Immersed Boundary Method for Fluid-Fluid-Solid Three Phase Flows

by

Berk Altunkeyik

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Mechanical Engineering

KOÇ ÜNİVERSİTESİ

July 1, 2020

Immersed Boundary Method for Fluid-Fluid-Solid Three Phase Flows

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Berk Altunkeyik

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. Dr. Metin Muradoğlu

_____

Prof. Dr. Kunt Atalık

_____

Asst. Prof. Savaş Taşoğlu

Date: _____

to my beloved family and dear friends

ABSTRACT

Immersed Boundary Method for Fluid-Fluid-Solid Three Phase Flows
Berk Altunkeyik
Master of Science in Mechanical Engineering
July 1, 2020

Multiphase flows can be observed in a broad range of industrial and natural processes, and their ubiquity puts them in a significant position in computational fluid dynamics (CFD). This importance also brings a requirement of understanding the underlying physics in detail and one way to achieve this goal is to do high fidelity simulations based on the first principles in a computational environment. These multiphase systems can contain fluid-fluid interactions, fluid-solid interactions and often both of them, therefore various methods have been developed and successfully applied to a wide range range of problems of practical importance in the last few decades. The multiphase flow systems are particularly of interest in microrobotic applications such as a chemotherapy delivery operation to a cancerous area by a remote-controlled microswimmer and in biological flows such as the flow in human lungs. The ultimate goal of this thesis is to develop a new high-fidelity numerical method for simulations of fluid-solid interactions and combine it with a front-tracking algorithm to simulate fluid-fluid-solid interactions in three phase systems.

In the present study, an immersed boundary method (IBM) is developed to treat fluid-solid interfaces that are fully or partially immersed in a host fluid. Using the front-tracking framework, the solid-fluid interfaces are represented by a Lagrangian grid that consists of marker points located at the vertices of a 3D triangular surface mesh. The direct forcing method developed by Uhlmann [152] is used to impose the no-slip and no-penetration boundary conditions on the interface. The Newton-Euler equations are solved for the rigid body dynamics of the fully immersed undeformable solid particles in a Lagrangian frame and tightly coupled with the continuum flow equations solved on a fixed Eulerian grid. For this purpose, quantities calculated on the Lagrangian grid are smoothed onto the Eulerian grid with a distribution function and the same distribution function is also used to transfer quantities from

the interface to the Eulerian grid in a conservative manner similar to the distribution/interpolation operations in the front-tracking method.

The hybrid immersed-boundary/front-tracking (IBM/FTM) solver is then validated for several benchmark cases. First, it is applied to simulate a single stationary rigid sphere exposed to a uniform flow at different Reynolds numbers and the results are found to be in good agreement with the results in the literature. Then, sedimentation of a single sphere in a resting fluid is simulated for various Reynolds numbers and the results are compared with the experimental results of Mordant and Pinton [102]. The method is demonstrated to be convergent in terms of spatial and temporal errors. The numerical method is also validated for a wide range of Reynolds numbers studied experimentally by Cate et al. [27]. Finally, the method is used to simulate particulate flows involving many particles and preliminary results are obtained. The method is found to be robust and the results look qualitatively accurate but a more rigorous quantitative comparison must be done for the validation, which is a subject of a future study. The method is totally developed in the front-tracking framework and can be readily used to simulate three phase flows involving fluid-fluid and fluid-solid interfaces.

ÖZETÇE

Yüksek Lisans Tez Başlığı
Berk Altunkeyik
Makine Mühendisliği, Yüksek Lisans
1 Temmuz 2020

Çok fazlı akışlar endüstriyel süreçlerden biyolojik akışlara kadar pek çok alanda gözlemlenebilir ve bunların yaygınlığı hesaplamalı akışkan dinamiğinde (CFD) önemli bir konuma getirir. Bu önem aynı zamanda temel fizikleri ayrıntılı olarak anlama gereksinimini de beraberinde getirir ve bu hedefi başarmanın bir yolu hesaplamalı bir ortamda temel prensiplere dayanan yüksek doğrulukta simülasyonlar yapmaktır. Bu çok fazlı sistemler sıvı-sıvı etkileşimlerini, sıvı-katı etkileşimlerini ve sıklıkla da her ikisini de içerebilir, bu nedenle son birka 10 yılda farklı yöntemler bulundu ve geni bir uygulamalı problem yelpazesinde çok başarılı oldukları gösterildi. Üç fazlı sistemler, özellikle uzaktan kumandalı bir mikroyüzücü ile kanserli bir bölgeye kemoterapi verme operasyonu gibi mikrorobotik uygulamalarında veya insan akciğerindeki akış gibi biyolojik akışlarla ilgilidir. Bu çalışma, sıvı-katı etkileşimleri için yeni bir sayısal yöntem geliştirmeyi ve bu yeni çözücüyü üç fazlı sistemlerin içindeki sıvı-sıvı-katı etkileşimlerini simüle etmek için bir ara-yüz izleme algoritması ile birleştirmeyi amaçlamaktadır.

Bu çalışmada, bir akışkana tamamen veya kısmen gömülmüş sıvı-katı ara yüzüne müdahale etmek için bir Daldırılmış sınır (IBM) algoritması geliştirilmiştir. Katı-sıvı ara yüzleri Ara-Yüz İzleme Metodunun iskeletini kullanarak, ara yüz izleme yöntemine benzer şekilde bir 3 boyutlu üçgensel yüzey ağının köşelerinde bulunan işaret noktalarından oluşan bir Lagrangian ızgarası ile temsil edilmektedir. Kaymazlık ve nüfuz etmezlik sınır koşullarını dayatmak için Uhlmann'ın [152] geliştirdiği doğrudan zorlama yöntemi kullanılmıştır. Tamamen akışkana gömülmüş katı partikülün cisim dinamikleri için Newton-Euler denklemleri bir Lagrangian ızgarasında çözülür ve Eulerian ızgarasında çözülen akışkan denklemleri ile sıkıca eşleştirilir. Çözüm süreci boyunca, Lagrangian ızgarasında hesaplanan değişkenler bir dağıtım fonksiyonu ile Eulerian ızgarasına yumuşatılmıştır ve aynı dağıtım fonksiyonu nicelikleri ara yüzden Eulerian ızgarasına ara-yüz metodundaki dağıtım/interpolasyon

operasyonlarına benzer şekilde korunarak transfer etmek için de kullanılmıştır.

Bu hibrit Daldırılmış Sınır Metodu/Ara-Yüz İzleme Metodu (IBM/FTM) çözücü, birkaç olayı simüle etmek için kullanılmıştır. İlk olarak, sabit bir katı küre, farklı Reynolds(Re) sayılarında tekdüze bir akışa maruz bırakılır ve sonuçların literatürdeki sonuçlarla oldukça uyuştuğu bulunur. Ardından, dingin bir akışkanda tek bir kürenin düşüşü çeşitli Reynolds sayıları için gözlenir ve sonuçlar Mordant ve Pinton'un [102] deneysel sonuçları ile kıyaslanır. Metodun mekansal ve zamansal hatalar açısından yakınsak olduğu gösterilmiştir. Sayısal yöntem ayrıca Cate et al.'ın [27] deneysel olarak çalıştığı geniş bir aralıktaki Reynolds sayıları ile de doğrulanır. Son olarak, IBM/FTM metodu pek çok parçacığın dahil olduğu çok partiküllü akışları simüle etmekte kullanılır ve ön sonuçlar elde edilmiştir. Yöntemin güçlü olduğu tespit edilmiştir ve sonuçlar niteliksel olarak doğru gözükür ama onaylama için daha titiz niceliksel kıyaslar yapılmalıdır ancak gelecekte yapılacaktır. Yöntem tamemen Ara-Yüz izleme iskeletinde geliştirilmiştir ve üç fazlı sıvı-sıvı ve sıvı-katı etkileşimlerinin olduğu akışlarda kullanılmaya hazırdır.

ACKNOWLEDGMENTS

First of all, I want to share my gratitudes and acknowledgements to my advisor Prof.Dr. Metin Muradoğlu for his support and believe in this project. His knowledge and guidance helped me to be able to complete this thesis and discover my path in the academia. I wish to thank all of my group mates but especially sincere Zaheer Ahmed for giving his time and ideas about the code. In addition, I would be glad to send my appreciations to Dr.Pedro Costa because of sharing his experiences with me. I am grateful to my beloved parents and sisters due to their endless efforts to help me. Finally, I would like to thank my "Goodfellas" for creating a cheerful environment.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

2D	:	Two Dimensional
3D	:	Three Dimensional
CFL	:	Courant-Friedrichs-Lewy Number
D_p	:	Diameter of the Particle
Fr	:	Froude Number
FTM	:	Front Tracking Method
$\mathbf{g}$	:	Gravitational Acceleration
h	:	mesh width
I	:	Indicator Function
IBM	:	Immersed Boundary Method
L_x, L_y, L_z	:	Channel Length in Given Direction
N_x, N_y, N_z	:	Grid Number in Given Direction
NS/NP	:	No-Slip / No-Penetration
Re	:	Reynolds Number
Re_p	:	Particle Reynolds Number
rhs	:	Right Hand Side
t_{ref}	:	Reference Time
$\mathbf{u}$	:	Velocity Vector
u_{ref}	:	Reference Velocity
$\mathbf{u}_p$	:	Particle Velocity
$\mathbf{u}^*$	:	First Unprojected Velocity Vector
$\mathbf{u}^{**}$	:	Second Unprojected Velocity Vector
$\mathbf{u}^d$	:	Desired Velocity Vector
$\tilde{\boldsymbol{U}}$	:	Predicted Velocity Vector
$\mathbf{x}$	:	Coordinates of any Given Point
$\mathbf{X}_l$	:	Coordinates of Lagrangian Element
$\mathbf{X}_c$	:	Coordinates of Particle Center
Δt	:	Time Step
$\Delta x, \Delta y, \Delta z$	:	Grid Size in Given Direction
ρ_f	:	Fluid Density
ρ_p	:	Particle Density
ν	:	Kinematic Viscosity
$\boldsymbol{\Omega}$	:	Rotational Velocity Vector

Chapter 1

INTRODUCTION

The multiphase flow refers to flow systems that contain fluid-fluid, fluid-solid and fluid-fluid-solid interfaces such as bubbles, droplets, interfacial flows and solid particles immersed in a host fluid. Multiphase flows are ubiquitous in natural processes and engineering applications [10, 94]. Examples of natural processes include pyroclastic flows from volcanoes, avalanches, sediment-laden river flows, sandstorms, cloud and planetary flows as well as biological flows such as blood cells flow in blood vessels, pulmonary flows, and the motion of suspended micro-organisms in water (e.g., plankton). Examples of engineering and technological applications include food processing, particulate flows in fluidized beds, slurry transportation, oil-well drilling and pulp fibers in paper making.

The multiphase flows are complex mainly due to discontinuous variations of material properties across the phase boundaries and non-linear interactions of phases. Therefore modeling of multiphase flows is a difficulty task which requires understanding of the inherently complex interactions of phases in a broad range of flow conditions. Although the effort to simulate multiphase flows started as early as the beginning of the computational fluid dynamics, the problem has proved to be formidable and thus the progress has become slow. The main difficulty arose from the requirement of tracking moving and deforming phase boundaries. Great progress has been made in interface-resolved simulations of multiphase flows in last few decades, see for instance, Tryggvason et al. [147]. However, the focus has been on two-phase flow fluid-fluid or fluid-solid systems and fluid-fluid-solid three phase flow systems have remained unexplored in spite of wide range of applications in conventional technologies such as fluidized bed reactors [113] as well as in emerging

technologies such as microfluidics [85] and soft robotics [134]. It is emphasized here that different methodologies have been developed to treat fluid-fluid and fluid-solid interfaces. Therefore, a unified approach needs to be used for interface-resolved simulations of three-phase fluid-fluid-solid flows. In this regard, the front-tracking method offers a distinct advantages since it is essentially based on the immersed boundary method developed to simulate fluid-solid interactions.

In the present work, we aim to develop a high-fidelity numerical method for treatment of fluid-solid interactions to rigid particle-laden incompressible viscous flows as part of a long term project in which the ultimate goal is to develop a computational tool for interface-resolved direct numerical simulations of flows involving both fluid-fluid and fluid-solid interfaces. For this purpose, an immersed boundary method is developed in the front-tracking framework and is shown to be a viable tool for simulation of the fluid-solid multiphase flows. Since the method is implemented in the existing front-tracking code that has been developed and successfully used to simulate a wide-range of fluid-fluid multiphase flows involving multi-physics effects such as surface tension, surfactant and viscoelasticity, its extension to fluid-fluid-solid systems is straightforward. Furthermore, the method can be readily extended to simulate flows involving strong fluid-solid interactions where solid undergoes a significant deformation. Since the front-tracking method plays a vital role and forms the backbone of the present immersed boundary method, we briefly describe the front-tracking method in the next section.

1.1 Front-Tracking Method

The front-tracking method was developed by Tryggvason and Univerdi [155] in early 90s and has been successfully applied to a vast class of fluid-fluid multiphase flow systems for a wide range of flow regimes from highly laminar to fully turbulent cases [146, 147]. It is a hybrid Eulerian-Lagrangian method and is essentially based on the immersed boundary method developed by Peskin [115] in 1970s to simulate blood flow in the heart. Before discussing the details of the front-tracking method, the numerical methods developed for interface-resolved simulations of fluid-fluid

multiphase flows are first briefly reviewed.

Since the very beginning of the computational fluid dynamics(CFD), multiphase flows have found a major position in the field, and some early techniques such as Marker-and-Cell (MAC) method [60] have been developed to simulate multiphase/multifluid flow systems including the effects of surface tension. Capturing the interface on a stationary Eulerian grid is the oldest and is still the most popular approach used, for instance, in the MAC method in which marker particles are used to identify each fluid and in the Volume-of-Fluid (VOF) method in which a marker function is used for the same purpose [62]. The VOF method can be considered as a successor of the MAC, because in the VOF formulation the marker points are changed with the marker function. The original MAC and VOF methods suffered from several deficiencies such as excessive parasite currents and artificial loss or increase of droplets/bubbles even when the flow is full incompressible. A number of algorithms have been developed to overcome these deficiencies in the VOF method and have been successfully applied to a wide range of multiphase and interfacial flows. A comprehensive review of the VOF method can be found in Scardovelli and Zaleski [131]. Zaleski et al. [58] have recently created a Parallel Robust Interface Simulator (PARIS) based on the VOF and demonstrated the performance of the method with impressive results for a wide range of multiphase flows of practical importance. Although the color function evolves by a simple looking advection equation, it turns out that tracking of interface between phases is a notoriously difficult task. Alternative Eulerian-Eulerian techniques such as the level set method [112, 141] and the phase-field method [70] have been developed. However, the main drawback of the Eulerian approaches is that the interface does not remain thin and approximation of surface tension results in excessive parasite currents [131].

Unlike the VOF and level-set methods, Ryskin and Leal [128, 129] used two separate boundary fitted grids in each fluid domain to simulate deformable bubbles with high accuracy. Since this pioneering work, similar methods have been developed and successfully applied to simulate fluid-fluid multiphase flows. The boundary-fitted grids are mostly employed for steady and unsteady motions of simple geometries and free surfaces. This method is usually based on the arbitrary

Lagrangian-Eulerian (ALE) methodology [61]. Takagi and Matsumoto [143] applied this method to simulate motion and deformation of a single 3D bubble and reported impressive results.

Lagrangian methods have a distinct advantage of eliminating numerical diffusion and thus keeping the phase boundary thin throughout the simulations. Lagrangian methods are also often called mesh-free methods or particle methods. The smoothed particle hydrodynamics (SPH) [100] is one of the most common example of Lagrangian methods. Even though SPH has been widely used in simulating free surface flows, it has been also extended to the other fields such as solid mechanics [89]. However, the method has suffered from difficulties in imposing the boundary conditions and high computational cost for each particle in kinetic-energy spectrum problems. Examples of the Lagrangian methods include the breakup of a viscous drop by Oran and Boris [111], initial deformation of a buoyant bubble by Shopov et al. [133], and the recent studies on the motion of the spherical particles in three dimensional flows by Johnson and Tezduyar [77]. Even if this method offers remarkable accuracy for simple geometries, it is troublesome to employ it for complex three-dimensional cases or increasing number of particles because of excessive computational cost.

To overcome disadvantages of the Lagrangian methods, Glimm et al. [52] developed the most primitive version of the front-tracking method (FTM), and Richtmyer and Morton [125] used the term "*Front-Tracking*" for the first time in the literature. In the original FTM, a front is employed to mark the interface with the Lagrangian points on a fixed Eulerian grid and each phase is treated separately. In this study, the framework developed by Tryggvason and Unverdi [155] is used to create the fluid-solid interface and it can be distinguished from the original FTM by the one-field formulation in which a single set of governing equation is written and solved for the entire computational domain. In this approach, a separate Lagrangian grid is used to represent the fluid-solid interface in the same way as done for the fluid-fluid interfaces in the front-tracking method. In fact, it is not necessary to use a separate Lagrangian mesh if the solid object is non-deformable since the interface can be determined from the information about the location of the center of the mass of the

object and its orientation. However, a separate Lagrangian grid is used because the ultimate goal of this project is to simulate the interactions of flexible soft bodies with the flow, where tracking the fluid-solid interface is required. This also facilitates the use of the same data structure both for solid and fluid particles in three-phase flow simulations. As far as in our knowledge, the combined strategy has been employed frequently for fluid-fluid interfacial flows [104,105,107,109,144,146] but not used for fluid-fluid-solid interactions in three-phase flows.

Following Tryggvason and Unverdi [155], the one-field formulation is used in the present study where one-set of flow equations is written for the entire computational domain and the surface tension is included as a body force in the momentum equations in the case of the fluid-fluid interfaces. In addition, a stationary staggered grid is used for the flow equations. This method was examined for two and three dimensional bubbly flows for various test cases. The motion of two dimensional bubbles for a range of Reynolds number between 1 to 100 were investigated by Jan [71] and the interactions of bubbles with each other and unsteady mixing layers were examined by Loth et al. [91]. In addition, Tryggvason and Juric [79] modified the algorithm for simulations of dendritic solidification of pure materials. Some other impressive applications of FTM include parallel 3D simulations of bubbly flows involving many bubbles [25, 103] and head-on collision and coalescence of drops by Nobari [108]. Olgac and Muradoglu [109] used FTM to investigate the effects of soluble and insoluble surfactants on liquid layer in the Bretherton problem. Izbassarov and Muradoglu [107] developed a robust and highly efficient finite-difference/front-tracking method (FD/FTM) for interface-resolved simulations of viscoelastic multiphase flow systems.

In this project, a Lagrangian grid is used to track the solid-fluid interface and a discontinuous marker or indicator function is constructed based on the Lagrangian grid to identify the fluid and solid domains. The Lagrangian grid consists of marker points located at the corners of triangular surface mesh. The Lagrangian marker points move according to the rigid-body dynamics and local flow velocity interpolated from the Eulerian grid on which the flow equations are solved. The marker function is computed at every time step using the standard procedure [146]. The

marker function and its properties will be discussed in Chapter 2 in detail. The Lagrangian grid is also called *front* that marks the location of the fluid-solid interface. Figures 1.1 and 1.2 show structure of 2D and 3D fronts, respectively. As seen, the marker points are located at the end points of line segment and at the corner of a triangular surface mesh in 2D and 3D, respectively. The line segments in 2D and the triangular elements in 3D are called *front elements*. Linked lists are used to track the connectivity of the marker points and the front elements. A separate front (or Lagrangian grid) is used for each solid-fluid interface to facilitate communication in parallel computations. In the case of deformable particles, the fronts are regularly restructured to keep the front elements nearly uniform and comparable to the Eulerian grid size since the front elements may get too small or too large as the interfaces undergoes large deformations. The details of the method can be found in Tryggvason et al. [146]. Note that restructuring of the interface is not needed in the present study since the solid particles are assumed to be rigid but the method is developed and implemented such that restructuring can be switched on if needed.

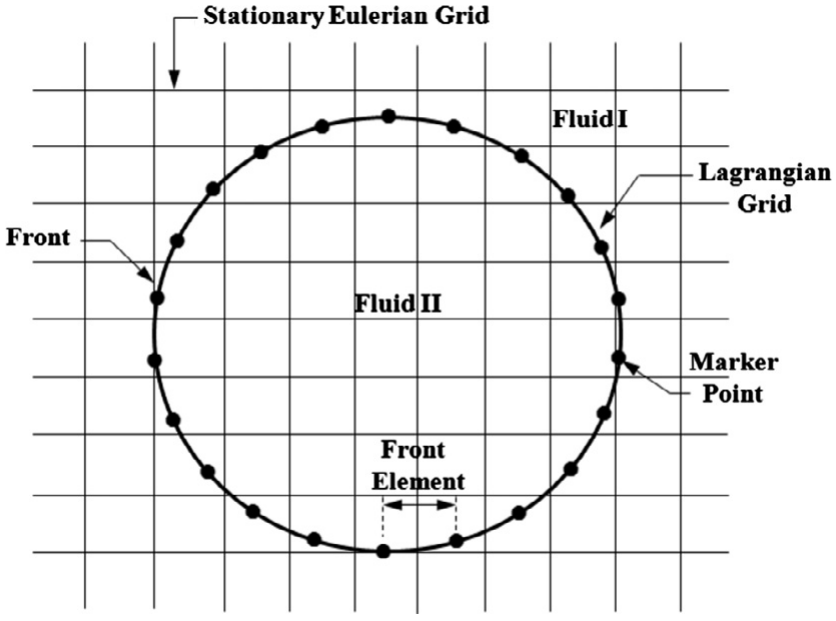

Figure 1.1: The structure of the Eulerian and the Lgrangian grids in 2D. The Lagrangian grid consists of marker points and a piece of the Lagrangian marker points is called a front element. Linked lists are used to define and maintain the connectivity of the marker points and front elements.

