Title
Dispersion of finite size droplets and solid particles in isotropic turbulence

Permalink
https://escholarship.org/uc/item/3m79068c

Author
Rosso, Michele

Publication Date
2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Dispersion of finite size droplets and solid particles in isotropic turbulence

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mechanical and Aerospace Engineering

by

Michele Rosso

Dissertation Committee:
Professor Said Elghobashi, Chair
Professor Derek Dunn-Rankin
Professor William A. Sirignano

2016

DEDICATION

This dissertation is dedicated to my mum, my dad, my sister and all my friends.
I owe to each and everyone of them the person I am today.

"If Britney made it through 2007, you can make it through this Ph.D."

A friend

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF SYMBOLS

Upper Case Roman

A	Linear system matrix
B_i	Component of the body forces of the Navier-Stokes equations
C_i	Component of the convective term of the Navier-Stokes equations
D	Dimensionless droplets/particles diameter
$\widetilde{D}$	Droplets/particles diameter
$E(k,t)$	Turbulence kinetic energy spectrum
$E(t)$	Dimensionless turbulent kinetic energy
F	Numerical flux in the x direction
$\boldsymbol{F}$	Lagrangian forcing
F_i	Component of the Lagrangian forcing
$\boldsymbol{F}_R^m$	Dimensionless repulsive force acting on the m-th solid particle
G	Numerical flux in the y direction
H	Numerical flux in the z direction
I_p^m	Dimensionless momentum of inertia of the m-th solid particle
L	Dimensionless size of a cubical domain
$L_1(\alpha)$	Norm 1 of the absolute error of the numerical variable α
$L_\infty(\alpha)$	Norm infinity of the absolute error of the numerical variable α
$\widetilde{L}$	Reference length scale
L_x	Dimensionless domain size in the x direction
L_y	Dimensionless domain size in the y direction
L_z	Dimensionless domain size in the z direction
N	Number of grid points along each side of a cubical domain
N_d	Number of liquid droplets

N_l	Number of Lagrangian points on the surface of each particle
N_p	Number of solid particles
N_x	Number of grid nodes in the x direction
N_y	Number of grid nodes in the y direction
N_z	Number of grid nodes in the z direction
P_{2h}^{h}	Prolongation operator
P_i	Component of the pressure term of the Navier-Stokes equations
N_p	Number of solid particles
R_h^{2h}	Restriction operator
$R_{ij}(k)$	Velocity cross-correlation spectra
R_p	Dimensionless particle radius
R_{u_i}	Component of the fluid acceleration
S	Sign function
S_u	Skewness of the velocity derivatives
U_0	Dimensionless velocity root-mean-square at $t = 0$
$\boldsymbol{U}$	Lagrangian velocity
U_i^*	Component of the Lagrangian intermediate velocity
U_{rms}	Dimensionless velocity root-mean-square
$\widetilde{U}$	Reference velocity scale
V_i	Component of the viscous term of the Navier-Stokes equations
$\boldsymbol{X}$	Lagrangian position
$\boldsymbol{X}_l$	Position vector of the l-th Lagrangian force point

Lower Case Roman

b	Right hand side of the 1D Poisson's equation
d	Distance function
$\boldsymbol{e}_i$	Unit vector in the i-direction
e^k	Error at the k-th iteration
$\boldsymbol{f}$	Dimensionless IBM forcing term per unit volume
f_i	Dimensionless component of the IBM forcing term per unit volume
$\widetilde{f}_i$	Component of the IBM forcing term per unit volume

g_i	Dimensionless gravitational acceleration in the i-direction
$\widetilde{g}_i$	Gravitational acceleration in the i-direction
$\widetilde{g}$	Gravitational constant
i, j, k	Tensor indices
k	Dimensionless wave number
k_{max}	Largest resolved wave number
k_{min}	Smallest non-zero wave number
k_p	Dimensionless wave number of peak energy
m	Solid particle index
$\boldsymbol{n}$	Local normal vector to the interface
n_i	Component of the local normal vector to the interface
p	Dimensionless pressure
$\widetilde{p}$	Pressure
r_{ij}	Dimensionless velocity mean correlations
r^k	Residual at the k-th iteration
s_i	Dimensionless capillary forcing term in the i-direction
s_{ij}	Dimensionless strain-rate tensor
$\widetilde{s}_i$	Capillary forcing term in the i-direction
t	Dimensionless time
$\widetilde{\tau}_{ij}$	Deviatoric stress tensor component
u	Component along x of the dimensionless velocity vector ($u = u_1$)
$\boldsymbol{u}$	Dimensionless velocity vector
$\boldsymbol{u}_\Gamma$	Interface velocity vector
u_i	Component of the dimensionless velocity vector
u_i^*	Component of the first tentative velocity vector
u_i^{**}	Component of the second tentative velocity vector
$\widetilde{u}_i$	Velocity vector component
$\boldsymbol{u}_p$	Dimensionless velocity of the center of a solid particle
$\widetilde{\boldsymbol{u}}$	Velocity vector
v	Component along y of the dimensionless velocity vector ($v = u_2$)
w	Component along z of the dimensionless velocity vector ($w = u_3$)
w_1, w_2	Adam-Bashforth weights
$\boldsymbol{x}$	Dimensionless position vector

$x_{c,i}$	Dimensionless position of a droplet/particle center of mass (i-axis)
$x_{c,i}^0$	Dimensionless initial position of a droplet/particle center of mass (i-axis)
x_Γ	Interface position in the x-axis
x_i	Component of the dimensionless position vector
$\widetilde{x}_i$	Component of the position vector
$\boldsymbol{x}_p$	Dimensionless position of the particle center
$\widetilde{\boldsymbol{x}}$	Position vector

Upper Case Greek

Γ	Interface
Ω	Control volume
Ω^-	Region of the control volume occupied by fluid ”$-$”
Ω^+	Region of the control volume occupied by fluid ”$+$”
Ω_p^m	Dimensionless volume of the m-th solid particle
Φ_v	Volume fraction
Ψ	Dimensionless two/four-way coupling rate of change of TKE

Lower Case Greek

α	Generic variable
$\overline{\alpha}$	Analytical expression of a generic variable α
β	Coefficient of the non-separable Poisson's equation
δ	Dirac's delta function per unit volume
δ_h	Three-dimensional regularized three-point delta function
$\widetilde{\delta}$	Dirac's delta function
$\varepsilon(t)$	Turbulent kinetic energy dissipation rate
η	Dimensionless Kolmogorov's length-scale
η_μ	Dynamic viscosity ratio
η_ρ	Density ratio
κ	Twice the interface local mean curvature (dimensionless)
$\widetilde{\kappa}_m$	Interface local mean curvature

$\widetilde{\kappa}$	Twice the interface local mean curvature
λ	Dimensionless Taylor's length-scale
λ_0	Dimensionless Taylor's length-scale at $t = 0$
$\widetilde{\lambda}$	Taylor's length-scale
μ	Dimensionless dynamic viscosity
μ_ϵ	Discrete viscosity field
$\widetilde{\mu}$	Dynamic viscosity
$\widetilde{\mu}^-$	Dynamic viscosity in region ”$-$”
$\widetilde{\mu}^+$	Dynamic viscosity in region ”$+$”
ν	Dimensionless kinematic viscosity
$\boldsymbol{\omega}_p$	Dimensionless angular velocity of a solid particle
ϕ	Level set function (signed distance function)
ϕ_0	Initial condition for the level set reinitialization equation
φ_l	Azimuthal angle of the l-th Lagrangian force point
ψ	Phase field function
ρ	Dimensionless density
ρ_ϵ	Discrete density field
ρ_p^m	Dimensionless density of the m-th solid particle
ρ^*	Density weighted average
$\widetilde{\rho}$	Density
$\widetilde{\rho}^-$	Density in region ”$-$”
$\widetilde{\rho}^+$	Density in region ”$+$”
$\widetilde{\sigma}$	Surface tension coefficient
τ	Fictitious time in the level set reinitialization equation
τ_ℓ	Dimensionless integral time-scale
τ_η	Dimensionless Kolmogorov's time-scale
τ_{ij}	Deviatoric stress-tensor component
τ_λ	Dimensionless Taylor's time-scale
θ_l	Polar angle of the l-th Lagrangian force point
$\widetilde{\nu}$	Kinematic viscosity
$\xi(\alpha)$	Relative error of a generic variable α

Operators

$\partial\alpha/\partial\xi$	Partial derivative of α with respect to a variable ξ
$\delta\alpha/\delta\xi$	Discrete partial derivative of α with respect to a variable ξ
$\delta_2\alpha/\delta_2\xi$	Discrete second order partial derivative of α with respect to a variable ξ
$\langle\alpha\rangle$	Ensemble average of a variable α
$\nabla\alpha$	Dimensionless gradient of α
$\widetilde{\nabla}\widetilde{\alpha}$	Gradient of $\widetilde{\alpha}$
$\overline{\alpha}^{x_i}$	Second order interpolation of α along x_i
$\overline{\alpha}^{x_ix_j}$	Second order interpolation of α in the plane x_ix_j
$[\alpha]_\Gamma$	Intensity of the jump of a variable α at the interface Γ

Other symbols

$\mathcal{C}$	Coefficient in the CFL condition
$\mathcal{C}_c$	Coefficient in the CFL condition (convective term)
$\mathcal{C}_g$	Coefficient in the CFL condition (gravity term)
$\mathcal{C}_s$	Coefficient in the CFL condition (surface tension term)
$\mathcal{C}_v$	Coefficient in the CFL condition (viscous term)
d	Exponent in Olsson's relation
$\mathcal{D}_{x_i}$	Dimensionless dispersion along x_i
δt	Time step size
δx	Grid spacing in the x
δy	Grid spacing in the y
δz	Grid spacing in the z
ℓ	Dimensionless turbulence integral scale
ϵ	ACLSM smoothing parameter
$\mathfrak{Fr}$	Froude number
$\mathsf{i},\mathsf{j},\mathsf{k}$	Grid indices in the x, y, z direction
$I^s_{\mathsf{i},\mathsf{j},\mathsf{k}}$	Scalar cell centered at node $(\mathsf{i},\mathsf{j},\mathsf{k})$
$\mathfrak{La}$	Laplace number
n	Time level index
$\mathfrak{Oh}$	Ohnesorge number

$\mathfrak{Re}$	Reynolds number
$\mathfrak{Re}_\ell$	Reynolds number based on the integral length-scale
$\mathfrak{Re}_\eta$	Reynolds number based on the Kolmogorov's length-scale
$\mathfrak{Re}_\lambda$	Reynolds number based on the Taylor's length-scale
$\mathfrak{Re}_{\lambda_0}$	Reynolds number based on the Taylor's length scale at $t = 0$
$\mathfrak{We}$	Weber number

LIST OF ABBREVIATIONS

ACLSM	Accurate Conservative Level Set Method
AMG	Algebraic Multigrid
BC	Boundary Condition
CFL	Courant-Friedrichs-Lewy
CG	Conjugate Gradient
CGC	Coarse Grid Correction
CSF	Continuum Surface Force
DCA	Discretization Coarse-grid Approximation
DNS	Direct Numerical Simulation
FD	Finite Difference
FFT	Fast Fourier Transform
FMM	Fast Marching Method
FN	Face Normals
FSM	Fast Sweeping Method
FTM	Front-Tracking Method
GCA	Galerkin Coarse-grid Approximation
GFM	Ghost Fluid Method
GMG	Geometric Multigrid
GS	Gauss-Seidel
IBM	Immersed Boundary Method
LHS	Left-Hand Side
LS	Least Squares

LSM	Level Set Method
LSR	Least Squares Reconstruction
MG	Multigrid
NSE	Navier-Stokes Equations
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PPE	Pressure Poisson's Equation
RHS	Right-Hand Side
SLSM	Standard Level Set Method
SOR	Successive Over-Relaxation
TKE	Turbulence Kinetic Energy
VOF	Volume Of Fluid

ACKNOWLEDGMENTS

First and foremost I would like to thank my family, in particular my wonderful parents, Elmo and Mirella, and my amazing sister Francesca. Without their encouragement, support and love I would not be the person I am today.

I also owe a big thank to all my Italian friends from my hometown, high school, swim team and Politecnico. Despite an ocean kept us apart for seven long years, our bond never faded and, I am sure, never will.

My friends at UCI deserve my gratitude as well: my stay in Irvine would have been lonely and much less interesting without them. They are my family away from home. A special mention goes to my roommate Lio, who lived with me for the entire duration of my graduate studies and can now be regarded as my brother.

Space constrain prevents me from mentioning each and everyone of my friends, but rest assure guys: you are all in my hearth!

Finally I want to thank my adviser, Prof. Said Elghobashi, my committee members, and the department of Mechanical and Aerospace Engineering at UCI. Finally, a sincere thank goes to Prof. Derek Dunn-Rankin who served as my mentor in all these years.

CURRICULUM VITAE

Michele Rosso

EDUCATION

Master of Science in Aerospace Enginnering **2009**
Politecnico di Torino *Torino, Italy*

Bachelor of Science in Aerospace Enginnering **2007**
Politecnico di Torino *Torino, Italy*

RESEARCH EXPERIENCE

Graduate Research Assistant **2010–2016**
University of California, Irvine *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2015**
University of California, Irvine *Irvine, California*

WORK EXPERIENCE

Intern **2014–2015**
MSC Software *Irvine, California*

REFEREED JOURNAL PUBLICATIONS

Eulerian-Lagrangian bridge for the energy and dissipation spectra in isotropic turbulence **2013**
Theoretical and Computational Fluid Dynamics

REFEREED CONFERENCE PUBLICATIONS

DNS of liquid droplets in a turbulent flow **May 2015**
Blue Waters Symposium

Dispersion of finite size droplets and solid particles in isotropic turbulence **May 2016**
International Conference of Multiphase Flows

Dispersion of finite size droplets and solid particles in isotropic turbulence **June 2016**
Blue Waters Symposium

ABSTRACT OF THE DISSERTATION

Dispersion of finite size droplets and solid particles in isotropic turbulence

By

Michele Rosso

Doctor of Philosophy in Mechanical and Aerospace Engineering

University of California, Irvine, 2016

Professor Said Elghobashi, Chair

Turbulent disperse two-phase flows, of either fluid/fluid or fluid/solid type, are common in natural phenomena and engineering devices. Notable examples are atmospheric clouds, i.e. dispersed liquid water droplets and ice particles in a complex turbulent flow, and spray of fuel droplets in the combustion chamber of internal combustion engines. However, the physics of the interaction between a dispersed phase and turbulence is not yet fully understood. The objective of this study is to compare the dispersion of deformable finite size droplets with that of solid particles in a turbulent flow in the absence of gravity, by performing Direct Numerical Simulation (DNS). The droplets and the particles have the same diameter, of the order of the Taylor's microscale of turbulence, and the same density ratio to the carrier flow. The solid particle-laden turbulence is simulated by coupling a standard projection method with the Immersed Boundary Method (IBM). The solid particles are fully resolved in space and time without considering particle/particle collisions (two-way coupling). The liquid droplet-laden turbulence is simulated by coupling a variable-density projection method with the Accurate Conservative Level Set Method (ACLSM). The effect of the surface tension is accounted for by using the Ghost Fluid Method (GFM) in order to avoid any numerical smearing, while the discontinuities in the viscous term of the Navier-Stokes equation are smoothed out via the Continuum Surface Force approach. Droplet/droplet interactions are allowed (four-way coupling). The results presented here show that in isotropic turbulence the dispersion of liquid droplets in a given direction is larger than that of solid particles due to the reduced decay rate of turbulence kinetic energy via the four-way coupling effects of the droplets.

Chapter 1

Introduction

Turbulent disperse two-phase flows, of either fluid/fluid or fluid/solid type, are ubiquitous in natural phenomena and engineering devices such as bubble clouds, bubble columns and reactors in the chemical industry, and spraying of liquid fuel and paint. The study of the interaction of solid/liquid particles with turbulence is particularly important to understand the fundamental properties of clouds. Clouds are dispersions of drops and ice particles in a complex turbulent flow and are characterized by a wide range of spatial and temporal scales. Their evolution depends on the strong couplings between the dispersed phase and the dynamics of the turbulent carrier flow [68]. As pointed out by Bodenschatz et al. [9], many of the questions regarding the physics of clouds, particularly nucleation, growth, and precipitation of water particles, are yet to be answered. The dispersion of spray droplets in the combustion chamber of an internal combustion engine is another example of application that could benefit from a deeper understanding of the physics involved in liquid/gas interfaces. In this case, the turbulence in the ambient gas influences the development of the spray characteristics, i.e. spreading rate, penetration length and cone angle, and is, in turn, affected by the dispersed phase dynamics. Elghobashi [20] classified the disperse turbulent flows from the point of view of the type of interaction between the particles and the turbulence. The level of this interaction is determined by the volume fraction, Φ_v, occupied by the dispersed phase. For $\Phi_v \leq 10^{-6}$, the dominant effect is that of the turbulent carrier flow on the dynamics of the dispersed phase, whereas the influence of the particles/droplets on the turbulence is negligible. This interaction is called *one-way coupling.* For larger values of Φ_v, i.e. $10^{-6} \leq \Phi_v \leq 10^{-3}$, the momentum transfer from the particles cannot be ignored since it is large enough to alter the structure of the turbulence. This interaction is termed *two-way coupling.* Both the regimes discussed above are characterized by the absence of particle/particle (or droplet/droplet) interaction due to the low volume fraction, and thus they are sometimes referred to as *dilute suspensions.* When the volume fraction is increased further, i.e. $\Phi_v \geq 10^{-3}$, the interactions between particles/droplets has to be considered in addition to the two-way coupling effects between dispersed phase and turbulence. This interaction is named *four-way coupling* and characterizes the so-called *dense suspensions.* The behavior of dispersions in turbulent flows with one-way coupling is reasonably understood, at least in the limits posed by the incomplete understanding of turbulence itself even in particle-free flows. On the other hand, the two-way and particularly the four-way coupling regimes are still in the process of being fully investigated due to the highly nonlinear nature of the interactions in these flows.

Literature Review

Over the past decades, a large body of research has been devoted to both the theoretical and the numerical study of turbulent dispersed two-phase flows. In 1922, Taylor [76] derived an analytic expression for the time evolution of the dispersion of a *fluid particle* in stationary isotropic turbulence. The term dispersion, hereinafter indicated by $\mathcal{D}_{x_i}$, is defined as the mean square displacement of solid or fluid particles along the direction x_i. Taylor's solution showed that the relation between $\mathcal{D}_{x_i}$ and the time, t, is quadratic, i.e. $\mathcal{D}_{x_i}\sim t^2$, for short times and linear, i.e. $\mathcal{D}_{x_i} \sim t$, for long times. Batchelor [8] and **?**] extended Taylor's theory to homogeneous turbulence. Tchen [77] linearized the equation of motion for a sphere in stationary homogeneous turbulence under the assumptions that the sphere diameter is smaller than the Kolmogorov's length scale of the turbulence and the relative motion between the particle and the carrier flow follows Stokes' law. Tchen's equation was later improved by Maxey and Riley [42] in their study of a small rigid sphere in a non-uniform flow. Reeks [58] and Pismen and Nir [55] analyzed the behavior of heavy particle dispersion in stationary, homogeneous, isotropic turbulence and showed that the effect of particle inertia is to increase the eddy diffusivity over that of the fluid in the absence of particles. Kim et al. [33] investigated the motion of a freely-moving sphere in an initially stationary or oscillating fluid and derived an equation for the particle motion that accounts for the effects of large relative acceleration or deceleration of the particle and the initial relative velocity between the fluid and the particle. Theoretical studies were also performed in order to model other important phenomena regarding disperse multi-phase flow, particularly in the context of spray combustion theory. The reader is referred to the comprehensive review by Sirignano [69], and the book by the same author [70] for an in-depth analysis of these studies. While theoretical approaches offer invaluable insight into the dynamics of particle dispersions in turbulence, they often rely on simplifying assumptions that greatly limit their range of applicability. With the increase in computational power in the last three decades, numerical simulations that do not rely on turbulence modeling, i.e. *Direct Numerical Simulation* (DNS), offered a valuable alternative for the study of dispersed turbulent flows. The early numerical studies accounted for the one-way coupling effects via either the *two-fluid* methods or the *Lagrangian* methods. In the *two-fluid* model, the dispersed phase is treated as a continuum described by a set of conservation laws similar to those of the carrier flow, with the momentum and energy exchange between the phases accounted for by adding source and sink terms in the momentum and energy equations for the particulate phase. This approach was used by Druzhinin and Elghobashi [19] to study bubble-laden isotropic decaying turbulence via DNS and by Février et al. [26] to investigate the velocity distribution of dilute suspensions of heavy particles in gas-solid turbulent flows. In the Lagrangian approach, the carrier flow equations are solved throughout the whole computational domain, while the particulate phase is represented by computational particles whose position, momentum, and energy are tracked by solving the equations of particle motion [20]. Riley and Patterson [59] were the first to use the Lagrangian approach in DNS to calculate the autocorrelation and the mean-square displacement of small particles in decaying isotropic turbulence. In their work, the particle diameter was smaller than the Kolmogorov's length scale of the carrier flow and only the *Stokes' drag* was included in the equation of particle motion. Their results showed that the velocity autocorrelation was increased for increasing values of particle inertia. McLaughlin [44] used DNS to examine particle deposition on the wall in a vertical channel flow and computed the particle trajectories by including only the *Stokes' drag* and *Saffman's lift force* in the particle motion equation. Squires and Eaton [73] measured the dispersion of heavy particles by performing DNS of forced and decaying isotropic turbulence. Their results showed that the effect of particle inertia is to increase the eddy diffusivity over that of the fluid in the absence of particle drift, as was demonstrated by the theoretical studies of Reeks [58] and Pismen and Nir [55]. Elghobashi and Truesdell [21] studied the dispersion of three different solid particles (corn, copper and glass) in decaying isotropic turbulence by using DNS and obtained good agreement with the experimental work by Snyder and Lumley [71]. They also showed that that the dispersion of small solid particles in zero gravity agrees with Taylor's theory [76]. Later works extended the Lagrangian approach to include the two-way coupling effects, see for example the research by Ferrante and Elghobashi [25] and Squires and Eaton [72]. The effects of the two-way interaction on particle dispersion in decaying isotropic turbulence was also investigated by Elghobashi and Truesdell [22]. They showed that, in zero gravity, the two-way coupling enhances the alignment of the surrounding fluid velocity vector with the direction of the solid particle trajectory. This alignment reduces the mean-square relative velocity and increases the Lagrangian velocity autocorrelation coefficient of the solid particle, the fluid point and the surrounding fluid, and thus increases the dispersion of the solid particles. However, the fluid point dispersion decreases because the larger inertia of the solid particles increases the decay rate of turbulence kinetic energy. All the numerical studies presented so far rely on the *point particle* approximation, i.e. the assumption is made that the instantaneous particle velocity can be calculated via the equations of particle motion. This is justified only if the following two conditions hold:

1. the particles/droplets have a diameter, $\widetilde{D}$, smaller than the Kolmogorov length scale, η, of the turbulent carrier flow, i.e. $\widetilde{D} << \eta$, and
2. the Reynolds number of the particles/droplets, $\mathfrak{Re}_p$, satisfies $\mathfrak{Re}_p < 1$.

When the above two conditions are not satisfied, the *point particle* assumption is no more valid since it is not possible to derive an analytic solution for the particle motion in a turbulent flow as proved by Lumley [40]. Consequently, an accurate numerical solution can be obtained only by fully resolving the flow field around each moving particle. This approach was used for the DNS of a single particle (Bagchi and Balachandar [2], Bagchi and Balachandar [3], Burton and Eaton [12]) and a collection of thousands of particles (Cate et al. [13], Lu and Tryggvason [37], Uhlmann [80]). Lucci et al. [39] pursued this technique to study the two-way coupling interaction of up to 6400 *finite size* solid particles and turbulence. Their results showed that the presence of the solid particles enhances the turbulence kinetic energy decay rate. The reader is referred to [4] for an exhaustive review of the research on dispersed turbulent flows.

Objectives

The objective of this study is to compare the dispersion of deformable finite size droplets with that of solid particles in a turbulent flow by performing DNS. The solid particles and liquid droplets have the same diameter and the same density ratio to the carrier flow. Both the dispersed phase and the carrier flow are fully resolved in 3D-space and time and all the scales of the turbulent motion are simultaneously resolved down to the smallest relevant length- and time-scales. In this work, the unsteady three-dimensional Navier-Stokes and continuity equations are solved throughout the whole computational domain, including the interior of the liquid droplets/solid particles. In the liquid droplet case, the motion and deformation of the interface are captured implicitly by using the *Accurate Conservative Level Set Method* (ACLSM) [18]. The effect of the surface tension is accounted for accurately via the *Ghost Fluid Method* (GFM) [31], while the discontinuities in the viscous term are smoothed out across the interface by means of the *Continuum Surface Force* (CSF) approach. Finally, a variable density projection method is used to impose the incompressibility constrain. In the solid particle case, the tracking of the interface is performed in a Lagrangian fashion by using the *Immersed Boundary Method* (IBM) [52, 79] and the divergence-free constrain on the velocity field is enforced by using a standard projection method. This thesis is structured as follows. Chapter 2 details the mathematical formulation of the problem, while Chapter 3 is devoted to the numerical techniques used to discretize the governing equations. Chapter 4 describes the tests that were performed to validate the computational code developed for this study. The DNS results regarding the dispersion of droplets/particles in decaying isotropic turbulence are reported in Chapter 5. Finally, Chapter 6 presents a summary of this work and discusses future research developments.

Technical challanges

The numerical simulation of deformable droplets poses three main challenges:

1. the numerical localization and transport of the interface (also called *front*) separating the phases involved,

2. the numerical treatment of the material properties that discontinuously change by many order of magnitude between the phases, and
3. the numerical discretization of a singular forcing term, i.e. the surface tension, acting only on the front.

Among the methods proposed for the numerical transport of the front over the past decades, the *Volume Of Fluid* (VOF) method, the *Front-Tracking Method* (FTM), and the *Level Set Method* (LSM) have become standard numerical tools for multi-phase simulations. The VOF [65] describes the interface through a volume fraction scalar, thus ensuring, at least theoretically, discrete mass conservation. Nonetheless the scalar function is discontinuous across the interface and thereby a specific advection scheme is required at the cost of putting constrains on the time step size and accuracy of the simulation. Moreover, the calculation of geometric quantities such as interface normals and curvature could be challenging. The FTM introduced by Unverdi and Tryggvason [81] makes use of an unstructured moving mesh for the discretization of the interface. The advantage of this method is that the transport of the mesh is purely Lagrangian; unfortunately any topological change or front-to-front interaction has to be handled manually by means of re-meshing, thus causing adverse effects on mass conservation. The LSM, devised by Osher and Sethian [49], represents the interface as the zero iso-level of a smooth function, typically a signed distance function. The latter is transported in an Eulerian fashion by solving a standard convection *Partial Differential Equation* (PDE) and kept smooth through a re-initialization process. Due to the smoothness of the level set function, the normals and curvature can be computed easily. The main drawback of the LSM is its inherent lack of conservation, although many strategies have been proposed to address the issue. In particular Enright et al. [23] used Lagrangian marker particles to correct the front position after the Eulerian advection step, while Sussman and Puckett [74] coupled the LSM with the VOF in order to take advantage of the good conservation properties of the latter. While these techniques alleviate the issue, they also lose the original simplicity of the LSM. A simpler method was proposed by Olsson and Kreiss [46] in which they replaced the signed distance function of the classical LSM with a hyperbolic tangent profile that is advected and re-initialialized using conservative equations. This approach was later improved by Desjardins et al. [18] under the name ACLSM and used for the simulation of turbulent atomization. In this study, the ACLSM was used to simulate deformable liquid droplets because of its simplicity, adaptivity, ease of parallelization and intrinsic ability of handling topological changes naturally.

Computational code

A computation code of more than 30,000 lines was developed for this study. The code was written in Fortran 2003 and designed according to modern object-oriented programming techniques. Furthermore, it includes distributed-memory parallelism using the MPI standard. DNS of turbulent flows is very demanding in terms of computational power and memory requirements, mostly because the computational grids need to be fine enough to accurately resolve the smallest flow structures. As the Taylor's scale Reynolds number is increased, such structures become smaller and smaller and consequently the mesh has to be refined accordingly. Moreover, an accurate time history of the flow is sought in order to compute time-dependent statistics, thus reducing the time-step interval needed to advance the solution in time. In response to these requirements, parallel computing became a standard tool in DNS codes. The code implemented for this research employs a parallel approach known as *3D domain decomposition*: the computational domain is partitioned along the three spatial dimensions into box-shaped sub-domains and the data associated with each sub-domain are hosted and operated on by a single CPU. This setup is particularly convenient in that it allows using up to $\mathcal{O}(N^3)$ CPUs where N is the number of grid points per direction. Thus the workload per single CPU is limited and it is unlikely that the memory saturation mark be reached even for very fine grids. Nevertheless the efficient implementation of inter-processor communications, especially for data transposition in the context of the *Fast Fourier Transform* (FFT), is not trivial and required considerable effort during the initial development of the code. The demand for computational efficiency is even more crucial when a two-phase flow is considered since the *Pressure Poisson's Equation* (PPE) is non-separable and therefore cannot be solved via a standard FFT. As a consequence the PPE has to be solved via an iterative method for sparse liner systems. In this work, the linear system associated to the PPE is solved by using the multigrid-preconditioned conjugate gradient provided by the PETSc library [5, 6, 7].

Chapter 2

Mathematical Model

The first step in a numerical study is the definition of a set of equations that model the physical phenomena under investigation. A suitable mathematical model for multi-phase flows is required to describe:

1. the motion of the fluids in the domain of interest, and
2. the motion of the interface between the different phases.

Item 1. in the above list is the topic of Section 2.1, while item 2. is dealt with in Section 2.2. In the following, all the variables are considered dimensionless unless they are superscripted with the diacritic symbol "$\sim$". It should be noted that the present chapter focuses exclusively on the mathematical description of the governing equations of multi-phase flows and interface motion, without considering any model that is specific to a discretization method.

2.1 Governing Equations for Incompressible Multi-Phase Flows

Before describing the governing equations for a multi-phase system, it is instructive to review the governing equation for a single-phase incompressible flow, namely the continuity equation

$$\frac{\partial \widetilde{u}_i}{\partial \widetilde{x}_i} = 0 \tag{2.1}$$

and the momentum equations

$$\frac{\partial \widetilde{u}_i}{\partial \widetilde{t}} + \frac{\partial}{\partial \widetilde{x}_j}(\widetilde{u}_i \widetilde{u}_j) = -\frac{1}{\widetilde{\rho}} \left[\frac{\partial \widetilde{p}}{\partial \widetilde{x}_i} + \frac{\partial \widetilde{\tau}_{ij}}{\partial \widetilde{x}_j} + \widetilde{\rho}\, \widetilde{g}_i \right] . \tag{2.2}$$

In Eq. (2.1) and (2.2), $\widetilde{u}_i$ is the component of the velocity vector $\widetilde{\boldsymbol{u}}$ in the i-direction, $\widetilde{\rho}$ the fluid density, $\widetilde{p}$ the pressure and $\widetilde{g}_i$ the gravitation acceleration in the i-direction. The present study considers only Newtonian fluids, for which the deviatoric stress tensor $\widetilde{\tau}_{ij}$ is defined as:

$$\widetilde{\tau}_{ij} = \widetilde{\mu} \left(\frac{\partial \widetilde{u}_i}{\partial \widetilde{x}_j} + \frac{\partial \widetilde{u}_j}{\partial \widetilde{x}_i} \right) , \tag{2.3}$$

where $\widetilde{\mu}$ is the dynamic viscosity. It should be noted that both $\widetilde{\rho}$ and $\widetilde{\mu}$ are constant for an incompressible single-phase flow. Consider now a volume Ω filled with two immiscible incompressible fluids of different phases. The two fluids are separated by an interface Γ that splits Ω into the sub-regions Ω^+, occupied by fluid "+", and Ω^-, occupied by fluid "$-$". Each phase is characterized by different material properties, i.e. $\widetilde{\mu}^+$, $\widetilde{\rho}^+$ in Ω^+ and $\widetilde{\mu}^-$, $\widetilde{\rho}^-$ in Ω^-. A typical configuration for this case is sketched in Figure 2.1. Since we are not considering *reactive* interfaces, i.e. the velocity field is continuous at the interface, Eq. (2.1) still holds in the whole domain Ω. In contrast, Eq. (2.2) can be used to describe the behavior of each of the two fluids only in their respective domains, but fails to model the physics at the interface Γ. As pointed out in the review paper by Scardovelli and Zaleski [65], the single-phase *Navier-Stokes Equations* (NSE) can be extended to model multi-phase flows by either:

1. re-writing Eq. (2.2) in weak form so that it can be used everywhere in the whole domain, including the interface, or
2. using Eq. (2.2) as is in each bulk-phase domain with proper jump conditions at the interface.

Approach 1. results in the so-called *whole-domain formulation* discussed in Subsection 2.1.1, while approach 2. leads to the *jump-conditions formulation* presented in Subsection 2.1.2.

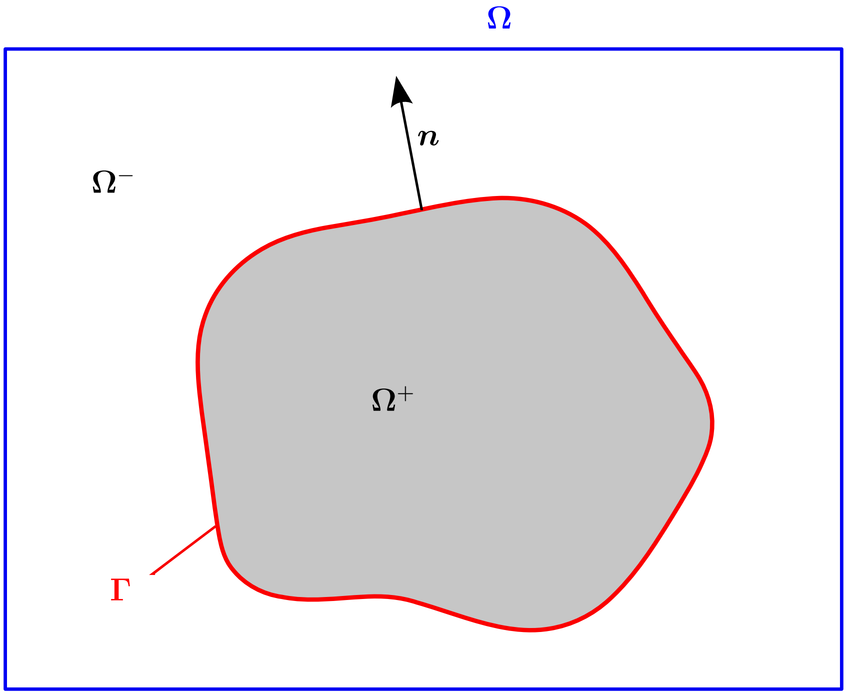


Figure 2.1 Typical configuration of a two-fluid system.

2.1.1 Whole-Domain Formulation

The whole-domain formulation of the NSE for multi-phase systems reads:

$$\frac{\partial \widetilde{u}_i}{\partial \widetilde{t}} + \frac{\partial}{\partial \widetilde{x}_j}(\widetilde{u}_i \widetilde{u}_j) = -\frac{1}{\widetilde{\rho}} \left[\frac{\partial \widetilde{p}}{\partial \widetilde{x}_i} + \frac{\partial \widetilde{\tau}_{ij}}{\partial \widetilde{x}_j} + \widetilde{\rho}\, \widetilde{g}_i + \widetilde{f}_i + \widetilde{s}_i \right] , \tag{2.4}$$

where the additional volumetric forcing terms $\widetilde{f}_i$ and $\widetilde{s}_i$ accounts for the presence of a solid boundary and the effect of capillary forces, respectively. When only fluid/solid interfaces are considered, $\widetilde{s}_i$ is neglected. In contrast, when the only interfaces present are of fluid/fluid type, $\widetilde{f}_i$ is set to zero. Equation (2.4) is essentially a weak form of the classic NSE and therefore it is valid everywhere in Ω despite the discontinuities at the interface. The forcing term $\widetilde{s}_i$ is defined as:

$$\widetilde{s}_i = 2\, \widetilde{\sigma}\, \widetilde{\kappa}_m\, \widetilde{\delta}\, n_i \ , \tag{2.5}$$

where $\widetilde{\sigma}$ is the surface tension coefficient, $\widetilde{\kappa}_m$ the interface mean curvature, n_i the i-component of the unit normal to Γ and $\widetilde{\delta}$ a one-dimensional Dirac delta function centered at the interface. It should be emphasized that Eq. (2.4) is singular at the interface because of the jumps in material properties and the singular terms $\widetilde{f}_i$ and $\widetilde{s}_i$ and therefore special numerical treatment is required for its discretization. The mean curvature $\widetilde{\kappa}_m$ is computed from the interface normal as:

$$\widetilde{\kappa}_m = -\frac{1}{2} \frac{\partial n_i}{\partial \widetilde{x}_i} \ . \tag{2.6}$$

In order to avoid the factor two in Eq. (2.5), it is common to define the variable $\widetilde{\kappa}$ as twice the mean curvature, i.e.

$$\widetilde{\kappa} = -\frac{\partial n_i}{\partial \widetilde{x}_i} \ , \tag{2.7}$$

and use it to re-write Eq. (2.5) as:

$$\widetilde{s}_i = \widetilde{\sigma}\,\widetilde{\kappa}\,\widetilde{\delta}\,n_i \; . \tag{2.8}$$

Hereinafter, the term *curvature* will be used to refer to $\widetilde{\kappa}$. More details on the derivation of this formulation can be found in the work by Tryggvason et al. [78] and Chang et al. [14].

