interface
MOF-AMR grid
Interpolation of
node velocities
Equivalent
volume fraction
Velocity-pressure
grid
(a)
Initialise velocity, pressure, volume fraction, MOF structure
Next iteration
Solve Navier-Stokes equations (fixed coarse grid)
Interpolate node velocities
Calculate equivalent volume fraction
Solve interface using MOF-ADG (adaptive grid)
(b)

Fig. 4.9 (a) Structure and (b) flowchart of the adaptive dual grid method. (a) Top grid is the interface MOF-AMR grid using the same colouring scheme described in the previous section. Bottom grid is the fluid solver grid, where dashed circles represent cell-centered pressure and volume fraction $F_{NS}(i,j)$, plain arrows represent horizontal velocities U and hollow arrows vertical velocities V, both face-centered. Communication between grids is required for velocities and volume fractions.

4.5 Closing remarks

In conclusion, this chapter has meticulously detailed a novel unconstrained adaptive mesh refinement framework using the moment-of-fluid method (MOF-AMR). The second part of this chapter has detailed the key components required in order to achieve machine precision mass conservation in the MOF-AMR method. This generalises to any level of refinement, from a fixed grid to at least two levels of refinement. The last part of this chapter has described the adaptive dual grid framework, elucidating its fundamental principles and integration within a fluid flow solver for numerical simulations. By emphasising the significance of adaptivity, this framework has explored the enhancement of accuracy and efficiency of interface calculations, addressing the intricacies of multiphase flow phenomena.

The next chapter will describe the transition from methodology to results, assessing the performance and outcomes of employing the MOF method and its integration with the Navier-Stokes equations on a fixed grid, validating the methodology and applying it to real test cases. These results will provide valuable insights into the performance and impact of the method and into understanding complex fluid behaviour.

Chapter 5

A filament MOF method for multiphase flows on a fixed grid : Benchmarks and Results

5.1 Introduction

Progress in computational fluid dynamics (CFD) has transformed the capacity of the community to simulate and analyse intricate fluid flow phenomena. As the accuracy and reliability of numerical simulations is enhanced, the validation of computational methods becomes crucial. The MOF method, a promising technique for capturing fluid interfaces, has gained prominence in recent years due to its ability to simulate multiphase flows with improved accuracy.

The primary objective of this chapter is to present a comprehensive overview of the validation process employed for the MOF method and to showcase its successful application to a range of carefully selected test cases. Validating a numerical method involves a rigorous assessment of its ability to reproduce physical phenomena accurately under diverse conditions. By scrutinising its performance against well-defined benchmarks, this research seeks to establish the credibility and effectiveness of the MOF method as a robust tool for simulating complex fluid dynamics.

Furthermore, the chapter will explore the method's versatility through its application to a series of challenging test cases that encompass a spectrum of physical scenarios. Through the analysis of the results, the method's capability is assessed and compared to other methods to highlight its potential for addressing real-world challenges. In essence, this chapter serves as a bridge between the theoretical foundation of the MOF method and practical applications.

5.2 Validation: the filament MOF method

In this section, several benchmark problems are considered with the aim of testing the performance of the new filament MOF method. Several problems are of considerable interest since the associated velocity field yields high deformation in the material. Maintaining the correct topology at maximum deformation is attractive and important for most engineering problems. However, in order to assess the predictive capability of interface capturing methods, each of the flows is reversed over the same time period and compared to its original configuration. Whilst comparison with the initial condition is possible, the MOF enables one to evaluate the difference between the final reconstruction and the original/reference configuration rather than the initial reconstruction. From a computational cost perspective, the proposed model uses an analytical reconstruction where possible. Indeed, when only the reconstruction of a piecewise linear interface between two materials in a Cartesian cell is required, this approach is significantly more efficient (Lemoine et al., 2017). In order to reduce the total error in reconstruction, the interface is reconstructed based on the material with the smallest volume fraction in a cell as suggested by Mukundan et al. (2022). For cases involving more than two materials reconstruction, the symmetric multi-material approach is chosen.

5.2.1 Error evaluation

Evaluating errors in interface reconstruction is a powerful tool to compare different interface tracking/capturing methods. The numerical errors in terms of volume fraction can be evaluated using the L_1 error norm E_{L_1}

$$E_{L_1} = \sum \left| F_{final} - F_{initial} \right| |\Omega| \ , \tag{5.1}$$

its relative error norm E_r

$$E_r = \frac{\sum \left| F_{final} - F_{initial} \right| |\Omega|}{\sum |F_{initial}| \, |\Omega|} \ , \tag{5.2}$$

and the maximum error norm L_∞

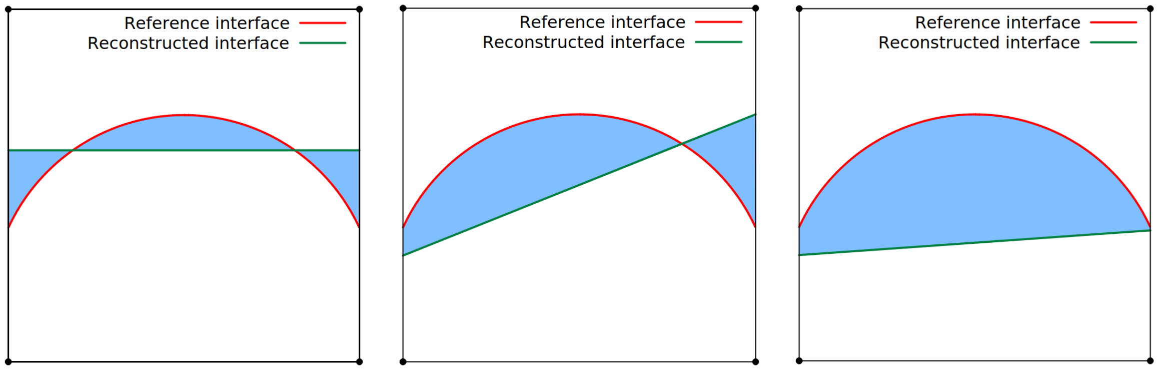
$$L_\infty = \max \left| F_{final} - F_{initial} \right| |\Omega| \tag{5.3}$$

A more representative error measure is the symmetric difference error which provides a better estimate of the interface reconstruction error. The symmetric difference error E_{sym} is given by

$$E_{sym} = \sum \left| \omega^{ref} \cup \omega^{act} - \omega^{ref} \cap \omega^{act} \right| \ , \tag{5.4}$$

where ω^{ref} denotes the initial state reference interface, which is potentially curved, and ω^{act} denotes the final state reconstructed polygon.

When comparing the reference interface with its reconstruction in individual cells, the symmetric difference error can be interpreted as the area between the two interfaces. Fig. 5.1 shows three different scenarios of intersecting interfaces and highlights the area corresponding to the symmetric difference error E_{sym}. Some simple calculations are necessary to evaluate the area of a segment.



(a) $n = 2$ **(b)** $n = 1$ **(c)** $n = 0$

Fig. 5.1 Symmetric difference error E_{sym} in a single cell. The area shaded in blue highlights the error corresponding to E_{sym} irrespective of the number of times, n, the reconstructed interface intersects the reference interface: (a) $n = 2$, (b) $n = 1$, (c) $n = 0$.

As well as evaluating the error in reconstruction, ensuring mass conservation is also crucial during these advection tests. In 2D, mass conservation corresponds to area preservation and mass loss is given by the expression:

$$\Delta m = \sum \left|F_{final}\right| |\Omega| - \sum \left|F_{initial}\right| |\Omega| \tag{5.5}$$

5.2.2 Benchmark: Zalesak slotted disc

In this benchmark test case, a slotted disc is advected in a rigid body rotation motion (Zalesak, 1979). A circle of radius $r = 0.15$, with a slotted rectangle of width $w_Z = 0.05$ and a maximum height of $h_Z = 0.85$, is centered at $(0.5, 0.75)$ in a unit square domain. The corresponding velocity field is given by:

$$\mathbf{u}(x,y) = \begin{bmatrix} 0.5 - y \\ x - 0.5 \end{bmatrix} \tag{5.6}$$

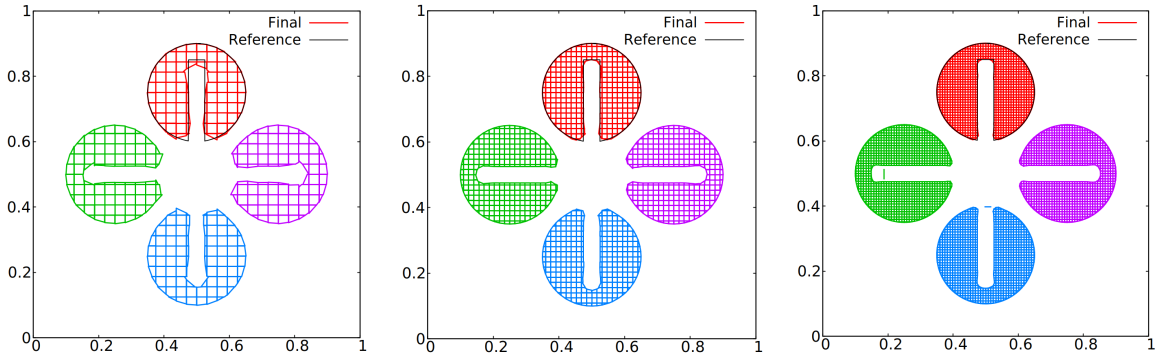
This case does not exhibit any filament formation, however it shows that the conglomeration algorithm works for velocity fields that rotate rather than deform the interface. Five different uniform grids have been used explicitly 32×32, 64×64, 128×128, 256×256 and 512×512. On the coarsest mesh 32×32, the number of iterations is set to be $n_{it} = 1256$ and $\Delta t = 2\pi/n_{it}$. The number of iterations is increased proportionally with increasing mesh refinement. Hence, Δt is decreased correspondingly.

In order to study the error convergence, the error measure used for this test case is the L_1 error. Table 5.1 summarises the L_1 error for different mesh sizes and highlights the convergence of the numerical approximation. In addition, Fig. 5.2 highlights the final solution after a full body rotation for the first three grids. The shape of the original interface is captured well. However, the sharp edges around the slotted rectangle have been smoothed out during the rotation. Indeed, the MOF method is not able to capture these edges regardless of the degree of mesh resolution. Potential improvements may arise with two half-planes MOF reconstruction. The maximum error L_∞ is a more relevant measure of the error for this problem in order to understand the order of convergence around sharp edges. In this case, second order convergence may be attained in some instances but it may depend on the alignment of the sharp edges of the slotted disc with the grid.

Table 5.1 Dependence of the L_1 error, E_{L_1}, relative error, E_r, and maximum error L_∞ on mesh size for the Zalesak slotted disc problem. Numbers in brackets indicate the order of convergence.

Mesh size	E_{L_1}	E_r	L_∞
32	3.17×10^{-3}(-)	5.45×10^{-2}(-)	2.77×10^{-4} (-)
64	9.02×10^{-4}(1.81)	1.55×10^{-2}(1.81)	5.23×10^{-5} (2.40)
128	3.81×10^{-4}(1.24)	6.54×10^{-3}(1.24)	3.47×10^{-5} (0.59)
256	1.35×10^{-4}(1.50)	2.31×10^{-3}(1.50)	1.52×10^{-5} (1.19)
512	4.93×10^{-5}(1.45)	8.47×10^{-4}(1.45)	2.47×10^{-6} (2.62)

The behaviour of the L_1 error over one rotation is shown in Fig. 5.3 for three different meshes mentioned in Fig. 5.2. The plot highlights that despite the interface only rotating, the



(a) 32 × 32 mesh **(b)** 64 × 64 mesh **(c)** 128 × 128 mesh

Fig. 5.2 Solution of rigid body rotation for the Zalesak slotted disc. Green depicts a quarter of rotation. Blue half rotation. Purple three quarter of rotation. Red depicts a full rotation and final solution. The black outline depicts the reference interface.

error increases during the rotation progresses as the interface reconstruction error accumulates at each time step.

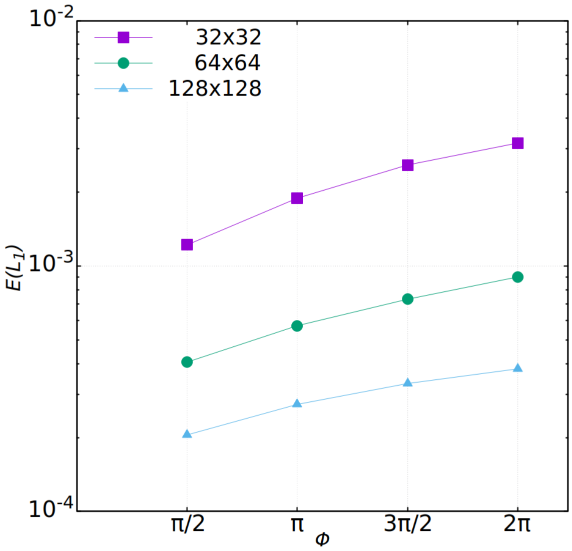


Fig. 5.3 Behaviour of the L_1 error during the rigid body rotation of the Zalesak slotted disc for different mesh sizes. Φ denotes the angle of full body rotation.

5.2.3 Benchmark: Reversible Vortex T=8

The reversible vortex is a benchmark test case for deforming advection cases (Rider and Kothe, 1998). A circle of radius $r = 0.15$ centered at $[0.5, 0.75]$ in a unit square domain is deformed in a divergence-free velocity field given by:

$$\mathbf{u}(x,y,t) = \begin{bmatrix} -\sin^2(\pi x)\sin(2\pi y) \\ \sin^2(\pi y)\sin(2\pi x) \end{bmatrix} \cos(\pi t/T) \ , \tag{5.7}$$

where T represents the full period and $T/2$ the time at maximum deformation. Here $T = 8$. The Courant-Friedrichs-Lewy (CFL) number is a dimensionless quantity defined as:

$$\mathrm{CFL} = \frac{u\Delta t}{\Delta x} \ , \tag{5.8}$$

where u is the characteristic velocity, Δt is the time step size, and Δx is the grid size. This number represents the ratio of physical distance a particle travels in one time step to its grid size. This ratio is a gauge to numerical stability in explicit methods. In this instance, the CFL number is 1, hence the number of iterations $n_{it} = 256$ and $\Delta t = \Delta x$ when a 32×32 uniform Cartesian mesh is considered. The number of iterations increases proportionally with mesh size.

The circle deforms in a filamentary structure at maximum deformation $t = T/2$. For this test case, filament detection is enabled. Several grids from 32×32 to 1024×1024 have been used to perform this dynamic test case.

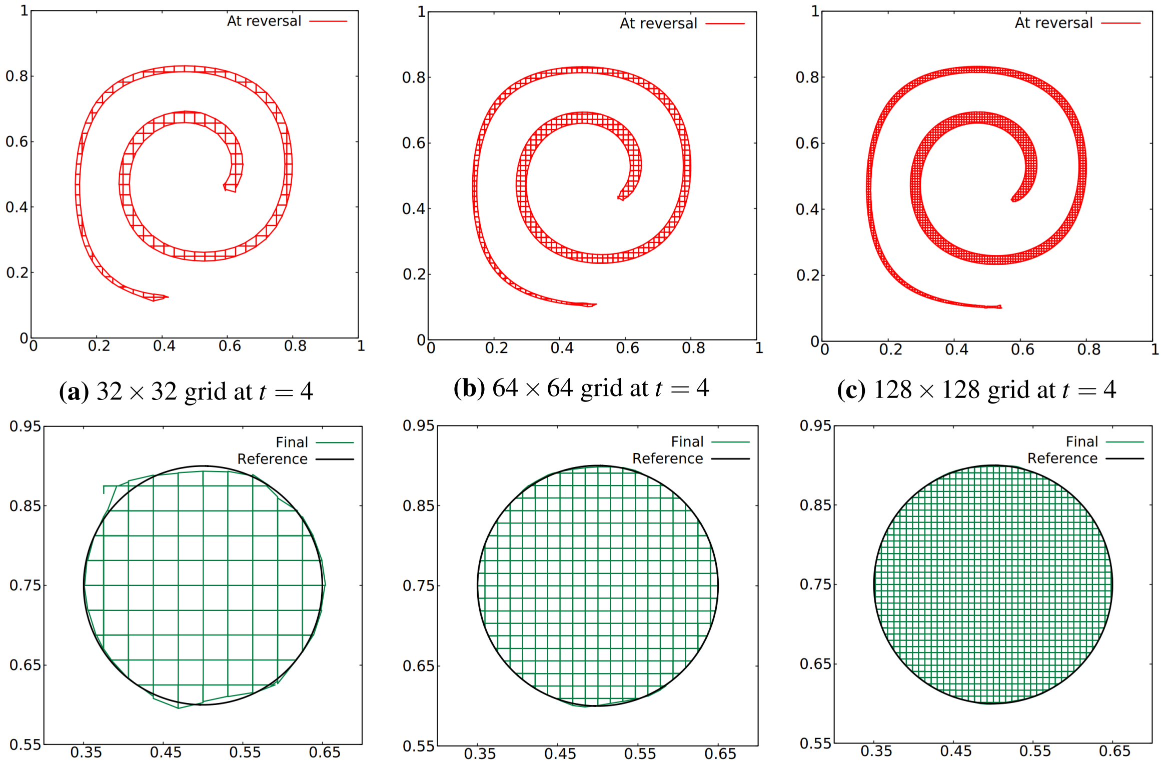
The symmetric difference error, E_{sym}, is shown in Table 5.2 for the initial reconstruction and at the final stage. The performance of the proposed method is compared with the results obtained using other MOF methods as well as with the standard MOF. Runtime, rounded to the next integer value, is also compared because the MOF method can be computationally expensive. Currently, the code has not been parallelised and so the computations are performed on a single core. The order of convergence of this method is also highlighted as well as the mass difference.

Fig. 5.4 shows the maximum deformation before reversal and the final reconstruction for different mesh sizes: 32×32, 64×64 and 128×128, respectively. Using a filament

Table 5.2 Reversible vortex test case data using $T = 8$ compared with the standard MOF (STD MOF) and with results generated using other MOF methods in the literature: a standard MOF with adaptive mesh refinement (AMR) method (Ahn and Shashkov, 2009), a filament AMR method (Jemison et al., 2015), a coupled level-set MOF (CLSMOF) (Mukundan et al., 2022).

Mesh size	32	64	128	256	512	1024
E_{sym} in (Ahn and Shashkov, 2009)	2.34×10^{-2}	3.31×10^{-3}	5.78×10^{-4}	1.22×10^{-4}	2.01×10^{-5}	-
Order of convergence	-	2.82	2.51	2.24	2.60	-
E_{sym} in (Jemison et al., 2015)	3.12×10^{-3}	6.91×10^{-4}	2.77×10^{-4}	-	-	-
Order of convergence	-	2.17	1.31	-	-	-
Runtime in (Jemison et al., 2015)	32.6	200	635.3	-	-	-
E_{sym} in (Mukundan et al., 2022)	1.32×10^{-3}	1.01×10^{-3}	5.44×10^{-4}	2.76×10^{-4}	1.38×10^{-4}	6.90×10^{-5}
Order of convergence	-	0.39	0.89	0.98	1.0	1.0
E_{sym} for STD MOF	1.42×10^{-2}	7.46×10^{-3}	1.29×10^{-3}	9.19×10^{-5}	1.45×10^{-5}	4.07×10^{-6}
Order of convergence	-	0.92	2.53	3.81	2.66	1.83
Initial E_{sym}	1.74×10^{-4}	4.06×10^{-5}	1.28×10^{-5}	2.99×10^{-6}	1.49×10^{-7}	4.19×10^{-8}
Final E_{sym}	2.80×10^{-3}	5.06×10^{-4}	1.54×10^{-4}	4.45×10^{-5}	1.48×10^{-5}	3.64×10^{-6}
Order of convergence	-	2.46	1.71	1.79	1.58	2.02
Mass difference	1.09×10^{-6}	2.54×10^{-7}	3.98×10^{-8}	2.31×10^{-9}	-1.50×10^{-6}	2.05×10^{-12}
Runtime (s)	17	31	95	447	4856	43942

approach, the vortex does not exhibit any spurious separated structures, even on a coarse mesh. In this test case, the trailing tail shows a thicker structure as the coarse cell cannot reconstruct the filament tail accurately. As the mesh is refined, the tail becomes well-defined but thicker than the filament width. The MOF method naturally creates these structures as it exhibits some cross-stream diffusion, leading to a shorter tail than expected. The symmetric difference error converges slightly faster than other MOF methods with a smaller error on the finest mesh. Runtime is also considerably faster by a factor of between two to five, although it may be related to computer and compiler performance. The error shows high order of convergence, almost matching the reference order two. The symmetric difference error for standard MOF exhibits a slower order of convergence on coarser grids. However, the symmetric difference error is almost indistinguishable on the finest meshes for the two approaches.

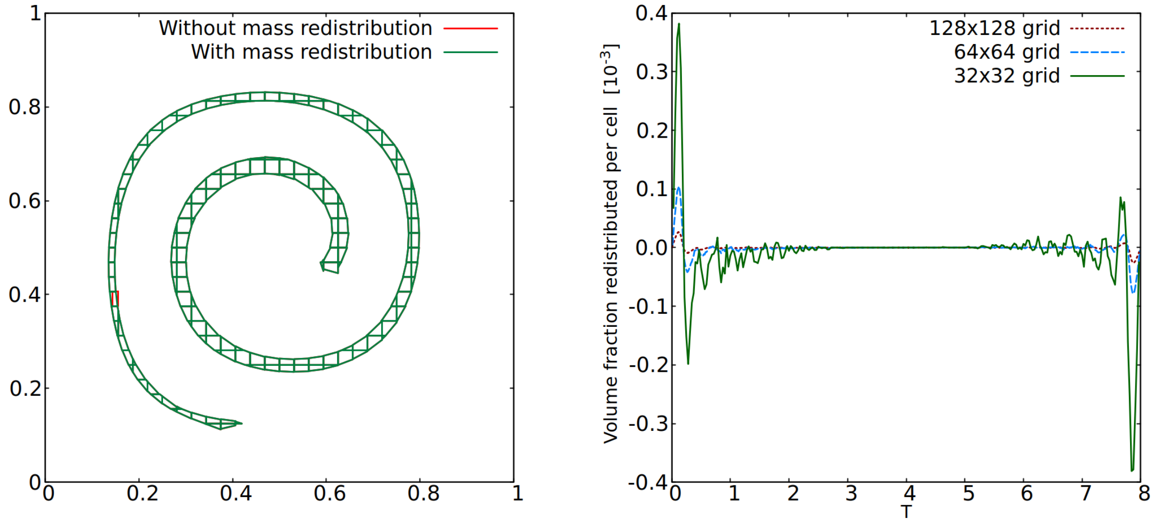


(a) 32×32 grid at $t = 4$
(b) 64×64 grid at $t = 4$
(c) 128×128 grid at $t = 4$
(d) 32×32 grid at $t = 8$
(e) 64×64 grid at $t = 8$
(f) 128×128 grid at $t = 8$

Fig. 5.4 Reversible vortex test case using the filament MOF method and $T = 8$ for 32×32, 64×64 and 128×128 grids. Top row of figures shows the maximum deformation. Bottom row of figures shows the final interface.

5.2.3.1 Influence of the mass redistribution

The investigation of the effect of mass redistribution is explored in this section. It is observed that the remapping, or mass redistribution procedure does not affect the topology greatly as the volume fraction that needs to be redistributed during the procedure is very small. Indeed, during the deformation of the vortex the volume fraction redistributed varies between 10^{-4} and 10^{-10}. During the early stages of the deformation, most mass has to be redistributed as there are many cells in the inner part of the circle that are over/under-filled and very few cells are mixed cells, i.e. cells containing an interface. On the contrary, at maximum deformation, very few cells are over/under-filled cells, most of them contain one interface, or two in the case of filaments. Fig. 5.5 summarises this. The difference between the interface shapes obtained with and without the post advection remapping procedure is highlighted in Fig. 5.5(a), whereas Fig. 5.5(b) shows the variation of volume fraction redistributed per iteration. Note that the mass is redistributed equally between mixed cells. Table 5.3 shows the mass difference for the reversible vortex case ($T = 8$) for a case where the mass was redistributed and when it was not. Two orders of magnitude of difference can be observed, which shows the advantage of using the proposed method for mass conservation. It is important to note that the mass redistribution does not achieve machine precision in the redistribution case as only one round of redistribution is performed in this work. This choice, made to reduce the computational effort, prevents additional rounds of redistribution from achieving complete mass balance.



(a) Filament MOF reconstruction including comparison with post advection remapping procedure

(b) Volume fraction redistributed as a function of the time period of the reversible vortex T

Fig. 5.5 Comparison showing (a) the effect on the interface shape of the post advection remapping procedure for mass conservation and (b) the actual mass redistributed per iteration for different grids.

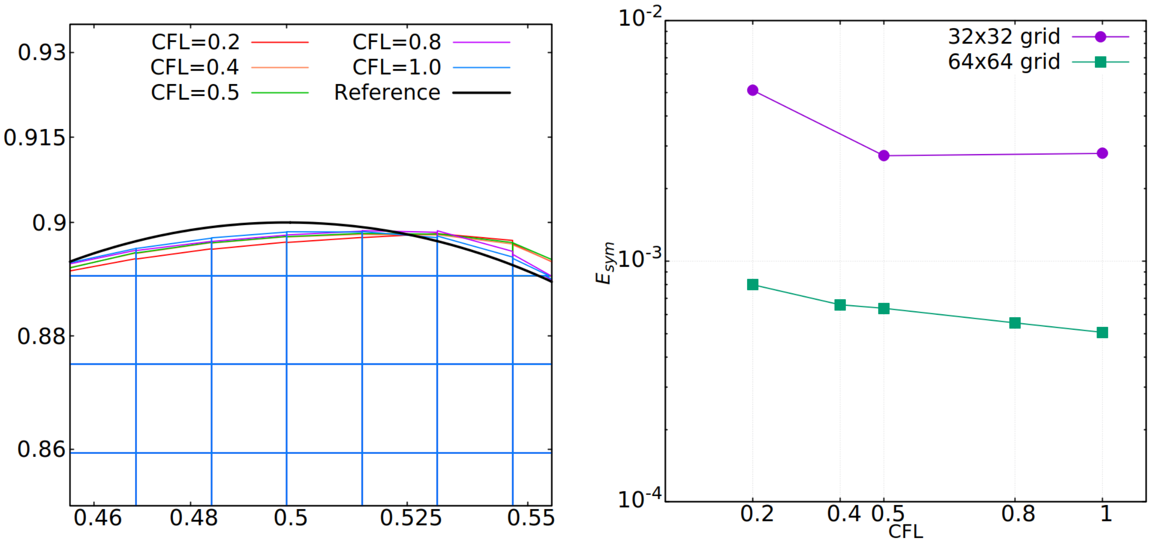
5.2.3.2 Influence of the CFL number on the interface

One expects the CFL number to influence the interface reconstruction. However, the Lagrangian advection procedure is not greatly affected by the CFL number. Therefore, most

Table 5.3 Mass difference for the reversible vortex with and without post advection remapping procedure.

Mesh size	With redistribution	Without redistribution
32	1.09×10^{-6}	1.35×10^{-4}
64	2.54×10^{-7}	2.58×10^{-5}
128	3.98×10^{-8}	3.75×10^{-6}
256	2.31×10^{-9}	5.72×10^{-6}

cases are performed with the maximum available CFL number which equals unity. In theory, a CFL number greater than unity can be used for such advection benchmarks. However, the stencil used in the dynamic test procedure encompasses only a 3×3 stencil and therefore limits larger CFL numbers. Fig. 5.6 shows a zoom on the final reconstruction for different CFL numbers 0.2, 0.4, 0.5, 0.8 and 1.0 and the error convergence. A lower CFL number will induce a larger number of iterations, therefore increasing the chances of error in reconstruction. However, the difference in error is relatively small in magnitude. The difference on the interface only occurs near the top of the circle which is near the tip of the filament at maximum deformation.