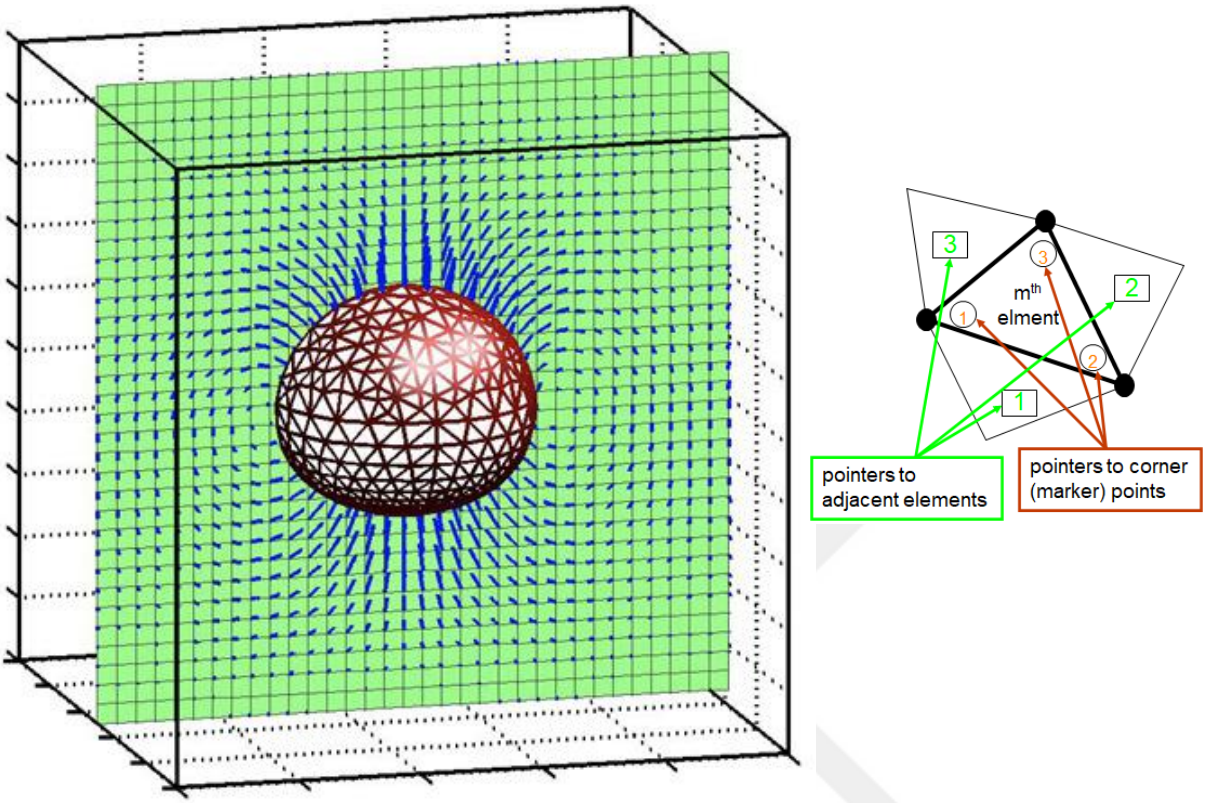

Figure 1.2: The structure of 3D Lagrangian and Eulerian grids (left) and connectivity of the front elements (right).

1.2 Fluid-Solid Interactions

In the most general definition, fluid-structure or fluid-solid interaction (FSI) is defined as the interaction of a movable or deformable solid structure with an internal or surrounding fluid flow. Some popular and common examples of fluid-solid interaction problems include aeroelasticity [26, 167], sedimentation, biological flows and biomechanics [3, 44, 101], complex flows in irregular domains [149, 151] and some other interdisciplinary problems [12, 18, 35, 47, 64, 90, 135, 136, 159]. Coupling of the fluid and solid domains has proved to a formidable task mainly due to inherent non-linearity of interactions and the stiffness caused by the large disparity of length and time scales of solid and fluid dynamics. In addition, treatment of the boundary conditions at the moving and deforming fluid-solid interfaces makes the problem extremely complicated and poses a challenge in computational sciences.

The numerical methods for the fluid-solid interactions can be categorized under two main classes according to mesh structure: (i) Body-conforming mesh and (ii) non-body conforming mesh methods. Both methods have some subclasses but before explaining them comprehensively, it is important to understand their differences. In the body-conforming mesh methods, as the name suggests, the interface is considered as a part of the solution, therefore, the mesh covers the interface as the part of the domain. When the solid body is moved or deformed, re-meshing is required to make the grids in the solid and fluid domains match at the interface, so the interface conditions are imposed in each domain as the physical boundary conditions. In the non-body conforming mesh methods, re-meshing is not required because a Lagrangian grid or marker points are employed to maintain communication between the solid and fluid domains. We next explain body-conforming and non-conforming methods separately.

1.2.1 Body-Conforming Mesh Methods

In body-conforming mesh methods, the aim is to solve fluid and solid parts in a synchronized way using grids on solid and fluid domains that perfectly match at the interface. A computational grid conforms with the body of the solid particle or the structure and as a result it employs an accurate description of the boundary conditions at the interface. The entire solution procedure can be summarized as follows. The flow equations are first advanced for a time step on the fluid domain assuming the front position is fixed at the previous time level. Then, the flow-induced stresses are evaluated at the interface and passed to the solid domain to be used as the boundary conditions in advancing the solid equations. Finally, the solid constitutive equations are advanced for a time step. At the end of this process, the location of the fluid-solid interface is advanced and the mesh is updated accordingly. For the solid-body or structure, it is important to compute the stresses on the bearing members while the location of the physical boundary is important for the fluid dynamics. This mismatch between the fluid domain and the solid body creates compelling challenges for the body-conforming mesh methods. In

some extreme cases, according to the fidelity of the grid, even gaps can be observed in the portions of the interface where the mismatch is large. The requirement of re-gridding at every time step is another disadvantage of body-conforming methods since computational cost becomes extremely large in complex geometries. Brown [24] and Onishi et al. [110] developed two different algorithms to transfer interface data based upon the shortest distance and normal projection to handle the fluid-solid coupling. Appa [6], Guruswamy [59] and Kapania et al. [80] developed and modified a method that creates a virtual front to cover the interface between fluid and solid object and called it *mortar method*. Farhat [38] and Zhang et al. [165] achieved a second order accuracy in time with a body-conformal method for fluid-structure interaction problems. Zhang et al. [165] used two different algorithms based from the Runge-Kutta and hybrid linear multi step method to simulate an aeroelastic flutter problem. Vierendeels [33] introduced a *reduced-order model (ROM)* to simulate the fluid-solid interactions of a heart valve and showed that the method is stable. Wood et al. [160] added an extra sub-iteration step between fluid dynamics and structure analysis computations to reduce the numerical error in the problem studied by Vierendeels [33]. The body-conforming methods are good at resolving particles with a high accuracy but numerical accuracy stats to deteriorate in the case of moving or deforming interfaces, computational cost also tends to increase. Hu et al. [65] developed a body-conforming method where the flow equations are solved implicitly with a finite-element formulation and the particle velocity is updated using an *arbitrary Lagrangian-Eulerian (ALE)* technique. They computed the migration and sedimentation of solid spheres in Newtonian and viscoelastic fluids. Although they obtained stable and accurate results, the computational cost and difficulty of the mesh generation make this method inconvenient for the multiphase problems with moving boundaries. Ferziger and Peric [76] observed that the body conforming methods may affect the accuracy and the convergence properties of the solver in an adverse way. Therefore, even for simple geometries, generating a body-conformal grid can involve a cumbersome iterative process. The level of difficulty of generating an acceptable grid hardens proportionally with the complexity of the shape of the body. Quarteroni and Valli [123] used a *structured grid* method where the volume

is divided into subdomains and an individual grid is generated in each subdomain. However, the communication between each subdomain required development of a highly complicated parallel algorithm. In recent years, Baum et al. [72], Ramamurti and Sandberg [124], and Tezduyar [145] have made some remarkable progress in moving boundary simulations with body-conformal grids but the body conforming mesh methods are still applicable to relatively simple geometries and the computational cost increases prohibitively with the complexity of the geometry.

1.2.2 Non-Body Conforming Mesh Methods

In the non-body conforming mesh methods, the momentum equations are discretized on the Cartesian grid which brings the advantage of generating the grid easily. While remeshing is eliminated with these methods, imposing the boundary conditions becomes complicated. The Cartesian grid methods or non-body conforming mesh methods are widely used for Euler equations [2, 14, 34, 114]. The cut-cell method belongs to the family of the Cartesian grid methods in which the boundary grid cells are enlarged and finite volume is used to approximate the fluxes [69]. The two dimensional viscous incompressible flows were simulated with a finite volume approach by Ye et al. [162]. In the cut-cell method, a co-located grid is usually used. The identification of the cells cut by the interface and how the boundary cells are combined are the key ingredients of the method. Implementation of the cut-cell method is relatively ease in 2D but it becomes extremely complicated in 3D especially in the case of existence of deforming interfaces. In addition, numerical instability arises when the boundary cells are too small or cell aspect ratio is too large. The fictitious domain method is another common non-body conforming method used to simulate the particle-laden flows and fluid-structure interactions. Glowinski et al. [53] combined finite-element method with the splitting technique to achieve a Lagrange multiplier-based fictitious domain algorithm to model viscous flow around a NACA0012 airfoil. The *viscosity penalization method (VPM)* is the remarkable and prominent concepts used in the fictitious domain method. Ritz and Caltagirone [126] used VPM to obtain a continuous hydrodynamics model for fluid-

particle systems, and Angot [5] published a benchmark paper about VPM in 1999. In this method, viscosity inside the particle is assumed to be much much larger than that of the bulk fluid. The method has been demonstrated to be highly successful for some applications but the large viscosity ratio may cause numerical instability when the solver is not capable of coping with the viscosity jumps between the bulk fluid and the alleged fluid inside the particle.

1.3 Immersed-Boundary Method

Immersed Boundary Method (IBM) developed by Peskin in 1972 [115] has become one of the most widely used non-body conforming methods for fluid-solid interactions. Peskin used the original IBM to model moving elastic boundaries in fluid-structure interaction problems, especially in biological flows such as cardiac mechanics and related blood flows. The main advantage of the IBM is to impose the boundary conditions on a non-body conforming grid to solve the flow equations. The IBM can be seen as a combination of a mathematical formulation and numerical scheme. The mathematical formulation consists of both Eulerian and Lagrangian variables, and linked by a Dirac delta function to transfer necessary variables between Eulerian to Lagrangian grids and vice versa. In the Peskin's original method, Eulerian variables are defined and solved on a fixed Cartesian mesh while the Lagrangian quantities move freely on a curvilinear Lagrangian mesh. The curvilinear mesh is chosen due to the wavy geometry of the heart valves. The interactions and exchange of the Eulerian and Lagrangian variables are done using a smoothed approximation to the Dirac delta function constructed in different ways.

The Cartesian grid methods can be seen as a premature version of the IBM to simulate inviscid flows with complex embedded solid boundaries. The Cartesian grid methods have evolved and been supported by similar but limited capabilities with the IBM [13, 30, 34, 150, 162]. Anderson et al. [4], and Scardovelli and Zaleski [131] have reviewed various applications of the Cartesian grid methods for the liquid-liquid and liquid-gas multiphase flow systems but and not the fluid-solid coupling. Using a Cartesian grid facilitates the combination of IBM method with the

numerical methods developed for simulations of multiphase flows including the commercial software packages and existing solvers such as the fast Poisson solver [142], the level-set method [1] and the structured multigrid solvers [88,163]. Although it is mostly used in the finite difference method context, the IBM can also be used with the finite element and finite volume methods because the cost of the grid generation is negligible especially in non-deformable body problems and only the points at the vicinity of the front need a special treatment [46, 121]. Moreover, Gilmanov and Sotiropoulos developed a hybrid Cartesian/IBM solver for 3D complex flows [49]. This versatility and capacity of operating with a variety of discretization methods make the IBM to be an appropriate solution strategy in a wide range of applications. Some successful and popular explications of IBM include the simulation of the cardiac mechanics, locomotion of aquatic animals [41, 43], sperm motility [42], 2D and 3D models of bacterial swimming, wave propagation in the cochlea [16] and spiral cochlear geometry [51], flow with suspensions of the particles and fibers [45,138,140], model of viscoelastic networks [19], and turbulent flow simulations [39].

The IBM was first used to simulate flow patterns around heart valves in the Peskin's landmark paper [115]. The leaflet of a human heart valve is an example for the flexible immersed boundaries that move with the fluid and exert forces on the flow. The key ingredient of the method is to replace the boundary with a force field defined on grid points of a rectangular domain. In Peskin's original IBM, the boundary is assumed to have no mass, or neutrally buoyant and there is the same fluid on each side of the boundary. As a result, the force field has the same effects as the physical boundary. This force is defined as a function of the position of the boundary. Peskin defined a force density in such way that the force field is singular and zero everywhere except along the boundary. Since the force is determined by the boundary configuration, it depends on the elastic properties of the interface. In this formulation, fluid domain is initialized on the fixed Eulerian grid, but forces are matched with the Lagrangian force points. Peskin extended the applicability spectrum of the algorithm with the Chorin's projection scheme [29] that contains the prediction and correction steps. Zhu and Peskin [168] came up with another IBM for the flow of a soap film between two nylon wires. The new IB method included an

added mass density term but the variations of the mass density made fast-Fourier transfer methods inapplicable. Therefore, a multigrid method was used to solve pressure Poisson equation. The algorithm was validated using the experimental data [164].

In 1999, Roma et al. [127] combined the IBM with a adaptive mesh refinement technique. The implicit formulation of IBM is improved and combined with a multi-level self adaptive grid on the locally refined meshes. They introduced a new computational method that employs an encapsulated adaptive mesh refinement [15,74,98]. In this implementation, they covered the vicinity of the immersed boundary with the finer rectangular grid patches that follow the boundary motions and each mesh communicated with each other to transfer properties. Elastic force density was a function of the unit tangent to the position and tension of the immersed boundary in this new method.

Peskin and McQueen [119] observed that a second order fluid solver only performs a first order behavior at the grid points near the boundary, which requires to increase the resolution of the grid at the vicinity of the immersed boundary. As a remedy to this problem, Berger [13] employed a composite grid which covers the refined regions with respect to their rankings. In this formulation, the physical variables such as pressure and velocity are located in a marker-and-cell staggered grid fashion. In addition, the Dirac-delta function used transfer quantities between Eulerian and Lagrangian grids is defined different than the Peskin's original method by the reducing the bandwidth of distribution from 4 to 3 points. Fauci and Peskin [43] developed a model for the aquatic organisms as a massless object with elastic boundary immersed in a viscous incompressible fluid. Fauci [40] used a similar numerical approximation for the boundary segment as used in the Peskin's original method. The method was then extended for the swimming sheet problem, and also applied to some other problems in 2D. Beyer [16] used the IBM to model the cochlea in the inner ear. This computational model can be seen as a modified version of Peskin's original method in 2D with the main difference that the force calculations are performed semi-implicitly. In addition, a fully implicit version of this algorithm has been developed later for the similar problem where the outer boundary is assumed

to be rigid while inside membranes are flexible. Beyer [17] also used the IBM to solve one dimensional heat transfer using a Crank-Nicholson algorithm on a uniform Cartesian grid. In 2002, Peskin [118] presented an immersed boundary method in which the temporal discretization is done by a second order Runge-Kutta method and the Eulerian variables are located on a fixed Cartesian mesh while the Lagrangian variables are located on a curvilinear mesh that moves in the same way as the original IBM. In this formulation, instead of mass density, an elastic (potential) energy is stored in the material at each time step.

In the all papers mentioned, the motion of the embedded body depends on the flow field and boundary configuration, and only the forces between artificial spring elements are known. These methods are generally called the *continuous forcing method*. In this approach, the solid boundaries are modeled as a product of very rigid connected springs, as a result, the force term depends on two free constants. The free constants, α and β as used in the literature need to be tuned according to the flow. Mohd-Yusof [99] understood the main drawbacks of the method developed by Peskin and McQueen [119] and its variants [56], and proposed a new IBM that makes the immersed force independent of these two free constants. Fadlun et al. [36] performed the first example of the full 3D simulations in complex flow with moving boundaries by this new IBM formulation called the *direct forcing method*. Fadlun et al. [36] demonstrated that the new method is a second order accurate and highly efficient in computing the laminar flow over a 3D ribbed channel.

Mittal and Iaccarino [96] reviewed IBM comprehensively with all the formulations including the *continuous* and the *direct forcing* algorithms. They also discussed the some drawbacks of IBM such as difficulties of imposing the boundary conditions and remedies for these deficiencies. They reviewed the applications of IBM to a broad spectrum of flow problems including the flow past flapping filaments [164, 168], the flow past a pick-up truck [67], the flow past a sphere [84], and bodies in free fall [92, 97].

In 2008, Mittal et al. [95] introduced a sharp interface immersed boundary method which employs a ghost-cell methodology in addition to the *direct forcing* method to handle highly complex shaped stationary or moving bodies. The method

shares the same fundamental ideas with the ghost-cell method of Ghias [48].

Since the boundary moves and deforms, and the flow field is generally unsteady, the force and the position of the boundary are functions of time. Feedback forcing is the method developed by Goldstein et al. [54,55,130] to address these issues. In this method, the forcing term contains two negative constants α and β, and they multiply the velocity difference between the velocity of the boundary and the velocity field of the flow. This force density is a feedback to the velocity difference between the flow field and solid object. It pushes the particle velocity to be equal to the flow velocity at the front, i.e., when the velocity on the boundary is different from that of the flow, the forcing is switched on. However, big values of α and β make the forcing equation too stiff making the time step extremely small and thus simulations are too expensive. Therefore, Goldstein et al. [54] had to restrict the time step to keep the CFL number constant at a finite value. This limitation renders the feedback forcing method too expensive for the fully three-dimensional multiphase flows. In the *direct forcing IBM*, a virtual force field that contains no free constants, is defined. Fadlun et al. [36]published a landmark paper about this method. The *continuous* and *direct forcing* methodologies are briefly described below.

1.3.1 The Continuous Forcing Immersed Boundary Method

In the *continuous forcing* approach, a suitable forcing function is applied to the entire computational domain, then flow equations are discretized and solved with this additional force term. This method is also known as *feedback forcing* IBM is a version of *continuous forcing* formulation. This method has versions for elastic boundaries, rigid boundaries, Lagrange multiplier methods and immersed interface method. In the Peskin's original method, the fluid-solid interface is represented as a set of massless elastic fibers and the boundary force is determined by the interface configuration according to the Hooke's law at every time step [86, 115, 117]. Goldstein et al. [54,55] have used various versions of *continuous forcing* approach to simulate objects with rigid boundaries. The *virtual boundary method* also belongs to the continuous forcing IBM and its distinctive feature is to treat the interface as

a fictitious boundary buried into the fluid region and desired body force depends on free constants. It has been found that the selection of the free constants is not easy and inappropriate selection may create oscillations and numerical stability [66, 139]. Khadra et al. [83] developed another formulation for the rigid boundaries based on the assumption that the entire simulation is done on a porous medium governed by the Navier-Stokes-Brinkman equations [23]. In this method, the force term is determined according to the permeability of the medium which also serves as a color function used to set the viscosity and velocity fields. When the corresponding free constants are selected appropriately, this method can be made identical to the *feedback forcing* method. Even though the continuous forcing methods are successfull for elastic boundaries, time step restrictions and unstable behaviors of force term make them burdensome for moving rigid body problems. Especially for high Reynolds numbers, this method poses requirement of increasing the number of grid points, and it may result in inaccurate representation of the sharp interface otherwise [96].

1.3.2 The Direct Forcing Immersed Boundary Method

In the *direct forcing* or *discrete forcing* IBM, a non-physical force is added to the flow momentum equations. Because of its simplicity and efficiency, this method is the most convenient approach at the present and unlike the *feedback forcing*, it doesn't require any free constants or parameters [75]. Since it doesn't have the free parameters such as α or β, it has no additional stability restrictions but is highly sensitive to the discretization method used in the numerical algorithm. Over the years, this method has undergone some modifications depending on the physical problems. Verzicco [39, 66, 157] utilized this approach to simulate various flows in 2D and 3D including turbulent flows about bluff bodies. Tseng and Ferziger [148] developed an efficient ghost-cell based IBM in the large-eddy simulations (LES) framework and successfully applied it to simulate complex ocean flows. Balaras [11] performed LES for the flow past 2D and 3D bodies, and also modeled solidification problems. His scheme includes a novel interpolation procedure for solid boundaries and it has been validated for the Stokes flow around a circular cylinder. Later, Balaras and

Vanella [156] combined the solver with an adaptive mesh refinement method and applied the new solver to study a wide range of fluid-structure interaction problems in a similar nested mesh formulation with Roma et al. [127]. Griffith [57] also developed a new adaptive IBM with second order accuracy for three dimensional blood flow analysis in the heart. Zhang and Zheng [166] improved the extrapolation and the interpolation schemes of [164] and used the *direct forcing* concept of [75] to achieve a second order accuracy. Uhlmann [152] published his landmark paper about the *direct forcing* approach in 2005. In this method, the solid particles are assumed to contain the same fluid as the continuous phase but the no-slip boundary conditions are imposed at the fluid-solid interface using the direct forcing method and the Newton-Euler equations are solved for the rigid body dynamics to update the position of the rigid bodies. Later, this method has been successfully applied to simulate the heavy-particles in laminar and turbulent flows [153, 154]. Breugem [21] used the Uhlmann's method as an elementary idea for the particle-laden flows and modified it for a better approximation of the boundary conditions, internal treatment of particle, and enhancement of numerical stability. The method has been also extended for porous media problems with a *penalization direct forcing* model [22], and a soft collision model was added to account for inter-particle and particle-wall collisions [20]. Costa [31] improved the collision model using the lubrication theory and parallelized the code, and then applied it to simulate particle-laden turbulent flows using more than 1000 fully-resolved particles. Wang et al. [158] used a *multi direct forcing* scheme to impose the no-slip and no-penetration conditions. They used an inner iteration to obtain the correct force field that gives the desired velocity on the Lagrangian points. Breugem et al. [8, 120] employed an analogous method to simulate several flows. Kempe and Fröhlich improved the IBM in several aspects including the cost reduction of Eulerin-Lagrangian transfer operations for a better imposition [82]. They used three-point version of the Roma's regularized delta function and the Lagrangian force points are distributed unevenly in contrast with the Uhlmann's algorithm where the points are distributed evenly. Deen et al. [32] combined the IBM with the front-tracking method for moving deformable and non-deformable particles, which took roots from the front tracking algorithm

of van Sint Annaland and IB model of van der Hoef et al. [63], and a hard sphere discrete particle approach was employed for the particle-particle interactions. Yang and Stern [161] took the advantage of combining the *direct forcing* with the fractional step method and taking the fluid solver part out of the predictor-corrector step. In addition, they investigated different complex geometries including rectangular plates and bodies in fluttering and tumbling motions. Some other considerable *discrete forcing* applications with finite-element, finite-volume and various other methods have been performed by Ikeno [68], Choi [28], Mark [93], Munjiza [73].

1.4 Outline of the Present Study

In this study, a direct forcing method is combined with a front-tracking algorithm to simulate dynamics and interaction of rigid particles dispersed in a continuous fluid. The method is developed entirely in the front-tracking framework so that it can be readily used to simulate three phase flows involving both immiscible fluid particles (bubbles or droplets) and solid particles dispersed in a continuous fluid. This thesis is organized as follows: In Chapter 2, the mathematical formulation of the flow equations and rigid body dynamics is explained in detail. In Chapter 3 the flow solver, IBM, and transfer operations between Eulerian grid and Lagrangian grid are discussed. The present IBM/FTM solver is validated for several test cases and the results are presented and discussed in Chapter 4 where the preliminary results for a particulate flow involving many rigid spherical particles are also presented. Finally some conclusions are drawn and the possible future extensions of this work are outlined in Chapter 5.

Chapter 2

MATHEMATICAL FORMULATION

In this section, the flow and solid body dynamics equations are discussed. The fluid flow is governed by the incompressible Navier-Stokes equations while the rigid body dynamics is modeled by the Newton-Euler equations. Flow equations are described here in the context of IBM/FTM.