2.1.2 Jump-Conditions Formulation

Instead of modifying the single-phase NSE via an additional term to account for the surface tension effects, one could directly use Eq. (2.2) in each separate bulk-phase domain together with proper *jump conditions* at the interface. In the following, the notation $[\alpha]_\Gamma$ indicates the *intensity* of the jump of a variable α at the interface Γ:

$$[\alpha]_\Gamma = \alpha^+ - \alpha^- \; . \tag{2.9}$$

The jump conditions at the interface between two immiscible, incompressible, Newtonian fluids are:

$$[\widetilde{\mu}]_\Gamma = \widetilde{\mu}^+ - \widetilde{\mu}^- \; , \tag{2.10}$$

$$[\widetilde{\rho}]_\Gamma = \widetilde{\rho}^+ - \widetilde{\rho}^- \; , \tag{2.11}$$

$$[\widetilde{\boldsymbol{u}}]_\Gamma = \widetilde{\boldsymbol{u}}^+ - \widetilde{\boldsymbol{u}}^- = 0 \; , \tag{2.12}$$

$$[\widetilde{p}]_\Gamma = \widetilde{p}^+ - \widetilde{p} \quad = \widetilde{\sigma}\,\widetilde{\kappa} + 2\,[\widetilde{\mu}]_\Gamma\, \boldsymbol{n} \cdot \widetilde{\nabla}\widetilde{\boldsymbol{u}} \cdot \boldsymbol{n}^t \; . \tag{2.13}$$

Equation (2.10) and (2.11) account for the jumps of material properties between the two phases, while Eq. (2.12) enforces the continuity of the velocity field at the interface, as required for a non-reactive interface.

The pressure jump in Eq. (2.13) is given by Landau and Lifshitz [34] and Brackbill et al. [10] and depends on

- the surface tension effects via the term $\widetilde{\sigma}\widetilde{\kappa}$, where the curvature $\widetilde{\kappa}$ is again defined according to Eq. (2.7), and
- the shear stresses jump via the term $2\,[\widetilde{\mu}]_\Gamma\,\boldsymbol{n}\cdot\widetilde{\nabla}\widetilde{\boldsymbol{u}}\cdot\boldsymbol{n}^t$.

This form of the NSE for multi-phase flows has been used by Desjardins et al. [18] and Kang et al. [31].

2.1.3 Normalization of the Governing Equations

In order to minimize the number of parameters involved in the physical models introduced so far, it is convenient to recast the governing equations in dimensionless form. In this work, the fluid "$-$" is considered the reference fluid, therefore $\widetilde{\mu}^-$ and $\widetilde{\rho}^-$ are taken to be the reference values for the dynamic viscosity and density respectively. Consequently, the dimensionless density ρ and viscosity μ are defined as follows:

$$\rho = \begin{cases} 1 & \text{in } \Omega^- \\ \eta_\rho & \text{in } \Omega^+ \end{cases} \qquad \text{with} \qquad \eta_\rho = \frac{\widetilde{\rho}^+}{\widetilde{\rho}^-} = \text{ density ratio }, \tag{2.14}$$

$$\mu = \begin{cases} 1 & \text{in } \Omega^- \\ \eta_\mu & \text{in } \Omega^+ \end{cases} \qquad \text{with} \qquad \eta_\mu = \frac{\widetilde{\mu}^+}{\widetilde{\mu}^-} = \text{ dynamic viscosity ratio }. \tag{2.15}$$

The reference value for the velocity and length scales are $\widetilde{U}$ and $\widetilde{L}$ respectively. Both of them are chosen according to the specific system being investigated.

Thus, the *Reynolds number* $\mathfrak{Re}$ is defined as:

$$\mathfrak{Re} = \frac{\widetilde{\rho}^- \, \widetilde{U}\widetilde{L}}{\widetilde{\mu}^-} \ , \tag{2.16}$$

the *Weber number* $\mathfrak{We}$ as:

$$\mathfrak{We} = \frac{\widetilde{\rho}^- \, \widetilde{U}^2\widetilde{L}}{\widetilde{\sigma}} \ , \tag{2.17}$$

and the *Froude number* as:

$$\mathfrak{Fr} = \frac{\widetilde{U}^2}{\widetilde{g} \, \widetilde{L}} \ , \tag{2.18}$$

where $\widetilde{g}$ is the gravitational constant. Using the above definitions, Eq. (2.4) can be re-cast in the following dimensionless form:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{1}{\rho}\left[\ \frac{\partial p}{\partial x_i} + \frac{1}{\mathfrak{Re}}\frac{\partial \tau_{ij}}{\partial x_j} + \frac{\rho g_i}{\mathfrak{Fr}} + f_i + s_i \ \right] \ , \tag{2.19}$$

where g_i is the dimensionless gravitational acceleration in the i-direction, and s_i the dimensionless surface tension term given by:

$$s_i = \frac{1}{\mathfrak{We}} \ \frac{\kappa \ \delta}{\rho} \ n_i \ . \tag{2.20}$$

Similarly, the non-dimensional momentum conservation in the jump conditions formulation can be written as:

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{1}{\rho}\left[\ \frac{\partial p}{\partial x_i} + \frac{1}{\mathfrak{Re}}\frac{\partial \tau_{ij}}{\partial x_j} + \frac{\rho g_i}{\mathfrak{Fr}} \ \right] \ , \tag{2.21}$$

together with the following dimensionless jump conditions:

$$[\mu]_\Gamma = \eta_\mu - 1 \ , \tag{2.22}$$

$$[\rho]_\Gamma = \eta_\rho - 1 \ , \tag{2.23}$$

$$[\boldsymbol{u}]_\Gamma = \boldsymbol{u}^+ - \boldsymbol{u}^- = 0 \ , \tag{2.24}$$

$$[p]_\Gamma = p^+ - p^- \quad = \frac{\kappa}{\mathfrak{We}} + \frac{2}{\mathfrak{Re}} [\mu]_\Gamma \, \boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{n}^t \ . \tag{2.25}$$

2.2 Mathematical Modeling of Moving Interfaces

One of the main challenges in the study of multi-phase flows is the accurate characterization of the motion of the interface $\Gamma(t)$ separating the phases involved. Some key issues that need to be addressed by a reliable mathematical model are:

- changes of topology, merging and breaking;
- formation of sharp corners, cusps, and singularities;
- dependence of the interface motion on delicate geometric quantities such as curvature and normal direction;
- coupling between the interface motion and the underlying velocity field moving the interface.

Among the methods proposed for the numerical transport of moving fronts over the past decades, the *Volume Of Fluid* (VOF) method, the *Front-Tracking Method* (FTM), the *Immersed Boundary Method* (IBM) and the *Level Set Method* (LSM) have become standard numerical tools for multi-phase simulations. The VOF method [65] describes the interface through a volume fraction scalar, thus ensuring, at least theoretically, discrete mass conservation. Nonetheless the scalar function is discontinuous across the interface and thereby a specific advection scheme is required at the cost of putting constrains on the time step size and accuracy of the simulation. Moreover, the calculation of geometric quantities such as interface normals and curvature could be challenging. The IBM by Peskin [52] and the FTM by Unverdi and Tryggvason [81] make use of an unstructured moving mesh for the discretization of the interface. In both methods the transport of the mesh is purely Lagrangian and thus they rely on delta functions to link the Lagrangian variables defined at the interface with the flow field variables defined on an Eulerian grid. Unfortunately any topological change or front-to-front interaction has to be handled manually by means of mesh rearrangements, thus causing adverse effects on mass conservation. The LSM, pioneered by the work by Osher and Sethian [49], represents the interface as the zero iso-level of a smooth function, typically a signed distance function. The latter is transported in an Eulerian fashion by solving a standard convection *Partial Differential Equation* (PDE) and kept smooth through a re-initialization process. In the present study, the IBM is used to model solid spherical particles, while the LSM is used to model liquid droplets.

2.2.1 Immersed Boundary Method

The IBM was originally developed by Peskin [52] to study flow patterns around heart valves. The distinguishing feature of this method was that the entire simulation was carried out on a Cartesian grid, which did not conform to the geometry of the heart, using a novel approach to impose the effect of the immersed boundary on the flow. Even though the original formulation was intended to model *elastic* membranes, the IBM has been extended to treat other types of fluid-structure interactions, including the interaction of a fluid with a *solid* boundary. This subsection will summarize the key ideas behind the IBM. For a more in-depth analysis, the reader is referred to the review papers by Peskin [53] and Mittal and Iaccarino [45]. The starting point of the IBM formulation is a Lagrangian perspective: the immersed boundary $\Gamma(t)$ is defined as a parametric surface via a set of curvilinear coordinates

(q, r). The position at time t in Cartesian coordinates of each material point in Γ is given by $\boldsymbol{X}(q, r, t)$ and its velocity by $\boldsymbol{U}(q, r, t)$. Since a *no-slip condition* is required at the boundary, $\boldsymbol{U}$ needs to satisfy $\boldsymbol{U}(q, r, t) = \boldsymbol{u}(\boldsymbol{X}(q, r, t), t)$. This requirement is imposed by introducing a boundary force $\boldsymbol{F}(q, r, t)$ acting on each material point in Γ. The effect of $\boldsymbol{F}$ is transferred to the fluid by defining the forcing term f_i in Eq. (2.19) as:

$$f_i = \boldsymbol{f}(\boldsymbol{x}, t) \cdot \boldsymbol{e}_i = \left(\int_\Gamma \boldsymbol{F}(q, r, t)\, \delta(\boldsymbol{x} - \boldsymbol{X}(q, r, t))\, dq\, dr \right) \cdot \boldsymbol{e}_i \ , \tag{2.26}$$

where δ is the one dimensional *Dirac delta function.* Similarly the velocity at each point of the boundary can be expressed in terms of the Eulerian velocity of the underlying fluid as:

$$\boldsymbol{U}(q, r, t) = \boldsymbol{u}(\boldsymbol{X}(q, r, t), t) = \int_\Omega \boldsymbol{u}(\boldsymbol{x}, t)\, \delta(\boldsymbol{x} - \boldsymbol{X}(q, r, t))\, d\boldsymbol{x} \ . \tag{2.27}$$

Equation (2.26) and (2.27) are *interaction equations* in that they convert from Lagrangian to Eulerian variables and vice-versa. The flow field is still described by the whole-domain Eq. (2.19), where the surface tension term s_i is now neglected. When the IBM is used to treat solid objects, the only effect of the forcing term f_i is the enforcement of the no-slip condition. Therefore the NSE can be solved throughout the entire computational domain, including the interior of the solid body. In this case, the IBM would correctly model the flow inside and outside the immersed boundary. However, an equation of motion has to be provided for each material point $\boldsymbol{X}(q, r, t)$ so that the interface can be tracked explicitly. For the specific case of spherical solid particles, this can be done by using the kinematic equations of rigid body motion

$$\boldsymbol{U}(\boldsymbol{X}) = \boldsymbol{u}_p + \boldsymbol{\omega}_p \times (\boldsymbol{X} - \boldsymbol{x}_p) \ , \tag{2.28}$$

where $\boldsymbol{x}_p$ and $\boldsymbol{u}_p$ are the position and linear velocity of the particle center, and $\boldsymbol{\omega}_p$ the particle angular velocity. These three quantities are computed by solving the equations of linear and angular motion of a rigid body. It should be emphasized that, by using Eq. (2.28), both translation and rotation of the particle are accounted for. The equations of conservation of linear and angular momenta for a solid spherical particle simulated via the IBM are described in Subsection 3.3.2. The IBM has been used successfully used for the *Direct Numerical Simulation* (DNS) of moving solid particles in a turbulent flow, see for example the work by Uhlmann [79] and Lucci et al. [39].

2.2.2 Standard Level Set Method

The *Standard Level Set Method* (SLSM) was originally developed by Osher and Sethian [49] to model fronts propagating with curvature-dependent speed. Over the years since its inception, it has been refined and applied to a multitude of moving interface problems in a wide variety of different fields. Examples of applications include semiconductor manufacturing, combustion and detonation, computer vision, surface-tension-driven flows and seismology. The SLSM represents $\Gamma(t)$ *implicitly* as the zero level set of a smooth function $\phi(\boldsymbol{x}, t)$, i.e. $\Gamma(t) = \{\boldsymbol{x} \mid \phi(\boldsymbol{x}, t) = 0\}$. It should be noted that the interface $\Gamma(t)$ has codimension 1 in Ω. Embedding the front into a higher dimensional function allows the adoption of an Eulerian perspective to treat a problem that is Lagrangian in nature. Moreover, due to the smoothness of ϕ, the interface unit normal can be computed via simple differentiation:

$$\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|} \ . \tag{2.29}$$

Similarly, the curvature can be obtained as:

$$\kappa = -\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|} \right) \ . \tag{2.30}$$

As shown by Peng et al. [51], the interface $\Gamma(t)$ is transported by the interface velocity $\boldsymbol{u}_\Gamma$ via:

$$\frac{\partial \phi}{\partial t} + u_{\Gamma_i} \frac{\partial \phi}{\partial x_i} = 0 \ . \tag{2.31}$$

Equation (2.31) is valid only at the interface because $\boldsymbol{u}_\Gamma$ in not defined anywhere but at Γ. Consider now a velocity field $\boldsymbol{u}$ defined in the whole domain Ω. Also assume that $\boldsymbol{u}$ has been computed by extending $\boldsymbol{u}_\Gamma$ off the interface and that $\boldsymbol{u} = \boldsymbol{u}_\Gamma$ at the front. Under these conditions, Eq. (2.31) can be generalized to be valid in the whole domain:

$$\frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i} = 0 \ . \tag{2.32}$$

Equation (2.32) is a standard transport equation and therefore a large number of well-established methods are available for its solution. In the case of multi-phase flows with non-reactive interfaces, $\boldsymbol{u}$ is the flow velocity field that satisfies $\boldsymbol{u} = \boldsymbol{u}_\Gamma$ at the interface, thus no dedicated algorithm is required to extend the front velocity off Γ. Moreover, since $\boldsymbol{u} = \boldsymbol{u}_\Gamma$ at each point on the interface, no normal fluid flux crosses Γ. For incompressible flows, Eq. (2.32) can be written in conservative form by using Eq. (2.1):

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_i}(u_i \phi) = 0 \ . \tag{2.33}$$

Even though a certain degree of arbitrariness is allowed in the choice of ϕ, it is convenient to adopt the classic definition:

$$\phi(\boldsymbol{x}, t) = \begin{cases} -d(\boldsymbol{x}) & \text{in } \Omega^+ \\ 0 & \text{at the interface} \\ d(\boldsymbol{x}) & \text{in } \Omega^- \end{cases} , \tag{2.34}$$

where $d(\boldsymbol{x})$ is a *distance function* given by:

$$d(\boldsymbol{x}) = \min_{\boldsymbol{y} \in \Gamma} |\boldsymbol{x} - \boldsymbol{y}| \tag{2.35}$$

and $\boldsymbol{y}$ a generic point on the front. A level set function ϕ defined via (2.34) is called *signed distance function* and offers many desirable properties:

- $\phi(\boldsymbol{x}, t)$ is differentiable almost everywhere on Γ;
- $|\nabla\phi| = 1$ for almost[1] every point in the domain;
- $\boldsymbol{n} = \nabla\phi$ and $\kappa = -\nabla^2\phi$;
- the closest interface point $\boldsymbol{x}_\Gamma$ to a generic point $\boldsymbol{x}$ is given by $\boldsymbol{x}_\Gamma = \boldsymbol{x} - |\phi(\boldsymbol{x}, t)|\boldsymbol{n}$.

While the solution of Eq. (2.33) correctly transports the level set $\phi = 0$, it does not keep ϕ a signed distance function, i.e. $|\nabla\phi| \neq 1$, and may introduce irregularities like steep or flat gradients in the solution. Maintaining ϕ as a distance function is essential for providing the interface with a constant width. Also, the computation of the normal and the curvature may prove difficult near a steep/flat gradient. In order to keep ϕ a signed distance function at all times, the following *reinitialization* equation is solved to steady state:

$$\frac{\partial\phi}{\partial\tau} + S(\phi_0)(|\nabla\phi| - 1) = 0 \ , \tag{2.36}$$

where τ is a fictitious time, S the sign function and ϕ_0 is the initial condition. This reinitialization technique was first introduced by Rouy and Tourin [62] and later perfected by Sussman et al. [75]. The main drawback of the LSM is its inherent lack of conservation properties. This is mainly due to the reinitialization Eq. (2.36) that tends to displace the zero level set from its correct location, thus causing the volume enclosed by the interface to

[1] This property is not satisfied by the set of points that are equidistant from at least 2 points on the interface. This set is called *skeleton* and, when it exists, it does not pose any numerical problem.

shrink. Many strategies have been proposed over the years to address the issue. In particular Enright et al. [23] used Lagrangian marker particles to correct the front position after the Eulerian advection step, while Sussman and Puckett [74] coupled the SLSM with the VOF method in order to take advantage of the good conservation properties of the latter. While these techniques alleviate the issue, they also lose the original simplicity of the SLSM.

2.2.3 Accurate Conservative Level Set Method

In order to overcome the lack of conservation of the SLSM, Olsson and Kreiss [46] and Olsson et al. [47] proposed to replace the level set function ϕ with a phase-field function ψ:

$$\psi(\boldsymbol{x}, t) = \frac{1}{2}\left[\tanh\left(\frac{\phi(\boldsymbol{x}, t)}{2\epsilon}\right) + 1\right] \ , \tag{2.37}$$

where ϵ is a smoothing parameter that determines the steepness of the hyperbolic tangent profile. It should be noted that the front, i.e. the level set $\phi = 0$, is now represented by the iso-contour $\psi = 0.5$, that is $\Gamma(t) = \{\boldsymbol{x} : \psi(\boldsymbol{x}, t) = 0.5\}$. The transport of the interface can still be performed by solving Eq. (2.33) for ψ:

$$\frac{\partial \psi}{\partial t} + \frac{\partial}{\partial x_i}(u_i \psi) = 0 \ . \tag{2.38}$$

The solution of Eq. (2.38) together with the definition (2.37) conserves the volume enclosed by the interface, as proved by Olsson and Kreiss [46] and Desjardins et al. [18]. As in the case of the SLSM, nothing ensures that solving Eq. (2.38) will preserve the hyperbolic tangent profile of ψ. Consequently, the following re-initialization equation needs to be solved to steady state to re-establish the shape of the profile:

$$\frac{\partial \psi}{\partial \tau} + \frac{\partial}{\partial x_i}[\psi(1-\psi)n_i] = \frac{\partial}{\partial x_i}\left[\epsilon\left(\frac{\partial \psi}{\partial x_i} n_i\right) n_i\right] \ , \tag{2.39}$$

where τ is a fictitious time. Equation (2.39) consists of a compression term on the left hand side that aims at sharpening the profile, and a diffusion term on the right hand size that guarantees that the interface remains of characteristic thickness ϵ. It should be emphasized that this form of reinitialization is written in a conservative form and thus, unlike Eq. (2.36), it conserves ψ, at least theoretically. Equation (2.39) was devised by Olsson et al. [47] in the context of the Finite Element Method and extended to the Finite Difference case by Desjardins et al. [18]. One drawback of using a hyperbolic tangent profile in place of a signed distance function is that calculating the normals and curvature by direct differentiation of ψ introduces spurious oscillations because of the steep-slope of the phase field function. Desjardins et al. [18] suggested to re-construct a level set function $\phi = d$ from ψ and compute normals and curvature from ϕ via Eq. (2.29) and (2.30). The method presented so far is called *Accurate Conservative Level Set Method* (ACLSM) and it is used in the present study to model deformable liquid droplets.

2.3 Models Used in the Present Study

In the present study, the flow field for the solid particle-laden turbulence case is modeled via the whole-domain Navier-Stokes Eq. (2.19), where the surface tension term s_i is set to zero. The no-slip condition at the solid boundaries is imposed via the IBM, i.e. via the forcing term f_i given by the Eq. (2.26). Consequently, the carrier flow occupies the entire domain, including the interior of the particles, and $\eta_\rho = \eta_\mu = 1$, i.e. the discontinuities in the material properties are eliminated. In contrast, the flow field for the droplet-laden turbulence case is modeled via the jump condition form of the Navier-Stokes Eq. (2.21). The jump conditions (2.22), (2.23), (2.24) and (2.25) account explicitly for the multi-phase nature of the flow and thus the material properties are now discontinuous. Finally, the gas/liquid interface is modeled via the ACLSM.

Chapter 3

Numerical Methods

This chapter describes the numerical methods of discretizing the equations used in the mathematical models presented in Chapter 2. The details of the computational domain mesh are given in Section 3.1, while the projection method for the solution of the *Navier-Stokes Equations* (NSE) is discussed in Section 3.2. Section 3.3 and 3.4 describe the discretization of the *Immersed Boundary Method* (IBM) and *Accurate Conservative Level Set Method* (ACLSM) equations respectively. Finally, the solution algorithms are summarized in Section 3.5. In the rest of the chapter, discrete operators applied to a variable α will be indicated by

$$\frac{\delta\alpha}{\delta x_i} \approx \frac{\partial\alpha}{\partial x_i} \qquad \text{and} \qquad \frac{\delta\alpha}{\delta t} \approx \frac{\partial\alpha}{\partial t} \ ,$$

where "$\approx$" stands for "is a discretization of". The symbols $\mathsf{i}, \mathsf{j}, \mathsf{k}$ (Serif font) indicate grid indices and are not to be confused with i, j, k (Roman font) used for tensor indices instead. The discrete value of a variable α at a grid node $(\mathsf{i}, \mathsf{j}, \mathsf{k})$ at time step n will be indicated by either $\alpha|^{\mathsf{n}}_{\mathsf{i,j,k}}$ or $\alpha^{\mathsf{n}}_{\mathsf{i,j,k}}$. The former is preferred for vector and tensor quantities to avoid confusion with tensor indices subscripts. The second order *interpolation* of a quantity α

along the x direction at $(\mathsf{i},\mathsf{j},\mathsf{k})$ and time level n is defined by

$$\overline{\alpha}^{x}\big|^{\mathsf{n}}_{\mathsf{i,j,k}} = \frac{\alpha|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j,k}} + \alpha|^{\mathsf{n}}_{\mathsf{i}-\frac{1}{2},\mathsf{j,k}}}{2} \ .$$

$\overline{\alpha}^{y}$ and $\overline{\alpha}^{z}$ are defined in the same manner. The second order *interpolation* of a quantity α in the xy plane at $(\mathsf{i},\mathsf{j},\mathsf{k})$ and time level n is defined by

$$\overline{\alpha}^{xy}\big|^{\mathsf{n}}_{\mathsf{i,j,k}} = \frac{\overline{\alpha}^{y}|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j,k}} + \overline{\alpha}^{y}|^{\mathsf{n}}_{\mathsf{i}-\frac{1}{2},\mathsf{j,k}}}{2} \ .$$

$\overline{\alpha}^{yz}$ and $\overline{\alpha}^{xz}$ are defined in the same manner. The second order *differentiation* operator acting on a variable α along the x direction at $(\mathsf{i},\mathsf{j},\mathsf{k})$ and time level n is expressed by:

$$\left.\frac{\delta_2\alpha}{\delta_2 x}\right|^{\mathsf{n}}_{\mathsf{i,j,k}} = \frac{\alpha|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j,k}} - \alpha|^{\mathsf{n}}_{\mathsf{i}-\frac{1}{2},\mathsf{j,k}}}{\delta x} \ .$$

$\dfrac{\delta_2\alpha}{\delta_2 y}$ and $\dfrac{\delta_2\alpha}{\delta_2 z}$ are defined in the same way. Finally, the components u_i of the velocity vector $\boldsymbol{u}$ will also be indicated by u, v and w so that $\boldsymbol{u} = (u_1, u_2, u_3) = (u, v, w)$.

3.1 Grid Arrangement

All the numerical simulations in the present study were performed in a rectangular domain of sizes $L_x \times L_y \times L_z$, discretized via a Cartesian grid with $N_x \times N_y \times N_z$ nodes. The position of each node is given by the triplet of coordinates $(x_{\mathsf{i}}, y_{\mathsf{j}}, z_{\mathsf{k}})$ defined as:

$$\begin{aligned}
x_{\mathsf{i}} &= \left(\mathsf{i} + \frac{1}{2}\right) \cdot \delta x \quad \text{with} \quad \delta x = \frac{L_x}{N_x} \qquad \forall\ \mathsf{i} = 0, 1, ..., N_x - 1 \quad , \\
y_{\mathsf{j}} &= \left(\mathsf{j} + \frac{1}{2}\right) \cdot \delta y \quad \text{with} \quad \delta y = \frac{L_y}{N_y} \qquad \forall\ \mathsf{j} = 0, 1, ..., N_y - 1 \quad , \qquad (3.1) \\
z_{\mathsf{k}} &= \left(\mathsf{k} + \frac{1}{2}\right) \cdot \delta z \quad \text{with} \quad \delta z = \frac{L_z}{N_z} \qquad \forall\ \mathsf{k} = 0, 1, ..., N_z - 1 \quad ,
\end{aligned}$$

where δx, δy and δz are the grid spacings in the x, y and z direction respectively. N_x, N_y and N_z are chosen in a such a way that the resulting grid is *uniform*, i.e. $\delta x = \delta y = \delta z$. The definition (3.1) provides a unique identification of each node by the triplet of indices $(\mathsf{i}, \mathsf{j}, \mathsf{k})$. Each node is the centroid of the *cell* $I^s_{\mathsf{i},\mathsf{j},\mathsf{k}}$ defined as follows:

$$I^s_{\mathsf{i},\mathsf{j},\mathsf{k}} = [\, x_{\mathsf{i}-\frac{1}{2}},\ x_{\mathsf{i}+\frac{1}{2}}\,] \times [\, y_{\mathsf{j}-\frac{1}{2}},\ y_{\mathsf{j}+\frac{1}{2}}\,] \times [\, z_{\mathsf{k}-\frac{1}{2}},\ z_{\mathsf{k}+\frac{1}{2}}\,]\ . \tag{3.2}$$

A discretization of the computational domain based on the definitions (3.1) and (3.2) is called *cell-centered*. The type of discretization determines which kind of restriction/prolongation operators should be used in the multigrid algorithm for the solution of the *Pressure Poisson's Equation* (PPE) as discussed in Appendix B. All the scalar variables of interest, namely p, ρ, μ, ψ and ϕ, are defined at the grid nodes. In contrast, the velocity vector $\boldsymbol{u} = (u, v, w)$ is defined component by component at *staggered* locations as described in [29]. For example, the discrete values of u are stored at the locations given by the triplet $(x_{\mathsf{i}+\frac{1}{2}}, y_{\mathsf{j}}, z_{\mathsf{k}})$ that is obtained by shifting the node $(x_{\mathsf{i}}, y_{\mathsf{j}}, z_{\mathsf{k}})$ in the x-direction by $\delta x/2$. Similarly, v and w are defined at the locations $(x_{\mathsf{i}}, y_{\mathsf{j}+\frac{1}{2}}, z_{\mathsf{k}})$ and $(x_{\mathsf{i}}, y_{\mathsf{j}}, z_{\mathsf{k}+\frac{1}{2}})$ respectively. A 2D sketch of the staggered grid arrangement described above is shown in Figure 3.1.

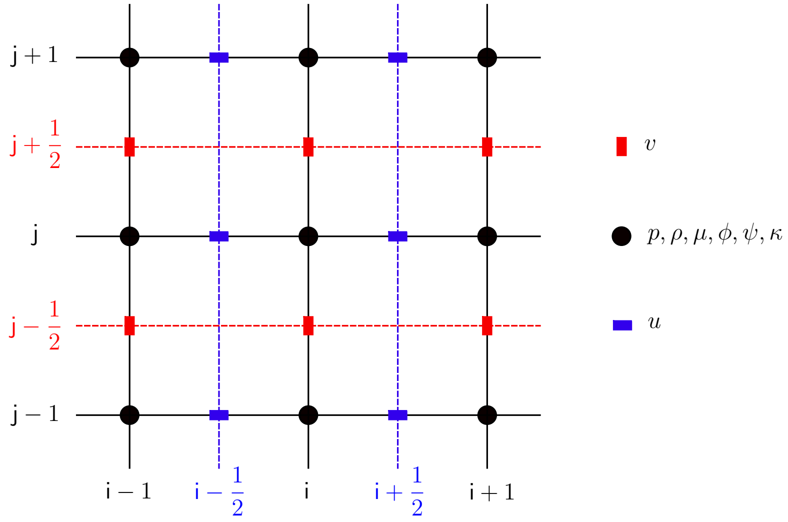


Figure 3.1 Sketch of a staggered grid in the xy plane. The scalar variables are defined at the grid nodes (black circles), while u (blue rectangles) and v (red rectangles) at the faces of a scalar cell.

3.2 Projection Method

The numerical solution of the Navier-Stokes Equations (2.19) and (2.21) is performed via the *projection method* by Chorin [16, 17]. Both equations can be re-written in the following compact form:

$$\frac{\partial u_i}{\partial t} = C_i + P_i + V_i + B_i \ , \tag{3.3}$$

where the convective term C_i, the pressure term P_i, the viscous term V_i and the body forcing term B_i are:

$$C_i = -\frac{\partial}{\partial x_j}(u_i u_j) \ , \tag{3.4}$$

$$P_i = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} \ , \tag{3.5}$$

$$V_i = \frac{1}{\rho\, \mathfrak{Re}}\frac{\partial \tau_{ij}}{\partial x_i} = \frac{1}{\rho\, \mathfrak{Re}} \cdot \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] , \tag{3.6}$$

$$B_i = \frac{g_i}{\mathfrak{Fr}} + \frac{f_i}{\rho} + \frac{s_i}{\rho} \ . \tag{3.7}$$

It should be noted that not all the terms in B_i need to be retained. When solving for immersed solid boundaries via the IBM and Eq. (2.19), the surface tension term, s_i, is set to zero. In contrast, when solving for a two-fluid system via Eq. (2.21) and the ACLSM, both f_i and s_i are set to zero, and the effect of the surface tension is included via the jump condition (2.25). Equation (3.3) can be further simplified by defining the *fluid acceleration* $R_{u_i} = C_i + V_i$:

$$\frac{\partial u_i}{\partial t} = R_{u_i} + P_i + B_i \ . \tag{3.8}$$

The numerical solution of Eq. (3.8) via the projection method consists of the following steps.

1. A tentative velocity u_i^* is computed by advancing in time Eq. (3.8) without P_i and B_i. This is accomplished by using a *second order explicit Adam-Bashforth* method:

$$\begin{aligned}
u_1^*\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} &= u_1\big|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} + w_1 \cdot R_{u_1}\big|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} + w_2 \cdot R_{u_1}\big|^{\mathsf{n}-1}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} \ , \\
u_2^*\big|_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} &= u_2\big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} + w_1 \cdot R_{u_2}\big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} + w_2 \cdot R_{u_2}\big|^{\mathsf{n}-1}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} \ , \\
u_3^*\big|_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} &= u_3\big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} + w_1 \cdot R_{u_3}\big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} + w_2 \cdot R_{u_3}\big|^{\mathsf{n}-1}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} \ ,
\end{aligned} \tag{3.9}$$

   where n indicates the time level. The weights w_1 and w_2 in Eq. (3.9) are computed as:

$$\begin{aligned}
w_1 &= \frac{\delta t^{\mathsf{n}} + 2\delta t^{\mathsf{n}-1}}{2\delta t^{\mathsf{n}-1}} \delta t^{\mathsf{n}} \ , \\
w_2 &= -\frac{\delta t^{\mathsf{n}}}{2\delta t^{\mathsf{n}-1}} \delta t^{\mathsf{n}} \ ,
\end{aligned} \tag{3.10}$$

   where $\delta t^{\mathsf{n}} = t^{\mathsf{n}+1} - t^{\mathsf{n}}$ and $\delta t^{\mathsf{n}-1} = t^{\mathsf{n}} - t^{\mathsf{n}-1}$ are time step widths. This formulation of the second order Adam-Bashforth scheme allows for variable time stepping. The discretization of C_i and V_i in R_{u_i} is discussed in Subsection 3.2.2 and 3.2.3.

2. A second tentative velocity u_i^{**} is obtained by considering the effect of B_i via a *first order explicit Euler scheme*:

$$\begin{aligned}
u_1^{**}\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} &= u_1^*\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} + \delta t^{\mathsf{n}} \cdot B_1\big|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} \ , \\
u_2^{**}\big|_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} &= u_2^*\big|_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} + \delta t^{\mathsf{n}} \cdot B_2\big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} \ , \\
u_3^{**}\big|_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} &= u_3^*\big|_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} + \delta t^{\mathsf{n}} \cdot B_3\big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} \ .
\end{aligned} \tag{3.11}$$

3. A PPE is solved to compute the pressure p at time level $\mathsf{n}+1$:

$$-\frac{\delta P_i}{\delta x_i}\bigg|^{\mathsf{n}+1}_{\mathsf{i},\mathsf{j},\mathsf{k}} = \frac{\delta}{\delta x_i}\left(\frac{1}{\rho}\frac{\delta p}{\delta x_i}\right)\bigg|^{\mathsf{n}+1}_{\mathsf{i},\mathsf{j},\mathsf{k}} = \frac{1}{\delta t^{\mathsf{n}}} \left.\frac{\delta_2 u_i^{**}}{\delta_2 x_i}\right|_{\mathsf{i},\mathsf{j},\mathsf{k}} \ . \tag{3.12}$$

   The discretization of the PPE is described in Subsection 3.2.4.

