DIRECT FORCING IMMERSED BOUNDARY METHODS: FINITE ELEMENT VERSUS FINITE VOLUME APPROACH

A Dissertation

by

ANGELO FRISANI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Yassin A. Hassan
Committee Members,	Kalyan Annamalai
	Hamn-Ching Chen
	William H. Marlow
Head of Department,	Yassin A. Hassan

December 2012

Major Subject: Nuclear Engineering

ABSTRACT

Two immersed boundary methods (IBM) for the simulation of conjugate heat transfer problems with complex geometries are introduced: a finite element (IFEM) and a finite volume (IFVM) immersed boundary methods are discussed. In the IFEM a projection approach is presented for the coupled system of time-dependent incompressible Navier-Stokes equations (NSEs) and energy equation in conjunction with the immersed boundary method for solving fluid flow and heat transfer problems in the presence of rigid objects not represented by the underlying mesh. The IBM allows solving the flow for geometries with complex objects without the need of generating a body-fitted mesh. Dirichlet boundary constraints are satisfied applying a boundary force at the immersed body surface. Using projection and interpolation operators from the fluid volume mesh to the solid surface mesh (i.e., the "immersed" boundary) and vice versa, it is possible to impose the extra constraint to the NSEs as a Lagrange multiplier in a fashion very similar to the effect pressure has on the momentum equations to satisfy the divergence-free constraint. The IFEM approach presented shows third order accuracy in space and second order accuracy in time when the simulation results for the Taylor-Green decaying vortex are compared to the analytical solution.

For the IFVM a ghost-cell approach with sharp interface scheme is used to enforce the boundary condition at the fluid/solid interface. The interpolation procedure at the immersed boundary preserves the overall second order accuracy of the base solver. The developed ghost-cell method is applied on a staggered configuration with the Semi-Implicit Method for Pressure-Linked Equations Revised algorithm. Second order

accuracy in space and first order accuracy in time are obtained when the Taylor-Green decaying vortex test case is compared to the IFVM analytical solution.

Computations were performed using the IFEM and IFVM approaches for the two-dimensional flow over a backward-facing step, two-dimensional flow past a stationary circular cylinder, three-dimensional flow past a sphere and two and three-dimensional natural convection in an enclosure with/without immersed body. The numerical results obtained with the discussed IFEM and IFVM were compared against other IBMs available in literature and simulations performed with the commercial computational fluid dynamics code STAR-CCM+/V7.04.006. The benchmark test cases showed that the numerical results obtained with the implemented immersed boundary methods are in good agreement with the predictions from STAR-CCM+ and the numerical data from the other IBMs. The immersed boundary method based of finite element approach is numerically more accurate than the IBM based on finite volume discretization. In contrast, the latter is computationally more efficient than the former.

ACKNOWLEDGMENTS

The author acknowledges the Texas A&M Supercomputing Facility for providing computing resources useful in conducting the research reported in this dissertation.

I would like to thank the Associate Director of the Supercomputing facility Spiros Vellas and his research assistant Ping Luo for their help and support throughout the development of this research project.

I would like to thank Dr. Wolfgang Bangerth, associate professor in the department of mathematics at Texas A&M University for his help and guidance in developing the finite element immersed boundary method as part of the current research.

I also would like to thank my committee chair, Dr. Yassin A. Hassan, and my committee members, Dr. Annamalai, Dr. Chen and Dr. Marlow, for their guidance and support throughout the course of this research.

Finally, thanks to my fiancé Leila June Nahavandi for helping and motivating me in these three years and being there every time I needed support. Thank you.

NOMENCLATURE

ALE	Arbitrary Lagrangian-Eulerian
BDF	Backward Difference Formula
BEs	Boussinesq Equations
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CFL	Courant Friedrichs Lewy
DNS	Direct Numerical Simulation
dg	Degree
EBI	Embedded Boundary Integral
EE	Energy Equation
FD	Fictitious Domain
FDM	Fictitious Domain Method
FMG	Full Multi-Grid
Gr	Grashof Number
IB	Immersed Boundary
IB-LBM	Immersed Boundary-Lattice Boltzmann Method
IBM	Immersed Boundary Method
IIM	Immersed Interface Method
IFEM	Immersed Finite Element Method
IFVM	Immersed Finite Volume Method
ILU	Incomplete Lower-Upper

KKT	Karush-Kuhn-Tuckler
LBE	Lattice Boltzmann Equation
LBB	Ladyzhenskaya-Babuska-Brezzi
LBM	Lattice Boltzmann Method
LM	Lower-Upper
MPI	Message-Passing Interface
NSEs	Navier-Stokes Equations
PDDF	Particle Density Distribution Function
Pe	Peclet Number
Pr	Prandlt Number
Ra	Rayleigh Number
Re	Reynolds Number
SIMPLER	Semi Implicit-Method for Pressure-Linked Equation Revised
sm	Surface Method
St	Strouhal Number
TDMA	Tri-Diagonal Matrix Algorithm
UMFPACK	Unsymmetric Multifrontal Sparse LU Factorization Package
vm	Volume Method

Greek symbols

$\Gamma_i(t)$	Immersed Interface
μ	Fluid Dynamic Viscosity

ρ	Fluid Density
Ω	Computational Domain
$\Omega_f(t)$	Fluid Region
$\Omega_s(t)$	Solid Region

Mathematical Operators

B	Divergence Operator
B^T	Gradient Operator
$\underline{F}$	Volumetric Force Vector
$\underline{f}$	Non-Dimensional Volumetric Force Vector
L	Implicit Viscous Matrix
M	Mass Matrix
N	Implicit Advection Matrix
P	Pressure
p	Non-Dimensional Pressure
p'	Non-Dimensional Pressure Correction
$\underline{U}$	Velocity Vector
$\underline{u}$	Non-Dimensional Velocity Vector
U_0	Reference Velocity
U	Reference Velocity

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

The numerical solution of the Navier-Stokes equations (NSEs) requires a spatial and temporal discretization for Finite Difference, Finite Volume and Finite Element methods. The former is performed through a triangulation or mesh of the spatial domain. Most of the commercial flow solvers are based on body-fitted mesh tools (i.e., the mesh conforms to all computational domain surfaces). This approach allows to have a direct imposition of the initial and boundary conditions at the surfaces of immersed bodies. In the Computational Fluid Dynamics (CFD) community an increased interest in solution algorithms with non-body conforming grids has been shown, in particular for flow through geometries that change (e.g., vibration of the spacer grids in a fuel assembly of a nuclear reactor core). It is difficult to simulate meshes that accurately resolve this possibly moving geometry. One approach to deal with this situation is to use a fixed mesh for the overall volume of interest, without trying to resolve the moving geometry, and to enforce fluid boundary conditions on these "immersed" boundaries through constraints posed in addition to the flow equations. The same approach can also be used for cases where the geometry may not be moving but is so complex that it cannot be adequately resolved using non-body fitted meshes.

Methods that follow this approach have been introduced in a number of contexts and are typically referred to as Immersed Boundary Method (IBM), Immersed Interface Method (IIM), Embedded Mesh, Fictitious Domain Method (FDM), differing in the details of the additional constraints implementation and enforcement at the fluid/solid

interface. These methodologies share the idea that the spatial discretization of both fluid and solid regions belongs to the same underlying mesh, and the mesh points in general do not conform to the fluid/solid interface. The fact that the mesh does not have to conform to the solid objects present in the computational domain makes the whole process of meshing orders of magnitude less costly than the body-fitted approach. In general a regular parallelepiped is used with uniform grids (i.e., Cartesian mesh), or an adaptive mesh refinement to have a better discretization close to body surface. The immersed boundaries (i.e., the obstacles which are usually considered to be "immersed" bodies) are discretized with an overlapping mesh from which the only information required is the location of the points describing the solid surface. The IBM is particularly efficient for immersed moving objects, where it does not require a new mesh at each time step, but only the updated position of the points describing the immersed body surface. In fact, while the immersed body may move and/or change shape, this elastic deformation is typically primarily a translation and updating the immersed body mesh is therefore much simpler than it would be to update a mesh that always tracks the shape of the fluid-filled part of the domain.

1.1 Overview of Existing IBM Methodologies

Two possible approaches can be used to determine the boundary force in the IBM: feedback forcing, and direct-forcing methods. In feedback forcing methods the boundary force is computed as a feedback effect on the position/velocity at the immersed body boundary points which must satisfy the no-slip constraint at the next iteration. In

contrast, in direct-forcing methods, the boundary force is directly determined by using the flow equations, such as the Navier-Stokes momentum equations, or the lattice Boltzmann equation.

When IBMs are implemented, interface schemes are required at the solid surface, since in general, the immersed body interface does not match the underlying computational mesh. Diffuse or sharp interface schemes can be used. In diffuse interface schemes, the forcing points, where the boundary force is evaluated, are located on the solid boundary. In the sharp interface scheme the forcing points are distributed on the computational nodes close to the fluid/solid interface. Using diffuse interface schemes, it is necessary to project the boundary force effect on the computational nodes. The projection operation adds a diffuse effect of the boundary force, hence the name diffuse interface schemes. In sharp interface schemes, the velocity on the forcing nodes is directly determined by interpolation, so that the corresponding boundary point satisfies the no-slip condition.

The feedback forcing IBM was originally introduced by Peskin [1] to simulate flow inside a heart with flexible valves. The forcing function was computed using Hooke's law [1, 2]. This technique was later extended to rigid bodies by taking a very large value for the spring constant [3, 4]. Goldstein et al. [5] applied the concept of feedback control to compute the force on the rigid immersed surface. For this methodology the choice of gain (stiffness) remains *ad hoc* and large gains result in stiff equations. More details on the feedback forcing IBM can be found in [6 – 9].

The IBM received more attention in recent years. It has been applied also for interaction of moving body and fluid flow [10 – 12]. In the original IBM, the structure is considered embedded into the fluid. Additional source terms are imposed to simulate the force applied by the structure to the fluid. Depending on how the source terms describing the presence of the immersed object are imposed it is possible to have a continuous forcing approach, where the source terms are directly applied in the continuous equations, or a discrete forcing approach, where the forcing terms are imposed in the resulting system of linear equations.

Lima E Silva et al. [13] proposed a different approach to compute the forcing term, based on the evaluation of the various momentum equation terms at the rigid boundary (direct-forcing). Second order Lagrange polynomials approximation are used to determine the forcing term. Another similar approach that combines the original IBM with direct and explicit forcing was introduced by Uhlmann [14] for the simulation of particulate flows. The forcing term at the boundary is evaluated based on the no-slip constraint for the fluid velocity at the immersed boundary surface given by the rigid-body prescribed motion and a preliminary velocity obtained explicitly without the application of the forcing term.

Other Cartesian grid methods have been developed. We mention the ghost-cell finite difference approach of Tseng and Ferziger [15], Majumdar [16], Ghias et al. [17, 18], Mittal et al. [19] and the cut-cell finite volume approach [20, 21]. In the ghost-cell approach, the ghost cells are interior to the body and have at least one neighbor in the fluid region. The required ghost-cell values are extrapolated to impose the boundary

condition implicitly. In the cut-cell approach, the cells in the Cartesian grid cut by the boundary are identified and reshaped. Polynomial interpolation functions are then used to approximate the fluxes and gradients on the faces of these cells while preserving second-order accuracy. The extension of cut-cell methods to three dimensions is not straightforward because of the complex cut-cell procedure.

Another approach for treating the immersed boundary is the Embedded Boundary Integral (EBI) method [22], where the irregular flow domain is embedded inside a larger regular domain for which fast solvers are available. The EBI method uses an integral formulation to compute the jumps of the velocity and its derivatives at the interface. Then the jumps are used as source terms at the grid points close to the interface. This idea is similar to the Immersed Interface Method (for more details see [23 – 26]).

The Lattice Boltzmann Method (LBM) has been extensively used to simulate complex flows as an alternative to the NSEs (see [27, 28]). The Lattice Boltzmann Equation (LBE) is a kinetic equation of particle density distribution functions (PDDF) discretized on a Cartesian grid. The moments of PDDFs provide hydrodynamic variables such as density, velocity, etc., at mesh points. The use of Cartesian grids motivates the coupling of LBM with IBM, which is referred as the Immersed Boundary-Lattice Boltzmann Method (IB-LBM). By replacing the NSEs with the LBE for flow field calculations, the same IBM discussion applies to both methodologies. For brevity we refer to Kang and Hassan [29] for an extensive discussion of IB-LBM and further references therein.

Boffi et al. [30] presented a finite element implementation of the IBM using Dirac function to determine the fluid-structure interaction force. Space-time discretization stability issues are addressed by Boffi [31] and Heltai [32]. Zhang et al. [33] present the immersed finite element method for the solution of fluid-structure interactions with deformable solids, where a Lagrangian mesh for the solid moves on top of an Eulerian mesh for the fluid covering the entire domain. A discretized delta function is used for the fluid/solid coupling at the interface.

Glowinski et al. [34, 35] proposed a Distributed Lagrange Multiplier (DLM) Fictitious Domain (FD) method which was further extended to obtain finite element solutions around moving rigid bodies using meshes which are not body-fitted [36 – 39].

The DLM-FD method was improved by avoiding the explicit calculation of the Lagrange multipliers by Patankar et al. [40, 41] and introducing body forces to enforce the rigidity constraint which they named non-DLM/FD method. The same method is used by Yu and Shao [42] and Diaz-Goano et al. [43] with minor differences in the non-DLM-FD formulation. Later Sharma and Patankar [44], Veeramani et al. [45] and Apte et al. [46] used the same non-DLM-FD procedure. Sharma and Patankar [44] implemented the method with finite volume and Semi Implicit Method for Pressure Linked Equation Revised (SIMPLER) algorithm [47, 48] for pressure-velocity coupling. Apte et al. [46] used a standard fractional step finite volume method, while Veeramani et al. [45] implemented the non-DLM/FD method in a finite element framework.

An IBM implemented inside an edge-based finite element solver is presented by Löhner et al. [49, 50]. The embedded surfaces are represented by a triangulation, while

the immersed bodies are given by a tetrahedral mesh independent of the computational mesh.

Codina et al. [51] present a finite element solution of the flow around immersed objects using the fixed mesh Arbitrary Lagrangian-Eulerian (ALE) approach. The solid boundary is represented using a level set function defined on a background fixed mesh, with the moving boundary treated by an ALE technique in the region close to the immersed surface, and least-squares approximations are used for the boundary conditions on the immersed body.

Ilinca and Hétu describe in [52, 53] a 3-D finite element with IBM represented using a level set function. The immersed boundary is discretized by the same interpolation functions used to solve the flow problem.

Taira and Colonius [54] use finite volume approach for a new formulation of the immersed boundary method with a structure identical to the traditional fractional step method for incompressible flow over bodies with prescribed surface motion. The boundary force applied at the immersed surface to satisfy no-slip constraint is determined using regularization and interpolation operators. The immersed boundary force acts as a Lagrange multiplier to satisfy the no-slip condition, in a way similar to the effect of pressure on the momentum equations to satisfy the divergence-free constraint. This approach completely removes all tuning parameters for the boundary force, allowing to formulate the IBM in a general framework for rigid bodies and bodies with prescribed surface motion.

1.2 Current Approach

In this work a finite element approach and a finite volume approach coupled with the Immersed Boundary Method and direct-forcing are used to solve the time-dependent incompressible NSEs and the energy equation for flow around stationary rigid bodies. In the Immersed Finite Element Method (IFEM) implementation, the boundary forces are treated in a way similar to the discretized pressure. For the incompressible NSEs, the pressure can be assumed as a Lagrange multiplier required to guarantee the divergence-free constraint (see [36 – 38]). Introducing a projection operator for the velocity field, the no-slip boundary condition at the fluid/solid interface can be viewed as an additional constraint. Then, the force acting on the fluid at the immersed surface can be easily determined in a fully implicit way. The discretized incompressible NSEs with IBM still present a structure algebraically similar to the traditional fractional step method. Although fractional step methods with immersed boundary techniques have been previously implemented, the IBM has not been considered as part of the fractional step method for the finite element formulation to the authors knowledge. As pointed out in [52], the use of finite element methods allows to discretize the immersed boundary surface with the same shape functions used to solve the flow problems, and an *ad hoc* projection and interpolation operators are not necessary.

On the other hand, for the Immersed Finite Volume Method (IFVM) implemented, an explicit sharp interface scheme was used, where the velocity on the forcing nodes inside the embedded body was directly determined by interpolating the velocity in the fluid regions nodes close to the immersed body surface, in such a way to

satisfy the no-slip boundary condition at the at the fluid/solid interface. The boundary conditions on the immersed body are imposed by using the ghost-cell approach, where ghost cells are defined as additional cells inside the immersed body, with at least one neighbor in the fluid domain. No *ad hoc* constants are used for the IFVM procedure, and no momentum forcing terms are necessary for the fluid cells. Consequently the method results in a "sharp" representation of the immersed boundary. The boundary conditions for the immersed body are imposed exactly at the fluid/solid interface without any spreading of the boundary forcing term into the fluid. The coupling between pressure and velocity was achieved by implementing a Semi-Implicit Method for Pressure-Linked Equation Revised (SIMPLER) algorithm on a staggered grid arrangement [47, 48]. A fully implicit scheme is used for time discretization, and a second order upwind scheme is used for discretizing the flux term (convection fluxes). Central difference discretization is used for the viscous term. A Geometric Multigrid Method was used to accelerate the convergence rate of the SIMPLER algorithm. Note that in literature are available only a few studies using pressure correction methods for solving the NSEs coupled with the IBM. Shen et al. [55, 56] used SIMPLEC [48, 57] type methods for pressure-velocity coupling.

In the finite element formulation we follow the approach of Perot [58] and Chang et. al. [59] where the fractional step method is written as a block-LU decomposition. The force generated at the immersed boundary surface and the no-slip constraint are determined in a way similar to the Distributed Lagrange Multiplier (DLM) method of Glowinski [36].

In Section 2 is presented the mathematical formulation for the IFEM approach, the traditional fractional step method is discussed and the immersed boundary formulation coupled with the projection method chosen is introduced. In Section 3 is presented the mathematical formulation for the IFVM approach. The spatial and temporal accuracy of the presented direct-forcing IBMs for time-dependent incompressible NSEs with fractional step method and finite element/volume discretization was investigated for the Taylor-Green decaying vortex for which an analytical solution is available. For the IFEM the comparison shows a third order accuracy in space as expected by using bi-quadratic shape functions for discretizing the velocity field, and a second order accuracy in time congruent with the discretization scheme used. For the IFVM a second order accuracy in space and a first order accuracy in time were determined.

Computations were performed for two-dimensional and three-dimensional benchmark cases considering fluid flow and heat transfer problems. The numerical results were compared with both experimental data, an extensive review of numerical simulations present in literature, and simulation performed with the commercial code STAR-CCM+/V7.04.006. In Section 4 are discussed the analyses performed and the comparison of the numerical results are shown. In Section 5 are outlined the main differences between the IFEM and IFVM approaches developed. In Section 6 is presented the scalability analysis for the parallel version of the IFVM approach. In Section 7 are presented the conclusions and future developments for the implemented immersed boundary formulations.

2. IMMERSED FINITE ELEMENT METHOD PROBLEM ASSESSMENT

In this work are considered incompressible fluid flow problems on a bounded computational domain Ω, formed by the fluid region $\Omega_f(t)$ and the solid region $\Omega_s(t)$. In Figure 1 is shown the reference computational domain. The immersed interface $\Gamma_i(t) = \partial\Omega_f \cap \partial\Omega_s$ at time t represents a boundary for the fluid region. Therefore boundary conditions (i.e., forces) on the fluid contained in $\Omega_f(t)$ must be present to respect the no-slip constraint.

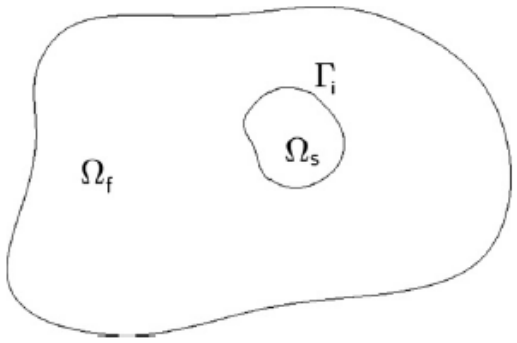

Figure 1. Reference computational domain.

2.1 Time-Dependent Incompressible Navier-Stokes Equations and Boundary Conditions

The equations describing the fluid flow motion considered are the time-dependent incompressible NSEs:

$$\rho\left(\frac{\partial \underline{U}}{\partial t} + \underline{U}\cdot\nabla\underline{U}\right) = -\nabla P + \mu\nabla\cdot\left(\nabla\underline{U} + \left(\nabla\underline{U}\right)^T\right) + \underline{F} \quad in\ \Omega_f \times]0,T[\qquad (1)$$
$$\nabla\cdot\underline{U} = 0$$

where $\underline{U}$ is the velocity vector, ρ is the fluid density, P is the pressure, μ is the dynamic viscosity and $\underline{F}$ is the volumetric force vector. If we use a reference velocity U_0 and reference length L, the time-dependent incompressible NSEs can be written in non-dimensional form and eq. (1) reads as:

$$\begin{aligned} &\frac{\partial \underline{u}}{\partial t} + \underline{u}\cdot\nabla\underline{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right) + \underline{f} \qquad in\ \Omega_f \times]0,T[\\ &\nabla\cdot\underline{u} = 0 \end{aligned} \tag{2}$$

where $\underline{u} = \underline{U}/U_0$ is the non-dimensional velocity vector, $p = P/\rho U_0^2$ is the non-dimensional pressure, $\underline{f} = \underline{F}\bullet L/\rho U_0^2$ is the non-dimensional volumetric force vector, and Re is the Reynolds number defined as:

$$\mathrm{Re} = \frac{\rho U L}{\mu} \tag{3}$$

The initial conditions are:

$$\underline{u}\big|_{t=0} = \underline{u}_0 \qquad in\ \Omega_f \tag{4}$$

and the boundary conditions associated with the momentum-continuity equations are:

$$\underline{u} = \underline{u}_D\left(\underline{x}\right), \quad for\ \underline{x} \in \Gamma_D \tag{5}$$

$$-\frac{1}{\mathrm{Re}}\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right)\cdot\underline{n} + p\underline{n} = \underline{t}\left(\underline{x}\right), \quad for\ \underline{x} \in \Gamma_t \tag{6}$$

where Γ_D is the portion of fluid boundary $\partial\Omega_f$ where Dirichlet conditions are imposed, and $\underline{t}\left(\underline{x}\right)$ is the total stress imposed on the remaining fluid boundary $\Gamma_t = \partial\Omega_f \backslash \Gamma_D$. Since no-slip constraint is assumed at the interface fluid/solid region, Dirichlet boundary conditions would be imposed at the fluid/solid interface Γ_i. We are using the IBM to determine the effect of the solid body on the fluid flow, therefore, Γ_i is not directly

represented by the finite element discretization. Section 2.4 discusses the immersed boundary formulation used to enforce velocity boundary conditions on the immersed surface.

2.2 Finite Element Formulation of the Time-Dependent Incompressible Navier-Stokes Equations

If we write the time-dependent incompressible NSEs system in vector form:

$$\begin{pmatrix} \frac{\partial \underline{u}}{\partial t} + \underline{u}\cdot\nabla\underline{u} + \nabla p - \frac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right) \\ -\nabla\cdot\underline{u} \end{pmatrix} = \begin{pmatrix} \underline{f} \\ 0 \end{pmatrix} \tag{7}$$

the weak form is obtained by forming the dot product from the left of eq. (7) with a vector-valued test function $\Phi = (\underline{v}, q)$, and integrating over the entire domain Ω_f. The weak form read as:

$$\begin{aligned} &\left(\underline{v}, \frac{\partial \underline{u}}{\partial t}\right)_{\Omega_f} + \left(\underline{v}, \underline{u}\cdot\nabla\underline{u}\right)_{\Omega_f} - \frac{1}{\mathrm{Re}}\left(\underline{v}, \nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right)\right)_{\Omega_f} + \\ &\left(\underline{v}, \nabla p\right)_{\Omega_f} - \left(q, \nabla\cdot\underline{u}\right)_{\Omega_f} = \left(\underline{v}, \underline{f}\right)_{\Omega_f} \end{aligned} \tag{8}$$

where $(\bullet,\bullet)_\Omega$ represents a volume integral over Ω. Eq. (8) has to hold for all vector-valued test function $\Phi = (\underline{v}, q)$. Defining the symmetric tensor $\underline{\underline{\varepsilon}}\left(\underline{u}\right) = \frac{1}{2}\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right)$, and integrating by parts the third and fourth terms on the lhs of eq. (8) we have:

$$\begin{aligned} &\left(\underline{v}, \frac{\partial \underline{u}}{\partial t}\right)_{\Omega_f} + \left(\underline{v}, \underline{u}\cdot\nabla\underline{u}\right)_{\Omega_f} + \frac{2}{\mathrm{Re}}\left(\nabla\underline{v}, \underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\Omega_f} - \frac{2}{\mathrm{Re}}\left(\underline{n}\otimes\underline{v}, \underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\partial\Omega_f} \\ &-\left(\nabla\cdot\underline{v}, p\right)_{\Omega_f} + \left(\underline{n}\cdot\underline{v}, p\right)_{\partial\Omega_f} - \left(q, \nabla\cdot\underline{u}\right)_{\Omega_f} = \left(\underline{v}, \underline{f}\right)_{\Omega_f} \end{aligned} \tag{9}$$

The Dirichlet boundary conditions for the velocity are:

$$\underline{u} = \underline{u}_D\left(\underline{x}\right), \quad for\ \underline{x} \in \Gamma_D \tag{10}$$

Since the test function $\underline{v}$ belongs to the tangent space of the solution variable $\underline{v} = 0, \quad for\ \underline{x} \in \Gamma_D$, then:

$$-\frac{2}{\mathrm{Re}}\left(\underline{n} \otimes \underline{v}, \underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\Gamma_D} + \left(\underline{n} \cdot \underline{v}, p\right)_{\Gamma_D} = 0 \tag{11}$$

As expected, strongly imposed boundary conditions do not appear in the weak form. For Neumann-type boundary conditions on the rest of the boundary $\Gamma_N = \partial\Omega_f \backslash \Gamma_D$ we rewrite the boundary terms as:

$$-\frac{2}{\mathrm{Re}}\left(\underline{n} \otimes \underline{v}, \underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\Gamma_N} + \left(\underline{n} \cdot \underline{v}, p\right)_{\Gamma_N} = \left(\underline{v}, \underline{n} \cdot \left[pI - \frac{2}{\mathrm{Re}}\underline{\underline{\varepsilon}}\left(\underline{u}\right)\right]\right)_{\Gamma_N} = \left(\underline{v}, \underline{t}\left(\underline{x}\right)\right)_{\Gamma_N} \tag{12}$$

where:

$$\underline{t}\left(\underline{x}\right) = \underline{n} \cdot \left[pI - \frac{2}{\mathrm{Re}}\underline{\underline{\varepsilon}}\left(\underline{u}\right)\right] \tag{13}$$

Eq. (13) is formally identical to eq. (6). With the Neumann boundary conditions available, the weak form eq. (9) can be written as:

$$\begin{aligned}\left(\underline{v}, \frac{\partial \underline{u}}{\partial t}\right)_{\Omega_f} + \left(\underline{v}, \underline{u} \cdot \nabla \underline{u}\right)_{\Omega_f} + \frac{2}{\mathrm{Re}}\left(\nabla \underline{v}, \underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\Omega_f} \\ -\left(\nabla \cdot \underline{v}, p\right)_{\Omega_f} - \left(q, \nabla \cdot \underline{u}\right)_{\Omega_f} = \left(\underline{v}, \underline{f}\right)_{\Omega_f} - \left\langle \underline{v}, \underline{t}\left(\underline{x}\right)\right\rangle_{\Gamma_N}\end{aligned} \tag{14}$$

where $<\bullet,\bullet>_\Gamma$ represents an integral over the surface Γ. It is also possible to combine Dirichlet and Neumann boundary conditions by enforcing each of them for some of the velocity components. The outflow boundary condition is obtained requiring the flow to be perpendicular to the boundary (i.e., the tangential velocity to the outflow

surface is set to zero). The other velocity components are constrained setting to zero the normal component of the normal stress on the outflow surface:

$$\begin{aligned} &\underline{u}_t = \left(1 - \underline{n} \otimes \underline{n}\right)\underline{u} = 0 \\ &\underline{n} \cdot \left(\underline{n} \cdot \left[pI - \frac{2}{\mathrm{Re}} \underline{\underline{\varepsilon}}\left(\underline{u}\right) \right] \right) = 0 \end{aligned} \tag{15}$$

where $\underline{n}$ is the normal to the outflow surface. The discretization of the weak form for the incompressible NSEs with Dirichlet and Neumann boundary conditions reads:

$$\begin{aligned} &find\ \underline{u} \in V_g = \left\{ \varphi \in H^1\left(\Omega\right)^d : \varphi_{\Gamma_D} = \underline{u}_D \right\},\ p \in Q = L^2\left(\Omega\right) s.t. \\ &\left(\underline{v}, \frac{\partial \underline{u}}{\partial t} \right)_\Omega + \left(\underline{v}, \underline{u} \cdot \nabla \underline{u} \right)_\Omega + \frac{2}{\mathrm{Re}} \left(\nabla \underline{v}, \underline{\underline{\varepsilon}}\left(\underline{u}\right) \right)_\Omega - \\ &\qquad \left(\nabla \cdot \underline{v}, p \right)_\Omega - \left(q, \nabla \cdot \underline{u} \right)_\Omega = \left(\underline{v}, \underline{f} \right)_\Omega - \left\langle \underline{v}, \underline{t}\left(\underline{x}\right) \right\rangle_{\Gamma_N} \\ &\forall\ \underline{v} \in V_g = \left\{ \varphi \in H^1\left(\Omega\right)^d : \varphi_{\Gamma_D} = 0 \right\}, q \in Q \end{aligned} \tag{16}$$

where for generality we used Ω referring to the fluid region Ω_f. These equations represent a saddle point problem. A solution only exists if the function spaces in which we search for a solution satisfy certain conditions, referred as the Babuska-Brezzi or Ladyzhenskaya-Babuska-Brezzi (LBB) conditions [61]. The continuous functions chosen satisfy the required conditions. When we discretize the equations by replacing the continuous variables and test functions by finite element functions (i.e., polynomials of order n) in finite dimensional spaces $V_{g,h} \subset V_g, Q_h \subset Q$, we have to make sure that also $V_{g,h}, Q_h$ satisfy the LBB conditions. For the time-dependent incompressible NSEs a number of possible choices is available which will respect the LBB conditions. A simple and accurate choice used for the present work is $\underline{u}_h \in Q_{p+1}^d,\ p_h \in Q_p^d$ i.e., to use elements

one order higher for the velocity vector than for the pressure. Then the discretized problem for the time-dependent incompressible NSEs reads:

$$
\begin{aligned}
&\textit{find } \underline{u}_h \in Q_{p+1}^d,\ p_h \in Q_p^d \ s.t. \\
&\left(\underline{v}_h, \frac{\partial \underline{u}_h}{\partial t}\right)_\Omega + \left(\underline{v}_h, \underline{u}_h \cdot \nabla \underline{u}_h\right)_\Omega + \frac{2}{\mathrm{Re}}\left(\nabla \underline{v}_h, \underline{\underline{\varepsilon}}\left(\underline{u}_h\right)\right)_\Omega \\
&\qquad -\left(\nabla \cdot \underline{v}_h, p_h\right)_\Omega - \left(q_h, \nabla \cdot \underline{u}_h\right)_\Omega = \left(\underline{v}_h, \underline{f}\right)_\Omega - \left\langle \underline{v}_h, \underline{t}\left(\underline{x}\right)\right\rangle_{\Gamma_N} \\
&\forall\, \underline{v}_h \in V_{g,h}, q_h \in Q_{g,h}
\end{aligned}
\tag{17}
$$

2.3 Fractional Step Method

The weak form of the Navier-Stokes equations leads to a linear system for the nodal values of the velocity vector and pressure. It is necessary to address the formulation to be chosen for the transient term, the viscous term and the convective term in the momentum equations. For the sake of generality, let's assume that appropriate discretization for these three terms are chosen. Then the system of eq. (17) can be written in matrix form as:

$$
\begin{bmatrix} M+N+L & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \underline{u}^{n+1} \\ p^{n+1} \end{pmatrix} = \begin{pmatrix} \underline{f}^{\,n} \\ 0 \end{pmatrix} + b.c. \tag{18}
$$

where M, N and L are sparse matrices representing the mass matrix, the implicit advection term and the implicit viscous term in the momentum equation, respectively. B^T represent the gradient operator acting on the pressure, B is the divergence operator projecting the velocity vector on a divergence-free field. The right hand side of eq. (18) consists of the discretized volumetric force and the explicit terms from the momentum equations. Inhomogeneous terms from the boundary conditions must also be considered.

One possible approach to solve the system of eq. (18) would be to use the Schur complement. However, the condition number for the Schur complement is proportional to $\left(\frac{1}{\Delta t}\right)^2$, where Δt is the time step used for the temporal discretization. This makes the system very difficult to solve. The reason why the time-dependent incompressible NSEs show this behavior is the character of the pressure term. The velocity and the pressure are coupled through the constraint $\nabla \cdot \underline{u} = 0$, for which the pressure is a Lagrange multiplier. The fractional step method (or equivalently projection method) is a well-established approach (see [58, 59, 62, 63, 64, 65]) used to decouple the pressure from the velocity field. In the present work, the second order backward difference formula (BDF2) is used to discretize the time derivative; the viscous term is treated in a fully implicit way. A complete description of the projection method used can be found in [66, 67] referred to as second-order pressure-correction scheme.

An overview of the steps involved and the equations solved is presented below. The objective is to obtain a sequence of velocities and pressures $\left\{\underline{u}^k, p^k\right\}$ and pseudo pressure $\left\{\phi^k\right\}$. From the initial conditions, using a first order method we compute $\left\{\underline{u}^0, p^0, \phi^0\right\}$ and $\left\{\underline{u}^1, p^1, \phi^1 = p^1 - p^0\right\}$. The fractional step method consists of the following steps:

- Step 0: Extrapolation. Define:

$$\underline{u}^* = 2\underline{u}^k - \underline{u}^{k-1}, p^t = p^k + \frac{4}{3}\phi^k - \frac{1}{3}\phi^{k-1} \tag{19}$$

- Step 1: Diffusion step. We solve the time-dependent incompressible NSE linear system for $\underline{u}^{k+1}$:

$$\frac{1}{2\Delta t}\left(3\underline{u}^{k+1}-4\underline{u}^{k}+\underline{u}^{k-1}\right)+\underline{u}^{*}\cdot\nabla\underline{u}^{k+1}+\frac{1}{2}\left(\nabla\cdot\underline{u}^{*}\right)\underline{u}^{k+1} \\ -\frac{2}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u}^{k+1}+\left(\nabla\underline{u}^{k+1}\right)^{T}\right)+\nabla p^{t}=\underline{f}^{k+1} \tag{20}$$

$$\underline{u}^{k+1}\Big|_{\Gamma_D}=\underline{u}_D,\underline{u}^{k+1}\times\underline{n}\Big|_{\Gamma_N}=0 \tag{21}$$

- Step 2: Projection step. A Poisson equation for the pseudo pressure ϕ^{k+1} is solved with the velocity field computed from the diffusion step:

$$\Delta\phi^{k+1}=\frac{3}{2\Delta t}\nabla\cdot\underline{u}^{k+1} \tag{22}$$

$$\partial_n\phi^{k+1}\Big|_{\Gamma_D}=0,\phi^{k+1}\Big|_{\Gamma_N}=0 \tag{23}$$

The boundary conditions chosen for $\underline{u}^{k+1}$ and ϕ^{k+1} are a special case of the outflow boundary conditions addressed in eq. (15).

- Step 3: Pressure correction. Two formulations can be used:
- The Standard form. The pressure is updated by:

$$p^{k+1}=p^{k}+\phi^{k+1} \tag{24}$$

- and the Rotational form. The pressure is updated by:

$$p^{k+1}=p^{k}+\phi^{k+1}-\frac{1}{\mathrm{Re}}\nabla\cdot\underline{u}^{k+1} \tag{25}$$

Following there are a few comments on the projection method used for this work. The advection term $\underline{u}\cdot\nabla\underline{u}$ is replaced by its skew symmetric form $\underline{u}\cdot\nabla\underline{u}+\frac{1}{2}\left(\nabla\cdot\underline{u}\right)\underline{u}$. This

is consistent with the continuous equations (i.e., at convergence $\nabla \cdot \underline{u} = 0$). This choice guarantees unconditional stability of the time-stepping scheme. The second order extrapolation $\underline{u}^*$ of $\underline{u}^{k+1}$ is used. The projection step is a realization of the Helmholtz decomposition:

$$L^2(\Omega)^d = H \oplus \nabla H^1_{\Gamma_2}(\Omega)$$

where:

$$H = \left\{ \underline{v} \in L^2(\Omega)^d : \nabla \cdot \underline{v} = 0, \underline{v} \cdot \underline{n}\big|_{\Gamma_1} = 0 \right\}$$

and:

$$H^1_{\Gamma_2}(\Omega) = \left\{ q \in H^1(\Omega) : q\big|_{\Gamma_2} = 0 \right\}$$

Using this decomposition on $\underline{u}^{k+1}$ we obtain:

$$\underline{u}^{k+1} = \underline{v}^{k+1} + \nabla\left(\frac{2\Delta t}{3} \phi^{k+1} \right) \tag{26}$$

where $\underline{v}^{k+1} \in H$. The divergence of eq. (26) gives eq. (22).

2.4 Immersed Boundary Projection Method

In the following section is described the immersed boundary approach coupled with the fractional step method for the incompressible time-dependent NSEs. The form of eq. (18) is known as the Karush-Kuhn-Tuckler (KKT) system used for constrained optimization problems [68]. It is worth to note that the pressure does not play a direct role in time advancement, acting as a set of Lagrange multipliers to minimize the system energy and satisfy the kinematic constraint of divergence-free velocity field. Following

this idea, it is possible to think of the algebraic constraint of no-slip boundary conditions present at the immersed boundary surface as additional Lagrange multipliers. Therefore, we add the no-slip constraint from the IBM into the fractional step method as additional Lagrange multipliers in a style similar to the approach used by Glowinski et al. [36] in their distributed Lagrange multiplier/fictitious domain method.

2.4.1 The Discretized Time-Dependent Incompressible NSEs with Surface Boundary Force

The IBM introduces a set of Lagrangian points ($\underline{X}_K$) representing the surface $\Gamma_i(t)$ of the immersed body at time t within the computational domain which does not need to conform to the "immersed" body shape (i.e., non-body fitting mesh). In general the IB surface is provided in the form of a CAD file from which we generate a surface mesh with a mesh size sufficiently refined to have a correct coupling of the fluid mesh and the immersed body mesh. At the Lagrangian points, approximate surface forces $\underline{f}_K$ are applied to enforce the no-slip condition along $\Gamma_i(t)$. In Figure 2 is shown a reference setup of the immersed body surface mesh and the underlying fluid mesh with the location of the Lagrangian points where the surface forces are applied. From the figure it is possible to note that, in general, the location of the Lagrangian boundary points does not coincide with the underlying fluid mesh discretization. Therefore, it is necessary to build a projection operator P from the fluid volume mesh to the immersed body surface mesh and an interpolation operator P^T from the surface mesh to the fluid volume mesh.

Writing the time-dependent incompressible NSEs with immersed boundary method, it is possible to have a discretized form of the projection operator, which consists of interpolating the shape functions used for the fluid volume mesh on the immersed boundary surface mesh. This can be easily handled in finite element method framework by the *deal.II* library chosen [69] to perform the numerical simulations.

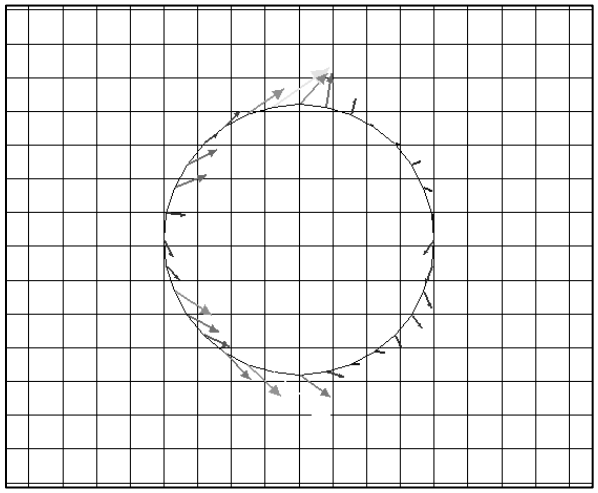

Figure 2. Setup of the Lagrangian points for the IB surface mesh approach.

The incompressible NSEs with a boundary force $\underline{f}_K$, and the no-slip condition at the fluid/solid interface, can be considered as the continuous analog of the IBM:

$$\begin{aligned}
&\frac{\partial \underline{u}}{\partial t} + \underline{u}\cdot\nabla\underline{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right) + \underline{f} \\
&\qquad - \int_{\Gamma_i(t)} \underline{f}_K\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X} - \underline{x}\right)ds \qquad\qquad in\,\Omega\times]0,T[\\
&\nabla\cdot\underline{u} = 0 \qquad\qquad in\,\Omega\times]0,T[\\
&\underline{u}\left(\underline{X}\left(s,t\right)\right) = \int_{\Gamma_i(t)} \underline{u}\left(\underline{x},t\right)\delta\left(\underline{x} - \underline{X}\right)d\underline{x} = \underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right) \qquad on\,\Gamma_i\left(t\right)
\end{aligned} \tag{27}$$

where $\underline{x}$ in Ω (volume mesh), $\underline{X}(s,t)$ in $\Gamma_i(t)$ (surface mesh) is the position of a point on the immersed boundary surface indexed by a position s in the reference configuration at time t, and the last term on the rhs of the first in eq. (27) represents the effect of the immersed body on the fluid, that is the integral of the surface forces $\underline{f}_K$ applied at the fluid/solid interface $\Gamma_i(t)$. For the time being, only rigid fixed immersed boundaries are considered where $\underline{X}(s,t) = \underline{X}(s,0) = \underline{X}(s)$ and $u_{IB}(\underline{X}(s,t)) = 0$. The application of the discussed IBM to rigid immersed bodies moving at a predetermined velocity $u_{IB}(\underline{X}(s,t)) \neq 0$ is straightforward. By forming the dot product from left of eq. (27) with a vector-valued function $\Phi = (\underline{v}, q, \underline{\varphi})$, the weak form of the system of eq. (27) is obtained:

$$\begin{aligned}
&\left(\underline{v}, \frac{\partial \underline{u}}{\partial t}\right)_\Omega + \left(\underline{v}, \underline{u}\cdot\nabla\underline{u}\right)_\Omega - \frac{1}{\mathrm{Re}}\left(\underline{v}, \nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right)\right)_\Omega + \left(\underline{v}, \nabla p\right)_\Omega - \left(q, \nabla\cdot\underline{u}\right)_\Omega = \\
&\qquad \left(\underline{v}, \underline{f}\right)_\Omega - \left(\underline{v}, \int_{\Gamma_i(t)} \underline{f}_K\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X} - \underline{x}\right)ds\right)_\Omega \qquad \text{in } \Omega\times]0,T[\\
&\left\langle \underline{\varphi}, \underline{u}\left(\underline{X}\left(s,t\right)\right)\right\rangle_{\Gamma_i(t)} = \left\langle \underline{\varphi}, \int_\Omega \underline{u}\left(\underline{x},t\right)\delta\left(\underline{x} - \underline{X}\right)d\underline{x}\right\rangle_{\Gamma_i(t)} = \\
&\qquad \left\langle \underline{\varphi}, \underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right)\right\rangle_{\Gamma_i(t)} \qquad \text{on } \Gamma_i\left(t\right)
\end{aligned} \tag{28}$$

where $\left(\bullet,\bullet\right)_\Omega$ and $\left\langle \bullet,\bullet\right\rangle_\Gamma$ represent volume and surface integrals, respectively. Eq. (28) has to hold $\forall\, \Phi = (\underline{v}, q, \underline{\varphi})$. The last term in the first of eq. (28) can be written as:

$$\begin{aligned}
&\left(\underline{v}, \int_{\Gamma_i(t)} \underline{f}_K\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X} - \underline{x}\right)ds\right)_\Omega = \\
&\qquad \int_\Omega \underline{v}\int_{\Gamma_i(t)} \underline{f}_K\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X} - \underline{x}\right)ds\,d\underline{x} = \int_\Omega \underline{v}\left(\underline{x}\right)\underline{\lambda}\left(\underline{x},t\right)d\underline{x}
\end{aligned} \tag{29}$$

where $\underline{\lambda}(\underline{x},t)$ are the projection of the Lagrange multipliers due to the force present at the immersed body interface from the surface mesh to the fluid volume mesh:

$$\underline{\lambda}\left(\underline{x},t\right)=\int_{\Gamma_i(t)}\underline{f}_K\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X}-\underline{x}\right)ds=P^T\underline{\lambda}\left(\underline{X}\left(s,t\right)\right) \tag{30}$$

and P^T represents the interpolation operator (immersed body surface mesh → fluid volume mesh). In the same way, the second of eq. (28) can be written as:

$$\begin{aligned}\left\langle\underline{\varphi},\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}\right\rangle_{\Gamma_i(t)}&=\int_{\Gamma_i(t)}\underline{\varphi}\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}ds=\\ \int_{\Gamma_i(t)}\underline{\varphi}\underline{u}\left(\underline{X}\left(s,t\right)\right)ds&=\left\langle\underline{\varphi},\underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right)\right\rangle_{\Gamma_i(t)}\end{aligned} \tag{31}$$

where $\underline{u}(\underline{X}(s,t))$ represents the fluid velocity projected on the immersed body mesh:

$$\underline{u}\left(\underline{X}\left(s,t\right)\right)=\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}=P\underline{u}\left(\underline{x},t\right) \tag{32}$$

After integrating by parts the third and fourth terms on the lhs in the first of of eq. (28), the discretization of the weak form for the time-dependent incompressible NSEs with Dirichlet and Neumann boundary conditions and IBM reads:

$$\begin{aligned}&\textit{find }\underline{u}\in V_g=\left\{\phi\in H^1\left(\Omega\right)^d:\phi_{\Gamma_D}=\underline{u}_D\right\},\ p\in Q=L^2\left(\Omega\right),\ \underline{\lambda}\in\Lambda_g=\left\{\varphi\in H^1\left(\Omega\right)^{d-1}\right\}\ s.t.\\ &\left(\underline{v},\frac{\partial\underline{u}}{\partial t}\right)_{\Omega}+\left(\underline{v},\underline{u}\cdot\nabla\underline{u}\right)_{\Omega}+\frac{2}{\mathrm{Re}}\left(\nabla\underline{v},\underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\Omega}-\left(\nabla\cdot\underline{v},p\right)_{\Omega}-\left(q,\nabla\cdot\underline{u}\right)_{\Omega}=\\ &\qquad\left(\underline{v},\underline{f}\right)_{\Omega}-\left(\underline{v},P^T\underline{\lambda}\left(\underline{X},t\right)\right)_{\Omega}-\left\langle\underline{v},\underline{t}\left(\underline{x}\right)\right\rangle_{\Gamma_N}\\ &\left\langle\underline{\varphi},P\underline{u}\left(\underline{x},t\right)\right\rangle_{\Gamma_i(t)}=\left\langle\underline{\varphi},\underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right)\right\rangle_{\Gamma_i(t)}=\left\langle\underline{\varphi},P\underline{u}_{IB}\left(\underline{x},t\right)\right\rangle_{\Gamma_i(t)}\\ &\forall\,\underline{v}\in V_g=\left\{\phi\in H^1\left(\Omega\right)^d:\phi_{\Gamma_D}=0\right\},q\in Q,\forall\,\underline{\varphi}\in\Lambda_g=\left\{\varphi\in H^1\left(\Omega\right)^{d-1}\right\}\end{aligned} \tag{33}$$

where strongly imposed boundary conditions (i.e., Dirichlet-type) do not appear in the weak form, whereas Neumann-type boundary conditions are still present (see Section 2.2). The solution for the velocity, pressure and Lagrangian multipliers fields is found in the function spaces $\underline{u}\in V_g=\{\Phi\in H^1(\Omega)^d:\Phi_{\Gamma D}=\underline{u}_D\}$, $p\in Q=L^2(\Omega)$, $\underline{\lambda}\in\Lambda_g=\{\varphi\in H^1(\Omega)^{d-}$

[1]}. When we discretize the NSEs with IBM by replacing the continuous variables and test functions by finite element functions in finite dimensional spaces $V_{g,h} \subset V_g$, $Q_h \subset Q$, $\Lambda_{g,h} \subset \Lambda_g$, we make sure that $V_{g,h}$, Q_h and $\Lambda_{g,h}$ satisfy the LBB conditions by choosing elements one order higher for the velocity vector than for the pressure. The Lagrange multipliers elements can have either the same order of the velocity vector or pressure field. Discretizing the time-dependent incompressible NSEs with IBM and replacing the continuous variables and test functions by finite element functions in finite dimensional spaces we obtain:

$$
\begin{aligned}
&\text{find } \underline{u}_h \in Q_{p+1}^d,\ p_h \in Q_p^d,\ \underline{\lambda}_h \in Q_p^{d-1}\ s.t.\\
&\left(\underline{v}_h, \frac{\partial \underline{u}_h}{\partial t}\right)_\Omega + \left(\underline{v}_h, \underline{u}_h \cdot \nabla \underline{u}_h\right)_\Omega + \frac{2}{\mathrm{Re}}\left(\nabla \underline{v}_h, \underline{\underline{\varepsilon}}\left(\underline{u}_h\right)\right)_\Omega - \left(\nabla \cdot \underline{v}_h, p_h\right)_\Omega \\
&\qquad -\left(q_h, \nabla \cdot \underline{u}_h\right)_\Omega = \left(\underline{v}_h, \underline{f}\right)_\Omega - \left(\underline{v}_h, \underline{\lambda}_h\left(\underline{x}, t\right)\right)_\Omega - \left\langle \underline{v}_h, \underline{t}\left(\underline{x}, t\right)\right\rangle_{\Gamma_N} \qquad (34)\\
&\left\langle \underline{\varphi}_h, \underline{u}_h\left(\underline{X}, t\right)\right\rangle_{\Gamma_i(t)} = \left\langle \underline{\varphi}_h, \underline{u}_{IB}\left(\underline{X}, t\right)\right\rangle_{\Gamma_i(t)}\\
&\forall\, \underline{v}_h \in V_{g,h}, q_h \in Q_{g,h}, \underline{\varphi}_h \in \Lambda_{g,h}
\end{aligned}
$$

2.4.2 The Discretized Time-Dependent Incompressible NSEs with Volume Boundary Force

The immersed body can be also discretized with a volume mesh overlapping on the fluid volume mesh. In this case, the IBM introduces a set of Lagrangian points ($\underline{X}_K$) representing the immersed object $\omega(t)$. The fluid volume mesh does not need to conform to the body shape (i.e., non-body fitting mesh). In general the IB is provided in the form of a CAD file from which we generate a volume mesh with a mesh size sufficiently refined to have a correct coupling of the fluid mesh and the immersed body mesh. At the

Lagrangian points, approximate volume forces f_K are applied to enforce the no-slip condition along the fluid/solid body interface Γ_i and inside the immersed body volume $\omega(t)$. In Figure 3 is shown a reference setup of the immersed body volume mesh and the underlying fluid mesh with the location of the Lagrangian points where the volume forces are applied. Since, in general, the location of the Lagrangian boundary points does not coincide with the underlying fluid mesh discretization, it is necessary to build a projection operator P from the fluid volume mesh to the immersed body volume mesh and an interpolation operator P^T from the immersed boundary volume mesh to the fluid volume mesh. The approach used is analogous to that used for the IBM surface mesh approach, where the discretized form of the projection operator is obtained interpolating the shape functions used for the fluid volume mesh on the immersed boundary volume mesh. The *deal.II* library [69] was used to perform the numerical simulations.

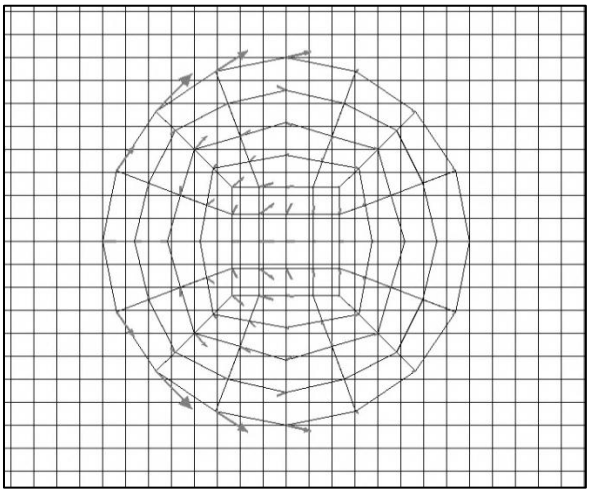

Figure 3. Setup of the Lagrangian points for the immersed boundary volume mesh.

The incompressible NSEs with a IB volume force $\underline{f}_K$, and the no-slip condition enforced on the entire IB volume $\omega(t)$, can be considered as the continuous analog of the IBM:

$$\begin{aligned}
&\frac{\partial \underline{u}}{\partial t} + \underline{u}\cdot\nabla\underline{u} = -\nabla p + \frac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right) + \underline{f} \\
&\qquad - \int_{\omega(t)} \underline{f}_K\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X} - \underline{x}\right)ds \qquad in\,\Omega\times]0,T[\\
&\nabla\cdot\underline{u} = 0 \qquad in\,\Omega\times]0,T[\\
&\underline{u}\left(\underline{X}\left(s,t\right)\right) = \int_{\omega(t)} \underline{u}\left(\underline{x},t\right)\delta\left(\underline{x} - \underline{X}\right)d\underline{x} = \underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right) \quad in\,\omega\left(t\right)
\end{aligned} \tag{35}$$

where $\underline{x}$ in Ω (fluid volume mesh), $\underline{X}(s,t)$ in $\omega(t)$ (immersed body volume mesh) is the position of a point on the immersed body indexed by a position s in the reference configuration at time t, and the last term on the rhs of the first in eq. (35) represents the effect of the immersed body on the fluid, that is the integral of the volume forces $\underline{f}_K$ applied on the immersed body volume $\omega(t)$. For the time being, only rigid fixed immersed boundaries are considered where $\underline{X}(s,t) = \underline{X}(s,0) = \underline{X}(s)$ and $u_{IB}(\underline{X}(s,t)) = 0$. The application of the discussed IBM to rigid immersed bodies moving at a predetermined velocity $u_{IB}(\underline{X}(s,t)) \neq 0$ is straightforward. By forming the dot product from left of eq. (35) with a vector-valued function $\Phi = (\underline{v}, q, \underline{\varphi})$, the weak form of the system of eq. (35) is obtained:

$$\left(\underline{v},\frac{\partial \underline{u}}{\partial t}\right)_{\Omega}+\left(\underline{v},\underline{u}\cdot\nabla\underline{u}\right)_{\Omega}-\frac{1}{\mathrm{Re}}\left(\underline{v},\nabla\cdot\left(\nabla\underline{u}+\left(\nabla\underline{u}\right)^{T}\right)\right)_{\Omega}+\left(\underline{v},\nabla p\right)_{\Omega}-\left(q,\nabla\cdot\underline{u}\right)_{\Omega}=$$

$$\left(\underline{v},\underline{f}\right)_{\Omega}-\left(\underline{v},\int_{\omega(t)}\underline{f}_{K}\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X}-\underline{x}\right)ds\right)_{\Omega} \qquad in\,\Omega\times]0,T[\tag{36}$$

$$\left(\underline{\varphi},\underline{u}\left(\underline{X}\left(s,t\right)\right)\right)_{\omega(t)}=\left(\underline{\varphi},\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}\right)_{\omega(t)}=$$

$$\left(\underline{\varphi},\underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right)\right)_{\omega(t)} \qquad in\,\omega(t)$$

Note that the integral in the second of eq. (36) are carried out over the entire immersed body volume mesh *ω(t)* whereas in the immersed body surface mesh approach the no-slip condition at the solid boundary was enforced by imposing a constraint on the velocity at the fluid/solid interface $\Gamma_i(t)$ as shown in eq. (28). Eq.(36) has to hold $\forall\, \Phi = (\underline{v}, q, \underline{\varphi})$. The last term in the first of eq. (36) can be written as:

$$\left(\underline{v},\int_{\omega(t)}\underline{f}_{K}\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X}-\underline{x}\right)ds\right)_{\Omega}=$$

$$\int_{\Omega}\underline{v}\int_{\omega(t)}\underline{f}_{K}\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X}-\underline{x}\right)ds\,d\underline{x}=\int_{\Omega}\underline{v}\left(\underline{x}\right)\underline{\lambda}\left(\underline{x},t\right)d\underline{x} \tag{37}$$

where $\underline{\lambda}\left(\underline{x},t\right)$ are the projection of the Lagrange multipliers due to the force present on the immersed body from the immersed body volume mesh to the fluid volume mesh:

$$\underline{\lambda}\left(\underline{x},t\right)=\int_{\omega(t)}\underline{f}_{K}\left(\underline{X}\left(s,t\right)\right)\delta\left(\underline{X}-\underline{x}\right)ds=P^{T}\underline{\lambda}\left(\underline{X}\left(s,t\right)\right) \tag{38}$$

and P^T represents the interpolation operator (immersed body volume mesh → fluid volume mesh). In the same way, the second of eq. (36) can be written as:

$$\left(\underline{\varphi},\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}\right)_{\omega(t)}=\int_{\omega(t)}\underline{\varphi}\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}ds=$$
$$\int_{\omega(t)}\underline{\varphi}\underline{u}\left(\underline{X}\left(s,t\right)\right)ds=\left(\underline{\varphi},\underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right)\right)_{\omega(t)} \tag{39}$$

where $\underline{u}(\underline{X}(s,t))$ represents the fluid velocity projected on the immersed body mesh:

$$\underline{u}\left(\underline{X}\left(s,t\right)\right)=\int_{\Omega}\underline{u}\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}=P\underline{u}\left(\underline{x},t\right) \tag{40}$$

After integrating by parts the third and fourth terms on the lhs of eq. (40), the discretization of the weak form for the time-dependent incompressible NSEs with Dirichlet and Neumann boundary conditions and IBM reads:

$$\begin{aligned}
&find\ \underline{u}\in V_g=\left\{\phi\in H^1\left(\Omega\right)^d:\phi_{\Gamma_D}=\underline{u}_D\right\},\ p\in Q=L^2\left(\Omega\right),\ \underline{\lambda}\in\Lambda_g=\left\{\varphi\in H^1\left(\Omega\right)^d\right\}\ s.t.\\
&\left(\underline{v},\frac{\partial\underline{u}}{\partial t}\right)_{\Omega}+\left(\underline{v},\underline{u}\cdot\nabla\underline{u}\right)_{\Omega}+\frac{2}{\mathrm{Re}}\left(\nabla\underline{v},\underline{\underline{\varepsilon}}\left(\underline{u}\right)\right)_{\Omega}-\left(\nabla\cdot\underline{v},p\right)_{\Omega}-\left(q,\nabla\cdot\underline{u}\right)_{\Omega}=\\
&\qquad\left(\underline{v},\underline{f}\right)_{\Omega}-\left(\underline{v},P^T\underline{\lambda}\left(\underline{X},t\right)\right)_{\Omega}-\left\langle\underline{v},\underline{t}\left(\underline{x},t\right)\right\rangle_{\Gamma_N}\\
&\left(\underline{\varphi},P\underline{u}\left(\underline{x},t\right)\right)_{\omega(t)}=\left(\underline{\varphi},\underline{u}_{IB}\left(\underline{X}\left(s,t\right)\right)\right)_{\omega(t)}=\left(\underline{\varphi},P\underline{u}_{IB}\left(\underline{x},t\right)\right)_{\omega(t)}\\
&\forall\,\underline{v}\in V_g=\left\{\phi\in H^1\left(\Omega\right)^d:\phi_{\Gamma_D}=0\right\},q\in Q,\forall\,\underline{\varphi}\in\Lambda_0=\left\{\varphi\in H^1\left(\Omega\right)^d\right\}
\end{aligned} \tag{41}$$

Note the absence of strongly imposed boundary conditions (i.e., Dirichlet-type) in the weak form, whereas Neumann-type boundary conditions are still present.

The solution for the velocity, pressure and Lagrangian multipliers fields is found in the function spaces $\underline{u}\in V_g=\{\Phi\in H^1(\Omega)^d:\Phi_{\Gamma D}=\underline{u}_D\}$, $p\in Q=L^2(\Omega)$, $\underline{\lambda}\in\Lambda_g=\{\varphi\in H^1(\Omega)^{d-1}\}$. When we discretize the NSEs with IBM by replacing the continuous variables and test functions by finite element functions in finite dimensional spaces $V_{g,h}\subset V_g$, $Q_h\subset Q$, $\Lambda_{g,h}\subset\Lambda_g$, we make sure that $V_{g,h}$, Q_h and $\Lambda_{g,h}$ satisfy the LBB conditions by choosing elements one order higher for the velocity vector than for the pressure. The Lagrange

multipliers elements can have either the same order of the velocity vector or pressure field. Discretizing the time-dependent incompressible NSEs with IBM and replacing the continuous variables and test functions by finite element functions in finite dimensional spaces we obtain:

$$
\begin{aligned}
&\textit{find } \underline{u}_h \in Q_{p+1}^d, p_h \in Q_p^d, \underline{\lambda}_h \in Q_p^{d-1} \textit{ s.t.} \\
&\left(\underline{v}_h, \frac{\partial \underline{u}_h}{\partial t}\right)_\Omega + \left(\underline{v}_h, \underline{u}_h \cdot \nabla \underline{u}_h\right)_\Omega + \frac{2}{\mathrm{Re}}\left(\nabla \underline{v}_h, \underline{\underline{\varepsilon}}\left(\underline{u}_h\right)\right)_\Omega - \left(\nabla \cdot \underline{v}_h, p_h\right)_\Omega \\
&\qquad -\left(q_h, \nabla \cdot \underline{u}_h\right)_\Omega = \left(\underline{v}_h, \underline{f}\right)_\Omega - \left(\underline{v}_h, \underline{\lambda}_h\left(\underline{x}, t\right)\right)_\Omega - \left\langle \underline{v}_h, \underline{t}\left(\underline{x}, t\right)\right\rangle_{\Gamma_N} \\
&\left(\underline{\varphi}_h, \underline{u}_h\left(\underline{X}, t\right)\right)_{\omega(t)} = \left(\underline{\varphi}_h, \underline{u}_{IB}\left(\underline{X}, t\right)\right)_{\omega(t)} \\
&\forall\, \underline{v}_h \in V_{g,h}, q_h \in Q_{g,h}, \underline{\varphi}_h \in \Lambda_{g,h}
\end{aligned}
\tag{42}
$$

2.4.3 The Discretized Time-Dependent Incompressible NSEs with Immersed Boundary Projection Method

In Section 2.3 we described the discretization in time and space chosen for the NSEs unsteady term, advection and viscous term, respectively. We also described the fractional step method. Applying the same fractional step method for time discretization to the system of eq. (34) for the IBM with immersed body surface mesh approach or eq. (42) for the IBM with immersed body volume mesh approach, the following linear system needs to be solved for each time step:

$$
\begin{bmatrix} M+N+L & P^T & B^T \\ P & 0 & 0 \\ B & 0 & 0 \end{bmatrix}
\begin{pmatrix} \underline{u}^{n+1}\left(\underline{x}\right) \\ \underline{\lambda}^{n+1}\left(\underline{X}\right) \\ p^{n+1}\left(\underline{x}\right) \end{pmatrix}
= \begin{pmatrix} \underline{f}^{n} \\ \underline{h}^{n} \\ 0 \end{pmatrix} + b.c.
\tag{43}
$$

where P is the projection operator from the fluid volume mesh to the immersed body surface/volume mesh, $\underline{h}^n$ and $\underline{\lambda}^{n+1}(\underline{X})$ are the no-slip boundary condition and the Lagrange multipliers at the immersed body surface/volume mesh, respectively, and P^T is the interpolation operator from the immersed body surface/volume mesh to the fluid volume mesh. The other terms in eq. (43) are identical to those defined in eq. (18). The system of eq. (43) is very similar to the system of eq. (18). Algebraically speaking, there is no need to make a distinction between p^{n+1} and $\underline{\lambda}^{n+1}$, and they could be considered as a single vector of Lagrange multipliers (see [54]). For algorithm reasons we kept p^{n+1} and $\underline{\lambda}^{n+1}$ separate in our implementation.

In solving the system of eq. (43), we first consider the subsystem:

$$\begin{bmatrix} A & P^T \\ P & 0 \end{bmatrix} \begin{pmatrix} \underline{u}^{n+1}(\underline{x}) \\ \underline{\lambda}^{n+1}(\underline{X}) \end{pmatrix} = \begin{pmatrix} \underline{f}^n \\ \underline{h}^n \end{pmatrix} + b.c. \tag{44}$$

where $A = M + N + L$. This is equivalent to solve the diffusion step of the projection method described in Section 2.3. At this point we have two options for the Lagrange multipliers of the IBM. Either solve the system for both $\underline{u}^{n+1}$ and $\underline{\lambda}^{n+1}$ at the same time, or find a LU decomposition of eq. (44) and use an approximation of A^{-1}. In the first approach, after applying the boundary conditions to the matrix A, we solve for:

$$\begin{aligned} &A\underline{u}^{n+1}(\underline{x}) + P^T \underline{\lambda}^{n+1}(\underline{X}) = \underline{f}^n \\ &P\underline{u}^{n+1}(\underline{x}) = \underline{h}^n \end{aligned} \tag{45}$$

Multiplying the first of eq. (45) by PA^{-1} we obtain:

$$\begin{aligned} &PA^{-1}A\underline{u}^{n+1}(\underline{x}) + PA^{-1}P^T \underline{\lambda}^{n+1}(\underline{X}) = PA^{-1}\underline{f}^n \\ &P\underline{u}^{n+1}(\underline{x}) = \underline{h}^n \end{aligned} \tag{46}$$

And using the second of eq. (46), we can solve for $\underline{\lambda}^{n+1}$ and $\underline{u}^{n+1}$:

$$\begin{aligned} PA^{-1}P^T\underline{\lambda}^{n+1}(\underline{X}) &= PA^{-1}\underline{f}^n - \underline{h}^n \\ A\underline{u}^{n+1}(\underline{x}) &= \underline{f}^n - P^T\underline{\lambda}^{n+1}(\underline{X}) \end{aligned} \tag{47}$$

We calculate the inverse of A with the Conjugate Gradient iterative method, and use as preconditioner for A UMFPACK in two dimensions and ILU in three dimensions. For the Schur complement $S = PA^{-1}P^T$ we used a very simple and efficient preconditioner $\tilde{S}^{-1} = \left(P\left(diagA\right)^{-1}P^T\right)^{-1}$. With this preconditioner, the Conjugate Gradient method for the first of eq. (47) converged in 5-7 iterations depending on the mesh size. Without preconditioner, the number of iterations required to obtain convergence for the Lagrange multipliers is strongly dependent on mesh size (i.e., becomes larger than 200 iterations for the finest mesh considered).

In the second approach we apply a LU decomposition to the system of eq. (44):

$$\begin{bmatrix} A & 0 \\ P & -PA^{-1}P^T \end{bmatrix}\begin{bmatrix} I & A^{-1}P^T \\ 0 & I \end{bmatrix}\begin{pmatrix} \underline{u}^{n+1}(\underline{x}) \\ \underline{\lambda}^{n+1}(\underline{X}) \end{pmatrix} = \begin{pmatrix} \underline{f}^n \\ \underline{h}^n \end{pmatrix} \tag{48}$$

It is easy to show that the system of eq. (48) is equivalent to eq. (44). Eq. (48) can be written as:

$$\begin{bmatrix} A & 0 \\ P & -PA^{-1}P^T \end{bmatrix}\begin{pmatrix} \underline{u}^*(\underline{x}) \\ \underline{\lambda}^{n+1}(\underline{X}) \end{pmatrix} = \begin{pmatrix} \underline{f}^n \\ \underline{h}^n \end{pmatrix} \tag{49}$$

where $\begin{pmatrix} \underline{u}^*(\underline{x}) \\ \underline{\lambda}^{n+1}(\underline{X}) \end{pmatrix}$ are equal to:

$$\begin{bmatrix} I & A^{-1}P^T \\ 0 & I \end{bmatrix}\begin{pmatrix} \underline{u}^{n+1}(\underline{x}) \\ \underline{\lambda}^{n+1}(\underline{X}) \end{pmatrix} = \begin{pmatrix} \underline{u}^*(\underline{x}) \\ \underline{\lambda}^{n+1}(\underline{X}) \end{pmatrix} \tag{50}$$

In this second approach the diffusion step of the projection method is decomposed in three more sub-steps:

- Step 1: solve for intermediate velocity:

$$A\underline{u}^{*}\left(\underline{x}\right)=\underline{f}^{n} \tag{51}$$

The GMRES iterative method is used for solving eq. (51), with ILU decomposition for the preconditioner.

- Step 2: solve for the Lagrange multipliers with an approximate Schur complement $\tilde{S}=P\tilde{A}^{-1}P^{T}$ using $\tilde{S}^{-1}=\left(P\left(diagA\right)^{-1}P^{T}\right)^{-1}$ as preconditioner:

$$P\tilde{A}^{-1}P^{T}\underline{\lambda}^{n+1}\left(\underline{X}\right)=P\underline{u}^{*}\left(\underline{x}\right)-\underline{h}^{n} \tag{52}$$

The preconditioner calculated in step 1 for the GMRES iterative method is used as $\tilde{A}^{-1}$ in the approximate Schur complement. This completely eliminates the need for calculating the inverse of the matrix A as required in the first approach (computationally very expensive for three-dimensional problems).

- Step 3: project the velocity field:

$$\underline{u}^{n+1}\left(\underline{x}\right)=\underline{u}^{*}\left(\underline{x}\right)-\tilde{A}^{-1}P^{T}\underline{\lambda}^{n+1}\left(\underline{X}\right) \tag{53}$$

The second approach is approximately 20% faster than the first one for 2D simulations, but the difference is expected to be larger for 3D simulations, where the matrix inversion operation becomes very expensive and time consuming. The use of preconditioning in calculating the Lagrange multipliers for both approaches reduces the computational time of an additional 20%.

The velocity $\underline{u}^{n+1}$ satisfies the no-slip constraint at the immersed body surface and inside the immersed body volume, but it is not a divergence-free field. As discussed in the previous section, a projection step on the pseudo-pressure and a correction step on the pressure are still necessary to have a divergence-free velocity vector.

It is worth noting that no discrete delta functions have been used to discretize the projection operator from the fluid volume to the immersed body surface/volume meshes; the same shape functions were used for the fluid volume mesh and immersed body surface/volume mesh. This allowed using adaptive mesh refinement for the fluid volume mesh as well as for the immersed body mesh. The Kelly-Error-Estimator criterion was used for adaptive mesh refinement of the fluid volume mesh. A uniform discretization was used for the immersed body mesh in the analyses performed. The conjugate gradient method was chosen to solve the projection step for both approaches discussed above.

In the case of moving rigid immersed body, the location of the Lagranginan points must be updated at each time step, so the projection operator P and the immersed body no-slip constraint $\underline{h}^n$ must be recomputed at each time step as well.

On the other hand, if stationary immersed body are considered, the Lagrangian points belonging to the IB surface/volume mesh are stationary and correspond to the surface/volume mesh points at every time step (i.e., the immersed body is not moving). The surface force at the immersed body interface can be written as:

$$\underline{\lambda}\left(\underline{x},t\right)=\int_{\Gamma_i(t)}\underline{f}_K\left(\underline{X}\left(s,t\right),t\right)\delta\left(\underline{X}-\underline{x}\right)ds=P^T\underline{\lambda}\left(\underline{X},t\right) \tag{54}$$

where $\underline{X}$ in Γi(t) are not anymore the Lagrangian points, but represent the immersed body surface mesh points.

In the same way the volume forces for the immersed body can be written as:

$$\underline{\lambda}\left(\underline{x},t\right)=\int_{\omega(t)}\underline{f}_K\left(\underline{X}\left(s,t\right),t\right)\delta\left(\underline{X}-\underline{x}\right)ds=P^T\underline{\lambda}\left(\underline{X},t\right) \tag{55}$$

where $\underline{X}$ in $\omega(t)$ are not anymore the Lagrangian points, but represent the immersed body volume mesh points. Following the IBM approach described above, also for stationary rigid bodies it is possible to define projection and interpolation operators. The system of time-dependent incompressible NSEs with IBM can be discretized in the same way as eq. (43) with the only difference that $\underline{X}$ must be reinterpreted as the immersed body surface/volume mesh points.

2.5 NSEs Coupled with the Energy Equation and Boundary Conditions

For flows where the change of density is relatively small, the assumption of incompressible fluid is reasonable as long as the fluid velocity is well below the speed of sound in the medium considered. This is to say that the Mach number $M = V/a << 1$, where V is the fluid velocity and a is the speed of sound.

For conjugate heat transfer problems where small density changes in the fluid associated with temperature gradients are large enough to generate a driving force (i.e., natural convection problems), the Boussinesq approximation for buoyancy can be used.

The Boussinesq approximation states that density differences are sufficiently small to be neglected in the momentum equations, except for the gravity term. The idea behind this assumption is that the effect of density differences can be neglected in the inertia term, but the effect of gravity on density differences across the flow is sufficiently strong to generate a driving force.

