AIX-MARSEILLE UNIVERSITÉ
DIRECTION GÉNÉRALE DE L'ARMEMENT

ECOLE DOCTORALE 353 Sciences pour l'Ingénieur : Mécanique, Physique, Micro et Nanoélectronique Laboratoire de Mécanique, Modélisation & Procédés Propres (M2P2)

UMR7340 Centrale Marseille 38 rue Joliot-Curie 13451 Marseille

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline: Mécanique et Physique des Fluides
Spécialité: Mécanique des fluides / Interaction Fluide-Structure

Eddy CONSTANT

Développement d'un solveur de frontières immergées dans OpenFOAM: vers le contrôle des vibrations induites par vortex dans le sillage d'un cylindre

A new IBM in OpenFOAM: towards the control of VIV in the wake of a cylinder

Soutenue le 18/12/2017 devant le jury composé de:

Elie HACHEM	Prof. Mines Paris Tech.	Rapporteur
Alfredo PINELLI	Prof. City, Univ. of London	Rapporteur
Julien REVEILLON	Prof. Univ. de ROUEN, CORIA	Examinateur
Michael SCHAEFER	Prof. TU Darmstadt	Examinateur
Loïc BOUDET	DGA	Tuteur, invité
Eric SERRE	Directeur de Recherche CNRS, M2P2	Directeur de thèse
Julien FAVIER	Maître de Conf. Aix-Marseille Univ., M2P2	Co-Directeur de thèse
Philippe MELIGA	Chargé de Recherche CNRS, M2P2	Co-Directeur de thèse

Numéro national de thèse/suffixe local: 2017AIXM0001/001ED62

Résumé

Cette thèse s'inscrit dans le contexte de la simulation et du contrôle des vibrations de structures montées sur ressort qui peuvent apparaître sous l'effet de l'interaction avec l'écoulement de sillage instationnaire. Le contrôle de ce phénomène, appelé Vibrations Induites par Vortex (VIV), est un enjeu critique dans l'optimisation de nombreux systèmes, notamment en aérodynamique autour des voilures d'avion et en hydrodynamique autour de structures offshore.

Dans cette thèse, une méthode de frontières immergées (IBM) a été intégrée dans l'algorithme PISO du code OpenFOAM, dédié à la simulation d'écoulements fluides incompressibles. La méthode de frontières immergées permet une représentation précise de corps fixes ou en mouvement, tout en conservant des maillages structurés conduisant à des algorithmes plus précis et efficaces en termes de performances numériques. Un schéma itératif basé sur des sous-itérations entre l'IBM et la correction de pression a été intégré dans le solveur PISO, permettant de conserver un solveur de Poisson rapide tout en satisfaisant simultanément la condition d'incompressibilité de l'écoulement et la condition de non-glissement à la surface. Pour calculer la divergence de l'équation de quantité de mouvement dans la boucle PISO et l'interpolation des flux, un calcul hybride orignal a été proposé avec une résolution analytique utilisant l'équation de la fonction noyau des quantités impliquant le terme force de l'IBM (quantités singulières). Un soin particulier a été apporté à la vérification et à la validation du nouvel algorithme. La convergence en maillage de différentes erreurs a été montrée au moyen d'une solution manufacturée, permettant d'analyser aussi bien les erreurs de discrétisation que les erreurs relatives à l'IBM. Le nouvel algorithm a été par la suite étendu au formalisme RANS et DDES proposés dans OpenFOAM pour la simulation d'écoulements en régimes turbulents. Une loi de paroi a été intégrée dans la méthode IBM permettant de modéliser les fines couches limites qui se développent autour des corps à grand nombre de Reynolds. Le travail de validation a été réalisé au regard des données expérimentales et numériques disponibles dans la littérature pour l'étude d'écoulements autour de cylindres et de sphères, sur une large gamme de nombres de Reynolds. Avec l'objectif de développer des lois de contrôle optimal pour le VIV, basées sur les mécanismes d'instabilité linéaire du système couplé dans le cadre de la théorie du contrôle, un solveur adjoint a été développé et validé dans OpenFOAM.

Mots clés: Interactions fluide/structure - OpenFOAM - Frontières immergées - Simulations d'écoulements turbulents - Solveur adjoint.

Abstract

This thesis is related to the simulation and the control of the Vortex Induced Vibrations phenomenon (VIV), which can result from the fluid structure interactions between an unsteady wake and the body, when the shedding frequency in the wake is close to the natural frequency of the body. The control of VIV is a critical issue when optimizing many systems, notably in aerodynamics, around aircraft wings, and in hydrodynamics, around offshore structures.

In this thesis, an Immersed Boundaries Method (IBM) was implemented into the PISO algorithm as a new library of OpenFOAM, in order to perform reliable simulations of incompressible flows around bluff bodies. The IBM allows an accurate description of fixed or moving solid obstacles embedded in the physical domain, using uniform or stretched Cartesian meshes. Owing to this feature, the maximum level of accuracy and scalability of the numerical solvers can be systematically achieved. An iterative scheme based on sub-iterations between IBM and pressure correction has been implemented into the native PISO solver of OpenFOAM. This allows one to use fast optimized Poisson solvers while satisfying simultaneously the divergence-free flow condition and the no-slip condition at the body surface. To compute the divergence of the momentum equation (in the PISO loop) and the interpolation of the fluxes, an hybrid calculation with an analytical resolution (using the kernel function equation) of the quantities involving the force term (singular quantities) has been proposed. A particular attention was paid to the verification and validation of the new algorithm. The mesh convergence of several errors was shown by means of a manufactured solution, allowing to analyze both the errors irelated to the discretization and to the IBM. The new algorithm was subsequently extended to the RANS and DDES formalism proposed in OpenFOAM for the simulation of turbulent flows. A wall law was integrated into the IBM method to model the thin boundary layers that develop around the bodies at large Reynolds numbers. Various 2D and 3D well-documented test cases of academic flows around fixed or moving solid bodies (cylinder and sphere) have been simulated and carefully validated against existing data from the literature in a large range of Reynolds numbers. With the objective of developing optimal control laws for VIV, based on the linear instability mechanisms of the coupled system within the framework of the control theory, a new adjoint solver was also developed and validated in OpenFOAM.

Keywords: Fluid/structure interactions - OpenFOAM - Immersed Boundaries Method - Turbulent flows simulations - Adjoint method

Remerciements

Je remercie chaleureusement toutes les personnes qui m'ont aidé lors de l'élaboration de ma thèse, notamment mes directeurs de thèse Eric S., Julien F. et Philippe M. . Eric pour m'avoir fait une totale confiance, pour m'avoir encouragé, Julien pour son intérêt, son soutien, sa grande disponibilité et ses nombreux conseils technique durant la thèse et Philippe pour son expertise et ses conseils avisés. Je remercie également mon tuteur Loic B. pour son accueil chaleureux à la DGA, ses nombreux conseils et son engagement quant à mon avenir professionnel malgré un emploi du temps chargé.

Ce travail n'aurait pas été possible sans le soutien d'Aix-Marseille Université ainsi que la Direction Générale de l'Armement, qui m'ont permis, grâce à leur financement de me consacrer sereinement à l'élaboration de ma thèse.

Merci à Alfredo P. et Elie H. pour leur participation à ma thèse en tant que rapporteur, leur relecture scrupuleuse du manuscrit et leurs suggestions avisées. Merci également aux membres du jury qui ont su, par leurs remarques et leurs conseils, faire avancer ma réflexion, notamment Julien R., qui m'a permis entre autre de participer aux journées OpenFoam et partager mon travail ainsi que Michael S. pour ses remarques pertinentes quant à mon travail de thèse.

Cette thèse n'aurait pu être menée à bien sans la disponibilité et l'accueil chaleureux que m'ont témoignés Michel P., Hery M., Sarah M., Chloé D., Sadia B. Sophie B. qui ont tous été d'une aide précieuse pour avancer dans cette recherche. Je remercie également les nombreuses personnes que j'ai eu l'occasion de rencontrer dans ce cadre.

Un grand merci à Marcello M., son expertise sur le logiciel OpenFoam, ses conseils professionnels et personnels, son soutien dans divers conférences et sa bonne humeur. Merci également à Aymeric L. pour ses nombreux conseils sur OpenFoam, son amitié, les échanges que nous avons eus, notamment musicaux.

Je tiens également à remercier mes collègues de bureau au laboratoire tout au long de ma thèse, Sylvain C, Jean Antoine S., Roua B.D. Jinming L. qui ont contribué à mon bien être au cours de ces 3 ans. Merci encore aux doctorantes qui m'ont accueilli et soutenu dès le début de ma thèse, Rachael A. et Marianna P. Leur amitié m'a permis de me sentir bien afin d'appréhender mon doctorat correctement. Merci aux italiens Matteo V. et Davide G pour tous les fous rires et l'aide qu'ils m'ont apporté afin d'acquérir les bases de l'italien. Merci à Elisabetta C. et Eunok Y. pour leur bonne humeur. Merci à Carlos R. B. B. pour tous les bons moments passés, son aide dans divers domaines et les débats sur des questions de la vie. Merci enfin à tous les doctorants et post-doc de l'association qui m'ont soutenu dans le projet de rassembler les jeunes chercheurs du technopole.

Также я хотел бы поблагодарить двух моих друзей из Украины, Максима С. и Сашу Д., с которыми я мог ежедневно обмениваться идеями о жизни на русском языке. Спасибо Вам за плодотворные беседы и дискуссии, которые не просто помогали развеяться и отвлечься, но так же являлись отличным уроком по улучшению моего русского языка.

Merci à la société R. Tech Engineering, Martin S., Vincent R. ainsi que mes nouveaux collaborateurs qui m'ont permis de continuer en partie mon travail sous OpenFoam et ouvert d'autres possibilités. Merci pour leur accueil chaleureux et leur soutien afin de me permettre de soutenir ma thèse dans de bonnes conditions.

Un grand merci à Liliane B. et Alexis B. pour leur soutien, leur amitié et pour avoir pris le temps de me conseiller malgré leurs emplois du temps très chargé. Merci également à tous les membres de l'ancien service « Re-entry Aérothermodynamics » d'Astrium (Ariane group) qui m'ont donné le goût de la recherche et encouragé à aller plus loin.

Je remercie mes amis de longue date qui m'ont permis de décompresser afin de mener à bien mon projet. Notamment Laetitia D., Guillaume A., David D., Thulasy B., Bee, André Mathieu, Poyoyo et beaucoup d'autres.

Au terme de ce parcours, je remercie enfin celles et ceux qui me sont chers et que j'ai quelque peu délaissé ces derniers mois pour achever cette thèse. Leurs attentions et encouragements m'ont accompagné tout au long

de ces années. Je suis redevable à mes parents. Gutemberg C. pour son soutien et sa confiance dans ce que j'ai entrepris depuis le début. Merci pour son implication dans mes différents changements de mode de vie malgré des épreuves très difficiles, il a toujours été présent.

J'ai une pensée particulière pour ma première fan, ma mère Odile JA. Sa présence et ses encouragements sont pour moi les piliers fondateurs de ce que je suis devenu et de ce que je fais. Elle s'est acharnée depuis tout petit afin que je réussisse et que je ne manque de rien.

Ces remerciements ne peuvent s'achever sans une pensée pour ma compagne Elena H., pour son soutien moral et pratique et sa confiance indéfectible dans mes choix. C'est en très grande partie grâce à elle que j'ai pu réaliser ceci et de notre amour qui a grandi et énormément évoluer durant ces 3 années de thèse.

Il n’est pas question de livrer
le monde aux assassins
d’aube.

"Nouvelle bonté",
Aimé Césaire

N’y eût-il dans le désert
qu’une seule goutte d’eau
qui rêve tout bas, dans le
désert n’y eût-il qu’une
graine volante qui rêve tout
haut.

"Blanc à remplir sur la carte
voyageuse du pollen",
Aimé Césaire

Contents

List of Figures

List of Tables

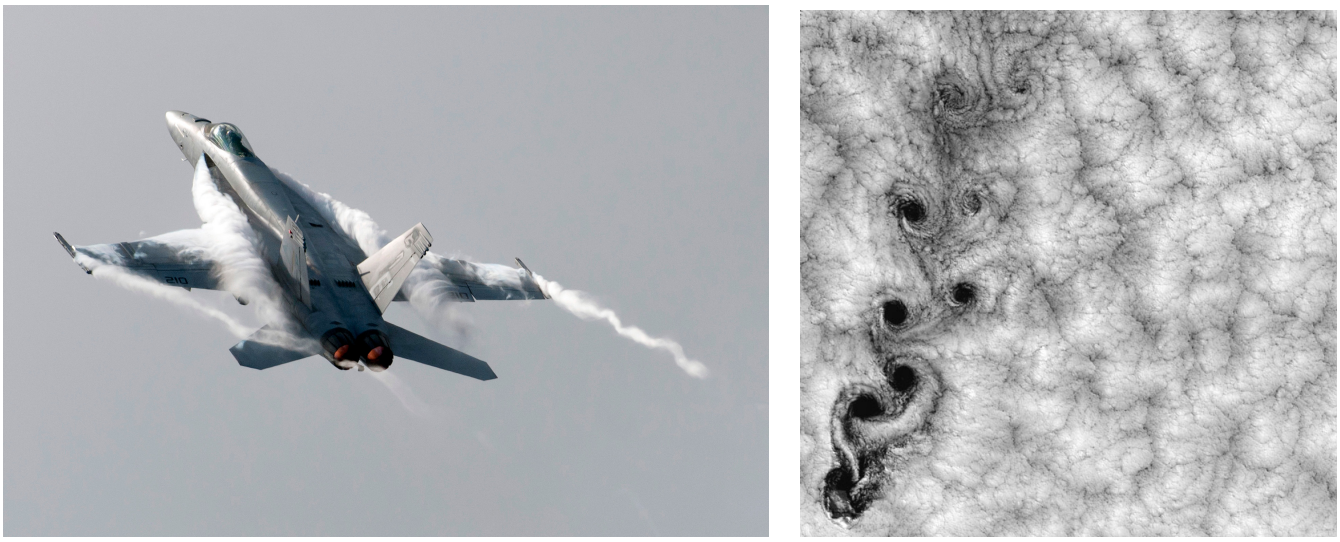
Chapter 1

Introduction

This chapter presents the context of the thesis as well as the general physical and numerical concepts that will be used later on in the manuscript.

1.1. Context of the study

In this thesis, we are interested in the numerical modelling of flows around obstacles using the open source toolbox, OpenFOAM. OpenFOAM provides an efficient coding and a suitable environment for the implementation and the rapid dissemination of new algorithms to the users community, with the target to be used further to investigate real industrial problems.

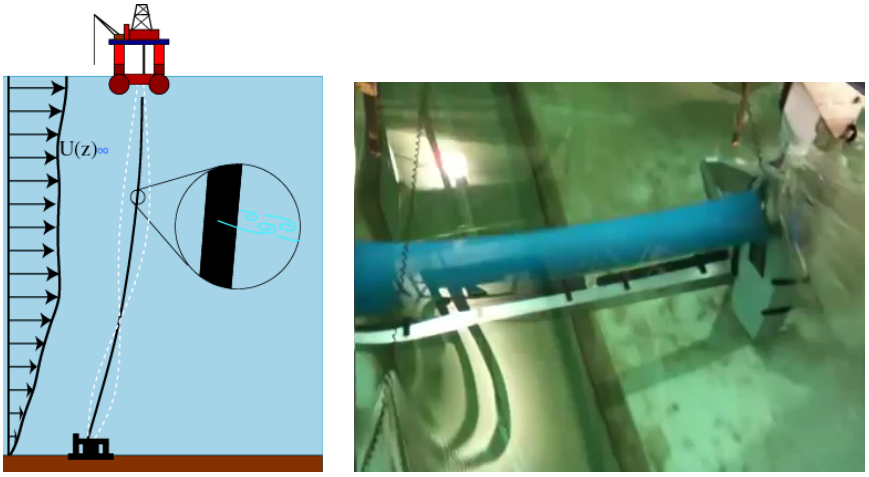


(a) (b)

Figure 1.1.: *Examples of engineering and environmental configurations involving flows around obstacles: (a) Aerodynamics. Flow around an aircraft showing vortex tip. (b) Flow around the Selkirk island in Pacific ocean showing a von-Karman vortex street in the atmosphere.*

Flows around obstacles are of practical interest for many engineering applications and natural systems (Figures 1.1, 1.2). Engineering applications, for which system's performances can be improved, range from aerodynamics (ex: flows over vehicles, ..), civil engineering (ex: flows around building, long span bridges, ...), to hydrodynamics (ex: offshore petroleum industry, naval engineering, ...). Natural systems concern environmental and geophysical flows (ex: flow around rocks in a river, atmospheric flow past an island, ...).

The obstacle, embedded in a surrounding fluid, always leads to complex physical features at high Reynolds numbers, such as unsteady flow separations and vortex shedding. Vortex shedding is accompanied by unsteady forces exerted by the fluid on the obstacle. Taking into account these fluid/solid interactions adds complexity, and requires to model both the fluid flow and the solid motion in a coupled manner. This is an area of active research in many complex fields such as computational aeroelasticity, biomechanics or turbomachinery, where each domain has its own specificity leading to the development of different methods of resolution. A famous example of fluid/structure interactions in civil engineering is related to long span bridges, which are susceptible to wind loads excitation due to the shape of the deck and their flexibility. The adequacy and safety of the structures are vital when exposed to wind action. In the case of flexible bodies with bluff cross-section, these unsteady forces may lead to structural vibrations. These vibrations lead to noise generation and drag increase, and may be detrimental to industrial systems. On the other hand, they may be desired, as in the context of flow energy harvesting, where they can be used as a mechanical energy converter in air (VIVACE shown Figure 1.2).When the body oscillation and the unsteady wake synchronize, a mechanism referred to as lock-in, the



(a) (b)

Figure 1.2.: *Examples of engineering configurations involving flows around obstacles: riser oscillations in offshore petroleum extraction (a),VIVACE test to extract energy from cylinder oscillations. MHL, University of Michigan (right)* .

phenomenon is called Vortex-Induced Vibrations (VIV). In offshore petrol industry, these VIV lead to large amplitude vibrations of offshore risers leading to premature fatigue or even failure of these structures (Figure 1.1); understanding and predicting VIV is thus crucial. This mechanism will be described in detail below in Sec. 1.2.2.

An accurate description of these flows can be achieved using efficient numerical methods implemented in versatile and powerful numerical tools that can be further used to study real industrial problems. In *OpenFOAM*, immersed bodies are primarily accounted by the use of wall-boundary conditions. However, when dealing with complex geometries, this approach leads to significant deformations of the computational mesh. On the one hand, this yields non-negligible numerical errors that are usually difficult to estimate. On the other hand, although body-fitted coordinate systems may yield a well-suited discretization of given geometry (Ferziger and Peric [28]), the grid generation may become a prohibitive issue if the geometry varies in time, as is commonly encountered in fluid-structure interaction problems. This clearly stresses the need to develop specific, advanced numerical techniques to address such complex configurations. An alternative, and more recent approach, is a class of numerical methods gathered under the name Immersed Boundary Method (IBM). In these methods, the body is modeled via an additional discrete source terms in the Navier-Stokes equations. The location of this source is defined by a set of Lagrangian points, which describes the geometry of the body.

When turbulent flows are considered, the prediction of the flow dynamics becomes much more challenging. On one side, the space and time evolution of the thin boundary layers and flow separation around the obstacle as well as the large range of scales developping in the near wake are very demanding in terms of resolution. It is why reduced order simulations (RANS), have been developed for a long time, particularly when complex flow configurations of industrial interest are involved. On the other side, the representation of complex geometries usually results in poor characteristics of the computational mesh, which has a strong impact on the final prediction. With the IBM the Cartesian mesh allows us to reduce the complex non-linear interactions between turbulence modelling and discretization error.

In the present work, the IBM is modified in order to incorporate a turbulence hybrid model (DES) with wall-functions, which has received attention in the last decade for industrial studies. The wall distance, which

is not directly available when using the IBM method, is derived exploiting information in the location of the Lagrangian points. These developments allow us for the prediction of turbulent flows around bluff bodies, combining IBM and turbulence reduced order models.

1.2. Flow past a bluff body

When spanwise or streamwise dimensions of the obstacle compare with its height, the shape of the obstacle significantly disturbs the flow, and the obstacle is called *bluff body.* Flows past circular or square cylinders, past spheres are often used in the literature as canonical cases of bluff-body flows.

The obstacle diverts the flow, which must locally accelerate to pass by, and vortex lines become stretched and clustered. Flow separation may occur on the upstream side of the body, the resulting vortices being stretched along the flanks, to cluster around the wake downstream.

The flow past the bluff body is generally characterized by the Reynolds number, Re:

$$Re = \frac{U_\infty D}{\nu} \tag{1.1}$$

where :

- D is the characteristic length of the body,
- U_∞ is the inflow velocity,
- $\nu = \frac{\mu}{\rho}$ is the kinematic fluid viscosity.

A fluid past the surface of a body exerts a force which can be decomposed into lift and drag forces, into the crossflow and streamwise direction of the flow, respectively. They both depend on time and pressure near to the wall according to :

$$F_L(t) = \oint p_{wall}(t).\mathbf{n} \cdot \mathbf{e_1} dS \tag{1.2}$$

$$F_D(t) = \oint p_{wall}(t).\mathbf{n} \cdot \mathbf{e_0} dS + \textit{viscous effects} \tag{1.3}$$

with F_L and F_D define the lift and drag forces, p_{wall} the pressure at the wall, $\mathbf{n}$ the normal to the wall, $\mathbf{e_0}$ and $\mathbf{e_1}$ the horizontal and vertical base vectors.

1.2.1. Example of the vortex shedding past a circular cylinder

The flow past a circular cylinder is a canonical flow around bluff body. The flow regime depends on the Reynolds number, as shown in Table 1.3. When increasing Reynolds number, vortex shedding occurs, known as Von Karman street, and the flow becomes unsteady, characterized by the vortex shedding frequency. The non-dimensional vortex shedding frequency, also called Strouhal frequency, varies as a function of the Reynolds number as :

$$St = \frac{fD}{U_\infty} \tag{1.4}$$

where f is the frequency of vortex shedding.

This phenomenon is amplified when increasing Reynolds number. The transition from a symmetric steady wake to an antisymmetric unsteady wake accompanied by the alternate shedding of counter-rotating vortices occurs near $Re \simeq 47$ as shown Figure 1.3. Even though the flow behavior greatly varies when the Reynolds number is increased, the vortex shedding phenomenon persists up to very high Reynolds numbers. Therefore, most of the systems involving an immersed bluff body involve vortex shedding.

As already mentionned, the vortex shedding phenomenon is accompanied by unsteady fluid forces induced by the vortices, as illustrated on Figure 1.5. The pressure in the vicinity of the structure is modified, leading to a significant unsteady cross-flow force (i.e. perpendicular to the oncoming flow), occurring at the Strouhal frequency and exerted on the body (shown Figure 1.4) . The amplitude of the cross-flow force greatly varies as a function of Re.