4. Finally, u_i^{**} is corrected by $P_i^{\mathsf{n+1}}$ to obtain the divergence-free velocity at the next time level $u_i^{\mathsf{n+1}}$:

$$\begin{aligned}
u_1\Big|_{\mathsf{i+\frac{1}{2},j,k}}^{\mathsf{n+1}} &= u_1^{**}\Big|_{\mathsf{i+\frac{1}{2},j,k}} + \delta t^{\mathsf{n}} P_1\Big|_{\mathsf{i+\frac{1}{2},j,k}}^{\mathsf{n+1}} &&= u_1^{**}\Big|_{\mathsf{i+\frac{1}{2},j,k}} - \delta t^{\mathsf{n}} \left(\frac{1}{\rho}\frac{\delta p}{\delta x}\right)\Bigg|_{\mathsf{i+\frac{1}{2},j,k}}^{\mathsf{n+1}} \quad , \\
u_2\Big|_{\mathsf{i,j+\frac{1}{2},k}}^{\mathsf{n+1}} &= u_2^{**}\Big|_{\mathsf{i,j+\frac{1}{2},k}} + \delta t^{\mathsf{n}} P_2\Big|_{\mathsf{i,j+\frac{1}{2},k}}^{\mathsf{n+1}} &&= u_2^{**}\Big|_{\mathsf{i,j+\frac{1}{2},k}} - \delta t^{\mathsf{n}} \left(\frac{1}{\rho}\frac{\delta p}{\delta y}\right)\Bigg|_{\mathsf{i,j+\frac{1}{2},k}}^{\mathsf{n+1}} \quad , \\
u_3\Big|_{\mathsf{i,j,k+\frac{1}{2}}}^{\mathsf{n+1}} &= u_3^{**}\Big|_{\mathsf{i,j,k+\frac{1}{2}}} + \delta t^{\mathsf{n}} P_3\Big|_{\mathsf{i,j,k+\frac{1}{2}}}^{\mathsf{n+1}} &&= u_3^{**}\Big|_{\mathsf{i,j,k+\frac{1}{2}}} - \delta t^{\mathsf{n}} \left(\frac{1}{\rho}\frac{\delta p}{\delta z}\right)\Bigg|_{\mathsf{i,j,k+\frac{1}{2}}}^{\mathsf{n+1}} \quad .
\end{aligned} \tag{3.13}$$

3.2.1 Discretization of the Material Properties

The dimensionless density, ρ, and molecular viscosity, μ, appear in the pressure gradient term, P_i, in the viscous term, V_i, and in the forcing term, B_i. The discretization of the material properties depends on the method used to describe the interface. In the IBM framework, only one fluid exists in the entire domain (cf. Section 2.2.1), i.e. the carrier fluid. Consequently, the discrete density field, ρ_ϵ, and the discrete molecular viscosity field, μ_ϵ, are constant. Moreover, the carrier fluid is taken as reference (cf. Section 2.1.3), and thus

$$\rho_\epsilon = \mu_\epsilon = 1 \ . \tag{3.14}$$

The above definition implies that the PPE (3.12) is *separable*, i.e. it can be re-cast in the following form:

$$\frac{\delta}{\delta x_i}\left(\frac{\delta p}{\delta x_i}\right)\Bigg|_{\mathsf{i,j,k}}^{\mathsf{n+1}} = \frac{1}{\delta t^n}\frac{\delta_2 u_i^{**}}{\delta_2 x_i}\Bigg|_{\mathsf{i,j,k}} \ . \tag{3.15}$$

The discretization of the material properties for a fluid/fluid interface is more challenging because both ρ and μ are discontinuous. In the context of the ACLSM framework, the discrete material properties ρ_ϵ and μ_ϵ are defined at the grid nodes via

$$\rho_\epsilon = 1 + (\eta_\rho - 1)\psi \qquad \text{and} \qquad \mu_\epsilon = 1 + (\eta_\mu - 1)\psi \ , \tag{3.16}$$

where the density ratio, η_ρ, and the molecular viscosity ratio, η_μ, are defined in Eq. (2.14) and (2.15). By using the above definitions, the discontinuities in ρ and μ are smeared out over a band of size proportional to ϵ, the smoothing parameter that defines the steepness of the phase field function ψ (cf. Section 3.4). The smooth treatment of the material properties corresponds to the *Continuum Surface Force* (CSF) approach developed by Brackbill et al. [10]. A direct consequence of smearing the jump in the molecular viscosity is that the pressure jump condition (2.25) reduces to:

$$[p]_\Gamma = p^+ - p^- \ = \frac{\kappa}{\mathfrak{We}} \ . \tag{3.17}$$

In this study, the definitions (3.16) are used to discretize the density and viscosity in V_i and B_i. In contrast, the density jump in P_i, as well as the pressure jump condition (3.17), are captured via the *Ghost Fluid Method* (GFM) without introducing any numerical smearing, as shown in Subsection 3.2.4.

3.2.2 Discretization of the Convective Term

The convective terms C_i are discretized via second order centered finite difference as follows:

$$\begin{aligned}
C_1\Big|^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}} &= -\left[\frac{\delta_2(\overline{u}^x\overline{u}^x)}{\delta_2 x} + \frac{\delta_2(\overline{u}^y\overline{v}^x)}{\delta_2 y} + \frac{\delta_2(\overline{u}^z\overline{w}^x)}{\delta_2 z}\right]^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}} \quad , \\
C_2\Big|^{\mathsf{n}}_{\mathsf{i,j+\frac{1}{2},k}} &= -\left[\frac{\delta_2(\overline{v}^x\overline{u}^y)}{\delta_2 x} + \frac{\delta_2(\overline{v}^y\overline{v}^y)}{\delta_2 y} + \frac{\delta_2(\overline{v}^z\overline{w}^y)}{\delta_2 z}\right]^{\mathsf{n}}_{\mathsf{i,j+\frac{1}{2},k}} \quad , \\
C_3\Big|^{\mathsf{n}}_{\mathsf{i,j,k+\frac{1}{2}}} &= -\left[\frac{\delta_2(\overline{w}^x\overline{u}^z)}{\delta_2 x} + \frac{\delta_2(\overline{w}^y\overline{v}^z)}{\delta_2 y} + \frac{\delta_2(\overline{w}^z\overline{w}^z)}{\delta_2 z}\right]^{\mathsf{n}}_{\mathsf{i,j,k+\frac{1}{2}}} \quad .
\end{aligned} \tag{3.18}$$

The discretization given by Eq. (3.18) is conservative. It should be noted that C_1, C_2 and C_3 are defined at the same staggered locations of u, v and w respectively.

3.2.3 Discretization of the Viscous Term

The viscous terms V_i are discretized via second order centered finite difference as follows:

$$\begin{aligned}
V_1\Big|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} &= \frac{1}{\mathfrak{Re}}\left[\frac{\delta_2(\tau_{xx})}{\delta_2 x} + \frac{\delta_2(\tau_{xy})}{\delta_2 y} + \frac{\delta_2(\tau_{xz})}{\delta_2 z}\right]^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} \cdot \frac{1}{\rho_\varepsilon}\Bigg|^{\mathsf{n}+1}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} \quad , \\
V_2\Big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} &= \frac{1}{\mathfrak{Re}}\left[\frac{\delta_2(\tau_{yx})}{\delta_2 x} + \frac{\delta_2(\tau_{yy})}{\delta_2 y} + \frac{\delta_2(\tau_{yz})}{\delta_2 z}\right]^{\mathsf{n}}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} \cdot \frac{1}{\rho_\varepsilon}\Bigg|^{\mathsf{n}+1}_{\mathsf{i},\mathsf{j}+\frac{1}{2},\mathsf{k}} \quad , \qquad (3.19) \\
V_3\Big|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} &= \frac{1}{\mathfrak{Re}}\left[\frac{\delta_2(\tau_{zx})}{\delta_2 x} + \frac{\delta_2(\tau_{zy})}{\delta_2 y} + \frac{\delta_2(\tau_{zz})}{\delta_2 z}\right]^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} \cdot \frac{1}{\rho_\varepsilon}\Bigg|^{\mathsf{n}+1}_{\mathsf{i},\mathsf{j},\mathsf{k}+\frac{1}{2}} \quad ,
\end{aligned}$$

where ρ_ϵ is the discrete density field defined in Eq. (3.14) and (3.16). It should be noted that V_1, V_2 and V_3 are defined at the same staggered locations of u, v and w respectively. The stress tensor components τ_{xx}, τ_{xy} and τ_{xz} are discretized at staggered locations consistently with the position of the discrete derivatives in Eq. (3.19):

$$\begin{aligned}
\tau_{xx}\Big|^{\mathsf{n}}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} &= 2\mu_\epsilon\Big|^{\mathsf{n}+1}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} \cdot \frac{\delta_2 u}{\delta_2 x}\Bigg|^{\mathsf{n}}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} \quad , \\
\tau_{xy}\Big|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j}+\frac{1}{2},\mathsf{k}} &= \overline{\mu_\epsilon}^{xy}\Big|^{\mathsf{n}+1}_{\mathsf{i}+\frac{1}{2},\mathsf{j}+\frac{1}{2},\mathsf{k}} \cdot \left(\frac{\delta_2 u}{\delta_2 y} + \frac{\delta_2 v}{\delta_2 x}\right)\Bigg|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j}+\frac{1}{2},\mathsf{k}} , \qquad (3.20) \\
\tau_{xz}\Big|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}+\frac{1}{2}} &= \overline{\mu_\epsilon}^{xz}\Big|^{\mathsf{n}+1}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}+\frac{1}{2}} \cdot \left(\frac{\delta_2 u}{\delta_2 z} + \frac{\delta_2 w}{\delta_2 x}\right)\Bigg|^{\mathsf{n}}_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}+\frac{1}{2}} \quad ,
\end{aligned}$$

where μ_ϵ is the discrete dynamic viscosity field defined in Eq. (3.14) and (3.16). The remaining stress tensor components are computed in the same fashion as τ_{xx}, τ_{xy} and τ_{xz}. It should be emphasized that ρ_ϵ and μ_ϵ in Eq. (3.19) and (3.20) are computed at time step $\mathsf{n}+1$.

3.2.4 Discretization of the Pressure Poisson's Equation

The PPE (3.12) is discretized on the grid nodes where the pressure p is defined, by using second order centered finite differencing. As discussed in Subsection 3.2.1, $\rho_\epsilon = 1$ in the whole domain for the case of solid particles simulated via the IBM. Therefore, Eq. (3.12) can be re-cast to Eq. (3.15) and discretized with standard finite differencing. Its solution can then be performed efficiently via the *Fast Fourier Transform* (FFT) or via an iterative method for sparse linear system. The case of liquid droplets presents additional difficulties because of the jump-condition form of the governing equations used in this case. Particularly, in the pressure gradient term P_i:

1. the density is variable, and
2. both density and pressure are singular at the interface.

Item 1. implies that the PPE (3.12) is *non-separable*, that is, it cannot be re-cast into the standard Poisson's Eq. (3.15). As a result, the FFT cannot be used. Item 2. implies that standard finite differencing is not a good choice in the neighborhood of the interface where the jumps in density and pressure have to be accounted for. A suitable discretization for the *Left-Hand Side* (LHS) of Eq. (3.12) for this case can be obtained by using the GFM [36, 31]. The GFM assumes that the jumps at the interface are known and can be extended off the interface by continuity through a Taylor series. More details on the GFM and its derivation are given in Appendix A. In what follows the focus will solely be on how the method is used in a 1D domain: the extension to 3D can be obtained by applying the 1D method in the three directions. In a 1D domain, the PPE given by Eq. (3.12) reads:

$$\frac{\delta}{\delta x}\left(\frac{1}{\rho}\frac{\delta p}{\delta x}\right)\Bigg|_{\mathsf{i}}^{\mathsf{n}+1} = \frac{1}{\delta t^{\mathsf{n}}}\frac{\delta_2 u^{**}}{\delta_2 x}\Bigg|_{\mathsf{i}} \tag{3.21}$$

The LHS of Eq. (3.21) is discretized at node i via second order centered finite difference together with the GFM and reads:

$$\frac{\delta}{\delta x}\left(\frac{1}{\rho}\frac{\delta p}{\delta x}\right)\Bigg|_{\mathsf{i}}^{\mathsf{n+1}} = \frac{p_{\mathsf{i+1}} - p_{\mathsf{i}} - [p]_R}{\rho_R\,\delta x^2} - \frac{p_{\mathsf{i}} - p_{\mathsf{i-1}} - [p]_L}{\rho_L\,\delta x^2}\ , \tag{3.22}$$

where $[p]_R$ and $[p]_L$ account for the pressure jump condition (3.17). In particular, the pressure jump $[p]_R$ at the interface located between i and $\mathsf{i}+1$ is defined as:

$$[p]_R = \begin{cases} k_R/\mathfrak{We} & \text{if } \phi_{\mathsf{i+1}}^{\mathsf{n+1}} > 0,\ \phi_{\mathsf{i}}^{\mathsf{n+1}} \leq 0 \\ -k_R/\mathfrak{We} & \text{if } \phi_{\mathsf{i+1}}^{\mathsf{n+1}} \leq 0,\ \phi_{\mathsf{i}}^{\mathsf{n+1}} > 0 \\ 0 & \text{otherwise (no interface)} \end{cases} . \tag{3.23}$$

Linear interpolation is used to evaluate k_R, the curvature at the interface located between i and $\mathsf{i}+1$:

$$k_R = \frac{k_{\mathsf{i+1}}|\phi_{\mathsf{i}}^{\mathsf{n+1}}| + k_{\mathsf{i}}|\phi_{\mathsf{i+1}}^{\mathsf{n+1}}|}{|\phi_{\mathsf{i}}^{\mathsf{n+1}}| + |\phi_{\mathsf{i+1}}^{\mathsf{n+1}}|}\ , \tag{3.24}$$

while the *effective* density ρ_R is computed as

$$\rho_R = \frac{\rho_{\mathsf{i}}|\phi_{\mathsf{i}}^{\mathsf{n+1}}| + \rho_{\mathsf{i+1}}|\phi_{\mathsf{i+1}}^{\mathsf{n+1}}|}{|\phi_{\mathsf{i}}^{\mathsf{n+1}}| + |\phi_{\mathsf{i+1}}^{\mathsf{n+1}}|}\ . \tag{3.25}$$

The quantities $[p]_L$, k_L and ρ_L associated with the left side of the stencil can be calculated similarly by applying stencil symmetry. The full discretization of equation Eq. (3.21) is therefore:

$$\frac{p_{\mathsf{i+1}}^{\mathsf{n+1}} - p_{\mathsf{i}}^{\mathsf{n+1}}}{\rho_R\ \delta x^2} - \frac{p_{\mathsf{i}}^{\mathsf{n+1}} - p_{\mathsf{i-1}}^{\mathsf{n+1}}}{\rho_L\ \delta x^2} = \frac{u_{\mathsf{i+1/2}}^{**} - u_{\mathsf{i-1/2}}^{**}}{\delta x\ \delta t^{\mathsf{n}}} + \frac{[p]_R}{\rho_R\ \delta x^2} + \frac{[p]_L}{\rho_L\ \delta x^2}\ . \tag{3.26}$$

It should be noted that the pressure gradient in the correction step (3.13) must be treated in the same way that is used for the discretization of Eq. (3.12). In a 1D case, the correction step is

$$u^{\mathsf{n+1}}_{\mathsf{i+1/2}} = u^{**}_{\mathsf{i+1/2}} - \delta t^{\mathsf{n}} \left(\frac{p^{\mathsf{n+1}}_{\mathsf{i+1}} - p^{\mathsf{n+1}}_{\mathsf{i}} - [p]_R}{\rho_R \, \delta x} \right) . \tag{3.27}$$

3.2.5 Time Step Restriction

In order to ensure the stability of the simulations, the time step, δt, was computed to satisfy the *Courant-Friedrichs-Lewy* (CFL) condition. The projection method coupled with the IBM proved to be stable for $\delta t \leq \delta x/2$. In contrast, the projection method coupled with the ACLSM required adaptive time stepping. For this case, the δt was computed to satisfy:

$$\delta t \, \mathcal{C} \leq 1 \ , \tag{3.28}$$

where $\mathcal{C}$ is a coefficient that accounts for the stability requirements of the convective, viscous, gravity term, and surface tension term. Kang et al. [31] derived the following definition of $\mathcal{C}$:

$$\mathcal{C} = \frac{(\mathcal{C}_c + \mathcal{C}_v) + \sqrt{(\mathcal{C}_c + \mathcal{C}_v)^2 + 4\mathcal{C}_g^2 + 4\mathcal{C}_s^2}}{2} \ , \tag{3.29}$$

where the coefficients $\mathcal{C}_c$, $\mathcal{C}_v$, $\mathcal{C}_g$, and $\mathcal{C}_s$ are:

$$\begin{aligned}
\mathcal{C}_c &= \frac{\max(u)}{\delta x} + \frac{\max(v)}{\delta y} + \frac{\max(w)}{\delta z} \\
\mathcal{C}_d &= \frac{2}{\mathfrak{Re}} \max\left(1, \frac{\eta_\mu}{\eta_\rho}\right) \left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2} \right) \\
\mathcal{C}_g &= \sqrt{\frac{1}{\mathfrak{Fr}}} \\
\mathcal{C}_s &= \sqrt{\frac{\kappa_{max}}{\mathfrak{We} \ \min(1, \eta_\rho) \ \min(\delta x^2, \delta y^2, \delta z^2)}}
\end{aligned} \tag{3.30}$$

In the above definitions, η_ρ and η_μ are the density and molecular viscosity ratio defined in Subsection 2.1.3, while κ_{max} is the maximum value of the local curvature of the interface.

3.3 Discretization of the IBM Equations

As mentioned in Subsection 2.2.1, the IBM relies on a set of curvilinear coordinates attached to the interface being tracked. In discrete terms, this means one has to provide a Lagrangian discretization of the interface in addition to the Eulerian grid used for the discretization of the computational domain. In this work, the IBM is used to model solid spherical particles. The surface of each particle is discretized via a uniform distribution of N_l Lagrangian points. The location $\boldsymbol{X}_l = (X_l, Y_l, Z_l)$ of the l-th Lagrangian point is given as a function of the position $\boldsymbol{x}_p = (x_p, y_p, z_p)$ of the center of the particle by:

$$\begin{aligned} X_l &= x_p + R_p \sin(\theta_l)\cos(\varphi_l) \\ Y_l &= y_p + R_p \sin(\theta_l)\sin(\varphi_l) \\ Z_l &= z_p + R_p \cos(\theta_l) \end{aligned} \tag{3.31}$$

where θ_l is the polar angle, φ_l the azimuthal angle and R_p the particle radius. It should be emphasized that R_p is constant since the particles are solid spheres. θ_l and φ_l are calculated by using the explicit spiral set proposed by Saff and Kuijlaars [64]:

$$\begin{aligned} &\theta_l = \arccos(c_l)\ , \quad c_l = -1 + \frac{2(l-1)}{N_l - 1}\ , \quad 1 \le l \le N_l\ , \\ &\varphi_0 = \varphi_{N_l} = 0\ , \\ &\varphi_l = \varphi_{l-1} + \frac{3.6}{\sqrt{N_l\,(1-c_l^2)}}\ , \quad 1 < l < N_l\ . \end{aligned} \tag{3.32}$$

3.3.1 Computation of the Forcing Term

The effect of the solid particles on the flow is accounted for via the forcing term f_i at step 2. of the projection algorithm (cf. Section 3.2). In order to compute f_i, the intermediate velocity u_i^*, defined on the Eulerian grid, is mapped to the Lagrangian points via

$$U_i^*(\boldsymbol{X}_l^m) = \sum_{\boldsymbol{x}\in\Omega} u_i^*(\boldsymbol{x})\,\delta_h(\boldsymbol{x}-\boldsymbol{X}_l^m)\,\delta x^3 \qquad \forall\; l, m\ , \tag{3.33}$$

where $m \in [1, N_p]$ is an integer indexing a specific particle and N_p is the total number of tracked particles. The above equation is a discretization of Eq. (2.27). In Eq. (3.33), δ_h represents the three-dimensional regularized three-point delta function proposed by Roma et al. [61]:

$$\delta_h(\boldsymbol{x}-\boldsymbol{X}_l^m) = \frac{1}{\delta x^3}\delta(r_1)\delta(r_2)\delta(r_3) \quad \text{with} \quad r_i = \frac{x_i-(X_l^m)_i}{\delta x}\ , \tag{3.34}$$

where

$$\delta(r) = \begin{cases} \dfrac{1}{3}(1+\sqrt{1-3r^2}), & \text{if } |r| \le 0.5 \\ \dfrac{1}{6}(5-3|r|-\sqrt{1-3(1-|r|)^2}), & \text{if } 0.5 < |r| \le 1.5 \\ 0, & \text{if } |r| > 1.5 \end{cases} \ . \tag{3.35}$$

A graphical representation of δ_h is given in Figure 3.2. Once the Lagrangian intermediate velocity U_i^* is known, the Lagrangian forcing F_i required to impose the no-slip condition at the interface can be calculated as:

$$F_i(\boldsymbol{X}_l^m) = \frac{U_i(\boldsymbol{X}_l^m)|^{\mathsf{n}+1} - U_i^*(\boldsymbol{X}_l^m)}{\delta t^{\mathsf{n}}} \qquad \forall\; l, m\ , \tag{3.36}$$

where $U_i(\boldsymbol{X}_l^m)$ is the velocity at the Lagrangian point $\boldsymbol{X}_l^m$ obtained by applying the rigid-body motion (2.28):

$$\boldsymbol{U}(\boldsymbol{X}_l^m)|^{\mathsf{n+1}} = \boldsymbol{u}_p^m|^{\mathsf{n+1}} + \boldsymbol{\omega}_p^m|^{\mathsf{n+1}} \times (\boldsymbol{X}_l^m - \boldsymbol{x}_p^m|^{\mathsf{n+1}}) \ , \tag{3.37}$$

where $\boldsymbol{u}_p$ and $\boldsymbol{\omega}_p$ are the linear and angular velocity of the particle center. Finally the forcing term f_i is obtained by mapping F_i to the Eulerian grid via:

$$f_i(\boldsymbol{x}) = \sum_{m=1}^{N_p} \sum_{l=1}^{N_l} F_i(\boldsymbol{X}_l^m) \delta_h(\boldsymbol{x} - \boldsymbol{X}_l^m) \Delta V_l^m \tag{3.38}$$

where ΔV_l^m is the volume associated with the l-th force point:

$$\Delta V_l^m = \frac{\pi \delta x}{3 N_l} (12 R_p^2 + \delta x^2) \ . \tag{3.39}$$

Equation (3.38) is the discrete version of Eq. (2.26).

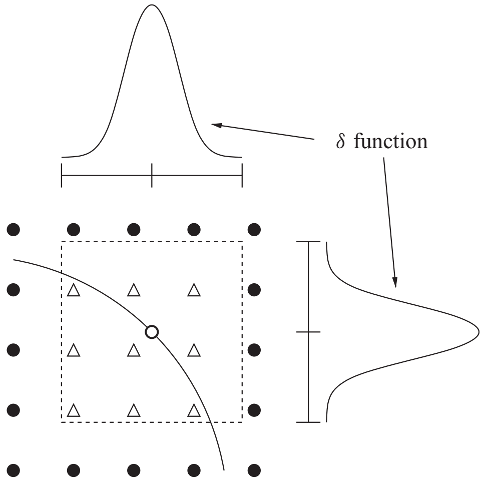


Figure 3.2 Eulerian grid points (triangles) influenced by the force at the Lagrangian grid point (white circle) of the particle spherical surface (black line) (from [39]). Three-dimensional, three-point δ_h function by Roma et al. [61].

3.3.2 Computation of the Particle Motion

The linear and angular velocities $\boldsymbol{u}_p$ and $\boldsymbol{\omega}_p$ and the center position $\boldsymbol{x}_p$ are computed for each particle by solving the following *Ordinary Differential Equations* (ODEs) that were derived by Lucci et al. [39]:

$$\begin{aligned}
\frac{d\boldsymbol{x}_p^m}{dt} &= \boldsymbol{u}_p^m \ , \\
\frac{d\boldsymbol{u}_p^m}{dt} &= \frac{1}{(1-\rho_p^m)\Omega_p^m}\sum_{l=1}^{N_l}\boldsymbol{F}(\boldsymbol{X}_l^m)\Delta V_l^m + \boldsymbol{g} + \frac{\boldsymbol{F}_R^m}{(\rho_p^m-1)\Omega_p^m} \ , \\
\frac{d\boldsymbol{\omega}_p^m}{dt} &= \frac{1}{I_p^m}\sum_{l=1}^{N_l}(\boldsymbol{X}_l^m - \boldsymbol{x}_p^m)\times\boldsymbol{F}(\boldsymbol{X}_l^m)\Delta V_l^m + \frac{1}{I_p^m}\frac{d}{dt}\iiint_{\Omega_m}[(\boldsymbol{x}-\boldsymbol{x}_p^m)\times\boldsymbol{u}]d\boldsymbol{x} \ ,
\end{aligned} \tag{3.40}$$

where ρ_p^m, Ω_p^m and I_p^m are the dimensionless density, volume and momentum of inertia of the m-th particle respectively, and $\boldsymbol{F}_R^m$ is a repulsive force that avoids the overlapping of particles (or penetration into each other). A detailed discussion on $\boldsymbol{F}_R^m$ is given in [39]. The above equations are integrated in time via a second-order Adam-Bashforth method.

3.4 Discretization of the ACLSM Equations

The phase-field function ψ defined by Eq. (2.37) is discretized at the grid nodes. As pointed out by Olsson et al. [47], choosing a small value for the smoothing parameter ϵ is crucial to achieve conservation of the volume bounded by the $\psi = 0.5$ iso-surface. On the other hand, if ϵ is too small compared to the grid size δx, over- or under-shoots may affect the steady-state solution of Eq. (2.39). ϵ is set to be proportional to the grid size by using the relation given by Olsson and Kreiss [46]:

$$\epsilon = \frac{(\delta x)^{1-\mathsf{d}}}{2} \ , \tag{3.41}$$

where $\mathsf{d} \geq 0$ is a parameter that influences the steepness of ψ. $\mathsf{d} > 0$ improves the stability of the method, while $\mathsf{d} = 0$ delivers better conservation properties. All the tests performed showed that $\mathsf{d} = 0$ was the best choice for the purposes of this study. This correspond to an interface width of about three to four grid spacings.

3.4.1 Interface Transport Equation

The transport equation (2.38) is advanced in time by using a *second order explicit Adam-Bashforth* method:

$$\psi|^{\mathsf{n+1}}_{\mathsf{i,j,k}} = \psi|^{\mathsf{n}}_{\mathsf{i,j,k}} + w_1 \cdot R_\psi|^{\mathsf{n}}_{\mathsf{i,j,k}} + w_2 \cdot R_\psi|^{\mathsf{n-1}}_{\mathsf{i,j,k}} \ , \tag{3.42}$$

where the weights w_1 and w_2 are given by Eq. (3.10). The term R_ψ represents a conservative discretization of the *Right-Hand Side* (RHS) of Eq. (2.38):

$$\begin{aligned} R_\psi|^{\mathsf{n}}_{\mathsf{i,j,k}} = -\frac{\delta}{\delta x_i}(u_i\psi)\Big|^{\mathsf{n}}_{\mathsf{i,j,k}} &= -\frac{F|^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}} - F|^{\mathsf{n}}_{\mathsf{i-\frac{1}{2},j,k}}}{\delta x} - \frac{G|^{\mathsf{n}}_{\mathsf{i,j+\frac{1}{2},k}} - G|^{\mathsf{n}}_{\mathsf{i,j-\frac{1}{2},k}}}{\delta y} + \\ &- \frac{H|^{\mathsf{n}}_{\mathsf{i,j,k+\frac{1}{2}}} - H|^{\mathsf{n}}_{\mathsf{i,j,k-\frac{1}{2}}}}{\delta z} \ , \end{aligned} \tag{3.43}$$

where the numerical fluxes F, G and H are approximations of the convective terms $u\psi$, $v\psi$ and $w\psi$ respectively. In order to evaluate R_ψ at the grid nodes, the numerical fluxes need to be computed at the velocity locations, i.e. at the faces of the scalar cell $I^s_{\mathsf{i,j,k}}$. The procedure to compute F will now be described; G and H are obtained by applying the same procedure in the y and z direction respectively. $F|^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}}$ is calculated via the following upwind scheme:

$$F|^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}} = u^+\big|^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}}\,\psi^L_{\mathsf{i+\frac{1}{2},j,k}} + u^-\big|^{\mathsf{n}}_{\mathsf{i+\frac{1}{2},j,k}}\,\psi^R_{\mathsf{i+\frac{1}{2},j,k}} \ , \tag{3.44}$$

where the superscripts + and − indicates the positive and negative part[1]. $\psi^L_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}}$ and $\psi^R_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}}$ are left and right approximations of the phase field function at the cell face $(\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k})$ where ψ is not directly available. These approximations are computed via the following linear reconstruction:

$$\begin{aligned} \psi^L_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} &= \psi|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}} + \frac{\delta x}{2} s_{\mathsf{i},\mathsf{j},\mathsf{k}} \ , \\ \psi^R_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} &= \psi|^{\mathsf{n}}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} - \frac{\delta x}{2} s_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} \ , \end{aligned} \tag{3.45}$$

where the slopes $s_{\mathsf{i},\mathsf{j},\mathsf{k}}$ and $s_{\mathsf{i}+1,\mathsf{j},\mathsf{k}}$ are calculated by using a slope limiter $\theta(a,b)$ as follows:

$$\begin{aligned} s_{\mathsf{i},\mathsf{j},\mathsf{k}} &= \theta\left(\frac{\psi|^{\mathsf{n}}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} - \psi|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}}}{\delta x}, \frac{\psi|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}} - \psi|^{\mathsf{n}}_{\mathsf{i}-1,\mathsf{j},\mathsf{k}}}{\delta x}\right) \ , \\ s_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} &= \theta\left(\frac{\psi|^{\mathsf{n}}_{\mathsf{i}+2,\mathsf{j},\mathsf{k}} - \psi|^{\mathsf{n}}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}}}{\delta x}, \frac{\psi|^{\mathsf{n}}_{\mathsf{i}+1,\mathsf{j},\mathsf{k}} - \psi|^{\mathsf{n}}_{\mathsf{i},\mathsf{j},\mathsf{k}}}{\delta x}\right) \ . \end{aligned} \tag{3.46}$$

The choice was made to use the *Superbee* slope limiter developed by Roe [60]:

$$\theta(a,b) = \begin{cases} \operatorname{sign}(a)\max(|a|,|b|) & \text{if } \ a\cdot b > 0 \ \wedge \ |a|/2 \le |b| \le 2|a| \ , \\ 2\operatorname{sign}(a)\min(|a|,|b|) & \text{if } \ a\cdot b > 0 \ \wedge \ (\ |a|/2 \ge |b| \vee |b| \ge 2|a| \ ) \ , \\ 0 & \text{if } \ a\cdot b \le 0 \ . \end{cases} \tag{3.47}$$

The discretization method presented here was proposed by Olsson and Kreiss [46].

3.4.2 Reconstruction of the Signed Distance Function

Using a phase-field ψ function instead of a signed distance function ϕ is beneficial to volume conservation, but reduces the accuracy of the calculation of normals and curvature. Des-

[1] The positive part a^+ of a real variable a is defined as $a^+ = \max(a,0)$. Similarly the negative part a^- is $a^- = \min(a,0)$.

jardins et al. [18] pointed out that $\boldsymbol{n}$ and κ are strongly sensitive to spurious oscillations in ψ if they are obtained by direct differentiation via

$$\boldsymbol{n} = \frac{\nabla\psi}{|\nabla\psi|} \tag{3.48}$$

and

$$\kappa = \nabla \cdot \left(\frac{\nabla\psi}{|\nabla\psi|} \right) \ . \tag{3.49}$$

In fact, an oscillation in ψ will appear as a large change in direction of the normal vector and therefore normals computed using Eq. (3.48) are not suitable for the solution of Eq. (2.39). The problem is even more serious for a curvature calculated via Eq. (3.49) since two levels of differentiations are involved in the process. In order to alleviate this problem, Desjardins et al. [18] proposed to compute $\boldsymbol{n}$ and κ from the signed distance function ϕ defined in Eq. (2.34). Thus, ϕ has to be reconstructed from the phase-field function ψ. This is accomplished by using the *Fast Sweeping Method* (FSM) described in Appendix C. The initial condition used to start the fast sweeping algorithm is

$$d_{\mathsf{i,j,k}} = \begin{cases} \Big| 2\epsilon \, \mathrm{atanh}(\ 2\psi_{\mathsf{i,j,k}} - 1) \Big| & \text{if} \quad \Big| 2\epsilon \, \mathrm{atanh}(\ 2\psi_{\mathsf{i,j,k}} - 1) \Big| \leq \delta x \ , \\ +\infty & \text{otherwise} \ \ , \end{cases} \tag{3.50}$$

where the time index has been dropped for clarity. Therefore d is initialized with large positive values everywhere in the domain except in a small band around the interface, where the exact distance is computed by inverting Eq. (2.37). After the distance function d has been reconstructed, the signed distance function ϕ is obtained by applying the definition (2.34). It should be emphasized that the nodes initially set to the exact solution via Eq. (3.50) are never updated during the whole reconstruction procedure.

3.4.3 Calculation of the Normal Vector

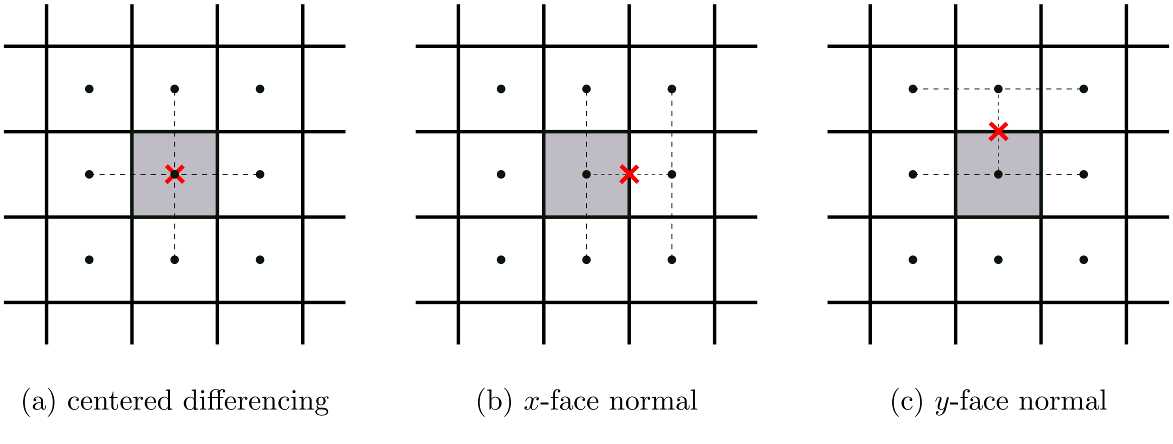
The solution of the reinitialization Eq. (2.39) requires the computation of the normal vector $\boldsymbol{n}$ at the faces of the scalar cell $I^s_{\mathsf{i,j,k}}$, as explained in Subsection 3.4.5. $\boldsymbol{n}$ could easily be computed at the scalar nodes via standard centered finite difference and then interpolated at the cell faces (cf. Figure 3.3a). However, this approach would introduce spurious oscillations as shown in [18]. A better alternative is to define *face normals* directly at the cell face locations. The x-face normal $\boldsymbol{n}^x$ at $(\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k})$ is computed as follows (cf. Figure 3.3b):

$$\boldsymbol{n}^x\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} = \left[\frac{(N^x_x, N^x_y, N^x_z)}{\sqrt{(N^x_x)^2+(N^x_y)^2+(N^x_z)^2}}\right]_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}}, \tag{3.51}$$

where

$$N^x_x\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} = \left.\frac{\delta_2\phi}{\delta_2 x}\right|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}}, \qquad N^x_y\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} = \left.\frac{\delta_2\overline{\phi}^x}{\delta_2 y}\right|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}}, \qquad N^x_z\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}} = \left.\frac{\delta_2\overline{\phi}^x}{\delta_2 z}\right|_{\mathsf{i}+\frac{1}{2},\mathsf{j},\mathsf{k}}. \tag{3.52}$$

The y- and z-face normals $\boldsymbol{n}^y$ and $\boldsymbol{n}^z$ are computed in the same fashion (cf. Figure 3.3). This scheme was devised by Desjardins et al. [18].



(a) centered differencing (b) x-face normal (c) y-face normal

Figure 3.3 Stencils for the discretization of the normal vector. The black circles represent grid nodes, while the red cross symbol shows where the normals are computed. The dashed lines indicate the stencil used.