2.1 Flow Equations

Since the ultimate goal is to simulate fluid-fluid-solid three phase flows, following Tryggvason and Unverdi [146, 155], the one-field formulation is used for the flow equations. In this formulation, a single set of flow equations is written for the entire computational domain with variable material properties in each phase and the surface tension is included into the momentum equations as a source term acting at the fluid-fluid interface using a three-dimensional Dirac delta function. In the front-tracking framework, a color function (also called the indicator function), I, is employed to distinguish domains occupied by different phases. In a two-phase system, the indicator function varies between zero and unity, and the front is located at $I = 0.5$ contour. Note that the indicator function is constructed based on the location of the front as will be explained later. The indicator function can be defined as

$$I(x, y, z, t) = \int_V \delta(x - x')\delta(y - y')\delta(z - z')dV \tag{2.1a}$$

where δ is a three-dimensional Dirac delta function, V is the volume, and x', y', z' are the points where the function is evaluated. In the two-phase fluid-solid system, the indicator function can be given by

$$I(x, y, z, t) = \begin{cases} 0 & solid \\ 1 & fluid. \end{cases} \tag{2.1b}$$

Once the indicator function is determined, then the material properties are set in each phase, for instance, the density field is computed as

$$\rho(x, y, z, t) = \rho_f I(x, y, z, t) + \rho_p(1 - I(x, y, z, t)), \tag{2.2}$$

where ρ_f and ρ_p are the fluid and solid densities, respectively. Note that the viscosity field is defined only in the fluid domain. In the present IBM formulation, the flow equations are solved in the entire computational domain using a uniform viscosity field, i.e., the viscosity in the solid domain is set equal to the fluid viscosity, but the force field is constructed such that the no-slip and no-penetration boundary conditions are imposed at the fluid-solid interface. The flow inside the solid domain does not have any physical meaning. The material properties remain constant following a fluid particle, therefore density and viscosity evolve by

$$\frac{D\rho}{Dt} = 0, \quad \frac{D\mu}{Dt} = 0, \tag{2.3}$$

where D/Dt is the material derivative. The mass conservation equation is given by

$$\nabla \cdot \mathbf{u} = 0 \tag{2.4}$$

The Navier-Stokes equations can be written for a Newtonian fluid as

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

where $\mathbf{u}$ is the velocity field, p is the pressure, μ is the viscosity, $\mathbf{f}_b$ is the body forces and the last term $\mathbf{f}$ is the immersed boundary force. In the case of fluid-fluid interactions, an additional surface tension force can be added to Eq. 2.5 as done in [106, 146].

2.2 Rigid Body Dynamics

Various methods have been proposed to treat the boundary conditions at the fluid-solid boundaries. In some studies, solid particles are allowed to move with the fluid flow. In the present formulation, we use the direct forcing method developed by Uhlmann [152] in which the solid spheres obeys their own linear and angular

momentum equations known as the Newton-Euler equations. The Navier-Stokes equations are solved fully coupled with the Newton-Euler equations. For the p^{th} solid particle, the Newton-Euler equations can be written as

$$m_p \frac{d\mathbf{U}_c}{dt} = \oint_{\partial V} \boldsymbol{\sigma} \cdot \mathbf{n} dA + (\rho_p - \rho_f) V_p \mathbf{g} + \mathbf{F}_c \tag{2.6a}$$

$$I_p \frac{d\boldsymbol{\Omega}}{dt} = \oint_{\partial V} \mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) dA + \mathbf{T}_c \tag{2.6b}$$

where $\mathbf{U}_c$ and $\boldsymbol{\Omega}$ denote the translational velocity and the rotational velocity vectors of the particle, respectively, $\boldsymbol{\sigma} = -p\mathbf{I} + \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$ is the stress tensor exerted by the fluid on the solid particles, $\mathbf{g}$ is the gravitational acceleration, $\mathbf{r} = \mathbf{r} = (\mathbf{x} - \mathbf{X}_c)$ is the position vector with respect to the centroid of the particle and $\mathbf{n}$ is the unit vector normal. In a many particle system, the particle-particle interactions result in the collision force, $\mathbf{F}_c$, and the collision torque $\mathbf{T}_c$. In the present study, the particle-particle interactions are ignored, i.e., $\mathbf{F}_c = 0$, $\mathbf{T}_c = 0$. The Navier-Stokes equations (Eqs. 2.4, 2.5) and the Newton-Euler equations (Eqs. 2.6a-2.6b) are solved fully coupled. The no-slip boundary conditions require that the flow velocity at the solid-fluid interface is given by

$$\mathbf{u}_f = \mathbf{u}_p = \mathbf{U}_c + \boldsymbol{\Omega} \times \mathbf{r}, \tag{2.7}$$

where $\mathbf{u}_p$ is the particle velocity used to compute the virtual force field using the IBM as discussed in the next chapter.

Chapter 3

NUMERICAL METHOD

The ultimate goal of the present study is to develop a high-fidelity numerical method for efficient simulations of fluid-fluid-solid three-phase flows. For this purpose, we aim to develop a direct-forcing method of Uhlmann [152] in the framework of the front-tracking method [107, 146]. The flow solver is essentially the same as that described by Muradoglu and Tryggvason [107]. The numerical method including the flow solver is described here for completeness but emphasis is placed on the new ingredient, the direct-forcing immersed boundary method.

3.1 Flow Solver

The flow equations are discretized on a Cartesian staggered grid on which the scalar fields are located at the cell centers while the velocity components are located at the cell faces. For the spatial discretization, second order central differences are used except for the convective terms for which a third order QUICK scheme [87] is used. The mass and momentum conservation equations of the flow are solved via a modified version of Chorin's projection method [29]. In the Chorin's original method, the time integration is divided into two sub-steps. In the first step, the effects of pressure are ignored and the Navier-Stokes equations are integrated in time to obtain the unprojected velocity field $\mathbf{u}^*$ that does not satisfy the mass conservation. In the second step, the mass conservation is used to compute the pressure field, and then the velocity field is corrected by adding the effects of the pressure field to obtain the physical velocity field the new time level $\mathbf{u}^{n+1}$ that satisfies both the momentum and mass conservation equations. The superscript n denotes the n^{th} time level. However, there is an additional time step in our formulation when the immersed boundary force is added into the unprojected velocity. The discretized forms of the

momentum and mass conservation equations can be written as

$$\frac{\rho^n \mathbf{u}^{n+1} - \rho^n \mathbf{u}^n}{\Delta t} = \mathbf{rhs}^n - \nabla p + \mathbf{f}^{n+1/2}, \tag{3.1a}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \tag{3.1b}$$

where **rhs** denotes the contributions from the advective, the diffusive and the body force terms. In the original projection method, Eq. 3.1a is divided into two steps as

$$\frac{\rho^n \mathbf{u}^{*} - \rho^n \mathbf{u}^n}{\Delta t} = \mathbf{rhs}^n + \mathbf{f}^n, \tag{3.2a}$$

$$\frac{\rho^n \mathbf{u}^{n+1} - \rho^n \mathbf{u}^{*}}{\Delta t} = -\nabla_h p^{n+1}, \tag{3.2b}$$

where $\mathbf{u}^*$ is the temporary unprojected velocity field. In the present study, the projection method is slightly modified to accommodate the IBM. In the modified projection method, Eq. 3.2a is further divided into two substeps as

$$\frac{\rho^n \mathbf{u}^{*} - \rho^n \mathbf{u}^n}{\Delta t} = \mathbf{rhs}^n, \tag{3.3a}$$

$$\frac{\rho^n \mathbf{u}^{**} - \rho^n \mathbf{u}^{*}}{\Delta t} = \mathbf{f}^n, \tag{3.3b}$$

where $\mathbf{u}^*$ and $\mathbf{u}^{**}$ are the provisional velocities that do not satisfy the mass conservation equation. The approximation of the force field $\mathbf{f}^n$ is discussed in the next section. Finally, the effects of the pressure are added as

$$\frac{\rho^n \mathbf{u}^{n+1} - \rho^n \mathbf{u}^{**}}{\Delta t} = -\nabla p. \tag{3.4}$$

Taking divergence of Eq. 3.4 and using the incompressibility condition $\nabla \cdot \mathbf{u}^{n+1} = 0$, we get the pressure Poisson equation given by

$$\nabla \cdot \frac{1}{\rho^n} \nabla p = -\frac{1}{\Delta t} \nabla \cdot \mathbf{u}^{**}, \tag{3.5}$$

which is solved using HYPRE library [37]. Once the pressure field is obtained, the velocity field is corrected as

$$\mathbf{u}^{n+1} = \mathbf{u}^{**} - \frac{\Delta t}{\rho^n} \nabla p, \tag{3.6}$$

where $\mathbf{u}^{n+1}$ is the projected velocity that satisfies both the mass and momentum equations at the new time level.

The flow equations are discretized on a Cartesian staggered grid on which the scalar fields are located at the cell centers while the velocity components are located at the cell faces. The method described above is only first order accurate in time but a second order time accuracy is recovered by using the trapezoidal rule as described by Tryggvason et al. [146].

3.2 Immersed Boundary Method

In the present scheme, the direct forcing method [152] is adapted to be used in the front-tracking framework such that it can be readily combined with the front-tracking method designed for the fluid-fluid multiphase flow simulations. The main idea is to enforce the desired velocity at the selected grid nodes that coincide with the surface of the immersed object or front as described below. Since we use Lagrangian marker points to create the front and the Eulerian grid is used for solving the flow equations, interpolation and distribution schemes are required to transfer variables between the Eulerian grid and Lagrangian grids. The force term $\mathbf{f}$ is computed on the Lagrangian grid and it needs to be distributed onto the Eulerian grid to be added to the flow momentum equations in a conservative manner. In addition, the particle satisfies the rigid body dynamics, as a result, its position with respect to the domain should be updated at each time step. This operation is done on the Lagrangian elements but the marker points and consequently the interface are advanced in a similar way with the original FTM in *advancefront* subroutine. An elaborated summary of the code can be found in *Appendix*.

Using an explicit Euler method, the time-discretized momentum equation can be written as

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\Delta t} = \mathbf{rhs}^{n} + \mathbf{f}^{n}. \tag{3.7}$$

In Eq. 3.7, $\mathbf{rhs}^{n}$ includes the convective, the viscous and pressure terms. Since the flow velocity must match the velocity of the solid boundary, the force term can

be computed as

$$\mathbf{f}^n = \frac{\mathbf{u}^d - \mathbf{u}^n}{\Delta t} - \mathbf{rhs}^n, \tag{3.8}$$

where $\mathbf{u}^d$ is the desired velocity computed at the front element centers using Eq. 3.11. For the sake of simplicity, ther terms at the right hand side can be written as

$$\tilde{\boldsymbol{u}} = \mathbf{u}^n - \Delta t \mathbf{rhs}^n. \tag{3.9}$$

In each time step this $\tilde{\boldsymbol{u}}$ is evaluated on the Eulerian grid and interpolated onto the Lagrangian element centers. The transfer operation in both directions will be discussed in Section 3.3. The force acts only on the fluid-solid interface but is smoothed onto the neighboring Eulerian grid cells using a distribution function that approximates the Dirac delta function. In addition to that, the net force and moment acting on the rigid sphere should be known to solve the Newton-Euler equations. The force term $\mathbf{F}$ is first evaluated at each element center and then distributed on the neighboring Eulerian grid using Peskin distribution function in a similar fashion as done for the surface tension in the fluid-fluid multiphase flow simulations. Even though it is harder to transfer quantities from the Eulerian grid to the Lagrangian grid [122] and more vulnerable to loss of accuracy during the operation, it is compulsory for our combined solver due to fluid-solid coupling. The force at the interface is computed as

$$\mathbf{F}^n = \frac{\mathbf{U}^d - \tilde{\boldsymbol{U}}}{\Delta t} \tag{3.10}$$

In Equation 3.10 and henceforth, the upper case letters denote the Lagrangian variables while lower case letters are used for the Eulerian variables. Here, $\mathbf{U}^d$ is the desired particle velocity on the Lagrangian element centers of the immersed boundary and is computed as

$$\mathbf{U}^d(\mathbf{X}_l) = \mathbf{U}_c + \boldsymbol{\Omega} \times (\mathbf{X}_l - \mathbf{X}_c). \tag{3.11}$$

In Equation 3.11, $\mathbf{X}_l$ is the coordinates of the corresponding marker element in vector notation, and $\mathbf{U}_c, \boldsymbol{\Omega}$, and $\mathbf{X}_c$ are the translational velocity, the rotational velocity and the coordinates of the centroid of the embedded body, respectively. Since Eq. 3.11

is satisfied on the Lagrangian elements, all of the variables are denoted by capital letters. In the present work, the body is rigid and non-deformable, therefore $\mathbf{X}_l - \mathbf{X}_c$ remains constant. To calculate the desired velocity $\mathbf{U}^d$, linear and angular velocities of the particle should be updated at each time step, which is done as follows:

$$\mathbf{U}_c^{n+1} = \mathbf{U}_c^n - \frac{\Delta t}{V_p}\frac{\rho_f}{\rho_p}\sum \mathbf{F}^n \Delta V_l + \frac{1}{V_p}\frac{\rho_f}{\rho_p}\left(\int_{V_{pi}} \mathbf{u}^{n+1} dV^{n+1} - \int_{V_{pi}} \mathbf{u}^n dV^n\right) + \Delta t\left(1 - \frac{\rho_f}{\rho_p}\right)\mathbf{g} + \Delta t \frac{{F_c}^{n+1} + F_c^n}{\rho_p V_p} \tag{3.12a}$$

$$\mathbf{\Omega}_c^{n+1} = \mathbf{\Omega}_c^n - \Delta t \frac{\rho_f}{I_p}\sum \mathbf{r}_l \times \mathbf{F}^n \Delta V_l + \frac{\rho_f}{I_p}\left(\int_{V_{pi}} \mathbf{r} \times \mathbf{u}^{n+1} dV^{n+1} - \int_{V_{pi}} \mathbf{r} \times \mathbf{u}^n dV^n\right) + \Delta t \frac{{T_c}^{n+1} + T_c^n}{I_p} \tag{3.12b}$$

Equations 3.12a and 3.12b denote the linear and angular momentum conservation of the rigid particle or front in our terms. 2nd terms at these equations are the opposites of IBM forces and torques exerted on the interface which are equal to 0 elsewhere, 3rd terms are the inertia of fluid inside the particle where V_{pi} is the volume inside the particle and its detailed calculations can be found in *Appendix*. However, since Uhlmann [152, 154] observed that doesn't affect much we consider it as 0 and didn't apply any special treatment inside the particle. The next two correspond to buoyancy and particle-particle collision terms. In this work, only single particles are considered or in many particle systems interactions aren't taken into account therefore the last term is also 0 but it will be useful in the future fluid-fluid-solid interaction problems.

3.3 Transfer Operations

One of the key ingredients and distinguishing common features of FTM and IBM is the need to exchange quantities between the Eulerian and Lagrangian grids. This is achieved by means of regularized delta functions in a similar fashion as the Peskin's original distribution function [116, 118] to interpolate and spread quantities between Eulerian and Lagrangian grids. For instance, the predicted velocity, $\tilde{\boldsymbol{u}}$, is transferred

from the Eulerian grid to the Lagrangian grid to compute the force at the element centers. Then, the force is smoothed back onto the neighboring Cartesian grid point to be added to the flow momentum equations as body force. The torque is only needed in the rigid body dynamics, so there is no need to transfer it to the Eulerian grid. The smoothing operations can be considered as an approximation of the δ function on the Eulerian grid because the interface is represented by a δ function. We need a function δ_h that approximates δ function in the discrete form, so that the predicted velocity is interpolated from the Eulerian grid onto the marker points, i.e.,

$$\tilde{\boldsymbol{U}}(\mathbf{X}_l) = \sum \tilde{\boldsymbol{u}}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}_l)h^3, \tag{3.13}$$

where h denotes the Eulerian grid spacing, $\mathbf{X}_l$ represents the position vector of the element centers on the interface and $\mathbf{x}$ represents the location of the velocity on the staggered Eulerian grid. After interpolating $\tilde{\boldsymbol{U}}$ onto Lagrangian grid, the force term is calculated. This force term is stored to be used in the Newton-Euler equations and is also distributed onto neighboring Eulerian grid points to enforce the boundary conditions at the fluid-solid interface. In Eq. 3.14, the immersed force is the sole quantity transferred to the Eulerian grid because the torque is not needed for the Navier-Stokes equations. Transfer operation of the force from the Lagrangian grid onto the Eulerian grid can be written as

$$\mathbf{f}(\mathbf{x}) = \sum^{N_p} \sum^{N_l} \mathbf{F}(\mathbf{X}_l)\delta_h(\mathbf{x} - \mathbf{X}_l)\Delta V_l \tag{3.14}$$

where N_p the total number of rigid particles, N_l is the number of the Lagrangian elements and ΔV_l is the volume of each Lagrangian element. ΔV_l is obtained by the multiplication of the element area with the unit grid size. The discrete delta function, δ_h, must have some desirable properties as summarized below:

- It is a smooth functions, i.e., it is differentiable everywhere.
- It is a compact support.
- It satisfies the divergence free condition when it is used to interpolate the incompressible velocity field.

- Its bandwidth is not small to avoid numerical instabilities and wiggles and not too large to avoid excessive numerical error.

- It approximates the Dirac delta function and converges to the Dirac delta function in the limit as the grid is refined, i.e., as $h \Rightarrow 0$. This property can be expressed mathematically as

$$\sum \delta_h(\mathbf{x} - \mathbf{X}_l)h^3 = 1, \tag{3.15}$$

$$\sum (\mathbf{x} - \mathbf{X}_l)\delta_h(\mathbf{x} - \mathbf{X}_l)h^3 = 0. \tag{3.16}$$

  Equations 3.15 and 3.16 make sure that the transfer operations are consistent and conservative, i.e., the total amount of force exerted on the fluid is the same as that computed at the interface. This condition is expressed as:

$$\sum \mathbf{f}(\mathbf{x})h^3 = \sum^{N_p} \sum^{N_l} \mathbf{F}(\mathbf{X}_l)\Delta V_l \tag{3.17}$$

$$\sum \mathbf{x} \times \mathbf{f}(\mathbf{x})h^3 = \sum^{N_p} \sum^{N_l} \mathbf{X}_l \times \mathbf{F}(\mathbf{X}_l)\Delta V_l \tag{3.18}$$

Three dimensional discrete delta function δ_h used for the transfer operations can be written as the product of three one dimensional delta functions as

$$\delta_h^{3D}(\mathbf{x} - \mathbf{X}) = \delta_h^{1D}(x - X)\delta_h^{1D}(y - Y)\delta_h^{1D}(z - Z). \tag{3.19}$$

The one-dimensional discrete delta functions is given by

$$\delta_h^{1D}(x - X) = \frac{1}{h}d(r) \tag{3.20}$$

where *d(r)* is a continuous distribution function also known as Peskin's cosine function defined as

$$d(r) = \begin{cases} (1/4h)(1 + \cos(\pi r/2h)) & |r| < 2h \\ 0 & |r| \geq 2h \end{cases} \tag{3.21}$$

In the parallel implementation, before the Eulerian to Lagrangian interpolations, point coordinates, element coordinates, position of the element centers, and the

position of the front centers are taken from the *Front* side, and $\tilde{\boldsymbol{u}}$ is interpolated onto the element centers. This information is transferred from the *Domain* to *Front*, and force calculation is done there. At spreading operation from the Lagrangian to the Eulerian grid, the force terms and element centers are sent back to *Domain* and entire process is repeated for every time step.

Chapter 4

RESULTS AND DISCUSSION

In this chapter, the results are presented and discussed. The method is first applied to simulate the uniform flow past a single stationary sphere to observe the qualitative behavior of the force vector and the vorticity field. In particular, the force and pressure distributions, and velocity field are examined inside and outside of the sphere. The method is then validated for the various test cases by comparing the computational results with the experimental data. Note that we only considered and presented 3D case in this study although the method can also be used to simulate 2D flows. Since the base solver is essentially the same as the FTM code of Muradoglu [106] and it ha been validated for a wide range of single and fluid-fluid multiphase flows, validation for single phase flow is not repeated here. For the validation of the IBM, simulations are performed for rigid particle sedimentation and the results are compared with the experimental results of Mordant and Pinton [102]. Additional validation simulations are also performed to simulate rigid sphere sedimentation for the cases of density ratio of around unity and for relatively low Re numbers, and the results are compared with the experimental data of Cate et al. [27]. Note that it has been reported in the previous studies that the density ratio close to the unity may cause some numerical difficulties. We aim to overcome these difficulties. Finally, the method is applied to simulate particulate flows involving many solid spherical particles to demonstrate the capability of the present algorithm for such cases albeit the simulations are done only for highly dilute concentrations to avoid particle-particle interactions as particle-particle interactions are not yet included in the present solver. We note, however, that a particle-particle interaction can be easily incorporated into the present solver.