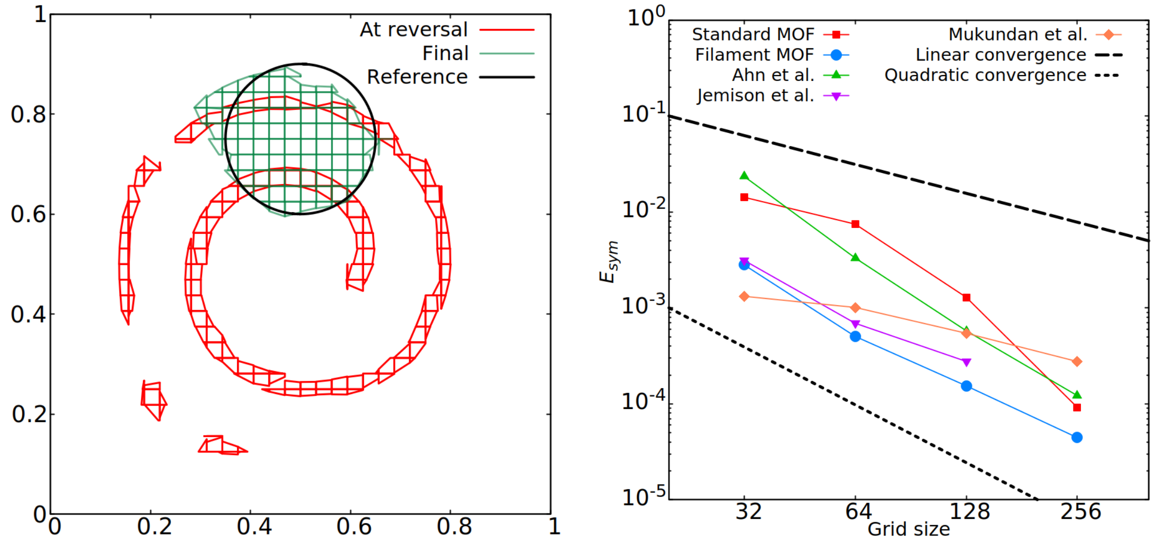
(a) Zoom on final reconstruction on a 64×64 grid

(b) E_{sym} as a function of the CFL number on two grids

Fig. 5.6 (a) Influence of the CFL number on the final reconstruction of part of the interface; (b) symmetric difference error, E_{sym}, as a function of the CFL number for two grids: 32×32 and 64×64.

5.2.3.3 Influence of the filament capable method

Filament capable MOF is able to reconstruct a moving interface with a greater accuracy and better topology. Indeed, under strong deformations, materials tend to break up when they are not supposed to do so. At the instant of maximum deformation, a continuous interface is more likely and will result in better modelling of multiphase flows. Fig. 5.7 highlights both visual reconstruction and convergence of the standard and filament solution. Fig. 5.7(a) shows that several breakups of the dynamic interface occur when a standard MOF reconstruction is implemented. The final reconstruction does not match the reference circle. Fig. 5.7(b) compares the order of convergence between a standard and the proposed filament approach, together with other MOF methods. Note that for the finer grids, the error tends to the same values as the thickness of the structure is larger than a cell size, hence the filamentary approach is not used as frequently during the dynamic test.



(a) Standard MOF reconstruction for a 32 × 32 grid at reversal (in red) and final (in green). Black outline denotes the reference interface.

(b) Convergence behaviour of the standard MOF and filament MOF compared with the literature using E_{sym}.

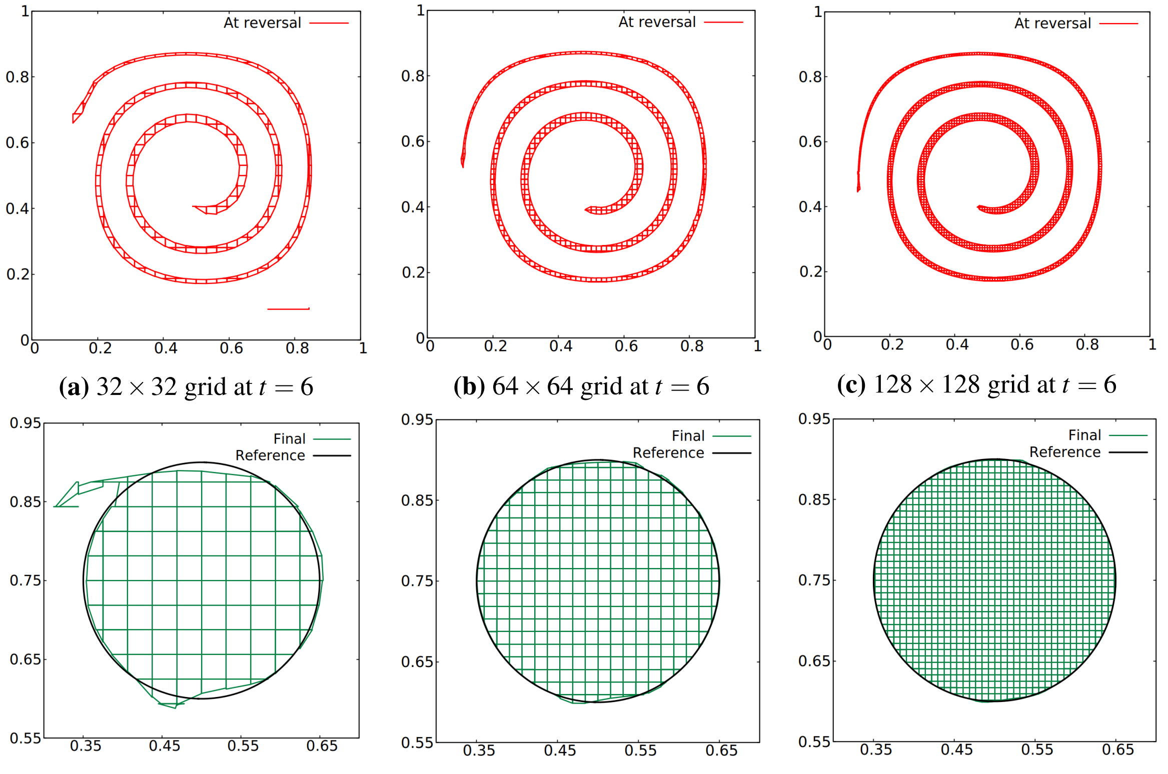
Fig. 5.7 (a) Influence of the filament capable method on the reconstruction; (b) Symmetric difference error E_{sym} compared with other MOF methods. Convergence rate is compared with a linear and quadratic reference.

5.2.4 Test case: Reversible Vortex T=12

This is the same benchmark test case as considered in Section 5.2.3 except that the full period is increased to $T = 12$. A larger period increases the deformation and thinner filaments are exhibited. Table 5.4 summarises the symmetric difference error for six different mesh sizes from 32×32 to 1024×1024. As the material is more deformed than in the previous benchmark with $T = 8$, the expected symmetric difference error is larger. As the mesh is refined, there are no longer significant benefits associated with using the filament method as the thickness of the deformed filament is greater than a cell width. Consequently, the order of convergence decreases from quadratic to linear until the order of convergence of the filament MOF follows that for standard MOF. Unexpected tendencies are observed in the finest mesh of this test case, which according to the author may be attributed to infinitesimal area calculation behaviour. The mass difference is very comparable. However, runtime is increased significantly. Indeed, the number of cells containing a filament structure compared to a standard interface is very large. Fig. 5.8 highlights the morphology of the very thin interface. Because filament reconstruction is computationally more expensive, the runtime is increased by a factor of three.

Table 5.4 Symmetric difference error, L_1 error, mass difference and runtime for the reversible vortex test case using $T = 12$ and its comparison with the standard MOF (STD MOF).

Mesh size	32	64	128	256	512	1024
Final E_{sym}	4.98×10^{-3}	9.91×10^{-4}	2.48×10^{-4}	1.27×10^{-4}	2.06×10^{-5}	6.33×10^{-6}
Order of convergence	-	2.32	1.99	0.96	2.62	1.70
E_{L_1}	4.18×10^{-3}	9.62×10^{-4}	2.58×10^{-4}	1.23×10^{-4}	4.11×10^{-5}	6.29×10^{-6}
Order of convergence	-	2.11	1.89	1.06	1.58	2.70
Mass difference	3.12×10^{-6}	3.08×10^{-7}	4.93×10^{-8}	4.01×10^{-9}	-8.93×10^{-12}	-7.33×10^{-7}
Runtime (s)	51	95	180	958	7334	65418
E_{sym} for STD MOF	2.66×10^{-2}	1.81×10^{-2}	3.37×10^{-3}	1.18×10^{-3}	4.05×10^{-5}	7.70×10^{-6}
Order of convergence	-	0.55	2.42	1.51	4.86	2.39



(a) 32×32 grid at $t = 6$ **(b)** 64×64 grid at $t = 6$ **(c)** 128×128 grid at $t = 6$

(d) 32×32 grid at $t = 12$ **(e)** 64×64 grid at $t = 12$ **(f)** 128×128 grid at $t = 12$

Fig. 5.8 Reversible vortex test case using $T = 12$ for 32×32, 64×64, 128×128 grids. Top row of figures shows the maximum deformation. Bottom row of figures shows the final interface.

5.2.5 Benchmark: Droplet flow

The droplet flow test case has a nonlinear divergence free velocity field (Ahn and Shashkov, 2009). The deformation of material tears an initial circle of radius $r = 0.125$ centred in a unit domain into a V-shape. The velocity field is given by:

$$\mathbf{u}(x,y,t) = \begin{bmatrix} 0.125(8x-4) \\ 0.125\left[-(8y-4)-4-\left(1-(8x-4)^2-(8x-4)^4\right)\right] \end{bmatrix} f(t) \qquad (5.9)$$

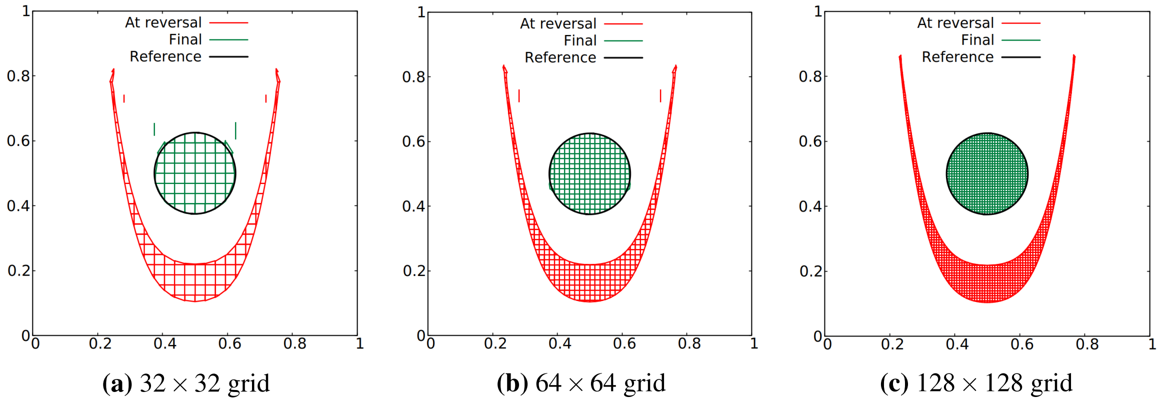
The velocity field is a function of time as the amplitude, $f(t)$, varies in time according to:

$$f(t) = \begin{cases} 1 & 0 \leq t < T_{max} - t_\varepsilon/2 \\ \cos\left(\frac{\pi(t - T_{max} + t_\varepsilon/2}{t_\varepsilon}\right) & T_{max} - t_\varepsilon/2 \leq t \leq T_{max} + t_\varepsilon/2 \\ -1 & T_{max} + t_\varepsilon/2 < t \leq 2T_{max} \end{cases} \tag{5.10}$$

At $T_{max} = 0.8$, time at maximum deformation, the flow is reversed. The flow is reversed smoothly during a transition period of $t_\varepsilon = 0.1$.

This test case provides a good insight into the filamentary formation of materials as the filament tip is leading as opposed to trailing in the previous benchmark. For the base grid, 32×32, the number of iterations is set to $n_{it} = 160$ for the entire simulation and $\Delta t = 0.01$. The number of iterations is increased proportionally with the mesh and therefore Δt is decreased proportionally with the mesh.

The dynamic test is performed for different grids from 32×32 to 256×256 using a filamentary MOF method. As the mesh is refined, this approach becomes less relevant. The symmetric difference error is compared with Jemison et al. (2015) despite an AMR capability being used in that paper. In addition, details of mass conservation and runtime are given in Table 5.5. Fig. 5.9 shows the maximum deformation and final reconstruction for 32×32, 64×64, 128×128 grids, respectively. It can be seen that coarser meshes lead to larger errors in reconstruction. In addition, the method exhibits some diffusion in the sense of "floating" elements. These "floating" elements could be attenuated with a higher tolerance in available cell volume fraction. Lower volume fraction tends to create long and thin polygons, hence a larger error in reconstruction. The lower bound of volume fraction available in a cell is set to 10^{-5} in the proposed model, compared to 10^{-8} in most comparative studies. Both maximum deformation and final reconstruction show a symmetric left-right deformation. As the tip of the filament gets thinner, even the filamentary approach cannot reconstruct the structure accurately. This leads to a shrinked filament structure. When the grid is refined, the tip of the filaments are well-defined and the final solution shows acceptable errors. The mass difference is acceptable, bearing in mind the choice of only one round of redistribution. In terms of runtime, the performance is compared with Jemison et al. (2015) which uses an AMR scheme and thus fewer cells than in the proposed method. The order of convergence of the solution shows a remarkable performance compared to methods described in other papers.



(a) 32 × 32 grid **(b)** 64 × 64 grid **(c)** 128 × 128 grid

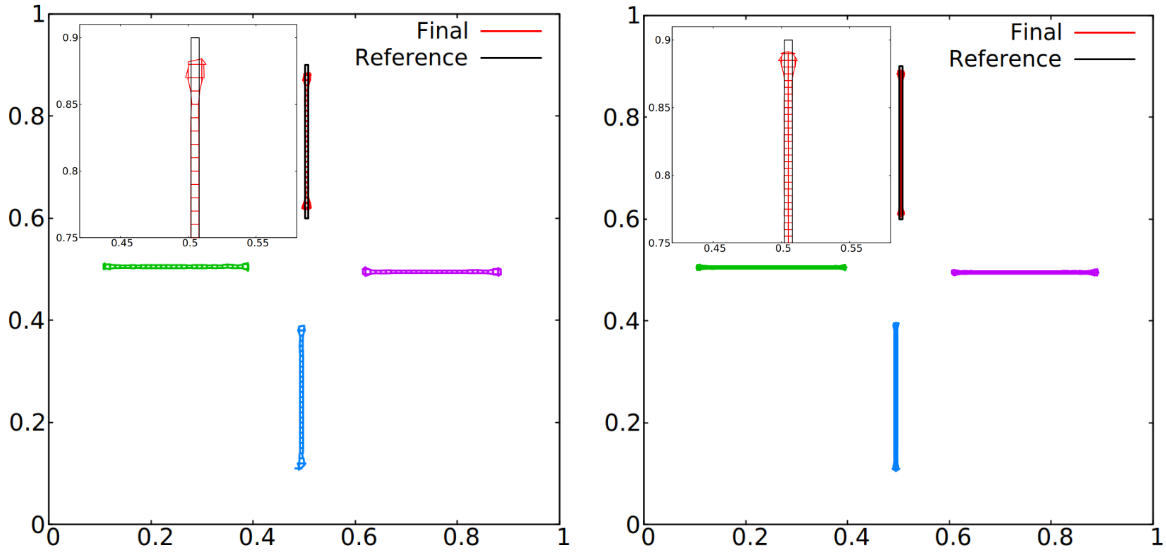
Fig. 5.9 Intermediate and final reconstruction for the droplet flow test case for different mesh sizes. Red depicts the maximum deformation before reversal. Green depicts the final reconstruction. The black outline is the reference circle.

Table 5.5 Symmetric difference error, order of convergence, mass difference and runtime for the droplet flow test case at final reconstruction compared to reference papers.

Mesh size	32	64	128	256
E_{sym} in (Jemison et al., 2015)	2.48×10^{-3}	6.37×10^{-4}	2.96×10^{-4}	-
Order of convergence	-	1.96	1.10	-
Runtime (s)	191.3	529.3	940.4	-
E_{sym}	1.71×10^{-3}	7.36×10^{-4}	1.26×10^{-4}	5.09×10^{-5}
Order of convergence	-	1.21	2.54	1.30
Mass difference	-1.16×10^{-9}	-1.07×10^{-7}	-4.49×10^{-11}	9.31×10^{-12}
Runtime (s)	3	9	29	166

5.2.6 Benchmark: Rotating filament

The rotating filament benchmark is a test case where a thin rectangle is advected anti-clockwise in a rigid body rotation motion (Ahn and Shashkov, 2009). The velocity field is the same as in Section 5.2.2 and is given in Eq. (5.6). The rectangle is centered at $(0.505, 0.75)$ in a unit square domain. Its initial width is $w_f = 0.006$ and height is $h_f = 0.3$. For a coarse mesh, here a 100×100 grid, the initial condition may already contain a filament structure. The corresponding number of iterations is set to $n_{it} = 300$ and $\Delta t = 2\pi/n_{it}$.



(a) 100×100 grid

(b) 200×200 grid

Fig. 5.10 Solution of rigid body rotation for the rotating filament. Green depicts a quarter of rotation. Blue half rotation. Purple three quarter of rotation. Red depicts a full rotation and final solution. The black outline depicts the reference interface.

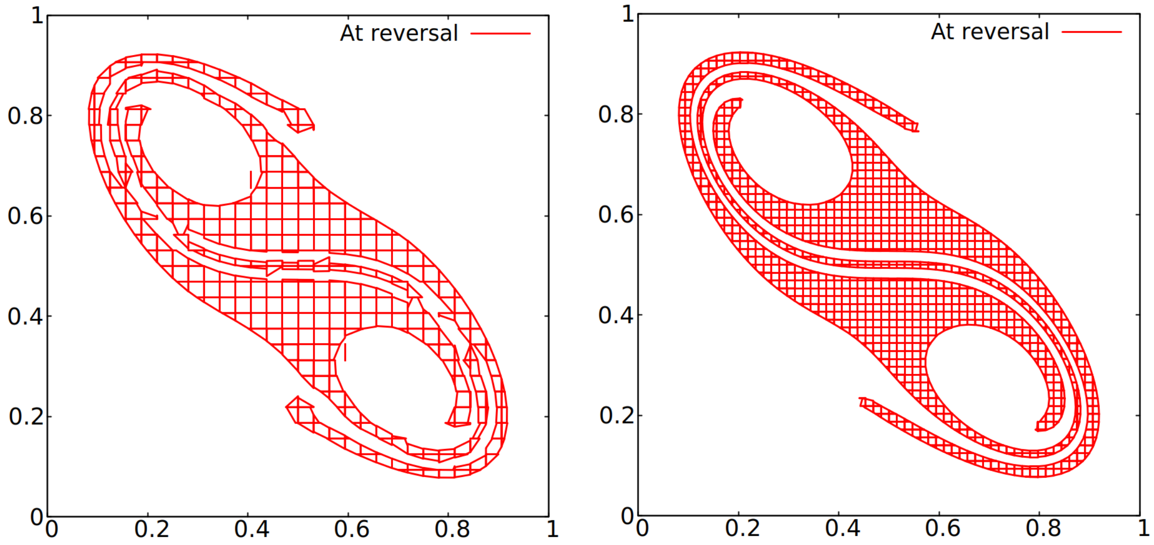
This benchmark can only be tested with a filament enabled approach on such coarse meshes. Indeed, even with a 200×200 grid, the filament body is subject to under-resolved filamentary structures. Fig. 5.10 shows the rotating filament at different stages of the full body rotation. The filament body is well reconstructed. However, both ends of the filament show cross-stream diffusion because the MOF method cannot reconstruct sharp edges accurately. In addition, the filament height is shortened due to the reconstruction error. The filament shortening matches with the height shown in Jemison et al. (2015). The zoom on the top left of the figure highlights both shortening of the filament and cross-stream diffusion compared to the reference rectangle outlined in black.

5.2.7 Benchmark: S-shape

The S-Shape test case comprises a circle of radius $r = 0.25$ located at the centre of a unit square domain at $t = 0$ (Ahn and Shashkov, 2009). The associated velocity field is nonlinear, divergence free and given by:

$$\mathbf{u}(x,y,t) = \begin{bmatrix} 0.25[(4x-2)+(4y-2)^3] \\ -0.25[(4y-2)+(4x-2)^3] \end{bmatrix} f(t) \tag{5.11}$$

The advection process creates a highly deformed and thin structure which means the filamentary capability is also enabled here. The amplitude $f(t)$ is given in Eq. (5.10). However, in this benchmark problem the maximum deformation occurs at $T_{max} = 4$ and the smooth transition period is $t_\varepsilon = 2$.



(a) 32 × 32 grid

(b) 64 × 64 grid

Fig. 5.11 Maximum deformation for the S-shape benchmark.

This case shows strong deformation and thin structures, mainly in the centre of the domain. A coarse mesh would struggle to reconstruct these structures. Indeed, for the 32 × 32 grid in Fig. 5.11, the central part may have three interfaces within a cell. Therefore, capping to three materials is a limiting factor, creating larger errors in reconstruction. Because of large reconstruction errors, several structures may merge and lead to different end results. The 64×64 grid is fine enough to have a maximum of two interfaces in a cell. The deformed interface shows an accurate representation at maximum deformation.

5.3 Application: the MOF method for simulating multiphase flows on a fixed grid

In this section, the coupling between a flow solver and the proposed MOF method is validated. Results obtained from numerical simulations using the MOF method for simulating multiphase flows are presented. The filament capability is enabled for this series of simulations. This implies that if the topology does not produce thin structures, the reconstruction will remain standard. The simulations were performed on a two-dimensional domain using a Cartesian grid in order to validate the numerical implementation by considering well-known benchmark test cases.

We have shown that standard MOF and filament MOF methods have shown good agreement on pure advection test cases where a prescribed divergence-free velocity field is imposed (Hergibo et al., 2023). In this section, these methods are applied to dynamic physical test cases where the velocity field is not prescribed a priori but determined by solving the momentum equation. The velocity field will not be identically divergence-free due to numerical rounding errors.

In this context, the incorporation of subgrid-scale filaments (shown in Fig. 5.12 for example and also discussed in Section 3.4 for more details) within the MOF method constitutes a novel contribution to the simulation of multiphase flows. This innovative approach addresses the challenges of resolving subgrid-scale features and complex changes in topology. By introducing subgrid filaments, the MOF method enhances the representation of thin films and droplets, while operating on a relatively coarse computational grid.

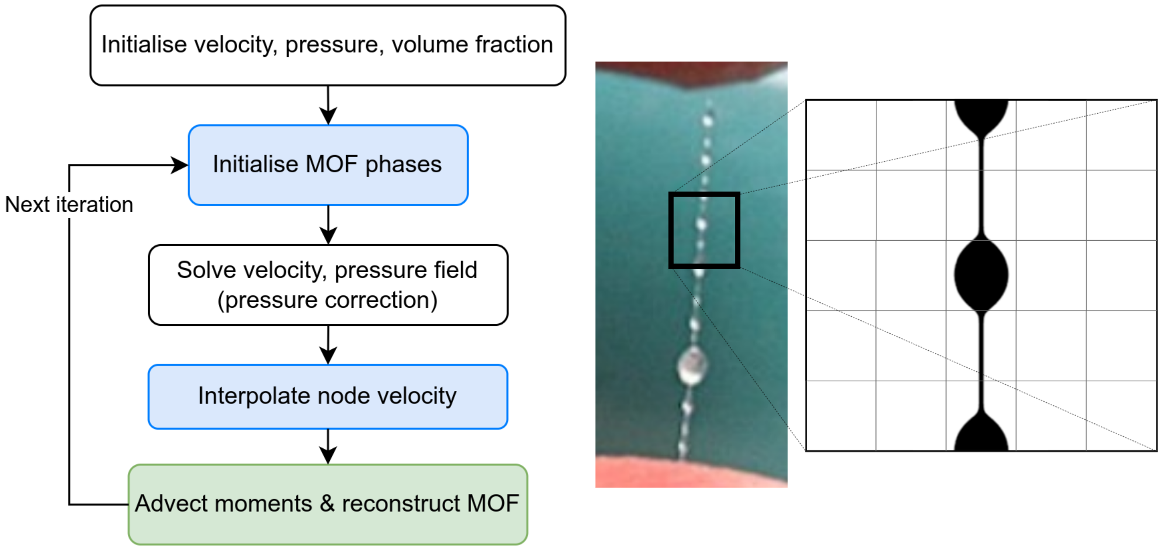
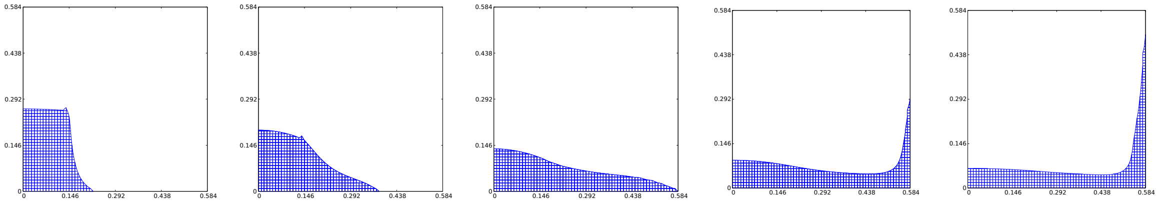

Initialise velocity, pressure, volume fraction
Initialise MOF phases
Next iteration
Solve velocity, pressure field (pressure correction)
Interpolate node velocity
Advect moments & reconstruct MOF

Fig. 5.12 Flowchart highlighting in blue the extra steps needed for the MOF method based multiphase flow solver. In green, the MOF method can select a standard or filament MOF. On the right hand side, subgrid structures are highlighted on a mesh-based approach using the example of saliva beads (Bhat et al., 2010).

5.3.1 Dam break problem

We examine the classical 2D dam break benchmark problem, that has been investigated extensively both experimentally (Martin et al., 1952) and numerically (Pavlidis et al., 2016; Zhang et al., 2010). The computational domain has dimensions of $4a \times 4a$ in the streamwise and vertical directions, respectively. Initially, a water column of height $2a$ and width a is at rest. In this study, $a = 0.146$ m is chosen. Three computational grids are used, 32×32, 64×64, 128×128, respectively. Time is nondimensionalised by $\sqrt{9.81/a}$. In terms of boundary conditions, the top boundary can be assumed to be an outflow, allowing material to leave the domain. For the other boundaries, left, right and bottom, a no-slip condition can be applied as the fluid adheres to the wall. However, in terms of the MOF method, a slip condition may be more appropriate for these boundaries. A slip condition ensures a zero velocity in the normal direction to the wall while allowing a calculated velocity in the tangential direction. Since the proposed MOF method is node-based, the node velocity at the wall cannot be zero; otherwise the material would be unable to *glide* on the surface. This boundary condition scenario is illustrated in Fig. 3.12.

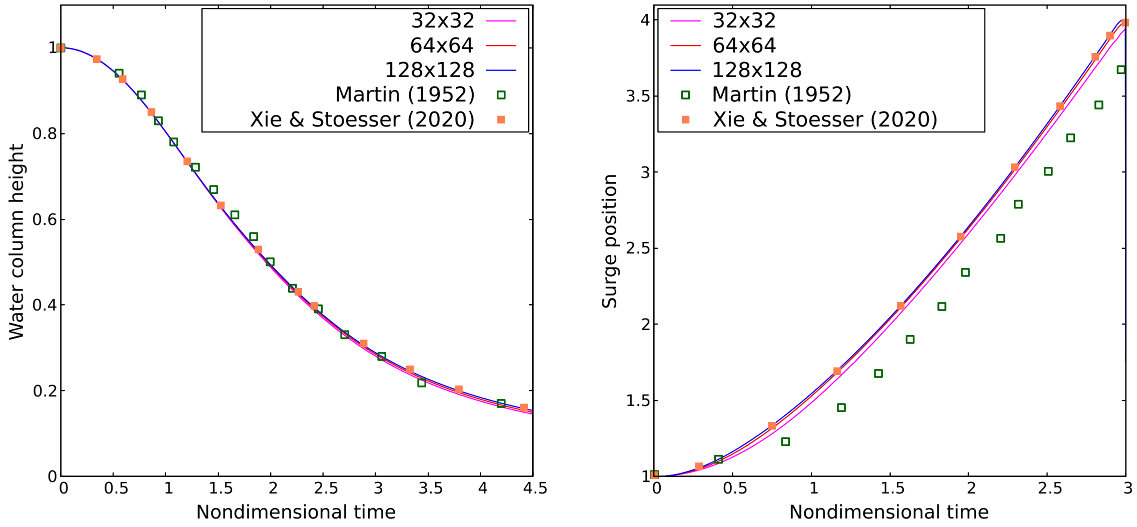


(a) $T = 1$ **(b)** $T = 2$ **(c)** $T = 3$ **(d)** $T = 4$ **(e)** $T = 5$

Fig. 5.13 Snapshots of the interface for the dam break test case for a 64×64 grid using nondimensional time.

