



Ben-Gurion University of the Negev  
Faculty of Engineering Sciences  
Department of Mechanical Engineering

**Development of a numerical methodology  
based on the Immersed Boundary method  
for simulation of incompressible viscous  
flow in the presence of rigid bodies with  
periodically moving boundaries**

Thesis submitted in partial fulfillment  
of the requirements for the M.Sc. degree

**By: Rafi Sela**

July 2020



Ben-Gurion University of the Negev  
Faculty of Engineering Sciences  
Department of Mechanical Engineering

**Development of a numerical methodology  
based on the Immersed Boundary method  
for simulation of incompressible viscous  
flow in the presence of rigid bodies with  
periodically moving boundaries**

Thesis submitted in partial fulfillment  
of the requirements for the M.Sc. degree

By: Rafi Sela

Supervised by: Dr. Yuri Feldman

Author: ...  .....

Date: 27.08.2020

Supervisor: ..... 

Date: 27.08.2020

Chairman of Graduate Studies Committee: .....

Date: .....

July 2020

## **ACKNOWLEDGEMENT**

I would like to express my deepest gratitude and appreciation to my research supervisor, Dr. Yuri Feldman, for his invaluable guidance and unwavering support throughout the entire research. In his diligence, inventiveness, "hands-on" philosophy, and constant effort to keep up with the state-of-the-art practices, he set a personal example for me as a young researcher. I would also like to thank him for his empathy and kindness.

I am extremely grateful to my family and to my partner, Ayelet, for being an endless source of comfort and solace during the most stressful hours. I am especially thankful to my mother whose heroic personal struggle during the past few years has inspired me to overcome any obstacle along the way. Above all, this achievement is the result of her nurturing and the sacrifices she made to educate me and prepare me for my future.

# Abstract

An extended immersed boundary (IB) methodology utilizing a semi-implicit direct forcing approach was formulated for the simulation of incompressible flows in the presence of periodically moving immersed bodies. The methodology utilizes a Schur complement approach to enforce the kinematic constraints of no-slip for immersed surfaces. The methodology is split into an "embarrassingly" parallel pre-computing stage and a time integration stage, both of which take advantage of the general parallel file system (GPFS) for efficient writing and reading of large amounts of data. The developed methodology can be straightforwardly embedded into the whole current family of pressure-velocity segregated solvers of incompressible Navier-Stokes (NS) equations based on the projection or fractional step approaches. It also accurately meets the no-slip kinematic constraints on the surfaces of immersed oscillating bodies. The developed methodology was extensively verified by applying it in the simulation of a number of representative flows developing in the presence of an oscillating sphere.

The capabilities of the methodology in simulating incompressible flow generated by one or several bodies, characterized by generic periodic kinematics, were demonstrated by simulation of the flow developing in the presence of two out-of-phase oscillating spheres and in the presence of a rotating ellipsoid. The physical characteristics of the generated flows, in terms of the time evolutions of the total drag coefficient or, in the latter case, of the torque, are presented as a function of Reynolds values. The vortical structures inherent in the generated flows were visualized by presenting the isosurfaces of the  $\lambda_2$  criterion.

# Table of Contents

<b>1 Introduction and literature survey .....</b>	<b>1</b>
1.1 Motivation of the study .....	1
1.1.1 Periodic movement of bodies in an incompressible fluid .....	1
1.1.2 Moving boundary simulations .....	5
1.2 Objectives of the study.....	9
<b>2 Theoretical background .....</b>	<b>10</b>
2.1 Chapter overview.....	10
2.2 3D incompressible flow induced by a periodically moving immersed body: IB formulation.....	10
2.2.1 Governing equations .....	10
2.2.2 Numerical approach .....	15
2.3 Implementation details.....	18
2.3.1Pre-computing stage.....	18
2.3.2 Time integration stage.....	20
<b>3 Results and discussion .....</b>	<b>23</b>
3.1 Chapter overview.....	23
3.2 Flow around a transversely oscillating sphere .....	24
3.2.1 Verification study .....	27
3.2.2 Grid and time step convergence.....	32
3.2.3 Efficiency characteristics .....	34
3.2.4 Flow simulations.....	37
3.3 Flow around a pair of transversely out-of-phase oscillating spheres.....	42
3.4 Flow around a rotating ellipsoid.....	47
<b>4 Summary and conclusions.....</b>	<b>53</b>
<b>5 Bibliography .....</b>	<b>56</b>

# List of Figures

1. Four classical categories of fish undulatory propulsion illustrated with fish outlines and midlines derived from experimental data (reproduced from [1]). The top half of the figure shows the outlines of typical species with displacements illustrating forward progression over time, while the lower half shows superimposed midlines at equally spaced time intervals throughout a fish tail beat. The presented times refer to the duration of the tail beat, while the velocities refer to the fish swimming speed. ....3
2. (a) Paramecium, covered with thousands of cilia; (b) The locomotion of a single cilium; (c) The collective wave-like motion of an array of cilia on a surface (reproduced from [9]) .....4
3. (a) Experimental set-up for characterizing flow manipulation by magnetically actuated artificial cilia. A rotating magnet, causing each cilium to rotate along a tilted cone, externally actuates the cilia array. This action generates simultaneous pumping and mixing regimes. (b) Measured fluid velocity profile in the microfluidic flow cell. Panels (a) and (b) are both reproduced from [10]. .....4
4. A schematic representation of the major principles of the IBM. The immersed body of elliptical shape is described by set of Lagrangian points indicated by the full black dots. A dashed shell of thickness equal to one grid step, attached to the immersed body, corresponds to the set of discrete volumes surrounding each Lagrangian point. The dashed and dotted circles show the range of action of the regularized Dirac delta functions smearing the Lagrangian forces over the Eulerian grid and interpolating the Eulerian velocities at the Lagrangian points, respectively. Adapted from [54]. .....13
5. A schematic representation of the physical model. A sphere of diameter  $D$  is confined by a prismatic enclosure of dimensions  $4D \times 4D \times 6D$ . The sphere oscillates in the  $z$  direction with amplitude  $A$ . .....25

6. Lagrangian points evenly distributed over the surface of a sphere by the non-iterative method of Leopardi. ....	27
7. Comparison between the time evolutions of non-dimensional drag forces, $F_D$ , obtained for: (a) $(Re, A/D) = (40, 5)$ ; (b) $(Re, A/D) = (40, 0.3125)$ . Solid lines correspond to the currently obtained results, while $\circ$ markers correspond to the values that were digitally scanned from [64]. ....	29
8. Comparison of the peak drag coefficient, $C_{D_{max}}$ , obtained for $Re = 10$ ( $\diamond$ ), $Re = 20$ ( $\times$ ), $Re = 50$ ( $*$ ) and $Re = 100$ ( $\bullet$ ) as a function of the $A/D$ ratio with the corresponding $C_{D_{max}}$ values reported in [65]. ....	30
9. Efficiency characteristics: (a) Memory consumption in pre-computing stage; (b) Memory consumption in time integration stage; (c) Time step duration in pre-computing and time integration stages. ....	37
10. Time evolution of the drag coefficient, $C_D$ , obtained for $Re = 50, 100, 150, 200$ and $A/D = 1$ , superimposed on the time evolution of the position of the sphere, $z - z_0$ (dashed line). ....	38
11. Visualization of the vortical structures generated by a transversely oscillating sphere over a single oscillating period calculated for $Re = 100, 150, 200$ and $A/D = 1$ . ....	40
12. Time evolution of the drag coefficient, $C_D$ , obtained for $Re = 50, 100, 150$ and $200$ (solid line) and $A/D = 0.5$ for: (a) The lower sphere superimposed on the time evolution of its position, $z - \tilde{z}_l$ (dashed line), where $\tilde{z}_l$ is the lower sphere's oscillation center; (b) The upper sphere superimposed on the time evolution of its position, $z - \tilde{z}_u$ (dashed line), where $\tilde{z}_u$ is the upper sphere oscillation center. ....	43

13. Typical pattern of vortical structures characterized by the isosurfaces of $\lambda_2 = -0.1$ , generated by a pair of out-of-phase transversely oscillating spheres over a single oscillation period calculated for $Re = 100, 150$ and $200$ and for $A/D = 0.5$ .....	47
14. A schematic representation of the physical model. The ellipsoid is confined by a prismatic enclosure of dimensions $4D \times 4D \times 6D$ and rotates with a constant angular velocity $\vartheta$ around the $z$ axis. ....	48
15. Time evolution of the torque, $M$ , exerted by the flow on the rotating ellipsoid, obtained for $Re = 100, 200, 300$ , $N = 2$ and $h = 2b$ . ....	51
16. Typical patterns of vortical structures characterized by the isosurfaces of $\lambda_2 = -0.1$ , generated by the rotating ellipsoid for $Re = 100, 200, 300$ , $N = 2$ and $h = 2b$ at: (a) The starting point of the rotation; (b) $1/8$ of the whole turn; (c) $1/4$ of the whole turn; (d) $3/8$ of the whole turn. ....	52

## List of Tables

1. Table 1: Comparison between the present and the previously published $C_{D_{max}}$ values .....	33
2. Table 2: Absolute values of the minimum and maximum $C_D$ acquired for the upper sphere as a function of the Reynolds number .....	44



# Nomenclature

## Initials

$A$	Amplitude of oscillation	[ $m$ ]
$a$	First radii	[ $m$ ]
$b$	Second radii	[ $m$ ]
$c$	Third radii	[ $m$ ]
$D$	Diameter	[ $m$ ]
$C_D$	Drag coefficient	
$F$	Volumetric boundary force (non-dimensional)	
$f$	Volumetric force field	
$F_D$	Drag force (non-dimensional)	
$f_D$	Drag force	[ $N$ ]
$H$	Helmholtz operator	
$h$	Grid spacing	
$I$	Interpolation operator	
$L$	Laplace operator	
$M$	Torque (non-dimensional)	
$N$	Non-linear convective terms of Navier-Stokes equation	
$N$	Number of grid points	
$p$	Pressure (non-dimensional)	
$\tilde{p}$	Order of convergence	
$R$	Regularization operator	
$r$	Distance (non-dimensional)	
$\tilde{r}$	Grid refinement ratio	
$Re$	Reynolds number	
$S$	Corresponds to all Lagrangian grid cells	
$T$	Time	[ $sec$ ]
$t$	Time (non-dimensional)	
$U^I$	Boundary velocity vector (non-dimensional)	
$u$	Velocity vector field (non-dimensional)	
$V$	Volume	[ $m^3$ ]
$X$	Parameterized surface of the immersed body	
$x$	Position vector (non-dimensional)	
$Z$	Generic vector	

### Superscript

$n$	Time step index
*	Intermediate predicted velocity field

### Subscript

0	Corresponds to the initial position
$i$	Index of Eulerian grid points
$i$	General index
$j$	General index
$k$	Index of Lagrangian grid points
$k$	General index
$L$	Corresponds to the Lagrangian grid
$l$	Lower
$u$	Upper
$x$	x direction component
$y$	y direction componet
$z$	z direction component

### Greek letters

$\varepsilon$	X coordinate of the Lagrangian point	
$\eta$	Y coordinate of the Lagrangian point	
$\zeta$	Z coordinate of the Lagrangian point	
$\theta$	Polar angle	[rad]
$\lambda_2$	Median eigenvalue	
$\nu$	Kinematic viscosity	[m <sup>2</sup> /sec]
$\rho$	Mass density	[kg/m <sup>3</sup> ]
$\phi$	Azimutal angle	[rad]
$\omega$	Frequency of oscillation	[1/sec]
$\Omega$	Corresponds to all Eulerian grid cells	

# Chapter 1

## Introduction and literature survey

### 1.1 Motivation of the study

#### 1.1.1 Periodic movement of bodies within incompressible fluid

The periodic movement of bodies immersed in an incompressible Newtonian fluid is ubiquitous in various biological systems and in many engineering and industrial applications. Indeed, the locomotion of most elongated aquatic vertebrates is based on undulatory propulsion, i.e., the sequential periodic activation of the different segments of a body. In an extensive review done by Lauder and Tytell [1], fish swimming is classified into four classical categories while a different general "mode" of undulatory motion characterizes each one (see Figure 1). In engineering, this mechanism is mimicked in bio-inspired synthetic micro- and nano-swimmers, which rely on the periodic external actuation of small rigid members in their locomotion [2]-[4]. In this type of application, propulsion is typically provided by the symmetry-breaking excitation of the surrounding flow by the periodically moving body of the swimmer.

The undulatory propulsion mechanism is also present in a number of other organisms on the cellular level, as reported in [5]-[8]. One fascinating example can be found in the fluid manipulation mechanism of ciliary systems, extensively reviewed in [7]. Ciliary systems consist of arrays of microscopic slender flexible rods (cilia), which may beat back and forth in recovery and effective strokes, respectively. These systems derive a beneficial propulsive effect by synergetic action obtained from a phase relationship between the beats of neighboring cilia. This phase relationship results in a wave travelling over the array of cilia. An example of ciliary motion enhanced effectiveness in flow generation can be found

in micro-organisms such as the Paramecium [9], which is about 100  $\mu\text{m}$  long and contains about 4000 cilia on its outer surface; the swimming speed of the Paramecium can reach up to ten times its own length per second. Figure 2 shows arrays of cilia covering the surface of the Paramecium, as well as the asymmetric motion of a single cilium and the wave-like motion of an entire array. Recently the idea of a bio-mimetic ciliary system, composed from functional arrays of artificial cilia, has been envisaged as a novel strategy for microfluidic manipulation [10]-[15]. In this kind of application, microscopic cilium-like actuators - moving in a periodic cycle consisting of effective and recovery strokes as mentioned previously - are externally actuated by various kinds of stimuli, such as an electrostatic field, a magnetic field or light. These devices have shown the capability of low-Reynolds-number fluid propulsion and mixing. Figure 3 shows the experimental design used in [10] to characterize flow manipulation by magnetically actuated artificial cilia, along a measured induced fluid velocity profile in an experimental microfluidic flow cell. Numerical modeling of the fluid flow induced by the artificial cilia's periodic movement can be of great importance in analyzing the working principle of the cilia, as well as in designing and optimizing these kinds of devices. Similarly, this kind of numerical modeling can produce clear benefits in the design of any active micro- and macro-mixer based on the periodic movement of externally activated rigid bodies. These kinds of mixer, implemented with rotating or oscillating stir-bars of various shapes, are typical in the chemical and medical industries [16]-[19].

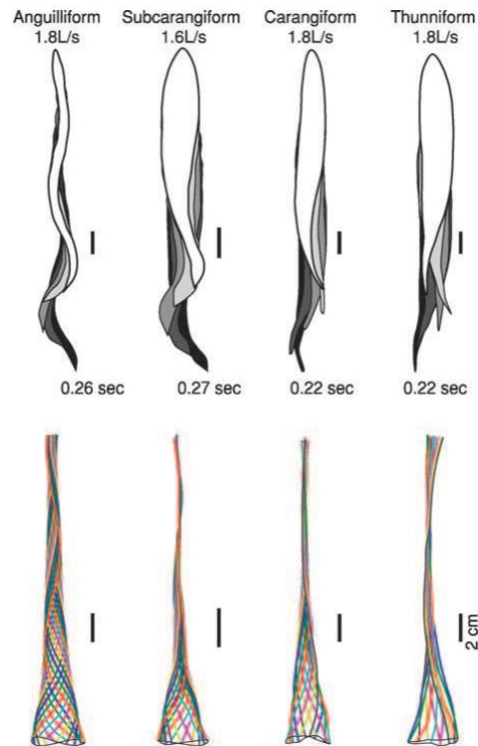
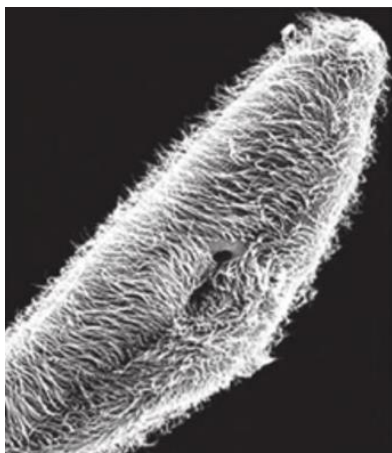
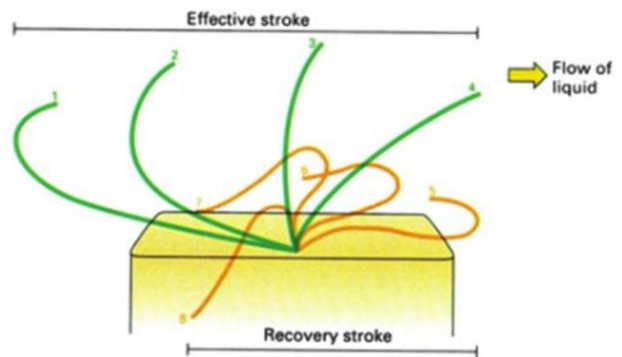


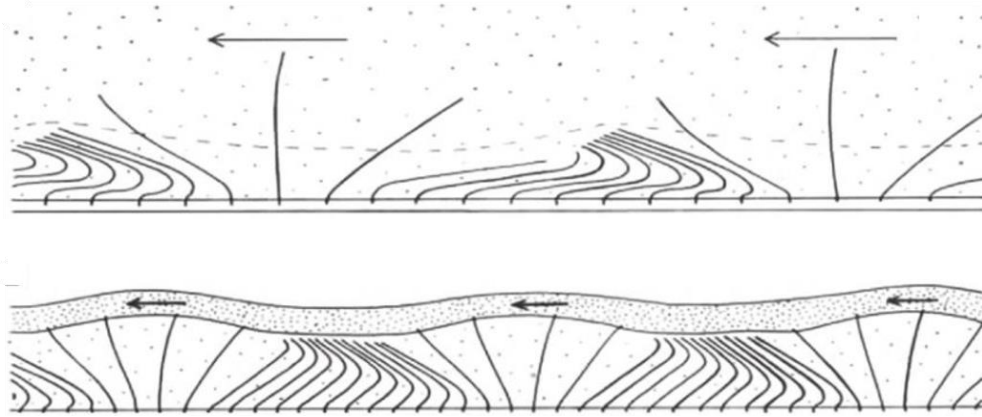
Figure 1: Four classical categories of fish undulatory propulsion illustrated with fish outlines and midlines derived from experimental data (reproduced from [1]). The top half of the figure shows the outlines of typical species with displacements illustrating forward progression over time, while the lower half shows superimposed midlines at equally spaced time intervals throughout a fish tail beat. The presented times refer to the duration of the tail beat, while the velocities refer to the fish swimming speed.