Under this assumption, the system of NSEs and the energy equation (EE) can be coupled to become the Boussinesq Equations (BEs):

$$\begin{aligned}
&\rho_0\left(\frac{\partial \underline{U}}{\partial t}+\underline{U}\cdot\nabla\underline{U}\right)=-\nabla P+\mu\nabla\cdot\left(\nabla\underline{U}+\left(\nabla\underline{U}\right)^T\right)+\rho(T)\underline{g}+\underline{F}\\
&\nabla\cdot\underline{U}=0 \qquad\qquad in\ \Omega_f\times]0,T[\\
&\rho_0 c_p\left(\frac{\partial T}{\partial t}+\underline{U}\cdot\nabla T\right)=k\nabla^2 T+Q
\end{aligned} \tag{56}$$

In the first of eq. (56) we explicitly wrote the external force acting on the fluid due to gravity. The second of eq. (56) is the incompressibility constraint, and the third of eq. (56) is the energy equation, where T is the fluid temperature, c_p is the fluid coefficient of specific heat, k is the fluid thermal conductivity, and Q is the volumetric heat source. The implicit assumption in the Boussinesq's equations is that the fluid density is slightly affected by the temperature field, and the density change is so small that needs to be considered for the gravity force only. The fluid density can be expressed as function of the fluid temperature:

$$\rho(T)=\rho_0\left[1-\beta(T-T_0)\right] \tag{57}$$

where T_0 is a reference temperature, ρ_0 is the fluid density at T_0 and β is the coefficient of thermal expansion. If eq. (57) is used into the first of eq. (56):

$$\rho_0\left(\frac{\partial \underline{U}}{\partial t}+\underline{U}\cdot\nabla\underline{U}\right)=-\nabla P+\mu\nabla\cdot\left(\nabla\underline{U}+\left(\nabla\underline{U}\right)^T\right)+\rho_0\left[1-\beta(T-T_0)\right]\underline{g}+\underline{F} \tag{58}$$

Eq. (58) can be rearranged by adding the time-independent term of the gravity force to the pressure term:

$$\rho_0\left(\frac{\partial \underline{U}}{\partial t}+\underline{U}\cdot\nabla\underline{U}\right)=-\nabla P_{dyn}+\mu\nabla\cdot\left(\nabla\underline{U}+\left(\nabla\underline{U}\right)^T\right)-\rho_0\beta T\underline{g}+\underline{F} \tag{59}$$

Where P_{dyn} is the dynamic pressure:

$$P_{dyn} = P - \rho_0 \left(1 + \beta T_0\right) \underline{g} = P + \rho_0 \left(1 + \beta T_0\right) \nabla \varphi \tag{60}$$

In eq. (60) we used the fact that the gravity force results from a gravity potential: $\underline{g} = -grad\varphi$. The system of Boussinesq equations read as:

$$\begin{aligned} &\rho_0 \left(\frac{\partial \underline{U}}{\partial t} + \underline{U} \cdot \nabla \underline{U} \right) = -\nabla P_{dyn} + \mu \nabla \cdot \left(\nabla \underline{U} + \left(\nabla \underline{U} \right)^T \right) - \rho_0 \beta T \underline{g} + \underline{F} \\ &\nabla \cdot \underline{U} = 0 \qquad\qquad in\ \Omega_f \times]0, T[\\ &\rho_0 c_p \left(\frac{\partial T}{\partial t} + \underline{U} \cdot \nabla T \right) = k \nabla^2 T + Q \end{aligned} \tag{61}$$

By using a reference velocity U_0, a reference length L and the reference temperature difference $\Delta T = T_{max} - T_0$, we can write the system of eq. (61) in non-dimensional form:

$$\begin{aligned} &\frac{\partial \underline{u}}{\partial t} + \underline{u} \cdot \nabla \underline{u} = -\nabla p + \frac{\mu}{\rho_0 U_0 L} \nabla \cdot \left(\nabla \underline{u} + \left(\nabla \underline{u} \right)^T \right) - \frac{\rho_0 g \beta \Delta T}{\rho_0 U_0^2 / L} \theta \underline{g}^* + \frac{\underline{F}}{\rho_0 U_0^2 / L} \\ &\nabla \cdot \underline{u} = 0 \\ &\frac{\partial \theta}{\partial t} + \underline{u} \cdot \nabla \theta = \frac{L}{\rho_0 c_p U_0 \Delta T} \cdot \frac{k \Delta T}{L^2} \nabla^2 \theta + \frac{Q}{\rho_0 c_p U_0 \Delta T / L} \end{aligned} \tag{62}$$

where g^* is the non-dimensional gravity vector field and $p = P_{dyn}/\rho_0 U_0^2$ is the non-dimensional pressure field. Beside the Reynolds number $Re = \rho_0 U_0 L/\mu$ introduced in eq. (3), two new non-dimensional numbers are introduced: the Grashof number Gr and the Peclet number Pe:

$$Gr = \frac{\rho_0^2 g \beta \Delta T L^3}{\mu^2}; Pe = \mathrm{Re} \cdot \mathrm{Pr} = \frac{\rho_0 U_0 L}{\mu} \cdot \frac{\mu c_p}{k} \tag{63}$$

where Pr is the Prandlt number $Pr = \mu c_p/k$. Using the definition of Gr and Re non-dimensional numbers the first of eq. (62) can be written as:

$$\frac{\partial \underline{u}}{\partial t}+\underline{u}\cdot\nabla\underline{u}=-\nabla p+\frac{\mu}{\rho_0 U_0 L}\nabla\cdot\left(\nabla\underline{u}+\left(\nabla\underline{u}\right)^T\right)-\frac{\rho_0^2 g\beta\Delta T L^3}{\mu^2}\frac{\mu^2}{\rho_0^2 U_0^2 L^2}\theta\underline{g}^*+\frac{\underline{F}}{\rho_0 U_0^2/L}=$$
$$-\nabla p+\frac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u}+\left(\nabla\underline{u}\right)^T\right)-\frac{Gr}{\mathrm{Re}^2}\theta\underline{g}^*+\underline{f} \tag{64}$$

where $\underline{f}$ is the non-dimensional volumetric force vector. By using the definition of the *Pe* non-dimensional number the third of eq. (62) can be written as:

$$\frac{\partial\theta}{\partial t}+\underline{u}\cdot\nabla\theta=\frac{L}{\rho_0 c_p U_0\Delta T}\cdot\frac{k\Delta T}{L^2}\nabla^2\theta+\frac{Q}{\rho_0 c_p U_0\Delta T/L}=$$
$$=\frac{\mu}{\rho_0 U_0 L}\cdot\frac{k}{\mu c_p}\nabla^2\theta+q=\frac{1}{Pe}\nabla^2\theta+q \tag{65}$$

where q is the normalized volumetric heat source. The system of Boussinesq equations in non-dimensional form reads as:

$$\begin{aligned}&\frac{\partial\underline{u}}{\partial t}+\underline{u}\cdot\nabla\underline{u}=-\nabla p+\frac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u}+\left(\nabla\underline{u}\right)^T\right)-\frac{Gr}{\mathrm{Re}^2}\theta\underline{g}^*+\underline{f}\\&\nabla\cdot\underline{u}=0 \qquad\qquad in\ \Omega_f\times]0,T[\\&\frac{\partial\theta}{\partial t}+\underline{u}\cdot\nabla\theta=\frac{1}{Pe}\nabla^2\theta+q\end{aligned} \tag{66}$$

The initial and boundary conditions for the NSEs are described in Section 2.1. The energy equation requires both initial and boundary conditions. The initial condition is of the form:

$$\theta\big|_{t=0}=\theta_0 \qquad in\ \Omega_f \tag{67}$$

and the boundary conditions can be either of Dirichlet-type or Neumann-type:

$$\theta=\theta_D\left(\underline{x}\right),\ \ for\ \underline{x}\in\Gamma_D \tag{68}$$

$$\underline{n}\cdot k\nabla\theta\big|_{\Gamma_N}=\phi \tag{69}$$

where θ_D is the value of the non-dimensional temperature at Dirichlet boundary Γ_D, and ϕ is the prescribed non-dimensional thermal flux at Neumann boundary Γ_N. For adiabatic boundaries $\phi = 0$. Since the temperature is assumed known at the interface fluid/solid region for the simulations performed, Dirichlet boundary conditions would be imposed at the immersed boundary interface Γ_i. We are using the IBM to determine the effect of the solid body on the fluid temperature distribution. Γ_i is not directly represented by the finite element discretization. Section 2.4 discusses the immersed boundary formulation used to enforce no-slip boundary conditions on the immersed surface. In Section 2.5.2 we will discuss the immersed boundary formulation used to enforce temperature boundary conditions at the solid body surface.

2.5.1 Finite Element Formulation of the Boussinesq Equations

For sake of brevity in the discussion of the finite element formulation for the Boussinesq equations we will illustrate only the differences with respect to the finite element formulation for the NSEs presented in Section 2.2.

Writing the system of Boussinesq equations [eq. (66)] in vector form:

$$\begin{pmatrix} \dfrac{\partial \underline{u}}{\partial t} + \underline{u}\cdot\nabla\underline{u} + \nabla p - \dfrac{1}{\mathrm{Re}}\nabla\cdot\left(\nabla\underline{u} + \left(\nabla\underline{u}\right)^T\right) + \dfrac{Gr}{\mathrm{Re}^2}\theta\underline{g}^* \\ -\nabla\cdot\underline{u} \\ \dfrac{\partial\theta}{\partial t} + \underline{u}\cdot\nabla\theta - \dfrac{1}{Pe}\nabla^2\theta \end{pmatrix} = \begin{pmatrix} \underline{f} \\ 0 \\ q \end{pmatrix} \tag{70}$$

The weak form for the NSEs with the buoyancy term is obtained by forming the dot product from the left with a vector-valued test function $\Phi = (\underline{v}, q)$, and integrating over the entire domain Ω_f. The weak form reads as:

$$\left(\underline{v},\frac{\partial \underline{u}}{\partial t}\right)_{\Omega_f}+\left(\underline{v},\underline{u}\cdot\nabla\underline{u}\right)_{\Omega_f}-\frac{1}{\mathrm{Re}}\left(\underline{v},\nabla\cdot\left(\nabla\underline{u}+\left(\nabla\underline{u}\right)^T\right)\right)_{\Omega_f}+\frac{Gr}{\mathrm{Re}^2}(\underline{v},\theta\underline{g}^*)$$
$$\left(\underline{v},\nabla p\right)_{\Omega_f}-\left(q,\nabla\cdot\underline{u}\right)_{\Omega_f}=\left(\underline{v},\underline{f}\right)_{\Omega_f} \tag{71}$$

A more interesting question is what to do with the energy equation. By default not all discretization of this equation are equally stable unless we either use upwinding, stabilization or a combination of both. One way to achieve this is to use discontinuous elements and to define a flux at the interface that takes into account upwinding. Discontinuous elements have the drawback that the use of numerical fluxes introduces an additional numerical diffusion on the entire solution field. Our goal is to keep the numerical diffusion to a minimum and apply it where it is necessary to stabilize the scheme. A different approach is to add some nonlinear (i.e., "artificial") viscosity to the diffusion term in the third of eq. (70), which becomes:

$$\frac{\partial\theta}{\partial t}+\underline{u}\cdot\nabla\theta-\frac{1}{Pe}\nabla\cdot(1+\frac{\eta(\theta)}{k})\nabla\theta=q \tag{72}$$

where $\eta(\theta)$ is the additional viscosity that only acts in the vicinity of shocks and other discontinuities. The value of the "artificial" viscosity is determined on the residual of the energy equation. Due to the relatively low Gr/Re^2 number chosen for the simulations performed, it was found not necessary to turn on the stability model implemented. Further details on this model are left out from the discussion for sake of brevity.

For the discretization in time of the energy equation we used the backward differentiation formula of second order (BDF2) to be consistent with the discretization in time of the momentum equations. The time derivative becomes:

$$\left.\frac{\partial\theta}{\partial t}\right|_{n+1} = \frac{\frac{3}{2}\theta^{n+1} - 2\theta^{n} + \frac{1}{2}\theta^{n-1}}{\Delta t} \tag{73}$$

The energy equation discretized in time with the artificial viscosity model turned off then becomes:

$$\frac{3}{2}\theta^{n+1} - \frac{\Delta t}{Pe}\nabla^2\theta^{n+1} = 2\theta^{n} - \frac{1}{2}\theta^{n-1} - \Delta t(2\underline{u}^{n} - \underline{u}^{n-1})\cdot\nabla(2\theta^{n} - \theta^{n-1}) + \Delta t\cdot q \tag{74}$$

where the diffusion term is treated in a fully implicit way and the advection term is treated in a fully explicit way using a forward extrapolation (projection) of velocity and temperature. This projection maintains the second order accuracy of the BDF2 scheme.

The weak form of the energy equation is obtained by forming the dot product from the left of eq. (74) with a scalar test function τ, and integrating over the entire domain Ω_f. The weak form of the energy equation reads as:

$$\begin{aligned}\left(\tau, \frac{3}{2}\theta^{n+1}\right)_{\Omega_f} + \frac{\Delta t}{Pe}\left(\nabla\tau, \nabla\theta^{n+1}\right)_{\Omega_f} &= \left(\tau, \left(2\theta^{n} - \frac{1}{2}\theta^{n-1}\right)\right)_{\Omega_f} \\ \Delta t\left(\tau, \left(q - (2\underline{u}^{n} - \underline{u}^{n-1})\cdot\nabla(2\theta^{n} - \theta^{n-1})\right)\right)_{\Omega_f} &+ \frac{\Delta t}{Pe}\left\langle\tau, \underline{n}\cdot\nabla\theta^{n+1}\right\rangle_{\Gamma_N}\end{aligned} \tag{75}$$

where $(\bullet,\bullet)_\Omega$ represents a volume integral over Ω, and the heat flux at Neumann boundaries is $\phi = 0$. Eq. (75) has to hold for all scalar test function τ. The discretization of the weak form for the time-dependent incompressible NSEs with buoyancy term and Dirichlet/Neumann boundary conditions eq. (71) reads:

$$
\begin{aligned}
&find\ \underline{u} \in V_g = \left\{ \varphi \in H^1(\Omega)^d : \varphi_{\Gamma_D} = \underline{u}_D \right\},\ p \in Q = L^2(\Omega)\ s.t. \\
&\left(\underline{v}, \frac{\partial \underline{u}}{\partial t} \right)_\Omega + \left(\underline{v}, \underline{u} \cdot \nabla \underline{u} \right)_\Omega + \frac{2}{\mathrm{Re}} \left(\nabla \underline{v}, \underline{\underline{\varepsilon}}(\underline{u}) \right)_\Omega + \frac{Gr}{\mathrm{Re}^2} (\underline{v}, \theta \underline{g}^*) - \\
&\qquad \left(\nabla \cdot \underline{v}, p \right)_\Omega - \left(q, \nabla \cdot \underline{u} \right)_\Omega = \left(\underline{v}, \underline{f} \right)_\Omega - \left\langle \underline{v}, \underline{t}(\underline{x}) \right\rangle_{\Gamma_N} \\
&\forall\ \underline{v} \in V_g = \left\{ \varphi \in H^1(\Omega)^d : \varphi_{\Gamma_D} = 0 \right\}, q \in Q
\end{aligned}
\tag{76}
$$

where for generality we used Ω referring to the fluid region Ω_f. The same stability issues discussed in Section 2.2 apply to eq. (76). The discretization of the weak form for the energy equation eq. (75) reads:

$$
\begin{aligned}
&find\ \theta \in T_g = \left\{ \varphi \in H^1(\Omega)^d : \varphi_{\Gamma_D} = \theta_D \right\}\ s.t. \\
&\left(\tau, \frac{3}{2}\theta^{n+1} \right)_\Omega + \frac{\Delta t}{Pe} \left(\nabla \tau, \nabla \theta^{n+1} \right)_\Omega = \left(\tau, \left(2\theta^n - \frac{1}{2}\theta^{n-1} \right) \right)_\Omega \\
&\qquad \Delta t \left(\tau, \left(q - (2\underline{u}^n - \underline{u}^{n-1}) \cdot \nabla (2\theta^n - \theta^{n-1}) \right) \right)_\Omega + \frac{\Delta t}{Pe} \left\langle \tau, \underline{n} \cdot \nabla \theta^{n+1} \right\rangle_{\Gamma_N} \\
&\forall\ \tau \in T_g = \left\{ \varphi \in H^1(\Omega)^d : \varphi_{\Gamma_D} = 0 \right\}
\end{aligned}
\tag{77}
$$

The continuous functions chosen satisfy the required conditions. When we discretize the equations by replacing the continuous variables and test functions by finite element functions (i.e., polynomials of order n) in finite dimensional spaces $V_{g,h} \subset V_g, Q_h \subset Q, T_{g,h} \subset T_g$, we have to make sure that also $V_{g,h}$, Q_h and T_h satisfy the LBB conditions. For the time-dependent incompressible NSEs a number of possible choices is available which will respect the LBB conditions. A simple and accurate choice used for the present work is $\underline{u}_h \in Q_{p+1}^d, p_h \in Q_p^d, \tau_h \in Q_{p+1}^d$ i.e., to use elements one order higher for the velocity vector and temperature field than for the pressure field. Then the discretized system of Boussinesq equations reads as:

$$
\begin{aligned}
&\text{find } \underline{u}_h \in Q_{p+1}^d,\ p_h \in Q_p^d,\ \tau_h \in Q_{p+1}^d \text{ s.t.} \\
&\left(\underline{v}_h, \frac{\partial \underline{u}_h}{\partial t}\right)_\Omega + \left(\underline{v}_h, \underline{u}_h \cdot \nabla \underline{u}_h\right)_\Omega + \frac{2}{\mathrm{Re}}\left(\nabla \underline{v}_h, \underline{\underline{\varepsilon}}\left(\underline{u}_h\right)\right)_\Omega + \frac{Gr}{\mathrm{Re}^2}\left(\underline{v}_h, \theta_h \underline{g}^*\right) \\
&\qquad -\left(\nabla \cdot \underline{v}_h, p_h\right)_\Omega - \left(q_h, \nabla \cdot \underline{u}_h\right)_\Omega = \left(\underline{v}_h, \underline{f}\right)_\Omega - \left\langle \underline{v}_h, \underline{t}\left(\underline{x}\right)\right\rangle_{\Gamma_N} \\
&\left(\tau_h, \frac{3}{2}\theta_h^{n+1}\right)_\Omega + \frac{\Delta t}{Pe}\left(\nabla \tau_h, \nabla \theta_h^{n+1}\right)_\Omega = \left(\tau_h, \left(2\theta_h^n - \frac{1}{2}\theta_h^{n-1}\right)\right)_\Omega \\
&\qquad \Delta t\left(\tau_h, \left(q - (2\underline{u}_h^n - \underline{u}_h^{n-1}) \cdot \nabla (2\theta_h^n - \theta_h^{n-1})\right)\right)_\Omega + \frac{\Delta t}{Pe}\left\langle \tau_h, \underline{n} \cdot \nabla \theta_h^{n+1}\right\rangle_{\Gamma_N} \\
&\forall\, \underline{v}_h \in V_{g,h}, q_h \in Q_{g,h}, \tau_h \in T_{g,h}
\end{aligned}
\tag{78}
$$

We use the fractional step method described in Section 2.3 to solve for the NSEs. The buoyancy term is calculated explicitly using the temperature from the previous time step. Writing the discretized form of the energy equation in a matrix notation we obtain:

$$\left(M_T + L_T\right)\theta_h^{n+1} = f_T \tag{79}$$

where M_T and L_T are the mass matrix and the thermal diffusive term and f_T is the source term function of velocity and temperature at the previous time steps and the volumetric heat source. The solution of eq. (79) is straightforward. We invert the matrix $A_T = M_T + L_T$ with the conjugate gradient algorithm using a simple preconditioner, and solve for:

$$\theta_h^{n+1} = A_T^{-1} f_T \tag{80}$$

2.5.2 Immersed Boundary Method for the Boussinesq Equations

The application of the immersed boundary method to the velocity and pressure fields of the Boussinesq equations follows the procedure outlined in Section 2.4. The

presence of the buoyancy term in the momentum equations does not alter the immersed boundary projection method solution procedure described in detail in Section 2.4.3.

In this section we focus on the application of the immersed boundary method to the energy equation with the surface and volume boundary force approaches. For the IBM surface approach the set of Lagrangian points ($\underline{X}_K$) used to represent the surface $\Gamma_i(t)$ of the immersed body at time t within the computational domain is the same as that used in the NSEs. At the Lagrangian points, approximate heat sources forcing terms q_K are applied to enforce the temperature Dirichlet boundary condition along $\Gamma_i(t)$. The approach is exactly the same as for the no-slip boundary constraint in the momentum equations applied to the fluid/solid interface. In general, the location of the Lagrangian boundary points does not coincide with the underlying fluid mesh discretization. Therefore, it is necessary to build a projection operator P from the fluid volume mesh to the immersed body surface mesh and an interpolation operator P^T from the surface mesh to the fluid volume mesh. It is worth mentioning that, the projection operator for the temperature field is different from the one used for the velocity field even if the same fluid mesh and immersed body mesh are used, since the latter field uses vector-valued test function and the temperature field uses scalar test functions.

Writing the energy equation with immersed boundary method, it is possible to have a discretized form of the projection operator, which consists of interpolating the shape functions used for the fluid volume mesh on the immersed boundary surface mesh. This can be easily handled in finite element method framework by the *deal.II* library chosen [69] to perform the numerical simulations.

The energy equation with a heat source forcing term q_K can be considered as the continuous analog of the IBM:

$$\begin{aligned} &\frac{\partial \theta}{\partial t} + \underline{u} \cdot \nabla \theta = \frac{1}{Pe} \nabla^2 \theta + q - \int_{\Gamma_i(t)} q_K \left(\underline{X}(s,t) \right) \delta \left(\underline{X} - \underline{x} \right) ds \qquad in \, \Omega \times]0,T[\\ &\theta \left(\underline{X}(s,t) \right) = \int_{\Gamma_i(t)} \theta(\underline{x},t) \delta \left(\underline{x} - \underline{X} \right) d\underline{x} = \theta_{IB} \left(\underline{X}(s,t) \right) \qquad on \, \Gamma_i(t) \end{aligned} \tag{81}$$

where we follow the same notation as in Section 2.4 and $\underline{x}$ in Ω (volume mesh), $\underline{X}(s,t)$ in $\Gamma_i(t)$ (surface mesh) is the position of a point on the immersed boundary surface indexed by a position s in the reference configuration at time t. The last term on the rhs of the first in eq. (81) represents the effect of the immersed body on the fluid temperature field, that is the integral of the heat source forcing terms q_K applied at the fluid/solid interface $\Gamma_i(t)$. By forming the dot product from left of eq. (81) with a scalar test function $\Phi = (\tau, \eta)$, the weak form of the system of eq. (81) is obtained:

$$\begin{aligned} &\left(\tau, \frac{\partial \theta}{\partial t} \right)_\Omega + \left(\tau, \underline{u} \cdot \nabla \theta \right)_\Omega - \frac{1}{Pe} \left(\tau, \nabla^2 \theta \right)_\Omega = \left(\tau, q \right)_\Omega \\ &\qquad - \left(\tau, \int_{\Gamma_i(t)} q_K \left(\underline{X}(s,t) \right) \delta \left(\underline{X} - \underline{x} \right) ds \right)_\Omega \qquad in \ \Omega \times]0,T[\\ &\left\langle \eta, \theta \left(\underline{X}(s,t) \right) \right\rangle_{\Gamma_i(t)} = \left\langle \eta, \int_\Omega \theta(\underline{x},t) \delta \left(\underline{x} - \underline{X} \right) d\underline{x} \right\rangle_{\Gamma_i(t)} = \\ &\qquad \left\langle \eta, \theta_{IB} \left(\underline{X}(s,t) \right) \right\rangle_{\Gamma_i(t)} \qquad on \ \Gamma_i(t) \end{aligned} \tag{82}$$

Eq.(82) has to hold $\forall \ \Phi = (\tau, \eta)$. The last term in the first of eq. (82) can be written as:

$$\left(\tau, \int_{\Gamma_i(t)} q_K\left(\underline{X}(s,t)\right)\delta\left(\underline{X}-\underline{x}\right)ds\right)_{\Omega} = \int_{\Omega}\tau\int_{\Gamma_i(t)} q_K\left(\underline{X}(s,t)\right)\delta\left(\underline{X}-\underline{x}\right)ds\,d\underline{x} = \int_{\Omega}\tau\left(\underline{x}\right)\lambda_T\left(\underline{x},t\right)d\underline{x} \tag{83}$$

where $\lambda_T(\underline{x},t)$ are the projection of the Lagrange multipliers due to the heat source forcing terms present at the immersed body interface from the surface mesh to the fluid volume mesh:

$$\lambda_T\left(\underline{x},t\right) = \int_{\Gamma_i(t)} q_K\left(\underline{X}(s,t)\right)\delta\left(\underline{X}-\underline{x}\right)ds = P^T\lambda_T\left(\underline{X}(s,t)\right) \tag{84}$$

and P^T represents the interpolation operator (immersed body surface mesh → fluid volume mesh) for the temperature scalar test and shape functions. In the same way, the second of eq. (82) can be written as:

$$\left\langle \eta, \int_{\Omega}\theta\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}\right\rangle_{\Gamma_i(t)} = \int_{\Gamma_i(t)}\eta\int_{\Omega}\theta\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x}ds = \int_{\Gamma_i(t)}\eta\theta\left(\underline{X}(s,t)\right)ds = \left\langle \eta, \theta_{IB}\left(\underline{X}(s,t)\right)\right\rangle_{\Gamma_i(t)} \tag{85}$$

where $\theta(\underline{X}(s,t)$ represents the temperature field projected on the immersed body mesh:

$$\theta\left(\underline{X}(s,t)\right) = \int_{\Omega}\theta\left(\underline{x},t\right)\delta\left(\underline{x}-\underline{X}\right)d\underline{x} = P\theta\left(\underline{x},t\right) \tag{86}$$

After applying the backward differentiation formula of second order (BDF2) for the discretization in time and integrating by parts the second term on the lhs of eq. (82) (for more details refer to Section 2.5.1), the discretization of the weak form for the energy equation with Dirichlet and Neumann boundary conditions and IBM reads:

$$
\begin{aligned}
&find\ \theta \in T_g = \left\{ \phi \in H^1(\Omega)^d : \varphi_{\Gamma_D} = \theta_D \right\}, \lambda_T \in \Lambda_g = \left\{ \eta \in H^1(\Omega)^{d-1} \right\} s.t \\
&\left(\tau, \frac{3}{2}\theta^{n+1} \right)_\Omega + \frac{\Delta t}{Pe}\left(\nabla\tau, \nabla\theta^{n+1} \right)_\Omega = \left(\tau, \left(2\theta^n - \frac{1}{2}\theta^{n-1} \right) \right)_\Omega - \Delta t\left(\tau, P^T \underline{\lambda}_T^{n+1}\left(\underline{X} \right) \right)_\Omega \\
&\qquad + \Delta t\left(\tau, q \right)_\Omega - \Delta t\left(\tau, (2\underline{u}^n - \underline{u}^{n-1}) \cdot \nabla(2\theta^n - \theta^{n-1}) \right)_\Omega + \frac{\Delta t}{Pe}\left\langle \tau, \underline{n} \cdot \nabla\theta^{n+1} \right\rangle_{\Gamma_N} \qquad (87) \\
&\left\langle \eta, P\theta^{n+1}\left(\underline{x} \right) \right\rangle_{\Gamma_i(t)} = \left\langle \eta, \theta_{IB}^{n+1}\left(\underline{X}\left(s \right) \right) \right\rangle_{\Gamma_i(t)} = \left\langle \eta, P\theta_{IB}^{n+1}\left(\underline{x} \right) \right\rangle_{\Gamma_i(t)} \\
&\forall \tau \in T_g = \left\{ \phi \in H^1(\Omega)^d : \varphi_{\Gamma_D} = 0 \right\}, \forall \eta \in \Lambda_g = \left\{ \eta \in H^1(\Omega)^{d-1} \right\}
\end{aligned}
$$

where strongly imposed boundary conditions (i.e., Dirichlet-type) do not appear in the weak form, whereas Neumann-type boundary conditions are still present.

The solution for the temperature and Lagrangian multipliers fields is found in the function spaces $\theta \in T_g = \{\Phi \in H^1(\Omega)^d : \Phi_{\Gamma D} = \theta_D\}$, $\underline{\lambda}_T \in \Lambda_g = \{\eta \in H^1(\Omega)^{d-1}\}$. When we discretize the NSEs with IBM by replacing the continuous variables and test functions by finite element functions in finite dimensional spaces $T_{g,h} \subset T_g$, $\Lambda_{g,h} \subset \Lambda_g$, we choose elements for the temperature field of the same order as those chosen for the velocity vector. Discretizing the energy equation with IBM and replacing the continuous variables and test functions by finite element functions in finite dimensional spaces we obtain:

$$
\begin{aligned}
&find\ \tau_h \in Q_{p+1}^d, \lambda_{T,h} \in Q_p^{d-1}\ s.t. \\
&\left(\tau_h, \frac{3}{2}\theta_h^{n+1} \right)_\Omega + \frac{\Delta t}{Pe}\left(\nabla\tau_h, \nabla\theta_h^{n+1} \right)_\Omega = \left(\tau_h, \left(2\theta_h^n - \frac{1}{2}\theta_h^{n-1} \right) \right)_\Omega - \Delta t\left(\tau_h, \lambda_{T,h}^{n+1}\left(\underline{x} \right) \right)_\Omega \\
&\qquad + \Delta t\left(\tau_h, q \right)_\Omega - \Delta t\left(\tau_h, (2\underline{u}_h^n - \underline{u}_h^{n-1}) \cdot \nabla(2\theta_h^n - \theta_h^{n-1}) \right)_\Omega + \frac{\Delta t}{Pe}\left\langle \tau_h, \underline{n} \cdot \nabla\theta_h^{n+1} \right\rangle_{\Gamma_N} \qquad (88) \\
&\left\langle \eta_h, \theta_h^{n+1}\left(\underline{X} \right) \right\rangle_{\Gamma_i(t)} = \left\langle \eta_h, \theta_{IB}^{n+1}\left(\underline{X} \right) \right\rangle_{\Gamma_i(t)} \\
&\forall \tau_h \in V_{g,h}, \eta_h \in \Lambda_{g,h}
\end{aligned}
$$

The immersed body can be also discretized with a volume mesh overlapping the fluid volume mesh. In this case, the IBM introduces a set of Lagrangian points ($\underline{X}_K$) representing the immersed object $\omega(t)$. The fluid volume mesh does not need to conform to the body shape (i.e., non-body fitting mesh). At the Lagrangian points, approximate heat source forcing terms q_K are applied to enforce the Dirichlet boundary condition along the fluid/solid body interface Γ_i and inside the immersed body volume $\omega(t)$. Since, in general, the location of the Lagrangian boundary points does not coincide with the underlying fluid mesh discretization, it is necessary to build a projection operator P from the fluid volume mesh to the immersed body volume mesh and an interpolation operator P^T from the immersed boundary volume mesh to the fluid volume mesh. The approach used is analogous to that used for the IBM surface mesh, where the discretized form of the projection operator is obtained interpolating the shape functions used for the fluid volume mesh on the immersed boundary volume mesh. The *deal.II* library [69] was used to perform the numerical simulations. In Section 2.4.2 were outlined the differences between the immersed body surface mesh and volume mesh approaches when applied to the time-dependent incompressible NSEs. Therefore, we will provide here the final result of the discretized energy equation with IBM when finite element functions in finite dimensional spaces replace the continuous variables and test functions.

$$
\begin{aligned}
&\text{find } \tau_h \in Q_{p+1}^d, \lambda_{T,h} \in Q_p^{d-1} \text{ s.t.} \\
&\left(\tau_h, \frac{3}{2}\theta_h^{n+1}\right)_\Omega + \frac{\Delta t}{Pe}\left(\nabla\tau_h, \nabla\theta_h^{n+1}\right)_\Omega = \left(\tau_h, \left(2\theta_h^n - \frac{1}{2}\theta_h^{n-1}\right)\right)_\Omega - \Delta t\left(\tau_h, \lambda_{T,h}^{n+1}\left(\underline{x}\right)\right)_\Omega \\
&\qquad + \Delta t\left(\tau_h, q\right)_\Omega - \Delta t\left(\tau_h, (2\underline{u}_h^n - \underline{u}_h^{n-1})\cdot\nabla(2\theta_h^n - \theta_h^{n-1})\right)_\Omega + \frac{\Delta t}{Pe}\left\langle\tau_h, \underline{n}\cdot\nabla\theta_h^{n+1}\right\rangle_{\Gamma_N} \qquad (89) \\
&\left(\eta_h, \theta_h^{n+1}\left(\underline{X}\right)\right)_{\omega_i(t)} = \left(\eta_h, \theta_{IB}^{n+1}\left(\underline{X}\right)\right)_{\omega_i(t)} \\
&\forall\, \tau_h \in V_{g,h}, \eta_h \in \Lambda_{g,h}
\end{aligned}
$$

2.5.3 The Discretized Boussinesq Equations with Immersed Boundary Projection Method

Since we have decoupled the NESs from the energy equation by calculating the buoyancy term in the momentum equations in an explicit way (i.e., the temperature from the previous time step is used in eq. (76), the solution of the NSEs with immersed boundary method is the same as the one outlined in Section 2.4.3 with the only difference that the buoyancy term is added to the external force $\underline{f}$ in eq. (43).

If we write the discretized energy equation with immersed body surface mesh approach eq. (88) or immersed body volume mesh approach eq (89) in matrix form, the following linear system needs to be solved for each time step:

$$
\begin{bmatrix} M_T + L_T & P^T \\ P & 0 \end{bmatrix}\begin{pmatrix} \theta^{n+1}\left(\underline{x}\right) \\ \lambda_T^{n+1}\left(\underline{X}\right) \end{pmatrix} = \begin{pmatrix} q^n \\ h_T^n \end{pmatrix} + b.c. \qquad (90)
$$

where P is the projection operator from the fluid volume mesh to the immersed body surface/volume mesh, $h_T{}^n$ and $\lambda_T{}^{n+1}(\underline{X})$ are the Dirichlet boundary condition for temperature at the fluid/solid interface and the Lagrange multipliers at the immersed body surface/volume mesh, respectively, and P^T is the interpolation operator from the

immersed body surface/volume mesh to the fluid volume mesh. To simplify the notation in eq. (90) we added all the known terms on the rhs of the energy equation in the q^n term, that is:

$$q^n = \left(\tau_h, \left(2\theta_h^n - \frac{1}{2}\theta_h^{n-1} \right) \right)_\Omega + \Delta t \left(\tau_h, q \right)_\Omega - \Delta t \left(\tau_h, (2\underline{u}_h^n - \underline{u}_h^{n-1}) \cdot \nabla (2\theta_h^n - \theta_h^{n-1}) \right)_\Omega \tag{91}$$

The other terms in eq. (90) are identical to those defined in eq. (79). The system of eq. (90) is very similar to the system of eq. (44), which is the system of momentum equations coupled with the Lagrange multipliers for the immersed boundary and with the pressure term calculated explicitly from the previous time step. This is equivalent to solve the diffusion step of the projection method described in Section 2.3. We solve the system of eq. (90) for both θ^{n+1} and λ_T^{n+1} at the same time. After applying the boundary conditions to the matrix $A_T = M_T + L_T$, we solve for:

$$\begin{aligned} &A_T \theta^{n+1}(\underline{x}) + P^T \lambda_T^{n+1}(\underline{X}) = q^n \\ &P\theta^{n+1}(\underline{x}) = h_T^n \end{aligned} \tag{92}$$

Multiplying the first of eq. (92) by PA_T^{-1} we obtain:

$$\begin{aligned} &PA_T^{-1}A_T\theta^{n+1}(\underline{x}) + PA_T^{-1}P^T\lambda_T^{n+1}(\underline{X}) = PA_T^{-1}q^n \\ &P\theta^{n+1}(\underline{x}) = h_T^n \end{aligned} \tag{93}$$

And using the second of eq. (93), we can solve for λ_T^{n+1} and θ^{n+1}:

$$\begin{aligned} &PA_T^{-1}P^T\lambda_T^{n+1}(\underline{X}) = PA_T^{-1}q^n - h_T^n \\ &A_T\theta^{n+1}(\underline{x}) = q^n - P^T\lambda_T^{n+1}(\underline{X}) \end{aligned} \tag{94}$$

We calculate the inverse of A_T with the Conjugate Gradient iterative method, and use as preconditioner for A_T UMFPACK in two dimensions and ILU in three dimensions. For the Schur complement $S = PA_T^{-1}P^T$ we used a very simple and efficient

preconditioner $\tilde{S}^{-1} = \left(P \left(diag A_T \right)^{-1} P^T \right)^{-1}$. With this preconditioner, the Conjugate Gradient method for the first of eq. (94) converged in a few iterations.

2.6 Immersed Boundary Methods Comparison

The immersed boundary formulation discussed in the previous section will be compared to other IBMs. We choose the original IBM (see [1]), the direct-forcing approach (see [60, 70]), the Immersed Interface Method (IIM) (see [23, 26]), and the Distributed Lagrange Multipliers (DLM) (see [36]). A more extensive review of the IBMs is presented in [52, 53]. We briefly discuss the IBM formulation of the methods addressed above, omitting the details for clarity of discussion.

2.6.1 The Original Immersed Boundary Method

The original IBM (see [1]) can be seen as a modification of the traditional fractional step method, where an explicit boundary force is used to take into account the effect of the immersed body:

$$\begin{aligned} A\underline{u}^* &= \underline{F}^n - P^T \underline{\lambda}^n + bc_1 \\ BA^{-1}B^T p &= B\underline{u}^* + bc_2 \\ \underline{u}^{n+1} &= \underline{u}^* - A^{-1}B^T p \end{aligned} \tag{95}$$

and the force applied at the immersed body surface is calculated explicitly with Hooke's law. Source/sink can be applied to the Poisson equation to correct the continuity equation (see [64]). To determine the force at the immersed body surface, also a feedback controller has been used (see [5]). As discussed in the introduction, to simulate

rigid immersed bodies, large gains are required. The system of equations becomes very stiff, prohibiting the use of larger CFL numbers. Lowering the gains to have larger CFL would result in a relaxation of the no-slip boundary constraint. The proposed fractional step/immersed boundary method solves for the boundary force implicitly with no constitutive laws. It is worth to point out that the method implemented satisfies the continuity equation and the no-slip condition to round-off error.

2.6.2 The Direct-Forcing Method

In the direct-forcing method (see [60]) the boundary force for rigid bodies is approximated with an intermediate velocity field $\underline{u}^*$. The force is implemented directly into the momentum equation by substituting the regularized no-slip condition near the immersed boundary. The momentum equation is modified with an interpolation operator to extract the velocity field near the immersed body surface. The difference between the modified momentum equation and the momentum equation from the traditional fractional step method represents the boundary force for the direct-forcing method. This method enforces the no-slip condition on $\underline{u}^*$ and not on $\underline{u}^{n+1}$, therefore, slip at the immersed surface will be present.

2.6.3 The Immersed Interface Method (IIM)

In the IIM (see [23, 26]) the boundary force is decomposed into tangential and normal components. The tangential component is included in the momentum equation as an explicit term, while the explicit normal component is included into the pressure

Poisson equation in terms of a pressure jump condition across the interface. This will generate a sharp velocity solution in the vicinity of the immersed body surface. Linnick et al. [73] developed a high order IIM employing one-sided finite differences to obtain jump conditions for higher-order derivatives.

2.6.4 The Distributed Lagrange Multiplier (DLM) Method

Glowinski et al. [36] introduce Lagrange multipliers on the immersed rigid body to satisfy the no-slip boundary conditions through a projection operator for a variational principle (i.e., finite element) approach. In our formulation we use the second order pressure-correction scheme (see Section 2.3), while Glowinski et al. used for the DLM the Marchuk-Yanenko fractional step scheme (see [74, 75]), which is composed of three operations: (i) the divergence-free condition and pressure; (ii) the convective and diffusive operators; (iii) the no-slip condition and boundary force.

3. FINITE VOLUME IMMERSED BOUNDARY METHOD

For the finite volume discretization a ghost-cell methodology was implemented to impose the boundary conditions at the immersed body surface. The methodology is similar to the approaches proposed by Tseng and Ferziger [15], Majumdar et al. [16], and later used by Ghias et al. [17, 18].

In general, the immersed body geometry is imported from a CAD program. Once the immersed body is embedded to an underlying volume mesh, it is necessary to flag the computational cells as either fluid or solid (i.e., inside the immersed body) cells. We used a ray-tracing technique to determine which cells belong to the fluid region and which cells belong to the solid region, respectively. This methodology is extremely efficient for both 2D and 3D simulations from a computational point of view, easily parallelizable, and does not represent an overhead in case of moving immersed bodies. Once the solid cells are flagged, it is necessary to determine the intersection of the ghost cells centroid with the immersed body surface. Using the normal intercept to the immersed boundary it is possible to determine the closest point on the immersed boundary surface for each ghost cell; this point and the fluid cells around it are used to construct a bi/tri-linear interpolation scheme in a fashion similar to that described in Kim et al. [64].

Since the exterior sharp interface scheme implemented is explicit, the “fresh cell” problem (i.e., fluid cells which at the previous time step were inside the immersed body) for moving boundaries is less restrictive than for other direct-forcing IBMs. It is worth

mentioning that no velocity field is computed inside the immersed body, and computing the Poisson equation for the pressure correction field on the entire computational domain or on the fluid cells only gives the same pressure correction distribution for the fluid region. This is in agreement with what expected. The Poisson equation for the pressure correction is derived from the continuity equation. Since no velocity field in the solid boundary is computed, the pressure correction is zero inside the immersed boundary and does not influence the solution in the fluid region. With a fractional step method for the solution of the NSEs, if the constraint at the immersed body surface is added before the incompressibility constraint, the finial field at the end of each time step will be solenoidal, but the no-slip boundary condition at the fluid/solid interface will be approximately respected. On the other hand, if the constraint at the immersed boundary interface is added after the incompressibility, the no-slip boundary condition will be satisfied at the immersed body surface, but the flow will be approximately solenoidal. With an iterative process this problem is eliminated. For pressure correction type NSEs solvers (see [47, 48]) the iterative nature of the methods allows to satisfy both constraints upon convergence to the predefined accuracy.

3.1 Numerical Implementation of the Time-Dependent Incompressible NSEs and Energy Equation

The non-dimensional time-dependent incompressible NSEs and energy equation governing incompressible flows under the Boussinesq approximation for the entire computational domain Ω can be written as:

$$\frac{\partial \underline{u}}{\partial t} + \underline{u} \cdot \nabla \underline{u} = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \underline{u} - \frac{Gr}{\mathrm{Re}^2} \theta \cdot \underline{g} + \underline{f} + \underline{f}_{IB} \tag{96}$$
$$\nabla \cdot \underline{u} = 0$$

$$\frac{\partial \theta}{\partial t} + \underline{u} \cdot \nabla \theta = \frac{1}{P\mathrm{e}} \nabla^2 \theta + q + q_{IB} \tag{97}$$

In eq. (96) the dimensionless external force $\underline{f}_{IB}$, which represent the effect of the immersed body on the fluid flow, can be expressed as:

$$\underline{f}_{IB} = \int_{\Gamma_i(t)} \underline{f}_K \left(\underline{X}(s,t) \right) \delta \left(\underline{X} - \underline{x} \right) ds \tag{98}$$

In eq. (97) the dimensionless heat source forcing term q_{IB}, which represent the effect of the immersed body on the fluid temperature distribution, can be expressed as:

$$q_{IB} = \int_{\Gamma_i(t)} q_K \left(\underline{X}(s,t) \right) \delta \left(\underline{X} - \underline{x} \right) ds \tag{99}$$

The immersed body introduces a set of Lagrangian points ($\underline{X}_K$) representing the surface $\Gamma_i(t)$ of the immersed body at time t within the computational domain which does not need to conform to the "immersed" body shape (i.e., non-body fitting mesh). At the Lagrangian points, approximate surface forces $\underline{f}_K$ are applied to enforce the no-slip condition along $\Gamma_i(t)$. For the energy equation, approximate heat source q_K are applied to enforce the Dirichlet boundary condition along $\Gamma_i(t)$.

In the IFVM approach implemented we use "ghost-cells" inside the immersed body such that the boundary conditions are imposed at the precise location of the fluid/solid interface. Consequently this method results in a sharp representation of the immersed boundary. No spurious spreading of the boundary forcing into the fluid region is present as in diffuse interface scheme. Since in general the location of the ghost-cells

centroids is not at the fluid/solid interface, an interpolation scheme is necessary to project the constraint from the fluid/solid interface Lagrangian points $\underline{X}_K$ to the relative ghost-cells. The interpolation operations are performed in a direction normal to the immersed body surface. More details on the implementation of the interpolation operators for the IFVM will be given in the next section.

The heat source forcing term q_{IB} at the Lagrangian points ($\underline{X}_K$) representing the surface $\Gamma_i(t)$ enforces the Dirichlet boundary condition on the temperature field at the immersed body interface, in a fashion similar to the one used for the NSEs where $\underline{f}_{IB}$ enforces the no-slip boundary conditions at the fluid/solid interface. It is worth mentioning that Neumann boundary conditions for the energy equations can be imposed as well at the immersed body interface. In this case the forcing term q_{IB} must be chosen in such a way that the gradient of the temperature in the normal direction to the fluid/solid interface satisfies the required boundary conditions instead of the temperature itself.

3.2 Implementation of the Direct Forcing Scheme for the IFVM

Next the implementation of the direct forcing immersed boundary method for the finite volume approach is described. In order to have the velocity at the Lagrangian points of the immersed body satisfy the no-slip boundary condition, a forcing term $\underline{f}_K(\underline{X}_K(s,t))$ is imposed on the Lagrangian points $\underline{X}_K(s,t)$ to modify its velocity, which must equal the immersed body velocity $\underline{u}_{IB}(\underline{X}_K(s,t))$. The forcing term $\underline{f}_K(\underline{X}_K(s,t))$ is determined as follows. From eq. (96), the force at the immersed body $\underline{f}_{IB}^{n+1}$ can be determined as:

$$\underline{f}_{IB}^{n+1} = \frac{\partial \underline{u}}{\partial t} + \underline{u}\cdot\nabla\underline{u} + \nabla p - \frac{1}{\mathrm{Re}}\nabla^2\underline{u} + \frac{Gr}{\mathrm{Re}^2}\theta\cdot\underline{g} + \underline{f} = \frac{\partial \underline{u}}{\partial t} - rhsu = \frac{\underline{u}^{n+1} - \underline{u}^{n}}{\Delta t} - rhsu^{n} \quad (100)$$

Where *n* and *n+1* represent the time step n and the following time step and *rhsu* is equal to:

$$rhsu^{n} = -\left(\underline{u}^{n}\cdot\nabla\underline{u}^{n} + \nabla p^{n} - \frac{1}{\mathrm{Re}}\nabla^2\underline{u}^{n} + \frac{Gr}{\mathrm{Re}^2}\theta^{n}\cdot\underline{g} + \underline{f}^{n} \right) \quad (101)$$

Eq. (100) must be satisfied at all Lagrangian points $\underline{X}_K(s,t)$ on the immersed body surface:

$$\underline{f}_K^{n+1}\left(\underline{X}_K\right) = \frac{\underline{u}_K^{n+1} - \underline{u}_K^{n}}{\Delta t} - rhsu_K^{n} = \frac{\underline{u}_K^{n+1} - \underline{u}_K^{*}}{\Delta t} + \frac{\underline{u}_K^{*} - \underline{u}_K^{n}}{\Delta t} - rhsu_K^{n} \quad (102)$$

where $\underline{u}_K^{*}$ is a temporary velocity vector which satisfies the NESs:

$$\frac{\underline{u}_K^{*} - \underline{u}_K^{n}}{\Delta t} - rhsu_K^{n} = 0 \quad (103)$$

Since we want to satisfy the no-slip boundary condition at the fluid/solid interface, the velocity $\underline{u}_K^{n+1}$ is equal to $\underline{u}_{IB}(\underline{X}_K)$ at time step *n+1*. Therefore, the forcing exerted on the Lagrangian points at the immersed body surface is:

$$\underline{f}_K^{n+1}\left(\underline{X}_K\right) = \frac{\underline{u}_K^{n+1} - \underline{u}_K^{*}}{\Delta t} = \frac{\underline{u}_{IB}^{n+1}\left(\underline{X}_K\right) - \underline{u}_K^{*}}{\Delta t} \quad (104)$$

By the way of example if the immersed body is at rest $\underline{u}_{IB}(\underline{X}_K) = 0$ for all $\underline{X}_K$ at the immersed body surface. From solving the NSEs on the fluid volume we will determine a fluid velocity at the locations $\underline{X}_K$ ($\underline{u}_K^{*}$) which in general is not equal to zero. Therefore, the forcing term to apply at the Lagrangian points $\underline{X}_K$ will be:

$$\underline{f}_K^{n+1}\left(\underline{X}_K\right) = \frac{\underline{u}_{IB}^{n+1}\left(\underline{X}_K\right) - \underline{u}_K^{*}}{\Delta t} = -\frac{\underline{u}_K^{*}}{\Delta t} \quad (105)$$

It is not a surprise that the forcing term is equal in module and opposite in sign to the fluid velocity at the Lagrangian point $\underline{X}_K$ if at that location the immersed body was not present. In other words the effect of the immersed body is to generate a force at the fluid/solid interface which will produce the desired value of velocity at the Lagrangian points.

When Dirichlet boundary conditions are considered for the immersed boundary in the energy equation, the fluid/solid interface is at temperature θ_{IB}. In order to satisfy the boundary condition at the immersed body Lagrangian points $\underline{X}_K$, a heat source $q_{IB}(\underline{X}_K)$ is determined as flows.

From the energy equation eq. (97) it is possible to get:

$$q_{IB} = \frac{\partial \theta}{\partial t} + \underline{u} \cdot \nabla \theta - \frac{1}{P\mathrm{e}} \nabla^2 \theta - q = \frac{\partial \theta}{\partial t} - rhs_T = \frac{\theta^{n+1} - \theta^n}{\Delta t} - rhs_T \tag{106}$$

where rhs_T is:

$$rhs_T = -(\underline{u} \cdot \nabla \theta - \frac{1}{P\mathrm{e}} \nabla^2 \theta - q) \tag{107}$$

At the Lagrangian points $\underline{X}_K$ on the immersed boundary surface the heat source forcing term $q_{IB}(\underline{X}_K)$ reads:

$$q_K^{n+1}\left(\underline{X}_K\right) = \frac{\theta_K^{n+1} - \theta_K^n}{\Delta t} - rhs_{T,K} = \frac{\theta_K^{n+1} - \theta_K^*}{\Delta t} + \frac{\theta_K^* - \theta_K^n}{\Delta t} - rhs_{T,K} \tag{108}$$

where θ_K^* is the temporary temperature which satisfies the energy equation:

$$\frac{\theta_K^* - \theta_K^n}{\Delta t} - rhs_{T,K} = 0 \tag{109}$$

Since we want to satisfy the Dirichlet boundary condition for the temperature at the fluid/solid interface the heat source at the Lagrangian points must be:

$$q_K^{n+1}\left(\underline{X}_K\right)=\frac{\theta_K^{n+1}-\theta_K^n}{\Delta t}=\frac{\theta_{IB}^{n+1}(\underline{X}_K)-\theta_K^*}{\Delta t} \tag{110}$$

The forcing on the NSEs and the energy equation is direct in the sense that the desired velocity and temperature at the immersed body surface are imposed directly on the boundary without any projection on a Lagrangian mesh and interpolation of the corresponding forcing term back to the Eulerian fluid mesh as for the diffuse interface scheme implemented in the IFEM described in Section 2.

3.3 Coupling of Momentum and Energy Equations with Immersed Boundary Method

To solve the NSEs we used a finite volume approach on a staggered grid coupled with the Semi-Implicit Method for Pressure-Linked Equation Revised (SIMPLER) algorithm ([47, 48]). A fully implicit scheme is used for time discretization, and a second order upwind scheme is used for discretizing the flux term (convection fluxes). Central difference discretization is used for the viscous term. The momentum equations when discretized on a Cartesian mesh can be written as:

$$u_{i,J,K}=\frac{\sum a_{nb}u_{nb}+b_{i,J,K}}{a_{i,J,K}}+\frac{A_{i,J,K}}{a_{i,J,K}}(p_{I,J,K}-p_{I+1,J,K}) \tag{111}$$

$$v_{I,j,K}=\frac{\sum a_{nb}v_{nb}+b_{I,j,K}}{a_{I,j,K}}+\frac{A_{I,j,K}}{a_{I,j,K}}(p_{I,J,K}-p_{I,J+1,K}) \tag{112}$$

$$w_{I,J,k}=\frac{\sum a_{nb}w_{nb}+b_{I,J,k}}{a_{I,J,k}}+\frac{A_{I,J,k}}{a_{I,J,k}}(p_{I,J,K}-p_{I,J,K+1}) \tag{113}$$

where *I*, *J* and *K* refer to the cell center, and *i*, *j* and *k* refer to the cell faces for a staggered approach. *u*, *v* and *w* are the three components of the velocity vector, a_{nb} are the cell neighbor coefficients for the finite volume discretization used and u_{nb}, v_{nb} and w_{nb} represent the velocity at the neighbor cells for the three velocity components, respectively. The *b* term represents the cell source term in the momentum equations and the last term on the rhs of eq. (111) – (113) represent the discretized pressure gradient across the cell faces.

In the SIMPLER algorithm we first determine the "pseudo-velocities" $\underline{\hat{u}}$ defined as:

$$u_{i,J,K} = \frac{\sum a_{nb} u_{nb} + b_{i,J,K}}{a_{i,J,K}} \tag{114}$$

$$v_{I,j,K} = \frac{\sum a_{nb} v_{nb} + b_{I,j,K}}{a_{I,j,K}} \tag{115}$$

$$w_{I,J,k} = \frac{\sum a_{nb} w_{nb} + b_{I,J,k}}{a_{I,J,k}} \tag{116}$$

Using Eq. (114) – (116) we can rewrite eq. (111) – (113) as:

$$u_{i,J,K} = \hat{u}_{i,J,K} + d_{i,J,K}(p_{I,J,K} - p_{I+1,J,K}) \tag{117}$$

$$v_{I,j,K} = \hat{v}_{I,j,K} + d_{I,j,K}(p_{I,j,K} - p_{I,j+1,K}) \tag{118}$$

$$w_{I,J,k} = \hat{w}_{I,J,k} + d_{I,J,k}(p_{I,J,K} - p_{I,J,K+1}) \tag{119}$$

where $d = A/a$ are the pressure coefficients for the three momentum equations. For the time-dependent incompressible NSEs, the continuity equation is replaced by a

pressure-Poisson equation, which after discretization with a finite volume approach reads as:

$$a^{p}_{I,J,K}\, p_{i,J,K} = \sum a^{p}_{nb}\, p_{nb} + b^{p}_{I,J,K} \tag{120}$$

The source term b^p in the rhs of eq. (120) is determined by using the pseudo-velocities. Once the discretized pressure equation [eq. (120)] is solved, the new pressure field is used to solve the discretized momentum equations for the intermediate velocity field $\underline{u}^*$:

$$u^{*}_{i,J,K} = \frac{\sum a_{nb} u^{*}_{nb} + b_{i,J,K} + b^{IB}_{i,J,K}}{a_{i,J,K}} + \frac{A_{i,J,K}}{a_{i,J,K}} (p_{I,J,K} - p_{I+1,J,K}) \tag{121}$$

$$v^{*}_{I,j,K} = \frac{\sum a_{nb} v^{*}_{nb} + b_{I,j,K} + b^{IB}_{I,j,K}}{a_{I,j,K}} + \frac{A_{I,j,K}}{a_{I,j,K}} (p_{I,J,K} - p_{I,J+1,K}) \tag{122}$$

$$w^{*}_{I,J,k} = \frac{\sum a_{nb} w^{*}_{nb} + b_{I,J,k} + b^{*}_{I,J,k}}{a_{I,J,k}} + \frac{A_{I,J,k}}{a_{I,J,k}} (p_{I,J,K} - p_{I,J,K+1}) \tag{123}$$

At this step we use the forcing terms calculated explicitly from the previous iteration to enforce the no-slip boundary condition at the immersed body surface (i.e., b^{IB}). Once the discretized momentum equations are solved, the flow field will satisfy the no-slip constraint at the fluid/solid interface, but it will not be divergence-free. Therefore, it is necessary to solve a pressure correction equation for p` to correct the velocity field and satisfy the divergence-free constraint:

$$a^{p}_{I,J,K}\, p^{'}_{i,J,K} = \sum a^{p}_{nb}\, p^{'}_{nb} + b^{'}_{I,J,K} \tag{124}$$

The term *b'* in eq. (124) represents the mass imbalance for each cell. At convergence the flow satisfies the divergence-free constraint and *b'* is zero in each cell

of the computational domain Ω. Since eq. (124) is a Poisson equation with homogeneous boundary conditions, at convergence p' is zero, hence the name pressure correction. With the pressure correction determined at the internal iteration n, the intermediated velocity field $\underline{u}^*$ is corrected:

$$u_{i,J,K} = u^*_{i,J,K} + d_{i,J,K}(p'_{I,J,K} - p'_{I+1,J,K}) \tag{125}$$

$$v_{I,j,K} = v^*_{I,j,K} + d_{I,j,K}(p'_{I,j,K} - p'_{I,j+1,K}) \tag{126}$$

$$w_{I,J,k} = w^*_{I,J,k} + d_{I,J,k}(p'_{I,J,K} - p'_{I,J,K+1}) \tag{127}$$

With the corrected velocity field the interpolation scheme determines the corrected forcing terms to be applied at the ghost-cells for the next iteration in such a way to satisfy the no-slip constraint at the immersed body surface, and if the check for residuals convergence is passed the computation moves to the next time step, otherwise it is necessary to perform one additional internal iteration.

Due to the iterative nature of the algorithm, for each iteration in time, it is necessary to perform a certain number of internal iterations to satisfy the incompressibility constraint. Adding the no-slip constraint at the immersed body in an explicit way (i.e., the immersed boundary force is determined from the velocity field at the previous internal iteration) determines a system which satisfies both no-slip condition at the fluid/solid surface and incompressibility constraint upon convergence. In fact, if the forcing term due to the immersed body keeps changing at the internal iterations, no converged solution can be obtained, and the incompressibility constraint cannot be satisfied. On the other hand, if the flow field does not satisfy incompressibility in the fluid cells close to the immersed boundary, very large pressure gradients are

present there, which will affect the solution at the next internal iteration. We insist on this point due to the importance of the issue. The iterative process will enforce both the no-slip and divergence-free constraints by progressively correcting the velocity field and the forcing terms at the immersed boundary until both reach the correct value (to the numerical error limit imposed). This issue will be clearly seen in problems where the forcing term at the immersed body is one of the parameters we use to benchmark the implemented immersed boundary method. By the way of example, in the two-dimensional steady channel flow, we monitor the lift coefficient at the cylinder surface. Due to the symmetric and steady nature of the problem, no lift is present on the cylinder, and the total force acting at the body surface in the cross flow direction is zero. Both the IFEM and IFVM predict a lift coefficient for this kind of problem of the same order of magnitude as the threshold on the residuals imposed to stop the internal iteration convergence. This is to say, if we set a threshold of 10^{-10} for the residuals, the lift coefficient for both the IFEM and IFVM is of the same order of magnitude.

This is not enough to state that the no-slip constraint at the immersed body is satisfied for the IFEM due to the diffusive nature of the scheme (i.e., the no-slip constraint is satisfied in an integral way on the fluid cell cut by the immersed body surface). On the other hand, the sharp nature of the IFVM means that the forcing term applied at the ghost-cells has a local nature. Only a correct satisfaction of the no-slip condition at the immersed body surface will provide the right value for the forcing terms at the ghost-cells inside the immersed body and the right lift coefficient.

For the energy equation the same procedure applied to the solution of the momentum equations coupled with an explicit immersed boundary treatment is used. The discretized energy equation with immersed body forcing term can be written as:

$$\theta_{I,J,K} = \frac{\sum a_{nb}\theta_{I,J,K} + b_{I,J,K} + b^{IB}_{I,J,K}}{a_{I,J,K}} \tag{128}$$

where θ is the non-dimensional temperature field, a_{nb} are the cell neighbors coefficients after discretizing the energy equation, b is the cell source term, and b^{IB} is the forcing term for the ghost-cells which are considered as boundary cells (i.e., cells where the known boundary conditions are imposed). For problems where the fluid flow equations are strongly coupled with the energy equation the NSEs coupled with IBM are solved first and the discretized energy equation is solved next. The forcing terms at the ghost-cells are determined explicitly from the temperature field at the previous internal iteration in a way equivalent to the calculation of the forcing term in the momentum equations for the no-slip constraint at the immersed body surface. For the energy equation if Dirichlet boundary conditions are imposed, the temperature at the fluid/solid interface is known. The interpolation scheme will determine the forcing term in the ghost-cell necessary to satisfy the known value of the temperature field at the corresponding point on the interface. The iterative process will converge when the temperature at the immersed body surface equals the known value and the forcing term from the previous internal iteration reaches a converged value.

3.4 Immersed Boundary Treatment

The implemented immersed boundary method employs a multi-dimensional ghost-cell approach to impose the boundary condition on the immersed body surface. It is similar to the methodology proposed by Tseng and Ferziger [15], Majumdar et al. [16], Ghias et al. [17, 18] and Mittal et al. [19].

The current method is designed with the spirit of simulating flows over arbitrary complex 2D and 3D immersed stationary and moving rigid boundaries. The approach chosen to represent the immersed body surface is believed to be flexible enough in such a way not to limit the type of geometries that can be handled. Moreover, the immersed body surface representation method introduces a negligible overhead in terms of memory and CPU time for all the operations associated with the immersed boundary surface (e.g., tracing of the fluid/solid interface, determination of the forcing terms at the ghost-cells interpolated from the flow field, etc.). It is also important to mention the compatibility of the IFVM solver developed with pre-processing software as CAD programs from which to import the immersed body geometry, and post-processing tools, such as Tecplot, for visualization purposes.

In the IFVM approach we choose to represent the surface of the IB with an unstructured mesh made of triangular elements. Many fast and efficient algorithms are available for generation of triangular meshes [76, 77]. Accurate and efficient representation of surfaces can be obtained through the use of non-uniform, non-isotropic meshes. In Figure 4 is shown the surface mesh before a uniform mesh triangulation was generated with STAR-CCM+ (STL file imported from SolidWorks) and after the

uniform mesh generation. This apparently simple mesh will be used to point out some aspects of the immersed boundary interface tool developed.

The unstructured surface mesh is embedded into a Cartesian volume mesh. In Figure 5 is shown an example of a cubic enclosure domain with a sphere at the center of the domain (left), and the sphere surface mesh embedded to the Cartesian mesh (right).

Once the immersed body mesh is read from the IFVM solver, the next step is to build the fluid/solid interface. This allows to identify the Cartesian cells whose centroid is inside the immersed body or at the interface (identified as “solid cells”) and the Cartesian cells whose centroid is outside the solid boundary (identified as “fluid cells”). We used a ray tracing technique to determine if the Cartesian mesh cells are inside or outside the immersed body. By generating a ray at the cell centroid in an arbitrary direction and counting the number of intersections of this ray with the immersed body surface, it is possible to determine if the fluid cell is inside (odd number of intersections) or outside (even number of intersections). This technique is very efficient and easily parallelizable. For stationary boundaries, the fluid cells flagging needs to be performed only once, and represents a very small fraction of the total computational time. For moving boundaries, the fluid cell flagging needs to be performed at every time step. Even for this case this operation takes only a very small fraction of the total CPU time. Therefore, very fine immersed body surface meshes can be used to provide highly accurate representations of the immersed body geometry without any significant overhead for the computing time.

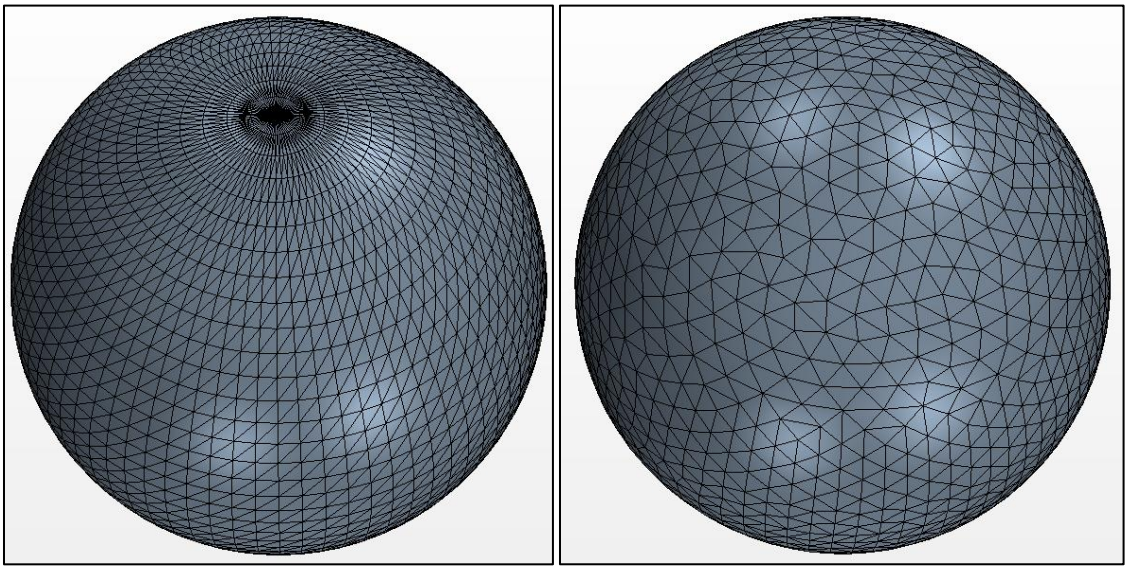

Figure 4. Reference geometry: sphere. Left: before uniform triangulation; right: after uniform triangulation.

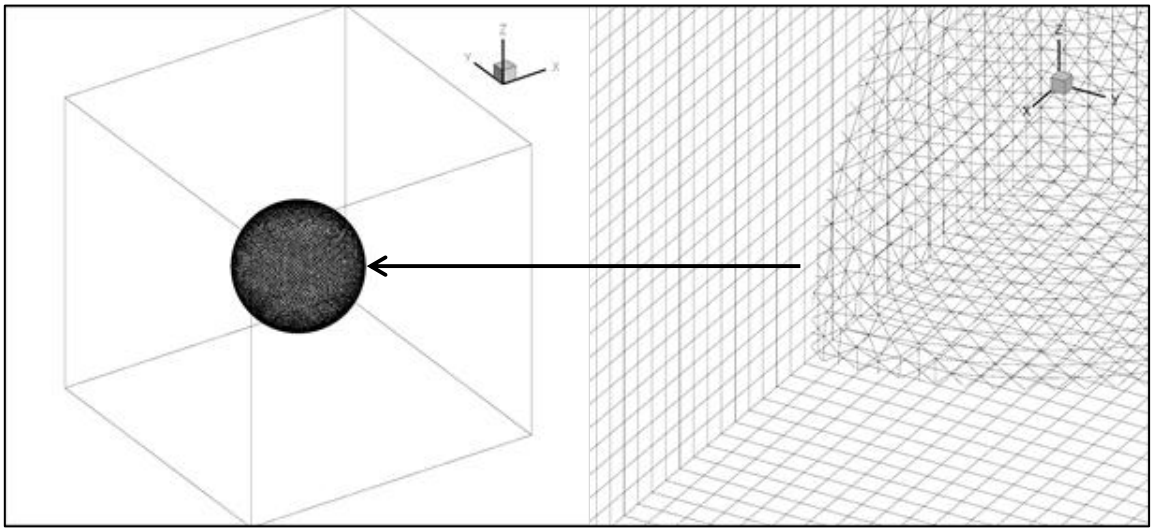

Figure 5. Example of the immersed body (sphere) embedded in a three-dimensional Cartesian grid.

The next step in building the fluid/solid interface for the IBM is the determination of the ghost cells, which are those cells with centroid inside the solid body but with at least one neighbor cell in the fluid region. This is a straightforward operation to perform for Cartesian meshes. Once the ghost cells are identified, it is necessary to determine an appropriate equation for these ghost-cells which leads to an implicit satisfaction of the boundary condition at the fluid/solid interface in the vicinity of each ghost-cell. We follow the approach of Kim et al. [64], where the interpolation procedure for the ghost-cell uses second-order bi-linear (for 2D problems) and tri-linear (for 3D problem) interpolations. The value of the forcing function at the ghost-cell is determined by using the point on the immersed body surface closest to the ghost-cell (i.e., the normal intercept to the immersed body surface passing for the ghost cell considered) and the fluid cells closest to the ghost-cell. In two-dimensional problems the procedure is straightforward since at most four points are used to determine the value of the forcing term at the ghost-cell, with one point being on the immersed body surface (i.e., the immersed boundary condition we want to satisfy) and the other three points being located in the fluid region. In Figure 6 (a) is shown a schematic of the interpolation scheme used for two-dimensional problems, where P_1 represents the point on the immersed body surface for which we want to satisfy the no-slip boundary condition, $\tilde{u}_2$, $\tilde{u}_3$ and $\tilde{u}_4$ are the fluid velocity in the ghost-cell neighbors, and U_1 is the ghost-cell for which the forcing term needs to be determined. If any of the ghost-cell neighbors happens to be inside the immersed body as shown in Figure 6 (b), the interpolation scheme reverses to a linear interpolation in the only direction left (a ghost-cell is by

definition a cell in the immersed boundary with at least one neighbor fluid cell). Once the set of fluid cells neighbors for each ghost-cell is determined, it becomes trivial to determine the value of U_1 for which the no-slip condition at pointP_1 (or P_2 for linear interpolation) is satisfied. In case of linear interpolation, when the ghost-cell fluid neighbor centroid (i.e., $\tilde{u}_4$) is very close to P_2, to avoid stability problems as suggested by Kim et al. [64], the velocity in the fluid region to be used for the interpolation scheme is reconstructed by using the first and second velocity points outside the solid body region. If the ghost-cell fluid neighbor is very close to the interface, the absolute value in the ghost-cell may be greater than the nearby fluid point values and the solution may not converge. By using the first two points in the fluid region close to the ghost-cell this problem is eliminated.

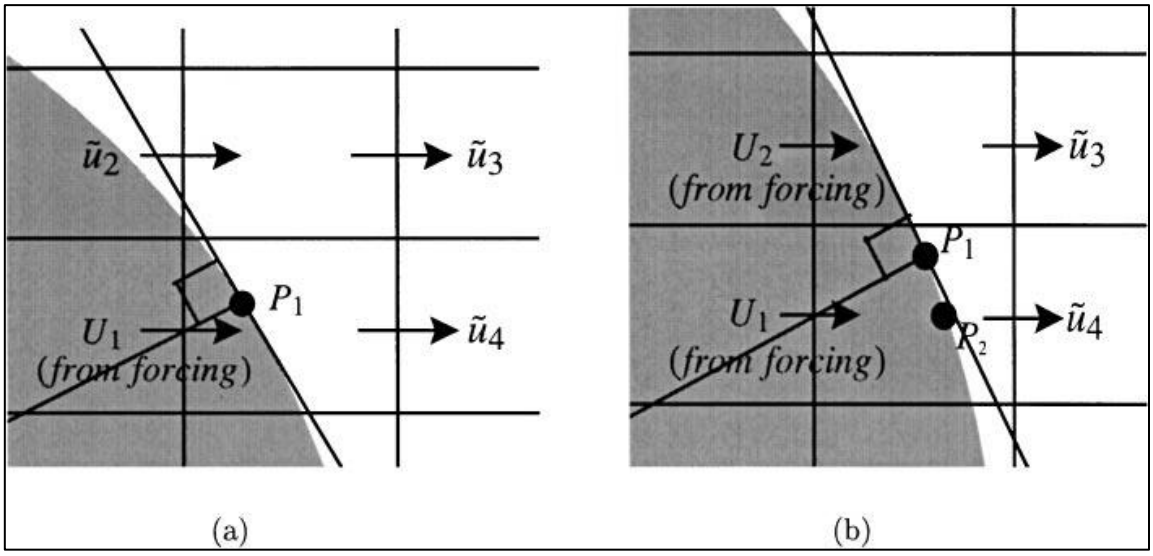

Figure 6. Two-dimensional schematic diagram for the interpolation scheme: (a) bilinear interpolation; (b) linear interpolation.

For three-dimensional problems at most eight points can be used to determine the value of the forcing term at the ghost cells. One of this is located on the immersed body (the immersed boundary condition we want to satisfy) and the other seven points are located in the fluid region. If any of these points is inside the immersed body, that direction is eliminated by the interpolation procedure which reverses to bi-linear or linear interpolation. Mode details about the handling of the interpolation scheme when the immersed body surface is very close to the first fluid cell centroid and the corresponding stability issues can be found in Kim et al. [64].

It is worth mentioning that the IFVM implemented does not require any internal treatment of the body except at the ghost-cells since the SIMPLER algorithm is used and the forcing needs to be applied only at the solid body boundary. Therefore, the little extra computational effort in building the fluid/solid interface and the interpolation procedure at the ghost-cells is offset by the reduction in the number of cells where the discretized fluid equations need to be solved. The same does not apply to the IFEM where the entire domain Ω is solved for the fluid field (i.e., even the flow inside the immersed body), with the additional constraints on the fluid cells overlapping with the immersed body cells as was described in Section 2.

Another important point about the location of the ghost-cells at fluid/solid interface is the use of a staggered grid arrangement. For staggered grids the velocity components are computed at the cells faces, meanwhile pressure, temperature and scalar fields are computed at the cells centroids. This implies that the location of the ghost-cells for each component of the velocity vector and the pressure field will be at different

positions, i.e., we need to perform an interpolation procedure for each of the velocity components and for the pressure field. In Figure 7 is shown a two-dimensional schematic of the velocity components location on a staggered grid and at the ghost cells. The figure clearly shows that the ghost-cells for the *U*-velocity component (bold horizontal arrow) and the ghost-cells for the *V*-velocity component (bold vertical arrow) are located at different positions inside the solid body. The staggered grid arrangement increases the required storage and computational effort to track the fluid/solid interface. However, this increase is not significant since the boundary is of co-dimension one, i.e., is lower dimensional than the fluid domain. On the other hand, the solver improvements in stability and robustness when using a staggered approach versus a collocated arrangement have been widely documented in literature.