Fig. 5.13 displays snapshots of the computed remaining water interface for a coarse mesh. Physically, two variables are good indicators of high-fidelity simulations : the surge front (location of the leading edge of the flood wave at the bottom boundary) and height position (vertical elevation of the water surface at the wall). These can be compared with experiments. Fig. 5.14 shows the comparison between simulations using several meshes with the VOF simulations of Xie and Stoesser (2020) and experiments of Martin et al. (1952) for surge front position and water column height. Generally, the numerical results converge as the mesh is refined, with only a minor disparity in surge position between the coarse and fine mesh simulations. In terms of surge position, since the dam cannot be instantly removed in the experiment, a slight time delay in the experimental data is expected, which is also observed in other numerical simulations. A delay of 0.27 in nondimensional units is observed, which corresponds to 0.023 s.

In order to investigate the convergence rate of the present method further, the free surface profiles obtained from the simulations on the three meshes at $T = 1$ and $T = 2$ are compared to the benchmark solution. Additionally, an extra simulation using an even finer mesh (256×256) is performed; the resulting air-water interface is considered as the benchmark solution for the purpose of the convergence study. Fig. 5.15 presents the calculated L_1, L_2, and L_∞ errors in relation to the benchmark solution. The convergence rate is observed to

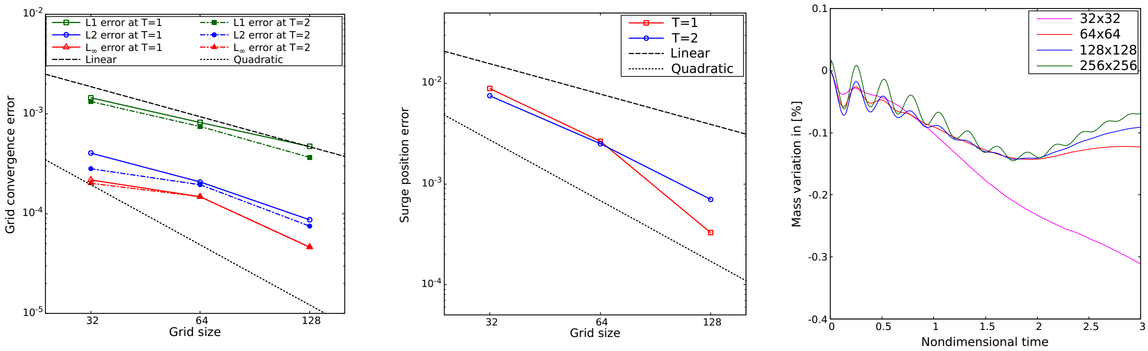


(a) Water column height

(b) Surge front position

Fig. 5.14 Convergence of (a) water column height, and (b) surge front position, with mesh refinement and comparison with the numerical predictions of Xie and Stoesser (2020) and the experimental data of Martin et al. (1952).

lie between first-order and second-order, tending towards second-order as the fine mesh is approached. The surge position error shows similar trends. It is interesting to note the oscillation in mass variation during the simulation due to the advection of the interface only and redistribution procedures. On the finest mesh, mass loss is below 0.15%.



(a) Grid convergence

(b) Surge position convergence

(c) Mass variation

Fig. 5.15 Convergence study for the dam break case, using grid convergence and surge position convergence as well as mass variation.

5.3.2 Rayleigh-Taylor instability

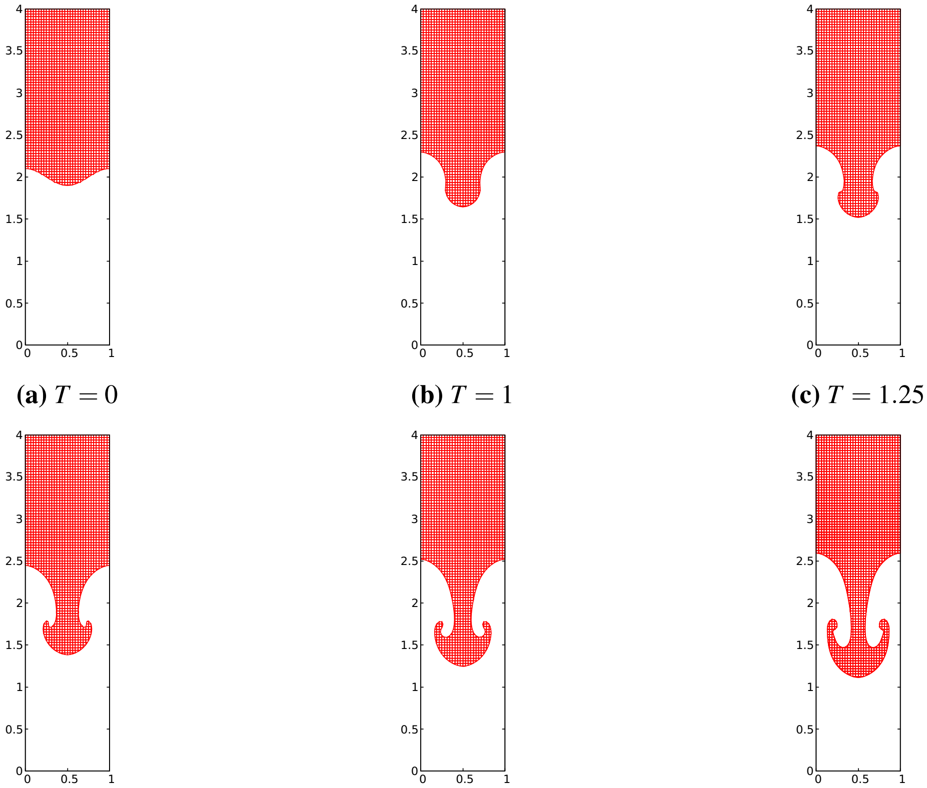
This well-known instability is a phenomenon that emerges when a gravitational field causes a heavy fluid initially at rest on top of a lighter fluid to deform the interface between them. In this study, the same configuration as previous investigations (Ding et al., 2007; Tryggvason, 1988; Xie et al., 2014) is adopted. The rectangular domain is $[0,d] \times [0,4d]$. The Atwood ratio A, a measure of the density difference between the heavier and lighter fluids, is defined by

$$A = \frac{\rho_H - \rho_L}{\rho_H + \rho_L}, \tag{5.12}$$

where the subscripts H and L, respectively, denote a heavy and light fluid. Time is nondimensionalised by $\sqrt{d/Ag}$, and surface tension and turbulence effects are ignored in this study. In this test case, $A = 0.5$. Additionally, the Reynolds number is set to $Re_A = \rho_A d^{3/2} g^{1/2} / \mu_A = 3000$, where ρ_A and ρ_B represent the densities of the heavier and lighter fluids, respectively, and μ_A corresponds to the dynamic viscosity of the heavier fluid. The interface between the two fluids is initially perturbed with a sinusoidal waveform of amplitude $0.1d$. Following the work of Tryggvason (1988), nondimensional variables are employed, scaling length by d, time by $\sqrt{d/Ag}$, and velocity by $\sqrt{Agd}$. Similar to the dam break case, slip conditions are applied on the left and right walls in order to maintain the no-penetration conditions but allows the fluid to move along the wall. No-slip conditions are applied on the top and bottom boundaries. Surface tension is not considered here.

We conduct computations using three fixed Cartesian meshes, 32×128, 64×256 and 128×512, respectively. Fig. 5.16 highlights the different stages of the deformation of the interface from a nondimensional time $T = 0$ to $T = 2$. The author observes that the vortex formed during the spike penetration is well reconstructed. The interface remains sharp and does not exhibit any diffusion.

The predictions of the present numerical scheme for this test case are compared to those of other methods in order to ensure the correct behaviour is achieved. Fig. 5.17 shows the



(a) $T = 0$ **(b)** $T = 1$ **(c)** $T = 1.25$

(d) $T = 1.5$ **(e)** $T = 1.75$ **(f)** $T = 2$

Fig. 5.16 Evolution of the Rayleigh-Taylor instability interface using the standard MOF method for a 32×128 grid.

position of the perturbation for both the heavy and light fluid. Good agreement is found with other methods for both fluids (Tryggvason, 1988; Xie et al., 2014). Whilst the position of the perturbation is a good criterion to ensure that the physics is correctly captured, the precision of the vortex is also a good criterion. However, the vortex is reconstructed more precisely with finer meshes. Grid convergence is not appropriate for this test case. In numerical simulations, a finer grid tends to approach the true solution, however the work of Tryggvason (1988) showed that finer grid will only provide more details in the vortex. Instead, the spike penetration position and its error relative to a benchmark solution on the finest mesh are more appropriate measures of the accuracy of the approximation.

In this instance, the benchmark considered is the solution on a 256×1024 grid. The error corresponds to the distance of the spike position relative to the benchmark solution.

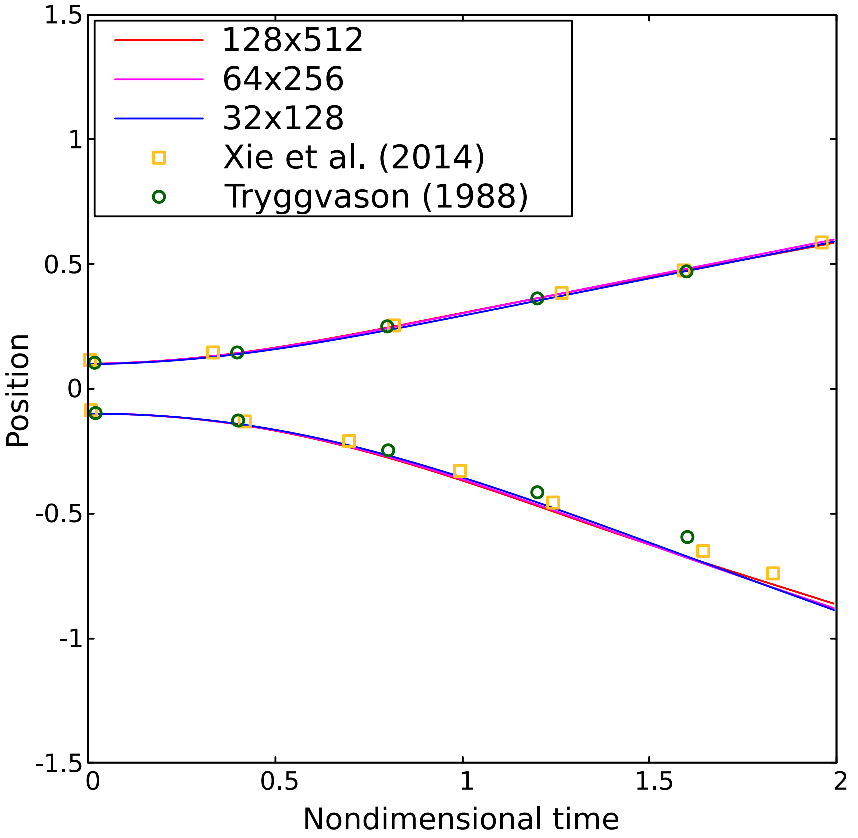
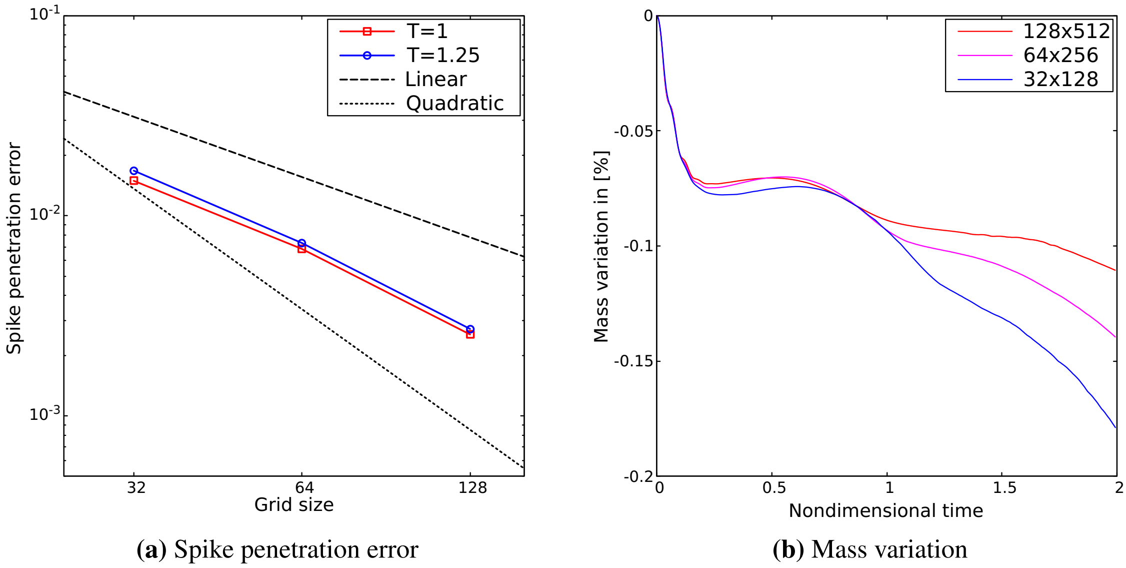


Fig. 5.17 Evolution of the position of the heavier and lighter fluids in nondimensional units.

The convergence is then evaluated for each grid. Fig. 5.18 highlights that near quadratic convergence is obtained. In addition, the mass loss during the evolution of the flow is kept to within 0.2%. A sudden loss of mass occurs at the beginning when the interface evolves slowly.

5.3.3 Kelvin-Helmholtz instability

This well-known instability is a phenomenon that occurs when fluids with different velocities interact. It is formed due to the shearing motion between fluids, leading to the formation of vortices. It is commonly observed in natural settings such as cloud formations, ocean currents, and atmospheric phenomena like jet streams. The domain of ratio 4 : 1 contains two fluids of the same densities and viscosities. Gravity is neglected. At the initial stage, the interface



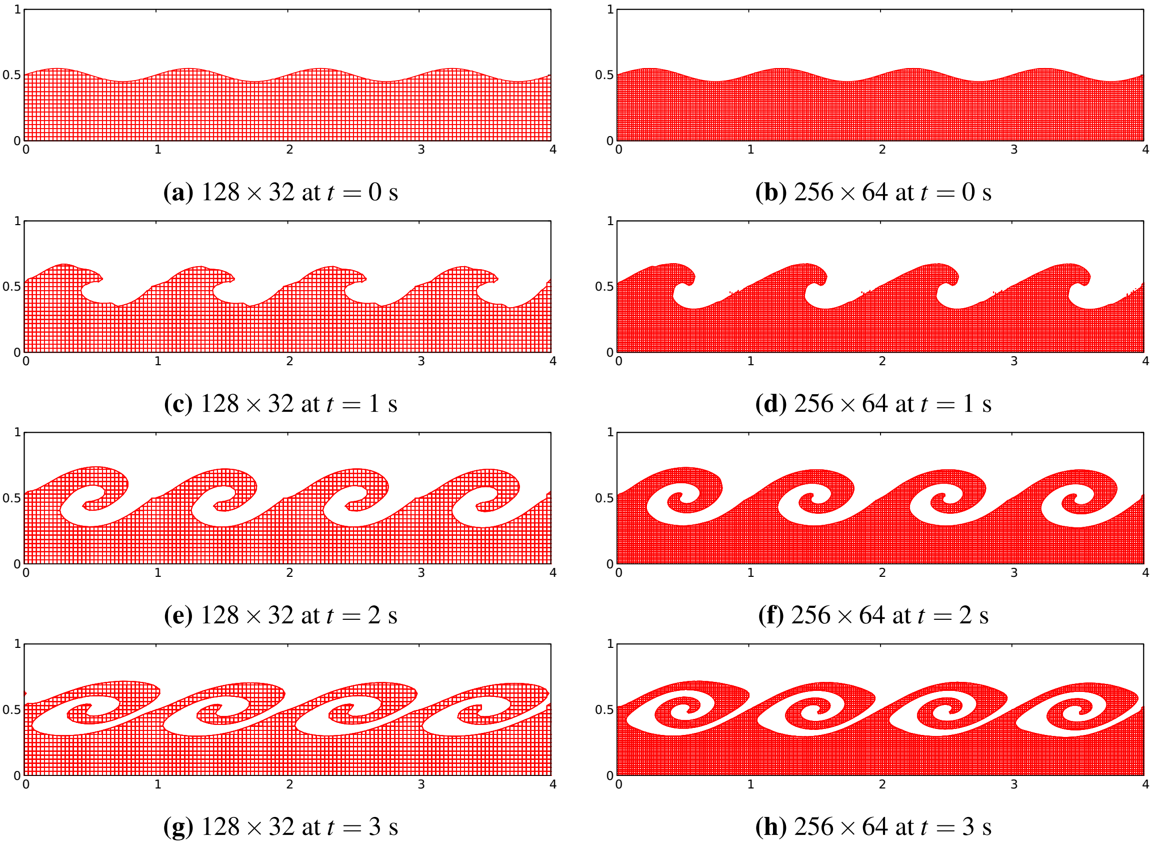
(a) Spike penetration error

(b) Mass variation

Fig. 5.18 Convergence study on the Rayleigh-Taylor instability problem, (a) spike penetration error, (b) mass variation (%).

between the two fluids takes the form of a sinusoidal function $0.5(1+0.01\sin(2\pi x))$. The fluid at the bottom has a constant velocity magnitude of 0.5 towards the left hand side of the domain. The fluid at the top has a constant velocity magnitude of 0.5 towards the right hand side of the domain in order to create the shear motion. Periodic boundary conditions are employed in the horizontal direction, while no-slip wall boundary conditions are imposed on top and bottom boundaries.

Numerical predictions for this test case are performed for two different grid sizes, respectively 128×32 and 256×64. Fig. 5.19 shows snapshots of the evolution of the interface at $t = 0$ s, $t = 1$ s, $t = 2$ s and $t = 3$ s, respectively, in physical time. The vortices are well reconstructed. Note that even for a coarse mesh, the gap between the two fluids remains larger than a cell size, which justifies the decision not to use a filament MOF method for this test case.



(a) 128 × 32 at $t = 0$ s

(b) 256 × 64 at $t = 0$ s

(c) 128 × 32 at $t = 1$ s

(d) 256 × 64 at $t = 1$ s

(e) 128 × 32 at $t = 2$ s

(f) 256 × 64 at $t = 2$ s

(g) 128 × 32 at $t = 3$ s

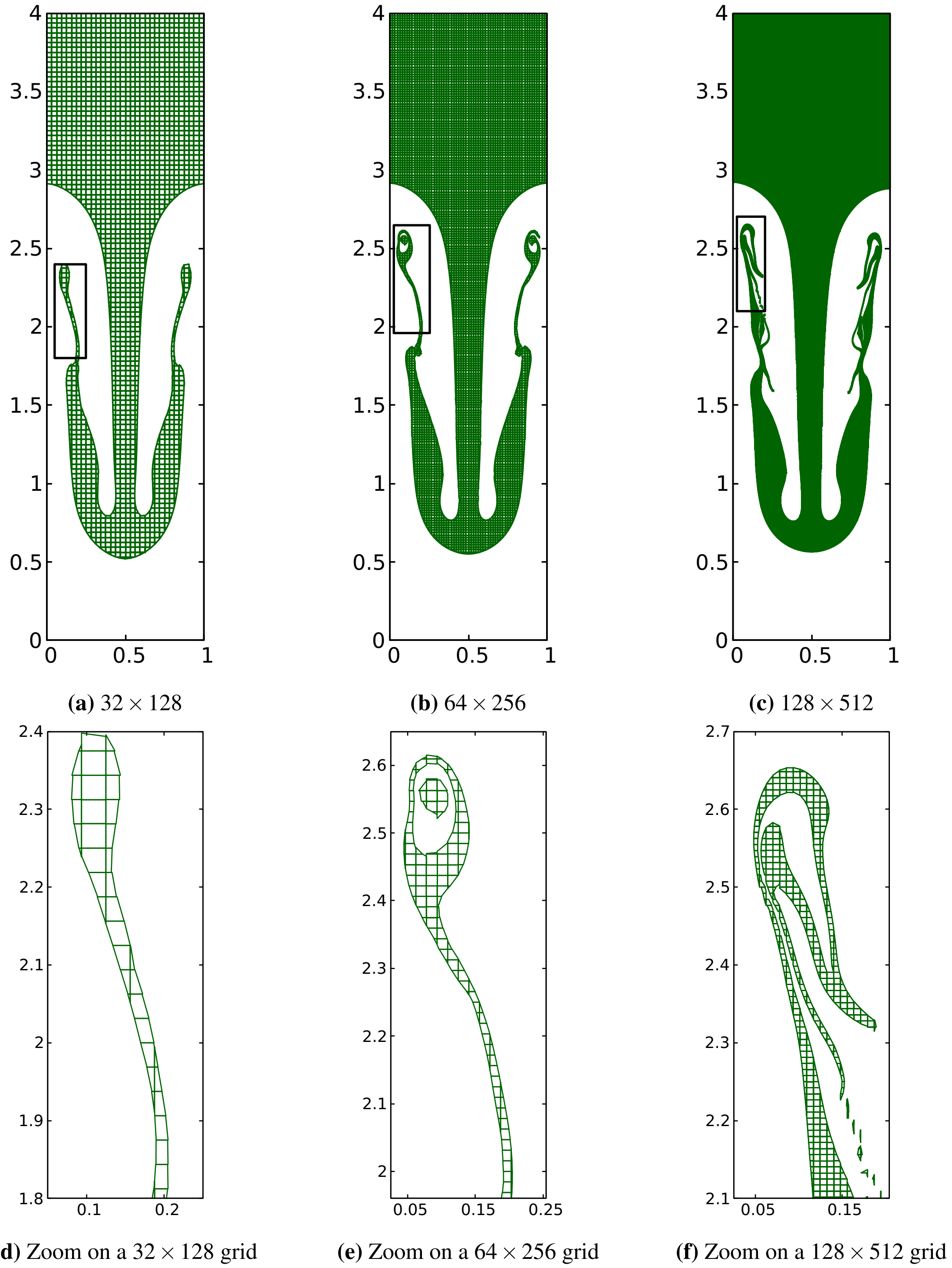
(h) 256 × 64 at $t = 3$ s

Fig. 5.19 Snapshots of the Kelvin-Helmholtz instability interface using the MOF method on 128 × 32 and 256 × 64 grids.

5.4 Filament MOF method coupled with Navier-Stokes equations

In this section, the coupling between a flow solver and the proposed filament MOF method, in scenarios where filaments are present, is discussed. While the previous section also addressed this coupling, the test cases did not involve filament structures. Resolving subgrid-scale structures for multiphase flows is challenging. It requires an extremely accurate method. The newly developed filament MOF method (Hergibo et al., 2023) possesses good reconstruction features mainly for filamentary flows. The Rayleigh-Taylor instability problem exhibits these characteristics. However, most numerical techniques for this benchmark problem are unable to resolve these filamentary structures and are limited to simulation times before they develop. On the one hand, finer grids are used to fully resolve the physical phenomena occurring in these instances. On the other hand, if coarser meshes are used, unphysical filament breakups would likely arise. In this section, some results, illustrating the coupling between the filament MOF method and the finite volume Navier-Stokes solver where filaments develop and are resolved, are presented.

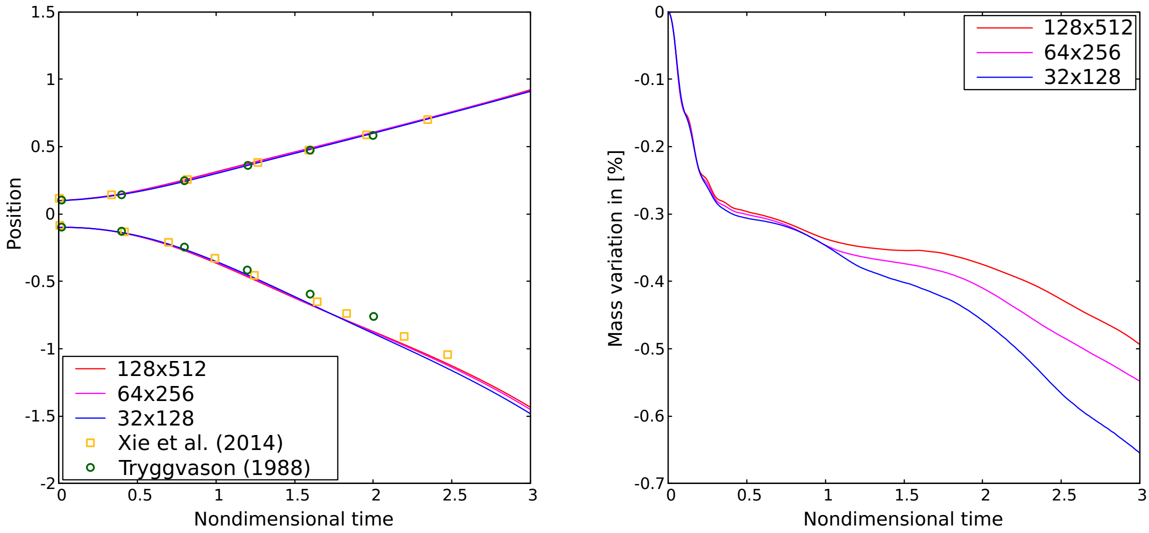
Fig. 5.20 shows snapshots of the Rayleigh-Taylor instability problem at later stages where filaments develop. These snapshots are taken at a nondimensional time of $T = 3$ for three different grids, 32×128, 64×256 and 128×512, respectively. The black rectangle denotes the area where a zoomed solution is provided. Note that the filament solution seems to be reconstructing filaments in a reasonable fashion. Filament breakups are reduced drastically, although they may still occur. In addition, at this later nondimensional time, here $T = 3$, the different grids do not seem to overlap, reducing any chance of grid convergence study. The finer grid shows an asymmetry in the lighter fluid position. This may be due to dynamics other than gravitational buoyancy occurring in the wake generation.



(a) 32×128 **(b)** 64×256 **(c)** 128×512

(d) Zoom on a 32×128 grid **(e)** Zoom on a 64×256 grid **(f)** Zoom on a 128×512 grid

Fig. 5.20 Snapshots of the Rayleigh-Taylor instability interface using the filament MOF method at $T = 3$ on different grids. Black rectangle highlights zoomed in areas on the top row. Bottom row shows zoomed in areas.