(a)

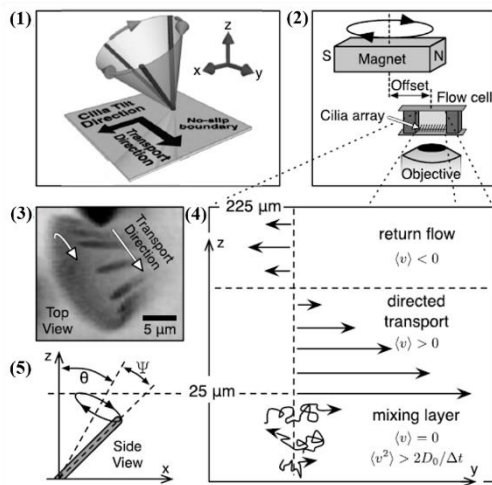


(b)

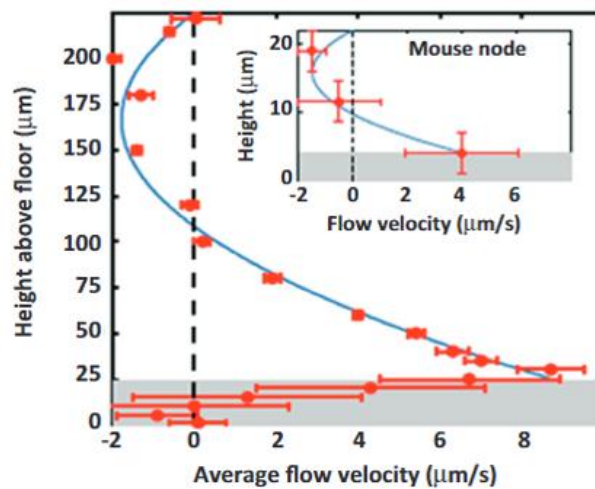


(c)

Figure 2: (a) *Paramecium*, covered with thousands of cilia; (b) The locomotion of a single cilium; (c) The collective wave-like motion of an array of cilia on a surface (reproduced from [9])



(a)



(b)

Figure 3: (a) Experimental set-up for characterizing flow manipulation by magnetically actuated artificial cilia. A rotating magnet, causing each cilium to rotate along a tilted cone, externally actuates the cilia array. This action generates simultaneous pumping and mixing regimes. (b) Measured fluid velocity profile in the microfluidic flow cell. Panels (a) and (b) are both reproduced from [10].

### 1.1.2 Moving boundary simulations

To simulate the fluid flow developing in the presence of periodically moving immersed bodies, it is necessary to perform moving boundary simulations. This kind of simulation is based on the key idea of determining the kinematics of the immersed body, and then imposing no-slip boundary conditions on the body surface at each computational time step. The major challenge in performing such simulations with a traditional body-conformal grid approach lies in the generation of a high-quality mesh that should, firstly, be capable of handling complex body geometries and, secondly, be rebuilt in each computational time step. Attempts to handle this challenge have been made in fluid-structure interaction analyses based on the arbitrary Lagrangian-Eulerian (ALE) formulation [20]. The ALE formulation uses a computational mesh that is neither a priori fixed in space, nor attached to a body, but, rather, can move arbitrarily to meet the boundary conditions on the surface of the solid body in a more flexible and accurate manner. Nevertheless, using this technique to tackle a freely moving solid body in a fluid domain poses a serious challenge, as it is necessary to ensure that the fluid mesh following the interface of the solid is not distorted or tangled upon large motions of the solid body. Although a possible remedy for this problem could come from the use of adaptive re-meshing techniques [21], these techniques are typically time prohibitive and require remapping of the field variables between the source and target meshes. Recently, the use of novel high-order sliding mesh methods [22]-[24] has been proven effective for solving moving interface problems, yet the ever-increasing demand for the analysis of flows typical in realistic complex engineering systems can lead to serious deterioration of the computational efficiency of these simulations. A promising alternative to these body-conformal grid approaches is the immersed boundary method (IBM). Initially developed by Peskin [25], the method has become very popular over the years and is currently widely

used for flow simulations in the presence of stationary and moving immersed bodies of complex geometry, as extensively reviewed in [26].

Among a number of existing methodologies, the present study focuses on the direct forcing approach, initially introduced in [27]. According to this approach, the impact of the immersed body (determined as a set of discrete Lagrangian points) on the surrounding flow is expressed by introducing additional unknowns in the form of volumetric forces, each associated with a corresponding Lagrangian point. The forces act as Lagrange multipliers that enforce a no-slip kinematic constraint on the surface of the immersed body, without involving any dynamical process. In the most general case, the Lagrangian points of the body surface do not coincide with the underlying Eulerian Cartesian grid. Information is exchanged between the grids through two adjoint operators, namely, the interpolation operator, which interpolates the velocity values from the Eulerian grid to the Lagrangian points, and the regularization operator, which smears the volumetric forces from the Lagrangian points on the underlying Eulerian grid. Chronologically, the implementation of the direct forcing IBM was first based on explicit calculation of the Lagrangian forces. The explicit formulation gained great popularity, since it does not require any modification to the original solver of the Navier-Stokes (NS) equations. In practice the forces are directly solved from the NS equations according to the following formulation:

$$\mathbf{F} = \frac{\mathbf{U}^\Gamma - \mathbf{u}^n}{\Delta t} - \mathbf{RHS}^n \quad (1.1)$$

where  $\mathbf{F}$  is the direct forcing term,  $\mathbf{u}^n$  is the velocity at the n-th time step,  $\mathbf{U}^\Gamma$  is the preset boundary velocity field and the  $\mathbf{RHS}^n$  term contains the convective, viscous and pressure gradient terms of the NS equations. After  $\mathbf{F}$  is determined, it is smeared over the neighboring Eulerian grid nodes using the regularization operator. In other words, to equip any existing NS solver with the immersed boundary (IB) functionality, the original solver



should be employed twice: the first time without taking into account the existence of the IB and the second time by employing the same differential operator with the modified right hand side (RHS) containing explicitly calculated Lagrangian forces to reflect the impact of the IB on the surrounding flow. The high flexibility of the explicit direct forcing IBM is quite impressive. It has been successfully employed over the past two decades in a broad spectrum of fields, including in the simulation of particulate flows [28]-[33], thermally driven confined flows [34]-[37], two-phase immiscible flows [38] and in the phenomenological modeling of the mobility and growth of cancerous tumors [39]-[41].

There are, however, two major drawbacks in the explicit calculation of the Lagrangian forces. First, to satisfy no-slip kinematic constraints with acceptable accuracy the simulations should be run with very small time steps. A number of attempts have been made to relax the small time step limitation; these have included solving the coupled system of the boundary forces [35]-[36], iteratively updating the velocity at a sub-time step based on its difference from the desired boundary velocity [42]-[43] and introducing an additional forcing loop [30]-[31] for more accurate imposition of the interface velocity. However, despite the evident success of the aforementioned studies in imposing no-slip kinematic constraints, none of the proposed strategies can be seen as an ultimate remedy to the second – and the most important – drawback, related to the explicit calculation of the Lagrangian forces, namely, the violation of the elliptic character of the NS equations. The problem shows up in the simulation of flows characterized by low and moderate Reynolds numbers, but it can be solved by employing a fully coupled formulation. The key idea is to express the Lagrangian forces as additional unknowns in the form of distributed Lagrange multipliers (DLM) implicitly embedded into the corresponding NS equations. In this case, the kinematic constraints of no-slip are imposed with a machine zero-precision. Recent progress in this direction is due to the works in [44]-[48], which successfully established

the DLM approach as a powerful tool for investigating incompressible flows for a wide range of Reynolds numbers, including linear stability analysis of pressure and thermally driven flows [49] and analysis of two-phase immiscible flows [50]. It is also worth mentioning the work in [51] that can be seen as a close relative of the DLM approach. In this approach, the pressure and boundary forces are treated as a single set of Lagrange multipliers that are determined implicitly from a modified Poisson equation; then, a single projection step is introduced to satisfy the continuity equation and the no-slip condition, in a manner analogous to the traditional fractional step method. This approach has been successful, under certain conditions, in achieving second-order accuracy in time. Unfortunately, these purely implicit implementations of the IBM typically involve substantial modification of the original solvers, which are not initially equipped with the IBM capability. For this reason, a semi-implicit implementation of the IBM, in which the Lagrangian forces are implicitly coupled with a non-solenoidal velocity field, further projected to the divergence-free subspace, has gained increasing interest over the last decade [52]-[55]. The semi-implicit formulation of the IBM can be straightforwardly embedded into the whole family of pressure-velocity-segregated NS solvers based on the projection or the fractional step algorithms, while maintaining the accuracy of the imposed constraints of no-slip bounded by the discretization error of the numerical scheme [54]. The key idea that stands at the basis of the semi-implicit implementation is the analytic decomposition of the operator coupling the NS equations with the constraints of no-slip, and the pre-computation of the contribution of the latter at the beginning of the computational process. It was shown in [54] that, upon completing the pre-computation step, the efficiency of the time integration performed by the algorithm based on the semi-implicit implementation of the IBM is comparable with its explicit counterpart when flows in the presence of stationary immersed bodies are considered.

## 1.2 Objectives of the study

The present work aims to extend the methodology previously developed in [54] to flow simulations in the presence of periodically moving immersed bodies whose kinematics is governed by periodic functions, and can therefore be described by a finite amount of repeating boundary configurations. This feature is exploited in the present algorithm to execute the pre-computation step described in [54] on each of the boundary configurations in an "embarrassingly" parallel fashion, which enables its efficient acceleration without introducing any significant modifications to the original solver. Then, by taking advantage of the general parallel file system (GPFS), and the massively parallel MUMPS solver, every pre-computed state can be saved, and, most importantly, read later very efficiently, making it possible to perform this kind of simulation in a reasonable amount of time.

The current study presents the extended methodology, including a detailed description of the algorithm, a verification study of the obtained results, and an appraisal of the efficiency characteristics in terms of memory and time consumption. The developed method was extensively verified by comparing the flow characteristics obtained from simulation of confined flows developing in the presence of an oscillating sphere with results available in the literature. The capability of the developed methodology to resolve the flows developing in the presence of rotating bodies is demonstrated by performing simulations of the flow generated by a rotating ellipsoid for different values of Reynolds number.

# Chapter 2

## Theoretical background

### 2.1 Chapter overview

In this chapter, the mathematical formulations of the research problem and of the novel methodology developed and utilized for the numerical solution of the problem are described. The study focuses on incompressible flows driven by immersed bodies characterized by some kind of pre-determined periodic kinematics. The equations governing the flow developing in the presence of a general periodically moving immersed body are given in section 2.2. Emphasis is put on building the theoretical model in accordance with the formalism of the immersed body method. Additionally, the numerical approach is elaborated and the main ideas of the semi-implicit direct forcing IB implementation for periodically moving boundaries are laid out. Finally, in section 2.3, the implementation details of the developed methodology are given, including a detailed explanation of the techniques used to exploit the structure of the problem in order to extract parallelism in the time direction.

### 2.2 3D incompressible flow induced by a periodically moving immersed body: IB formulation

#### 2.2.1 Governing equations

The incompressible flow developing in the presence of an immersed body whose parameterized surface  $\mathbf{X}(\xi, \eta)$  is moving in accordance with a priori determined kinematics, given the surface velocity  $\mathbf{U}^\Gamma(\mathbf{X})$ , is governed by the incompressible

momentum and continuity equations. Formulated in Cartesian coordinates and written in dimensionless form, the governing equations read as:

$$\nabla \cdot \mathbf{u} = 0, \quad (2.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u} + \mathbf{f} \quad (2.2)$$

where  $\mathbf{u}(u, v, w)$ ,  $p$ , and  $t$  are the dimensionless velocity, pressure and time, respectively.

The volumetric force field  $\mathbf{f}(x)$  in (2.2) reflects the impact of the moving boundary of the immersed body on the surrounding flow. Note that, generally, the value of  $\mathbf{f}(x)$  is unknown, and it is therefore necessary to introduce a kinematic constraint of no-slip:

$$\mathbf{u}(\mathbf{X}) = \mathbf{U}^\Gamma(\mathbf{X}) \quad (2.3)$$

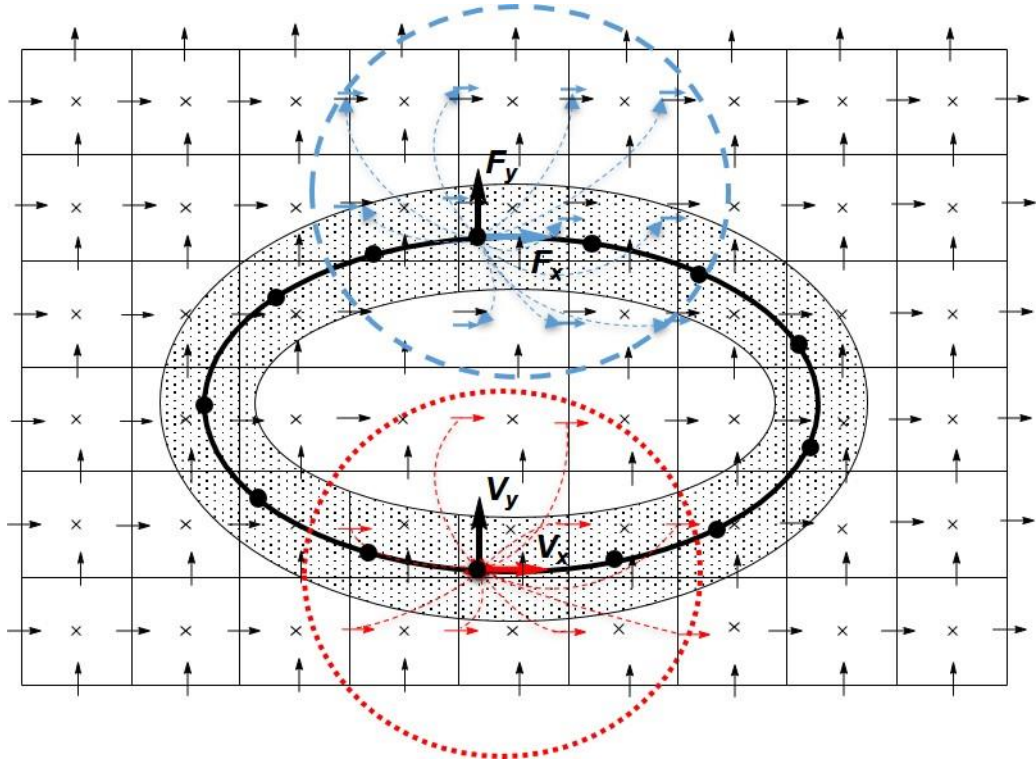
providing closure of the overall system of Eqs. (2.1)-(2.3). Note that while Eqs. (2.1)-(2.2) are formulated on a Cartesian grid, Eq. (2.3) holds only for the domain  $(\mathbf{X})$ , which includes the surface of the immersed body. Distinction between variables related to the Cartesian grid and variables related to the grid coinciding with the immersed body surface is provided by assigning them lowercase and uppercase, respectively. In accordance with the IBM formalism, all the variables related to the Cartesian grid are called Eulerian variables, while the Cartesian grid is called a Eulerian grid. The surface of the immersed body is determined by a set of discrete points, which are called Lagrangian points, and all the variables related to these points are called Lagrangian variables. Figure 1 shows the set-up of a uniform structured staggered 2D Eulerian grid underlying a set of Lagrangian points determining the surface of an immersed body of elliptical shape. In general, the Lagrangian points do not coincide with the underlying Eulerian grid. Therefore, there is a need to introduce two adjoint operators for the exchange of information between the grids, namely, the

interpolation operator,  $\mathbf{I}$ , used to interpolate the Eulerian velocities at the locations of the Lagrangian points, and the regularization operator,  $\mathbf{R}$ , used to smear the volumetric Lagrangian forces on the adjacent Eulerian grid:

$$\mathbf{I}(u(x_i)) = \int_{\Omega} (u(x_i)) \cdot \delta(\mathbf{X}_k - x_i) dV_{\Omega_i} \quad (2.4)$$

$$\mathbf{R}(\mathbf{F}_k(\mathbf{X}_k)) = \int_S (\mathbf{F}_k(\mathbf{X}_k)) \cdot \delta(x_i - \mathbf{X}_k) dV_{S_k} \quad (2.5)$$

where the indices  $i$  and  $k$  run for the whole discrete range of Eulerian and Lagrangian coordinates, respectively, and the indices  $\Omega$  and  $S$  read for the Eulerian and Lagrangian grid cells, respectively, thus determining  $dV_{\Omega_i}$  as the  $i^{\text{th}}$  volume of the Eulerian flow domain and  $dV_{S_k}$  as the virtual volume confining the  $k^{\text{th}}$  Lagrangian point. To achieve the best accuracy, a uniform grid in the vicinity of the immersed body surface is utilized. The surface of the immersed body is determined by a set of equispaced Lagrangian points, and the distance between the neighboring Lagrangian points is approximately the same as the cell width of the underlying Eulerian grid, also yielding  $dV_{\Omega_i} \approx dV_{S_k}$ .



*Figure 4: A schematic representation of the major principles of the IBM. The immersed body of elliptical shape is described by set of Lagrangian points indicated by the full black dots. A dashed shell of thickness equal to one grid step, attached to the immersed body, corresponds to the set of discrete volumes surrounding each Lagrangian point. The dashed and dotted circles show the range of action of the regularized Dirac delta functions smearing the Lagrangian forces over the Eulerian grid and interpolating the Eulerian velocities at the Lagrangian points, respectively. Adapted from [54].*

Among a wide variety of possible regularized Dirac delta functions that can be utilized in (2.4)-(2.5), the function introduced by Roma et al. in [56] was chosen:

$$\delta(r) = \begin{cases} \frac{1}{6\Delta r} \left[ 5 - 3\frac{|r|}{\Delta r} - \sqrt{-3\left(1 - \frac{|r|}{\Delta r}\right)^2 + 1} \right] & \text{for } 0.5\Delta r \leq |r| \leq 1.5\Delta r \\ \frac{1}{3\Delta r} \left[ 1 + \sqrt{-3\left(\frac{|r|}{\Delta r}\right)^2 + 1} \right] & \text{for } |r| \leq 0.5\Delta r \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

Here,  $\Delta r$  is the cell width in the  $r$  direction, which means that the above delta function supports three grid cells in each spatial direction, while interpolating Eulerian velocities and regularizing Lagrangian forces (the same delta function is utilized in both the interpolation and regularization operators). The chosen delta function has been specifically designed for performing calculations on staggered grids and has gained popularity over recent years [28],[30],[42],[48],[52] due to its compact kernel (only 3 cells in each direction of the computational domain). Interpolation of the discrete Eulerian velocities  $\mathbf{u}_i$  and regularization of the discrete Lagrangian forces  $\mathbf{F}_k$  for the 3D configuration are performed by employing the following formulas:

$$\mathbf{U}_k^\Gamma = \Delta x^3 \sum_i \mathbf{u}_i \delta(x_i - \varepsilon_k) \delta(y_i - \eta_k) \delta(z_i - \zeta_k) \quad (2.7)$$

$$\mathbf{f}_i = \Delta x^3 \sum_k \mathbf{F}_k \delta(\varepsilon_k - x_i) \delta(\eta_k - y_i) \delta(\zeta_k - z_i) \quad (2.8)$$

yielding the resultant discrete boundary velocity at the  $k^{\text{th}}$  Lagrangian point,  $\mathbf{U}_k^\Gamma$  and the discrete volumetric force  $\mathbf{f}_i$  at the  $i^{\text{th}}$  point  $(x_i, y_i, z_i)$  of the Eulerian staggered grid.