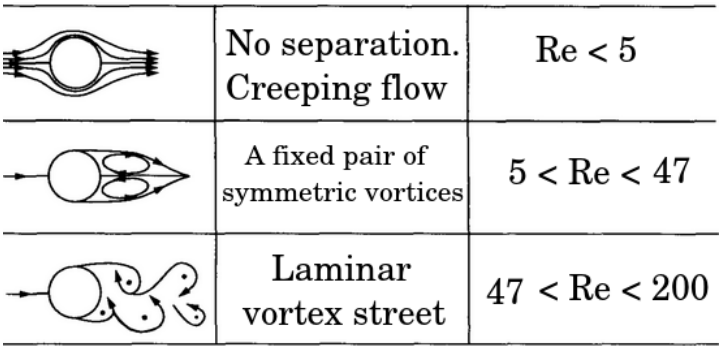


Figure 1.3.: *Laminar flows topology past a cylinder.*

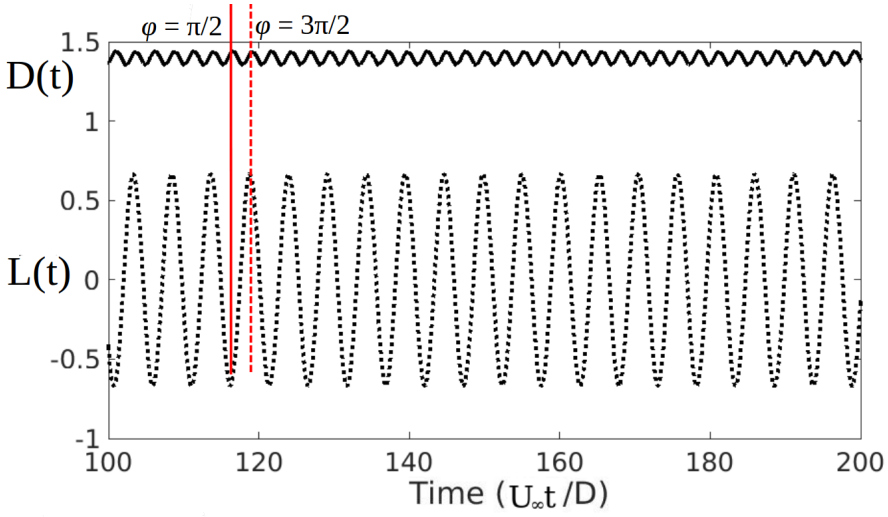


Figure 1.4.: *Time evolution of drag $D(t)$ and lift $L(t)$. Flow past a fixed cylinder at $Re = 185$. OpenFOAM computations.*

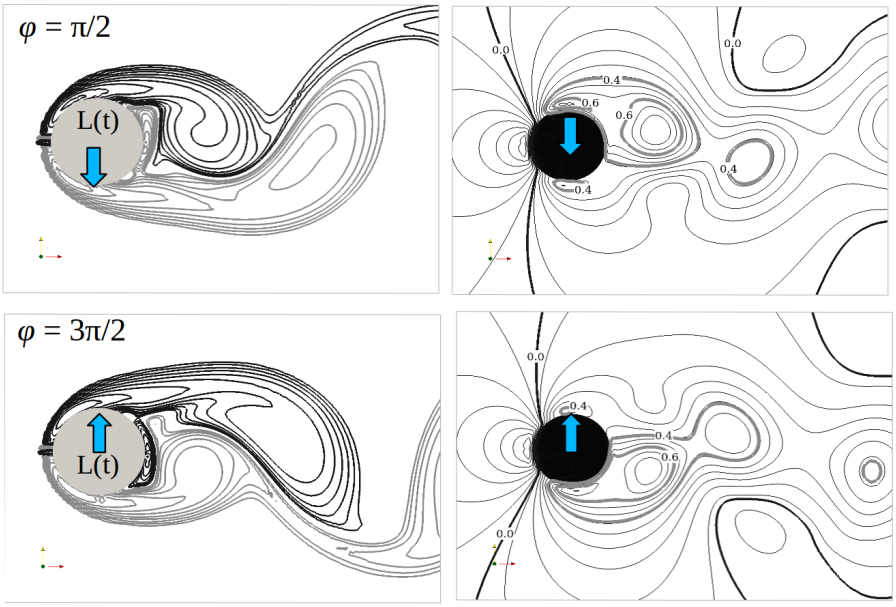


Figure 1.5.: *Flow past a fixed cylinder at* $Re = 185$ *for 2 differents lift phases* $\phi = \frac{\pi}{2}$ *(Up) and* $\phi = \frac{3\pi}{2}$ *(down). OpenFOAM computations showing iso-contours of vorticity (left) and pressure (right).*

1.2.2. The Vortex-Induced Vibrations (VIV)

VIV occur when the Strouhal frequency approaches a natural frequency of the flexible body. The flow can then transfer energy to the structure, and an oscillatory response may occur. When the body oscillates, it substantially alters the flow around it. A particular aspect of this flow-structure coupling is the synchronization of the vortex shedding and body oscillation, a mechanism referred to as lock-in (see Figure 1.6). The lock-in is defined as the state when the frequency of the periodic wake vortex mode f matches the cylinder oscillation frequency f_N. These two frequencies may, however, be different from the natural frequency of the spring–mass system, i.e. f/f_N need not be necessarily equal to 1.0 at lock-in.

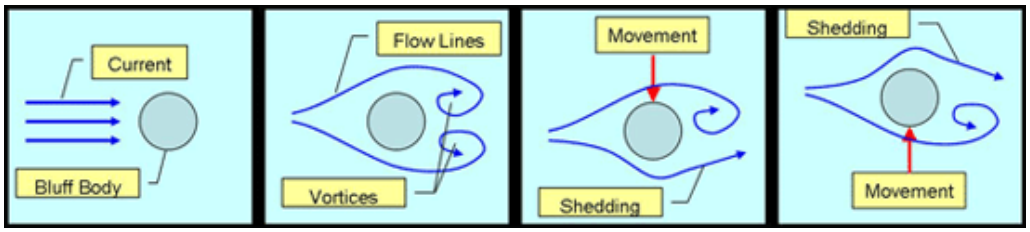


Figure 1.6.: *Sketch showing VIV for the flow past a cylinder.*

The lock-in condition is generally used as a criterion defining VIV among other flow-induced vibrations. Wake-body synchronization can occur in conditions where the Strouhal frequency and the structure natural frequency significantly depart from each other. For a given Strouhal frequency, the structure may therefore vibrate over a wide range of natural frequencies, called the lock-in range. The width of the lock-in range, as well as the evolution of the oscillation amplitude and frequency over this range, are difficult to predict.

The wide range of applications of VIV as well as the complexity of their physical behavior have motivated many studies. Even though most of real systems subjected to VIV involve flexible slender bodies, VIV have

been extensively studied through the canonical problem of a rigid circular cylinder mounted on an elastic support allowing oscillations in the cross-flow direction. This simplified configuration allows to study the synchronized oscillations with a limited number of structural parameters and vibration modes. Large amplitude vibrations occur over a well-defined range of the reduced velocity U_r, defined as the inverse of the oscillator natural frequency f_n normalized by the cylinder diameter and tne oncoming flow velocity ($U_r = U_\infty / f_n D$).

The alteration of this typical behavior when moving towards conditions closer to those encountered in natural or industrial systems is a crucial issue, of which many aspects remain to be clarified. This aspect is particularly true when *turbulent flows* are considered. This is one of the motivation of the present work that we would like to address in a close future.

1.3. The incompressible Navier-Stokes equations

The incompressible flow of a viscous, Newtonian fluid is governed by the following dimensionless Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0, \tag{1.5}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} \tag{1.6}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure. Re is the Reynolds number defined in Eq. 1.1.

1.4. *OpenFOAM* as simulation tool

The long-term goal of this thesis being the development of an efficient code to investigate flows around bluff bodies in versatile configurations relevant with real industrial flows, our choice has been made on *OpenFOAM* for solving incompressible Navier-Stokes above, Eq.1.6.

OpenFOAM (for "Open source Field Operation And Manipulation") is an extended repository of C++ libraries, released under the GNU Public license (GPL) and running on massive parallel computers. *OpenFOAM* (ESI-OpenCFD [23]) is able to simulate a wide range of flows relevant with many industrial systems. Finally, it also allows us to rapidly disseminate our new algorithms and physcial knowledge of these flows to the industry, where an increasing number of companies use it in their R & D department.

Indeed, *OpenFOAM* has gained a vast popularity during the recent years as the user is provided with existing solvers and tutorials allowing for a quick start to using the code. The software is now extensively used both in academic research (see among others the papers by Tabor and Baba-Ahmadi [131], Meldi et al. [74], Lysenko et al. [70], and Komena and Shamsa [58]) and for industrial flows analysis (Ref [115] ,[29], [31]). *OpenFOAM* solvers can also be freely modified to become more efficient, and several papers in the literature deal with the implementation of new numerical techniques or models in *OpenFOAM* (see among others the papers by Flores et al. [29], Towara et al. [138], and Vuorinen et al. [144]).

1.4.1. The finite-volume discretization

OpenFOAM is based on a finite-volume discretization. The method is locally conservative because it is based on a "local balance" approach. On each discretization cell, which is called "control volume", an integral formulation of the fluxes over the boundary of the control volume is done. Then based on these fluxes, it will be particularly well-suited to describe the dynamics of the flows under consideration in this thesis, which are non-linear with a notion of transport. The method is based on the integral form of incompressible Navier-Stokes

equations (see Eq.). The flow domain is divided into a finite number of control volumes in order to integrate the flow quantities. The equation writes:

$$\frac{\partial \mathbf{\Psi}}{\partial t} + \nabla \cdot \mathbf{F}_i = \nabla \cdot \mathbf{F}_v \tag{1.7}$$

where $\mathbf{\Psi} = \{1, u, v, w\}$ is the conservative variables vector, $\mathbf{F}_i = f \cdot \mathbf{e_0} + g \cdot \mathbf{e_1} + h \cdot \mathbf{e_2}$ is the inviscid flux vector, $\mathbf{F}_v = F \cdot \mathbf{e_0} + G \cdot \mathbf{e_1} + H \cdot \mathbf{e_2}$ is the viscous flux vector and where $\mathbf{f} = \{u, u^2 + p, uv, uw\}$, $\mathbf{g} = \{u, vw, v^2 + p, uw\}$, $\mathbf{h} = \{u, vw, uv, w^2 + p\}$, $\mathbf{F} = \{0, \tau_{xx}, \tau_{xy}, \tau_{xz}\}$, $\mathbf{G} = \{0, \tau_{yx}, \tau_{yy}, \tau_{yz}\}$, $\mathbf{H} = \{0, \tau_{zx}, \tau_{zy}, \tau_{zz}\}$.

Integrating the Eq. 1.10 over a control volume Ω leads to :

$$\frac{\partial}{\partial t} \iiint_\Omega \mathbf{\Psi} dV + \iiint_\Omega [\nabla \cdot \mathbf{F}_i(\mathbf{\Psi}, \mathbf{n}, p) - \nabla \cdot \mathbf{F}_v(\tau)] dV = 0 \tag{1.8}$$

where $\partial\Omega$ denotes the surface of the control volume.

Using the Gauss theorem :

$$\iiint_\Omega (\boldsymbol{\nabla} \bullet \mathbf{\Psi})\, d\Omega \equiv \oiint_{\partial\Omega} \mathbf{\Psi} \bullet \mathbf{n} dS \tag{1.9}$$

we get :

$$\frac{\partial}{\partial t} \iiint_\Omega \mathbf{\Psi} dV + \oiint_{\partial\Omega} [F_i \cdot \mathbf{n} - F_v \cdot \mathbf{n}] dS = 0 \tag{1.10}$$

In *OpenFOAM*, all flow variables are computed at the cell center (collocated method). This method, introduced by [107], is much easier too implement than a staggered grid approach since the integration can be done locally, all the information being contain on each cell. The grid generation is also easier for the user, and the mesh can easily be composed of polyhedral elements without any restriction on the number of faces enclosing a control volume. Among its weaknesses however, the cell center velocity is only approximately divergence-free (see Ref [27] ,[109],[73]), as the exact mass conservation is ensured on the interpolated velocity on the faces thanks to the fluxes. It is even possible to obtain unreal pressure field using pressure velocity coupling like SIMPLE (see for example in Ref [92] ,[152],[16]).

Although higher-order schemes are available in *OpenFOAM* to integrate the equations at the cell center, the second-order accurate midpoint rule is used here. This method makes the hypothesis that the value of the variable at the center of a control volume represents the mean value throughout the control volume in order to get a finite number of linear equations that can be solved using matrix methods. This assumption leads to:

$$\iiint_\Omega \mathbf{\Psi} dV \approx \boldsymbol{f}(\Psi_p) V \tag{1.11}$$

where V is the volume of a control volume Ω and Ψ_p is the mean value of Ψ through the p^{th} control volume.

To pass from cell centered quantities to face centered quantities we choose the linear interpolation (central-differences scheme) in *OpenFOAM* among several other available schemes. This scheme is second-order accurate (whereas an upwind is only first-order) and it offers an acceptable robustness and CPU cost. Volume and surface integrals of Eq. 1.10 are then linearised using appropriate schemes:

- Viscous term

$$\iiint_\Omega \nabla \cdot (F_v) dV = \iiint_\Omega \nabla \cdot (\frac{1}{Re} \nabla \Psi) dV \tag{1.12}$$
$$= \oiint_{\partial\Omega} (\frac{1}{Re} \nabla \Psi) \cdot \mathbf{n} dS \tag{1.13}$$
$$\equiv \sum_f \frac{1}{Re} S_f \mathbf{n} \cdot (\nabla \Psi) \tag{1.14}$$

with f all the faces, S_f the surface area. For a cartesian mesh, $S_f \mathbf{n} \cdot (\nabla \Psi) = S_f \frac{\Psi_N - \Psi_P}{d_{\overline{NP}}}$, N the neighbourg cells (north,south,east,west) and $d_{\overline{NP}}$ the distance between the local and neighbour cell.

- Inviscid term

$$\iiint_\Omega \nabla \cdot (F_i) dV = \iiint_\Omega [\nabla \cdot (\mathbf{U}\Psi) + \nabla p] dV \tag{1.15}$$
$$= \oiint_{\partial\Omega} [(\mathbf{U}\Psi) \cdot \mathbf{n} + p\mathbf{n}] dS \tag{1.16}$$
$$\equiv \sum_f S_f \mathbf{n} p + S_f \mathbf{n} \cdot \mathbf{U}_f \Psi_f \tag{1.17}$$
$$= \sum_f S_f \mathbf{n} p + F \Psi_f \tag{1.18}$$

with $F = S_f \mathbf{n} \cdot \mathbf{U}_f$ is the flux at the face f. The face flux Ψ_f can be evaluated thanks to a large panel of schemes in *OpenFOAM*. The scheme used here is the central scheme $\Psi_f = \frac{\overline{fN}}{\overline{PN}} \Psi_P + (1 - \frac{\overline{fN}}{\overline{PN}}) \Psi_N$ with $\overline{fN}$ the distance between the local and the neighbour cell.

- Temporal term
Backward differencing scheme that is second-order accurate in time is used here. The variables at the $(n-2)^{th}$ time-step are stored in order to solve the problem. The equation reads :

$$\frac{\partial}{\partial t} \iiint_\Omega \mathbf{\Psi} dV = \frac{3(\Psi_P \mathbf{U})^n - 4(\Psi_P \mathbf{U})^{n-1} + (\Psi_P \mathbf{U})^{n-2}}{2\Delta t} \tag{1.19}$$

with Δt the time step. The system is then integrated in time.

Then the resolution of the equation is reduced to a set of algebraic system of equations expressed as $[A]\{x\} = \{b\}$ with $[A]$ a square matrix, x the column vector of dependent variables and b defined by boundaries.

1.4.2. The SIMPLE and PISO algorithms for solving the velocity-pressure coupling

The incompressibilty of the fluid is guaranteed by the continuity equation (1.5). Taking the divergence of equation (1.6) and using the continuity equation (1.5) yields classically a Poisson equation for the pressure :

$$\nabla^2 p = -\nabla \cdot (\mathbf{u} \nabla \mathbf{u}) \tag{1.20}$$

Eqs. (1.6) and (1.20) couple pressure and velocity in an elliptic manner that requires specific numerical algorithms.

The SIMPLE algorithm (Semi-Implicit Method for Pressure Linked Equations) proposed by Patankar and Spalding [93] is widely used in the commercial softwares (CFX,FLUENT...). The idea of the method is to introduce a predictor and a corrector step. To do so, a velocity guess is first calculated from the momentum equation (1.6) using the pressure at the previous time step. The pressure is then updated through the Poisson equation, and the velocity is finally recalculated into a loop. Several iterations are required to satisfy the divergence free condition. In order to do so, the general discretization of the equations reads on one cell :

$$a_{ii}\mathbf{u}_{ii}^{n} = \sum_{j}\sum_{k} a_{jk}\mathbf{u}_{jk}^{n} - \nabla p^{n} + \mathbf{S}^{n-1} \text{with} j \neq k \tag{1.21}$$

$$\nabla \mathbf{u}_{ii}^{n+1} = 0 \tag{1.22}$$

with ii refering to the local cell and jk to its neighbour.

In this original method, the contribution of the neighbour cells on the calculation of the pressure correction was neglected. To avoid an eventual divergence of the method as detailed in Versteeg and Malalasekera [143] , an under relaxation scheme is used that may however slow down the calculation. It is why, later on Patankar [92] and Doormaal and Raithby [21] introduced the SIMPLER (SIMPLE Revised) and SIMPLEC (SIMPLE Consistent) respectively, in order to improve the original SIMPLE algorithm. These improved versions of the algorithm deal with different consideration for the treatment of the neighbour cells on the calculation of the pressure correction.

1.4.2.1. original PISO algorithms

In this thesis, we will use the native PISO algorithm of *OpenFOAM*. Introduced by Issa [48] , the original PISO follows as well the method of the SIMPLE algorithm. Designed initially for transient flows, the original method is composed by a 3 steps procedure at each time step n :

1. Predictor step:

   a) An estimate velocity $\mathbf{u}^{\star,1}$ is calculated from the Navier-Stokes equations:

   $$\frac{\partial \mathbf{u}^{\star,1}}{\partial t} + \nabla \cdot (\mathbf{u}^{\star,1}\mathbf{u}^{\star,1}) = -\nabla p + \frac{1}{Re}\nabla^{2}\mathbf{u}^{\star,1} \tag{1.23}$$

   $\mathbf{u}^{\star,1}$ is the guess value of the velocity in the iterative PISO loop

2. PISO loop:
   For the sub-iteration $m = 1$ to $M - 1$, and up to convergence:

   a) At each sub-iteration, a pressure field $p^{\star,m}$ is calculated from the Poisson equation :

   $$\nabla^{2} p^{\star,m} = -\nabla \cdot (\mathbf{u}^{\star,m}\nabla \mathbf{u}^{\star,m}) \tag{1.24}$$

   which gives in discretized form :

   $$\nabla[\{a_{ii}^{-1}\}(\nabla p^{\star,m})] = \nabla(\sum_{j}\sum_{k} a_{jk}\mathbf{u}_{jk}^{\star,m} + \mathbf{S}^{n-1}) \tag{1.25}$$

b) The velocity field is thus corrected using:

$$\mathbf{u}^{\star,m+1} = \{a_{ii}^{-1}\}(-\nabla p^{\star,m} + \sum_j \sum_k a_{jk}\mathbf{u}_{jk}^{\star,m}) \tag{1.26}$$

with a prescribed number of PISO loop of $M = 2$ in Ref [48] . This method is then more efficient than the SIMPLE family of algorithms because it requires less sub-iterations. Nevertheless, as the method is semi-implicit the time step has to remain relatively low in order to satisfy stability conditions.

1.4.2.2. OpenFoam PISO algorithms

As an improvement, an undocumented extra term on the mass flux in the pressure correction loop is implemented in the OpenFOAM PISO solver. The flux calculation from the velocity field then reads :

$$\phi^n = \phi_1^n + \phi_2^n \tag{1.27}$$

$$\phi_1^n = \mathbf{u}^n \cdot \mathbf{n}dS| \tag{1.28}$$

$$\phi_2^n = \frac{\alpha}{\Delta t}(a_p^n)^-1|_f \phi^{n-1} - (a_p^n \mathbf{u}^{n-1})|_f \cdot \mathbf{n}dS \tag{1.29}$$

$$\alpha = 1 - min(\frac{|\phi^{n-1} - \mathbf{u}^{n-1} \cdot \mathbf{n}dS|}{|\phi^{n-1}| + \epsilon}, 1) \tag{1.30}$$

with :

- $|...|_f$ the values evaluated on the faces
- S the area of the face
- ϕ the flux used in the PISO loop
- ϕ_1 the original calculation of the flux in the PISO loop of Ferziger and Peric [28]
- ϕ_2 the corrected term of the flux in the PISO loop
- ϵ a small value in order to avoid a division by zero

In a recent paper, Vuorinen et al. [144] study the influence of the additional term ϕ_2 which act as a stabilizing extra term. Moreover, Vuorinen et al. evidence that this term is used to add some diffusion inside the equations.

1.5. The fluid/structure coupling

In this kind of problem, the fluid and the structure have their own governing equations that interact thanks to a common interface. The way to couple both systems is a numerical issue that requires very efficient and accurate numerical algorithms. The strategy strongly depends on the needs related to the problem under consideration. There exist two families of solvers (Ref [39]), based on a monolithic or a partitioned approach.

- The monolithic approach implies to solve the fluid and structural equations at the same time (see for example Ref [85]). The condition at the interface is then directly embbeded into the equations. This approach is robust and accurate but requires the development of a dedicated code. Besides, fully coupled equations are most of the time computationally challenging for complex applications (Ref [100]).

- On the contrary, the partitioned approach does not require many changes as it allows us to keep the original solvers for each subproblem. This makes it easier to implement.

The main issue of all these methods is how to model and compute the interface between the fluid and the structure. Hereafter several methods to describe the fluid/structure coupling and boundary modeling are briefly reviewed. The reader is referred to reading the book of Heil et al. [39] for more details.

1.5.1. Analytic Element Method and Boundary Element Method

The analytic element method (AEM) introduced by Strack [129] , and the Boundary Element Method (see Ref [102]) rely on the discretization of internal and external boundaries. The differential equations are transformed into boundary integral equations using Green identities and the divergence theorem. One of the difference between AEM and BEM is that the boundary integrals are calculated analytically for the AEM (see Ref [64]). For the BEM the boundary is approximated by linear combination of Green functions. Then, the discretization is much smaller, meshes can easily be generated, and design changes do not require a complete remeshing. These methods can be really efficient as for the case of a red blood cell motion (see Ref [87]) Nevertheless, they are only valid for infinite or semi-infinite domains and for flows where inertial effects are negligible, that is to say to really low Reynolds number flows.