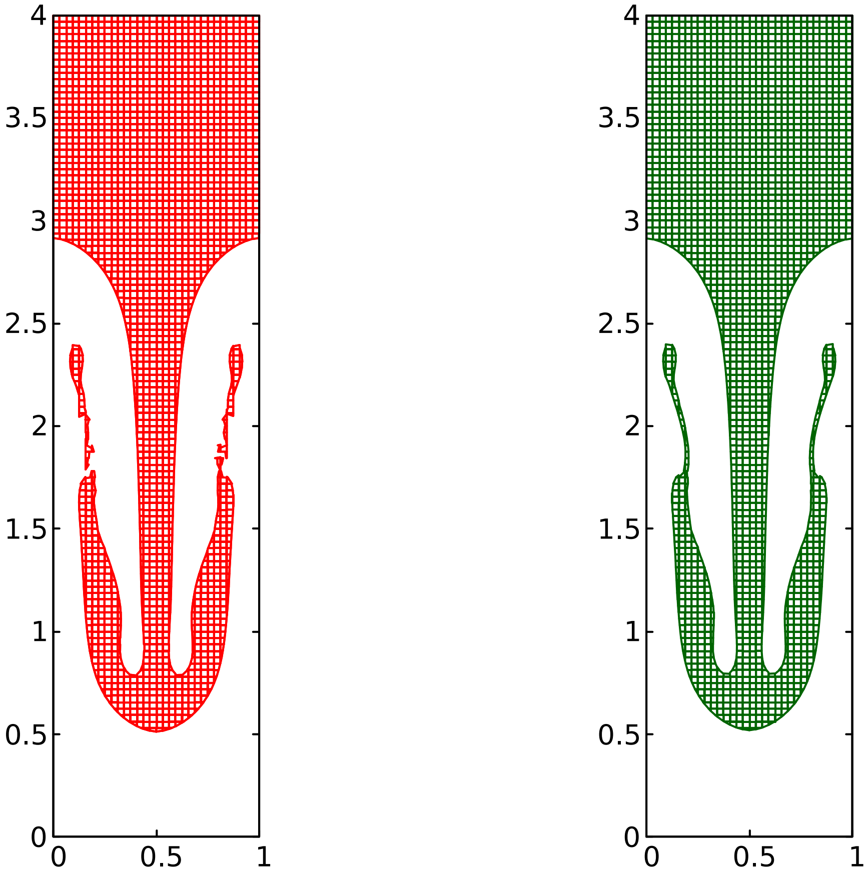
(a) Evolution of spike position compared to other methods

(b) Mass variation when using filament MOF

Fig. 5.21 Study on the Rayleigh-Taylor instability case using filament MOF. Spike penetration position and mass variation are presented.

However, the physical spike penetration position remains a good quantity to evaluate. Fig. 5.21 highlights the positions of the heavy and light fluid during the simulation. In this case, the nondimensional time has been extended to $T = 3$. Note the heavy fluid spike position continues to move downwards and even seems to accelerate at these later stages. To the best of the author's knowledge, no comparisons have been made for the physical spike position at later stages. For mass variation, it shows a similar trend to the standard MOF results shown above. A maximum of 0.654% mass loss is reached at $T = 3$ for the coarsest mesh 32×128.

As a comparison, Fig. 5.22 shows the exact comparison between the standard MOF and the filament MOF method. The author observes that despite the filaments not being resolved appropriately, the standard MOF cannot resolve any feature that is thinner than a cell size. The standard MOF simulation exhibits unphysical breakups. As subgrid-scale structures get resolved more accurately, effort needs to be made to ensure that the solution is physical and is validated with high-fidelity numerical simulations. Quantifying features of the mushroom shaped tail would be an interesting addition to the community.



(a) Standard MOF
Runtime: 27 s

(b) Filament MOF
Runtime: 38 s

Fig. 5.22 Comparison of the standard and filament MOF method for the Rayleigh-Taylor instability problem at $T = 3$ for a 32×128 grid and their respective runtimes.

Runtime is also compared between the standard and filament MOF methods. On a single core, the test case takes 27 s to run for a standard MOF simulation, whereas the filament MOF simulation takes 38 s. As expected, understanding the topology through the conglomeration algorithm and performing filament reconstruction is more expensive. However, a 40% increase in runtime is not significant when taking into consideration to the quality of the reconstruction. Once filaments break up, coalescence is not possible for this case. This leaves the user with a choice of balance between fast runtime and accuracy in reconstruction.

5.5 Closing remarks

In conclusion, this chapter has provided a detailed validation process for the moment-of-fluid (MOF) method, showcasing its efficacy in dealing with dynamic interface reconstruction. The successful validation process has demonstrated the reliability and accuracy of the MOF method in capturing sharp interfaces while considering subgrid structures. The filament MOF method has demonstrated its novel approach and capability in resolving multiphase flow problems.

The application of the MOF method to real-world cases has further demonstrated its versatility and practical relevance. The method robustly handles and accurately predicts fluid behaviour in diverse situations. The application of the MOF method to these cases allows the reader to gain insights and enhance understanding of the intricacies involved in complex multiphase flow dynamics. Its successful application to real-world problems demonstrates that it is a reliable and effective computational tool. This opens avenues for further exploration and utilisation in various engineering and scientific domains.

The next chapter will address the implementation of adaptivity within the MOF framework, exploring its capability and potential to enhance computational precision and efficiency. Through the incorporation of adaptive techniques, the adaptive MOF method will be validated. Complex real-world applications will be demonstrated using the novel MOF-ADG method.

Chapter 6

A MOF-adaptive dual grid method for multiphase flows: Benchmarks and Results

6.1 Introduction

The previous chapter has set the stage for subgrid structures or filament reconstruction. The quest for increased accuracy and efficiency remains crucial, in addition to capturing fine-scale features. One prominent technique is to use computational resources intelligently by refining grids when needed using mesh adaptivity.

This chapter explores the concept of adaptive mesh refinement for the moment-of-fluid (MOF-AMR) and its innovative application through the novel adaptive dual grid moment-of-fluid (MOF-ADG). This dual grid system allows dynamic adjustment of grid resolution solely on interfacial features. The primary focus of this chapter is on validating the MOF-AMR method using a selection of well-known benchmark problems and extending the capability of the complexity of the solver compared to a fixed grid setting. A rigorous assessment of the efficiency of the method is carried out, showcasing some promising results.

The integration of the adaptive dual grid technique with the moment-of-fluid method adds a layer of sophistication, presenting a promising capacity for overcoming computational challenges in complex scenarios. This chapter aims to unravel the potential of this adaptive approach, providing complex real-world cases for multiphase flow dynamics.

6.2 Validation: the MOF-AMR method

6.2.1 Error evaluation

Computing errors play an important part of interface capturing methods as it is the primary indicator of the effectiveness of a method. Comparing errors enables one to evaluate the merits of different methods. The L_1 error norm E_{L_1}, which is based on a volume fraction approach, is one of these numerical indicators. On a refined grid, the L_1 error is evaluated on the base mesh, which means that refined subcells are grouped together to form a single volume fraction on the base mesh. Using the data structure described in Section 4.2.1, the error can be evaluated using

$$E_{L_1} = \sum_{i,j} \left| F_{final}(i,j) - F_{initial}(i,j) \right| |\Omega|(0) \ , \tag{6.1}$$

where $|\Omega|(0)$ represents the cell area at level 0, $F_{final}(i,j)$ and $F_{initial}(i,j)$ are calculated in similar fashion:

$$F_{final}(i,j) = \sum_{lev} \sum_{is,js \subset \{\Phi_{lev}\}} \frac{F(i,j,is,js,lev)\,|\Omega|\,(lev)}{|\Omega|\,(0)} \ , \tag{6.2}$$

where F represents the volume fraction in a subcell and Φ_{lev} corresponds to the set including subsets Φ_0, Φ_1 and Φ_2, respectively. If a cell contains subcells at level 1 and level 2, then Eq. (6.2) aims at summing up their volume fractions with respect to their subcell grid size. If a cell has not been refined, then level 0 remains and $F_{final}(i,j)$ is the volume fraction at level 0. The relative error norm E_r is given by:

$$E_r = \frac{E_{L_1}}{\sum_{i,j} |F_{initial}(i,j)| \, |\Omega| \, (0)} \tag{6.3}$$

The symmetric error is another indicator of the error in reconstruction. This error indicator provides an estimation of the discrepancy in the area between the initial and final states. On a refined grid, the error is evaluated on any mixed subcells. Their sum forms the symmetric error E_{sym}, which is given by Equation (5.4).

Finally, the mass difference is also used as an indicator. Mass conservation is critical during dynamic cases. In this thesis, mass corresponds to the area encompassed within the original interface i.e.

$$\Delta m = \sum \left|F_{final}\right| |\Omega| \, (lev) - \sum |F_{initial}| \, |\Omega| \, (lev) \tag{6.4}$$

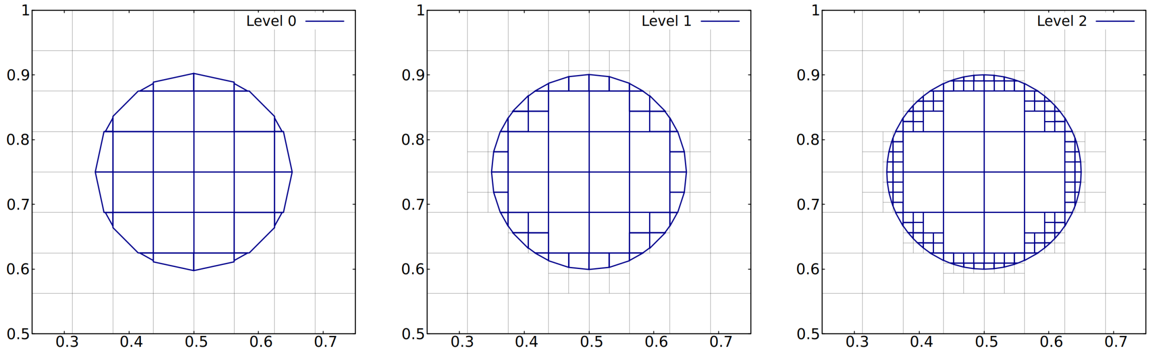
While the order of convergence is always calculated on a uniform mesh, an attempt at finding the pseudo-order of convergence of the mesh is presented here. Indeed, the order of convergence calculated with a uniform mesh is related to the mesh size but also to the ratio of number of cells between refinement levels. In that respect, the aim is to give a ratio of maximum number of grid cells. The pseudo-order of convergence OC is given by

$$OC = \log\left(\frac{E_1^{sym}}{E_2^{sym}}\right) / \log\left(\sqrt{\frac{n_2^{max}}{n_1^{max}}}\right) \ , \tag{6.5}$$

where E^{sym} corresponds to the symmetric difference error of a particular grid and n^{max} its maximum number of grid cells. Note that Eq. (6.5) is also valid for a uniform mesh and so the ratio of maximum number of cells in a constant environment gives the same order of convergence as for a uniform mesh.

6.2.2 Static reconstruction

Static reconstruction consists of reconstructing the interface of a material using the same AMR logic as that described in Section 4.2.1. The only variation is that no advection is necessary. The refinement procedure still applies and the refinement criterion remains. The intersection of a circle of radius $r = 0.15$ centred at $[0.5, 0.75]$ in a unit domain on a finer grid is determined using the exact interface rather than the material configuration at the previous time step. Exact mass conservation is achieved at all levels. Fig. 6.1 highlights the difference in precision during reconstruction when refining the interface using zero, one or two levels of refinement. The symmetric difference error gives a good insight into the increased precision and accuracy obtained when using a higher level of refinement.



(a) Zero refinement level; $E_{sym} = 1.04 \times 10^{-4}$
(b) One refinement level; $E_{sym} = 4.24 \times 10^{-5}$
(c) Two refinement levels; $E_{sym} = 1.29 \times 10^{-5}$

Fig. 6.1 Static reconstruction for a 16×16 base grid with zero, one and two levels of refinement and the associated symmetric difference error.

6.2.3 Benchmark: Zalesak slotted disc

This benchmark test case involves a slotted disc which is rotated anti-clockwise in a rigid body rotation around the centre of the domain (Zalesak, 1979). The circle of radius $r = 0.15$ has a rectangular slot of width $w_Z = 0.05$ in its centre with a maximum height of $h_Z = 0.85$. The velocity field for this test case is given by:

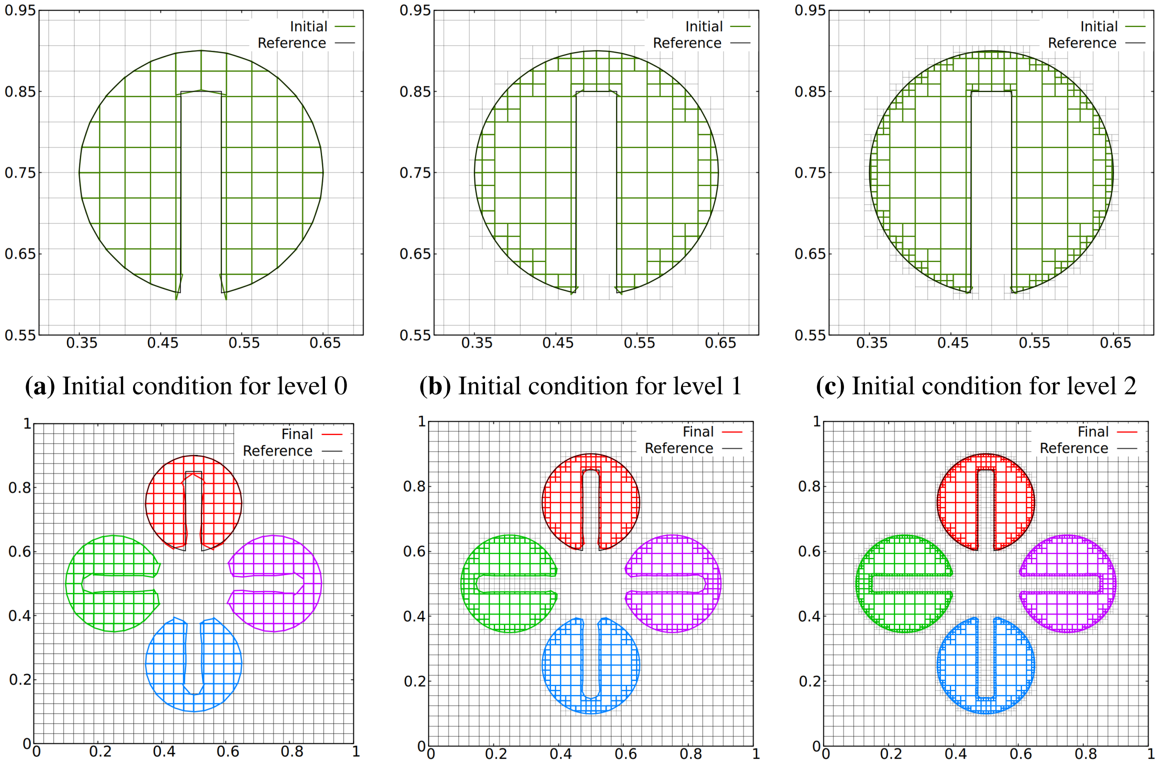
$$\mathbf{u}(x,y) = \begin{bmatrix} 0.5 - y \\ x - 0.5 \end{bmatrix} \tag{6.6}$$

Even though no filaments are formed during the advection process, the filament capability of the proposed code is still enabled. The rotational nature of this test case also highlights the fact that no deformation occurs in the material, hence the mass redistribution algorithms are enabled but not used as the backtrace is always of the same size as the cell area. This highlights the powerful choice of backtrace when refining a mesh as described in Section 4.2.3. Three different grids are presented, explicitly 32 × 32 as base mesh and a level 1 and level 2 of refinement. The number of iterations is $n_{it} = 300$ and $\Delta t = 2\pi / n_{it}$.

Table 6.1 Dependence of the L_1 error, E_{L_1}, and relative error, E_r, on refinement level for the Zalesak slotted disc problem using a 32 × 32 base mesh.

Refinement level	E_{L_1}	E_r
0	2.55×10^{-3}	4.38×10^{-2}
1	5.31×10^{-4}	9.13×10^{-3}
2	1.98×10^{-4}	3.41×10^{-3}

The error indicator used in this test case is the interpolated L_1 error. Table 6.1 presents the error for different levels of refinement. Fig. 6.2 shows the difference between initial and final reconstructions, as well as the intermediate reconstructions captured during the full rigid body rotation. The shape of the interface is maintained well, except around the sharp edges of the rectangular slot. The MOF method, as it stands, is not able to reconstruct these sharp edges even when refining the grid locally. Note, however, that the straight interfaces around the longer edges of the rectangle are not refined at the initial stage. Indeed, because MOF reconstructs the cells along these edges exactly, the refinement criterion is not triggered. The main difference compared with the method of Ahn and Shashkov (2009) is that the tolerance used in that paper is independent of the cell dimension. This means that, with a tolerance set to be smaller than machine precision, even cells that are reconstructed exactly will be refined. Fig. 6.3 which is taken from Ahn and Shashkov (2009) shows that the neighbourhoods of the slot are refined while it is not in the author's initial reconstruction.



(a) Initial condition for level 0

(b) Initial condition for level 1

(c) Initial condition for level 2

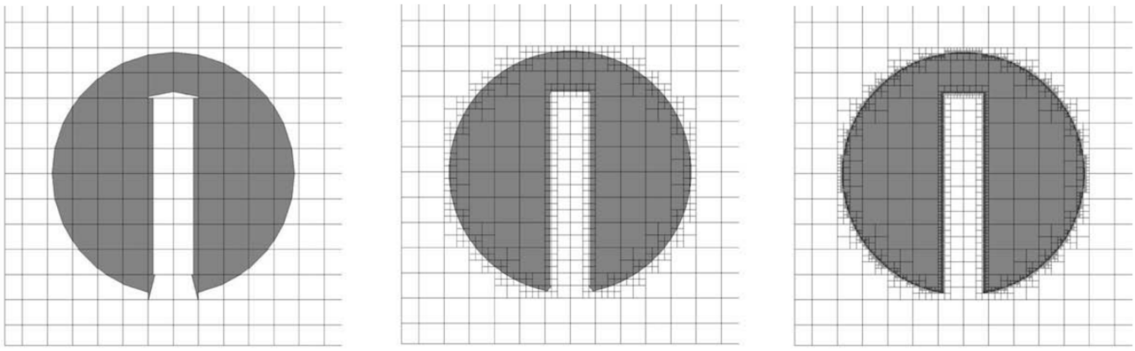
(d) Final solution for level 0

(e) Final solution for level 1

(f) Final solution for level 2

Fig. 6.2 Zalesak slotted disc test case for a 32 × 32 base grid and one and two levels of refinement. Top row of figures shows the initial reconstruction. Bottom row of figures shows the evolution of the shape of the interface.

The Zalesak slotted disc is also a good benchmark to evaluate the efficiency of the method through a quantitative analysis of the distribution of computational effort between different components of the method. This data is reported using the average percentage of time spent on each component per iteration. Five main blocks exist in this code, the first involves identifying level 0 cells that will need to be advected. This second and third blocks involves backtracking cells at any levels and also the intersection procedure. The final two blocks involve global mass redistribution and interface reconstruction. Fig. 6.4 highlights



(**a**) AMR up to level 0 (**b**) AMR up to level 2 (**c**) AMR up to level 4

Fig. 6.3 Figure taken from Ahn and Shashkov (2009) highlighting their initial reconstruction and choice of refinement for the Zalesak slotted disc.

the percentage of time taken in each block of the code for both a 32 × 32 and 64 × 64 grid. The second plot shows data for the same finest level of refinement. Note the percentage of time spent on the advection identification and the redistribution procedure subroutines is insignificant. Most of the time is spent in the intersection procedure as expected due to looping through all cells and subcells. As more levels are considered, the percentage of time increases in the intersection procedure. However, the percentage of time spent in reconstruction does not increase significantly because of the limited number of cells reaching higher levels. In addition, the likelihood of them not being filament reconstruction cells mitigates the computational cost.

6.2.4 Benchmark: Reversible vortex T=8

The reversible vortex is an advection benchmark that has been widely studied in the literature (Rider and Kothe, 1998). In this deformation problem, a circle of radius $r = 0.15$ located within a unit square domain centered at $[0.5, 0.75]$ is sheared by a divergence-free velocity field given by

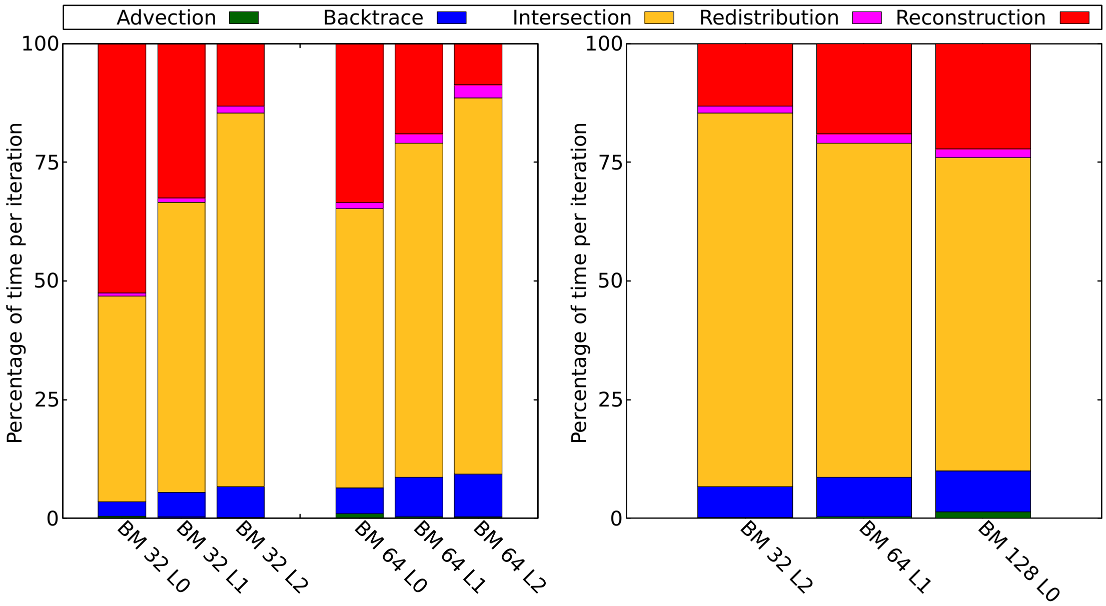
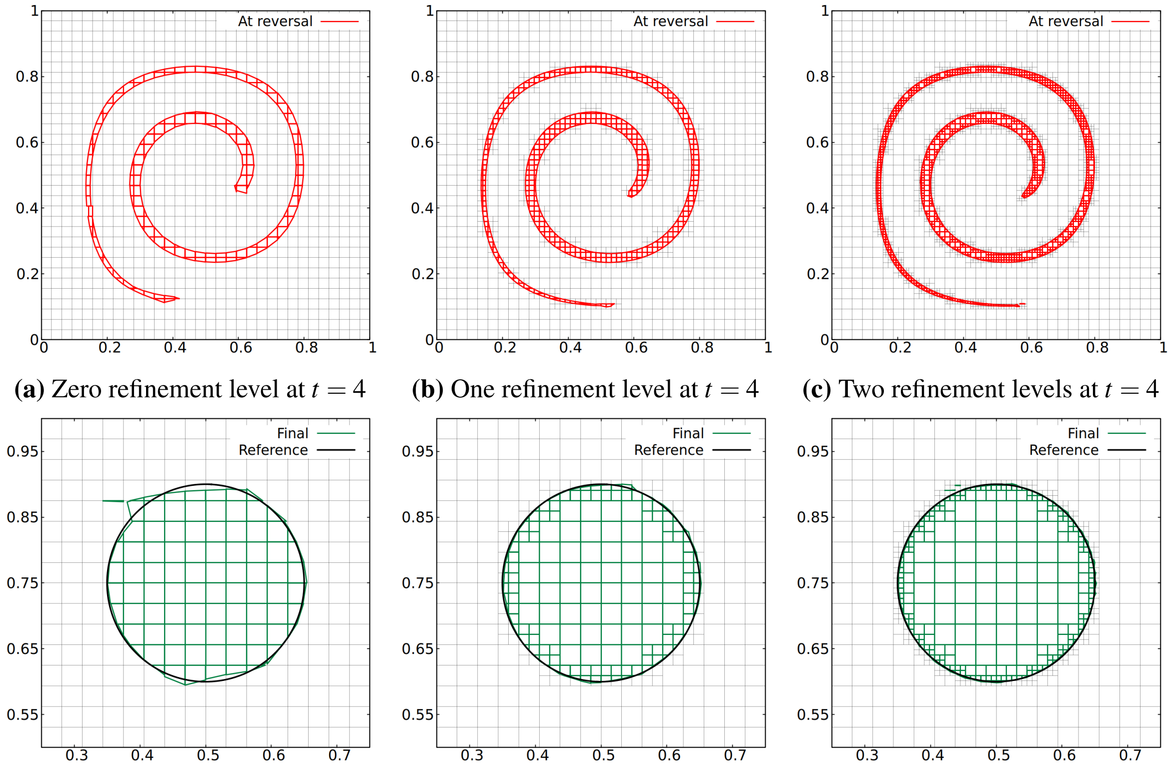


Fig. 6.4 Percentage of time spent on key MOF processes per iteration for a 32 × 32 and 64 × 64 grid for level 0, level 1 and level 2, respectively. Comparison of time distribution for same finest level of refinement.

$$\mathbf{u}(x,y,t) = \begin{bmatrix} -\sin^2(\pi x)\sin(2\pi y) \\ \sin^2(\pi y)\sin(2\pi x) \end{bmatrix} \cos(\pi t/T)\ , \tag{6.7}$$

in which T represents the full period. In most cases, $T = 8$ and in this case, the Courant-Friedrichs-Lewy (CFL) number is 1. In that respect, the number of iterations $n_{it} = 256$ and $\Delta t = \Delta x$. The structure of the deformed interface exhibits filaments which indicates that the filament procedure is activated within the proposed AMR scheme. Fig. 6.5 presents the results for a base mesh of 32 × 32 with 0, 1 and 2 levels of refinement. The maximum deformation at $t = T/2$ is shown, as well as the final state at $t = T$. Indeed, during the final state, the symmetric difference error can be used to compare with the initial reconstruction. It is important to note that during the refinement process, the local CFL number reaches 2 and 4, respectively, for refinement at level 1 and level 2. Mass difference and runtime are also explicitly displayed in Table 6.2.



(a) Zero refinement level at $t = 4$ **(b)** One refinement level at $t = 4$ **(c)** Two refinement levels at $t = 4$

(d) Zero refinement level at $t = 8$ **(e)** One refinement level at $t = 8$ **(f)** Two refinement levels at $t = 8$

Fig. 6.5 Reversible vortex test case using $T = 8$ for the base grid 32×32 with zero, one and two levels of refinement. Top row of figures shows the maximum deformation. Bottom row of figures shows the final interface.

Table 6.2 Symmetric difference error, order of convergence, mass difference and runtime for the reversible vortex test case at final reconstruction using a filamentary approach. The pseudo-order of convergence is given in parenthesis.

Refinement level	0	1	2
E_{sym}	3.05×10^{-3}	1.14×10^{-3}	8.93×10^{-4}
Order of convergence	-	1.41(4.25)	0.35(0.77)
Mass difference	3.3×10^{-15}	-6.7×10^{-15}	2.5×10^{-13}
Runtime (s)	15.7	35.2	92.2

The evolution of the number of cells is displayed in Fig. 6.6. As expected, level 0 offers a constant number of cells throughout the iterations, while the number of cells for level 1 and level 2 increase gradually until the vortex is reversed. Note the small drop in the number of cells in the final iteration before reversal. At this instant in time the magnitude of the velocity field vanishes which limits the error in reconstruction.