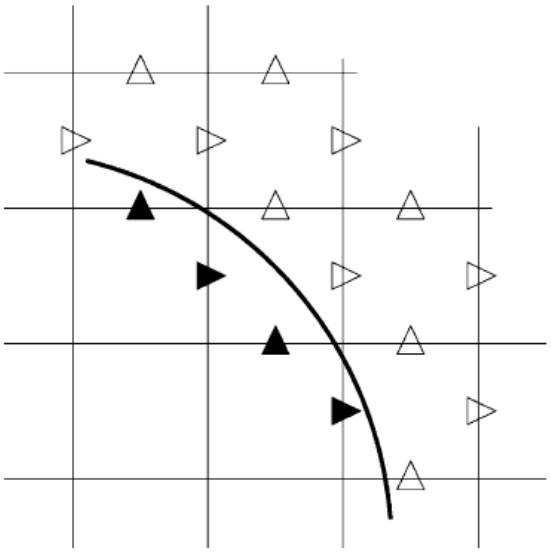

Figure 7. Two-dimensional schematic of a computational domain with an immersed boundary for a staggered grid approach. Ghost-cell U location (►) and V location (▲).

Once the forcing term at every ghost-cell is determined from the flow field calculated at the previous iteration, the discretized system of NSEs can be solved for the current iteration with the ghost-cells having the same effect that boundary cells have for unstructured solvers, which means a direct imposition of the boundary condition explicitly determined at every ghost-cell is added to the source term in the *rhs* of the discretized NSEs. The same argumentation applies to the energy equation. This implies that, in contrast to the feedback forcing method, the stability limit of the current integration scheme is the same as that without the immersed boundary. In turn, this makes possible the simulation of complex three-dimensional conjugate heat transfer problems practical. As Udaykumar et al. [20] and Ye at al. [21] pointed out the use higher order extrapolation/interpolation schemes to evaluate the variables at ghost-cells can preserve at least second-order spatial accuracy.

Due to the bilinear interpolation scheme used, the Dirichlet boundary conditions for the velocity field are prescribed to second order accuracy. Also for the fluid cells a second order accuracy discretization was used. For the solution of the discretized pressure and pressure correction equations, central difference was used, which has second order accuracy. Therefore, we expected local and global second order accuracy for the IFVM solver implemented. The results shown in Section 4 confirm this assumption.

The described ghost-cell method has some similarities with the methodology implemented by Ghias et al. [17], Mittal et al. [19] and the ghost-fluid method (GFM) of Gibou et al. [78]. As Mittal et al. [19] point out, the accuracy, robustness and efficiency

of the GFM method is limited by the 1D interpolation along Cartesian direction. The current method as the one described in Mittal et al. [19] constructs the interpolation along the solid body boundary-normal direction and this not only improves stability, accuracy and robustness of the IFVM solver implemented, but also has implications for the implementation of Neumann boundary conditions at the immersed body surface.

3.5 Immersed Boundary Motion

For rigid immersed boundary, the solid motion can be included into the implemented IFVM relatively easily. Either if the boundary motion is predetermined or if it is determined by the Newton's equations of motion, when the field equations are advanced in time to the next iteration, it is necessary to update the location of the immersed body. This is accomplished by moving the nodes of the surface triangles with a known $\Delta\underline{x} = (\Delta x, \Delta y, \Delta z)$, predetermined in the case of known body motion, or determined with the Newton's equation of motion from the previous time step, for a freely moving immersed body. Once the location of the immersed body is updated, the process of building the immersed body interface, determining the ghost-cells location, and the interpolation operators previously described needs to be updated as well.

We can consider the IFVM as an Eulerian-Lagrangian approach, where the immersed boundary interface is explicitly tracked in a Lagrangian fashion, while the flow equations are solved on a fixed Eulerian grid.

For sharp interface schemes, one stability issue in moving boundary problems is the so-called "fresh-cell" problem [20]. In fact, for moving immersed bodies, cells that

are inside the solid at one time step, might be into the fluid region at the next time step due to the boundary motion. In Figure 8 is shown a 2D schematic of an immersed body boundary motion creating "fresh-cells" from time step *n* to time step *n*+1.

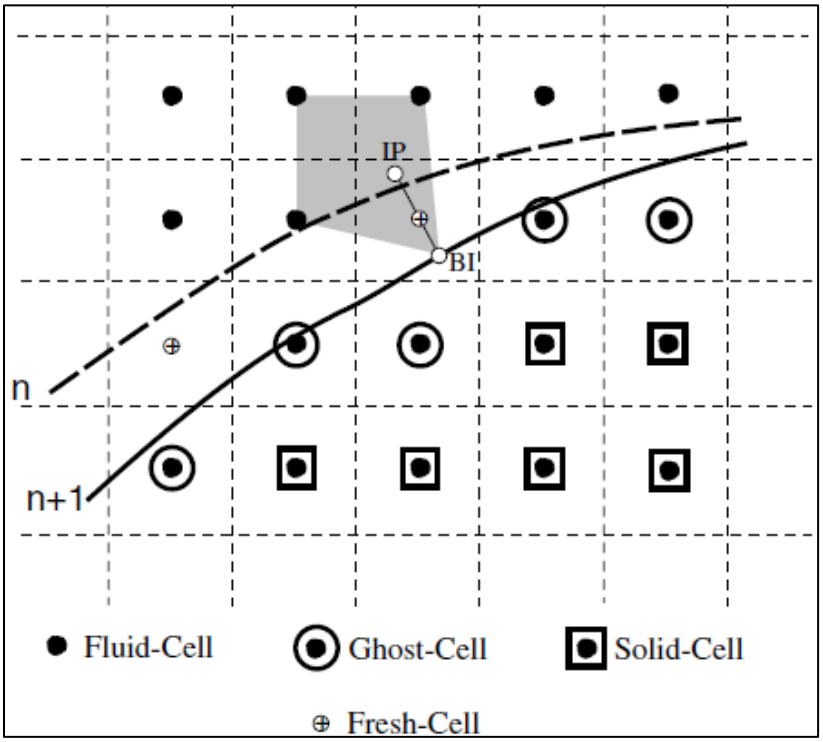

Figure 8. Two-dimensional schematic of an immersed body boundary motion showing the appearance of "fresh-cells" at the next time step.

Due to the iterative nature of the scheme adopted, the "fresh-cell" problem is of no concern for the implemented IFVM. At the current time step some "fresh-cells" will appear in the simulation which do not have any reasonable value from the previous iteration (since they were inside the solid body). The solver starts the internal iterative process to enforce no-slip constraint at the fluid/solid interface and divergence-free

constraint for the fluid region. At each internal iteration the forcing terms at the current ghost-cells are updated explicitly from the previous internal iteration. Even if the first guess of velocity field at the "fresh-cells" and forcing terms at the new ghost-cells is incorrect, after a few internal iterations the process will converge. We do not expect to see a large increase in the number of internal iterations necessary to reach convergence for transient problems with immersed body motion if compared to transient problems with fixed body.

3.6 Multi-Grid Methodology

It is a well-known fact that the discretization error reduces with mesh spacing. Increasing the number of cells (i.e., finer mesh) improves the accuracy of CFD simulations. The SIMPLER algorithm is iterative in nature, so we use the Three-Diagonal Matrix Algorithm (TDMA) in a line-by-line fashion to progressively converge to the "true" numerical solution. Unfortunately the convergence rate of the TDMA rapidly reduces when the mesh is refined. The reason for this behavior is that the iterative solution procedure used removes efficiently only those Fourier modes of the error whose wavelengths are smaller than or comparable to the grid spacing. The multigrid procedure aims at covering a wider spectrum of wavelengths by iterating on various grids, so that on each grid the corresponding error components are efficiently damped.

Consider the discretized system of NSEs in matrix form:

$$A \cdot x = b \tag{129}$$

where A represents the coefficient matrix, x is the numerical solution and b is the source term. Using an iterative method we obtain an intermediate solution y, which satisfies eq. (129) up to a residual r:

$$A \cdot y = b - r \tag{130}$$

The error vector between the true numerical solution and the approximate one is:

$$e = x - y \tag{131}$$

Subtracting eq. (130) from eq. (129) we have a relationship between the error e and the residual r:

$$A \cdot e = r \tag{132}$$

The residual vector is obtained at every time step by using eq. (130). We can use the iterative process to solve eq. (132) and obtain a better approximation of the error vector. It has been established that the solution error has components with a range of wavelengths that are multiples of the mesh size. Iterative methods cause rapid reduction of error components with short wavelengths up to a few multiples of the mesh size. The long-wavelengths components of the error tend to decay very slowly as the iteration process advances. Coarse meshes can attenuate the error components with large wavelengths and, therefore, all error components reduce rapidly. Fine mesh can attenuate only error components with relative short wavelengths. Consequently the convergence process requires a very large number of iterations. Multigrid methods are designed to take advantage of the error behavior and use iteration on meshes of different sizes. The short-wavelength error components are effectively reduced on the finest meshes, whereas the long-wavelength error components are eliminated on coarse

meshes. The computational cost on coarse meshes is a fraction if compared to the one of the finest meshes and is offset by the benefit of much improved convergence rate.

In our approach we use a Full Multi-Grid (FMG) W cycle. Following will be outlined only the main steps of this procedure. We start by solving the discretized equations (e.g., the NSEs, the energy equation or both) on a very coarse mesh. Once convergence is obtained for this coarse mesh, the solution is "prolongated" to a mesh with a refinement factor of two in all directions. The word "prolongation" is used in Multi-Grid approaches to state that the solution is interpolated from the coarse mesh to the fine mesh. We use the solution find on the coarse mesh as a starting guess for solving the discretized equations on the fine mesh. We solve one on the fine mesh and transfer the residual to the coarse mesh. This is the "restriction" step. We use the residual from the fine mesh to solve an equation for the error [eq. (131)] on the coarse mesh knowing that the coarse mesh is more effective in damping the long-wavelength error components than the fine mesh. We solve the error equation on the coarse mesh and prolongate the result to the fine mesh where we add the interpolated error to the approximate solution found at the previous multi-grid step. This approximate solution is "filtered" of the long-wavelength error components and will converge faster than if we kept iterating on the fine mesh. At this point we prolongate the approximate solution from the fine mesh to a finer mesh and start one more cycle on the multi-grid process, by "restricting" the residual from this finer mesh to the coarser meshes, solving for the error equation on these finer meshes, and "prolongating" the improved error back to the finer mesh. At

every cycle we get an error filtered of the long-wavelength components, and the convergence rate is strongly improved.

The cost of every iteration is increased by a factor 5 when we introduce the FMG approach, but the convergence rate is increased of two orders of magnitude, with a total reduction in computational time larger than 10.

4. NUMERICAL SIMULATIONS AND RESULTS

In this section we present the numerical results to validate the IBM implemented for solving the time-dependent incompressible NESs with fractional step method applied to finite element discretization and with Semi-Implicit Method for Pressure-Linked Equations Revised (SIMPLER) applied to the finite volume discretization. First we address the spatial and temporal accuracy of both methods by comparing the numerical results to the analytical solution for the Taylor-Green decaying vortex. Them we consider the simulation of flow past a backward-facing step at Re = 100, 200 and 400. This benchmark case is particularly interesting because of the recirculating zone downstream of the step. Open boundary simulations were performed for two-dimensional flow past a circular cylinder at Re = 20, 40, 100, 150 and 200, and the numerical analyses were compared to experimental data and an extensive collection of other numerical simulations present in literature to address qualitatively and quantitatively the accuracy of the implemented method. Simulations were performed for three-dimensional flow past a sphere at Re = 50, 100, 150, 200, 250 and 300. These simulations allowed to test the accuracy of the implemented IFEM for steady state and transient problems in 2D and 3D configurations, respectively.

To test the accuracy of the coupled NSEs with the energy equation, two and three-dimensional simulations of natural convection in an enclosure at different Rayleigh numbers were performed. Then the IFEM and IFVM approaches were tested against

conjugate heat transfer problems for two and three dimensional natural convection in an enclosure with immersed bodies.

For all the numerical simulations mesh convergence was extensively studied and the converged results from the IFEM and IFVM were compared to experimental data and other numerical data available from literature and against the commercial code STAR-CCM+/V7.04.006, where a body fitted finite volume discretization is implemented.

4.1 Taylor-Green Decaying Vortex

The spatial and temporal accuracy of the IFEM and IFVM implemented is addressed considering the two-dimensional Taylor-Green decaying vortex flow (see [29]) in a square box, for which the analytical solution is available:

$$u(x,y,t) = -u_0 \cos\left(\frac{\pi}{L}x\right)\sin\left(\frac{\pi}{L}y\right)\exp\left(-\frac{2}{\mathrm{Re}}\left(\frac{\pi}{L}\right)^2 t\right) \tag{133}$$

$$v(x,y,t) = u_0 \sin\left(\frac{\pi}{L}x\right)\cos\left(\frac{\pi}{L}y\right)\exp\left(-\frac{2}{\mathrm{Re}}\left(\frac{\pi}{L}\right)^2 t\right) \tag{134}$$

$$p(x,y,t) = p_0 - \frac{u_0^2}{4}\left[\cos\left(\frac{2\pi}{L}x\right)+\cos\left(\frac{2\pi}{L}y\right)\right]\exp\left(-\frac{4}{\mathrm{Re}}\left(\frac{\pi}{L}\right)^2 t\right) \tag{135}$$

The square box dimensions are *[-L,L] x [-L,L]*. A circle of diameter $D = L$ is embedded at the center of the box. Initial conditions can be obtained by eqs. (133) - (135) with $t = 0$. Time dependent boundary conditions at the box and the embedded circle are given by eq. (133) and eq. (134). Since we use a diffuse interface scheme for the finite element discretization, our forcing points are located on the circle boundary

with $\Delta s = \pi/2 \bullet \Delta x$, where Δs represents the surface mesh size and Δx represents the fluid volume mesh size (for uniform volume meshes). Le et al. [23] point out that for too many surface control points, the system becomes ill-conditioned and oscillations in the surface forces are observed which cause stability problems at high Reynolds numbers. They suggested as criterion $\Delta s \approx 0.9\,\Delta x$. We observed stability problems for $\Delta s < \pi/4 \bullet \Delta x$ (i.e., $\Delta s \approx 0.785\,\Delta x$) and in general, a reduction in the CFL number is determined with increasing the number of control points above this threshold. Due to the third order accuracy in space and excellent agreement with the experimental data (see below) this is of no concern. Actually, the use of less control points if compared to traditional IBMs, determines a significant reduction in the computational time due to the reduced sizes of the projection and interpolation operator which require additional numerical computations for solving the Lagrange multipliers (i.e., the force at the immersed body) at every time step. Su et al. [79] address $\Delta s = 2/3\,\Delta x$ as spacing-independent solution when they use delta functions as interface scheme. On the other hand, Kang and Hassan [29] and Wu [80] point out that for the explicit diffuse interface scheme streamlines penetrate the boundary of the cylinder. This means that the no-slip boundary condition at the fluid/solid interface is satisfied up to a certain approximation.

The approximation introduced at the IB interface by the diffuse interface scheme can be completely removed if the immersed body is discretized with a volume mesh having the same dimension as the fluid volume mesh, instead of a surface mesh of co-dimension one. Using a volume mesh for the immersed body allows to have forcing points inside the immersed body, where the same boundary conditions applied at the

fluid/solid interface are used [see last of eq. (42)]. This means that the fluid inside the immersed body will be forced to have the same velocity as the fluid at the fluid/solid interface. By the way of example, if the immersed body is at rest, then the fluid at the immersed body interface must have zero velocity because of the no-slip condition at the interface. Using a surface mesh for the immersed body of co-dimension one, the no-slip condition is imposed only at the fluid/solid interface and, if the surface mesh used is too coarse, some fluid might enter the solid region, which means streamlines penetrate the boundary of the cylinder. Using a volume mesh for the immersed body, the no-slip condition is enforced in the entire fluid region overlapping the solid region, and no fluid crosses the cylinder boundary.

This problem is completely eliminated in the IBM with finite volume discretization, where the fluid volume mesh and the immersed body volume mesh are the same (i.e., overlap). For each fluid volume cell which happens to be crossed by the immersed body interface, the only information needed is the distance from the ghost-cell centroid to the immersed body surface. A finer triangulation of the immersed body will provide a better approximation of the immersed body geometry, but it is of no concern from a numerical stability point of view. This is due to the fact that the number of points actually used to discretize the immersed body surface is always equal to the number of fluid volume cells crossed by the immersed body (i.e., the number of ghost-cells found at the fluid/solid interface).

The Taylor-Green decaying vortex simulation was performed at $Re = 10$. Six uniform volume mesh refinements (i.e., cycle = 0 through cycle = 5) were used to

determine the accuracy in space, with $\Delta x = L / 2^{2+cycle}$, $\Delta s = \pi L / (2^{2+cycle} \bullet 2)$ in the finite element discretization. The numerical results were compared to the analytical solution at normalized time $t^* = t \bullet U_0 / L$. In Table 1 are shown the volume and surface mesh sizes (relative to the reference length L) for the 6 refinements considered.

For the finite volume discretization the simulations were performed with the same number of fluid volume cells used in the finite element case, whereas 100 cell points for the embedded circle were used for all cycles. As mentioned previously, for the finite volume approach the actual number of forcing points is not correlated to the triangulation of the immersed body surface but depends only on the fluid volume mesh. Table 2 shows the fluid volume mesh size (relative to the reference length L) and the total number of fluid cells used in the 6 refinements considered for the finite volume approach simulations.

Table 1. Volume and immersed body surface mesh sizes for the refinement considered (finite element discretization)

Cycle	Fluid volume mesh Δx	IB surface mesh Δs
0	L/4	πL/8
1	L/8	πL/16
2	L/16	πL/32
3	L/32	πL/64
4	L/64	πL/128
5	L/128	πL/256

Table 2. Volume mesh sizes and total number of cells for the refinement considered (finite volume discretization)

Cycle	*Volume mesh Δx*	*Number of fluid cells*
0	L/4	64 (8x8)
1	L/8	256 (16x16)
2	L/16	1024 (32x32)
3	L/32	4096 (64x64)
4	L/64	16384 (128x128)
5	L/128	65536 (256x256)

The numerical accuracy in space of the finite element IBM method implemented was determined comparing the L_2, H^1 and L_∞ error norms inside the embedded circle. For each cell of the triangulation, the three error norms are defined as:

$$\left\|\underline{u}-\underline{u}_h\right\|_{L_2(K)}=\left(\int_K\left|\underline{u}-\underline{u}_h\right|^2 d\underline{x}\right)^{1/2} \tag{136}$$

$$\left\|\underline{u}-\underline{u}_h\right\|_{H^1(K)}=\left(\left\|\underline{u}-\underline{u}_h\right\|_{L_2(K)}+\left|\underline{u}-\underline{u}_h\right|^2_{H^1(K)}\right)^{1/2} \tag{137}$$

$$\left\|\underline{u}-\underline{u}_h\right\|_{L_\infty(K)}=\max_{\underline{x}\in K}\left|\underline{u}\left(\underline{x}\right)-\underline{u}_h\left(\underline{x}\right)\right| \tag{138}$$

and:

$$\left|\underline{u}-\underline{u}_h\right|_{H^1(K)}=\left(\int_K\left|\nabla\left(\underline{u}-\underline{u}_h\right)\right|^2 d\underline{x}\right)^{1/2} \tag{139}$$

Once the error norms at every cell have been evaluated, it is possible to determine the global error norms as:

$$E = \|\underline{e}\| = \left(\sum_i \underline{e}_i^2 \right)^{1/2} \tag{140}$$

The L_2 error norm is an integral measure of how much the numerical results deviate from the analytical solution, meanwhile the H^1 error norm is a combination of the L_2 error norm and the H^1 error semi-norm, where the latter is an integral measure of the difference between the gradient of the numerical results and the gradient of the analytical solution. Since Lagrange multipliers present at the embedded circle introduce a discontinuity in the solution (i.e., the solution is not smooth there), the H^1 error norm gives a very good indication of the IBM effect on the analytical solution. The L_∞ error norm determines the maximum deviation of the numerical results with respect to the analytical solution.

In Figure 9 are shown the Taylor-Green decaying vortex velocity magnitude (left) and velocity vector (right) at normalized time $t^* = 1.0$, with the implicit diffuse interface scheme implemented for the finite element IBM used. The figures show a central vortex and the embedded circle rotating at the same fluid velocity. This means that in the limit for very small meshes the Lagrange multipliers that represent the forces at the embedded circle surface would become infinitesimal. The numerical results confirm what expected providing a maximum Lagrange multiplier of $4.2 \bullet 10^{-5}$ at the embedded circle surface.

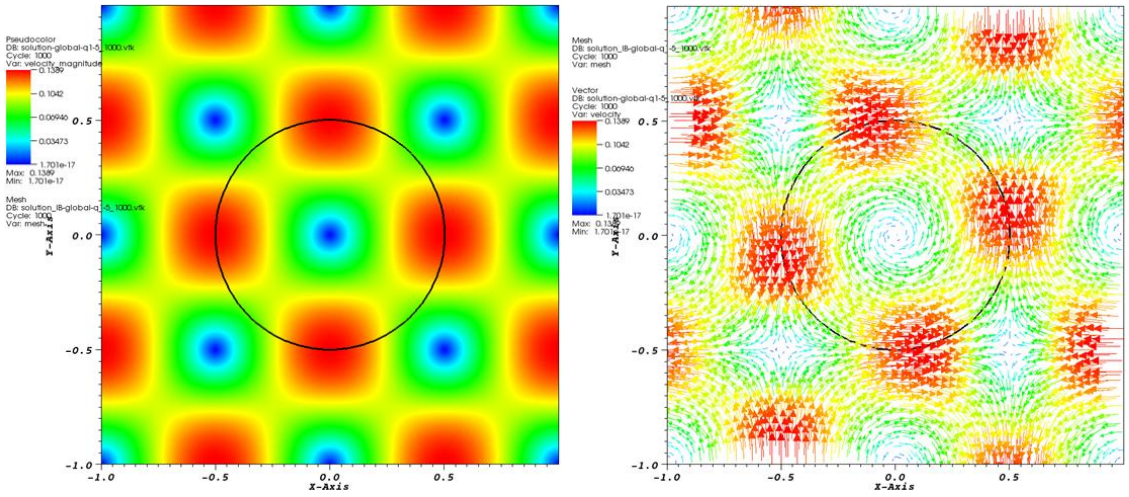

Figure 9. Taylor-Green decaying vortex velocity magnitude (left) / velocity vector (right) at t* = 1.0, and Δx = L/128 (IFEM). The solid line represents the embedded circle.

In Figure 10 are shown the Taylor-Green decaying vortex velocity magnitude (left) and velocity vector (right) at normalized time $t^* = 1.0$, with the explicit sharp interface scheme implemented for the finite volume IBM used. The velocity contours and velocity vectors are qualitatively and quantitatively in excellent agreement with the results obtained from the finite element simulations of the Taylor-Green Decaying vortex. The numerical results confirm what expected providing a maximum forcing term of $5.3 \cdot 10^{-9}$ at the embedded circle surface.

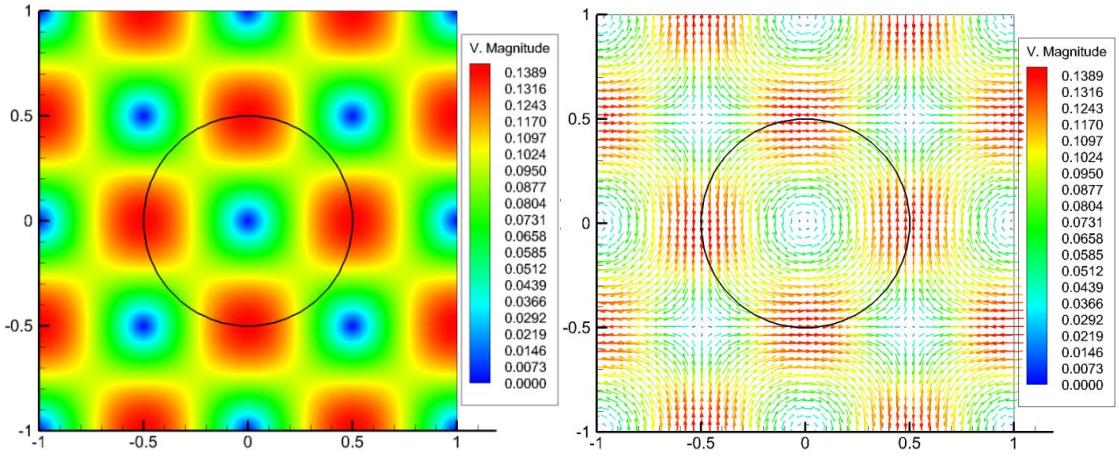

Figure 10. Taylor-Green decaying vortex velocity magnitude (left) / velocity vector (right) at t* = 1.0, and Δx = L/128 (IFVM). The solid line represents the embedded circle.

In Table 3 and Table 4 are shown the Taylor-Green decaying vortex finite element IBM error table and convergence rate table, respectively. Table 1 provides the finite element volume and immersed body surface mesh sizes for the refinements (i.e., cycles) considered. Table 3 addresses for each cycle the volume mesh number of cells, how many degrees of freedom were used for the volume mesh (# dofs) and the embedded circle surface mesh (# dofs IB) and the L_2, H^1 and L_∞ error norms defined in eq. (136) – (139), respectively.

The implicit diffuse interface scheme was used to take into account the presence of the embedded circle in the IFEM approach, and the explicit sharp interface scheme was used for the IFVM approach. From Table 3 it is possible to address the excellent quality of the numerical simulations due to the use of bi-quadratic shape functions for the velocity field and the Lagrange multipliers, and bi-linear shape functions for the pressure field in the IFEM. With relatively coarse mesh (i.e., for cycle 5: $\Delta x = L/128$, Δs

$= \pi L/256$) the L_2 error norm is of the order $9.6 \bullet 10^{-8}$, while the maximum error norm L_∞ is of the order $2.4 \bullet 10^{-7}$.

Table 4 shows the IFEM convergence rate in log scale for the H^1 and L_2 error norms in columns four and six, respectively. As expected, for bi-quadratic shape function the numerical scheme shows an H^1 error norm convergence rate of 2.0, and an L_2 error norm convergence rate of 3.0. The last value in the L_2 error norm convergence rate points out that the accuracy cutoff with which the linear system of discretized NSEs is solved was reached, and the error norm convergence starts to deteriorate. Increasing the accuracy of the conjugate gradient solver for the NSEs will result in a larger number of iterations to get the converged solution at each time step, preserving the third order accuracy of the implemented IBM scheme also for the finest meshes. The implicit diffuse interface scheme was used to take into account the presence of the embedded circle.

Table 3. IFEM error table for the Taylor-Green decaying vortex – spatial accuracy

Cycle	*# fluid volume cells*	*Δx*	*Δs*	L_2 *– error*	H^1 *– error*	L_∞ *– error*
0	16	L/4	πL/8	2.595e-03	3.421e-02	5.819e-02
1	64	L/8	πL/16	2.975e-04	8.794e-03	7.500e-04
2	256	L/16	πL/32	3.588e-05	2.176e-03	7.584e-05
3	1024	L/32	πL/64	4.426e-06	5.347e-04	1.240e-05
4	4096	L/64	πL/128	5.485e-07	1.327e-04	1.468e-06
5	16384	L/128	πL/256	9.662e-08	3.319e-05	2.409e-07

Table 4. IFEM convergence rate table for the Taylor-Green decaying vortex – spatial accuracy

Cycle	*# fluid volume cells*	*Δx*	H^1 *– error*		L_2 *– error*	
0	16	L/4	3.421e-02	conv. rate	2.595e-03	conv. rate
1	64	L/8	8.794e-03	1.96	2.975e-04	3.12
2	256	L/16	2.176e-03	2.01	3.588e-05	3.05
3	1024	L/32	5.347e-04	2.03	4.426e-06	3.02
4	4096	L/64	1.327e-04	2.01	5.485e-07	3.01
5	16384	L/128	3.319e-05	2.00	9.662e-08	2.50

Table 5 shows the Taylor-Green decaying vortex IFVM error table. Table 2 provides the finite volume mesh sizes and the total number of fluid cells for the refinements (i.e., cycles) considered. For each refinement cycle are shown in Table 5 the total number of fluid volume cells used, the L_2 and L_∞ error norms defined in eq. (137) and eq. (139) respectively. The explicit sharp interface scheme was used to take into account the presence of the embedded circle. There is no need to check for the gradient of the solution at the fluid/solid interface (i.e., H^1 error norm) since the sharp interface scheme used for the IFVM approach does not introduce any discontinuity at the fluid/solid interface. Due to the second order accuracy for the space discretization of the NSEs and the projection-interpolation operators at the immersed body surface (see discussion below), both the L_2 and L_∞ error norms for the finite element discretization are superior to the L_2 and L_∞ error norms for the finite volume discretization if the same mesh size is considered, as the comparison of Table 3 and Table 5 points out. By the

way of example, for the finest mesh considered the finite volume discretization gives L_2 and L_∞ error norms two orders of magnitude larger than the respective error norms for the finite element discretization. The loss of accuracy in the numerical results is counterbalanced by the reduction in the computing time for the finite volume approach with respect to the finite element one of at least one order of magnitude, as will be better addressed in the following sections.

Table 5. IFVM error table for the Taylor-Green decaying vortex – spatial accuracy

cycle	*# fluid volume cells*	*Δx*	*#dofs*	L_2 *– error*	L_∞ *– error*
0	64	L/4	262	4.854e-03	9.997e-03
1	256	L/8	902	2.327e-03	5.0286e-03
2	1024	L/16	3334	5.507e-04	1.248e-04
3	4096	L/32	12806	1.622e-04	4.116e-04
4	16384	L/64	50182	4.508e-05	1.171e-04
5	64536	L/128	198662	2.367e-05	5.476e-05

In Figure 11 is shown the L_2 error norm function of the number of mesh points across the embedded circle in a log scale for the implemented IFEM with implicit diffuse interface scheme compared to the L_2 error norm for the finite element simulations without embedded boundary (i.e., no IBM), and a reference slope -3 error norm. With and without the embedded circle the finite element discretization shows a third order convergence rate in space as expected by using bi-quadratic shape functions for

discretizing the velocity field. It is worth mentioning that in our approach we do not use discrete delta functions, which are well known to reduce the spatial accuracy (see [29, 72, 80, 81] among others) of the numerical method implemented. As pointed out before, the use of finite element methods allows to discretize the immersed boundary surface with the same shape functions used to solve the flow problems, and an *ad hoc* projection and interpolation operators (i.e., discrete delta functions) are not necessary. This is the reason why we get the same error norms convergence rate with and without IBM.

For the finite volume IBM approach with explicit sharp interface scheme, a second order accuracy was used for the space discretization of the NSEs, and bi-linear/tri-linear (i.e., second-order) interpolation scheme were used for evaluating the momentum forcing at the immersed body surface. Figure 12 shows the L_2 error norm function of the number of mesh points across the embedded circle in a log scale for the finite volume approach, compared to the L_2 error norm for the finite volume simulations without embedded boundary, and a reference slope -2 error norm. Both the simulations with and without the embedded circle show a second order accuracy in space in agreement with the numerical and interpolation schemes implemented.

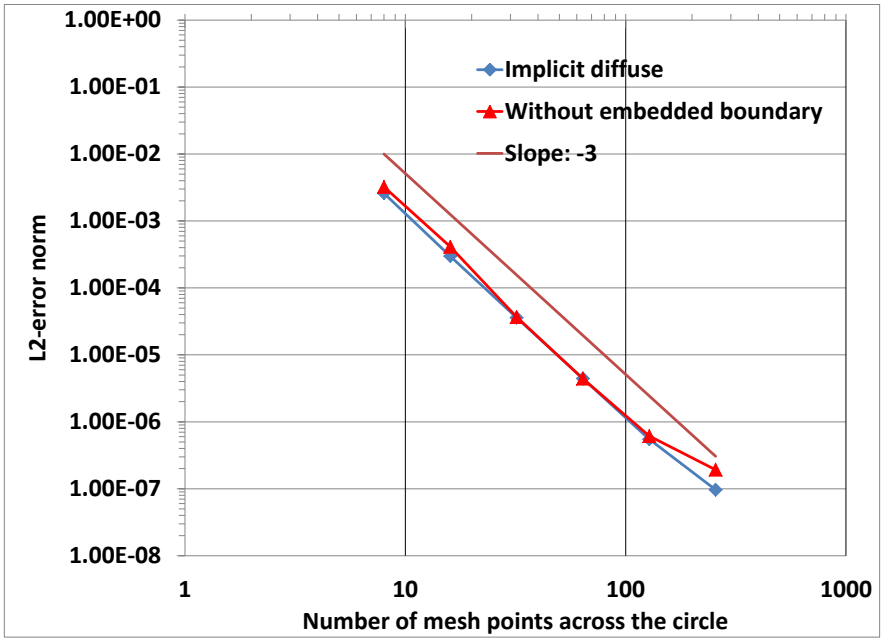

Figure 11. Spatial accuracy of the IFEM with implicit diffuse interface scheme applied to the Taylor-Green decaying vortex.

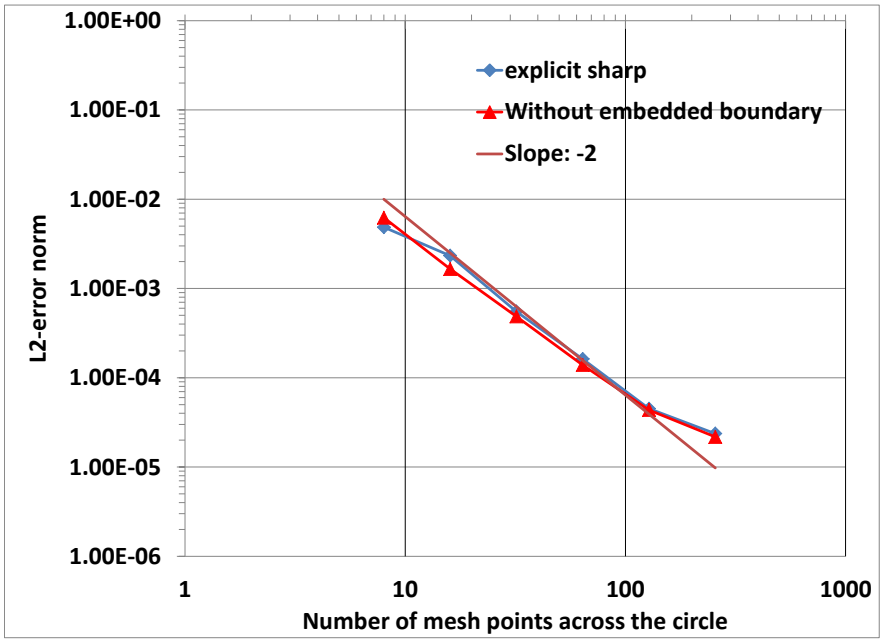

Figure 12. Spatial accuracy of the IFVM with explicit sharp interface scheme applied to the Taylor-Green decaying vortex.

To assess the temporal accuracy of the implemented finite element and finite volume IBMs, we simulated the Taylor-Green decaying vortex with the finest volume and surface meshes ($\Delta x = L/128$, $\Delta s = \pi L/256$) considered (i.e., cycle 5), reducing the time step from 0.2 to $6.25 \cdot 10^{-3}$, and determined the L_2, H^1 and L_∞ error norms. In Table 6 are shown the calculated error norms for the temporal accuracy test (finite element discretization). In Figure 13 are shown the three error norms function of the time stepping in a log scale (finite element discretization). The comparison with a reference -2 slope indicates that the accuracy in time of the implemented finite element IBM is second order as expected by using the BDF-2 second order discretization in time discussed in Section 2.2.

Table 6. Finite element error table for the Taylor-Green decaying vortex – temporal accuracy

Cycle	*# fluid volume cells*	Δx	*time step*	L_2 *– error*	H^1 *– error*	L_∞ *– error*
0	16384	L/128	0.2	7.570e-03	4.031e-02	1.543e-02
1	16384	L/128	0.1	1.276e-03	7.502e-03	2.550e-03
2	16384	L/128	0.05	2.372e-04	1.651e-03	4.658e-04
3	16384	L/128	0.025	6.235e-05	4.733e-04	1.163e-04
4	16384	L/128	0.0125	1.790e-05	1.461e-04	3.615e-05
5	16384	L/128	0.00625	5.605e-06	5.587e-05	1.277e-05

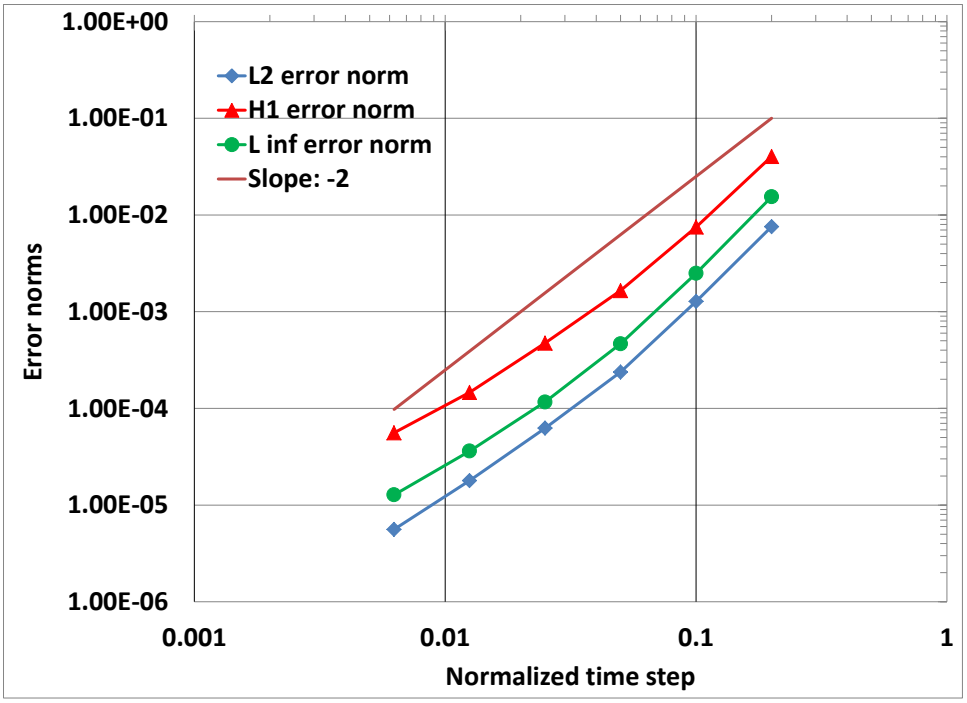

Figure 13. Temporal accuracy for the IFEM with implicit diffuse interface scheme applied to the Taylor-Green decaying vortex.

For the finite volume IBM approach, a first order discretization in time was used. When the L_2 and L_∞ error norms are plotted against the normalized time step, the temporal accuracy of the IFVM is first order, as shown in Table 7 and Figure 14. The comparison of the L_2 and L_∞ error norms for the finite element and finite volume IBM approaches shows the superior accuracy of the former. For a 256x256 cells mesh with a time step of $6.25 \cdot 10^{-3}$ the finite volume discretization error norms are two orders of magnitude larger than the finite element ones.

Table 7. Finite volume error table for the Taylor-Green decaying vortex – temporal accuracy

cycle	*# fluid volume cells*	*Δx*	*Time step*	L_2 *– error*	L_∞ *– error*
0	16384	L/128	0.2	1.852e-01	4.027e-01
1	16384	L/128	0.1	6.786e-02	1.588e-01
2	16384	L/128	0.05	3.674e-02	7.959e-02
3	16384	L/128	0.025	1.510e-02	3.180e-02
4	16384	L/128	0.0125	4.652e-03	9.673e-03
5	16384	L/128	0.00625	1.333e-03	2.516e-03

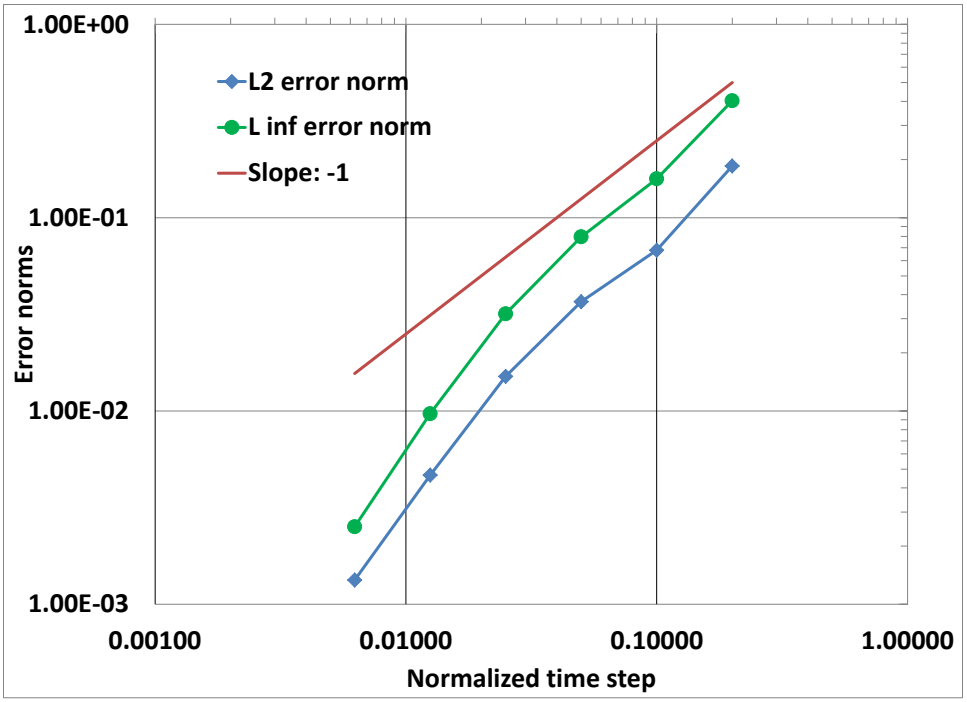

Figure 14. Temporal accuracy for the IFVM with explicit sharp interface scheme applied to the Taylor-Green decaying vortex.

To test the accuracy of the IFEM proposed with adaptive mesh refinement, the same problem was solved using adaptive meshes. The Kelly error estimator was chosen as refinement-coarsening method. The Kelly error estimator is based on refining the mesh in those regions where the solution (i.e., velocity vector) shows larger gradients, and vice versa, coarsen the mesh in regions where the solution is smooth. When adaptive mesh refinement is used, the determination of convergence rates is not as clear as for uniform meshes.

In Table 8 we show the error tables to address the reduction of the L_2, H^1 and L_∞ error norms with adaptive mesh refinement. In Figure 15 (left) is shown the adaptive mesh for cycle 4 of Table 8 and the velocity magnitude (right) at normalized time $t^* = 1.0$. As expected, using adaptive mesh refinement the error convergence rate is reduced with respect to the uniform mesh refinement. Due to the smoothness of the solution in the central region, adaptive mesh refinement is performed mainly close to the embedded circle and to the box boundaries (see Figure 15), where large gradients in the velocity vector field are present. The spatial accuracy, on the other hand, is considered in the region internal to the embedded circle. For this reason the error norms show second order accuracy in Figure 16.

Table 8. Error table for the finite element IBM with adaptive mesh refinement applied to the Taylor-Green decaying vortex – spatial accuracy

Cycle	*# fluid volume cells*	*#dofs*	*# dofs IB*	*L2 – error*	*H1 – error*	*L∞ – error*
0	64	659	16	1.563e-03	3.145e-02	4.267e-03
1	208	2107	32	9.270e-04	1.862e-02	2.327e-03
2	652	6623	64	2.743e-04	6.442e-03	6.953e-04
3	2023	19966	128	7.564e-05	2.487e-03	1.980e-04
4	6277	61601	256	1.672e-05	6.504e-04	5.064e-05
5	19483	191599	512	3.870e-06	1.536e-04	1.326e-05
6	60400	592742	1024	9.610e-07	4.222e-05	2.626e-06

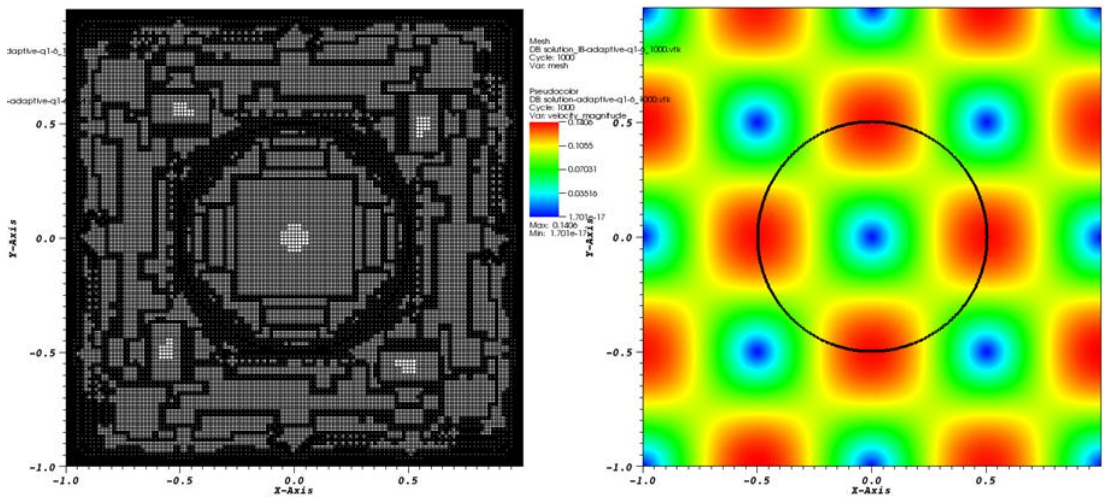

Figure 15. Taylor-Green decaying vortex adaptive mash (left) / velocity magnitude (right) at t* = 1.0 for the IFEM. The solid line represents the embedded circle.

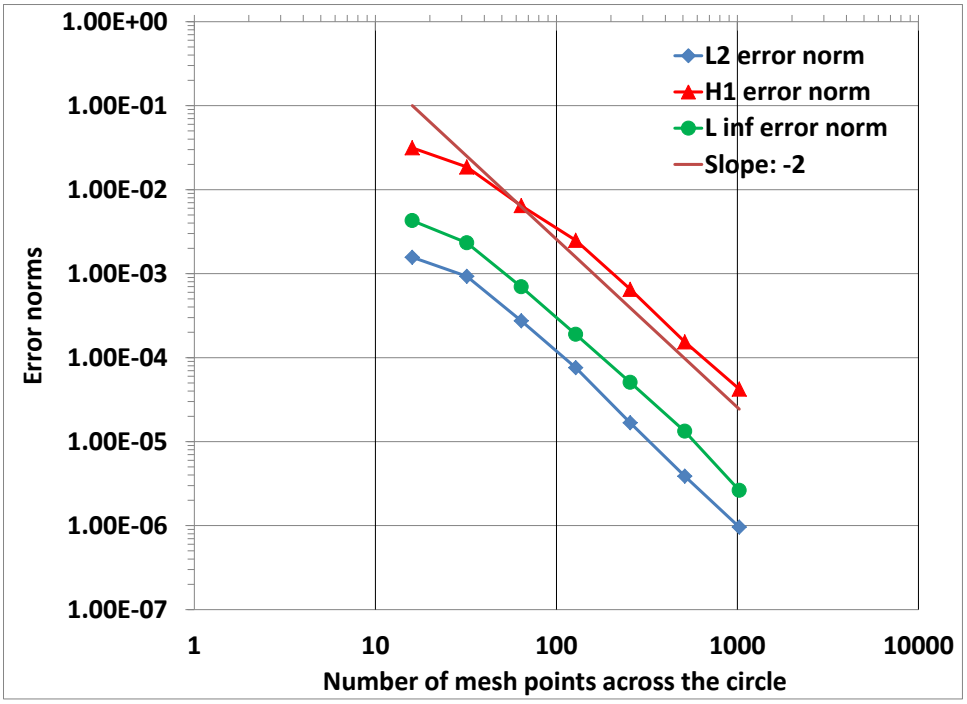

Figure 16. Spatial accuracy of the IFEM with adaptive mesh refinement, fractional step method and implicit diffuse interface scheme applied to the Taylor-Green decaying vortex.

4.2 Two-Dimensional Flow Over a Backward-Facing Step

The flow over a backward-facing step in a two-dimensional channel represents an excellent test case for the accuracy of numerical methods because of the strong dependence of the reattachment length x_r on the Reynolds number. An excessive numerical smoothing to guarantee numerical stability will results in an underestimation of the predicted reattachment length.

The objective of this simulation was to test both the implemented immersed finite element method and immersed finite volume method against the commercial code STAR-CCM+ for the two-dimensional laminar flow over a backward-facing step at *Re* =

100, 200 and 400, respectively, where the Reynolds number is defined as: Re = U_0•H/ν. U_0, H and ν are the inlet reference velocity, the channel height, and the fluid kinematic viscosity, respectively. The channel height was set equal to twice the step height h (aspect ratio of two), where h = 0.05.

At the inlet boundary, located at the step, a constant velocity profile of U_0 = 1.0 was imposed, with V = 0.0. Wall boundary conditions were set for the top and bottom boundaries, and homogeneous Neumann boundary conditions were imposed at the outlet, located at x = 20 h. The step length is l = 5 h.

The reattachment length x_r is defined as the length of the recirculation zone behind the step normalized to the step height h. In Table 9, Table 10 and Table 11 are shown the mesh sensitivity comparison for x_r computed at *Re* = 100, 200 and 400 with the IFEM, the IFVM and STAR-CCM+, respectively. The results show that for each case mesh convergence was obtained with the IFVM and STAR-CCM+, meanwhile the IFEM results for Re = 200 and 400 might require further mesh refinement. The numerical results obtained with the IFVM and STAR-CCM+ are in good agreement with the numerical results of Kim and Moin [63] and the experimental data of Armaly et al. [82] for the range of *Re* number considered. It is worth mentioning that for higher *Re* numbers three-dimensional effects are present and, therefore, three-dimensional simulations would be necessary as pointed out by Armaly et al. [82].

The results obtained with the IFEM approach are in good agreement with the results from the IFVM and STAR-CCM+ for the *Re* = 100 case, with the predicted

recirculation length equal to: x_r = 2.85, 2.99 and 2.98 for the IFEM, IFVM and STAR-CCM+, respectively.

Also the flow pattern predicted with the finite element approach is qualitatively and quantitatively similar to that predicted with the finite volume approach and the body-fitted commercial code, as the streamlines and velocity contours show in Figure 17 and Figure 18, respectively.

Table 9. IFEM mesh sensitivity analysis for the reattachment length x_r in the two-dimensional laminar flow over a backward-facing step *Re* = 100, 200 and 400

Fluid #cells (#dofs)	*Δx*	*IB #cells (#dofs)*	*Δs*	x_r		
				Re = 100	Re = 200	Re = 400
160x16 (23923)	h/16	40x8 (320)	h/8	2.65	4.51	6.95
320x32 (93923)	h/32	80x16 (1280)	h/16	2.81	4.71	7.15
640x64 (372163)	h/64	160x32 (5120)	h/32	2.85	4.81	7.22

Table 10. IFVM mesh sensitivity analysis for the reattachment length x_r in the two-dimensional laminar flow over a backward-facing step *Re* = 100, 200 and 400

Fluid #cells	*Δx*	x_r		
		Re = 100	Re = 200	Re = 400
200x20	h/10	2.81	4.71	7.91
500x50	h/25	2.91	4.91	7.98
1000x100	h/50	2.96	4.92	8.01
2000x200	h/100	2.99	4.96	8.04

Table 11. STAR-CCM+ mesh sensitivity analysis for the reattachment length x_r in the two-dimensional laminar flow over a backward-facing step *Re* = 100, 200 and 400

Fluid #cells	*Δx*	x_r		
		Re = 100	Re = 200	Re = 400
500x50	h/25	2.92	4.92	7.92
1000x100	h/50	2.96	4.95	8.02
2000x200	h/100	2.98	4.97	8.06

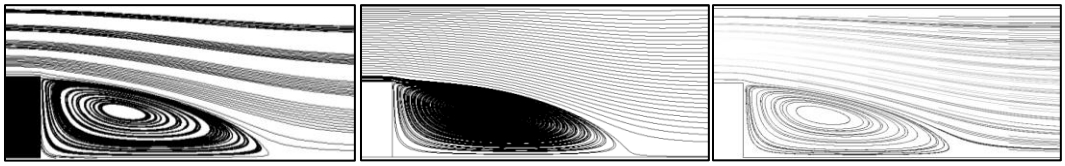

Figure 17. Streamlines for two-dimensional laminar flow over a backward-facing step at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

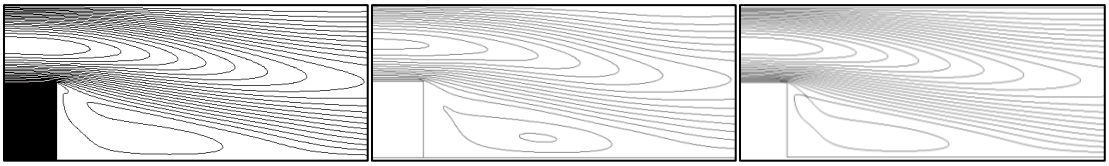

Figure 18. Contours of velocity magnitude for two-dimensional laminar flow over a backward-facing step at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

For *Re* = 200, the IFEM slightly underestimates the recirculation length if compared to the IFVM and STAR-CCM+, with the finest meshes predicting: x_r = 4.81, 4.96 and 4.97 for the IFEM, IFVM and STAR-CCM+, respectively. The flow patterns determined with the two immersed boundary methodologies are very similar to the ones

determined with the body fitted approach as can be seen from the streamlines and velocity contours plotted in Figure 19 and Figure 20, respectively.

For *Re* = 400, the IFEM results deviates from the IFVM and STAR-CCM+ results, underpredicting the recirculation length: x_r = 7.22, 8.04 and 8.06 for the IFEM, IFVM and STAR-CCM+, respectively. The streamlines plotted in Figure 21 point out that the recirculation zone predicted with the IFEM is visibly smaller than that predicted with the IFVM and STAR-CCM+. This behavior is due to the diffusive nature (i.e., artificial or numerical viscosity) of finite element methods if compared to finite volume methods. As Kim and Moin [63] point out excessive numerical smoothing in favor of stability determines an incorrect prediction of the reattachment length.

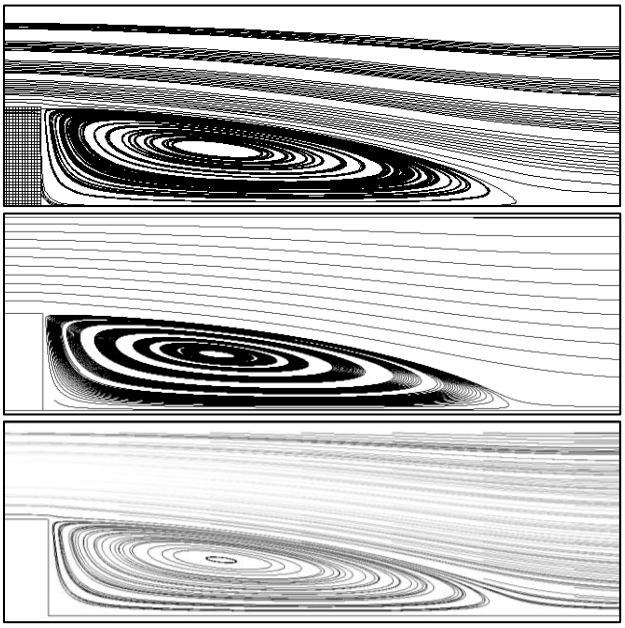

Figure 19. Streamlines for two-dimensional laminar flow over a backward-facing step at *Re* = 200. Top: IFEM vm, dg = 1; middle: IFVM; bottom: STAR-CCM+.

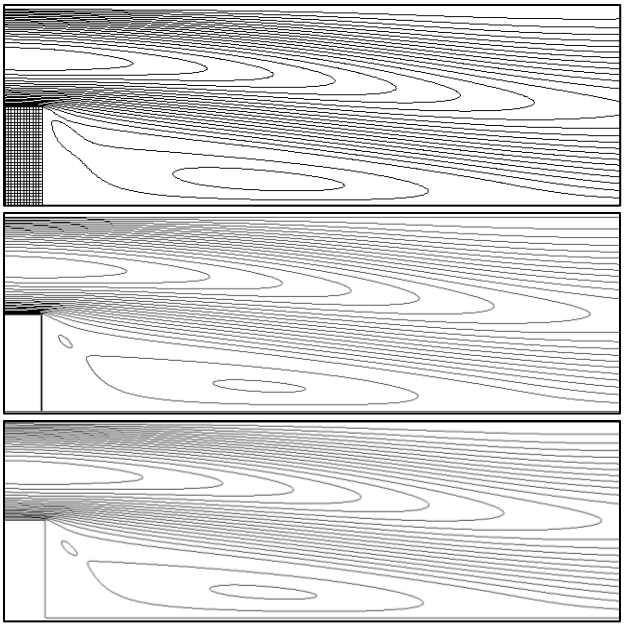

Figure 20. Contours of velocity magnitude for two-dimensional laminar flow over a backward-facing step at *Re* = 200. Top: IFEM vm, dg = 1; middle: IFVM; bottom: STAR-CCM+.

The flow distribution determined with the IFEM is qualitatively similar to that determined with the IFVM and STAR-CCM+ as can be seen from the velocity contours plotted in Figure 22.

It is also worth mentioning that the computations performed with the finest meshes for the IFVM and STAR-CCM+ have a mesh size the same order of the smallest turbulent structures for the *Re* number considered. Although the two-dimensional simulations were performed as steady state and, therefore, the use of Direct Numerical Simulations (DNS) would be improper, the IFVM and STAR-CCM+ simulations were performed with a very large number of cells if compared to the IFEM simulations, due to the higher

computational effort of the latter. This might have an impact on the accuracy of the IFEM results.

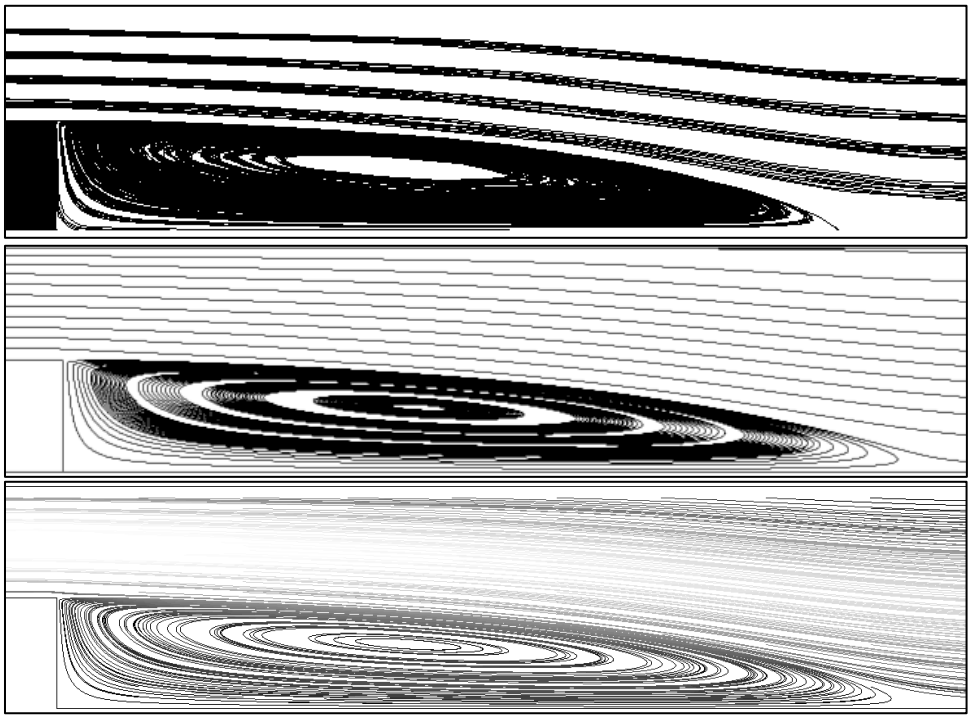

Figure 21. Streamlines for two-dimensional laminar flow over a backward-facing step at *Re* = 400. Top: IFEM vm, dg = 1; middle: IFVM; bottom: STAR-CCM+.

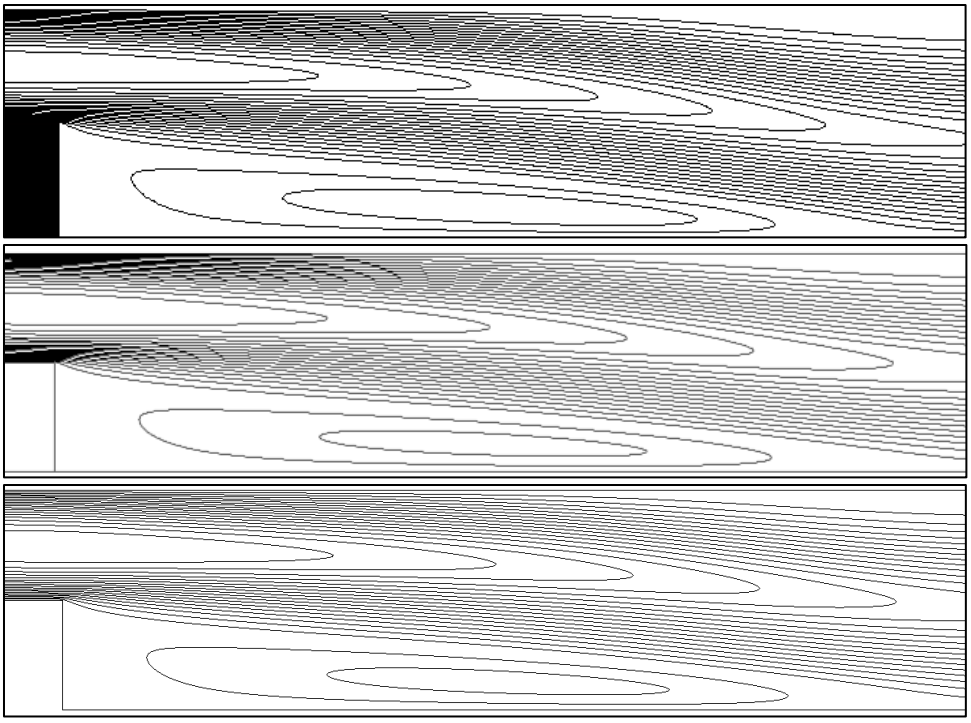

Figure 22. Contours of velocity magnitude for two-dimensional laminar flow over a backward-facing step at *Re* = 400. Top: IFEM vm, dg = 1; middle: IFVM; bottom: STAR-CCM+.

4.3 Two-Dimensional Flow Past a Circular Cylinder

Flow past a stationary circular cylinder is considered next. This is one of the representative benchmark problems for which a vast literature is available of both experimental and numerical data. For this problem, the flow is stationary at low Reynolds numbers (i.e., *Re* < 46). A pair of counter-rotating vortices are generated symmetrically respect to the cylinder centerline in the wake. We investigated simulations at *Re* = 20 and 40 for steady flows. For larger Re numbers, small instabilities in the flow upstream the cylinder are amplified and unsteadiness arises spontaneously. Vortex shedding occurs in the cylinder wake. If the *Re* number is further increased

transition to 3D is induced in the near wake. For unsteady flows, Re = 100, 150 and 200 were investigated. It is worth mentioning, that above Re = 150, 2D flow simulations do not completely reflect the transition in the near wake (see [83, 84]). For the flow past a stationary circular cylinder, analyses were performed using the IFEM implemented with both surface and volume mesh approaches for the embedded cylinder, and the IFVM with explicit sharp interface scheme. For the finite element IBM, bi-linear and bi-quadratic shape functions were used to address the effect of the interpolation (i.e., projection operator) on the implicit diffuse interface scheme used.

4.3.1 Two-Dimensional Steady Flow Past a Circular Cylinder

For the steady flow simulation, the cylinder of diameter D is located at the center of a rectangular domain $\Omega = [-10D, 10D] \times [-10D, 10D]$. In Table 12 is shown the mesh sensitivity analysis for the IFEM simulations of steady flow past a circular cylinder at Re = 20. Table 12 addresses the number of cells used for the fluid volume mesh and the immersed body mesh, respectively. In the second column of Table 12 is shown the cell size normalized to the cylinder diameter D. For the IB both surface and volume meshes were considered, and bi-linear and bi-quadratic shape functions were used. For the finest mesh were used 641 x 641 dofs for each component of the fluid velocity field and 128 forcing points for each component of the Lagrange multipliers at the surface mesh (i.e., $\Delta x = D / 32$, $\Delta s = \pi D / 128$). For inlet and lateral boundaries, Dirichlet boundary conditions were used (i.e., free stream velocity $U_\infty = 1.0$, $V = 0.0$). For the outlet, homogeneous Neumann (i.e., convective) boundary conditions were used.

For steady flow, the drag coefficient C_D and the recirculation length L_w of the counter-rotating vortices in the cylinder wake are used as reference parameters for benchmark with the experimental and numerical data available. The drag coefficient C_D is defined as:

$$C_D = \frac{F_D}{\frac{1}{2}\rho U_\infty^2 D} \tag{141}$$

Where ρ, U_∞ and D represent the fluid density, the fluid free stream velocity and the cylinder diameter, respectively. F_D is the drag force on the cylinder and can be determined using eq. (30) if the immersed body surface mesh approach is used:

$$F_D(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t) \cdot \underline{n}_x d\underline{x} = \int_{\underline{x}} \int_{\Gamma_i(t)} \underline{F}_K\left(\underline{X}(s,t)\right) \delta\left(\underline{X} - \underline{x}\right) ds \cdot \underline{n}_x d\underline{x} \tag{142}$$

and eq. (38) for the immersed body volume mesh approach:

$$F_D(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t) \cdot \underline{n}_x d\underline{x} = \int_{\underline{x}} \int_{\omega(t)} \underline{F}_K\left(\underline{X}(s,t)\right) \delta\left(\underline{X} - \underline{x}\right) ds \cdot \underline{n}_x d\underline{x} \tag{143}$$

The definition of the recirculation length follows the notation used in Coutanceau and Bouard [85]. In the following figures the flow is from left to right (or top to bottom) unless otherwise specified. The last column in Table 12 shows the drag coefficient function of the mesh sensitivity for the IFEM. Refining the mesh the drag coefficient decreases, approaching the experimental value determined by Tritton [86]. Using for the immersed body a surface mesh with bi-linear shape function (i.e., degree = 1) instead of bi-quadratic shape function (i.e., degree = 2) allows to have smaller surface cells, and the diffusive effect of the interpolation scheme is reduced. If volume meshes are used for the

immersed body, even with very coarse meshes, the drag coefficient is very close to the experimental value.

Table 12. IFEM mesh sensitivity analysis for the drag coefficient in the steady flow past a circular cylinder at $Re = 20$

Fluid #cells	Δx	Immersed body #cells	Δs	C_D
20x20	D/2	32 (surface mesh, degree = 1)	πD/32	2.776
40x40	D/4	64 (surface mesh, degree = 1)	πD/64	2.423
80x80	D/8	64 (surface mesh, degree = 1)	πD/64	2.290
160x160	D/16	64 (surface mesh, degree = 1)	πD/64	2.220
320x320	D/32	128 (surface mesh, degree = 1)	πD/128	2.184
320x320	D/32	32 (surface mesh, degree = 2)	πD/64	2.229
80x80	D/8	20 (volume mesh degree = 1)	πD/8	2.314
160x160	D/16	80 (volume mesh degree = 1)	πD/16	2.187
320x320	D/32	320 (volume mesh degree = 1)	πD/32	2.168

In Table 13 is shown the mesh sensitivity analysis for the drag coefficient determined with the IFVM and that calculated by STAR-CCM+, respectively. The mesh convergence study shows that the drag coefficient reached an asymptotic value for both the finite volume IBM and body fitted approaches. It is worth mentioning that the drag coefficient value decreases with mesh refinement for the finite volume IBM approach, and slightly increases with mesh refinement for the body-fitted approach.

Table 13. IFVM and STAR-CCM+ mesh sensitivity analysis for the drag coefficient in the steady flow past a circular cylinder at $Re = 20$

IBM Fluid #cells	Δx	C_D	STAR-CCM+ Fluid #cells	Δx	C_D
200x200	D/10	2.239	200x200	D/10	2.150
400x400	D/20	2.263	500x500	D/25	2.165
800x800	D/40	2.141	1000x1000	D/50	2.167
1600x1600	D/80	2.124	//		//

In Table 14 is shown a comparison of the drag coefficient and recirculation length for $Re = 20$ with experimental data (see [85, 86, 87]) and other numerical results available in literature (see [13, 21, 23, 25, 29, 54, 73, 79, 80, 88-100]). The numerical results obtained with the implemented IFEM and IFVM are in good agreement with the experimental data and the other IBM approaches. Also good agreement with the results obtained by running the commercial CFD code STAR-CCM+ is found. Considering the experimental data as reference value for the drag coefficient, the IFEM with a surface mesh approach and bi-linear or bi-quadratic shape functions gives a -0.09% and 0.315% error on the drag coefficient, respectively.

Using a volume mesh for the immersed body with bi-linear shape function the error on the drag coefficient is -2.430% for the IFEM; the IFVM and STAR-CCM+ give an error equal to -4.41% and -2.475%, respectively. The recirculation length determined with the IFEM for the different immersed body discretizations and shape functions used are very close to the experimental data of Coutanceau and Bouard [85] when the IB is discretized with a surface mesh. If the IB is discretized with a volume mesh, smaller

recirculation lengths are predicted. The recirculation length predicted with IFVM and STAR-CCM+ is very close to the experimental data of Coutanceau and Bouard [85].

Table 14. Comparison of the drag coefficient and recirculation length in the two-dimensional steady flow past a circular cylinder at $Re = 20$

	Year	Characteristics	l/d	C_D
Coutanceau and Bouard [85]	1977	Experiment	0.93	-
Tritton [86] – set #6	1959	Experiment	-	2.222
Tritton [86] – set #5	1959	Experiment	-	2.022
Wieselsberger [87]	1922	Experiment	-	2.050
Lima E Silva et al. [13]	2003	Implicit diffuse direct-forcing, NSEs	1.04	2.040
Ye et al. [21]	1999	Cut cell method, NSE	0.92	2.030
Le et al. [23]	2006	IIM, NSE	0.93	2.050
Le et al. [25]	2008	Implicit diffuse direct-forcing, NSEs	1.05	2.070
Kang and Hassan [29]	2010	Exterior sharp direct-forcing, LBE	0.91	2.057
Taira and Colonius [54]	2007	Implicit diffuse direct-forcing, NSEs	0.97	2.070
Linnick and Fasel [73]	2005	IIM, streamfunction-vorticity	0.93	2.160
Su et al. [79]	2007	Implicit diffuse direct-forcing, NSEs	-	2.200
Wu and Shu [80]	2009	Implicit diffuse direct-forcing, LBE	0.93	2.091
Dennis and Chang [88]	1970	Numerical solution	0.94	2.045
Fornberg [89]	1980	Numerical solution	0.91	2.000

Table 14 Continued

	Year	Characteristics	*l/d*	C_D
Calhoun [90]	2002	Interior sharp direct-forcing, streamfunction-vorticity	0.91	2.190
Russel and Wang [91]	2003	Interior sharp direct-forcing, streamfunction-vorticity	0.94	2.130
Niu et al. [92]	2006	Explicit sharp direct-forcing, LBE	0.95	2.144
He and Doolen [93]	1997	Implicit diffuse direct-forcing, LBE	0.92	2.152
Wang et al. [94]	2009	Implicit diffuse direct-forcing, NSEs	0.98	2.250
Park et al. [95]	1998	Body-fitted method, NSEs	-	2.010
Xu [96]	2008	IIM, NSEs	0.93	2.230
Xu and Wang [97]	2006	IIM, NSEs	0.92	2.230
Choi et al. [98]	2007	Interior sharp direct-forcing, NSEs	0.90	2.020
Rogers and Kwak [99]	1991	Body-fitted method, NSEs	0.93	2.080
Pacheco et al. [100]	2005	Exterior sharp direct-forcing, NSEs	0.91	2.080
Present (IFEM sm, dg = 1)	2012	Implicit diffuse direct-forcing, NSEs	0.90	2.220
Present (IFEM sm, dg = 2)	2012	Implicit diffuse direct-forcing, NSEs	0.93	2.229
Present (IFEM vm, dg = 1)	2012	Implicit diffuse direct-forcing, NSEs	0.85	2.168
Present (IFVM)	2012	Explicit sharp direct-forcing, NSEs	0.92	2.124
STAR-CCM+/V7.04.006	2012	Body-fitted method, NSEs	0.92	2.167

In Figure 23 are shown the vorticity contours for the IFEM numerical simulation of the steady flow past a circular cylinder at *Re* = 20. For the immersed body, the simulations were performed considering a surface mesh and a volume mesh approach, and sensitivity over the shape functions used (i.e., bi-linear and bi-quadratic shape functions) were performed. The figure shows that there is good agreement among the different IFEM approaches considered.

In Figure 24 are shown the vorticity contours for the IFEM with volume mesh for the immersed body and bi-linear shape functions (left), the vorticity contours for the IFEM (center), and the vorticity contours obtained from STAR-CCM+ (right).. Also good agreement is found between the IFEM and the IFVM approaches. The vorticity contours predicted with the two immersed boundary methodologies are qualitatively in good agreement with the vorticity contours determined by the body-fitted commercial code.