A uniform flow past a stationary sphere is usually used to compute the drag force, F_D and compare it with the experimental data. However, instead of the drag

force, the general behavior of the force field and IBM solver are considered in this study to examine accuracy of the method qualitatively only. The flow configuration is similar to that used by Johnson and Patel [9, 50, 78]. We consider a uniform laminar flow pasts a sphere at varying Reynolds numbers from 50 to 300. The rigid solid sphere is placed at the center of the square domain and a uniform velocity is specified in z-direction at the inlet (the south boundary) and the pressure boundary conditions are applied at the outlet. The no-slip boundary conditions are applied on other boundaries. All the units are in the SI units. Figure 4.1 shows the constant contours of the IB force, *fzsolid*, in z direction. Note that the force is multiplied by 1000 for ease of interpretation and since we are only interested in how symmetric the force is distributed rather than its magnitude. As seen, the force is distributed in the vicinity of the interface and the distribution is quite symmetric showing the accuracy of the force distribution algorithm. Figure 4.2 shows the pressure distribution around the sphere. This figure clearly shows the expected symmetry of the pressure distribution with respect to the axial line passing through the center of the sphere, which can be considered a good indication for the accurate distribution of the force as well as accurate computation of the flow. The pressure distribution is qualitatively in good agreement with [137]. Figure 4.3 shows the velocity vectors in and around the stationary rigid sphere in a uniform flow. This case is similar to that studied by Mittal et al. [95] that has been widely studied both experimentally and computationally. The simulation is first performed for the Reynolds number, $Re = 100$ and then repeated for different Reynolds numbers. As also shown by Gilmanov et al. [50], the flow separation occurs behind the sphere as can be clearly seen in the velocity vector plot in Fig. 4.3. The wake region behind the sphere is also clearly seen in this figure. The vortices are formed in wake region and it is observed that the increasing Re results in shortening of the wake region. However, the horizontal distance in the wake region is not affected by the Reynolds number. These results are consistent with the computational results of Gilmanov et al. [50]. For the same case, the streamlines and velocity vectors are plotted in Fig. 4.4. The red arrows indicate the velocity vectors while black lines are the streamlines. Streamlines are created with random seeds issuing from the surface of the rigid

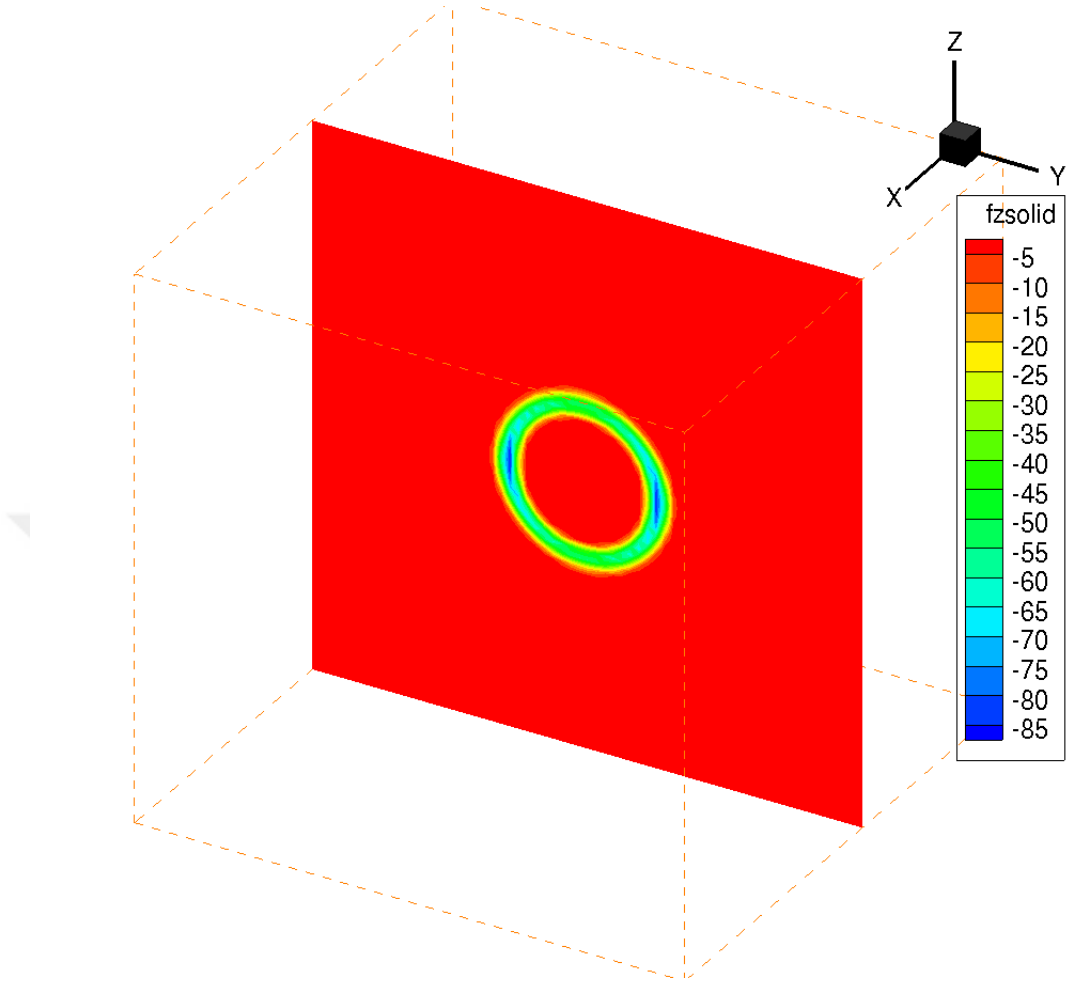

Figure 4.1: Immersed boundary force distribution around a spherical solid sphere in a uniform flow.

sphere. Total number of 50 random seeds are used to visualize the streamlines in TECPLOT. The deflection of the flow velocity is clearly seen in the wake region as well as the velocity vectors follow the streamlines. Figure 4.5 shows the flow inside the rigid solid sphere. As can be clearly seen in this figure, the flow velocity is zero at the solid-fluid interface showing that the no-slip boundary conditions are properly enforced on the surface of the sphere. As mentioned before, the flow inside the sphere is not physical and has no influence on the flow around the sphere.

Since the results in stationary sphere fulfill the aim of capturing the velocity field correctly and characteristics of wake region, and satisfaction of NS/NP, the dynamic motions of the rigid particle can be examined in the next section.

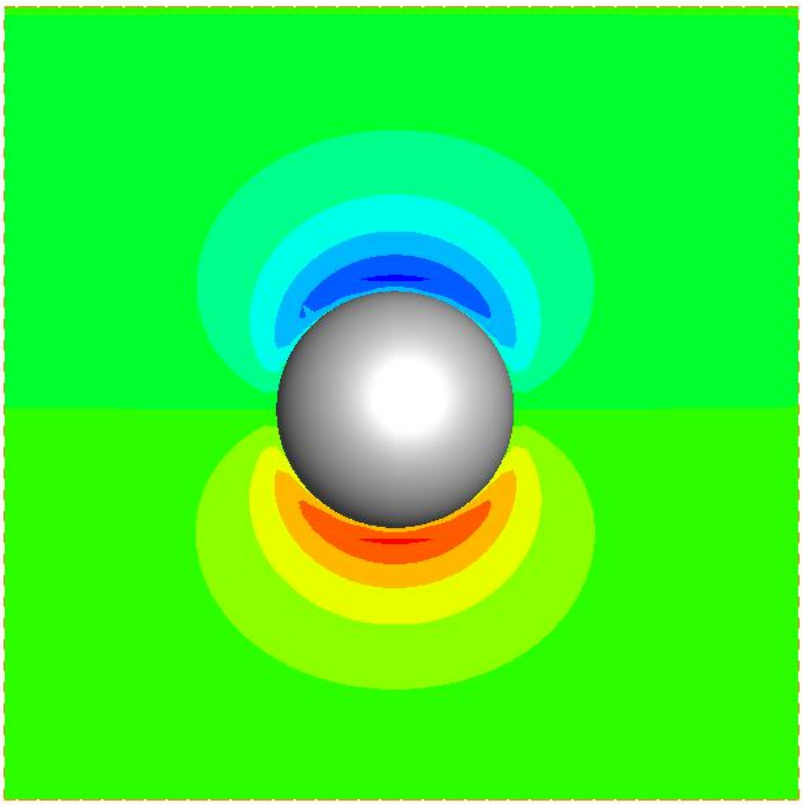

Figure 4.2: Pressure distribution around the stationary sphere in a uniform flow.

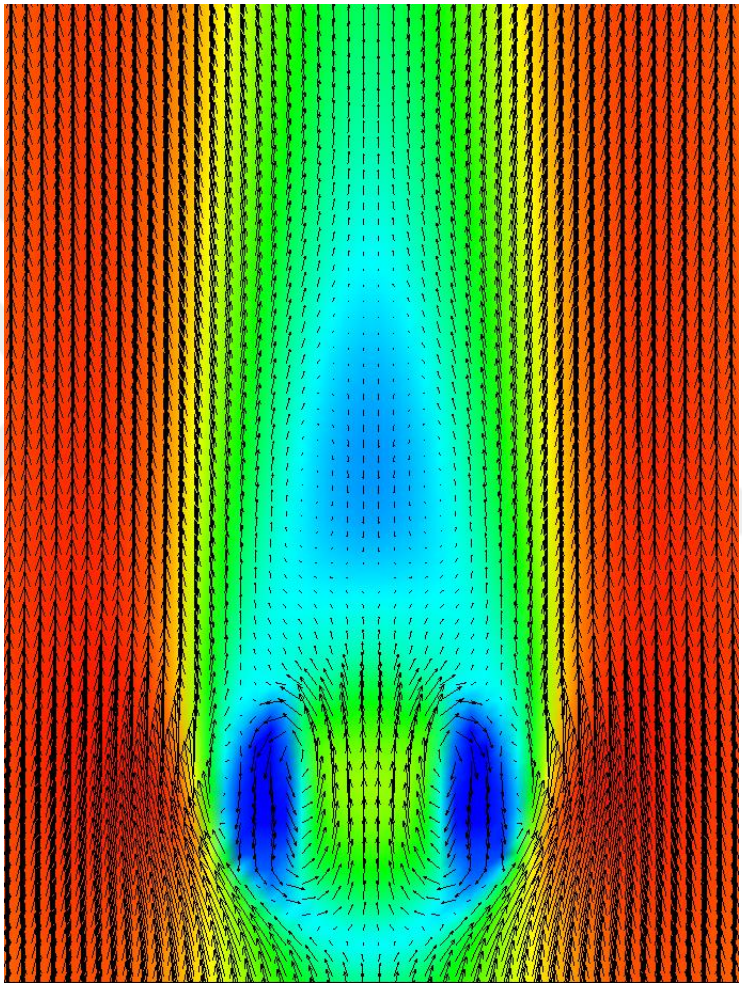

Figure 4.3: Vorticity and velocity fields around the sphere for $Re = 100$.

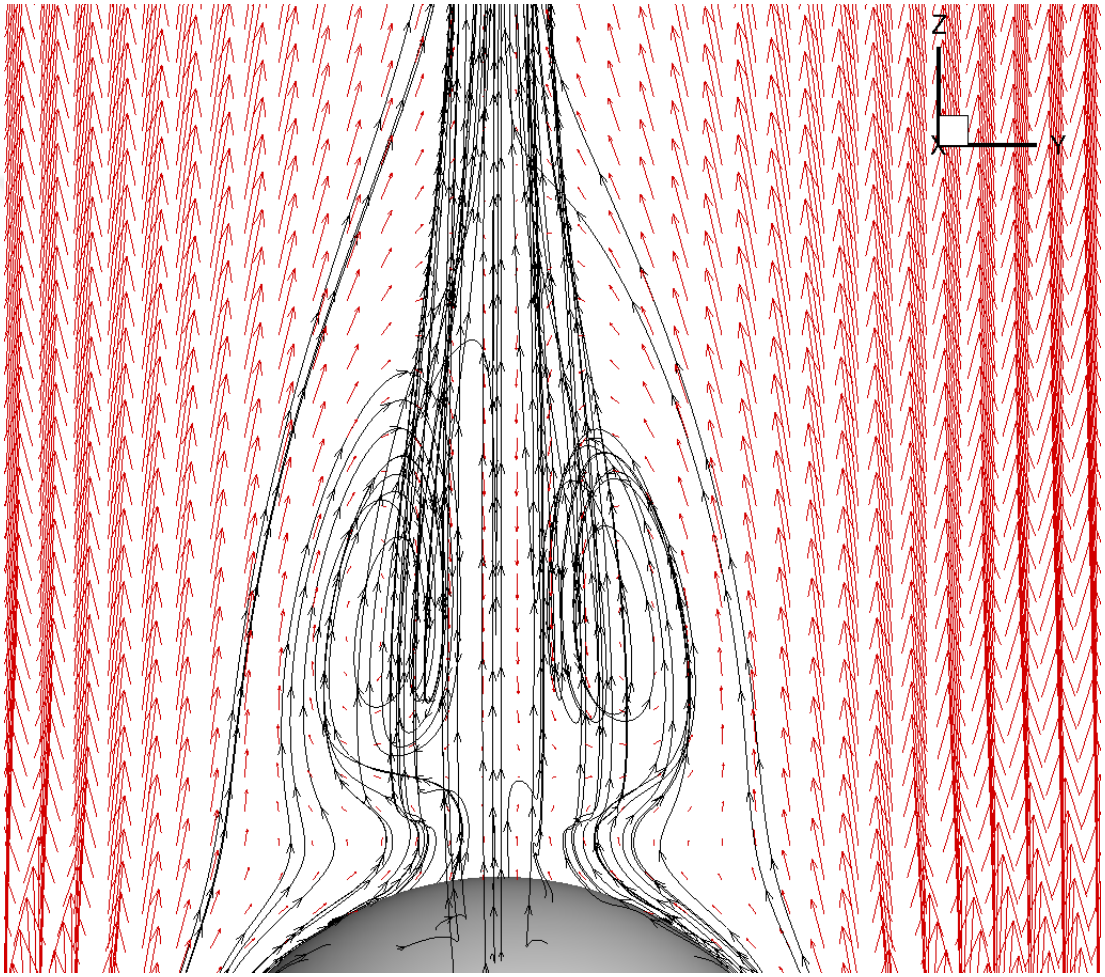

Figure 4.4: Streamlines around the stationary sold sphere in a uniform stream for $Re = 100$.

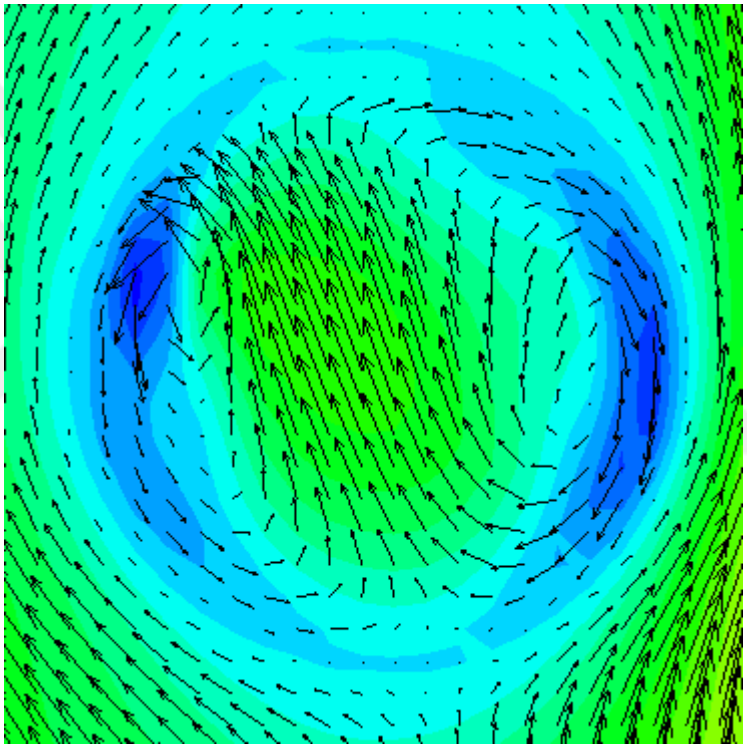

Figure 4.5: The flow field inside the solid rigid sphere for $Re = 100$. The flow inside the sphere is not physical and has no influence on the physical flow around the sphere.

4.1 A Single Sedimenting Sphere

In these 3 cases, a single rigid sphere is released and the sedimentation is observed and compared with Mordant and Pinton's experimental results [102]. Previously, these cases were simulated by Uhlmann [153] and initially the flow field and the particle are at rest. Then, particle is released by two tweezers without any velocity. In Mordant's experimental setup, ultrasonic waves were used to detect the velocity. The sedimentation is only under the influence of gravity and buoyancy as the driven forces at the experiment and also simulations. In addition, Mordant and Pinton compared their experimental data with a simulation by a Runge-Kutta based solver. In Figure 4.6, a zoomed image of the rigid spherical body is attached and triangular elements on the particle can be seen.

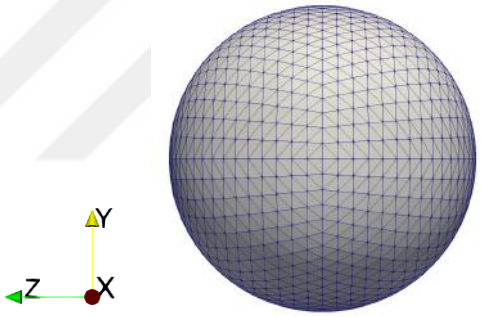

Figure 4.6: The rigid spherical body with triangular elements

In our test cases, all quantities associated with the particle are non-dimensionalized according to reference values. The reference velocity and time are given as respectively, $u_{ref} = \sqrt{|\mathbf{g}|D_p}$, $t_{ref} = \sqrt{D_p/|\mathbf{g}|}$ where D_p is the diameter of the particle. The flow field is periodic in all directions. The reason of using periodic boundaries is to create the effects of large container as the experiment. In the cases that no-slip walls are used in boundaries, the velocity of the particle was less than the periodic results which contradicts the objectives of this study. We aimed to complete development of the flow field with triply periodic boundaries, and prevent wall effects in horizontal plane. In cases density ratio, viscosity and channel length are changed to determine their effects on the particle velocity and the flow field. The particle Reynolds num-

bers, Re_p are kept constant to obtain a duplicate model of the experiments. Re_p is given as

$$Re_p = \frac{\mathbf{u}_p D_p}{\nu} \tag{4.1}$$

where $\mathbf{u}_p$ is the particle velocity. The physical properties of the fluid and solid sphere used in the experimental study by Mordant and Pinton [102] are summarized in Table 4.1. The domain length in y-direction, L_y is changed in Case 3 but in other directions it is held constant in all cases and domain corresponds to $\Omega = 1.25 \times L_y \times 1.25$. The particle position is $\mathbf{x}_c = 0.625 \times y_c \times 0.625$, where y_c corresponds to initial location of the particle center in y axis and 9.5 in Case 1 and 2, and 14.5 in the last case. The gravity is taken into consideration in its pyhsical value, $g_y = -9.81 m/s^2$. Particle diameter is given as $D_p = 1/6m$. Since the horizontal velocities are negligeble compared to vertical velocity component, grid points in y-direction where the motion occurs, is chosen more than the other directions. However, huge differences in number of grid points in different directions resulted in inaccuracies in terminal velocities.

Case Number	1	2	3
Density Ratio ρ_p/ρ_f	2.56	2.56	7.71
Domain Length $L_y[m]$	10	10	15
Kinematic Viscosity $\nu[m^2/s]$	0.005416368	0.00104238	0.00267626
Re_p	41.17	362.70	280.42

Table 4.1: The physical properties of the fluid and solid sphere used in the experimental study by Mordant and Pinton [102], and the domain length in y-direction.

In the following Figure 4.7, our results and experimental data are presented and found in a good agreement. The dashed and straight lines with the same color represent the same Re_p case for experiment and simulation results in Figure 4.7. The number of grid points in y-direction, N_y corresponds to 480, 512, and 512 in the given order for 3 cases and for other directions various grid sizes are used. The respective CFL numbers are 0.3 for $Re_p = 41.17$ and $Re_p = 362.70$ cases and 0.1 for

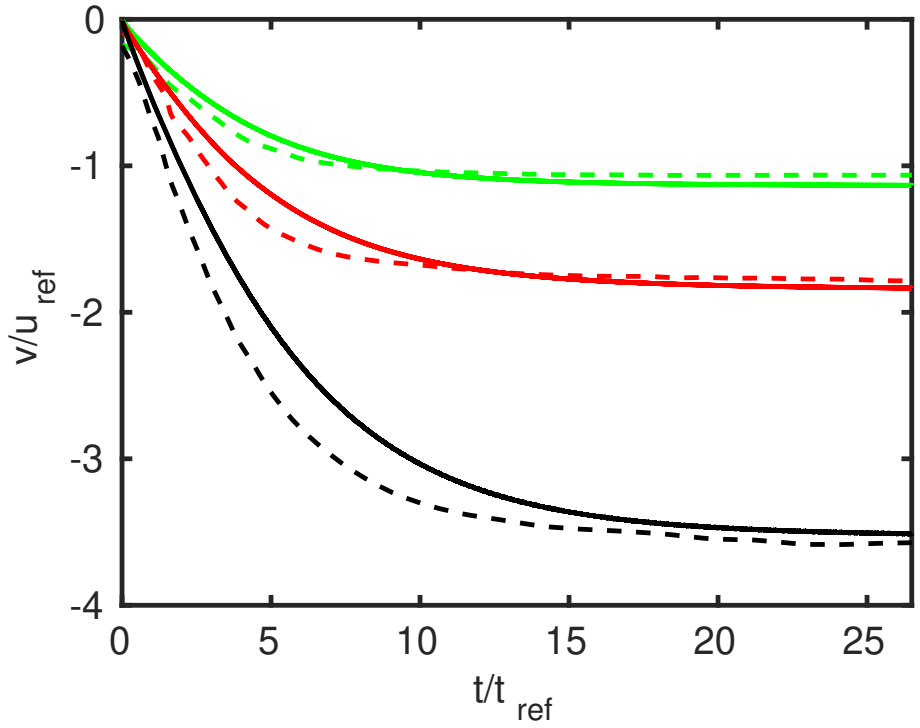

Figure 4.7: Change of vertical terminal velocity in time is investigated for $Re_p = 41.17$, $Re_p = 362.70$ and $Re_p = 280.42$ cases denoted with green, red, and black in order and compared with the experimental data. Dashed lines represent experiments and straight lines are used for present study for the same Re_p number

$Re_p = 280.42$ in the figure. In the presented solver, it is optional to select fixed CFL number or fixed time step, and all simulations are done with fixed CFL numbers which selects the necessary time step itself. In first case $Re_p = 41.17$, time step corresponds to 0.002 which is close to Uhlmann and Patankar's [132]. On the other hand, in Case 3 $\Delta t \approx 1.5 \times 10^{-3}$ at the beginning but in time it is diminished up to $\Delta t \approx 7.5 \times 10^{-4}$. For all 3 cases change of the horizontal velocity components are given in Figures 4.8 and 4.9. The blue, red and black lines represent the cases in numerical order. In $Re_p = 41.17$ case horizontal velocities, u and w didn't change significantly because of the lack of asymmetric vortex shedding. However, in other cases $Re_p = 362.70$ and $Re_p = 280.42$, the lateral motion can be observed from the shift in u/u_{ref} and w/u_{ref}. Horizontal velocities in both direction reach the $O(10^{-4})$. In Uhlmann's report [152], these variables are given with the ups and

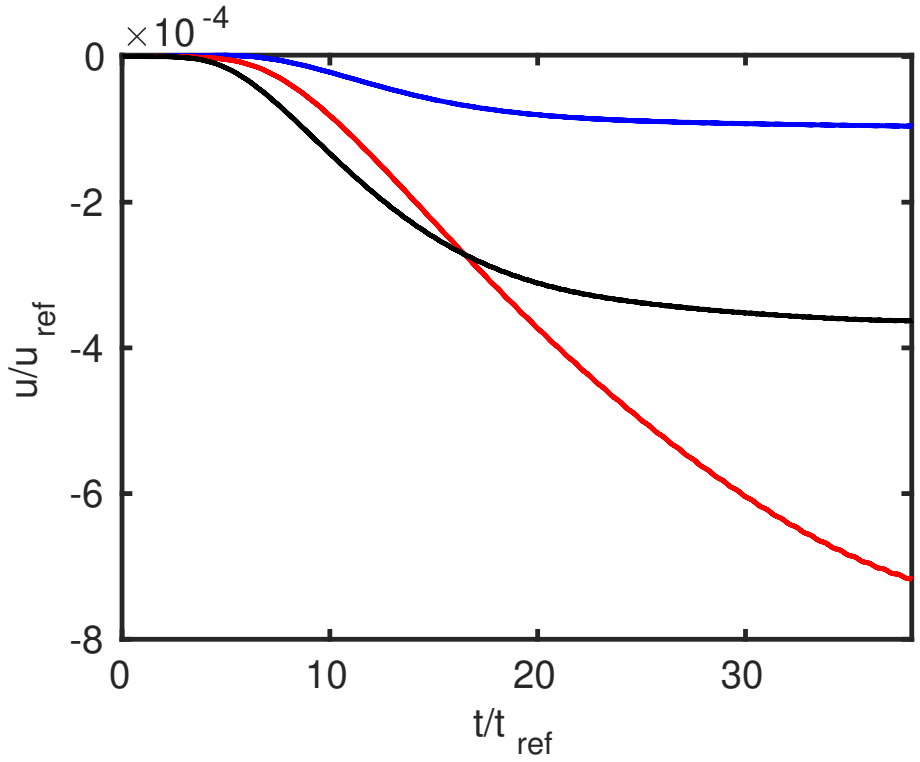

Figure 4.8: Horizontal velocity component u is non-dimensionalized and its change over dimensionless time is investigated for $Re_p = 41.17$, $Re_p = 362.70$, and $Re_p = 280.42$ that are represented with blue, red and black lines.

downs, however in the present study change in horizontal velocities are much stable and linear. Moreover, since the sphere is located symmetrically in x and z axis, the velocity components u/u_{ref} and w/u_{ref} evolved almost identically.