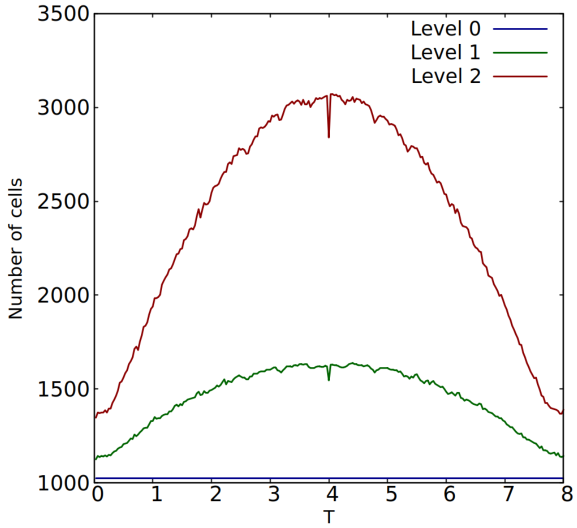


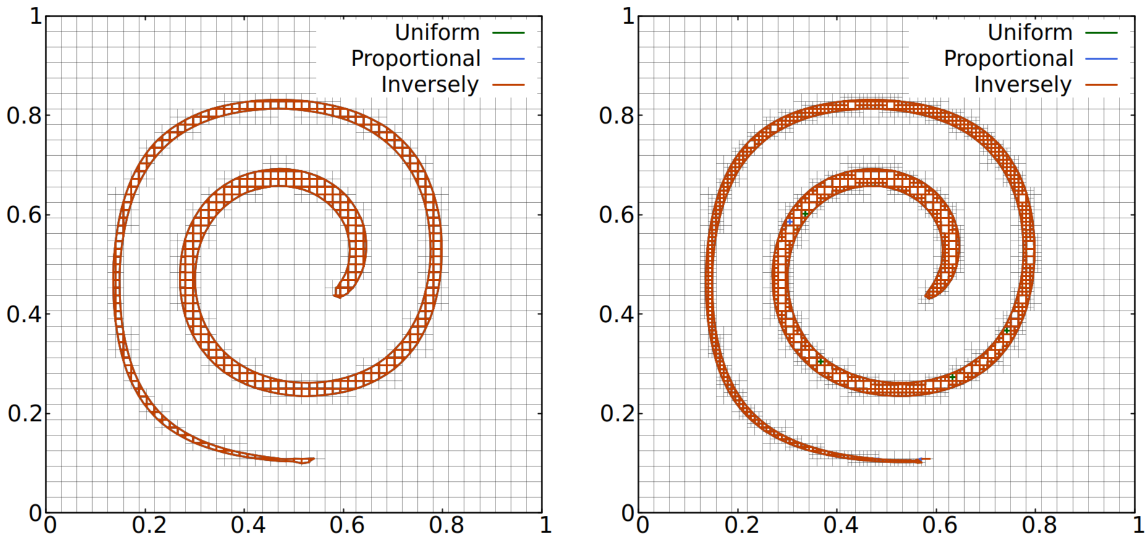
Fig. 6.6 Evolution of the number of cells when using different levels of refinement during the reversible vortex test case.

6.2.4.1 Influence of the mass redistribution procedure

In this section the influence of the mass redistribution procedure is examined. In most cases, mass is redistributed uniformly. However, as discussed in Section 4.3, directly proportional and inversely proportional redistributions are implemented and explored in this thesis. Fig. 6.7 shows the seemingly marginal differences between these approaches in terms of reconstruction. Runtimes are also comparable with a uniform distribution. However, in terms of mass conservation, machine precision is not achieved. The main difference lies in the way the redistribution of mass is achieved. While a directly proportional approach seems to be a natural way to follow, the number of iterations necessary to redistribute mass is increased compared to a uniform approach. Similarly, the inversely proportional approach iterates more times without increasing the runtime significantly.

6.2.4.2 Influence of the initial refinement

The initial reconstruction is the lower limit of error possible when reconstructing the interface. Indeed, it may differ between the initial reconstruction and the dynamic case. In general,

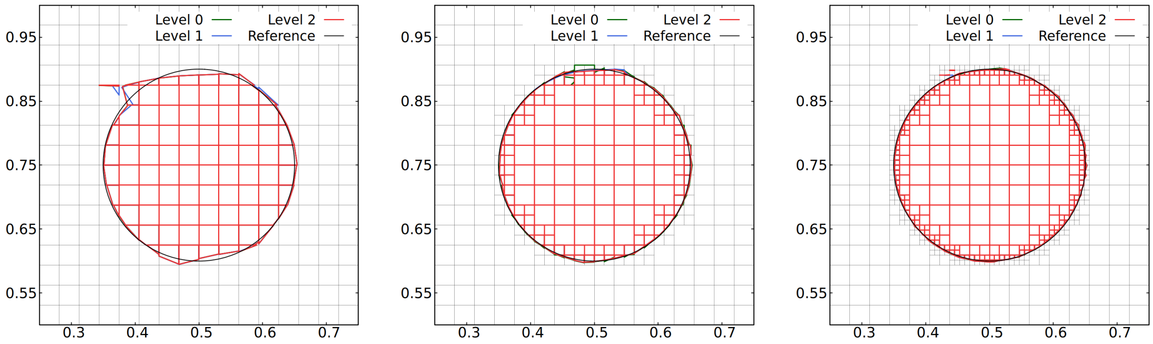


(a) One level of refinement

(b) Two levels of refinement

Fig. 6.7 Visual comparison between a uniform, directly proportional and inversely proportional mass redistribution at maximum deformation.

when using a refinement structure, the initial refinement is the same as the advection process. This is the case for all other cases in this present study. Hence, this section assesses the influence of the initial refinement on the final reconstruction. In the following case, the circle is reconstructed using different levels of refinement at the initial stage, then advected using either level 0, level 1 or level 2. Fig. 6.8 shows the final reconstruction for different levels of refinement at the initial stage.



(a) Use of level 0 for advection

(b) Use of level 1 for advection

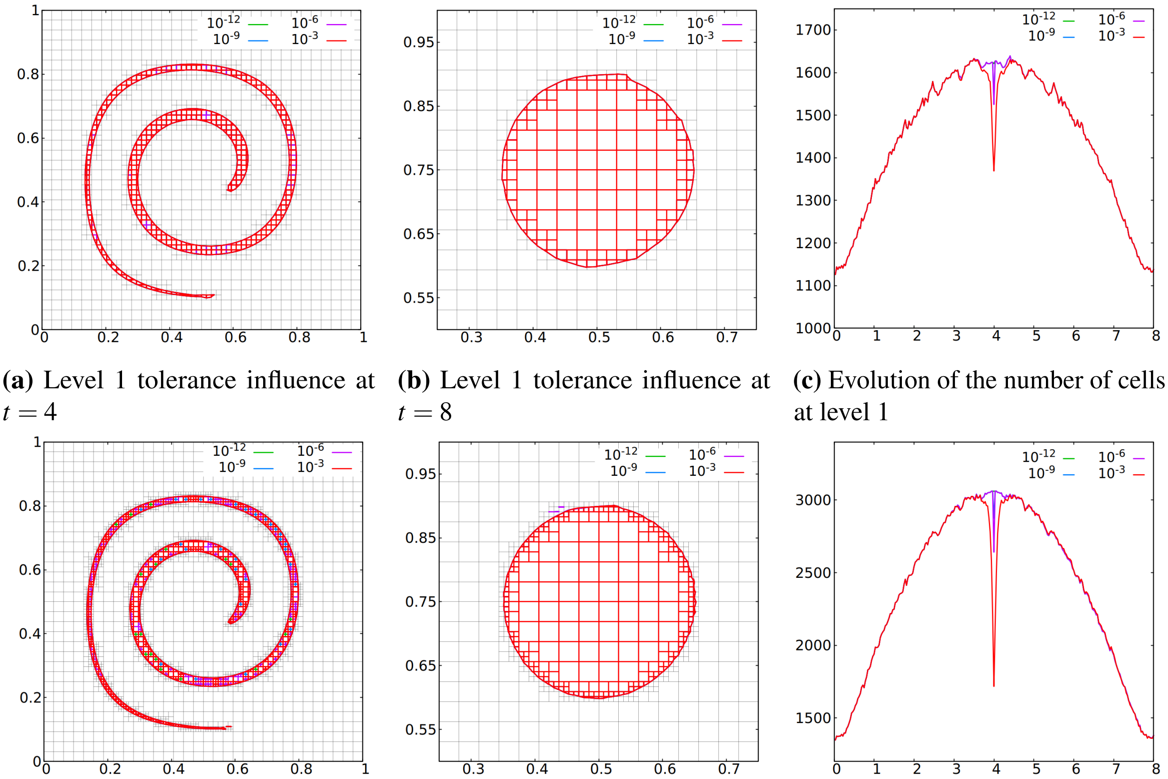
(c) Use of level 2 for advection

Fig. 6.8 Final reconstruction for the reversible vortex test case using $T = 8$ for the base grid 32×32, one and two levels of refinement. Levels indicate the level of refinement at the initial stage.

A slightly adapted data structure is used to accommodate the correct segmentation. In the proposed code, (i, j, is, js, lev) is the data structure used for adaptive mesh refinement. However, space allocation is performed at the start using the desired maximum refinement level div_max, i.e. *is* has an allocation of 2^{div_max} and so has js. Note that this allocation would not work if the maximum level was 0 at the initial stage but then 2 during the advection process as the allocation would not be performed. This allows the user to use any initial condition in terms of refinement levels as one can see in Fig. 6.1. Note that the final state is not highly dependent on the initial level of refinement.

6.2.4.3 Influence of the mesh refinement criteria

As described above the refinement criteria in a MOF framework is the discrepancy between the reference and reconstructed centroid. This section discusses the influence of having a finer or coarser criterion. Note that the criterion is nondimensionalised by the cell size so that it is more meaningful than using machine precision. A lower tolerance has a great influence on the reconstruction precision, but it also has implications on the number of cells in the domain and indeed the runtime. In setting a suitable tolerance, one has to consider the trade-off between accuracy and runtime. Runtime is comparable for all test cases and a significant difference is not found. Fig. 6.9 shows the intermediate and final reconstruction as well as the evolution of the number of cells in the domain. The maximum number of cells is also comparable, however the evolution shows an interesting feature where the last iteration before reversal exhibits a large drop in the number of cells. Indeed, the last iteration corresponds to the $\cos(\pi t/T)$ term vanishing, meaning the reconstruction is an almost-static reconstruction. The level 0 advection is able to reconstruct more filaments. Combined with a low tolerance, the number of cells in the domain decreases significantly.



(a) Level 1 tolerance influence at $t = 4$

(b) Level 1 tolerance influence at $t = 8$

(c) Evolution of the number of cells at level 1

(d) Level 2 tolerance influence at $t = 4$

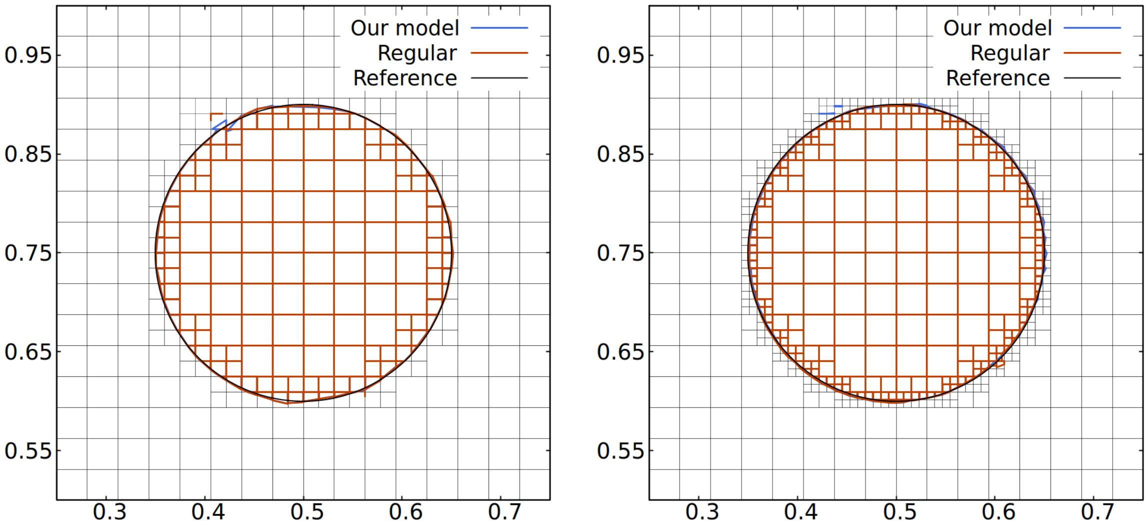
(e) Level 2 tolerance influence at $t = 8$

(f) Evolution of the number of cells at level 2

Fig. 6.9 Influence of the mesh refinement criterion tolerance on intermediate and final reconstruction and evolution of the number of cells in the domain.

6.2.4.4 Influence of the backtrace on interface reconstruction

The choice of backtrace within a refinement framework can influence results greatly. Indeed, the natural choice is to perform backtracking on the subcell itself, ascribed here as *regular*. However, there are some advantages and disadvantages which are explained below. On the one hand, the intersection procedure of the author's approach must intersect the entirety of the desired material at all times. In this regard, the backtracking approach is to use level 0 as reference and make sure that all refined levels intersect the same area as previous levels. This ensures exact mass conservation. However, the refined backtrace subcells are slightly deformed, which means the reference volume fraction and centroid are somewhat distorted. On the other hand, the regular backtrace creates gaps and overlaps that are very small (Ahn and Shashkov, 2009). This does not guarantee a full intersection of the material, leading to poor conservation of mass. Despite this loss of mass, the interface reconstruction is not distorted which may indicate a smoother interface reconstruction. The correct backtracking consisting of advected hanging nodes may also create non-convex cells, which means more complex algorithms are needed. This approach has been discarded. Fig. 6.10 emphasises the difference between a regular backtrace and the proposal for one and two levels of refinement.



(a) One level of refinement

(b) Two levels of refinement

Fig. 6.10 Comparison between a regular backtrace and the author's choice of backtracking a refined subcell at the final state.

Table 6.3 shows a comparison of error, mass difference and runtime between the proposed model and a regular backtrace method. Although the symmetric difference error is smaller when using a regular backtrace, this may not guarantee good mass conservation. In addition, the regular backtrace choice seems to be computationally faster. Indeed, the backtracking procedure is only relevant for the subcell itself, whereas in the proposed model, higher levels need to account for previous levels of refinement for its backtracking procedure. For this case, mass conservation approaches machine precision, which is an essential quantity to conserve in numerical multiphase flows. The proposed model compares well with other AMR-MOF methods such as Ahn and Shashkov (2009). Moreover, it shows a good balance between runtime and mass conservation while maintaining an acceptable error in reconstruction.

Table 6.3 Comparison between the proposed model and a regular backtrace regarding interface reconstruction.

	The proposed model	**Regular backtrace**
	Level 1	
E_{sym}	1.14×10^{-3}	7.84×10^{-4}
Mass difference	-6.7×10^{-15}	9.8×10^{-5}
Runtime (s)	35.3	31.0
	Level 2	
E_{sym}	8.93×10^{-4}	6.26×10^{-4}
Mass difference	2.5×10^{-13}	-2.6×10^{-4}
Runtime (s)	92.2	75.2

6.2.5 Benchmark: Droplet flow

Originally proposed by Ahn and Shashkov (2009) and further developed by Jemison et al. (2015), the droplet flow test case deforms an initial circle of radius $r = 0.125$ centred in a unit square domain using a nonlinear divergence-free velocity field given by

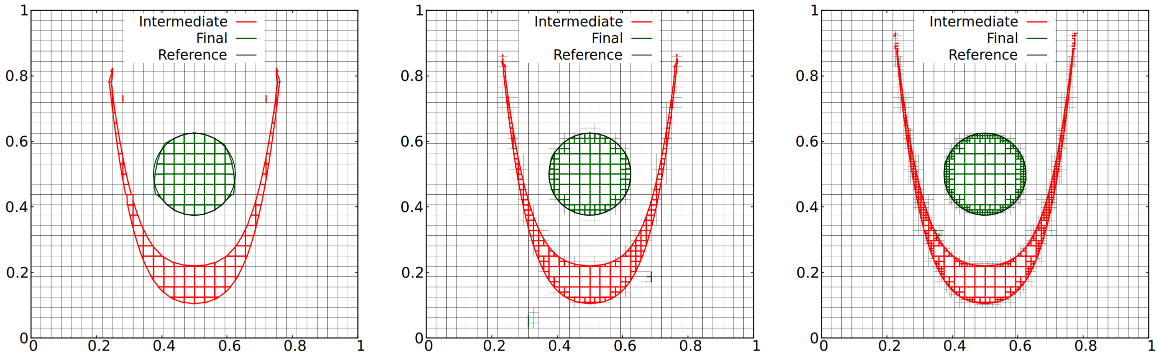
$$\mathbf{u}(x,y,t) = \begin{bmatrix} 0.125(8x-4) \\ 0.125\left[-(8y-4)-4-\left(1-(8x-4)^2-(8x-4)^4\right)\right] \end{bmatrix} f(t)\ , \tag{6.8}$$

where

$$f(t) = \begin{cases} 1 & 0 \le t < T_{max} - t_\varepsilon/2 \\ \cos\left(\frac{\pi(t-T_{max}+t_\varepsilon/2)}{t_\varepsilon}\right) & T_{max}-t_\varepsilon/2 \le t \le T_{max}+t_\varepsilon/2 \\ -1 & T_{max}+t_\varepsilon/2 < t \le 2T_{max}\ , \end{cases} \tag{6.9}$$

represents the amplitude of the velocity field which varies in time so that at time $t = T_{max}$ the initial droplet is recovered to its original position.

Filaments are formed during the advection process. A leading tip is generated, making this case challenging. The base mesh is 32 × 32, the number of iterations is $n_{it} = 160$ and $\Delta t = 0.01$. Two levels of refinement are tested. Fig. 6.11 shows the shape of the interface at the intermediate time $t = T_{max}$ and at the final time $t = 2T_{max}$ using different levels of refinement. All figures show very good agreement compared with the original circle. In addition, filaments are well reconstructed except when the tip needs to be reconstructed using refinements. This tends to lead to spurious breakups in the material.



(a) Zero refinement level **(b)** One refinement level **(c)** Two refinement levels

Fig. 6.11 Droplet flow test case for a 32 × 32 base grid with zero, one and two levels of refinement: maximum deformation and location of the interface at final time.

Table 6.4 Symmetric difference error, mass difference and runtime for the droplet flow test case at final reconstruction using a 32 × 32 base mesh compared to reference solutions.

Refinement level	0	1	2
E_{sym} in Jemison et al. (2015)	2.48×10^{-3}	6.37×10^{-4}	2.96×10^{-4}
Order of convergence	-	1.96	1.10
Runtime (s)	191.3	529.3	940.4
E_{sym}	1.53×10^{-3}	2.55×10^{-4}	1.90×10^{-4}
Mass difference	-2.82×10^{-4}	-4.78×10^{-7}	-4.92×10^{-16}
Runtime (s)	2.9	7.8	21.2

Table 6.4 provides the information on the symmetric difference error, mass difference and runtime. Note that the mass difference is not as accurate as expected. Indeed, for the coarser refinement, some material tends to leave the domain near the bottom edge. The

level 1 figure shows that some material at the final state was advected very near the edge of domain, suggesting that for levels 0 and 1, some has left the domain. This highlights a major drawback of the author's approach. Indeed, when using a level 0 advection scheme combined with filaments, material that breaks away from the main material tends to stay detached, or is reconstructed poorly even when using some levels of refinement.

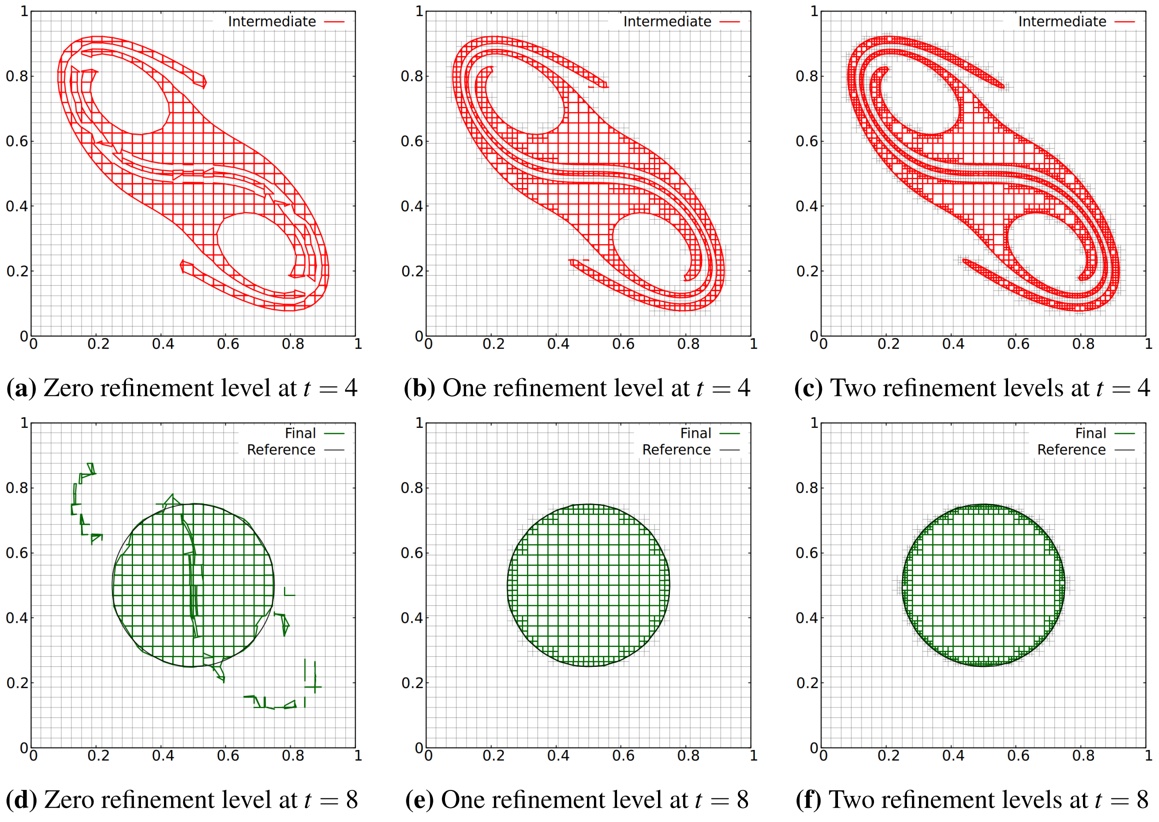
6.2.6 Benchmark: S-shape

First tested by Ahn and Shashkov (2009) and Jemison et al. (2015), the S-shape benchmark case is a challenging material deformation test where an initial circle of radius $r = 0.25$ centred in a unit square domain is deformed in a nonlinear divergence-free velocity field given by

$$\mathbf{u}(x,y,t) = \begin{bmatrix} 0.25[(4x-2)+(4y-2)^3] \\ -0.25[(4y-2)+(4x-2)^3] \end{bmatrix} f(t) \,, \tag{6.10}$$

where $f(t)$ is given in Eq. (6.9). In this case, $T_{max} = 4$ and $t_\varepsilon = 2$. The total number of iterations for a base mesh 32×32 is $n_{it} = 320$ and $\Delta t = 0.025$.

The deformation creates a highly deformed material creating thin filamentary structures in the centre of the domain. For this benchmark, the proposed filament capable MOF procedure is used. Fig. 6.12 shows the maximum deformation of the material and its final state. The level 0 grid shows poor reconstruction because the thin strand of material in the centre of the domain is difficult to reconstruct even with a filament approach using three conglomerates. When more than three conglomerates exist, a standard MOF reconstruction is used which tends to merge materials together (Hergibo et al., 2023). Table 6.5 provides the information on the symmetric difference error, mass difference and runtime. Using one or two levels of refinement exhibits a better reconstructed interface than Jemison et al. (2015). However, mass conservation is not well maintained for this challenging case due to the reversal of a large portion of thin filamentary structures. The significant loss of mass affects the symmetric difference error at level 2, which is larger than the reconstruction at level 1.



(a) Zero refinement level at $t = 4$ **(b)** One refinement level at $t = 4$ **(c)** Two refinement levels at $t = 4$

(d) Zero refinement level at $t = 8$ **(e)** One refinement level at $t = 8$ **(f)** Two refinement levels at $t = 8$

Fig. 6.12 S-shape test case for a 32 × 32 base grid with zero, one and two levels of refinement: (a)-(c) interface at maximum deformation; (d)-(f) final interface.

Table 6.5 Symmetric difference error, mass difference and runtime for the S-shape test case at final reconstruction compared to Jemison et al. (2015).

Refinement level	0	1	2
E_{sym} in Jemison et al. (2015)	2.11×10^{-2}	1.34×10^{-3}	4.74×10^{-4}
Runtime (s)	157.2	773.1	1871.5
E_{sym}	1.57×10^{-2}	1.11×10^{-3}	1.41×10^{-3}
Mass difference	-3.47×10^{-10}	-2.44×10^{-4}	3.30×10^{-3}
Runtime (s)	35.9	61.7	180.7

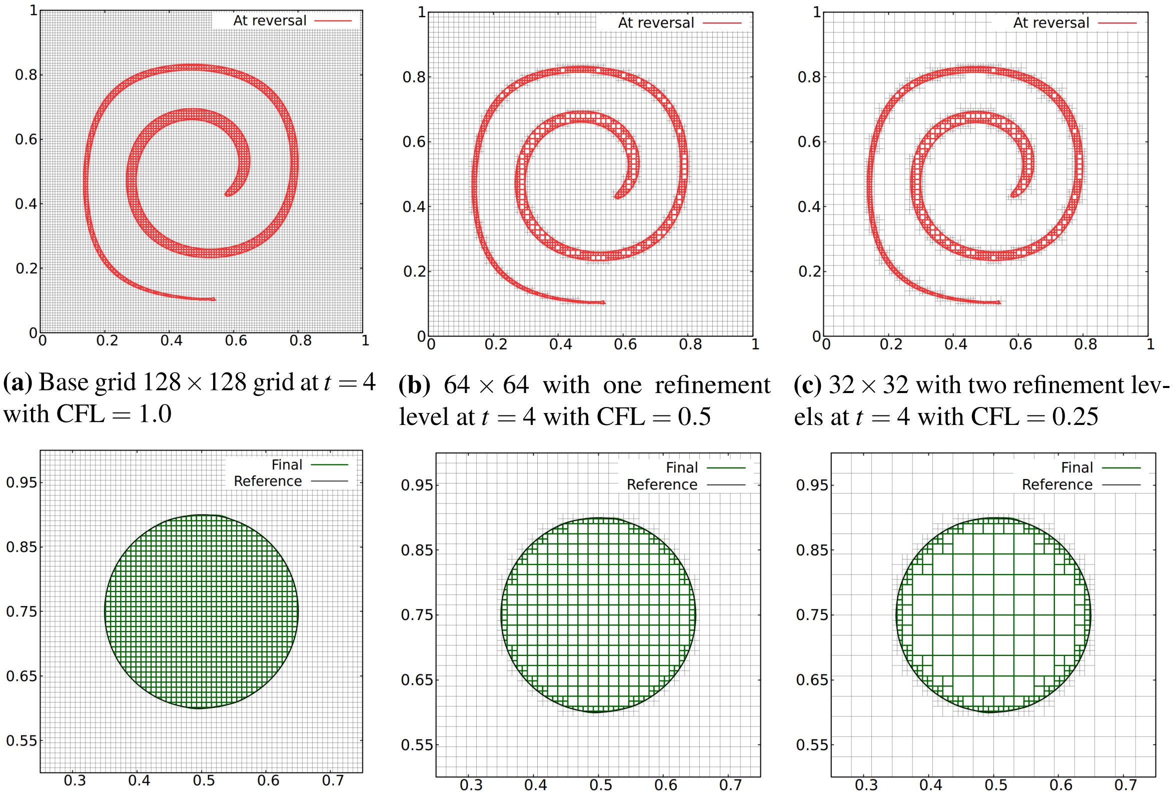
6.3 Discussion on the efficiency of MOF-AMR filament capability

Any AMR framework is known to use a reasonable trade-off between accuracy and runtime, refining regions of interest while decreasing the total number of cells used in computation compared to a uniform grid. In general AMR practices, runtime increases with refinement levels while the error decreases (or the region of interest becomes more accurately defined). However, in the MOF context, regions of high deformation can be reconstructed with ease using filaments while maintaining a reasonable computational cost. In this regard, one can try to compare the efficiency of different levels of a MOF-AMR filament capable procedure. Indeed, a filament reconstruction with a higher base resolution but with a lower level of refinement may be equivalent to a lower base resolution reconstruction but with a higher level of refinement. This section tries to give an insight into compromising runtime and error for the well-known reversible vortex benchmark. At first the author uses a constant unity CFL number on the base mesh, meaning the local CFL number for refined grids is 2 and 4, respectively, for level 1 and level 2. Secondly, the author considers an effective CFL number for the finest resolution meaning that the number of iterations is constant for all three configurations. The base mesh CFL number for one level of refinement is 0.5 and for two levels of refinement 0.25, which is effectively half of the time step for each refinement level simulations.

One can see from Table 6.6 that with a constant CFL number, runtime is better for one level of refinement, which is also better than two levels of refinement. This is due to the fact that level 1 has to be reconstructed first. In addition, the number of cells used is very small compared to a uniform mesh even with the highest refinement levels. When using the same effective CFL number, i.e. equivalent at the finest resolution, runtime increases significantly with the increased number of iterations. Similarly, the symmetric difference error increases.