1.5.2. Arbitrary Lagrangian-Eulerian method

The Arbitrary Lagrangian-Eulerian method (ALE) provides an hybrid description of a Lagrangian method (see Ref [63]), where the grid points move with the fluid, and an Eulerian formulation (Ref [45]) which can handle excessive mesh distortion as the meshed region is fixed.

Then the computational mesh inside the domains can move with the continuum in a Lagrangian fashion, while the mesh on the boundaries and interfaces of the domains can be either fixed, in order to precisely track the boundaries and interfaces of the structure, or moved in some arbitrarily specified way to give a continuous rezoning capability. The method was first proposed for finite differences and finite volumes by Noh [88] , Franck and Lazarus [30] , Trulio [140] , and Hirt et al. [42] .

This method is widely used in the literature for fluid structure interactions, see for example in Ref [151] , [22], [54]. However it is limited to relatively small displacement or deformation, as the mesh distortion induced by this method should not be too large when the mesh is only moved. As discussed in Ref [50] , handling mesh deformation and data mapping (in order to transfer the flow field from the old tothe new mesh) can overcome this difficulty, as really distorded cells could lead to a really bad result if it converges. Nevertheless a remeshing strategy is not desirable at every time step as it introduces numerical errors, slows down the computation and there is still an issue to handle really large displacements and deformations.

1.5.3. Chimera method

Chimera is a variant of the Schwarz's algorithm, which is used in CFD to avoid meshing complicated objects (see Ref [128] ,[9]). Different components are meshed independently, and then join together using a domain decomposition technique to couple the equations solved on each component. This coupling is achieved via transmission conditions (in the finite element method) or by imposing the continuity of fluxes (in the finite volume context). Historically, the method has then been extensively used to treat moving objects, as the independent meshes are free to move with respect to the others. At each time step, the main task consists in recomputing the interpolation of the transmission conditions or fluxes.

The Chimera method is well suited for treating problems where components are moving (see Ref [72]). The independent meshes are moved as rigid bodies and the solution is recoupled when suited. This recoupling can

be very costly. However, they enable to maintain the boundary layers and local refinement around the bodies in a natural way.

1.6. The Immersed Boundary Methods

A wide spectrum of methods included in this family has proven to be efficient to simulate complex and moving geometries, such as Lagrangian multipliers (Ref [35]), level-set methods (Ref [10]), fictitious domain approaches and surface (Ref [96]) and volume penalization approaches (Ref [79] ,[47]). The present work, deals with the IBM primarily proposed in the seminal work of Peskin [97] , who introduced this method to simulate fluid-structure interactions into a cardio-vascular system (see the late, seminal paper by Peskin [98] for the mathematical foundation). A common feature of all IBM techniques is that the Navier-Stokes equations are discretized over a simple structured Cartesian grid, which significantly improves the computational efficiency and the stability. The Peskin's method mixes Euler-Lagrangian grids in order to compute the flow interactions with a flexible immersed boundary.

The way that boundary conditions are imposed differs from one IBM to another. For the simulation of a viscous incompressible flow past a body, the geometry is immersed into a larger computational domain, and the boundary conditions are represented by the addition of an ad-hoc body force in the momentum equations. This force imposes indirectly the effect of no-slip boundary conditions or other wall Dirichlet or Neumann conditions. The introduction of this forcing function into the momentum equation can be done in two ways, called either continuous or discrete forcing.

Continuous forcing method -The original method of Peskin [97] is an example of continuous forcing method. The fluid is represented on an Eulerian system of coordinate, whereas the structure is represented on a Lagrangian one, where markers define immersed solid boundaries. The forcing function f, is included into the momentum equation that evolves to :

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f} \tag{1.31}$$

After choosing an appropriate forcing function in the continuous method the equations are then discretized and solved on the whole domain. The immersed boundary is modeled as massless elastic fibers and their locations are defined using the Lagrangian grid, where each Lagrangian marker tracks a massless point that moves depending on the local fluid velocity defined on the Eulerian mesh. To model the force exerted on the elastic fibers by the blood, Peskin used the Hook's law which reads :

$$\mathbf{F}(s,t) = -K\mathbf{X}(s,t) - \mathbf{X}_0 \tag{1.32}$$

where $\mathbf{F}(s,t)$ is the force exerted on the elastic fiber at position s, K a spring constant, $\mathbf{X}_0$ the equilibrium position of the fiber, and $\mathbf{X}(s,t)$ defines the position of the elastic fiber at time t. The motion of the elastic fiber at time t, $\mathbf{X}(s,t)$, can be computed from the equation :

$$\frac{\partial \mathbf{X}(s,t)}{\partial t} = \mathbf{U}(s,t) \tag{1.33}$$

with $\mathbf{U}(s,t)$ the velocity evaluated at position s of the lagrangian space Ω_s.

The velocity $\mathbf{U}(s,t)$ then needs to be computed. It is done by the interpolation of the velocity from the Eulerian space (where the Navier-Stokes equation is computed) into the Lagrangian space (where the solid motion and boundary conditions are computed). This interpolation reads :

$$\mathbf{U}(s,t) = \mathcal{I}[\mathbf{u}](s) = \int_{\Omega_j} u(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t))d\mathbf{x} \tag{1.34}$$

where δ represent the Dirac delta function and Ω_j the eulerian space.

Then, once the solid equation is computed on the Lagrangian grid, the effect of the force should be integrated in the fluid equations (computed on the eulerian mesh) in order to compute the next time step. It is done through another interpolation of the forcing term from the Lagrangian grid to the Eulerian mesh. This operation, called spreading, is achieved by :

$$f(\mathbf{x},t) = \mathcal{S}[\mathbf{F}](x) = \int_{\Omega_s} u(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t))ds \tag{1.35}$$

The force f(x,t) is modeled by a Dirac δ function applied on the fluid. The issue is that the Lagrangian markers almost never coincide with the points of the Lagrangian grid. Then, the forcing is done through several points of the Eulerian mesh around the Lagrangian markers. The Dirac function is then smoothed through a distribution on several points. The interpolation step shows the same issue, and then, when the equations are discretized, Eulerian and Lagrangian quadratures are defined in order to take into account the discretization. The discretized functions reads :

$$\mathbf{U}_s(t) = \mathcal{I}[\mathbf{u}]_s = \sum_{D_j} u_j^n \delta(\mathbf{x} - \mathbf{X}_S(t))\Delta V \tag{1.36}$$

$$f(\mathbf{x},t) = \mathcal{S}[\mathbf{F}_s] = \sum_{D_s} u(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}_S(t))\Delta s \tag{1.37}$$

where D_j and D_s represents the discretized eulerian and lagrangian space, ΔV and Δs the eulerian and lagrangian quadratures.

Nevertheless the IBM defined by Peskin can not properly model rigid bodies characterized by a large value of the spring constant that would leading to a numerical instability. Since then, other formulations have been proposed. For instance, the so called "virtual boundary method" (Ref [4] ,[36]) defines a body surface as a virtually boundary directly embedded into the fluid. The idea is to drive the boundary velocity to rest. Then the no-slip boundary condition is directly integrated into the Navier-Stokes equations. The equations read :

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f} + \int_{\Omega} \mathbf{F}(s,t)\delta(\mathbf{x} - \mathbf{X}(s))ds \tag{1.38}$$

$$\nabla \cdot u = 0 \text{ on} \Omega_f \tag{1.39}$$

$$\mathbf{u} = \mathbf{u_b} \text{ on} \Omega_b \tag{1.40}$$

$$u(\mathbf{X}(s),t) = \int_{\Omega_j} u(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(s,t))\Delta\mathbf{x} \tag{1.41}$$

with Ω_f and Ω_b the fluid and solid domain

Goldstein et al. [36] makes the Lagrangian and Eulerian points to coincide, in order to avoid the interpolation step $\mathcal{I}[\mathbf{u}](s)$. However, the forcing term is unknown a priori that leads to estimate this value afterall through a feedback process (which leads to the fact that the method is also called "Feedback Forcing"). The forcing step reads :

$$\mathbf{F}(s,t) = \alpha \int_{\Omega_j}^{t} \mathbf{u}(s,\tau)d\tau + \mathbf{u}(s,t) \tag{1.42}$$

where u the velocity at the surface points. The form of the forcing feedback looks like a PI controller, with the P model as the proportional part and I as the integral part.

In order to model a smooth surface instead of a step-like surface, the boundary force is multiplied by a Gaussian distribution so that the neighbour Eulerian points receive a part of the force. With α and β chosen appropriety, the method gives rather good results. Nevertheless, the method produces spurious oscillations, and the associated computational time step is drastricly restricted due to numerical stability issues (see Ref [112]).

Especially for highly unsteady flows, stability problems arise due to considerable stiffness. Saiki *et al.* 1996 extended this feedback forcing approach in order to remove these spurious oscillations. They modified the forcing term which reads :

$$\mathbf{F}(s,t) = \alpha \int_{\Omega_j}^{t} [\mathbf{u}(\mathbf{X}_s,\tau) - \mathbf{v}(\mathbf{X}_s,\tau)]d\tau + [\mathbf{u}(\mathbf{X}_s,t) - \mathbf{v}(\mathbf{X}_s,t)] \tag{1.43}$$

where the velocity of the body $\mathbf{v}$ is controlled as well. Nonetheless, the feedback forcing models suffer from severe CFL restrictions related to stiffness constants (Ref [81]).

Discrete forcing method - The discrete forcing approach, also termed the direct approach, aims at overcoming the drawbacks of the continuous forcing approach, as the introduction of the force term at the discretization stage leads to a more stable and efficient algorithm (Ref [81]). This method first introduced by Mohd-Yusof and LeVeque [84] , has been developed in numerous original research works (see for examples Ref [24] ,[55], [2], [132]) including a dedicated solver in OpenFOAM (Ref [51]). In this approach, the governing equations are discretized on a Cartesian grid, neglecting the immersed boundary. After that, the discretization in the cells near the IB is adjusted to account for their presence. The grid points in the vicinity of the immersed boundary will be computed using a interpolation scheme. This approach first introduced by Mohd-Yusof and LeVeque [84] considers a forcing term which can be expressed as :

$$\frac{\mathbf{u}_i^{n+1} - \mathbf{u}_i^n}{\Delta t} = \mathbf{RHS}_i + \mathbf{f}_i \tag{1.44}$$

At each time step the velocity at the surface of the solid u_i^{n+1} should reach the desired moving boundary velocity U_d. Then we get the following relation for the forcing function :

$$\mathbf{f}_i = \frac{\mathbf{U}_d^{n+1} - \mathbf{u}_i^n}{\Delta t} - \mathbf{RHS}_i \tag{1.45}$$

Then the force is defined as :

$$\mathbf{f} = \begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) + \nabla p - \dfrac{1}{Re}\nabla^2 \mathbf{u} + \dfrac{\mathbf{U}_d^{n+1} - \mathbf{u}_i^n}{\Delta t} & \text{near } D_s \\ 0 & \text{otherwise} \end{cases} \tag{1.46}$$

The drawback of these methods is that they are sensitive to the discretization, especially that of the time derivative. In this context, the semi-implicit treatment of the viscous terms to reduce the viscous stability constraint has a direct influence on the computation of the force term (Ref [24] ,[55]). Kim et al. [55] suggested to perform a first step explicitly to compute the force, and then to add the obtained force term to the equations, treated in a semi-implicit way. Although the method is computationally efficient, the velocity field and the force term are not evaluated at the same time instant in the algorithm, which can lead to stability issues. Another important aspect which is targeted in the present work is the analysis of moving boundaries. The related velocity fields generally suffer from spurious oscillations occurring during the time-marching of the algorithm, when a mesh element occupied by the flow suddenly becomes a *solid* cell. In order to overcome these difficulties, Uhlmann [141] proposed a direct forcing method combining the strengths of both continuous

and direct forcing approaches. The method relies on the evaluation of the force term in the Lagrangian space, thus using the δ-functions originally proposed by Peskin. It has been successively improved by Pinelli et al. [99] , who introduced a new efficient quadrature for the spreading step and extended the method to non-uniform and curvilinear meshes. Owing to its modularity, stability, computational efficiency and accuracy in the analysis of moving/deformable configurations, this method has been identified as the best candidate to be implemented in the *OpenFOAM* solver. Compared to the IBM method recently implemented in *OpenFOAM* by Jasak et al. [51] , the present approach appears to be more accurate and more versatile for the study of unsteady/deforming structures, as it relies only on the accuracy of the interpolation and spreading steps, which are independent of the complexity of the geometry.

Although it is not systematically mentioned explicitly in the literature, the application of discrete forcing approaches in the context of incompressible flow solvers with predictor-corrector schemes is not straightforward. In fact, it is a two-constraints problem: on the one hand, the force term, needed to impose the no-slip condition at the solid boundary, must be calculated, and on the other hand, a divergence-free velocity at the boundaries must be satisfied. This means that enforcing divergence free conditions on the velocity affects the accuracy of the immersed boundary force at the wall. Although this issue has been claimed to be negligible by Fadlun et al. [24] , it may actually lead to significant differences depending on the configuration considered. It has been shown to systematically introduce a first-order error in time on the actual boundary values (Ref [20]). A solution has been proposed by Ikeno and Kajishima [46] which changes the matrix structure of the Poisson problem solved to compute the value of the projector term (i.e., pressure or pressure correction), by directly imposing Neumann type conditions on the immersed boundary on the corresponding matrix terms. In order to avoid changing the matrix structure, Taira and Colonius [132] have suggested to use Lagrangian multipliers associated to boundary values to impose the expected velocity condition on the immersed boundary. Those Lagrangian multipliers are obtained solving a system derived from an algebraic splitting of the full spatial operator of the Navier-Stokes equations. In the present work, we choose an iterative scheme based on sub-iterations between (IBM) and pressure correction. This allows to use fast optimized Poisson solvers while keeping control of the error made on both the velocity at the immersed boundary and the divergence of the velocity field.

1.7. The simulation of turbulents flows past a bluff body

When turbulent flows are considered, the prediction of the flow dynamics becomes much more challenging. The space and time evolution of the thin boundary layers, the flow separation around the obstacle as well as the large range of scales developing in the near wake are very demanding in terms of resolution. A theoretical estimate of the resolution requirements is provided by the Kolmogorov's theory for isotropic homogeneous turbulence, and the notion of energy cascade. On average, kinetic energy is transferred over a range of scales that have to be theoretically discretized, from the large scales of energy injection (production scales, of the same size that the obstacle) to the small scales of energy dissipation at molecular level (Kolmogorv scale) (see Ref [56] [57]).

According to Kolmogorov theory the ratii between the smallest to the largest scales behave like $Re^{-3/4}$ for the lengths and $Re^{-1/2}$ for the time. That means, for example, that at $Re = 10^6$, there are about five orders of magnitude between the sizes of the largest eddies and the smallest.

In addition, the discretization of complex geometries usually results in poor characteristics of the computational mesh. It is why models have been developped for a long time to perform reduced order simulations, particularly when complex flow configurations of industrial interest are involved. Three generic classes of simulations have been developed to solve Navier-Stokes equations in turbulent regimes (Figure 1.7): Direct Numerical Simulation (DNS), Reynolds Navier-Stokes Simulation (RANS) and Large Eddy Simulation (LES). The choice depends on the flow features, on the accuracy needed, and on the availability of numerical resources, Table 1.1.

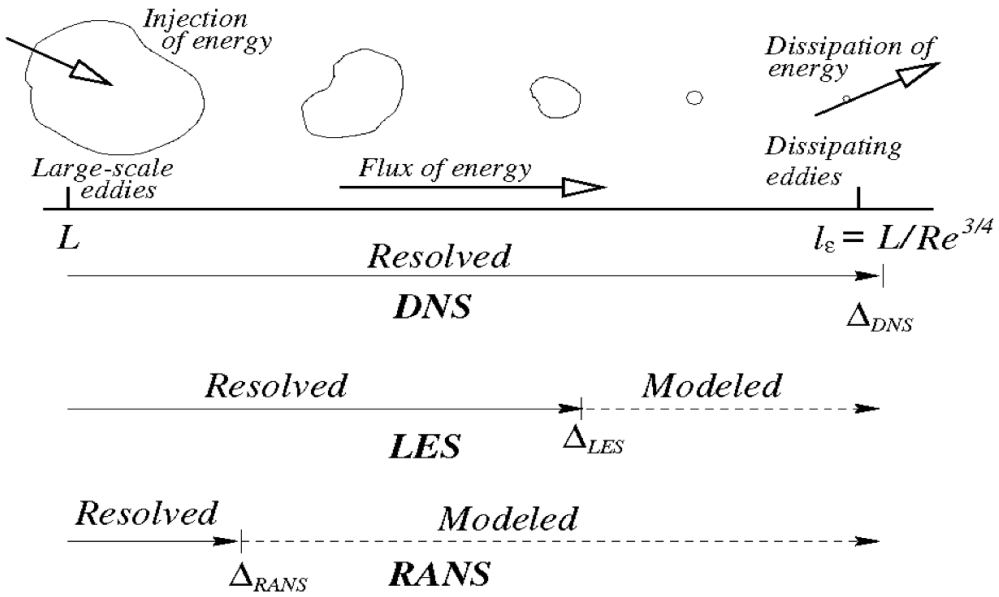


Figure 1.7.: *Sketch of the energy cascade in the frame of the Kolmogorov theory and generic numerical techniques for turbulent flows simulation.*

Table 1.1.: *Summary of strategies*

Method	Re-dependence	Empiricism	Grid Size	Grid Ready
U-RANS	Weak	Strong	10^7	1995
DES	Weak	Strong	10^8	2000
LES	Weak	Weak	$10^{11.5}$	2070
DNS	Strong	Weak	10^{16}	2080

1.7.1. Direct Numerical Simulation (DNS)

In DNS, all space and time scales of the flow are resolved, as shown in Figure 1.7. That involves extremely high resolutions associated to huge computer resources. In addition, the computation time has to be sufficient so that the fluid properties reach a statistical equilibrium. It is the most accurate approach, but so much resources are required that, when the complexity of the problem increases it becomes unaffordable. It is therefore limited to fundamental studies in simple computational domain and for low to moderate Reynolds numbers.

1.7.2. Reynolds Navier-Stokes Simulation (RANS)

In RANS the equations are averaged. It is relevant in many engineering applications where only average statistics of the flow are required. All variables are decomposed into a steady and fluctuating part such that, $\mathbf{u} = \mathbf{U} + \mathbf{u}^{'}$. The averaging process leads to a new set of equations called Reynolds-equations, written in a Cartesian coordinates system:

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 U_i}{\partial x_i \partial x_j} - \frac{\overline{u_i^{'} u_j^{'}}}{\partial x_j} \tag{1.47}$$

Reynolds equations are solved to determine the mean velocity field $\mathbf{U}$ and the turbulence is parameterized as shown in Figure 1.7. The averaging of the nonlinear terms leads to a new unknown, the Reynolds stresses tensor, $\tau_{ij} = \overline{u_i^{'} u_j^{'}}$. A closure has thus to be introduced to define these Reynolds stresses in terms on known averaged quantities. There exists a large literature on the topic proposing various models depending on the flow under consideration and the available numerical resources. The most popular models are based on either the Reynolds stress transport, by solving equations directly derived manipulating the Navier-Stokes equations, or on the Boussinesq hypothesis, which assumes a simple relationship between Reynolds stresses and velocity gradients through the eddy viscosity (similar to molecular viscosity). In this thesis, we follow the latter approach and the reader is referred to Hanjalic and Launder [38] for more details. The Boussinesq hypothesis, derived by analogy from the relation between the stress tensor and the rate-of-strain for an incompressible Newtonian fluid, writes here:

$$\tau_{ij} = \rho < u_i u_j > \tag{1.48}$$

where k is the turbulent kinetic energy and S_{ij} the mean rate of strain. Using this hypothesis, the closure problem comes down to find the turbulent viscosity ν_t (or eddy viscosity). This new viscosity is not related to physical property of the fluid but depends on the flow conditions. Several ways exist in the literature to model ν_t, based on dimensional arguments, physical behaviours (parameters based on experimental observations) and empirical parameters. In this thesis we have chosen the native Spalart Allmaras model of OpenFOAM. This model is a one-equation model that solves a modelled transport equation for a viscosity-like variable proportional to ν_t. The Spalart–Allmaras model was designed specifically for aerospace applications involving wall-bounded flows and has been shown to give good results for boundary layers subjected to adverse pressure gradients. Details on it will be provided on Chapter 3.

1.7.3. Large-Eddy Simulation (LES)

First proposed by Smagorinsky [120] for atmospheric studies, LES is based on the solution of the filtered Navier-Stokes equations, and consequently only resolves the largest energy containing scales, the small "universal" ones being modeled. In practice, the filtering is done most of the time by the mesh, which fixes the size of the eddies that will be resolved or modeled. That results in a significant reduction of the computational cost compared to DNS. The accuracy can be however better than in RANS, since the large eddies contain almost all the turbulent energy, and are responsible for most of the momentum transfer and turbulent mixing. The resulting field obtained is then 3D and unsteady. The filtering of Navier-Stokes equations leads to an extra term called residual-stress tensor that has to be modelled. It represents the impact of the unresolved velocity components on the resolved ones. The small scales being more isotropic and homogeneous than the large ones, the Boussinesq approximation (isotropic) provides a rather accurate hypothesis. A wide range of subgrid scales models exists as sysnthetized in the book of Pope [101] LES remains however time consuming for industrial flows, as shown in Smagorinsky [122] and reported on Table 1.1. The Reynolds number dependence refers to the number of grid points as function of the Reynolds number.

1.7.4. Detached-Eddy Simulation (DES)

In wall bounded flows, the computational cost of a LES becomes quickly unaffordable as the Reynolds number increases. The idea came to make hybrid models mixing RANS and LES in a single approach. In this kind of approach, the domain is ideally divided into two sub-domains: a RANS region, where a suitable RANS model is solved, typically near the boundary layer and a LES region where the LES equations are solved. Following this idea, Spalart et al. [126] introduced DES. In DES, a single model in the RANS and LES regions is used and in the LES region the RANS model itself is used as a SGS model. This is practically done by switching the turbulent viscosity which depends on the grid itself and the distance from the wall. In OpenFOAM, the minimum wall distance d in the SA–RANS model is modified by using a length scale C_{DES} depending on the grid cell dimension. The formula of DES length scale inside the turbulence model which controls the eddy viscosity reads :

$$\tilde{d} = min(d, C_{DES}\Delta) \tag{1.49}$$

$$\Delta = max(\Delta x, \Delta y, \Delta z) \tag{1.50}$$

with $C_{DES} = 0.65$ a constant of the model.

Others filter width can be used as $\Delta = (\Delta x \Delta y \Delta z)^{\frac{1}{3}}$ or $\Delta = (\Delta x^2 + \Delta y^2 + \Delta z^2)^{\frac{1}{2}}$

Inside a thin boundary layer the wall distance d is much smaller than the largest dimensions of the cell; and then the DES length scale is on the RANS equations $\tilde{d} = d$. Outside the boundary layer the distance d is much larger than the cell dimension and the LES model is active, $\tilde{d} = C_{DES}\Delta$. Being dependent on the grid size, this length scale enables the DES model to work like LES in separated flow regions, which are typically located away from the walls. Additional details will be provided in Chapter 3.