In Figure 4.10, first test case is tested and the transient behaviour of the particle is achieved to capture successfully. However, the difference between the terminal velocities can be seen although the error percentage is below 5%. The experimental outcomes are drawn again for comparision with the simulation results. To understand the grid effects, 128 and 480 as the grid numbers in y-direction are tried but for this case it doesn't provide a significant change. In both N_y number, the CFL number is fixed to 0.3. It can be seen that, the solver performs higher accuracy in higher particle Reynolds number cases in terms of terminal velocity. On the other hand, declines in viscosity affect the accuracy in transient region. One of the possible reasons is the internal treatment of the solid particle. The IBM force in only

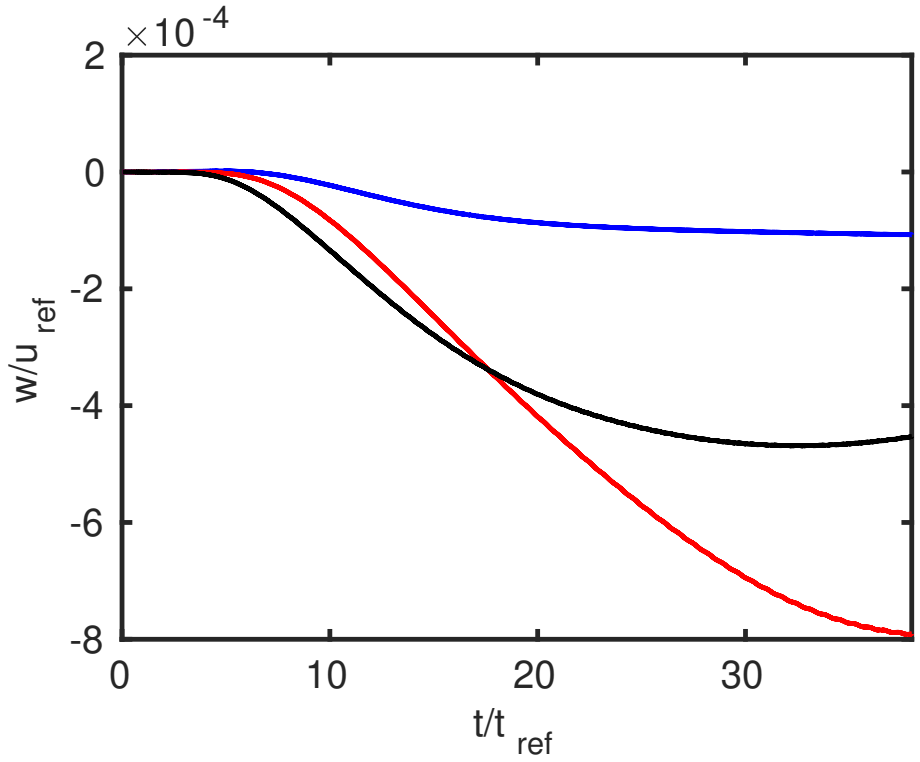

Figure 4.9: Horizontal velocity component w is non-dimensionalized and its change over dimensionless time is investigated for $Re_p = 41.17$, $Re_p = 362.70$, and $Re_p = 280.42$ that are represented with blue, red and black lines.

applied on the boundaries and inside of the rigid body isn't treated. At the low viscosity cases with higher density ratios such as $Re_p = 280.42$, the viscosities inside and outside of particle are same but density jump is higher which may lead to create this problem as observed in VPM studies mentioned in *Introduction*.

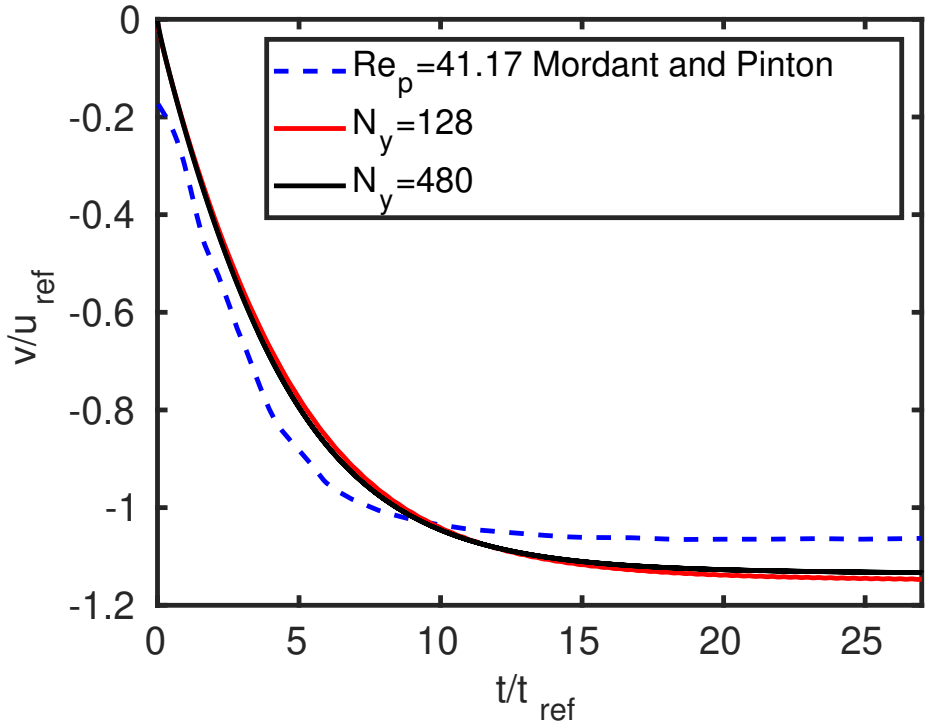

Figure 4.10: Effects of the grid size in y-direction for $Re_p = 41.17$ case on dimensionless vertical terminal velocity comparision with the experimental data over dimensionless time

As mentioned earlier, In Figure 4.11, $Re_p = 362.70$ case is simulated with different grid sizes in y-direction. The positive effect of increasing grid number is evident. It does not only change the accuracy of terminal velocity but also makes the velocity change near to the transition zone. The CFL number is 0.3, and the grid numbers N_x and N_z are kept constant at 100, which resulted in 8.5 grid to resolve particle which will be explained in the next Section. Before 100 as the grid number in x and z directions, less numbers were tried and it is seen that the increments in both direction resulted well in terms of accuracy but also altered the total computational time negatively. The error calculation in terminal velocity is given as

$$\epsilon = \frac{u_{tr} - v_{ter}}{u_{tr}} \tag{4.2}$$

where ϵ is the error, and u_{tr}, v_{ter} are the experimental terminal velocity and terminal velocity in simulations respectively. In Case 2, $\epsilon\% \approx 3$ which can be considered as satisfactory.

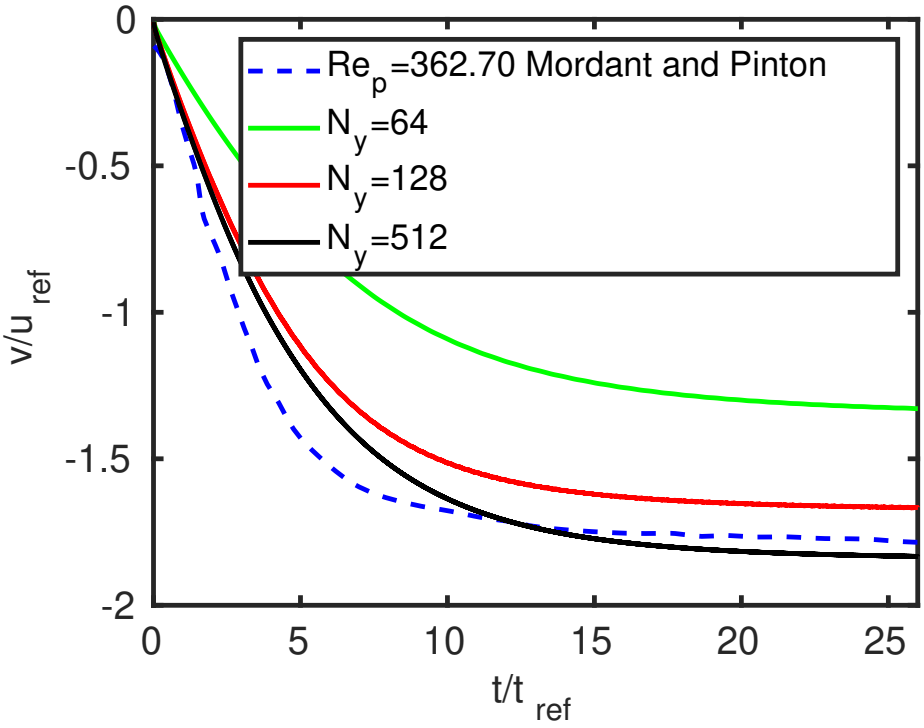

Figure 4.11: Effects of the grid size in y-direction for $Re_p = 362.70$ case on dimensionless vertical terminal velocity comparision with the experimental data over dimensionless time

In Figure 4.12, density ratio is the highest in all cases. The grid and CFL number effects are investigated in this $Re_p = 280.42$ case together. First, CFL number, N_x and N_z are kept fixed at 0.3, 100 and 100 respectively. $N_y = 128$ and $N_y = 512$ cases are compared and as expected the latter one performed better and closer to experimental. Then CFL number effects are investigated on $N_y = 512$ case with 2 CFL numbers 0.3 and 0.1. Decline in CFL number also lead to lower error percentage in terminal velocity, even though it only changed transition region slightly.At CFL=0.3, the error percentage is around 5% but at CFL=0.1 it turns into 2% in terminal velocity.

In addition, several simulations are perfomed for the same computational setup with $Re_p = 362.70$ case in Figure 4.13 but the density ratios are less or equal to 1. Since the configuration didn't give exactly in [81], a similar setup is created. When the density ratio corresponds to $\rho_p/\rho_f = 1$, the rigid object stays stationary.

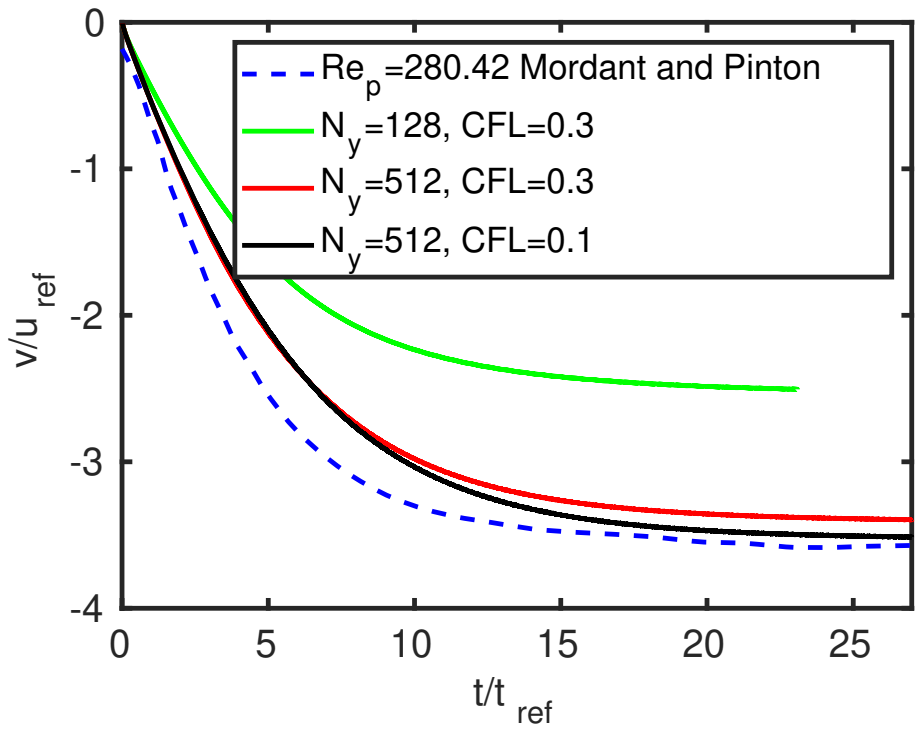

Figure 4.12: Effects of the grid size in y-direction for $Re_p = 280.42$ case and CFL number for $N_y = 512$ cases on dimensionless vertical terminal velocity comparision with the experimental data over dimensionless time.

In given dimensionless time, the vertical velocity in each simulation with different density ratio tends to plot an inclined line, and the acceleration has an inverse proportion with the ratio. On the other hand, it is observed that when $\rho_p/\rho_f \leq 0.5$ instabilities start to occur. Between $t/t_{ref} = 0.5$ and $t/t_{ref} = 1.0$, the cyan line expresses $\rho_p/\rho_f = 0.5$ critically increase and decrease at the same time step, which shows that the solver can not handle this value. Therefore, it is showed that the combined numerical method can be employed for buoyant particles with a ratio limitation.

According to results given in this Section, the presented algorithm is able to capture some characteristic features of sedimentation cases. The solver is examined with various settings from different aspects of dynamic movements of suspended particles. The accuracy in terminal velocity is higher than the transient field because of the viscosity effects. However, it should be highlighted that this combined

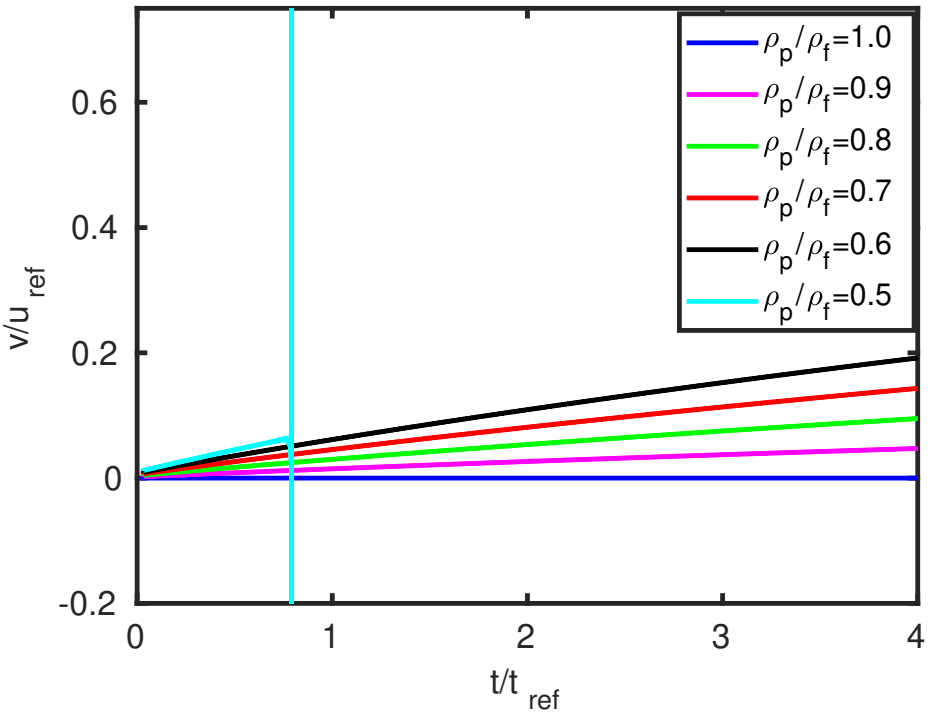

Figure 4.13: Terminal velocity in y-direction is non-dimensionalized and its change over dimensionless time is investigated for buoyant cases with various density ratios, $\rho_p/\rho_f = 0.9, 0.8, 0.7, 0.6, 0.5, 1.0$

method can operate in a broad range of Reynolds number and reflect the behaviour of rigid particles driven by gravity for three dimensional flows. In addition to that, the combined algorithm can be used for buoyant rigid particles with a density ratio restriction at $\rho_p/\rho_f = 0.5$. In the next section, sedimentation problems are investigated with higher resolution for density ratios close to one.

4.2 Sedimentation of Particles with Density Ratio Around Unity

In these validation section, sedimentation of a rigid sphere in a viscous fluid is simulated for density ratios around unity. The ambient fluid is quiescent and flow is induced solely due to the motion of the sphere. Previously, Apte et al. [7], Kempe and Frohlich [82] have simulated these cases and compared their results with the experimental data of Cate et al. [27] who examined four cases with the particle Reynolds number ranging between 1.5 and 31.9. In the experimental study, a nylon

sphere was released into a quiescent container filled with various silicon oils. They also performed Lattice-Boltzman (LB) simulations and the results are compared with the experimental data. The parameters used in the experimental study are summarized in Table 4.2. In the present simulations, all parameters are taken equal to the physical values and the results are presented in SI units.

Case Number	$\rho_f[kg/m^3]$	$\mu_f[Ns/m^2]$	$u_t[m/s]$	Re_p
1	970	373	0.038	1.5
2	965	212	0.060	4.1
3	962	113	0.091	11.6
4	960	58	0.128	31.9

Table 4.2: The experimental conditions for the four cases of the sedimentation of a nylon sphere used in the experimental study by Cate et al. [27] and are simulated in the present thesis.

In all the cases, the particle density and diameter are $\rho_p = 1120kg/m^3$ and $d_p = 0.015m$, respectively. The computational domain is $6.67D_p \times 13.33D_p \times 6.67D_p$ in the x, y and z directions, respectively, and the initial position of the particle is $(x_p, y_p, z_p) = (3.33D_P, 8.5D_p, 3.33D_p)$. The gravitational acceleration acts in the negative z direction. The no-slip wall conditions are applied on the top and bottom walls of the container while the periodic boundary conditions are applied in other directions. All the four cases are simulated but the Re_p=11.6 and 31.9 cases are chosen for a further investigation. In Figure 4.14, the cases summarized in 4.2 are simulated using a $64 \times 256 \times 64$ grid resolution. The maximum error percentage in terminal velocity is obtained for the $Re_p = 31.9$ case as 4.7% based on the particle Reynolds number Re_p in the steady state.

In Figure 4.15, the numerical results for $Re_p = 11.6$ and $Re_p = 31.9$ cases are compared with the experiment data. The simulations are performed using a $64 \times 256 \times 64$ grid resolution. In this figure, the symbols indicate the experiment data while the solid lines are the numerical results. The same color is used for the experimental and numerical results for the same case. Note that the experimental

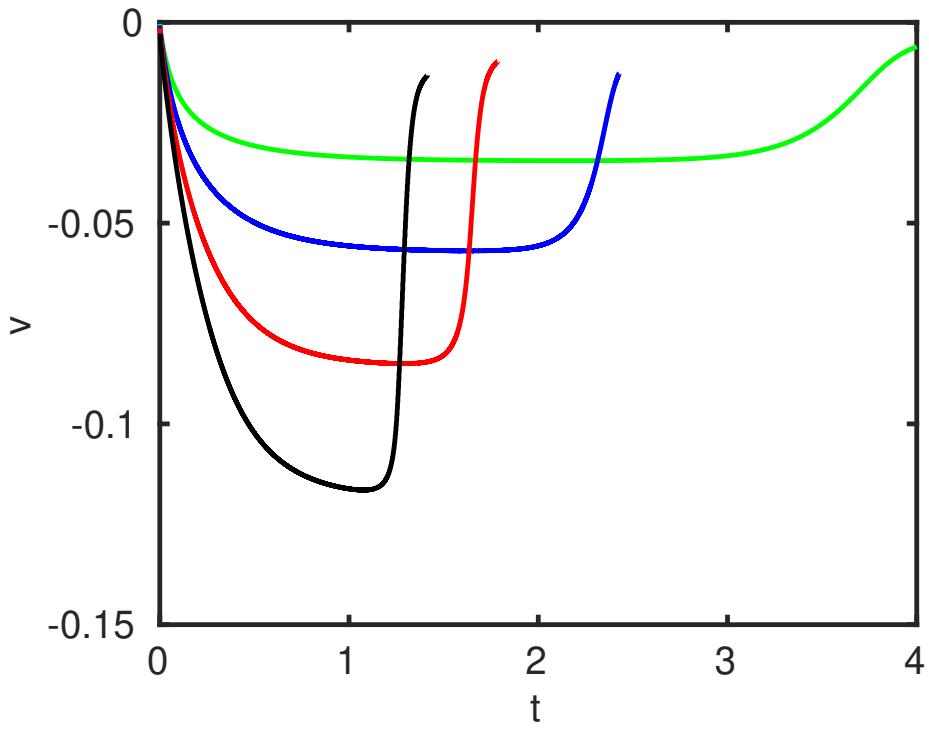

Figure 4.14: Change of the terminal velocity v in time for $Re_p = 1.5$, $Re_p = 4.1$, $Re_p = 11.6$, and $Re_p = 31.9$ corresponds to green, blue, red and black straight lines in order.

data are not provided for the time interval between $t = 0.2$ and $t = 0.4$. In the previous numerical studies, it has been reported that the density ratio around unity causes some numerical instabilities and some special treatments for stable solutions. In the simulations, no special treatment is required and stable solutions are obtained for all the cases without any sign of numerical instability. As can be seen in Fig. 4.15, the present results are in good agreement with the experimental data and the difference is well within the experimental uncertainty. Note that, the density ratio is about the same and only difference is the viscosity in these two cases. Therefore, the results can be interpreted as the impact of the viscosity on the terminal velocity of a solid sphere.

The grid convergence is examined for $Re_p = 31.9$ case and the results are shown in Fig. 4.16. The number of the grid points used to resolve the solid sphere can be

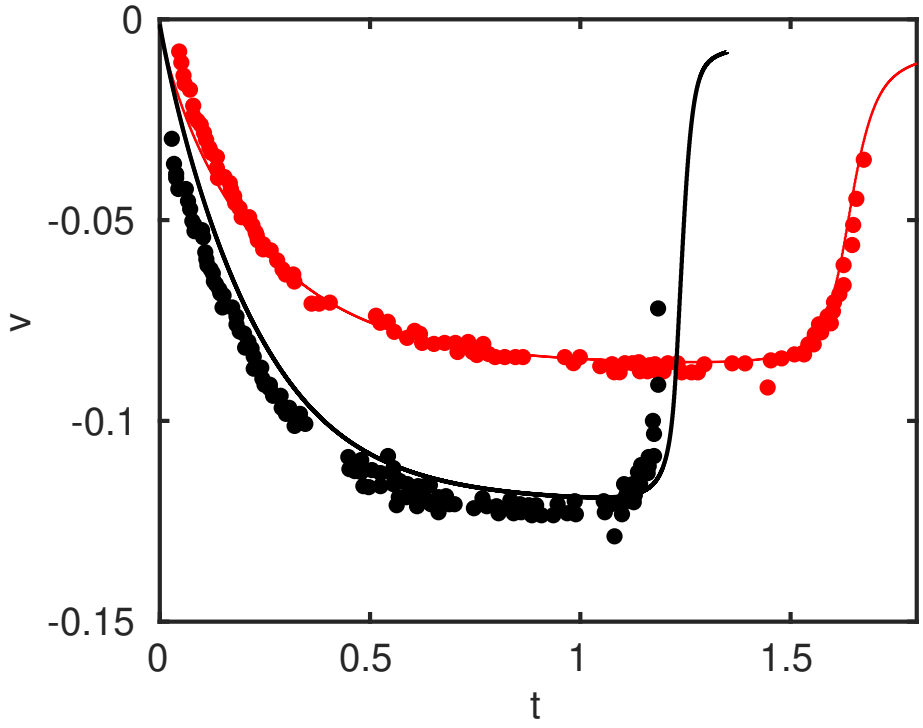

Figure 4.15: Comparison of the computational results with the experimental data for $Re_p = 11.6$ and $Re_p = 31.9$ cases for the time evolution of the particle velocity. The symbols and the lines denote the experimental data and the computational results, respectively. Red and black colors indicate $Re_p = 11.6$ and $Re_p = 31.9$ cases, respectively.

given by

$$N_r = \frac{d_p}{h} = \frac{d_p}{L_y/N_y}, \tag{4.3}$$

where N_r is the number of grid points inside the particle, h is the Eulerian mesh width, L_y is the length of the domain in the y-direction, and N_y is the total number of grid points in the y-direction. In Fig. 4.16, the simulations are performed using $64 \times 64 \times 64$, $64 \times 192 \times 64$, and $64 \times 256 \times 64$ grid resolutions which correspond to the number of points in the particle about 5, 15, and 20, respectively. As can be seen in this figure, a full grid convergence is not yet achieved but the difference between results obtained using successively refined grid get smaller, which can be considered as an evidence that method is grid convergent. Further simulations must be performed on finer grids to achieve fully grid independent solutions.