3.4.4 Calculation of the Curvature

In order to compute the curvature κ at the grid nodes, one could simply discretize Eq. (3.49) by using the face normals:

$$\kappa_{\mathsf{i,j,k}} = \frac{n_x^x\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j,k}} - n_x^x\big|_{\mathsf{i}-\frac{1}{2},\mathsf{j,k}}}{\delta x} + \frac{n_y^y\big|_{\mathsf{i,j}+\frac{1}{2},\mathsf{k}} - n_y^y\big|_{\mathsf{i,j,k}-\frac{1}{2}}}{\delta y} + \frac{n_z^z\big|_{\mathsf{i,j,k}+\frac{1}{2}} - n_z^z\big|_{\mathsf{i,j,k}-\frac{1}{2}}}{\delta z} \ . \tag{3.53}$$

Since ϕ is recomputed at every time step via the FSM, inaccuracies in the distance field are to be expected. Therefore Eq. (3.53) cannot produce an accurate curvature because it relies on two levels of numerical differentiation of ϕ. In this study κ was computed via a *Least Squares Reconstruction* (LSR) as proposed by Marchandise et al. [41]. This approach will now be described for a 2D domain. Consider a generic variable α, known at each node, that is polluted by some inaccuracies. The goal is to reconstruct the values of the derivatives of α so that the effect of these inaccuracies is minimal. In order to do so the data surrounding each node (i,j) is assumed to behave quadratically via a second order Taylor's expansion:

$$\begin{aligned} \alpha_{\mathsf{i+p,j+m}} - \alpha_{\mathsf{i,j}} = {} & \left.\frac{\delta\alpha}{\delta x}\right|_{\mathsf{i,j}} \cdot \mathsf{p}\,\delta x + \left.\frac{\delta\alpha}{\delta y}\right|_{\mathsf{i,j}} \cdot \mathsf{m}\,\delta y + \\ & + \frac{1}{2}\left.\frac{\delta^2\alpha}{\delta x^2}\right|_{\mathsf{i,j}} \cdot (\mathsf{p}\,\delta x)^2 + \frac{1}{2}\left.\frac{\delta^2\alpha}{\delta y^2}\right|_{\mathsf{i,j}} \cdot (\mathsf{m}\,\delta y)^2 + \left.\frac{\delta^2\alpha}{\delta x\delta y}\right|_{\mathsf{i,j}} \cdot (\mathsf{p}\,\delta x)\cdot(\mathsf{m}\,\delta y) \ . \end{aligned} \tag{3.54}$$

The above equation links the value of α at (i,j) with that at a surrounding node $(\mathsf{i}+\mathsf{p},\mathsf{j}+\mathsf{m})$. If Eq. (3.54) is applied to each node defined by the pair of integer indeces $(\mathsf{p},\mathsf{m}) \in [-p,p] \times [-q,q] \setminus (0,0)$, one obtains $(2p+1)(2q+1)-1$ equations of the form (3.54). The associated linear system reads:

$$\begin{bmatrix} -p\delta x & -q\delta y & \frac{1}{2}(p\delta x)^2 & \frac{1}{2}(q\delta y)^2 & pq\delta x\delta y \\ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ p\delta x & q\delta y & \frac{1}{2}(p\delta x)^2 & \frac{1}{2}(q\delta y)^2 & pq\delta x\delta y \end{bmatrix} \cdot \begin{bmatrix} \left.\frac{\delta\alpha}{\delta x}\right|_{\mathsf{i,j}} \\ \left.\frac{\delta\alpha}{\delta y}\right|_{\mathsf{i,j}} \\ \left.\frac{\delta^2\alpha}{\delta x^2}\right|_{\mathsf{i,j}} \\ \left.\frac{\delta^2\alpha}{\delta y^2}\right|_{\mathsf{i,j}} \\ \left.\frac{\delta^2\alpha}{\delta x\delta y}\right|_{\mathsf{i,j}} \end{bmatrix} = \begin{bmatrix} \alpha_{\mathsf{i}-p,\mathsf{j}-q} - \alpha_{\mathsf{i,j}} \\ \\ \vdots \\ \vdots \\ \\ \alpha_{\mathsf{i}+p,\mathsf{j}+q} - \alpha_{\mathsf{i,j}} \end{bmatrix} \tag{3.55}$$

If p and q are chosen so that $(2p+1)(2q+1)-1=5$, i.e. equals the number of unknowns, the system (3.55) is determined and can be solved. This, however, would simply give derivatives still affected by the inaccuracies in α. If instead, p and q are chosen so that $(2p+1)(2q+1)-1>5$, the system (3.55) is over-determined but can still be solved in a least square sense[2]. A least square solution has the advantage of minimizing the expected deviation between α and the unpolluted solution that α approximates. In order to compute the curvature at node (i,j), the system (3.55) is first solved with $\alpha=\phi$ and $p=q=2$; the normal $\boldsymbol{n}$ at node (i,j) is then computed as:

$$\boldsymbol{n}|_{\mathsf{i,j}} = \frac{(N_x, N_y)}{\sqrt{N_x^2+N_y^2}} \qquad \text{with} \qquad N_x = \left.\frac{\delta\alpha}{\delta x}\right|_{\mathsf{i,j}} \qquad \text{and} \quad N_y = \left.\frac{\delta\alpha}{\delta y}\right|_{\mathsf{i,j}} \ . \tag{3.56}$$

Next, the system (3.55) is solved again twice, for $\alpha=n_x$ and $\alpha=n_y$, and the curvature κ is finally obtained as:

$$\kappa|_{\mathsf{i,j}} = \left.\frac{\delta n_x}{\delta x}\right|_{\mathsf{i,j}} + \left.\frac{\delta n_y}{\delta y}\right|_{\mathsf{i,j}} \ . \tag{3.57}$$

[2] Given an over-determined linear system $Ax=b$, the least square solution is obtained by solving $A^tAx=A^tb$.

3.4.5 Reinitialization Equation

The reinitialization Eq. (2.39) is a key step in the ACLSM in that it allows maintaining a fixed interface width at all times. The integration in pseudo-time τ of Eq. (2.39) was performed via a second order Runge-Kutta scheme that comprised the following steps (the node indices have been dropped for clarity):

$$
\begin{aligned}
\psi^{\mathsf{n}+\frac{1}{2}} &= \psi^{\mathsf{n}} + \frac{\delta\tau}{2} \cdot R_{\psi}^{\mathsf{n}} \ , \\
\psi^{\mathsf{n}+1} &= \psi^{\mathsf{n}} + \delta\tau \cdot R_{\psi}^{\mathsf{n}+\frac{1}{2}} \ ,
\end{aligned}
\tag{3.58}
$$

In Eq. (3.58), $\delta\tau = \delta x/2$ is the time step and R_ψ represents a conservative discretization of the RHS of Eq. (2.39):

$$
R_\psi\big|_{\mathsf{i,j,k}}^{\mathsf{n}} = -\frac{F\big|_{\mathsf{i}+\frac{1}{2},\mathsf{j,k}}^{\mathsf{n}} - F\big|_{\mathsf{i}-\frac{1}{2},\mathsf{j,k}}^{\mathsf{n}}}{\delta x} - \frac{G\big|_{\mathsf{i,j}+\frac{1}{2},\mathsf{k}}^{\mathsf{n}} - G\big|_{\mathsf{i,j}-\frac{1}{2},\mathsf{k}}^{\mathsf{n}}}{\delta y} - \frac{H\big|_{\mathsf{i,j,k}+\frac{1}{2}}^{\mathsf{n}} - H\big|_{\mathsf{i,j,k}-\frac{1}{2}}^{\mathsf{n}}}{\delta z} \ , \tag{3.59}
$$

with the numerical fluxes F, G and H defined as:

$$
\begin{aligned}
F &= \left[-\overline{\psi(1-\psi)}^{x} + \epsilon \left(\frac{\delta_2 \psi}{\delta_2 x} n_x^x + \frac{\delta_2 \overline{\psi}^{x}}{\delta_2 y} n_y^x + \frac{\delta_2 \overline{\psi}^{x}}{\delta_2 z} n_z^x \right) \right] n_x^x \ , \\
G &= \left[-\overline{\psi(1-\psi)}^{y} + \epsilon \left(\frac{\delta_2 \overline{\psi}^{y}}{\delta_2 x} n_x^y + \frac{\delta_2 \psi}{\delta_2 y} n_y^y + \frac{\delta_2 \overline{\psi}^{y}}{\delta_2 z} n_z^y \right) \right] n_y^y \ , \\
H &= \left[-\overline{\psi(1-\psi)}^{z} + \epsilon \left(\frac{\delta_2 \overline{\psi}^{z}}{\delta_2 x} n_x^z + \frac{\delta_2 \overline{\psi}^{z}}{\delta_2 y} n_z^z + \frac{\delta_2 \psi}{\delta_2 z} n_z^z \right) \right] n_z^z \ .
\end{aligned}
\tag{3.60}
$$

The face normals in Eq. (3.60) are computed as described in Subsection 3.4.3.

3.5 Full Solution Algorithm

Steps of the projection method coupled with the ACLSM

1. Advance the phase-field function ψ from time level n to $\mathsf{n}+1$ by using the schemes outlined in Subsection 3.4.1.

2. Reconstruct the signed distance function $\phi^{\mathsf{n}+1}$ from $\psi^{\mathsf{n}+1}$ by using the FSM as described in Subsection 3.4.2.

3. Compute the face normals by using the discretization presented in Subsection 3.4.3.

4. Compute the curvature via the LSR outlined in Subsection 3.4.4.

5. Reinitialize the hyperbolic tangent profile of $\psi^{\mathsf{n}+1}$ by solving the reinitialization equation as described in Subsection 3.4.5.

6. Compute a tentative velocity u_i^{**} by performing steps 1. and 2. of the projection method detailed in Section 3.2. For this case, $f_i = s_i = 0$.

7. Compute the pressure $p^{\mathsf{n}+1}$ by solving the PPE (3.12). The GFM is used to discretize the PPE near the interface; the curvature computed at step 4. is used in the jump condition.

8. Compute the velocity field $u_i^{\mathsf{n}+1}$ at time level $\mathsf{n}+1$ by applying the correction step.

A graphical representation of the above solution algorithm is given in the flowchart in Figure 3.4.

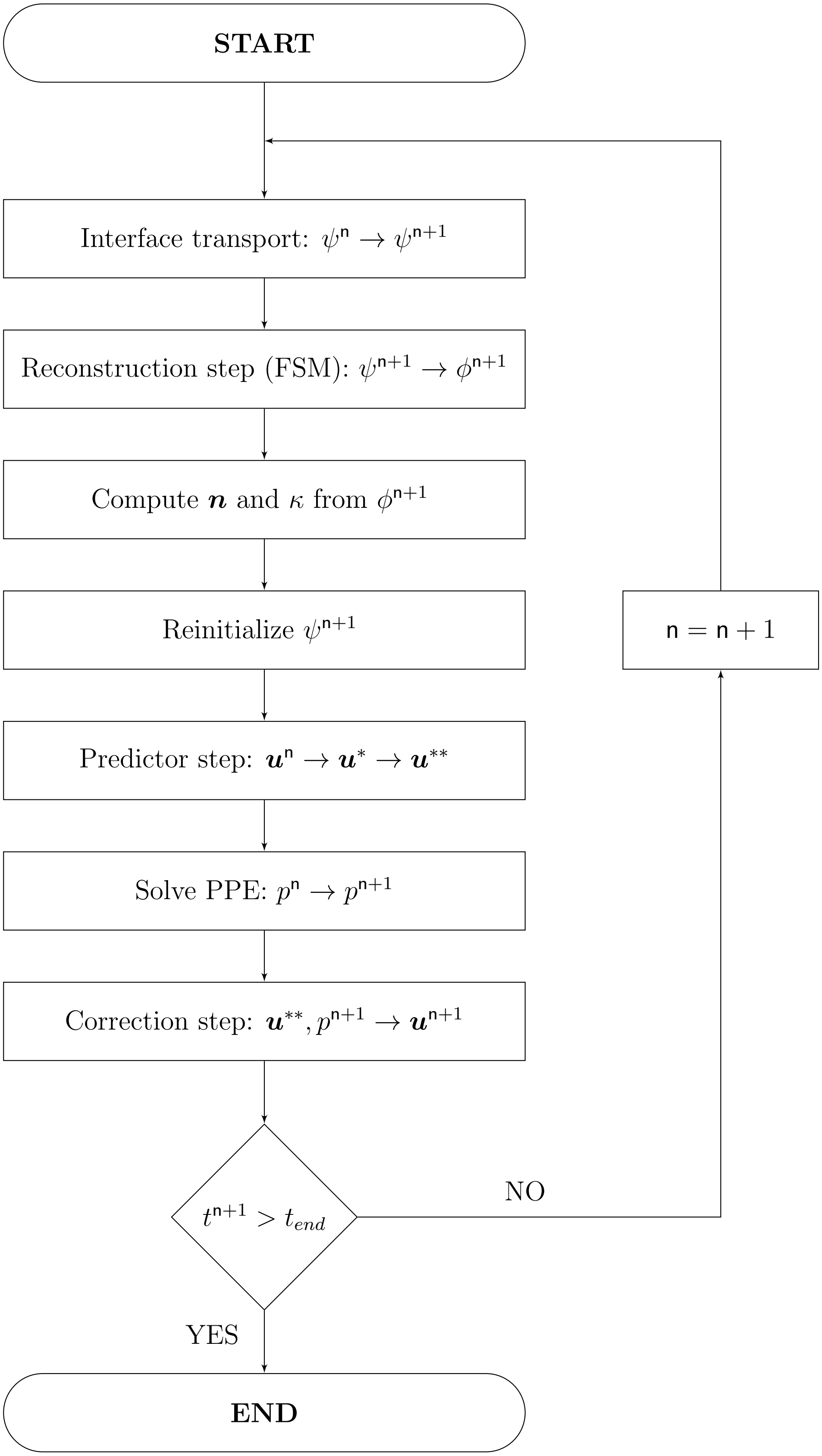

START
Interface transport: $\psi^{\mathsf{n}} \to \psi^{\mathsf{n}+1}$
Reconstruction step (FSM): $\psi^{\mathsf{n}+1} \to \phi^{\mathsf{n}+1}$
Compute $\boldsymbol{n}$ and κ from $\phi^{\mathsf{n}+1}$
Reinitialize $\psi^{\mathsf{n}+1}$
Predictor step: $\boldsymbol{u}^{\mathsf{n}} \to \boldsymbol{u}^{*} \to \boldsymbol{u}^{**}$
Solve PPE: $p^{\mathsf{n}} \to p^{\mathsf{n}+1}$
Correction step: $\boldsymbol{u}^{**}, p^{\mathsf{n}+1} \to \boldsymbol{u}^{\mathsf{n}+1}$
$t^{\mathsf{n}+1} > t_{end}$
NO
$\mathsf{n} = \mathsf{n} + 1$
YES
END

Figure 3.4 Flowchart describing the steps of the projection method coupled with the ACLSM. This algorithm is used for the simulation of liquid droplet-laden turbulence.

Steps of the projection method coupled with the IBM

1. For each solid particle, solve the ODEs in Eq. (3.40) to compute $\boldsymbol{x}_p$, $\boldsymbol{u}_p$ and $\boldsymbol{\omega}_p$ at the next time level $\mathsf{n}+1$.

2. Compute a tentative velocity u_i^* by performing step 1. of the projection method detailed in Section 3.2.

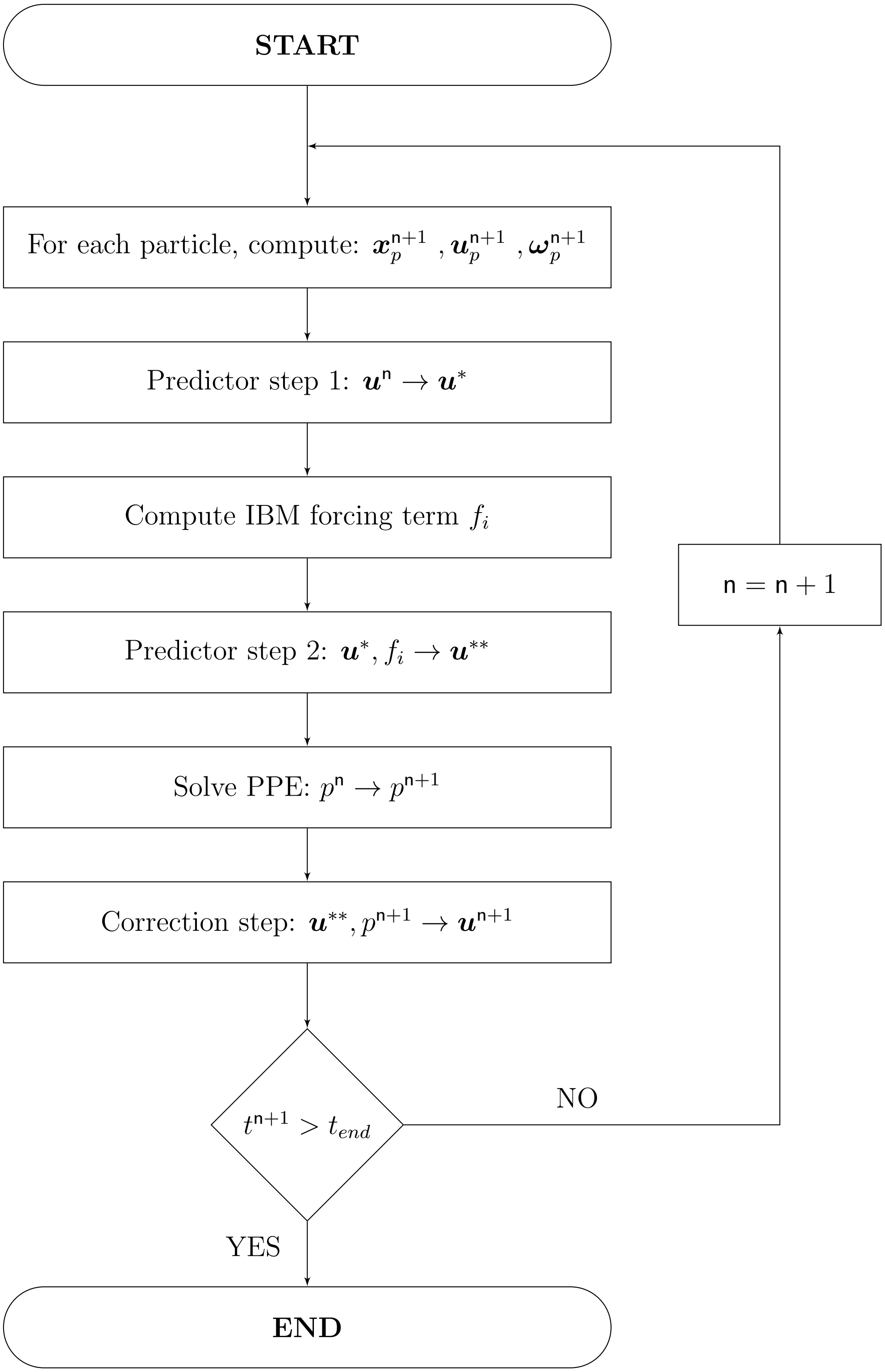
3. Compute the forcing term f_i by using the method outlined in Subsection 3.3.1

4. Compute a second tentative velocity u_i^{**} by performing step 2. of the projection method detailed in Section 3.2. For this case, $s_i = 0$.

5. Compute the pressure $p^{\mathsf{n}+1}$ by solving the PPE (3.12). Standard centered finite differencing is used to discretize the PPE. In this case, the PPE is *separable.*

6. Compute the velocity field $u_i^{\mathsf{n}+1}$ at time level $\mathsf{n}+1$ by applying the correction step.

A graphical representation of the above solution algorithm is given in the flowchart in Figure 3.5.



START
For each particle, compute: $\boldsymbol{x}_p^{\mathsf{n}+1}$, $\boldsymbol{u}_p^{\mathsf{n}+1}$, $\boldsymbol{\omega}_p^{\mathsf{n}+1}$
Predictor step 1: $\boldsymbol{u}^{\mathsf{n}} \to \boldsymbol{u}^*$
Compute IBM forcing term f_i
Predictor step 2: $\boldsymbol{u}^*, f_i \to \boldsymbol{u}^{**}$
Solve PPE: $p^{\mathsf{n}} \to p^{\mathsf{n}+1}$
Correction step: $\boldsymbol{u}^{**}, p^{\mathsf{n}+1} \to \boldsymbol{u}^{\mathsf{n}+1}$
$t^{\mathsf{n}+1} > t_{end}$
NO
$\mathsf{n} = \mathsf{n} + 1$
YES
END

Figure 3.5 Flowchart describing the steps of the projection method coupled with the IBM. This algorithm is used for the simulation of solid particle-laden turbulence.

Chapter 4

Validation of the Numerical Methods

The numerical methods presented in Chapter 3 were validated by comparing the numerical with the exact solution for a number of test cases. Each test problem was designed to assess the behavior and accuracy of the solution algorithm under different conditions. A two dimensional domain of size $L \times L$ was considered in all the tests. In what follows, N indicates the number of grid nodes in each direction. The relative error of a numerical variable α at node (i, j) is represented by $\xi(\alpha)$:

$$\xi(\alpha)\big|_{\mathsf{i,j}} = \left|1 - \frac{\alpha}{\overline{\alpha}}\right|_{\mathsf{i,j}}, \quad \forall\, \alpha \neq 0 \ , \tag{4.1}$$

where $\overline{\alpha}$ is the exact solution. The numerical accuracy of α was measured via the norm 1, $L_1(\alpha)$, and the norm infinity, $L_\infty(\alpha)$, of the absolute error:

$$L_1(\alpha) = \frac{1}{N_\Sigma^2} \sum_{\mathsf{i,j}} |\alpha - \overline{\alpha}|_{\mathsf{i,j}} \ , \qquad L_\infty(\alpha) = \max |\alpha - \overline{\alpha}|_{\mathsf{i,j}} \ , \tag{4.2}$$

where N_Σ is the number of grid points used to compute L_1. Both norms are calculated by considering all the grid nodes in the domain, unless otherwise stated. This chapter focuses exclusively on the validation of the numerical methods used to treat fluid/fluid interfaces, namely the *Accurate Conservative Level Set Method* (ACLSM), the *Ghost Fluid Method* (GFM) and their coupling with the projection method for the solution of the *Navier-Stokes Equations* (NSE). The reader is referred to [38] for a detailed analysis of the accuracy and performance of the *Immersed Boundary Method* (IBM) used to simulate fluid/solid interfaces.

4.1 Zalesak's Disk

The algorithm for the transport and reinitialization of the phase field function, ψ, was tested by simulating the rigid body rotation of a notched circle, also known as *Zalesak's disk* [84]. This test checks whether the numerical schemes used to discretize the ACLSM equations is capable of correctly resolving sharp corners and thin geometric structures. Initially, the disk is centered at the point $(0.5, 0.5)$ in a square domain of size $L = 1$, with a notch of height 0.4 and width 0.1. A constant velocity field given by

$$u(x,y) = \frac{\pi}{2}(y - 0.5) \quad \text{and} \quad v(x,y) = \frac{\pi}{2}(0.5 - x) \tag{4.3}$$

is applied to the disk for the entire simulation. Consequently, the disk is expected to rotate around the point $(0.5, 0.5)$ without changing shape. The test was run until $t = 4$, that is until the interface completed a full rotation. The simulation was repeated for four different meshes of 32, 64, 128, and 256 grid points in order to assess convergence under grid refinement. In all cases, the time step width, δt, was set to $\delta x/4$. After each advection step, the reinitialization Eq. (2.39) was solved for four pseudo-time steps. Figure 4.1 displays the shape of the disk after one full rotation. The coarsest grid tested ($N = 32$) does not allow the accurate resolution of the sharp corners, resulting in large deformations of the interface. By increasing the grid size ($N = 64$, 128, 256), the solution becomes satisfactory.

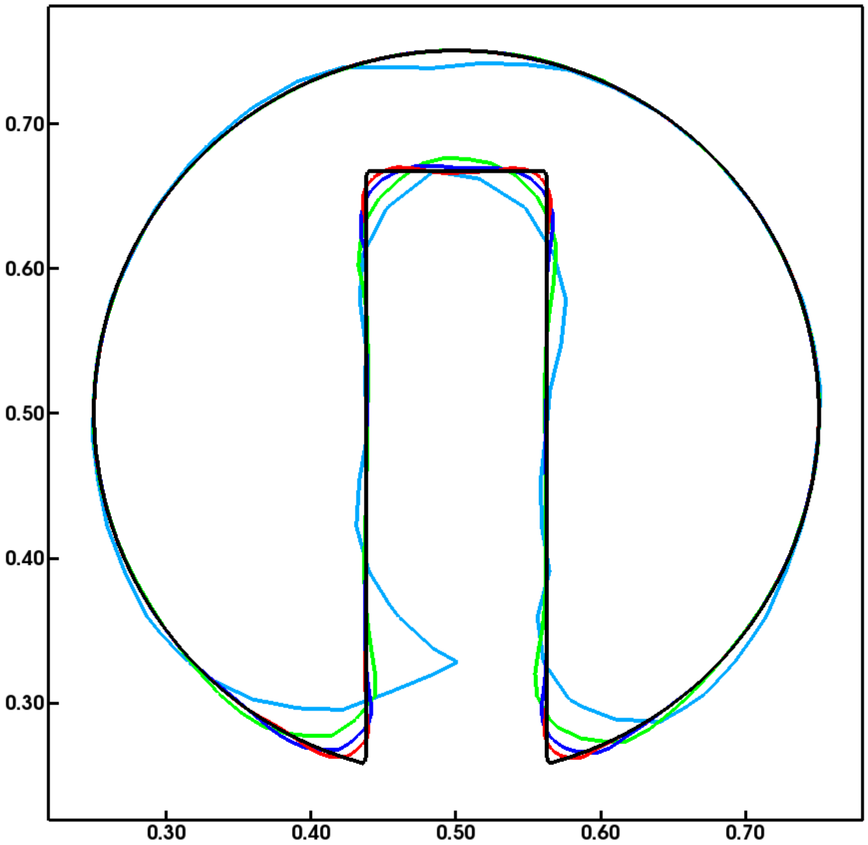


Figure 4.1 Notched disk after one full rotation. The black line is the exact solution obtained by a rigid body rotation. The cyan, green, blue and red lines are the numerical solutions obtained on grids of 32, 64, 128, and 256 mesh points respectively.

Table 4.1 shows the norm 1 of the absolute error of the numerical solution after one full rotation for increasingly finer meshes. The norm is computed in a band of width $8\delta x$ around the interface. It is seen that the order of convergence is between 1 and 2. This is expected since Eq. (2.38) for the transport of ψ is discretized via a numerical scheme that is of order less than two for non-smooth solutions (cf. Subsection 3.4.1). The conservation properties of the ACLSM were tested also by computing the volume enclosed by the interface, Ω^+, at time step n via

$$\Omega^+|^{\mathsf{n}} = \delta x \delta y \sum_{\mathsf{i,j}} \psi|^{\mathsf{n}}_{\mathsf{i,j}} \ . \tag{4.4}$$

Figure 4.2 shows the temporal development of error of the volume relative to the initial volume for different grids. It should be noted that the conservation properties of the ACLSM are excellent, even when the interface is subjected to strong deformations.

Table 4.1 Norm 1 of the absolute error of the numerical solution in a band of width $8\delta x$.

N	$L_1(\psi)$	Order
32	$2.38 \cdot 10^{-5}$	-
64	$7.43 \cdot 10^{-6}$	1.68
128	$1.97 \cdot 10^{-6}$	1.92
256	$7.80 \cdot 10^{-7}$	1.33

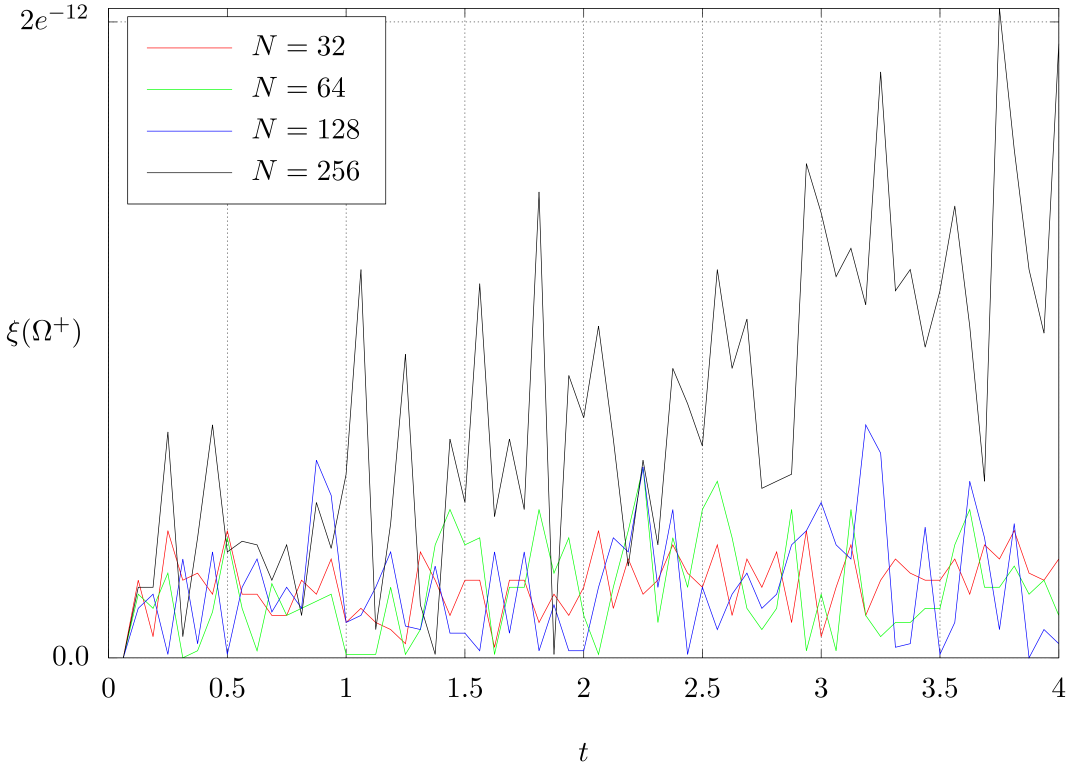


Figure 4.2 Temporal development of the volume error relative to the initial volume (%).

4.2 Accuracy of the FSM Reconstruction

The accuracy of the *Fast Sweeping Method* (FSM) was tested by reconstructing a signed distance function, ϕ, from a given phase field function, ψ, for the case of a 2D circle of radius $R = 0.25$. The circle was centered in a square domain of size $L = 1$. The exact signed distance function, $\overline{\phi}$, and the corresponding phase field function, $\overline{\psi}$, for this configuration are given by

$$\overline{\phi}(x, y) = \sqrt{(x - 0.5)^2 - (y - 0.5)^2} - R \tag{4.5}$$

and

$$\overline{\psi}(x, y) = \frac{1}{2}\left(\tanh\left(\frac{-\overline{\phi}}{\epsilon}\right) + 1\right) \ . \tag{4.6}$$

The initial condition for the FSM is obtained from $\overline{\psi}$ via Eq. (3.50). Initially, a tentative signed distance function, ϕ^*, is computed via the first order FSM. Subsequently, ϕ^* is refined in a small band around the interface by using a third order FSM[1]. It is seen in Table 4.2 that the reconstruction method used in this work yields a fourth-order numerical signed distance function, ϕ, despite the scheme used in the refinement step being only third order accurate.

Table 4.2 Development of the error of the signed distance function ϕ for increasingly finer grids.

N	$L_1(\phi)$	Order	$L_\infty(\phi)$	Order
32	$8.95 \cdot 10^{-5}$	-	$1.82 \cdot 10^{-3}$	-
64	$5.23 \cdot 10^{-6}$	4.10	$4.13 \cdot 10^{-5}$	5.46
128	$3.10 \cdot 10^{-7}$	4.08	$1.88 \cdot 10^{-6}$	4.46
256	$1.98 \cdot 10^{-8}$	3.97	$1.10 \cdot 10^{-7}$	4.10
512	$1.22 \cdot 10^{-9}$	4.02	$5.70 \cdot 10^{-9}$	4.27

[1] The third order accuracy is achieved via a third order WENO scheme (see [85]).

4.2.1 Accuracy of the Normal Vector Calculation

The exact normal, $\overline{\boldsymbol{n}}$, for the signed distance function defined in Eq. (4.5) is given by

$$\overline{\boldsymbol{n}} = \frac{\nabla\overline{\phi}}{||\nabla\overline{\phi}||} = \frac{(\, x - 0.5\,,\; y - 0.5\,)}{\sqrt{(x-0.5)^2 + (y-0.5)^2}}\ . \tag{4.7}$$

The numerical normal vector, $\boldsymbol{n}$, was computed from the level set function after the FSM reconstruction. Two different methods of computing $\boldsymbol{n}$ were tested, namely

1. a standard second order centered *Finite Difference* (FD) scheme, and
2. a second order *Least Squares* (LS) scheme.

The FD scheme to compute $\boldsymbol{n}$ at the grid node (i,j) reads:

$$\boldsymbol{n}|_{\mathsf{i,j}} = \frac{(N_x, N_y)}{\sqrt{N_x^2 + N_y^2}} \quad \text{with} \quad N_x = \frac{\phi|_{\mathsf{i+1,j}} - \phi|_{\mathsf{i-1,j}}}{2\delta x} \quad \text{and} \quad N_y = \frac{\phi|_{\mathsf{i,j+1}} - \phi|_{\mathsf{i,j-1}}}{2\delta y}\ , \tag{4.8}$$

while the LS solution is obtained by solving Eq. (3.55) with $\alpha = \phi$, and then computing $\boldsymbol{n}$ at the grid nodes via Eq. (3.56). Table 4.3 shows that both methods achieve second order accuracy, with the FD scheme producing slightly smaller absolute errors. It should be noted that the normal vector comes into play only in the re-initialization Eq. (2.39). As mentioned in Subsection 3.4.5, a conservative discretization of Eq. (2.39) requires $\boldsymbol{n}$ to be computed at the faces of a scalar cell via FD. Therefore the results for the circle test presented here are not meant to verify the accuracy of the face normals used in the reinitialization equation, but only to analyze the effect of the high order FSM on the geometric properties of ϕ.

Table 4.3 Development of the absolute error of the numerical normal $\boldsymbol{n}$ under grid refinement. The norms were computed by considering a band of width $8\,\delta x$ around the interface.

N	FD				LS			
	$L_1(\boldsymbol{n})$	Order	$L_\infty(\boldsymbol{n})$	Order	$L_1(\boldsymbol{n})$	Order	$L_\infty(\boldsymbol{n})$	Order
32	$4.02 \cdot 10^{-3}$	-	$2.11 \cdot 10^{-2}$	-	$6.87 \cdot 10^{-3}$	-	$5.50 \cdot 10^{-2}$	-
64	$9.87 \cdot 10^{-4}$	2.03	$2.67 \cdot 10^{-3}$	2.98	$6.87 \cdot 10^{-3}$	2.04	$8.87 \cdot 10^{-2}$	2.63
128	$2.44 \cdot 10^{-4}$	2.01	$5.93 \cdot 10^{-4}$	2.17	$4.10 \cdot 10^{-4}$	2.03	$1.63 \cdot 10^{-3}$	2.44
256	$6.10 \cdot 10^{-4}$	2.00	$1.35 \cdot 10^{-4}$	2.14	$1.01 \cdot 10^{-4}$	2.02	$3.57 \cdot 10^{-4}$	2.19
512	$1.53 \cdot 10^{-5}$	2.00	$3.22 \cdot 10^{-5}$	2.07	$2.53 \cdot 10^{-5}$	2.00	$8.40 \cdot 10^{-5}$	2.09

4.2.2 Accuracy of the Curvature Calculation

The exact curvature, $\overline{\kappa}$, for the signed distance function defined in Eq. (4.5) is given by

$$\overline{\kappa} = \nabla \cdot \left(\frac{\nabla \overline{\phi}}{||\nabla \overline{\phi}||} \right) = \frac{1}{\sqrt{(x-0.5)^2 + (y-0.5)^2}} \; . \tag{4.9}$$

Three different methods for computing the numerical curvature, κ, were tested, namely:

1. a second order centered FD scheme applied to ϕ,
2. a second order centered FD scheme applied to the face normals defined in Subsection 3.4.3 (FD+FN), and
3. a second order LS scheme.

The FD method relies on the analytical expression for κ, that is

$$\kappa = \frac{\nabla \phi}{||\nabla \phi||} = \frac{\phi_x^2 \phi_{yy} - 2\phi_x \phi_y \phi_{xy} + \phi_y^2 \phi_{xx}}{(\phi_x^2 + \phi_y^2)^{1.5}} \; , \tag{4.10}$$

where the subscripts indicate partial derivatives. The above equation is discretized at the grid nodes via centered FD. The methods 2. and 3. are described in Subsection 3.4.4. The development of the absolute error under grid refinement in a band of width $8\delta x$ around the interface is shown in Table 4.4 for the three methods. Second order accuracy is achieved by all the schemes, but the norms of the absolute error are almost an order of magnitude smaller for the FD-based methods. Figure 4.3, 4.4, and 4.5 display the contours plot of the absolute error of κ in a band of width $8\delta x$ for the three methods. It is seen that the absolute error produced by methods 1. (cf. Figure 4.4) and 2. (cf. Figure 4.5) is oscillatory. In contrast, the LS solution exhibits a much smoother behavior, as shown in Figure 4.3. This is not surprising, since the advantage of method 3. over the methods 1. and 2. is its ability of minimizing the effect of the noise polluting ϕ on the curvature. Since oscillations in the curvature may cause parasitic currents that are detrimental to the stability and accuracy of the solution algorithm (cf. Section 4.3), the LS scheme was chosen over the FD-based methods.

Table 4.4 Development of the absolute error of the numerical curvature κ under grid refinement. The norms were computed by considering a band of width $8\delta x$ around the interface.