1.7.5. Boundary layer and wall model

Flow separation and reattachment are strongly dependent on a correct prediction of the development of turbulence near walls. Due to no-slip boundary condition, the turbulent eddy viscosity ν_t tends to zero in these flow regions. However, RANS models are usually based on assumptions relying on a high Reynolds number, requiring the viscosity-affected region of the boundary layer to be properly resolved ($y^+ \sim 1$ meshes, with $y^+ = \frac{y_{wall} * U_\tau}{\nu}$). Without such a resolution, it is known from DNS and experimental observations that these high-Re models predict the wrong behaviors near solid walls. Low-Re models cannot be obtained as a limit of high-Re models. As a first attempt to model these thin boundary layers (even if we know that this method provides satisfying results only in very simple near-wall flows as flate plate), we have considered a wall function approach to avoid the resolution of this viscous laminar sublayer. This approach allows us to apply a boundary condition at a sufficiently large distance away from the wall to be in the logarithmic layer ($y^+ > 30$), (see Ref [61]). With these laws it is possible to express the mean velocity parallel to the wall and turbulence quantities outside the viscous sublayer in terms of the distance to the wall and wall conditions such as wall shear stress. These relations based on asymptotic developments (see Ref [121]) reads :

$$u^+ \begin{cases} = y^+ & 0 \leq y^+ \leq 5 \\ = \frac{1}{\kappa} ln\ y^+ + C & 30 \leq y^+ \end{cases} \tag{1.51}$$

with :

- $u^+ = \frac{u_\parallel}{u_\tau}$ is the dimensionless velocity depending on the $u_\parallel$ parallel to the wall
- $u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ the friction velocity or shear velocity

- $\tau_w = \mu \left(\frac{\partial u_\parallel}{\partial y}\right)_{y=0}$ the wall shear stress
- ρ is the fluid density
- μ is the dynamic viscosity
- $\kappa = 0.41$ is the Von Kármán constant
- $C \simeq 5$ is a constant

Then, the wall functions can be used to provide near-wall boundary conditions for the momentum and turbulence transport equations, rather than conditions at the wall itself, so that the viscous sublayer does not have to be resolved and the need for a very fine mesh is reduced.

Nowadays a lot of wall functions exist in literature. Launder and Spalding [61] introduced the following wall function to model the near-wall region. This condition is computed directly inside the log-layer of the boundary layer (see Figure 1.8) and the calculation of the wall shear stress reads :

$$\tau_w = \begin{cases} \dfrac{\mu}{y_P} u_P^+ & y^+ \leq 11.3 \\ \dfrac{\rho c_\mu^{\frac{1}{4}} \kappa \sqrt{k_P}}{ln(E y_P^+)} u_P^+ & 11.3 \leq y^+ \end{cases} \tag{1.52}$$

with

- $y^+ = \rho c_\mu^{\frac{1}{4}} y_P \sqrt{k} \mu$
- c_μ and E constants
- k_p the turbulent kynetic energy

1.8. Towards control method

The control of bluff body flows remains a large challenge in different fields of industry. In VIV phenomenon for example, the control of vibrations can be done through the control of vortex shedding, which leads to a reduction of the forces acting on the body and of theirs frequency. The controlled flow can be done by momentum transfer, in boundary layer or wake, which affects the wake dynamics.

Depending on the energy input to activate the control process, two kinds of method exist:

- Passive when no external energy is added to the system. The control is done through changes in the geometry. Some geometrical modifications are done to suppress the vortex shedding, including end-plates and helical strakes as shown on Figure 1.9 and described in Ref [8] or small control cylinders strategically located in the wake of the main cylinder as mentioned in Strykowski and Sreenivasan [130] ,[19]. It can also be done by poroelastic actuators, inspired by birds feathers, to control boundary layer separation o a wing (Ref [26] , Ref [25]), compliant walls inspired by Dolphin's skin to control turbulent shear stress (Ref [43]), or undulated leading-edge inspired by Humpback whales to control boundary separation and delay stall on a wing (Ref [119])

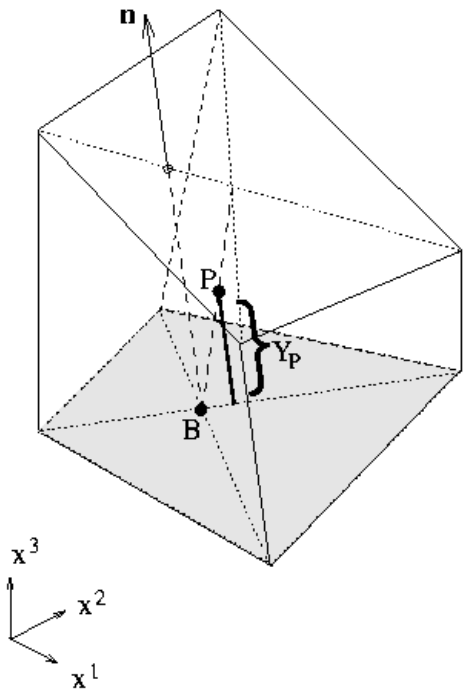


Figure 1.8.: *Point inside the log layer.*

- Active otherwise, with both closed and open loop. This control technique has been intensively studied with several control features as shown on Figure 1.10. Other control techniques exist as acoustic excitations studied by Blevins [6] , temperature changes as studied in Ref [62] ,[145], and moving surface boundary layer control (MSBC) in Ref [59] or in Ref [94] , where the control is done by the momentum injection into the flow field near the body by small rotating cylinders.

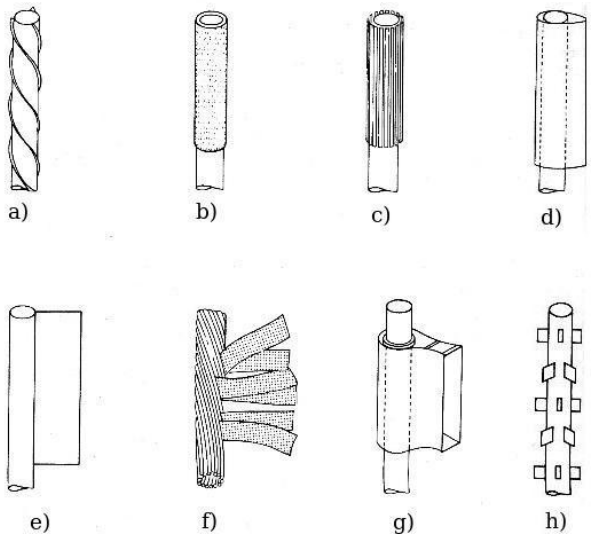


Figure 1.9.: *Passive control devices with a) helical strake, b) shroud, c) axial slats, d) streamlined fairing, e) splitter, f) ribboned cable, g) pivoted guiding van, h) spoiler plates.*

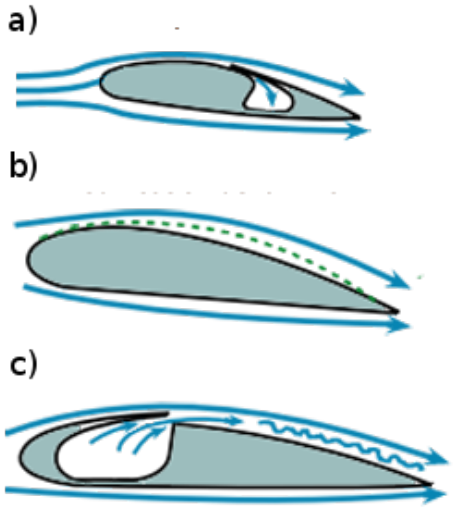


Figure 1.10.: *Active control devices with a) Suction features, b) moving wall features, c) blowing features.*

The use of such techniques require the optimization of many parameters as the cylinder diameter, the location of the actuator, the blowing intensity. Find an optimum from empirical results for all these variables can be really expensive, because each parameter variation would require a simulation or an experiment. This motivated the work of Hill [41] on the sensitivity theory. This theory consists in evaluating the effects of an added force on the primary transition of the wake with much less simulations: base flow, stability, and adjoint problem in order in this case to analyze the stabilization of wake flows. Sensitivity analysis to steady force has been used then by Giannetti and Luchini [34] , Marquet et al. [71] or Pralits et al. [103] . The concept of wavemaker (or sensitivity to flow modifications) has been introduced by Giannetti and Luchini in the wake of a circular cylinder, and defined regions where instabilities can be amplified and regions where flow control can be achieved. Marquet et al. [71] achieved the theoretical formulation to assess the effect of a steady force on the base flow and on the unstable mode in order to find the change on the growth and frequency of the instability. They have shown that putting a small control structure modifies the flow stability by inducing a structural modification of the equations at the perturbation level and modifies the base flow on which the perturbations evolve which modify the all dynamics. They compared their prediction of a circular cylinder wake control using a small cylinder to the experimental observations of Strykowski and Sreenivasan [130] . In the frame work of the wake of a fixed circular cylinder (see Ref [94]) did simulations in the presence of small rotating to achieve passive control following the work of Marquet et al. .

Later on Pralits et al. [103] have extended the sensitivity analysis to the wake of moving objects with a rotating circular cylinder in order to find different regions of instability growth due to the modified base flow around the rotating main cylinder in order to achieve passive control.

1.9. Objective and outline of this thesis

The ultimate objective of the thesis is to propose an efficient numerical solver to predict and control turbulent flows around bluff body in versatile industrial configurations, with a coupling between the fluid flow and the solid motion. This requires to design an efficient solver for the incompressible Navier-Stokes equations with the

following main properties:

- handle laminar and turbulent flow regimes in versatile configurations,
- couple the fluid flow and the solid motion,
- perform sensibility analysis with the perspective of developing new control strategies,
- run on massive parallel machine to reach high resolutions and long computational times for steady state,

To reach this objective, the improved direct forcing approach of Pinelli et al. [99] has been incorporated, then carefully verified and validated, into the PISO solver of the open source toolbox OpenFOAM (Chapter 2). The method has been firstly extended to simulate the coupling of the fluid flow with the solid motion (Chapter 3), then afterwards to turbulent flows simulations in the frame of OpenFOAM native DES models (Chapter 4). Towards flow control, an adjoint solver has been developed to perform sensibility analysis around a cylinder (Chapter 5).

Chapter 2

Development of a new IBM PISO flow solver in OpenFOAM

In this chapter we propose a modified PISO algorithm integrating the efficient Immersed Boundary Method (IBM) of Ref [99] in the open source toolbox OpenFOAM. A rigorous characterization of the IBM is proposed, using an original verification technique, which allows to estimate precisely the numerical errors at various stages of the algorithm. The chapter also provides a thorough validation of the solver on relevant literature test-cases.

2.1. Flow configuration

The computational domain is made of a box of height H, width W and length $L_i + L_0$. In our simulation, a body is embedded within the domain by IBM, as exemplified in Figure 2.1 for the case of a simple cylinder.

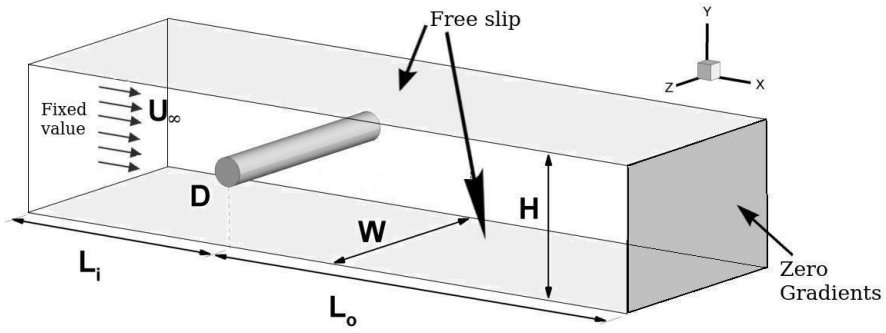


Figure 2.1.: *Computational domain. Example of a flow configuration around a cylinder of diameter D.*

Boundary conditions: At the inlet, a steady uniform velocity is imposed along the streamwise direction x together with a zero pressure gradient. A mass conservation condition is imposed at the outlet. We assume periodic conditions in the spanwise direction z. Free-slip boundary conditions on the velocity are applied at the top and bottom of the domain. Note that no-slip boundary conditions are not required at the body wall which is modelled by the IBM.

2.2. The numerical discretization

The governing Eqs. 1.6, 1.5 are discretized into *OpenFOAM* using a standard finite volume integration with a gaussian quadrature. Gaussian integration is based on summing values on cell faces, which must be interpolated from cell centres (see Sec. 1.4.1). The interpolation schemes are described in Sec. 1.4.1. The meshes are fixed and structured, and composed of hexaedral elements [49].

The time discretization is based on the implicit Backward Euler scheme which has been chosen for its simplicity. However, it could be staithforwardly extended to more accurate second-order *OpenFOAM* schemes, as the IBM implementation does not depend on the time discretization scheme.

The velocity-pressure coupling is solved by the built-in solver *pisoFoam*. As we don't want to add some diffusive term which will lead to a smoothed description of the IB forcing and then of a more diffuse boundary, we decide to remove the correction term ϕ_2^n from the OpenFoam PISO Solver (describes Sec. 1.4.2.2) since no diffusion term is needed. As a consequence, it is the general PISO solver describe in Ref [28] which is used.
The solver is thus the classical Pressure Implicit with Splitting of Operators (PISO) algorithm described in the paper by Ferziger and Peric [28] . Three and one iterations were set for a PISO loop and for non orthogonal corrections, respectively (see in Ref [18]).

Linear algebraic systems are solved using the Diagonal Incomplete LU Preconditioned Biconjugate Gradient *DILUPBG* (for the momentum equation (Eq. 1.6)) and the Diagonal Incomplete Cholesky Preconditioned Conjugate Gradient *DICPCG* (for the Poisson equation (Eq. 1.20)). For the present simulations, involving low to moderate Reynolds numbers, and regular structured meshes, no preconditionning was needed. This

conjugate gradient method without preconditionnining provides a good compromise between stability, accuracy and numerical cost. For all independent variables, the required accuracy is 10^{-7} at each time step.

Note - This algorithm is not optimized for the structured meshes, and could be similarly used for unstructured meshes. For structured meshes used on this thesis (even stretched in one or two directions), the geometric-algebraic multi-grid (GAMG) proposed in OpenFOAM is much more efficient. However the mesh used here is pseudo-structured as unstructured elements is used between the structured blocks to ensure a better continuity of the solution, as shown on the example in Figure 2.2. Then in order to focus on convergence issues related to the IBM based on structured meshes, and to avoid other possible numerical issues we decide to use an algorithm adapted for unstructured grids in this section. Nevertheless the algorithm optimized for structured meshes as GAMG work similarly with the IBM as it has been used in the last chapter (5)

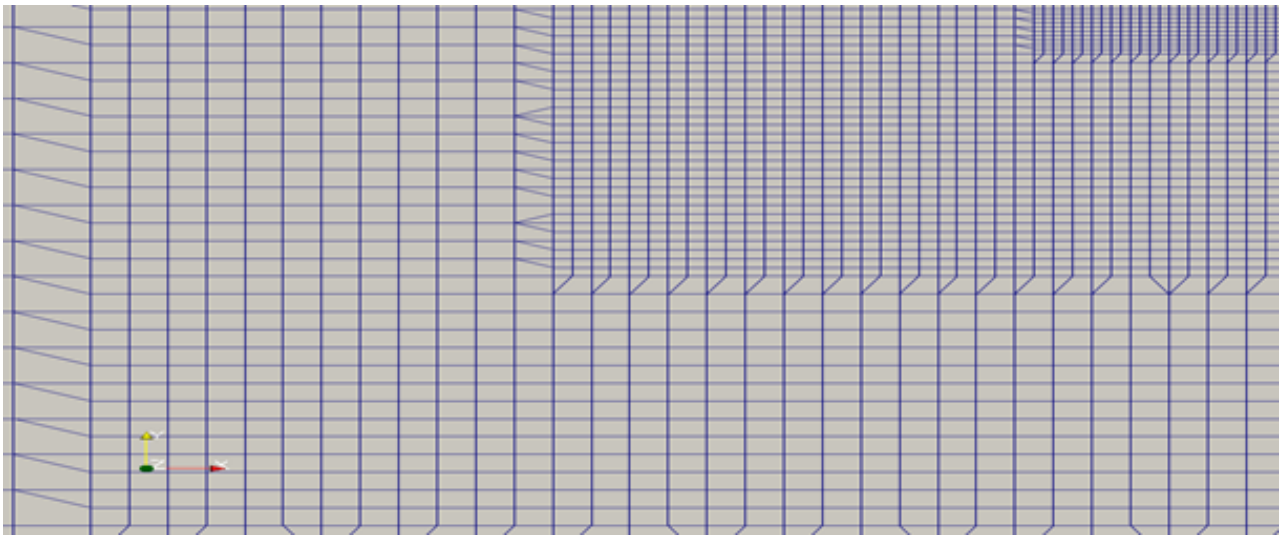


Figure 2.2.: *Example of 2D mesh. Structured blocks are connected by unstructured elements to ensure a better continuity of the solution.*

2.2.1. The modified PISO algorithm

The predictor-corrector solver *pisoFOAM* has been modified to integrate the direct forcing approach proposed by Pinelli et al. [99] . The Navier-Stokes equations are discretized on a fixed mesh (Eulerian) while the solid boundary is discretized by a set of Lagrangian markers free to move over the Eulerian mesh, depending on the motion of the solid.

The numerical issue is here to satisfy the two-constraints problem formed by no-slip and divergence-free conditions. The new following 3 steps procedure is proposed at each time step n :

1. **Predictor step:**
   a) An estimate velocity $\widehat{\mathbf{u}}$ is obtained by solving the momentum Navier–Stokes equations without any force term, and using the pressure p computed at the previous time step $n-1$:

$$\frac{\partial \widehat{\mathbf{u}}}{\partial t} + \nabla \cdot (\widehat{\mathbf{u}}\widehat{\mathbf{u}}) = -\nabla p + \frac{1}{Re}\nabla^2 \widehat{\mathbf{u}} \tag{2.1}$$

b) The calculation of the IBM force $\mathbf{F}_s$ is detailed in 2.2.2. It is calculated on the Lagrangian markers using $\widehat{\mathbf{u}}$ (Eq. 2.7), and its values are spread on the Eulerian mesh to calculate $\mathbf{f}$ (Eq. 2.20).

c) A new velocity $\mathbf{u}^{\star,1}$ is calculated from the Navier-Stokes equations accounting now the immersed boundary force term f:

$$\frac{\partial \mathbf{u}^{\star,1}}{\partial t} + \nabla \cdot (\mathbf{u}^{\star,1}\mathbf{u}^{\star,1}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u}^{\star,1} + f(\widehat{\mathbf{u}}) \tag{2.2}$$

$\mathbf{u}^{\star,1}$ is the guess value of the velocity in the iterative PISO loop

2. **PISO loop:**
For the sub-iteration $m = 1$ to $M - 1$, and up to convergence:

a) At each sub-iteration, a pressure field $p^{\star,m}$ is calculated from the following Poisson equation :

$$\nabla^2 p^{\star,m} = -\nabla \cdot (\mathbf{u}^{\star,m}\nabla \mathbf{u}^{\star,m}) + \nabla \cdot f(\widehat{\mathbf{u}}) \tag{2.3}$$

b) The velocity field is thus corrected using:

$$\mathbf{u}^{\star,m+1} = g\left(\mathbf{u}^{\star,m},\ \nabla p^{\star,m},\ f(\widehat{\mathbf{u}})\right) \tag{2.4}$$

where g as well as all discretized operators used in the algorithm are defined in Annex B.

3. **Final step:**
The velocity and the pressure are finaly updated at time $n + 1$:

$$\mathbf{u}^{n+1} = \mathbf{u}^{\star,M-1} \tag{2.5}$$
$$\mathbf{p}^{n+1} = \mathbf{p}^{\star,M-1} \tag{2.6}$$

2.2.2. Calculation of the IBM body forces

As in classical direct forcing methods, the target velocity U^d is directly imposed at the boundary nodes. This velocity is equal to the local fluid velocity.

2.2.2.1. Calculation of the body force term F on the Lagrangian markers: the interpolation step

The body force is computed into the Lagrangian space, i.e. at all Lagrangian markers. On the s^{th} Lagrangian marker, and at time step $(n + 1)$, the force term $\mathbf{F}_s^{n+1}$, is given by:

$$\mathbf{F}_s^{n+1} = \frac{\mathbf{U}_s^d - \mathcal{I}[\hat{\mathbf{u}}]_s}{\Delta t} \tag{2.7}$$

where $\mathbf{U}_s^d$ is the target velocity to be imposed on the s^{th} Lagrangian marker. $\mathcal{I}[\hat{\mathbf{u}}]_s$ stands for the interpolation on the s^{th} Lagrangian marker of the fluid velocity known on the Eulerian mesh at time step n, and computed without any force term (2.1). As presented in Li et al. [65] , the discrete expression of the interpolation operator is given by :

$$\mathcal{I}[\mathbf{u}^n]_s = \sum_{j \in D_s} \mathbf{u}_j^n \delta_h(\mathbf{x}_j - \mathbf{X}_s)\Delta v \tag{2.8}$$

where the j-index refers to the discrete value of the fluid velocity on the Eulerian mesh, $\mathbf{X}_s$ refers to the coordinates of the s^{th} Lagrangian marker and Δv refers formally to an Eulerian quadrature, i.e. $\Delta v = \Delta x \Delta y \Delta z$ for the case of a Cartesian uniform mesh. The interpolation kernel is the discretized delta-function δ_h used in Roma et al. [110] :

$$\delta_h(r) \begin{cases} \frac{1}{3}\left(1 + \sqrt{-3r^2+1}\right) & 0 \leq r \leq 0.5 \\ \frac{1}{6}\left[5 - 3r - \sqrt{-3(1-r)^2+1}\right] & 0.5 \leq r \leq 1.5 \\ 0 & \text{otherwise} \end{cases} \tag{2.9}$$

with $r = (\mathbf{x}_j - \mathbf{X}_k)/h$. The delta-function δ_h satifies the following discrete properties:

1. $\delta_h(r)$ is continuous, $\forall r \in \mathbb{R}$
2. $\sum_{j \in D_j} \delta_h(r) = 1$
3. $\sum_{j \in D_j} r\delta_h(r) = 0$

It is centered on each Lagrangian marker s and takes non-zero values inside a finite domain D_s, called the support of the s^{th} Lagrangian marker. The variable h of the supporting box is defined in order to insure at least 3 points in the supporting box.

The delta function choice has been made regarding the avantages of the Roma et al. [110] function compare to others described in Pepona [95] . It is built on three points, which ensures a sharp description of the forcing thanks to the conditions. A wider extent of the forcing over the Eulerian mesh has to be avoided as it would increase the smoothing error due to IBM in the near wall region, which is an important issue for turbulent flows. It is also continuous, which will allow us to derivate it easily for the needs of the PISO algorithm (see section 2.2).