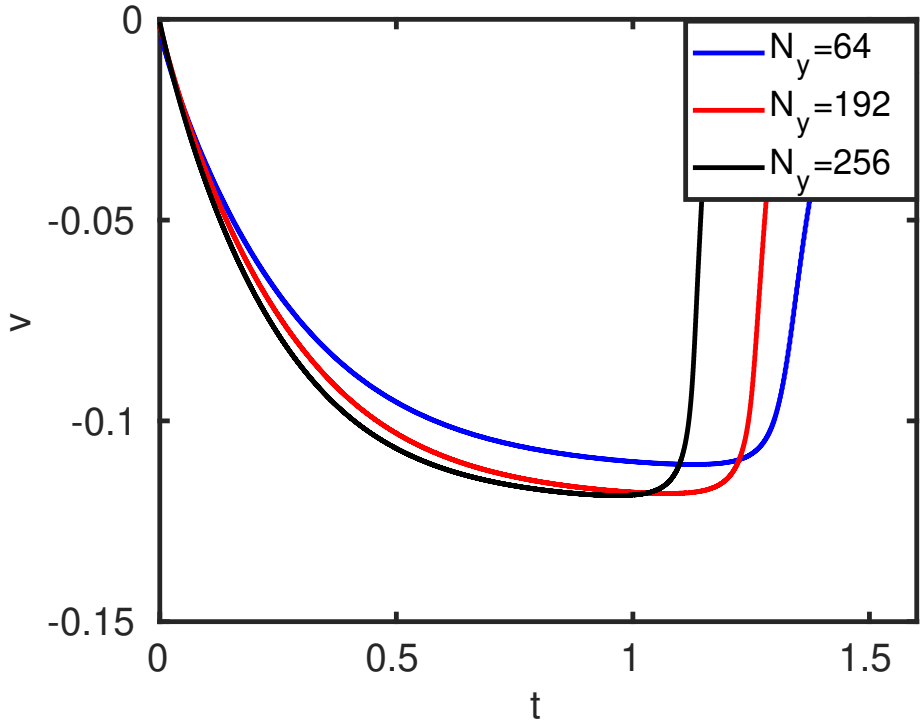

Figure 4.16: The grid convergence of the numerical method for the $Re_p = 31.9$ case. Simulations are performed using $64 \times 64 \times 64$, $64 \times 192 \times 64$, and $64 \times 256 \times 64$ grid resolutions which correspond to the number of points in the particle about 5, 15, and 20, respectively.

The velocity contours are plotted in In the Figs. 4.17-4.24 for the $Re_p = 1.5$ and $Re_p = 31.9$ cases at four different times. In the case 1, from $t = 0$ s to $t = 0.63$ s (Fig. 4.17), the velocity field gets larger in time. However, between $t = 0.63$ s to $t = 1.56$ s (Fig. 4.18), the contours concentrate at sides but elongate through the top region. At $t = 3.5$ s (Fig. 4.19), when the sphere is about to reach the bottom of the container, the velocity field expands and finally becomes nearly invisible and stationary at $t = 4.0$ s (Fig. 4.20). In the case 4, the velocity contours expand vertically between $t = 0.3$ s and $t = 0.6$ s (Fig. 4.22), and also the flow speed behind the sphere increases. In Fig. 4.23, the sphere has a long region at the back and contours spread to the sides at the bottom part of the container. In Fig. 4.24, the sphere hits the ground and the contours take the longest and narrowest shape. It can be seen that the contours for the higher Reynolds number case, i.e.,

$Re_p = 31.9$, the velocity contours become narrower and longer compared to the $Re_p = 1.5$ case. In the case 1, the velocity contours become asymmetric about the direction of the motion, which is consistent with the observations by Apte et al. [7]. In both cases, the velocity contours at the time when the sphere hits he bottom wall are qualitatively similar to those observed by Cate et al. [27] experimentally and Apte et al. [7, 27] computationally.

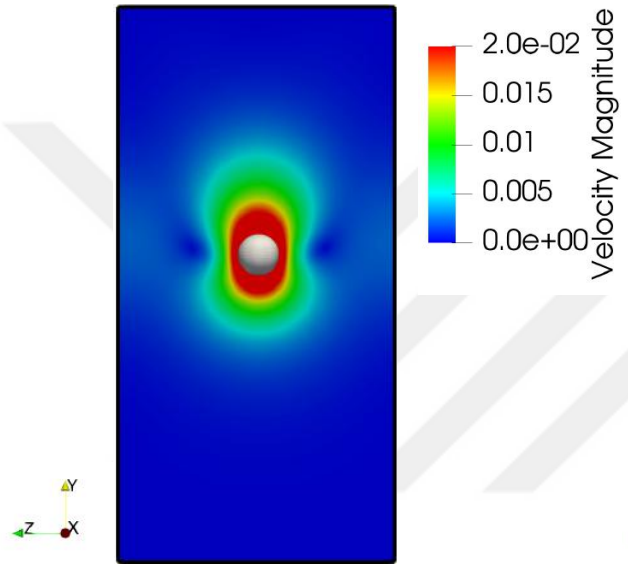

Figure 4.17: Velocity contour for the $Re_p = 1.5$ case at $t = 0.63$ s.

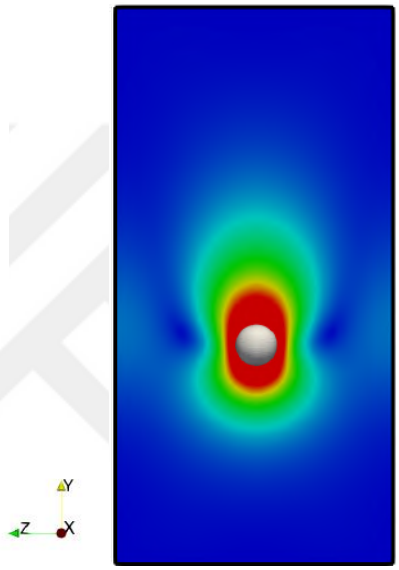

Figure 4.18: Velocity contour for the $Re_p = 1.5$ case at $t = 1.56$ s.

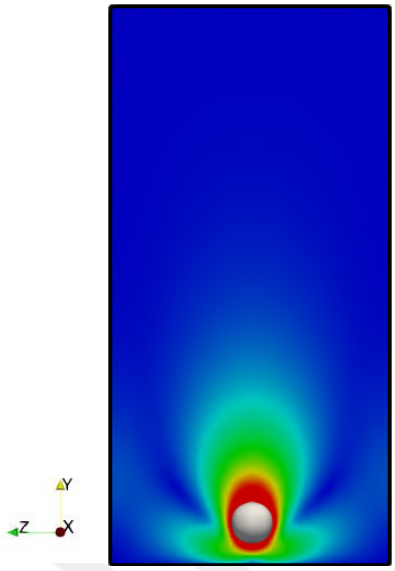

Figure 4.19: Velocity contour for the $Re_p = 1.5$ case at $t = 3.5$ s.

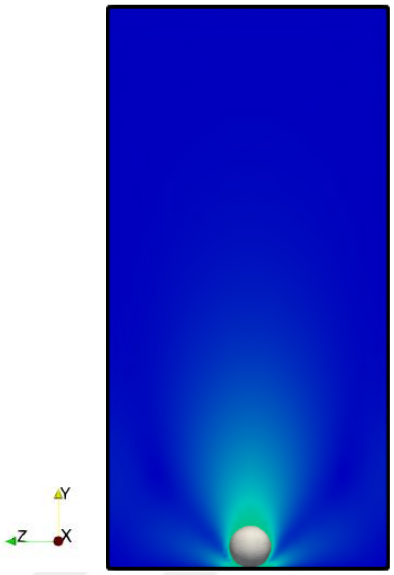

Figure 4.20: Velocity contour for the $Re_p = 1.5$ case at $t = 4.0$ s.

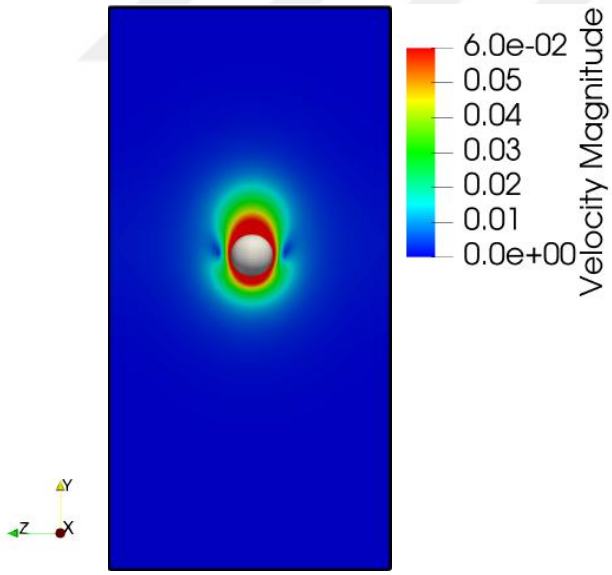

Figure 4.21: Velocity contour for the $Re_p = 31.9$ case at $t = 0.3$ s.

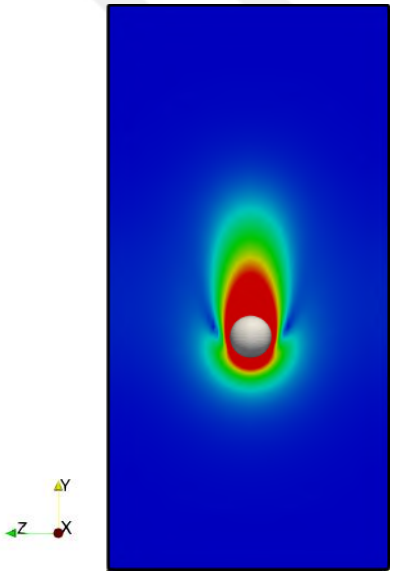

Figure 4.22: Velocity contour for the $Re_p = 31.9$ case at $t = 0.6$ s.

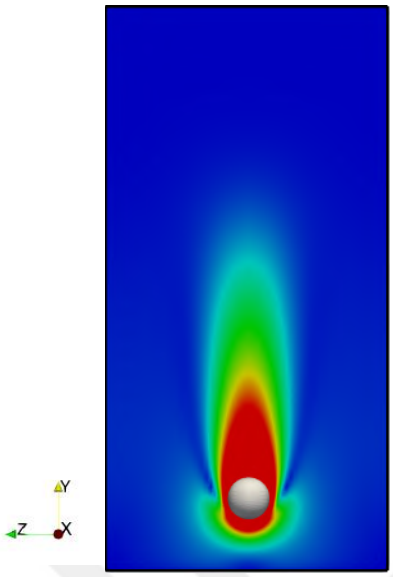

Figure 4.23: Velocity contour for the $Re_p = 31.9$ case at $t = 1.1$ s.

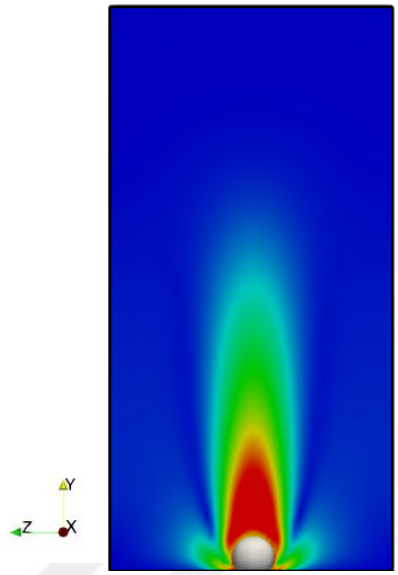

Figure 4.24: Velocity contour for $Re_p = 31.9$ case at $t = 1.3$ s.

Figures 4.25, 4.26 and 4.27 show the streamlines plotted for the case 4 at $t = 0.6$, 1.0 and 1.3. As seen in these figures, large vortices are created on both sides of the sphere as it falls on the bottom wall. The effects of the bottom wall can also be seen clearly in Fig. 4.27.

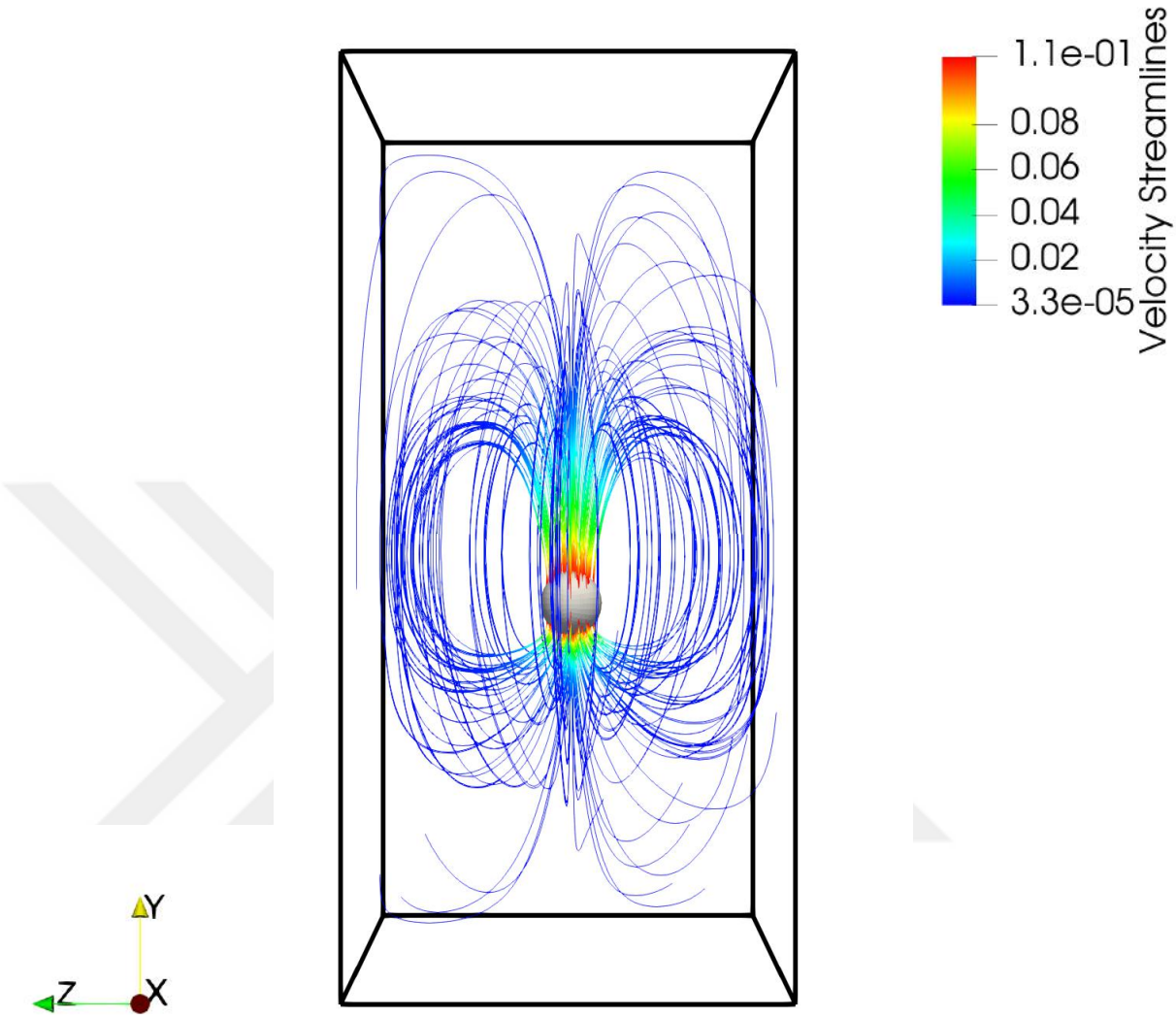

Figure 4.25: The streamlines plotted for the case 4 at $t = 0.6$ s. ($Re_p = 31.9$)

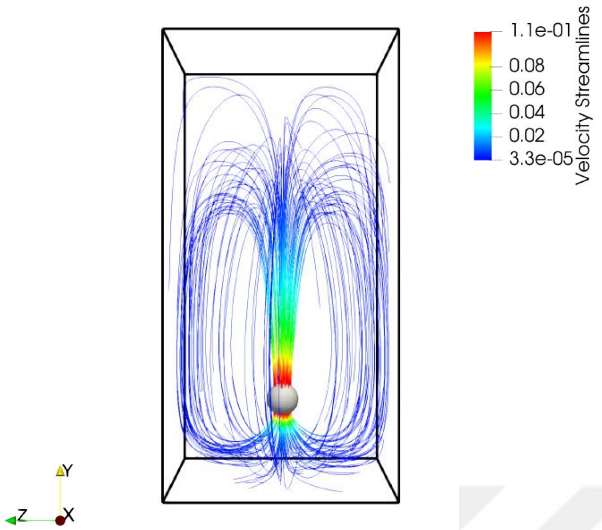

Figure 4.26: The streamlines plotted for the case 4 at $t = 1\ s.$ ($Re_p = 31.9$)

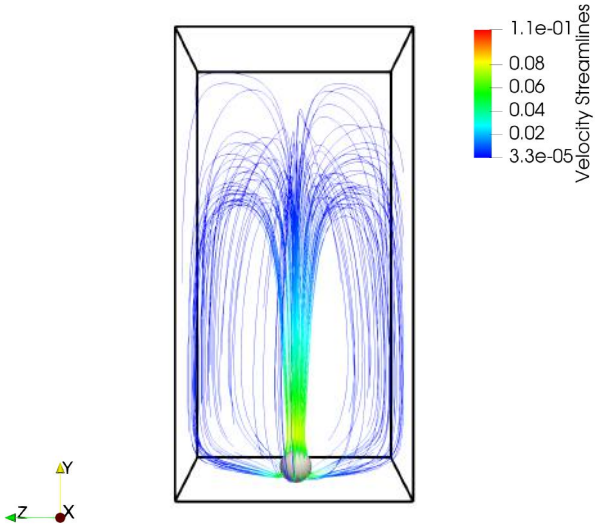

Figure 4.27: The streamlines plotted for the case 4 at $t = 1.3\ s.$ ($Re_p = 31.9$)

4.3 Many Particle System

In this section, combined solver is used to simulate particulate flows including many particle systems. In this case, sedimentation of the multiple particles is observed similar to particle-laden flows. Since the presented work does not have any collision model, particle-particle interactions are not taken into consideration in these simulations, which is justified only for very dilute cases. The results are evaluated only qualitatively as preliminary results for the fluid-fluid-solid three phase flows. Simulations are preformed for the case of 40 particles that are initially located uniformly in the rectangular domain filled with a liquid of The kinematic viscosity $\nu = 0.01m^2/s$ as shown in Fig. 4.28. The particles have the same size with the diameter of $D_p = 0.015$ m and start to sink solely due to gravitational acceleration of $g = 9.81$ m/s^2 from the rest. The size of the computational domain is $0.2\ m \times 0.2\ m \times 0.2\ m$. All particles are assumed to be identical with the density ratio of $\rho_p/rho_f = 2.56$. The flow parameters are the same as those used by Uhlmann [153] except for that the number of the particles is reduced to 40 instead of 63 for the sake of simplicity.

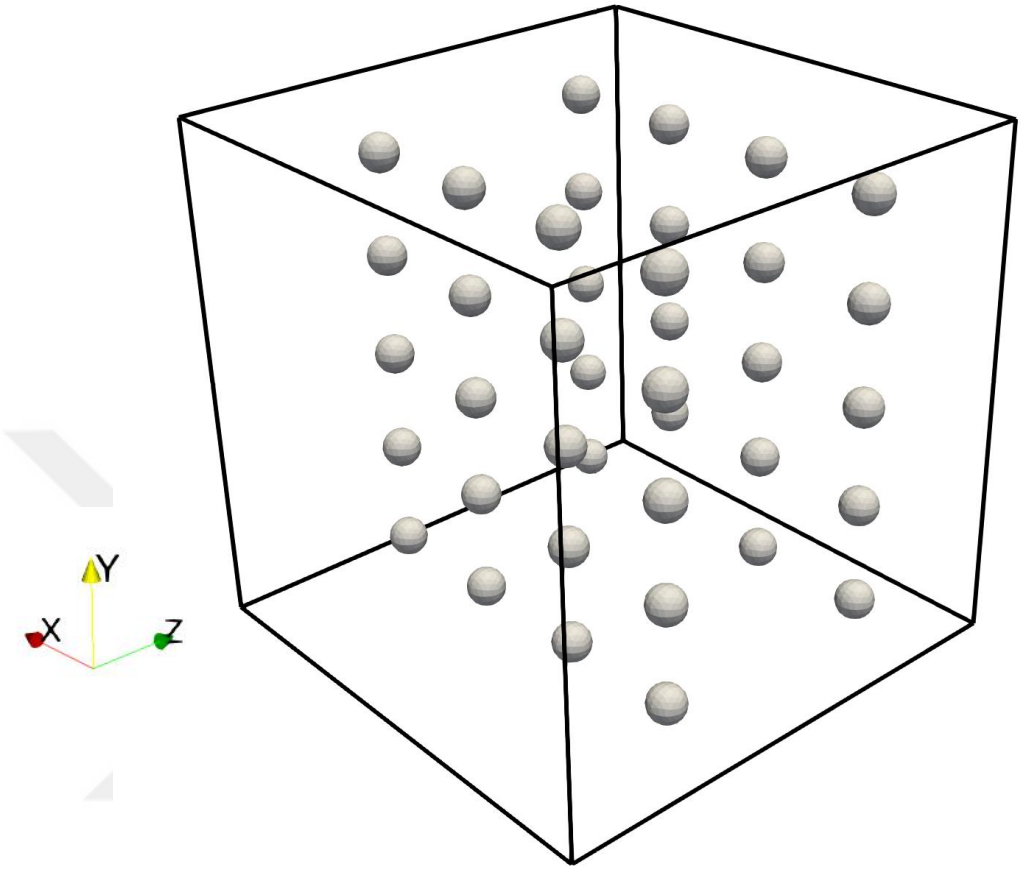

Figure 4.28: The initial distribution of particles for the particulate flow simulations involving 40 particles. All other flow parameters are the same as those used by Uhlmann [153].