### 2.2.2 Numerical approach

Following the SIMPLE method [57], the system of equations (2.1)-(2.3) equipped with the IB capability are transformed into the following set of discrete equations:

$$\frac{1}{\text{Re}} \mathbf{L}(\mathbf{u}^*) - \frac{3\mathbf{u}^*}{2\Delta t} + \mathbf{R}(\mathbf{F}_k(\mathbf{X}_k)) = \frac{-4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t} + \mathbf{N}(\mathbf{u}^n) + \nabla p^n \quad (2.9)$$

$$\mathbf{I}(\mathbf{u}^*(x_i)) = \mathbf{U}^\Gamma(\mathbf{X}_k) \quad (2.10)$$

$$\Delta(\delta p) = \frac{3}{2\Delta t} \nabla \cdot \mathbf{u}^* \quad (2.11)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{2\Delta t}{3} \nabla(\delta p), \quad p^{n+1} = p^n + \delta p \quad (2.12)$$

where the second-order backward finite difference scheme was utilized for the temporal discretization, while a standard staggered mesh second-order conservative finite-volume formulation [57] was used for the discretization of all spatial derivatives. Eqs. (2.9)-(2.10) play the role of predictor, yielding a non-solenoidal velocity field  $\mathbf{u}^*$  by taking the values of the pressure field  $p^n$  from the previous time step, and Eqs. (2.11)-(2.12) are formulated to correct the pressure field and to project the predicted velocity onto the divergence-free subspace. Neumann boundary conditions with a single reference Dirichlet point introduced anywhere within the computational domain are used for the solution of the Poisson equation, Eq. (2.11). The linear terms  $\mathbf{L}$  and  $\mathbf{R}$  that enter into the momentum equation (2.9) and correspond to the Laplace and regularization operators, respectively, are treated implicitly, while the nonlinear convective terms  $\mathbf{N}$  are treated explicitly by calculating their values from the previous time step.

Explicit treatment of the nonlinear terms allows us to solve Eqs. (2.9)-(2.10) successively for each component of the predicted velocity vector  $\mathbf{u}^*$  and then to continue with a standard projection-correction step, determined by Eqs. (2.11)-(2.12). Eqs. (2.9)-(2.10) can be rewritten in a compact block-matrix form:

$$\begin{bmatrix} \mathbf{H} & \mathbf{R} \\ \mathbf{I} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^* \\ \mathbf{F} \end{bmatrix} = \begin{bmatrix} \mathbf{RHS}^{n-1,n} \\ \mathbf{U}^\Gamma \end{bmatrix} \quad (2.13)$$

where  $\mathbf{H} = \frac{1}{\text{Re}} \mathbf{L} - \frac{3}{2\Delta t} \mathbf{I}$  is the Helmholtz operator acting on each component of the predicted vector  $\mathbf{u}^*$ ,  $\mathbf{I}$  is the unity matrix,  $\mathbf{R}$  and  $\mathbf{I}$  are rectangular matrices that contain terms resulting from applying regularization and interpolation operators, respectively. and  $\mathbf{RHS}^{n-1,n}$  stands for the right-hand side vector containing the pressure gradient and nonlinear convective terms known from the previous time steps. The next section will focus on the solution strategy of the system of Eqs. (2.13) by utilizing the domain decomposition technique. Suppose that a generic package capable of solving the Helmholtz equation (2.13) is available:

$$[\mathbf{H}][\mathbf{u}^*] = [\mathbf{RHS}^{n-1,n}] \quad (2.14)$$

Assume also that the above package can be used as a black box driver (i.e., without introducing any modifications into the package itself) to obtain  $\mathbf{u}^*$ . In general, such a driver can be one of two types: either a standalone solver capable of solving the Helmholtz equation (2.13) (e.g., FISHPACK [58]) or a whole CFD package for the simulation of incompressible flows from the family of projection or fractional step solvers (e.g., openFoam [59]). When utilizing a driver of the first type, one should explicitly build and provide to the driver the matrix  $\mathbf{H}$  and the vector  $\mathbf{RHS}^{n-1,n}$  with appropriate boundary conditions, while utilizing a driver of the second type will require only proper determination

of the boundary conditions. Furthermore, drivers of both types can also be used for calculating the product of  $\mathbf{H}^{-1}$  by any generic vector  $\mathbf{Z}$ . For a driver of the first type, the product is obtained by providing the package with an already existing matrix  $\mathbf{H}$  and with a modified *RHS* vector whose values are now equal to the values of the  $\mathbf{Z}$  vector. For a driver of the second type, the only requirement is a modification of the *RHS* vector. To demonstrate the capabilities of the developed methodology, the solver of the second type developed in [60] is used as the driver. Keeping in mind the above capabilities, we perform an analytic transformation of the system of Eqs. (2.14), known as a Schur complement decomposition:

$$\mathbf{F} = [\mathbf{IH}^{-1}\mathbf{R}]^{-1} [\mathbf{IH}^{-1}\mathbf{RHS}^{n-1,n} - \mathbf{U}^\Gamma] \quad (2.15)$$

$$\mathbf{u}^* = \mathbf{H}^{-1} [\mathbf{RHS}^{n-1,n} - \mathbf{RF}] \quad (2.16)$$

The idea is first to find the values of the distributed Lagrange multiplier (DLM),  $\mathbf{F}$ , providing the kinematic constraints of no-slip, and thereafter to use them to find the values of the predicted velocity vector,  $\mathbf{u}^*$ . Note that, despite the fact that the process is separated into two stages, the obtained  $\mathbf{F}$  and  $\mathbf{u}^*$  values are fully coupled, as the solution procedure is implicit. Obtaining the solution of the whole problem by a straightforward application of Eqs. (2.15) and (2.16) would require us to perform inversion of the matrices  $[\mathbf{H}]$  and  $[\mathbf{IH}^{-1}\mathbf{R}]$ , which is computationally prohibitive. Instead, the solution can be obtained by calculation of the series of the matrix-vector products of the matrix  $[\mathbf{H}^{-1}]$  and further LU decomposition of the small matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$ . Most of the calculations required to complete the stages described in Eqs. (2.15) and (2.16) can be pre-computed once at the beginning

of the computational process and then reused throughout the simulation. The implementation details of the developed methodology are given in the next section.

## 2.3 Implementation details

Prior to describing the numerical procedure developed for the solution of Eqs. (2.15) and (2.16), we make a number of observations regarding the structure of matrices  $\mathbf{I}$  and  $\mathbf{R}$ . We recall that the matrices contain the terms that were obtained by employing the interpolation and the regularization operators, respectively, to enforce the kinematic constraints of no-slip on the surface of the immersed body. The characteristic property of these two matrices is their extreme sparseness, resulting from the compact kernel of the utilized discrete delta function. In fact, considering a typical 3D problem characterized by  $O(10^6)$ - $O(10^7)$  degrees of freedom for each velocity component, a row of matrix  $\mathbf{I}$  (or alternatively a column of matrix  $\mathbf{R}$ ) contains only  $O(10^2)$  non-zero values. As a result, both the  $\mathbf{I}$  and  $\mathbf{R}$  matrices can be stored in compressed sparse row (CSR) format, while their matrix-vector product can be efficiently performed by employing standard routines from the Intel Math Kernel Library (MKL). We next give a detailed description of both the pre-computing and the time integration stages of the developed methodology.

### 2.3.1 Pre-computing stage

Recalling that the present methodology was developed for flow simulation in the presence of periodically oscillating immersed bodies, we divide the oscillating period into an integer number of cyclically repeated time steps. All the procedures described in this section are performed for each time step entering a single oscillating period.

Calculation of matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$

The matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$  is a square matrix of dimensions  $(m \times m)$ , where  $m$  is the total number of Lagrangian points determining the surfaces of all the immersed bodies involved in the simulation. For a typical 3D problem,  $m$  lies within the range of  $m \in [O(10^3) \div O(10^4)]$ . Note, also, that following the study in [54] an absolute value of the sparsing threshold,  $s$ , was set to  $s = 10^{-21}$ . Consequently, only entries with an absolute value higher than the value of  $s$  were stored in the matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$ . The matrix itself was stored in a row-column-value (RCV) format, as follows:

- For each column  $[\mathbf{R}]_k$ ,  $1 \leq k \leq m$  of the matrix  $\mathbf{R}$  :
  - Employ a generic driver of either the first or the second type, with a modified right hand side  $[RHS^{n-1,n}] = [\mathbf{R}]_k$  to calculate the product  $\mathbf{H}^{-1}[\mathbf{R}]_k$  ;
  - Multiply the matrix  $\mathbf{I}$  stored in CSR format by the obtained vector, employing standard routines for the Intel MKL;
  - Fill the  $k^{th}$  column  $[\mathbf{IH}^{-1}\mathbf{R}]_k$  of the matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$  in the RCV format, while employing the sparsing threshold,  $s$  .

Note that the above process makes it possible for us to avoid the creation of intermediate matrices of large dimensions and directly builds the matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$  of dimensions  $(m \times m)$ . An additional important observation is that all 3 stages of the above algorithm are independent of both the column number  $k$  and the time step  $\Delta t$ , which makes the whole pre-computing stage "embarrassingly" parallel. The parallelism can be exploited on two levels: first, while building the matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$ , i.e., the columns of the matrix can be calculated separately and then collected into the whole matrix in order to perform the

calculation stage<sup>1</sup>, and second, while calculating the matrices  $[\mathbf{IH}^{-1}\mathbf{R}]$  corresponding to different time steps entering into a single period<sup>2</sup>.

*Factorization of matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$*

Instead of direct calculation of the inverse of the matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$  we perform *LU* factorization, which then allows us to obtain the product of  $[\mathbf{IH}^{-1}\mathbf{R}]^{-1}$  by any generic vector of appropriate length. *LU* factorization was performed by utilizing an open-source MUMPS solver [61], [62], and the *LU* factors calculated for each time step were stored on the hard disk<sup>3</sup>.

### 2.3.2 Time integration stage

After completing the pre-computing stage, the time integration stage can be initiated. The implementation details of all the steps required to complete the solution of Eqs.(2.15) and (2.16) for one time step are as follows:

- Calculate  $[\mathbf{IH}^{-1}\mathbf{RHS}^{n-1,n}]$ :
  - Employ a generic driver of either the first or the second type, with its original RHS to calculate the product  $[\mathbf{IH}^{-1}\mathbf{RHS}^{n-1,n}]$ ;
  - Multiply the matrix  $\mathbf{I}$  stored in CSR format by the obtained vector, employing standard routines from the Intel MKL.
- Calculate  $[\mathbf{IH}^{-1}\mathbf{RHS}^{n-1,n} - \mathbf{U}^\Gamma]$  simply by subtracting two vectors.

---

<sup>1</sup> This strategy is preferable for stationary immersed bodies.

<sup>2</sup> This strategy is preferable for periodically oscillating immersed bodies, and was utilized in the present study.

<sup>3</sup> A still not released version of MUMPS, allowing for the storage and reconstruction of *LU* factors, was provided by the MUMPS developers, who can be contacted through the software website <http://mumps.enseeiht.fr/>.

- Calculate  $\mathbf{F}^4$  (See Eq.(2.15)):
  - Retrieve the  $LU$  factors of the matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$ , which were calculated at the pre-computing stage for every time step and stored on the hard disk;
  - Assign  $[\mathbf{IH}^{-1}RHS^{n-1,n} - \mathbf{U}^\Gamma]$  to the RHS of Eq. (2.15);
  - Perform standard backward and forward substitutions of the factorized matrix  $[\mathbf{IH}^{-1}\mathbf{R}]$  with respect to the RHS built in the previous step.
- Calculate  $\mathbf{u}^*$  (See Eq. (2.16)):
  - Multiply the matrix  $\mathbf{R}$  stored in CSR format by the calculated DLM vector  $\mathbf{F}$ , employing standard routines from the Intel MKL;
  - Calculate  $[RHS^{n-1,n} - \mathbf{RF}]$  simply by subtracting two vectors;
  - Employ a generic driver of either the first or the second type, with a modified RHS equal to  $[RHS^{n-1,n} - \mathbf{RF}]$ .

Note that after completing the pre-computing stage, the time integration of the presently developed methodology is based on double implementation of an original generic driver (of either the first or the second type). Namely, for the first time when calculating  $[\mathbf{IH}^{-1}RHS^{n-1,n}]$  and for the second time when calculating  $\mathbf{H}^{-1}\mathbf{R}$  products - the same as when utilizing any fully explicit formulation of the direct forcing IBM. As a result, not only does the developed semi-implicit methodology provide a more accurate imposition of the kinematic constraints of no-slip on the surfaces of the periodically oscillating immersed

---

<sup>4</sup> All the stages within the bullet are automated in MUMPS.

bodies, but it is also as fast as its fully explicit counterpart in terms of time efficiency of the time integration stage.



# Chapter 3

## Results and discussion

### 3.1 Chapter overview

In this chapter, the results obtained in the framework of the current study are presented and discussed. The first part of the chapter, section 3.2, is dedicated to extensive verification of the developed methodology through a series of benchmark tests using the well-studied configuration of an incompressible flow driven by a transversely oscillating sphere in an otherwise quiescent fluid. This flow configuration is further used to assess the sensitivity of the developed method to the spatial grid size and time step values (grid convergence study), and to analyze its efficiency characteristics in terms of RAM memory usage and time step duration.

The latter part of this chapter (starting from 3.2.4) presents a series of demonstrations of the capabilities of the developed method in simulating the incompressible flow generated by a body characterized by periodic kinematics. The already mentioned oscillating sphere configuration, as well as the configuration of a rotating ellipsoid, are simulated for different values of Reynolds number. The physical characteristics of the flows in terms of the time evolutions of the total drag coefficient and the torque are presented as a function of Reynolds values. The vortical structures inherent in these flows are visualized by presenting the isosurfaces of the  $\lambda_2$  criterion.

### 3.2 Flow around a transversely oscillating sphere

An oscillating sphere of diameter  $D$  in an otherwise quiescent fluid confined by a rectangular prism of dimensions  $4D \times 4D \times 6D$  is considered. A schematic of the physical model is given in Figure 2. The sphere oscillates in the  $z$  direction with periodic velocity  $U_z$  given by:

$$U_z = U_{\max} \sin(\omega T) \quad (3.1)$$

where  $\omega$  is the angular oscillating frequency and  $T$  is the dimensional time. Setting the coordinate system origin at the center of the prism (as shown in Figure 2), the time dependent location of sphere's center of mass can be obtained by integrating Eq. (3.1) over time to yield:

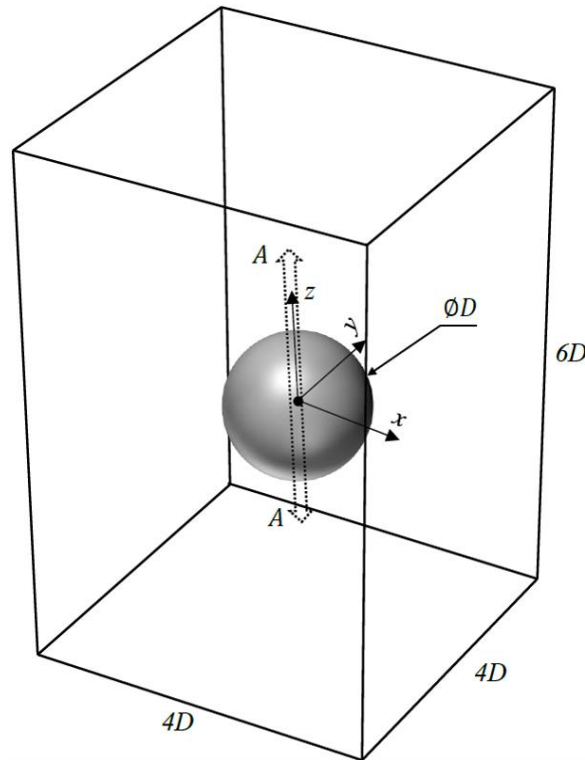
$$\mathbf{x}_c = \left[ 0 \quad 0 \quad -\frac{U_{\max}}{\omega} \cos(\omega T) \right] \quad (3.2)$$

No-slip boundary conditions are applied to the surface of the sphere and to all the walls of the computational domain. The no-slip boundary conditions are chosen as: first, they can be easily reproduced in future realistic experiments; and second, the computational domain chosen is large enough so that the wall shear effect is negligibly small. After utilizing the values of  $D$ ,  $U_{\max}/D$  and  $\rho U_{\max}^2$  for scaling the length, velocity, time and pressure fields, respectively, the kinematics of the oscillating sphere is governed by the following non-dimensional equations:

$$\mathbf{u}_c = \left[ 0 \quad 0 \quad u_z = \sin\left(\frac{D}{A} t\right) \right] \quad (3.3)$$

$$\mathbf{x}_c = \left[ 0 \quad 0 \quad z = -\frac{A}{D} \cos\left(\frac{D}{A} t\right) \right] \quad (3.4)$$

where  $A = U_{\max}/\omega$  corresponds to the oscillation amplitude of the sphere. After substituting the above kinematics in (2.1)-(2.3), it is clear that the flow is governed by two non-dimensional parameters, namely, the Reynolds number,  $Re = U_{\max}D/\nu$ , and the ratio of the oscillation amplitude to the sphere diameter,  $A/D$ . It can be noticed that for the choice of  $A = U_{\max}/\omega$ , the  $A/D$  ratio is equivalent to the Strouhal number ( $St = \omega D/U_{\max}$ ), typically used to normalize frequency in oscillatory flows.