Non uniform Cartesian meshes

In order to extend the method to non uniform Cartesian meshes, we follow the approach described in Liu et al. [66] and used in Pinelli et al. [99] . The delta-function δ_h is modified by introducing a polynomial function, in order to take in account the unequal spreading of the Eulerian points inside the Lagrangian supporting box D_s.

The new function $\tilde{\delta}_h$ reads :

$$\tilde{\delta}_h(r) = \sum_{l=0}^{L} b_l(r,h)(\mathbf{x}_j - \mathbf{X}_k)^l \delta_h(\mathbf{x}_j - \mathbf{X}_k) \tag{2.10}$$

with L the number of polynomial coefficients $b_l(r,h)$ obtained by imposing the following properties to $\tilde{\delta}_h$:

$$\int_\Omega \tilde{\delta}_h(r)d\Omega = 1 \tag{2.11}$$

$$\int_\Omega r^l \tilde{\delta}_h(r)d\Omega = 0 \qquad (l = 1, ..., L) \tag{2.12}$$

These properties are the continous equivalent of the discrete ones defined for the regular delta-function δ_h Eq. 2.9 (see Pinelli et al. [99]).

The $b_l(r, h)$ polynomial coefficients are determined by solving the following system of equation :

$$\int_\Omega r^l \tilde{\delta}_h(r)d\Omega = \sum_{l=0}^{L} b_l(r, h)m_l(r) = T \tag{2.13}$$

with $T = \{1; 0...0\}$, and $m_l(r)$ is defined as

$$m_l(r) = \int_\Omega r^l \delta_h(r)d\Omega = T \tag{2.14}$$

We chose here a $2^n d$-order polynomial function, so $L = 2$. Then, in 2D (the extension to 3D is straightfoward), $\tilde{\delta}_h$ reads n a Cartesian frame:

$$\begin{aligned}\tilde{\delta}_h(x_j - X_k, y_j - Y_k) = [b_0 + b_1(x_j - X_k) + b_2(y_j - Y_k) + b_3(x_j - X_k)(y_j - Y_k) \\ + b_4(x_j - X_k)^2 + b_5(y_j - Y_k)^2]\delta_h(x_j - X_k, y_j - Y_k)\end{aligned} \tag{2.15}$$

with

$$\delta_h(x_j - X_k, y_j - Y_k) = \delta_h(x_j - X_k)\delta_h(y_j - Y_k) \tag{2.16}$$

Rewriting the system (2.13) as a linear problem, we get:

$$[m_{l,a}]\mathbf{b} = \mathbf{e}_0 \tag{2.17}$$

where

$$m_{l,a} = \sum_{j \in D_s} (x_j - X_k)^l (y_j - Y_k)^a \delta_h(x_j - X_k)\delta_h(y_j - Y_k)\Delta A_j \tag{2.18}$$

with e_0 is the first component of the identity vector of size 6, and ΔA_j defines the area of the j^{th} Eulerian cell.

Due to low values of the original window function on some points, and in order to prevent the matrix to be singular, we solved the equivalent system:

$$\mathbf{HMH}^{-1} = \mathbf{e_0} \tag{2.19}$$

with $H = diag(1, \frac{1}{h_x}, \frac{1}{h_y}, \frac{1}{h_x h_y}, \frac{1}{h_x^2}, \frac{1}{h_y^2})$.

The system is solved in two steps with $\mathbf{HMc} = \mathbf{e}_0$ and $\mathbf{Hc} = \mathbf{b}$.

Once the linear system 2.19 is solved, the polynomial coefficients are known and thus a new window function deltatilde is built, on which the reproducing properties of the Eq. 2.9 have been enforced at the discrete level. In the case of non uniform meshes this process can be seen as a correction of the original Roma function 2.9 to

take into account the non uniformity of the mesh.

In the following, the delta-function $\tilde{\delta}_h$ will be used instead of δ_h.

2.2.2.2. Calculation of the body force f on the Eulerian mesh: the spreading step

Once the force term is computed from Eq. 2.7, one needs to transfer its value to the Eulerian mesh. This is done by the spreading step, which is the inverse operation of the interpolation. The value of the force term evaluated on the Eulerian mesh, $\mathbf{f}^{n+1}(\mathbf{x}_j)$, is given by:

$$\mathcal{S}[\mathbf{F}_k^{n+1}] = \mathbf{f}^{n+1}(\mathbf{x}_j) = \sum_{k \in D_j} \mathbf{F}_k^{n+1} \delta_h(\mathbf{x}_j - \mathbf{X}_k)\boldsymbol{\epsilon}_k \tag{2.20}$$

The k-index refers to a loop over the Lagrangian markers whose support contains the Eulerian node j. $\boldsymbol{\epsilon}_k$ is the Lagrangian quadrature, which is calculated solving a linear system :

$$A\boldsymbol{\epsilon} = \mathbf{1} \tag{2.21}$$

where the vectors $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_{N_s})^T$ and $\mathbf{1} = (1, \ldots, 1)^T$ have a dimension of N_s, N_s being the number of Lagrangian markers, and A is the matrix defined by the product between the k^{th} and the m^{th} interpolation kernels such that:

$$A_{km} = \sum_{j \in D_m} \delta_h(\mathbf{x}_j - \mathbf{X}_k)\delta_h(\mathbf{x}_j - \mathbf{X}_m) \tag{2.22}$$

2.2.3. IBM improvement for the divergence free condition

The discretization of the boundary of the structure leads to some errors due to the smoothness of the body force term (ideally discretized over 3 markers). Thus the the PISO solver requires the computation of the divergence of the forcing term (Eq. 2.3). As illustrated below the forcing term, really sharp and then almost singular lead to non-negligeable errors on the divergence estimation with classical derivative estimation as the 2^{nd} order central scheme.

The Figure 2.3 shows the IBM forcing distribution around one Lagrangian marker in the Eulerian space computed analytically $\mathbf{f}^a(\mathbf{x})$, and discretized on the Eulerian mesh $\mathbf{f}(\mathbf{x_j})$ with :

$$\mathbf{f}^a(\mathbf{x}) = \sum_{k \in D_j} \mathbf{F}_k \delta_h(\mathbf{x} - \mathbf{X}_k)\boldsymbol{\epsilon}_k \tag{2.23}$$

$$\mathbf{f}(\mathbf{x_j}) = \sum_{k \in D_j} \mathbf{F}_k \delta_h(\mathbf{x}_j - \mathbf{X}_k)\boldsymbol{\epsilon}_k \tag{2.24}$$

Figure 2.4 shows the divergence of the IBM forcing term around one Lagrangian marker in the Eulerian space computed analytically $\nabla \cdot \mathbf{f}^a(\mathbf{x})$, discretized using a central scheme on the Eulerian mesh $\nabla \cdot \mathbf{f}(\mathbf{x_j})|_{central}$, and using the value obtained by the analytical derivation $\nabla \cdot \mathbf{f}(\mathbf{x_j})|_{new}$ with :

$$\nabla \cdot \mathbf{f}^a(\mathbf{x}) = \nabla \cdot \Big(\sum_{k \in D_j} \mathbf{F}_k \delta_h(\mathbf{x} - \mathbf{X}_k)\boldsymbol{\epsilon}_k \Big) \tag{2.25}$$

$$\nabla \cdot \mathbf{f}(\mathbf{x_j})|_{central} = \sum_{l=1}^{3} \frac{F_{j+1,l} - F_{j-1,l}}{2\Delta \mathbf{x}_{j,l}} \tag{2.26}$$

$$\nabla \cdot \mathbf{f}(\mathbf{x_j})|_{new} = \nabla \cdot \mathbf{f}^a(\mathbf{x_j}) \tag{2.27}$$

where l refer to the space dimension Note that we assume in the derivation that the value F_k and $\boldsymbol{\epsilon}_k$ are constant on the all k^{th} supporting box leading to :

$$\nabla \cdot \mathbf{f}^a(\mathbf{x}) = \nabla \cdot (\sum_{k \in D_j} \mathbf{F}_k \delta_h(\mathbf{x} - \mathbf{X}_k \boldsymbol{\epsilon}_k)) = \sum_{k \in D_j} \mathbf{F}_k \boldsymbol{\epsilon}_k \nabla \cdot \delta_h(\mathbf{x} - \mathbf{X}_k) \tag{2.28}$$

As we can see Figure 2.4 the error induced by the calculation of the derivative is significant and the pseudo-analytical derivative improves the results. Another solution to improve the results would be to enlarge the stencil by adding points in the derivation and interpolation of the force term. This solution would however lead to a more diffuse, and thus less accurate definition of the boundary.

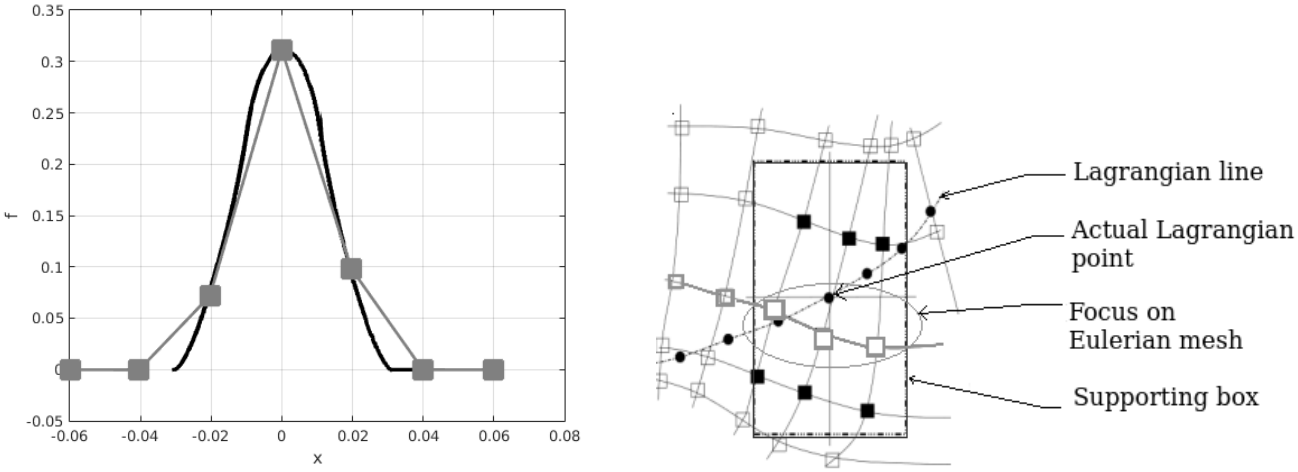


Figure 2.3.: *Eulerian discretization of the IBM force term f: comparison between discretized forcing term $\mathbf{f}(\mathbf{x_j})$ (grey squares) and the analytical forcing term $\mathbf{f}^a(\mathbf{x})$ (black line) (left). Zoom on the mesh discretization (right).*

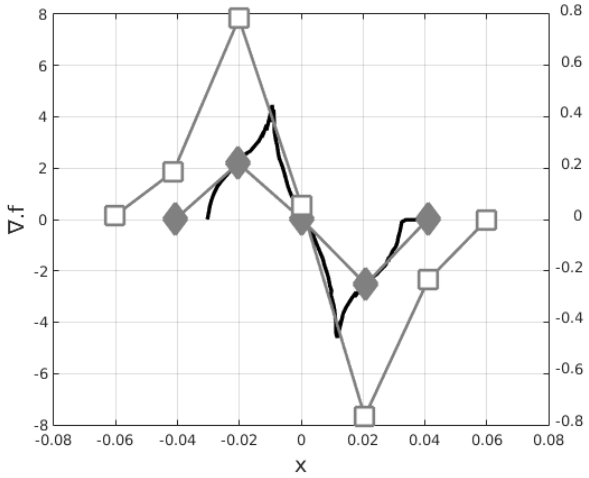


Figure 2.4.: *Divergence of the force term f using two interpolation schemes: 2^{nd}-order centered scheme $\nabla \cdot \mathbf{f}(\mathbf{x_j})|_{central}$ (empty grey symbols using a 10 times larger scale for the y-axis), the new derivative calculated with the kernel function $\nabla \cdot \mathbf{f}(\mathbf{x_j})|_{new}$ (full grey symbols) and the theoretical value $\nabla \cdot \mathbf{f}^a(\mathbf{x})$ (black line).*

Numerical results in Figure 2.5 show the efficiency of this correction on the derivative, which can provide an error on the mass conservation per cell which is even smaller than the one obtained with a body fitted mesh (by a factor of 15%).

A part of residual error on the mass for the IBM come from the fact that we use a non-monolithic approach that satisfies no-slip condition and the incompressibility within the same system of equations. Such an approach may require a large number of iterations to achieve convergence, hence higher CPU and memory costs. Here, both constraints are treated separately and independently, i.e. there is no back and forth between the two. This approach could lid to some errors on the divergence as those seen Figure 2.5.

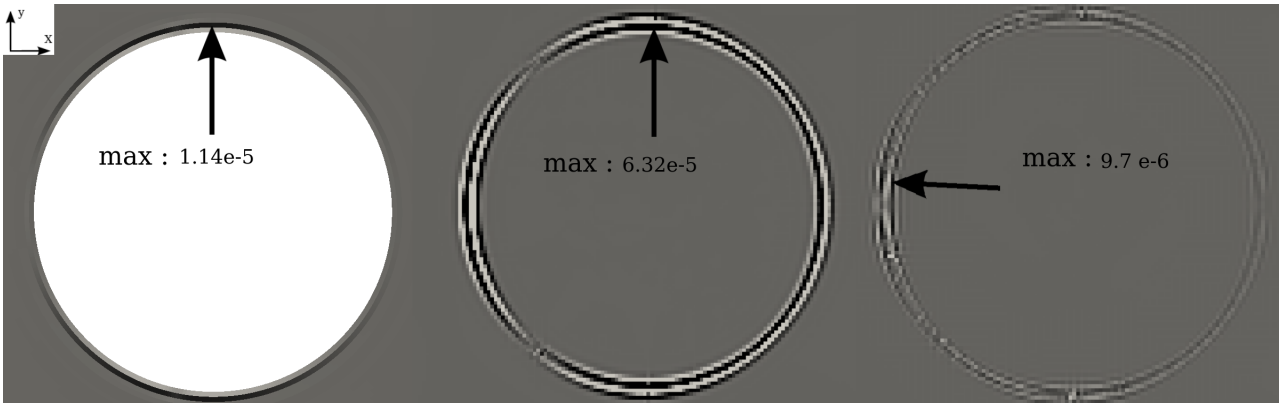


Figure 2.5.: *Error on the mass conservation defined by $\rho_{ij} div(u_{ij}) S_{ij}$, with S_{ij} the area of the 2D cell (i, j). Classical no-slip boundary condition (left), IBM calculations without (center) and with (right) correction of the derivative. The divergence is calculated at the end of the PISO loop. The error magnitude varies from -10^{-5} (black) to 10^{-5} (white). Flow around a 2D cylinder at $Re = 30$ using 312 Lagrangian markers*

Nevertheless this so-called partitioned approach provides satisfying results in our simulations, as shown below in Figure 2.6. The fluxes at the surface of cylinder (theoretically zero) are calculated here at the Lagrangian markers in the normal (Figure 2.6a), tangential (Figure 2.6b) and vertical (Figure 2.6c) directions, and their magnitude is satisfactorily low (with a flux about $3 \cdot 10^{-5}$ on the normal direction).

Besides the error on the body fitted mesh has been investigated as the mass conservation should be improved especially for a body fitted grid which is the object of next section.

2.2.4. Body fitted error investigation

To do so a body fitted lid-driven cavity flow has been perform in order to assess the error on the divergence of the velocity. The details of the calculation are shown Figure 2.7.

Figure 2.8 shows the divergence obtained on the velocity field computed on the cell centers. We can observe especially near the walls a large error of the divergence on four layers of cells. As mentioned in the paragraph (see 1.4.1), this type of error can be observed for a collocated finite volume method as the divergence free equation is imposed on the flux rather than the center of the cells. Moroever, the error depends on the error of the interpolation used in the simulation. As a linear first order interpolation is performed some errors could come from the problem of interpolation at the boundaries with a linear scheme.

To investigate this aspect in more details, we compute as well the error evaluated on the fluxes. The divergence shown in Figure 2.9 is computed directly from the fluxes using the Gauss theorem 1.9. The error corresponds in this Figure only to the error imposed for the solver algorithm which shows that it is only due to the interpolation error and finite volume discretization.

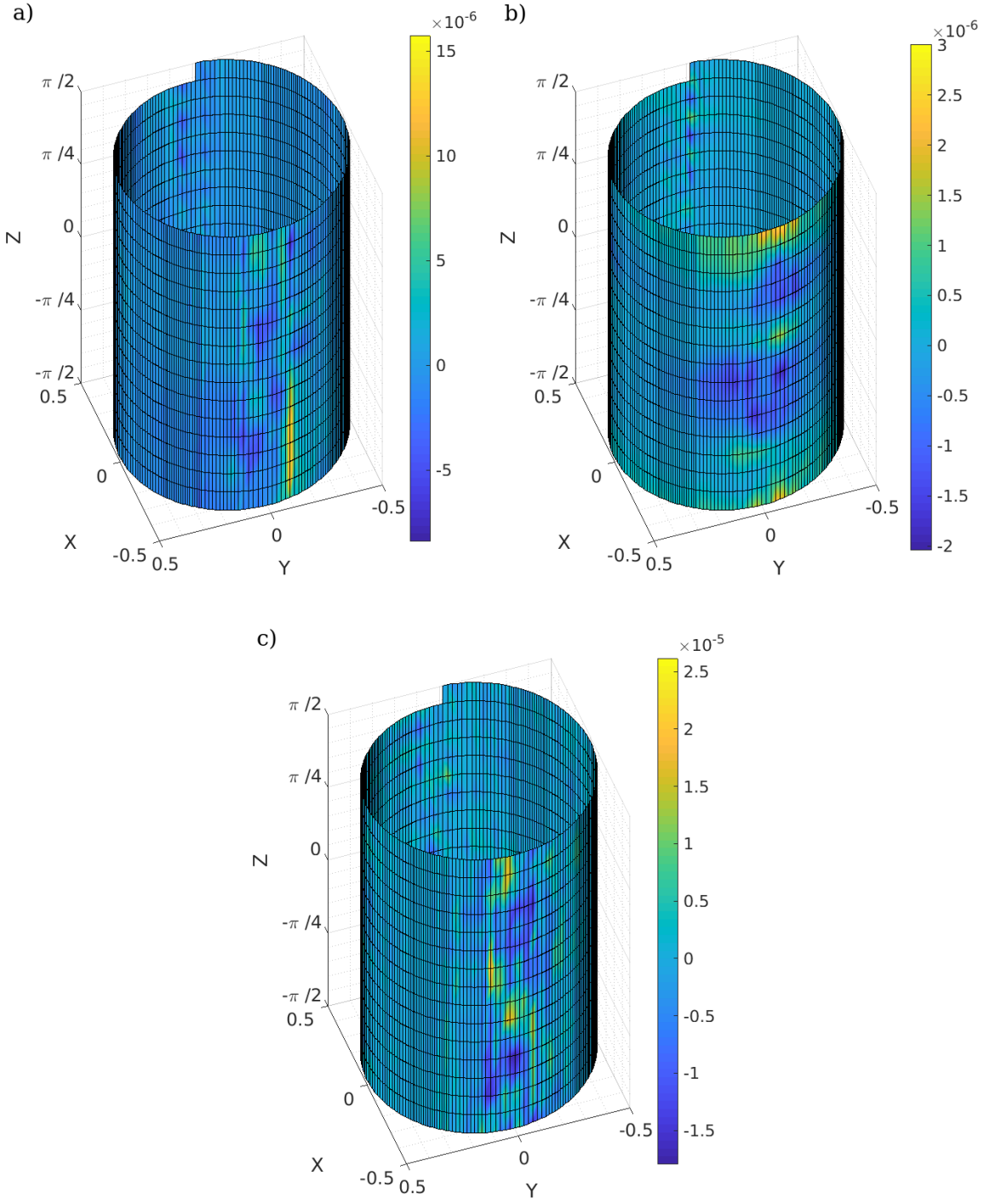


Figure 2.6.: *Fluxes at the cylinder calculated on the Lagrangian points in the normal (a), tangential (b) and vertical directions (c). Flow simulation at Re=3900. On z axis only 1/4 of the Lagrangian markers are shown for clarity.*

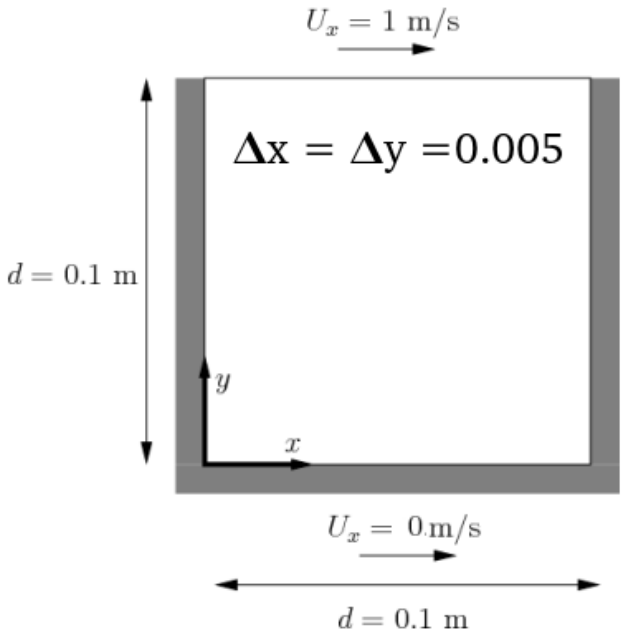


Figure 2.7.: *Lid driven cavity at $Re = 10$. Details of the computation.*

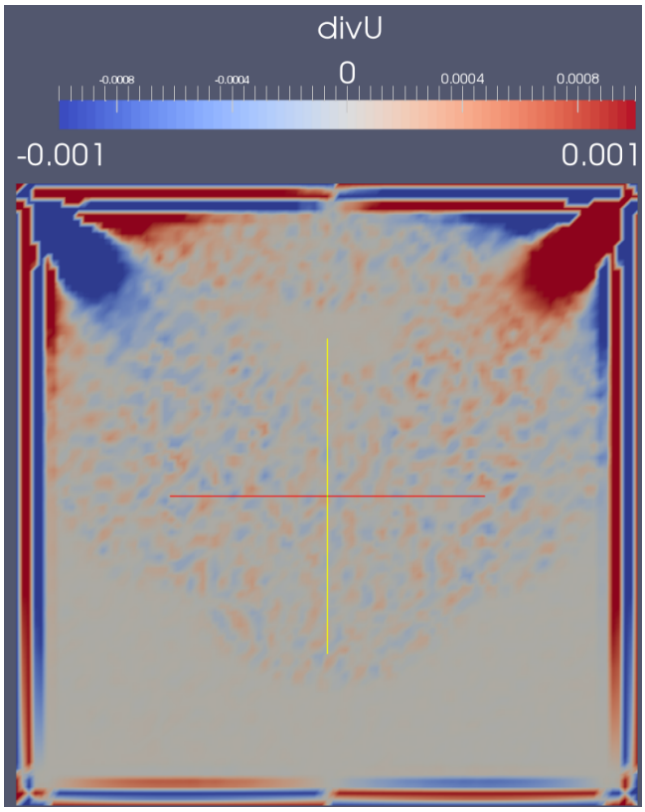


Figure 2.8.: *Lid driven cavity at $Re = 10$, Divergence of the velocity field calculated on the cell center of the cells.*

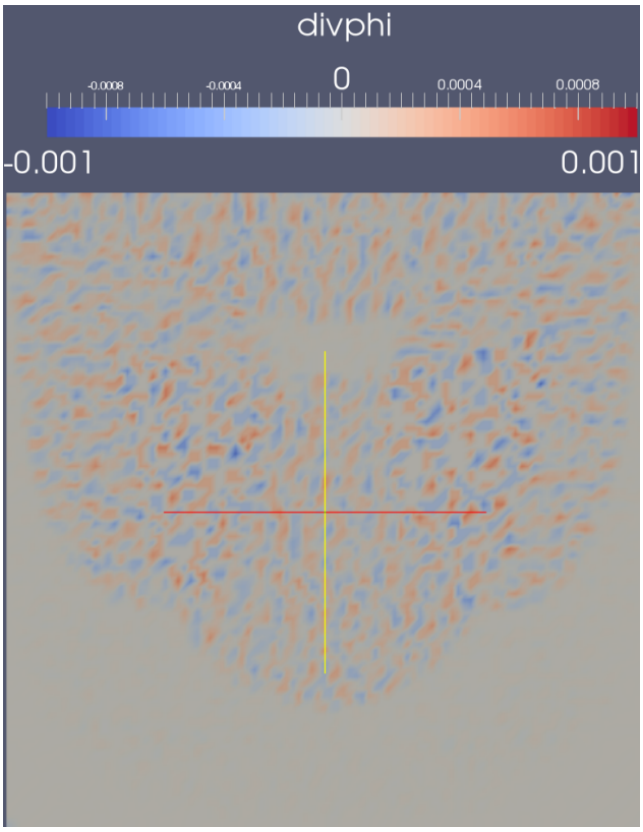


Figure 2.9.: *Lid driven cavity at $Re = 10$, Divergence of the velocity field calculated thanks to the gauss theorem 1.9 from the fluxes at the faces center.*

As a conclusion, the divergence on boddy fitted grids, i.e. without using IMB, is introducing as mass conservation error near the boundaries equivalent to IBM errors due to interpolation scheme error at the wall.