Figure 4.29 shows time evolution of the average vertical velocity of the particles v_{avg} defines as

$$v_{avg} = \frac{\sum_{i=1}^{N_p} v_{pi}}{N_p}, \tag{4.4}$$

where N_p and v_{pi} are the total number of particles and the vertical velocity of i^{th} particle, respectively. The magnitude of the average vertical velocity increases until the dimensionless time $t/t_{ref} \approx 5$ when the particles that are initially located at the lowest position reach the bottom of the container. The particle distribution is shown at $t/t_{ref} \approx 5$ in Fig. 4.30. Note that, since some of the particles are positioned at lower in vertical direction, the average velocity is less than the reference velocity,

u_{ref} as in [102, 153].

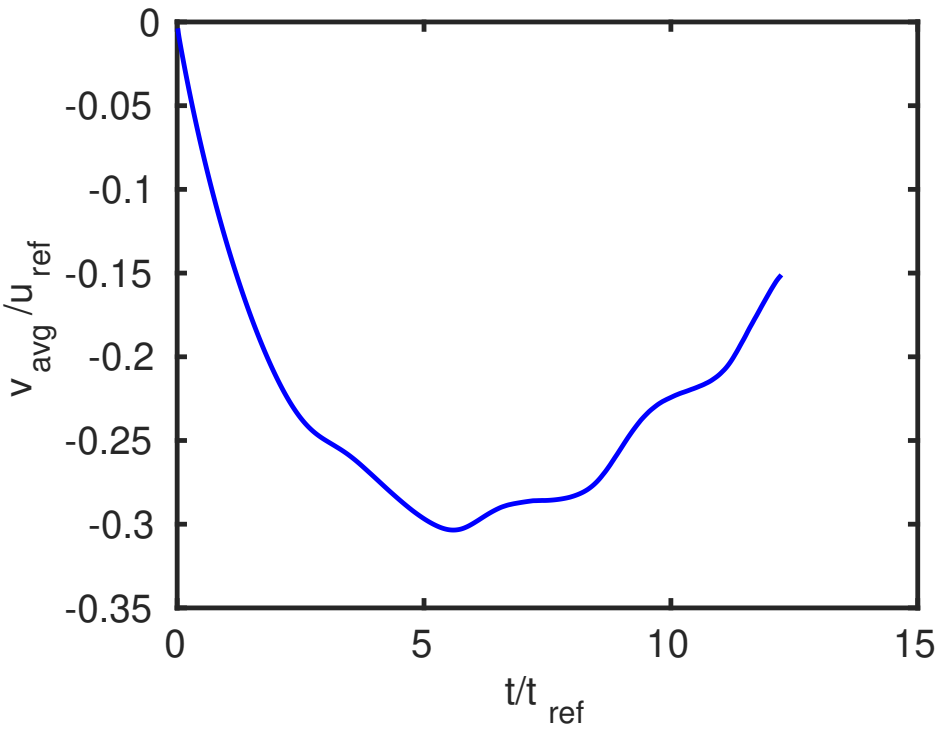

Figure 4.29: The time evolution of the dimensionless average terminal velocity of particles in the 40 particle system.

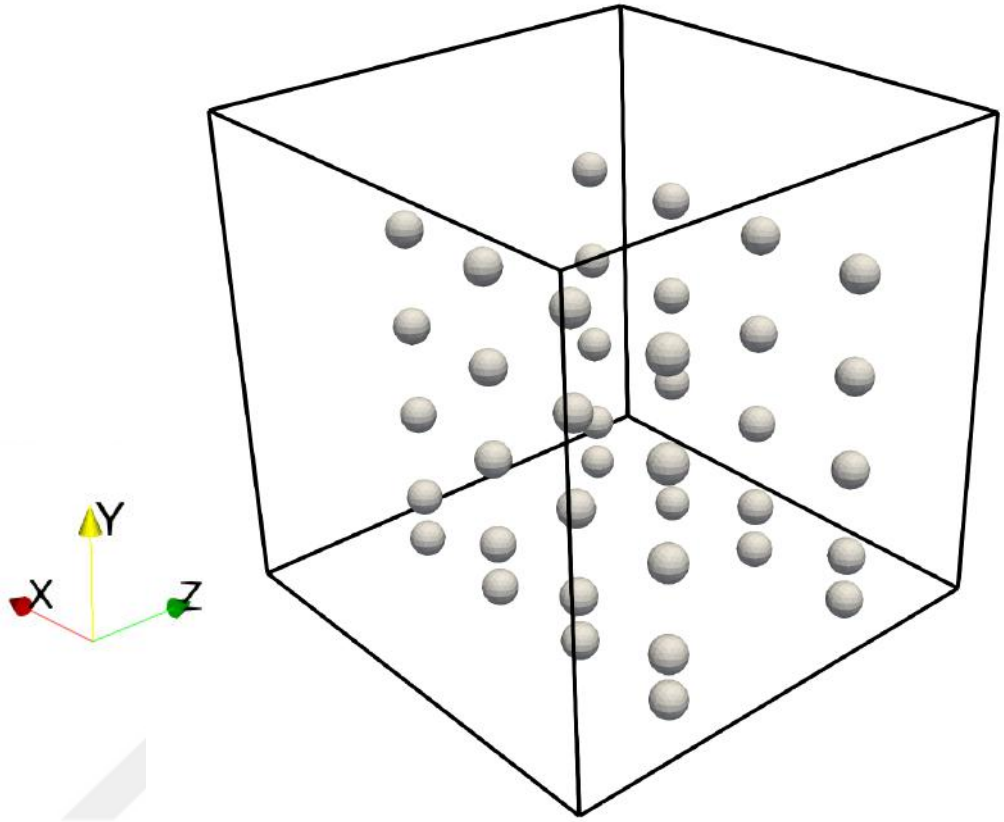

Figure 4.30: Particle distribution at $t/t_{ref} = 5$ in the 40 particle system.

The steamlines and constant contours of the magnitude of the vertical velocity component are shown In Figs. 4.31 and 4.32, respectively, for the many particle system. Since the particles become stationary at the bottom, streamlines can not be observed in that region.

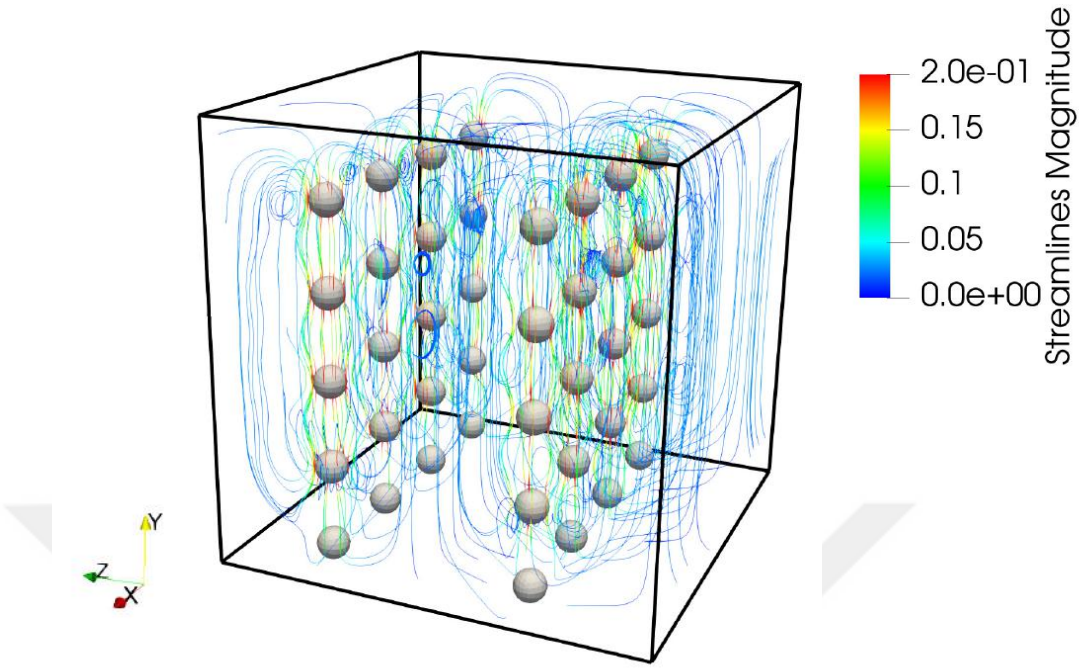

Figure 4.31: Streamlines computed for the 40-particle system is plotted at $t/t_{ref} = 5$.

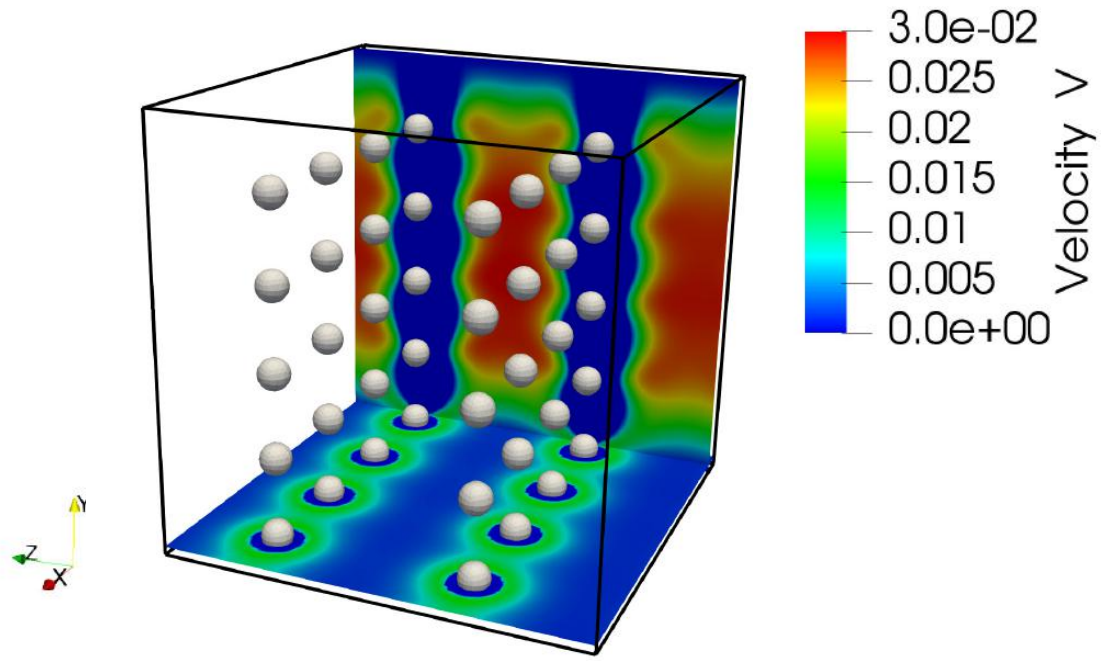

Figure 4.32: Constant contours of the magnitude of the vertical velocity component for the 40-particle system are plotted $t/t_{ref} = 5$.

Even though the interaction between particles are not implemented in the present hybrid solver, the preliminary results show that the method is capable of simulating particulate flows involving many rigid particles in a dilute particle concentration. A model for particle-particle interactions is needed for simulations of particulate flows with a dense particle concentration. Since the present results are only preliminary, they can be interpreted only qualitatively. A more rigorous study is required for a quantitative interpretation, which will be subject of a future study.

Chapter 5

CONCLUSIONS

In this study, a combined IBM/FTM solver is developed for interface-resolved simulations of viscous incompressible three phase fluid-fluid-solid flows in 3D. The front-tracking framework is employed to treat the fluid-rigid body boundary conditions, and coupled with the one-field formulation and Chorin's projection scheme to handle 3 phase problems. Direct forcing immersed boundary method (IBM) is used to impose the boundary conditions on the solid surface. The direct forcing IBM is selected since it is easy to implement, does not require specification of any free parameter and highly compatible with the front-tracking method. In the present method, a separate Lagrangian grid is employed to represent the fluid-solid interface in a similar fashion as used for the fluid-fluid interfaces in the front-tracking method. The communication between the Lagrangian and Eulerian grids is maintained using the distributions and interpolation schemes similar to the front-tracking method. Although the method has been implemented and validated only for the fluid-solid interfaces, the fluid-fluid interface treatment is retained in the code so that it can be readily used for simulations of three-phase fluid-fluid-solid multiphase flows.

The method is first applied to simulate flow around a stationary solid sphere in a uniform flow for various Reynolds numbers. These simulations are performed only using relatively coarse grids because the main purpose of this test is to examine the numerical properties of the method qualitatively, i.e., we examine the distribution of the IBM force field, satisfaction of the no-slip boundary conditions at the fluid-solid interface, symmetry of the solution at low Reynolds number. The numerical results are found to be qualitatively convincing.

After the qualitative validation, the method is validated quantitatively for two test cases for which experimental data are available. The first quantitative test case deals with the sedimentation of a rigid spherical particle in an otherwise quiescent

viscous fluid. Simulations are performed for various values of the kinematic viscosity, the density ratio, and channel length to access the performance of the method for a wide range of flow conditions. In particular, simulations are performed for the particle Reynolds numbers of $Re_p \approx 41$ and $Re_p \approx 362$. The results are generally found to be in good agreement with the experimental data with the maximum deviation of 5%. It is found that the discrepancy between the numerical and experimental results is slightly larger in the transient period (i.e., 5%) and decreases down to 3% in the steady motion. Simulations are also performed for various grid resolutions and time steps, and the method is shown to be convergent both in terms of spatial and temporal errors. Then, simulations are performed for the density ratio close to the unity and the method is found to be stable and robust, which is in contrast with the Uhlmann's *direct forcing* method [152] which requires some special treatment to prevent numerical instability for the density ratio close to unity. The present solver is applied to simulate four different cases for Re_p ranging from 1.5 to 31.9 and the density ratio is close to unity. For all four case, the results are found to be in good agreement with the experimental results, i.e., deviation of the numerical particle velocity from the experimental one is about 3% in the steady motion. The constant contours of vertical velocity and streamlines are also computed and plotted for all the cases and found to match with the previous studies.

Finally, the method is applied to simulate flows involving many particles and the preliminary results are presented. The results are found to be qualitatively convincing but a more rigorous analysis is required for a quantitative assessment. The results clearly show that the present method is applicable to the particulate flows involving many particles in dilute concentrations. However, a particle-particle collision model must be incorporated to reliably use the present method for the simulation of flows with denser concentrations.

Several issues are left for a future study. The performance of the method deteriorates in the transient motion of particles, which is attributed to neglecting the effects of the inertia of the fluid in the solid particle. As mentioned in thesis, the solid particle is assume to contain the same fluid as the ambient fluid. Breugem [20] stated that neglecting the fluid inertial in the volume occupied by the solid particle

can affect the transient motion of the particle and results in 9% deviation. Addition of the inertia term that can be calculated numerically or analytically can improve the performance of the present method especially in the transient motion. Another important issue is that the immersed boundary force is computed and added to the momentum equation only once in the present implementation. It is known that the predicted IBM force may not fully enforce the no-slip boundary conditions at the fluid-solid interface, and it may be needed to perform an inner iteration to make sure that the boundary conditions are fully satisfied. Instead of applying direct force just once at each time step as is done here, the force can be computed and applied for several times in an inner loop until the no-slip boundary conditions are fully satisfied. This is called as multi-forcing method. Such an inner iteration has not been incorporated in the present algorithm but it is a good idea to examine the effectiveness of multi-forcing method in the future. The multi-forcing and internal treatment can improve the accuracy of the solver significantly especially in the cases involving complex geometries. In the present study, the method is only applied to simulate rigid particles. However, in many applications such as biological flow and soft robotics [3], the solid body undergoes significant deformations. The present framework is suitable to allow the flexibility of the solid objects but with considerable algorithm development and implementation. It is however relatively easier to modify the present algorithm to simulate rigid objects with different geometrical shapes such as an ellipsoidal body as studied by Zaleski et al. [58]. Finally, the method can be readily used to simulate the fluid-fluid-solid multiphase flows involving only rigid spherical particles. However, in a three phase model, solid and fluid particles interact with each other, so proper collision models must be incorporated. The lubrication theory and soft-sphere collision model may be used to make the model more realistic as well as computationally efficient.

Acknowledgments

We acknowledge financial support by Koc University and Turkish Academy of Sciences (TUBA).

BIBLIOGRAPHY

[1] L. M. Adams. A multigrid algorithm for immersed interface problems. *SIAM Journal on Scientific Computing*, 1996.

[2] A. Almgren, J. Bell, P. Colella, and T. Marthaler. A cartesian grid projection method for the incompressible euler equations in complex geometries. *SIAM Journal on Scientific Computing*, 18, 10 1996.

[3] M. Amjadi, K. U. Kyung, I. Park, and M. Sitti. Stretchable, skin-mountable, and wearable strain sensors and their potential applications: A review. *Advanced Functional Materials*, 26, 11 2015.

[4] D. Anderson, G. McFadden, and A. Wheeler. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics*, 30, 06 1997.

[5] P. Angot, C. H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in viscous flows. *Numerische Mathematik*, 81:497–520, 02 1999.

[6] K. Appa. Finite-surface spline. *Journal of Aircraft*, 1989.

[7] S. V. Apte, M. Martin, and N. A. Patankar. A numerical method for fully resolved simulation (frs) of rigid particle flow interactions in complex flows. *Journal of Computational Physics*, 228(8):2712 – 2738, 2009.

[8] M. N. Ardekani, P. Costa, W. P. Breugem, and L. Brandt. Numerical study of the sedimentation of spheroidal particles. *International Journal of Multiphase Flow*, 87:16 – 34, 2016.

[9] G. Atefi, H. Niazmand, and M. Meigounpooy. Numerical analysis of 3d flow past a stationary sphere with slip condition at low and moderate reynolds numbers. *Journal of Dispersion Science and Technology*, 28:591–602, 04 2007.

[10] S. Balachandar and J. Eaton. Turbulent dispersed multiphase flow. *Annu. Rev. Fluid Mech.*, 42:111–33, 2010.

[11] E. Balaras. Modeling complex boundaries using an external force field on fixed cartesian grids in large-eddy simulations. *Computers & Fluids*, 33:375–404, 03 2004.

[12] Y. Bazilevs, K. Takizawa, and T. Tezduyar. *Computational Fluid-Structure Interaction: Methods and Applications*. Wiley Series in Computational Mechanics, 02 2013.

[13] M. Berger and M. Aftosmis. Aspects (and aspect ratios) of cartesian mesh methods. In *16th International Conference on Numerical Methods in Fluid Dynamics*, 11 2000.

[14] M. Berger and A. Leveque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis*, 35, 12 1997.

[15] M. Berger and I. Rigoutsos. An algorithm for point clustering and grid generation. *Systems, Man and Cybernetics, IEEE Transactions on*, 21:1278 – 1286, 10 1991.

[16] R. P. Beyer. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics*, 98(1):145 – 162, 1992.

[17] R. P. Beyer and R. J. LeVeque. Analysis of a one-dimensional model for the immersed boundary method. *SIAM Journal on Numerical Analysis*, 29:332–364, 1992.

[18] R. Bisplinghoff, H. Ashley, and R. Halfman. *Aeroelasticity*. Dover Books on Aeronautical Engineering Series. Dover Publications, 1996.

[19] D. C. Bottino. Modeling viscoelastic networks and cell deformation in the context of the immersed boundary method. *Journal of Computational Physics*, 147(1):86 – 113, 1998.

[20] W. P. Breugem. A combined soft-sphere collision/immersed boundary method for resolved simulations of particulate flows. *American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FEDSM*, 1, 01 2010.

[21] W.-P. Breugem. A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows. *Journal of Computational Physics*, 231(13):4469 – 4498, 2012.

[22] W. P. Breugem, V. Dijk, and R. Delfos. An efficient immersed boundary method based on penalized direct forcing for simulating flows through real porous media. *American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FEDSM*, 1, 07 2012.

[23] H. C. Brinkman. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Flow, Turbulence and Combustion*, 1949.

[24] S. A. Brown. Displacement extrapolations for cfd + csm aeroelastic analysis. In *38th Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, apr 1997.

[25] B. Bunner and G. Tryggvason. An examination of the flow induced by the motion of many buoyant bubbles. *Journal of Visualization - J VIS*, 2:153–158, 06 1999.

[26] M. Calaf, C. Meneveau, and J. Meyers. Large eddy simulation study of fully developed wind-turbine array boundary layers. *Physics of Fluids*, 22(1):015110, 2010.

[27] A. Cate, C. Nieuwstad, J. Derksen, and H. Van den Akker. Piv experiments and lattice-boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids*, 14:4012–4025, 11 2002.

[28] J. I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224(2):757 – 784, 2007.

[29] A. Chorin. Numerical solution of the navierstokes equations. *Mathematics of Computation*, 22, 10 1968.

[30] D. K. Clarke, H. Hassan, and M. D. Salas. Euler calculations for multielement airfoils using cartesian grids. *AIAA Journal*, 24, 1985.

[31] P. S. Costa. *Interface-Resolved Simulations of Dense Turbulent Suspension Flows*. PhD thesis, Technical University of Delft, 2017.

[32] N. G. Deen, M. van Sint Annaland, and J. Kuipers. Direct numerical simulation of complex multi-fluid flows using a combined front tracking and immersed boundary method. *Chemical Engineering Science*, 64(9):2186 – 2201, 2009.

[33] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in fsi applications. *Computers & Structures*, 86(23):2224 – 2234, 2008.

[34] D. DeZeeuw and K. G. Powell. An adaptively refined cartesian mesh solver for the euler equations. *Journal of Computational Physics*, 104(1):56 – 68, 1993.

[35] E. H. Dowell. *A Modern Course in Aeroelasticity*. Springer, 1995.

[36] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35 – 60, 2000.

[37] R. Falgout, A. Baker, E. Chow, V. Henson, E. Hill, J. Jones, T. Kolev, B. Lee, J. Painter, C. Tong, P. Vassilevski, and U. Yang. Users manual, hypre high performance preconditioners. *http://www.llnl.gov/CASC/hypre/*, 110(2), 2019.

[38] C. Farhat, M. Lesoinne, and P. L. Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1):95 – 114, 1998.

[39] M. Fatica, G. Iaccarino, B. Khalighi, and P. Moin. Large eddy simulation of a road vehicle with drag-reduction devices. *AIAA Journal*, 40:2447–2455, 12 2002.

[40] L. Fauci. A grid-free method for high reynolds number flow around an immersed elastic structure. Technical report, Scientific and Technical Aerospace Reports, 05 1991.

[41] L. J. Fauci. Interaction of oscillating filaments: A computational study. *Journal of Computational Physics*, 86(2):294 – 313, 1990.

[42] L. J. Fauci and A. McDonald. Sperm motility in the presence of boundaries. *Bulletin of Mathematical Biology*, 57:679–699, 1995.

[43] L. J. Fauci and C. S. Peskin. A computational model of aquatic animal locomotion. *Journal of Computational Physics*, 77(1):85 – 108, 1988.