*Figure 5: A schematic representation of the physical model. A sphere of diameter  $D$  is confined by a prismatic enclosure of dimensions  $4D \times 4D \times 6D$ . The sphere oscillates in the  $z$  direction with amplitude  $A$ .*

Next, the force balance of the accelerating sphere is considered. Recalling that the sphere is filled with the same fluid as that outside of it (in accordance with the IB method formalism), the force balance can be expressed as [31]:

$$\frac{d}{dt} \int_{V_{\text{sphere}}} \mathbf{u} dV = \mathbf{f}_D + \int_{V_{\text{sphere}}} \mathbf{f} dV \quad (3.5)$$

The first term of the RHS of Eq. (3.5) represents the instantaneous drag force exerted on the sphere by the fluid, while the second term corresponds to an instantaneous external force that should be exerted on the body to provide its prescribed kinematics. This term can be directly calculated by summing all the IB forces in accordance with:

$$\int_{V_{sphere}} \mathbf{f} dV = \sum_{ijk} f_{ijk} \Delta x \Delta y \Delta z \quad (3.6)$$

Eqs. (3.5) and (3.6) enable computation of the drag force,  $\mathbf{f}_D$ . Under the assumption of rigid-body motion on the sphere interface [28], the LHS term in Eq. (3.5) can be approximated as:

$$\frac{d}{dt} \int_{V_{sphere}} \mathbf{u} dV = \frac{d\mathbf{u}_c}{dt} V_{sphere} \quad (3.7)$$

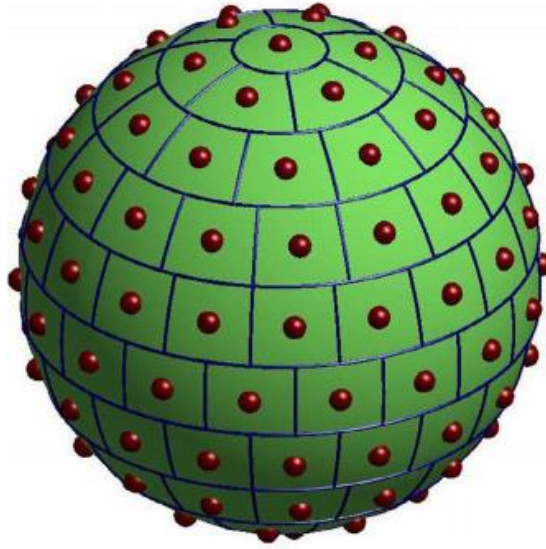
where  $d\mathbf{u}_c/dt$  is the acceleration of the sphere center of mass, which can be obtained analytically by taking the time derivative of Eq. (3.3). For purposes of comparison, the drag force,  $\mathbf{f}_D$ , can also be expressed in terms of the drag coefficient determined for a spherical geometry as:

$$C_{D_i} = \frac{8f_{D_i}}{\rho U_{max}^2 \pi D^2} \quad (3.8)$$

where  $f_{D_i}$  is the value of the  $i^{th}$  component of the dimensional drag force  $\mathbf{f}_D$ . Utilizing the scaling of the present study, the drag coefficient is expressed in terms of the non-dimensional drag force  $\mathbf{F}_D$  as  $C_{D_i} = 8F_{d_i}/\pi$ . In compliance with the IB method, the surface of the sphere is described by a set of uniformly distributed Lagrangian points, obtained by employing the non-iterative method of Leopardi [63]. The number of Lagrangian points used for the discretized sphere is determined by:

$$N_L = \frac{\pi D^2}{\Delta x^2} \quad (3.9)$$

where  $\Delta x$  is the typical cell width of the uniform Eulerian grid. Thus, the spacing between the Lagrangian points is kept approximately the same as the Eulerian grid cell width, providing the high accuracy of the IB method. Figure 3 shows the distributed points over the sphere surface, enclosed in virtual surfaces of equal area.

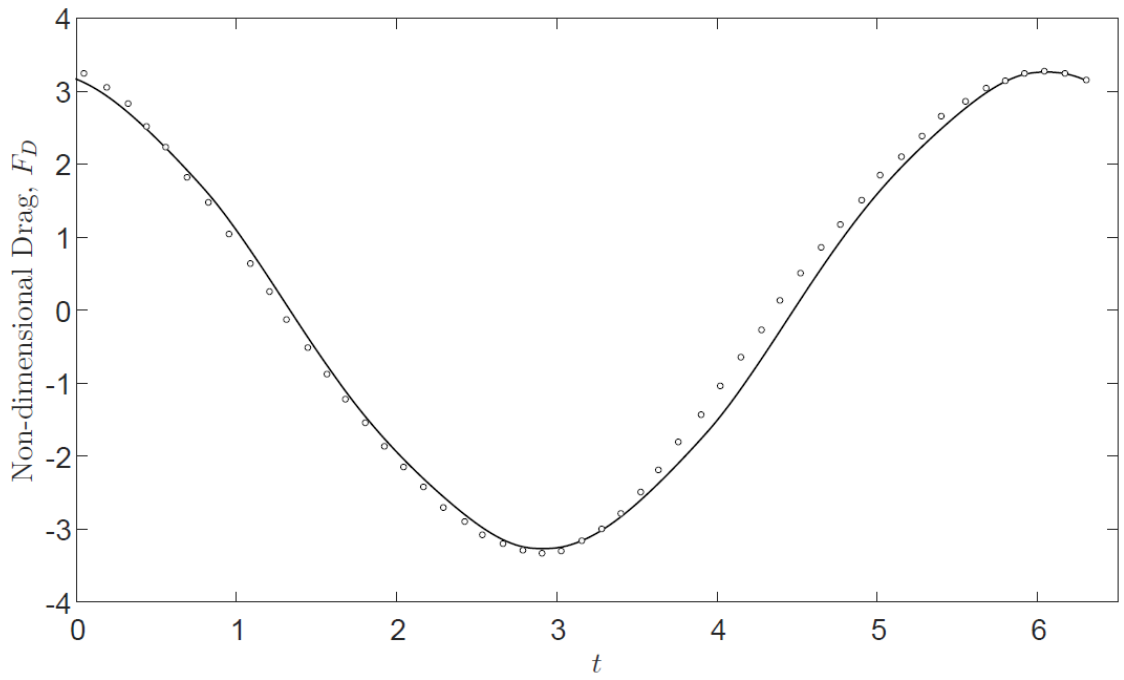


*Figure 6: Lagrangian points evenly distributed over the surface of a sphere by the non-iterative method of Leopardi.*

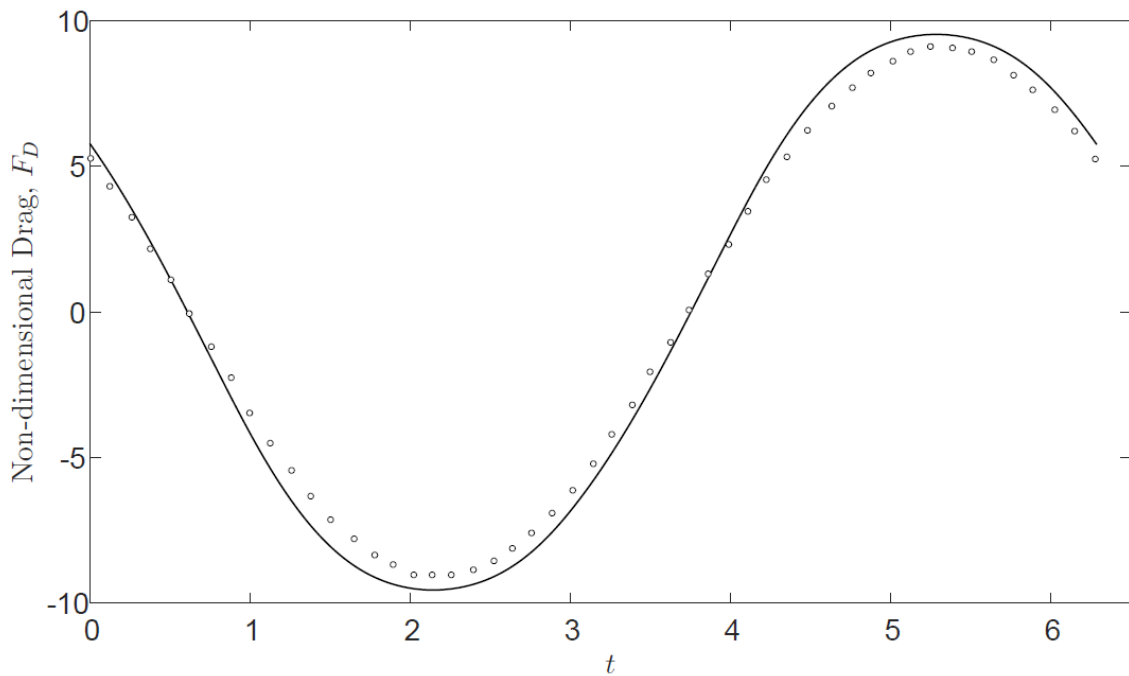
### **3.2.1 Verification study**

The developed methodology was verified by comparing the results obtained for the transversely oscillating sphere with the corresponding results available in the literature. First, a comparison is conducted between the currently calculated time evolutions of the non-dimensional drag force in the  $z$  direction and the corresponding values previously reported in [64]. The comparison is done for two sets of  $Re$  and  $A/D$  values:  $(Re, A/D) = (40, 5)$  and  $(Re, A/D) = (40, 0.3125)$ , as shown in Figure 7. We note in

passing that the non-dimensional time,  $t$ , and the drag force,  $F_D$ , were multiplied by the factors of  $A/D$  and  $3\pi/Re$ , respectively, to meet the scaling used in [64].



(a)



(b)

*Figure 7: Comparison between the time evolutions of non-dimensional drag forces,  $F_D$ , obtained for: (a)  $(Re, A/D) = (40, 5)$ ; (b)  $(Re, A/D) = (40, 0.3125)$ . Solid lines correspond to the currently obtained results, while  $\circ$  markers correspond to the values that were digitally scanned from [64].*

The calculations were performed on a  $200 \times 200 \times 300$  uniform grid with a time step  $\Delta t$  equal to  $10^{-3}$  of the oscillation period. An acceptable agreement between the present and the previously reported  $F_D$  values is observed for the entire length of the oscillation period, while the existing, insignificant, deviation between the results can be attributed to the assumptions of an infinite computational domain and an axisymmetric flow regime made in [64].

As an additional means of verification of the developed methodology, a comparison was made between the values of the peak drag coefficient,  $C_{D_{\max}}$ , with the corresponding values reported in [65], as shown in Figure 8. The  $C_{D_{\max}}$  values were calculated based on the drag force in the  $z$  direction exerted on the vertically oscillating sphere by the flow. Characteristics of the computational set-up, in terms of grid resolution and time step value, were the same as those used in the previous verification test.

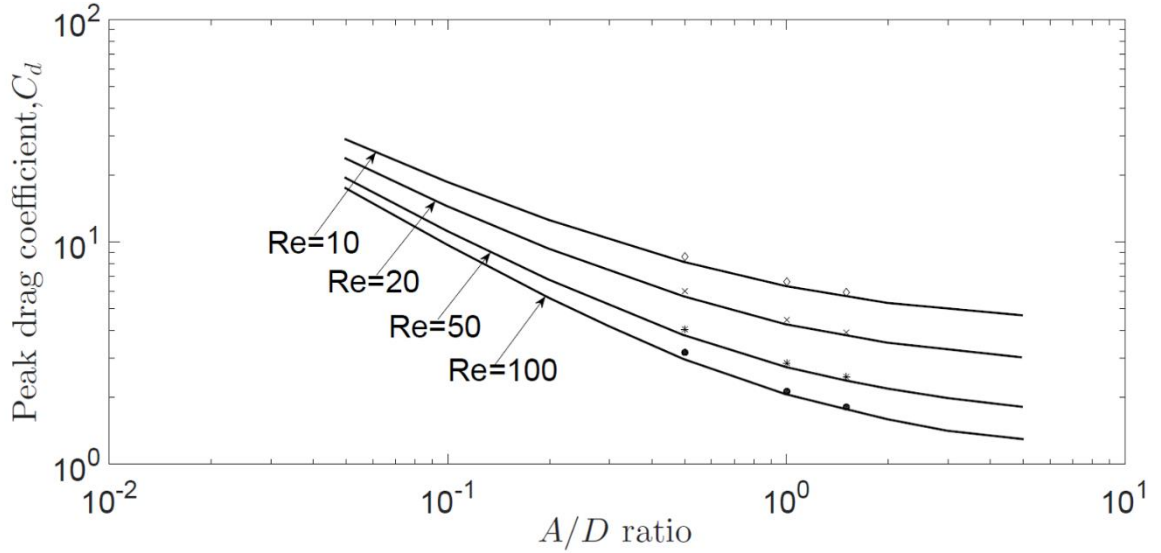


Figure 8: Comparison of the peak drag coefficient,  $C_{D_{\max}}$ , obtained for  $Re = 10$  ( $\diamond$ ),  $Re = 20$  ( $\times$ ),  $Re = 50$  ( $*$ ) and  $Re = 100$  ( $\bullet$ ) as a function of the  $A/D$  ratio with the corresponding  $C_{D_{\max}}$  values reported in [65].

The current results successfully reproduce two general trends reported in [65]: first, for a given value of the  $A/D$  ratio, the  $C_{D_{\max}}$  values decrease with increasing values of Reynolds number; and second, for a given value of  $Re$ , the  $C_{D_{\max}}$  values decrease with increasing values of the  $A/D$  ratio. The first trend can be attributed to the fact that the value of the drag coefficient is inversely proportional to  $Re$ , while the second trend is related to the inertia effects of the flow, which are inversely correlated to the  $A/D$  ratio (the fact becomes obvious after taking the time derivative of Eq. (3.3)). Our results also predict the asymptotic value of the peak drag coefficient at  $A/D \rightarrow \infty$ , representing the steady state configuration. It can be clearly seen that in this case the drag coefficient is significantly lower than its unsteady counterpart. This observation can be explained by an impact of the acceleration of the body inducing a pressure force which contributes to the drag coefficient.

It can be seen that there is an acceptable level of agreement between the present and the previously reported  $C_{D_{\max}}$  values over the entire range of the  $Re$  and  $A/D$  values. As



a general trend, the discrepancy between the results increases with increasing  $Re$  and decreasing  $A/D$  ratio values. The maximal discrepancy (not exceeding 7%) was observed for  $(Re, A/D) = (100, 0.5)$ . The discrepancies between the results of the current study and those reported in [65] can be explained by differences in the size of the computational domains and in the basic assumptions made when performing the numerical simulations. Particularly, the results of [65] were obtained under the assumption of axisymmetric flow in a computational domain extending 50 diameters in both radial and axial directions, while the present simulations were performed for a much smaller ( $4D \times 4D \times 6D$ ) non axisymmetric domain. The difference in the computational domain size can also serve as a possible explanation for the categorically higher values of peak drag coefficient obtained in the present study, as the impact on the sphere is increased by the no-slip boundaries of the confining prism. Another possible reason for the observed discrepancies could be the assumption of rigid body motion of the fluid within the sphere used in the present study for calculating the instantaneous drag force,  $F_D$  (according to the IB formalism the body is represented by a series of discrete points representing its surface, while the confined volume is occupied by a fluid of the same density as the surrounding fluid). This assumption is a good approximation as long as the flow inertia is low, i.e., for high values of the  $A/D$  ratio. However, it may not hold very well for low  $A/D$  values, a fact that is also consistent with the slightly higher value of the presently obtained peak drag compared to the corresponding value of [64] obtained for  $(Re, A/D) = (40, 0.3125)$  (see Figure 6-b). The observed discrepancy, however, shows up in the post-processing stage, and has no impact on the accuracy of the characteristics of the calculated flow field.

### 3.2.2 Grid and time step convergence

The sensitivity of the developed methodology to grid size and time step values was investigated. For this purpose, the convergence of the values of the peak drag coefficients obtained on coarse ( $100 \times 100 \times 150$ ) and fine ( $200 \times 200 \times 300$ ) grids was studied. The results obtained on both grids were further extrapolated to their zero-grid-size asymptotic values,  $C_{D_{\max},h=0}$ , by employing Richardson extrapolation in the following manner:

$$C_{D_{\max},h=0} = C_{D_{\max},2} + \frac{(C_{D_{\max},1} - C_{D_{\max},2})}{\tilde{r}^{\tilde{p}} - 1} + O(h^{\tilde{p}+1}) \quad (3.10)$$

where  $C_{D_{\max},1}$  and  $C_{D_{\max},2}$  are the values obtained on the fine and coarse grids, respectively,  $\tilde{r}$  is the grid refinement ratio (in this case  $\tilde{r} = 2$ ) and  $\tilde{p}$  is the order of the used method (in this case  $\tilde{p} = 2$ ). Theoretically, equation (3.10) for the Richardson extrapolation provides a  $\tilde{p}+1$  order estimate for the continuum value of  $C_{D_{\max}}$ . Therefore, considering that calculations are done with a second order method employed with grid spacing of  $h = 0.02$  (as in the present study), this estimation is the equivalent of performing a grid refinement with a grid refinement ratio of  $\tilde{r} \approx 7$ . The calculated fine and coarse grid values, along with the extrapolated values, were compared with the numerical results reported in [65], as shown in Table 1. It can be seen that the  $C_{D_{\max}}$  values monotonically decrease with mesh refinement, while the deviation between the fine grid and the zero-grid-size asymptotic values does not exceed 3%.

Table 1: Comparison between the present and the previously published  $C_{D\max}$  values

Re = 10				
$A/D$	$C_{D\max,h=0.04}$	$C_{D\max,h=0.02}$	$C_{D\max,h\approx 0}$	[65]
0.5	8.88	8.59	8.49	8.14
1	6.77	6.62	6.57	6.32
1.5	6.02	5.92	5.89	5.8
Re = 20				
$A/D$	$C_{D\max,h=0.04}$	$C_{D\max,h=0.02}$	$C_{D\max,h\approx 0}$	[65]
0.5	6.26	6.01	5.93	5.7
1	4.58	4.47	4.43	4.27
1.5	3.98	3.93	3.91	3.83
Re = 50				
$A/D$	$C_{D\max,h=0.04}$	$C_{D\max,h=0.02}$	$C_{D\max,h\approx 0}$	[65]
0.5	4.27	4.04	3.96	3.82
1	2.93	2.85	2.82	2.74
1.5	2.51	2.47	2.46	2.39
Re = 100				
$A/D$	$C_{D\max,h=0.04}$	$C_{D\max,h=0.02}$	$C_{D\max,h\approx 0}$	[65]
0.5	3.44	3.18	3.09	2.97
1	2.33	2.13	2.06	2.06
1.5	1.89	1.82	1.8	1.77

An additional important observation is related to the time evolution of the  $C_D$  obtained on different grids. Spurious high-frequency oscillations (in the form of saw teeth) were observed for the time evolution of  $C_D$  calculated on the coarse grid. These oscillations were

smoothed completely with mesh refinement. For this reason, all further results for the simulation of the present flow configuration were obtained using a  $200 \times 200 \times 300$  grid. Investigation of the time step value sensitivity of the obtained results revealed that sustainable numerical stability of the developed method is achieved for the Courant number values  $C \leq 0.2$ . Time step independence of the obtained results (including the elimination of the spurious saw teeth oscillations) is achieved for the values of  $\Delta X \leq 2 \times 10^{-2}$  and  $\Delta t \leq 10^{-3}$  for the values of  $Re \leq 200$ .