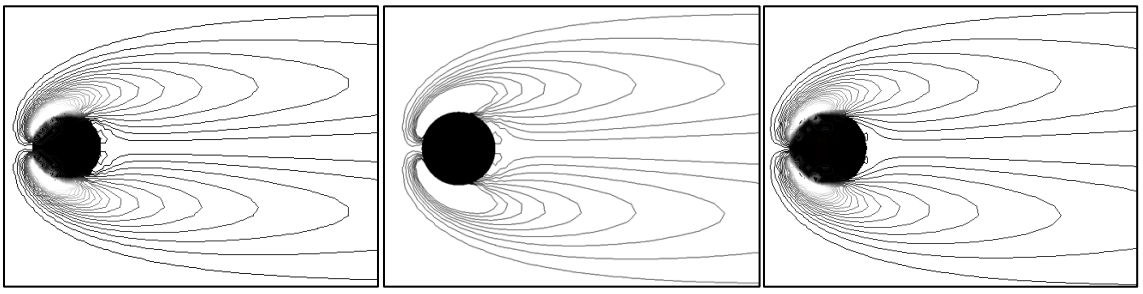

Figure 23. Vorticity contours for steady flow past a circular cylinder at *Re* = 20. IFEM; left: sm, dg = 1; center: sm, dg = 2; right: vm, dg = 1.

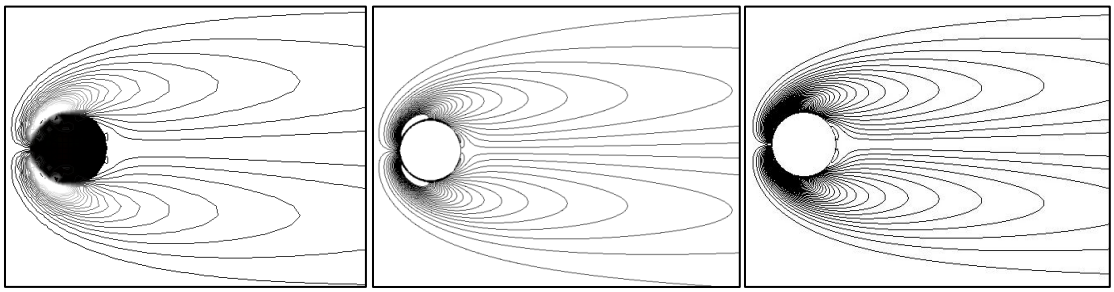

Figure 24. Vorticity contours for steady flow past a circular cylinder at *Re* = 20. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

In Figure 25 are shown the streamlines for the steady flow past a circular cylinder at *Re* = 20 determined with the direct-forcing IFEM and implicit diffuse interface scheme using surface mesh for the immersed body with bi-linear and bi-quadratic shape functions, and volume mesh for the immersed body with bi-linear shape functions. As expected, the diffuse interface scheme allows some fluid to penetrate the cylinder surface when the IB is discretized with a surface mesh (left and center of Figure 25). The pair of counter-rotating vortices appear slightly shifted downwards in the cylinder wake due to the fluid penetration at the cylinder surface. When a volume mesh is used for the immersed body, the fluid inside the cylinder is at rest, and the streamlines do not cross the boundary of the cylinder, with the result that the pair of counter-rotating vortices in the wake are very close to the cylinder surface as shown in the right of Figure 25. In Figure 26 the streamlines for the steady flow past a circular cylinder at *Re* = 20 obtained with the IFEM and volume mesh approach for the immersed body (left) are compared to the streamlines obtained with the IFVM (center) and with the streamlines

computed with STAR-CCM+ (right). The figure shows that for the IFVM there is no fluid penetration at the cylinder surface, and the streamlines predicted with the three different approaches look very similar.

In Figure 27 is shown a comparison of the streamlines for *Re* = 24.3 (left) from the experimental data of Coutanceau and Bouard [85], the streamlines for the IFEM numerical simulations at *Re* = 20 (center) and the streamlines for the IFVM numerical simulations at *Re* = 20 (right). The numerical results show that the symmetric counter-rotating vortices in the cylinder wake are qualitatively and quantitatively in good agreement with the experimental visualization.

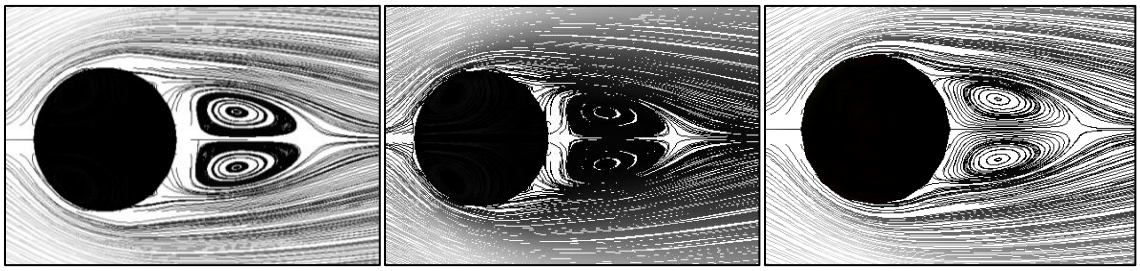

Figure 25. Streamlines for steady flow past a circular cylinder at *Re* = 20. IFEM; left: sm, dg = 1; center: sm, dg = 2; right: vm, dg = 1.

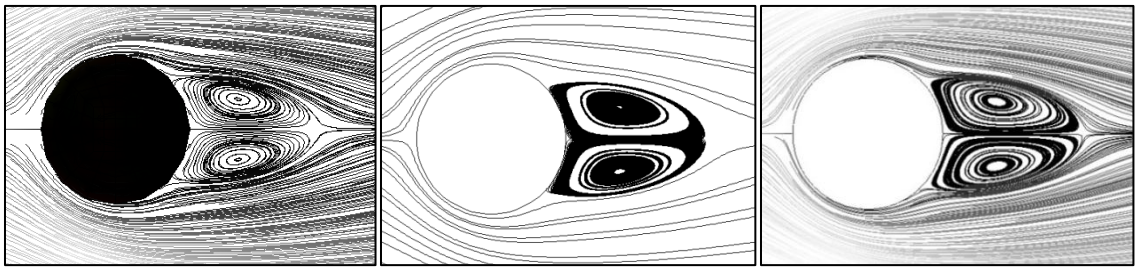

Figure 26. Streamlines for steady flow past a circular cylinder at *Re* = 20. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

Figure 27. Streamlines for steady flow past a circular cylinder at *Re* = 24.3 from [85] (left); *Re* = 20 IFEM vm, dg = 1 (center); *Re* = 20 IFVM (right).

Table 15 addresses the number of cells and mesh size used for the IFEM fluid volume mesh and the immersed body mesh, respectively, in the steady flow past a circular cylinder at *Re* = 40. For the immersed body both surface and volume mesh approaches were considered, and bi-linear and bi-quadratic shape functions were used. For the finest mesh were used 641×641 dofs for each component of the fluid velocity field and 128 forcing points for each component of the Lagrange multipliers at the surface mesh (i.e., $\Delta x = D / 32$, $\Delta s = \pi D / 128$). The same boundary conditions applied for the steady flow past a circular cylinder at *Re* = 20 were used. The drag coefficient reached an asymptotic value for the finest mesh considered.

In Table 16 is shown the drag coefficient determined with the IFVM and that calculated by STAR-CCM+, respectively. The mesh convergence study shows that the

drag coefficient reached an asymptotic value for both the finite volume IBM and body fitted approaches. Also for the steady flow past a circular cylinder at *Re* = 40 the drag coefficient value decreases with mesh refinement for the IFVM approach, and slightly increases with mesh refinement for the body-fitted approach.

Table 15 IFEM mesh sensitivity analysis for the drag coefficient in the steady flow past a circular cylinder at *Re* = 40

Fluid #cells	Δx	IB #cells	Δs	C_D
20x20	D/2	8 (surface mesh, degree = 1)	πD/8	2.043
40x40	D/4	16 (surface mesh, degree = 1)	πD/16	1.910
80x80	D/8	32 (surface mesh, degree = 1)	πD/32	1.786
160x160	D/16	64 (surface mesh, degree = 1)	πD/64	1.707
320x320	D/32	128 (surface mesh, degree = 1)	πD/128	1.661
20x20	D/2	4 (surface mesh, degree = 2)	πD/8	2.069
40x40	D/4	8 (surface mesh, degree = 2)	πD/16	1.912
80x80	D/8	16 (surface mesh, degree = 2)	πD/32	1.828
160x160	D/16	32 (surface mesh, degree = 2)	πD/64	1.686
320x320	D/32	32 (surface mesh, degree = 2)	πD/64	1.656
80x80	D/8	20 (volume mesh degree = 1)	πD/8	1.726
160x160	D/16	80 (volume mesh degree = 1)	πD/16	1.627
320x320	D/32	320 (volume mesh degree = 1)	πD/32	1.615

Table 16 IFVM and STAR-CCM+ mesh sensitivity analysis for the drag coefficient in the steady flow past a circular cylinder at $Re = 40$

IBM Fluid #cells	Δx	C_D	STAR-CCM+ Fluid #cells	Δx	C_D
200x200	D/10	1.646	200x200	D/10	1.605
400x400	D/20	1.667	500x500	D/25	1.610
800x800	D/40	1.588	1000x1000	D/50	1.611
1600x1600	D/80	1.587	//		//

In Table 17 is shown a comparison of the drag coefficient and recirculation length at $Re = 40$ with experimental data (see [85, 86, 87]) and other numerical results available in literature (see [13, 15, 19, 21, 23, 25, 29, 54, 64, 73, 79, 80, 88-103). Also for this case, the numerical results obtained with both the IFEM and IFVM are in excellent agreement with the experimental data. Good agreement with the other numerical IBM approaches and with the numerical results from STAR-CCM+ is achieved. For the IFEM, if the immersed body is discretized with a surface mesh using either bi-linear or bi-quadratic shape functions, the diffuse interpolation scheme predicts larger counter-rotating vortices in the cylinder wake. If the immersed body is discretized with a volume mesh, the presence of Lagrange multipliers in the fluid region inside the cylinder determines smaller counter-rotating vortices in the cylinder wake. Considering the experimental data of Tritton [86] as reference value, The IFEM with volume mesh approach and bi-linear shape functions gives an error on the drag coefficient of -2.298%. For the IFVM and STAR-CCM the error on the drag coefficient is equal to -3.992% and -2.541%, respectively.

Table 17 Comparison of the drag coefficient and recirculation length in the two-dimensional steady flow past a circular cylinder at $Re = 40$

	Year	Characteristics	l/d	C_D
Coutanceau and Bouard [85]	1977	Experiment	2.13	-
Tritton [86] – set #6	1959	Experiment	-	1.653
Wieselsberger [87]	1922	Experiment	-	1.700
Lima E Silva et al. [13]	2003	Implicit diffuse direct-forcing, NSEs	2.55	1.540
Tseng and Ferziger [15]	2003	Interior sharp direct-forcing, NSEs	2.21	1.530
Mittal et al. [19]	2008	Exterior sharp direct-forcing, NSEs	-	1.530
Ye et al. [21]	1999	Cut cell method, NSEs	2.27	1.520
Le et al. [23]	2006	IIM, NSEs	2.22	1.560
Le et al. [25]	2008	Implicit diffuse direct-forcing, NSEs	2.59	1.580
Kang and Hassan [29]	2010	Exterior sharp direct-forcing, LBE	2.25	1.538
Taira and Colonius [54]	2007	Implicit diffuse direct-forcing, NSEs	2.33	1.550
Kim et al. [64]	2000	Exterior sharp direct-forcing, NSEs	-	1.510
Linnick and Fasel [73]	2005	IIM, streamfunction-vorticity	2.23	1.610
Su et al. [79]	2007	Implicit diffuse direct-forcing, NSEs	-	1.630
Wu and Shu [80]	2009	Implicit diffuse direct-forcing, LBE	2.31	1.565
Dennis and Chang [88]	1970	Numerical solution	2.35	1.522
Fornberg [89]	1980	Numerical solution	2.24	1.500
Calhoun [90]	2002	Interior sharp direct-forcing, streamfunction-vorticity	2.18	1.620
Russel and Wang [91]	2003	Interior sharp direct-forcing, streamfunction-vorticity	2.29	1.600

Table 17 Continued

	Year	Characteristics	*l/d*	C_D
Niu et al. [92]	2006	Explicit sharp direct-forcing, LBE	2.26	1.589
He and Doolen [93]	1997	Implicit diffuse direct-forcing, LBE	2.25	1.499
Wang et al. [94]	2009	Implicit diffuse direct-forcing, NSEs	2.35	1.660
Park et al. [95]	1998	Body-fitted method, NSEs	-	1.510
Xu [96]	2008	IIM, NSEs	2.24	1.660
Xu and Wang [97]	2006	IIM, NSEs	2.21	1.660
Choi et al. [98]	2007	Interior sharp direct-forcing, NSEs	2.25	1.520
Rogers and Kwak [99]	1991	Body-fitted method, NSEs	2.29	1.549
Pacheco et al. [100]	2005	Exterior sharp direct-forcing, NSEs	2.28	1.530
Dias and Majumdar [101]	//	//	2.69	1.540
Marella et al. [102]	2005	//	2.30	1.520
Haeri and Shrimpton [103]	2012	//	2.23	1.580
Haeri and Shrimpton [103]	2012	non-DLM/FD, NSEs	2.27	1.620
Present (IFEM sm, dg = 1)	2012	Implicit diffuse direct-forcing, NSE	2.40	1.661
Present (IFEM sm, dg = 2)	2012	Implicit diffuse direct-forcing, NSE	2.35	1.656
Present (IFEM vm, dg = 1)	2012	Implicit diffuse direct-forcing, NSE	2.09	1.615
Present (IFVM)	2012	Explicit sharp direct-forcing, NSE	2.29	1.587
STAR-CCM+/V7.04.006	2012	Body-fitted method, NSE	2.40	1.611

In Figure 28 are shown the vorticity contours for the IFEM numerical simulation of the steady flow past a circular cylinder at $Re = 40$. For the immersed body,

simulations were performed considering a surface mesh approach and a volume mesh approach, respectively, and sensitivity over the shape functions used (i.e., bi-linear and bi-quadratic shape functions) were performed. In Figure 29 are shown the vorticity contours for the IFEM with volume mesh for the immersed body and bi-linear shape functions (left), the vorticity contours for the IFEM (center), and the vorticity contours obtained from STAR-CCM+ (right), respectively. The figure shows that the vorticity contours for both IBM approaches are in good agreement with the commercial code predictions, and in general with the vorticity contours reported in literature for steady flow past a circular cylinder at *Re* = 40.

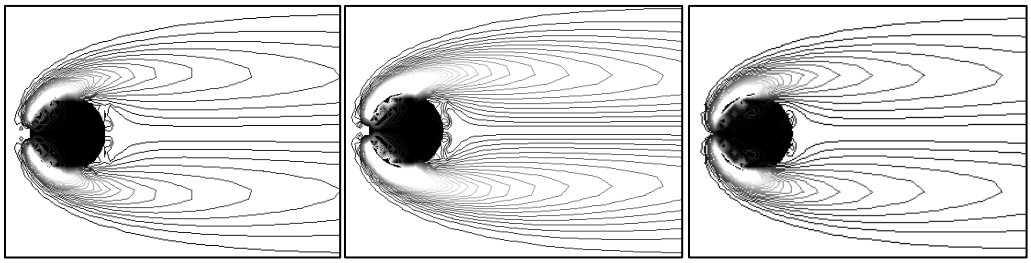

Figure 28. Vorticity contours for steady flow past a circular cylinder at *Re* = 40. IFEM; left: sm, dg = 1; center: sm, dg = 2; right: vm, dg = 1.

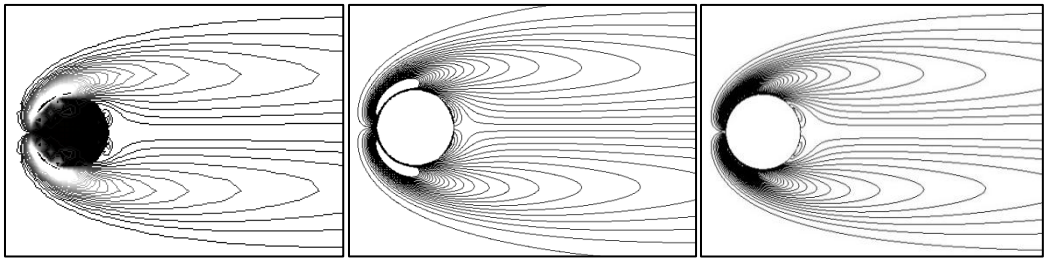

Figure 29. Vorticity contours for steady flow past a circular cylinder at *Re* = 40. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

In Figure 30 are shown the streamlines for the steady flow past a circular cylinder at *Re* = 40 determined with the direct-forcing IFEM and implicit diffuse interface scheme using surface mesh for the immersed body with bi-linear and bi-quadratic shape function, and volume mesh for the immersed body with bi-linear shape function. As discussed for the steady flow past a circular cylinder at *Re* = 20, the diffuse interface scheme allows some fluid to penetrate the cylinder surface, with the pair of counter-rotating vortices slightly shifted downwards in the cylinder wake (left and center). For the immersed body discretized with a volume mesh, the fluid inside the cylinder is at rest, and the streamlines do not cross the boundary of the cylinder. The pair of counter-rotating vortices in the wake are very close to the cylinder surface as shown in the left of Figure 31, and look very similar to the streamlines determined with the IFVM (center) and STAR-CCM+ (right). Due to the sharp nature of the interpolating functions at the immersed body surface, the no-slip boundary conditions is directly imposed in the IFVM approach and there is no fluid penetration inside the cylinder.

In Figure 32 is shown a comparison of the streamlines for *Re* = 40.3 (left) from the experimental data of Coutanceau and Bouard [85], the streamlines from the IFEM numerical simulations at *Re* = 40 (center) and the streamlines from the IFVM numerical simulations at *Re* = 40 (right). The numerical results show that the length of the symmetric counter-rotating vortices in the wake is qualitatively and quantitatively in good agreement with the experimental visualization. In Table 17 is shown that the recirculation length determined numerically is very close to the one determined in [85] with an error of -1.878% for the IFEM and 7.511% for the IFVM, respectively.

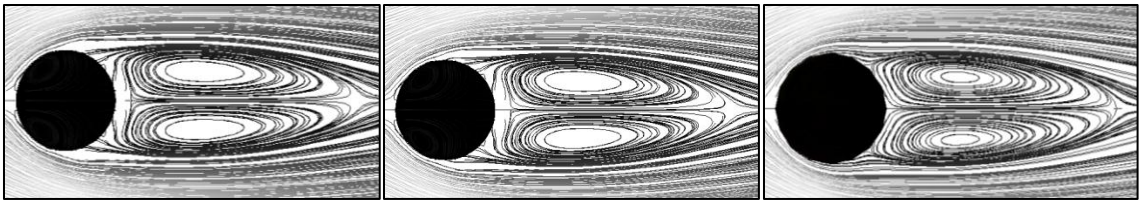

Figure 30. Streamlines for steady flow past a circular cylinder at *Re* = 40. IFEM. Left: sm, dg = 1; center: sm, dg = 2; right: vm, dg = 1.

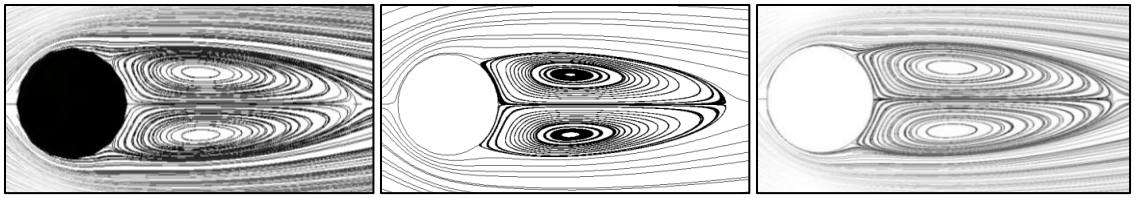

Figure 31. Streamlines for steady flow past a circular cylinder at *Re* = 40. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

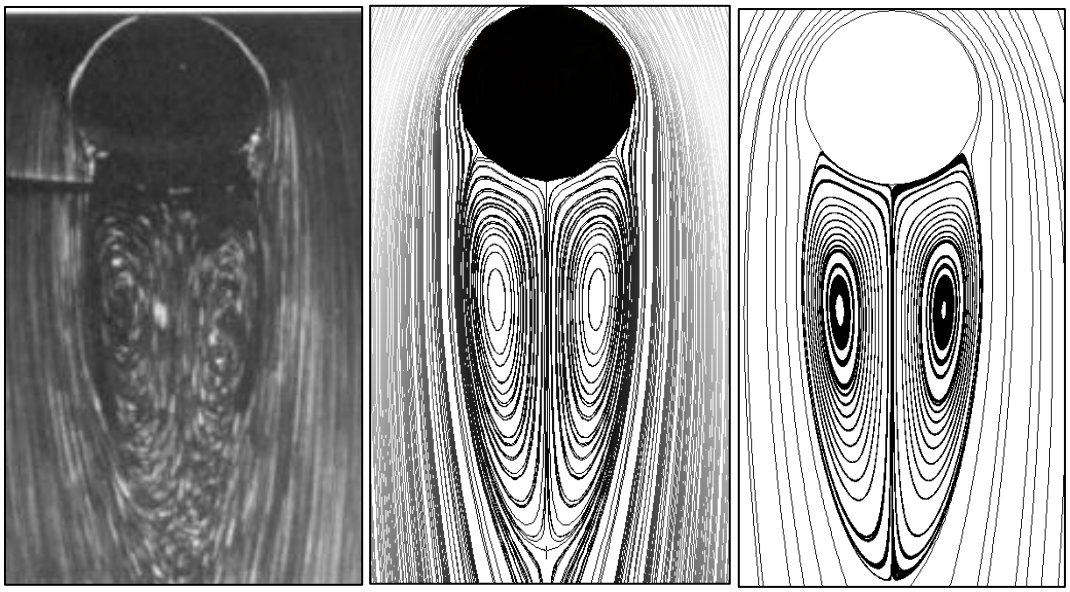

Figure 32. Streamlines for steady flow past a circular cylinder at *Re* = 40.3 from [85] (left); *Re* = 40 IFEM vm, dg = 1 (center); *Re* = 40 IFVM (right).

4.3.2 Two-Dimensional Unsteady Flow Past a Circular Cylinder

For the two-dimensional unsteady flows with *Re* = 100, 150 and 200, due to the extended recirculation zone in the cylinder wake, the computational domain for the IFEM simulations is extended: $\Omega = [-5D, 25D] \times [-10D, 10D]$, with the circular cylinder of diameter *D* being positioned 5*D* downstream the inlet and -0.1*D* off the axial symmetry line. This will introduce a small perturbation into the system, generating instability which will trigger unsteadiness. If the cylinder is positioned on the symmetry line, a much longer simulation time is required to trigger the instability. For the IFVM simulations the computational domain is: $\Omega = [-5D, 35D] \times [-10D, 10D]$, with the circular cylinder of diameter *D* positioned 5*D* downstream the inlet and -0.1*D* off the axial symmetry line. For the finite volume discretization, the instability is generated even if the cylinder is positioned on the symmetry line. For consistency with the IFEM simulations an offset of -0.1*D* was set. It is worth noting that the finite volume simulations required the outlet boundary condition to be imposed further downstream the cylinder location if compared to the finite element simulations. No converged results could be obtained with the finite volume discretization if the outlet boundary condition was set 25*D* downstream the cylinder.

Two new coefficients are introduced: the lift coefficient C_L and the Strouhal number *St*, which can be compared to the experimental and numerical data available in literature. The lift coefficient is defined as:

$$C_L = \frac{F_L}{\frac{1}{2}\rho U_\infty^2 D} \tag{144}$$

where F_L is the lift force on the cylinder and can be determined using eq. (30) for immersed body surface mesh approach:

$$F_L(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t)\cdot \underline{n}_y d\underline{x} = \int_{\underline{x}} \int_{\Gamma_i(t)} \underline{F}_K\left(\underline{X}(s,t)\right)\delta\left(\underline{X}-\underline{x}\right)ds\cdot \underline{n}_y d\underline{x} \tag{145}$$

and eq. (38) for immersed body volume mesh approach:

$$F_L(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t)\cdot \underline{n}_y d\underline{x} = \int_{\underline{x}} \int_{\omega(t)} \underline{F}_K\left(\underline{X}(s,t)\right)\delta\left(\underline{X}-\underline{x}\right)ds\cdot \underline{n}_y d\underline{x} \tag{146}$$

The Strouhal number is defined as:

$$S_t = \frac{f_q D}{U_\infty} \tag{147}$$

where U_∞ and D represent the free stream velocity and the cylinder diameter, respectively, and f_q is the vortex shedding frequency. The vortex shedding frequency can be obtained by the lift coefficient, due to its periodic oscillatory trend. Eq. (147) can be normalized using reference velocity $U_{\infty,0}$ and length scales D_0 as:

$$S_t = \frac{f_q D}{U_\infty} = \frac{f_q D_0}{U_{\infty,0}}\cdot\frac{D/D_0}{U_\infty/U_{\infty,0}} = \frac{f_q D_0}{U_{\infty,0}}\cdot\frac{D^*}{U_\infty^*} = \frac{f_q D_0}{U_{\infty,0}}\cdot\frac{1}{1} = \frac{1}{t_q\frac{U_{\infty,0}}{D_0}} = \frac{1}{t_q^*} = f_q^* \tag{148}$$

where f_q^* is the normalized vortex shedding frequency.

In Table 18, Table 19 and Table 20 are shown the mesh sensitivity for the drag and lift coefficients and the Strouhal number using the IFEM, the IFVM and the commercial code STAR-CCM+, respectively, for the unsteady flow past a circular cylinder at *Re* = 100. The tables show that for the mesh refinements chosen both the IBMs implemented and STAR-CCM+ computations give converged results in good agreement with each other.

Table 18 IFEM mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 100

Fluid #cells	Δx	IB #cells	Δs	C_D	C_L	*St*
120x80	D/8	20 (vm, dg = 1)	πD/8	1.483±0.011	±0.325	0.169
240x160	D/16	80 (vm, dg = 1)	πD/16	1.429±0.008	±0.322	0.179
480x320	D/32	320 (vm, dg = 1)	πD/32	1.425±0.008	±0.321	0.177

Table 19 IFVM mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 100

Fluid #cells	Δx	C_D	C_L	*St*
200x100	D/5	1.678±0.001	±0.008	0.121
400x200	D/10	1.586±0.008	±0.279	0.150
800x400	D/20	1.505±0.009	±0.315	0.167
1600x800	D/40	1.431±0.010	±0.325	0.174

Table 20 STAR-CCM+ mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 100

Fluid #cells	Δx	C_D	C_L	*St*
300x200	D/10	1.484±0.010	±0.337	0.175
1500x1000	D/50	1.458±0.010	±0.347	0.164
3000x2000	D/100	1.459±0.010	±0.354	0.171

Table 21 provides a comparison of the drag and lift coefficients and the Strouhal number for the IFEM, the IFVM, the commercial code STAR-CCM+, an extensive

literature of both experimental data (see [86, 104-106]) and numerical simulations performed with other IBM approaches (see [4, 13-15, 19, 23, 25, 29, 46, 64, 73, 79, 80, 90, 91, 94, 96, 97, 98, 100-103, 107-112]) and body fitted approaches (see [95, 108]). The numerical results obtained with the IFEM and IFVM approaches implemented are in good agreement with the experimental data available, the results obtained with other IBM approaches and the STAR-CCM+ numerical predictions. The drag coefficient average value determined with STAR-CCM+ is slightly larger than the value predicted with the IFEM and IFVM approaches and the data available from literature. Also for the lift coefficient STAR-CCM+ predicts a value larger than the IFEM and the IFVM numerical results. The Strouhal number predictions for the IFEM, the IFVM and STAR-CCM+ are in satisfactory agreement.

Table 21 Comparison of drag and lift coefficients and Strouhal number in the two-dimensional unsteady flow past a circular cylinder at Re = 100

	Year	Characteristics	C_D	C_L	St
Tritton [86] – set #8	1959	Experiment	1.257 (avg.)	-	-
Roshko [104]	1953	Experiment	-	-	0.164
Williamson [105]	1989	Experiment	-	-	0.166
Berger and Wille [106]	1972	Experiment	-	-	0.16-0.17
Lai and Peskin [4]	2000	Explicit diffuse feedback-forc., NSEs	1.447 (avg.)	±0.330	0.165
Lima E Silva et al. [13]	2003	Implicit diffuse direct-forcing, NSEs	1.390 (avg.)	-	0.160
Uhlmann [14]	2005	Explicit sharp direct-forcing, NSEs	1.453±0.011	±0.339	0.169
Tseng and Ferziger [15]	2003	Interior sharp direct-forcing, NSEs	1.420 (avg.)	±0.290	0.164
Mittal et al. [19]	2008	Exterior sharp direct-forcing, NSEs	1.350 (avg.)	-	0.165
Le et al. [23]	2006	IIM, NSEs	1.370±0.009	±0.323	0.160
Le et al. [25]	2008	Implicit diffuse direct-forcing, NSEs	1.390±0.009	±0.346	0.160
Kang and Hassan [29]	2010	Exterior sharp direct-forcing, LBE	1.336 (avg.)	±0.329	0.165
Apte et al. [46]	2009	non-DLM/FD, NSEs	1.360 (avg.)	-	0.160
Kim et al. [64]	2000	Exterior sharp direct-forcing, NSEs	1.330 (avg.)	±0.320	0.165
Linnick and Fasel [73]	2005	IIM, streamfunction-vorticity	1.380±0.010	±0.337	0.169
Su et al. [79]	2007	Implicit diffuse direct-forcing, NSEs	1.400 (avg.)	±0.340	0.168
Wu and Shu [80]	2009	Implicit diffuse direct-forcing, NSEs	1.364 (avg.)	±0.344	0.163
Calhoun [90]	2002	Interior sharp direct-forcing, streamf.-vortic.	1.330±0.014	±0.298	0.175
Russel and Wang [91]	2003	Interior sharp direct-forcing, streamf.-vortic.	1.380±0.007	±0.300	0.169
Wang et al. [94]	2009	Implicit diffuse direct-forcing, NSEs	1.379 (avg.)	±0.357	0.170

Table 21 Continued

	Year	Characteristics	C_D	C_L	*St*
Park et al. [95]	1998	Body-fitted method, NSEs	1.330 (avg.)	±0.330	0.165
Xu [96]	2008	IIM, NSEs	1.420±0.010	±0.353	0.172
Xu and Wang [97]	2006	IIM, NSEs	1.423±0.013	±0.340	0.171
Choi et al. [98]	2007	Interior sharp direct-forcing, NSEs	1.340 (avg.)	±0.315	0.164
Pacheco et al. [100]	2005	Exterior sharp direct-forcing, NSEs	1.410 (avg.)	-	0.167
Dias and Majumdar [101]	//	//	1.395 (avg.)	±0.283	0.171
Marella et al. [102]	2005	//	1.36 (avg.)	-	-
Haeri and Shrimpton [103]	2012	//	1.320 (avg.)	-	0.174
Haeri and Shrimpton [103]	2012	non-DLM/FD, NSEs	1.380 (avg.)	-	0.168
Braza et al. [107]	1986	Numerical solution	1.364±0.015	±0.250	-
Liu et al. [108]	1998	Body-fitted method, NSEs	1.350±0.012	±0.339	0.165
Sui et al. [109]	2007	Explicit diffuse direct-forcing, LBE	1.438 (avg.)	±0.344	0.166
Saiki et al. [110]	1996	Explicit diffuse feedback-forcing, NSEs	1.260 (avg.)	-	0.171
Engelman and Jaminia [111]	1990	Body-fitted method, NSEs	1.411 (avg.)	-	0.173
Ji et al. [112]	2012	Interior sharp direct-forcing, NSEs	1.376±0.010	0.339	0.169
Present (IFEM vm, dg = 1)	2011	Implicit diffuse direct-forcing, NSEs	1.425±0.008	±0.321	0.177
Present (IFVM)	2012	Explicit sharp direct-forcing, NSEs	1.431±0.010	±0.325	0.174
STAR-CCM+/V7.04.006	2012	Body-fitted method, NSEs	1.459±0.010	±0.354	0.171

In Figure 33 and Figure 34 are shown the time evolution of the drag and lift coefficient computed with the IFEM, the IFVM and STAR-CCM+, respectively. Following an initial transient where the periodic vortex shedding develops, both

methodologies are qualitatively and quantitatively in good agreement with the STAR-CCM+ results. The IFEM and IFVM predic a slightly larger vortex shedding frequency, and STAR-CCM+ giving slightly larger drag and lift coefficients (see Table 21).

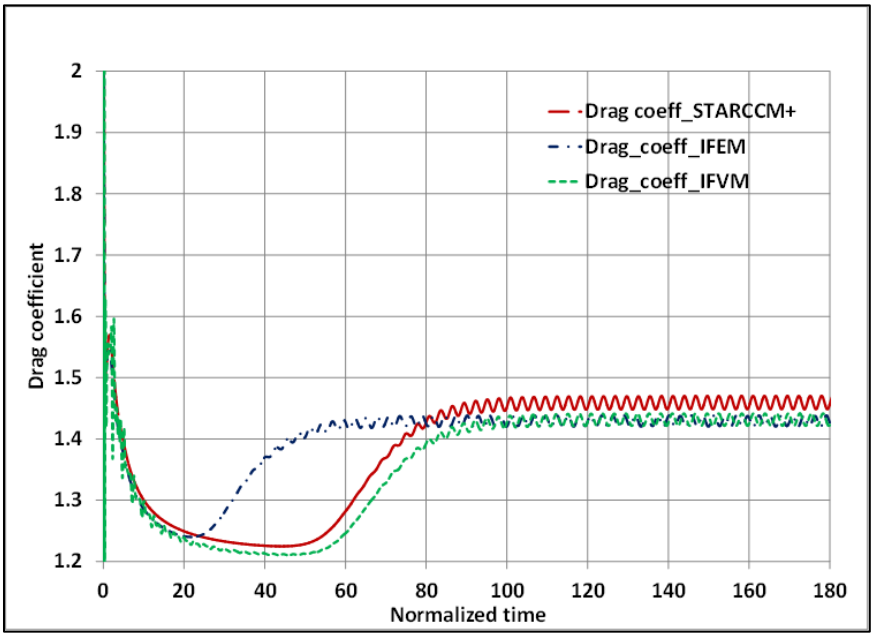

Figure 33. Time evolution of C_D for unsteady flow past a circular cylinder at Re = 100.

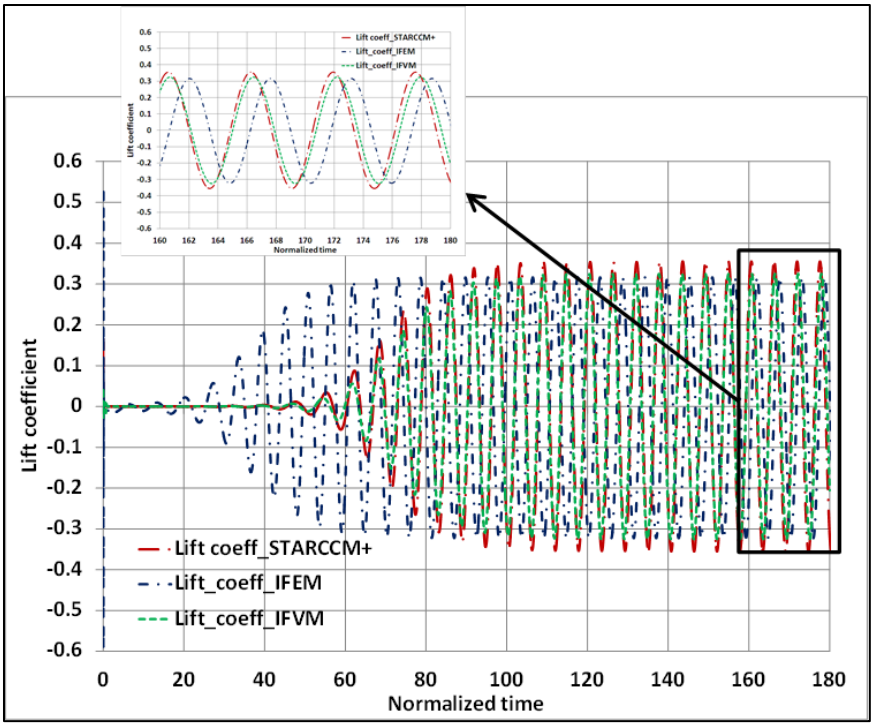

Figure 34. Time evolution of C_L for unsteady flow past a circular cylinder at *Re* = 100.

In Figure 35 and Figure 36 are shown the pressure distribution and velocity magnitude distribution close to the IB surface computed with the IFEM, IFVM and STAR-CCM+, respectively. The figures show that close to the IB the pressure and velocity distribution computed with the IFEM and IFVM approaches are qualitatively very similar to those obtained using the finite volume commercial code for a very fine mesh (6 million cells were used for the STAR-CCM+ simulation as shown in Table 20). The resemblance of the flow distribution obtained with the IFEM, the IFVM and that obtained with the commercial code STAR-CCCM+ can be noted as well from the velocity magnitude in the simulation domain considered shown in Figure 37 and the velocity and vorticity contours shown in Figure 38 and Figure 39, respectively.

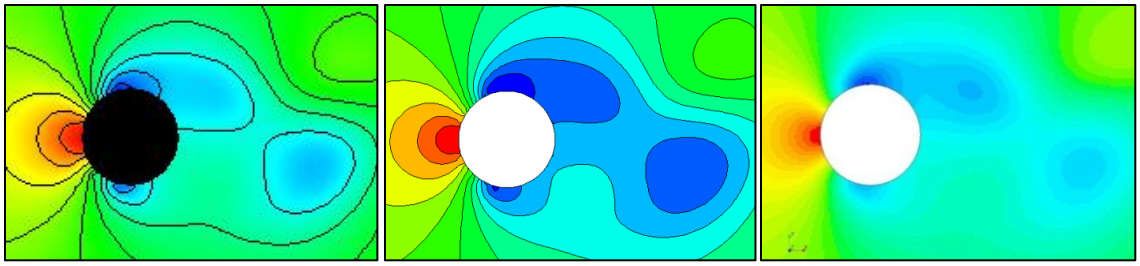

Figure 35. Pressure distribution for unsteady flow past a circular cylinder at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

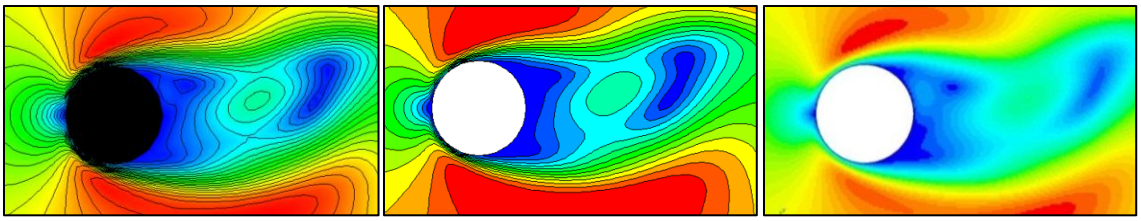

Figure 36. Velocity magnitude close to the IB surface for unsteady flow past a circular cylinder at *Re* = 100. Left: IB vm, dg = 1; right: STAR-CCM+.

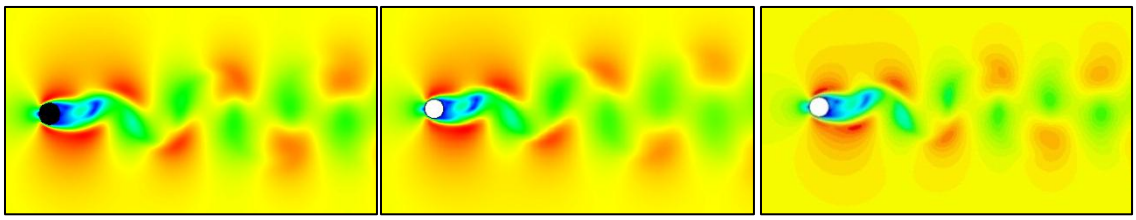

Figure 37. Velocity magnitude for unsteady flow past a circular cylinder at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

Figure 40 shows the streamlines determined with the IFEM, IFVM and those determined by STAR-CCM+ close to the IB surface. As pointed out previously, using a surface mesh to discretize the IB surface, some streamlines might penetrate the IB surface, due to the diffusive nature of the interpolation scheme. If a volume mesh is used

to discretize the immersed boundary, Lagrange multipliers are present not only on the surface of the immersed body, but also in the volume region inside the immersed body, and this guarantees no penetration of streamlines inside the cylinder boundary. The figure clearly shows that the IFEM and the IFVM properly simulate the vortex shedding at the cylinder surface.

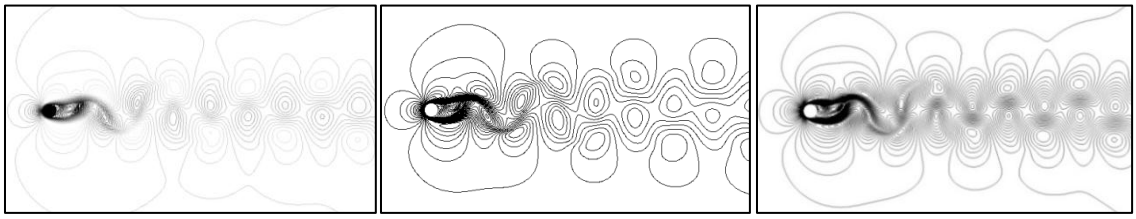

Figure 38. Velocity contours for unsteady flow past a circular cylinder at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

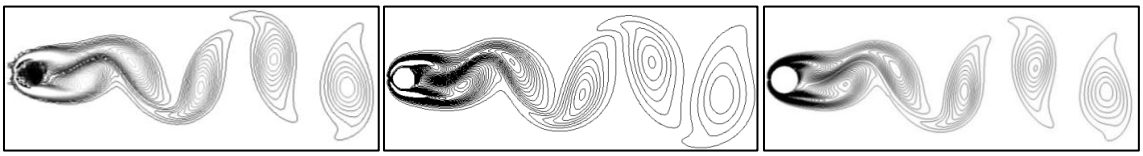

Figure 39. Vorticity contours for unsteady flow past a circular cylinder at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

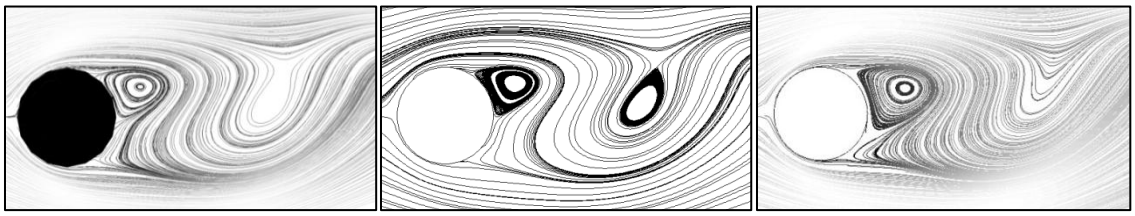

Figure 40. Streamlines for unsteady flow past a circular cylinder at *Re* = 100. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

In Table 22, Table 23 and Table 24 are shown the mesh sensitivity for the drag and lift coefficients and the Strouhal number using the IFEM, the IFVM and the commercial code STAR-CCM+, respectively, for the unsteady flow past a circular cylinder at *Re* = 150. The tables show that for the mesh refinements chosen the IFEM and the IFVM implemented give results in good agreement with the STAR-CCM+ predictions.

Table 22 IFEM mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 150

Fluid #cells	Δx	Immersed boundary #cells	Δs	C_D	C_L	*St*
120x80	D/8	20 (vm, dg = 1)	πD/8	1.627±0.030	±0.583	0.193
240x160	D/16	80 (vm, dg = 1)	πD/16	1.400±0.025	±0.518	0.199
480x320	D/32	320 (vm, dg = 1)	πD/32	1.395±0.021	±0.508	0.195

Table 23 IFVM mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 150

Fluid #cells	Δx	C_D	C_L	*St*
200x100	D/5	1.643±0.001	±0.071	0.124
400x200	D/10	1.566±0.013	±0.324	0.155
800x400	D/20	1.484±0.023	±0.448	0.177
1600x800	D/40	1.405±0.026	±0.513	0.193

Table 24 STAR-CCM+ mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at Re = 150

Fluid #cells	Δx	C_D	C_L	St
300x200	D/10	1.415±0.022	±0.458	0.194
1500x1000	D/50	1.431±0.028	±0.551	0.177
3000x2000	D/100	1.434±0.029	±0.563	0.195

Table 25 provides a comparison of the drag and lift coefficients and the Strouhal number for the IFEM, the IFVM, the commercial code STAR-CCM+, an extensive literature of both experimental data (see [104, 105]) and numerical simulations performed with other IBM approaches (see [4, 13, 29, 79, 109, 112]) and body fitted approaches (see [108]) for the unsteady flow past a circular cylinder at Re = 150. The numerical results obtained with the IFEM and the IFVM approaches implemented are in good agreement with the experimental data available, the results obtained with other IBM approaches and the STAR-CCM+ numerical predictions.

Table 25 Comparison drag and lift coefficients and Strouhal number in the two-dimensional unsteady flow past a circular cylinder at *Re* = 150

	Year	Characteristics	C_D	C_L	*St*
Roshko [104]	1953	Experiment	-	-	0.182
Williamson [105]	1989	Experiment	-	-	0.183
Lai and Peskin [4]	2000	Explicit diffuse feedback-forcing, NSEs	1.440 (avg.)	-	0.184
Lima E Silva et al. [13]	2003	Implicit diffuse direct-forcing, NSEs	1.370 (avg.)	-	0.180
Kang and Hassan [29]	2010	Exterior sharp direct-forcing, LBE	1.312 (avg.)	±0.513	0.184
Su et al. [79]	2007	Implicit diffuse direct-forcing, NSEs	1.390 (avg.)	-	0.187
Liu et al. [108]	1998	Body-fitted method, NSE	1.334±0.012	±0.530	0.182
Sui et al. [109]	2007	Explicit diffuse direct-forcing, LBE	1.449 (avg.)	±0.709	0.197
Ji et al. [112]	2012	Interior sharp direct-forcing, NSEs	1.354±0.026	±0.524	0.188
Present (IFEM vm, dg=1)	2012	Implicit diffuse direct-forcing, NSEs	1.395±0.021	±0.508	0.195
Present (IFVM)	2012	Explicit sharp direct-forcing, NSEs	1.405±0.026	±0.513	0.193
STAR-CCM+/V7.04.006	2012	Body-fitted method, NSEs	1.434±0.029	±0.563	0.195

In Figure 41 and Figure 42 are shown the time evolution of the drag and lift coefficient computed with the IFEM, the IFVM and STAR-CCM+, respectively, for the unsteady flow past a circular cylinder at *Re* = 150. Following an initial transient where the periodic vortex shedding develops, both immersed boundary methodologies are qualitatively and quantitatively in good agreement with STAR-CCM+ numerical data, with the IFEM predicting a slightly larger vortex shedding frequency, and STAR-CCM+ giving slightly larger drag and lift coefficients.

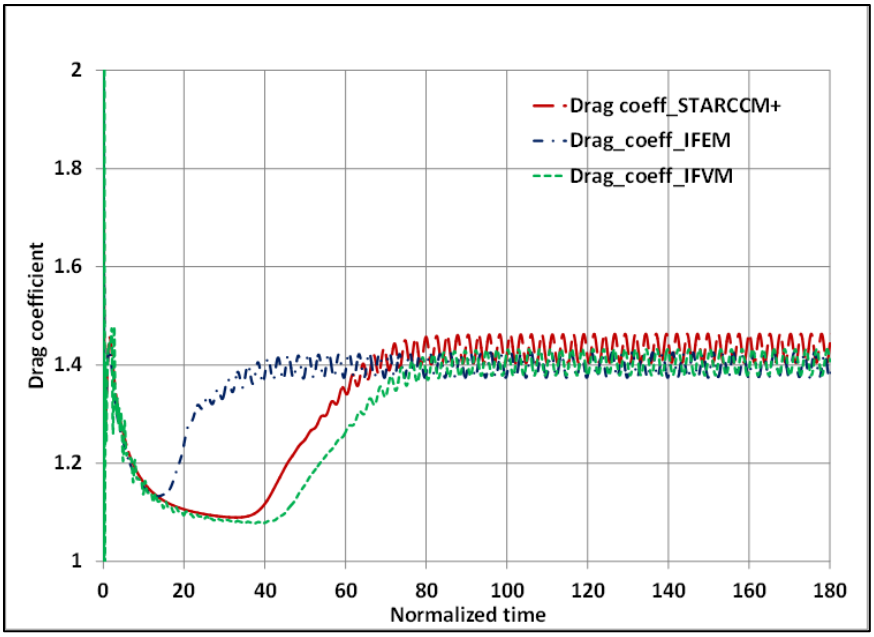

Figure 41. Time evolution of C_D for unsteady flow past a circular cylinder at Re = 150.

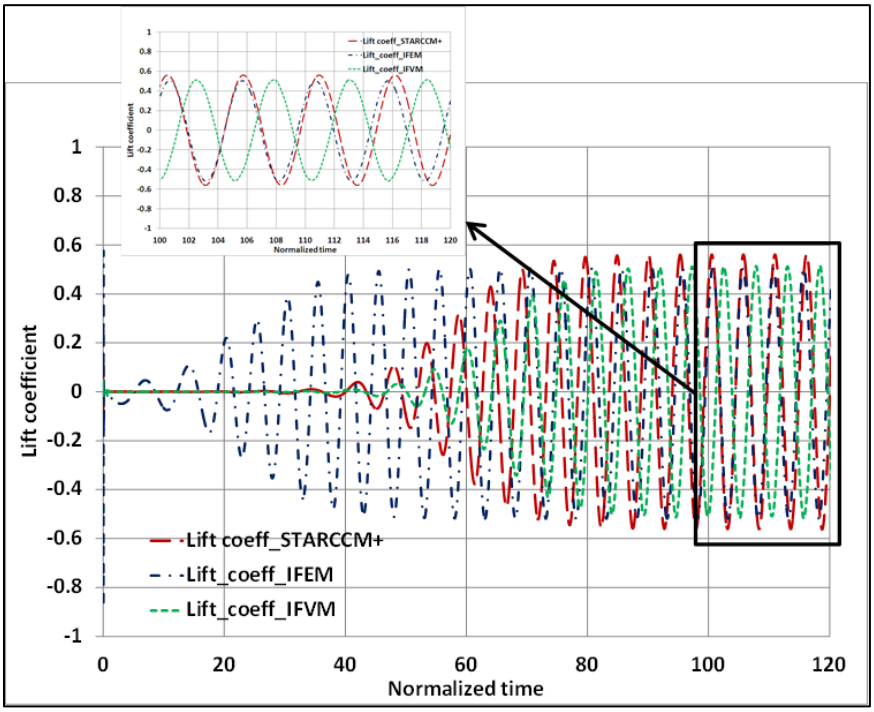

Figure 42. Time evolution of C_L for unsteady flow past a circular cylinder at Re = 150.

In Figure 43 and Figure 44 are shown the pressure distribution and velocity magnitude distribution close to the IB surface computed with the IFEM, the IFVM and STAR-CCM+, respectively. The figures show that close to the IB the pressure and velocity distribution computed with the IFEM and the IFVM approaches are qualitatively very similar to those obtained using the finite volume commercial code for a very fine mesh (6 million cells were used for the STAR-CCM+ simulation as shown in Table 24).

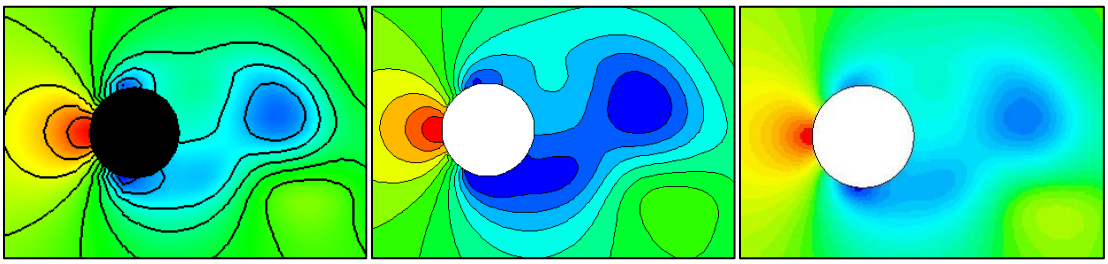

Figure 43. Pressure distribution for unsteady flow past a circular cylinder at *Re* = 150. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

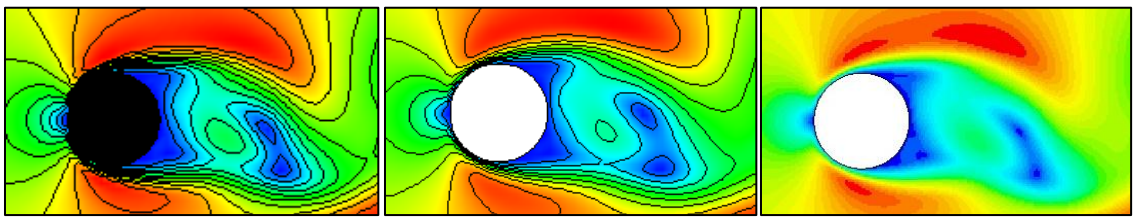

Figure 44. Velocity magnitude close to the IB surface for unsteady flow past a circular cylinder at *Re* = 150. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

The resemblance of the flow distribution obtained with the IFEM and the IFVM to that obtained with the commercial code STAR-CCCM+ can be noted as well from the

velocity magnitude in the simulation domain considered shown in Figure 45 and the velocity and vorticity contours shown in Figure 46 and Figure 47, respectively.

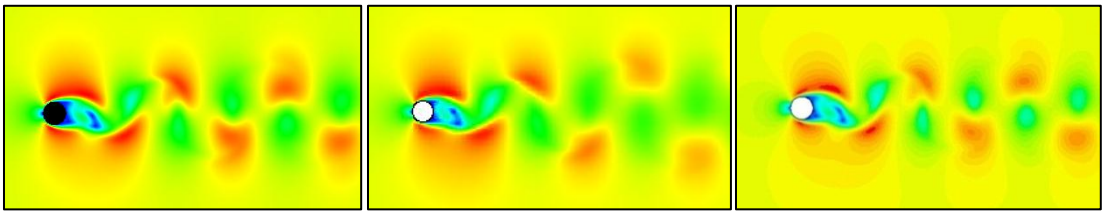

Figure 45. Velocity magnitude for unsteady flow past a circular cylinder at *Re* = 150. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

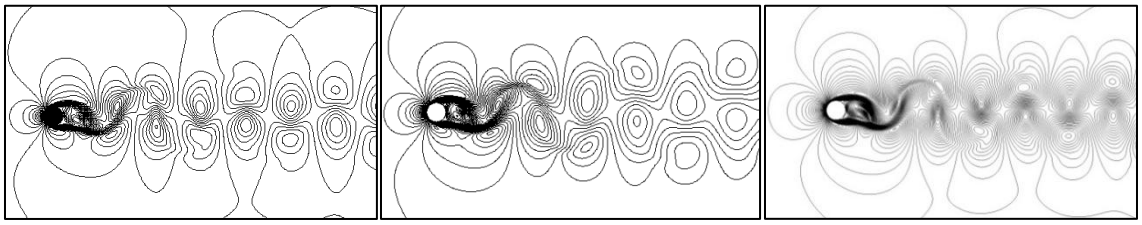

Figure 46. Velocity contours for unsteady flow past a circular cylinder at *Re* = 150. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

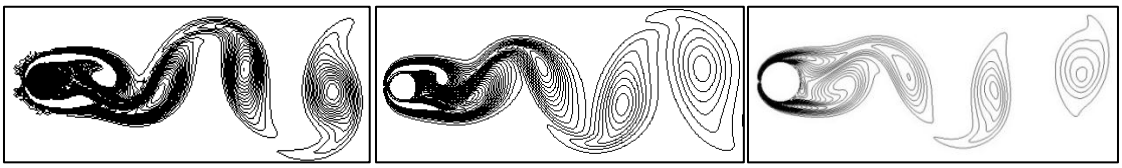

Figure 47. Vorticity contours for unsteady flow past a circular cylinder at *Re* = 150. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

Figure 48 shows the streamlines determined with the IFEM, the IFVM and those determined by STAR-CCM+ close to the IB surface. As pointed out previously, using a

surface mesh to discretize the IB surface for the IFEM, some streamlines might penetrate the IB surface, due to the diffusive nature of the interpolation scheme. If a volume mesh is used to discretize the immersed boundary for the IFEM, Lagrange multipliers are present not only on the surface of the immersed body, but also in the volume region inside the immersed body, and this guarantees no penetration of streamlines inside the cylinder boundary. The figure shows that the IFEM and the IFVM properly simulate the vortex shedding in the cylinder wake, and the agreement with the streamlines from STAR-CCM+ is clearly visible.

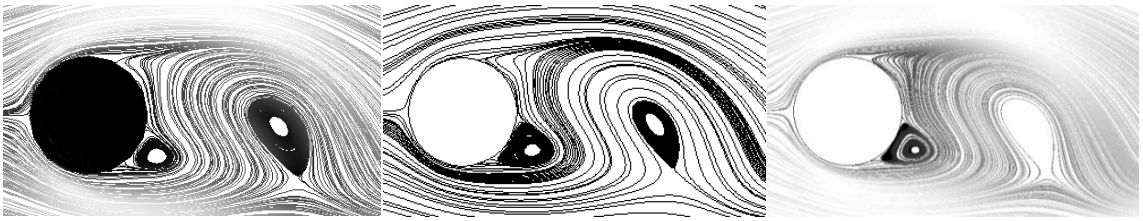

Figure 48. Streamlines for unsteady flow past a circular cylinder at *Re* = 150. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

In Table 26, Table 27 and Table 28 are shown the mesh sensitivity for the drag and lift coefficients and the Strouhal number using the IFEM, the IFVM and the commercial code STAR-CCM+, respectively, for the unsteady flow past a circular cylinder at *Re* = 200. The tables show that for the mesh refinement chosen both the IBM implemented and the finite volume computations give converged results in good agreement with each other.

Table 26 IFEM mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 200

Fluid #cells	Δx	Immersed boundary #cells	Δs	C_D	C_L	*St*
120x80	D/8	20 (vm, dg = 1)	πD/8	1.668±0.049	±0.696	0.197
240x160	D/16	80 (vm, dg = 1)	πD/16	1.413±0.045	±0.682	0.211
480x320	D/32	320 (vm, dg = 1)	πD/32	1.401±0.042	±0.675	0.205

Table 27 IFVM mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 200

Fluid #cells	Δx	C_D	C_L	*St*
200x100	D/5	1.620±0.001	±0.580	0.124
400x200	D/10	1.549±0.016	±0.330	0.157
800x400	D/20	1.487±0.032	±0.515	0.182
1600x800	D/40	1.407±0.041	±0.648	0.202

Table 28 STAR-CCM+ mesh sensitivity analysis for drag and lift coefficients and Strouhal number in the unsteady flow past a circular cylinder at *Re* = 200

Fluid #cells	Δx	C_D	C_L	*St*
300x200	D/10	1.344±0.029	±0.502	0.206
1500x1000	D/50	1.440±0.050	±0.681	0.203
3000x2000	D/100	1.443±0.050	±0.743	0.207

Table 29 provides a comparison of the drag and lift coefficients and the Strouhal number for the IFEM, the IFVM, the commercial code STAR-CCM+, an extensive

literature of numerical simulations performed with other IBM approaches (see [4, 23, 25, 73, 80, 90, 91, 94, 96, 97, 98, 107, 109, 110, 112]) and body fitted approaches (see [108, 113-115]). The numerical results obtained with the IFEM and IFVM approaches implemented are in good agreement with the experimental data available, the results obtained with other IBM approaches and the STAR-CCM+ numerical predictions.

Table 29 Comparison for drag and lift coefficients and Strouhal number in the two-dimensional unsteady flow past a circular cylinder at $Re = 200$

	Year	Characteristics	C_D	C_L	St
Lai and Peskin [4]	2000	Explicit diffuse feedback-forcing, NSEs	-	-	0.190
Le et al. [23]	2006	IIM, NSEs	1.340±0.030	±0.430	0.200
Le et al. [25]	2008	Implicit diffuse direct-forcing, NSEs	1.380±0.040	±0.676	0.208
Linnick and Fasel [73]	2005	IIM, streamfunction-vorticity	1.370±0.046	±0.700	0.199
Wu and Shu [80]	2009	Implicit diffuse direct-forcing, NSEs	1.349 (avg.)	-	0.193
Calhoun [90]	2002	Interior sharp direct-forcing, streamf.-vortic.	1.172±0.058	±0.668	0.202
Russel and Wang [91]	2003	Interior sharp direct-forcing, streamf.-vortic.	1.290±0.022	±0.500	0.195
Wang et al. [94]	2009	Implicit diffuse direct-forcing, NSEs	1.262 (avg.)	±0.708	0.195
Xu [96]	2008	IIM, NSEs	1.430±0.050	±0.710	0.202
Xu and Wang [97]	2006	IIM, NSEs	1.420±0.040	±0.660	0.202
Choi et al. [98]	2007	Interior sharp direct-forcing, NSEs	1.360±0.048	±0.640	0.191
Braza et al. [107]	1986	Numerical solution	1.400±0.050	±0.750	-
Liu et al. [108]	1998	Body-fitted method, NSEs	1.310±0.049	±0.690	0.192
Sui et al. [109]	2007	Explicit diffuse direct-forcing, LBE	1.449 (avg.)	±0.708	0.197

Table 29 Continued

	Year	Characteristics	C_D	C_L	St
Saiki et al. [110]	1996	Explicit diffuse feedback-forcing, NSEs	1.180 (avg.)	-	0.197
Ji et al. [112]	2012	Interior sharp direct-forcing, NSEs	1.354±0.044	±0.681	0.200
Belov et al. [113]	1995	Body-fitted method, NSEs	1.190±0.042	±0.640	0.193
Rosefeld et al. [114]	1991	Body-fitted method, NSEs	1.310±0.040	±0.650	0.200
Wright and Smith [115]	2001	Body-fitted method, NSEs	1.330±0.040	±0.680	0.196
Present (IFEM vm, dg = 1)	2011	Implicit diffuse direct-forcing, NSEs	1.401±0.042	±0.675	0.205
Present (IFVM)	2012	Explicit sharp direct-forcing, NSEs	1.407±0.041	±0.648	0.202
STAR-CCM+/V7.04.006	2011	Body-fitted method, NSEs	1.443±0.050	±0.743	0.207

In Figure 49 and Figure 50 are shown the time evolution of the drag and lift coefficient computed with the IFEM, the IFVM and STAR-CCM+, respectively. Following an initial transient where the periodic vortex shedding develops, both methodologies are qualitatively and quantitatively in good agreement with STAR-CCM+ numerical results, with the IFEM and IFVM predicting a slightly larger vortex shedding frequency, and STAR-CCM+ giving slightly larger drag and lift coefficients.

In Figure 51 and Figure 52 are shown the pressure distribution and velocity magnitude distribution close to the IB surface computed with the IFEM, the IFVM and STAR-CCM+, respectively. The figures show that close to the IB the pressure and velocity distribution computed with the IFEM and IFVM approaches are qualitatively very similar to those obtained using STAR-CCM+.

The resemblance of the flow distribution obtained with the IFEM, the IFVM and that obtained with the commercial code STAR-CCCM+ can be noted as well from the velocity magnitude in the simulation domain considered shown in Figure 53 and the velocity and vorticity contours shown in Figure 54 and Figure 55, respectively.

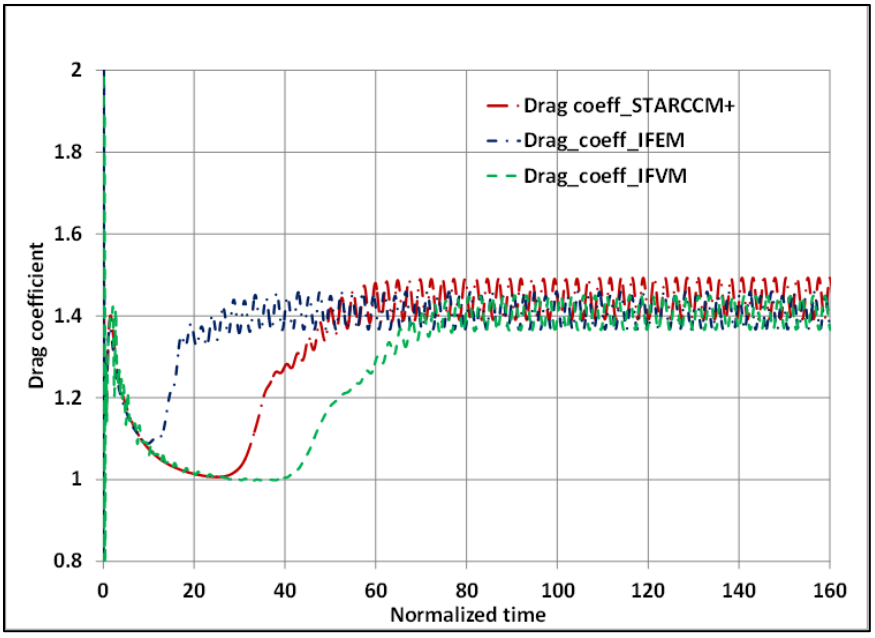

Figure 49. Time evolution of C_D for unsteady flow past a circular cylinder at Re = 200.

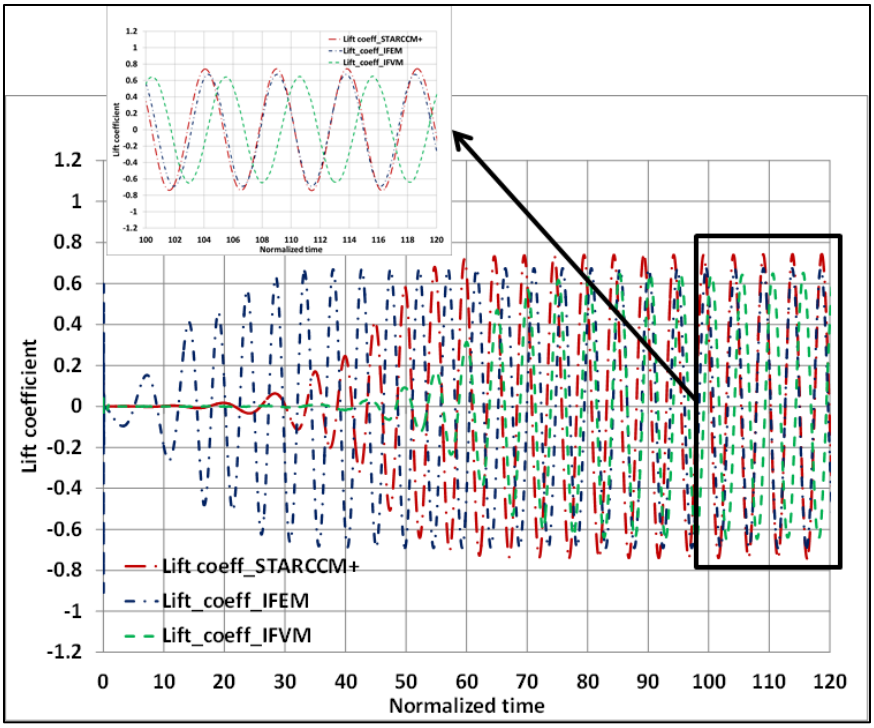

Figure 50. Time evolution of C_L for unsteady flow past a circular cylinder at Re = 200

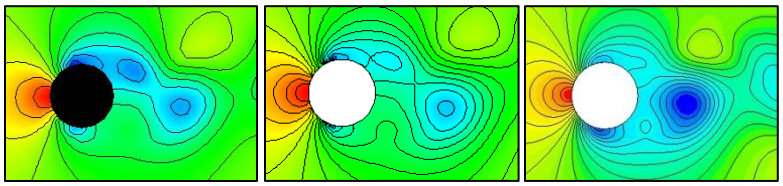

Figure 51. Pressure distribution for unsteady flow past a circular cylinder at Re = 200. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

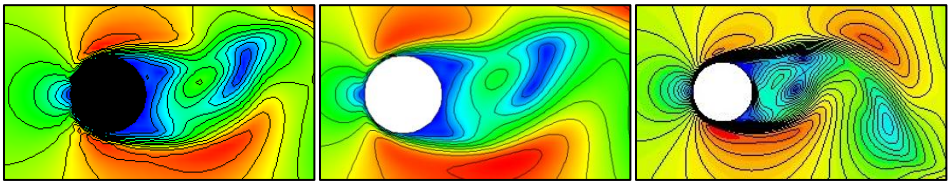

Figure 52. Velocity magnitude close to the IB surface for unsteady flow past a circular cylinder at *Re* = 200. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

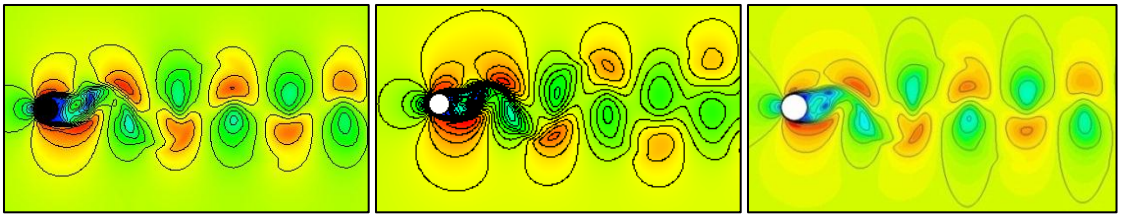

Figure 53. Velocity magnitude for unsteady flow past a circular cylinder at *Re* = 200. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

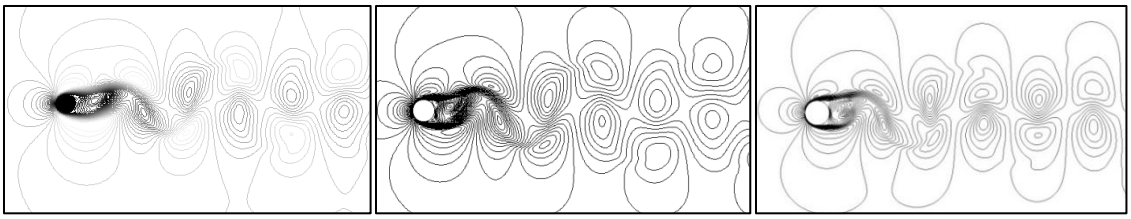

Figure 54. Velocity contours for unsteady flow past a circular cylinder at *Re* = 200. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

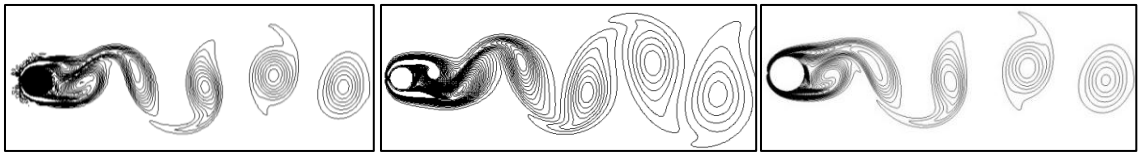

Figure 55. Vorticity contours for unsteady flow past a circular cylinder at *Re* = 200. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

Figure 56 shows the streamlines determined with the IFEM, the IFVM and those determined by STAR-CCM+ close to the IB surface. As pointed out previously, using a surface mesh to discretize the IB surface, some streamlines might penetrate the IB surface, due to the diffusive nature of the interpolation scheme. If a volume mesh is used to discretize the immersed boundary, Lagrange multipliers are present not only on the surface of the immersed body, but also in the volume region inside the immersed body,

and this guarantees no penetration of streamlines inside the cylinder boundary. Figure 56 shows that the IFEM and the IFVM properly simulate the vortex shedding at the cylinder surface, and the results from both immersed boundary methodologies are qualitatively in good agreement with STAR-CCM+ predictions.

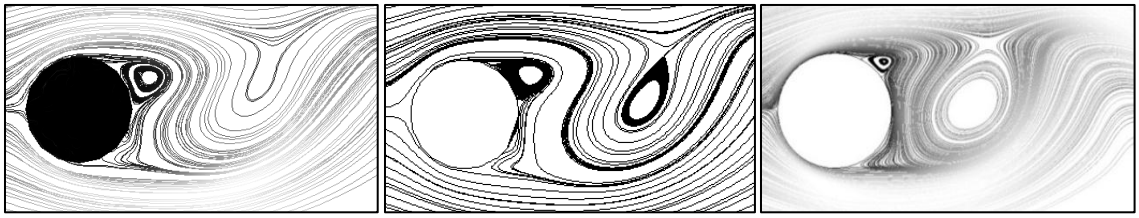

Figure 56. Streamlines for unsteady flow past a circular cylinder at *Re* = 200. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

4.4 Three-Dimensional Flow Past a Sphere

To test the implemented immersed boundary methodologies against three-dimensional problems, the laminar flow past a sphere was simulated next. Simulations were performed at *Re* = 50, 100, 150, 200, 250 and 300, where the Reynolds number is measured based on the sphere diameter *D*, the free stream velocity U_∞, and the fluid kinematic viscosity ν ($Re = U_\infty \bullet D/\nu$). For $Re \leq 200$, the flow regime is steady axisymmetric. For $210 \leq Re \leq 270$ the flow regime becomes steady nonaxisymmetric, and for $280 \leq Re$ the flow regime becomes unsteady. The computational domain for both finite element and finite volume discretization is: $\Omega = [0, 20D] \times [0, 10D] \times [0,10D]$, where *D* is the sphere diameter *D* = 1.0. The sphere is centered at (10.0, 5.0, 5.0). Free stream velocity (U_∞ = 1.0, *V* = 0.0, *W* = 0.0) conditions were specified at the inlet, north

and south boundaries. Homogeneous Neumann boundary conditions were imposed at the top, bottom and exit boundaries. The time step for all simulations was $\Delta t = 0.01$.