[44] R. S. Fearing, K. H. Chiang, M. H. Dickinson, D. L. Pick, M. Sitti, and J. Yan. Wing transmission for a micromechanical flying insect. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings*, volume 2, pages 1509–1516, 2000.

[45] A. L. Fogelson and C. S. Peskin. A fast numerical method for solving the three-dimensional stokes' equations in the presence of suspended particles. *Journal of Computational Physics*, 79(1):50 – 69, 1988.

[46] P. Friedl and K. Wolf. Tumour-cell invasion and migration: diversity and escape mechanisms. *Nature Reviews Cancer*, 3:362–374, 2003.

[47] Y. Fung. *An Introduction to the Theory of Aeroelasticity*. Dover Books on Aeronautical Engineering. Dover Publications, 2008.

[48] R. Ghias, R. Mittal, and H. Dong. A sharp interface immersed boundary method for compressible viscous flows. *Journal of Computational Physics*, 225(1):528 – 553, 2007.

[49] A. Gilmanov and F. Sotiropoulos. A hybrid cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies. *Journal of Computational Physics*, 207(2):457 – 492, 2005.

[50] A. Gilmanov, F. Sotiropoulos, and E. Balaras. A general reconstruction algorithm for simulating flows with complex 3d immersed boundaries on cartesian grids. *Journal of Computational Physics*, 191(2):660 – 669, 2003.

[51] E. Givelberg and J. Bunn. A comprehensive three-dimensional model of the cochlea. *Journal of Computational Physics*, 191(2):377 – 391, 2003.

[52] J. Glimm, J. Grove, X. Li, W. Oh, and D. Sharp. A critical analysis of rayleightaylor growth rates. *Journal of Computational Physics*, 169(2):652 – 677, 2001.

[53] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A distributed lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow. *Computer Methods in Applied Mechanics and Engineering*, 184(2):241 – 267, 2000.

[54] D. Goldstein, R. Handler, and L. Sirovich. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics*, 105(2):354 – 366, 1993.

[55] D. Goldstein, R. Handler, and L. Sirovich. Direct numerical simulation of turbulent flow over a modeled riblet covered surface. *Journal of Fluid Mechanics*, 302, 11 1995.

[56] D. B. Goldstein and T. Tuan. Secondary flow induced by riblets. *Journal of Fluid Mechanics*, 1998.

[57] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin. An adaptive, formally second order accurate version of the immersed boundary method. *Journal of Computational Physics*, 223(1):10 – 49, 2007.

[58] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational physics*, 152(2):423–346, 1999.

[59] G. Guruswamy and C. Byun. Direct coupling of euler flow equations with plate finite element structures. *AIAA Journal*, 1995.

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

[61] C. Hirt, A. Amsden, and J. Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *JOURNAL OF COMPUTATIONAL PHYSICS*, 14(3):227–253, 1974.

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

[63] M. Hoef, M. Van, M. Annaland, N. Deen, and H. Kuipers. Numerical simulation of dense gas-solid fluidized beds: A multiscale modeling strategy. *Annu. Rev. Fluid Mech*, 40:47–70, 01 2008.

[64] G. Hou, J. Wang, and A. Layton. Numerical methods for fluid-structure interaction a review. *Communications in Computational Physics*, 12(2):337377, 2012.

[65] H. H. Hu, N. Patankar, and M. Zhu. Direct numerical simulations of fluid-solid systems using the arbitrary lagrangian eulerian technique. *Journal of Computational Physics*, 169(2):427 – 462, 2001.

[66] G. Iaccarino. Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews - APPL MECH REV*, 56, 05 2003.

[67] G. Iaccarino, G. Kalitzin, and B. Khalighi. Towards a fast rans solver based on the immersed boundary technique. In *41st Aerospace Sciences Meeting and Exhibit*, 01 2003.

[68] T. Ikeno and T. Kajishima. Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations. *Journal of Computational Physics*, 226, 10 2007.

[69] D. Ingram, D. Causon, and C. Mingham. Developments in cartesian cut cell methods. *Mathematics and Computers in Simulation*, 61(3):561 – 572, 2003. MODELLING 2001 - Second IMACS Conference on Mathematical Modelling and Computational Methods in Mechanics, Physics, Biomechanics and Geodynamics.

[70] D. Jacqmin. Calculation of two-phase navierstokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96 – 127, 1999.

[71] Y. Jan. *Computational studies of bubble dynamics*. PhD thesis, University of Michigan, 1994.

[72] B. JD, L. H, and L. R. The numerical simulation of strongly unsteady flows with hundreds of moving bodies. *AIAA Pap*, 1998.

[73] C. Ji, A. Munjiza, and J. Williams. A novel iterative direct-forcing immersed boundary method and its finite volume applications. *Journal of Computational Physics*, 231(4):1797 – 1821, 2012.

[74] J.J.Quirk. *An Adaptive Mesh Refinement Algorithm for Computational Shock Hydrodynamics*. PhD thesis, Cranfield Institute of Technology College of Aeronautics, 1991.

[75] J.Mohd-Yusof. Combined immersed-boundary/b-spline method for simulations of flow in complex geometries. In *Center for Turbulence Research Annual Research Briefs*, 1997.

[76] M. P. Joel H. Ferziger. *Computational Methods for Fluid Dynamics*. Springer-Verlag Berlin Heidelberg, 1996.

[77] A. Johnson and T. Tezduyar. 3d simulation of fluid-particle interactions with the number of particles reaching 100. *Comput. Methods Appl. Mech. Eng.*, 145(3):301–321, 1997.

[78] T. A. Johnson and V. C. Patel. Flow past a sphere up to a reynolds number of 300. *Journal of Fluid Mechanics*, 378:1970, 1999.

[79] D. Juric and G. Tryggvason. A front-tracking method for dendritic solidification. *Journal of Computational Physics*, 123(1):127 – 148, 1996.

[80] R. K. Kapania, M. Bhardwaj, E. A. Reichenbach, and G. P. Guruswamy. Aeroelastic analysis of modern complex wings. In *th Symposium on Multidisciplinary Analysis and Optimization*, 1996.

[81] T. Kempe and J. Fröhlich. Collision modelling for the interface-resolved simulation of spherical particles in viscous fluids. *Journal of Fluid Mechanics*, 709:445, 10 2012.

[82] T. Kempe and J. Fröhlich. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *Journal of Computational Physics*, 231(9):3663 – 3684, 2012.

[83] K. Khadra, P. Angot, S. Parneix, and J. Caltagirone. Fictitious domain approach for numerical modelling of navierstokes equations. *International Journal for Numerical Methods in Fluids*, 34:651–684, 12 2000.

[84] J. Kim, D. Kim, and H. Choi. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1):132 – 150, 2001.

[85] J. Kobayashi, Y. Mori, K. Okamoto, R. Akiyama, M. Ueno, T. Kitamori, and S. Kobayashi. A microfluidic device for conducting gas-liquid-solid hydrogenation reactions. *SCIENCE*, 304(5675):1305–1308, 2004.

[86] M. C. Lai and C. S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705 – 719, 2000.

[87] B. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19(1):59–98, 1979.

[88] H. Levine and M. Nilsen-Hamilton. Angiogenesis-a biochemical/mathematical perspective. *Lecture Notes in Mathematics*, 1872, 11 2006.

[89] L. D. Libersky, A. G. Petschek, T. C. Carney, J. R. Hipp, and F. A. Allahdadi. High strain lagrangian hydrodynamics: A three-dimensional sph code for dynamic material response. *Journal of Computational Physics*, 109(1):67 – 75, 1993.

[90] H. Lomax, T. Pulliam, and D. Zingg. *Fundamentals of Computational Fluid Dynamics*. Scientific Computation. Springer Berlin Heidelberg, 2003.

[91] E. Loth, M. Taeibi-Rahni, and G. Tryggvason. Deformable bubbles in a free shear layer. *International Journal of Multiphase Flow*, 23(5):977 – 1001, 1997.

[92] H. J. Lugt. Autorotation. *Annual Review of Fluid Mechanics*, 15(1):123–147, 1983.

[93] A. Mark and B. Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227:6660–6680, 06 2008.

[94] V. Mathai, D. Lohse, and C. Sun. Turbulent dispersed multiphase flow. *Annu. Rev. Condens.Matter Phys.*, 11:529–559, 2020.

[95] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. V. Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825 – 4852, 2008.

[96] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 01 2005.

[97] R. Mittal, V. Seshadri, S. E. Sarma, and H. S. Udaykumar. Computational modeling of fluidic micro-handling processes. *TechConnect Briefs*, pages 388–391, 2002.

[98] M.J.Berger. *Adaptive Mesh Refinement for Hyperbolic Partial Differntial Equations*. PhD thesis, Stanford University, Department of Computer Science, 1982.

[99] J. Mohd-Yusof, P. Orlandi, and D. Haworth. Les of flows in complex geometries using immersed boundaries. *Proceedings of the Summer Program*, 11 1998.

[100] J. MONAGHAN. Simulating free-surface flows with sph. *JOURNAL OF COMPUTATIONAL PHYSICS*, 110(2):399–406, 1994.

[101] T. D. Montenegro-Johnson, D. J. Smith, and D. Loghin. Physics of rheologically enhanced propulsion: Different strokes in generalized stokes. *Physics of Fluids*, 25(8):081903, 2013.

[102] N. Mordant and J.-F. Pinton. Velocity measurement of a settling sphere. *The European Physical Journal B*, 18:343–352, 11 2000.

[103] S. MORTAZAVI and G. TRYGGVASON. A numerical study of the motion of drops in poiseuille flow. part 1. lateral migration of one drop. *Journal of Fluid Mechanics*, 411:325350, 2000.

[104] M. Muradoglu and A. D. Kayaalp. An auxiliary grid method for computations of multiphase flows in complex geometries. *Journal of Computational Physics*, 214(2):858 – 877, 2006.

[105] M. Muradoglu and H. A. Stone. Motion of large bubbles in curved channels. *Journal of Fluid Mechanics*, 570:455466, 2007.

[106] M. Muradoglu and G. Tryggvason. A front-tracking method for computation of interfacial flows with soluble surfactants. *Journal of Computational Physics*, 227(4):2238 – 2262, 2008.

[107] M. Muradoglu and G. Tryggvason. Simulations of soluble surfactants in 3d multiphase flow. *Journal of Computational Physics*, 274:737 – 757, 2014.

[108] M. Nobari and G. Tryggvason. Numerical simulations of three-dimensional drop collisions. *AIAA Journal*, 34:750–755, 04 1996.

[109] U. Olgac and M. Muradoglu. Computational modeling of unsteady surfactant-laden liquid plug propagation in neonatal airways. *Physics of Fluids*, 25, 07 2013.

[110] R. Onishi, T. Kimurat, Z. Guo, and T. Iwamiya. Coupled aero-structural model - approach and application to high aspect-ratio wing-box structures. In *Symposium on Multidisciplinary Analysis and Optimization*, 1998.

[111] E. S. Oran and J. P. Boris. *Numerical Simulation of Reactive Flow*. Cambridge University Press, 2nd edition, 2000.

[112] S. OSHER and J. SETHIAN. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *JOURNAL OF COMPUTATIONAL PHYSICS*, 79(1):12–49, 1988.

[113] H. Pan, X.-Z. Chen, X.-F. Liang, L.-T. Zhu, and Z.-H. Luo. Cfd simulations of gas-liquid-solid flow in fluidized bed reactors - a review. *Powder Technology*, 299:235–258, 2016.

[114] R. B. Pember, J. B. Bell, P. Colella, W. Y. Curtchfield, and M. L. Welcome. An adaptive cartesian grid method for unsteady compressible flow in irregular regions. *Journal of Computational Physics*, 120(2):278 – 304, 1995.

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

[116] C. S. Peskin. Flow patterns around heart valves: a digital computer method for solving the equations of motion. *IEEE Transactions on Biomedical Engineering*, pages 316–317, 1973.

[117] C. S. Peskin. The fluid dynamics of heart valves: Experimental, theoretical, and computational methods. *Annual Review of Fluid Mechanics*, 14(1):235–259, 1982.

[118] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479517, 2002.

[119] C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart i. immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics*, 81(2):372 – 405, 1989.

[120] F. Picano, W.-P. Breugem, and L. Brandt. Turbulent channel flow of dense suspensions of neutrally-buoyant spheres. *Journal of Fluid Mechanics*, 764:463–487, 01 2015.

[121] J. Pouyssegur, F. Dayan, and N. M. Mazure. Hypoxia signalling in cancer and approaches to enforce tumour regression. *Nature*, 441:437–443, 2006.

[122] A. Prosperetti and G. Tryggvason. *Computational Methods for Multiphase Flow*. Cambridge University Press, 2007.

[123] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Oxford Science Publications, 01 1999.

[124] R. Ramamurti and W. Sandberg. Simulation of flow about flapping airfoils using finite element incompressible flow solver. *AIAA Journal*, 39, 02 2001.

[125] R. Richtmyer and K. Morton. *Difference methods for initial-value problems*. Interscience tracts in pure and applied mathematics. Interscience Publishers, 1967.

[126] J. B. Ritz and J. Caltagirone. A numerical continuous model for the hydrodynamics of fluid particle systems. *International Journal for Numerical Methods in Fluids*, 30:1067 – 1090, 08 1999.

[127] 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.

[128] G. Ryskin and L. G. Leal. Numerical solution of free-boundary problems in fluid mechanics. part 1. the finite-difference technique. *Journal of Fluid Mechanics*, 148:117, 1984.

[129] G. Ryskin and L. G. Leal. Numerical solution of free-boundary problems in fluid mechanics. part 2. buoyancy-driven motion of a gas bubble through a quiescent liquid. *Journal of Fluid Mechanics*, 148:1935, 1984.

[130] E. Saiki and S. Biringen. Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method. *Journal of Computational Physics*, 123(2):450 – 465, 1996.

[131] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annu. Rev. Fluid Mech.*, 31, 01 1999.

[132] N. Sharma and N. A. Patankar. A fast computation technique for the direct numerical simulation of rigid particulate flows. *Journal of Computational Physics*, 205(2):439 – 457, 2005.

[133] P. J. Shopov, P. D. Minev, I. B. Bazhlekov, and Z. D. Zapryanov. Interaction of a deformable bubble with a rigid wall at moderate reynolds numbers. *Journal of Fluid Mechanics*, 219:241271, 1990.

[134] M. Sitti. Miniature soft robots - road to the clinic. *NATURE REVIEWS MATERIALS*, 3(6):74–75, 2018.

[135] W. H. Snyder and J. L. Lumley. Some measurements of particle velocity autocorrelation functions in a turbulent flow. *Journal of Fluid Mechanics*, 48(1):4171, 1971.

[136] B. Solenthaler, J. Schlfli, and R. Pajarola. A unified particle model for fluid solid interactions. *Computer Animation and Virtual Worlds*, 18(1):69–82, 2007.

[137] J. Southard. Introduction to fluid motions, sediment transport, and current-generated sedimentary structures. `https://ocw.mit.edu/courses/earth-atmospheric-and-planetary-sciences/12-090-introduction-to-fluid-motions-sediment-transport-and-current-generat course-textbook/ch3.pdf`, Fall 2006. Massachusetts Institute of Technology: MIT OpenCourseWare.

[138] J. M. Stockie and S. I. Green. Simulating the motion of flexible pulp fibres using the immersed boundary method. *Journal of Computational Physics*, 147(1):147 – 165, 1998.

[139] J. M. Stockie and B. R. Wetton. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. *Journal of Computational Physics*, 154(1):41 – 64, 1999.

[140] D. Sulsky and J. Brackbill. A numerical method for suspension flow. *Journal of Computational Physics*, 96(2):339 – 368, 1991.

[141] 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.

[142] P. Swarztrauber. Fast poisson solver. *Studies in Numerical Analysis*, 24, 1984.

[143] S. Takagi and Y. Matsumoto. Three-dimensional deformation of a rising bubble. In *Proceedings of theGerman-Japanese Symposium on Multiphase Flow*, 1998.

[144] S. Tasoglu, G. Kaynak, A. J. Szeri, U. Demirci, and M. Muradoglu. Impact of a compound droplet on a flat surface: A model for single cell epitaxy. *Physics of fluids*, 22 8, 2010.

[145] T. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 06 2001.

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

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

[148] Y. H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593 – 623, 2003.

[149] H. Udaykumar, R. Mittal, P. Rampunggoon, and A. Khanna. A sharp interface cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics*, 174(1):345 – 380, 2001.

[150] H. Udaykumar, W. Shyy, and M. Rao. Elafint - a mixed eulerian-lagrangian method for fluid flows with complex and moving boundaries. *International Journal of Numerical Methods in Fluids*, 1996.

[151] H. S. Udaykumar, W. Shyy, and M. M. RAO. Elafint: A mixed eulerianlagrangian method for fluid flows with complex and moving boundaries. *International Journal for Numerical Methods in Fluids*, 22(8):691–712, 1996.

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

[153] M. Uhlmann. Experience with dns of particulate flow using a variant of the immersed boundary method. In *European Conference on Computational Fluid Dynamics*, 01 2006.

[154] M. Uhlmann. Interface-resolved dns of vertical particulate channel flow in the turbulent regime. *Physics of Fluids - PHYS FLUIDS*, 20, 08 2011.

[155] 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.

[156] M. Vanella, P. Rabenold, and E. Balaras. A direct-forcing embedded-boundary method with adaptive mesh refinement for fluidstructure interaction problems. *Journal of Computational Physics*, 229(18):6427 – 6449, 2010.

[157] R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. Haworth. Large eddy simulation in complex geometric configurations using boundary body forces. *AIAA Journal*, 38(3):427–433, 2000.

[158] Z. Wang, J. Fan, and K. Luo. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow - INT J MULTIPHASE FLOW*, 34:283–302, 03 2008.

[159] J. Ward. *Turbulent Flow in Porous Media.* Journal of the Hydraulics Division. University of Arkansas, Engineering Experiment Station, 1965.

[160] C. Wood, A. Gil, O. Hassan, and J. Bonet. Partitioned block-gaussseidel coupling for dynamic fluidstructure interaction. *Computers & Structures*, 88(23):1367 – 1382, 2010.

[161] J. Yang and F. Stern. A simple and efficient direct forcing immersed boundary framework for fluidstructure interactions. *Journal of Computational Physics*, 231(15):5029 – 5061, 2012.

[162] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy. An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156(2):209 – 240, 1999.

[163] P. D. Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *Journal of Computational and Applied Mathematics*, 33(1):1 – 27, 1990.

[164] J. Zhang, S. Childress, A. Libchaber, and M. Shelley. Flexible filaments in a

flowing soap film as a model for one-dimensional flags in a two-dimensional wind. *Nature*, 408:835–9, 01 2001.

[165] L. Zhang and M. Gay. Immersed finite element method for fluid-structure interactions. *Journal of Fluids and Structures*, 23(6):839 – 857, 2007.

[166] N. Zhang and Z. Zheng. An improved direct-forcing immersed-boundary method for finite difference applications. *Journal of Computational Physics*, 221, 09 2006.

[167] W. Zhang, Y. Jiang, and Z. Ye. Two better loosely coupled solution algorithms of cfd based aeroelastic simulation. *Engineering Applications of Computational Fluid Mechanics*, 1:253–262, 12 2007.

[168] L. Zhu and C. S. Peskin. Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method. *Journal of Computational Physics*, 179(2):452 – 468, 2002.

Appendix A

A.1 General Schematic of the Code

The combined algorithm is built on the FTM code [106]. Entire code is written with Fortran language and operated with 2 processes. MPI and HYPRE libraries [37] are used for parallelization and pressure solver. The general scheme is given in items:

- Initialization of the domain and front according to *input* file.
- Start of the computational time.
- Calculation of the diffusive, convective, and body force terms to obtain first unprojected velocity from Equation 3.2a.
- Calculation of the $\tilde{\boldsymbol{u}}$ with additional discretized pressure gradient term.
- Interpolation of the $\tilde{\boldsymbol{u}}$ from Eulerian grid to Lagrangian element centers as $\tilde{\boldsymbol{U}}$.
- Computation of the force and torque terms, $\mathbf{F}$ and $\mathbf{T}$ on the immersed boundary from Equation 3.10.
- Transfer of the force term to Eulerian grid as f, and calculation of the second unprojected velocity in Equation 3.3b.
- Solving Poisson Equation and velocity correction.
- Updating the density field.
- Updating the linear and angular velocities of the particle $\mathbf{U}_c$ and $\boldsymbol{\Omega}_c$, and advancing the front.
- New time step.

During these operations, element centers and other interface information transferred to Lagrangian to Eulerian. Results are saved in MATLAB format to get necessary graphics and Paraview format to visualize them in 3D. In addition, TECPLOT is used for stationary sphere figures.

A.2 Spherical Particles

A.2.1 Force Points

In [152] and following articles, the *Lagrangian Force Points*(*LFPS*) are distributed according to volume of the particle. This model was designed to place *LFPS* evenly and number of the *LFPS* is given in Equation A.1 and detailed analysis can be found in [152, 154].

$$N_l \approx \frac{\pi}{3}(12\frac{r_p^2}{h^2} + 1) \tag{A.1}$$

N_l is the number of *LFPS*, r_p is the radius of the particle, and h is the mesh width. However, it is observed that change in number of *LFPS* doesn't impact the solution significantly. Because of this and easiness of the FTM formulation, the sphere is divided into triangular elements in the presented study. While the spherical object is added to the domain, points and elements are created simultaneously. The advantage of this method is preventing to recreate and relocate force points on each time step.

A.2.2 Newton-Euler Equations

In Equations 3.12a and 3.12b, discretized linear and angular momentum equations are provided for rigid particle. These equations can be taken into consideration as a final version of Newton-Euler Equations. In this section, the evolution of these equations will be explained in detailed. Equations are provided simply in 2.6a and 2.6b and the integral terms are defined on the fluid-solid interface ∂V. According to Cauchy, we can write the hydrodynamic force and torque terms as given at the below;

$$\oint_{\partial V} \boldsymbol{\sigma} \cdot \mathbf{n} dA = -\rho_f \int_{V_p} \mathbf{f} dV + -\rho_f \frac{d}{dt}(\int_{V_{pi}} \mathbf{u} dV) \tag{A.2a}$$

$$\oint_{\partial V} \mathbf{r} \times (\boldsymbol{\sigma} \cdot \mathbf{n}) dA = -\rho_f \int_{V_p} \mathbf{r} \times \mathbf{f} dV + -\rho_f \frac{d}{dt} (\int_{V_{pi}} \mathbf{r} \times \mathbf{u} dV) \tag{A.2b}$$

The first terms of A.2a and A.2b are the negative versions of the immersed forces of fluid-solid coupling. The properties that were given in Section 3.3 turn these terms to sum over element centers. It can be written as;

$$-\rho_f \int_{V_p} \mathbf{f} dV = -\rho_f \sum_{ijk} \mathbf{f}_{ijk} \Delta x \Delta y \Delta z \tag{A.3}$$

The other terms at right hand side of Equations A.2a and A.2b are named as rate-of-change terms and comprehensive calculation can be found at [20, 152].

When all the terms given in Equations A.2a, A.2b are put into 2.6a and 2.6b, the final version of the Newton-Euler equations are obtained and they are discretized in this thesis.