### 3.2.3 Efficiency characteristics

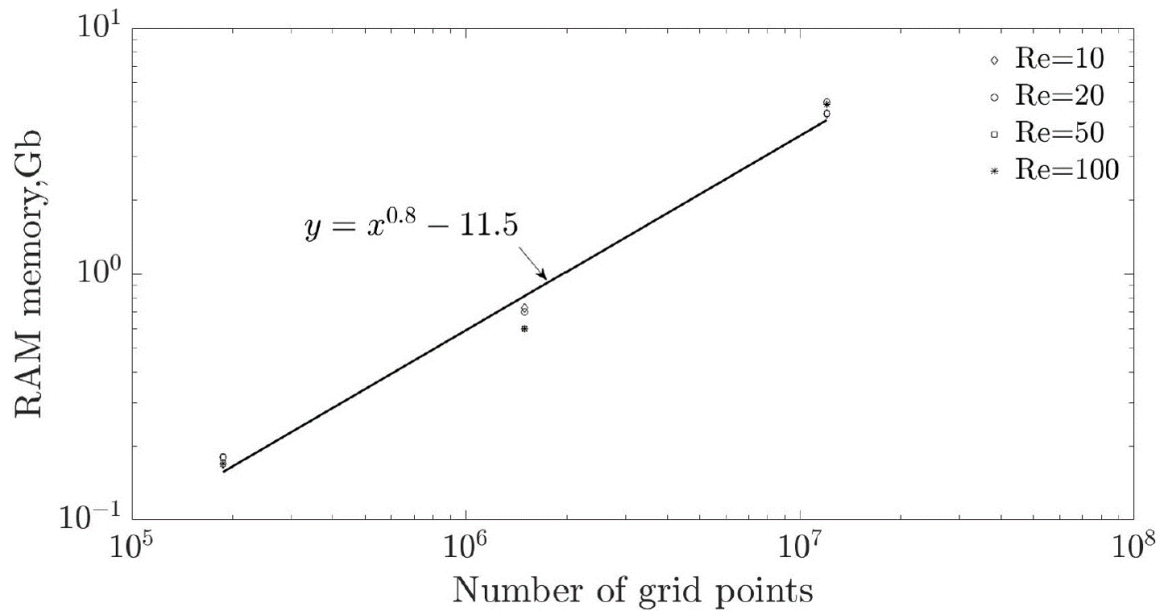
Focus has been put on the efficiency characteristics of the developed methodology in terms of memory consumption and time step duration, for both the pre-computation and the time integration stages. The trends for both the efficiency characteristics were obtained by utilizing a power law best fit to the corresponding measurements of these characteristics. The measurements were taken for three different grid resolutions and four different values of the Reynolds number, as shown in Figure 9. It is remarkable that the pre-computing stage is characterized by higher absolute values of consumed RAM compared to the time integration stage. This behavior can be explained by the extra memory consumed by the MUMPS solver when performing the  $LU$  factorization procedure. Once the factorization stage is completed, only the  $LU$  factors are stored on the hard disk<sup>5</sup>, while all the dynamically allocated auxiliary memory is automatically released. It is also worth mentioning that the configurations characterized by a lower Reynolds number typically consume more memory, which can be explained by the more pronounced elliptical character of the low-Reynolds-number-flows. This feature ultimately leads to a higher

---

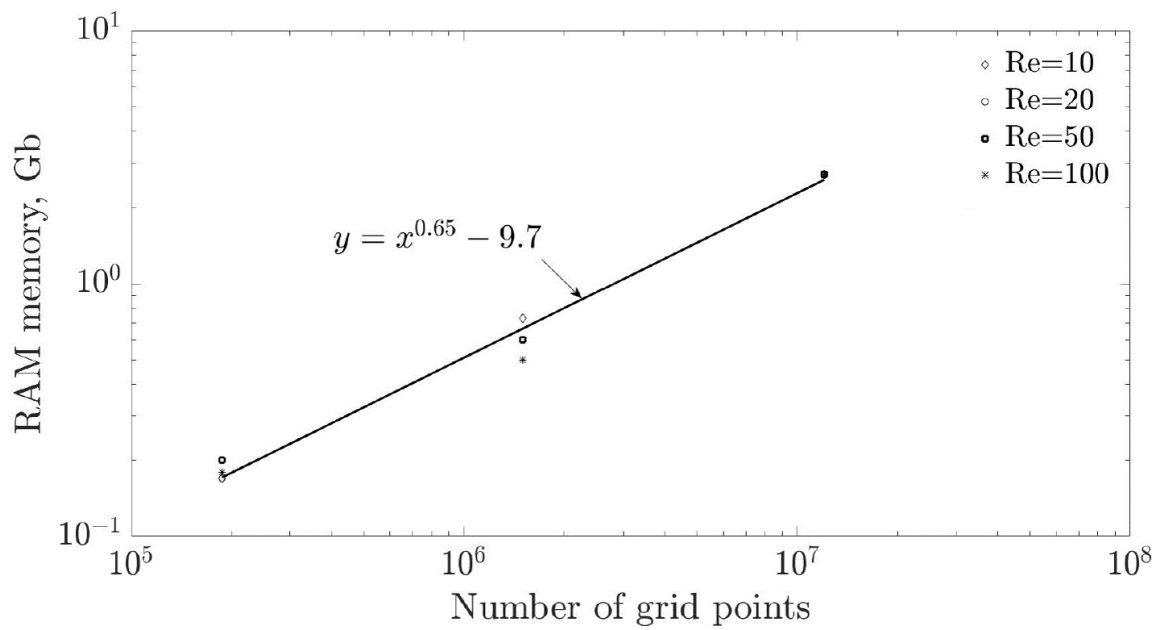
<sup>5</sup> In accordance with the developed algorithm, the stored factors will be cyclically read during the time integration stage.

number of non-zero entries in the  $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$  matrix satisfying the determined sparsing threshold  $s = 10^{-21}$ . Another important observation is related to the fact that exponent values of the memory consumption fits, built for both pre-computing and time integration stages, are significantly lower than unity, indicating the efficiency in exploiting the sparseness of the matrices involved in both stages.

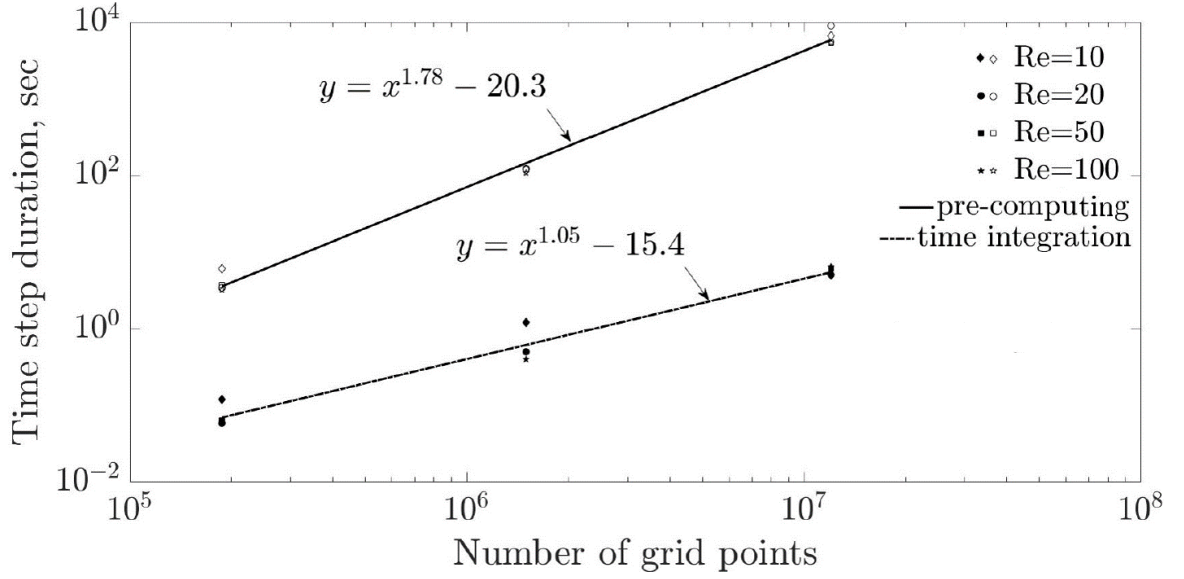
The efficiency of the developed methodology in terms of time step duration in both pre-computing and time integration stages can be assessed by examining the corresponding best fits shown in Figure 9. It can be seen that the exponent value corresponding to the best fit built for the time integration stage is very close to unity, indicating high efficiency of the developed methodology, whose time consumption grows almost linearly with increasing grid resolution. In the same instance, a much higher slope characterizes the best fit corresponding to the time consumption of the pre-computing stage. This observation can be explained by the fact that an increasing grid resolution has, in practice, a dual effect on the amount of calculations that have to be performed in the pre-computing stage. First, it increases the dimensions of the matrix  $[\mathbf{H}]$  corresponding to the Helmholtz operator of the original solver, not equipped with the IB functionality; and second, it increases the number of Lagrangian points (as follows from Eq. (3.9)) needed to meet the requirement of approximately the same distance between the neighboring Lagrangian points and the cell width of the underlying Eulerian grid, necessary to achieve high accuracy of the obtained results.



(a)



(b)



(c)

Figure 9: Efficiency characteristics: (a) Memory consumption in pre-computing stage; (b) Memory consumption in time integration stage; (c) Time step duration in pre-computing and time integration stages.

### 3.2.4 Flow simulations

Upon successful verification of the developed method for the simulation of flow developing in the presence of a transversely oscillating sphere, a number of representative results was obtained to gain a better understanding of the 3D flow configuration characteristics. Figure 10 presents the time evolution of the drag coefficient,  $C_D$ , obtained for the values of  $Re = 50, 100, 150$  and  $200$  and  $A/D = 1$ . The trend which was already observed in Figure 8, i.e., an inverse growth of the peak drag coefficient with the Reynolds number, is also preserved for the higher values of Reynolds number. Another important remark is that the inertia effects of the flow are quite significant. These effects are manifested in a clearly visible phase lag between the time evolution of the position of the sphere,  $z - z_0$ , and the  $C_D$  curves, which indicates that the fluid, driven by the oscillating sphere, continues its motion even after the sphere has completely stopped moving.

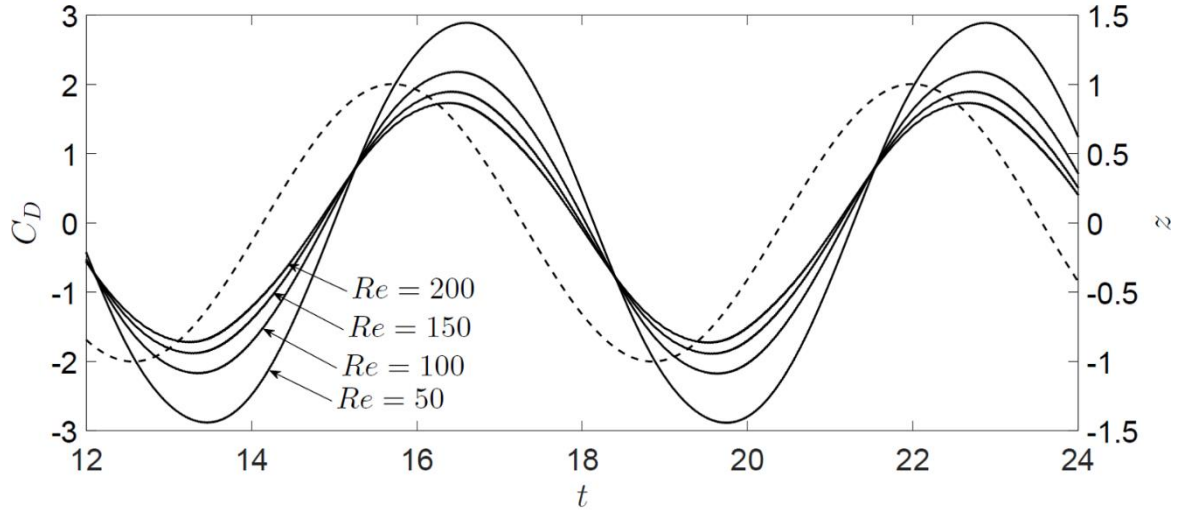
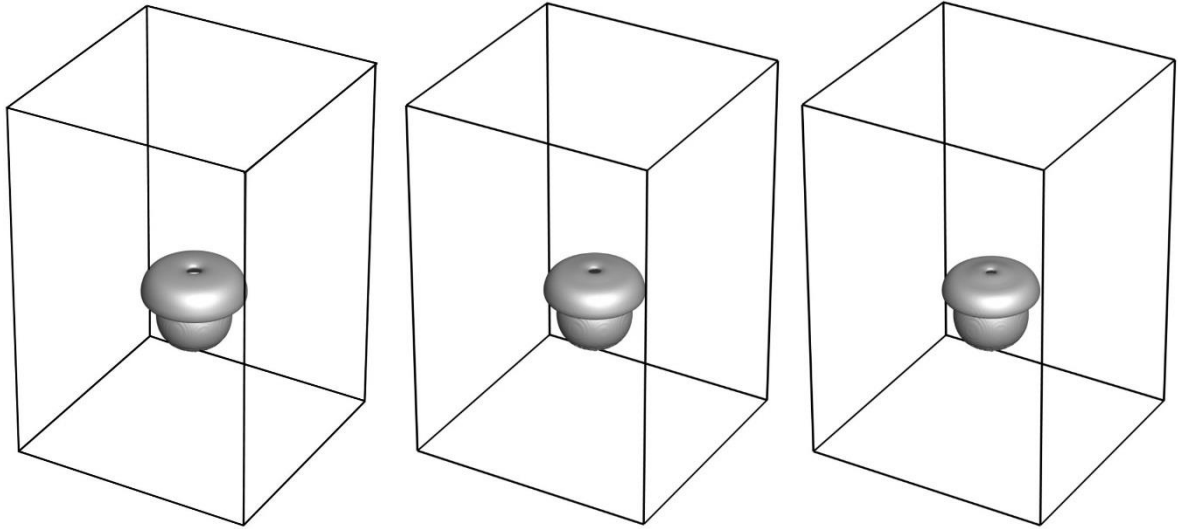


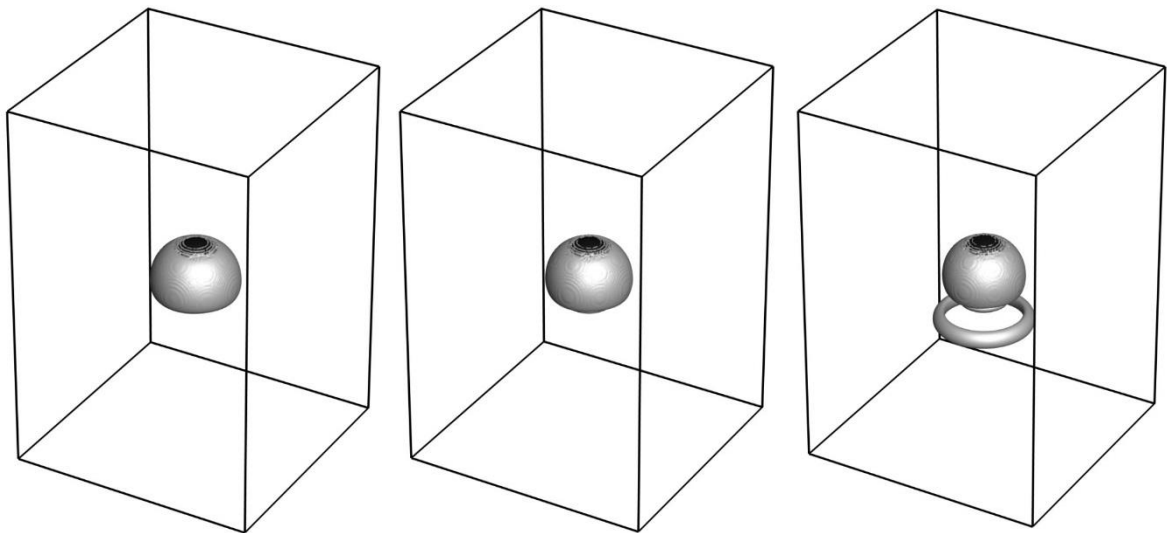
Figure 10: Time evolution of the drag coefficient,  $C_D$ , obtained for  $Re = 50, 100, 150, 200$  and  $A/D = 1$ , superimposed on the time evolution of the position of the sphere,  $z - z_0$  (dashed line).  $t$  is the non dimensional time.

Flow patterns developing on different parts of the sphere trajectory have been visualized by looking at the isosurfaces of the  $\lambda_2$  criterion corresponding to the value of  $\lambda_2 = -0.1$ , as shown in Figure 11. According to [66], the isosurfaces characterize the vortical structures of the flow. Because of the flow inertia, the annular vortical structures are formed at the lowest and the highest points of the sphere trajectory, shedding from the sphere surface and then propagating through the computational domain. It is also remarkable that the lifetime of the annular vortices increases with increasing values of the Reynolds number. In fact, for  $Re = 100$ , the vortical structure dissipates almost immediately after separating from the sphere surface, while for  $Re = 200$  the structure continues to evolve within about half a period of the sphere oscillation.