2.3. Solver verification

To evaluate quantitatively the order of accuracy of the IBM, a methodology based on a manufactured solution has been used, which allows to compare our numerical results to an analytical solution. A careful verification study is provided allowing to distinguish three different kinds of errors coming from the discretization and the IBM. The strength of this procedure is that it can identify any coding mistake that affects the order of accuracy of the numerical method. To do so we need to choose a function, which should be:

- composed of analytic functions like polynomials, trigonometric, or exponential functions as general as possible to describe the variable of our problem.
- sufficiently differentiable so that the differential operator needed make sense.
- physically relevant and in the range of the code applicability.

Thanks to the solution chosen, we calculate an analytic source term. Then we discretize the analytical solution and compute the resulting source term. We compare the error obtained between the two solutions (obtained with the 2 source terms) in order to deduce the order of convergence of the method.

This verification has been performed in 2D using a polynomial solution for the velocity and the pressure. Polynomial functions $f(x,y)$, $g(x,y)$ and $h(x,y)$ have been chosen ussing maple in order to get a divergence free velocity $\mathbf{u}_a(f(x,y);g(x,y))$ and a pressure $p_a(h(x,y))$ as function of the velocity such that:

$$\mathbf{u_a} = \begin{cases} f(x,y) = (1-0.01x^2)^2(1-0.03y^2)(1-0.01y^2) \\ \qquad -0.02(1-0.01x^2)^2(y-0.01)(y-0.01y^3) \\ g(x,y) = 0.5 + 0.04x(1-0.01x^2)(y-0.01y^3)(1-0.01y^2) \end{cases} \tag{2.29}$$

$$p_a = h(x,y) = f(x,y)g(x,y) \tag{2.30}$$

Different steps in the solver are verified according the definition of three errors:

- $e_{F_{IBM}}$ is the error on the estimate of the IBM force term (Eq. 2.7) (during Step 2 of the IBM/PISO solver) and integrated on the body, hence computed as:

$$e_{F_{IBM}} = \mid \sum_{k \in D_j} (\mathbf{F_k} - \mathbf{F_a})\boldsymbol{\epsilon}_k \mid \tag{2.31}$$

  where :

$$\mathbf{F}_a \quad = \quad \frac{\mathbf{U}_k^d - \mathbf{U_a}}{\Delta t} \tag{2.32}$$

  and $\mathbf{U_a}$ is the value of the analytical solution $\mathbf{u_a}$ (Eq. 2.29) evaluated on the Lagrangian markers.

- e_{noslip} is the error on the no-slip condition at the boundary of the obstacle. This error is evaluated during the calculation of the IBM force term on the Eulerian mesh (end of Step 2 of the IBM/PISO solver). It is

defined as the L_∞ norm of the difference between the velocity on one Lagrangian marker (Eq. 2.8), and the Eulerian velocity that has been spread and re-interpolated, i.e. :

$$e_{noslip} = \| U_s - \mathcal{I}[\mathcal{S}[U_k]]_s \|_\infty \tag{2.33}$$

- $e_{u_{tot}}$ is the error on the velocity at the end of the PISO loop (Step 6 of the IBM/PISO solver). It is calculated in terms of both L_2 and L_∞ norms:

$$e_{u_{tot}/L_2} = \| u - u_a \|_2 \tag{2.34}$$
$$e_{u_{tot}/L_\infty} = \| u - u_a \|_\infty \tag{2.35}$$

The verification is made in five steps summarized below:

1. Computation of $\mathbf{u}$ and p according to:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = \nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + S_a \tag{2.36}$$

where

$$S_a = \frac{\partial \mathbf{u_a}}{\partial t} + \nabla \cdot (\mathbf{u_a u_a}) - \nabla p_a - \frac{1}{Re}\nabla^2 \mathbf{u_a} \tag{2.37}$$

2. Computation on the Lagrangian markers of the analytical values of the IBM force term F_a using $\mathbf{u}_a$, and of the IBM force term F_s using the interpolated velocity u from the former step.
3. Calculation of $e_{F_{IBM}}$ and e_{noslip} using Eqs. (2.31) and (2.33).
4. Spreading of the residual force $F_s - F_a$ on the Eulerian mesh.
5. Execution of steps 3 to 6 of the PISO algorithm, and calculation of e_{u_{tot}/L_2} and e_{u_{tot}/L_∞}.

In order to quantify the errors $e_{F_{IBM}}$, e_{noslip} and $e_{u_{tot}}$ for different configurations, the simulations have been performed for different geometries :

- 2D flows past a circular cylinder (of diameter $L/5$)
- a square cylinder (of side $L/5$)

L being the size of the computational domain.

The simulation has been performed also for a circular cylinder with polynomial calibration of the window function (see Sec. 2.2.2.1) on the same uniform mesh in order to compare the interpolations with and without the polynomial calibration on the same mesh.

For a $(L \times L)$-domain, four uniform grids corresponding to five refinements have been tested: $\Delta x = \Delta y = 5 \times 10^{-2}L$, $3.3 \times 10^{-2}L$, $2.5 \times 10^{-2}L$, $1.25 \times 10^{-2}L$ and $5 \times 10^{-3}L$. To choose an interesting configuration to test the algorithm, the Lagrangian markers of the square cylinder is not aligned with the eulerian cells centers in order to avoid a trivial case. Results are shown on Figures 2.10, 2.11 and 2.12. All errors obviously decrease when the mesh is refined. Without RKPM method the error e_{noslip} exhibits a second-order rate of convergence whereas e_{u_{tot}/L_2} and e_{u_{tot}/L_∞} only exhibit a rate of convergence between 1 and 2 for both geometries. $e_{F_{IBM}}$ exhibits a rate of convergence that depends on the geometry (as could have been expected depending on points

repartition on the grid), namely 1 for the square cylinder and nearly 2 for the circular cylinder.

With the RKPM method the convergence error is between 1 and 2 for all the quantities but it is important to notice that the magnitude of the error is significantly lower. It can be explained by the fact that additional constraints are imposed Equation 2.13 to improve the reproduced capabilities of the window window function of Roma et al. . As a result the function obtained is sharper with the RKPM method. It is worth noticing that even if the interpolation and the spreading based on the RKPM method are improved, the overall error on the velocity at the end of the PISO loop doesn't change significantly as the accuracy of the flow solver is marginally impacted by the small error of interpolation generated by the IBM. Nevertheless, as the RKPM method leads to a sharper description of the boundary, the gradients are higher than with the delta function of Ref [110] , in particular, in the first iterations of a simulation, where the velocity gradients are large around the boundaries which can lead to unstable simulations. To overcome this issue, with the RKPM method, 2 strategies are used in the following:

- The calculations are initiated without RKPM during the first iterations,
- The supporting box is enlarged (in order to smooth the delta function and calculate the derivative properly on 5 points instead of 3 in each direction) to reduce the gradients at the beginning of the calculation.

Besides, as the improvement on the global accuracy of the flow solver is not significant, and as the cost of the method is important for a large number of Lagrangian markers due to the resolution of the linear system to compute the calibration coefficients, the RKPM will be used only when the mesh is non-uniform (see Chap. 3). The fact that e_{u_{tot}/L_2} and e_{u_{tot}/L_∞} are unsensible to the use of the RKPM method can show that, despite the fact they are widely used in the literature to evaluate the IBM accuracy, these errors mainly measure the error related to the solver (here the PISO solver of Openfoam) and other errors should be investigated in order to compare different methods.

In order to push further this analysis, a grid convergence study has been also carried out for the flow past a circular cylinder at $Re = 30$. Four grids have been used corresponding to $\Delta x = \Delta y = 8 \times 10^{-2} D, 4 \times 10^{-2} D$, $2 \times 10^{-2} D$ and $1 \times 10^{-2} D$ with D the diameter of the cylinder. The solution computed on the finest mesh is the reference solution. The error is estimated from the drag coefficient C_D defined in Eq. (2.39), Figure 2.13. The error descreases when the mesh is refined, with an order between 1 and 2 as shown previously using a manufactured solution.

2.4. Solver validation

The solver validation is performed using two- and three-dimensional (2D/3D) simulations of flows past a circular cylinder and a sphere of diameter D and at various Reynolds numbers ($Re = U_\infty D/\nu$) ranging between 30 and 300. Turbulent flows simulations will be presented in Chapter 3. Such flows are of great interest since they constitute a generic configuration for many applications in fields such as external aerodynamics, offshore engineering or environmental sciences. Vortex shedding is numerically challenging because the separation point on the surface of the cylinder is not fixed by the geometry. Since the pressure field changes rapidly near the separation and reatachment points, pressure prediction is therefore decisive for a correct estimation of the drag and lift coefficients. Numerically, in contrast to the square cylinder which can be tackled using Cartesian grids, these configurations require curvilinear body-fitted (or unstructured) grids. All these features makes these test cases excellent configurations to evaluate the precision and accuracy of the IBM solver we have developed.

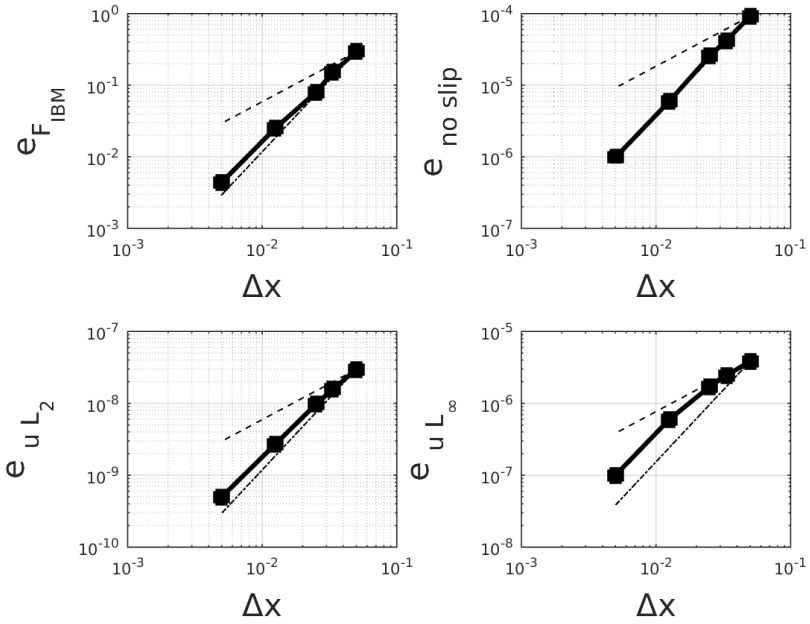


Figure 2.10.: *Log-log plots of $e_{F_{IBM}}$, e_{noslip} and $e_{u_{tot}}$ as a function of the mesh refinement. Flow past a circular (full thick line). The dashed black and grey lines show the slopes of order 1 and 2, respectively.*

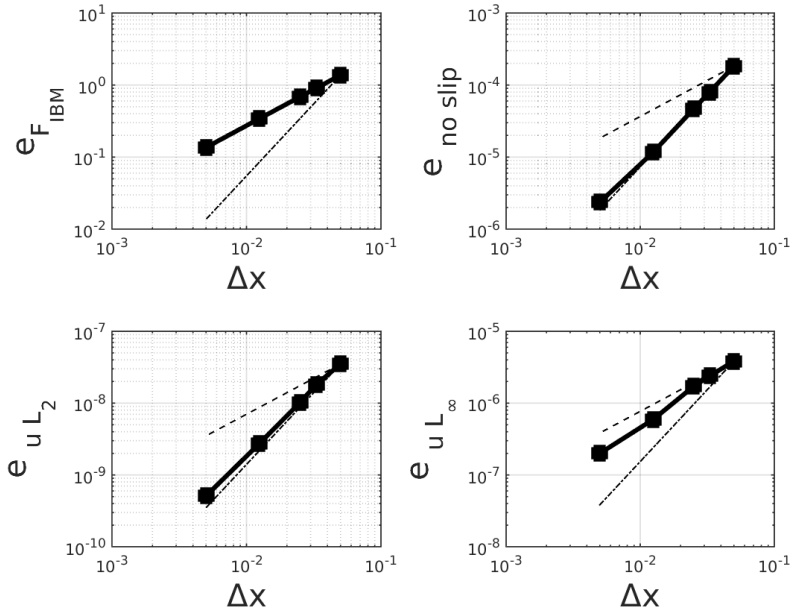


Figure 2.11.: *Log-log plots of $e_{F_{IBM}}$, e_{noslip} and $e_{u_{tot}}$ as a function of the mesh refinement. Flow past a square (full thick line) cylinder. The dashed black and grey lines show the slopes of order 1 and 2, respectively.*

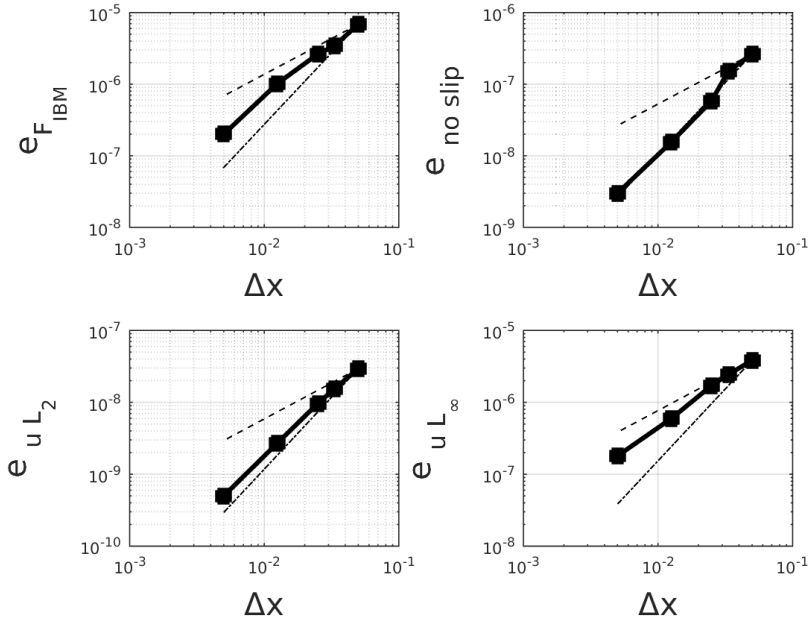


Figure 2.12.: *Log-log plots of $e_{F_{IBM}}$, e_{noslip} and $e_{u_{tot}}$ as a function of the mesh refinement using the RKPM method. Flows past a circular cylinder (full thick line). The dashed black and grey lines show the slopes of order 1 and 2, respectively.*

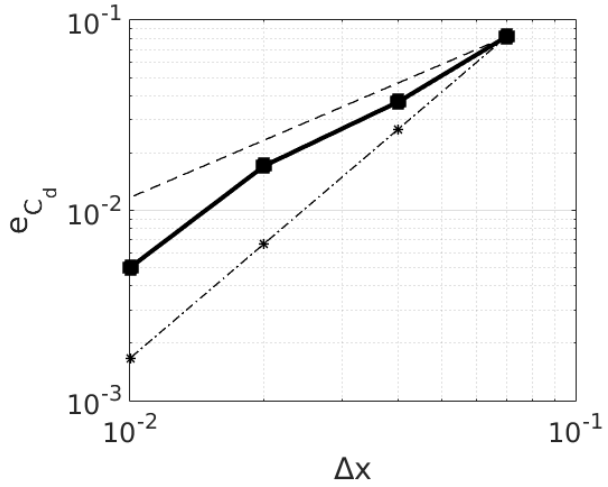


Figure 2.13.: *Log-log plot of the error on the drag coefficient computed for various mesh refinements comparing the value of the drag coefficient C_D to the reference value computed with the finest mesh. Flow past a circular cylinder at $Re = 30$. The dashed black and grey lines show the slopes of order 1 and 2, respectively.*

For the cylinder, the Strouhal number, drag and lift coefficients are defined by:

$$C_D = \frac{2F_d}{\rho u_\infty^2 D} \quad , \quad C_L = \frac{2F_l}{\rho u_\infty^2 D} \quad , \quad St = \frac{Df_v}{u_\infty}, \tag{2.38}$$

For the sphere, the drag, lift and side coefficients are defined by:

$$C_D = \frac{2F_d}{\frac{1}{2}\rho u_\infty^2 \pi \frac{D^2}{4}} \quad , \quad C_L = \frac{2F_l}{\frac{1}{2}\rho u_\infty^2 \pi \frac{D^2}{4}} \quad , \quad St = \frac{Df_v}{u_\infty}, \tag{2.39}$$

where f_v is the shedding frequency and F_d, F_l and F_s are the drag, the lift and the side force per unit length, respectively, computed by integrating the immersed boundary force term in the Lagrangian space.

2.4.1. Flows around a fixed cylinder

2D an 3D simulations have been performed for Reynolds number ranging from $Re = 30$ to 300. Solutions have been favorably compared to data of reference of the literature.

2.4.1.1. Numerical details

The center of the cylinder is the origin of the domain at $(0, 0)$.

The dimensions of the computational domain are those proposed by Pinelli et al. [99] and Vanella and Balaras [142] , respectively $[-16D, 48D] \times [-16D, 16D] \times [-5.12D, 5.12D]$ in the streamwise (x), vertical (y) and spanwise (z) directions (Figure 2.14).

The grid is uniform in the neighborhood of the cylinder, i.e. in the region $-D \leq x \leq D$ and $-D \leq y \leq D$. For 3D computations, the 2D mesh has been extruded in the spanwise direction. Details on the resolution, as well as the number of Lagrangian markers and their relative spacing with respect to the Eulerian mesh are given in Table 2.1. Outside this region, the mesh size is stretched with a factor of 2.0 on five grid levels in the (x, y)-plane (as shown in Figure 2.14).

Table 2.1.: *Mesh resolutions in the neighborhood of the cylinder: 2D cases 1 and 2* $[-D, D] \times [-D, D]$*, 3D case 3* $[-D, D] \times [-D, D] \times [-5.12D, 5.12D]$*. The* α *parameter defines the ratio of the distance between Lagrangian markers over the local Eulerian grid size Pinelli et al. [99] .*

Case	Resolution	Lagrangian markers	α
1	$\Delta x = \Delta y = 0.02D$	147	1.061
2	$\Delta x = \Delta y = 0.01D$	312	1.004
3	$\Delta x = \Delta y = 0.02D, \Delta z = 0.16D$	9792	1.004

All 2D and 3D simulations have been performed on 12 and 96 cpu of the AMU computing facilities, respectively. The CFL has been fixed to 0.5 and the number of PISO loop to 3. Simulations time varies from 24 hours (2D simulations with 200 000 points) to 168 hours (3D simulations with 4 millions of points) depending on the mesh size and the flow regime.

2.4.1.2. 2D steady flow

The characteristic geometrical parameters of the flow are defined on Figure 2.15.

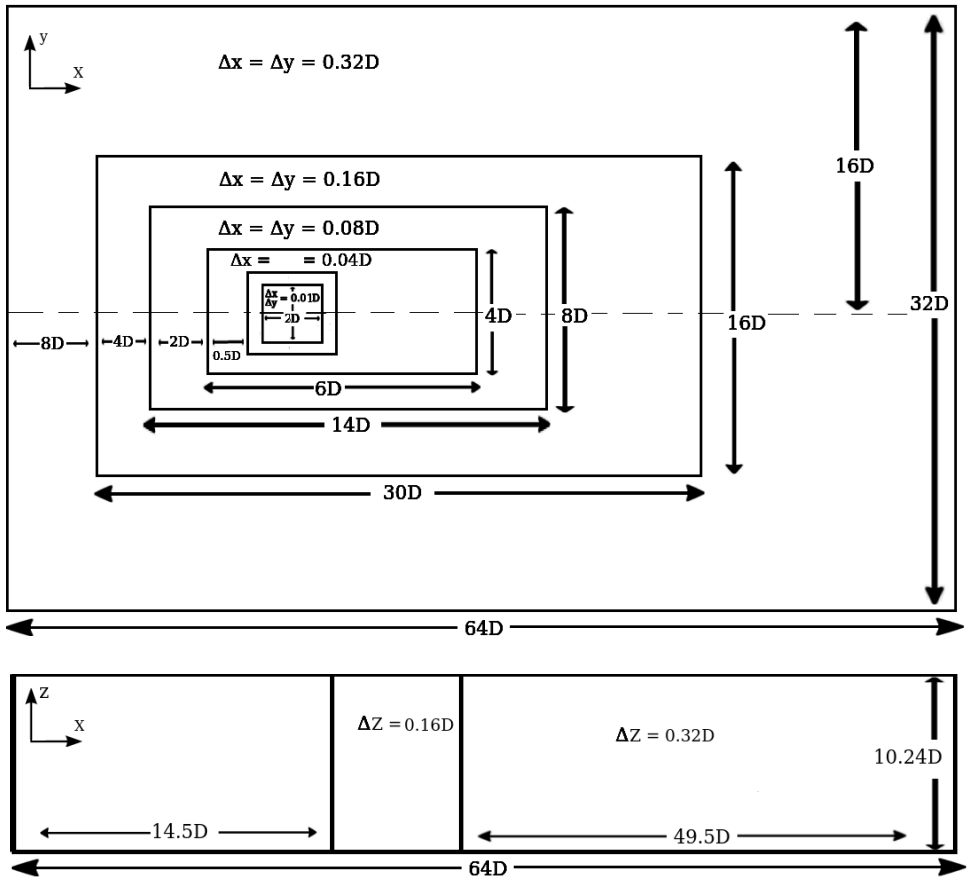


Figure 2.14.: *Computational domain decomposition and grid spacings. (x, y)-plane (top) and spanwise direction (bottom).*

At $Re = 30$, the flow is characterized by a steady recirculating region located just behind the cylinder. All characteristic geometrical parameters reported in Table 2.2 compare well with the data of the literature with differences less than 6% for the most refined grid.