N	FD		FD+FN		LS	
	$L_1(\kappa)$	Order	$L_1(\kappa)$	Order	$L_1(\kappa)$	Order
32	$3.20 \cdot 10^{-3}$	-	$3.08 \cdot 10^{-3}$	-	$2.90 \cdot 10^{-2}$	-
64	$8.31 \cdot 10^{-4}$	1.94	$7.01 \cdot 10^{-4}$	2.14	$7.51 \cdot 10^{-3}$	1.90
128	$1.96 \cdot 10^{-4}$	2.08	$1.52 \cdot 10^{-4}$	2.20	$1.88 \cdot 10^{-3}$	1.99
256	$5.05 \cdot 10^{-5}$	1.96	$3.99 \cdot 10^{-5}$	1.93	$4.66 \cdot 10^{-4}$	2.01
512	$1.27 \cdot 10^{-5}$	1.99	$1.01 \cdot 10^{-5}$	1.99	$1.17 \cdot 10^{-4}$	2.00

Table 4.5 Development of the absolute error of the numerical curvature κ under grid refinement. The norms were computed by considering the interpolated values of κ at the interface.

N	FD		FD+FN		LS	
	$L_1(\kappa)$	Order	$L_1(\kappa)$	Order	$L_1(\kappa)$	Order
32	$9.78 \cdot 10^{-4}$	-	$1.47 \cdot 10^{-3}$	-	$2.74 \cdot 10^{-2}$	-
64	$4.65 \cdot 10^{-4}$	1.07	$5.19 \cdot 10^{-4}$	1.51	$7.43 \cdot 10^{-3}$	1.90
128	$9.71 \cdot 10^{-5}$	2.26	$1.07 \cdot 10^{-4}$	2.28	$1.88 \cdot 10^{-3}$	1.96
256	$3.18 \cdot 10^{-5}$	2.61	$3.30 \cdot 10^{-5}$	1.70	$4.70 \cdot 10^{-4}$	2.00
512	$7.01 \cdot 10^{-6}$	2.18	$7.48 \cdot 10^{-6}$	2.14	$1.18 \cdot 10^{-4}$	1.99

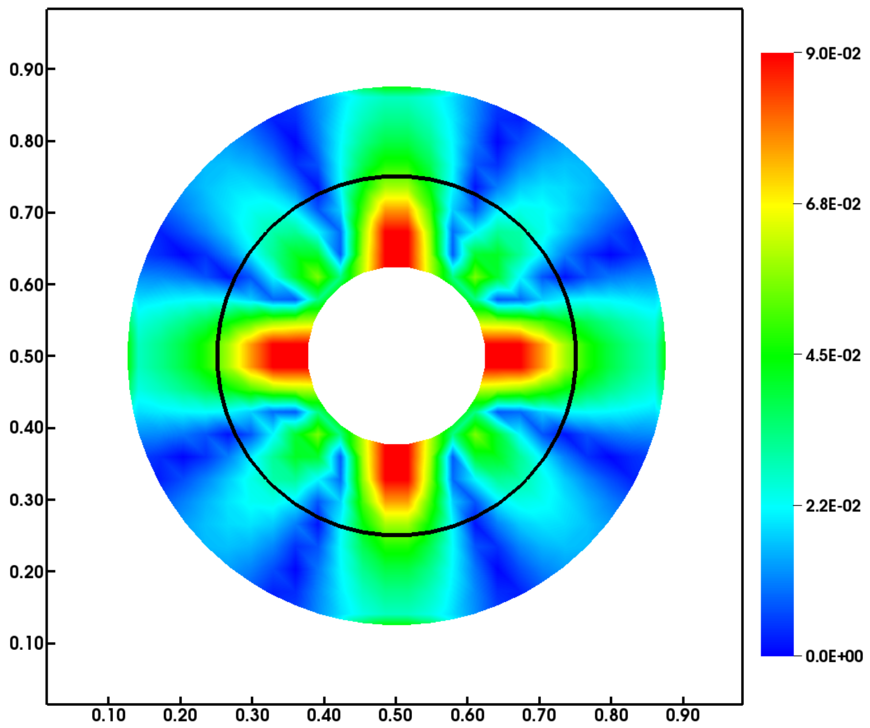


Figure 4.3 Contours of the absolute error of the numerical curvature obtained via the LS method. Only a band of width $8\delta x$ around the interface (black line) is shown.

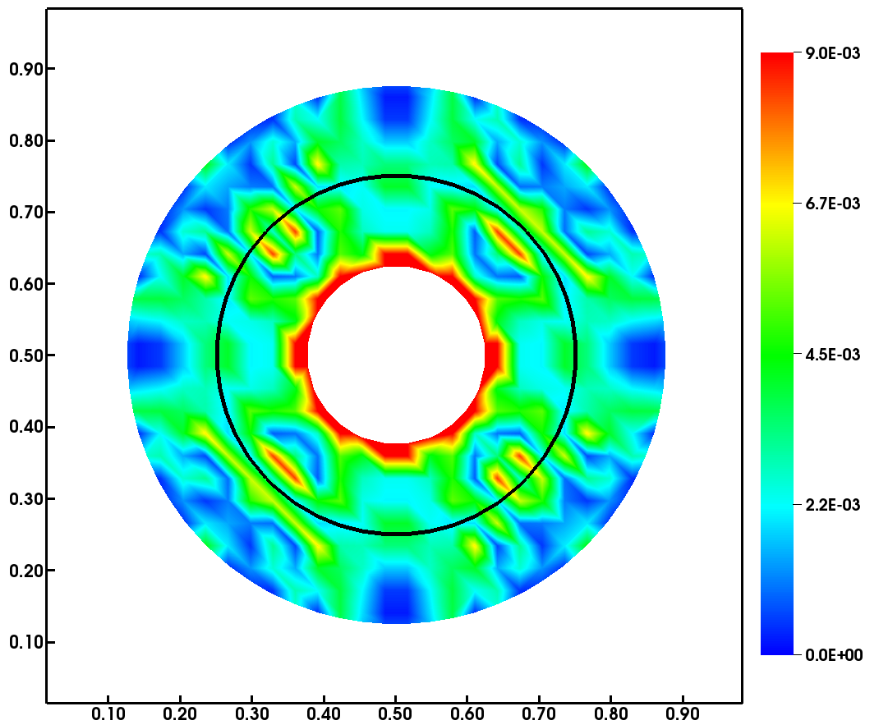


Figure 4.4 Contours of the absolute error of the numerical curvature obtained via the FD method. Only a band of width $8\delta x$ around the interface (black line) is shown.

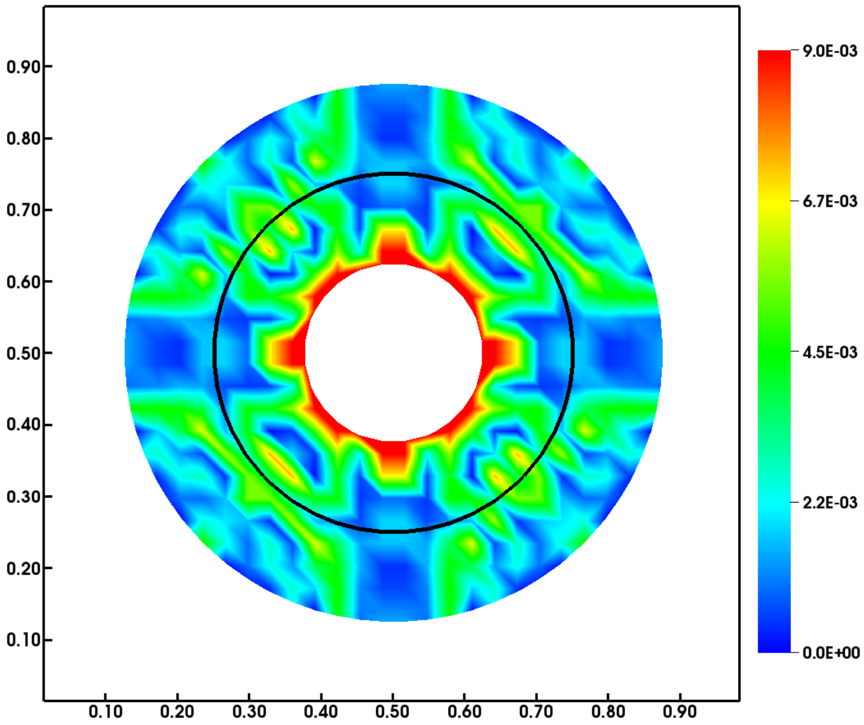


Figure 4.5 Contours of the absolute error of the numerical curvature obtained via the FD+FN method. Only a band of width $8\delta x$ around the interface (black line) is shown.

4.3 Stationary Interface

Interface simulation methods are known to be affected by the so-called *spurious* or *parasitic currents*. These are non-physical vortices that appear in the numerical solution in close proximity of the interface, even in the absence of any external forces. The causes of this phenomenon are:

1. an inconsistent discretization of the surface tension force and the pressure gradient, and

2. an inaccurate numerical curvature.

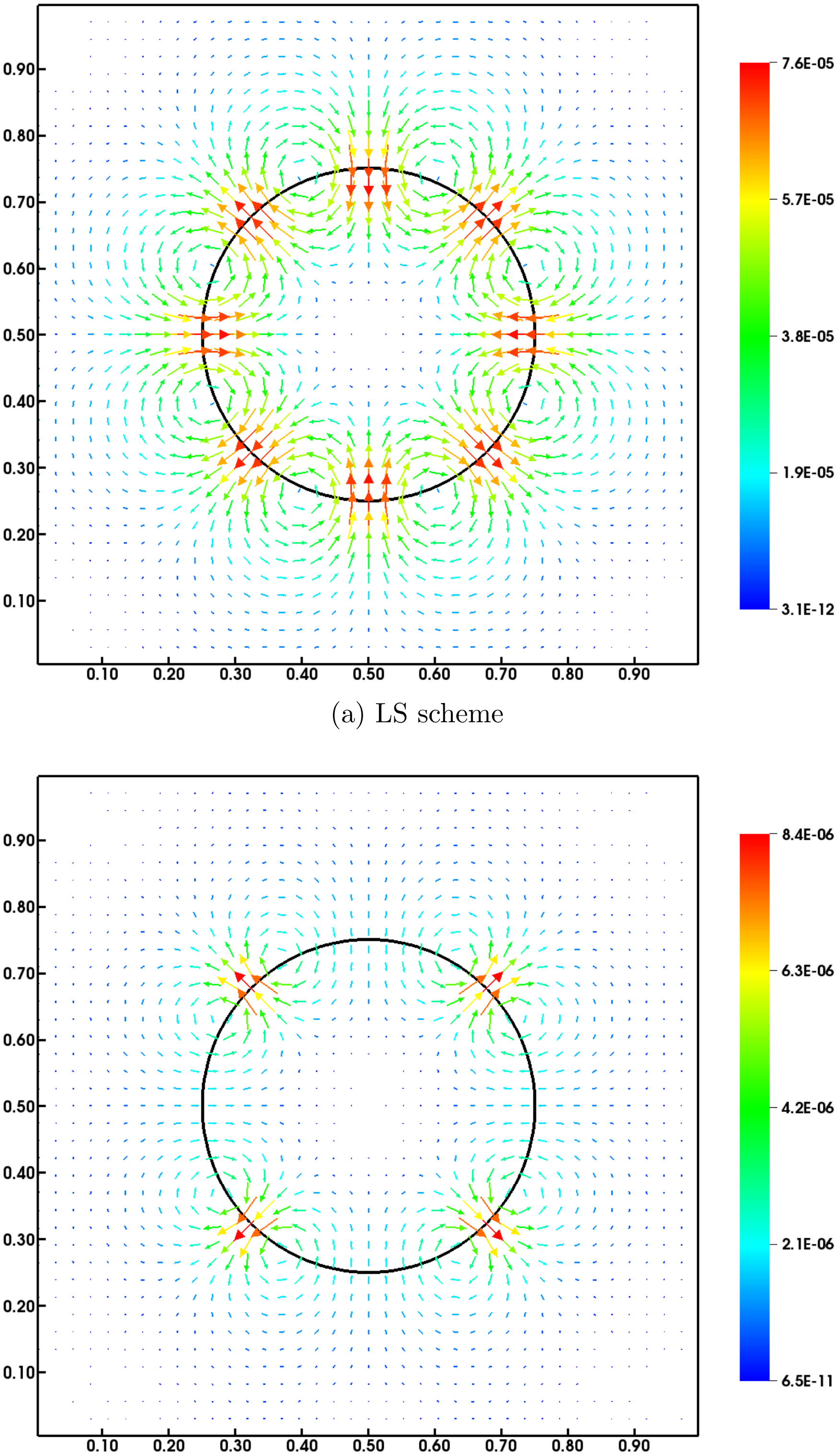
An inconsistent numerical scheme does not enforce the discrete balance between pressure and surface tension, thus producing spurious local fluid accelerations. *Continuum Surface Force* (CSF)-based methods are likely to be affected by this issue unless proper discretization techniques are used. In contrast, the GFM produces a consistent discretization by design and therefore automatically guarantees the discrete force balance at the interface. The issue of inaccurate curvature computations is more serious. First, a discretization of the curvature that is very sensitive to small changes in the shape of the interface causes large errors in the value of the capillary forces. Second, oscillations in the curvature field may produce non-physical local forces that result in spurious currents. Many FD-based schemes for the computation of the curvature are affected by the inaccuracies in the signed distance function and show order of convergence less than one under grid refinement. This means that the magnitude of the parasitic currents cannot be reduced by refining the mesh. In order to test whether these issues affect the numerical method used in this work, a stationary interface was considered. The interface was a circle of radius $R = 0.25$ centered in a 2D domain of size $L = 1$. The circle can be thought of as the cross section of a cylinder with infinite length. The viscosity and density were kept constant in the whole computational domain, including the interior of the interface. Initially the velocity field was set to zero and no forces except the surface tension were applied. Since no discontinuities in density and viscosity are present, and no external forces other than the capillarity force are accounted for, the stationary interface test is ideal to verify that the pressure gradient/surface tension balance is satisfied in a discrete sense. Under these conditions, the velocity is expected to be zero at all times and the interface to remain a circle. The dimensionless parameter that characterizes this case is the Laplace number

$$\mathfrak{La} = \frac{\mathfrak{Re}^2}{\mathfrak{We}} = \frac{\widetilde{\sigma}\widetilde{L}\widetilde{\rho}^-}{\widetilde{\mu}^-} \ , \tag{4.11}$$

where $\widetilde{\rho}^-$ and $\widetilde{\mu}^-$ are the reference density and molecular viscosity, $\widetilde{\sigma}$ is the surface tension coefficient, and $\widetilde{L}$ is the reference length. The flow field and the interface were evolved in time from $t = 0$ to $t = 10$ with $\mathfrak{La}$ set to 120 by using the algorithm outlined in Figure 3.4. In addition, both the FD and LS schemes for the computation of the curvature were tested. It is seen in the velocity vector plot of Figure 4.6 that, at a very early time ($t = 0.36$), both methods for the curvature computation create non-physical vortices surrounding the interface despite the mesh being quite fine ($N = 128$). Also, the maximum value of the velocity obtained by using the FD method (Figure 4.6b) is one order of magnitude smaller. This is consistent with the results on the accuracy of the two schemes reported in Section 4.2. By construction, the ACLSM assumes that each point on the interface, is transported by the underlying velocity field and therefore no normal fluid flux across the interface should be present (cf. Subsection 2.2.3). However, since the interface simulation methods that include the effect of surface tension are plagued by parasitic currents as discussed above, a normal flux is to be expected in the numerical solution. This can be observed in Figure 4.6. Nevertheless, it should be noted that such flux is only local, i.e. the overall net flux across the interface is zero. As the simulation progresses, the maximum value of the velocity is reduced by two orders of magnitude in the LS solution (Figure 4.7a and 4.8a), while it grows by four orders of magnitude in the FD solution (Figure 4.7b and 4.8b). It should also be noted that the symmetry of the flow field is lost in the FD case, and consequently the interface is distorted, as shown in Figure 4.8b. Lastly, Table 4.6 shows that the reduction of the parasitic currents does not converge under grid refinement for the FD case.

Table 4.6 Maximum value of the velocity magnitude at $t = 10$ for three different grid sizes.

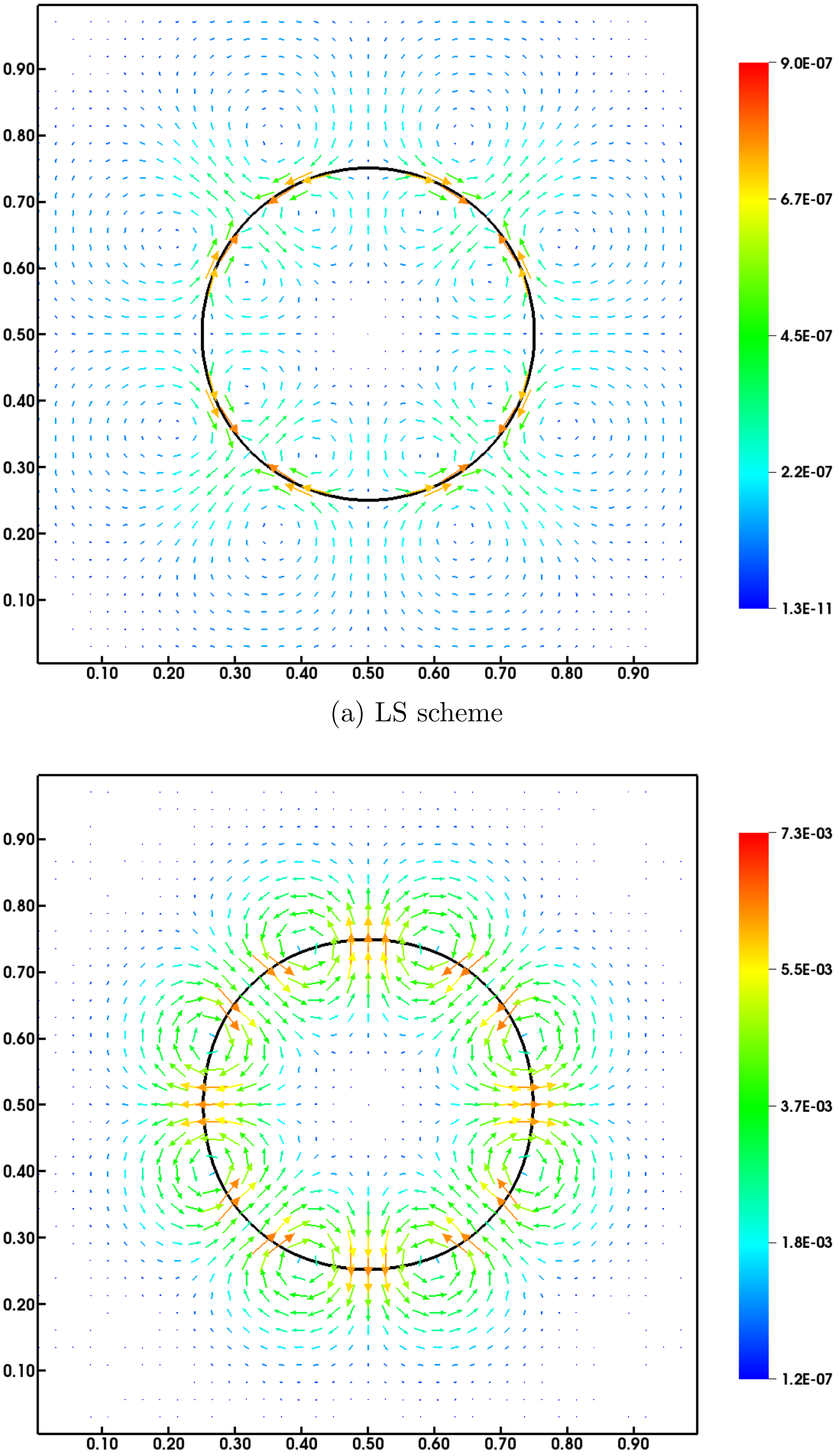
N	FD		LS	
	$L_\infty(\boldsymbol{u})$	Order	$L_\infty(\boldsymbol{u})$	Order
32	$1.70 \cdot 10^{-6}$	-	$2.00 \cdot 10^{-5}$	-
64	$2.20 \cdot 10^{-6}$	-0.37	$5.90 \cdot 10^{-6}$	1.76
128	$2.93 \cdot 10^{-2}$	-13.7	$8.80 \cdot 10^{-8}$	2.76



(a) LS scheme

(b) FD scheme

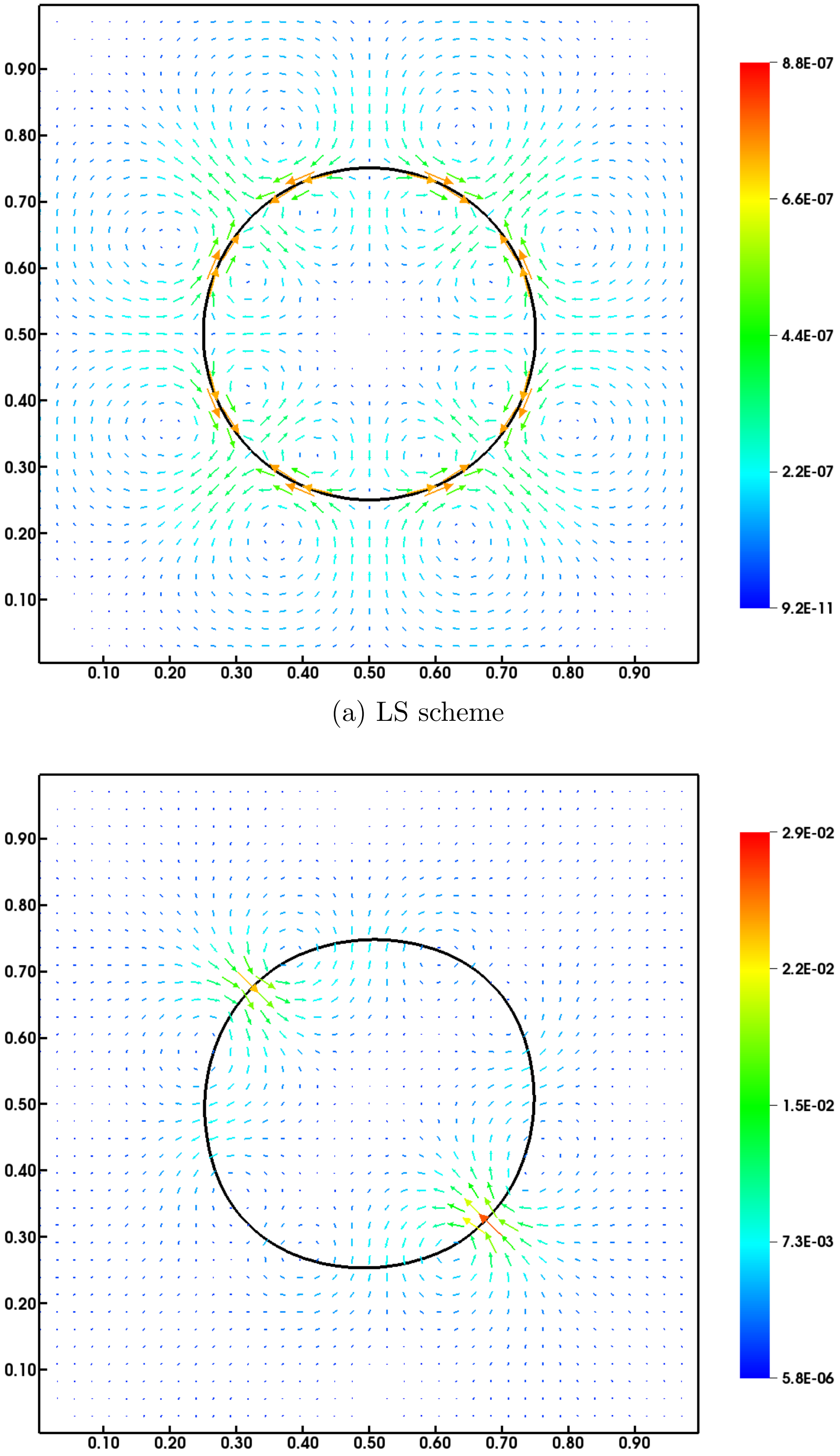
Figure 4.6 Vector plot of the velocity field at $t = 0.36$ for the stationary interface test. Both the FD and LS solution are shown for a grid of 128 nodes.



(a) LS scheme

(b) FD scheme

Figure 4.7 Vector plot of the velocity field at $t = 5.05$ for the stationary interface test. Both the FD and LS solution are shown for a grid of 128 nodes.



(a) LS scheme

(b) FD scheme

Figure 4.8 Vector plot of the velocity field at $t = 9.92$ for the stationary interface test. Both the FD and LS solution are shown for a grid of 128 nodes.

4.4 Standing Wave

In order to test the capability of the algorithm to handle the interplay of viscosity ratio, density ratio, and surface tension, the interface between two fluids was considered. The level set at $t = 0$, ϕ_0, was initialized in a domain of size $[0, 2\pi] \times [0, 2\pi]$ via:

$$\phi_0(x, y) = y - \pi + a_0 \cos(x) \ . \tag{4.12}$$

Equation (4.12) represents a flat interface slightly perturbed by a cosine wave with initial amplitude $a_0 = 0.01\lambda$, where $\lambda = 2\pi$ is the wavelength of the perturbation. Periodicity was enforced in the x direction, while slip conditions were used in the y direction. Initially, the velocity field was set to zero in both fluids. The goal of this test was to accurately compute the time evolution of the interface amplitude and compare it with the analytic solution derived by Prosperetti [57] for the case of two fluids with identical kinematic viscosity. The analytic solution reads:

$$\begin{aligned} a(\widetilde{t}) = & \frac{4(1-4\beta)\widetilde{\nu}^2\widetilde{k}^4}{8(1-4\beta)\widetilde{\nu}^2\widetilde{k}^4 + \widetilde{\omega}_0^2}\, a_0 \ \mathrm{erfc}\left(\sqrt{\widetilde{\nu}\widetilde{k}^2\widetilde{t}}\right) \ + \\ & + \sum_{i=1}^{4} \frac{z_i}{Z_i} \frac{\widetilde{\omega}_0^2 a_0}{z_i^2 - \widetilde{\nu}\widetilde{k}^2} \ \exp[(z_i^2 - \widetilde{\nu}\widetilde{k}^2)\widetilde{t}] \ \mathrm{erfc}\left(z_i\sqrt{\widetilde{t}}\right) \ , \end{aligned} \tag{4.13}$$

where $\widetilde{k} = 2\pi/\lambda$ is the wavenumber of the perturbation, z_i the complex roots of the quadratic equation

$$z^4 - 4\beta\sqrt{k^2\widetilde{\nu}}z^3 + 2(1-6\beta)k^2\widetilde{\nu}z^2 + 4(1-3\beta)(k^2\widetilde{\nu})^{3/2}z + (1-4\beta)k^4\widetilde{\nu}^2 + \widetilde{\omega}_0^2 = 0 \ , \tag{4.14}$$

and Z_i a coefficient computed via

$$Z_i = \prod_{\substack{j=1 \\ j \neq i}}^{4} (z_j - z_i) \ . \tag{4.15}$$

The frequency $\widetilde{\omega}_0$ and the coefficient β in Eq. (4.13) are defined as

$$\widetilde{\omega}_0 = \sqrt{\frac{\widetilde{\sigma}\widetilde{k}^3}{\widetilde{\rho}^+ + \widetilde{\rho}^-}} \qquad \text{and} \qquad \beta = \frac{\widetilde{\rho}^+\,\widetilde{\rho}^-}{(\widetilde{\rho}^+ + \widetilde{\rho}^-)^2} \tag{4.16}$$

By using $\widetilde{k}^{-1}$ as the reference length, $\widetilde{L}$, and $\widetilde{\omega_0}/\widetilde{k}$ as the reference velocity, $\widetilde{U}$, Eq. (4.13) can be recast in dimensionless form:

$$a(t) = \frac{4(1-4\beta)\nu^2}{8(1-4\beta)\nu^2+1} a_0 \operatorname{erfc}(\sqrt{\nu t}) + \sum_{i=1}^{4} \frac{y_i}{Y_i} \frac{a_0}{y_i^2 - \nu} \exp[(y_i^2 - \nu)t] \operatorname{erfc}(y_i \sqrt{t}) \ , \tag{4.17}$$

where now y_i are the complex roots of

$$y^4 - 4\beta\sqrt{\nu}y^3 + 2(1-6\beta)\nu y^2 + 4(1-3\beta)\nu^{3/2}y + (1-4\beta)\nu^2 + 1 = 0 \ , \tag{4.18}$$

and

$$Y_i = \prod_{\substack{j=1 \\ j\neq i}}^{4} (y_j - y_i) \ . \tag{4.19}$$

Given the reference quantities introduced above, the dimensionless viscosity, ν, the Reynolds' number, $\mathfrak{Re}$, and the Weber's number, $\mathfrak{We}$, for this case are:

$$\nu = \frac{\widetilde{k}^2\widetilde{\nu}}{\widetilde{\omega_0}} \ , \qquad \mathfrak{Re} = \frac{1}{\nu} \qquad \text{and} \qquad \mathfrak{We} = \frac{1}{\eta_\rho + 1} \ , \tag{4.20}$$

where $\eta_\rho = \widetilde{\rho}^+/\widetilde{\rho}^-$ is the density ratio. Clearly, in order to have the same kinematic viscosity in both fluids, η_ρ must equal the ratio of the molecular viscosities $\eta_\mu = \widetilde{\mu}^+/\widetilde{\mu}^-$ It is seen from the definitions (4.20) that, once η_ρ is set, $\mathfrak{We}$ becomes a constant. Therefore only two dimensionless parameter, namely $\mathfrak{Re}$ and η_ρ, are required to completely describe this case[2].

[2] Given than only one other dimensionless parameter is needed in addition to η_ρ, some authors prefer to use the Ohnesorge number $\mathfrak{Oh} = \sqrt{\mathfrak{We}}/\mathfrak{Re}$ in lieu of $\mathfrak{Re}$ and $\mathfrak{We}$.

A numerical solution was obtained for two sets of parameters:

1. $\mathfrak{Re} = 10$ and $\eta_\rho = 10$

2. $\mathfrak{Re} = 10$ and $\eta_\rho = 1000$

Figure 4.9 and 4.10 show that, in both cases, the temporal development of the numerical amplitude matches the exact solution for a grid of size $N = 256$. While other researchers showed comparable results by using coarser meshes, the accuracy of the method used in this work is limited by the order of the phase field function transport scheme.

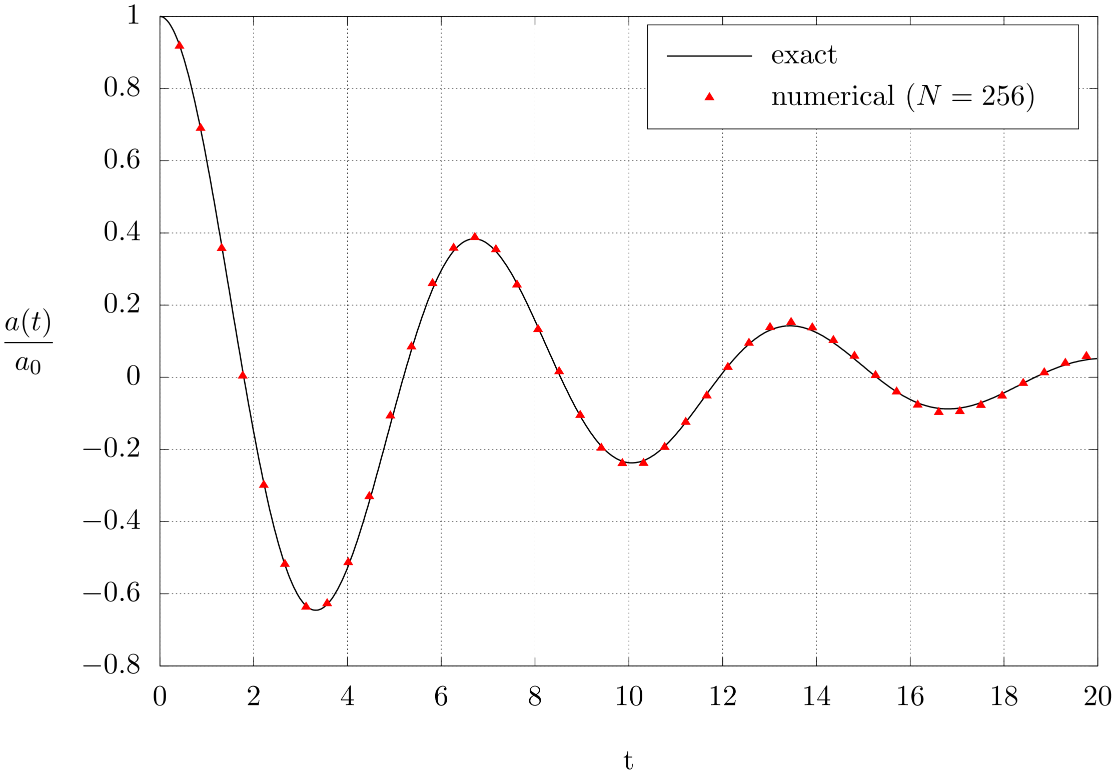


Figure 4.9 Temporal development of amplitude of the interface oscillation normalized by its initial value. The numerical solution is compared to the exact solution for $\mathfrak{Re} = 10$ and $\eta_\rho = 10$.

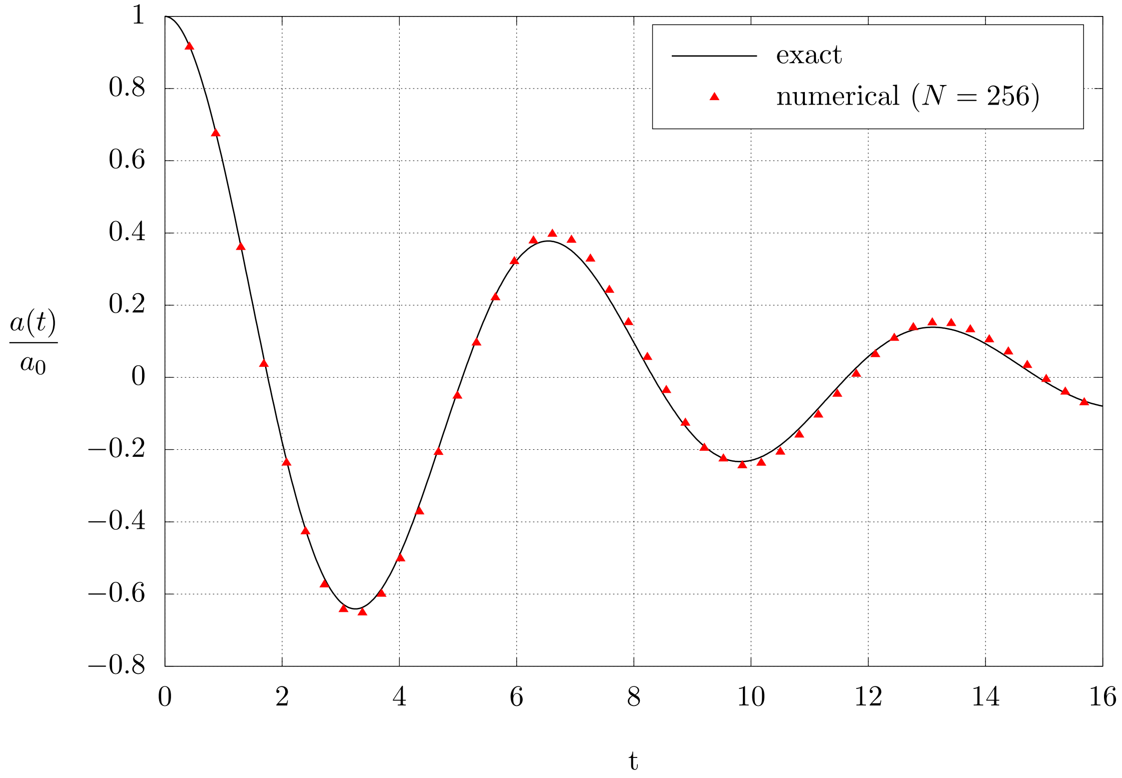


Figure 4.10 Temporal development of amplitude of the interface oscillation normalized by its initial value. The numerical solution is compared to the exact solution for $\mathfrak{Re} = 10$ and $\eta_\rho = 1000$.