For $Re \leq 200$ the flow gradually reaches steady state conditions and the wake behind the sphere is axisymmetric and topologically similar as addressed in Johnson and Patel [116].

In Figure 57, Figure 58, Figure 59 and Figure 60 are shown the streamlines on the (x, y)-plane for the three-dimensional flow past a sphere at Reynolds number Re = 50, 100, 150 and 200, obtained with the IFVM (left) and STAR-CCM+ (right), respectively. The figures show that the flow in the wake separates from the surface of the sphere, forming a closed recirculation zone. For these low Reynolds number the flow is axisymmetric and at steady state conditions. Only the length of the recirculation zone and the angle of separation change with the Re number. Both the IFVM and STAR-CCM+ show the appearance of minor three-dimensional effects for $Re \leq 200$ as Figure 57 through Figure 60 point out. This behavior is due to the relatively coarse mesh used to discretize the immersed body for both approaches. If a more refined mesh for the sphere is used, symmetry is reestablished in the simulations and these three-dimensional disturbances become negligible.

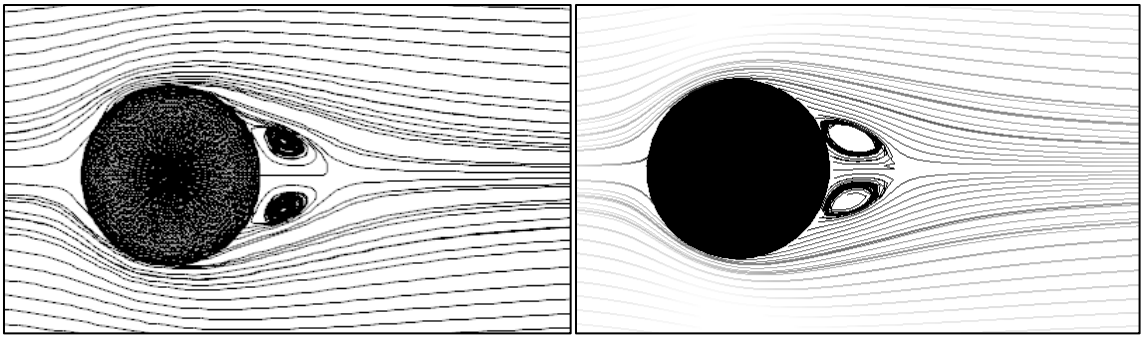

Figure 57. Streamlines on the (*x*,*y*)-plane for flow past a sphere at *Re* = 50: Left: IFVM; right: STAR-CCM+.

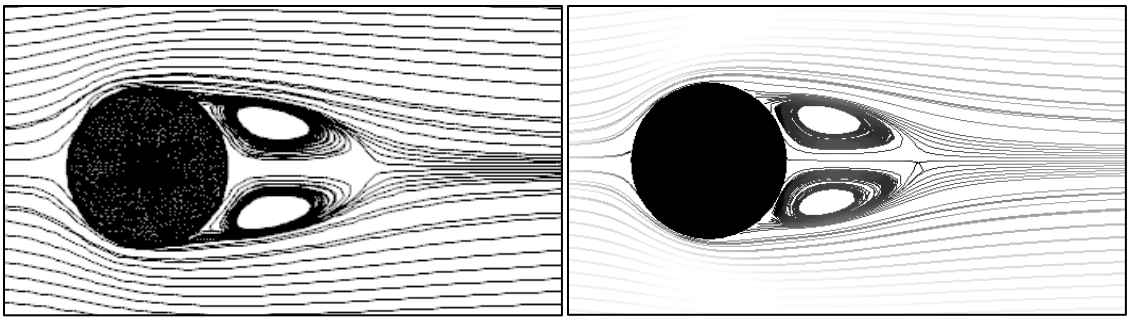

Figure 58. Streamlines on the (*x*,*y*)-plane for flow past a sphere at *Re* = 100: Left: IFVM; right: STAR-CCM+.

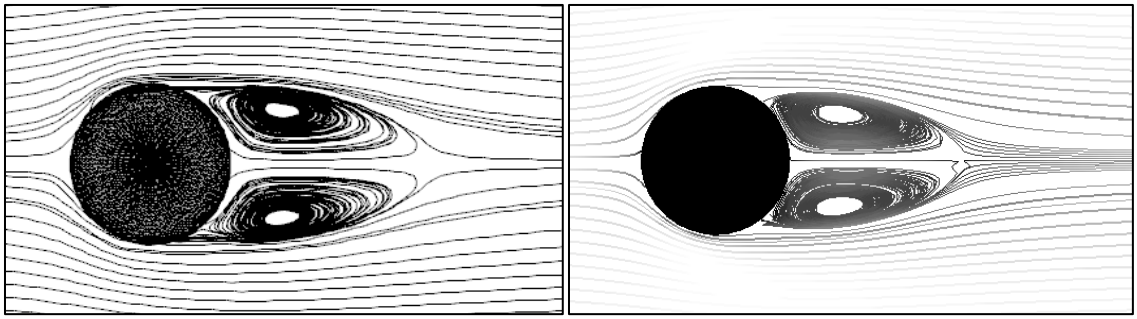

Figure 59. Streamlines on the (*x*,*y*)-plane for flow past a sphere at *Re* = 150: Left: IFVM; right: STAR-CCM+.

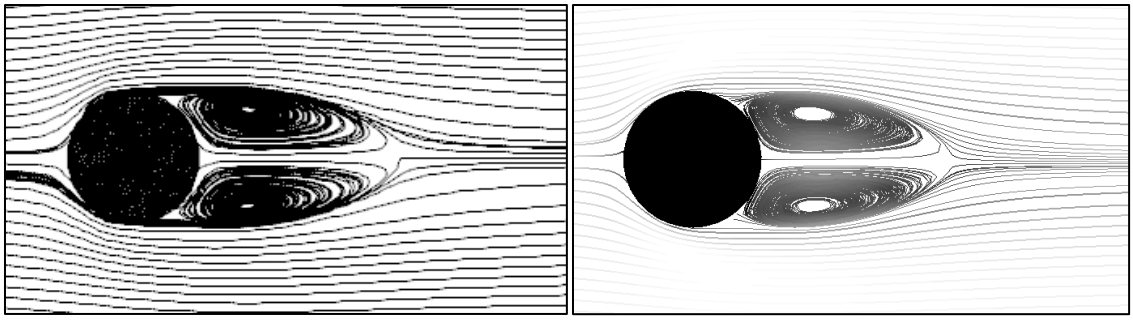

Figure 60. Streamlines on the (*x*,*y*)-plane for flow past a sphere at *Re* = 200: Left: IFVM; right: STAR-CCM+.

In Figure 61, Figure 62, Figure 63 and Figure 64 are shown the vorticity contours on the (*x*, *y*)-plane for the three-dimensional flow past a sphere at Reynolds number *Re* = 50, 100, 150 and 200, obtained with the IFVM (left) and STAR-CCM+ (right), respectively. From the figures it is possible to see that the flow distribution in the sphere wake determined with the IFVM approach is qualitatively similar to the one determined with the commercial code STAR-CCM+. Due to the very coarse mesh used for the IFEM method, the streamlines and vorticity contours form the IFEM were not presented.

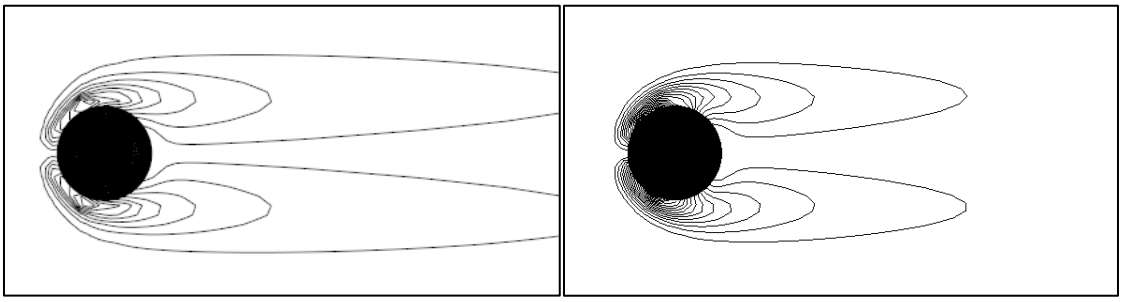

Figure 61. Vorticity contours on the (*x*,*y*)-plane for flow past a sphere at *Re* = 50: Left: IFVM; right: STAR-CCM+.

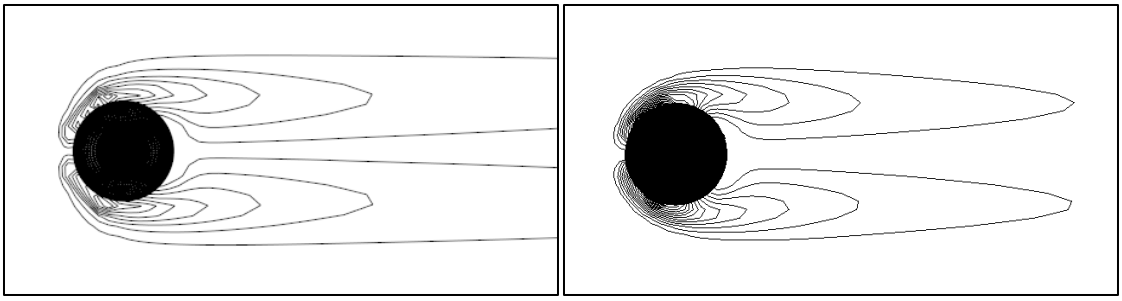

Figure 62. Vorticity contours on the (x,y)-plane for flow past a sphere at Re = 100: Left: IFVM; right: STAR-CCM+.

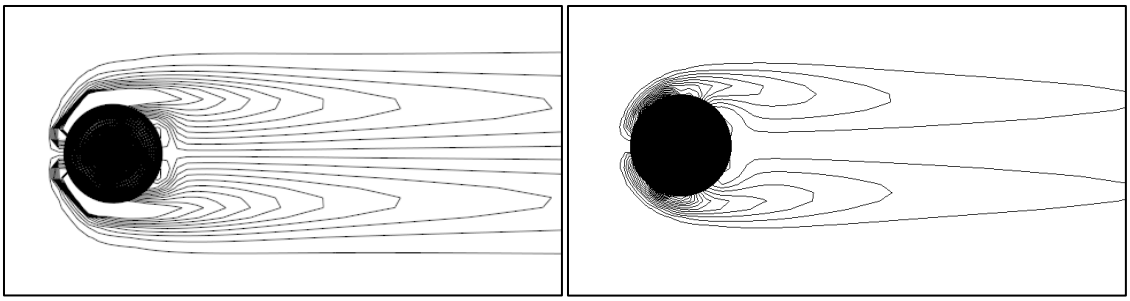

Figure 63. Vorticity contours on the (x,y)-plane for flow past a sphere at Re = 150: Left: IFVM; right: STAR-CCM+.

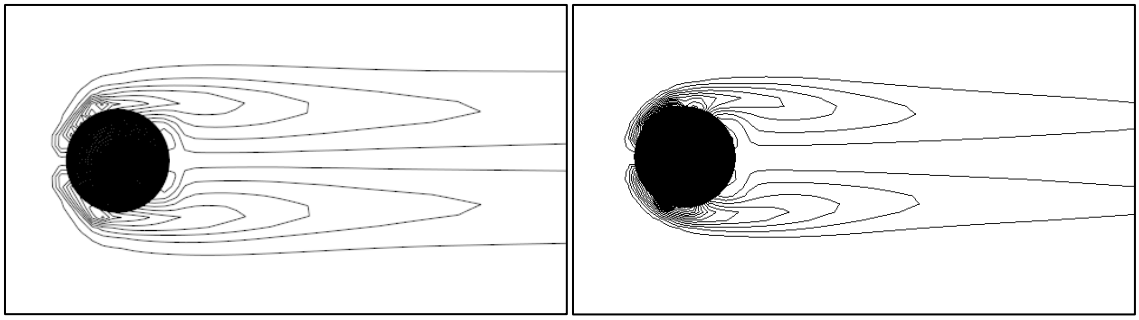

Figure 64. Vorticity contours on the (x,y)-plane for flow past a sphere at Re = 200: Left: IFVM; right: STAR-CCM+.

For a three-dimensional flow past a sphere the drag coefficient C_D is defined as:

$$C_D = \frac{F_D}{\frac{1}{2}\rho U_\infty^2 \pi \frac{D^2}{4}} \tag{149}$$

where ρ, U_∞ and D represent the fluid density, the fluid free stream velocity and the sphere diameter, respectively. F_D is the drag force on the sphere and can be determined using eq. (30) for immersed body surface mesh approach:

$$F_D(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t) \cdot \underline{n}_x d\underline{x} = \int_{\underline{x}} \int_{\Gamma_i(t)} \underline{F}_K\left(\underline{X}(s,t)\right) \delta\left(\underline{X} - \underline{x}\right) ds \cdot \underline{n}_x d\underline{x} \tag{150}$$

and eq. (38) for immersed body volume mesh approach:

$$F_D(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t) \cdot \underline{n}_x d\underline{x} = \int_{\underline{x}} \int_{\omega(t)} \underline{F}_K\left(\underline{X}(s,t)\right) \delta\left(\underline{X} - \underline{x}\right) ds \cdot \underline{n}_x d\underline{x} \tag{151}$$

In Table 30 are shown the numerical results for the drag coefficient at *Re* = 50, 100, 150 and 200 for the IFEM, and in Table 31 and Table 32 are presented the drag coefficient results at the same Reynolds numbers for the IFVM and STAR-CCM+, respectively. Also for the three-dimensional flow past a sphere the two immersed boundary methodologies give results in good agreement with the body-fitted commercial code results. The deviation in the IFEM results is due to the coarse mesh used for the finite element approach simulations. Due to the very large computational effort required by the IFEM for three-dimensional simulations, it was not possible to further refine the mesh for this test case using the IFEM approach. The IFVM results for the drag coefficient are in excellent agreement with the STAR-CCM+ predictions. Also for the three-dimensional flow over a sphere the IFEM shows a monotone decreasing drag coefficient with mesh refinement, meanwhile the two finite volume approaches have an opposite behavior, with the drag coefficient increasing with mesh refinement.

Table 30 IFEM mesh sensitivity analysis for drag coefficient in the three-dimensional flow past a sphere at *Re* = 50, 100, 150 and 200

Fluid #cells)	Δx	IB #cells	C_D			
			Re = 50	*Re = 100*	*Re = 150*	*Re = 200*
40x20x20	D/4	56	3.358	2.585	2.306	2.105
80x40x40	D/8	56	1.694	1.193	1.029	0.975

Table 31 IFVM mesh sensitivity analysis for drag coefficient in the three-dimensional flow past a sphere at *Re* = 50, 100, 150 and 200

		C_D			
Fluid #cells	Δx	*Re* = 50	*Re* = 100	*Re* =150	*Re* = 200
100x50x50	D/5	0.917	0.804	0.766	0.704
200x100x100	D/10	1.513	1.045	0.832	0.732
400x200x200	D/20	1.534	1.050	0.855	0.752
800x400x400	D/40	1.556	1.081	0.887	0.766

Table 32 STAR-CCM+ mesh sensitivity analysis for drag coefficient in the three-dimensional flow past a sphere at *Re* = 50, 100, 150 and 200

		C_D			
Fluid #cells	Δx	*Re* = 50	*Re* = 100	*Re* = 150	*Re* = 200
300x200x200	D/10	1.542	1.068	0.873	0.757
375x250x250	D/12.5	1.556	1.079	0.883	0.768

Good agreement was also found with the numerical data presented in Le et al. [25], Kim et al. [64], Fornberg [89], Johnson and Patel [116], Gilmanov et al. [117], Constantinescu and Squires [118] and the experimental data of Taneda [119]. In Table 33 the numerical predictions from the IFEM, the IFVM and STAR-CCM+ are compared against the data from Kim et al. [44] and Fornberg [89].

Table 33 Comparison of the drag coefficient in the three-dimensional flow past a sphere at Re = 50, 100, 150 and 200

	C_D			
	Re = 50	Re = 100	Re = 150	Re = 200
Kim et al [64]	//	1.087	//	//
Fornberg [89]	//	1.085	//	0.786
Present (IFEM, vm, dg = 1)	1.694	1.193	//	//
Present (IFVM)	1.556	1.081	0.887	0.766
Star-CCM+/V7.04.006	1.556	1.079	0.883	0.768

For $210 \leq Re \leq 270$ the flow is no longer axisymmetric, but it is still steady. The lateral force coefficient defined as:

$$C_L = \frac{F_L}{\frac{1}{2}\rho U_\infty^2 \pi \frac{D^2}{4}} \tag{152}$$

points out the loss of axial symmetry for the flow. In eq. (73) F_L represents the lateral force on the sphere and can be determined using eq. (30) for immersed body surface mesh approach:

$$F_L(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t) \cdot \underline{n}_y d\underline{x} = \int_{\underline{x}} \int_{\Gamma_i(t)} \underline{F}_K\left(\underline{X}(s,t)\right) \delta\left(\underline{X} - \underline{x}\right) ds \cdot \underline{n}_y d\underline{x} \tag{153}$$

and eq. (38) for immersed body volume meshes:

$$F_L(t) = \int_{\underline{x}} \underline{\lambda}(\underline{x},t) \cdot \underline{n}_y d\underline{x} = \int_{\underline{x}} \int_{\omega(t)} \underline{F}_K\left(\underline{X}(s,t)\right) \delta\left(\underline{X} - \underline{x}\right) ds \cdot \underline{n}_y d\underline{x} \tag{154}$$

For axisymmetric flow (i.e., $Re \leq 200$) C_L is zero. At $Re > 210$, the lateral force coefficient increases in absolute value and start to oscillate at $Re > 270$ for which the flow becomes unsteady. In Figure 65 are shown the streamlines for the IFVM on the (x,z)-plane at $Re = 250$. The IFVM simulation predicts loss of symmetry with respect to the (x,y)-plane for the flow at Re ~ 250, which is less than the value predicted by experimental data and other numerical simulations (Re ~ 270). We believe the loss of symmetry is due to the coarse discretization of the sphere surface which introduces some numerical instability on the flow distribution. This triggers the beginning of flow instability for $Re < 270$. In Figure 66 are shown the streamlines for the IFVM on the (x,y)-plane at $Re = 250$. The good qualitative prediction of the flow distribution in the (x,y)-plane for the IFVM with the results presented in Le et al. [25] gives us good confidence that a finer sphere surface discretization would reduce the instability which triggered the loss of symmetry for $Re < 270$. It is worth mentioning that STAR-CCM+ predicted loss of symmetry for $Re \sim 270$.

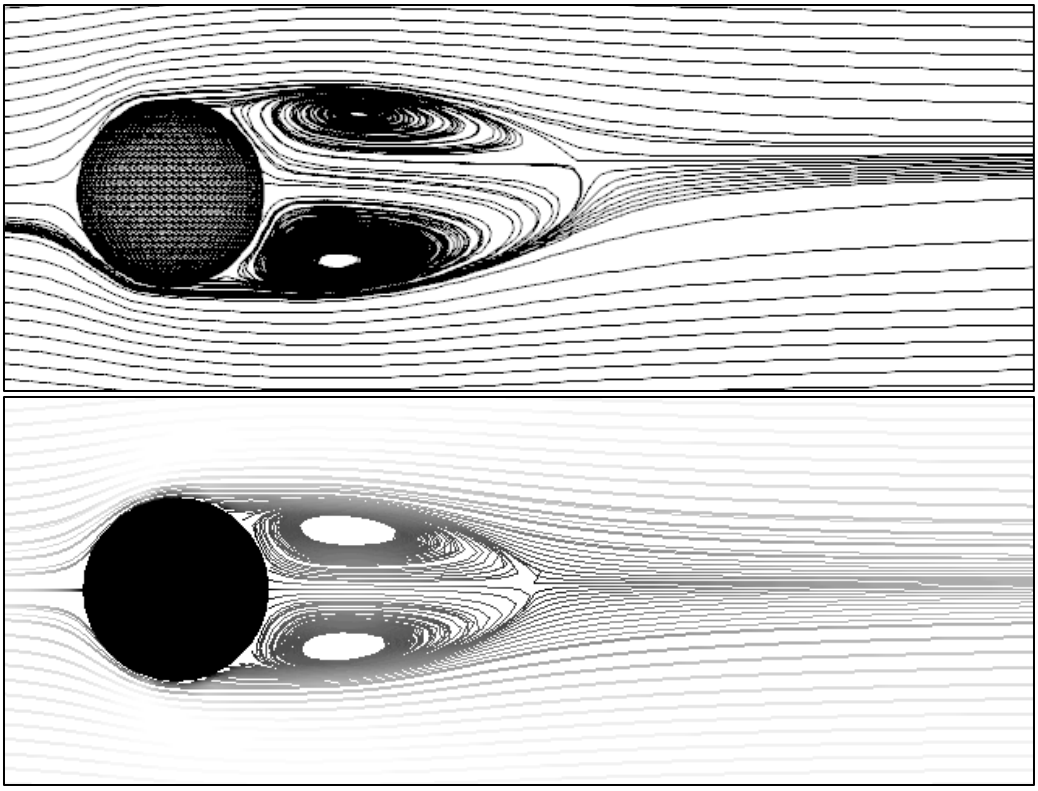

Figure 65. Streamlines on the (x,z)-plane for flow past a sphere at Re = 250: Top: IFVM; bottom: STAR-CCM+.

In Table 34 and Table 35 are shown the IFVM/STAR-CCM+ mesh sensitivity analysis for drag and lateral force coefficients at Re = 250, respectively. The drag coefficient is in good agreement between the two approaches. No converged results on the lateral force coefficient could be obtained with STAR-CCM+ code.

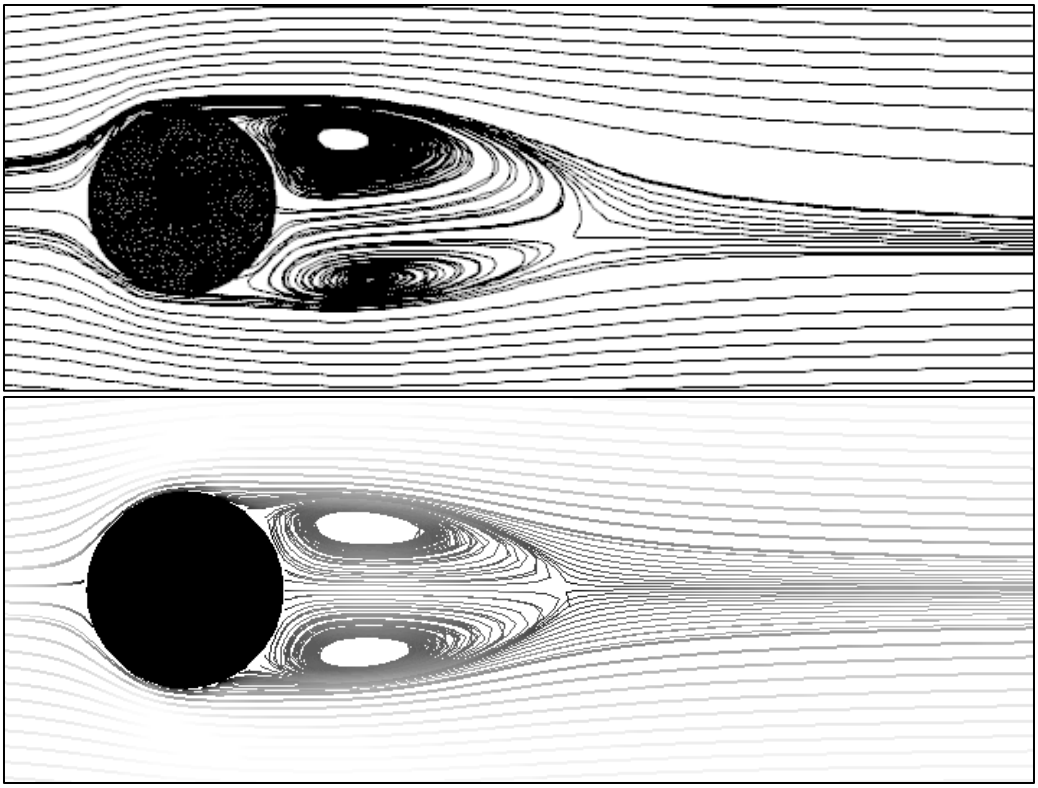

Figure 66. Streamlines on the (*x,y*)-plane for flow past a sphere at *Re* = 250: Top: IFVM; bottom: STAR-CCM+.

Table 34 IFVM and STAR-CCM+ mesh sensitivity analysis for drag and later force coefficients in the three-dimensional flow past a sphere at *Re* = 250

IFVM				STAR-CCM+/V7.04.006			
Fluid #cells	Δx	C_D	C_L	Fluid #cells	Δx	C_D	C_L
100x50x50	D/5	0.750	0.0002 (avg.)	300x200x200	D/10	0.678	0.0018 (avg.)
200x100x100	D/10	0.760	0.0008 (avg.)	375x250x250	D/20	0.698	0.0451 (avg.)
400x200x200	D20	0.702	0.02 (avg.)	//	//	//	//

Table 35 IFVM and STAR-CCM+ mesh sensitivity analysis for drag and later force coefficients in the three-dimensional flow past a sphere at *Re* = 300

IFVM				`STAR-CCM+/V7.04.006			
Fluid #cells	Δx	C_D	C_L	Fluid #cells	Δx	C_D	C_L
100x50x50	D/5	0.746	0.0001 (avg.)	300x200x200	D/10	0.637	//
200x100x100	D/10	0.717	0.0037 (avg.)	//	//	//	//
400x200x200	D20	0.673	0.027 (avg.)	//	//	//	//

In Table 36 are shown the comparison for the drag and lateral force coefficients and Strouhal number between the IFVM, STAR-CCM+ and data available in the literature. The drag coefficient obtained with the IFVM approach and STAR-CCM+ compares very well with the other numerical data at *Re* = 250, meanwhile there are some differences at *Re* = 300. The IFVM underestimates the lateral force coefficient for both simulations. No converged results on the lateral force coefficient could be obtained with STAR-CCM+.

In Figure 67 and Figure 68 are shown the IFVM streamlines in the (*x*,*z*)-plane and (*x*-*y*)-plane at *Re* = 300, respectively. Figure 67 confirms that the wake behind the sphere is not anymore symmetric about the (*x*-*y*)-plane in agreement with the loss of symmetry shown at *Re* = 250 in Figure 65. Nevertheless the flow distribution in the (*x*-*y*)-plane is qualitatively very similar to the one observed by Le et al. [25] in the three-dimensional flow past a sphere at Re = 300.

Overall the results presented for the IFVM give a reasonable level of confidence that the IFVM approach can correctly predict three-dimensional flow problems. Further

analyses are necessary to address the discrepancies on the lateral force coefficient and on the premature loss of symmetry in the sphere wake determined with the IFVM approach. The poor quality of the STAR-CCM+ simulations is most likely due to the very coarse mesh used. Further analyses with a more refined mesh are necessary before a sound judgment of the results can be performed. No results for the IFEM were presented at *Re* = 250 and 300. The number of time steps required to reach a fully developed flow for the three-dimensional flow past a sphere at *Re* = 250, 300 would have required unreasonable computational time even with very coarse meshes using the IFEM approach.

Table 36 Comparison of the drag and lateral force coefficients and Strouhal number in the three-dimensional flow past a sphere at Re = 250, 300

	C_D		C_L		*St*
	Re = 250	*Re* = 300	*Re* = 250	*Re* = 300	*Re* = 300
Kim et al [64]	0.701	0.657	0.059	0.067	0.134
Johnson and Patel [116]	0.70	0.656	0.062	0.069	0.137
Constantinescu and Squires [118]	0.70	0.655	0.062	0.065	0.136
Present (IFVM)	0.702	0.673	0.020	0.027	0.115
Star-CCM+/ V7.04.006	0.698	0.637	0.045	//	//

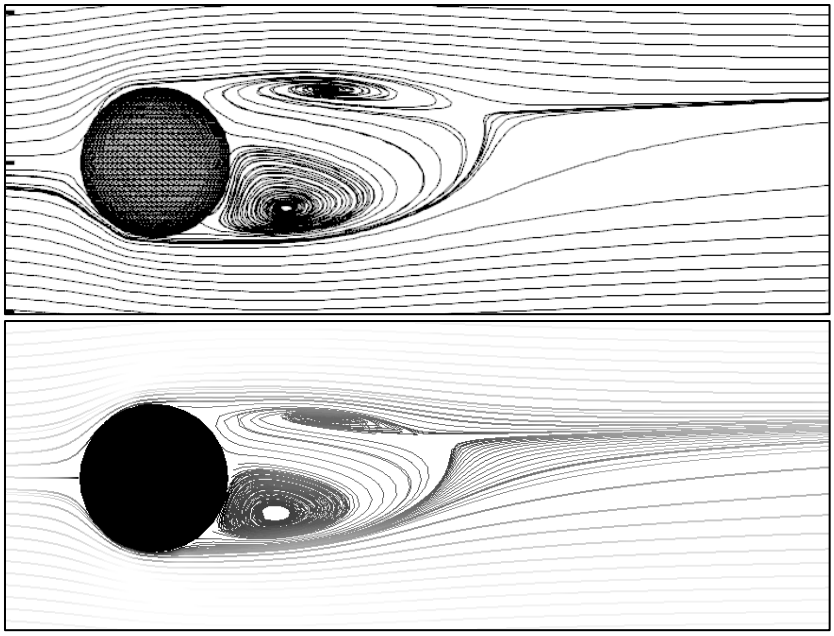

Figure 67. Streamlines on the (*x*,*z*)-plane for flow past a sphere at *Re* = 300: Top: IFVM; bottom: STAR-CCM+.

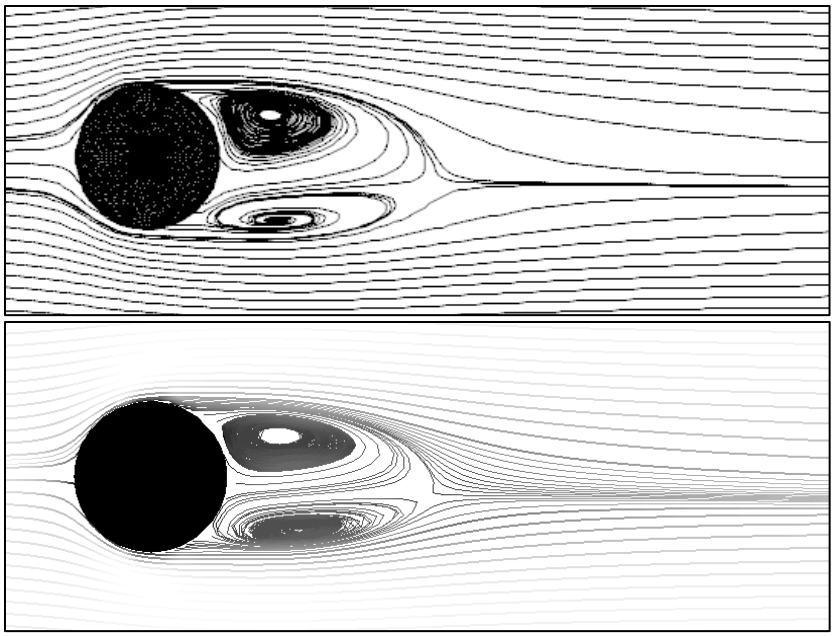

Figure 68. Streamlines on the (*x*,*y*)-plane for flow past a sphere at *Re* = 300: Top: IFVM; bottom: STAR-CCM+.

4.5 Two-Dimensional Natural Convection in a Square Cavity

To test the accuracy of the implemented energy equation for the finite element and finite volume discretizations, we considered the two-dimensional natural convection in a square cavity problem. This test case has been extensively used as benchmark for the validation of numerical methods in simulating natural convection problems. The square cavity has hot and cold isothermal boundary conditions at the left and right vertical walls, respectively, and adiabatic boundary conditions at the top and bottom horizontal walls, as shown in Figure 69.

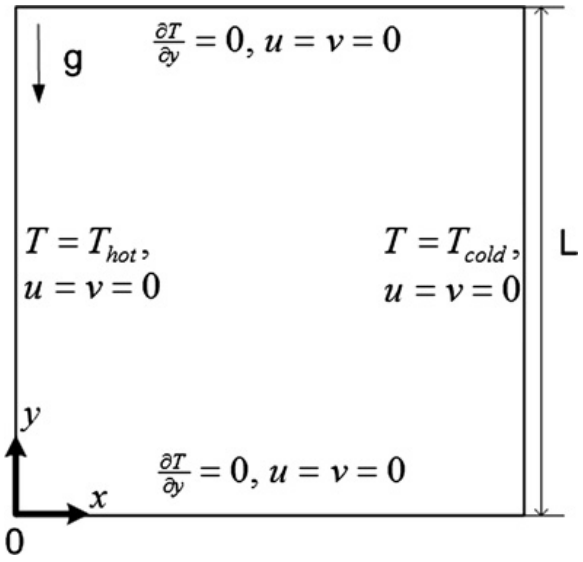

Figure 69. Geometry and boundary conditions for the two-dimensional natural convection in a square cavity.

In natural convection problems the ratio of Grashof number (*Gr*) to the square of Reynolds number (*Re*) is small, that is: $Gr/Re^2 \leq 1$, where Gr/Re^2 represents the importance of buoyancy forces compared to inertia forces. Using the definition of *Gr*

number from eq. (63) and *Re* number from eq. (3), it is possible to define a characteristic velocity U_c:

$$\frac{Gr}{\mathrm{Re}^2}=1\Rightarrow\frac{\dfrac{\rho^2 g\beta\Delta TL^3}{\mu^2}}{\left(\dfrac{\rho UL}{\mu}\right)^2}=\frac{g\beta\Delta TL}{U^2}=1\Rightarrow U=U_c=\sqrt{g\beta\Delta TL} \tag{155}$$

The characteristic velocity should be selected to be small so that the approximation of incompressible flow is still applicable. If buoyancy forces are of the same order or larger than inertia forces, it is possible to use two non-dimensional numbers to characterize natural convection problems: the Prandlt number (*Pr*) and the Rayleigh number (*Ra*), defined as:

$$\mathrm{Pr}=\frac{\mu c_p}{k}=\frac{\nu}{\alpha} \tag{156}$$

$$Ra=Gr\bullet\mathrm{Pr}=\frac{\rho^2 g\beta\Delta TL^3}{\mu^2}\bullet\frac{\mu c_p}{k}=\frac{g\beta\Delta TL^3}{\nu\alpha} \tag{157}$$

where μ is the fluid dynamic viscosity, c_p is the fluid specific heat, *k* is the fluid thermal conductivity, ν is the kinetic viscosity ($\nu = \mu/\rho$); α is the thermal diffusivity ($\alpha = k/\mu c_p$); *g* represents the gravitational constant, β is the coefficient of thermal expansion, ΔT represents the temperature difference between the hot and cold walls (i.e., $\Delta T = T_{hot} - T_{cold}$) and is the driving force for the natural convection in a square cavity and *L* is the cavity reference length (i.e., the cavity height). The *Pr* number is a measure of the fluid molecular diffusivity to the thermal diffusivity. The *Ra* number is a measure of the strength of buoyancy forces (the driving phenomena) compared to viscous forces (the dissipative phenomena).

Qualitative comparisons were performed for the maximum horizontal velocity u_{max} and its vertical position $z_{u,max}$ on the middle plane of the cavity ($x = L/2$), the maximum vertical velocity w_{max} and its horizontal position $x_{w,max}$ on the middle plane of the cavity ($z = L/2$), the average Nusselt number throughout the cavity Nu_{avg}, the average Nusselt number on the middle plane of the cavity ($x = L/2$) $Nu_{1/2}$, the average Nusselt number at the hot wall Nu_0, the maximum value of the local Nusselt number at the hot wall Nu_{max} and its location $z_{Nu,max}$ and the minimum value of the local Nusselt number at the hot wall Nu_{min} and its location $z_{Nu,min}$. We define a Nusselt number based on the heat transfer across the cavity by using the local heat flux in the horizontal direction:

$$Nu(x,z) = Pe \bullet q_x(x,z) = Pe\left[u(x,z) \bullet T(x,z) - \frac{1}{Pe}\frac{\partial T(x,z)}{\partial x} \right] \tag{158}$$

The average Nusselt number throughout the cavity is defined as:

$$Nu_{avg} = \int_0^L \int_0^L q_x(x,z)dxdz \tag{159}$$

where all variables have been normalized to their reference values. In a similar way it can be defined the average Nusselt number on the middle plane and at the hot wall:

$$Nu_{avg,x} = \int_0^L q_x(x,z)dz \tag{160}$$

Simulations were performed at Rayleigh numbers $Ra = 10^3$, 10^4, 10^5 and 10^6, for the IFEM, the IFVM and STAR-CCM+, and the results were compared with the numerical simulations of Kang and Hassan [29] and De Vahl Davis [120], Hortmann et al. [121], Barakos et al. [122], Markatos and Pericleous [123] and Fusegi et al. [124].

Setting the fluid physical property, from eq. (76) it is possible to see that the two free parameters are the cavity height L and the temperature gradient across the cavity ΔT. All simulations were performed setting $\Delta T = 10$. The cavity height was changed to match the desired Rayleigh number. This also implies that a different characteristic velocity [see eq. (155)] is obtained for each case considered. The Prandlt number was kept constant in all simulations equal to $Pr = 0.71$ corresponding to that of air. In Table 37 are given the flow physical property used for the two-dimensional natural convection in a square cavity simulations, and the corresponding cavity height and characteristic velocity for the Rayleigh number range considered.

Table 37 Air physical property and geometric configuration used for the two-dimensional natural convection in a square cavity at $Ra = 10^3$, 10^4, 10^5 and 10^6

Ra	10^3	10^4	10^5	10^6
ρ (kg/m^3)	1.205	1.205	1.205	1.205
μ (Pa•s)	$1.81562 \cdot 10^{-5}$	$1.81562 \cdot 10^{-5}$	$1.81562 \cdot 10^{-5}$	$1.81562 \cdot 10^{-5}$
c_p (J/ kg K)	1005.0	1005.0	1005.0	1005.0
k (W / m K)	0.0257	0.0257	0.0257	0.0257
β (1/K)	0.00343	0.00343	0.00343	0.00343
L (m)	0.009831	0.021181	0.045633	0.09831
U_c (m/s)	0.057516	0.084422	0.123915	0.181882

In Table 38, Table 39 and Table 40 are shown the mesh sensitivity analyses for the two-dimensional natural convection in a square cavity at $Ra = 10^3$ performed with

the IFEM approach, the IFVM approach and STAR-CCM+, respectively. The results show that mesh convergence was obtained for the three different approaches. The results also show very close agreement between the IFEM and the IFVM. Considering that for the two-dimensional natural convection in a square cavity no immersed boundary is present, excellent agreement with the body-fitted commercial code STAR-CCM+ was found.

Table 38 IFEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^3$

Fluid #cells	Δx	Nu_{avg}	Nu_0	Nu_{min} ($z_{Nu,min}$)	Nu_{max} ($z_{Nu,max}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
8x8	L/16	1.118	1.123	0.691 (1.0)	1.516 (0.125)	3.650 (0.812)	3.692 (0.187)
16x16	L/32	1.118	1.119	0.691 (1.0)	1.508 (0.094)	3.650 (0.812)	3.692 (0.187)
32x32	L/64	1.118	1.118	0.691 (1.0)	1.507 (0.094)	3.650 (0.812)	3.695 (0.172)
64x64	L/128	1.118	1.118	0.691 (1.0)	1.506 (0.086)	3.650 (0.812)	3.698 (0.180)
128x128	L/256	1.118	1.118	0.691 (1.0)	1.506 (0.090)	3.650 (0.812)	3.698 (0.180)

Table 39 IVEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^3$

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
10x10	L/10	1.114	1.151	1.144	1.609 (0.050)	0.659 (1.0)	3.751 (0.850)	3.848 (0.150)
20x20	L/20	1.124	1.125	1.124	1.531 (0.075)	0.683 (1.0)	3.677 (0.825)	3.737 (0.175)
40x40	L/40	1.119	1.119	1.119	1.512 (0.087)	0.689 (1.0)	3.656 (0.812)	3.698 (0.187)
80x80	L/80	1.118	1.118	1.118	1.507 (0.094)	0.691 (1.0)	3.648 (0.819)	3.697 (0.181)
160x160	L/160	1.118	1.118	1.118	1.506 (0.091)	0.691 (1.0)	3.650 (0.816)	3.696 (0.178)
320x320	L/320	1.118	1.118	1.119	1.507 (0.089)	0.692 (1.0)	3.648 (0.814)	3.697 (0.180)
640x640	L/640	1.118	1.118	1.118	1.506 (0.088)	0.691 (1.0)	3.648 (0.813)	3.696 (0.179)

In Figure 70 and shown the temperature contours for the two-dimensional natural convection in a square cavity at $Ra = 10^3$ obtained with the IFEM, the IFVM and STAR-CCM+, respectively. The figures show that the temperature distribution across

the cavity is qualitatively very similar. For this small value of *Ra* number, the heat transfer mode is mainly dominated by conduction, convection being almost negligible.

Figure 71 shows the velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^3$ determined with the IFEM, the IFVM and STAR-CCM+, respectively. Also for the flow field the three approaches give very similar qualitative distributions.

Table 40 STAR-CCM+ mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^3$

Fluid #cells	Δx	Nu_0	Nu_{max} $(z_{Nu,max})$	Nu_{min} $(z_{Nu,min})$	u_{max} $(z_{u,max})$	w_{max} $(x_{w,max})$
100x100	L/100	1.119	1.508 (0.095)	0.691 (0.971)	3.654 (0.815)	3.702 (0.175)
200x200	l/200	1.121	1.510 (0.097)	0.694 (0.972)	3.651 (0.812)	3.698 (0.177)
300x300	L/300	1.125	1.514 (0.098)	0.695 (0.952)	3.651 (0.812)	3.699 (0.178)

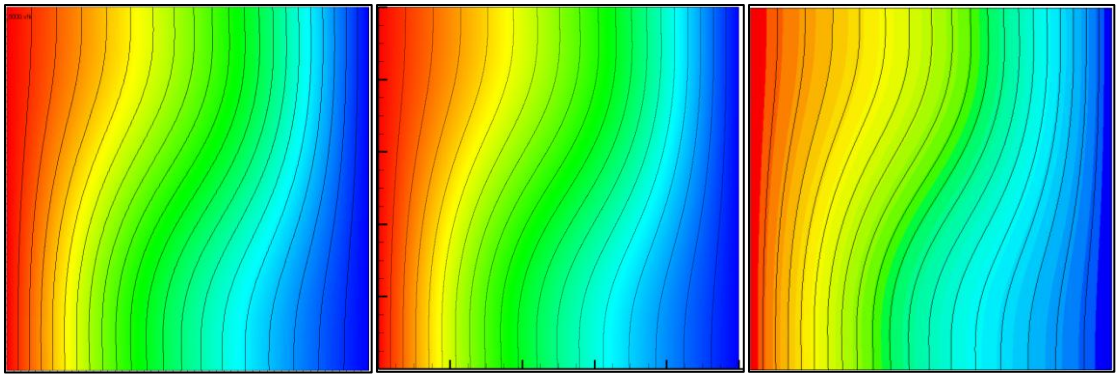

Figure 70. Temperature contours for the two-dimensional natural convection in a square cavity at $Ra = 10^3$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

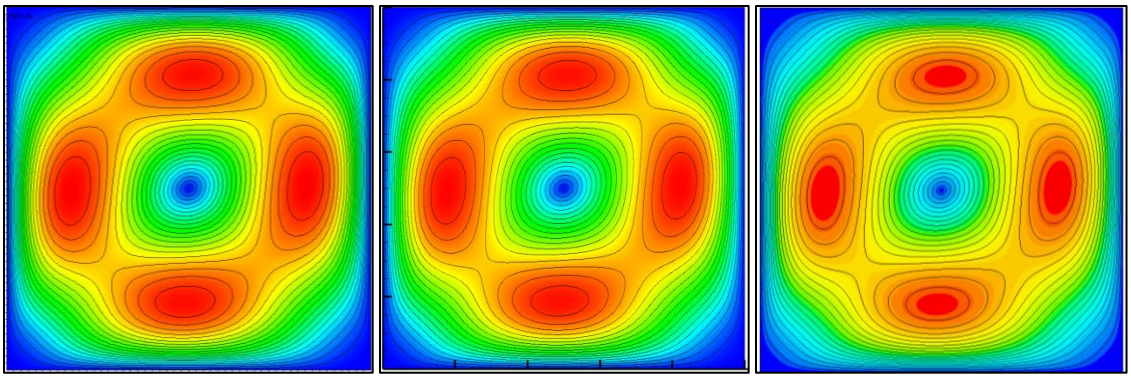

Figure 71. Velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^3$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

In Table 41, Table 42 and Table 43 are shown the mesh sensitivity analyses for the two-dimensional natural convection in a square cavity at $Ra = 10^4$ performed with the IFEM approach, the IFVM approach and STAR-CCM+, respectively. The results show that mesh convergence was obtained for the three different approaches. The IFEM, IFVM and STAR-CCM+ results are very close.

Table 41 IFEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^4$

Fluid #cells (#dofs)	Δx	Nu_{avg}	Nu_0	Nu_{min} ($z_{Nu,min}$)	Nu_{max} ($z_{Nu,max}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
8x8 (948)	L/16	2.241	2.325	0.594 (1.0)	3.687 (0.125)	16.155 (0.812)	19.612 (0.125)
16x16 (3556)	L/32	2.244	2.264	0.588 (1.0)	3.576 (0.156)	16.144 (0.812)	19.596 (0.125)
32x32 (13764)	L/64	2.245	2.248	0.586 (1.0)	3.539 (0.141)	16.174 (0.828)	19.595 (0.125)
64x64 (54148)	L/128	2.245	2.245	0.585 (1.0)	3.531 (0.141)	16.179 (0.820)	19.624 (0.117)
128x128 (214788)	L/256	2.245	2.245	0.585 (1.0)	3.530 (0.145)	16.182 (0.824)	19.624 (0.117)

Table 42 IVEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^4$

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
10x10	L/10	2.493	2.494	2.494	4.212 (0.050)	0.560 (1.0)	16.123 (0.850)	18.379 (0.150)
20x20	L/20	2.309	2.309	2.309	3.742 (0.125)	0.578 (1.0)	16.214 (0.825)	19.689 (0.125)

Table 42 Continued

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
40x40	L/40	2.260	2.260	2.261	3.585 (0.137)	0.584 (1.0)	16.146 (0.812)	19.642 (0.112)
80x80	L/80	2.248	2.249	2.249	3.546 (0.144)	0.585 (1.0)	16.177 (0.819)	19.636 (0.119)
160x160	L/160	2.246	2.246	2.246	3.536 (0.147)	0.585 (1.0)	16.183 (0.822)	19.621 (0.122)
320x320	L/320	2.245	2.245	2.246	3.532 (0.145)	0.585 (1.0)	16.183 (0.823)	19.627 (0.120)
640x640	L/640	2.245	2.245	2.246	3.533 (0.145)	0.586 (1.0)	16.186 (0.823)	19.631 (0.120)

In Figure 72 and Figure 73 are shown the IFEM, IFVM and STAR-CCM+ temperature contours and velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^4$, respectively. Also for $Ra = 10^4$ the temperature and flow distribution across the cavity is qualitatively very similar for the three approaches used. The heat transfer mode is still mainly dominated by conduction, but the distortion on the temperature contours due to convection is more evident than for the natural convection in a square cavity for $Ra = 10^3$ case.

Table 43 STAR-CCM+ mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^4$

Fluid #cells	Δx	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
100x100	L/100	2.248	3.541 (0.150)	0.585 (0.985)	16.177 (0.825)	19.648 (0.115)
200x200	l/200	2.248	3.536 (0.152)	0.587 (0.987)	16.183 (0.822)	19.623 (0.112)
300x300	L/300	2.251	3.540 (0.152)	0.588 (0.995)	16.186 (0.822)	19.634 (0.118)

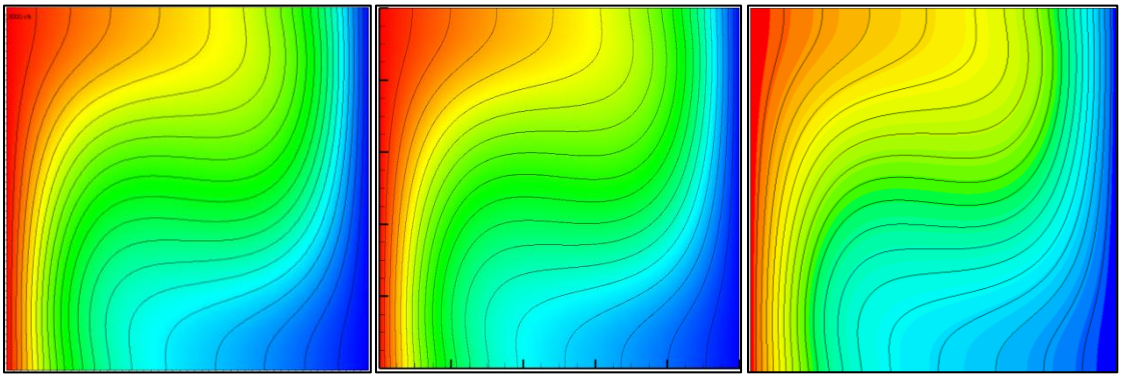

Figure 72. Temperature contours for the two-dimensional natural convection in a square cavity at $Ra = 10^4$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

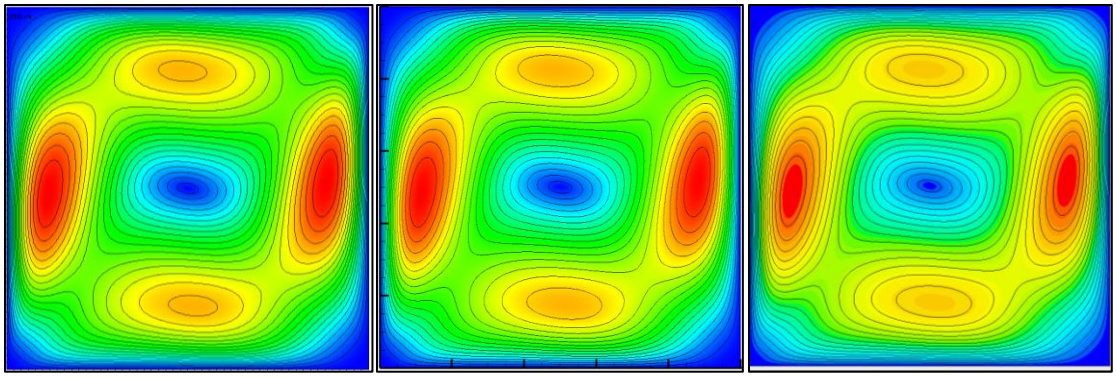

Figure 73. Velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^4$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

In Table 44, Table 45 and Table 46 are shown the mesh sensitivity analyses for the two-dimensional natural convection in a square cavity at $Ra = 10^5$ performed with the IFEM approach, the IFVM approach and STAR-CCM+, respectively. Also for $Ra = 10^5$ mesh convergence was obtained for the three different approaches and the results are in excellent agreement.

Table 44 IFEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^5$

Fluid #cells (#dofs)	Δx	Nu_{avg}	Nu_0	Nu_{min} ($z_{Nu,min}$)	Nu_{max} ($z_{Nu,max}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
8x8 (948)	L/16	4.417	4.811	0.752 (1.0)	7.948 (0.125)	33.247 (0.875)	68.420 (0.062)
16x16 (3556)	L/32	4.517	4.602	0.716 (1.0)	7.977 (0.094)	34.041 (0.844)	68.018 (0.062)
32x32 (13764)	L/64	4.521	4.557	0.731 (1.0)	7.838 (0.078)	34.702 (0.859)	68.510 (0.062)
64x64 (54148)	L/128	4.522	4.527	0.729 (1.0)	7.740 (0.078)	34.722 (0.852)	68.507 (0.062)
128x128 (214788)	L/256	4.522	4.522	0.728 (1.0)	7.722 (0.082)	34.733 (0.855)	68.627 (0.066)

Table 45 IVEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^5$

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
10x10	L/10	5.398	5.459	5.446	9.365 (0.050)	1.363 (1.0)	34.348 (0.850)	74.195 (0.050)
20x20	L/20	4.922	4.930	4.922	9.271 (0.025)	0.707 (1.0)	35.033 (0.875)	67.010 (0.075)

Table 45 Continued

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
40x40	L/40	4.627	4.629	4.627	8.220 (0.062)	0.720 (1.0)	34.880 (0.862)	69.048 (0.062)
80x80	L/80	4.547	4.548	4.548	7.843 (0.081)	0.726 (1.0)	34.790 (0.856)	68.616 (0.069)
160x160	L/160	4.528	4.528	4.529	7.751 (0.078)	0.728 (1.0)	34.752 (0.853)	68.659 (0.065)
320x320	L/320	4.523	4.523	4.524	7.728 (0.083)	0.728 (1.0)	34.747 (0.855)	68.612 (0.067)
640x640	L/640	4.522	4.522	4.523	7.723 (0.082)	0.728 (1.0)	34.744 (0.854)	68.631 (0.066)

Figure 74 and Figure 75 show the IFEM, IFVM and STAR-CCM+ temperature contours and velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^5$, respectively. The temperature and flow distribution across the cavity is qualitatively very similar for the three approaches used, with the heat transfer mode being driven predominantly by convection.

Table 46 STAR-CCM+ mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^5$

Fluid #cells	Δx	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
100x100	L/100	4.538	7.799 (0.075)	0.726 (0.995)	34.594 (0.855)	68.677 (0.065)
200x200	l/200	4.528	7.745 (0.082)	0.728 (0.997)	34.655 (0.852)	68.642 (0.067)
300x300	L/300	4.528	7.735 (0.085)	0.729 (0.998)	34.691 (0.855)	68.658 (0.065)

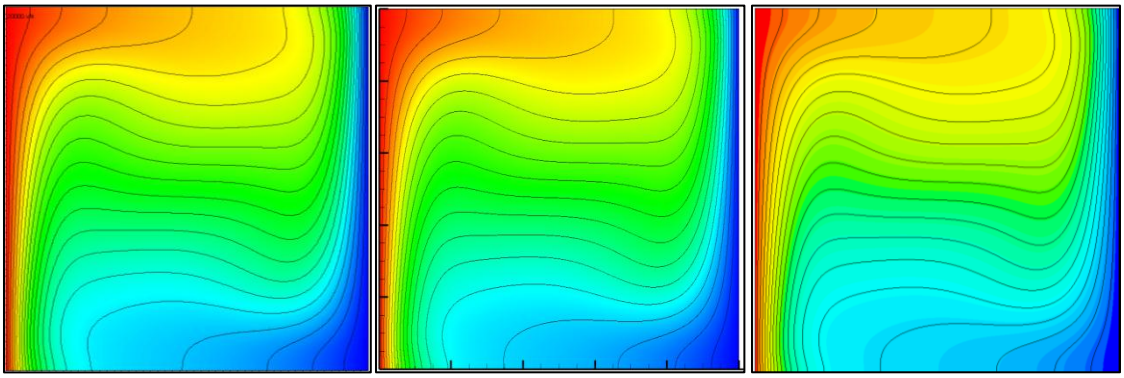

Figure 74. Temperature contours for the two-dimensional natural convection in a square cavity at $Ra = 10^5$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

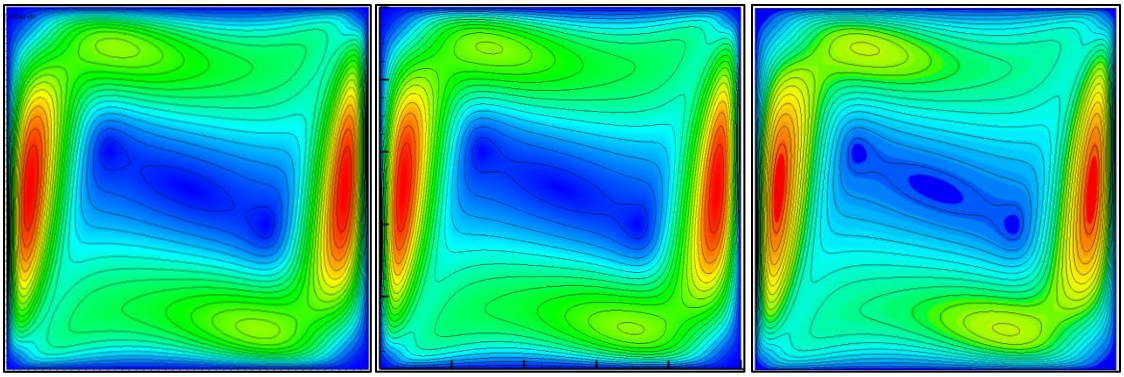

Figure 75. Velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^5$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

Table 47, Table 48 and Table 49 show the IFEM, the IFVM and STAR-CCM+ mesh sensitivity analyses for the two-dimensional natural convection in a square cavity at $Ra = 10^6$, respectively. Also for $Ra = 10^6$ mesh convergence was obtained for the three different approaches and the results are in good agreement with each other. In is worth mentioning that for the boundary conditions imposed, the flow velocity is of the same order of magnitude as the speed of sound for the fluid considered. For these conditions, the approximation of incompressible flow used for the three approaches is not valid anymore.

Table 47 IFEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^6$

Fluid #cells (#dofs)	Δx	Nu_{avg}	Nu_0	Nu_{min} $(z_{Nu,min})$	Nu_{max} $(z_{Nu,max})$	u_{max} $(z_{u,max})$	w_{max} $(x_{w,max})$
8x8 (948)	L/16	7.561	8.578	0.865 (1.0)	13.791 (0.187)	65.416 (0.875)	199.685 (0.062)
16x16 (3556)	L/32	8.669	9.578	1.032 (1.0)	18.133 (0.062)	63.723 (0.844)	215.789 (0.031)
32x32 (13764)	L/64	8.817	9.108	0.991 (1.0)	18.437 (0.047)	64.669 (0.844)	215.793 (0.031)
64x64 (54148)	L/128	8.825	8.890	0.983 (1.0)	17.859 (0.039)	64.816 (0.852)	220.378 (0.039)
128x128 (214788)	L/256	8.825	8.836	0.980 (1.0)	17.598 (0.039)	64.818 (0.852)	220.378 (0.039)

Table 48 IVEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity at $Ra = 10^6$

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} $(z_{Nu,max})$	Nu_{min} $(z_{Nu,min})$	u_{max} $(z_{u,max})$	w_{max} $(x_{w,max})$
10x10	L/10	8.048	8.284	8.286	11.530 (0.050)	4.629 (1.0)	88.143 (0.950)	191.923 (0.050)
20x20	L/20	10.437	10.467	10.471	20.535 (0.025)	1.664 (1.0)	65.732 (0.875)	240.822 (0.025)

Table 48 Continued

Fluid #cells	Δx	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
40x40	L/40	9.415	9.418	9.456	21.164 (0.012)	0.9860 (1.0)	66.106 (0.862)	223.927 (0.037)
80x80	L/80	8.982	8.983	8.983	18.796 (0.031)	0.973 (1.0)	65.210 (0.856)	218.334 (0.031)
160x160	L/160	8.864	8.864	8.865	17.851 (0.034)	0.977 (1.0)	64.923 (0.853)	219.896 (0.041)
320x320	L/320	8.834	8.834	8.835	17.613 (0.039)	0.979 (1.0)	64.854 (0.848)	220.421 (0.039)
640x640	L/640	8.827	8.827	8.827	17.554 (0.038)	0.979 (1.0)	64.838 (0.849)	220.519 (0.038)

Figure 76 and Figure 77 show the IFEM, the IFVM and STAR-CCM+ temperature contours and velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^6$, respectively. The temperature and flow distribution across the cavity is qualitatively very similar for the three approaches used. The heat transfer mode is driven by convection. Figure 76 also shows that the thermal boundary layer close to the hot wall is extremely thin. Very fine meshes are required to determine the correct values for the temperature gradients close to the hot wall. Since the

temperature gradient is the driving phenomena in the momentum equations, only for very fine meshes ($\Delta x/L \sim 10^{-3}$) a satisfying prediction of the flow and temperature distribution can be obtained.

Table 49 STAR-CCM+ mesh sensitivity comparison for the two-dimensional natural convection in a square cavity at $Ra = 10^6$

Fluid #cells	Δx	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
100x100	L/100	8.928	18.315 (0.035)	0.969 (0.995)	65.438 (0.845)	221.033 (0.035)
200x200	l/200	8.853	17.749 (0.037)	0.977 (0.997)	64.980 (0.847)	220.697 (0.037)
300x300	L/300	8.840	17.633 (0.038)	0.980 (0.998)	64.893 (0.848)	220.595 (0.038)

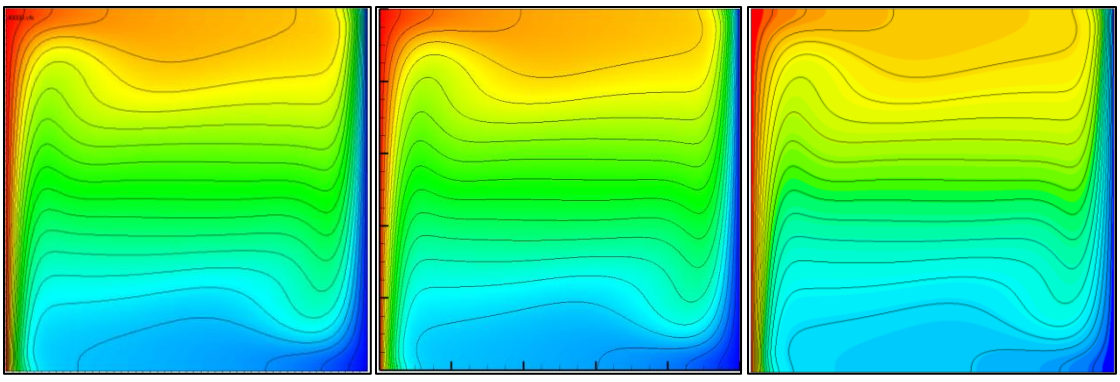

Figure 76. Temperature contours for the two-dimensional natural convection in a square cavity at $Ra = 10^6$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

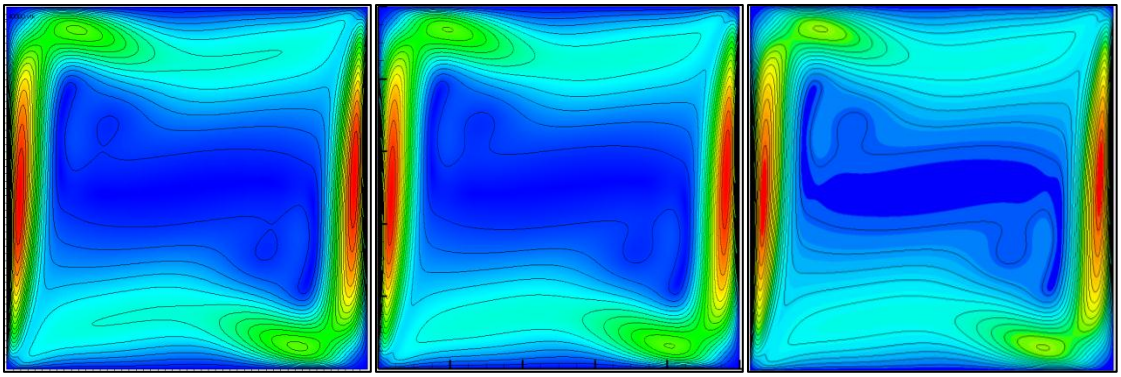

Figure 77. Velocity magnitude contours for the two-dimensional natural convection in a square cavity at $Ra = 10^6$. Left: IFEM vm, dg = 1; center: IFVM; right: STAR-CCM+.

For the two-dimensional natural convection in a square cavity no immersed boundary was present. The objective of these analyses was to test the accuracy of the finite element and finite volume discretization schemes when the time-dependent incompressible NSEs are strongly coupled with the energy equation.

In Table 50, Table 51, Table 52 and Table 53 are shown the comparison of the reference parameters for the two-dimensional natural convection in a square cavity at $Ra = 10^3$, 10^4, 10^5 and 10^6, respectively, for the IFEM, the IFVM, STAR-CCM+ and data from numerical simulations available in literature. The comparison of the reference parameters for the IFEM, the IFVM and STAR-CCM+ in the Rayleigh number range considered with the numerical data present in the literature shows very good agreement especially with the data of De Vahl Davis [120] and Hortmann et al. [121] numerical predictions.

Table 50 Comparison of the reference parameters for the two-dimensional natural convection in a square cavity at $Ra = 10^3$

	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
Kang and Hassan [29]	1.118	//	//	//	//	3.646 (0.810)	3.695 (0.180)
De Vahl Davis [120]	1.118	1.118	1.117	1.505 (0.092)	0.692 (1.0)	3.649 (0.813)	3.697 (0.178)
Barakos et al. [122]	1.114	//	//	1.581 (0.099)	0.670 (0.994)	4.105 (0.806)	4.159 (0.181)
Markatos and Pericleous [123]	1.108	//	//	1.596 (0.083)	0.720 (0.993)	//	//
Fusegi et al. [124]	1.105	//	//	1.420 (0.083)	0.764 (1.0)	3.542 (0.833)	3.515 (0.200)
Present (IFEM)	1.118	//	1.118	1.506 (0.090)	0.691 (1.0)	3.650 (0.812)	3.698 (0.180)
Present (IFVM)	1.118	1.118	1.118	1.506 (0.088)	0.691 (1.0)	3.648 (0.813)	3.696 (0.179)
STAR-CCM+/V7.04.006	//	//	1.125	1.514 (0.098)	0.695 (0.952)	3.651 (0.812)	3.699 (0.178)

Table 51 Comparison of the reference parameters for the two-dimensional natural convection in a square cavity at $Ra = 10^4$

	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
Kang and Hassan [29]	2.243	//	//	//	//	16.165 (0.820)	19.614 (0.120)
De Vahl Davis [120]	2.243	2.243	2.238	3.528 (0.143)	0.586 (1.0)	16.178 (0.823)	19.617 (0.119)
Hortmann et al. [121]	2.245	//	//	3.531 (0.146)	//	16.180 (0.825)	19.629 (0.120)
Barakos et al. [122]	2.245	//	//	3.539 (0.143)	0.583 (0.994)	16.262 (0.818)	19.717 (0.119)
Markatos and Pericleous [123]	2.201	//	//	3.482 (0.143)	0.643 (0.993)	//	//
Fusegi et al. [124]	2.302	//	//	3.652 (0.123)	0.611 (1.0)	16.936 (0.817)	18.959 (0.117)
Present (IFEM)	2.245	//	2.245	3.530 (0.145)	0.585 (1.0)	16.182 (0.824)	19.624 (0.117)
Present (IFVM)	2.245	2.245	2.246	3.533 (0.145)	0.586 (1.0)	16.186 (0.823)	19.631 (0.120)
STAR-CCM+/V7.04.006	//	//	2.251	3.540 (0.152)	0.588 (0.995)	16.186 (0.822)	19.634 (0.118)

Table 52 Comparison of the reference parameters for the two-dimensional natural convection in a square cavity at $Ra = 10^5$

	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} ($z_{Nu,max}$)	Nu_{min} ($z_{Nu,min}$)	u_{max} ($z_{u,max}$)	w_{max} ($x_{w,max}$)
Kang and Hassan [29]	4.514	//	//	//	//	34.680 (0.855)	68.545 (0.065)
De Vahl Davis [120]	4.519	4.519	4.509	7.717 (0.081)	0.729 (1.0)	34.730 (0.855)	68.590 (0.066)
Hortmann et al. [121]	4.522	//	//	7.720 (0.083)	//	34.740 (0.855)	68.637 (0.067)
Barakos et al. [122]	4.510	//	//	7.636 (0.085)	0.773 (0.999)	29.963 (0.859)	58.564 (0.066)
Markatos and Pericleous [123]	4.430	//	//	7.626 (0.083)	0.824 (0.993)	//	//
Fusegi et al. [124]	4.646	//	//	7.795 (0.083)	0.787 (1.0)	33.368 (0.855)	56.067 (0.065)
Present (IFEM)	4.522	//	4.522	7.722 (0.082)	0.728 (1.0)	34.733 (0.855)	68.627 (0.066)
Present (IFVM)	4.522	4.522	4.523	7.723 (0.082)	0.728 (1.0)	34.744 (0.854)	68.631 (0.066)
STAR-CCM+/V7.04.006	//	//	4.528	7.735 (0.085)	0.729 (0.998)	34.691 (0.855)	68.658 (0.065)

Table 53 Comparison of the reference parameters for the two-dimensional natural convection in a square cavity at $Ra = 10^6$

	Nu_{avg}	$Nu_{1/2}$	Nu_0	Nu_{max} $(z_{Nu,max})$	Nu_{min} $(z_{Nu,min})$	u_{max} $(z_{u,max})$	w_{max} $(x_{w,max})$
Kang and Hassan [29]	8.794	//	//	//	//	64.596 (0.848)	219.593 (0.036)
De Vahl Davis [120]	8.800	8.799	8.817	17.925 (0.038)	0.989 (1.0)	64.630 (0.850)	219.360 (0.038)
Hortmann et al. [121]	8.825	//	//	17.536 (0.039)	//	64.837 (0.852)	220.461 (0.039)
Barakos et al. [122]	8.806	//	//	17.442 (0.037)	1.001 (0.999)	62.994 (0.859)	219.360 (0.039)
Markatos and Pericleous [123]	8.754	//	//	17.872 (0.038)	1.232 (0.993)	//	//
Fusegi et al. [124]	9.012	//	//	17.670 (0.038)	1.257 (1.0)	68.720 (0.856)	216.848 (0.033)
Present (IFEM)	8.825	//	8.836	17.598 (0.039)	0.980 (1.0)	64.818 (0.852)	220.378 (0.039)
Present (IFVM)	8.827	8.827	8.827	17.554 (0.038)	0.979 (1.0)	64.838 (0.849)	220.519 (0.038)
STAR-CCM+/V7.04.006	//	//	8.840	17.633 (0.038)	0.980 (0.998)	64.893 (0.848)	220.595 (0.038)

4.6 Two-Dimensional Natural Convection in a Square Cavity with an Eccentric Cylinder

The next step is the validation of both finite element and finite volume immersed boundary approaches for the system of time-dependent incompressible NSEs coupled with the energy equation. The two-dimensional natural convection is a square cavity with an eccentric cylinder test case was chosen as benchmark exercise.

In a square cavity of height L, a circular cylinder with diameter $D = 0.4L$ is eccentrically located in the cavity $0.1L$ upward from the center, as shown in Figure 78. The circular cylinder surface is at temperature $T = T_{hot}$, with the left and right cavity walls at temperature $T = T_{cold}$. Adiabatic boundary conditions were imposed for the top and bottom cavity walls. No-slip boundary conditions were imposed for the momentum equations at the cavity walls. This benchmark problem has been simulated by different numerical methods under $Ra = 10^6$ and $Pr = 0.1$ and 10. The IFEM, the IFVM and STAR-CCM+ results were compared to the numerical data of Kang and Hassan [29], Pacheco et al. [100], Feng and Michaelidis [125], Demirdizic et al. [126] and Yu et al. [127]. In Table 54 are given the fluid properties and the geometric configurations necessary to have the specified Ra and Pr numbers inside the square cavity with eccentric cylinder.

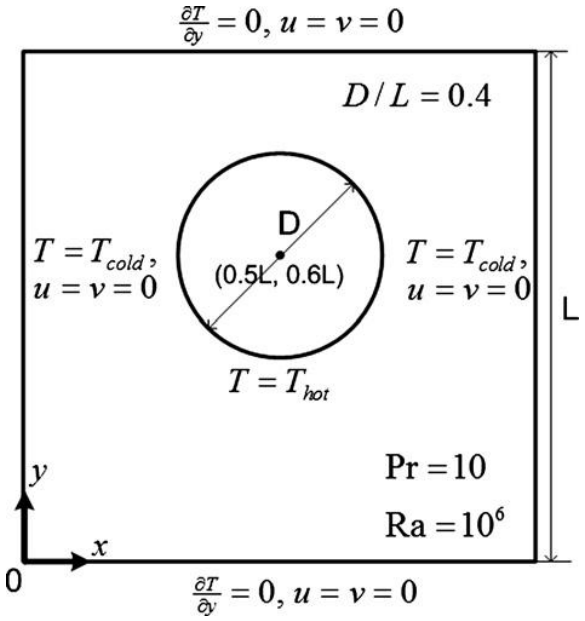

Figure 78. Geometry and boundary conditions for the two-dimensional natural convection in a square cavity with an eccentric cylinder.

Table 54 Fluid physical property and geometric configuration used for the two-dimensional natural convection in a square with an eccentric cylinder at $Ra = 10^6$

Pr	0.1	10
ρ (kg/m^3)	1.205	1.205
μ (Pa•s)	$2.5572 \cdot 10^{-6}$	$2.5572 \cdot 10^{-4}$
c_p (J/ kg K)	1005.0	1005.0
k (W / m K)	0.0257	0.0257
β (1/K)	0.00343	0.00343
L (m)	0.051152	0.131194
U_c (m/s)	0.237428	0.0282649

The average *Nu* number at the cold wall Nu_0 (i.e., east wall), the maximum *Nu* number at the cold wall Nu_{max} and its vertical position $y_{Nu,max}$ were considered as reference parameters to study mesh convergence and to compare the converged results to other numerical simulations of the natural convection in a square cavity with an eccentric cylinder present in literature.