Table 6.6 Efficiency table comparing computational accuracy and performance on three different grids with the same maximum level of refinement. BM 128 relates to Base Mesh and its resolution. CFL numbers are expressed for the base mesh.

	BM 128 Level 0	**BM 64 Level 1**	**BM 32 Level 2**
CFL	1.0	1.0	1.0
E_{sym}	1.56×10^{-4}	2.36×10^{-4}	8.93×10^{-4}
Max number of cells	16384	5530	3061
Number of iterations	1024	512	256
Runtime (s)	115.3	81.2	92.9
CFL	1.0	0.5	0.25
E_{sym}	1.56×10^{-4}	1.65×10^{-4}	2.42×10^{-4}
Runtime (s)	115.3	165.5	296.9



(**a**) Base grid 128×128 grid at $t = 4$ with CFL $= 1.0$

(**b**) 64×64 with one refinement level at $t = 4$ with CFL $= 0.5$

(**c**) 32×32 with two refinement levels at $t = 4$ with CFL $= 0.25$

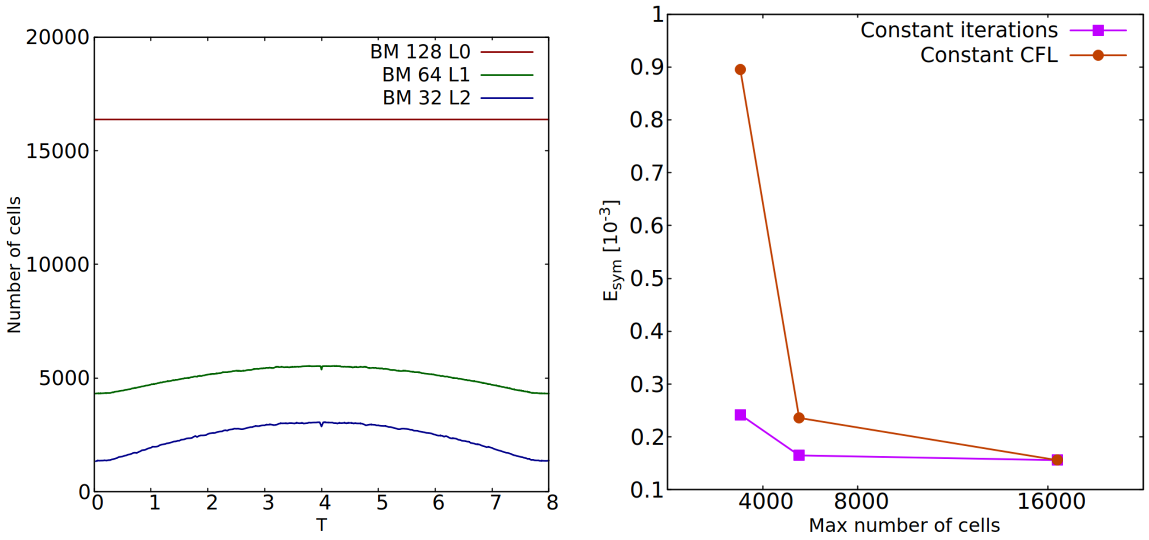
(**d**) Base grid 128×128 grid at $t = 8$ with CFL $= 1.0$

(**e**) 64×64 with one refinement level at $t = 8$ with CFL $= 0.5$

(**f**) 32×32 with two refinement levels at $t = 8$ with CFL $= 0.25$

Fig. 6.13 Visual results of the efficiency test of the MOF-AMR filament capable procedure using different CFL numbers and therefore a constant number of iterations.

Fig. 6.13 shows the improved final reconstruction. Comparison of performance on a fine uniform grid and a grid using one level of refinement, both using filament capable methods, shows that there is a significant improvement in runtime and a decrease in the number of cells for the latter while the error is very similar in both cases. This choice of base mesh and level of refinement may be a more desirable option. Fig. 6.14 shows that this approach uses a significantly smaller number of cells. In addition, much better reconstruction is achieved with a smaller CFL number. The influence of high CFL numbers (> 2) on interface accuracy has not been demonstrated.



(a) Evolution of number of cells during the reversible vortex test case

(b) Symmetric difference error as a function of max number of cells

Fig. 6.14 (a) Evolution of the number of cells and (b) dependence of the symmetric difference error on the maximum number of cells, for the efficiency test.

6.4 Application: the MOF-ADG method for multiphase flows

The MOF-ADG method represents a cutting-edge computational approach for modelling multiphase flows with great accuracy and efficiency. By integrating the moment-of-fluid (MOF) technique with adaptive dual grid (ADG), this method excels in capturing complex fluid dynamics phenomena. Its versatility in simulating diverse multiphase flow benchmark problems across various applications is demonstrated in this section.

6.4.1 Sloshing

6.4.1.1 Free sloshing of inviscid fluids

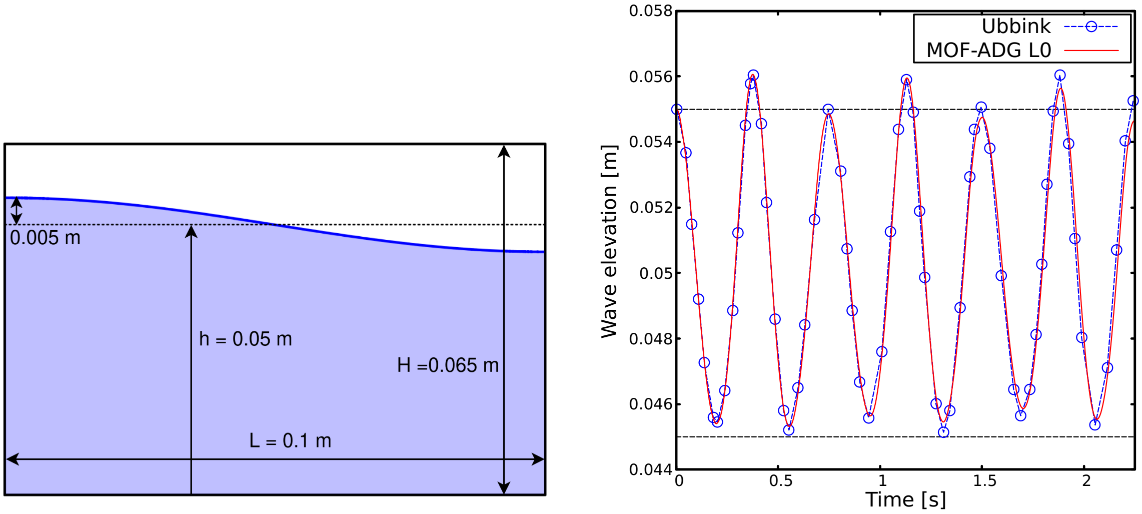
In this section, sloshing, the study of uncontrolled oscillatory motion of a liquid within a container, is examined. Typically caused by external forces or perturbations, complex topological change is created in the free surface. First, inviscid sloshing is considered, where no viscosity is applied resulting in no damping of the fluid motion.

For the inviscid sloshing case, the author replicates the configuration used by Ubbink (1997) who implemented an algebraic VOF method. A rectangular tank has length $L = 0.1$ m and height $H = 0.065$ m. The initial wave is a half cosine of amplitude 0.005 m with the still water level at rest at $h = 0.05$ m. Fig. 6.15(a) shows the domain and initial condition with a half cosine wave. The ratio of densities of the fluids is 1000 and both viscosities are set to 0. Similar to previous cases in earlier chapter, slip conditions are imposed on the left and right walls to enforce no-penetration while permitting fluid movement along the wall. No-slip conditions are applied on the top and bottom boundaries. The reader is guided towards the illustration of Fig. 3.12.

At the initial stage, the liquid exhibits a half wave length perturbation. As time progresses, characteristic wave patterns are formed. In this case, the period is given by

$$P = \frac{2\pi}{\sqrt{gk\tanh(kh)}} , \tag{6.11}$$

where $g = 9.8$ m/s^2 is the acceleration due to gravity, h is the still water height and k the wave number defined by $k = \pi/L$. Fig. 6.15(b) displays the wave elevation at the left boundary in physical units for the initial six periods only using a level 0 grid. The waves are not



(a) Schematic of the domain and initial condition

(b) Evolution of the wave elevation

Fig. 6.15 Inviscid sloshing case: (a) schematic of the initial conditions, (b) evolution of the wave elevation compared with the predictions of Ubbink (1997).

damped as this case is inviscid. The frequency aligns with the theoretical frequency. The wave elevation exhibits different behaviour for even and odd periods, related to wave modes. In both cases the amplitudes match the predictions of Ubbink (1997).

6.4.1.2 Free sloshing of viscous fluids

The second part considers the influence of viscosity on the sloshing motion by varying the Reynolds number, resulting in damping of the wave elevation over time to different degrees. In this instance, comparisons with the predictions of Liu and Lin (2008) are made. In this setting the dimensions of the container and the initial condition are changed. The tank has length of $L = 1$ m and height of $H = 0.65$ m. The initial wave is a full wavelength and its amplitude is set to 0.01 m, while the water depth is maintained around 0.5 m. Fig. 6.16 shows the computational domain and initial condition.

Two scenarios are studied: $Re = 20$ and $Re = 200$, where Re is defined by

$$Re = \frac{\rho h \sqrt{gh}}{\mu}\,, \tag{6.12}$$

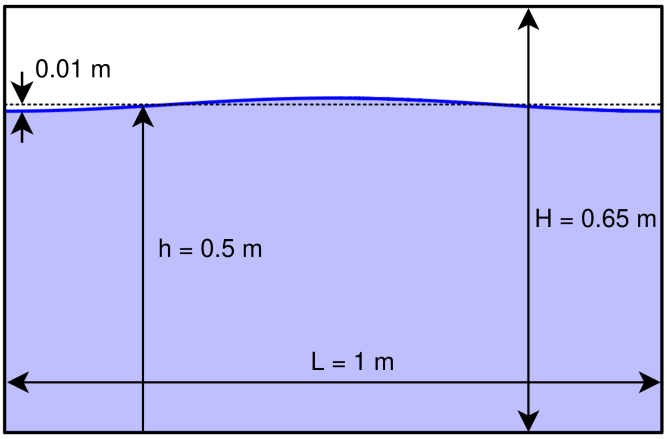
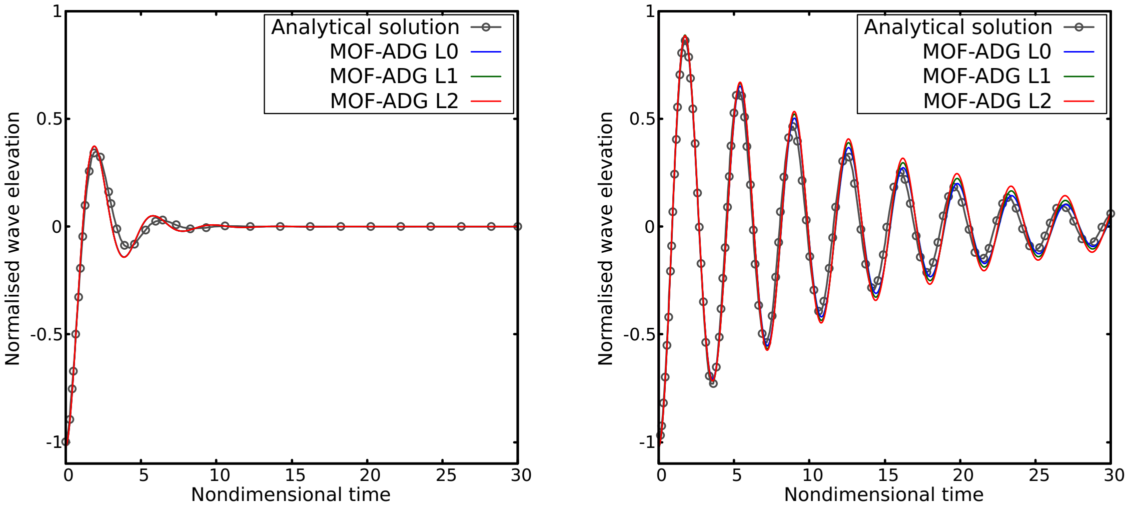


Fig. 6.16 Schematic of the tank dimensions and initial condition.

and ρ and μ are the density and dynamic viscosity of water, respectively. The air-water density ratio is maintained as per the inviscid case. The viscosity ratio is kept constant while studying different Reynolds numbers. In both cases, three levels of grid resolution were implemented using the novel MOF-ADG method. The first one corresponds to a fixed grid, level 0 (L0), the second has one level of refinement (L1) and the final one has two levels of refinement (L2). The simulation time is extended to 30 units, where time is multiplied by $\sqrt{g/h}$ to obtain a dimensionless time. The time step used across different refinements is constant.

Fig. 6.17 presents the normalised wave elevation at the left boundary as a function of the nondimensional time. For both cases, $Re = 20$ and $Re = 200$, numerical predictions on all three grids are compared with the analytical solution provided by Liu and Lin (2008) and Wu et al. (2001). The characteristic oscillatory behaviour reveals a gradual reduction in wave amplitude. Notably, the decay rate of the wave is more pronounced for a smaller Reynolds number. In both situations, the wave elevation at the left boundary matches favourably the analytical solution. Note that for a low Reynolds number the adaptive dual grid method exhibits very similar behaviour irrespective of the levels of refinement while a higher Reynolds number exhibits a slight discrepancy in wave elevation when using one or two levels of refinement.



(a) $Re = 20$

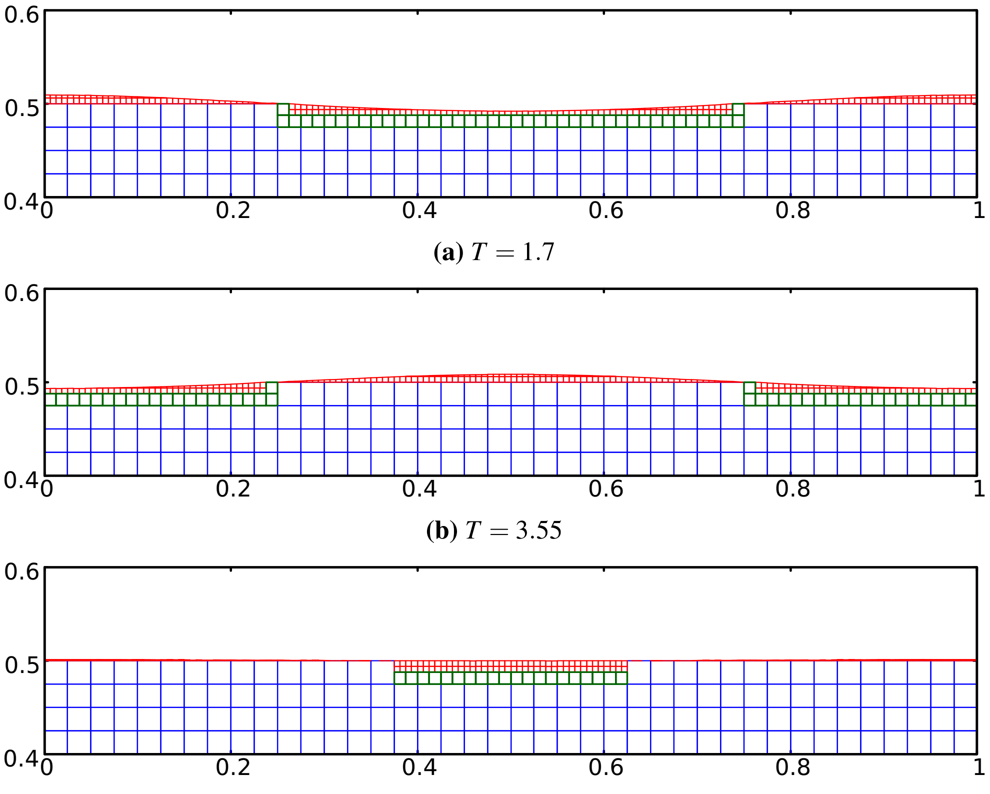
(b) $Re = 200$

Fig. 6.17 Evolution of water level at the left boundary using the MOF-ADG method with a 40×26 base grid compared with the analytical solution for (a) $Re = 20$, (b) $Re = 200$.

Fig. 6.18 depicts a natural sloshing sequence at various time intervals revealing the dynamic behaviour and evolution of liquid movement within a tank when $Re = 200$. The coarse base grid is composed of a 40×26 arrangement of Cartesian cells. As time progresses, subsequent figures capture the agitation at $T = 1.7$, $T = 3.55$ and $T = 30$. Note that for these two cases, the wave period is different from the inviscid case. These figures provide a comprehensive visual representation, illustrating the dynamics of sloshing using the MOF-ADG method.

6.4.1.3 Viscous sloshing under horizontal excitation

In this section, the sloshing of a liquid in a 2D rectangular tank that is horizontally agitated is assessed. The excitation occurs periodically, with velocity components $\mathbf{u}_{exc} = [-A\omega\cos(\omega t), 0]$, where A is the displacement amplitude and ω denotes the angular fre-



(a) $T = 1.7$

(b) $T = 3.55$

(c) $T = 30$

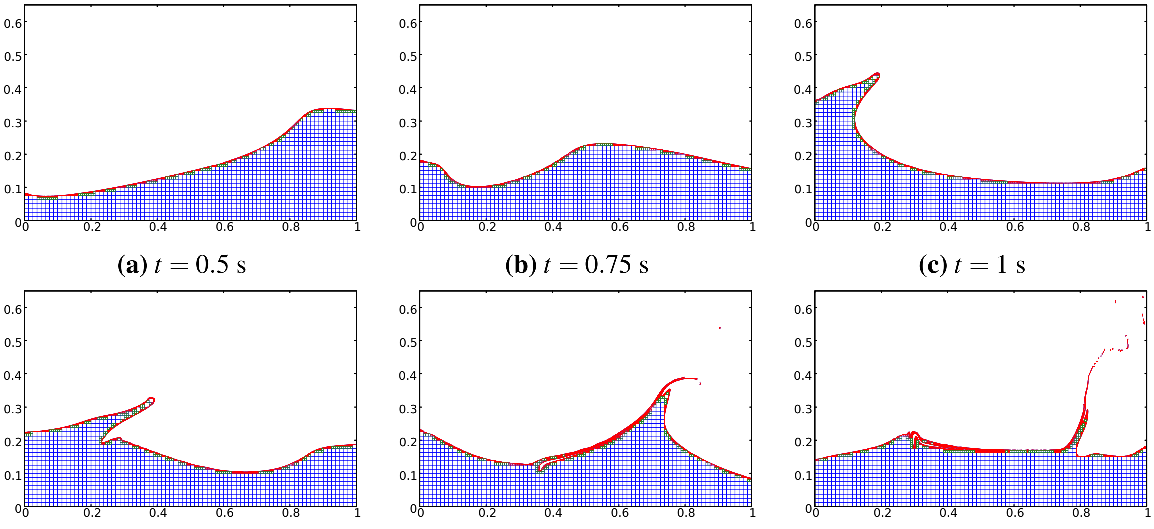
Fig. 6.18 Evolution of water level within the tank when $Re = 200$. Zoom is carried out on the interface to highlight the performance of the MOF-ADG method. The base grid (L0) is composed of 40×26 Cartesian cells. Blue depicts level 0 (L0), green level 1 (L1) and red level 2 (L2).

quency of the excitation. Within the momentum equation, this excitation is provided as an external force **f** in addition to gravity forces i.e.

$$\mathbf{f} = \mathbf{g} - \frac{d\mathbf{u}_{exc}}{dt} \tag{6.13}$$

The parameters and dimensions of the tank are similar to the free sloshing case above. The mesh used to generate the results is finer to better track the complex dynamics. The notable difference is the initial condition. In such a scenario, the natural frequency of the fluid in the tank is expressed as $\omega_0 = \sqrt{\pi g \tanh(\pi h)}\ \mathrm{s}^{-1}$ (Liu and Lin, 2008). In the following test case, the free surface is initially at rest in the tank at a new height $h = 0.175$ m. For this test case, the amplitude and frequency are specified as follows: $A = 0.06$ m and $\omega = 2.0\omega_0$.

This case highlights the ability of the MOF-ADG method to handle complex dynamics. Fig. 6.19 shows the evolution of the interface, which increases in complexity until the forming jet falls onto the free surface. With the motion of the tank, a first jet emerges from the interaction with the left-hand side wall, followed by a second thinner jet. Note that due to the use of two levels of refinement, the thin jet does not break up and maintains a sharp interface. This suggests that breakup is likely, however observations indicate that the breakup occurs only with two levels of refinement, highlighting that the method is capable of effectively capturing subgrid features. The series of snapshots captured during the simulation offer a visual narrative of the dynamic topology of the interface. It also highlights the interaction taking place within the domain near the boundary walls.



(a) $t = 0.5$ s **(b)** $t = 0.75$ s **(c)** $t = 1$ s

(d) $t = 1.25$ s **(e)** $t = 1.5$ s **(f)** $t = 1.75$ s

Fig. 6.19 Snapshots of the evolution of the air-water interface during horizontal excitation (sloshing) of a tank initially at rest and with $A = 0.06$ m and $\omega = 2.0\omega_0$ using the MOF-ADG with two adaptive levels of refinement.

6.4.2 Dam break problem

Demonstrating its significance as a benchmark problem in multiphase flow simulations, the phenomenon of dam breaking encompasses intricate dynamics, including high-impact pressures, surface fragmentation, and the formation of water jets, among others. The investigation presented in this section focuses on dam breaking and aims at replicating the experimental work of Zhou et al. (1999) and Buchner (2002) and numerical work of Greaves (2006). The computational domain consists of a rectangular tank measuring 3.22 m in length and 1.8 m in height. The lower-left corner accommodates the water phase, with a reference density of 1000 kg/m^3, occupying an area of 1.2 m in width and $H_c = 0.6$ m in height. The air medium is characterized by a reference density of 1.29 kg/m^3. Fig. 6.20 offers a visual schematic of the domain. Time is nondimensionalised by $\sqrt{g/H_c}$, pressure by $\rho g H_c$ and height by H_c.

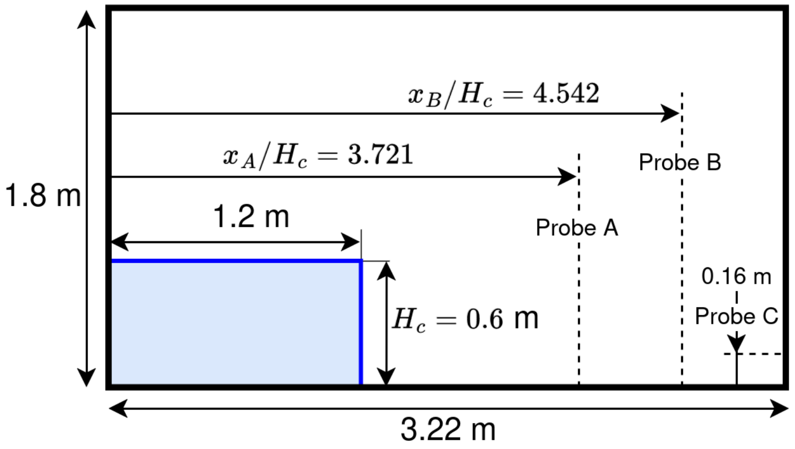
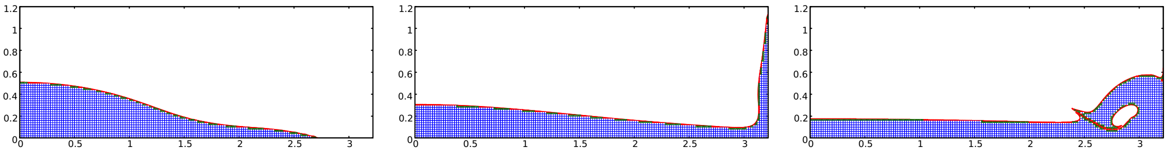


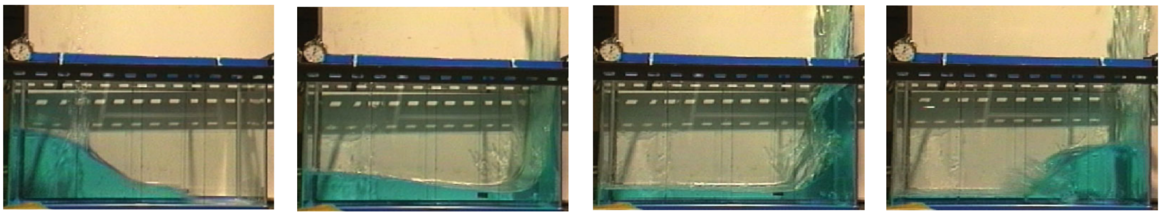
Fig. 6.20 Schematic of the domain, initial condition and location of probes A, B and C, which are used as instrumental validation.

Validating the numerical simulations of a dam break event against experimental and numerical data in the literature is vital in demonstrating the reliability of the method. Fig. 6.21 shows a sequence of three snapshots following the collapse of the dam under the influence of gravity. Snapshots show two levels of refinement where the base grid is composed of 161 × 90 Cartesian cells. When the dam breaks the surge position advances rapidly (Fig. 6.21(a)). This is followed by a strong vertical motion after the water impacts the right-hand wall (Fig. 6.21(b)). Eventually, the liquid falls back on itself creating a cavity (Fig. 6.21(c)). Using the filament MOF method and a large time step, this last snapshot reveals a thin air film that does not enclose the cavity completely. Fig. 6.22 illustrates snapshots of the experimental work of Koshizuka (1995), included as qualitative comparison despite dimensions being different and being captured at slightly different nondimensional time.



(a) $T = 2$ **(b)** $T = 4$ **(c)** $T = 6.2$

Fig. 6.21 Evolution of the air-water interface for dam break problem using the MOF-ADG method with two levels of refinement.

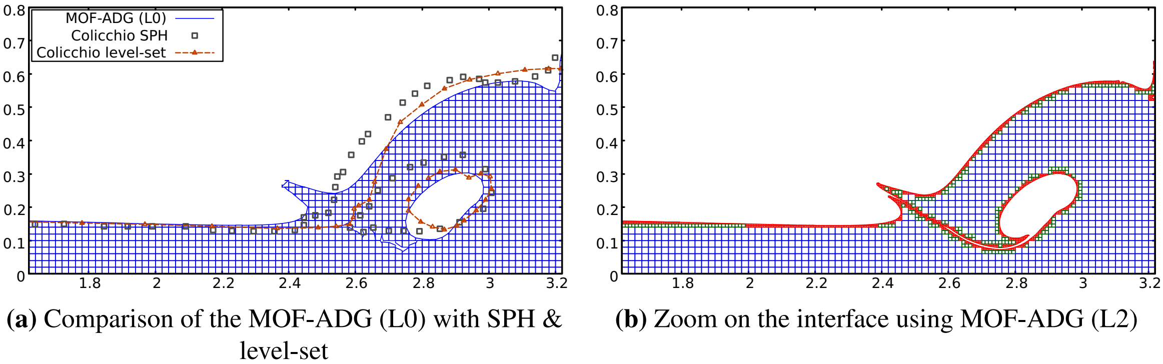


(a) $T = 1.61$ **(b)** $T = 3.23$ **(c)** $T = 4.85$ **(d)** $T = 6.46$

Fig. 6.22 Evolution of the experimental work of Koshizuka (1995).

The present adaptive dual grid MOF method preserves detailed phenomena, such as water splashing and the formation of flow jets. Fig. 6.23 shows the results of the present method for level 0, 1 and 2 and comparison is made with the level-set method and SPH methods provided by Colicchio et al. (2002). The air cavity is well captured and enclosed using MOF-ADG (L0) method, while still containing a few "bubbles of air". Other numerical methods also capture the air cavity but a large discrepancy occurs in predicting its location. The level-set method creates a large bridge of coalescence and a small cavity whilst the SPH method creates a large cavity. The associated jet also shows a significant difference between the three methods. Note that the process of interface refinement tends to create a thin film of air, rather than fully enveloping the cavity.