Chapter 5

Results

The objective of this study was to compare the dispersion characteristics of fully resolved liquid droplets with that of fully resolved solid particles in decaying isotropic turbulence. The motivation for selecting decaying isotropic turbulence was that it is the simplest turbulent flow that can be studied both numerically and experimentally (e.g. in a wind tunnel) and thus the numerical results could be validated or reproduced by other researchers. The dimensionless dispersion along x_i, $\mathcal{D}_{x_i}$, is defined as the mean square of the displacement of the droplet/particle center of mass, $x_{c,i}$, in the x_i direction:

$$\mathcal{D}_{x_i}(t) = \frac{\varepsilon_0^2}{E_0^3} \left\langle (x_{c,i} - x_{c,i}^0)^2 \right\rangle \ . \tag{5.1}$$

In Eq. (5.1), $x_{c,i}^0$ is the position of the center of mass of a droplet/particle in the i-direction at the time of releasing it in the flow. The ensemble average $\left\langle (x_{c,i} - x_{c,i}^0)^2 \right\rangle$ is normalized by ε_0^2/E_0^3, i.e. by the square of the initial length-scale characterizing the large eddies. All the results presented in this chapter were obtained by performing *Direct Numerical Simulation* (DNS) in a cubical domain of size $L = L_x = L_y = L_z$ that was discretized with N grid nodes per direction, i.e. $N = N_x = N_y = N_z$. Periodic boundary conditions were prescribed along the three coordinate directions. The effect of gravity was not accounted for, i.e. $\mathfrak{Fr} = \infty$. Section 5.1 presents the details of the turbulent carrier flow. The properties of the dispersed phase are discussed in Section 5.2. Finally, the analysis and comparison of the dispersion characteristics of particles and droplets are reported in Section 5.3.

5.1 Turbulence Properties

5.1.1 Turbulence Generation

The initial velocity field was generated for a single phase flow by prescribing:

1. the dimensionless wave number of peak energy, k_p, at $t = 0$,
2. the dimensionless velocity mean correlations, r_{ij}, at $t = 0$, and
3. the Reynolds number $\mathfrak{Re}_{\lambda_0}$ based on the Taylor's micro-scale λ_0:

$$\mathfrak{Re}_{\lambda_0} = \frac{U_0 \lambda_0}{\nu} \ , \tag{5.2}$$

   where λ_0 is the dimensionless Taylor's length scale at $t = 0$, ν the dimensionless kinematic viscosity and U_0 is the value of the dimensionless root-mean-square velocity U_{rms} at $t = 0$.

Given r_{ij} and k_p, the *Turbulence Kinetic Energy* (TKE) spectrum, $E(k,t)$, and the velocity field at $t = 0$ were computed by using the method proposed by Gerz and Schumann [28] that ensures that the initial random velocity field is isotropic, divergence-free with respect to the discretized form of the continuity equation, and that the velocity cross-correlation spectra, $R_{ij}(k)$, satisfy the realizability constraints (Schumann [66]).

Once $E(k,t=0)$ is known, the definition

$$E(k,t=0) = \left(\frac{3U_0^2}{2}\right)\left(\frac{k}{2\pi k_p^2}\right)\exp\left(-\frac{k}{k_p}\right) \tag{5.3}$$

can be used to compute U_0:

$$U_0 = \sqrt{\frac{2}{3}\int_0^{N/2} E(k,0)dk}\ . \tag{5.4}$$

In Eq. (5.3) and (5.4) k is the dimensionless wave number that ranges from 0 to $N/2$, i.e. $k \in [0, N/2]$. Both k and k_p are normalized by the smallest non-zero wave number $k_{min} = 2\pi/L$. Therefore the maximum resolved wavenumber is $k_{max} = \pi N/L$. Finally, the dimensionless kinematic viscosity ν is obtained from:

$$\nu = \frac{\sqrt{15}\,U_0^2}{\mathfrak{Re}_{\lambda_0}}\left(8\pi^2\int_0^{N/2} k^2 E(k,0)dk\right)^{-\frac{1}{2}}\ . \tag{5.5}$$

The following relation links ν with the dynamic viscosity, $\widetilde{\mu}^-$, and density, $\widetilde{\rho}^-$, of the reference fluid[1]:

$$\frac{\widetilde{\mu}^-}{\widetilde{\rho}^-} = \nu\,\widetilde{U}\,\widetilde{L}\ , \tag{5.6}$$

where $\widetilde{L}$ is a chosen reference length and $\widetilde{U}$ the reference velocity obtained from Eq. (5.6) as:

$$\widetilde{U} = \frac{\widetilde{\mu}^-}{\widetilde{\rho}^-\nu\widetilde{L}}\ , \tag{5.7}$$

As a consequence, the Reynolds number $\mathfrak{Re}$ defined in Eq. (2.16) becomes:

$$\mathfrak{Re} = \frac{1}{\nu}\ . \tag{5.8}$$

[1] Clearly, in the case of single-phase flow, $\widetilde{\rho}^-$ and $\widetilde{\mu}^-$ are material properties of the only phase present.

This ensures that the scaling used in the initialization procedure is consistent with the one use for the *Navier-Stokes Equations* (NSE) (cf. Subsection 2.1.3). In this study, the turbulent velocity field was generated by setting $k_p = 4$, $\mathfrak{Re}_{\lambda_0} = 75$ and

$$r_{ij} = \begin{cases} 0 & \text{if} \quad i \neq j \\ 0.0025959 & \text{if} \quad i = j \end{cases} . \tag{5.9}$$

These input values result in $\nu = 2.39 \times 10^{-5}$. By choosing liquid water at STP as reference fluid ($\nu^{-} = 10^{-6}$ m^2/s) and $\widetilde{L} = 0.01$ m, Eq. (5.7) gives $\widetilde{U} = 4.18$ m/s. The computational domain was given a dimensionless size $L = 1$ and was discretized by using 512 grid points per direction, i.e. $N = 512$. This allows to fully resolve all the turbulence scales down to the smallest one for the case of interest ($\mathfrak{Re}_{\lambda_0} = 75$), according to the estimate given by Pope [56]. The parameters of the simulation of decaying isotropic turbulence are summarized in Table 5.1. The dimensionless turbulence properties at $t = 0$ are reported in Table 5.2: ℓ and τ_ℓ are the integral length- and time scales; η and τ_η are the Kolmogorov's length- and time scales; $\mathfrak{Re}_\ell$ is the Reynolds number based on ℓ. Finally, the initial TKE spectrum $E(k, 0)$ is shown in Figure 5.1.

Table 5.1 Parameters of the simulation of decaying isotropic turbulence.

Parameter	Value	Unit	Description
$\mathfrak{Re}_{\lambda_0}$	75	-	Reynolds number based on the Taylor's microscale at $t = 0$.
N	512	-	Number of grid points per directions.
ν	2.39×10^{-5}	-	Dimensionless kinematic viscosity of the carrier flow
$\widetilde{L}$	0.01	m	Reference length
$\widetilde{U}$	4.18	m/s	Reference velocity
k_p	4	-	Dimensionless wave number of peak energy
r_{ii}	2.59×10^{-3}	-	Dimensionless velocity mean correlations

5.1.2 Turbulence Evolution

DNS of single-phase decaying isotropic turbulence was performed in order to verify the accuracy of the computational code as well as to obtain data to compare to the case of two-phase turbulence. The dimensionless turbulence properties of interest are reported in Table 5.2 for $t = 1$, 5, 10 and 15. It should be noted that the turbulence scales are well resolved at all times since the condition $\eta k_{max} > 1$ is always satisfied. Figure 5.1 shows the evolution in time of the TKE spectrum: $E(k,t)$ decays in time as expected for a turbulent flow in the absence of any forcing. It should be emphasized that forcing isotropic turbulence, whether spectrally at the small wave numbers or in the physical space, is not appropriate to study the effect of finite-size particles/droplets on turbulence as discussed in [39], and thus it was not considered in this work.

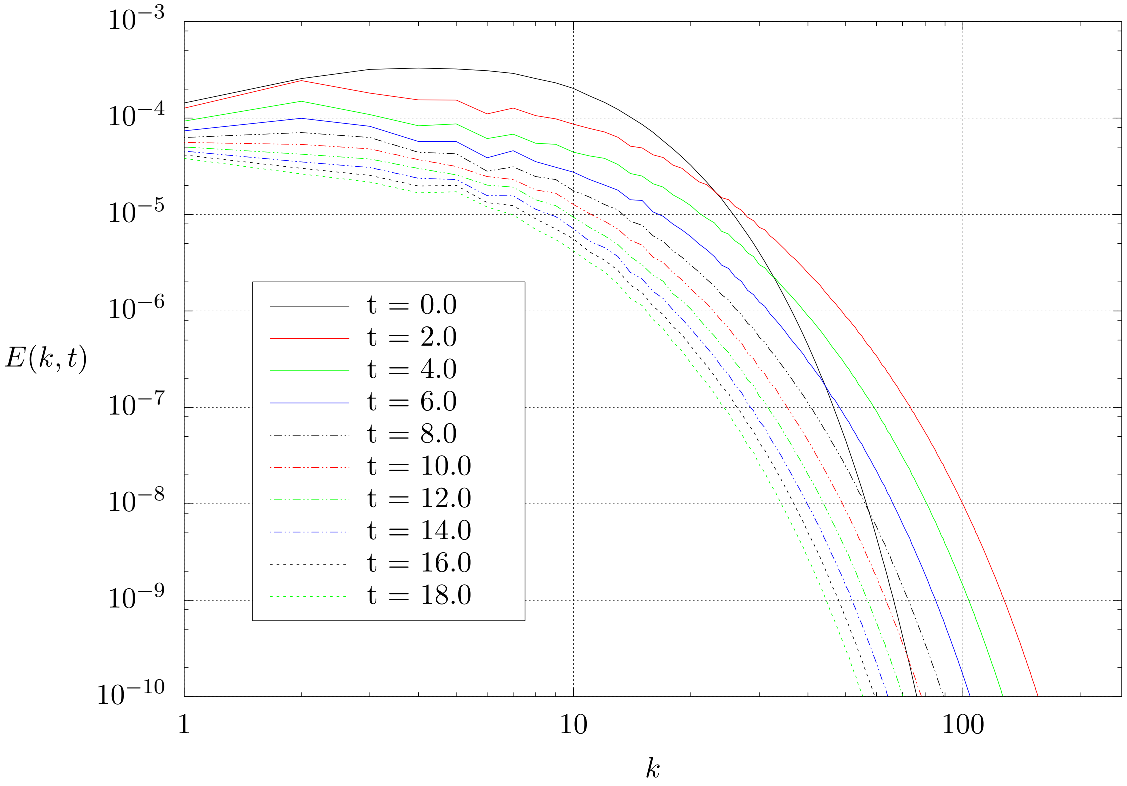


Figure 5.1 Temporal development of the energy spectrum $E(k,t)$ for decaying isotropic turbulence ($\mathfrak{Re}_{\lambda_0} = 75$, $N = 512$).

Table 5.2 Turbulence parameters (dimensionless) at initial time ($t = 0$), release time ($t = 1$), and only for the case of single-phase flow at time $t = 5$ and 15.

t	U_{rms}	ε	ℓ	λ	η	$\mathfrak{Re}_\ell$	$\mathfrak{Re}_\lambda$	ℓ/η	τ_ℓ	τ_λ	τ_η
0.0	0.0503	7.5×10^{-4}	0.0681	0.0342	0.00205	142	75	33.2	1.37	0.68	0.178
1.0	0.0436	8.8×10^{-4}	0.0736	0.0271	0.00195	135	50	37.7	1.67	0.59	0.159
5.0	0.0233	2.0×10^{-4}	0.0888	0.0321	0.00286	90	33	31.0	3.66	1.31	0.342
15.0	0.0142	2.0×10^{-5}	0.1234	0.0513	0.00510	63	26	24.2	10.2	4.54	1.089

Figure 5.2 shows the temporal development of the TKE normalized by initial value, $E(t)/E(0)$. For single-phase turbulence, the decay of TKE is due to the effect of the viscous dissipation:

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\varepsilon(t) \ , \tag{5.10}$$

where the viscous dissipation rate of TKE, $\varepsilon(t)$, is given by:

$$\varepsilon(t) = 2\nu\langle s_{ij}s_{ij}\rangle \ . \tag{5.11}$$

In Eq. (5.11), $\langle\alpha\rangle$ denotes the ensemble average of a generic variable α throughout the computational domain, whereas s_{ij} is the *strain-rate tensor* defined as:

$$s_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \ . \tag{5.12}$$

The temporal development of the viscous dissipation rate is shown in Figure 5.3: $\varepsilon(t)$ peaks at $t = 1$ and then decays monotonically. For $t < 1$, the turbulence develops until an established nonlinear transfer of turbulence kinetic energy across the spectrum is reached. This can be seen in Figure 5.4 that shows that the skewness of the velocity derivatives S_u, defined as

$$S_u = -\left\langle\frac{\partial u_i^3}{\partial x_i}\right\rangle \Bigg/ \left\langle\frac{\partial u_i^2}{\partial x_i}\right\rangle^{1.5} \ , \tag{5.13}$$

reaches the value -0.5 at $t = 1$.

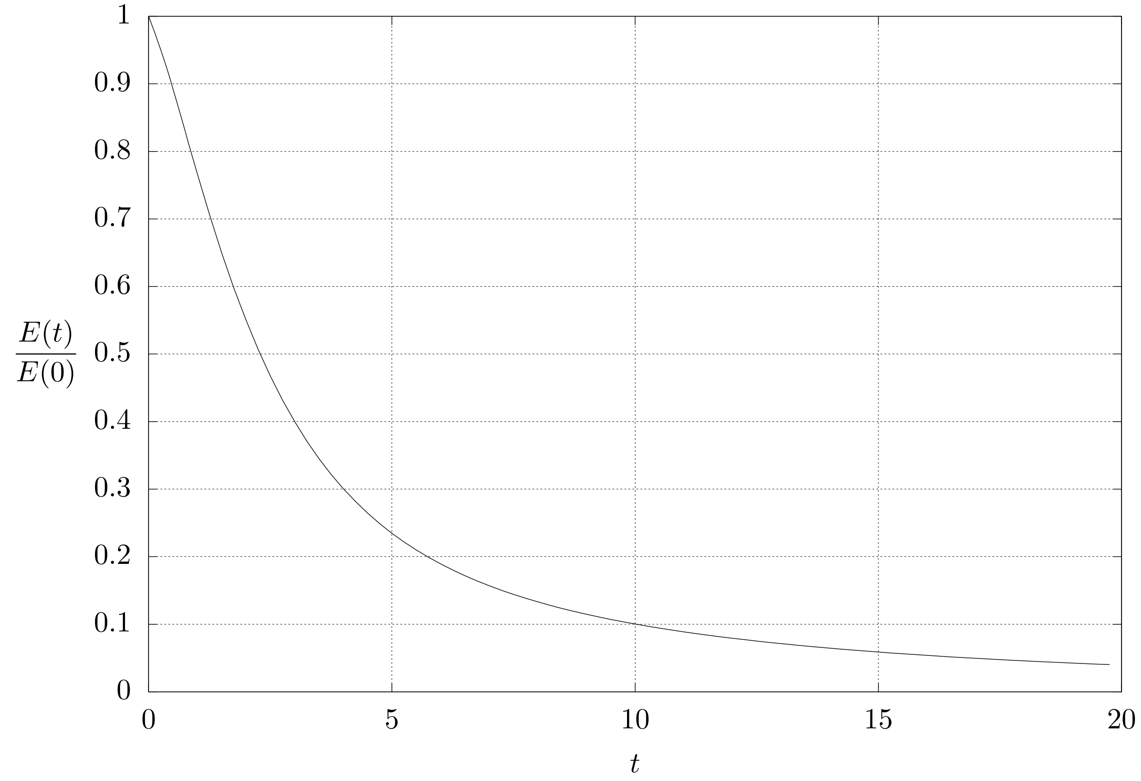


Figure 5.2 Temporal development of TKE normalized by its initial value for single-phase turbulence ($\mathfrak{Re}_{\lambda_0} = 75$, $N = 512$).

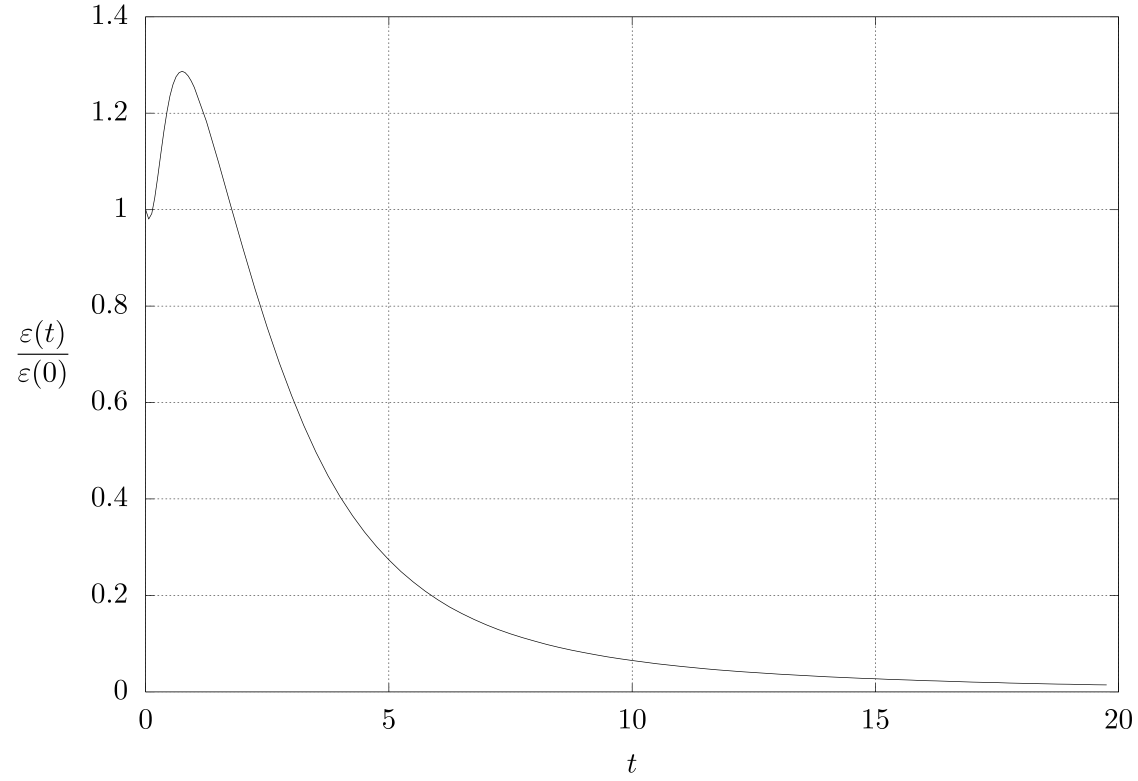


Figure 5.3 Temporal development of the viscous dissipation rate of turbulence kinetic energy normalized by its initial value for single-phase turbulence ($\mathfrak{Re}_{\lambda_0} = 75$, $N = 512$).

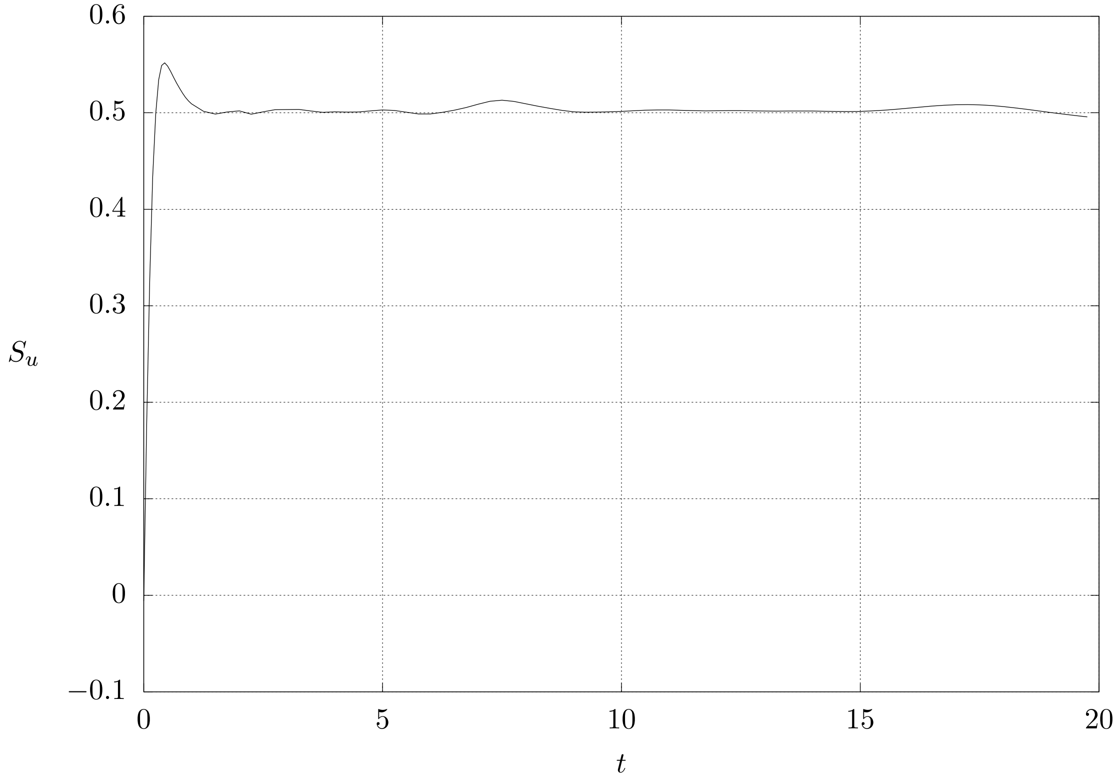


Figure 5.4 Temporal development of the skewness, S_u, of the velocity derivatives in the DNS of single-phase decaying isotropic turbulence ($\mathfrak{Re}_{\lambda_0} = 75$, $N = 512$).

5.2 Dispersed Phase Properties

Two different types of dispersed phase were considered *separately*:

1. solid spherical particles, modelled via the *Immersed Boundary Method* (IBM), and
2. liquid droplets, initially spherical, modelled via the *Accurate Conservative Level Set Method* (ACLSM).

The dispersed phase was released into the turbulent flow at $t = 1$, after the turbulence had reached a well developed stage, as explained in Section 5.1. At release time, the velocity inside the droplets/particles was set to a uniform value equal to the interpolated velocity at the center of mass. Initially, both particles and droplets have the same diameter, $\widetilde{D} = 310$ μm. The reference length, $\widetilde{L} = 0.01$ m, was chosen to guarantee a resolution of about 16 grid points per diameter, while maintaining a dimensionless domain size of 1. The dimensionless diameter, $D = 0.031$, is comparable to the initial Taylor's length-scale and about 16 times the initial Kolmogorov's length-scale. The density ratio of the dispersed phase to the carrier fluid, η_ρ, is 10. While the ACLSM can handle much larger density ratio, the IBM cannot and therefore $\eta_\rho = 10$ is the only case studied in this work. In addition to the aforementioned parameters, the simulation of liquid droplets requires the ratio of the molecular viscosity of the dispersed phase to the carrier flow, η_μ, and the Weber's number, $\mathfrak{We}$. In this study, $\eta_\mu = 0.1$ and $\mathfrak{We} = 159.6$. Since the goal of this work was to compare the dispersion of solid particles with that of liquid droplets with the same density ratio to the carrier flow, the values of η_μ and $\mathfrak{We}$ were selected to allow for a relatively large time step in order to speed up the simulations. Therefore the dispersed phase is a fictitious fluid. The physical and numerical parameters used in the simulation of particle-laden and droplet-laden turbulence are summarized in Table 5.3.

Table 5.3 Physical and numerical parameters in the simulation of particle-laden and droplets-laden turbulence.

Parameter	Value	Unit	Description
$\widetilde{D}$	310	μm	Droplet/particle diameter
D/η_0	15.86	-	Diameter/initial Kolmogorov's length scale
D/λ_0	1.15	-	Diameter/initial Taylor's length scale
η_ρ	10.8	-	Density ratio of dispersed phase to carrier fluid
η_μ	0.1	-	Molecular viscosity ratio of droplets to carrier fluid
$\mathfrak{Re}_{\lambda_0}$	75	-	Reynolds number based on the initial Taylor's length scale
$\mathfrak{We}$	159.6	-	Weber number of the droplet

5.3 Dispersion Characteristics

Figure 5.5 shows that the dispersion of the solid spherical particles along x is reduced *monotonically* as the number of particles, N_p, increases from 640 to 3200 to 6400. This corresponds to an increase of volume fraction, Φ_v, from 0.01 to 0.05 to 0.1. This result was already observed by Lucci et al. [39], who showed that the increase of Φ_v reduces the TKE, $E(t)$, due to the two-way coupling effects. In contrast, it is seen in Figure 5.6 that increasing the number of liquid droplets, N_d, from 100 to 500 to 1000 increases their dispersion. A comparison between the dispersion of solid particles and liquid droplets in the x direction for a volume fraction of $\Phi_v = 0.016$ is shown in Figure 5.7: the liquid droplets disperse more than an equal number of solid particles. The same effect is observed along y and z, as shown in Figure 5.8. In order to explain the opposite behavior of the dispersion of droplets, it is necessary to examine the effects of the liquid droplets on the TKE decay rate of the carrier fluid. For a two-phase flow, the decay of TKE is due to the combined effects of viscous dissipation and two/four-way coupling:

$$\frac{\mathrm{d}E(t)}{\mathrm{d}t} = -\varepsilon(t) + \Psi \ . \tag{5.14}$$

In the above equation, Ψ is the rate of change of TKE due to the forces acting at the interface separating the carrier flow and the dispersed phase. Thus, Ψ represents the two-way coupling effects in the particle-laden turbulence case, the four-way coupling effects in the droplet-laden turbulence case, and it is zero for single-phase turbulence (cf. Eq. (5.10)). It should be noted that the definition of $\varepsilon(t)$ given by Eq. (5.11) still applies to the two-phase case, as long as the ensemble average $\langle s_{ij}s_{ij}\rangle$ is taken over the sub-domain occupied by the carrier fluid only. By definition, $\varepsilon(t)$ is always a sink of TKE, while Ψ can act as either a source or a sink of TKE as shown by Ferrante and Elghobashi [25].

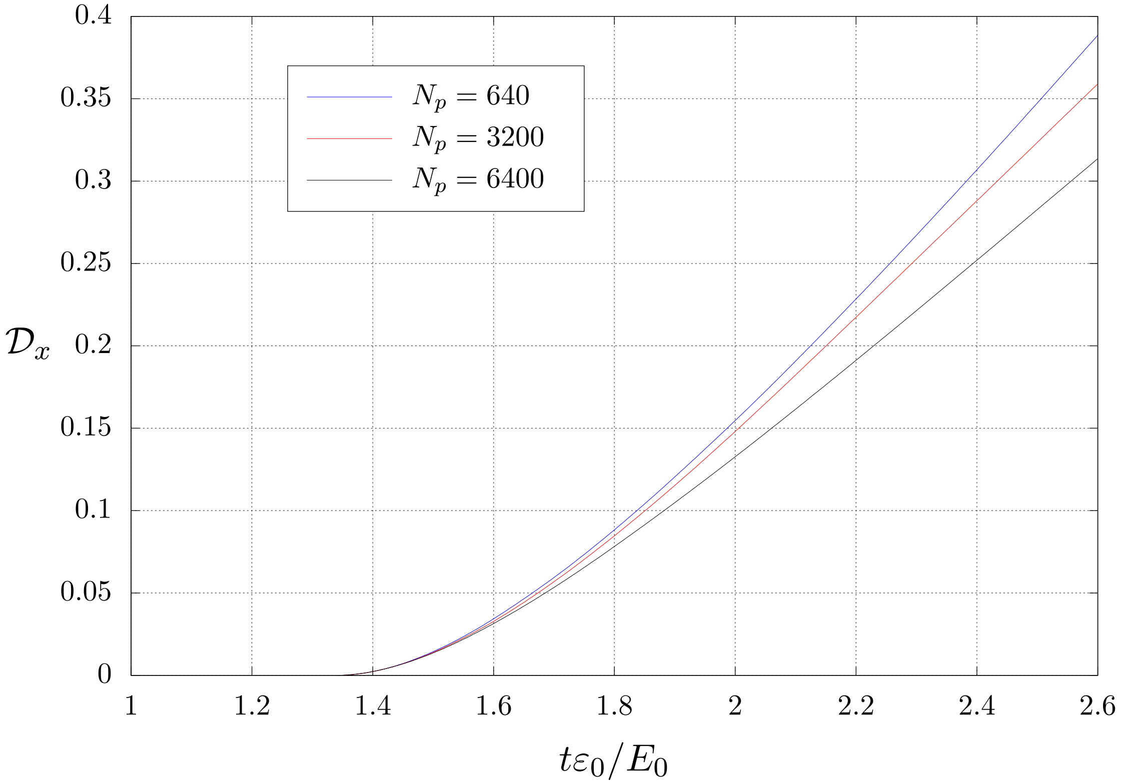


Figure 5.5 Dispersion in the x-direction of solid spherical particles ranging in number from 640 to 6400 in isotropic turbulence.

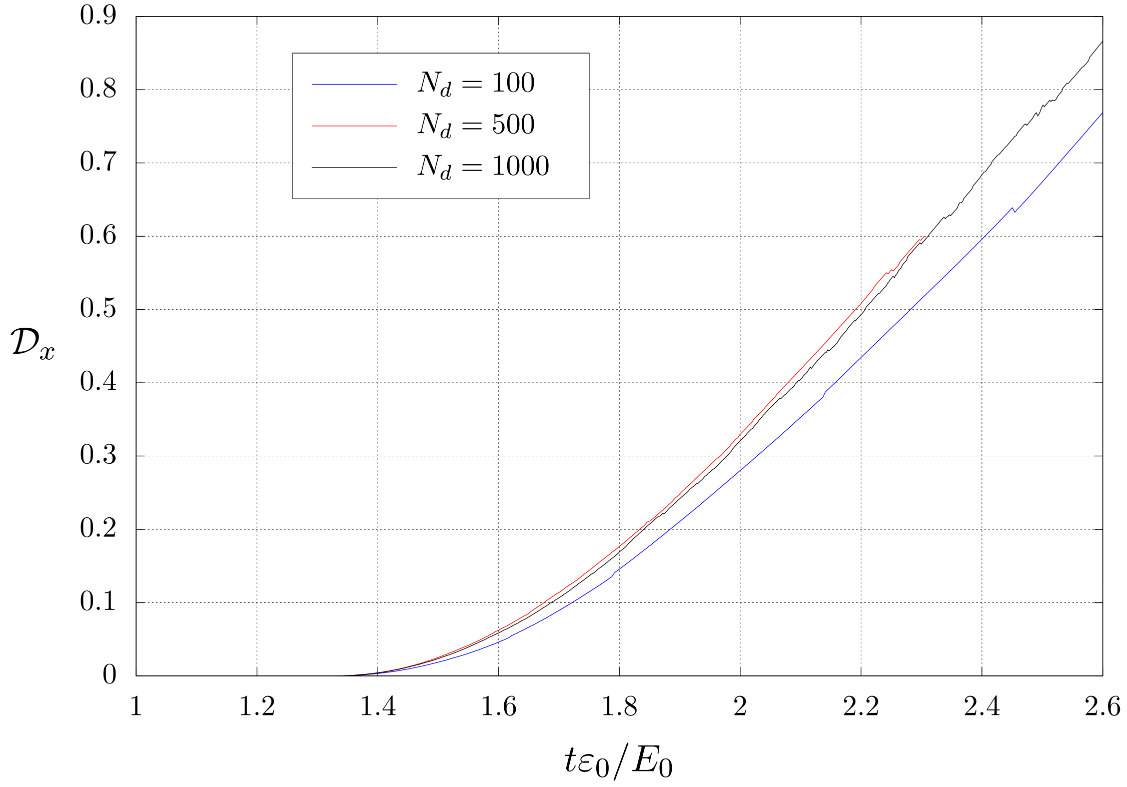


Figure 5.6 Dispersion of liquid droplets ranging in number from 100 to 1000 in isotropic turbulence.

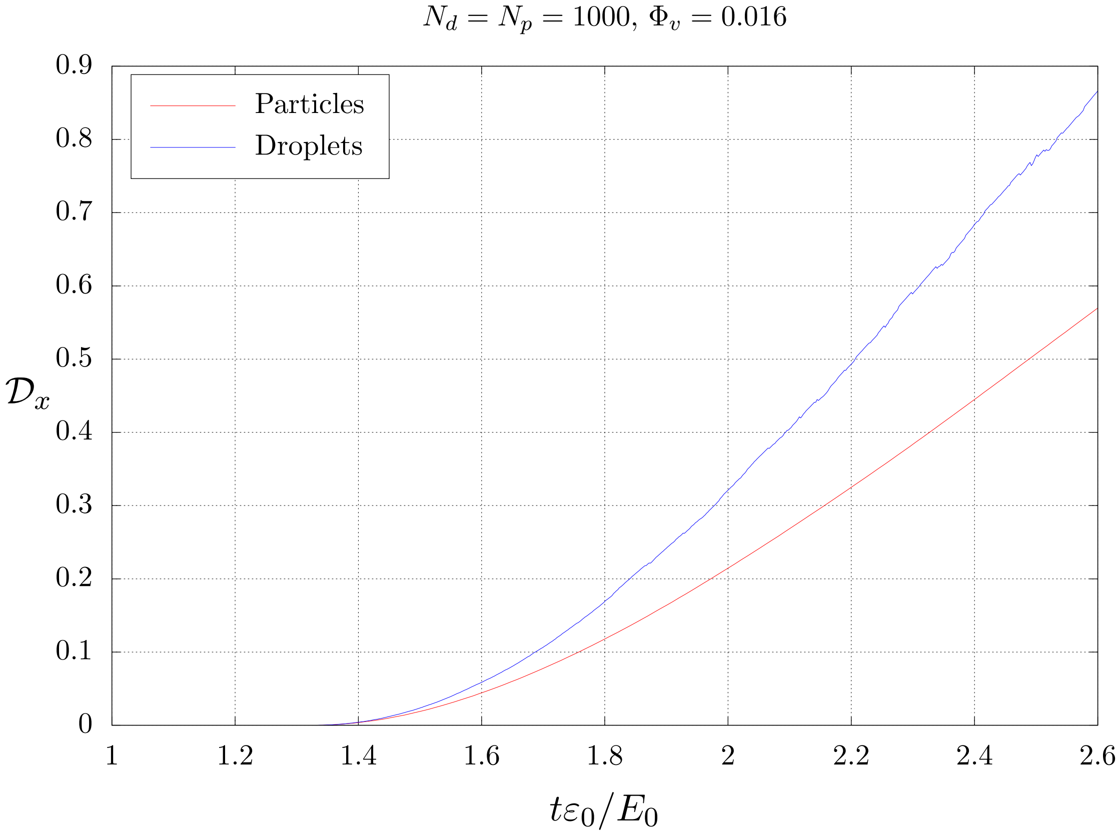


Figure 5.7 Dispersion of liquid droplets and solid particles in the x-direction in isotropic turbulence.