In Table 55 is shown the mesh sensitivity analysis for the IFEM approach. From the results it is possible to assess that mesh convergence is reached for the two finest mesh refinements. Table 56 shows the mesh sensitivity study for the IFVM approach when the natural convection in a square cavity with an eccentric cylinder is studied. The results show that for the three finest levels of mesh refinements the Nu number distribution at the cold has reached convergence for both cases $Pr = 0.1$ and $Pr = 10$.

In Table 57 is reported the mesh sensitivity analysis for the natural convection in a square cavity with an eccentric cylinder when the STAR-CCM+ code is used. The results show that mesh convergence is not reached. Further mesh refinements were considered, but the simulation becomes unstable and does not reach a numerically converged state.

Table 58 shows the comparison of the converged results for the IFEM, the IFVM and STAR-CCM+ (for STAR-CCM+ we considered the second refinement as converged results) with the numerical data of Demirdzic et al. [126] (reference data). The IFEM and STAR-CCM+ results are in satisfactory agreement with the reference data for $Pr = 0.1$. For the $Pr = 10$ case, some discrepancies can be noted. The IFVM results are in excellent agreement with the reference data for both the $Pr = 0.1$ and $Pr = 10$ case.

Table 55 IFEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ for Pr = 0.1 and 10

		$Pr = 0.1$			$Pr = 10$		
Fluid #cells	Δx	Nu_0	Nu_{max}	$y_{Nu,max}$	Nu_0	Nu_{max}	$y_{Nu,max}$
16x16	L/32	6.124	11.924	0.937	7.407	20.349	0.937
32x32	L/64	6.679	14.235	0.891	6.803	19.338	0.969
64x64	L/128	6.482	14.101	0.883	6.929	19.323	0.961
128x128	L/256	6.647	14.427	0.879	7.002	19.305	0.965

Table 56 IVEM mesh sensitivity analysis for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ for Pr = 0.1 and 10

		$Pr = 0.1$			$Pr = 10$		
Fluid #cells	Δx	Nu_0	Nu_{max}	$y_{Nu,max}$	Nu_0	Nu_{max}	$y_{Nu,max}$
10x10	L/10	5.624	10.908	1.0	7.348	12.170	1.0
20x20	L/20	6.631	16.525	0.937	10.663	25.392	1.0
40x40	L/40	6.510	15.783	0.937	8.064	24.692	1.0
80x80	L/80	6.549	15.269	0.906	7.517	21.390	0.969
100x100	L/100	6.569	15.071	0.905	7.460	20.836	0.965
200x200	L/200	6.670	14.656	0.887	7.397	20.145	0.967
400x400	L/400	6.724	14.553	0.879	7.385	19.985	0.964
800x800	L/800	6.724	14.553	0.879	7.382	19.943	0.963
1000x1000	L/1000	6.724	14.553	0.879	7.382	19.939	0.963

Table 57 STAR-CCM+ mesh sensitivity analysis for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ for Pr = 0.1 and 10

		$Pr = 0.1$			$Pr = 10$		
Fluid #cells	Δx	Nu_0	Nu_{max}	$y_{Nu,max}$	Nu_0	Nu_{max}	$y_{Nu,max}$
100x100	L/100	6.745	14.690	0.865	7.354	20.657	0.965
200x200	l/200	6.724	14.613	0.863	7.014	19.672	0.962
300x300	L/300	6.798	14.698	0.882	6.529	18.684	0.962

Table 58 Comparison of the reference parameters for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ for Pr = 0.1 and 10

	$Pr = 0.1$			$Pr = 10$		
	Nu_0	Nu_{max}	$y_{Nu,max}$	Nu_0	Nu_{max}	$y_{Nu,max}$
Demirdzic et al [126]	6.730	14.558	0.879	7.384	19.949	0.965
Present (IFEM)	6.482	14.101	0.883	6.929	19.323	0.961
Present (IFVM)	6.724	14.553	0.879	7.382	19.939	0.963
STAR-CCM+/V7.04.006	6.724	14.613	0.863	7.014	19.672	0.962

In Figure 79, Figure 80 and Figure 81 are shown the temperature contours, velocity magnitude contours and streamlines for the two-dimensional natural convection in a square cavity with eccentric cylinder at $Ra = 10^6$ and $Pr = 0.1$ computed with the IFEM approach, the IFVM approach and STAR-CCM+, respectively. The figures show an excellent agreement of the two immersed boundary method approaches with the body fitted STAR-CCM+ results. The numerical data presented in Table 55 through Table 58

demonstrate the quantitative good agreement of the three different approaches. Figure 79 through Figure 81 show also the good qualitative agreement particularly close to the hot cylinder surface (i.e., the immersed boundary) where very large temperature gradients are present and the plume develops. All three approaches were able to catch the secondary recirculation zones above the cylinder surface. These results are in very good agreement with the numerical simulation of Demirdzic et al. [126].

To have a better quantitative validation of the immersed boundary methods developed, we compared the Nusselt number distribution at the cold wall for the converged IFEM, IFVM and STAR-CCM+ results with the numerical data of Demirdzic et al. [126] and Yu et al. [127]. Figure 82 shows the very good agreement that the IFVM and STAR-CCM+ have with the data from Demirdzic et al. [126] and Yu et al. [127] throughout the cavity height, especially close to the cavity top wall, where the IFEM results underestimate the Nusselt value.

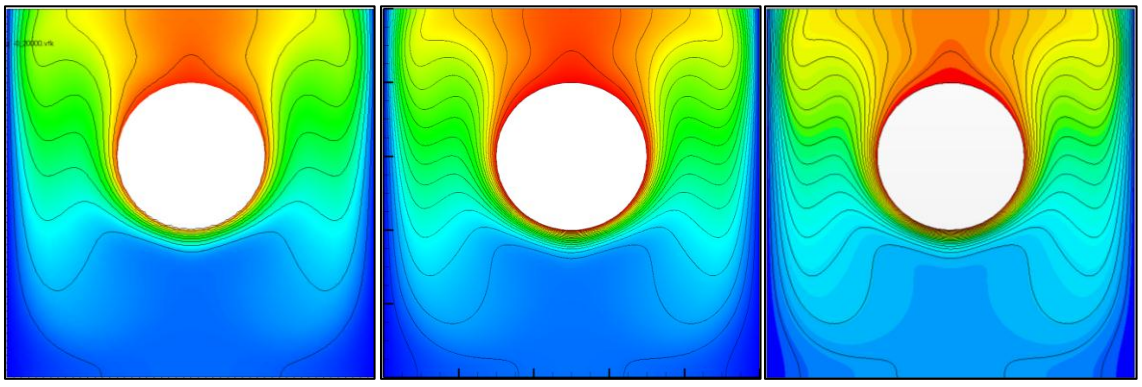

Figure 79. Temperature contours for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 0.1$. Left: IFEM; center: IFVM; right: STAR-CCM+.

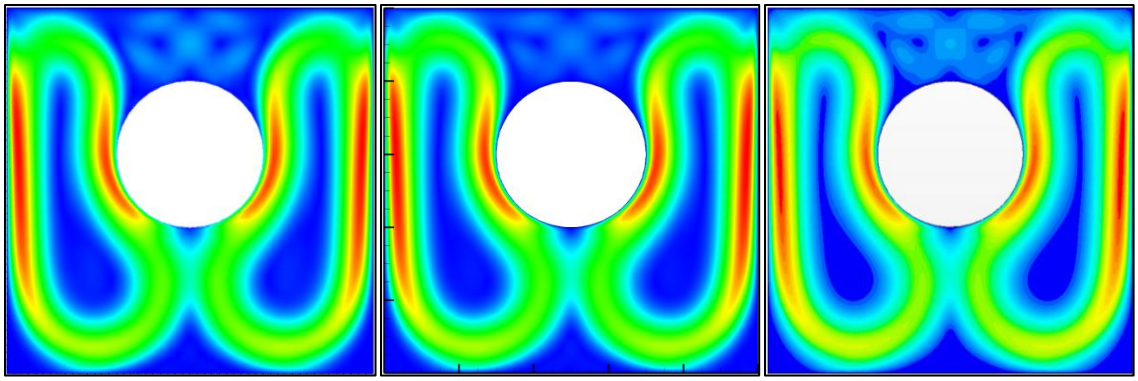

Figure 80. Velocity magnitude contours for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 0.1$. Left: IFEM; center: IFVM; right: STAR-CCM+.

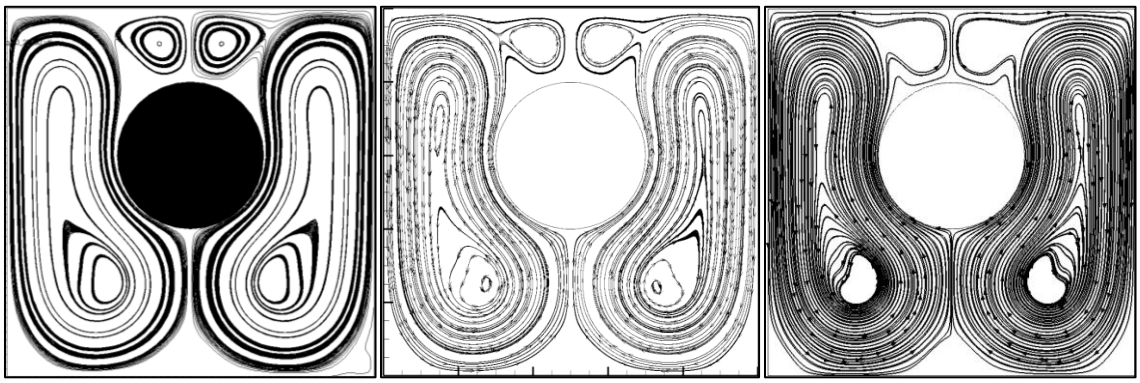

Figure 81. Streamlines for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 0.1$. Left: IFEM; center: IFVM; right: STAR-CCM+.

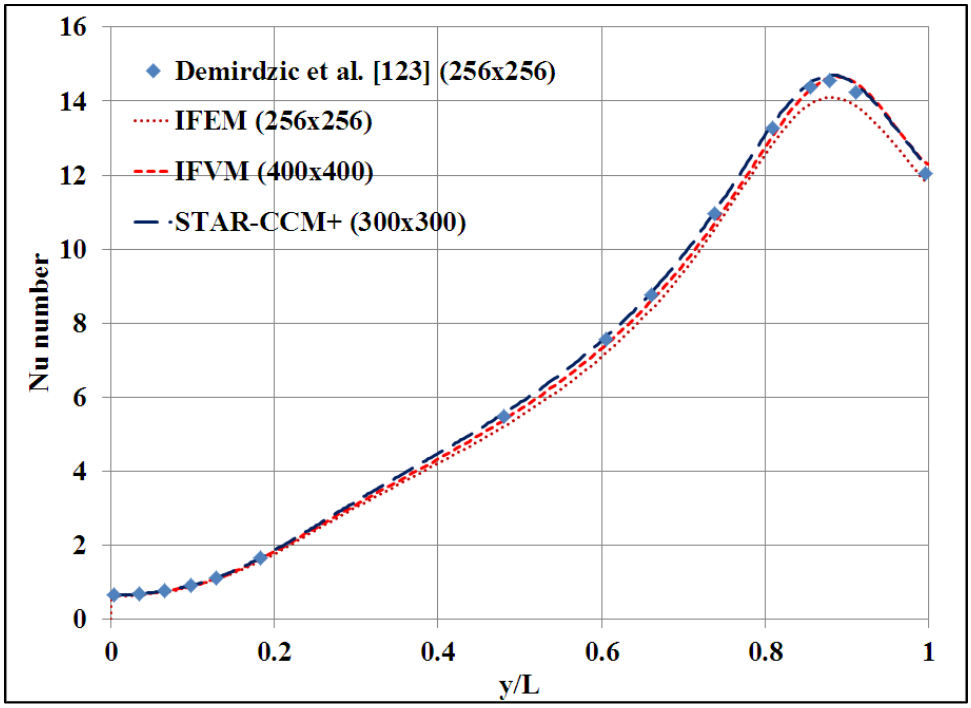

Figure 82. Cold wall Nusselt distribution for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 0.1$.

In Figure 83 and Figure 84 are shown the IFEM, the IFVM and STAR-CCM+ temperature and velocity magnitude contours for the two-dimensional natural convection in a square cavity with eccentric cylinder at $Ra = 10^6$ and $Pr = 10$, respectively. The IFEM and STAR-CCM+ predict a larger stratification region in the bottom part of the cavity with respect to the IFVM calculation. Comparing the temperature contours with those determined by Kang and Hassan [29], Pacheco et al. [100] and Demirdzic et al. [126], the IFVM is in better agreement then the IFEM and STAR-CCM+. In particular, none of the reference cited above predicts the secondary recirculation vortices close to the cavity bottom wall. Figure 85 shows the streamlines for the two-dimensional natural convection in a square cavity with eccentric cylinder at $Ra = 10^6$ and $Pr = 10$ determined

with the IFEM, the IFVM and STAR-CCM+, respectively. The IFEM and STAR-CCM+ predicts secondary recirculation regions close to the cavity bottom wall due to a larger stratification region in the lower part of the cavity, meanwhile the IFVM streamlines are in very good qualitative agreement with the streamlined determined by Kang and Hassan [29], Pacheco et al. [100] and Demirdzic et al. [126].

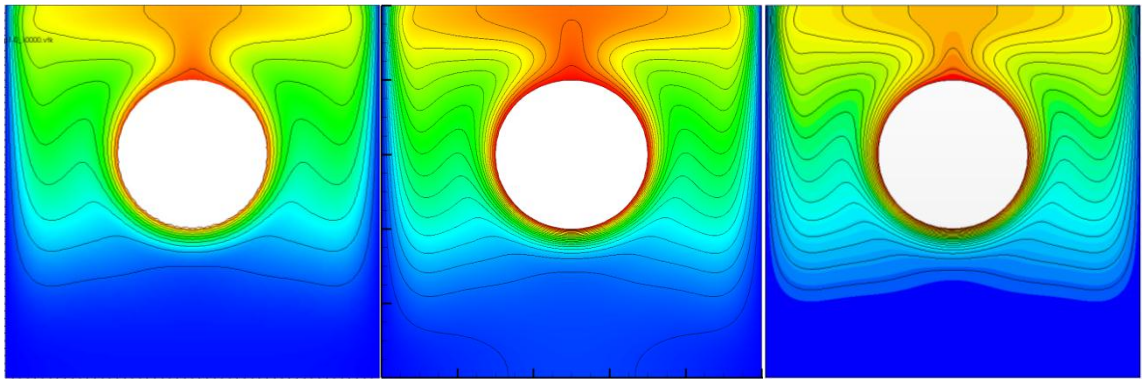

Figure 83. Temperature contours for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 10$. Left: IFEM; center: IFVM; right: STAR-CCM+.

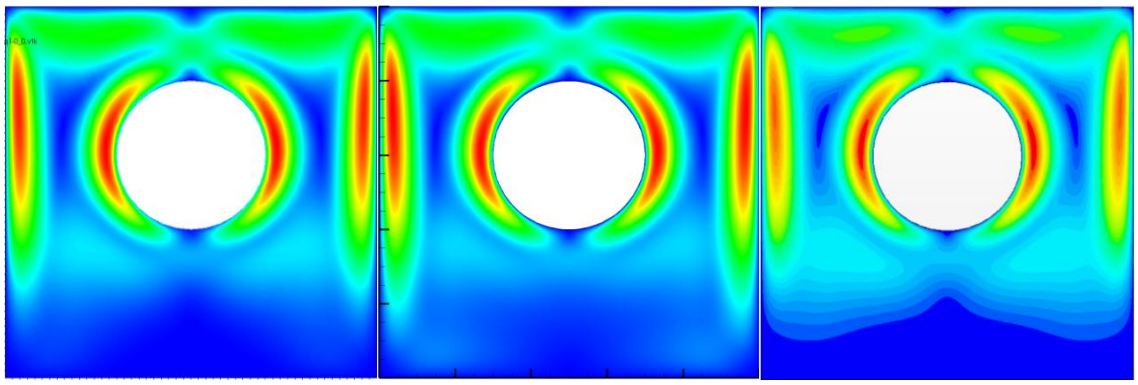

Figure 84. Velocity magnitude contours for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 10$. Left: IFEM; center: IFVM; right: STAR-CCM+.

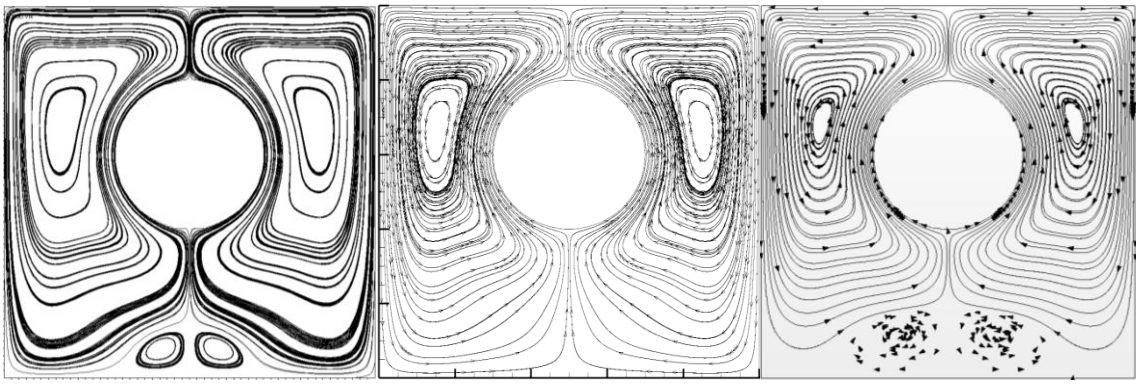

Figure 85. Streamlines for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 10$. Left: IFEM; center: IFVM; right: STAR-CCM+.

In Figure 86 is plotted the Nusselt distribution al the lateral cold wall for the two-dimensional natural convection in a square cavity with eccentric cylinder at $Ra = 10^6$ and $Pr = 10$ determined with the IFEM, the IFVM and STAR-CCM+ against the reference data of Demirdzic et al. [126]. The figure shows the better agreement of the IFVM results with the reference data if compared to the IFEM and STAR-CCM+ numerical predictions. In Figure 87 we compare the Nusselt distribution at the lateral cold wall for the same test case obtained with the IFVM to the data from Pacheco et al. [100], Demirdzic et al. [126] and Yu et al. [127]. The IFVM predictions are in excellent agreement with the other numerical data, throughout the cavity height.

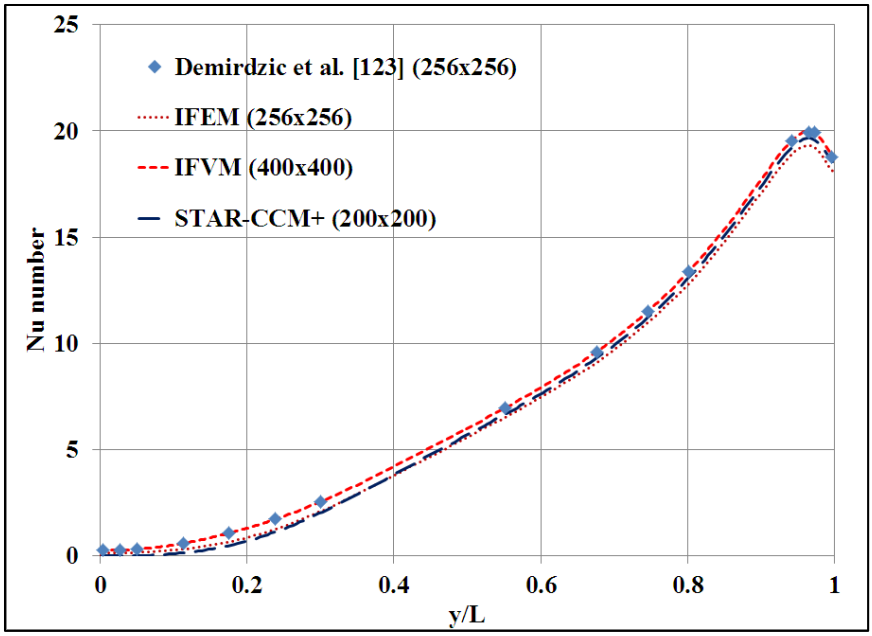

Figure 86. Cold wall Nusselt distribution for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 10$.

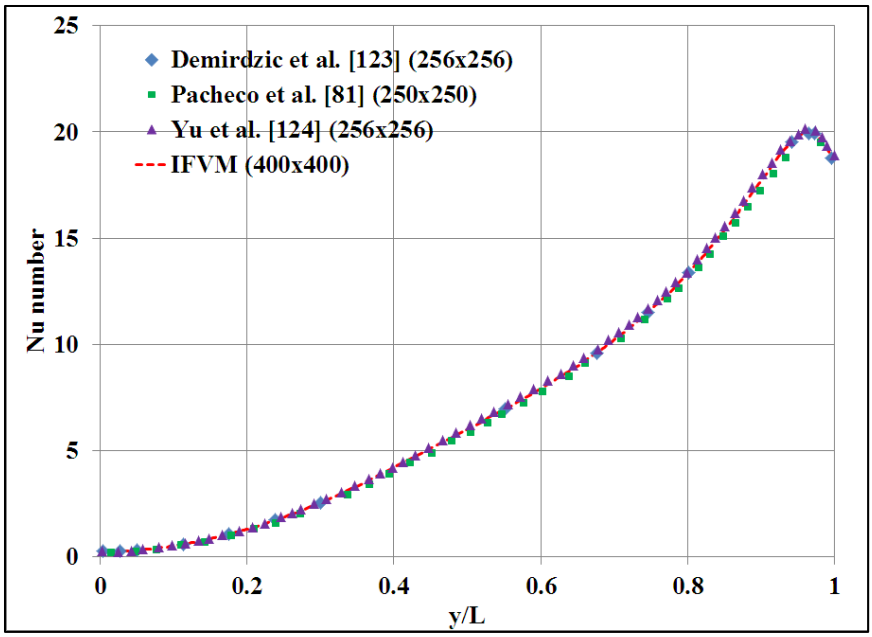

Figure 87. Comparison of the cold wall Nusselt distribution for the two-dimensional natural convection in a square cavity with an eccentric cylinder at $Ra = 10^6$ and $Pr = 10$.

4.7 Three-Dimensional Natural Convection in a Cubic Enclosure

The next step in the validation of the implemented finite volume immersed boundary method solver is to test the natural convection problem in a three-dimensional configuration. A reference benchmark is the three-dimensional natural convection in a cubic enclosure, which is an extension of the two-dimensional natural convection in a square cavity considered in Section 4.5. For this configuration one lateral wall it at constant temperature T_{hot} and the other lateral wall is at constant temperature T_{cold}. All the other walls are considered adiabatic. In Figure 88 is shown a schematic of the geometry and boundary conditions imposed for the three-dimensional natural convection in a cubic enclosure.

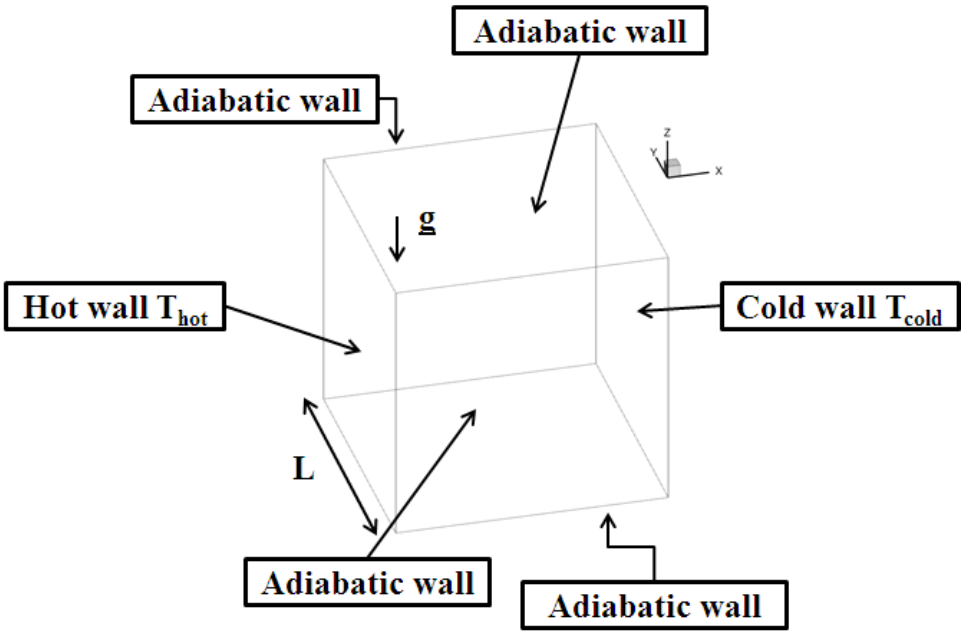

Figure 88. Schematic of the computational domain and boundary conditions used for the three-dimensional natural convection in a cubic enclosure.

Simulations were performed at Rayleigh number $Ra = 10^3$, 10^4 and 10^5, with a Prandlt number $Pr = 0.7$, corresponding to that of air. The cubic enclosure reference length is L. U_c represents the enclosure characteristic velocity as defined in eq. (155). In Table 59 are shown the air physical property and the domain geometrical configurations used for the three-dimensional natural convection in a cubic enclosure at $Ra = 10^3$, 10^4 and 10^5, respectively. The temperature difference $\Delta T = T_{hot}$-T_{cold} was the fixed parameter in the Rayleigh number. The enclosure Rayleigh numbers were matched by changing the cavity reference length as shown in Table 59.

Table 59 Air physical property and geometric configurations used for the three-dimensional natural convection in a cubic enclosure at $Ra = 10^3$, 10^4 and 10^5

Ra	10^3	10^4	10^5
ρ (kg/m^3)	1.205	1.205	1.205
μ (Pa•s)	$1.790 \bullet 10^{-5}$	$1.790 \bullet 10^{-5}$	$1.790 \bullet 10^{-5}$
c_p (J/ kg K)	1005.0	1005.0	1005.0
k (W / m K)	0.0257	0.0257	0.0257
β (1/K)	0.00343	0.00343	0.00343
L (m)	0.009785	0.021081	0.045418
U_c (m/s)	0.05738	0.08422	0.12362

The calculated surface-averaged Nusselt number at the hot wall determined with the IFVM approach and STAR-CCM+ was compared with the values of Fusegi et al. [124], Ha and Jung [128] and Yoon et al. [129]. In Table 60 is shown the IFVM mesh

convergence study of the surface-averaged Nusselt number at the hot wall for the three-dimensional natural convection in a cubic enclosure. The results show that mesh convergence is reached for the three finest mesh refinements for all Rayleigh numbers considered. Simulations of the three-dimensional natural convection in a cubic enclosure were performed with STAR-CCM+ with a mesh resolution of 300x300x300. In Table 61 the calculated hot wall surface-averaged Nusselt number for the IFVM and STAR-CCM+ are compared to the data from Fusegi et al. [124], Ha and Jung [128] and Yoon et al. [129]. All the computations are substantially in good agreement, with STAR-CMM+ slightly overestimating the Nusselt number at the hot wall. In particular excellent agreement of the IFVM results if found with the computations form Yoon et al. [129].

Table 60 IFVM mesh sensitivity analysis for the surface-averaged Nusselt number at the cubic enclosure hot wall for $Ra = 10^3$, 10^4 and 10^5

Fluid #cells	Δx	$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$
10x10x10	L/10	1.091	2.295	5.261
20x20x20	L/20	1.076	2.118	4.738
40x40x40	L/40	1.072	2.070	4.441
80x80x80	L/80	1.071	2.058	4.361
100x100x100	L/100	1.071	2.057	4.351
200x200x200	L/200	1.071	2.055	4.338
300x300x300	L/300	1.071	2.055	4.336

Table 61 Comparison of the surface-averaged Nusselt number at the cubic enclosure hot wall for $Ra = 10^3$, 10^4 and 10^5

	$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$
Fusegi et al. [124]	1.085	2.100	4.361
Ha et al. [128]	1.072	2.070	4.464
Yoon et al. [129]	1.072	2.055	4.339
Present (IFVM)	1.071	2.055	4.336
Present (STAR-CCM+)	1.080	2.070	4.463

In Figure 89 is shown the IFVM temperature distribution at the cubic enclosure boundaries for the $Ra = 10^3$, 10^4 and 10^5 case, respectively. At low Rayleigh numbers the heat flux across the cavity is mostly controlled by diffusion with convection being negligible (left of Figure 89). As the Rayleigh number increases, buoyancy becomes larger with convection being the predominant heat transfer mode (right of Figure 89).

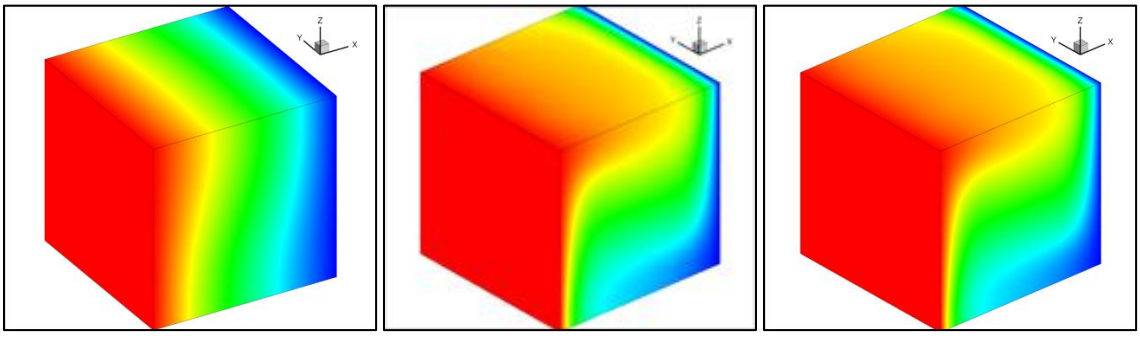

Figure 89. IFVM boundary temperature distribution for the three-dimensional natural convection in a cubic enclosure at $Ra = 10^3$ (left), 10^4 (center) and 10^5 (right).

In Figure 90 are shown the IFVM predictions of temperature contours at the enclosure symmetry plane (i.e., y = 0.5) for the $Ra = 10^3$, 10^4 and 10^5 cases, respectively. The figure shows the effect of buoyancy on the temperature contours and the formation of the thermal boundary layer at the enclosure hot wall as the Rayleigh number is increased.

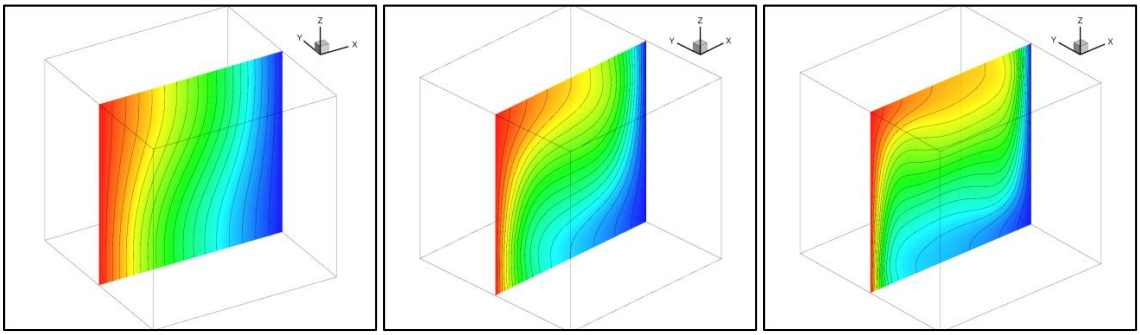

Figure 90. IFVM symmetry plane (y = 0.5) temperature distribution for the three-dimensional natural convection in a cubic enclosure at $Ra = 10^3$ (left), 10^4 (center) and 10^5 (right).

The velocity distribution inside the three-dimensional enclosure is very complex. In general for low Rayleigh numbers the fluid moves upwards close to the cavity hot wall due to buoyancy, and downwards close to the cavity cold wall due to the larger air density in this region, with a stagnant region in the center of the enclosure as the velocity magnitude contours show in Figure 91. Increasing the Rayleigh number, the flow patterns become more and more complicated. At low Rayleigh numbers, the flow rotates clockwise in the cavity on parallel (x, z)-planes and the number of streamlines crossing from one plane to another is very limited, as the streamlines plot shows in the left of

Figure 92. Increasing the Rayleigh number, the flow becomes unstable and the y-symmetry is progressively lost, with streamlines crossing from one (x, z)-plane to another, as shown in the right of Figure 92.

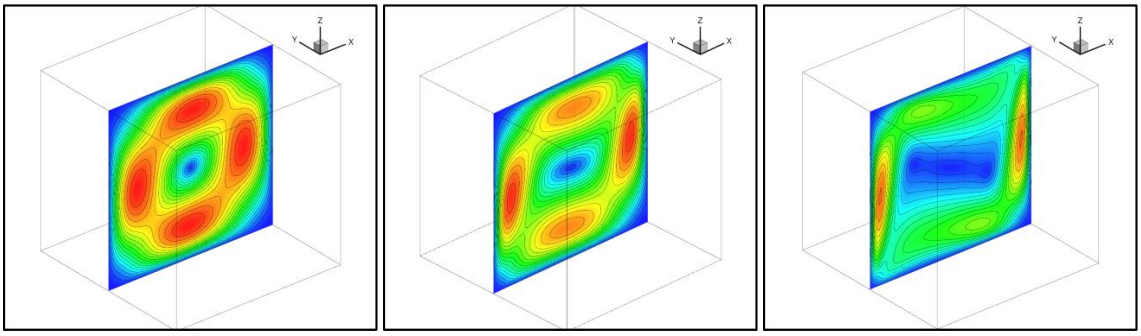

Figure 91. IFVM symmetry plane (y = 0.5) velocity magnitude distribution for the three-dimensional natural convection in a cubic enclosure at $Ra = 10^3$ (left), 10^4 (center) and 10^5 (right).

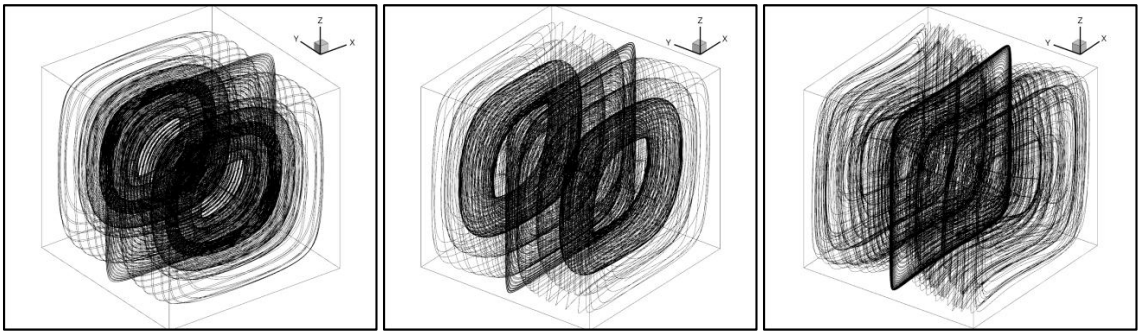

Figure 92. IFVM streamlines for the three-dimensional natural convection in a cubic enclosure at $Ra = 10^3$ (left), 10^4 (center) and 10^5 (right).

To have a quantitative validation of the IFVM, we plotted the temperature distribution determined with the IFVM and 100x100x100 mesh size against the STAR-CCM+ numerical predictions for a 300x300x300 mesh size at different locations in the

enclosure. In Figure 93 are shown the temperature predictions at the enclosure symmetry plane (y = 0.5) for different axial locations at $Ra = 10^3$ determined with the IFVM and STAR-CCM+, respectively. In Figure 94 are shown the IFVM and STAR-CCM+ numerical results for the axial temperature distribution at the $(x, y) = (0.5, 0.5)$ location for $Ra = 10^3$. For this low Rayleigh number, the predominant heat transfer mode is conduction and the IFVM temperature distribution across the three-dimensional enclosure is in excellent agreement with the STAR-CCM+ predictions.

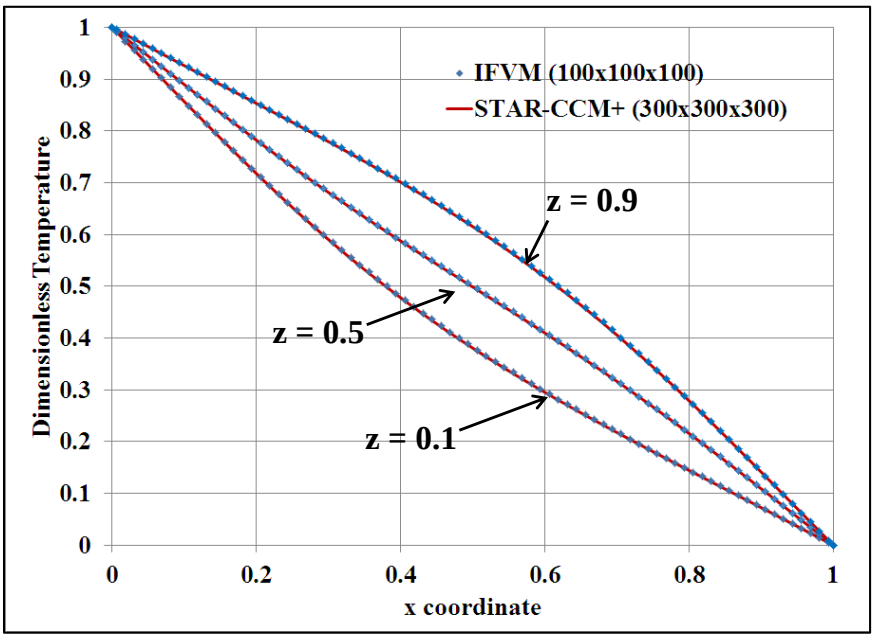

Figure 93. IFVM and STAR-CCM+ comparison of the temperature distribution at the symmetry plane y = 0.5 for different z positions ($Ra = 10^3$).

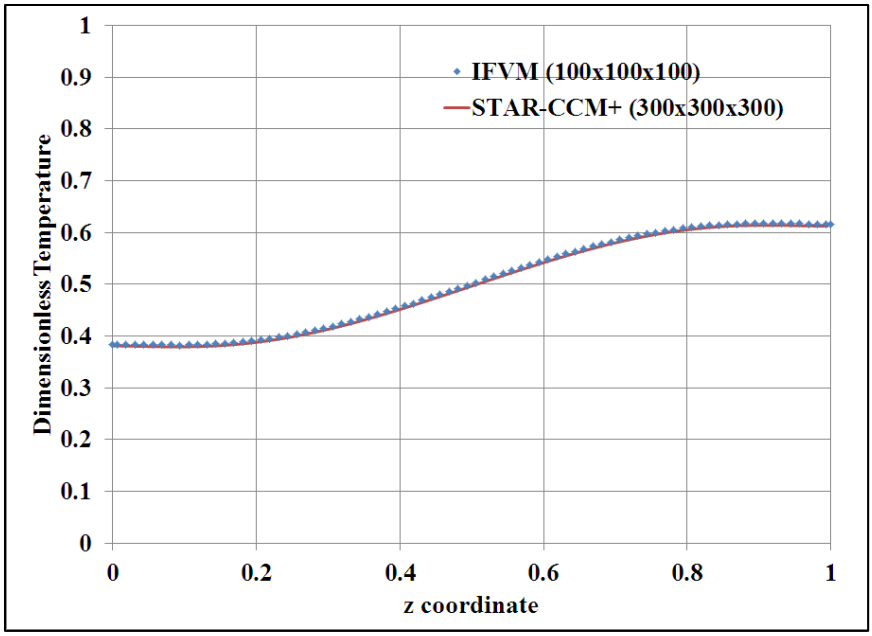

Figure 94. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location (Ra = 10^3).

Figure 95 and Figure 96 show the IFVM and STAR-CCM+ temperature predictions at the symmetry plane (y = 0.5) for different axial locations and in the axial direction at the (x, y) = (0.5, 0.5) location for Ra = 10^4, respectively. Also for this mixed conduction/convection heat transfer mode the IFVM temperature distribution across the cavity is very close to that predicted by STAR-CCM+.

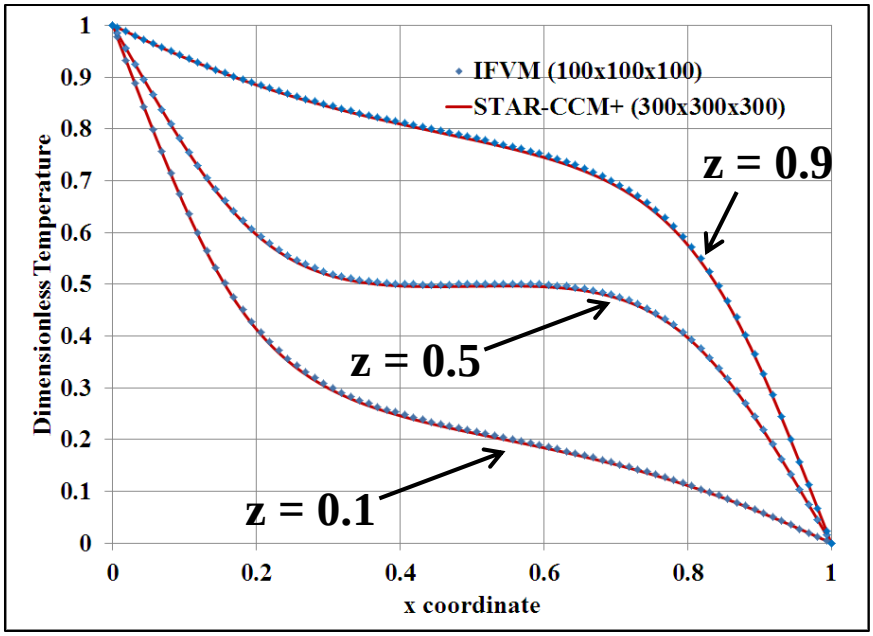

Figure 95. IFVM and STAR-CCM+ comparison of the temperature distribution at the symmetry plane y = 0.5 for different z positions ($Ra = 10^4$).

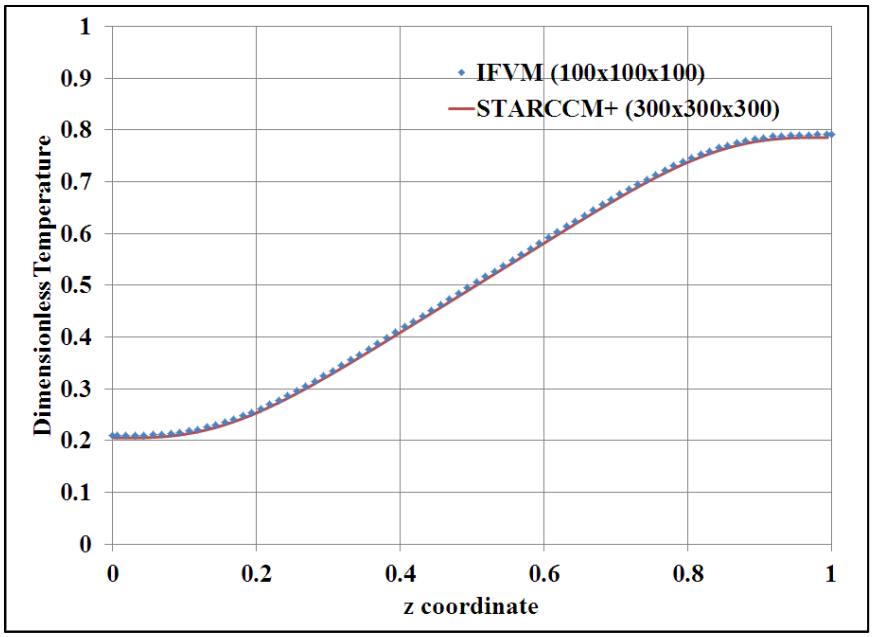

Figure 96. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location ($Ra = 10^4$).

In Figure 97 and Figure 98 are shown the IFVM and STAR-CCM+ temperature predictions at the symmetry plane (y = 0.5) for different axial location and in the axial direction at the (x, y) = (0.5, 0.5) location for $Ra = 10^5$, respectively. For this convection dominated heat transfer mode the IFVM predicts a slightly larger temperature distribution across the cavity if compared to the STAR-CCM+ numerical results. This disagreement can be explained by considering the surface-averaged Nusselt number at the hot wall predictions from the IFVM, STAR-CCM+, and other numerical data available in literature shown in Table 61. The Nu number at the hot wall represents a non-dimensional heat flux at that boundary. STAR-CCM+ numerical results point out that a larger heat flux at the hot wall is predicted. The heat flux at the hot wall is directly related to the gradient of the temperature normal to the wall. At the hot wall location the temperature is kept constant. This means that the temperature at the cells close to the enclosure hot wall predicted by STAR-CCM+ is lower than the one predicted by the IFVM and the other numerical data shown in Table 61. This trend is shown throughout the entire cavity, with STAR-CCM+ temperature distribution slightly below the IFVM predictions. If the surface-averaged Nusselt number at the hot wall is plotted against the number of iterations for STAR-CCM+, it is possible to see that after a relatively short transient with a sharp decrease in the Nusselt number, the solution does not reaches a converged value after 100,000 iterations. We expect to see a fully converged solution after about 1,000,000 iterations, for which the STAR-CCM+ surface-averaged Nusselt number at the hot wall prediction is in agreement with the other numerical data, and the temperature distribution across the cavity matches the IFVM results.

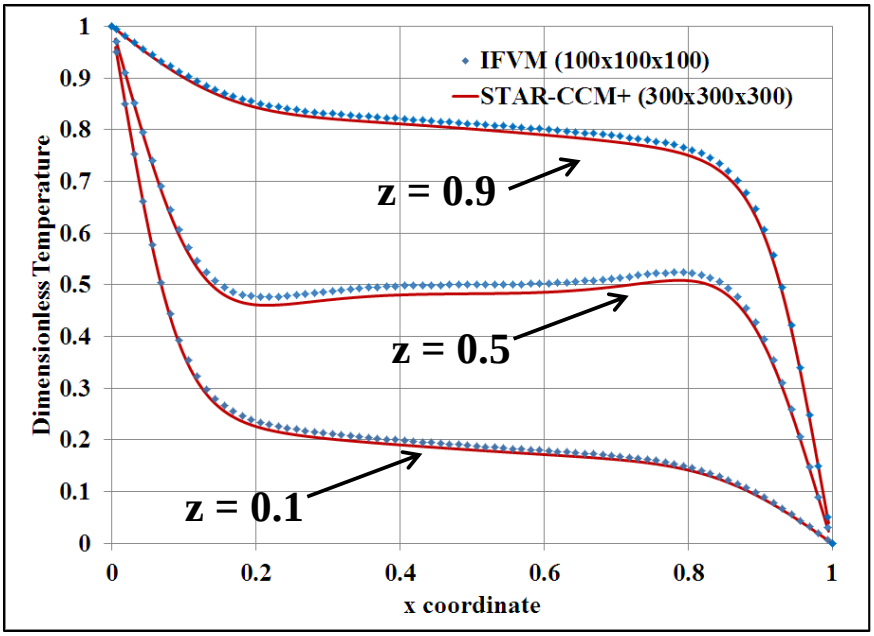

Figure 97. IFVM and STAR-CCM+ comparison of the temperature distribution at the symmetry plane y = 0.5 for different z positions ($Ra = 10^5$).

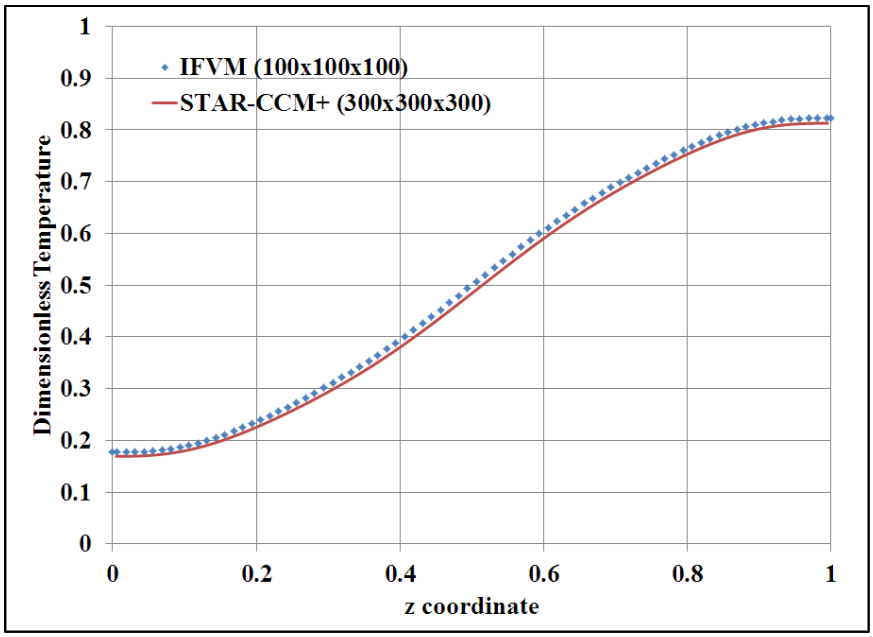

Figure 98. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location ($Ra = 10^5$).

4.8 Three-Dimensional Natural Convection in a Cubic Enclosure with an Embedded Sphere

To validate the IFVM approach for fluid flow and heat transfer problems, we considered the three-dimensional natural convection in a cubic enclosure with an embedded sphere at the center of the domain benchmark case. The cubic enclosure has reference length L and a sphere of radius $R = 0.2\ L$ is embedded at the center of the enclosure. The enclosure external boundary walls are at fixed Temperature T_{cold}, and the sphere wall is at temperature T_{hot}. No-slip boundary conditions are imposed at enclosure boundary walls and at the internal sphere wall. In Figure 99 is shown a schematic of the computational domain and boundary conditions used for this benchmark problem.

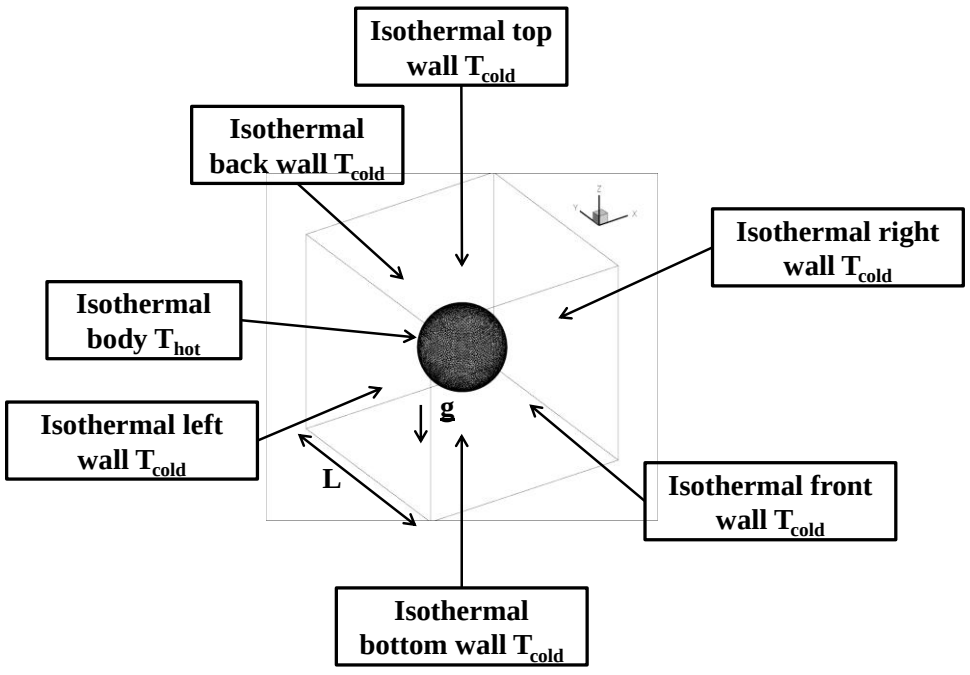

Figure 99. Schematic of the computational domain and boundary conditions used for the three-dimensional natural convection in a cubic enclosure with a sphere.

Simulations were performed at Rayleigh numbers $Ra = 10^3$, 10^4, 10^5 and 10^6, respectively. The Prandlt number was set equal to $Pr = 0.7$ corresponding to that of air. The temperature difference $\Delta T = T_{hot}$-$T_{cold} = 10$ K was the fixed parameter. The enclosure Rayleigh numbers were matched by changing the cavity reference length as shown in Table 62.

Table 62 Air physical property and geometric configurations used for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^3$, 10^4, 10^5 and 10^6

Ra	10^3	10^4	10^5	10^6
ρ (kg/m^3)	1.205	1.205	1.205	1.205
μ (Pa•s)	$1.790 \bullet 10^{-5}$	$1.790 \bullet 10^{-5}$	$1.790 \bullet 10^{-5}$	$1.790 \bullet 10^{-5}$
c_p (J/ kg K)	1005.0	1005.0	1005.0	1005.0
k (W / m K)	0.0257	0.0257	0.0257	0.0257
β (1/K)	0.00343	0.00343	0.00343	0.00343
L (m)	0.009785	0.021081	0.045418	0.09785
U_c (m/s)	0.05738	0.08422	0.12362	0.018145
R (m)	0.001957	0.0042162	0.0090836	0.01957

The surface-averaged Nusselt number at the enclosure side wall was taken as reference parameter to compare the IFVM predictions with the STAR-CCM+ results. In Table 63 is shown the mesh sensitivity study for the IFVM in the range of Rayleigh numbers considered. The results show that mesh convergence was reached for the two

finest mesh resolutions for all Rayleigh numbers considered. Simulations were performed with STAR-CCM+ considering a 100x100x100 grid resolution. In Table 64 are shown the IFVM and STAR-CCM+ predictions for the surface-averaged Nusselt number at the enclosure side wall at $Ra = 10^3$, 10^4, 10^5 and 10^6, respectively. The IFVM and STAR-CCM+ calculated Nusselt numbers are in good agreement.

Table 63 IVEM mesh sensitivity analysis for the side wall surface-averaged Nusselt number for the three-dimensional cubic enclosure with a sphere for $Ra = 10^3$, 10^4, 10^5 and 10^6

Fluid #cells	Δx	$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$	$Ra = 10^6$
10x10x10	L/10	0.620	0.589	0.691	1.278
20x20x20	L/20	0.639	0.617	0.656	1.638
40x40x40	L/40	0.642	0.623	0.649	1.102
80x80x80	L/80	0.643	0.624	0.653	1.048
100x100x100	L/100	0.643	0.625	0.655	1.031

Table 64 Comparison of the side wall surface-averaged Nusselt number for the three-dimensional cubic enclosure with a sphere for $Ra = 10^3$, 10^4, 10^5 and 10^6

	$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$	$Ra = 10^6$
Present (IFVM)	0.643	0.625	0.645	1.031
Present (STAR-CCM+)	0.637	0.618	0.646	0.939

4.8.1 Ra = 10^3, Flow and Temperature Fields

For this low Rayleigh number the temperature contours shown in Figure 100 and the streamlines shown in Figure 101 are almost symmetric about the horizontal and vertical centers of the enclosure. This result comes with no surprise since the heat transfer mode is dominated by conduction, with the effect of convection being negligible. The temperature and flow distribution across the enclosure determined with the IFVM is qualitatively very close to the STAR-CCM+ predictions as Figure 100 and Figure 101 show. To have a quantitative estimate of the IFVM predictions, we plotted in Figure 102 the temperature distribution at the symmetry plane (y = 0.5) for different axial locations obtained with the IFVM approach and STAR-CCM+, respectively. Figure 102 shows that the IFVM numerical results at z = 0.1 and z = 0.9 are in excellent agreement with the STAR-CCM+ predictions. Particularly interesting is the temperature distribution on the symmetry plane (y = 0.5) for z = 0.5 for which we can show that the IFVM the STAR-CCM+ numerical results are in good agreement approaching the immersed body hot wall surface. Since this is a conduction dominated heat transfer mode, the effect of buoyancy (i.e., velocity field) is negligible and, therefore, we can state that the IFVM approach is properly solving a Laplace problem for the temperature filed. The same argumentation applies to the temperature distribution in the axial direction at the (x, y) = (0.5, 0.5) location shown in Figure 103. The IFVM predictions are in excellent agreement with the STAR-CCM+ numerical results all the way to the sphere hot wall.

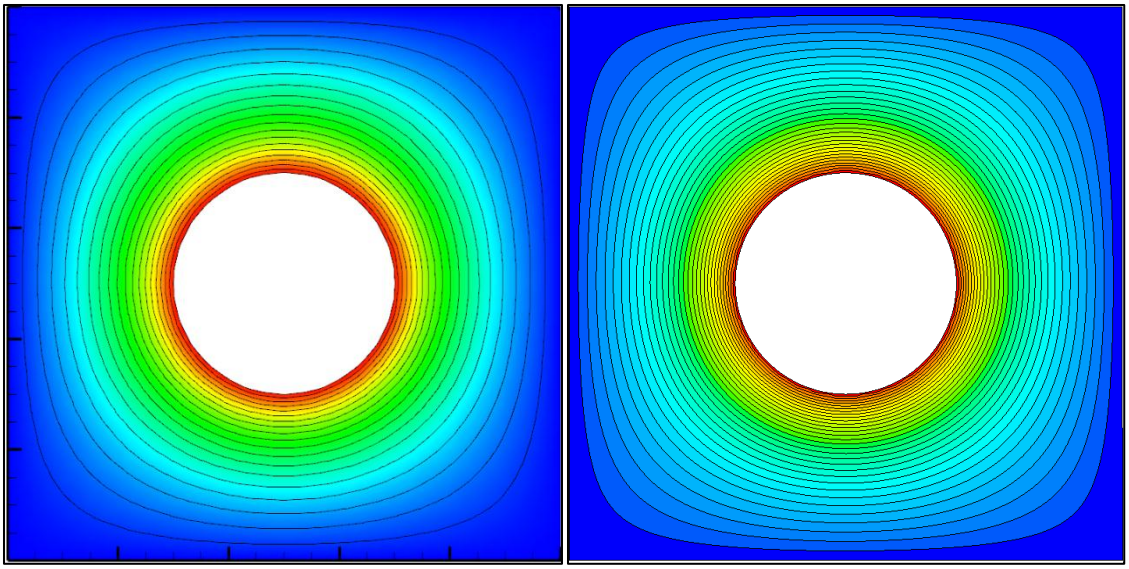

Figure 100. Symmetry plane (y = 0.5) temperature contours for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^3$. Left: IFVM; right: STAR-CCM+.

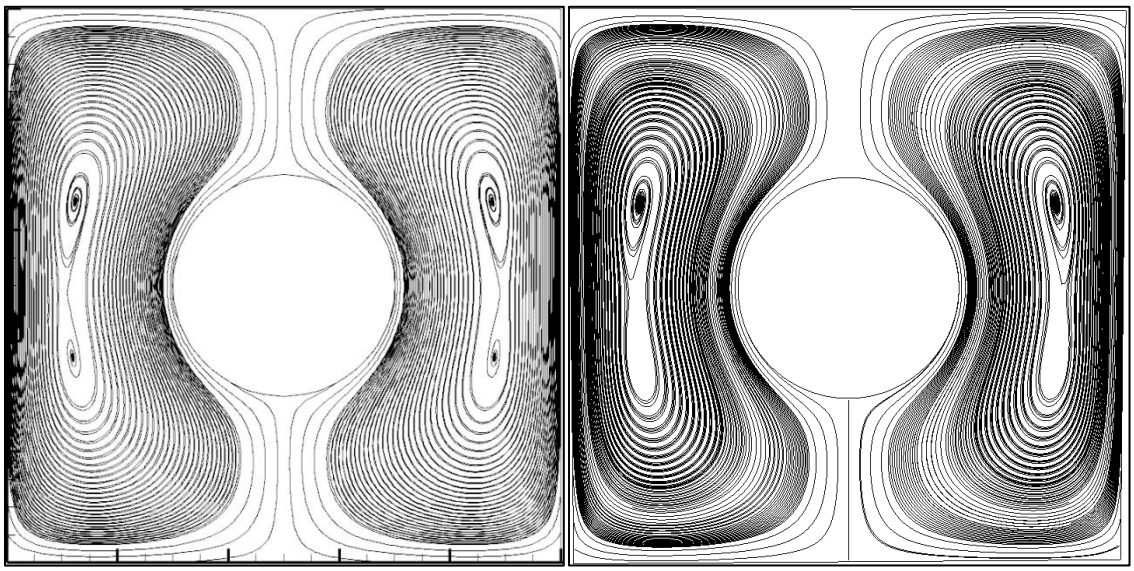

Figure 101. Symmetry plane (y = 0.5) streamlines for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^3$. Left: IFVM; right: STAR-CCM+.

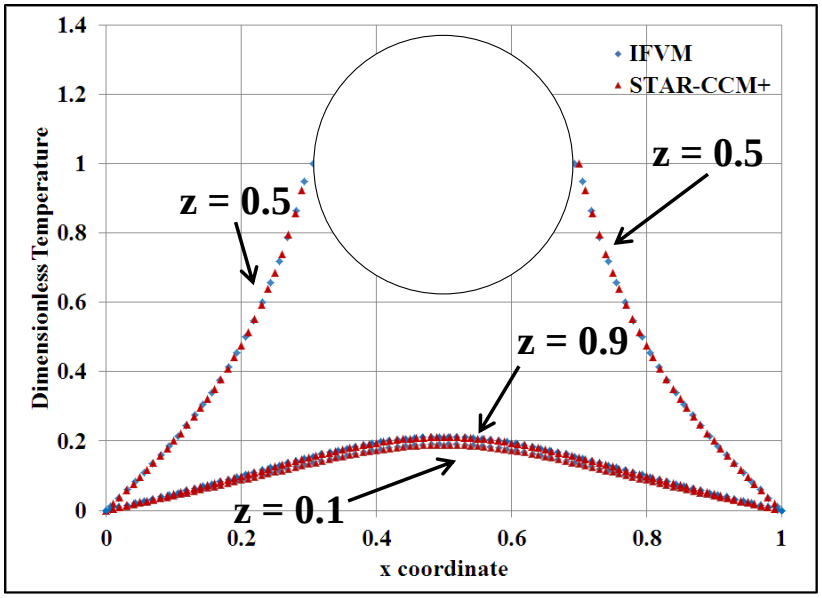

Figure 102. IFVM and STAR-CCM+ comparison of the temperature distribution at the symmetry plane y = 0.5 (different axial positions) for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^3$.

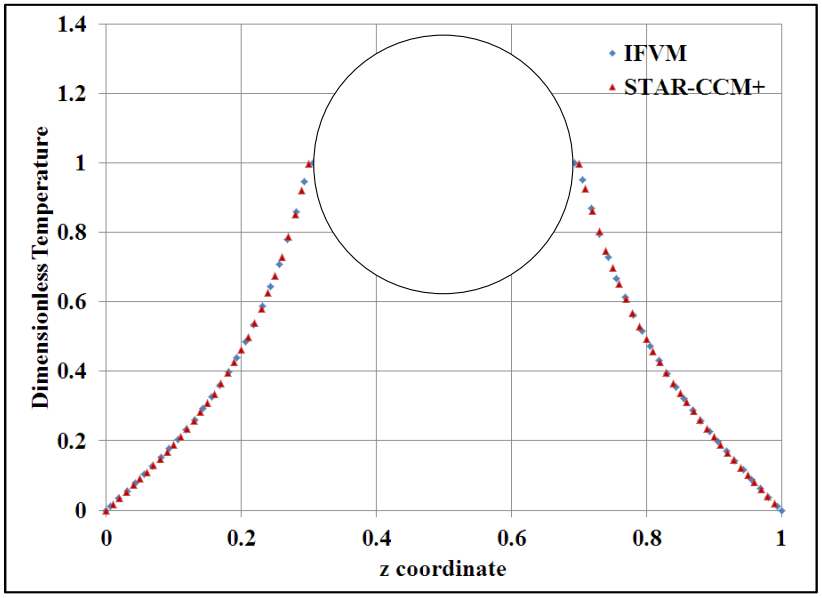

Figure 103. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^3$.

4.8.2 Ra = 10^4, Flow and Temperature Fields

As the Rayleigh number is increased to *Ra* = 10^4, a very weak upward thermal plume develops at the sphere hot wall. The temperature contours shown in Figure 104 and streamlines shown in Figure 105 for the IFVM approach and STAR-CCM+ are qualitatively very similar.

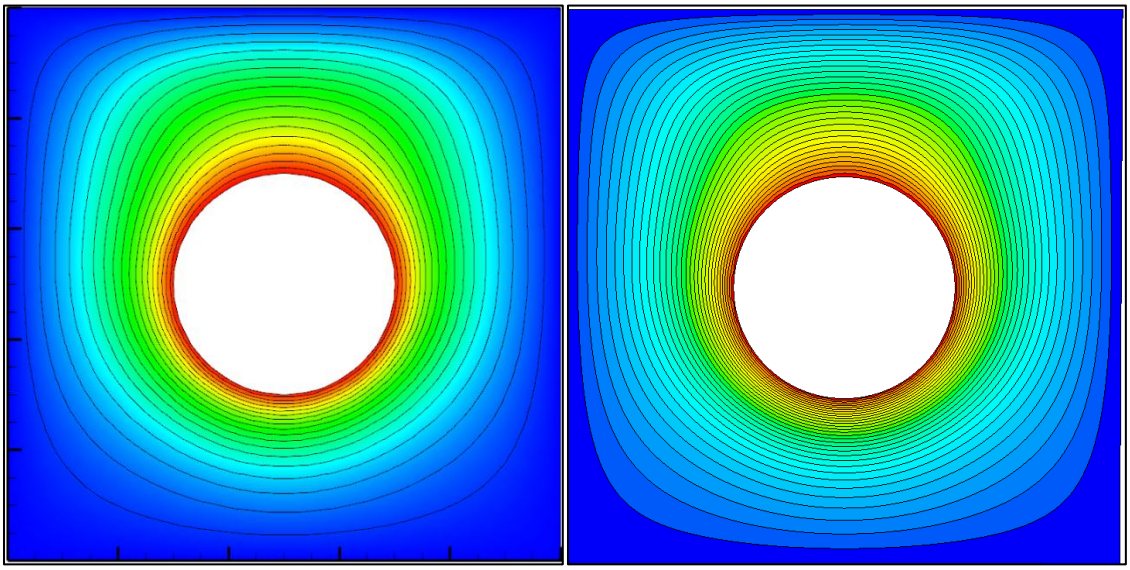

Figure 104. Symmetry plane (y = 0.5) temperature contours for the three-dimensional natural convection in a cubic enclosure with a sphere at *Ra* = 10^4. Left: IFVM; right: STAR-CCM+.

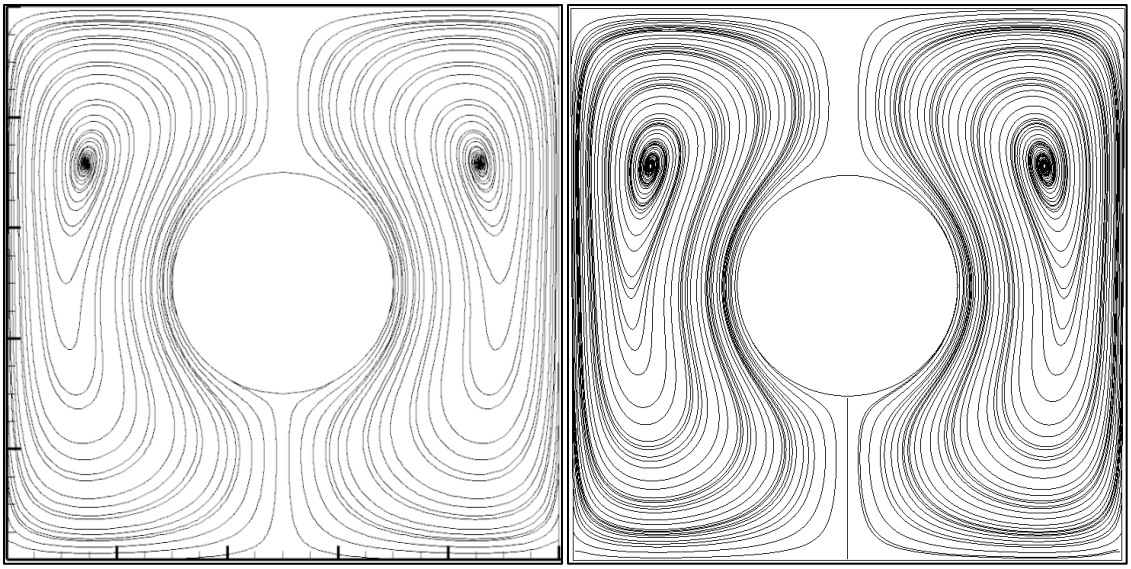

Figure 105. Symmetry plane (y = 0.5) streamlines for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^4$. Left: IFVM; right: STAR-CCM+.

The temperature distribution at the symmetry plane (y = 0.5) obtained with the IFVM approach is in excellent agreement with the numerical data from STAR-CCM+ for $z = 0.1$, $z = 0.5$ and $z = 0.9$, as shown in Figure 106. Buoyancy forces are driving the flow, but the effect of convection on the heat transfer mode is still limited. Close to the immersed body hot surface, the local temperature distribution predicted by the IFVM approach is very close to the one predicted by the body-fitted commercial code STAR-CCM+ both in the *x*-direction (see Figure 106) and in the *z*-direction (see Figure 107).

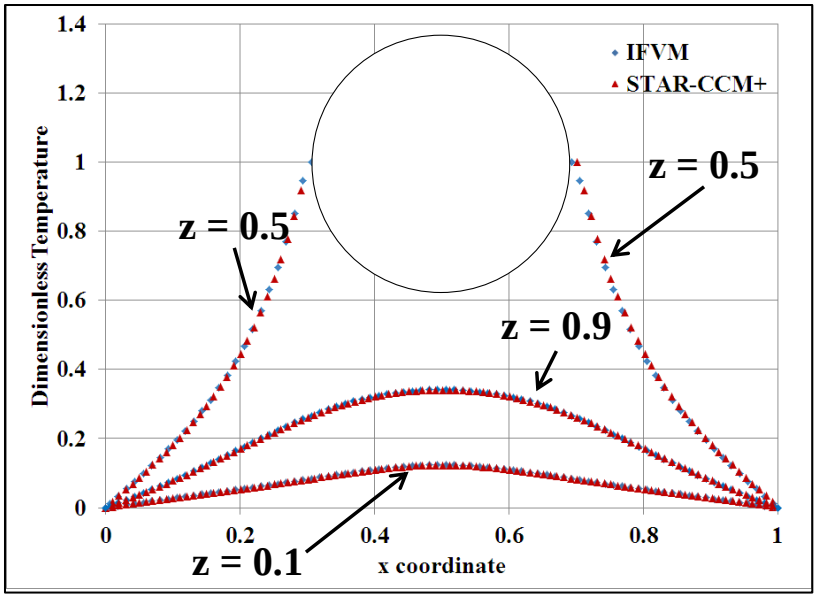

Figure 106. IFVM and STAR-CCM+ comparison of the temperature distribution at the symmetry plane y = 0.5 (different axial positions) for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^4$.

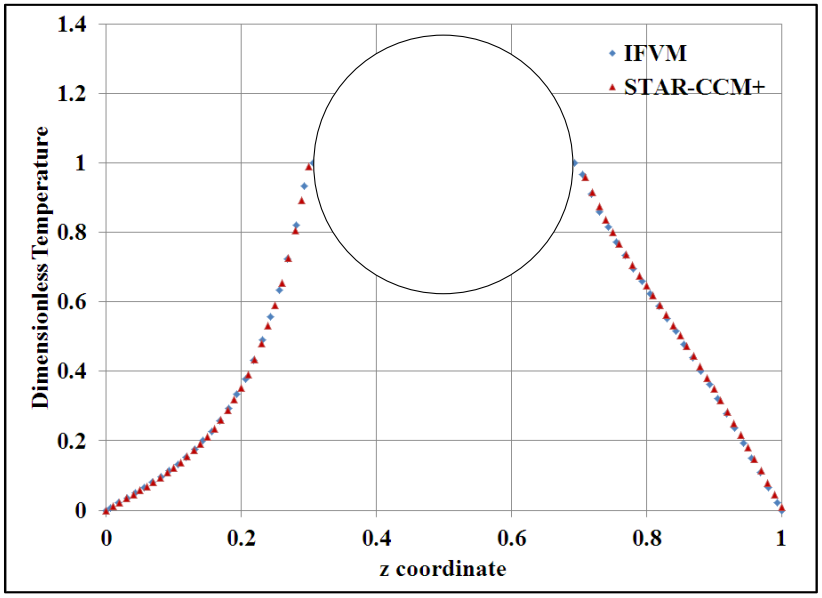

Figure 107. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^4$.

4.8.3 Ra = 10^5, Flow and Temperature Fields

As the Rayleigh number reaches *Ra* = 10^5, the buoyant plume developing along the hot sphere walls is clearly visible in Figure 108 showing the temperature contours at the symmetry plane (*y* = 0.5) for the IFVM approach and STAR-CCM+, respectively.

Convection dominates the heat transfer mode. The dominant flow is in the upper part of the enclosure, and correspondingly the core of the recirculating eddies is located only in the upper half, as the streamlines plot at the symmetry plane (*y* = 0.5) show in Figure 109. Qualitatively there is good agreement between the IFVM results and STAR-CCM+ numerical predictions.