Fig. 6.24 illustrates the temporal evolution of water height and pressure at three locations. These probes were used in the early experiments performed by Zhou et al. (1999) and

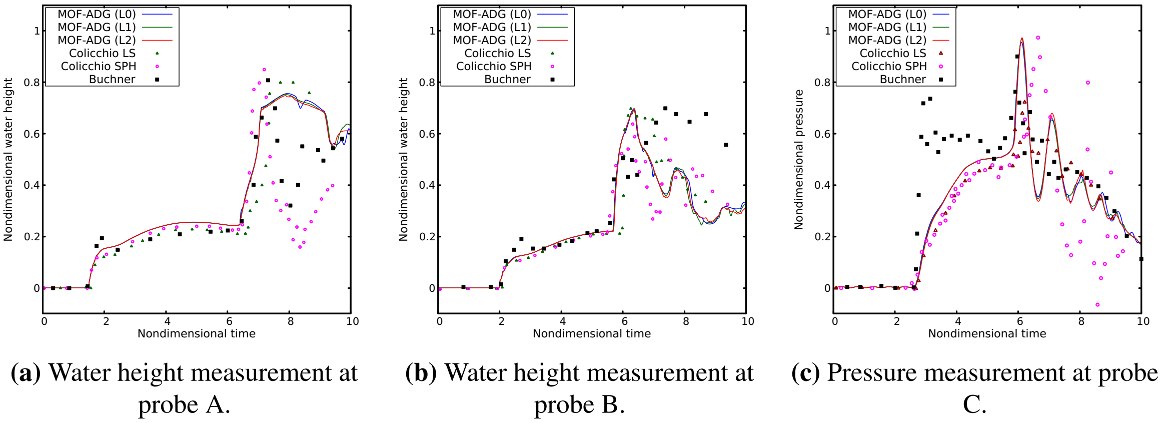


(a) Comparison of the MOF-ADG (L0) with SPH & level-set

(b) Zoom on the interface using MOF-ADG (L2)

Fig. 6.23 (a) Comparison of the air-water interface for the MOF-ADG, level-set and SPH methods at $T = 6.2$. (b) Interface generated with level 0 is in blue, level 1 in green and level 2 in red ($T = 6.2$).

Buchner (2002). Probe A is located at $x_A/H_c = 3.721$, probe B at $x_B/H_c = 4.542$ along the horizontal axis accounting for water height evaluation. Probe C is located at the right hand wall at a height of 0.16 m.



(a) Water height measurement at probe A.

(b) Water height measurement at probe B.

(c) Pressure measurement at probe C.

Fig. 6.24 Water height and pressure measurement analysis using MOF-ADG method compared to level-set and SPH methods from Colicchio et al. (2002) and experimental work from Buchner (2002).

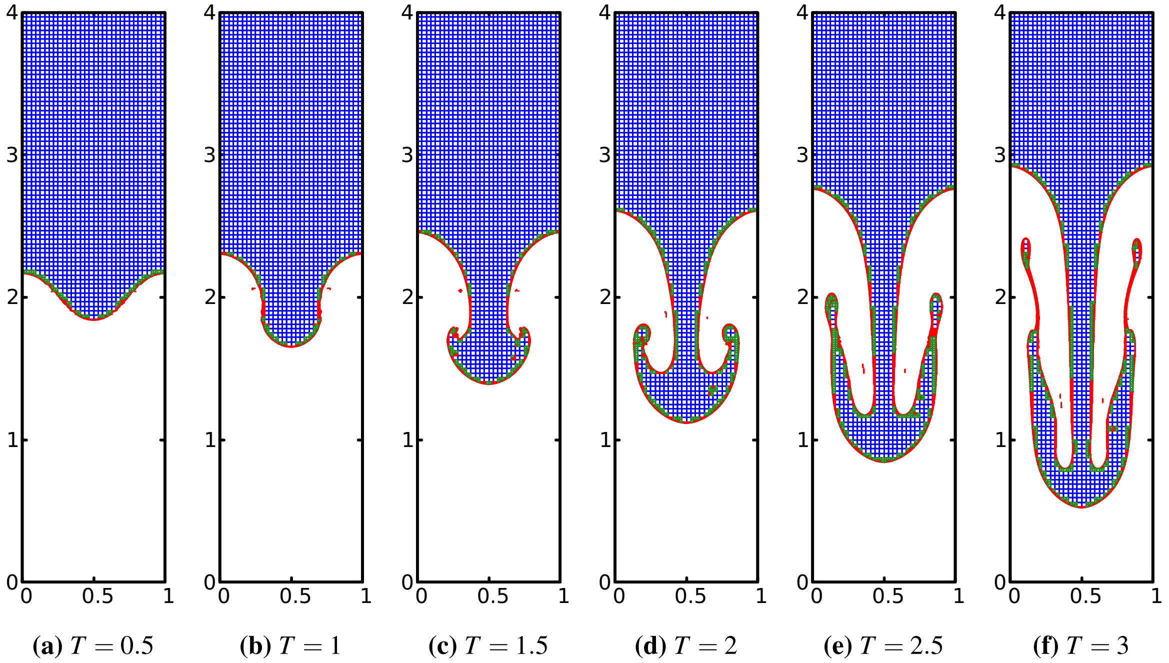
While the experimental and numerical trends exhibit a satisfactory level of agreement for water height measurements until $T = 6.5$, a notable disparity emerges throughout the second part of the simulation and the water level peak seems to be underestimated. The initial pressure rise agrees well with other numerical methods but there is a delay in attaining its peak value compared to experimental work. Note that the different resolutions of the MOF-ADG method exhibit nearly overlapping trends. In conclusion, there is a relatively limited level of agreement between the computational results and the experimental data which may be due to the pressure transducer location or variations in the water front propagation velocity. Challenges in achieving measurement repeatability has been previously documented leading to inconclusive findings when comparing with experimental data (Zhou et al., 1999). However, most numerical methods seems to show good agreement with each other.

6.4.3 Rayleigh-Taylor instability

This well-known instability is a phenomenon that emerges when gravity causes a heavy fluid initially at rest on top of a lighter fluid to deform the interface between them. In this study, the same configuration used in previous investigations (Ding et al., 2007; Tryggvason, 1988; Xie et al., 2014) is adopted. The rectangular domain is $[0,d] \times [0,4d]$ and the interface is initially perturbed with a sinusoidal waveform of amplitude $0.1d$. The Atwood ratio A, which is a measure of the density difference between the heavier and lighter fluids, is set to 0.5. Time is nondimensionalised by $\sqrt{d/Ag}$, and surface tension and turbulence effects are ignored in this study.

We conduct computations using three cases for the present MOF-ADG method with different levels of refinement while using the same finest Cartesian grid size. Therefore for these computations, the time step is the same. The coarsest grid is 32×128 with 2 refinement levels (L2), the medium grid is 64×256 (L1) and the finest grid is 128×512 (L0), respectively. Fig. 6.25 shows a sequence of snapshots revealing the evolution of the interface, where the heavy fluid falls into the lighter fluid. The early stages of the deformation of the interface exhibit a 'mushroom' like shape, then thinner structures begin to appear as the material becomes increasingly deformed. Some "bubbles"/"droplets" appear in the material and are highlighted with the colour scheme of the refinement.

Fig. 6.26 highlights the refined reconstruction of the interface during deformation at a nondimensional time of $T = 3.5$ for the three cases. The author observes that the "mushroom"

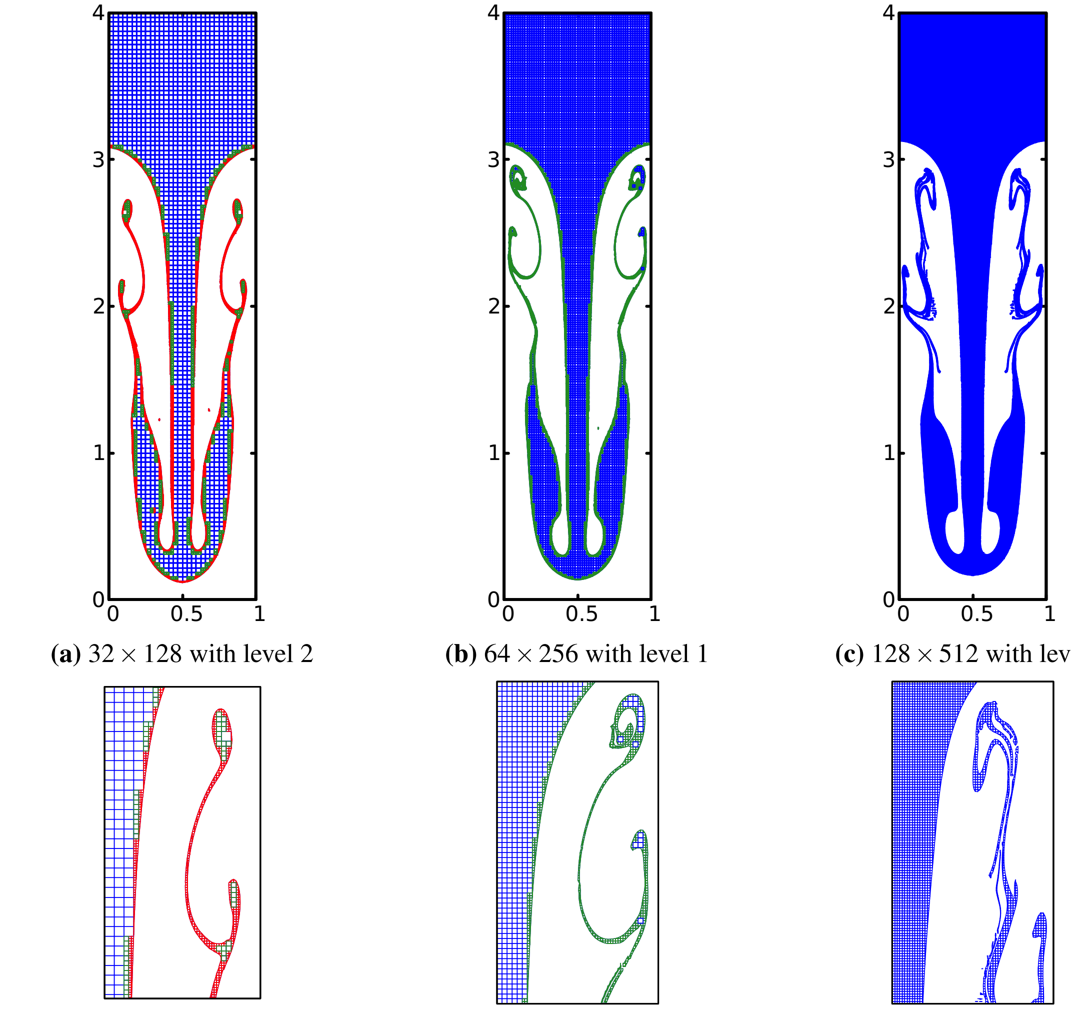


(a) $T = 0.5$ (b) $T = 1$ (c) $T = 1.5$ (d) $T = 2$ (e) $T = 2.5$ (f) $T = 3$

Fig. 6.25 Snapshots showing the evolution of the interface between the heavy fluid (blue) and the light fluid (white) using the novel MOF-ADG method with a 32×128 base grid with two levels of refinement. T is nondimensional time.

shape formation during the spike penetration is reconstructed well. The interface remains sharp and exhibits very limited diffusion. Many filaments are present, but grid convergence of these structures is elusive. To the best of the author's knowledge, such a late stage of simulation with a coarse grid and subgrid structures has not been achieved in the literature.

In order to investigate the robustness and performance of the MOF-ADG method, some insights into its performance in Fig. 6.27 is provided. The evolution of the spike position aligns well with other methods used in the field and the nondimensional time is extended to $T = 3.5$ (Fig. 6.27(a)). Acceptable mass conservation is achieved throughout the simulation (Fig. 6.27(b)). A finer mesh tends to have inferior mass conservation properties compared with a coarse mesh due to the number of over/under-filled cells and area of redistribution. Indeed, for this particular scenario and despite advecting only the interface, more mass needs to be redistributed around a small interface, which to a greater extent affects the mass variation for finer grids. This is valid both for fixed grids and adaptive grids. Reducing

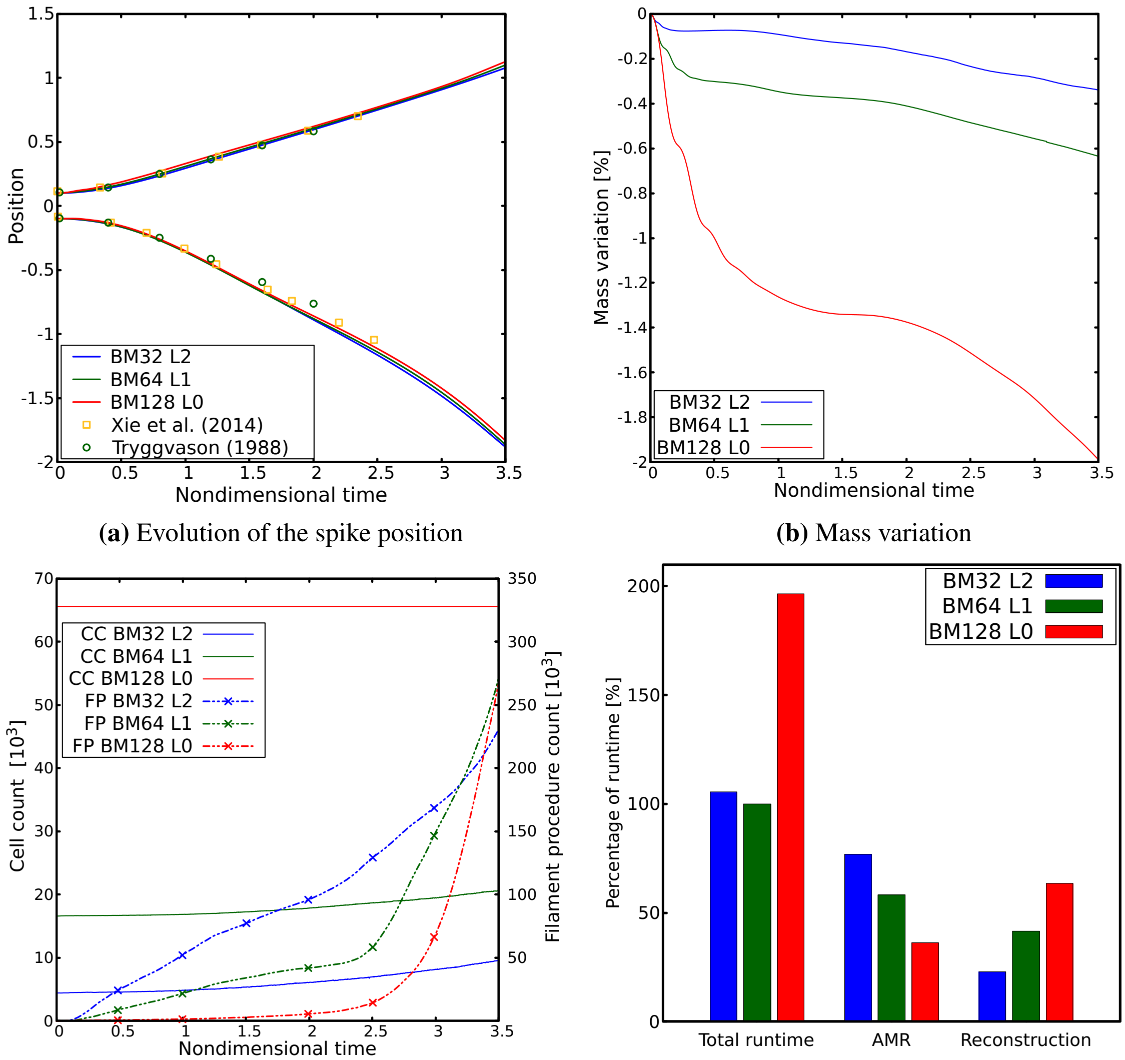


(a) 32 × 128 with level 2 **(b)** 64 × 256 with level 1 **(c)** 128 × 512 with level 0

(d) Zoom box $[0.5,1]\times[1.8,2.8]$ **(e)** Zoom box $[0.5,1]\times[2,3]$ **(f)** Zoom box $[0.5,1]\times[2,3]$

Fig. 6.26 Snapshots of the Rayleigh-Taylor instability for the MOF-ADG method at $T = 3.5$ for the same finest level of refinement. In these snapshots, blue denotes level 0, green level 1 and red level 2.

the time step would improve mass conservation even further but at a greater computational cost. The Courant number is not a limiting factor in the proposed MOF approach. Two fundamental variables, cell count and filament subroutine count are shown on a common plot in Fig. 6.27(c). The filament subroutine count cycles through the domain and iterations during the dynamic process in order to evaluate how many times the filament reconstruction subroutine has been called. Since two interfaces are reconstructed within a cell, it is important to understand its potential influence on the computational cost. The juxtaposition between



(a) Evolution of the spike position

(b) Mass variation

(c) Cell & filament procedure count

(d) Runtime

Fig. 6.27 Detailed analysis of several numerical data of the MOF-ADG method. BM stands for base mesh. CC stands for cell count and FP for filament procedure. 100% total runtime is based on the shortest simulation.

cell count and filament count allows for a clear and concise comparison of the influence of these outputs offering potential correlations with runtime. Moreover, the last figure (Fig. 6.27(d)) offers a crucial evaluation of the relative runtime performance of each principal component of the computation. The total runtime is compared with respect to the time of the shortest simulation, shown as a percentage, with the rest providing a more expensive computation. The AMR and reconstruction subroutines are responsible for a major part of the interface capturing procedure (Hergibo et al., 2024). The time spent in these subroutines is added together across all iterations and each value in the AMR and reconstruction histogram corresponds to the percentage of time shared between them. It clearly shows that significant time is spent in the AMR at level 2 while having nearly the same number of filaments to reconstruct. As reported in Hergibo et al. (2024), in many instances a single level of refinement is the most efficient approach.

6.5 Discussion on the efficiency of the MOF-ADG method

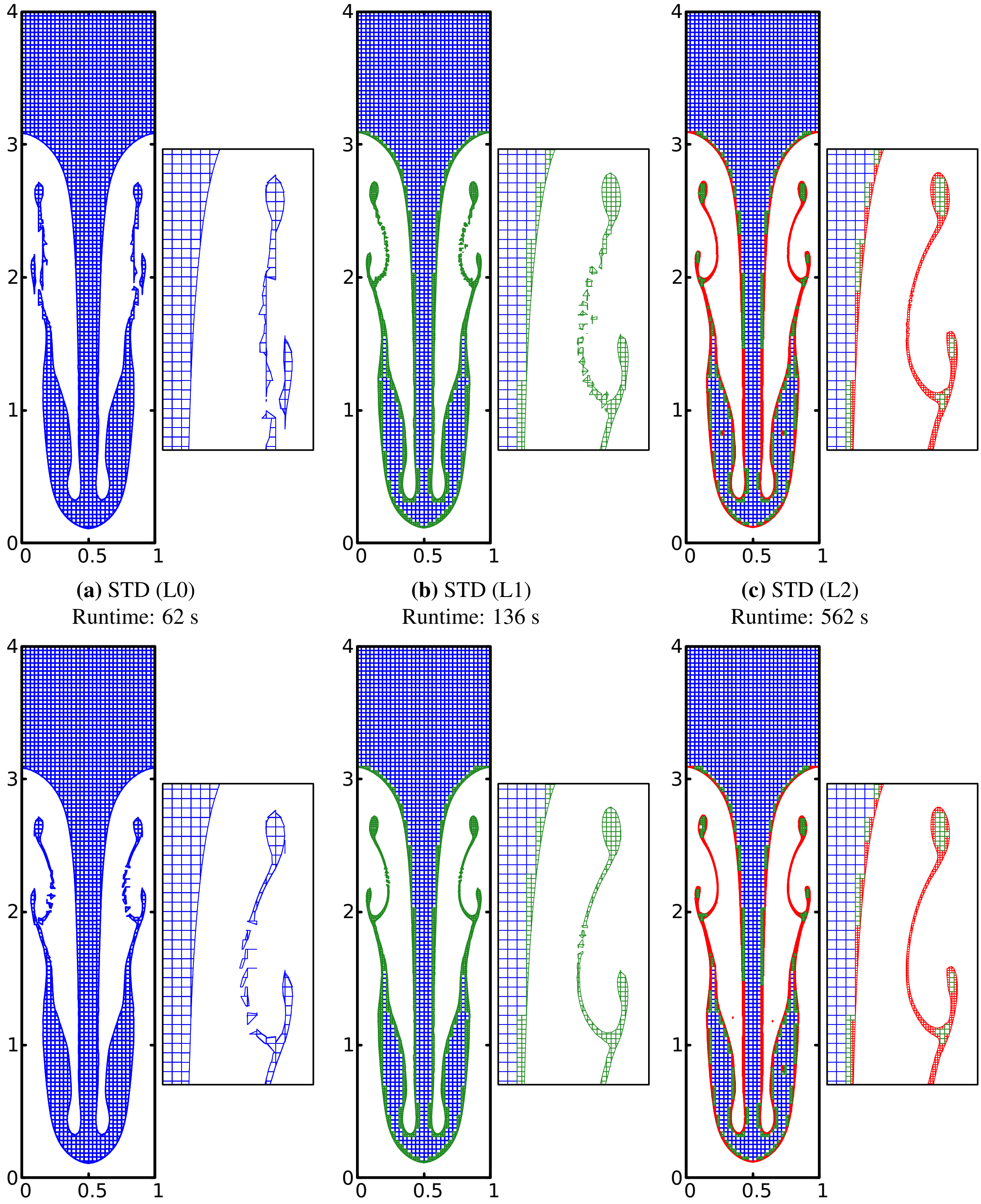
The MOF-ADG method framework offers a compromise between accuracy and runtime. It achieves this by using dual grid benefits and also refining regions of interest. It naturally reduces the overall number of cells used in a computation compared to a uniform grid or a single AMR grid. Typically, in AMR practices, runtime increases as the number of refinement levels increase, resulting in decreased error or a more precise region of interest.

However, in the context of the MOF method, regions experiencing high deformation may be able to be reconstructed effortlessly using filaments while maintaining a reasonable computational cost. Consequently, one can assess the accuracy of various levels of a MOF-ADG method using standard and filament procedures. The choice between filament, adaptivity or the combination of both can be assessed. Note that a filament reconstruction with a higher base resolution but a lower level of refinement may be equivalent to a lower base resolution reconstruction but with a higher level of refinement.

This section aims to provide insights into the trade-off between refinement and error using the Rayleigh-Taylor instability problem. The time step remains constant across all test cases, therefore, comparing runtime alone may not provide a comprehensive understanding of efficiency. The first set of cases uses a 32×128 grid with level 0, 1 and 2 and a standard reconstruction. The second set of cases uses the same resolution but with a filament reconstruction.

Fig. 6.28 shows a matrix of figures each with a zoom in for the standard and filament MOF reconstruction and with zero to two levels of refinement. The figure demonstrates that, in general, standard reconstruction does not provide a great deal of improvement and two levels of refinement are needed. However, for such extreme filamentary cases, a coarse grid and one level of refinement does not prevent the breakup of filaments, even when enabling filament reconstruction. Eventually, two levels of refinement are necessary to achieve an acceptable degree of precision, although using a standard reconstruction gives a poor result. The zoom provided shows the standard MOF method with two levels of refinement is on the verge of breaking up whereas the filament MOF method shows a smoother interface. The filament MOF-ADG shows a better reconstruction even on a fixed grid, for level 0. Filaments do breakup with zero and one level of refinement when using a filamentary reconstruction. This can be explained by the choice of multi-material reconstruction described in Hergibo et al. (2023). An interesting question concerns the choice of *best* variant of MOF-ADG method in order to obtain acceptable results. On the one hand, it seems that a filament method with L0 is faster and seems to breakup less than a standard MOF-ADG L1. On the other hand, a filament MOF-ADG L1 offers more breakup than the standard MOF-ADG L2, although significantly faster.

In terms of runtime, each runtime is provided in the caption. Filament reconstruction generally achieves comparable results in terms of accuracy or slightly better but at a small extra cost. Therefore, the comparison of runtime between standard and filament MOF-ADG is less pertinent. With this information however, the author can conclude that filament reconstruction is consistently a more advantageous choice than using adaptivity when considering both accuracy and runtime. Combining both filament and adaptivity capabilities remain the *best* choice.



(a) STD (L0)
Runtime: 62 s

(b) STD (L1)
Runtime: 136 s

(c) STD (L2)
Runtime: 562 s

(d) FIL (L0)
Runtime: 79 s

(e) FIL (L1)
Runtime: 198 s

(f) FIL (L2)
Runtime: 680 s

Fig. 6.28 Study of the efficiency of the MOF-ADG method. Accuracy against refinement using both standard and filament reconstruction. The base mesh is 32×128 and uses three different levels of refinement. The side figure shows a zoom on the tip of the filamentary mushroom, zoom box $[0.5, 1] \times [1.8, 2.8]$.

6.6 Closing remarks

To conclude this chapter, the adaptive mesh refinement moment-of-fluid (MOF-AMR) method and the adaptive dual grid moment-of-fluid (MOF-ADG) have been explored as novel approaches to enhance numerical multiphase flow dynamics. The primary objective was to showcase the accuracy, efficiency, and versatility of these methods by addressing their challenges and limitations.

This chapter has meticulously outlined the validation procedure for the adaptive mesh refinement moment-of-fluid (MOF-AMR) method. The challenges of maintaining a sharp interface in numerical multiphase flows are effectively addressed. The method has been successfully validated with simplicity and accuracy, offering the possibility to capture intricate subgrid structures for multiphase flows.

The real-world application of the adaptive dual grid moment-of-fluid (MOF-ADG) method has not only demonstrated its adaptability but also confirmed that it is an important newcomer in the arena of numerical methods for multiphase flows. The method manages and predicts complex fluid motions across multiple scales for diverse problems, and offers a valuable computational tool for multiphase flow dynamics.

Chapter 7

Conclusions and Future Research

The aim of this thesis is to create a computational method for understanding of complex multiphase flow problems in two dimensions. In this chapter, the summary of the key discoveries and conclusions is presented, followed by a discussion on recommendations for future research.

7.1 Conclusions

7.1.1 The MOF method: filaments

In conclusion, a new MOF method with a symmetric multi-material approach has been presented where thin structures are resolved using a filament approach for a fixed coarse mesh. A novel robust approach to solve the optimisation problem is proposed using a bisection method. No initial condition or parameters are necessary and the global minimum can always be found. A Lagrangian backtracking approach ensures that there is no limitation on the CFL number when advecting materials. Solving under-resolved filaments inherently involves a higher computational cost, which is reduced by choosing to cap the number of conglomerates at three and using a symmetric approach. As a result, almost quadratic order of convergence is achieved and the error converges as the grid is refined. However for complex and large material deformations, the limitation of this method is shown and the topology might not be well maintained at sharp edges.

This efficient approach is applied to several benchmark problems with different levels of deformation. Most of these benchmark problems are compared with different MOF approaches, filaments, AMR, CLSMOF and standard MOF method. First, the Zalesak slotted disc does not exhibit any filament behaviour, yet the chosen approach shows good qualitative results. Other benchmark problems such as the reversible vortex and the droplet flow case are tested for large deformation highlighting the quality of reconstruction of filaments on coarse meshes. Finally, the rotating filament benchmark is presented, which is only applicable using filament reconstruction for such coarse grids. The limitation of the method is shown in the S-shape deformation benchmark. For most benchmark problems, the error and runtime are at least comparable to other MOF methods. Furthermore, the accuracy in interface reconstruction is improved for large deformation. In addition, runtime has been decreased compared to most MOF methods, although its significance may be attenuated due to advancements in computer performance over the years.

7.1.2 The MOF-AMR method: adaptivity

A new quadtree-based adaptive MOF method has been developed in which filament structures are resolved using a symmetric multi-material approach on a refined grid. A simplified quadtree structure has been implemented with logical connection between parent and children cells up to two levels of refinement. A Lagrangian backtracking approach for refined grids is proposed that enables exact material intersection during the advection process, hence ensuring mass conservation. The refinement criterion is based on the centroid defect relative to the cell or subcell size, ensuring linear interfaces are reconstructed exactly without the need for refinement. As a result, the proposed framework achieved good results in terms of accuracy and runtime while using computational resources in a more efficient manner. Comparison between different levels of refinement for the same minimum cell size provides insight into the most efficient use of this framework and the MOF method in general.