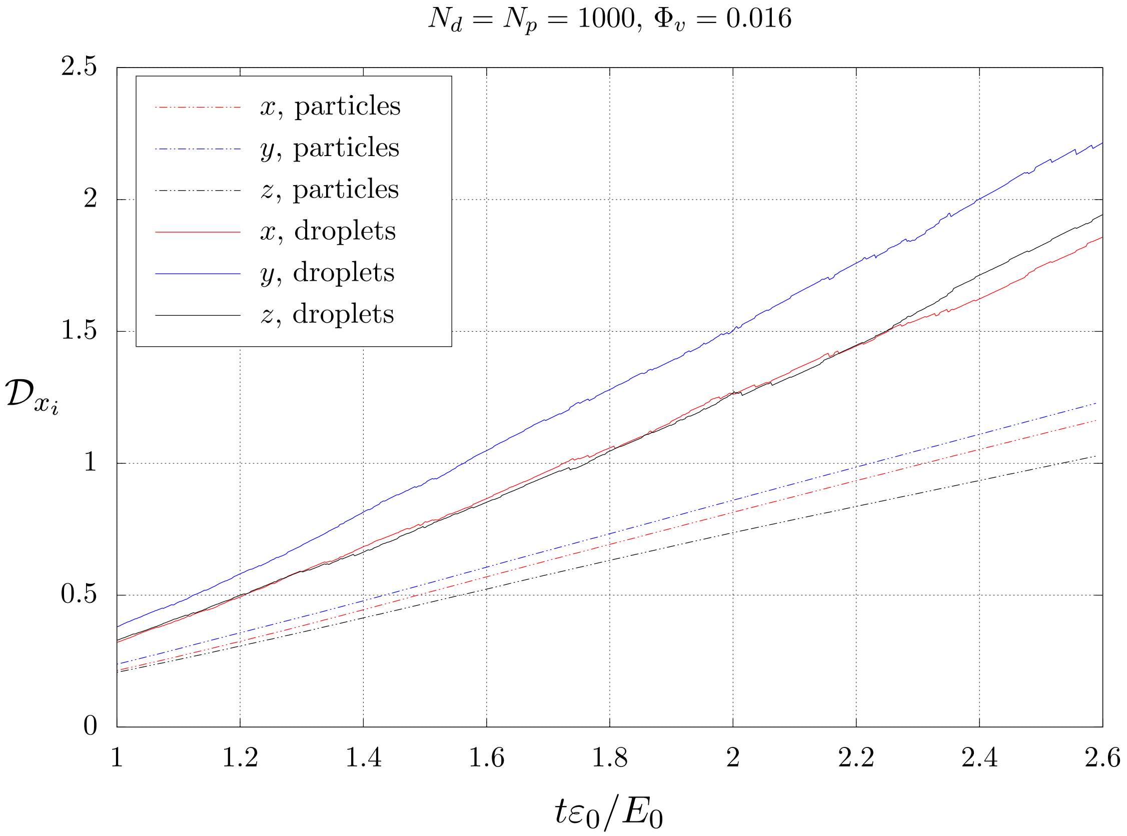


Figure 5.8 Dispersion of liquid droplets and solid particles in the three coordinate directions in isotropic turbulence.

5.3.1 Effect on the TKE Decay Rate

Figure 5.9 shows the temporal development of the TKE normalized by its value at $t = 0$ for N_d ranging from 0 (single-phase) to 1000. The main finding is that finite-size droplets, under the conditions listed in Table 5.3, *reduce* the decay rate of TKE compared to that of the single-phase (no droplets) turbulence. The reduction increases as the volume fraction of the droplet is increased. This is an unexpected result since the research over the past two decades on the effects of solid particles on turbulence is that solid particles *enhance* the decay rate of TKE of isotropic turbulence [22, 25, 39]. Figure 5.10 compares the temporal decay of TKE for the three cases of single-phase, particle-laden and droplet-laden flows. A direct consequence of the reduced decay rate of TKE is the enhanced dispersion as compared to the dispersion of solid particles of the same diameter and same density ratio (of solid to fluid) as shown in Fig 5.8.

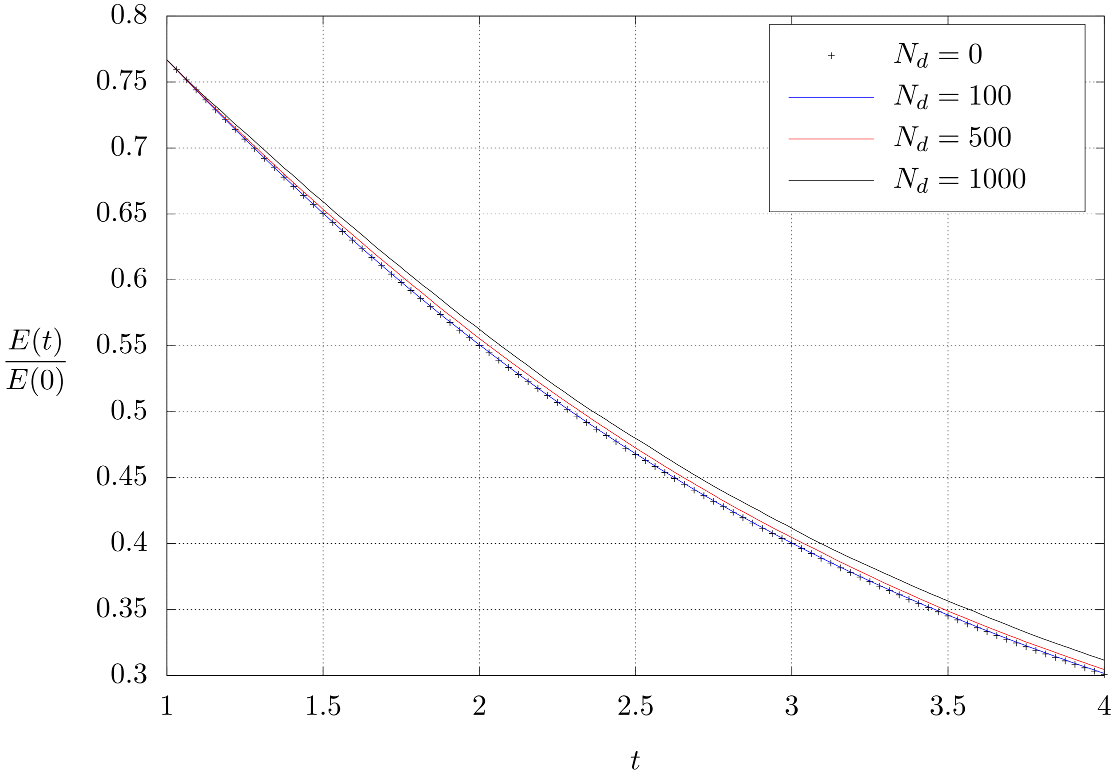


Figure 5.9 Temporal decay of kinetic energy of isotropic turbulence for four cases: no droplets, 100, 500 and 1000 droplets.

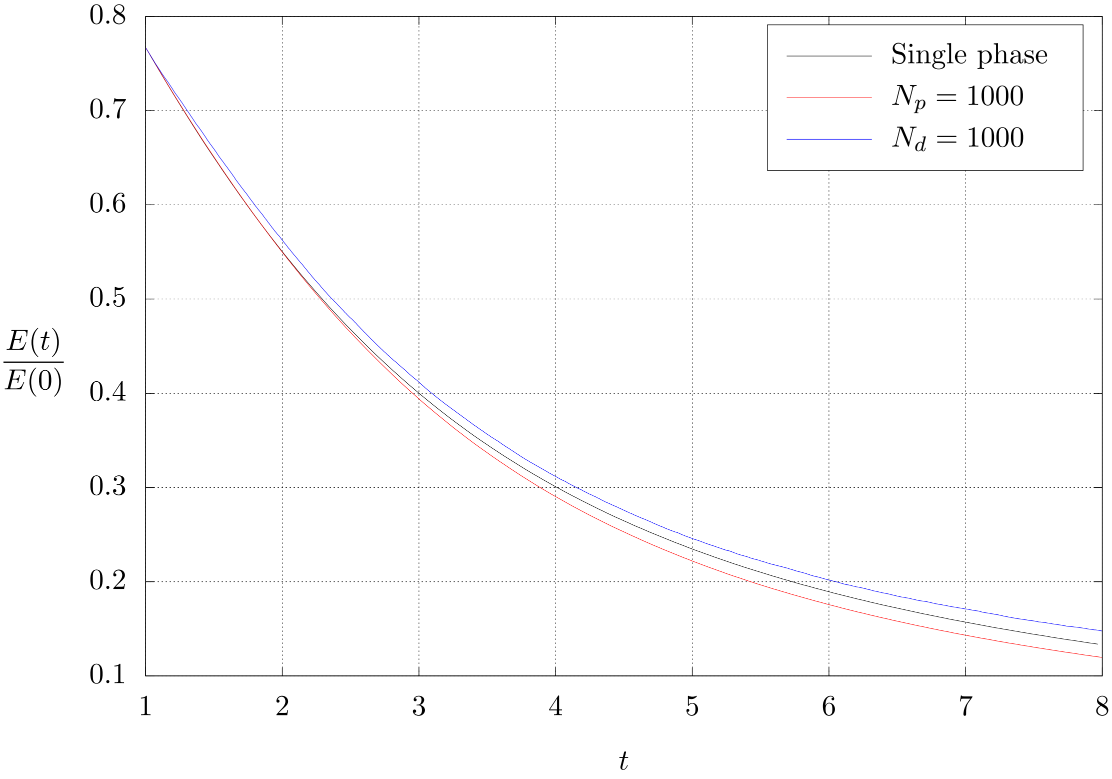


Figure 5.10 Temporal decay of TKE for single-phase, particle-laden, and droplet-laden isotropic turbulence

5.3.2 Effect on the TKE Viscous Dissipation Rate

Figure 5.11 compares the temporal distribution of the dissipation rate of TKE for single-phase, particle-laden and droplet-laden isotropic turbulence. $\varepsilon(t)$ immediately peaks higher than that of single-phase turbulence as soon as the dispersed phase is released into the flow at $t = 1$. As noted by Lucci et al. [39], the presence of solid particles in the turbulent flow increases the velocity gradients close to the particles surface through the *no-slip* condition. As a consequence, the local strain rate s_{ij} increases in proximity of the particles, resulting in a larger viscous dissipation rate compared to the single-phase case. For droplet-laden turbulence, the increase in $\varepsilon(t)$ is much larger compared to the particle-laden case. The liquid droplets do not maintain their initial spherical shape, particularly when they are in the process of merging/breaking-up and/or wobbling. Consequently, the local strain-rate around a deformed droplet is much larger than that around a spherical particle. A typical non-spherical droplet is shown in Figure 5.12 together with the fluid velocity vectors. Figure 5.13 displays the instantaneous contours of the TKE dissipation rate: $\varepsilon(t)$ is largest (red contours) at the surface of the droplets, especially when the droplet is no longer spherical and/or is merging with another droplet. The contours of $\varepsilon(t)$ around wobbling and merging droplets are shown in Figure 5.14 and 5.15 respectively.

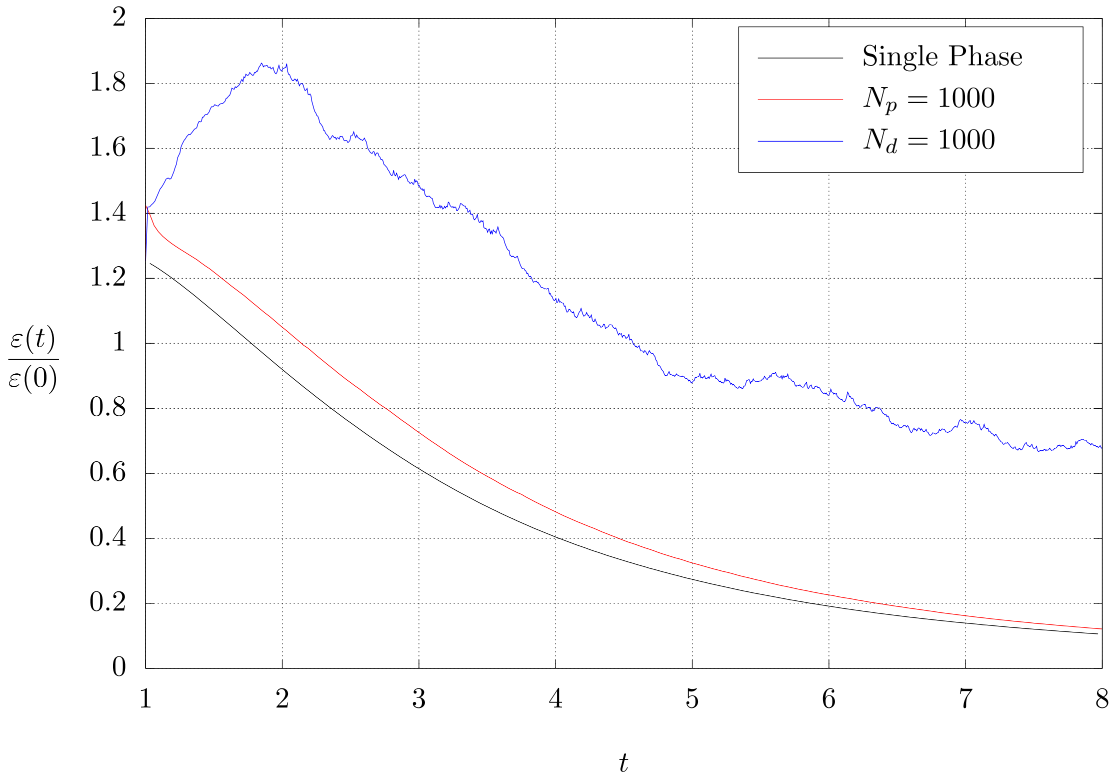


Figure 5.11 Temporal distribution of the dissipation rate of TKE for single-phase, particle-laden and droplet-laden isotropic turbulence.

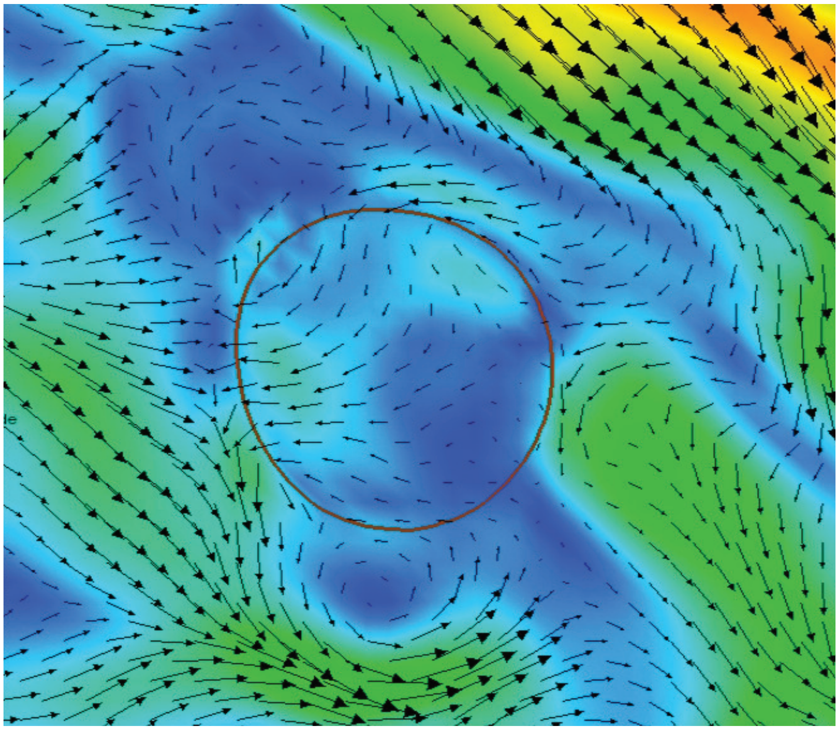


Figure 5.12 Velocity vectors outside and inside a single non-vaporizing droplet in isotropic turbulence. The vectors are projected on the middle plane of a three-dimensional domain. Contours of velocity component perpendicular to the plane are: blue(-), green , yellow (+).

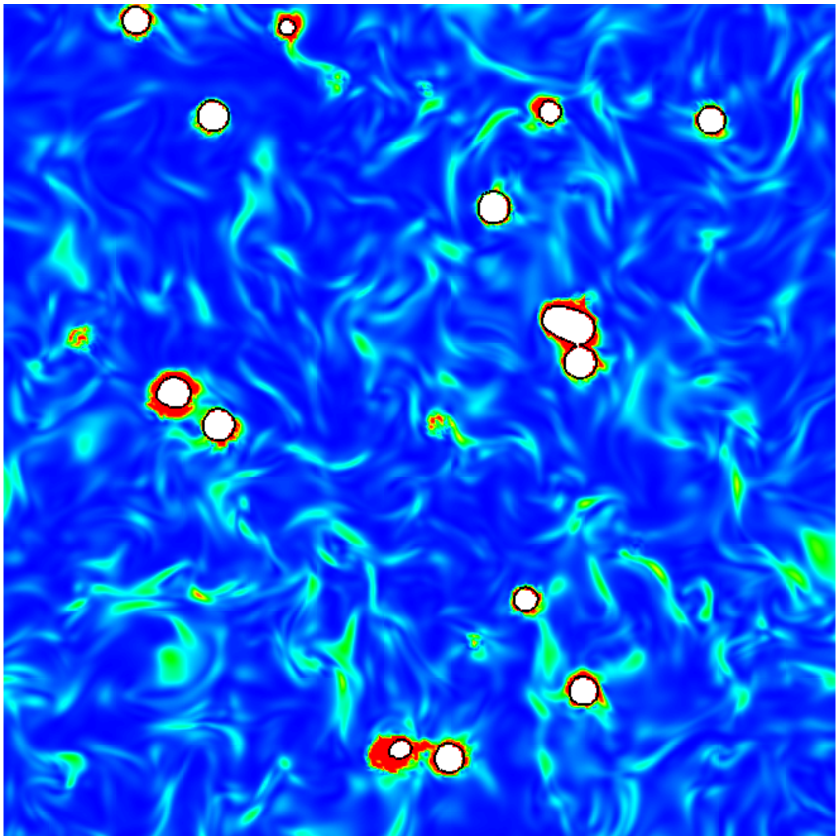
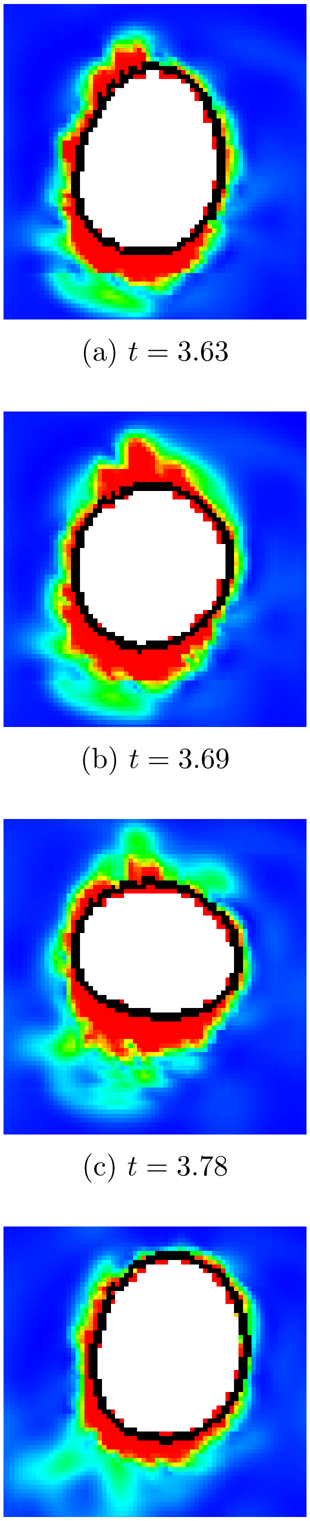


Figure 5.13 Contours of the dissipation rate of turbulence kinetic energy around non-vaporizing droplets in isotropic turbulence at $t = 2.59$. Red and blue contours indicate maximum and minimum dissipation rates, respectively.



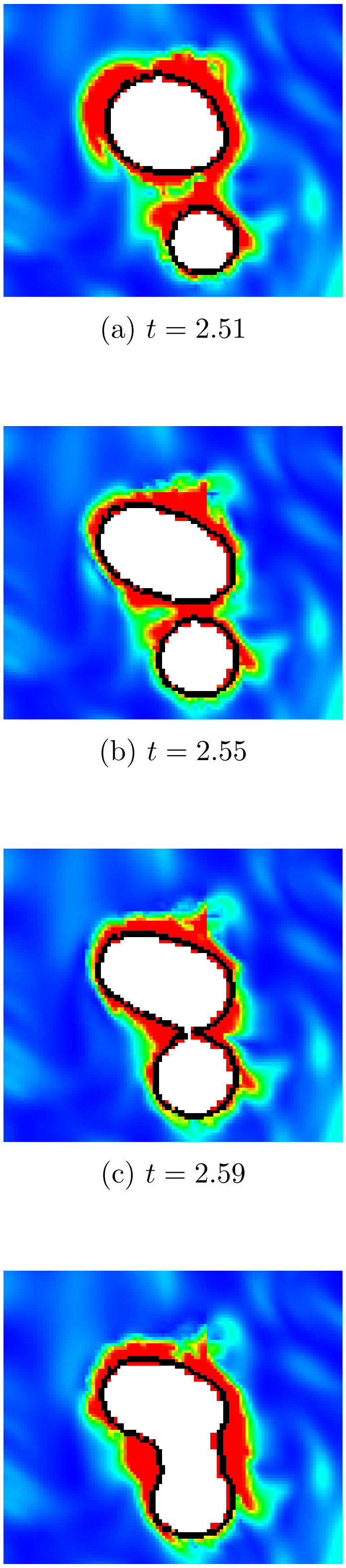
(a) $t = 3.63$

(b) $t = 3.69$

(c) $t = 3.78$

(d) $t = 3.95$

Figure 5.14 Contours of the dissipation rate of turbulence kinetic energy around non-vaporizing droplets in the process of wobbling at four different times. Red and blue contours indicate maximum and minimum dissipation rates, respectively.



(a) $t = 2.51$

(b) $t = 2.55$

(c) $t = 2.59$

(d) $t = 2.61$

Figure 5.15 Contours of the dissipation rate of turbulence kinetic energy around non-vaporizing droplets in the process of merging at four different times. Red and blue contours indicate maximum and minimum dissipation rates, respectively.

5.3.3 Coupling Effects

Lucci et al. [39] showed that the coupling term Ψ in Eq. (5.14) is always negative for particle-laden turbulence (two-way coupling). This means that dispersed solid particles enhance the decay rate of TKE as shown in Figure 5.10. In contrast, Figure 5.10 and 5.11 show that dispersed liquid droplets reduce the decay rate of TKE despite increasing the viscous dissipation rate $\varepsilon(t)$. Therefore, Ψ in Eq. (5.14) must be positive, i.e. act like a source of TKE, when four-way coupling effects are in place. In order to explain the positive sign of Ψ, it is useful to consider Figure 5.13, 5.15, and 5.14 that show that the droplets are no longer spherical, and thus they rotate more than a spherical object due to the surrounding velocity gradients. Consequently, the spinning droplets transfer their kinetic energy to the surrounding fluid. As an indication of the increased rotation of the droplets, Figure 5.16 shows the temporal variation of the probability density function (PDF) of the tangential velocity magnitude at the surfaces of the 1000 droplets. It is seen that the magnitude of the tangential velocity varies by two orders of magnitude, with peak velocities about one order of magnitude larger than the mean. These large tangential velocities correspond to the merging or the non-spherical droplets.

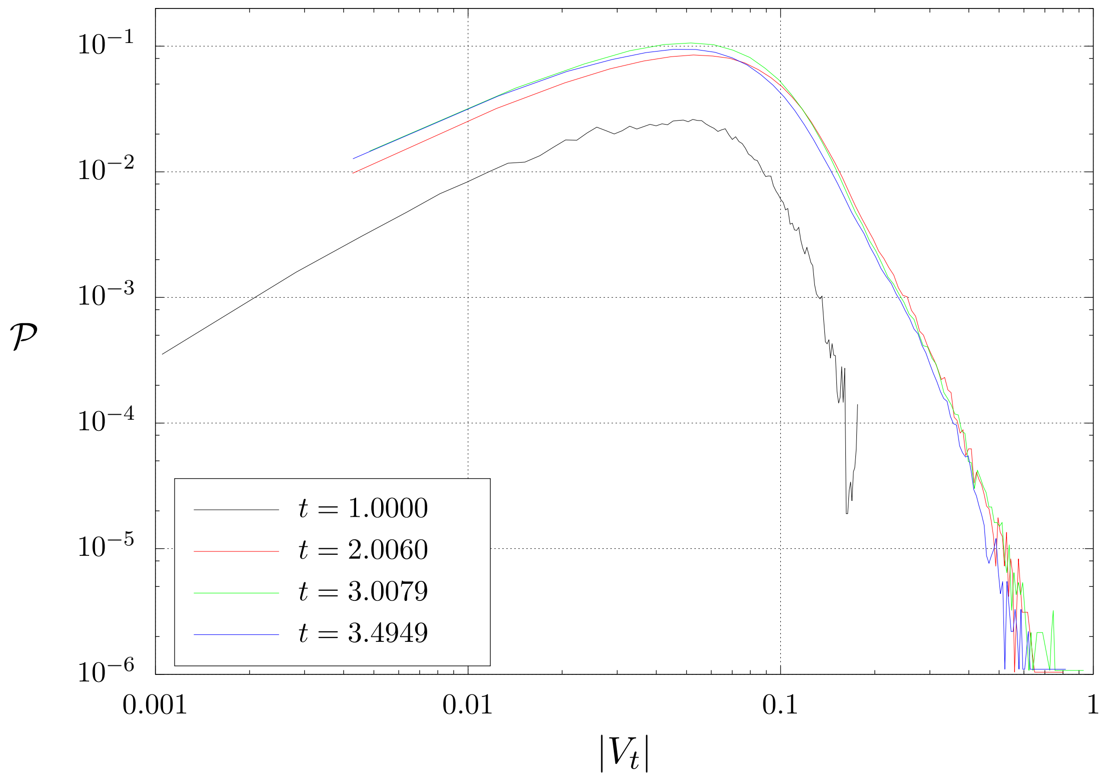


Figure 5.16 Instantaneous PDF (in log-log coordinates) of the tangential velocity of 1000 droplets in isotropic turbulence at four different times.

Chapter 6

Conclusions

The dispersion characteristics of deformable liquid droplets and solid particles in decaying isotropic turbulence and zero gravity were investigated by using *Direct Numerical Simulation* (DNS). The liquid droplets were allowed to freely move, deform and interact with each other (four-way coupling), while the solid particles were not allowed to collide (two-way coupling). For this study, a computational code for the solution of two-phase flow, either fluid/fluid or solid/fluid type, was developed. The code uses the *Accurate Conservative Level Set Method* (ACLSM) to implicitly track the motion and deformation of the liquid droplets surface and the *Immersed Boundary Method* (IBM) to explicitly track the solid particles in a Lagrangian fashion. The effect of the surface tension and the discontinuous material properties at the droplets interface were accounted for via the *Ghost Fluid Method* (GFM) and the *Continuum Surface Force* (CSF) approach, respectively. The unsteady three-dimensional Navier-Stokes and continuity equations were solved throughout the whole computational domain, including the interior of the dispersed phase, via a projection method. It should be noted that the liquid droplets and solid particles were fully resolved in 3D-space and time and all the scales of the turbulent motion were simultaneously resolved down to the smallest relevant length- and time-scales. The simulations were performed on grid with 512 grid points per direction and with an initial Taylor's microscale Reynolds number $\mathfrak{Re}_{\lambda_0} = 75$. The liquid droplets and the solid particles had the same diameter that was comparable to the initial Taylor's microscale of the turbulence. The density ratio of the dispersed phase to the carrier flow was set to 10, while the Weber's number and the ratio of the molecular viscosity for the case of droplet-laden turbulence were set to 159 and 0.1 respectively. The results presented in this work show that in isotropic turbulence and zero gravity the dispersion of liquid droplets in a given direction is larger than that of solid particles. It is seen that the presence of the solid particles enhances the turbulence kinetic energy decay rate as a result of the two-way coupling effects. Consequently, the dispersion of solid particles is reduced as their volume fraction is increased. In contrast, dispersed liquid droplets reduce the turbulence kinetic energy decay rate due to the four-way coupling effects. In particular, it is shown that the deformed droplets rotate more than the spherical solid particles, and thus the rotational kinetic energy is transferred to the surrounding flow. This phenomenon is responsible for the reduction of the turbulence kinetic energy decay rate for droplet-laden turbulence. Of course, the results presented in this work were obtained under the set of physical conditions listed in Chapter 5.

Future developments

The present study is a first step in understanding the interaction of a dispersed liquid phase with turbulence. Many aspects of this interaction were not investigated in this work and need to be researched in future studies. First and foremost, it is important to determine the range of conditions under which the results presented here apply. In particular, a parametric study by varying the droplets density ratio, viscosity ratio and Weber's number is recommended. Also, a deeper analysis on the effect of the droplets deformation on the surrounding flow field topology would be of great interest. The interactions between two or more droplets need to be more thoroughly analyzed. This may prove particularly challenging, since the accurate resolution of merging and break-ups will likely require the development and implementation of numerical schemes able to handle such interactions at a sub-grid level. Finally, it would be of great scientific and practical importance to include the effect of vaporization (fuel spray in combustion engines) and dispersions of liquid droplets (clouds) .

Bibliography

[1] D. Adalsteinsson and J. A. Sethian. The Fast Construction of Extension Velocities in Level Set Methods. *Journal of Computational Physics*, 148:2–22, 1997.

[2] P. Bagchi and S. Balachandar. Effect of turbulence on the drag and lift of a particle. *Physics of Fluids (1994-present)*, 15(11):3496–3513, 2003.

[3] P. Bagchi and S. Balachandar. Response of the wake of an isolated particle to an isotropic turbulent flow. *Journal of Fluid Mechanics*, 518:95–123, 2004.

[4] S. Balachandar and J. K. Eaton. Turbulent Dispersed Multiphase Flow. *Annual Review of Fluid Mechanics*, 42(1):111–133, 2010.

[5] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[6] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016. URL `http://www.mcs.anl.gov/petsc`.

[7] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Web page. `http://www.mcs.anl.gov/petsc`, 2016. URL `http://www.mcs.anl.gov/petsc`.

[8] G. K. Batchelor. Diffusion in a field of homogeneous turbulence. *Mathematical Proceedings of the Cambridge Philosophical Society*, 48(02):345–362, 1952.

[9] E. Bodenschatz, S. P. Malinowski, R. A. Shaw, and F. Stratmann. Can We Understand Clouds Without Turbulence? *Science*, 327(5968):970–971, 2010.

[10] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992.

[11] W. L. Briggs. A Multigrid Tutorial.

[12] T. M. Burton and J. K. Eaton. Fully resolved simulations of particle-turbulence interaction. *Journal of Fluid Mechanics*, 545:67, 2005.

[13] A. T. Cate, J. J. Derksen, L. M. Portela, and H. E. A. Van Den Akker. Fully resolved simulations of colliding monodisperse spheres in forced isotropic turbulence. *Journal of Fluid Mechanics*, 519:233–271, 2004.

[14] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *Journal of computational Physics*, 124(2):449–464, 1996.

[15] D. Chopp. Some Improvements of the Fast Marching Method. *SIAM Journal on Scientific Computing*, 23(1):230–244, 2001.

[16] A. J. Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967.

[17] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22(104):745–762, 1968.

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

[19] O. A. Druzhinin and S. Elghobashi. Direct numerical simulations of bubble-laden turbulent flows using the two-fluid formulation. *Physics of Fluids (1994-present)*, 10(3): 685–697, 1998.

[20] S. Elghobashi. On predicting particle-laden turbulent flows. *Applied Scientific Research*, 52(4):309–329, 1993.

[21] S. Elghobashi and G. C. Truesdell. Direct simulation of particle dispersion in a decaying isotropic turbulence. *Journal of Fluid Mechanics*, 242:655–700, 1992.

[22] S. Elghobashi and G. C. Truesdell. On the two-way interaction between homogeneous turbulence and dispersed solid particles. I: Turbulence modification. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 5(7):1790–1801, 1993.

[23] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A Hybrid Particle Level Set Method for Improved Interface Capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[24] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999.

[25] A. Ferrante and S. Elghobashi. On the physical mechanisms of two-way coupling in particle-laden isotropic turbulence. *Physics of Fluids (1994-present)*, 15(2):315–329, 2003.

[26] P. Février, O. Simonin, and K. D. Squires. Partitioning of particle velocities in gas–solid turbulent flows into a continuous field and a spatially uncorrelated random distribution: Theoretical formalism and numerical study. *Journal of Fluid Mechanics*, 533:1–46, 2005.

[27] M. Frigo and S. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[28] T. Gerz and U. Schumann. Influence of initial conditions on the development of stratified homogeneous turbulent shear flow. In *Finite Approximations in Fluid Mechanics II: DFG Priority Research Programme Results 1986-1988*, pages 142–156. 1989.

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

[30] M. Herbert. *Cell-Centered Multigrid with Higher-Order Transfer Operators in waL-Berla*. Bachelor thesis, FRIEDRICH-ALEXANDER-UNIVERSITÄT, ERLANGEN-NÜRNBERG, 2013.

[31] M. Kang, R. P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000.

[32] M. Khalil and P. Wesseling. Vertex-centered and cell-centered multigrid for interface problems. *Journal of Computational Physics*, 98(1):1–10, 1992.

[33] I. Kim, S. Elghobashi, and W. A. Sirignano. On the equation for spherical-particle motion: Effect of Reynolds and acceleration numbers. *Journal of Fluid Mechanics*, 367 (1):221–253, 1998.

[34] L. Landau and E. Lifshitz. *Fluid Mechanics, Second Edition: Volume 6 (Course of Theoretical Physics)*, volume 6. Butterworth-Heinemann, 2 edition, 1987.

[35] N. Li and S. Laizet. 2DECOMP&FFT - A Highly Scalable 2D Decomposition Library and FFT Interface. In *Cray User Group 2010 Conference*, Edinburgh, 2010.

[36] X.-D. Liu, R. P. Fedkiw, and M. Kang. A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains. *Journal of Computational Physics*, 160(1): 151–178, 2000.

[37] J. Lu and G. Tryggvason. Numerical study of turbulent bubbly downflows in a vertical channel. *Physics of Fluids (1994-present)*, 18(10):103302, 2006.

[38] F. Lucci. *Taylor Length-Scale Size Particles in Isotropic Turbulence*. Ph.d., University of California, Irvine, United States – California, 2011.

[39] F. Lucci, A. Ferrante, and S. Elghobashi. Modulation of isotropic turbulence by particles of Taylor length-scale size. *Journal of Fluid Mechanics*, 650:5, 2010.

[40] J. L. Lumley. Two-phase and non-Newtonian flows. In *Turbulence*, page 289. 1978.

[41] E. Marchandise, P. Geuzaine, N. Chevaugeon, and J.-F. Remacle. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *Journal of Computational Physics*, 225(1):949–974, 2007.

[42] M. R. Maxey and J. J. Riley. Equation of motion for a small rigid sphere in a nonuniform flow. *Physics of Fluids (1958-1988)*, 26(4):883–889, 1983.

[43] D. A. May, P. Sanan, K. Rupp, M. G. Knepley, and B. F. Smith. Extreme-scale Multigrid Components within PETSc. *arXiv:1604.07163 [cs]*, 2016.

[44] J. B. McLaughlin. Aerosol particle deposition in numerically simulated channel flow. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 1(7):1211–1224, 1989.

[45] R. Mittal and G. Iaccarino. IMMERSED BOUNDARY METHODS. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005.

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

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

[48] C. W. Oosterlee and T. Washio. On the Use of Multigrid as a Preconditioner. In *Proceedings of the 9th International Conference on Domain Decomposition Methods*, pages 441–448, Bergen, Norway, 1997.

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

[50] D. Pekurovsky. P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. *SIAM Journal on Scientific Computing*, 34(4):C192–C209, 2012.

[51] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-Based Fast Local Level Set Method. *Journal of Computational Physics*, 155(2):410–438, 1999.

[52] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of computational physics*, 10(2):252–271, 1972.

[53] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11, 2002.

[54] M. Pippig. PFFT: An Extension of FFTW to Massively Parallel Architectures. *SIAM Journal on Scientific Computing*, 35(3):C213–C236, 2013.

[55] L. M. Pismen and A. Nir. On the motion of suspended particles in stationary homogeneous turbulence. *Journal of Fluid Mechanics*, 84(01):193, 1978.

[56] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[57] A. Prosperetti. Motion of two superposed viscous fluids. *Physics of Fluids (1958-1988)*, 24(7):1217–1223, 1981.

[58] M. W. Reeks. On the dispersion of small particles suspended in an isotropic turbulent fluid. *Journal of Fluid Mechanics*, 83(03):529, 1977.

[59] J. J. Riley and J. Patterson. Diffusion experiments with numerically integrated isotropic turbulence. *Physics of Fluids (1958-1988)*, 17(2):292–297, 1974.

[60] P. L. Roe. Characteristic-Based Schemes for the Euler Equations. *Annual Review of Fluid Mechanics*, 18(1):337–365, 1986.

[61] A. M. Roma, C. S. Peskin, and M. J. Berger. An Adaptive Version of the Immersed Boundary Method. *Journal of Computational Physics*, 153(2):509–534, 1999.

[62] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.

[63] Y. Saad. *Iterative Methods for Sparse Linear System*, volume 170. 1992.

[64] E. B. Saff and A. B. J. Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1):5–11, 1997.

[65] R. Scardovelli and S. Zaleski. Direct Numerical Simulation of Free-Surface and Interfacial Flow. *Annual Review of Fluid Mechanics*, 31(1):567–603, 1999.