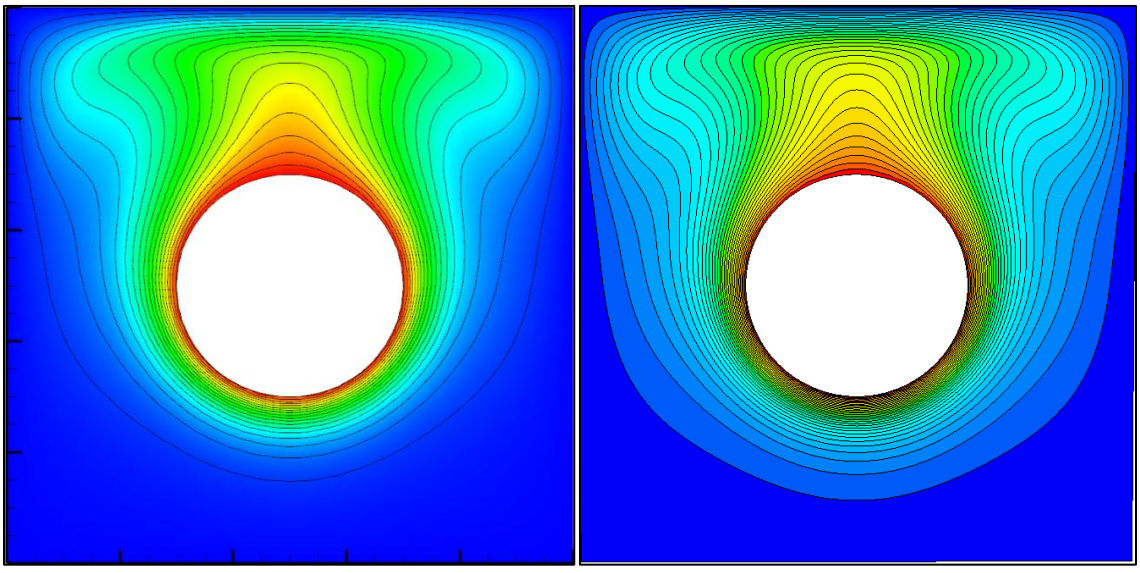

Figure 108. Symmetry plane (y = 0.5) temperature contours for the three-dimensional natural convection in a cubic enclosure with a sphere at *Ra* = 10^5. Left: IFVM; right: STAR-CCM+.

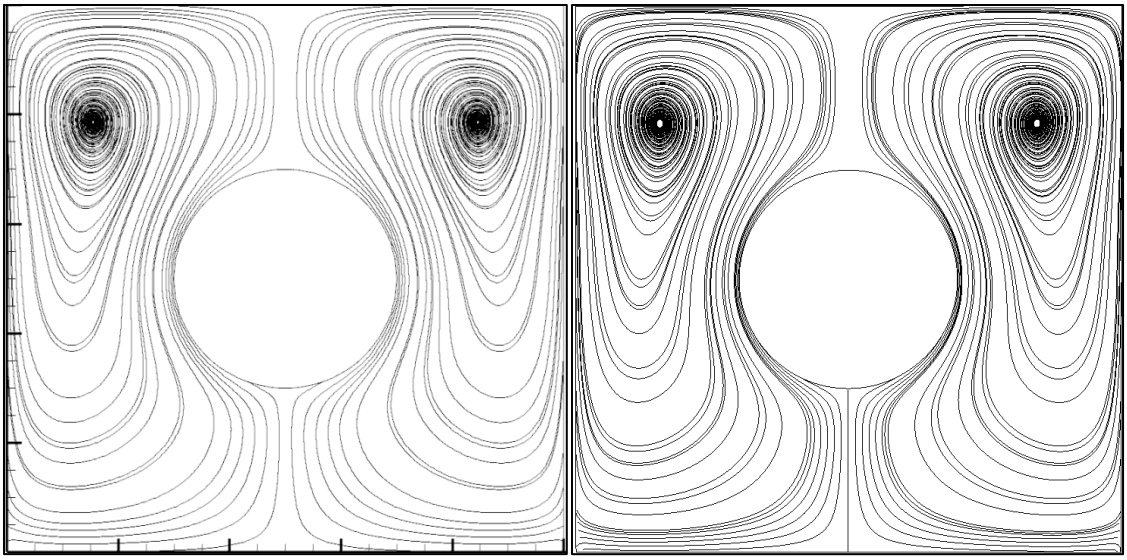

Figure 109. Symmetry plane (y = 0.5) streamlines for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^5$. Left: IFVM; right: STAR-CCM+.

Plotting the temperature distribution at the symmetry plane ($y = 0.5$) for different axial locations in Figure 110 and Figure 111, we confirm that even for flows where convection is the predominant heat transfer mode, the IFVM predictions are in excellent agreement with the results from STAR-CCM+ across the enclosure, and close to the sphere hot wall. Also for the axial temperature distribution at the $(x, y) = (0.5, 0.5)$ location shown in Figure 112, the predictions from the IFVM are in excellent agreement with the results for STAR-CCM+.

If the sharp interpolation scheme used for the IFVM were not able to give a satisfactory prediction for the temperature and/or velocity fields close to the immersed body interface, due to the strong coupling of the NSEs with the energy equation, the results from the IFVM approach would not agree with the STAR-CCM+ predictions.

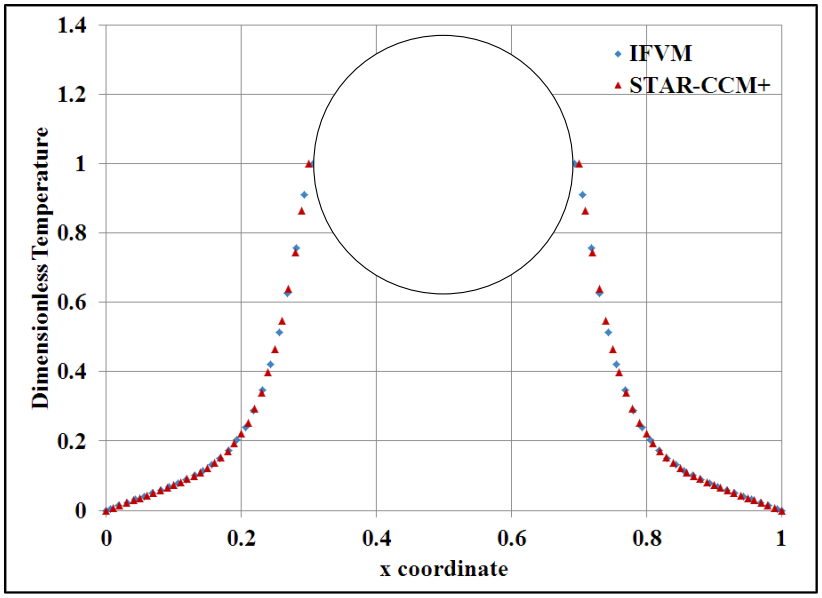

Figure 110. IFVM and STAR-CCM+ comparison of the temperature distribution in the *x* direction at the (*y*, *z*) = (0.5, 0.5) location for the three-dimensional cubic enclosure with a sphere at $Ra = 10^5$.

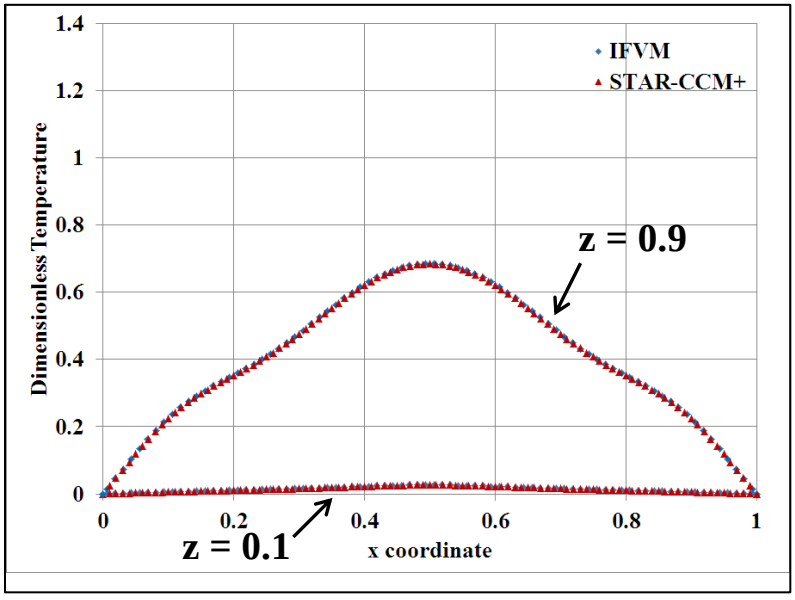

Figure 111. IFVM and STAR-CCM+ comparison of the temperature distribution in the *x* direction at the (*y*, *z*) = (0.5, 0.1) and (*y*, *z*) = (0.5, 0.9) locations for the three-dimensional cubic enclosure with a sphere at $Ra = 10^5$.

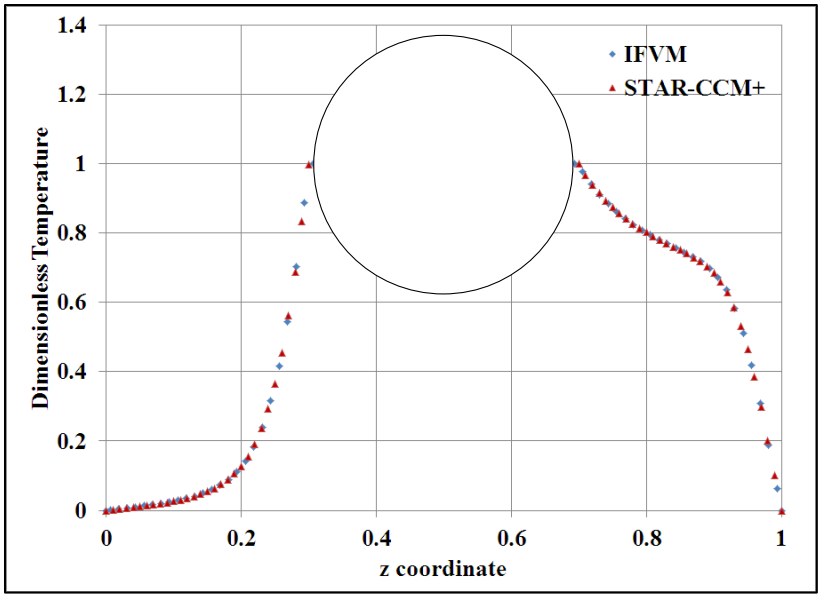

Figure 112. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location for the natural convection in a cubic enclosure with a sphere at $Ra = 10^5$.

4.8.4 Ra = 10^6, Flow and Temperature Fields

For Rayleigh number equal to $Ra = 10^6$ or larger, the heat transfer mode is completely dominated by convections. A very strong buoyant plume develops along the sphere hot wall as the temperature contours at the symmetry plane (y = 0.5) for the IFVM approach and STAR-CCM+ show in Figure 113. The isotherms are strongly distorted towards the upper part of the enclosure leading to a stable stratification region in the lower part. The temperature predictions from the IFVM approach are is good agreement with STAR-CCM+ numerical results. For these Rayleigh numbers the convection velocity is significantly larger than for the cases shown in the previous sections. From Figure 113 it is also possible to see the boundary layer formation at the sphere hot wall, and its separation from the surface near the top of the sphere which

determines the formation of a strong plume in the enclosure upper region. The core of the inner vortices moves upwards and towards the corner of the enclosure as the predictions from the streamlines at the symmetry plane (y = 0.5) plotted in Figure 114 show for both the IFVM approach and STAR-CCM+, respectively.

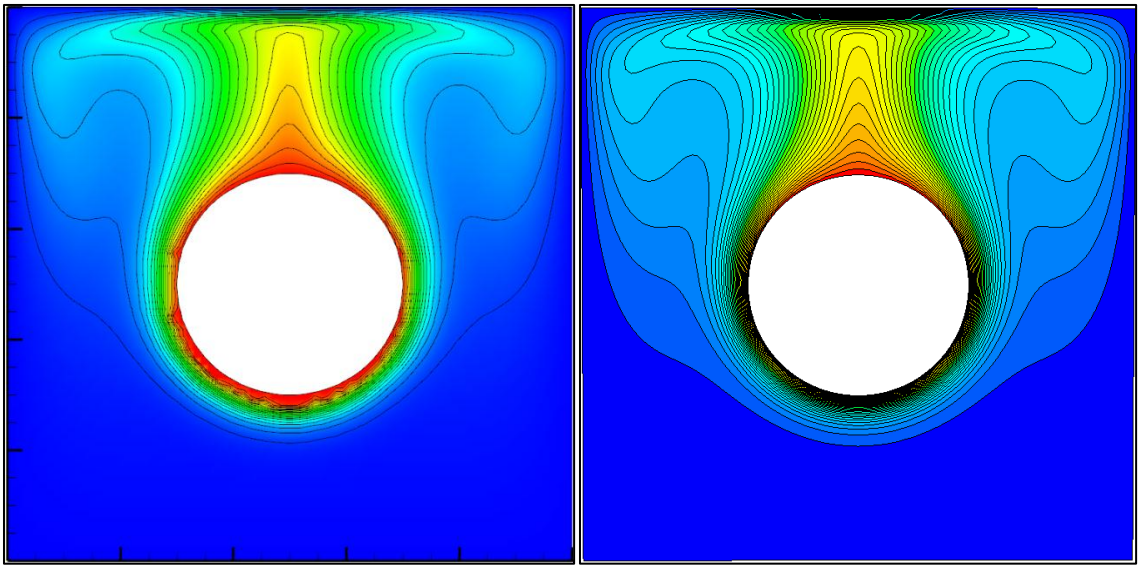

Figure 113. Symmetry plane (y = 0.5) temperature contours for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^6$. Left: IFVM; right: STAR-CCM+.

In Figure 115 is shown the temperature distribution at the symmetry plane (y = 0.5) and at the axial location z = 0.5 predicted with the IFVM approach and STAR-CCM+. The IFVM numerical results are in good agreement with STAR-CCM+ results in the enclosure region and close to the sphere hot wall.

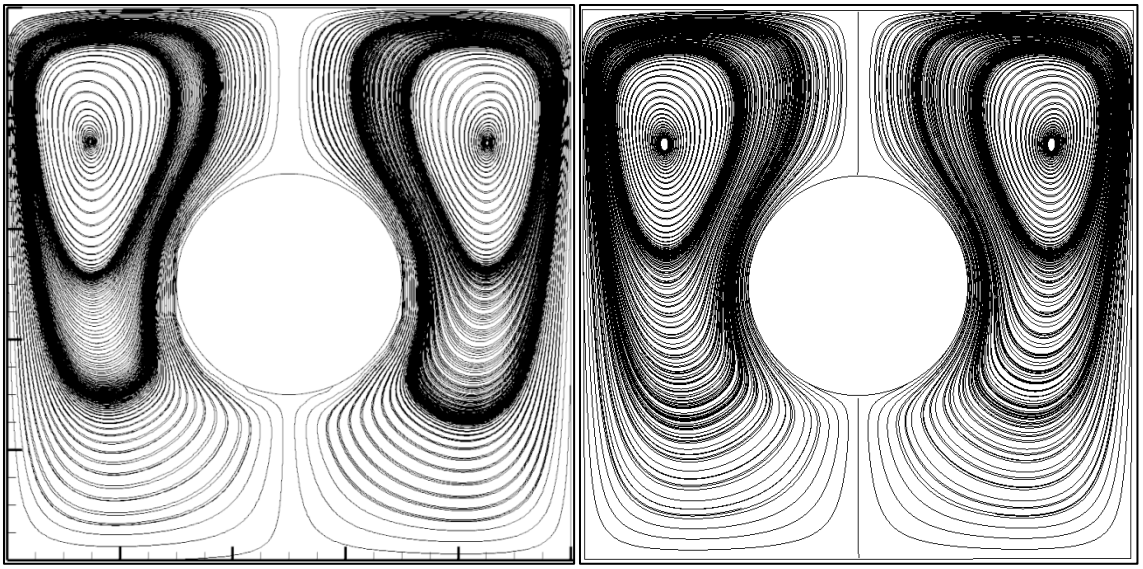

Figure 114. Symmetry plane (y = 0.5) streamlines for the three-dimensional natural convection in a cubic enclosure with a sphere at $Ra = 10^6$. Left: IFVM; right: STAR-CCM+.

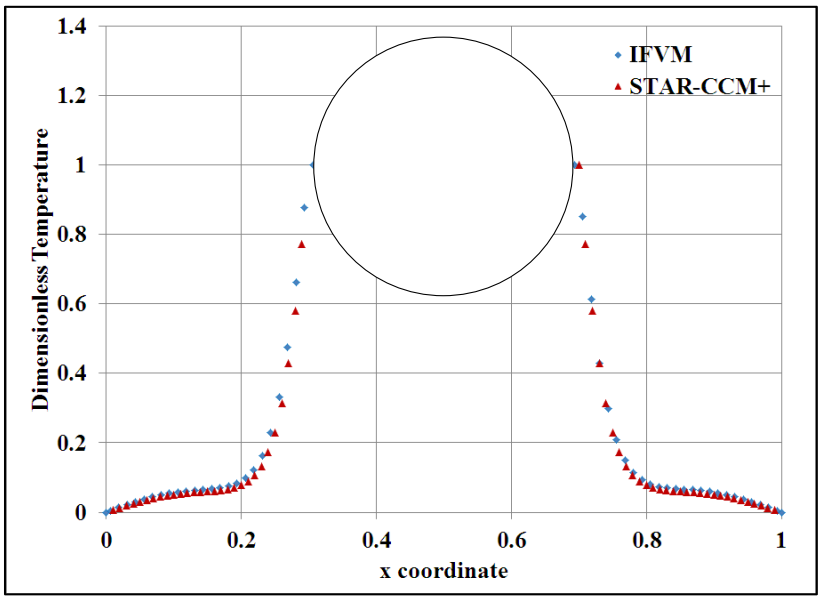

Figure 115. IFVM and STAR-CCM+ comparison of the temperature distribution in the x-direction at the (y, z) = (0.5, 0.5) location for the cubic enclosure with a sphere at $Ra = 10^6$.

Figure 116 shows the temperature predictions for the IFVM approach and STAR-CCM+ at the symmetry plane ($y = 0.5$) and at the axial locations $z = 0.1$ and $z = 0.9$, respectively. In the lower part of the cavity (i.e., $z = 0.1$) where stratification is present, the two approaches give very close predictions. In the upper part of the cavity (i.e., $z = 0.9$) there are some discrepancies between the IFVM predictions and the numerical data from STAR-CCM+. A closer look at Figure 113 reveals that the IFVM temperature distribution in proximity of the sphere hot wall shows some minor discontinuities. This unphysical effect is due to the coarse triangulation used for discretizing the sphere surface. If a finer mesh is used for the sphere to embed into the fluid Cartesian mesh, the discretization error due to the interpolation algorithm at the immersed boundary can be reduced, and the unphysical discontinuities in the temperature distribution would be smoothed out. The same considerations apply to the axial temperature distribution at the $(x, y) = (0.5, 0.5)$ location shown in Figure 117 for the IFVM approach and STAR-CCM+, respectively. In the bottom part of the cavity the IFMV predicts a temperature distribution slightly larger than STAR-CCM+ due to a too coarse mesh triangulation for the immersed body. In the upper part of the cavity this discrepancy disappears due to the reduced convection velocity right above the sphere upper part.

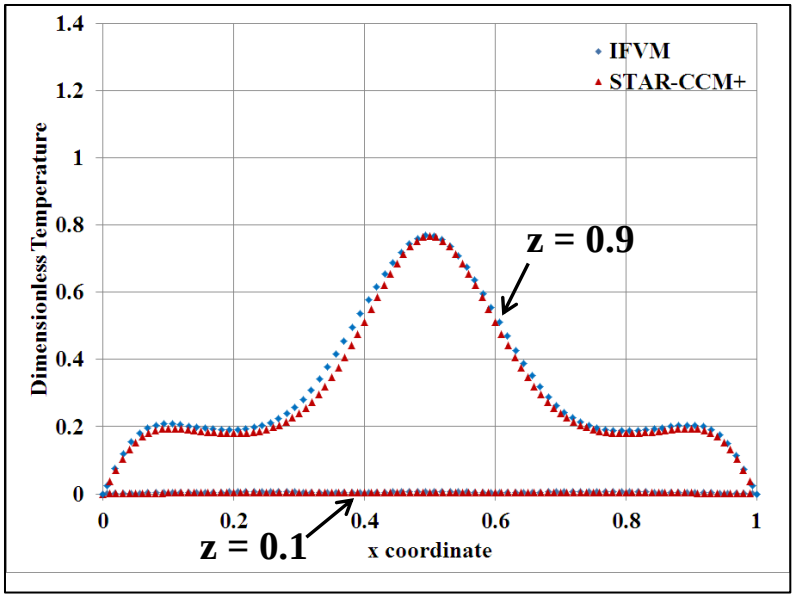

Figure 116. IFVM and STAR-CCM+ comparison of the temperature distribution in the x-direction at the (y, z) = (0.5, 0.1) and (y, z) = (0.5, 0.9) locations for the cubic enclosure with a sphere at $Ra = 10^6$.

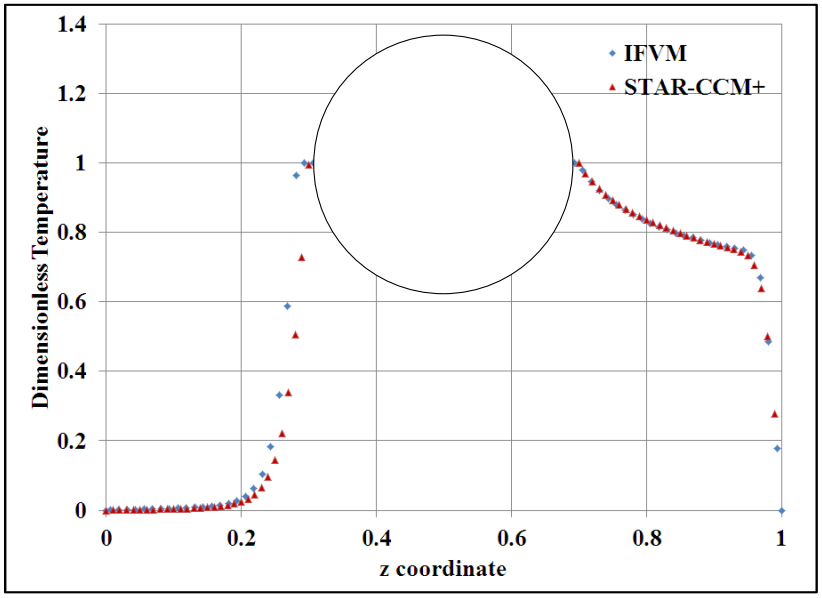

Figure 117. IFVM and STAR-CCM+ comparison of the temperature distribution in the vertical direction at the (x, y) = (0.5, 0.5) location for the natural convection in a cubic enclosure with a sphere at $Ra = 10^6$.

This slight discrepancy between the IFVM and STAR-CCM+ numerical results for the three-dimensional natural convection in a cubic enclosure with embedded body at large Rayleigh numbers points out that to have an accurate reconstruction on the fluid/solid interface from an interpolation point of view, immersed boundary methods need a fine triangulation mesh of the solid object to be embedded in the fluid region. If a coarse mesh is used, the interpolation scheme will inevitably determine approximate forcing terms at the ghost-cells, and this approximation will be reflected on the solution close to the fluid/solid interface.

5. IMMERSED BOUNDARY METHODS: IFEM VERSUS IFVM APPROACH

In this section we briefly discuss the strengths and weaknesses of the IFEM and IFVM approaches discussed in the previous sections. In Section 2 we addressed the mathematical formulation of the IFEM and in Section 3 we discussed the mathematical formulation of the IFVM. The use of Lagrange multipliers for the IFEM to be projected on the immersed body mesh, and the forcing terms to be interpolated back on the fluid volume mesh implies that two mesh discretizations are used to solve the problem at hand: one for the fluid and one for the immersed body. We can think of this as an Eulerian/Lagrangian approach, where the fluid flow equations are solved on an Eulerian mesh and the additional constraints due to the immersed body are solved on a Lagrangian mesh. On the other hand, in the IFVM we stressed the point that the forcing terms are directly applied at ghost-cells. There is no need to solve equations on an additional mesh, which is only needed to determine the location of the ghost-cells on the fluid mesh. Therefore, we can talk of an Eulerian IFVM approach meaning that we solve the fluid flow equations on an Eulerian mesh. Solving the discretized system of equations on one mesh is computationally more efficient than using two meshes.

In the IFEM approach we solve for the fluid flow equations on the entire computational domain, i.e., fluid and immersed body, and then we add additional constraints to the fluid cells overlapping the solid body. In the IFVM we solve only for the fluid region, neglecting the fluid cells overlapping with the immersed body, and

adding forcing terms directly to the ghost-cells. This represents a saving in terms of memory and number of unknowns to compute.

Concerning the interface scheme, we discussed the use of a diffuse interface scheme for the IFEM approach and a sharp interface scheme for the IFVM approach. Diffuse interface scheme require a projection of the field (e.g., velocity, temperature) from the Eulerian mesh to the Lagrangian mesh to determine the Lagrangian multipliers (i.e., forcing terms), and an interpolation of the forcing terms from the Lagrangian mesh back to the Eulerian mesh. The forcing term is "spread" or "diffused" back on the Eulerian mesh, hence the name diffuse interface scheme. In the sharp interface scheme we use the velocity field variables close to the fluid/solid interface and the constraint at the immersed body surface to determine the value of the forcing terms at the corresponding ghost-cells through an appropriate interpolation, where a ghost-cell is by definition a cell inside the solid body with at least one neighbor in the fluid region. The diffusive nature of the interface scheme for the IFEM will satisfy the constraint at the immersed body surface in an integral way, whereas the sharp nature of the interface scheme for the IFVM will satisfy the constraint at the fluid/solid interface in a local way. Since immersed boundary methods approximate the immersed boundary surface with a forcing term, the local nature of sharp interface schemes will results in a better approximation of the interface especially for coarse meshes. This is perhaps one of the key points in applying immersed boundary method to the solution of differential equations. The idea behind the immersed boundary approach is to use simple Cartesian meshes to represent complex geometries. By simple we mean from a numerical point of

view, i.e., the data structure of Cartesian meshes is order of magnitude less expensive that the one for body-fitted unstructured meshes. We also would like to use a reduced number of cells to simulate complex geometries. The use of coarser mesh reduces the computational time, making the simulation of very large domains possible. If we are forced by the interpolation scheme to use very fine meshes close to the immersed boundary interface, somehow we defeat the original goal of using immersed boundary methodologies. With a diffuse interface scheme we are forced to have a very fine mesh close to the fluid/solid interface, which can be easily accomplished with a body-fitted approach. With a sharp interface scheme, on the other hand, we directly impose the constraint at the fluid/solid interface as long as there are enough ghost-cells (i.e., forcing points) representing the geometric features of the immersed object we are trying to simulate, just like for body-fitted approaches where boundaries are properly represented only if the mesh discretization is fine enough. Therefore, the IFEM projection/interpolation process introduces an additional discretization error in the solution of the field equations, meanwhile the IFVM interpolation process leaves unchanged the base solver discretization error at the fluid/solid interface. This issue can be clarified considering the two-dimensional flow past a circular cylinder results shown in Section 4.3. The IFEM drag and lift coefficients are in satisfactory agreement with the IFVM predictions, STAR-CCM+ numerical results, and data from available literature. This means that from an integral point of view the IFEM predicts the right streamwise and cross-flow forces at the fluid/solid interface, just like the other approaches do. Then we mentioned the unphysical fluid penetration at the cylinder wall if a surface mesh is

used to discretize the immersed body in the IFEM approach. This unphysical behavior is due to the diffusive nature of the interface scheme used in the IFEM, where the no-slip condition at the fluid/solid interface is satisfied in an integral way across the cell. Locally the fluid velocity at the interface might not satisfy the no-slip boundary condition, but it does satisfy this constraint when integrated over the cell size. By refining the mesh close to the fluid/solid interface the number of Lagrange multipliers is increased and the quality of the solution at the immersed body surface improves (i.e., the number of streamlines crossing the fluid/solid interface is reduces). For the same problem we showed that by discretizing the cylinder with a volume mesh instead of a surface mesh the unphysical behavior of fluid penetration at the solid boundary is eliminated. Therefore, by using diffuse interface scheme we are forced to have a very fine mesh close to the immersed body interface, when our initial objective was to have a coarse representation of the solid object.

In Section 4.1 we showed that the IFEM is third order accurate in space and second order accurate in time, meanwhile the IFVM is second order accurate in space and first order accurate in time. One of the strength of finite element methods is to be numerically more accurate than finite volume methods, so we are not surprised by this result. On the other hand, the IFEM higher numerical accuracy comes with an order of magnitude increase in the computational effort if compared to the IFVM approach. From an engineering point of view this is perhaps the most limiting issue in the application of immersed boundary methods to finite element discretizations.

Another important issue is the numerical algorithm used to solve the time-dependent incompressible NSEs. In our IFEM we use the fractional step method coupled with the immersed boundary method to solve the NSEs, and we solve first for the velocity field and the Lagrange multipliers related to the no-slip constraint at the solid surface (diffusion step), then for the pseudo-pressure (projection step) and finally we correct the pressure field (correction step) with the value of the pseudo-pressure from the previous step. The projection and correction steps are necessary to guarantee the divergence-free constraint for the velocity field. The projection of the velocity vector on a divergence-free field might actually alter the no-slip constraint at the fluid/solid interface. On the other hand, for the IFVM, we use the SIMPLER algorithm to solve the time-dependent incompressible NSEs computed with a direct-forcing sharp interface immersed boundary method, where the forcing term is determined from the NSEs in a fully explicit way. The iterative nature of the SIMPLER algorithm determines a divergence-free velocity field upon convergence at every time step. The presence of an immersed boundary is discretized numerically by adding the forcing terms explicitly into the NSEs. At every time step, the NSEs are iterated until both divergence-free and no-slip constraints are satisfied to the desired residual accuracy.

The *deal.II* library [69] used for the IFEM implementation allows to construct the projection/interpolation operators with the same shape and test functions used to discretize the field equations, and no *ad hoc* projection and interpolation operators were necessary. In the IFVM approach, we had to develop an interpolation scheme to determine the forging terms at the ghost-cells.

Finally we end our discussion on the IFEM versus IFVM with a note on the Verification&Validation process through which any software should go before any claim can be made about the quality of its results. The *deal.II* library [69] is widely used across the scientific community and will probably become a cornerstone of numerical analysis in the years to come. Our IFVM approach was developed in the spirit of academic research for alternative immersed boundary methodologies.

In Table 65 is shown a summary of the IFEM and IFVM strengths and weaknesses rom an immersed boundary methodology application point of view.

Table 65 Comparison of IFEM and IFVM

Criteria	IFEM	IFVM
Mesh approach	2 –mesh approach: Eulerian/Lagrangian	1-mesh approach: Eulerian
Domain to compute	fluid and solid	fluid only
Interface scheme	diffuse direct-forcing	sharp direct-forcing
Nature of interface	diffusive: fluid penetration at the IB	Sharp: no fluid penetration at the IB
Numerical accuracy	very accurate (third order in space)	Less accurate (second order in space)
Computational effort	very large	reduced
Numerical algorithm to solve NSEs	fractional step method	Semi-Implicit Pressure-Linked Equation Revised (SIMPLER)
Solution method	iterative	iterative
Projection/interpolation operators	build-in	ad-hoc projection/ interpolation operators
V&V	validated	to validate

6. SCALABILITY ANALYSIS OF THE IFVM APPROACH

In this section we study the behavior of the IFVM approach from a parallel program point of view, that is we will present the performance of the IFVM code when it is parallelized with Message-Passing Interface [130, 131]. We will begin by showing the scalability analysis for the two-dimensional IFVM code with and without immersed body embedded to the Cartesian mesh. Then we will present the scalability analysis for the three-dimensional IFVM code with and without immersed boundary.

In Section 3 we discussed the discretization of the field equations (i.e., NSEs and energy equation) on a Cartesian mesh, and the application of the TDMA in a line-by-line fashion to sweep across the domain and iteratively get to a converged solution. In Figure 118 is shown a schematic of a Cartesian mesh for a two-dimension domain. If the entire domain is solved serially (i.e., on a single processor), the discretized equations are solved first by applying the TDMA to each row of the domain and sweeping in the *y*-direction, and then by applying the TDMA to each column of the domain and sweeping in the *x*-direction. This is the serial version of the IFVM code. Minor modifications to this solution procedure would allow the use of OpenMP [132] for shared memory architectures. This would speedup the performance of the serial IFMV to a theoretical limit equal to the number of processors available for shared memory architectures. The discussion of why there is a very low threshold in the number of processors that can be used in shared memory architectures is beyond the scope of the present research. Thing is that shared memory architectures are limited to tens of processors at the present time.

On the other hand, distributed memory architectures have theoretically no limit on the number of available processors. Supercomputers of small/medium size have thousands of processing units, with the largest supercomputer (IBM Sequoia) on earth having over 1,57 million parallel computing cores.

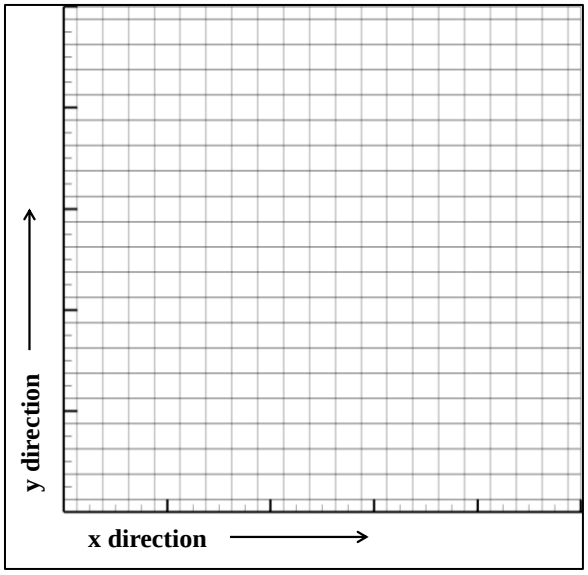

Figure 118. Solution of discretized equations on a two-dimensional mesh using the TDMA algorithm in a line-by-line fashion.

The objective of our present research was to develop a CFD code for the solution of the time-dependent incompressible NSEs coupled with the energy equation for very complex geometries and mesh sizes in the range 100 million – 1 billion cells. For this mesh sizes, trying to solve even steady-state problems on a single processors or on a shared memory architecture would require months/years. Therefore, the natural choice was to develop a parallel version of the IFVM code for distributed memory architectures

(i.e., supercomputers). Without going too much into the details of parallel computing, we can address the two key points in a parallel version of our IFVM code, or any parallel code. The first step is to partition the discretization mesh across the processors. In Figure 119 is shown the two-dimensional domain decomposed across 16 processors. There is a one-to-one correspondence between processes and physical processors (i.e., core units), that is to say each process is associated to a physical processor.

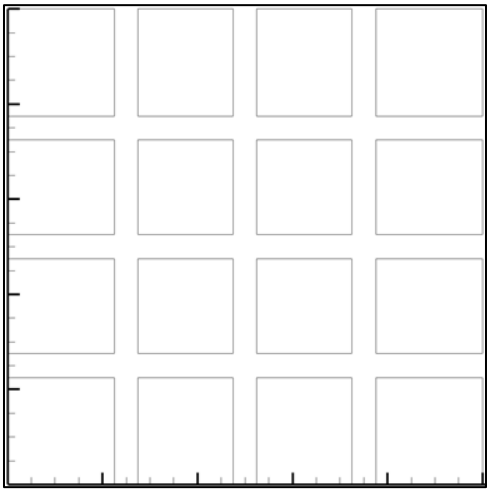

Figure 119. Two-dimensional Cartesian decomposition of a domain with sixteen MPI processes.

Once the domain is decomposed, the discretized equations can be solved for each process independently. In our parallel IFVM approach we used the same TDMA algorithm in a line-by-line fashion to solve the field equations on each core unit just like we did for the serial version of the code on the entire domain.

At every iterations, once the discretized equations are solved for each process, it is necessary to transfer the solution field at the boundary of every partition (i.e.,

interface) to the neighbor processes. This represents the second aspect of running a code on distributed memory system. Every process must have an interface to communicate data with the neighbor processes. The data transfer is handled by the Message-Passing Interface (MPI) library. We only need to concern about when to perform the data transfer. The how is left to the MPI implementation. In Figure 120 is shown a schematic of a two-dimensional mesh decomposed in 16 processes, with two-dimensional communication across the processors, where by two-dimensional communication we mean that the two-dimensional physical domain is decomposed in a two-dimensional fashion with communication at the four boundaries of every process.

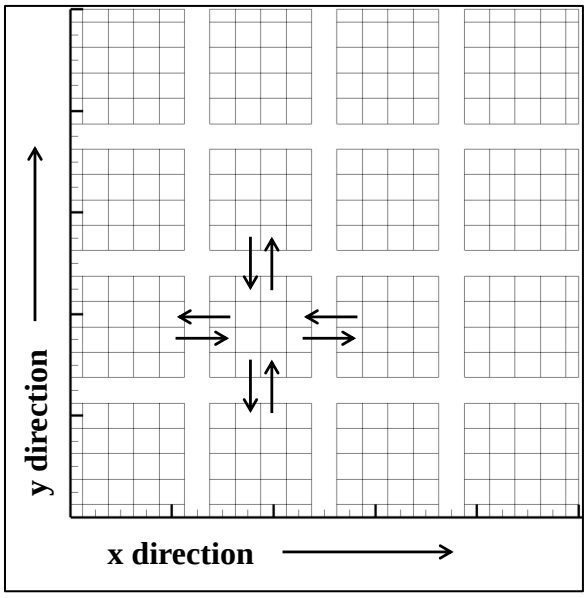

Figure 120. Solution of discretized equations on a two-dimensional mesh using the TDMA algorithm in a line-by-line fashion for sixteen MPI processes with two-dimensional decompositions.

If T_1 is the IFVM execution time on a single processor (sequential) and T_P is the execution time of the parallel IFVM implementation on p processors the speedup S_P is defined as:

$$S_P = \frac{T_1}{T_P} \tag{161}$$

In Figure 121 is shown the speedup for a two-dimensional 512x512 mesh without immersed body function of the number of processors used. The figure shows that for a reduced number of processors (i.e., up to 64) the actual speedup is very close to the theoretical value (red line). For 64 processors we observe super linear speedup due to an efficient use of cache memory. Increasing the number of processors, the fraction of computational effort for each process is reduced and communication across the processes is increased, due to the larger number of processes. Communication is an overhead (i.e., an additional task to perform which in not present in the serial version of the code). Reducing the computational effort on each process and increasing communication across nodes will determine a deterioration of the parallel IFVM performance if too many processes are used. For the mesh size considered (i.e., 512x512), Figure 121 shows that using more than 128 cores will actually slow down the computation (communication becomes the dominant fraction of the computational time). If the mesh size is increased to 1024x1024 cells, the workload of each process is correspondingly increased if compared to the 512x512 mesh case. Therefore, the number of processors for which the parallel IFVM code speedup starts to deteriorate is 256 cores

as shown in Figure 122. For less than 256 cores the parallel version of the IFVM code shows linear/super-linear speedup.

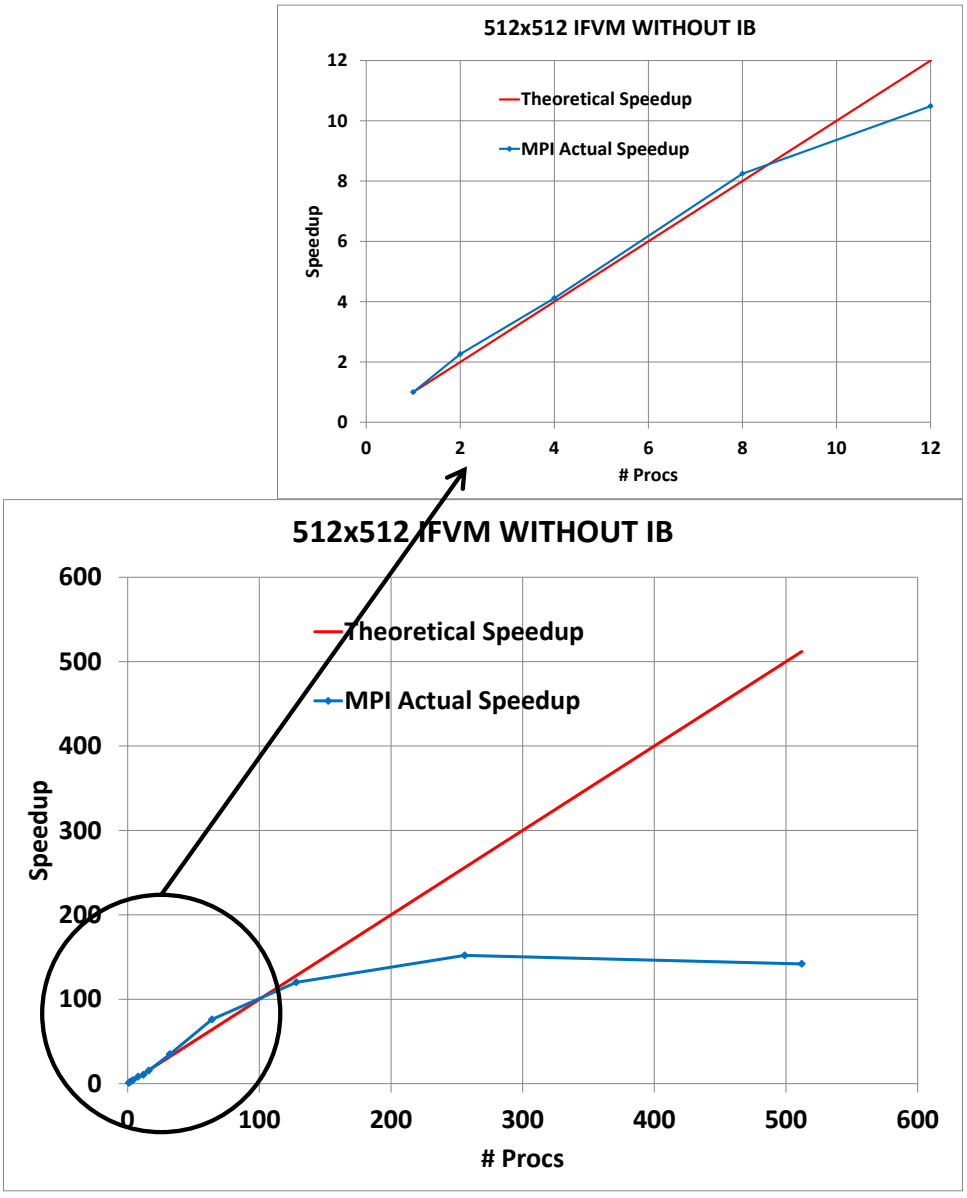

Figure 121. Speedup for a two-dimensional 512x512 mesh without immersed boundary (two-dimensional decomposition).

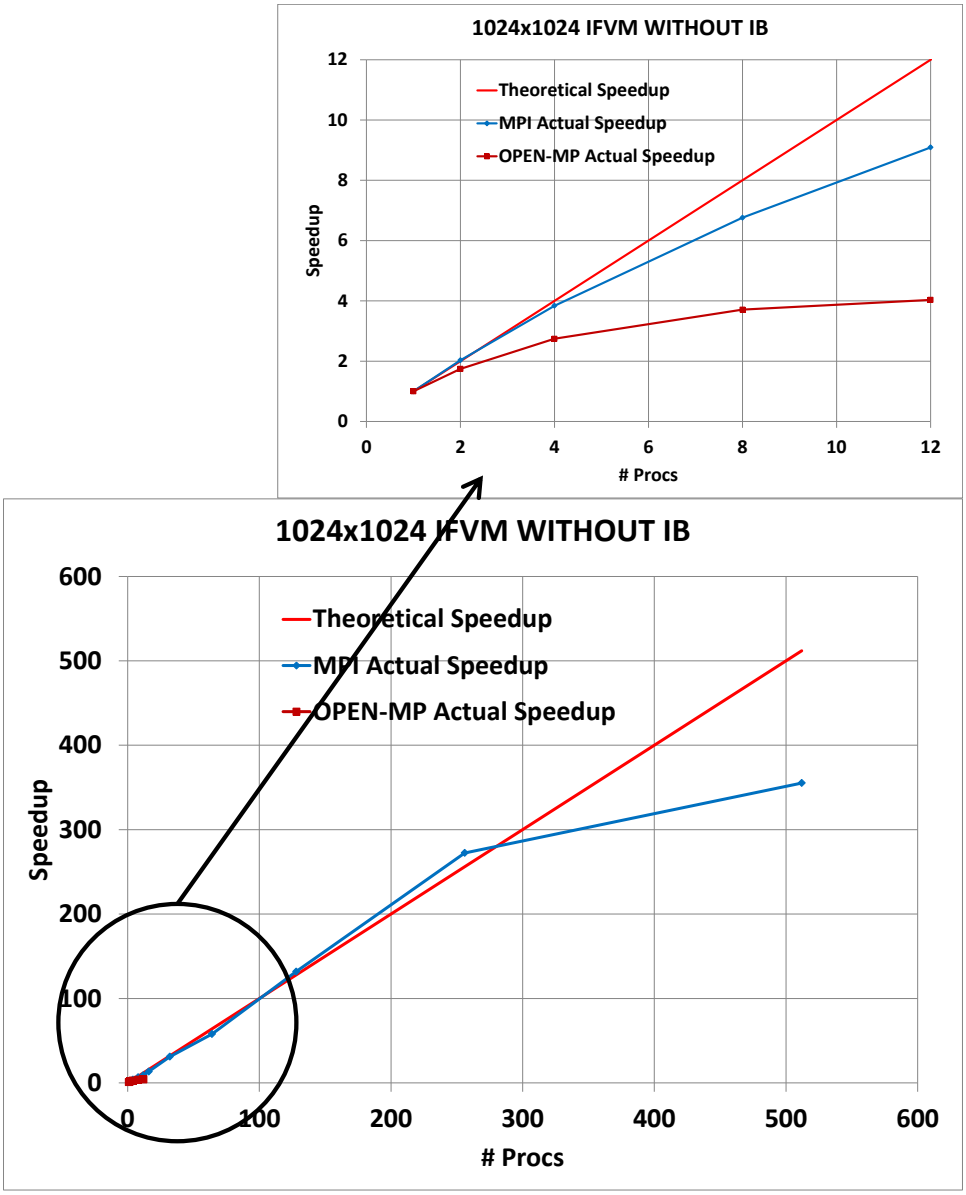

Figure 122. Speedup for a two-dimensional 1024x1024 mesh without immersed boundary (two-dimensional decomposition).

If the number of cells is increased to 2048x2048, the IFVM code almost shows a liner speedup up to 512 processors as pointed out in Figure 123. By increasing the mesh

size to 4096x4096, Figure 124 shows that that the parallel IFVM version shows a linear behavior form parallel performance point of view.

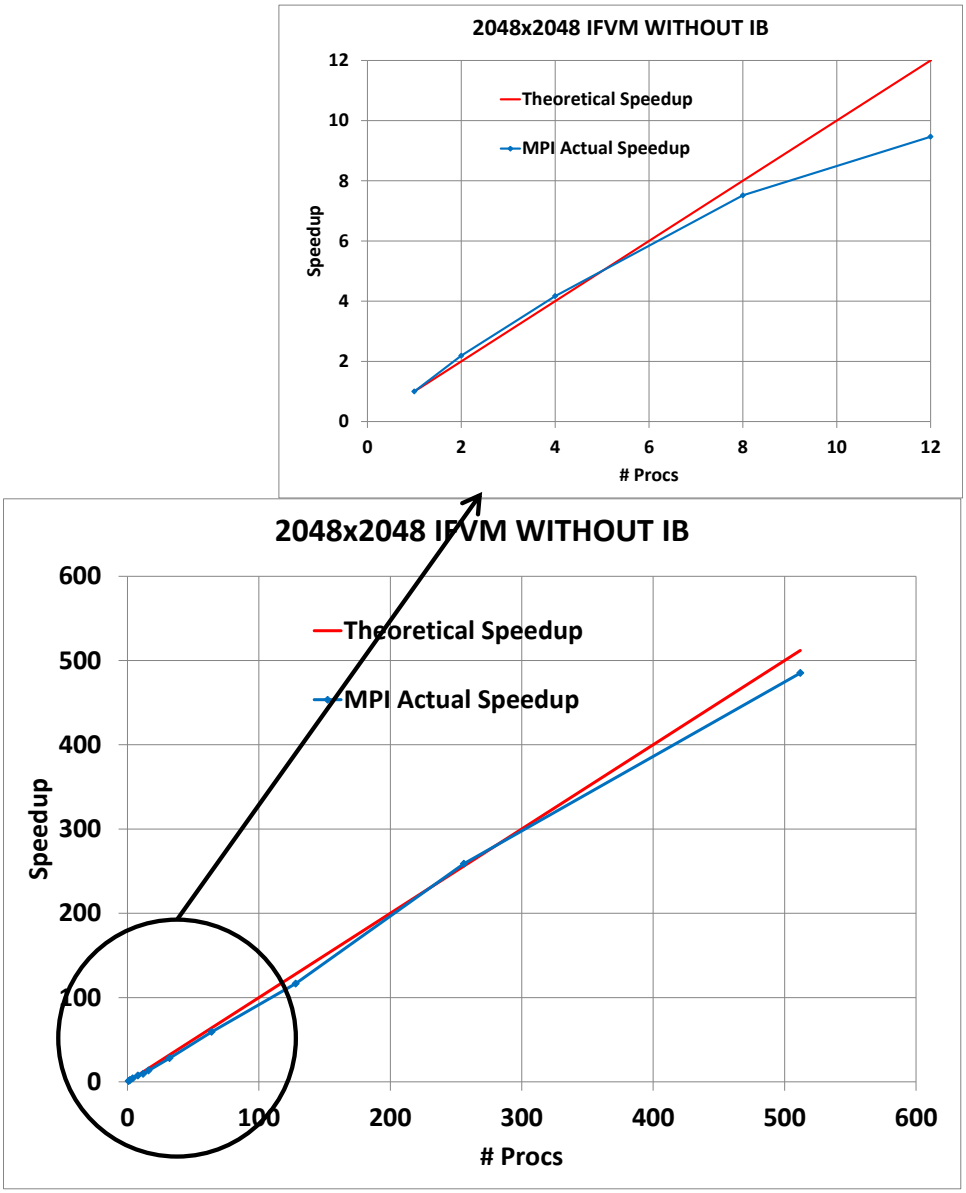

Figure 123. Speedup for a two-dimensional 2048x2048 mesh without immersed boundary (two-dimensional decomposition).

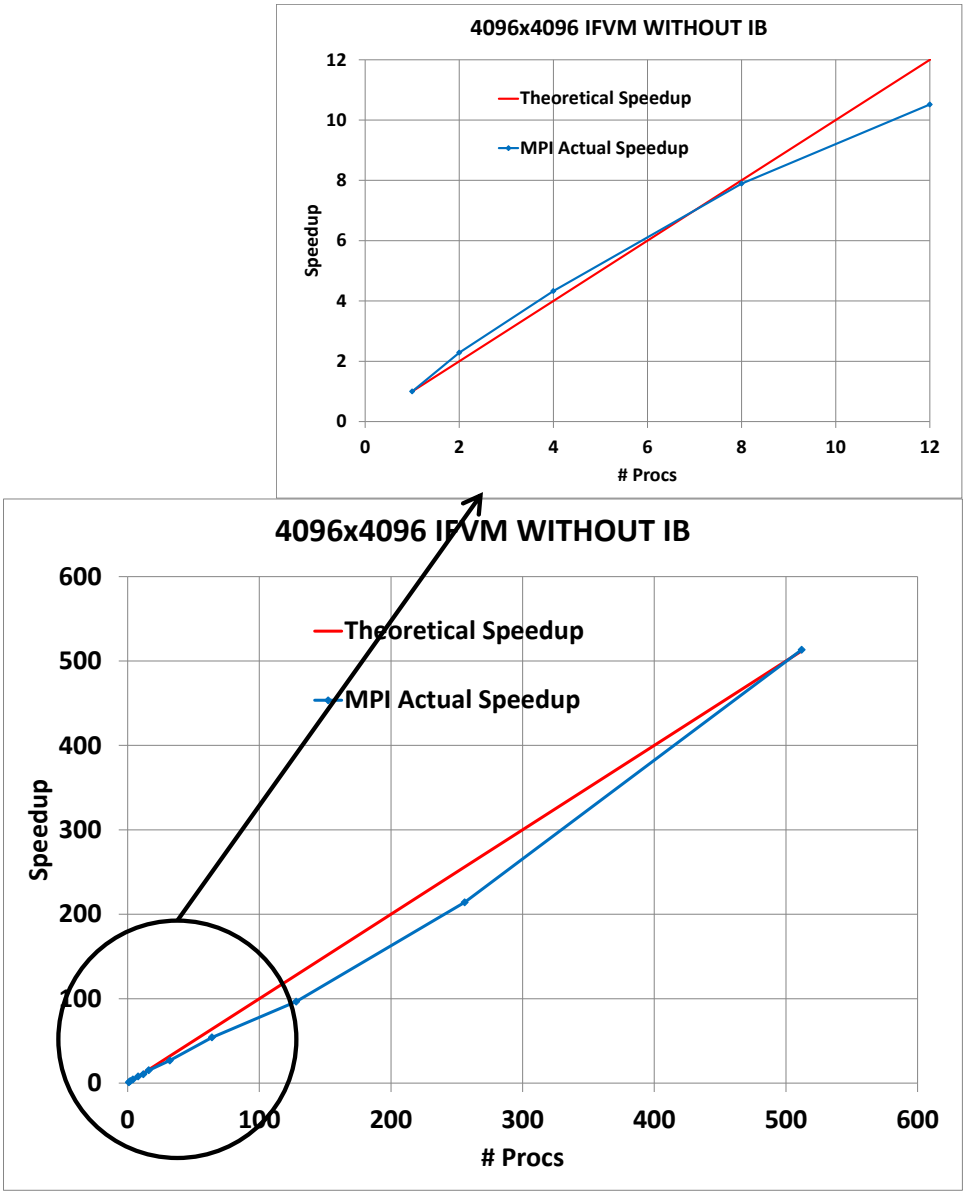

Figure 124. Speedup for a two-dimensional 4096x4096 mesh without immersed boundary (two-dimensional decomposition).

In Figure 125 is shown the computational time on a logarithmic scale function of the mesh size for different number of processors used. If the number of processors is fixed (e.g., one processor is used), increasing the mesh size will proportionally increase

the computational effort and, consequently, the computational time. Figure 125 points out the linear correlation between mesh size and computational time in the range 1-512 #processors. The IFVM shows a linear trend with respect to the number of cells used.

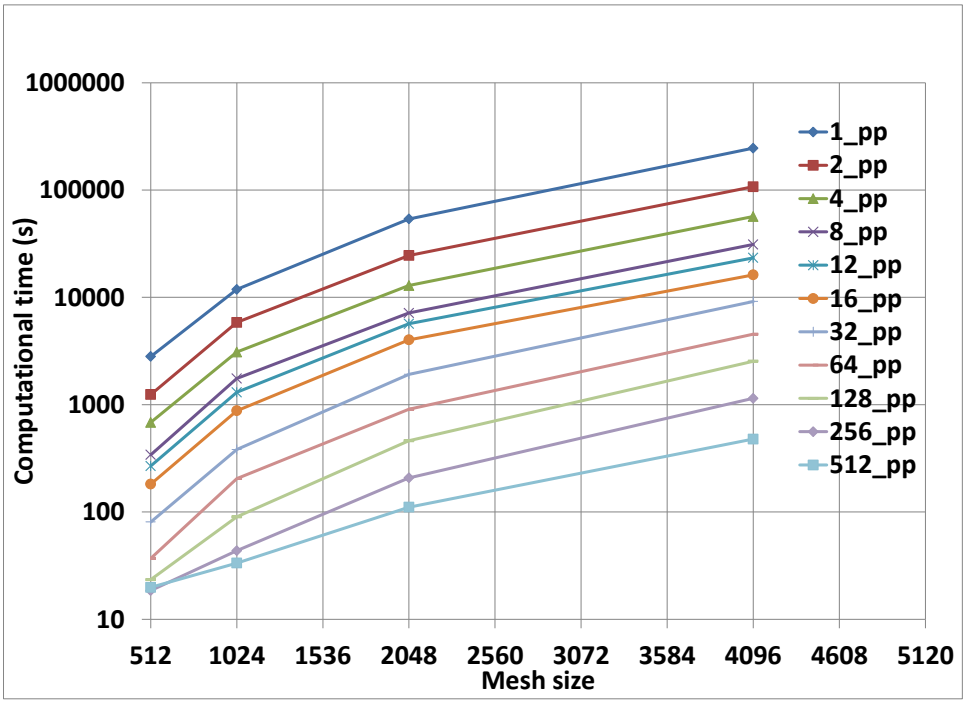

Figure 125. Computational time function of mesh size for a two-dimensional mesh without immersed boundary.

Next we look at the parallel version of the IFVM when an immersed body is embedded with the Cartesian mesh for a two-dimensional problem. In Section 3 we discussed how the effect of the solid body on the underlying fluid mesh is limited to the identification of the fluid and solid regions, the determination of the ghost-cells location at the fluid/solid interface, and the interpolation procedure at these ghost-cells. These steps can be addressed as the creation of a fluid/solid computational “interface”, that is

to say a way to generate correctly interpolated forcing terms at the ghost-cells of the fluid Cartesian mesh. In the serial version of the IFVM code, once we have created the computational interface, we solve the discretized equations by sweeping in the *x* and *y*-direction, respectively. The immersed boundary does not introduce any alteration of the base solver algorithm. This can be easily understood by looking at Figure 126, where we apply the TDMA in a line-by-line fashion to the two-dimensional Cartesian mesh with immersed boundary. The effect of the immersed boundary is an additional forcing term in the discretized equations at the ghost-cells.

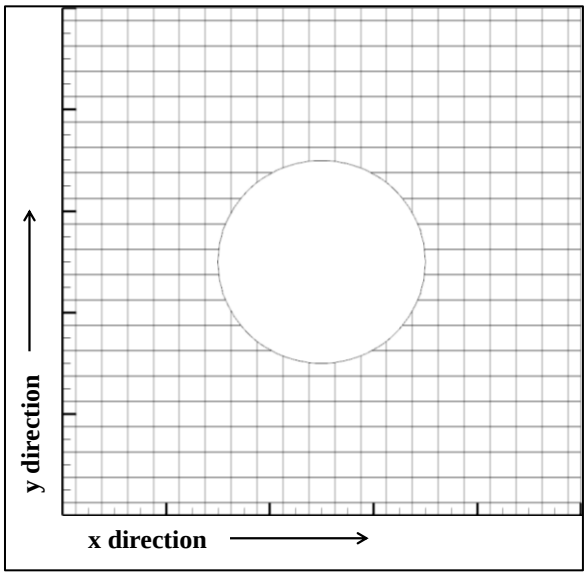

Figure 126. Solution of discretized equations on a two-dimensional mesh with immersed body using the TDMA in a line-by-line fashion.

When the computational domain is decomposed across many processors, which will communicate through the interface, each process will create the computational

“interface” for the immersed boundary following the same criteria used for the serial version of the IFVM code. Therefore, in the parallel IFVM approach, we split the immersed boundary interface across processors in the same way we decompose the Cartesian mesh. Processes which will happen to compute the flow equations in physical regions crossed by the immersed boundary will have the corresponding cells flagged as solid, and ghost-cells to apply the forcing terms> On the other hand, processes outside the immersed body will solve the discretized equations on a simple two-dimensional Cartesian mesh where all cells belong to the fluid region. This can be easily visualized with the sketch shown in Figure 127, where a two-dimensional domain with immersed body is distributed across 16 processes. The figure also points out that the processes crossed by the immersed body solve the discretized equations on a reduced number of computational cells if compared to the processors outside the solid body. This might generate work unbalance when many MPI processes are requested to perform computations. The computational overhead required to perform the interpolation for the forcing term at the ghost cells balances with the reduced number of cells for the processes overlapping with the solid body (i.e., reduced computational effort). The presence of solid objects in the parallel version of the IFVM code does not represent any significant load balance concern as the results from the two/three-dimensional version of the parallel IFVM code point out.

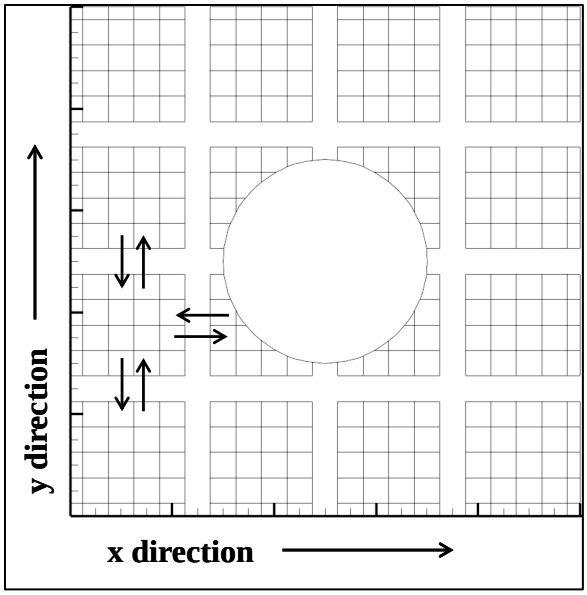

Figure 127. Solution of discretized equations on a two-dimensional mesh with immersed body using the TDMA in a line-by-line fashion for sixteen MPI processes with a two-dimensional decomposition.

In Figure 128 is shown the speedup for a two-dimensional 512x512 mesh with immersed boundary function of the number of processors used. The results are consistent with the IFVM code without immersed boundary (see Figure 121). For this relatively coarse mesh, above 64 processors the overhead due to communication represents most of the computational time and there is no gain in increasing the number of processors. Also for a two-dimensional 1024x1024 mesh with immersed boundary shown in Figure 129 the results are consistent with the case without immersed body (see Figure 122). The parallel IFVM code shows almost linear scaling up to 256 processors.

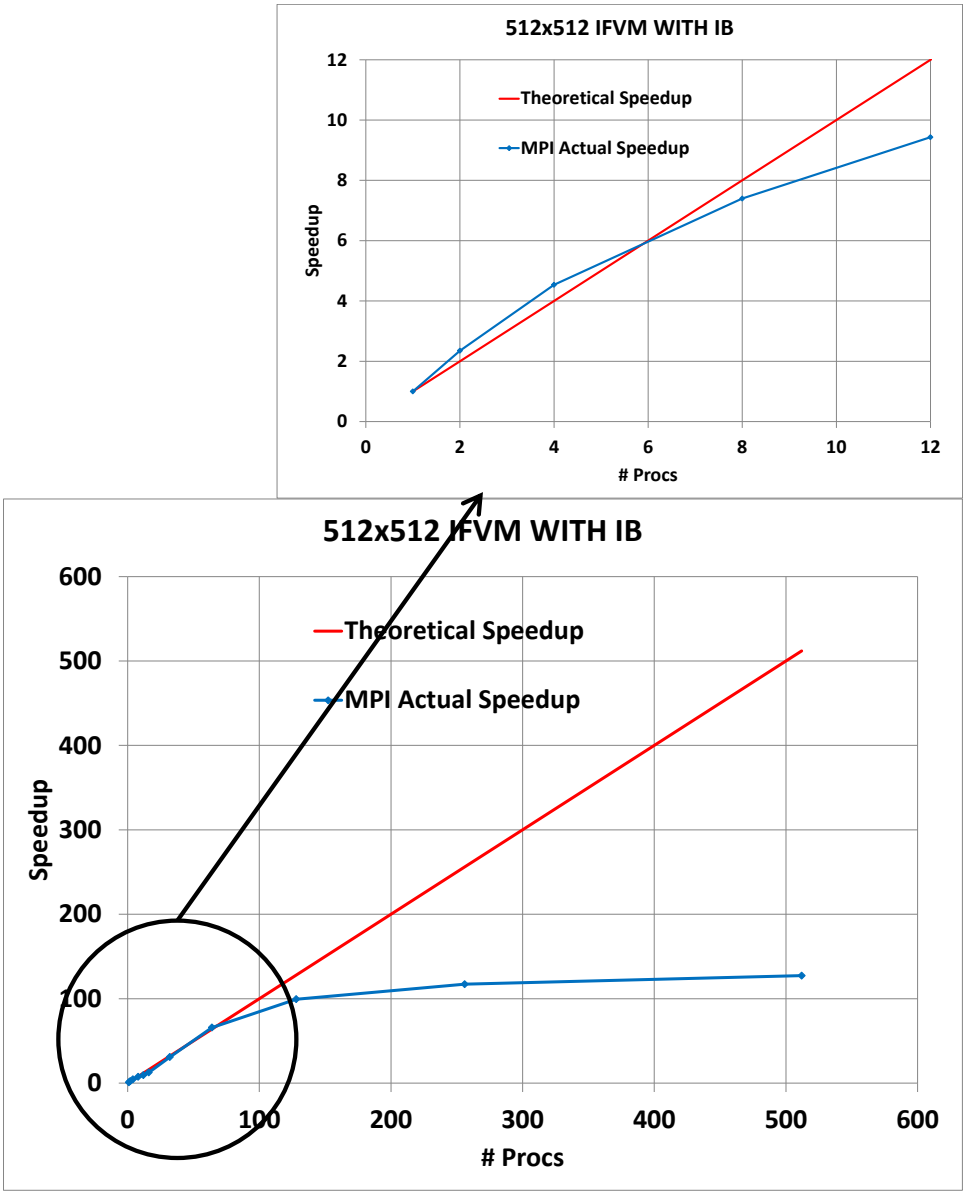

Figure 128. Speedup for a two-dimensional 512x512 mesh with immersed boundary (two-dimensional decomposition).

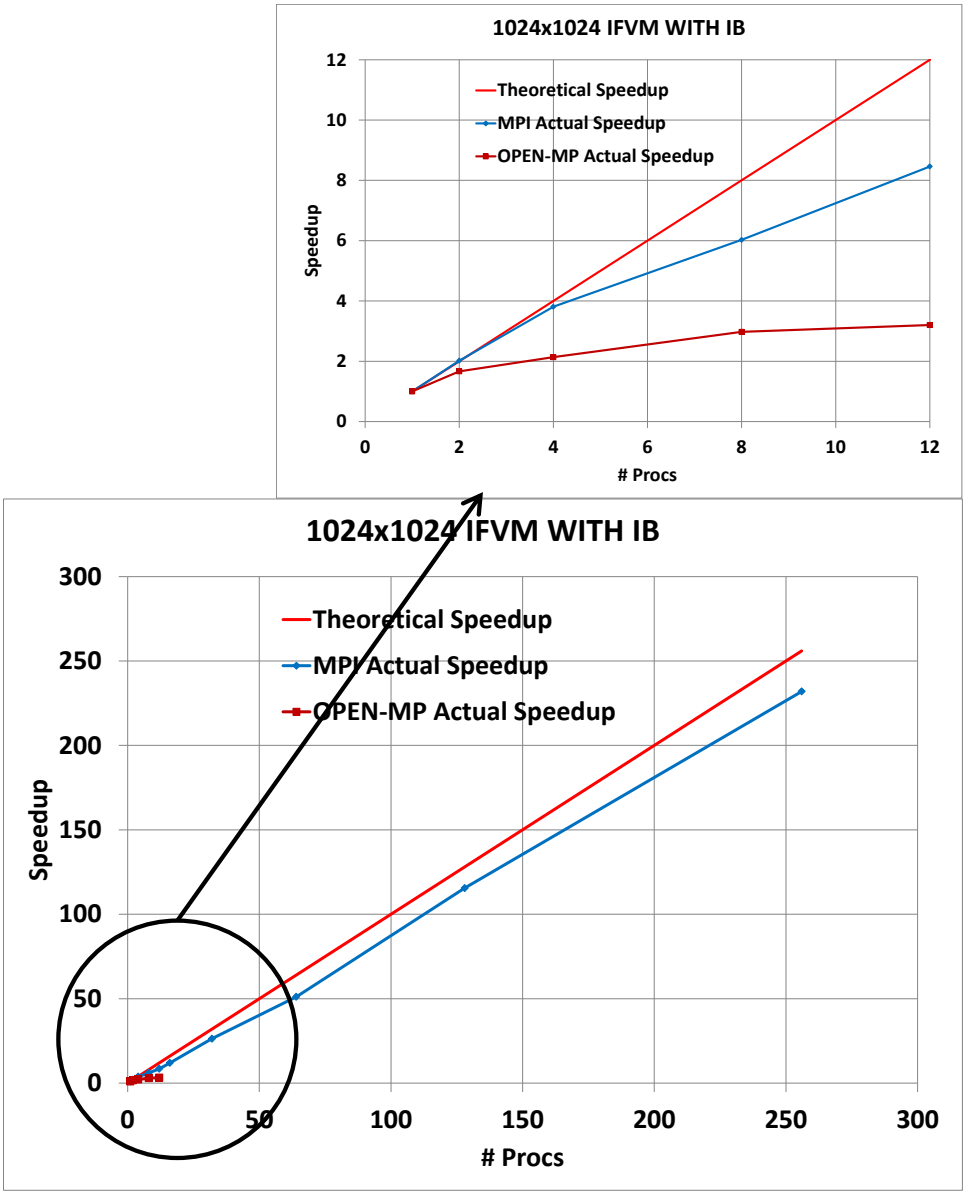

Figure 129. Speedup for a two-dimensional 1024x1024 mesh with immersed boundary (two-dimensional decomposition).

Considering a two-dimensional 2048x2048 mesh size with immersed boundary, the IFVM shows linear scaling up to 512 processors, in agreement with the same mesh size case without the solid body (see Figure 123).

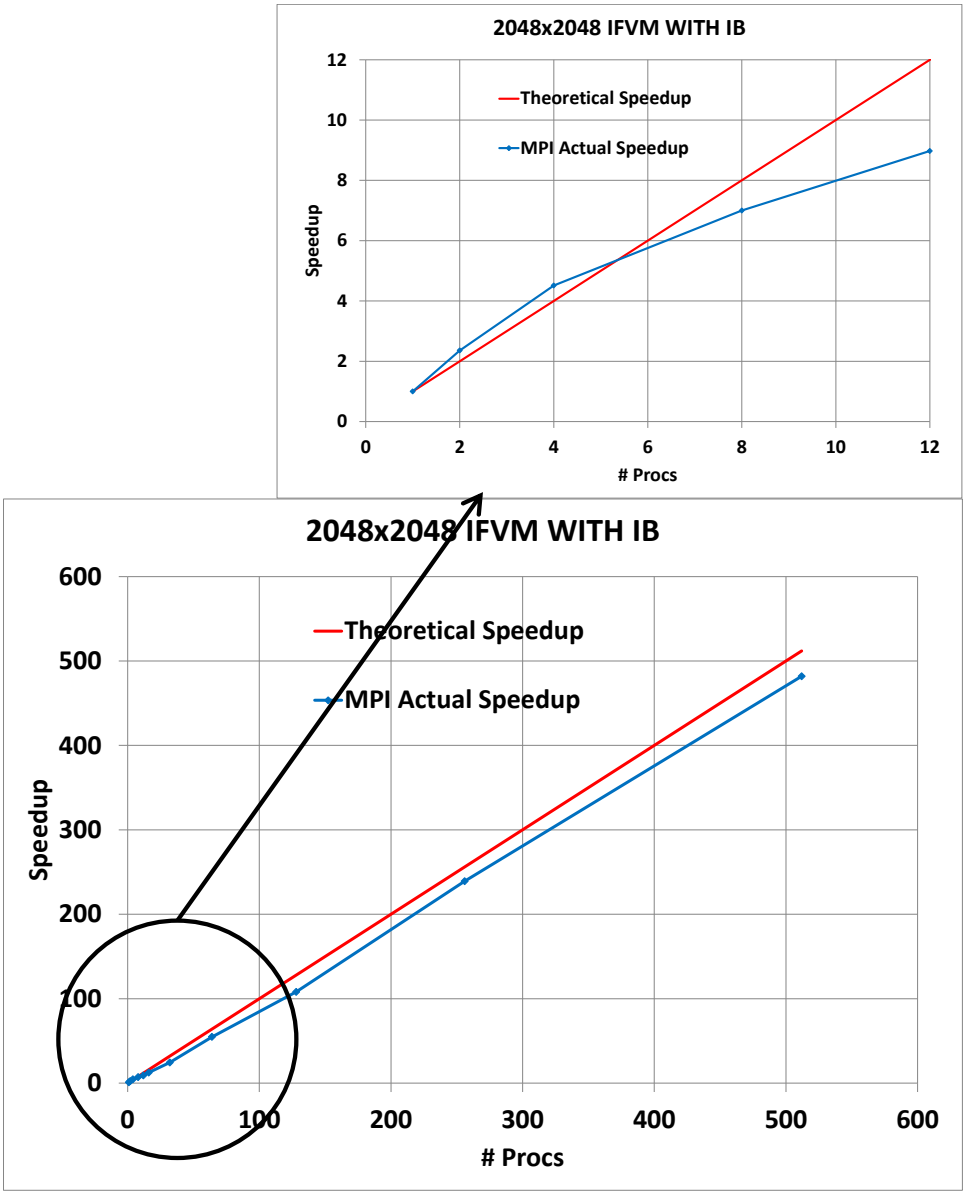

Figure 130. Speedup for a two-dimensional 2048x2048 mesh with immersed boundary (two-dimensional decomposition).