Table 2.2.: *Geometrical parameters of the wake, and drag coefficient for the configuration of a fixed cylinder at $Re = 30$. Numerical and experimental data from literature are provided for comparison.*

		L/D	a/D	b/D	θ^o	C_D
Present ($Re = 30$)	$\Delta x = \Delta y = 0.02D$	**1.66**	**0.556**	**0.53**	**47.80**	**1.78**
	$\Delta x = \Delta y = 0.01D$	**1.64**	**0.55**	**0.53**	**48.40**	**1.77**
Pinelli et al. [99] (Num.)		1.70	0.56	0.52	48.05	1.80
Blackburn and Henderson [5] (Num.)		-	-	-	-	1.74
Coutanceau and Bouard [13] (Expe.)		1.55	0.54	0.54	50.00	-
Tritton [139] (Expe.)		-	-	-	-	1.74

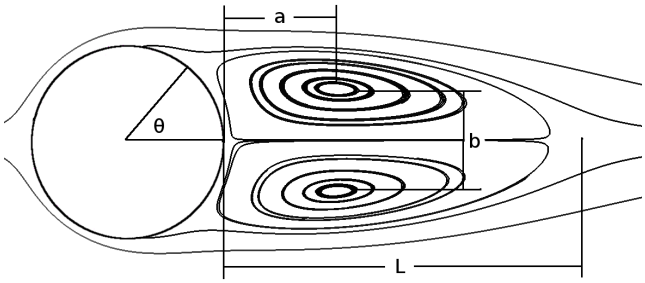


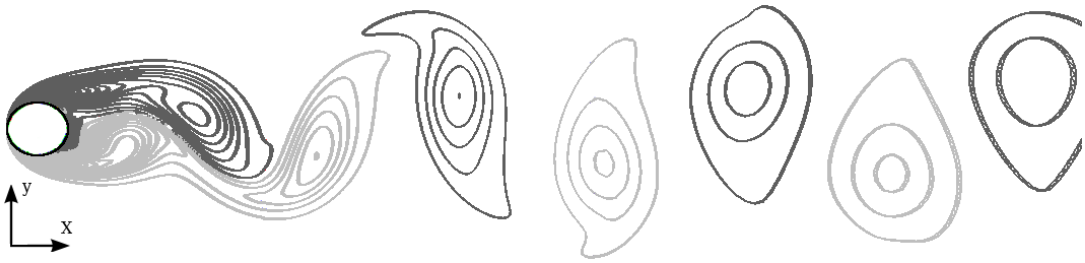
Figure 2.15.: *Characteristic geometrical parameters in the steady regime. L is the length of the recirculation, a is the distance between the cylinder and the recirculations centers, b is the vertical distance between the two recirculation centers, and θ is the separation angle measured from the rear stagnation point.*

2.4.1.3. 2D unsteady flows

Simulations in 2D unsteady regimes with vortex shedding have been performed at $Re = 100$ and 185, i.e. above $Re_c = 40$ for the transition to unsteadiness according to Williamson [147] and Norberg [90] . The vorticity contours shown in Figure 2.16 exhibit the well-known Karman vortex street featuring the periodic shedding of vortices, convected and diffused away from the cylinder.

The topology of the solutions compares well with that reported in several reference studies, see for instant in the papers by Guilmineau and Queutey [37] and Pinelli et al. [99] . The corresponding time evolutions of C_D and C_L are plotted in Figure 2.17 and show that the amplitude of the lift and drag fluctuations increase with the Reynolds number, in good agreement with the paper by Guilmineau and Queutey [37] . For both Reynolds numbers, the Strouhal number, the mean drag (computed over 10 time periods) and the rms lift coefficients compare well with the literature data summarized in Table 2.3.

(a)



(b)

Figure 2.16.: *Vorticity countours evidencing the shedding of large-scale vortices in 2D flow past a fixed circular cylinder at $Re = 100$ (a) and $Re = 185$ (b). Vorticity magnitude veries from -1 (black) to 1 (white)*

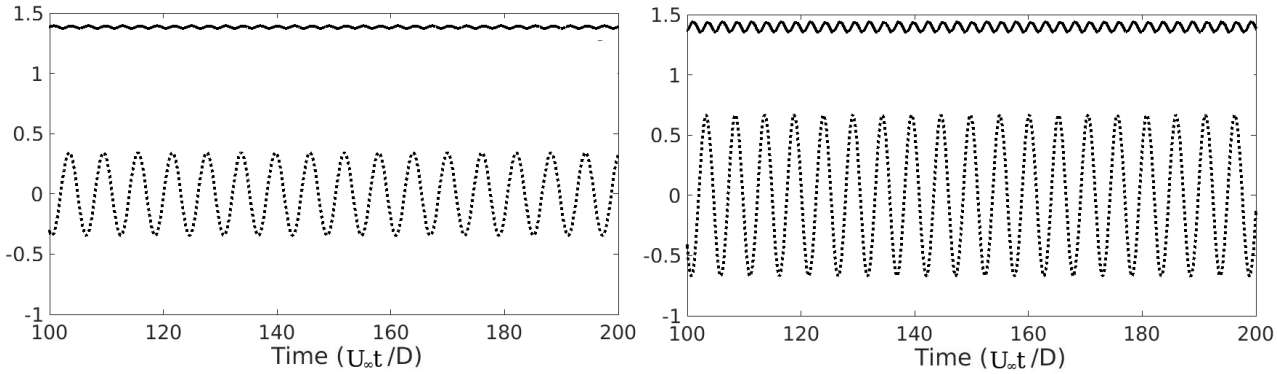


Figure 2.17.: *Temporal evolutions of C_D (full line) and C_L (dashed line) for the 2D flow at $Re = 100$ (left); $Re = 185$ (right).*

Table 2.3.: *Mean drag, rms lift coefficients and Strouhal number for 2D flow past a fixed cylinder at $Re = 100$ and $Re = 185$. Numerical and experimental data from the literature are provided for comparison.*

		C_D	C_L^{rms}	S_t	θ_{mean}
Present ($Re = 100$)	$\Delta x = \Delta y = 0.02D$	**1.38**	-	**0.165**	118.9
	$\Delta x = \Delta y = 0.01D$	**1.37**	-	**0.165**	118.9
Blackburn and Henderson [5] (Num.)		1.35	-	-	-
Barkley and Henderson [3] (Num.)		-	-	0.165	-
Williamson [148] (Expe.)		-	-	0.164	-
Henderson [40] (Num.)		1.35	-	-	-
Norberg [90] (Expe.)		-	-	0.164	-
Present ($Re = 185$)	$\Delta x = \Delta y = 0.02D$	**1.387**	**0.436**	**0.198**	110.8
	$\Delta x = \Delta y = 0.01D$	**1.379**	**0.427**	**0.198**	110.8
Pinelli et al. [99]	$\Delta x = \Delta y = 0.005D$	1.430	0.423	0.196	-
	$\Delta x = \Delta y = 0.01D$	1.509	0.428	0.199	-
Vanella and Balaras [142] (Num.)		1.377	0.461	-	-
Guilmineau and Queutey [37] (Num.)		1.287	0.443	0.195	-
Lu and Dalton [68] (Num.)		1.310	0.422	0.195	-
Williamson [147] (Expe.)		-	-	0.193	-

In addition, Figure2.18 also shows a good agreement with the literature on the prediction of the separation angle at $Re = 100$ and $Re = 185$ as well as on its evolution when increasing Reynolds number.

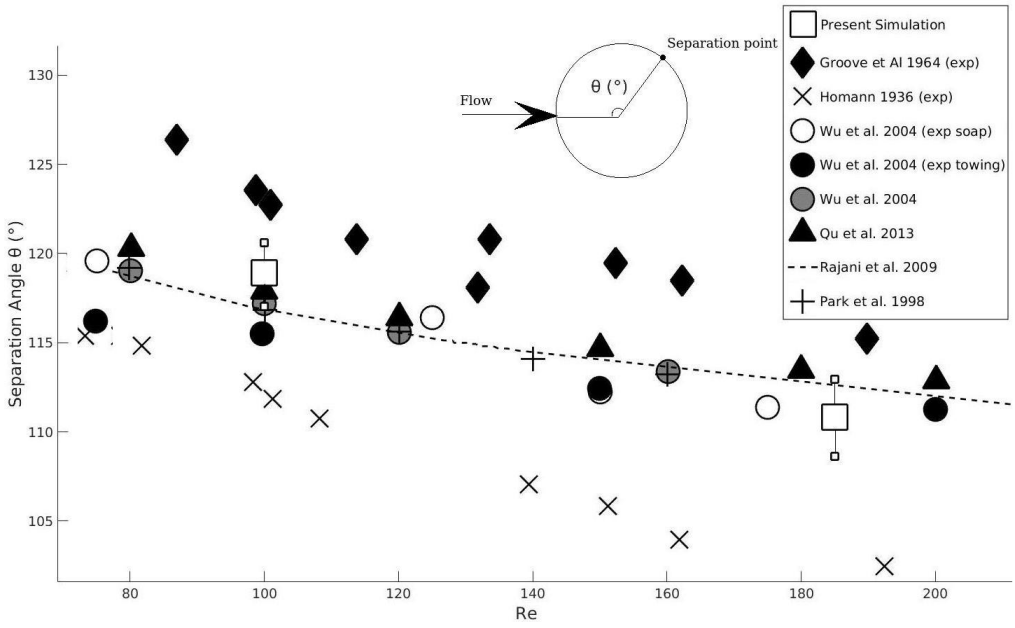


Figure 2.18.: *Mean separation angle as a function of the Reynolds number. Error bars correspond to the min/max values achieved during the duration of the averaging process.*

2.4.1.4. 3D unsteady flows

In order to show the capacity of the code to accurately predict 3D unsteady flows, additional simulations have been performed at $Re = 200$ and 300, i.e., above the critical value $Re_c = 190$ for the transition to 3D flow, and within the range of Reynolds numbers where the 3D pattern transitions from mode A to mode B, according to the reference study of Williamson [148] .

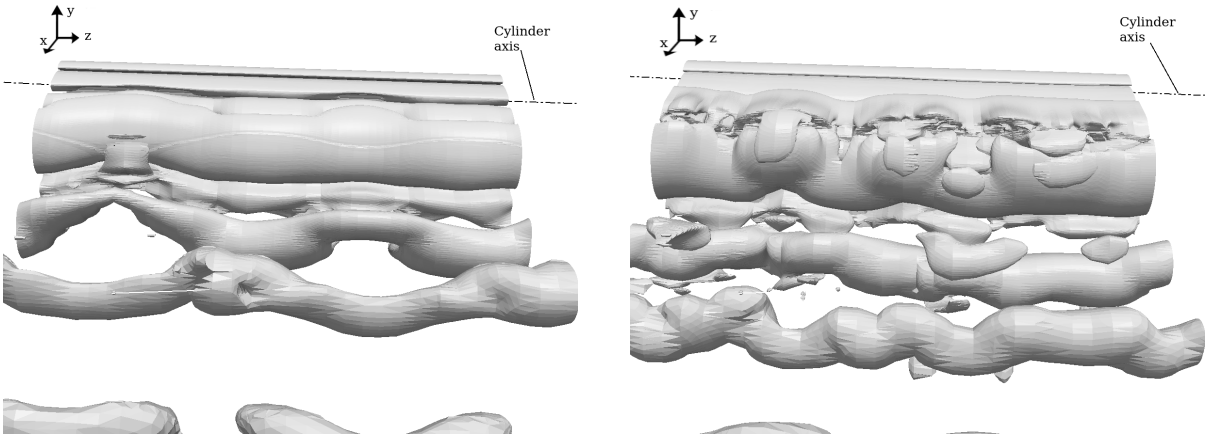


Figure 2.19.: *Iso-surfaces of the instantaneous Q-criterion* $(-0.8 < Q < 0.8)$ *at* $Re = 200$ *(left) and* $Re = 300$ *(right).*

The present simulations predict well the occurrence of 3D vortex shedding, as shown by the instantaneous Q-criterion iso-surfaces in Figure 2.19. When increasing Reynolds number from $Re = 200$ to $Re = 300$, the solution shows a strong decrease of the spanwise wavelength λ_z, from $\lambda_z/D \simeq 4.5$ to $\lambda_z/D \simeq 1.25$ as previously observed by Williamson [148] at the transition between mode A and mode B. The temporal evolution of C_D and C_L in Figure 2.20 shows a modulated behaviour characteristic of these 3D flows, all values

being in agreement with the literature data, as seen from Table 2.4.

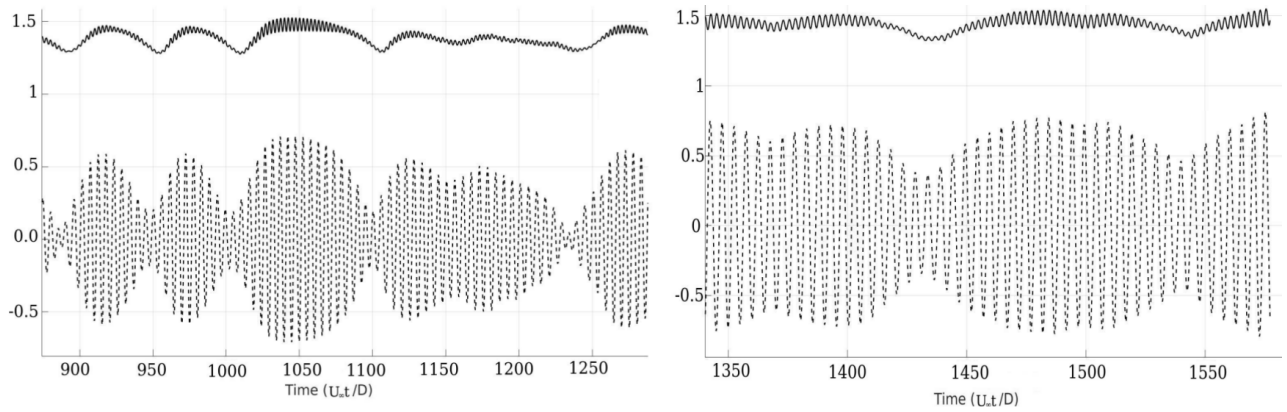


Figure 2.20.: *Temporal evolutions of C_D (full line) and C_L (dashed line) for the 3D flow at $Re = 200$ (left); $Re = 300$ (right).*

Table 2.4.: *Mean drag, rms lift coefficients and Strouhal number for 3D flow past a fixed cylinder at $Re = 200$ and $Re = 300$. Numerical and experimental data from the literature are provided for comparison.*

		C_D	C_L^{rms}	St
Present ($Re = 200$)	$\Delta x = \Delta y = 0.02D$ & $\Delta Z = 0.16D$	**1.384**	**0.346**	**0.1802**
Rajani et al. [106] (Num.)		1.338	0.4216	0.1936
Qu et al. [105] (Num.)		1.24	0.339	0.1801
Williamson [148] (Expe.)		-	-	0.1800
Pinelli (private Communication)		1.371	0.163	0.1915
Present ($Re = 300$)	$\Delta x = \Delta y = 0.02D$ & $\Delta Z = 0.16D$	**1.43**	**0.453**	**0.198**
Rajani et al. [106] (Num.)		1.28	0.499	0.195
Mittal and Balachandar [80] (Num.)		1.26	0.38	0.203
Williamson [148] (Expe.)		-	-	0.203
Norberg [89] (Expe.)		-	0.435	0.203
Wieselsberger [146] (Expe.)		1.22	-	-

2.4.2. Flow around a sphere

The performance of the IBM solver to accurately represent three dimensional configurations is assessed by the analysis of the laminar flow around a sphere. Different Reynolds numbers in the range $100,\ 300$ are considered. Because of the larger degree of freedom for three-dimensional development if compared with the circular cylinder, this case exhibits the emergence of more complex dynamics and vortex interactions.

2.4.2.1. Computational details

The center of the sphere is at the origin of the domain at $(0, 0, 0)$. The dimensions of the computational domain are those used above for the cylinder, respectively $[-16D, 48D] \times [-16D, 16D] \times [-H/2, H/2]$ in

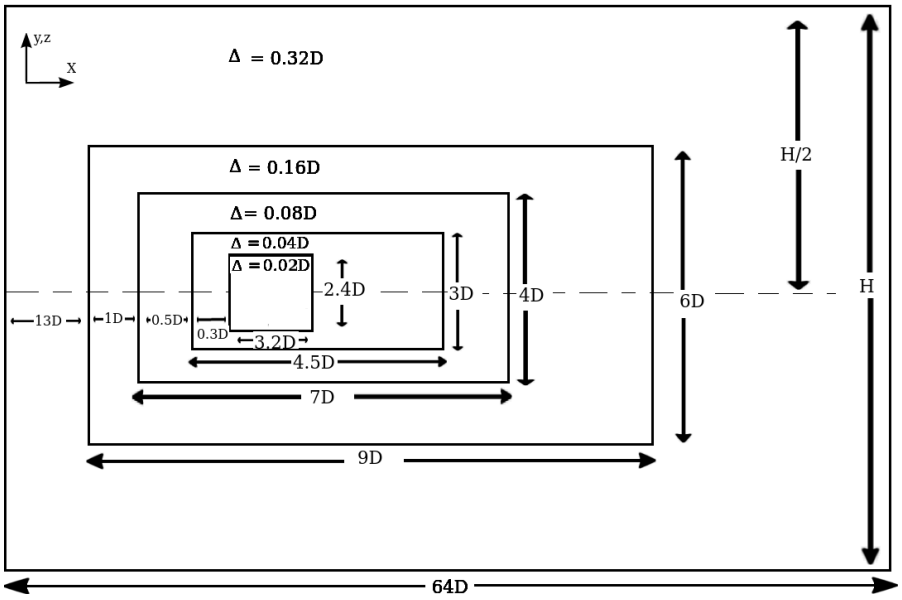


Figure 2.21.: *Computational domain decomposition and grid spacings.* (x, y)*-plane and* (x, z)*-plane.*

the streamwise (x), vertical (y) and spanwise (z) directions as shown in Figure 2.21. The size in the spanwise direction has been set to $H = 10$ (Domain 1) and $H = 32$ (Domain 2).

The grid is uniform in the neighborhood of the sphere, i.e. in the region $-1.2D \leq x \leq 2D$, $-1.2D \leq y \leq 1.2D$ and $-1.2D \leq z \leq 1.2D$. The body is discretized using 7652 Lagrangian markers ($\alpha = 1.012$). Outside this region, the mesh size is coarsened, with a factor of 2.0 on four grid levels in the (x, y)-plane and (x, z)-plane (as shown Figure 2.21) with $\Delta x = \Delta y = \Delta z = \Delta$.

All simulations have been performed on 96 cpu of the AMU computing facilities. Compared to the cylinder case, the CFL had to be reduced to 0.2 and the number of PISO loops remains equal to 3.

2.4.2.2. 3D steady axisymmetric flow

At $Re = 100$ the flow is characterized by a steady axisymmetric recirculating region located just behind the sphere. All characteristic geometrical parameters defined on Figure 2.15 compare well with the data available in the literature, which are reported in Table 2.5. Differences lower than 6% are observed. The results also compare well with the data of Johnson and Patel [53] as shown on Figure 2.22 for the instantaneous flow field pressure coefficient contours.

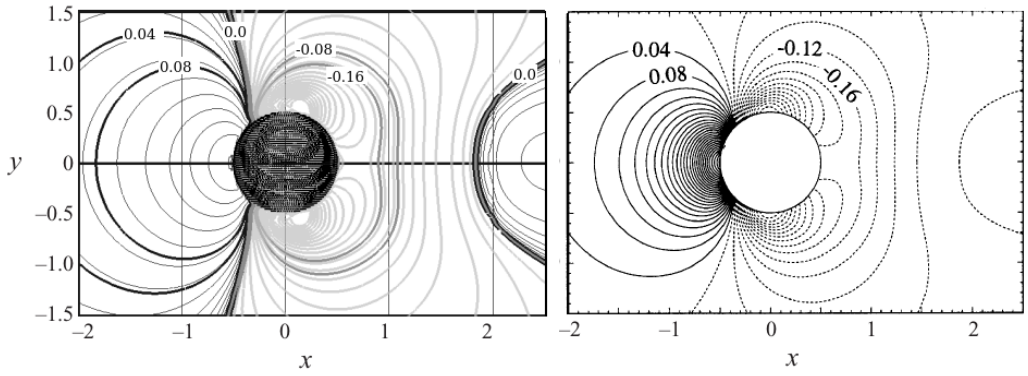


Figure 2.22.: *Snapshot of flow field pressure coefficient contours. Axisymmetric flow at* $Re = 100$*. Present Results (left), results of Johnson and Patel [53] (right).*

Table 2.5.: *Geometrical parameters of the wake and drag coefficients. Flow past a fixed sphere at* $Re = 100$*. Numerical and experimental data from literature are provided for comparison.*

	L/D	θ^o	C_D
Present ($Re = 100$) $\Delta x = \Delta y = 0.02D$	**0.92**	**53.03**	**1.14**
Taneda [134] (Expe.)	0.89	-	-
Nakamura [86] (Expe.)	-	53	-
Johnson and Patel [53] (Num.) $\Delta x = \Delta y = 0.005D$	0.88	53	1.08
Giacobello et al. [33] (Num.)	0.88	53	-
Tomboulides and Orszag [136] (Num.)	0.88	53	-

2.4.2.3. 3D steady non-axisymmetric flow

For Reynolds numbers in the range $[211,\ 270]$, the axial symmetry of the flow is broken but the field keeps a plane of symmetry. In this range of Reynolds numbers, the flow remains steady.

Current calculations has been performed at $Re = 250$ and the location of the symmetry plane was allowed to arise naturally, emerging by the numerical perturbation of the solver only. For a clear presentation of the results, the flow field has been rotated such that the symmetry plane coincides with the (x, y)-plane. Results are found to be very close to reference data as shown in Table 2.6. We notice that results obtained in the larger domain in the spanwise direction (dom2) are in better agreement with reference data, with differences of less than 3%.