[66] U. Schumann. Realizability of Reynolds-stress turbulence models. *Physics of Fluids (1958-1988)*, 20(5):721–725, 1977.

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

[68] R. A. Shaw. PARTICLE-TURBULENCE INTERACTIONS IN ATMOSPHERIC CLOUDS. *Annual Review of Fluid Mechanics*, 35(1):183–227, 2003.

[69] W. A. Sirignano. Fuel droplet vaporization and spray combustion theory. *Progress in Energy and Combustion Science*, 9(4):291–322, 1983.

[70] W. A. Sirignano. *Fluid Dynamics and Transport of Droplets and Sprays.* Cambridge University Press, 1999.

[71] W. H. Snyder and J. L. Lumley. Some measurements of particle velocity autocorrelation functions in a turbulent flow. *Journal of Fluid Mechanics*, 48(01):41, 1971.

[72] K. D. Squires and J. K. Eaton. Particle response and turbulence modification in isotropic turbulence. *Physics of Fluids A: Fluid Dynamics (1989-1993)*, 2(7):1191–1203, 1990.

[73] K. D. Squires and J. K. Eaton. Measurements of particle dispersion obtained from direct numerical simulations of isotropic turbulence. *Journal of Fluid Mechanics*, 226: 1–35, 1991.

[74] M. Sussman and E. G. Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162(2):301–337, 2000.

[75] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[76] G. I. Taylor. Diffusion by Continuous Movements. *Proceedings of the London Mathematical Society*, s2-20(1):196–212, 1922.

[77] C.-M. Tchen. *Mean Value and Correlation Problems Connected with the Motion of Small Particles Suspended in a Turbulent Fluid*. Springer Netherlands, Dordrecht, 1947.

[78] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A Front-Tracking Method for the Computations of Multiphase Flow. *Journal of Computational Physics*, 169(2):708–759, 2001.

[79] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.

[80] M. Uhlmann. Interface-resolved direct numerical simulation of vertical particulate channel flow in the turbulent regime. *Physics of Fluids (1994-present)*, 20(5):053305, 2008.

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

[82] P. Wesseling. Cell-centered multigrid for interface problems. *Journal of Computational Physics*, 79(1):85–91, 1988.

[83] P. Wesseling. Introduction to Multigrid Methods. Technical Report 95-11, 1995.

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

[85] Y.-T. Zhang, H.-K. Zhao, and J. Qian. High Order Fast Sweeping Methods for Static Hamilton–Jacobi Equations. *Journal of Scientific Computing*, 29(1):25–56, 2005.

[86] H. Zhao. A fast sweeping method for Eikonal equations. *Mathematics of Computation*, 74(250):603–627, 2005.

[87] H. Zhao. Parallel Implementations of the Fast Sweeping Method. *Journal of Computational Mathematics*, 25(4):421–429, 2007.

[88] H.-K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and Nonparametric Shape Reconstruction from Unorganized Data Using a Variational Level Set Method. *Computer Vision and Image Understanding*, 80(3):295–314, 2000.

Appendix A

Derivation of the GFM for the discretization of the PPE

The *Ghost Fluid Method* (GFM) was developed by Fedkiw et al. [24] to capture the boundary conditions at a contact discontinuity in the inviscid compressible Euler equations. Liu et al. [36] used related techniques to develop a boundary condition capturing approach for the variable coefficient Poisson's equation on domains with an embedded interface. The method was extended by Kang et al. [31] to treat multi-phase incompressible flow including the effects of viscosity, surface tension and gravity. The key feature of the GFM is that it accurately simulates a sharp interface, whereas many previous finite difference methods, like the *Continuum Surface Force* (CSF) approach, involve numerical smearing of the equations near the interface. Consider a 1D domain where the interface Γ is a single point located at x_Γ between the grid nodes x_i and $x_{\mathrm{i}+1}$. Without loss of generality, assume that the phase ”$-$” fills the sub-domain $x < x_\Gamma$ and the phase ”+” occupies the sub-domain $x > x_\Gamma$. Also, $[p]_\Gamma = p_\Gamma^+ - p_\Gamma^-$ and $[\rho]_\Gamma = \rho_\Gamma^+ - \rho_\Gamma^-$ are known. The configuration described is sketched in Figure A.1.

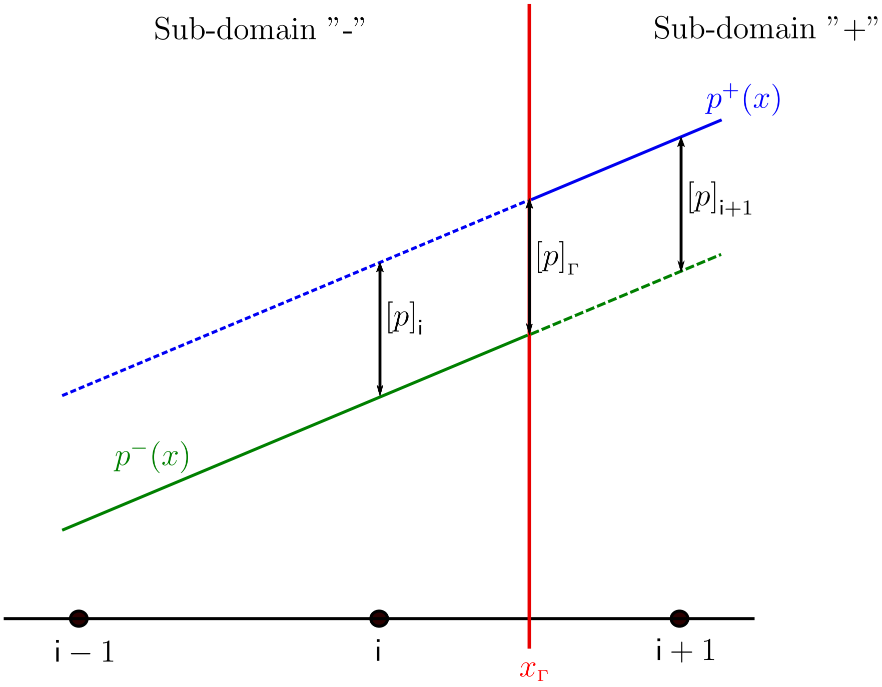


Figure A.1 Typical interface configuration in a one dimensional domain. The jump condition $[p]_\Gamma$ is known at the interface Γ located at x_Γ. p^+ and p^- represent the variable p in the sub-domain "+" and "-" respectively. The dashed lines indicate ghost values, while the continuous lines indicate actual values. The jumps $[p]_{\mathsf{i}}$ and $[p]_{\mathsf{i+1}}$ are extentions of $[p]_\Gamma$ at the grid nodes i and $\mathsf{i}+1$ surrounding the interface.

The goal here is to find a finite difference discretization of the 1D *Pressure Poisson's Equation* (PPE)

$$\frac{d}{dx}\left(\frac{1}{\rho}\frac{dp}{dx}\right) = b \ , \tag{A.1}$$

where b is a generic *Right-Hand Side* (RHS). A standard finite difference discretization of Eq. (A.1) is not appropriate at the nodes i and $\mathsf{i}+1$ because it would use pressure and density values from both sides (phases) of the interface, thus creating instabilities in the solution. Instead, it would be desirable to discretize the Laplacian in Eq. (A.1) by using only values of p and ρ that come from the same phase of the node being discretized. As a consequence, a suitable discretization of the Eq. (A.1) at node i is

$$\left.\frac{\delta}{\delta x}\left(\frac{1}{\rho}\frac{\delta p}{\delta x}\right)\right|_{\mathsf{i}} = \frac{p^-_{\mathsf{i+1}} - p^-_{\mathsf{i}}}{\rho^- \delta x^2} - \frac{p^-_{\mathsf{i}} - p^-_{\mathsf{i-1}}}{\rho^- \delta x^2} \ , \tag{A.2}$$

where $p^-_{\mathsf{i}+1}$ can be thought of as the pressure value at node $\mathsf{i}+1$ if p^- were extended by continuity in the sub-domain ”+”. Since $p^-_{\mathsf{i}+1}$ is fictitious and only $p^+_{\mathsf{i}+1}$ is available, the former is called *ghost value*. The GFM assumes that the jump $[p]_\Gamma$ and its derivatives $[dp/dx]_\Gamma$, $[d^2p/dx^2]_\Gamma$, ... are known at the interface and can be extended by continuity through a Taylor's series. Under these assumptions, the jump of p at node $\mathsf{i}+1$, $[p]_{\mathsf{i}+1}$, can be computed via

$$[p]_{\mathsf{i}+1} = p^+_{\mathsf{i}+1} - p^-_{\mathsf{i}+1} = [p]_\Gamma + (x_{\mathsf{i}+1} - x_\Gamma)\left[\frac{dp}{dx}\right]_\Gamma + \mathcal{O}((x_{\mathsf{i}+1} - x_\Gamma)^2)\ . \tag{A.3}$$

Since now $[p]_{\mathsf{i}+1}$ is known, $p^-_{\mathsf{i}+1}$ can be calculated as $p^-_{\mathsf{i}+1} = p^+_{\mathsf{i}+1} - [p]_{\mathsf{i}+1}$ and used in Eq. (A.2):

$$\frac{\delta}{\delta x}\left(\frac{1}{\rho}\frac{\delta p}{\delta x}\right)\bigg|_{\mathsf{i}} = \frac{p^+_{\mathsf{i}+1} - p^-_{\mathsf{i}} - [p]_{i+1}}{\rho^-\delta x^2} - \frac{p^-_{\mathsf{i}} - p^-_{\mathsf{i}-1}}{\rho^-\delta x^2} \tag{A.4}$$

It should be noted that Eq. (A.4) does not contain ghost values and thus can be used as is. For multi-phase flows, the pressure jump $[p]_\Gamma$ is a known quantity that depends on the local surface tension and shear stresses jump[1], but the pressure gradient $[\partial p/\partial x]_\Gamma$ in Eq. (A.3) is not known a priori and has to be derived. For a non-reactive interface, the velocity is continuous at Γ and so must be the RHS of the *Navier-Stokes Equations* (NSE). Therefore the jump of the pressure gradient divided by the density has to be zero as well:

$$\left[\frac{1}{\rho}\frac{\partial p}{\partial x}\right]_\Gamma = 0 \quad\Longrightarrow\quad \frac{1}{\rho^+}\frac{\partial p^+}{\partial x}\bigg|_\Gamma = \frac{1}{\rho^-}\frac{\partial p^-}{\partial x}\bigg|_\Gamma\ . \tag{A.5}$$

The equality (A.5) can then be manipulated as follows:

$$\frac{1}{\rho^+}\frac{\partial p^+}{\partial x}\bigg|_\Gamma - \frac{1}{\rho^+}\frac{\partial p^-}{\partial x}\bigg|_\Gamma = \frac{1}{\rho^-}\frac{\partial p^-}{\partial x}\bigg|_\Gamma - \frac{1}{\rho^+}\frac{\partial p^-}{\partial x}\bigg|_\Gamma \quad\Longrightarrow\quad \frac{1}{\rho^+}\left[\frac{\partial p}{\partial x}\right]_\Gamma = \left[\frac{1}{\rho}\right]_\Gamma \frac{\partial p^-}{\partial x}\bigg|_\Gamma\ . \tag{A.6}$$

[1] In this study, the viscous term was discretized by using the CSF approach (cf. Subsection 3.2.1). Thus, the shear stresses jump becomes zero and the pressure jump depends on the surface tension effects only.

As a result, the pressure gradient jump can be expressed as

$$\left[\frac{\partial p}{\partial x}\right]_{\Gamma} = -\rho^{+}\left[\frac{1}{\rho}\right]_{\Gamma}\left.\frac{\partial p^{-}}{\partial x}\right|_{\Gamma} = -\rho^{-}\left[\frac{1}{\rho}\right]_{\Gamma}\left.\frac{\partial p^{+}}{\partial x}\right|_{\Gamma} . \tag{A.7}$$

By using the Taylor's series expansion, the following equalities hold:

$$\begin{aligned} \left.\frac{\partial p^{-}}{\partial x}\right|_{\Gamma} &= \left.\frac{\partial p^{-}}{\partial x}\right|_{\mathsf{i}+\frac{1}{2}} + \mathcal{O}(x_{\mathsf{i}+\frac{1}{2}} - x_{\Gamma}) \ , \\ \left.\frac{\partial p^{+}}{\partial x}\right|_{\Gamma} &= \left.\frac{\partial p^{+}}{\partial x}\right|_{\mathsf{i}+\frac{1}{2}} + \mathcal{O}(x_{\mathsf{i}+\frac{1}{2}} - x_{\Gamma}) \ . \end{aligned} \tag{A.8}$$

Since we are discretizing at node i, i.e. in the "$-$" phase, the focus will be on $\partial p^{-}/\partial x$. The latter can be discretized as follows:

$$\left.\frac{\partial p^{-}}{\partial x}\right|_{\Gamma} \simeq \left.\frac{\partial p^{-}}{\partial x}\right|_{\mathsf{i}+\frac{1}{2}} \simeq \frac{p^{-}_{\mathsf{i}+1} - p^{-}_{\mathsf{i}}}{dx} = \frac{p^{+}_{\mathsf{i}+1} - p^{-}_{\mathsf{i}} - [p]_{i+1}}{dx} . \tag{A.9}$$

Substituting Eq. (A.9) into Eq. (A.7) gives an expression for the pressure gradient jump at the interface:

$$\left[\frac{\partial p}{\partial x}\right]_{\Gamma} = -\rho^{+}\left[\frac{1}{\rho}\right]_{\Gamma}\left.\frac{\partial p^{-}}{\partial x}\right|_{\Gamma} \simeq -\rho^{+}\left[\frac{1}{\rho}\right]_{\Gamma}\frac{p^{+}_{\mathsf{i}+1} - p^{-}_{\mathsf{i}} - [p]_{i+1}}{\delta x} . \tag{A.10}$$

By using Eq. (A.10) into Eq. (A.3) one obtains

$$[p]_{\mathsf{i}+1} \simeq [p]_{\Gamma} - \chi\rho^{+}\left[\frac{1}{\rho}\right]_{\Gamma}\left(p^{+}_{\mathsf{i}+1} - p^{-}_{\mathsf{i}} - [p]_{i+1}\right) \ , \tag{A.11}$$

where $\chi = (x_{\mathsf{i}+1} - x_{\Gamma})/\delta x$. Solving the above equation for $[p]_{\mathsf{i}+1}$ gives

$$[p]_{\mathsf{i}+1} \simeq \frac{\rho^{-}}{\rho^{*}}[p]_{\Gamma} + \left(1 - \frac{\rho^{-}}{\rho^{*}}\right)\left(p^{+}_{\mathsf{i}+1} - p^{-}_{\mathsf{i}}\right) \ , \tag{A.12}$$

where $\rho^{*} = \rho^{+}\chi + (1-\chi)\rho^{-}$ is the *weighted average* of the density between i and $\mathsf{i}+1$.

The RHS of Eq. (A.12) contains known quantities and therefore can be used in Eq. (A.4):

$$\frac{\delta}{\delta x}\left(\frac{1}{\rho}\frac{\delta p}{\delta x}\right)\Bigg|_{\mathsf{i}} = \frac{p^{+}_{\mathsf{i}+1} - p^{-}_{\mathsf{i}} - [p]_{\Gamma}}{\rho^{*}\delta x^{2}} - \frac{p^{-}_{\mathsf{i}} - p^{-}_{\mathsf{i}-1}}{\rho^{-}\delta x^{2}} \ . \tag{A.13}$$

Thus the GFM discretization of Eq. (A.1) at node i reads:

$$\frac{p_{\mathsf{i}+1} - p_{\mathsf{i}}}{\rho^{*}\delta x^{2}} - \frac{p_{\mathsf{i}} - p_{\mathsf{i}-1}}{\rho^{-}\delta x^{2}} = b_{\mathsf{i}} + \frac{[p]_{\Gamma}}{\rho^{*}\delta x^{2}} \ . \tag{A.14}$$

It can be shown that the above discretization is symmetric, i.e. it results in a symmetric matrix. The derivation presented here is based on [18].

Appendix B

Numerical Solution of the Pressure Poisson Equation

Any projection method for solving the incompressible *Navier-Stokes Equations* (NSE) relies on the numerical solution of an elliptic *Pressure Poisson's Equation* (PPE) of the form

$$\frac{\partial}{\partial x_i}\left(\beta \frac{\partial p}{\partial x_i}\right) = b \ , \tag{B.1}$$

where the coefficient β may be variable in space. The above equation can be discretized on a Cartesian uniform grid via finite differencing (and the *Ghost Fluid Method* (GFM) if jumps are present), resulting in the discrete PPE

$$\left.\frac{\delta}{\delta x_i}\left(\beta \frac{\delta p}{\delta x_i}\right)\right|_{\mathsf{i,j,k}} = b|_{\mathsf{i,j,k}} \ . \tag{B.2}$$

Two techniques are typically used to solve Eq. (B.2), namely:

1. *Fast Fourier Transform* (FFT): the discrete PPE is transformed to *spectral form* and solved in *Fourier* space;

2. *Iterative methods*: the sparse linear system associated with Eq. (B.2) is solved iteratively.

The FFT approach has the advantages of being very efficient, always delivering a solution accurate to machine precision and not requiring storage of a sparse matrix. It is also vastly available through a large number of highly optimized software libraries, particularly FFTW [27], and works without much tuning. On the other hand, the FFT method is not straightforward to apply to problems with *Boundary Conditions* (BCs) other then periodic and cannot be used when β is variable. Moreover, distributed FFT does not scale as well as some iterative methods, most notably *Multigrid* (MG) methods, because of the large number of inter-processors communications required. Finally, state of the art software packages such as 2DECOMP&FFT [35], P3DFFT [50] and PFFT [54] do not implement 3D domain decomposition[1], thus limiting the number of processors one can use given a certain grid.
In contrast, iterative methods offer more flexibility compared to the FFT approach in that they:

- work for any kind of BCs with little or no modification of the system matrix, and
- can be used for the solution of the system associated with Eq. (B.2) even when β is strongly anisotropic or even singular.

In addition, most iterative techniques scale well in parallel, are widely available via well known software packages and can be used in any type of domain decomposition setting. The main drawback is that they require more tuning than FFT-based approaches for complex problems and they do not deliver machine accuracy. Also, if not properly tuned, they would require more computational time than FFT to obtain a solution. In this work, a *Conjugate Gradient* (CG) method preconditioned via a MG approach was used for the solution of the

[1] The author is not aware on any existing parallel implementation of FFT based on a 3D domain decomposition.

PPE, since the singular nature of the pressure and density at the interface does not allow for a spectral approach. Hereinafter a basic description of the MG method is given; the interested reader is referred to the book by Saad [63] for a more in-depth analysis on the topic.

B.1 Basic Iterative Methods

The second order finite difference discretization of Eq. (B.1) results in a linear system of the form

$$Ap = b \ , \tag{B.3}$$

where A is a sparse symmetric (semi-)positive defined system matrix. A is tridiagonal in the 1D case, pentadiagonal in the 2D case and eptadiagonal in the 3D case. The most straightforward way of solving Eq. (B.3) is via basic iterative methods such as the *Jacobi*, *Gauss-Seidel* (GS) or *Successive Over-Relaxation* (SOR) iterations. The error e^k and residual r^k at the k-th iteration are defined as:

$$e^k = p - p^k \quad \text{and} \quad r^k = b - Ap^k \quad \text{, and thus} \quad Ae^k = r^k \ , \tag{B.4}$$

where p^k is the solution at the k-th iteration and $Ae^k = r^k$ is the *residual equation*. The definitions (B.4) imply that iterating over the original system $Ap = b$ with initial guess p^0 is equivalent to iterating over $Ae^0 = r^0$ with $r^0 = b - Ap^0$, since $p = p^0 + e^0$. Unfortunately, the aforementioned schemes are plagued by a deterioration of the rate of convergence[2] as the iterative cycle progresses. This effect is more prominent as the grid is refined. The stall in the error reduction is caused by the so-called *smoothing property*: the basic iterative

[2] The rate of convergence is a measure of how much the error is reduced between two successive iterations.

methods are very effective in eliminating the Fourier modes of the error with a wavelength comparable to the grid size (*oscillatory modes*), but damp the other modes (*smooth modes*) very slowly [11, 63, 83].

B.2 The Multigrid Method

The MG method takes advantage of the smoothing property described above by defining a sequence of coarser grids on which the smooth modes are seen as oscillatory ones and therefore can be damped very quick by using a basic iteration technique. As a result, the convergence rate becomes independent of grid refinement and does not degrade during the iterative process. In order to illustrate the concept, consider two grids only: a fine one with grid spacing of h, and a coarse one, with grid spacing of $2h$. The discretization of Eq. (B.2) is performed on the fine grid and results in the linear system $A_h p_h = b_h$, where the subscript h indicates the grid spacing. The application of few sweeps, say α_1, of an iterative solver[3] on $A_h p_h = b_h$ reduces the high-frequency components of the error and gives the approximate solution $p_h^{\alpha_1}$. In order to compute the exact solution p_h via $p_h = p_h^{\alpha_1} + e_h^{\alpha_1}$, the error $e_h^{\alpha_1}$ needs to be calculated by solving $A_h e_h^{\alpha_1} = r_h^{\alpha_1} = b_h - A p_h^{\alpha_1}$. On the fine grid, this is equivalent to solving the original system and thus it does not offer any advantage over iterating on $A_h p_h = b_h$ until convergence. On the other hand, solving $A_h e_h^{\alpha_1} = r_h^{\alpha_1}$ on a coarser grid is beneficial because:

- $e_h^{\alpha_1}$ and $r_h^{\alpha_1}$, unlike $p_h^{\alpha_1}$, are smooth on the fine grid, and thus appear oscillatory on the coarse grid where therefore relaxation is more effective, and
- solving a linear system on a coarse grid is computationally cheaper than on a fine grid.

[3] The application of an iterative method for few iterations with the goal of smoothing the error is also known as *relaxation*.

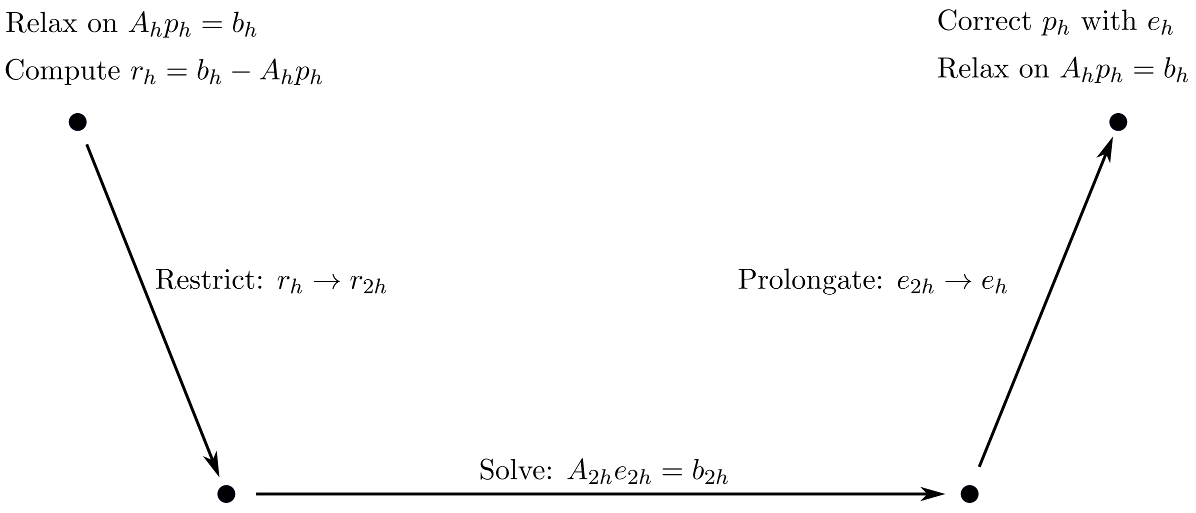

Relax on $A_h p_h = b_h$
Compute $r_h = b_h - A_h p_h$
Correct p_h with e_h
Relax on $A_h p_h = b_h$
Restrict: $r_h \to r_{2h}$
Prolongate: $e_{2h} \to e_h$
Solve: $A_{2h} e_{2h} = b_{2h}$

Figure B.1 Schematic of the CGC scheme. The iteration index has been dropped for clarity.

The *Coarse Grid Correction* (CGC) algorithm that follows is based on the concepts outlined above (see Figure B.1 for a visual representation of the method).

1. *Pre-smoothing*: a basic iterative scheme is applied to $A_h p_h = b_h$ for α_1 iterations with arbitrary initial guess p_h^0. The oscillatory components of the error are removed and $e_h^{\alpha_1}$ appears smooth on the fine grid. The residual is computed as $r_h^{\alpha_1} = b_h - A_h p_h^{\alpha_1}$.

2. *Restriction*: $r_h^{\alpha_1}$ is mapped onto the coarse grid to obtain $r_{2h}^{\alpha_1}$.

3. *Coarse-grid solve*: the residual equation $A_{2h} e_{2h}^{\alpha_1} = r_{2h}^{\alpha_1}$ is solved. $e_{2h}^{\alpha_1}$ contains the information necessary to remove the smooth errors on the fine grid.

4. *Prolongation*: $e_{2h}^{\alpha_1}$ is mapped onto the fine grid to obtain $e_h^{\alpha_1}$.

5. *Fine-grid correction*: an improved initial guess p^* is obtained by correcting $p_h^{\alpha_1}$ with $e_h^{\alpha_1}$ via $p^* = p_h^{\alpha_1} + e_h^{\alpha_1}$. This amends $p_h^{\alpha_1}$ by its smooth error $e_h^{\alpha_1}$.

6. *Post-smoothing*: a basic iterative scheme is applied to $A_h p_h = b_h$ for α_2 iterations with initial guess p^* to remove oscillatory components of the error introduced in the prolongation step.

The CGC scheme is not very useful as is, since solving $A_{2h}e_{2h}^{\alpha_1} = r_{2h}^{\alpha_1}$ is still very expensive. However, one could apply it a second time to solve the coarse problem: this would introduce an even coarser grid where a smaller system of the form $Ae = r$ needs to be solved. The MG method repeats this process recursively until a coarsest grid is reached where the corresponding residual equation is inexpensive to solve even via a direct method. The MG cycle can be of different types depending on how the CGC scheme is repeated in the recursion: the most common cycles are shown in Figure B.2. The MG approach described so far is known as *Geometric Multigrid* (GMG) since the construction of the multilevel hierarchy is based on the knowledge of the partial differential equation to be solved as well as the geometry of the problem. If instead the hierarchy of operators is computed directly from the system matrix, the method is called *Algebraic Multigrid* (AMG). In this case each level is simply a subset of unknowns without any geometric interpretation. A detailed description of both approaches can be found in the excellent tutorial by Briggs [11] and the introduction by Wesseling [83].

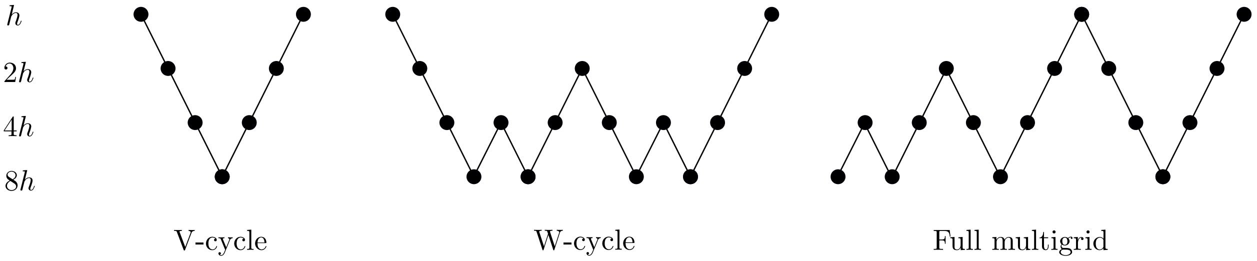


Figure B.2 Visual representation of common MG cycles.

B.2.1 Inter-Grid Operators

The transfer operators used for the prolongation (P_{2h}^{h}) and restriction (R_{h}^{2h}) between grids h and $2h$ play a key role in the MG method in that they are responsible for the mapping from coarse to fine grid and vice-versa. The choice of P_{2h}^{h} and R_{h}^{2h} depends on the type of discretization used on the original *Partial Differential Equation* (PDE) and the features of the problem at hand. The discretization of the PPE in a staggered grid system is *cell-centered*, i.e. the domain is divided into cells where the center point (node) stores the discrete values of the pressure (cf. Section 3.1). As a consequence, prolongation and restriction have to be performed cell-wise rather than node-wise. In this work the prolongation is based on a piece-wise constant interpolation that results in the following mapping for the 2D case (see Figure B.3):

$$\begin{aligned} (v_h)_a &= (P_{2h}^h v_{2h})_a = v_A \ , \qquad & (v_h)_b &= (P_{2h}^h v_{2h})_b = v_B \ , \\ (v_h)_c &= (P_{2h}^h v_{2h})_c = v_C \ , \qquad & (v_h)_d &= (P_{2h}^h v_{2h})_d = v_D \ , \end{aligned} \tag{B.5}$$

where v is the variable being prolongated. R_h^{2h} is constructed as the adjoint of P_{2h}^h multiplied by a suitable scaling factor. While the definitions (B.5) give first order polynomial accuracy only, they performed well for all the cases of interest in this research. Higher order inter-grid operators may speed up convergence and increase the robustness of the algorithm, particularly for cell-centered discretizations and discontinuous coefficients, see for example the works by Wesseling [82], Khalil and Wesseling [32] and Herbert [30]. Nevertheless they are more difficult to set up and are rarely included in software packages, thus they were not pursed in this work.

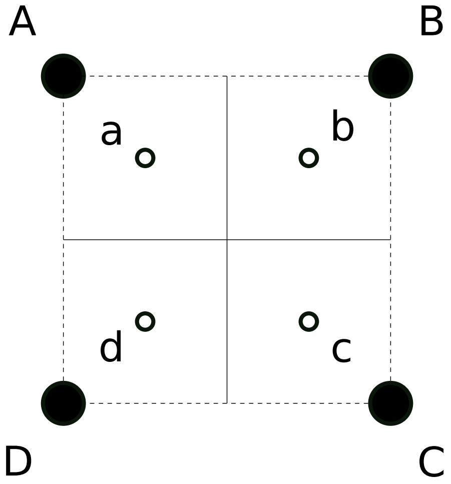


Figure B.3 Cell-centered grid point configuration in two dimensions. Coarse cell centers = A, B, C, D. Fine cell centers = a, b, c, d.

B.2.2 Coarse Grid Operators

Besides prolongation and restriction operators, the other main component of a MG algorithm is the coarse grid operator A_{2h}. The latter can be defined via two different approaches:

1. *Discretization Coarse-grid Approximation* (DCA): A_{2h} is obtained by directly discretizing Eq. (B.1) on the coarse grid;
2. *Galerkin Coarse-grid Approximation* (GCA): the coarse grid operator is obtained via $A_{2h} = P_h^{2h} A_h P_{2h}^h$.

The comparison between the two approaches is discussed by Wesseling [83] and will not be repeated here. In this work, the coarse grid operators were computed by using the GCA.

B.3 Solution Method Used in This Work

The solution of the PPE has been performed via the CG method with a V-cycle GMG as a preconditioner [48], that is GMG is used as a linear solver inside the CG scheme. This approach combines the well-known robustness of *Krylov-subspace methods* with the optimal convergence rate of MG. While many researchers use an AMG preconditioner, all the tests performed in this work showed that GMG outperforms AMG. The GS iteration was used for pre- and post-smoothing: one sweep proved to be enough on the finer grids, while 2 to 4 sweeps were used on the coarser grids to improve convergence speed. Finally the residual equation on the coarsest grid was solved directly via the *LU decomposition*.

The Krylov solver as well the preconditioner were provided by the PETSc library [5, 6, 7]. In order to achieve good parallel scalability on a large number of processor, the GMG preconditioner was coupled with PETSc's PCTELESCOPE preconditioner by May et al. [43] that we helped testing.

Appendix C

The Fast Sweeping Method

The reconstruction of a distance function d requires solving the following boundary value problem:

$$\begin{cases} |\nabla d(\boldsymbol{x})| = 1, & \boldsymbol{x} \in \Omega \setminus \Gamma \ , \\ d(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \Gamma \ . \end{cases} \tag{C.1}$$

The two most efficient methods for the task are the *Fast Marching Method* (FMM) and the *Fast Sweeping Method* (FSM). The FMM, developed by Sethian [67] and later improved by Chopp [15] and Adalsteinsson and Sethian [1], was the first optimally efficient algorithm for solving Eq. (C.1). Its efficiency stems from the fact that the solution is updated by following an ordering that is consistent with the causality relation implicit to the *Eikonal equation* $|\nabla d| = 1$. This means that the distance function is obtained in a sequential fashion by updating one grid point at a time so that the solution is strictly increasing (decreasing). Hence an upwind difference scheme, a *heapsort algorithm* and possibly a *heap data structure* are needed, resulting in an algorithmic complexity of order $\mathcal{O}(N \log N)$, where N is the size of the problem. While optimal, the FMM has some limitations: its sequential nature makes it difficult to parallelize, the data structures involved are complex and higher order extensions of the method are not straightforward. The FSM was developed by Zhao et al. [88] and Zhao [86] as a simpler alternative to the FMM. A parallel implementation was given by Zhao [87] and higher order versions were derived by Zhang et al. [85]. The main idea of the FSM is to use nonlinear upwind differencing and *Gauss-Seidel* (GS) iterations with alternating sweeping ordering to update the solution consistently with the causality expressed by $|\nabla d| = 1$. This approach has a complexity of $\mathcal{O}(N)$. After extensive testing of both methods, the FSM proved to be a better and simpler choice for the purpose of this work. In what follows, the first order serial FSM is presented for the case of a two-dimensional domain: the extension to the 3D case is straightforward. The first order FSM discretizes the Eikonal equation $|\nabla d| = 1$ via the following Godunov's upwind scheme [62]:

$$[\max(d_{\mathsf{i,j}} - a\,,\,0)]^2 + [\max(d_{\mathsf{i,j}} - b\,,\,0)]^2 = \delta x^2 \;, \tag{C.2}$$

where a and b are given by:

$$\begin{aligned} a &= \min(\,d_{\mathsf{i+1,j}}\,,\,d_{\mathsf{i-1,j}}\,)\;,\\ b &= \min(\,d_{\mathsf{i,j+1}}\,,\,d_{\mathsf{i,j-1}}\,)\;. \end{aligned} \tag{C.3}$$

Eq. (C.2) is a quadratic equation and can be solved for $d_{\mathsf{i,j}}$. Its solution is given by:

$$d_{\mathsf{i,j}} = \begin{cases} \min(a,b) + \delta x^2, & \text{if} \quad |a-b| \geq \delta x\;, \\ \dfrac{1}{2}\left[a + b + \sqrt{2\delta x^2 - (a-b)^2}\right], & \text{if} \quad |a-b| < \delta x\;. \end{cases} \tag{C.4}$$

The upwind scheme (C.2) is valid at the interior points of the domain, i.e. for $\mathsf{i} = 1, ..., N_x - 2$ and $\mathsf{j} = 1, ..., N_y - 2$, while one sided differencing is used at the boundaries. Given the discretization presented above, the fast sweeping algorithm solves the boundary value problem (C.1) via the following steps.

1. *Initialization.* All the nodes in the domain are initialized with large positive values except in a small band near the interface, where the correct solution is used instead. The nodes inside the small band are never updated during the whole process.

2. *Update.* The whole domain is swept with four alternating orderings, i.e.:

   - $\mathsf{i} = 0 \to (N_x - 1) \qquad \mathsf{j} = 0 \to (N_y - 1)$
   - $\mathsf{i} = (N_x - 1) \to 0 \qquad \mathsf{j} = 0 \to (N_y - 1)$
   - $\mathsf{i} = 0 \to (N_x - 1) \qquad \mathsf{j} = (N_y - 1) \to 0$
   - $\mathsf{i} = (N_x - 1) \to 0 \qquad \mathsf{j} = (N_y - 1) \to 0$

   During each sweep, for each node not included in the small band a value $d_{\mathsf{i,j}}^{new}$ is computed by using the current values of its neighbors via Eq. (C.4). The value at the node is updated by taking the minimum between $d_{\mathsf{i,j}}^{new}$ and its current value.

3. *Convergence check.* The difference between the solution before and after the update step is computed in some norm. If it is larger than a chosen tolerance, the process is repeated from 2. Otherwise, the process is stopped.