This MOF-AMR method is tested on several benchmark problems with high material deformation. All of these benchmark problems are compared with a couple of similar MOF approaches using mesh refinement. First, the Zalesak slotted disc has less refined cells at the initial stage and achieved good qualitative results. Other benchmark problems such as the reversible vortex show highly precise reconstruction at maximum deformation under different levels of refinement. The droplet flow and the S-shape test case yielding highly deformed structures are presented with filament reconstruction. Quantitatively, results are comparable to other MOF methods. The limitation of the proposed method lies in the number of refinement levels available in an unconstrained adaptive grid structure. The chosen refinement approach differs from other MOF-AMR reference methods. Machine precision mass conservation algorithms are achieved for benchmark problems such as the reversible vortex, whilst further improvements are required for other problems such as the droplet flow or the S-shape case. Furthermore, runtime has been significantly decreased compared to alternative MOF methods. In this study, no high-performance libraries are used and calculations are carried out on a single core (Intel Core i7, 8M cache, base 1.8 GHz). High-performance frameworks would offer strong scalability and efficient algorithms for handling large parallel octree operations. Yet, complexity and potential resource requirements may be challenging. In comparison, the proposed data structure offers ease of use and accessibility, suitable for smaller-scale efforts. Many advantages follow from this decision such as the absence of load balancing, numbering, and neighbouring search. However, the author is aware of potential issues related to limited scalability and versatility, memory access, parent node data optimisation and general computing performance.

7.1.3 Application of the MOF method: simulating multiphase flows

The moment-of-fluid method coupled to a finite volume Navier-Stokes solver has been presented for multiphase flows on a fixed grid. First, the MOF method is implemented for different benchmark test cases where the complexity of the flow is relatively low. Then, the filament MOF method is introduced where subgrid-scale structures are resolved for a highly complex flow structure. This represents a significant advancement in the understanding and modelling of complex multiphase flows. Higher levels of accuracy and efficiency in capturing the intricate interactions between two phases are achieved. A direct comparison has shown the differences of approach between a standard and a filament MOF method. The representation of filaments is relatively well captured using the latter method, despite some remaining filament breakups. Promising results in resolving subgrid-scale structures have been demonstrated.

The proposed method is tested on well-known benchmark problems featuring different levels of complexity. Good qualitative predictions of the evolution of the height of the water column height and position of the surge front are achieved with both experimental measurements and previous numerical studies for the dam break problem. In the case of the Rayleigh-Taylor instability problem, the method is able to reconstruct the interface in a precise manner during the early stages of development, while the filament MOF preserves highly deformed subgrid-scale structures during the later stages. Both standard and filament MOF methods maintain satisfactory agreement with the predictions of other numerical methods. Finally, the method exhibits a good level of accuracy for the Kelvin-Helmholtz instability problem.

7.1.4 The MOF-ADG method for multiphase flow simulations

In conclusion, this study introduces a novel adaptive dual grid moment-of-fluid (MOF-ADG) method coupled with an implicit finite volume Navier-Stokes solver. This method represents a novel and promising approach for simulating complex multiphase flow dynamics using two separate Cartesian grids: a fixed coarse grid for the fluid solver and an adaptive grid for the interface capturing method. The present approach exhibits remarkable adaptivity in refining computational grids, offering a significant improvement in the numerical efficiency and accuracy of fluid flow simulations. In addition, numerical complexity has been enhanced by including subgrid-scale elements such as filaments in order to reduce the persisting breakup of numerical filaments through the moment-of-fluid method.

The method proposed in this thesis is evaluated using established benchmark problems in which challenges are of varying complexities. The method demonstrates an excellent level of accuracy in addressing the natural sloshing problem, displaying quantitative agreement with existing literature and analytical solutions. Moreover, the dam break problem successfully provides accurate quantitative estimations of water height evolution and pressure measurement, aligning well with both experimental data and previous numerical studies for dam break problems. In the Rayleigh-Taylor instability problem, the method precisely reconstructs the interface during its early stages and effectively captures deformed subgrid-scale structures in later stages with the particularly well developed filament MOF-ADG. In the final section, the MOF-ADG method shows its relevance by comparing reconstructions for a single case exploring different refinement scenarios. In summary, the MOF-ADG demonstrates excellent performances in accurately capturing complex flow features and the intrinsic physical behaviour.

7.2 Future research

The MOF method, like most interface capturing methods, diffuses when advecting sharp edges. In addition, the tip of filaments is not well-resolved regardless of the mesh resolution. However, the method shows greater ability to reconstruct accurate interfaces in complex situations such as near boundaries or filament/multi-material over VOF methods. In future work the author would like to include the ability to reconstruct four conglomerates within a cell while maintaining a high level of accuracy in reconstruction with an acceptable runtime. For a fixed coarse mesh, this may lead to increased precision in thin layered filaments while reducing the natural diffusion of material. This approach could involve an optimised selection of which material to reconstruct. As well as being able to reconstruct multiple materials within one cell, future work may involve extending the proposed MOF method to more than two materials in the domain in order to be able to handle complex scenarios such as fluid-structure interaction or three fluids interacting (air-water-oil). Undoubtedly, expanding this will introduce several challenges such as increasing complexity, conserving mass for all materials and maintaining a reasonable computational cost. Addressing challenging 3D reconstruction is also a development that the author wish to pursue in the future.

The present MOF-AMR method tends to decrease the cross-stream diffusion of advected material and can reconstruct sharp edges or tips of filaments with greater accuracy using up to two levels of refinement. Further improvement may be made to address these numerical issues by using the recent new moment-of-fluid method (Shashkov and Kikinzon, 2023) or the parabolic interface reconstruction (PPIC) (Remmerswaal and Veldman, 2022). In addition, the number of conglomerates has less influence on the centroid defect as these scenarios tend to trigger refinement. In future work, advecting and reconstructing several materials within the same domain using the present MOF-AMR method will be a priority, which will most likely involve reconstructing more than three materials. In this AMR framework, optimising the levels of refinement could reduce the computational cost by creating predictive algorithms. Indeed, based on geometrical features and characteristics, predictive algorithms can strategically refine cells and regions without the need to reconstruct iteratively, therefore enhancing efficient and effective use of computational resources.

Despite advances in interface capturing methods for multiphase flows, surface tension has been neglected in this thesis which may modify vortices created in certain complicated flow events. The complexity of the method and the associated computational cost, although mitigated with analytical solution on Cartesian grids, may be a limiting factor for its use. Nevertheless, the moment-of-fluid method remains a robust and promising improvement to interface capturing/tracking methods and in numerical multiphase flows in general. Increasing the levels of detail in subgrid-scale structure as well as including surface tension and turbulence modelling, whether for Newtonian or non-Newtonian flows is the next priority while maintaining acceptable levels of computational resources. The moment-of-fluid method is an example of the next generation of interface capturing methods and is well-placed to play a pivotal role in the analysis and prediction of complex multiphase systems.

In addition, certain limitations of the MOF-ADG method arise notably in situations with breakup and coalescence or with complex geometries despite offering sharper interface reconstruction. The computational expense of the proposed method is somewhat mitigated by employing analytical solutions on Cartesian grids for interface capturing features when possible. However, the use of a fixed coarse grid for the flow solver is not negligible in terms of computational efficiency. For future work, the focus will be on addressing these limitations by including additional features such as surface tension on an adaptive dual grid, which may involve some additional complex numerical challenges, and extending the application of the MOF-ADG method to a wider range of multiphase problems such as droplet and bubbles dynamics. Furthermore, the proposed approach should be extended to three-dimensional problems. In addition to 3D reconstruction mentioned previously, this extension in 3D may involve challenges such as conglomeration algorithm and octree formation validity. The adaptive dual grid moment-of-fluid (MOF-ADG) method has been shown to be a robust and promising enhancement to interface capturing and tracking techniques in complex numerical multiphase flows.

Parallelisation is crucial for the practical applicability of the MOF method and its derivative techniques, especially when dealing with larger systems. Nevertheless, this thesis has been carried out on a single core, leaving room for improvements through parallelisation. Devoting efforts in this area involves exploring advanced high-performance computing techniques. The goal is to enable simulations of large scale, allowing for representations of real-world scenarios.

Some other future work may involve exploring coupling mechanisms with other complex area of CFD such as mass and heat transfer or fluid-structure interaction. It is a fundamental aspect in numerous real-world scenarios, from industrial processes to environmental studies. Extending validation efforts to include simulations in extreme conditions is a crucial step in ensuring the robustness and reliability of the MOF method. Simulating extreme conditions or environments, such as high-speed compressible flows, provides an opportunity to test the accuracy and stability of the method in scenarios where traditional approaches might struggle.

References

Ahn, H. T. and Shashkov, M. (2007). Multi-material interface reconstruction on generalized polyhedral meshes. *Journal of Computational Physics*, 226(2):2096–2132.

Ahn, H. T. and Shashkov, M. (2009). Adaptive moment-of-fluid method. *Journal of Computational Physics*, 228(8):2792–2821.

Ahn, H. T., Shashkov, M., and Christon, M. A. (2009). The moment-of-fluid method in action. *Communications in Numerical Methods in Engineering*, 25(10):1009–1018.

Alexandrov, O. and Santosa, F. (2005). A topology-preserving level set method for shape optimization. *Journal of Computational Physics*, 204(1):121–130.

Anderson, D. M., McFadden, G. B., and Wheeler, A. A. (1998). Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30(1):139–165.

Antepara, O., Balcázar, N., and Oliva, A. (2021). Tetrahedral adaptive mesh refinement for two-phase flows using conservative level-set method. *International Journal for Numerical Methods in Fluids*, 93(2):481–503.

Banerjee, S., Lian, Y., Liu, Y., and Sussman, M. (2022). A new method for estimating bubble diameter at different gravity levels for nucleate pool boiling. *Journal of Heat Transfer*, 144(2):021601.

Berger, M. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512.

Bhat, P. P., Appathurai, S., Harris, M. T., Pasquali, M., McKinley, G. H., and Basaran, O. A. (2010). Formation of beads-on-a-string structures during break-up of viscoelastic filaments. *Nature Physics*, 6(8):625–631.

Biswas, R., Flaherty, J. E., and Arney, D. C. (1993). An adaptive mesh-moving and refinement procedure for one-dimensional conservation laws. *Applied Numerical Mathematics*, 11(4):259–282.

Breil, J., Harribey, T., Maire, P.-H., and Shashkov, M. (2013). A multi-material ReALE method with mof interface reconstruction. *Computers & Fluids*, 83:115–125.

Buchner, B. (2002). *Green water on ship-type offshore structures*. PhD thesis, Delft University of Technology, The Netherlands.

Cantwell, C., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., De Grazia, D., Yakovlev, S., Lombard, J.-E., Ekelschot, D., Jordi, B., Xu, H., Mohamied, Y., Eskilsson, C., Nelson, B., Vos, P., Biotto, C., Kirby, R., and Sherwin, S. (2015). Nektar++: An open-source spectral/hp element framework. *Computer Physics Communications*, 192:205–219.

Caretto, L., Gosman, A., Patankar, S., and Spalding, D. (1973). Two calculation procedures for steady, three-dimensional flows with recirculation. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics: Vol. II Problems of Fluid Mechanics*, pages 60–68. Springer.

Ceniceros, H. D., Roma, A. M., Silveira-Neto, A., Villar, M. M., et al. (2010). A robust, fully adaptive hybrid level-set/front-tracking method for two-phase flows with an accurate surface tension computation. *Communications in Computational Physics*, 8(1):51–94.

Chen, X. and Yang, V. (2014). Thickness-based adaptive mesh refinement methods for multi-phase flow simulations with thin regions. *Journal of Computational Physics*, 269:22–39.

Chiodi, R. and Desjardins, O. (2017). A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows. *Journal of Computational Physics*, 343:186–200.

Chirco, L. and Zaleski, S. (2023). An edge-based interface-tracking method for multiphase flows. *International Journal for Numerical Methods in Fluids*, 95:491–497.

Colicchio, G., Colagrossi, A., Greco, M., and Landrini, M. (2002). Free-surface flow after a dam break: a comparative study. *Ship Technology Research*, 49(3):95–104.

Cossali, G. E., Coghe, A., and Marengo, M. (1997). The impact of a single drop on a wetted solid surface. *Experiments in Fluids*, 22(6):463–472.

Cutforth, M., Barton, P. T., and Nikiforakis, N. (2021). An efficient moment-of-fluid interface tracking method. *Computers & Fluids*, 224:104964.

Ding, H., Spelt, P. D., and Shu, C. (2007). Diffuse interface model for incompressible two-phase flows with large density ratios. *Journal of Computational Physics*, 226(2):2078–2095.

Doherty, W., Phillips, T. N., and Xie, Z. (2023). A stabilised finite element framework for viscoelastic multiphase flows using a conservative level-set method. *Journal of Computational Physics*, 477:111936.

Dyadechko, V. and Shashkov, M. (2005). Moment-of-fluid interface reconstruction. *Los Alamos Report LA-UR-05-7571*, page 49.

Dyadechko, V. and Shashkov, M. (2008). Reconstruction of multi-material interfaces from moment data. *Journal of Computational Physics*, 227(11):5361–5384.

Evrard, F., Chiodi, R., van Wachem, B., and Desjardins, O. (2023). Conservative two-phase flow simulation using piecewise-parabolic interface reconstructions. *Bulletin of the American Physical Society*.

Ferziger, J. H. and Peric, M. (2002). *Computational Methods for Fluid Dynamics*. Berlin, Springer.

Fu, L., Litvinov, S., Hu, X. Y., and Adams, N. A. (2017). A novel partitioning method for block-structured adaptive meshes. *Journal of Computational Physics*, 341:447–473.

Gada, V. H. and Sharma, A. (2011). On a novel dual-grid level-set method for two-phase flow simulation. *Numerical Heat Transfer, Part B: Fundamentals*, 59(1):26–57.

Galera, S., Breil, J., and Maire, P.-H. (2011). A 2D unstructured multi-material cell-centered arbitrary Lagrangian–Eulerian (CCALE) scheme using MOF interface reconstruction. *Computers & Fluids*, 46(1):237–244.

Ginzburg, I. and Wittum, G. (2001). Two-phase flows on interface refined grids modeled with VOF, staggered finite volumes, and spline interpolants. *Journal of Computational Physics*, 166(2):302–335.

Greaves, D. (2004). A quadtree adaptive method for simulating fluid flows with moving interfaces. *Journal of Computational Physics*, 194(1):35–56.

Greaves, D. M. (2006). Simulation of viscous water column collapse using adapting hierarchical grids. *International Journal for Numerical Methods in Fluids*, 50(6):693–711.

Han, A., Chiodi, R., and Desjardins, O. (2024). Capturing thin structures in VOF simulations with two-plane reconstruction. *arXiv preprint arXiv:2403.10729*.

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

Hartmann, D., Meinke, M., and Schröder, W. (2011). A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1038–1052.

Hay, A. and Visonneau, M. (2005). Computation of free-surface flows with local mesh adaptation. *International Journal for Numerical Methods in Fluids*, 49(7):785–816.

Hergibo, P., Liang, Q., Phillips, T. N., and Xie, Z. (2024). A quadtree-based adaptive moment-of-fluid method for interface reconstruction with filaments. *Journal of Computational Physics*, 499:112719.

Hergibo, P., Phillips, T. N., and Xie, Z. (2023). A moment-of-fluid method for resolving filamentary structures using a symmetric multi-material approach. *Journal of Computational Physics*, 491:112401.

Hill, R. N. and Shashkov, M. (2013). The symmetric moment-of-fluid interface reconstruction algorithm. *Journal of Computational Physics*, 249:180–184.

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

Iserles, A. (2009). *A first course in the numerical analysis of differential equations*. Number 44. Cambridge University Press.

Islam, A., Sussman, M., Hu, H., and Lian, Y. (2022). Simulation of drop impact on substrate with micro-wells. *Physics of Fluids*, 34(6):062108.

Issa, R. I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1):40–65.

Jemison, M., Loch, E., Sussman, M., Shashkov, M., Arienti, M., Ohta, M., and Wang, Y. (2013). A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing*, 54(2):454–491.

Jemison, M., Sussman, M., and Shashkov, M. (2015). Filament capturing with the multimaterial moment-of-fluid method. *Journal of Computational Physics*, 285:149–172.

Koshizuka, S. (1995). A particle method for incompressible viscous flow with fluid fragmentation. *Comp. Fluid Dyn. J.*, 4:29–46.

Koshizuka, S. and Oka, Y. (1996). Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, 123(3):421–434.

Kucharik, M., Garimella, R. V., Schofield, S. P., and Shashkov, M. J. (2010). A comparative study of interface reconstruction methods for multi-material ALE simulations. *Journal of Computational Physics*, 229(7):2432–2452.

Lakdawala, A. M., Sharma, A., and Thaokar, R. (2016). A dual grid level set method based study on similarity and difference between interface dynamics for surface tension and radial electric field induced jet breakup. *Chemical Engineering Science*, 148:238–255.

Lemoine, A., Glockner, S., and Breil, J. (2017). Moment-of-fluid analytic reconstruction on 2D Cartesian grids. *Journal of Computational Physics*, 328:131–139.

Leung, S. and Zhao, H. (2009). A grid based particle method for moving interface problems. *Journal of Computational Physics*, 228(8):2993–3024.

Li, G., Lian, Y., and Sussman, M. (2013). Simulations of gas-liquid two-phase jet flows using the moment of fluid method. In *Fluids Engineering Division Summer Meeting*, volume 55560, page V01CT17A014. American Society of Mechanical Engineers.

Li, J. (2013). An arbitrary Lagrangian Eulerian method for three-phase flows with triple junction points. *Journal of Computational Physics*, 251:1–16.

Liang, Q. (2012). A simplified adaptive Cartesian grid system for solving the 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 69(2):442–458.

Liu, D. and Lin, P. (2008). A numerical study of three-dimensional liquid sloshing in tanks. *Journal of Computational Physics*, 227(8):3921–3939.

López, J., Hernández, J., Gómez, P., and Faura, F. (2004). A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics*, 195(2):718–742.

López, J., Hernández, J., Gómez, P., and Faura, F. (2005). An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics*, 208(1):51–74.

Ma, Z., Qian, L., Martínez-Ferrer, P., Causon, D., Mingham, C., and Bai, W. (2018). An overset mesh based multiphase flow solver for water entry problems. *Computers & Fluids*, 172:689–705.

Malik, M., Fan, E. S.-C., and Bussmann, M. (2007). Adaptive VOF with curvature-based refinement. *International Journal for Numerical Methods in Fluids*, 55(7):693–712.

Martin, J. C., Moyce, W. J., Martin, J., Moyce, W., Penney, W. G., Price, A., and Thornhill, C. (1952). Part iv. an experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 244(882):312–324.

Milcent, T. and Lemoine, A. (2020). Moment-of-fluid analytic reconstruction on 3D rectangular hexahedrons. *Journal of Computational Physics*, 409:109346.

Milcent, T. and Lemoine, A. (2024). An analytic approach for the moment-of-fluid interface reconstruction method on tetrahedral meshes. *Journal of Computational Physics*, page 112758.

Mirzadeh, M., Guittet, A., Burstedde, C., and Gibou, F. (2016). Parallel level-set methods on adaptive tree-based grids. *Journal of Computational Physics*, 322:345–364.

Monaghan, J. J. (1992). Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574.

Mukundan, A. A., Ménard, T., de Motta, J. C. B., and Berlemont, A. (2022). A hybrid moment of fluid–level set framework for simulating primary atomization. *Journal of Computational Physics*, 451:110864.

Nakajima, M. and Stevenson, D. J. (2014). Investigation of the initial state of the Moon-forming disk: Bridging SPH simulations and hydrostatic models. *Icarus*, 233:259–267.

Olsson, E. and Kreiss, G. (2005). A conservative level set method for two phase flow. *Journal of Computational Physics*, 210(1):225–246.

Osher, S. and Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49.

Patankar, S. (2018). *Numerical Heat Transfer and Fluid Flow*. London, Taylor & Francis.

Patel, T. and Lakdawala, A. (2018). A dual grid, dual level set based cut cell immersed boundary approach for simulation of multi-phase flow. *Chemical Engineering Science*, 177:180–194.

Pavlidis, D., Gomes, J. L., Xie, Z., Percival, J. R., Pain, C. C., and Matar, O. K. (2016). Compressive advection and multi-component methods for interface-capturing. *International Journal for Numerical Methods in Fluids*, 80(4):256–282.

Pilliod Jr, J. E. and Puckett, E. G. (2004). Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502.

Popinet, S. (2009). An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228(16):5838–5866.

Pozzetti, G., Jasak, H., Besseron, X., Rousset, A., and Peters, B. (2019). A parallel dual-grid multiscale approach to CFD–DEM couplings. *Journal of Computational Physics*, 378:708–722.

Remmerswaal, R. A. and Veldman, A. E. (2022). Parabolic interface reconstruction for 2D volume of fluid methods. *Journal of Computational Physics*, 469:111473.

Rider, W. J. and Kothe, D. B. (1998). Reconstructing volume tracking. *Journal of Computational Physics*, 141(2):112–152.

Rudman, M. (1997). Volume-tracking methods for interfacial flow calculations. *International Journal for Numerical Methods in Fluids*, 24(7):671–691.

Rudman, M. (1998). A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for Numerical Methods in Fluids*, 28(2):357–378.

Sato, Y. and Ničeno, B. (2013). A sharp-interface phase change model for a mass-conservative interface tracking method. *Journal of Computational Physics*, 249:127–161.

Scardovelli, R. and Zaleski, S. (1999). Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31(1):567–603.

Schofield, S. P., Christon, M. A., Dyadechko, V., Garimella, R. V., Lowrie, R. B., and Swartz, B. K. (2010). Multi-material incompressible flow simulation using the moment-of-fluid method. *International Journal for Numerical Methods in Fluids*, 63(8):931–952.

Sethian, J. A. and Smereka, P. (2003). Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics*, 35(1):341–372.

Shao, S. and Lo, E. Y. (2003). Incompressible SPH method for simulating Newtonian and non-Newtonian flows with a free surface. *Advances in Water Resources*, 26(7):787–800.

Shashkov, M. (2023). An adaptive moments-based interface reconstruction using intersection of the cell with one half-plane, two half-planes and a circle. *Journal of Computational Physics*, 494:112504.

Shashkov, M. and Kikinzon, E. (2023). Moments-based interface reconstruction, remap and advection. *Journal of Computational Physics*, 479:111998.

Shashkov, M. and Wendroff, B. (2004). The repair paradigm and application to conservation laws. *Journal of Computational Physics*, 198(1):265–277.

Spainhour, J. (2020). Quadratic moment-of-fluid interface reconstruction. Undergraduate Honors Thesis, The Florida State University College of Arts, and Science.

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

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

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

Tryggvason, G. (1988). Numerical simulations of the Rayleigh-Taylor instability. *Journal of Computational Physics*, 75(2):253–282.

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

Ubbink, O. (1997). *Numerical prediction of two fluid systems with sharp interfaces*. PhD thesis, Imperial College of Science, Technology and Medicine.

Unverdi, S. O. and Tryggvason, G. (1992). A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37.

Wang, J., Borthwick, A., and Taylor, R. E. (2004). Finite-volume-type VOF method on dynamically adaptive quadtree grids. *International Journal for Numerical Methods in Fluids*, 45(5):485–508.

Wang, Z. and Stern, F. (2022). Volume-of-fluid based two-phase flow methods on structured multiblock and overset grids. *International Journal for Numerical Methods in Fluids*, 94(6):557–582.

Wu, G., Eatock Taylor, R., and Greaves, D. (2001). The effect of viscosity on the transient free-surface waves in a two-dimensional tank. *Journal of Engineering Mathematics*, 40:77–90.

Xiao, F., Honma, Y., and Kono, T. (2005). A simple algebraic interface capturing scheme using hyperbolic tangent function. *International Journal for Numerical Methods in Fluids*, 48(9):1023–1040.

Xie, Z. (2010). *Numerical modelling of breaking waves under the influence of wind*. PhD thesis, University of Leeds.

Xie, Z. (2022). An implicit Cartesian cut-cell method for incompressible viscous flows with complex geometries. *Computer Methods in Applied Mechanics and Engineering*, 399:115449.

Xie, Z., Pavlidis, D., Percival, J. R., Gomes, J. L., Pain, C. C., and Matar, O. K. (2014). Adaptive unstructured mesh modelling of multiphase flows. *International Journal of Multiphase Flow*, 67:104–110.

Xie, Z., Pavlidis, D., Salinas, P., Percival, J. R., Pain, C. C., and Matar, O. K. (2016). A balanced-force control volume finite element method for interfacial flows with surface tension using adaptive anisotropic unstructured meshes. *Computers & Fluids*, 138:38–50.

Xie, Z. and Stoesser, T. (2020). A three-dimensional Cartesian cut-cell/volume-of-fluid method for two-phase flows with moving bodies. *Journal of Computational Physics*, 416:109536.

Ye, Z., Sussman, M., Zhan, Y., and Zhao, X. (2021). A decision-tree based moment-of-fluid (DTMOF) method in 3D rectangular hexahedrons. *arXiv preprint arXiv:2108.02533*.

Youngs, D. L. (1982). Time-dependent multi-material flow with large fluid distortion. *Numerical Methods for Fluid Dynamics*.

Zalesak, S. T. (1979). Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362.

Zeng, Y., Liu, H., Gao, Q., Almgren, A., Bhalla, A. P. S., and Shen, L. (2023). A consistent adaptive level set framework for incompressible two-phase flows with high density ratios and high Reynolds numbers. *Journal of Computational Physics*, 478:111971.

Zhang, D., Jiang, C., Liang, D., and Cheng, L. (2015). A review on TVD schemes and a refined flux-limiter for steady-state calculations. *Journal of Computational Physics*, 302:114–154.

Zhang, Y., Zou, Q., and Greaves, D. (2010). Numerical simulation of free-surface flow using the level-set method with global mass correction. *International Journal for Numerical Methods in Fluids*, 63(6):651–680.

Zheng, X., Lowengrub, J., Anderson, A., and Cristini, V. (2005). Adaptive unstructured volume remeshing–ii: Application to two-and three-dimensional level-set simulations of multiphase flow. *Journal of Computational Physics*, 208(2):626–650.

Zhou, Z., De Kat, J., and Buchner, B. (1999). A nonlinear 3D approach to simulate green water dynamics on deck. In *Proceedings of the seventh international conference on numerical ship hydrodynamics, Nantes, France*, pages 1–15.

Zwart, P., Raithby, G., and Raw, M. (1999). The integrated space-time finite volume method and its application to moving boundary problems. *Journal of Computational Physics*, 154(2):497–519.

Publications & Conference Presentations

Publications

- Hergibo, Philippe, Phillips, Timothy N., and Xie, Zhihua. "A moment-of-fluid method for resolving filamentary structures using a symmetric multi-material approach." *Journal of Computational Physics* (2023), vol. 491, p. 112401.

- Hergibo, Philippe, Liang, Qiuhua, Phillips, Timothy N., and Xie, Zhihua. "A quadtree-based adaptive moment-of-fluid method for interface reconstruction with filaments." *Journal of Computational Physics*, (2024), vol. 499, p. 112719.

- Hergibo, Philippe, Phillips, Timothy N., and Xie, Zhihua. "Resolving subgrid-scale structures for multiphase flows using a moment-of-fluid method." *Computers & Fluids*, 2024, vol. 285, p. 106455.

- Hergibo, Philippe, Phillips, Timothy N., and Xie, Zhihua. "An adaptive dual grid moment-of-fluid method for multiphase flows." *(under revision). Journal of Computational Physics*

Conference Presentations

- "A symmetric filamentary approach to the Moment-Of-Fluid method", *11th International Conference on Multiphase Flow (ICMF-2023)*, Kobe, JAPAN

- "A quadtree-based adaptive MOF method for interface reconstruction with filaments", *Welsh Mathematics Colloquium 2023*, Gregynog, WALES

- "A symmetric filamentary approach to the Moment-Of-Fluid method", *Cardiff University School of Engineering Research Conference*, Cardiff, WALES

- "Resolving subgrid scale structures for multiphase flows using the moment-of-fluid method", *APS 76th Annual Meeting of the Division of Fluid Dynamics Meeting*, Washington DC, USA