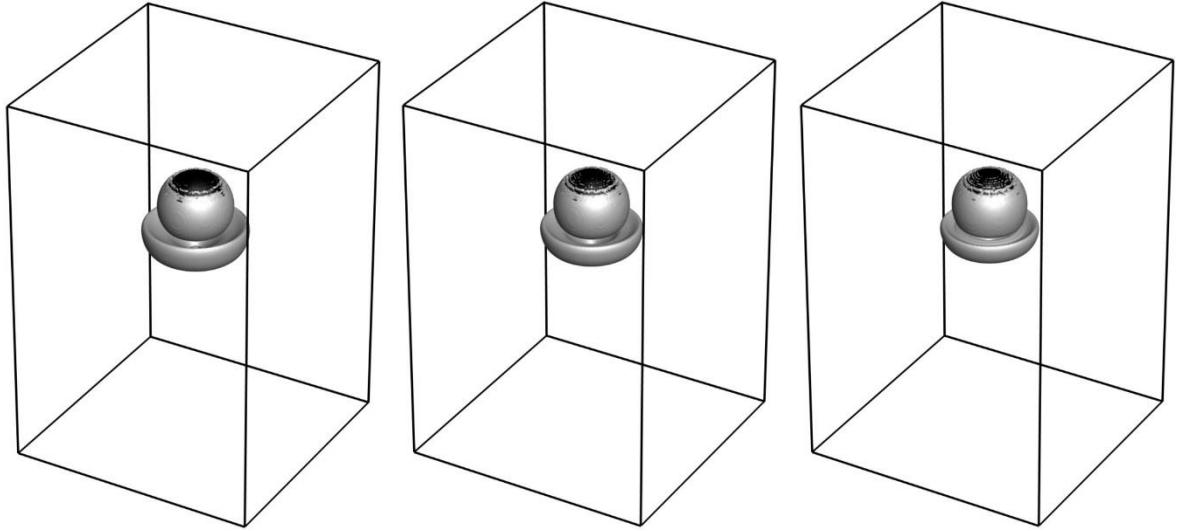




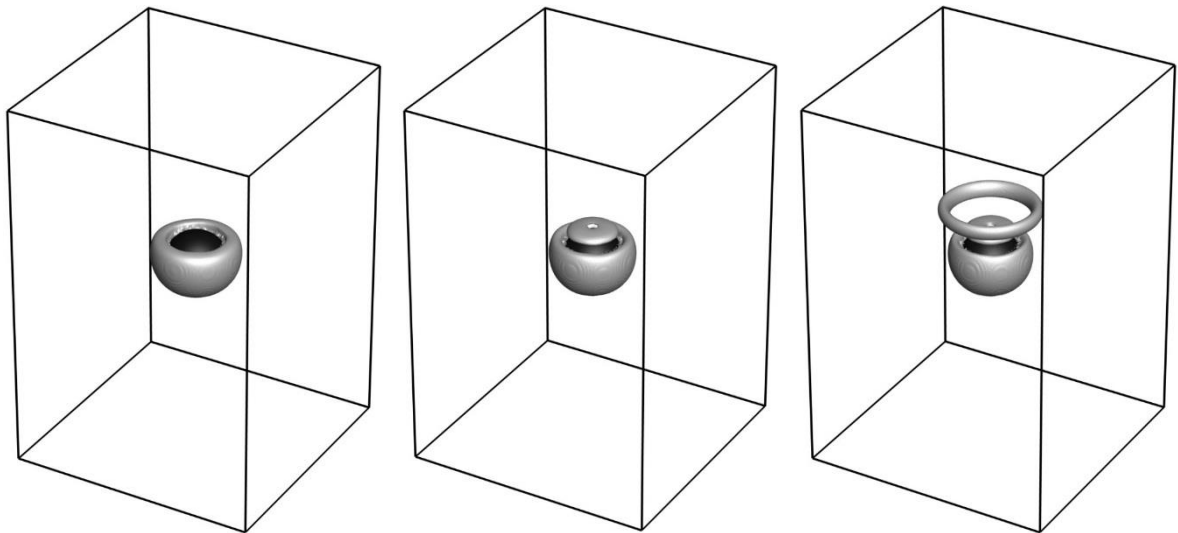
(a) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the lowest point of the sphere trajectory for  $Re = 100, 150, 200$  (from left to right).



(b) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the mid-point of the sphere trajectory on its way up for  $Re = 100, 150, 200$  (from left to right).



(c) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the highest point of the sphere trajectory for  $Re = 100, 150, 200$  (from left to right).



(d) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the mid-point of the sphere trajectory on its way down for  $Re = 100, 150, 200$  (from left to right).

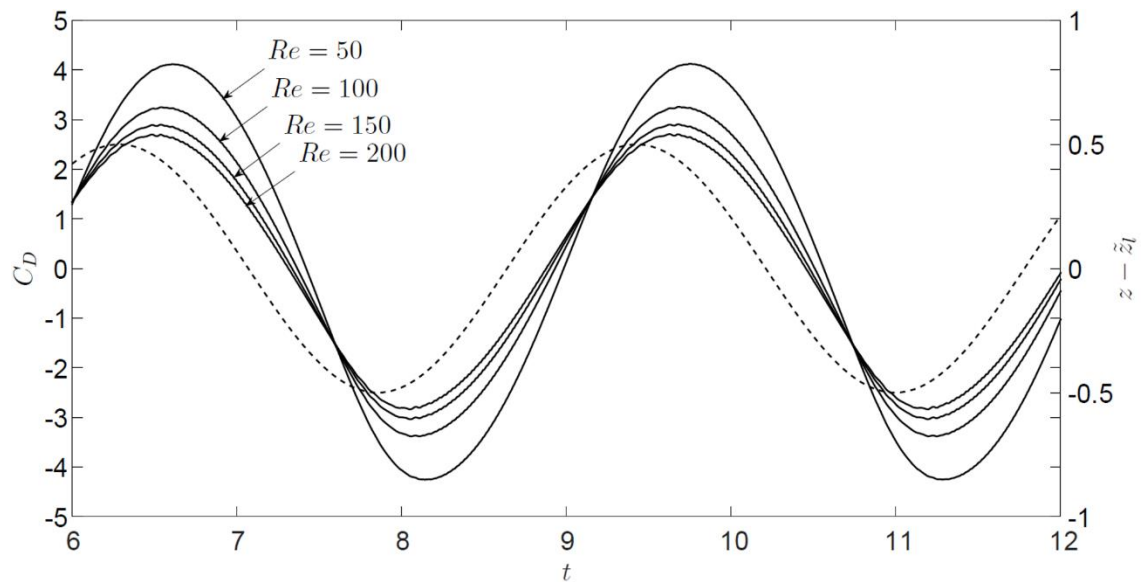
*Figure 11: Visualization of the vortical structures generated by a transversely oscillating sphere over a single oscillating period calculated for  $Re = 100, 150, 200$  and  $A/D = 1$ .*

It should be noted here that attempts to simulate the flow developing at even higher values of Reynolds numbers,  $Re \geq 300$ , have been made (for these values the symmetry of the flow is expected to break down, justifying the full 3D simulation for capturing non-axisymmetric flow phenomena). Unfortunately, when the simulations are performed on a

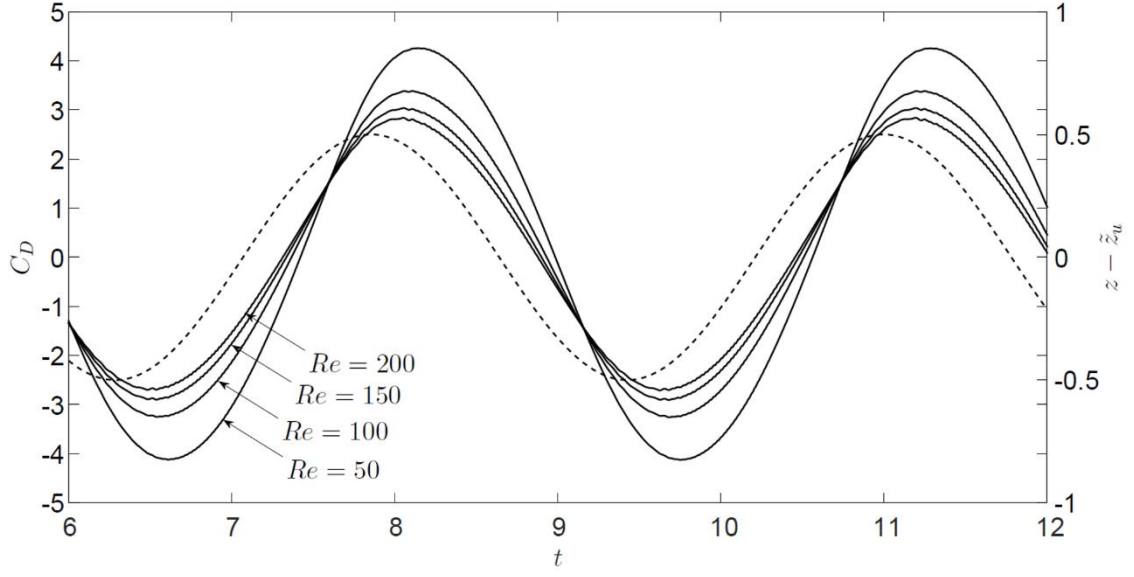
$200 \times 200 \times 300$  grid the time evolution of  $C_D$  showed reoccurrence of the spurious high-frequency oscillations reported in 3.2.2. This undesirable artifact of unphysical temporal oscillations of the pressure field has been previously reported for IBMs dealing with moving bodies on a fixed, non-body conformal computational grid [67],[68]. According to Lee et al. [67], this artifact manifests itself when using direct forcing IBMs for moving-body problems, as the result of a temporal discontinuity in the velocity, which is related to the fact that the numerical stencils applied for the forcing terms vary with boundary motion. In other words, when the IB approaches an external grid point (of the Eulerian grid), the velocity,  $\mathbf{u}_i$ , at that grid point should approach the body velocity,  $\mathbf{U}_i^T$ ; however, in practice, it converges to the body velocity with a spatial discretization error up to the order of the interpolation error. When this grid point is located on the IB, with the body shifting its position in the next time step, its velocity suddenly changes to satisfy the body kinematic constraint through the momentum forcing. This discontinuity in time ultimately generates non-physical behavior of the pressure near the IB. However, recalling that the magnitude of the velocity discontinuity is proportional to the grid spacing [67], this numerical artifact can be eliminated by performing simulations on even finer grids. This feat is feasible for stationary set-ups, albeit prohibitively expensive for configurations containing oscillatory moving immersed bodies. Future research will be focused on developing a sharp interface formulation of the developed methodology that will make it possible to implicitly impose kinematic no-slip constraints with the second order of accuracy, while keeping intrinsic portability of the methodology, i.e., utilizing the existing efficient time steppers of the NS equations in a black box manner.

### 3.3 Flow around a pair of transversely out-of-phase oscillating spheres

To further demonstrate the capabilities of the developed methodology, the configuration of a pair of out-of-phase oscillating spheres was considered. Similarly to the previous configuration, the flow was calculated for the values of  $Re = 50, 100, 150$  and  $200$ . Both spheres oscillate with a value of  $A/D = 0.5$ , each around its own oscillation center placed at a distance of  $1.25D$  from the top and bottom boundaries of the prism, respectively. As a result, a minimal distance of  $0.5D$  is reached between the spheres as their velocity reaches zero. This set-up was chosen carefully with the aim of using the same prismatic enclosure of dimensions  $4D \times 4D \times 6D$  as that utilized in previous simulations. The time evolution of the drag coefficient  $C_D$  calculated for the upper and lower spheres as a function of the Reynolds number is presented in Figure 12. The time evolution of  $C_D$  is superimposed on the time evolution of each sphere's position (relative to its oscillation center), as indicated by dashed lines.



(a)



(b)

Figure 12: Time evolution of the drag coefficient,  $C_D$ , obtained for  $Re = 50, 100, 150$  and  $200$  (solid line) and  $A/D = 0.5$  for: (a) The lower sphere superimposed on the time evolution of its position,  $z - \tilde{z}_l$  (dashed line), where  $\tilde{z}_l$  is the lower sphere's oscillation center; (b) The upper sphere superimposed on the time evolution of its position,  $z - \tilde{z}_u$  (dashed line), where  $\tilde{z}_u$  is the upper sphere oscillation center.  $t$  is the non-dimensional time.

As in previous configurations, there is a clearly distinguishable phase lag between the time evolutions of  $C_D$  and the corresponding position of the sphere, which again confirms the existence of non-negligible inertia effects of the surrounding flow. By virtue of the symmetric initial boundary conditions, the time evolutions of  $C_D$  of both spheres are symmetric relative to the corresponding oscillation centers for the entire range of Reynolds numbers, up to a slight bias reflected in a difference between the maximal and minimal absolute values of each sphere's  $C_D$ . In particular, the absolute  $C_D$  values are consistently higher when the spheres are close to each other over the entire range of Reynolds values, as follows from the data acquired over a single oscillating period for the upper sphere, without loss of generality, as detailed in Table 2. It is noteworthy that the  $C_D$  time

evolutions acquired for the two spheres are equal up to a multiplication by minus unity i.e., the mutual effect of the spheres is independent of the direction of their motion and is only a function of their acceleration and proximity. As the spheres move away from each other, their mutual effect decreases and eventually tends to zero, resulting in smaller absolute values of the  $C_D$  extrema. This observation is supported by a comparison between the current extremum values of  $C_D$  and the values of peak drag coefficients calculated for a single oscillating sphere at  $(\text{Re}, A/D)=(50,0.5)$  and at  $(\text{Re}, A/D)=(100,0.5)$ , as detailed in Table 1. It can be seen that for the case when the two spheres are at the maximal distance from one another, the extremum values of  $C_D$  are fairly close to the peak drag coefficient values acquired for a single oscillating sphere (1.5% deviation).

*Table 2: Absolute values of the minimum and maximum  $C_D$  acquired for the upper sphere as a function of the Reynolds number*

<b>Reynolds number</b>	<b><math>C_D</math> Maximum absolute value</b>	<b><math>C_D</math> Minimum absolute value</b>
<b>50</b>	4.1	4.25
<b>100</b>	3.24	3.37
<b>150</b>	2.89	3
<b>200</b>	2.69	2.8

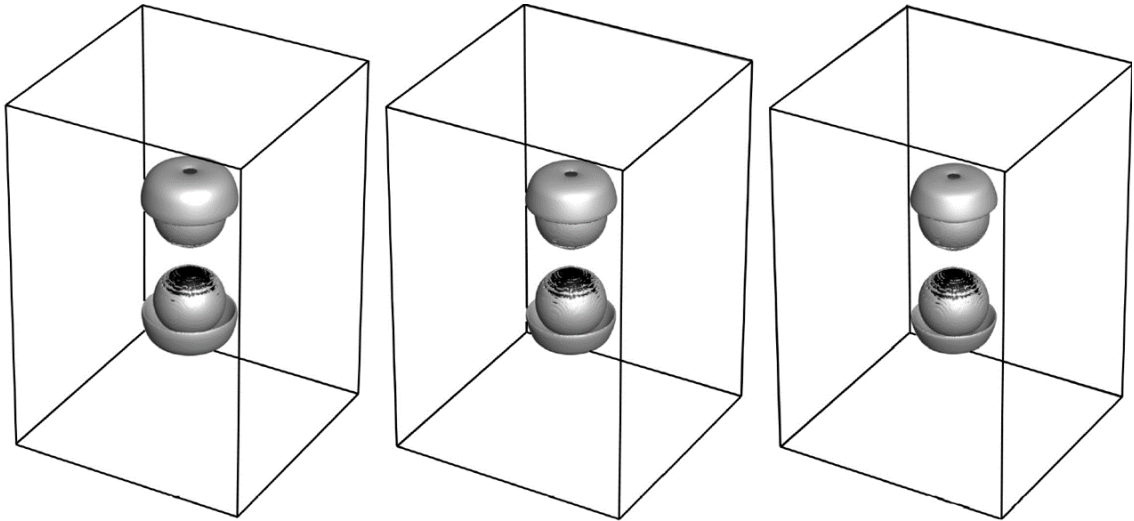
It should be mentioned that, although significantly alleviated, the insignificant non-physical high-frequency oscillations can again be recognized in the time evolution of the  $C_D$  values of the two spheres for  $\text{Re} \geq 100$ . These high-frequency oscillations are a

numerical artifact, reappearing as a result of the fact that the acceleration of the spheres<sup>6</sup> is twice that characterizing the previous configuration. As has already been mentioned, the observed high-frequency oscillations can be eliminated either by utilizing grids that are even denser or by employing a second-order sharp interface formulation of the developed methodology, which will be the focus of our future work.

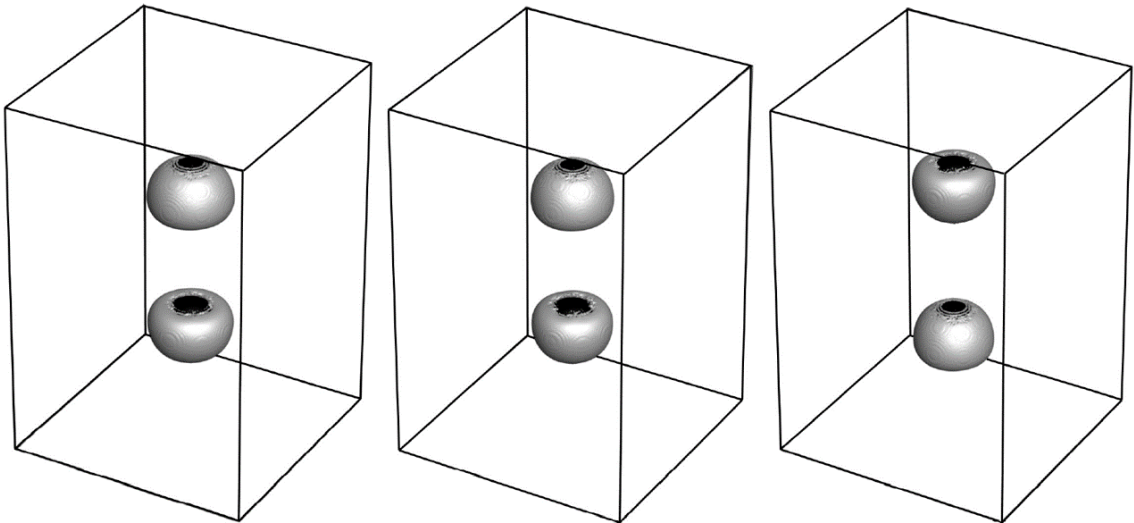
The obtained results were visualized by utilizing the same technique as that used for the previous configuration, i.e., by presenting vortical structures recognized by isosurfaces of the  $\lambda_2$  criterion corresponding the value of  $\lambda_2 = -0.1$ . The isosurfaces are presented for four representative time instances taken over the oscillation period, as shown in Figure 13. Surprisingly, the inertia of the surrounding flow does not have a strong effect on the intensity of the shedding phenomenon. In fact, despite the higher acceleration characterizing the flow regime under consideration, the intensity of the shedding of the annular vortex structures from the sphere surface is significantly less pronounced, as can be seen in Figure 13. This is apparently a consequence of the half length of the trajectory of the spheres, which leads to lower peak values of the local fluid flow rate generated by the spheres.

---

<sup>6</sup> A direct consequence of a double reduction of the  $A/D$  value.

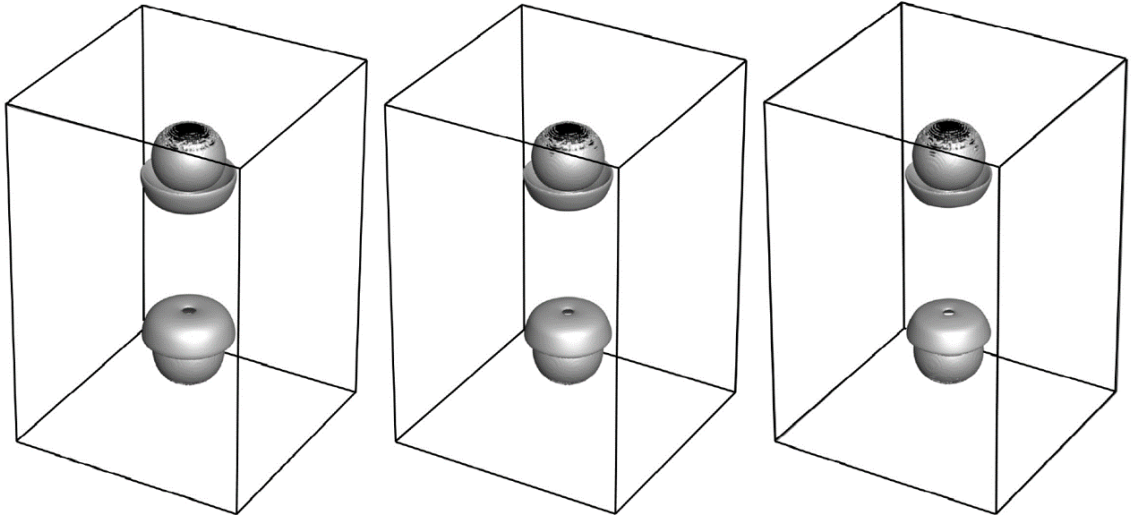


(a) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the lowest point of the sphere trajectory for  $Re = 100, 150, 200$  (from left to right).