Figure 131 shows the speedup for the two-dimensional 4096x4096 mesh case with immersed boundary. When we compare the speedup for this case to the one for the same mesh size without immersed boundary (shown in Figure 124), the effect of the

solid body on the overall computational time can be noticed. The presence of a bluff body inside the computational domain will prevent the effective use of all scheduled processors. Some of them might be completely inside the solid body for which the IFVM approach does not compute the field equations. These processors actually do not perform any computations. This is a limitation of the IFVM approach implemented and would require a major data structure change to eliminate the problem. Two aspects mitigate this limitation of the IFVM. In general complex geometries consider either immersed body which span most of the computational domain or are localized in a limited region of a very large computational domain (e.g., two-dimensional flow over a circular cylinder and three-dimensional flow over a sphere). For problems where the immersed body covers most of the computational domain, many processes will intersect the solid body and, therefore, the work load is balanced. On the other hand, for very large computational domains a limited number of processors will intersect the immersed boundary and the overall effect on performances is very limited. We presented the results for the worst case scenario possible, which is the one of an immersed body size comparable to that of the embedded fluid region. For this case, when a large number of processors are used, it is inevitable to have a reduction in performance due to the inefficient use of the processors overlapping the bluff solid object. Figure 131 shows that the reduction in performance due to the presence of the immersed body is still in the order of 20% reduction if compared to the case without embedded body.

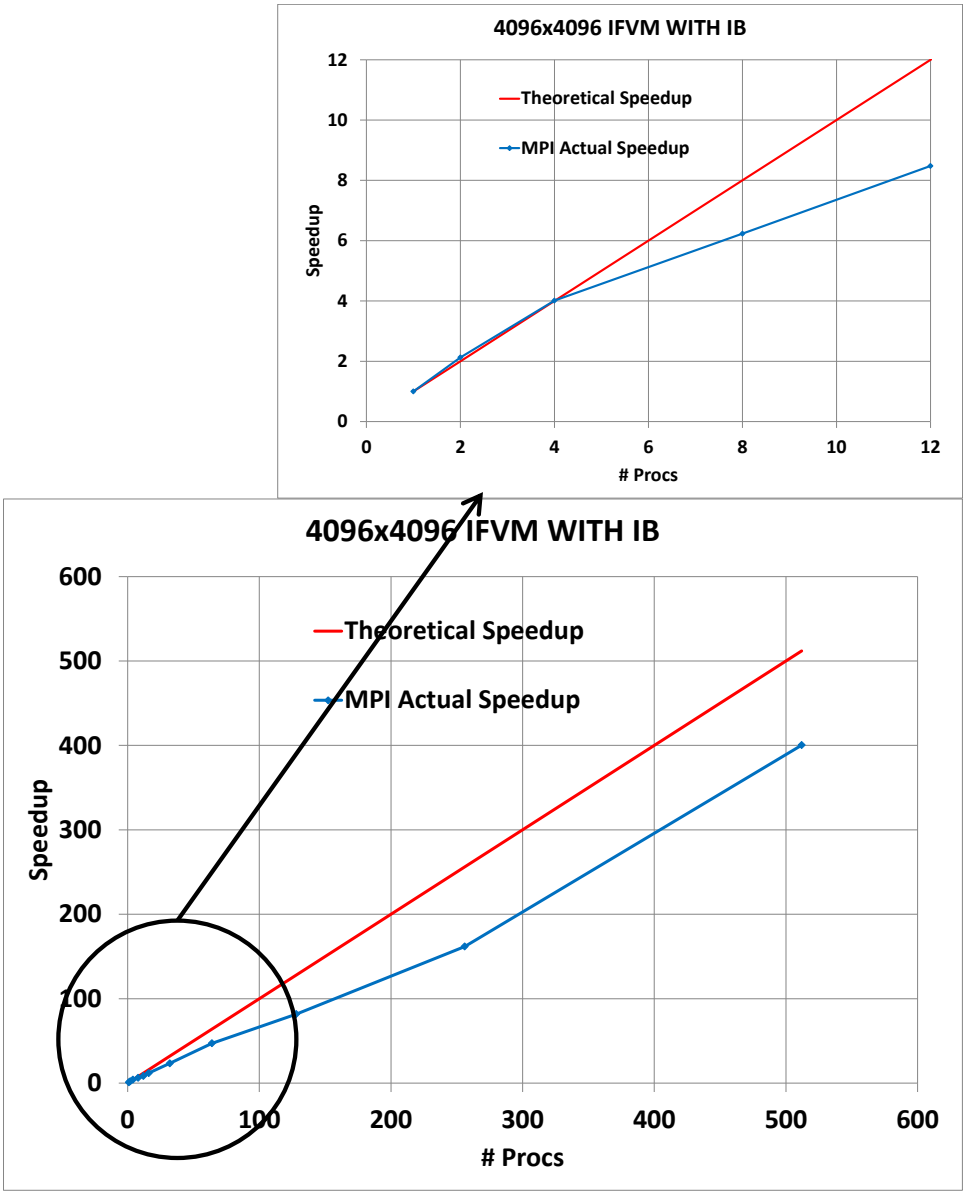

Figure 131. Speedup for a two-dimensional 4096x4096 mesh with immersed boundary (two-dimensional decomposition).

The presence of the immersed body does not alter the linear relation between the computational time and the mesh size for the IFVM, since the overhead associated with the interpolation procedure at the fluid/solid interface is proportional to the fluid mesh

size. Therefore, the computational effort is proportional to the mesh size for any number of processes used as shown in Figure 132.

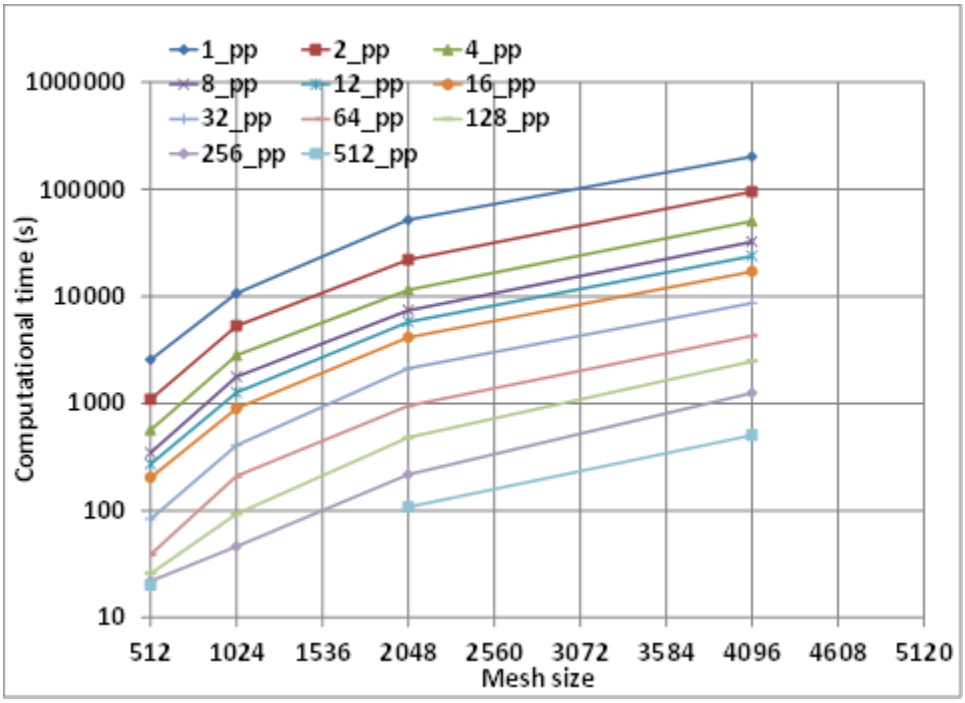

Figure 132. Computational time function of mesh size for a two-dimensional mesh with immersed boundary.

When three-dimensional problems are considered we apply the TDMA in a line-by-line fashion sweeping alternatively in the *x*, *y* and *z*-direction, respectively. The base solver is the same as for two-dimensional problems, and besides sweeping in the third dimension, the approach to the solution of the field equation remains unchanged for two/three-dimensional problems as the schematic shown in Figure 133 points out.

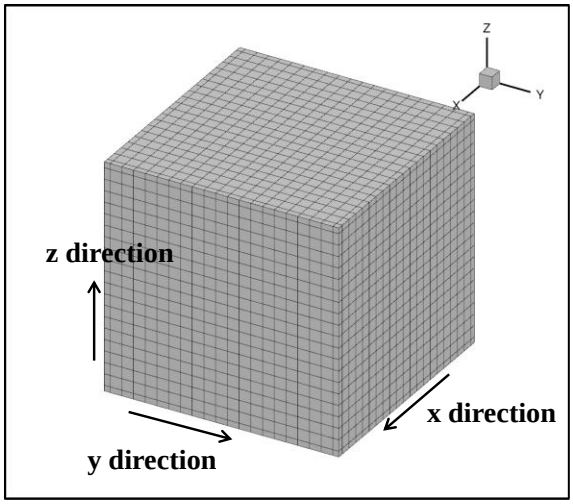

Figure 133. Solution of discretized equations on a three-dimensional mesh using the TDMA algorithm in a line-by-line fashion.

In Figure 134 is shown the two-dimensional decomposition of a reference three-dimensional domain across 16 MPI processes. This implies that for physical three-dimensional problems we split the computational mesh in such a way to have communication at the interfaces in the *x* and *y*-direction, with a continuous mesh in the *z*-direction as shown in Figure 135. We implemented both a two and three-dimensional decomposition of the physical domain, but for the time being we tested the two-dimensional decomposition only. The reason behind this choice is to keep communication overhead to a minimum even for very large meshes. The profiling of the IFVM code estimates a communication time inferior to 20% for the largest mesh considered (500x500x500 cells). This choice was made in the spirit of testing the performance of the implemented three-dimensional IFVM more from a computational effort point of view, trying to limit the overhead due to communication across processes.

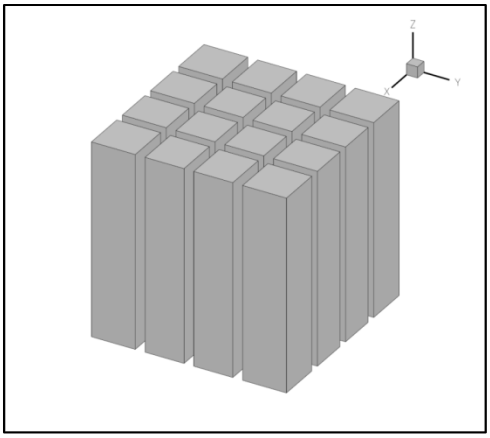

Figure 134. Two-dimensional decomposition of a three-dimensional Cartesian domain with sixteen MPI processes.

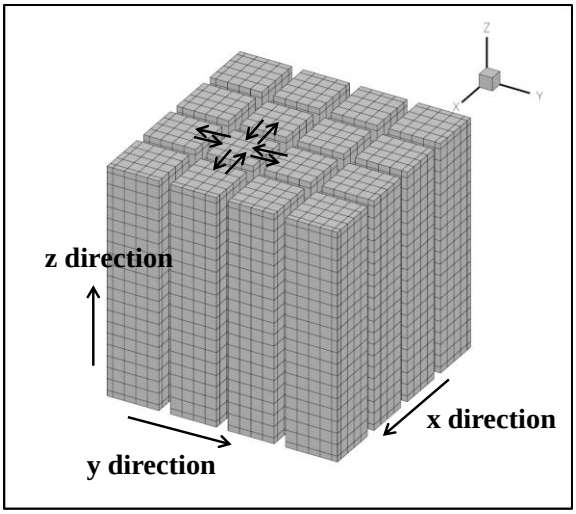

Figure 135. Solution of discretized equations on a three-dimensional mesh using the TDMA algorithm in a line-by-line fashion for a sixteen MPI processes with a two-dimensional decomposition.

In Figure 136 is shown the speedup for a three-dimensional 100x100x100 mesh without immersed boundary with two-dimensional decomposition. For a reduced number of processors, linear scaling performances are shown. If the number of

processors in increased, communication becomes the relevant part of the computational time for this relatively coarse mesh, and there is a limited gain in total computational time by increasing the number of processors above 32.

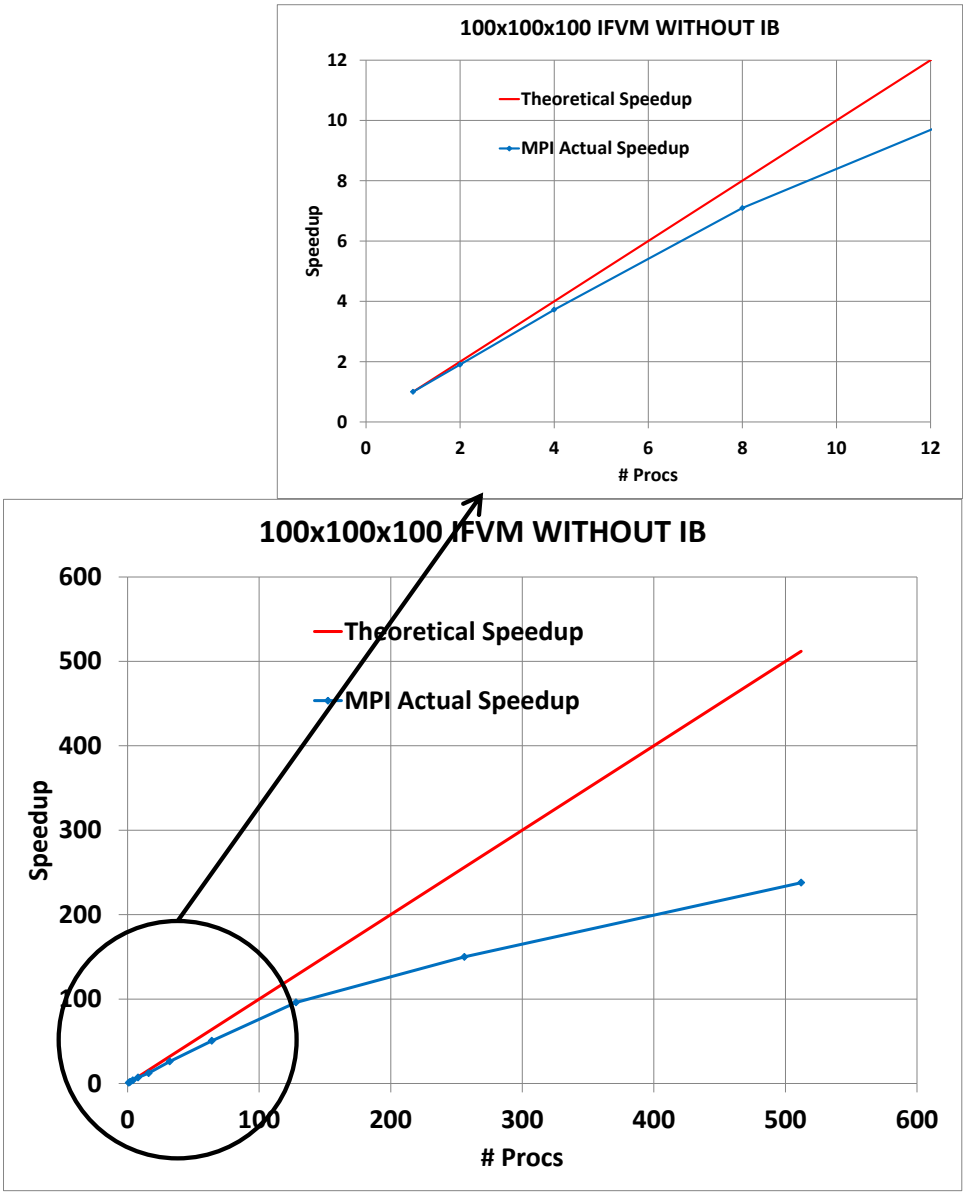

Figure 136. Speedup for a three-dimensional 100x100x100 mesh without immersed boundary (two-dimensional decomposition).

Considering a three-dimensional 200x200x200 mesh without immersed boundary with a two-dimensional decomposition, the IFVM shows linear scaling up to 64 processors, that is approximately 125,000 cells per processor. For a larger number of processes, the overhead due to communication becomes the dominant factor in the total computing time as shown in Figure 137.

Figure 138 shows the speedup for a three-dimensional 300x300x300 mesh without immersed boundary and with two-dimensional decomposition. The parallel IFVM code shows linear speedup up to 256 processors, and almost linear speedup for 512 processors.

When the number of cells is increased to 400x400x400, linear speedup is achieved up to 512 processors as shown in Figure 139. For these very large meshes, the computational time represents the relevant fraction of total computing time, with communication time being a few percent of the total computing time.

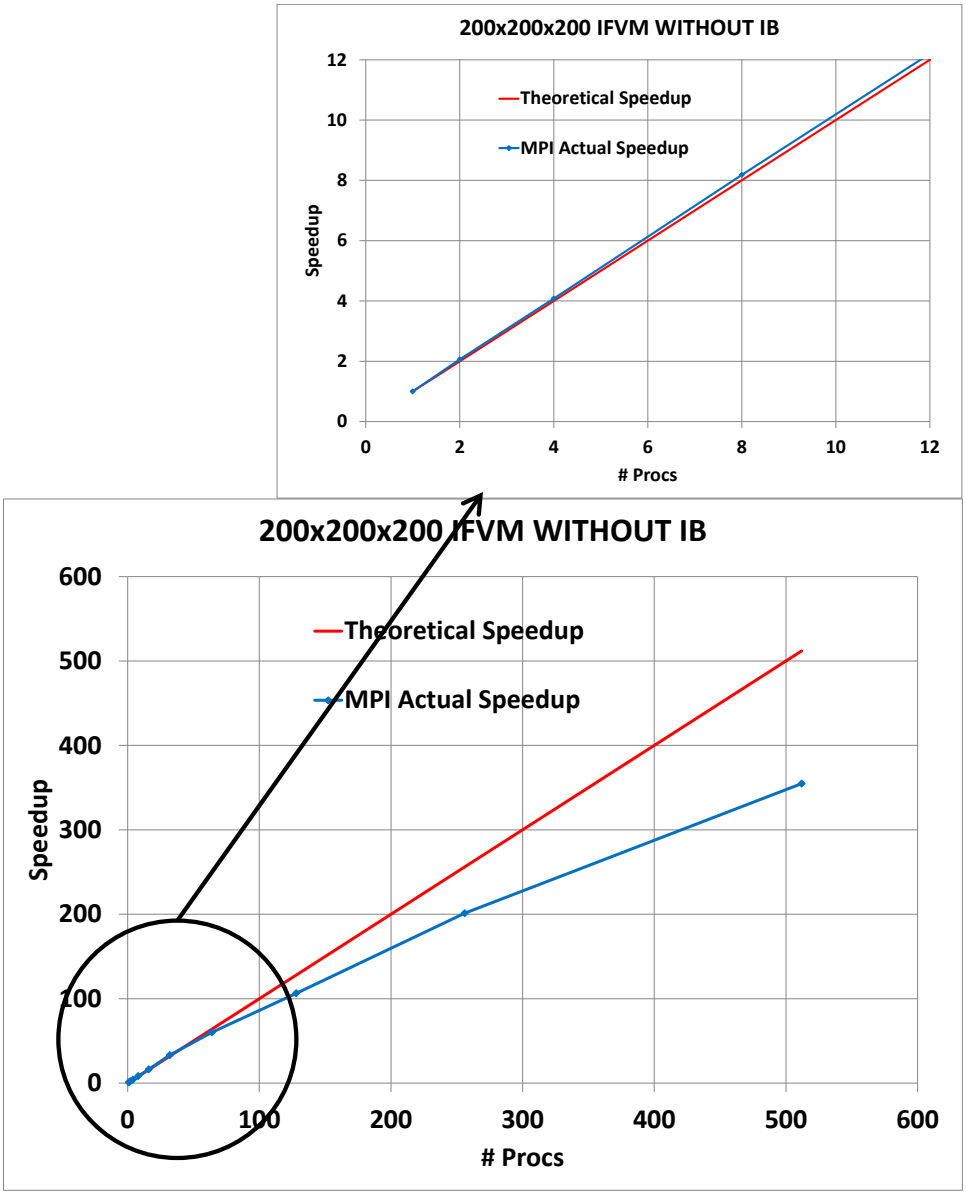

Figure 137. Speedup for a three-dimensional 200x200x200 mesh without immersed boundary (two-dimensional decomposition).

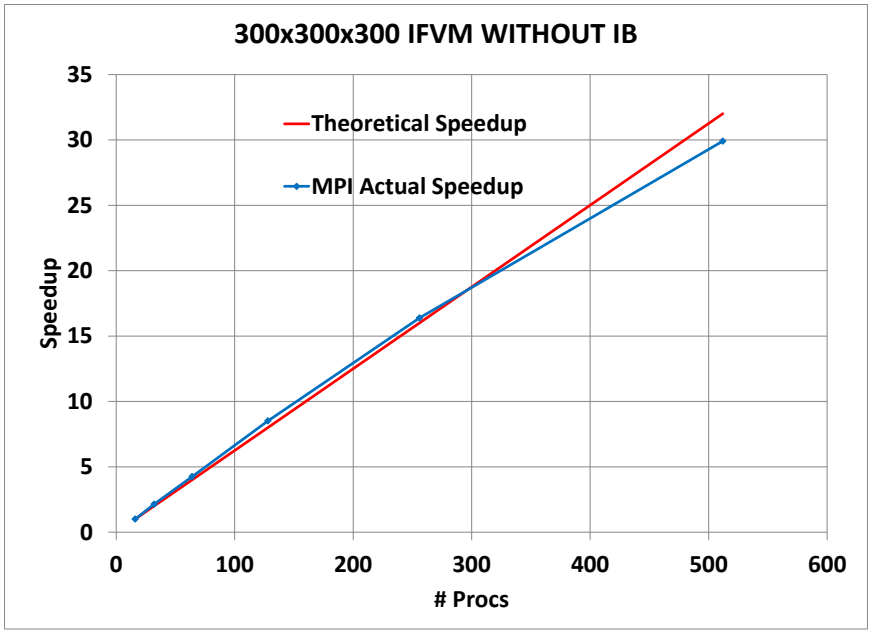

Figure 138. Speedup for a three-dimensional 300x300x300 mesh without immersed boundary (two-dimensional decomposition).

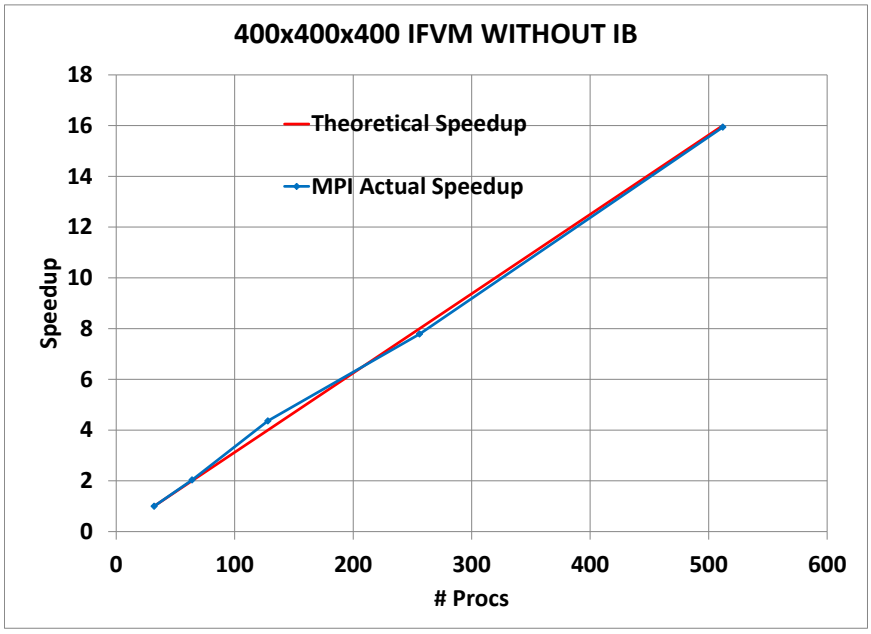

Figure 139. Speedup for a three-dimensional 400x400x400 mesh without immersed boundary (two-dimensional decomposition).

The results obtained for the three-dimensional 500x500x500 mesh without immersed boundary shown in Figure 140 are in contradiction with what expected. Increasing the mesh size, the parallel IFVM code speedup should be closer to the theoretical one. Figure 140 shows that for 512 cores the IFVM performances start to deteriorate, when a coarser mesh shows theoretical speedup for this number of processors (see Figure 139). We did not investigate further this result for the time being.

In Figure 141 we show the computational time function of the mesh size for three-dimensional problems without the immersed boundary. The figure shows that the computational time is linearly related to the mesh size for the number of cores considered to solve the problem.

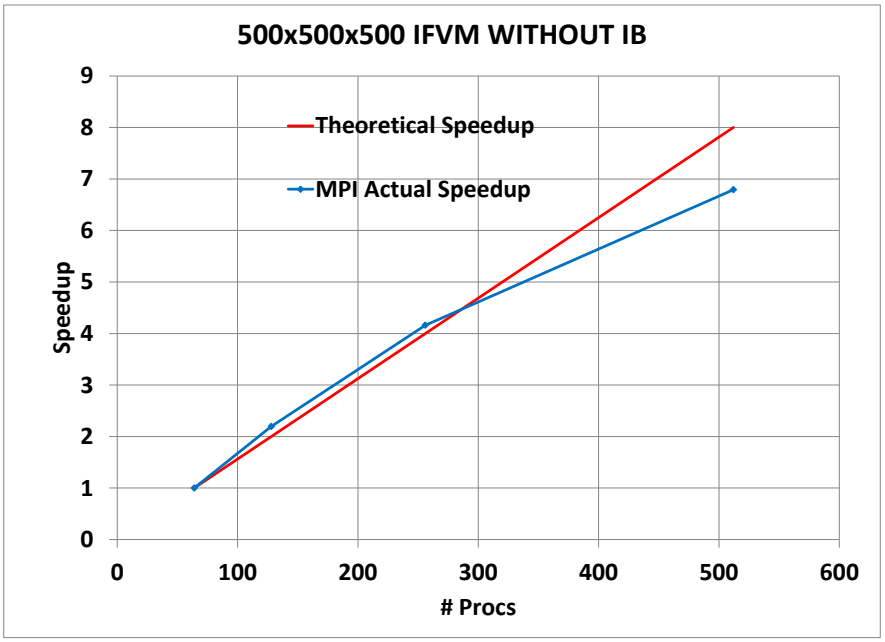

Figure 140. Speedup for a three-dimensional 500x500x500 mesh without immersed boundary (two-dimensional decomposition).

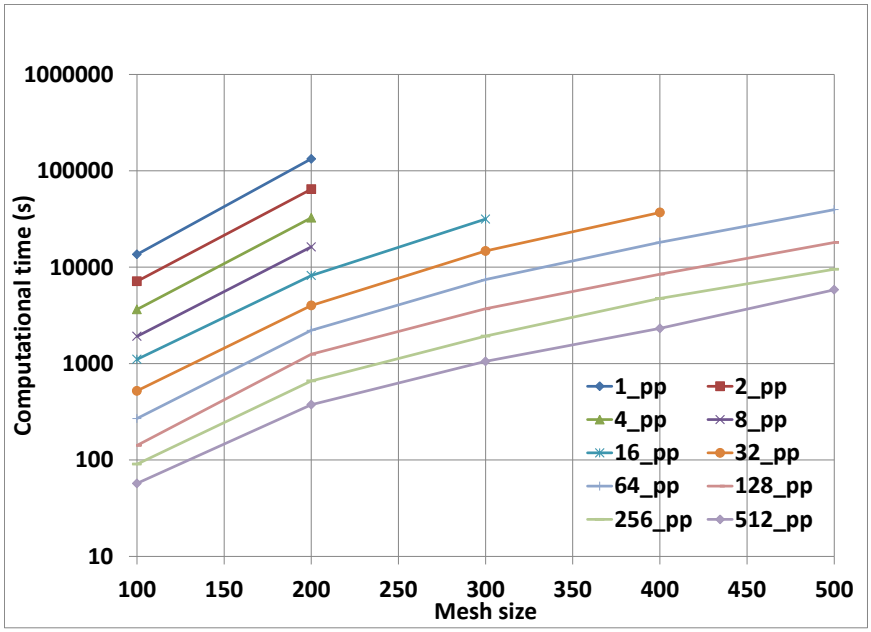

Figure 141. Computational time function of mesh size for a three-dimensional mesh without immersed boundary.

When an immersed body is considered in a three-dimensional domain with a two-dimensional MPI decomposition, we follow the same approach described for two-dimensional problems with two-dimensional decomposition, which is each process intersecting the solid body constructs the computational interface that competes to it. This is to say that the immersed body interface is split across the processes in the same way the fluid mesh decomposition is achieved. The solution algorithm for a serial IFVM code solves the TDMA in a line-by-line fashion alternating swept in the three directions, and using the computational interface to determine the forcing terms at the ghost-cells as shown in Figure 142. The parallel IFVM code with embedded immersed body solves the discretized equations following the same approach the only difference being fluid mesh

and immersed body interface are split across processes and two-dimensional communication at the processes interface is required as shown in Figure 143.

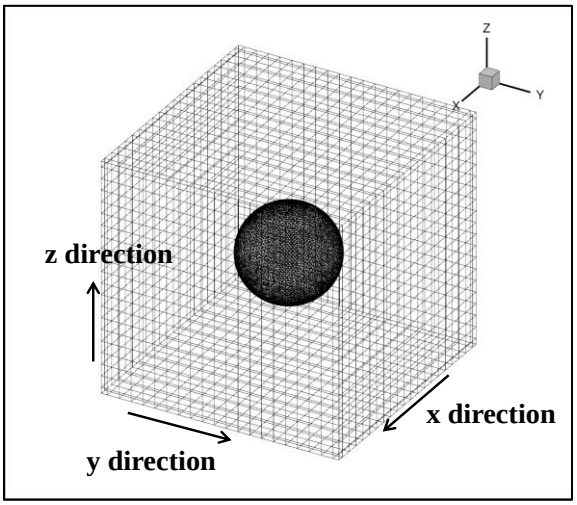

Figure 142. Solution of discretized equations on a three-dimensional mesh with embedded body using the TDMA algorithm in a line-by-line fashion.

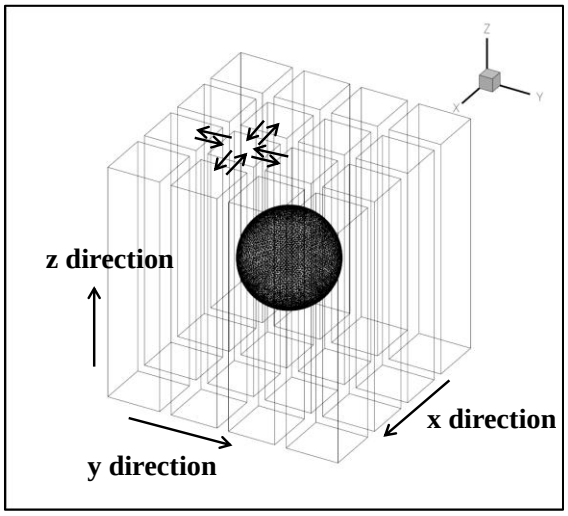

Figure 143. Solution of discretized equations on a three-dimensional mesh with embedded body using the TDMA algorithm in a line-by-line fashion for sixteen MPI processes with two-dimensional decomposition.

When the immersed body is included in the computational domain, the main issue related to parallel performance of the IFVM is the possibility that a large number of processes are located inside the solid body, which implies the corresponding cores are partially or fully unutilized. As we discussed for the two-dimensional configuration, the only way to completely avoid this problem would require a major change in the IFVM data structure organization. We tested the parallel IFVM three-dimensional configuration against a bluff body (i.e., a sphere) of diameter $D = 0.4\,L$, where L is the cubic enclosure reference length. As we pointed out before, this represents the worse scenario with a large region of the computational domain occupied by the solid object.

In Figure 144 is shown the speedup for a three-dimensional 100x100x100 mesh with immersed boundary and two-dimensional decomposition. The figure shows that for more than four processors the parallel IFVM deviates from the theoretical speedup in agreement with the same case without immersed body shown in Figure 136. For a 200x200x200 mesh size the parallel IFVM code with embedded body shows in Figure 145 a linear speedup up to 32 processors, meanwhile the parallel IFVM code without immersed body linear speedup is maintained up to 64 processors (see Figure 137). This slight reduction in performance is clearly due to the presence of the immersed body.

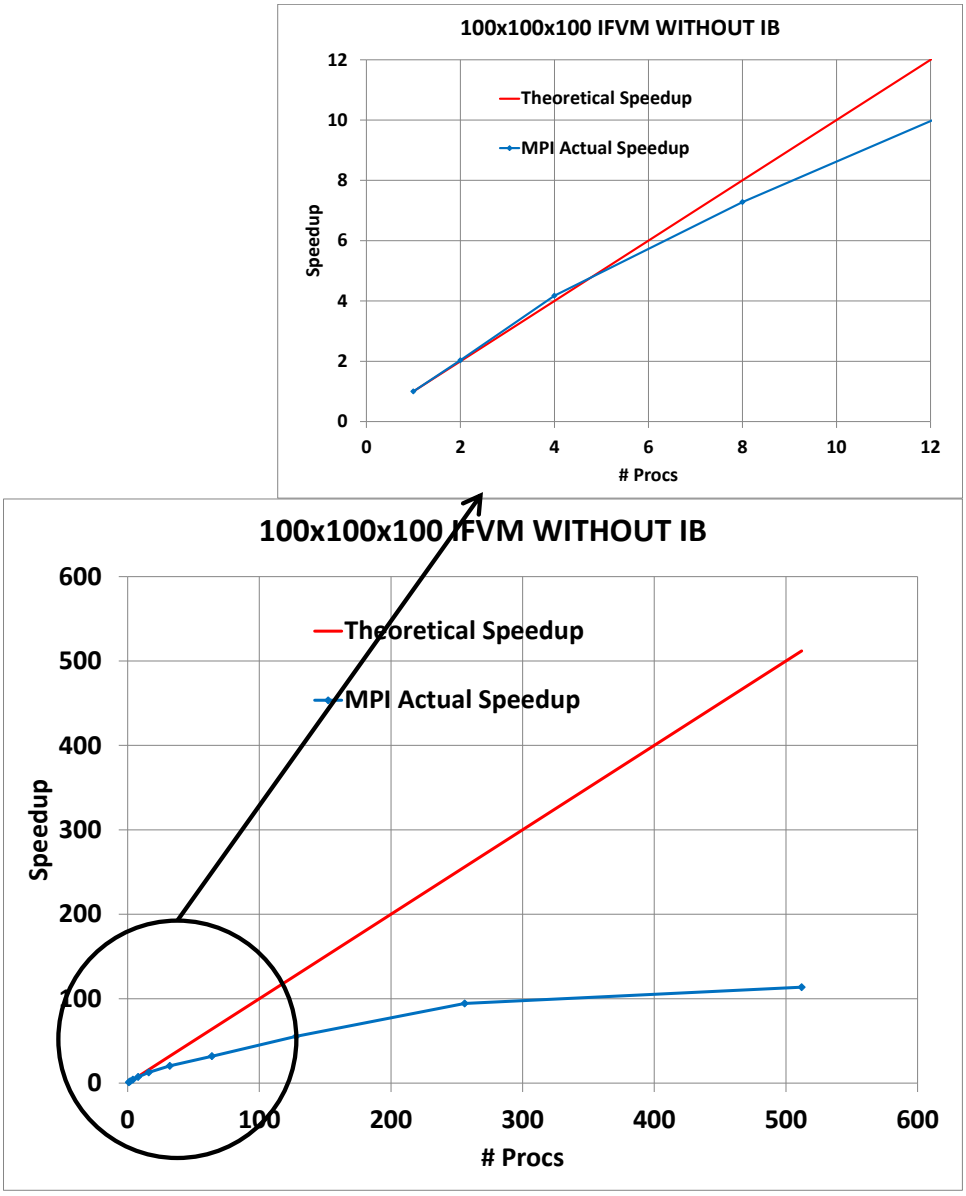

Figure 144. Speedup for a three-dimensional 100x100x100 mesh with immersed boundary (two-dimensional decomposition).

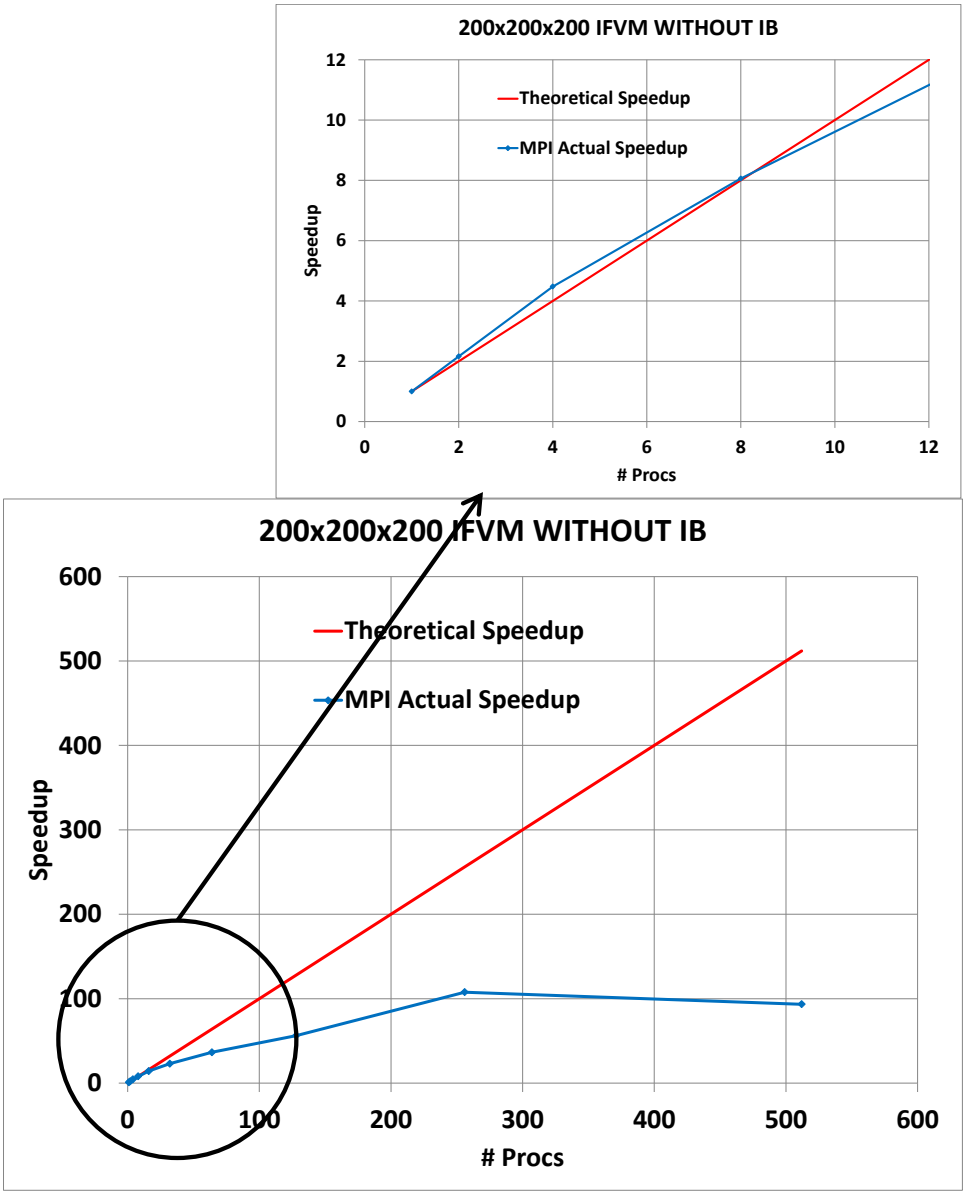

Figure 145. Speedup for a three-dimensional 200x200x200 mesh with immersed boundary (two-dimensional decomposition).

If the mesh is refined to 300x300x300 cells, 400x400x400 cells and 500x500x500 cells, the parallel IFVM code linear speedup trend is maintained for larger and larger number of processors, as shown in Figure 146, Figure 147 and Figure 148

respectively. The presence of the immersed body inevitably introduces a reduction in the parallel performance of the IFVM code, but even for the conservative configuration chosen, this reduction can be considered acceptable if compared to the speedup for the three-dimensional cases without immersed boundary (see Figure 138, Figure 139 and Figure 140, respectively).

In Figure 149 is shown the computational time function of the mesh size for the three-dimensional mesh with embedded body. As expected the computational time is a linear function of the mesh size.

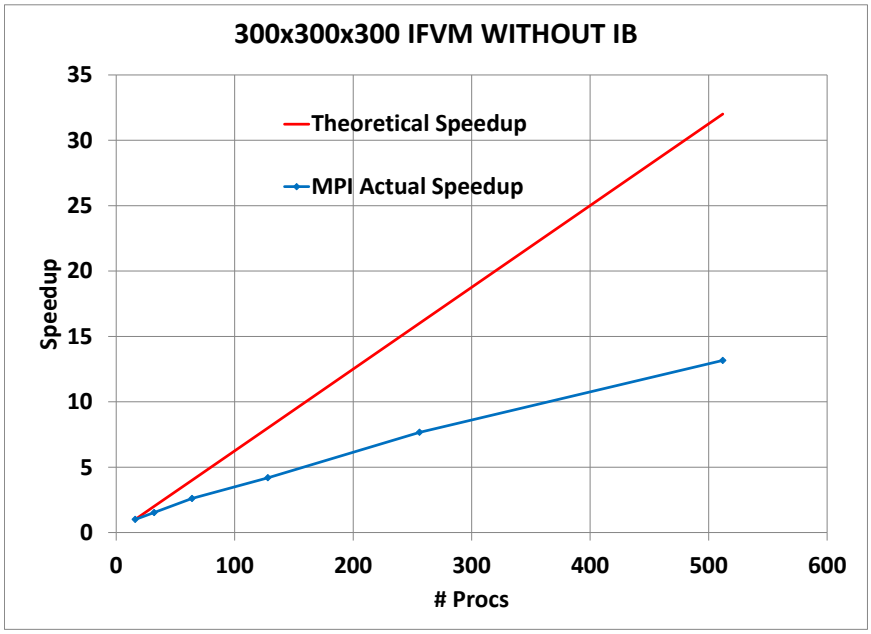

Figure 146. Speedup for a three-dimensional 300x300x300 mesh with immersed boundary (two-dimensional decomposition).

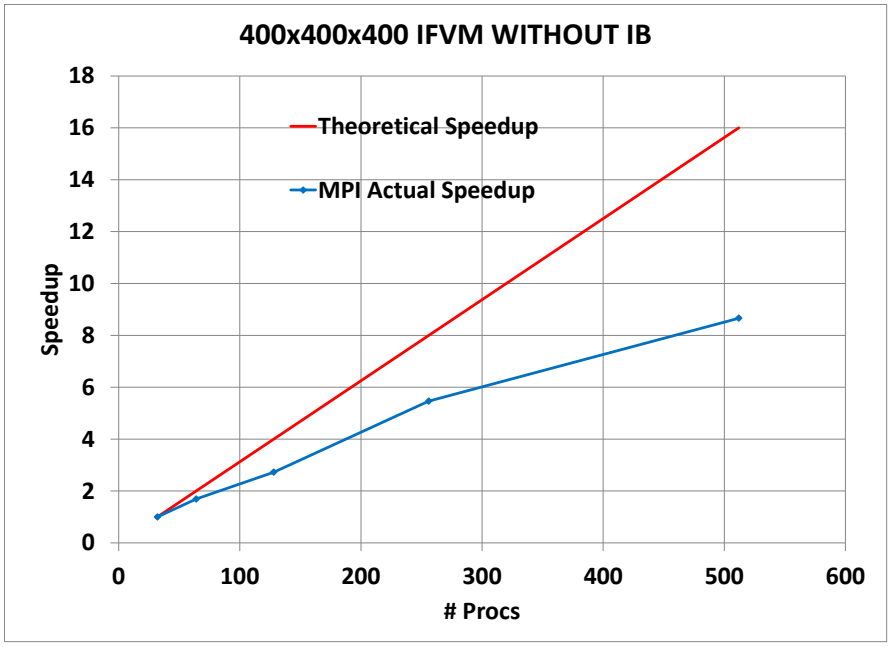

Figure 147. Speedup for a three-dimensional 400x400x400 mesh with immersed boundary (two-dimensional decomposition).

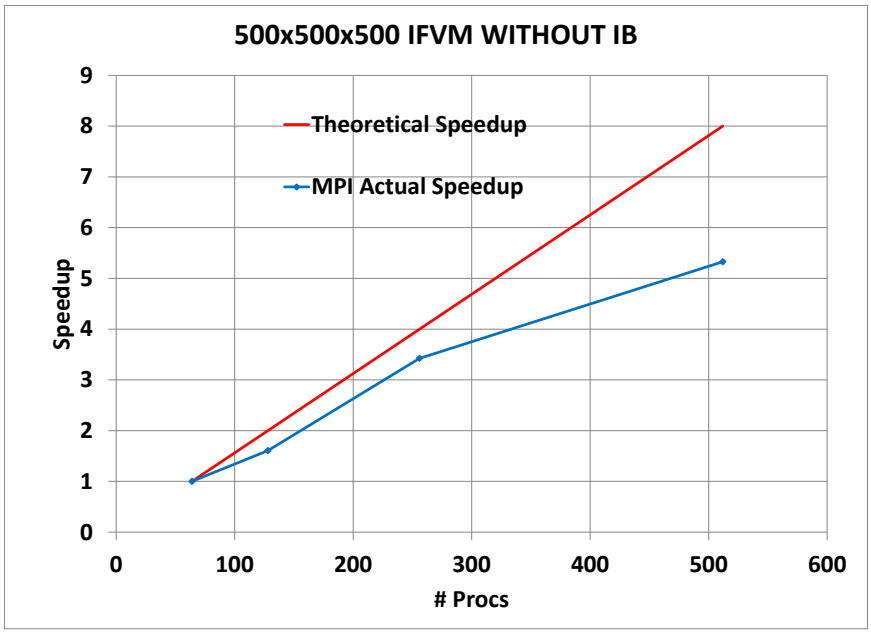

Figure 148. Speedup for a three-dimensional 500x500x500 mesh with immersed boundary (two-dimensional decomposition).

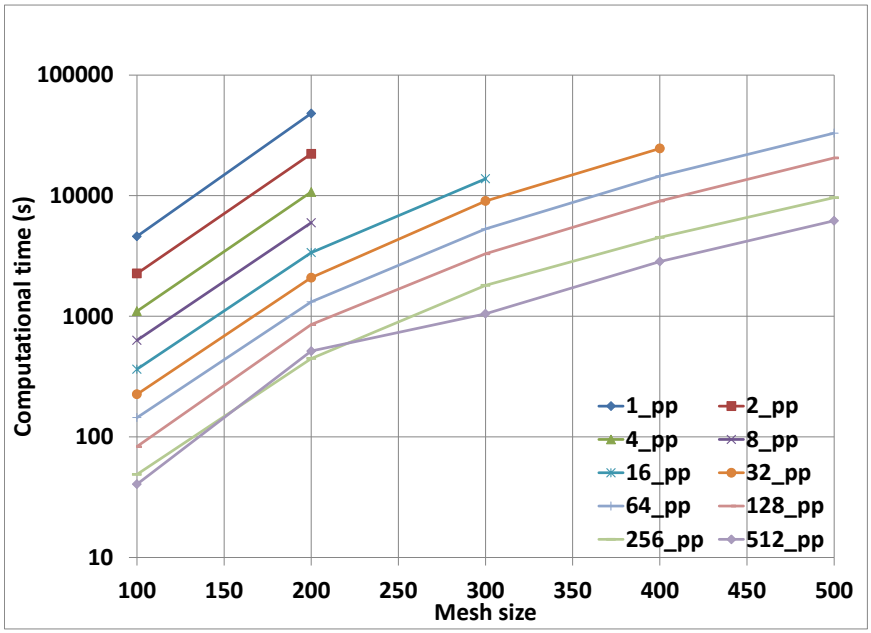

Figure 149. Computational time function of mesh size for a three-dimensional mesh with immersed boundary.

In Table 66 is shown a comparison of the computational time for the parallel IFVM code with STAR-CCM+ considering the three-dimensional natural convection in a cubic enclosure. The parameters chosen for the comparison were the mesh sizes and number of processors used to run the simulations. The number of iterations was kept constant for both codes (i.e., 50,000 iterations).

The results show that in the range 1 – 27 million cells and 16 – 64 processors the parallel IFVM code is almost three times faster than the commercial code STAR-CCM+, when a trim mesh is used for the latter. Using polyhedral meshes for STAR-CCM+ would sensibly increase the computational effort and the number of iterations required to reach convergence. It is worth mentioning that we did not consider the number of

iteration as a parameter for the IFVM/STAR-CCM+ comparison, due to the superior convergence rate of the IFVM, especially if the problem at hand is solved with the Full Multigrid Approach.

Table 66 Computational time comparison for the IFVM with STAR-CCM+ applied to the three-dimensional natural convection in a cubic enclosure for 50,000 iterations

Mesh size	IFVM (sec.)	STAR-CCM+ (sec.)	STARCCM+/IFVM
100x100x100 (16 pp.)	12723.43	32850.05	2.582
200x200x200 (32 pp.)	50136.20	146521.32	2.922
300x300x300 (64 pp.)	96217.28	271854.55	2.825

To test the performances of the parallel IFVM with immersed boundary we compared in Table 67 the IFVM computational time for a 100x100x100 mesh size with that of STAR-CCM+ when a three-dimensional natural convection in a cubic enclosure with embedded sphere problem is considered. The number of processors used was 16. The results show that with the introduction of the immersed body the parallel IFVM is approximately 2.5 times faster than STAR-CCM+ (using a trim mesh for the latter). This reduction in performance of the parallel IFVM is due to the partial utilization of computational power for the processes crossed by the solid body surface.

Table 67 Computational time comparison of the IFVM with STAR-CCM+ applied to the three-dimensional natural convection in a cubic enclosure with embedded sphere for a mesh size of 100x100x100 (16 pp.)

Rayleigh number	IFVM (sec.)	STAR-CCM+ (sec.)	STARCCM+/IFVM
10^3	3581.44	8792.02	2.455
10^4	3785.74	9364.05	2.473
10^5	3585.65	8946.08	2.495
10^6	3576.68	8895.01	2.487

7. CONCLUSIONS

A new formulation of the immersed boundary method for simulating the time-dependent incompressible Navier-Stokes equations was presented in this work. The direct-forcing IFEM implemented is algebraically equivalent to the fractional step method. Lagrange multipliers are introduced to take into account the force at the fluid/solid interface. The no-slip constraint is satisfied through a projection operator in a way similar to the divergence-free constraint for the velocity field in the classical projection approach. The boundary force is determined implicitly without using any constitutive relations (e.g., Hooke's law) for the rigid body. This allowed using larger CFL numbers. Rigid bodies for which the motion is imposed *a priori* can be simulated as well. The current scheme is found to have a second order convergence rate for the H^1 error norm, and a third order convergence rate for the L_2 error norm, respectively, when the Taylor-Green decaying vortex is simulated. Second order convergence rate for the temporal discretization was determined in agreement with the numerical scheme used.

For the direct-forcing IFVM implemented, a ghost-cell approach was used, where ghost cells are defined as additional cells inside the immersed body, with at least one neighbor in the fluid domain, and an explicit sharp interface scheme was implemented to determine the forcing terms at the ghost cells, in such a way to satisfy the no-slip boundary condition at the immersed interface points. Bi/tri-linear interpolation operators were used to satisfy the Dirichlet boundary conditions at the fluid/solid interface. Second order convergence rates for the L_2 error norm was achieved

for the Taylor-Green decaying vortex test case. First order convergence rate for the temporal discretization was determined in agreement with the numerical scheme used.

Computations were performed for the time-dependent incompressible Navier-Stokes equations in a two-dimensional flow over a backward-facing step, two-dimensional steady/unsteady flow past a stationary circular cylinder, three-dimensional flow past a sphere. Also conjugate heat transfer problems were considered to test the NSEs coupled with the energy equation. Two/three-dimensional natural convection in a square cavity and two/three-dimensional natural convection in an enclosure with embedded body benchmark test cases were considered.

The direct-forcing implicit diffuse interface scheme for the IFEM implemented was tested against bi-linear and bi-quadratic shape functions to address the accuracy of the projection operator (i.e., interpolation) on the Dirichlet boundary condition (i.e., no-slip and fixed temperature) imposed at the fluid/solid interface. Computations with either surface or volume mesh for the immersed body were performed, to investigate the effect of Lagrange multipliers imposition not only at the fluid/solid interface but in the entire immersed solid volume. The numerical results showed the volume mesh approach being more reliable than the surface mesh approach, due to some fluid "penetration" at the cylinder surface when the latter approach is used to discretize the immersed boundary. On the other hand, the sharp nature of the IFVM interpolation scheme allows a direct imposition of the boundary conditions at the fluid/solid interface. No fluid penetration or other unphysical behaviors close to the immersed body interface were observed for the benchmark problems considered. The IFEM and IFVM numerical results for the

benchmark test cases considered show good qualitative and quantitative agreement with the experimental data available, numerical predictions from other immersed boundary methods calculation present in literature, and the results from the commercial code STAR-CCM+.

The IFEM and IFVM discussed in this work represent two completely different approaches to immersed boundary methodologies. The IFEM uses an Eulerian/Lagrangian approach, where the field equations are solved on the Eulerian mesh and the additional constraints due to the immersed boundary are solved on a Lagrangian mesh representing the solid body. For IFVM the field equations are solved on an Eulerian mesh and the only information required from the immersed body mesh is the location of the fluid/solid interface.

The IFEM computational domain covers the fluid and solid regions, whereas the IFVM approach solves the discretized equations in the fluid region only.

The diffuse nature of the interface scheme used for the IFEM can give unphysical results at the fluid/solid boundary for coarse discretizations. The sharp nature of the IFVM interface scheme allows a direct imposition of the boundary condition at the immersed body surface avoiding any unphysical behavior at the fluid/solid interface.

The IFEM approach showed a superior numerical accuracy if compared to the IFVM approach. On the other hand, the former is computationally very expensive, meanwhile the latter showed a computational time three times smaller than the commercial CFD code STAR-CCM+.

For the IFEM we used the fractional step method to solve the NSEs. The no-slip constraint at the immersed boundary surface was solved implicitly with the momentum equations. Then a correction (i.e., projection) on the velocity field is necessary to satisfy the divergence-free constraint. This correction on the velocity field might compromise the accuracy with which the no-slip constraint is satisfied at the fluid/solid interface. The iterative nature of the SIMPLER algorithm used for the IFVM enforces divergence-free and Dirichlet boundary conditions at the fluid/solid interface upon convergence of the scheme.

A high performance IFVM for massive parallel computations was developed. The scalability analysis with/without embedded body showed linear scaling for very large meshes with hundreds of processors. This means that a correct and efficient use of the allocated resources is made, that is the first step towards high performance computational fluid dynamics. Communication is the predominant effect in controlling the total computational time for massive CFD problems where hundreds/thousands of processors are required to perform the simulation. By overlapping communication to computation we believe it will be possible to substantially increase the performance of the parallel IFVM. In this work we tested our IFEM and IFVM against benchmark problems for low Reynolds numbers (i.e., laminar flows). Industrial applications often require computations at very large Reynolds numbers. For this kind of flow problems turbulence modeling and a very fine mesh close to the wall are a must. Therefore, as future work we will focus on coupling implementing turbulence modeling and adaptive-mesh-refinement techniques for the parallel IFVM.

REFERENCES

[1] C.S. Peskin, Flow patterns around heart valves: a numerical method. J. Comput. Phys. 10 (1972) 252-271.

[2] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220-252.

[3] R.P. Beyer, R.J. LeVeque, Analysis of a one-dimensional model for the immersed boundary method, SIAM J. Numer. Anal. 29 (1992) 332-364.

[4] M. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705-719.

[5] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (1993) 354-366.

[6] A. Mayo, C.S Peskin, An implicit numerical method for fluid dynamics problems with immersed elastic boundaries, In: Fluid Dynamics in Biology: Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference, Contemporary Mathematics, 1993, pp. 261-277.

[7] E.P. Newrer, A.L. Fogelson, R.D. Guy, R.M. Kirby, Unconditionally stable discretization of the immersed boundary equations, J. Comput. Phys. 222 (2007) 702-719.

[8] E.P. Newrer, A.L. Fogelson, R.D. Guy, R.M. Kirby, A comparison of implicit solvers for the immersed boundary equations, Comput Methods Appl. Mech. Eng. 197 (2008) 2290-2304.

[9] Y. Mori, C.S. Peskin, Implicit second order immersed boundary methods with boundary mass, Comput. Meth. Appl. Mech. Eng. 197 (2008) 2049-2067.

[10] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (2005) 239-261.

[11] R. Tyson, C.E. Jordan, J. Hebert, Modeling anguilliform swimming at intermediate Reynolds number: a review and a novel extension of immersed boundary method applications, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2105-2118.

[12] Y. Kim, C.S. Peskin, 3-D parachute simulation by the immersed boundary method, Computers and Fluid 38 (2009) 1065-1079.

[13] A.L.F. Lima E Silva, A. Silveira-Neto, J.J.R. Damasceno, Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method, J. Comput. Phys. 189 (2003) 351-370.

[14] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 174 (2005) 345-380.

[15] Y.H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593-623.

[16] S. Majumdar, G. Iaccarino, P. Durbin, RANS solver with adaptive structured boundary non-conforming grids, Annu. Res. Briefs, Cent. Turbul. Res. (2001) 353-364.

[17] R. Ghias, R. Mittal, T. Lund, A non-body conformal grid method for simulation of compressible flows with complex immersed boundaries, AIAA (2004) 80.

[18] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, J. Comput. Phys. 225 (2007) 528-553.

[19] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (2008) 4825-4852.

[20] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flow with complex moving boundaries, J. Comput. Phys. 174 (2001), 345-380.

[21] T. Ye, R. Mittal, H.S. Udaykumar. W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundary, J. Comput. Phys. 156 (1999) 113-142.

[22] G. Biros, L. Ying, D. Zorin., The embedded boundary integral equation solver for the incompressible Navier-Stokes equations, Technical report, Courant Institute, New York University, 2002.

[23] D.V. Le, B.C. Khoo, J. Peraire, An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries, J. Comput. Phys. 220 (2006) 109-138.

[24] Z. Li, C. Wang, A fast finite difference method for solving Navier-Stokes equations on irregular domains, Commun. Math. Sci. 1 (2003) 180-196.

[25] D.V. Le, B.C. Khoo, K.M. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2119-2130.

[26] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, SIAM, J. Sci Comput. 25 (2003) 832-856.

[27] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annual Review of Fluid Mechanics 30 (1998) 329-364.

[28] D. Yu, M.R. Mei, L-S. Luo, W. Shyy, Viscous flow computations with the method of lattice Boltzmann equation, Progress in Aerospace Sciences 39 (2003) 329-367.

[29] S.K. Kang, Y.A. Hassan, A comparative study of direct-forcing immersed boundary-lattice Boltzmann methods for stationary complex boundaries, Int. J. Numer. Methods Fluids 66 (2011) 1132-1158.

[30] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, Computers and Structures 81 (2003) 491-501.

[31] D. Boffi, L. Gastaldi, L. Heltai, On the CFL condition for the finite element immersed boundary method, Computers and Structures 85 (2007) 775-783.

[32] L. Heltai, On the stability of the finite element immersed boundary method, Computers and Structures 86 (2008) 598-617.

[33] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, Comput. Methods Appl. Mech. Engrg 193 (2004) 2051-2067.

[34] R. Glowinski, T-W. Pan, J. Périaux, A fictitious domain method for Dirichlet problem and applications, Comput. Methods Appl. Mech. Engrg 111 (1994) 283-303.

[35] R. Glowinski, T-W. Pan, J. Périaux, A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations, Comput. Methods Appl. Mech. Engrg 112 (1994) 133-148.

[36] R. Glowinski, T-W. Pan, J. Périaux, Distributed Lagrange multipliers methods for incompressible viscous flow around moving rigid bodies, Comput. Methods Appl. Mech. Engrg 151 (1998) 181-194.

[37] R. Glowinski, T-W. Pan, T.I. Hesla, D.D. Joseph, J. Périaux, A distributed Lagrange multiplier/fictitious domain method for flow around moving rigid

bodies: application to particulate flows, Comput. Methods Appl. Mech. Engrg 184 (1999) 241-267.

[38] R. Glowinski, T-W. Pan, T.I. Hesla, D.D. Joseph, J. Périaux, A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flows, Int. J. Num. Meth. Fluids 30 (2000) 1043-1066.

[39] T-W. Pan, V. Sarin, R. Glowinski, A. Sameh, J. Périaux, A fictitious domain method with distributed Lagrange multipliers for the numerical simulation of particulate flow and its parallel implementation, Parall. Comput. Fluid Dynam 53 (1998) 467-474.

[40] N.A. Patankar, P. Singh, D. Joseph, R. Glowinski, T.W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow 26 (2000) 1509-1524.

[41] N.A. Patankar, A formulation for fast computations of rigid particulate flows, Cent. Turb. Res. Annu. Res. Briefs. (2001) 185-196.

[42] Z. Yu, X. Shao, A direct-forcing fictitious domain method for particulate flows, J. Comput. Phys. 227 (2007) 292-314.

[43] C. Diaz-Goano, P. Minev, K. Nandakumar, A fictitious domain/finite element method for particulate flows, J. Comput. Phys. 192 (2003) 105-123.

[44] N. Sharma, N.A. Patankar, A fast computation technique for the direct numerical simulation of rigid particulate flows, J. Comput. Phys. 205 (2005) 439-457.

[45] C. Veeramani, P. Minev, K. Nandakumar, A fictitious domain formulation for flow with rigid particles: A non-Lagrangian multiplier version, J. Comput. Phys. 205 (2007) 439-457.

[46] S.V. Apte, M. Martin, N.A. Patankar, A numerical method for fully resolved simulation (FRS) of rigid particle-flow interactions in complex flows, J. Comput. Phys. 228 (2009) 2712-2738.

[47] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor & Francis, 1980.

[48] H. Versteeg, W. Malalasekra, 2007. An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Prentice Hall.

[49] R. Löhner, J.R. Cebral, F.E. Camelli, S. Appanaboyina, J.D. Baum, E.L. Mestreau, O.A. Soto, Adaptive embedded and immersed unstructured grid techniques, Comput. Methods Appl. Mech. Engrg 197 (2008) 2173-2197.

[50] R. Löhner, S. Appanaboyina, J.R. Cebral, Comparison of body-fitted, embedded and immersed solutions of low Reynolds-number 3-D incompressible flows, Int J. Numer. Methods Fluids 57 (2008) 13-30.

[51] R. Codina, G. Hozeaux, H. Coppola-Owen, J. Baiges, The fixed mesh ALE approach for the numerical approximation of lows in moving domains, J. Comput. Phys. 228 (2009) 1591-1611.

[52] R. Ilinca, J.F. Hétu, A finite element immersed boundary method for fluid flow around rigid objects, Int. J. Numer. Methods Fluids 65 (2010) 856-875.

[53] R. Ilinca, J.F. Hétu, A finite element immersed boundary method for fluid flow around moving objects, Int. J. Numer. Methods Fluids 39 (2010) 1656-1671.

[54] K. Taira, T. Colonius, The immersed boundary method: A projection approach, J. Comput. Phys. 225 (2007) 2118-2137.

[55] L. Shen, E. Chan, Numerical simulation of fluid-structure interaction using a combined volume of fluid and immersed boundary method, Ocean Eng. 35 (2008) 939-952.

[56] L. Shen, E. Chan, P. Lin, Calculation of hydrodynamic forces acting on a submerged moving object using immersed boundary method, Comput. Fluids 38 (2009) 691-702.

[57] J.H. Ferziger, M. Peric, 2002. Computational Methods for Fluid Dynamics. Springer.

[58] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51-58.

[59] W. Chang, F. Giraldo, B. Perot, Analysis of an exact fractional step method, J. Comput. Phys. 180 (2002) 183-199.

[60] J. Mohd-Yusof, Combined immersed boundary/B-splines methods for simulations of flow in complex geometries, Center for Turbulence Research, Annual Research Briefs, 1997, pp. 317-327.

[61] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics, vol. 15, Springer-Verlag. New York, 1991.

[62] A.J. Chorin, Numerical solution of the Navier-Stokes equations. Math. Comput. 22 (1968) 745-762.

[63] J. Kim, P. Moin, Application of a fractional step method to incompressible Navier-Stokes equations, J. Comput. Phys. 59 (1985) 308-323.

[64] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132-150.

[65] H. Le, P. Moin, An improvement of fractional step method for the incompressible Navier-Stokes equations, J. Comput Phys. 92 (1991) 369-379.

[66] J.L. Guermond, L. Quartapelle, On the approximation of the unsteady Navier-Stokes equations by finite element projection methods, Numer Math. 80 (1998) 207-238.

[67] J.L. Guermond, P. Minev, J. Shen, Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions, SIAM J. Numer. Anal. 43 (2005) 239-258.

[68] J. Nocedal, S.J. Wright, Numerical Optimization, Springler, 1999.

[69] W. Bangerth, R. Hartmann, G. Kanschat, deal.II – a general purpose object oriented finite element library, AMC Trans. Math. Softw. 33 (2007) 1-17.

[70] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35-60.

[71] R. Mittal, G. Iaccarino, 2005, Immersed boundary methods, Ann. Rev. Fluid Mech. 37 (2000) 39-261.

[72] C.S. Peskin, The immersed boundary method, Acta Number 11 (2002) 479-517.

[73] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157-192.

[74] G.I. Marchuk, Methods of Numerical Mathematics, Springer-Verlag, 1975.

[75] N.N. Yanenko, The Method of Fractional Steps: The Solution of Problems of Mathematical Physics in Several Variables, Springer-Verlag, 1971.

[76] H. Borouchaki, S.H. Lo, Fast Delaunay triangulation in three dimensions, Comput. Methods Appl. Mech. Eng. 128 (1995) 153-167.

[77] D.J. Mavriplis, An advancing front Delaunay triangulation algorithm designed for robustness, J. Comput. Phys. 117 (1995) 90-101.

[78] F. Gibou, R.J. Fedkiw, L.T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput Phys. 176 (2002) 205-227.

[79] W-S. Su, M-C. Lai, C-A. Lin, An immersed boundary technique for simulating complex flows with rigid boundary, Computers and Fluids 36 (2007) 313-324.

[80] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (2009) 1963-1979.

[81] X. Yang, X. Zhang, Z. Li, G-W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, J. Comput. Phys. 228 (2009) 7821-7836.

[82] B.F. Armaly, F. Durst, J.C.F. Pereira, J. Fluid Mech. 127 (1983) 473.

[83] H. Mansy, P-M. Yang, D.R. Williams, Quantitative measurements of three-dimensional structures in the wake of a circular cylinder, J. Fluid Mech. 270 (1994) 277-296.

[84] D. Barkley, R.D. Handerson, Three-dimensional Floquet stability analysis of the wake of a circular cylinder, J. Fluid Mech. 322 (1996) 215-241.

[85] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (1977) 231-256.

[86] D.J. Tritton, Experiments on the flow past a circular cylinder al low Reynolds number, J. Fluid Mech. 6 (1959) 547-567.

[87] C. Wieselsberger, New data on the laws of fluid resistance, TN 84, NACA, 1922.

[88] S.C.R. Dennis, G. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100, J. Fluid Mech. 42 (1970), 471-489.

[89] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (1980), 819-855.

[90] D. Calhoun, A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, J. Comput. Phys. 176 (2002) 231-275.

[91] D. Russell, Z.J. Wang, A Cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, J. Comput. Phys. 191 (2003) 177-205.

[92] X.D. Niu, C. Shu, Y.T. Chew, Y. Peng, A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows, Physics Letters A 354 (2006) 173-182.

[93] X. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinate system: flow around a circular cylinder, J. Comput. Phys. 134 (1997) 306-315.

[94] Z. Wang, J. Fan, K. Cen, Immersed boundary method for the simulation of 2D viscous flow based on vorticity-velocity formulations, J. Comput. Phys. 228 (2009) 1504-1520.

[95] J. Park, K. Kwon, H. Choi, Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160, KSME Int. Journal 12 (1998) 1200-1205.

[96] S. Xu, The immersed interface method for simulating prescribed motion of rigid objects in an incompressible viscous flows, J. Comput. Phys. 227 (2008) 5045-5071.

[97] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454-493.

[98] J-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys. 224 (2007) 757-784.

[99] S.E. Rogers, D. Kwak, Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations, J. App. Num. Math. 8(1991) 43-64.

[100] J.R. Pacheco, A. Pacheco-Vega, T. Rodic, R.E. Peck, Numerical simulations of heat transfer and fluid flow problems using an immersed-boundary finite-volume method on nonstaggered grids, J. Numer. Heat Tr. B-Fund 48 (2005) 1-24.

[101] A. Dias, S. Majumdar, Numerical computation of flow around a circular cylinder, Technical Report, PS II Report, BITS Pilani, India.

[102] S. Marella, S. Krishnan, H. Liu, H.S. Udaykumar, Sharp interface Cartesian grid method I: An easily implemented technique for 3d moving boundary computations, J. Comput Phys. 210 (2005) 1-31.

[103] S. Haeri, J.S. Shrimpton, On the application of immersed boundary, fictitious domain and body-conformal methods to many particle multiphase flows, Int. J. Multiphase Flow 40 (2012) 38-55.

[104] Roshko, On the development of turbulent wakes from vortex streets, National Advisory Committee for Aeronautics, Washington, D.C. http://resolver.caltech.edu/CaltechAUTHORS:ROSnacarpt1191, 1954.

[105] C.H.K. Williamson, Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers, J. Fluid Mech. 206 (1989) 579-627.

[106] E. Berger, R. Wille, Periodic flow phenomena, Ann. Rev. Fluid Mech. 4 (1972) 313-340.

[107] M. Braza, P. Chassaing, H. Ha Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, J. Fluid Mech. 165 (1986) 79-130.

[108] C. Liu, X. Zheng, C.H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, J. Comput. Phys. 139 (1998) 35-57.

[109] Y. Sui, Y-T. Chew, P. Roy, H-T. Low, A hybrid immersed-boundary and multi-block lattice Boltzmann method for simulating fluid and moving-boundaries interactions, Int. J. Numer. Methods Fluids 53 (2007) 1727-1754.

[110] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a visual boundary method, J. Comput. Phys. 123 (1996) 450-465.

[111] M.S. Engelman, M. Jaminia, Int. J. Numer. Methods Fluids 11 (1990) 985.

[112] C. Ji, A. Munjiza, J.J.R. Williams, A novel iterative direct-forcing immersed boundary method and its finite volume applications, J. Comput. Phys. 231 (2012) 1797-1821.

[113] A. Belov, L. Martinelli, A. Jamenson, A new implicit algorithm with multigrid for unsteady incompressible flow calculations, AIAA 95-0049 (1995) January 9-12.

[114] M. Rosefeld, D. Kwak, M. Vinokur, A fractional step solution method for the unsteady incompressible Navier-Stokes equations in generalized coordinate system, J. Comput. Phys. 94 (1991) 102-137.

[115] J.A. Wright, R.W. Smith, An edge-based method for the incompressible Navier-Stokes equations on polygonal meshes, J. Comput. Phys. 169 (2001) 24-43.

[116] T.A. Johnson, V.C. Patel, Flow past a sphere up to a Reynolds number of 300, J. Fluid Mech. 378 (1999) 19-70.

[117] A. Gilmanov, F. Sotiropoulos, E. Balaras, A general reconstruction algorithm for simulating flow with complex 3D immersed boundaries on Cartesian grids, J. Comput. Phys. 191 (203) 660-669.

[118] G.S Constantinescu, K.D. Squires, LES and DNS investigations of turbulent flow over a sphere, AIAA Paper 2000-0540, AIAA Press, Washington, 1989.

[119] S. Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, J. Phys. Soc. Japan 11 (1956) 1104-1108.

[120] G.De Vahl Davis, Natural convection of air in a square cavity: a benchmark numerical solution, Int. J. Numer. Methods Fluid 3 (1983) 249-264.

[121] M. Hortmann, M. Peric, G. Scheuerer, Finite volume multigrid prediction of laminar natural convection: benchmark solutions, Int. J. Numer. Methods Fluids 11 (1990) 189-207.

[122] G Barakos, E. Mitsoulis, D. Assimacopoulos, Natural convection flow in a square cavity revised: laminar and turbulent models with wall functions, Int. J. Numer. Methods Fluids 18 (1994) 695-719.

[123] N.C. Markatos, K.A. Pericleous, Laminar and turbulent natural convection in an enclosed cavity, Int. J. Heat Mass Transfer 27 (1984) 765-772,

[124] T. Fusegi, J.M. Hyun, K. Kuwahara, B. Farouk, A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure, Int. J. Heat Mass Transfer, 34 (1991) 1543-1557.

[125] Z.G. Feng, E.E. Michaelides, Heat transfer in particulate flow with direct numerical simulations (DNS), Int. J. Heat Mass Transfer 52 (2009) 777-786.

[126] I. Demirdzic, Z. Lilek, M. Peric, Fluid flow and heat transfer test problems for non-orthogonal grids: benchmark solutions, J. Numer. Methods Fluids 15 (1992) 329-354.

[127] Z.S. Yu, X.M. Shao, A. Wachs, A fictitious domain method for particulate flows with heat transfer, J. Comput. Phys. 217 (2006) 424-452.

[128] M.Y. Ha, M.J. Jung, A numerical study on three-dimensional conjugate heat transfer of natural convection and conduction in a differentially heated cubic enclosure with a heat-generating cubic conducting body, J. Heat Mass Transfer 43 (2000) 4229-4248.

[129] H.S. Yoon, D.H. Yu, M.Y. Ha, Y.G. Park, Three-dimensional natural convection in an enclosure with a sphere at different vertical locations, J. Heat Mass Transfer 53 (2010) 3143-3155.

[130] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message Passing Interface, 2nd edition, MIT Press, Cambridge, MA, 1999.

[131] W. Gropp, E. Lusk, R. Thakur, Using MPI-2: Advanced Features of the Message-Passing Interface, MIT Press, Cambridge, MA, 1999.

[132] B. Chapman, G. Jost, R. Van Der Pas, Using OpenMP: Portable Shared Memory Parallel Programming MIT Press, Cambridge, MA, 2007.