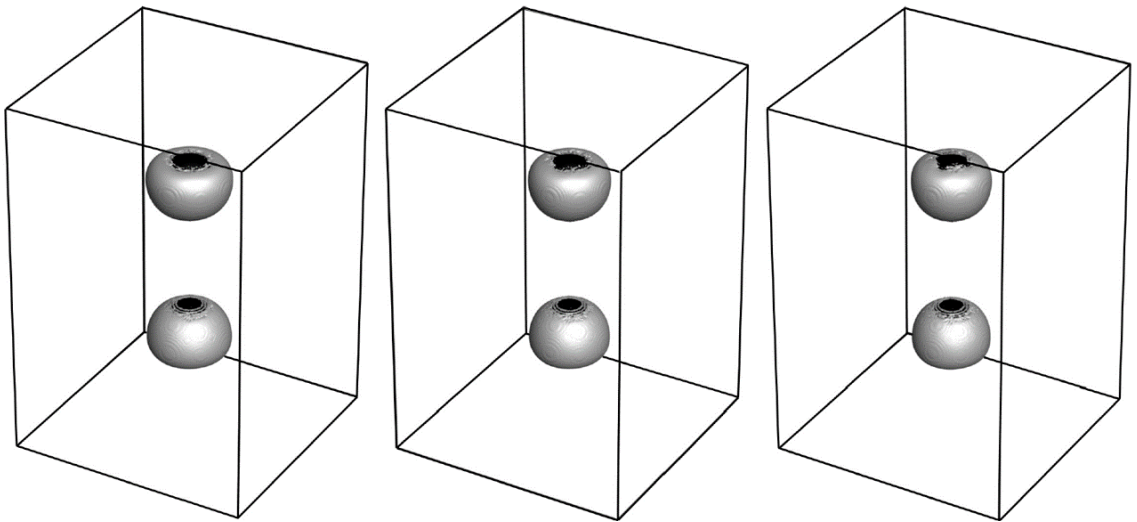


(b) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the mid-point of the sphere trajectory on its way up for  $Re = 100, 150, 200$  (from left to right).





(c) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the highest point of the sphere trajectory for  $Re = 100, 150, 200$  (from left to right).



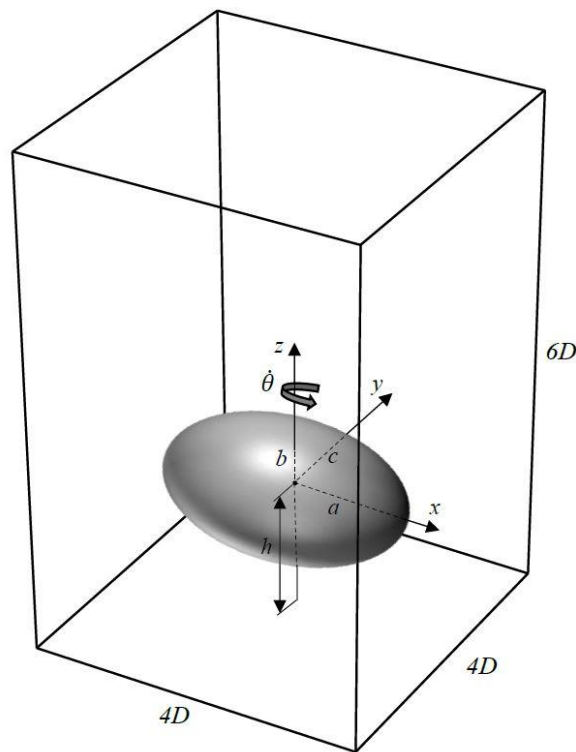
(d) Isosurfaces of the  $\lambda_2 = -0.1$  criterion calculated at the mid-point of the sphere trajectory on its way down for  $Re = 100, 150, 200$  (from left to right).

*Figure 13: Typical pattern of vortical structures characterized by the isosurfaces of  $\lambda_2 = -0.1$ , generated by a pair of out-of-phase transversely oscillating spheres over a single oscillation period calculated for  $Re = 100, 150$  and  $200$  and for  $A/D = 0.5$ .*

### 3.4 Flow around a rotating ellipsoid

In this section the flow generated by an ellipsoid rotating with uniform angular velocity,  $\dot{\theta}$ , is considered. This kind of flow configuration is of great importance in

simulating the active mixing phenomenon that is an integral part of many processes in bio- and chemical engineering. The physical model of the problem is shown in Figure 14. The ellipsoid is placed within a prismatic computational domain of dimensions  $4D \times 4D \times 6D$ . The center of the ellipsoid is aligned with the vertical centerline of the prism and is placed at a distance  $h$  from its bottom. The direction of the ellipsoid's rotation around the  $z$  axis is as indicated in Figure 14 by a curved arrow. The ellipsoid radii are  $a$ ,  $c$  and  $b$  along the  $x$ ,  $y$  and  $z$  axes, respectively. No-slip boundary conditions are applied to the surface of the ellipsoid and to all the walls, confining the computational domain.



*Figure 14: A schematic representation of the physical model. The ellipsoid is confined by a prismatic enclosure of dimensions  $4D \times 4D \times 6D$  and rotates with a constant angular velocity  $\dot{\theta}$  around the  $z$  axis.*

Similarly to for the previous configuration, the surface of the ellipsoid is discretized by a set of uniformly distributed Lagrangian points. An in-house developed algorithm, using a set of equispaced circular sections to describe the ellipsoid, is used to implement the

discretization procedure. For the special case of  $b = c$ , the number of circular sections used to describe the ellipsoid is determined by:

$$N_s = \frac{2a}{\Delta x} \quad (3.11)$$

where  $\Delta x$  is the typical cell width of the uniform Eulerian grid, and the number of Lagrangian points used for the discretization of the surface of the ellipsoid is given by the sum of the equispaced Lagrangian points at each circular section:

$$N_L = \sum_{i=-\frac{N_s}{2}}^{\frac{N_s}{2}} \frac{2\pi \left( b \sqrt{1 - \left( \frac{i\Delta x}{a} \right)^2} \right)}{\Delta x} \quad (3.12)$$

Simulation results are calculated on a  $100 \times 100 \times 150$  uniform grid with a time step equal to  $\Delta t = 10^{-3}$  of the rotation period. Utilizing the values of  $2b$ ,  $U_{\max} = a\dot{\theta}$  and  $2b/U_{\max}$  for normalizing length, velocity and time scales, respectively, the flow is governed by two non-dimensional parameters, namely, the Reynolds number ( $\text{Re} = U_{\max} 2b/\nu$ ) and the ratio of the longest to the shortest radius,  $N = a/b$ . The  $(x, y, z)$  position of any point on the rotating ellipsoid's surface is governed by the following non-dimensional equations:

$$x = N/2 \cos(\phi) \cos(\theta_0 + 2t/N) + X_0/2b \quad (3.13)$$

$$y = 1/2 \cos(\phi) \sin(\theta_0 + 2t/N) + Y_0/2b \quad (3.14)$$

$$z = 1/2 \sin(\phi) + Z_0/2b \quad (3.15)$$

where  $-\pi/2 \leq \phi \leq \pi/2$ ,  $-\pi \leq \theta \leq \pi$  and  $(X_0, Y_0, Z_0)$  are the coordinates of the ellipsoid's center relative to the global coordinate system origin. The ellipsoid's instantaneous velocity vector  $(u_x, u_y, u_z)$  can be obtained by an analytical time differentiation of the corresponding coordinate, yielding:

$$u_x = -\cos(\phi)\sin(\theta_0 + 2t/N) \quad (3.16)$$

$$u_y = 1/N \cos(\phi)\cos(\theta_0 + 2t/N) \quad (3.17)$$

$$u_z = 0 \quad (3.18)$$

Assuming rigid-body motion on the ellipsoid interface [28], and taking into account that the ellipsoid rotated with a constant angular velocity  $\dot{\theta}$ , the torque exerted on the ellipsoid by the flow as a result of the drag force may be estimated by:

$$M = -\int_{V_{\text{ellipsoid}}} \mathbf{r} \times \mathbf{f} dV = -\sum_{i,j,k} [\mathbf{r} \times \mathbf{f}]_{i,j,k} \Delta x \Delta y \Delta z \quad (3.19)$$

The simulations were performed for  $h=2b$ ,  $N=2$  and  $\text{Re}=100, 200, 300$ . The time evolution of the torque,  $M$ , exerted on the rotating ellipsoid by the flow, as a function of the whole range of Reynolds number values is shown in Figure 15. It can be seen that the previously reported numerical artifact of spurious high-frequency oscillations is present in all the results. Unfortunately, subsequent attempts to perform these simulations on a denser  $200 \times 200 \times 300$  grid have failed to achieve a significant mitigation of these oscillations. As previously stated, a remedy to this problem must come from a higher order of IB formulation accuracy, which will be the focus of future research. At the same time, the obtained results still enable studying the qualitative attributes of the flow when focusing on the low frequency harmonics of the time evolutions. As expected, the absolute value of  $M$

decreases with decreasing Reynolds number. In each curve, the four highest peaks appearing during a single turnover time are attributed to the prismatic shape of the computational domain. The peaks are related to four symmetric positions of the ellipsoid when its longest axis is perpendicular to the face of the domain, resulting in an abrupt deviation of the  $M$  value as the distance between the two tips of the ellipsoid and the corresponding faces of the prism changes. The above observations are made even more substantial by examining the distribution of the vortical structures visualized in Figure 16, as shown by the isosurfaces of  $\lambda_2 = -0.1$ . It can be seen that the pairs of patterns observed in the figures: Figure 16-a and Figure 16-c, and Figure 16-b and Figure 16-d, are symmetric up to  $90^\circ$  rotation, which explains the quarter turn periodicity of the time evolution of  $M$ . It can be seen that the patterns are also somewhat reminiscent of Görtler vorticities, which occur at boundary layer flows along concave walls, although the detailed examination of this hypothesis remained beyond the scope of the current study.

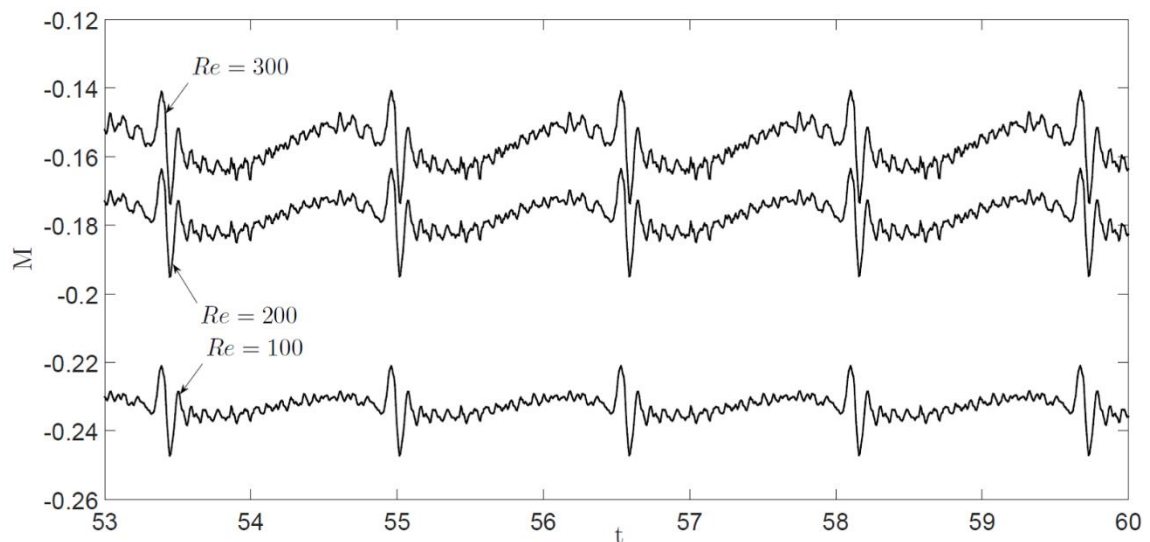


Figure 15: Time evolution of the torque,  $M$ , exerted by the flow on the rotating ellipsoid, obtained for  $Re = 100, 200, 300$ ,  $N = 2$  and  $h = 2b$ .

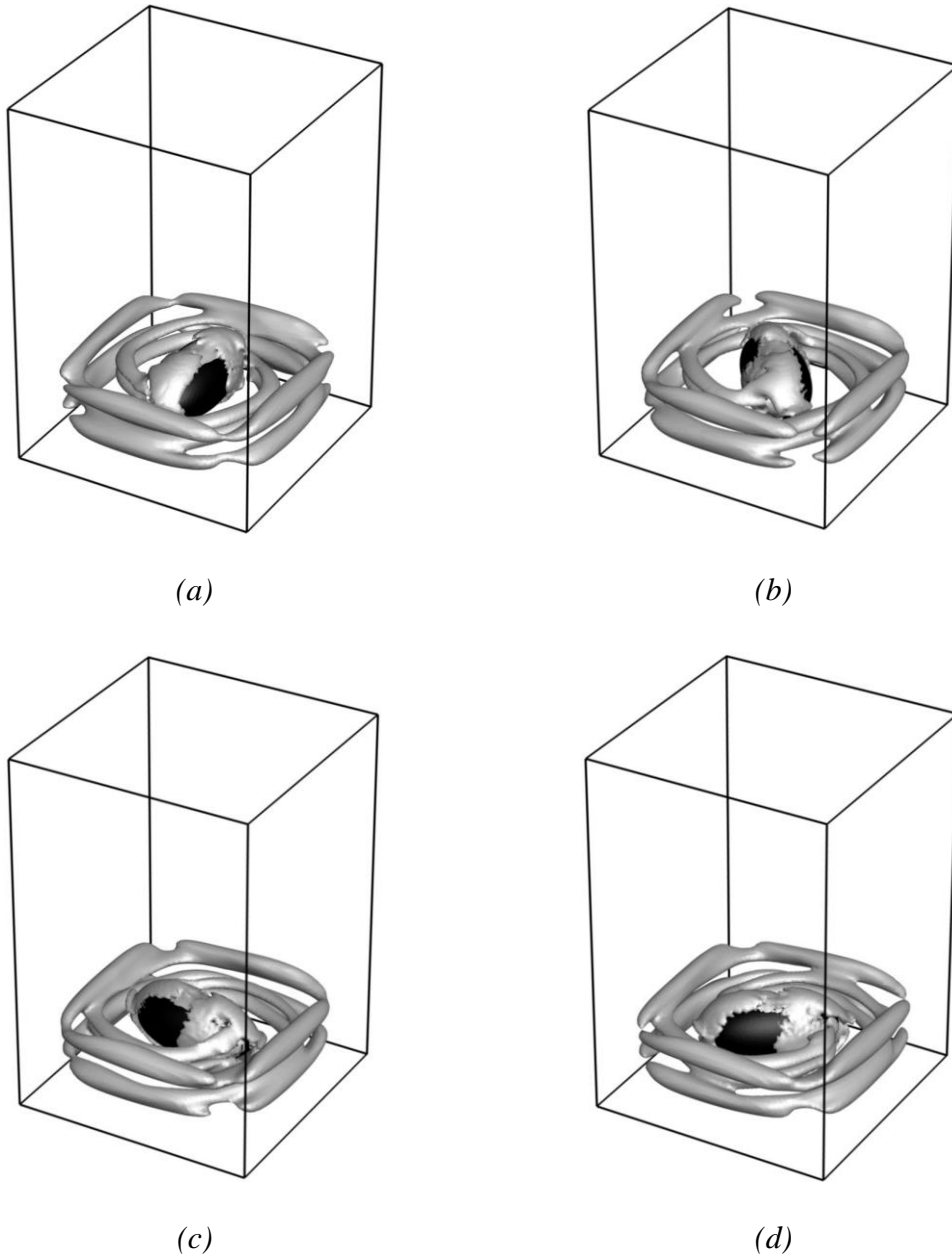


Figure 16: Typical patterns of vortical structures characterized by the isosurfaces of  $\lambda_2 = -0.1$ , generated by the rotating ellipsoid for  $\text{Re} = 100, 200, 300$ ,  $N = 2$  and  $h = 2b$  at: (a) The starting point of the rotation; (b)  $1/8$  of the whole turn; (c)  $1/4$  of the whole turn; (d)  $3/8$  of the whole turn.

# Chapter 4

## Summary and conclusions

A novel semi-implicit IB formulation based on the Schur complement approach was developed for the simulation of incompressible flows in the presence of periodically moving immersed bodies. The developed formulation was successfully verified for the flow generated by an oscillating sphere over the entire examined range of Reynolds numbers and  $A/D$  ratios. As an extension of the work published in [54], the developed approach comprises a generic methodology that makes it possible for us to equip any existing time marching solver of the NS equations based on the segregated pressure-velocity coupling (e.g., SIMPLE, fractional step, projection methods and their derivatives) with the IBM functionality. A distinctive feature of the developed methodology is that it is composed of two stages, namely, pre-computing and time-integration stages. The present methodology intelligently employs parallelism in the time direction by exploiting the periodic character of the immersed bodies, rendering the whole pre-computation stage "embarrassingly" parallel. This fact allows us to boost its computational efficiency. Another distinctive feature of the developed methodology is the increased efficiency of the time integration stage, achieved by incorporating the MUMPS solver and the GPFS technology in the solution procedure. The MUMPS solver enables the efficient storage and reconstruction of matrices by using its  $LU$  factors, while the GPFS facilitates acceleration of the read-write procedures. Overall, the performance characteristics of the developed methodology for all the configurations simulated in the framework of the present study can be summarized in terms of the following quantitative data: RAM consumption – no more than 20 GB; hard disk space required for keeping  $LU$  factors for  $10^3$  time steps over a single period – no

more than 1.5 TB; typical time required for pre-computing a single time step on a single standard Linux server containing 24 cores (48 threads) – less than 10 minutes for a 100x100x150 grid and less than one hour for a denser, 200x200x300 grid (the time required for the pre-computing stage is linearly proportional to the number of servers involved); typical time required for performing a time integration of a single time step on a single standard Linux server containing 24 cores (48 threads) – less than 2 seconds.

Several issues related to the present formulations have been encountered throughout the study. First, the presence of spurious pressure oscillations in simulations of an oscillating sphere with  $Re \geq 300$  and in simulations done for a rotating ellipsoid. Although for the case of the rotating ellipsoid, this artifact could be the result of numerical instability due to violation of the CFL condition<sup>7</sup>, it has been found to be closely related the spatial order of the accuracy of the scheme. Following from the limited accuracy of the present formulation, mitigation of the pressure oscillations will require using much denser computational grids, which may be prohibitive in terms of the overall memory consumption and the time required for completing the pre-computing stage. In the near future, there is room to consider incorporating a higher order of IB formulation accuracy in the developed methodology, such as sharp-interface IB methods based on the cut-cell [68]-[71] or ghost cell [72]-[73] approaches. These formulations will make it possible for us to produce high fidelity results with a reasonable number of computational nodes, which will ultimately lead to reduced memory consumption and time duration of the pre-computing stage and less intensive usage of hard disk space. These formulations will also allow, to perform the simulation of higher Reynolds flows for the configurations investigated in the framework of the current study. As such they will allow for investigation of a the axisymmetry breaking

---

<sup>7</sup> As we are dealing with rotational motion of relatively large immersed bodies, very high values of linear velocity develop at the tips of the immersed body.



flows justifying the effort which was put for the development of the numerical tool capable for the simulation of fully 3D flows. Also, these modifications might also allow us to further investigate the nature of the vortical structures, including the conditions for their onset and shedding, typical of both oscillatory and rotational body configurations. Note however, that the procedure described above will inevitably demand modifications of the original NS solver, and therefore could deteriorate the portability of the developed methodology. In addition, these formulations involve the calculation of the body stress tensor, based on the calculated velocity and pressure fields, which adds further implementation challenges<sup>8</sup>.

The numerical simulations performed in the framework of the present study were restricted to configurations containing either oscillating or rotating bodies, yet it should be stressed that the developed methodology may be used to simulate a wide spectrum of incompressible flows in the presence of immersed bodies with any periodic kinematics. This capability is important for high-fidelity simulations of flows driven by various kinds of rotor machinery, for the undulatory motion of tiny sea creatures and for various biomedical applications in which the flow is driven by peristaltic contraction of the surrounding tissues. All the above applications will be the focus of our future studies, which will also address more efficient parallelizing of the time marching stage of the developed methodology by employing hybrid and distributed memory paradigms. Additionally, we intend to further investigate the accuracy of the developed methodology by examining features related to the instantaneous flow near the domain walls, including an extended analysis of the transport of passive scalar.

---

<sup>8</sup> In the present formulation, the pressure is treated with a DLM approach, which removes the mentioned complexity.

# Bibliography

- [1] S. Lauder & E. Tytell, Hydrodynamics of undulatory propulsion, *Fish Biomechanics (Fish Physiology)* 23 (2006) 425-468.
- [2] S. Ebbens & T. Howse, In pursuit of propulsion at the nanoscale, *Soft Matter* 6 (2010) 726-738.
- [3] W. Wang, S. Duan, S. Ahmed, T. Mallouk & A. Sen, Small power: Autonomous nano- and micromotors propelled by self-generated gradients, *Nano Today* 8 (2013) 531-554.
- [4] F. Nadal & E. Lauga, Asymmetric steady streaming as a mechanism for acoustic propulsion of rigid bodies, *Phys. Fluids* 26 (2014) 082001.
- [5] D. Bray, *Cell Movements: From molecules to Motility*, Garland Science, New York, 1998.
- [6] J. Lighthill, *Mathematical Bio-fluid Dynamics*, SIAM, Philadelphia, 1975.
- [7] C. Brennen & H. Winet, Fluid mechanics of propulsion by cilia and flagella, *Ann. Rev. Fluid Mech.* 9(1) (1977) 339-398.
- [8] E. Lauga & T. Powers, The hydrodynamics of swimming microorganisms, *Rep. Prog. Phys.* 72 (2009) 096601.
- [9] J. Toonder & P. Onck, *Artificial cilia* (RCS Publishing, Cambridge, 2013).
- [10] A. Shields, B. Fiser, B. Evans, M. Falvo, S. Washburn & R. Superfine, Biomimetic cilia arrays generate simultaneous pumping and mixing regimes, *PNAS* 107(36) (2010) 15670-15675
- [11] K. Ramesh, D. Tripathi & O. Bég, Cilia-assisted hydromagnetic pumping of biorheological couple stress fluids, *J. Propul. Power* 8(3) (2019) 221-233.
- [12] B. Evans, A. Shields, R. Carroll, S. Washburn, M. Falvo & R. Superfine, Magnetically actuated nanorod arrays as biomimetic cilia, *Nano Letters* 7(5) (2007) 1428-1434.

- [13] E. Milana, B. Gorissen, S. Peerlinck, M. Volder & D. Reynaerts, Artificial soft cilia with asymmetric beating patterns for biomimetic low-Reynolds-number fluid propulsion, *Adv. Funct. Mater* 29 (2019) 1900462.
- [14] J. Toonder & P. Onck, Microfluidic manipulation with artificial/bioinspired cilia, *Trends Biotechnol.* 31(2) (2013) 85-91.
- [15] M. Asadnia, A. Kottapalli, K. Karavitaki, M. Warkiani, J. Miao, D. Corey & M. Triantafyllou, From biological cilia to artificial flow sensors: biomimetic soft polymer nanosensors with high sensing performance, *Sci. Rep.* 6 (2016) 32955.
- [16] L. Lu, K. Ryu, C. Liu, A magnetic microstirrer and array for microfluidic mixing, *J. Electromech .Syst.* 11(5) (2002) 462-469.
- [17] Y. Huh, T. Park, E. Lee, W. Hong, S. Lee, Development of a fully integrated microfluidic system for sensing infectious viral disease, *Electrophoresis* 29(14) (2008) 2960-2969.
- [18] L. Capretto, W. Cheng, M. Hill, X. Zhang, Micromixing within microfluidic devices, *Topics in current chemistry* 304 (2011) 27-68.
- [19] R. Shamsoddini, M. Sefid, R. Fatehi, ISPH modelling and analysis of fluid mixing in a microchannel with an oscillating or a rotating stirrer, *Eng. Applicat. Comput. Fluid Mech.* 8(2) (2014) 289-298.
- [20] H. Hu, N. Patankar & M. Zhu, Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique, *J. Comput. Phys.* 169 (2001), 427-462.
- [21] E. Hachem. S. Feghali, R. Codina & T. Coupez, Anisotropic adaptive meshing and monolithic Variational Multiscale method for fluid–structure interaction, *Computers & Structures* 122 (2013) 88-100.

- [22] E. Ferrer & A. Willden, A high order discontinuous Galerkin-Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes, *J. Comput. Phys.* 231 (2012) 7037-7056.
- [23] L. Ramires, C. Foulquie, X. Nogueira, S. Khelladi, J. Chassaing & I. Colominas, New high-resolution-preserving sliding mesh techniques for higher-order finite volume schemes, *J. Comput. Phys.* 118 (2015) 114-130.
- [24] B. Zhang & C. Liang, A simple, efficient, and high-order accurate curved sliding-mesh interface approach to spectral difference method on coupled rotating and stationary domains, *J. Comput. Phys.* 295 (2015) 147-160.
- [25] C. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (1972) 252-271.
- [26] R. Mittal, G. Iaccarino, The immersed boundary method, *Ann. Rev. Fluid Mech.* 37 (2005) 239-261.
- [27] J. Mohd-Yusof, Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries, Center for Turbulence Research, Annual Research Briefs (1997) 317-327.
- [28] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *Journal of Computational Physics* 209 (2005) 448-476.
- [29] S. Lin, Y. Chin, J. Hu & Y. Chen, A pressure correction method for fluid-particle interaction flow: Direct-forcing method and sedimentation flow, *Int. J. Numer. Met. Fluids* 67 (2011) 1771-1798.
- [30] T. Kempe & J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle-laden flows, *Journal of Computational Physics* 231 (2012) 3663-3684.

- [31] W. Breugem, A second-order accurate immersed boundary method for fully resolved situations of particle-laden flows, *J. Comput. Phys.* 231 (2012) 4469-4498.
- [32] T. Kempe, M. Lennartz, S. Schwarz, & J. Fröhlich, Imposing the free slip condition with a continuous forcing immersed boundary method, *Journal of Computational Physics* 282 (2015) 183-209.
- [33] D. Lo, C. Lee & I. Lin, An efficient immersed boundary method for fluid flow simulations with moving boundaries, *Appl. Math. Computat.* 328 (2018) 312-337.
- [34] H. Yoon, D. Yu, M. Ha & Y. Park, Three-dimensional natural convection in an enclosure with a sphere at different vertical locations, *Int. J. Heat Mass Transfer* 53 (2010) 3143-3155.
- [35] W. Ren, C. Shu & W. Yang, Boundary condition-enforced immersed boundary method for thermal flow problems with Dirichlet temperature condition and its applications, *Comput. Fluid.* 57 (2012) 40-51.
- [36] W. Ren, C. Shu, W. Y., An efficient immersed boundary method for thermal flow problems with heat flux boundary conditions, *Int. J. Heat and Mass Transfer* 64 (2013) 694-705.
- [37] Y. Gulberg, Y. Feldman, On laminar natural convection inside multilayered spherical shells, *Int. J. Heat and Mass Transfer* 91 (2015) 908-921.
- [38] Y. Li, E. Jung, W. Lee, H. Lee & J. Kim, Volume preserving immersed boundary methods for two-phase fluid flows, *Int. J. Numer. Meth. Fluids* (2012) 842-858.
- [39] R. Dillon, M. Owen & K. Painter, A single-cell-based model of multicellular growth using the immersed boundary method, *AMS Contemp. Math.* 466 (2008) 1-15.
- [40] K. Rejniak, An immersed boundary framework for modelling the growth of individual cells: An application to the early tumor development, *J. Theor. Biol.* 936 (2007) 186-204.

- [41] K. Rejniak, Circulating tumor cells: When a solid tumor meets a fluid microenvironment, *Adv. Exp. Med. Biol.* 936 (2016) 93-106.
- [42] K. Luo, Z. Wang, J. Fan & K. Cen, Full-scale solutions to particle-laden flows: Multidirect forcing and immersed boundary method, *Phys. Rev. E* 76 (2007) 066709.
- [43] K. Luo, Z. Wang, J. Tan & J. Fan, An improved direct-forcing immersed boundary method with inward retraction of Lagrangian points for simulation of particle-laden flows, *J. Comput. Phys.* 376 (2019) 210-227.
- [44] B. Kallemov, A. Bhalla, B. Griffith & A. Donev, An immersed boundary method for rigid bodies, *Comm. App. Math. Comp. Sci.* 11 (2016) 79-141.
- [45] D. Stein, R. Guy & B. Thomases, Immersed boundary smooth extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods, *J. Comput. Phys.* 304 (2016) 252-274.
- [46] S. Liska & T. Colonius, A fast immersed boundary method for external incompressible viscous flows using lattice Green's functions, *J. Comput. Phys.* 331 (2016) 257-279.
- [47] Y. Bao, A. Donev, B. Griffith, D. McQueen & C. Peskin, An Immersed Boundary method with divergence-free velocity interpolation and force spreading, *J. Comput. Phys.* 347 (2017) 183-206.
- [48] D. Stein, R. Guy & B. Thomases, Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains, *J. Comput. Phys.* 335 (2017) 155-178.
- [49] Y. Feldman, Y. Gulbeg, An extension of the immersed boundary method based on the distributed Lagrange multiplier approach, *J. Comput. Phys.* 322 (2016) 248-266.
- [50] A. Spizzichino, S. Goldring & Y. Feldman, The immersed boundary method: application to two-phase immiscible flows, *Commun. Comput. Phys.* 25(1) (2019) 107-134.

- [51] K. Taira & T. Colonius, The immersed boundary method: A projection approach, *Journal of Computational Physics* 225 (2007) 3121-3133.
- [52] Y. Park, M. Ha, C. Choi & J. Park, Natural convection in a square enclosure with two inner circular cylinders positioned at different vertical locations, *Int. J. Heat and Mass Transfer* 77 (2014) 501-518.
- [53] D. Le, B. Khoo & K. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 2119-2130.
- [54] Y. Feldman, Semi-implicit direct forcing immersed boundary method for incompressible viscous thermal flow problems: a Schur complement approach, *Int. J. Heat Mass Transf* 127 (2018) 1267-1283.
- [55] S. Tschigale, T. Kempe & Jochen Fröhlich, A non-iterative immersed boundary method for spherical particles of arbitrary density ratio, *J. Comput. Phys.* 339 (2017) 432-452.
- [56] A. Roma, C. S. Peskin & M. J. Berger, An adaptive version of the immersed boundary method, *Journal of Computational Physics* 153 (1999) 509-534.
- [57] S. V. Patankar & D. B. Spalding, A calculation procedure for heat, mass and momentum in three-dimensional parabolic flows, *Int. J. Heat Mass Transf.* 15 (1972) 1787-1806.
- [58] J. Adams, P. Swarztrauber & R. Sweet, FISHPACK: Efficient FORTRAN subprograms for the solution of separable elliptic partial differential equations, <http://adsabs.harvard.edu/abs/2016ascl.soft09004A> (1999).
- [59] H. Weller, G. Tabor, H. Jasak & C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, *Computers in Physics* 12 (1998) 620-631.

- [60] H. Vitoshkin, A. Gelfgat, On direct inverse of Stokes, Helmholtz and Laplacian operators in view of time-stepper-based Newton and Arnoldi solvers in incompressible CFD, *Communications in Computational Physics* 14 (2013) 1103-1119.
- [61] P. Amestoy, I. Duff, J. Lexcellent & J. Koster, Multifrontal parallel distributed symmetric and unsymmetric solvers, *Comput. Methods Appl. Mech. Engrg.* 184 (1998) 501-520.
- [62] P. Amestoy, I. Duff, J. Lexcellent & J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM J. Matrix Anal. Appl.* 23 (2001) 15-41.
- [63] P. Leopardi, A partition of the unit sphere into regions of equal area and small diameter, *Electron. Trans. Numer. Anal.* 25 (2006) 309–327.
- [64] R. Mei, Flow due to an oscillating sphere and an expression for unsteady drag on the sphere at finite Reynolds number, *J. Fluid. Mech.* 270 (1994) 133-174.
- [65] H. Blackburn, Mass and momentum transport from a sphere in steady and oscillatory flows, *Phys. Fluid.* 14(11) (2002) 3997-4011.
- [66] J. Jeong & F. Hussain, On the identification of a vortex, *J. Fluid Mech.* 285 (1995) 69-94.
- [67] J. Lee, J. Kim, H. Choi & K. Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, *J. Comput. Phys.* 230 (2011) 2677-2695.
- [68] J. Seo & R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, *J. Comput. Phys.* 230(19) 7347-7363.
- [69] Y. Shen, Y. Ren, H. Ding, A 3D conservative sharp interface method for simulation of compressible two-phase flows, *J. Comput. Phys.* 403 (2020) 109107.



- [70] B. Muralidharan, S. Menon, A high-order adaptive Cartesian cut-cell method for simulation of compressible viscous flow over immersed bodies, *J. Comput. Phys.* 321 (2016) 342-368.
- [71] B. Muralidharan, S. Menon, Simulation of moving boundaries interacting with compressible reacting flows using a second-order adaptive Cartesian cut-cell method, *J. Comput. Phys.* 357 (2018) 230-262.
- [72] A. Mark, E. Svenning & F. Edelvik, An immersed boundary method for simulation of flow with heat transfer, *Int. J. Heat Mass Transf.* 56 (2013) 424-435.
- [73] C. Merlin, P. Domingo & L. Vervisch, Immersed boundaries in large eddy simulation of compressible flow, *Flow. Turbul. Combust.* 90(1) (2013) 29-68.

## תקציר

עבודה זו מציגה מתודולוגיית "Immersed Boundary" מורחבת, המבוססת על גישת האכיפה הישירה הבלתי-מפורשת למחצה, להדמיית זרימה בלתי-דחיסה בנוכחות גופים שקועים הנעים בתנועה מחזורית. המתודולוגיה נעזרת בפירוק מרחבי בגישת "משלים שור" (Schur Complement) לאכיפת האילוצים הקינמטיים של אי-החלקה על המשטחים השקועים. המתודולוגיה מחולקת לשלב חישוב מקדים מקבילי לחלוטין (Embarrassingly parallel) ולשלב אינטגרציה בזמן, אשר מנצלים את יתרונותיה של מערכת קבצים על דיסק משותף (GPFS) לצורך כתיבה וקריאה יעילים של כמויות מידע גדולות. המתודולוגיה שפותחה ניתנת להטמעה נוחה בכל פותרן נאווייה-סטוקס ממשפחת "Pressure-Velocity Segregated" המבוסס על גישת ה-"Projection" או ה-"Fractional Step". בנוסף, המתודולוגיה מקיימת בדיוק רב את תנאי אי-ההחלקה על משטחי הגוף השקוע המבצע תנועה מחזורית. המתודולוגיה אומתה בצורה מקיפה על-ידי יישומה בהדמיית מספר משטרי זרימה מייצגים, המתפתחים בנוכחות כדור תונד. יכולותיה בהדמיית זרימה בלתי-דחיסה הנוצרת על-ידי גוף אחד או יותר בעלי קינמטיקה מחזורית כללית, הודגמו על-ידי הדמיית הזרימות המתפתחות בנוכחות צמד כדורים תונדים באנטי-פאזה ובנוכחות אליפסואיד סובב. המאפיינים הפיזיקאליים של הזרימות המתפתחות, במושגים של השינוי בזמן של מקדם הגרר (במקרה הראשון) או של המומנט (במקרה השני), הוצגו כפונקציה של ערכי הריינולדס (Reynolds). החזיה (Visualization) של המבנים הערבוליים המאפיינים את הזרימות המתפתחות בוצעה על-ידי הצגת ה-Isosurfaces של קריטריון  $\lambda_2$ .



אוניברסיטת בן-גוריון בנגב  
הפקולטה למדעי ההנדסה  
המחלקה להנדסת מכונות


**פיתוח מתודולוגיה נומרית מבוססת שיטת**  
**Immersed Boundary**  
**להדמיית זרימה צמיגה בלתי דחיסה**  
**בנוכחות גופים קשים הנעים בתנועה מחזורית**

חיבור זה מהווה חלק מהדרישות לקבלת תואר "מגיסטר" בהנדסה

מאת : רפי סלע

מנחה : דוקטור יורי פלדמן

תאריך 27.08.2020

חתימת המחבר. 

תאריך 27.08.2020

אישור המנחה. 

תאריך

אישור יו"ר ועדת תואר שני מחלקתית.

יולי 2020

תמוז תש"פ



אוניברסיטת בן-גוריון בנגב  
הפקולטה למדעי ההנדסה  
המחלקה להנדסת מכונות

**פיתוח מתודולוגיה נומרית מבוססת שיטת**  
**Immersed Boundary**  
**להדמיית זרימה צמיגה בלתי דחיסה**  
**בנוכחות גופים קשים הנעים בתנועה מחזורית**

חיבור זה מהווה חלק מהדרישות לקבלת תואר "מגיסטר" בהנדסה

**מאת: רפי סלע**

יולי 2020

תמוז תש"פ