The presence of a symmetry plane in the flow can be clearly observed from the three-dimensional particles paths out of the (x,y)-plane upstream of the sphere shown on Figure 2.23 as well as from the snapshot of 3D streamwise shown on Figure 2.24. Similarly to the simulation of Johnson and Patel [53] , it can be seen that the upper spiral in the (x,y)-plane is actually fed by fluid from upstream while the lower spiral releases fluid into the wake after sending it up and around the upper spiral.

Present results exhibit agreement with the findings by Johnson and Patel [53] as shown on Figure 2.25 for the pressure coefficient contours for both the (x, z)- and (x, y)-planes.

The pressure field in the (x,z)-plane is completely symmetric but the pressure contours in the (x, y)-plane are not. The pressure minimum in the region of the lower vortex is lower than that in the region of the upper vortex, which corresponds to the phenomenon observed by Johnson and Patel [53] .

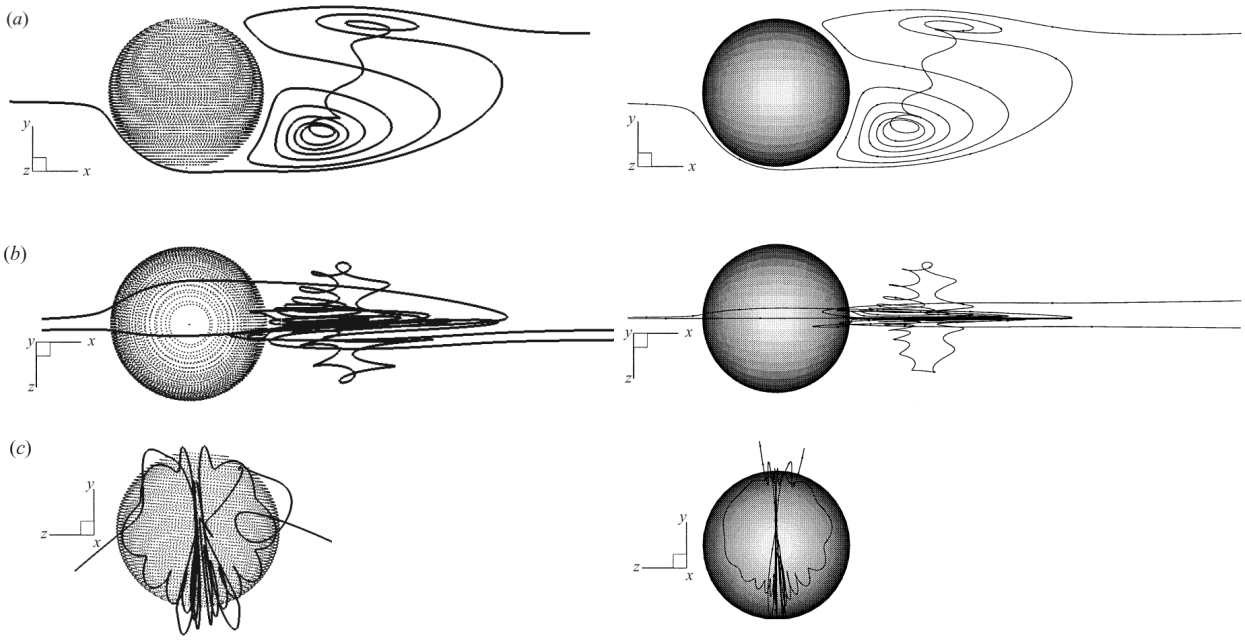


Figure 2.23.: *Snapshot of 3D particle path at $Re = 250$ for (x,y)-view (a), (b) (x,z)-view, (c) (y,z)-view. Present results (left), Results of Johnson and Patel [53] (right).*

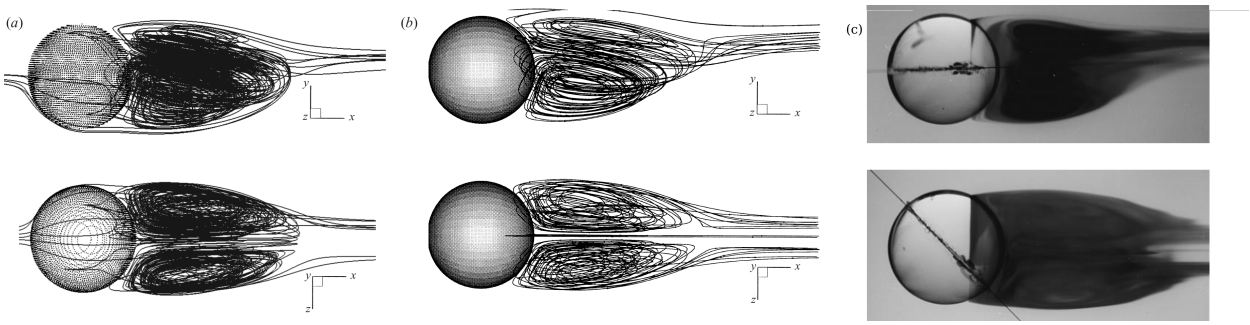


Figure 2.24.: *Snapshot of 3D streamlines at $Re = 250$ for (x,y)-view (up), (x,z)-view (down). Present results (a), results of Johnson and Patel [53] (b), experiments with dye injection of Johnson and Patel [53] (c).*

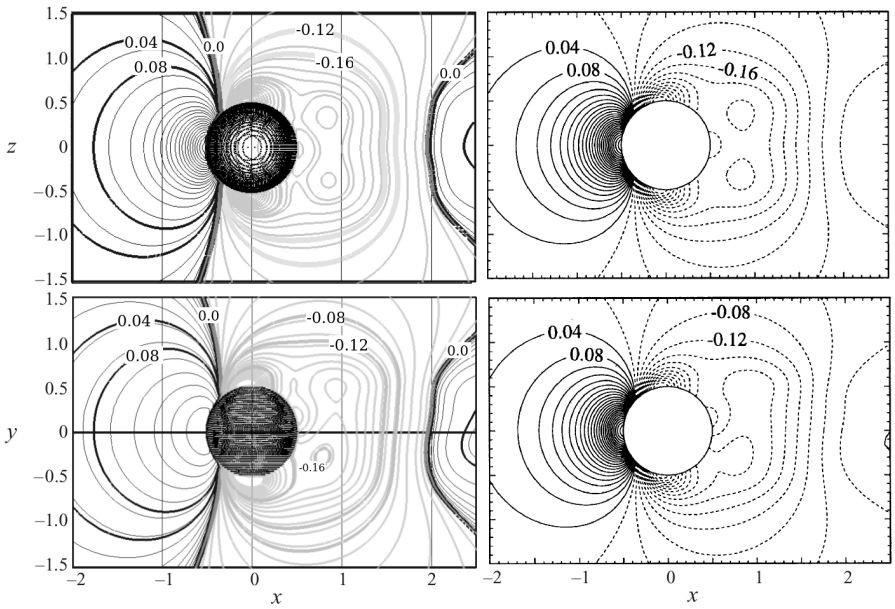


Figure 2.25.: *Snapshot of instantaneous flow field pressure coefficient contours. Non-axisymmetric flow at* $Re = 250$. *Present results (left) and results of Johnson and Patel [53] (right).*

Table 2.6.: *Drag and lift mean coefficients. Flow past a fixed sphere at* $Re = 250$. *Numerical and experimental data from literature are provided for comparison.*

		$C_{D_{mean}}$	$C_{L_{mean}}$
Present ($Re = 250$)	$\Delta x = \Delta y = 0.02D - Dom1$	**0.76**	**-0.057**
	$\Delta x = \Delta y = 0.02D - Dom2$	**0.72**	**-0.062**
Johnson and Patel [53] (Num.) $\Delta x = \Delta y = 0.005D$		0.70	-0.061
Giacobello et al. [33] (Num.)		0.702	-0.061

2.4.2.4. 3D unsteady non-axisymmetric flow

For Reynolds numbers greater than $Re = 270$ the flow around a sphere is expected to become unsteady Johnson and Patel [53] . The flow shows a highly organized periodic structure dominated by vortex shedding.

In the present study, simulations are performed at $Re = 300$ and results are summarized in Table 2.7 together with literature data.

As for the 3D steady case, results obtained in the larger (spanwise) domain (dom 2) are in closer agreement with the data of the literature. The agreement between numerical results is very good, with differences less than 4% for all quantities. Regarding experimental results, only few measurements are avalaible, and differences with present results are of about 6% and 8% for the St and the $C_{D_{mean}}$, respectively.

The near wake dynamics ($x \leq 5$) is well predicted as shown by Figure 2.26 both for the mean streamwise velocity and r.m.s. quantities. For ($x > 5$) the mean streamwise velocity remains well predicted while the r.m.s. becomes overestimated 15% probably because of the coarsening of the mesh in the far wake.

Table 2.7.: *Drag and lift mean coefficients. Flow past a sphere at $Re = 300$. Numerical and experimental data from literature are provided for comparison.*

		$C_{D_{mean}}$	$C_{L_{mean}}$	S_t
Present	$\Delta x = \Delta y = \Delta z = 0.02D - (dom1)$	$\mathbf{0.705 \pm 3.3 \times 10^{-3}}$	$\mathbf{0.0659 \pm 1.89 \times 10^{-2}}$	0.13
($Re = 300$)	$\Delta x = \Delta y = \Delta z = 0.02D - (dom2)$	$\mathbf{0.679 \pm 3.9 \times 10^{-3}}$	$\mathbf{0.066 \pm 2.03 \times 10^{-2}}$	0.139
Johnson and Patel [53] (Num.) $\Delta x = \Delta y = 0.005D$		$0.656 \pm 3.5 \times 10^{-3}$	$0.069 \pm 1.6 \times 10^{-2}$	0.136
Giacobello et al. [33] (Num.)		0.658	0.067	0.134
Tomboulides et al. [137] (Num.)		$0.671 \pm 2.8 \times 10^{-3}$	-	0.136
Roos and Willmarth [111] (Expe.)		0.629	-	-
Johnson and Patel [53] (Expe.)		-	-	0.148-0.165

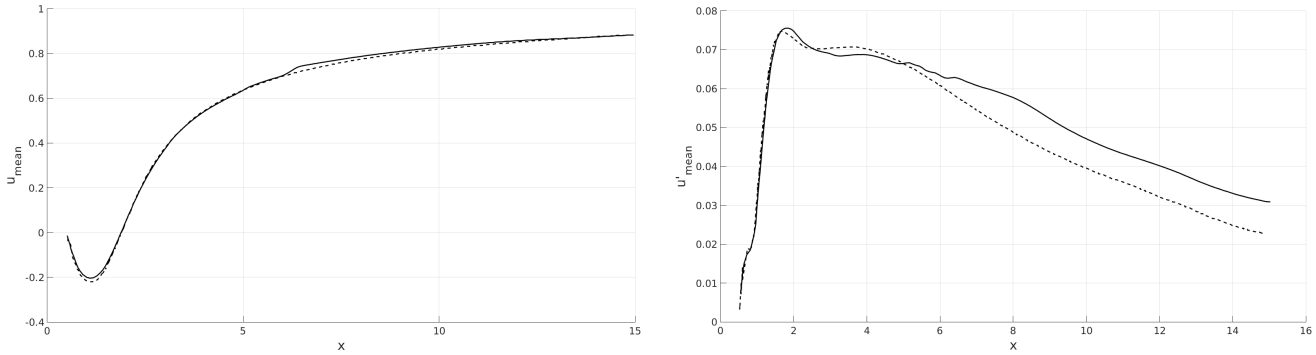


Figure 2.26.: *Averaged streamwise velocity (left) and r.m.s (right). Present result (–), Johnson and Patel [53] (- -). Flow past a sphere at $Re = 300$.*

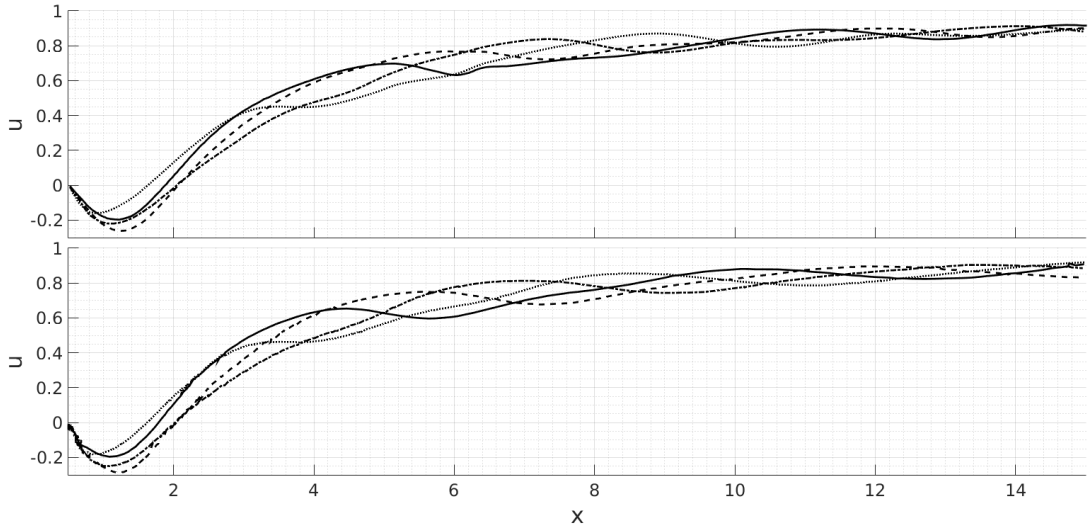


Figure 2.27.: *Streamwise velocity at every quarter period along the axis of motion from the rear of the sphere. $\phi = 0$ (–), $\phi = \pi/2$ (- -), $\phi = \pi$ (-.-), $\phi = 3\pi/2$ (...). Present results (up), Johnson and Patel [53] (down). Flow past a sphere at $Re = 300$.*

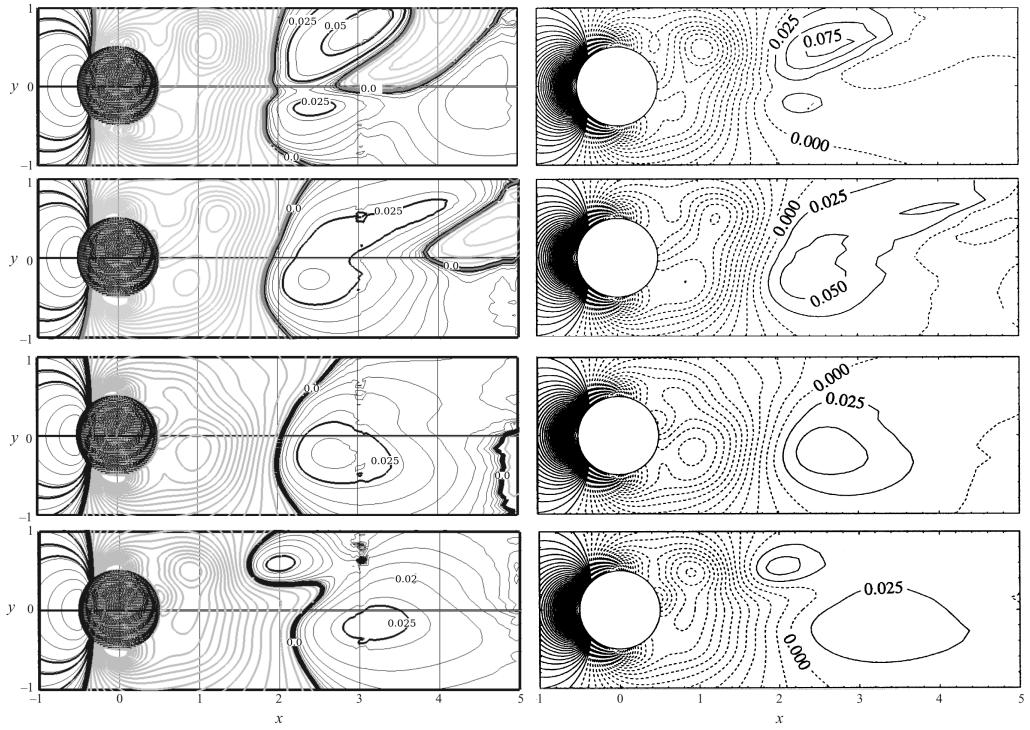


Figure 2.28.: *Instantaneous pressure coefficient contour on the (x,y)-plane at every quarter period from* $\phi = 0$ *(up) to* $\phi = 3\pi/2$ *(down). Present results (left), Johnson and Patel [53] (right).*

Figure 2.27 shows the streamwise velocity at every quarter period from the rear of the sphere. We can observe the same convecting structures in the wake as the one predicted by Johnson and Patel [53] . A small delay is observed, which is probably due to the mesh refinement in the wake. The traveling wave with a peak at $x = 5$ instead of $x_{ref} = 4.5$ for $\phi = 0$ move to $x = 9$ instead of $x_{ref} = 8.5$ for $\phi = 3\pi/2$. The location of zero velocity is well captured for every quarter of phase going from $x = 2$ to $x = 1.5$ for $\phi = \pi/2$ to $\phi = 3\pi/2$ respectively.

The pressure coefficient is also in good agreement with the results of Ref. Johnson and Patel [53] as shown on Figures 2.28 & 2.29 for each quarter phase.

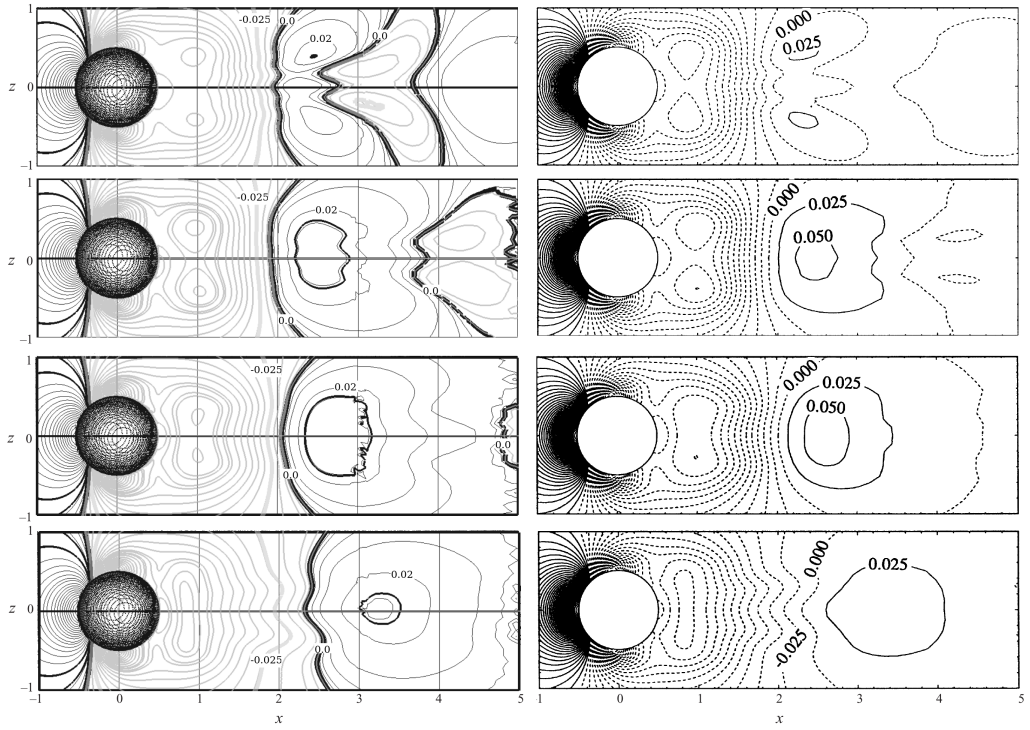


Figure 2.29.: *Instantaneous pressure coefficient contour on the (x,z)-plane at every quarter period from* $\phi = 0$ *(up) to* $\phi = 3\pi/2$ *(down). Present results (left), Johnson and Patel [53] (right).*

Chapter 3

Fluid/structure interactions for the flow past a cylinder

This chapter presents the developments undertaken to model ultimately fluid structure interactions using the new IBM solver developped in OpenFOAM. IBM has emerged as a powerful tool for tackling such problems due to their inherent ability to handle deformable or moving bodies without the need for expensive dynamic re-meshing strategies. We focus in this chapter on the flow past a cylinder at low Reynolds number, with the aim to address the lock-in phenomenon, when the cylinder experiences cross-flow vibrations. This problem occurs when the vortex-shedding of the cylinder gets close to its natural structural frequency. In order to validate the capacity of the new solver to deal with moving obstacles, we have first performed simulations with a forced sinusoidal motion directly imposed on the cylinder. Preliminary computations have been performed thereafter for free oscillations modelled by a partitioned approach using both weak and strong couplings.

3.1. Numerical models

When considering fluid/structure interactions a coupled system with both fluid and structure has to be modelled. In the monolithic method, both the fluid and structural equations are solved in a single solver. Monolithic approaches are unconditionnaly stable, but available solvers for the fluid and the structure cannot be used straightforwardly.

The system is here solved in a partitioned way. The fluid part and the solid part are solved using their own numerical methods, and interact with each other through the boundary conditions at the fluid/structure interface thanks to the IBM, and to an adequate coupling strategy depending on the problem studied. Hereafter are described the different strategies in order to model forced and free oscillations.

3.1.1. Forced oscillations case

The forced oscillations are characterized by a dimensionless frequency $F = f_o/S_t$ and an amplitude $A = y_{max}/D$, where f_o is the frequency of the forced oscillation and y_{max} is the maximal vertical displacement of the structure center $\mathbf{X}_c(X_c, Y_c, Z_c)$ defined by :

$$\mathbf{X}_c = \frac{1}{K} \sum_{k \in D_j} \mathbf{X}_k \tag{3.1}$$

with K the total number of lagrangian marker

The motion of the cylinder on the vertical direction is assumed to be governed here by the following equation :

$$Y_c(t) = A \cos(2\pi f_0 t) \tag{3.2}$$

The following time-dependent boundary condition on the cylinder surface is incorporated into the IBM force term defined in Eq. 2.7 using the IBM target velocity :

$$U_{body} = \begin{pmatrix} 0 \\ \dfrac{dY_c}{dt} \\ 0 \end{pmatrix} \tag{3.3}$$

In this configuration of forced oscillations, only the predictor step of the PISO algorithm introduced in Sec 2.2.2 has to be modified at each time step n as following:

1. **Predictor step:**
   a) An estimate velocity $\widehat{\mathbf{u}}$ is obtained by solving the momentum Navier–Stokes equations without any force term, and using the pressure p computed at the previous time step $n-1$:

$$\frac{\partial \widehat{\mathbf{u}}}{\partial t} + \nabla \cdot (\widehat{\mathbf{u}}\widehat{\mathbf{u}}) = -\nabla p + \frac{1}{Re} \nabla^2 \widehat{\mathbf{u}} \tag{3.4}$$

   b) **Forcing and body motion step:**
      i. The new position of the body center is computed thanks to the equation 3.2. Then all Lagrangian markers are moved thanks to the equation :

$$\mathbf{X}_s = \mathbf{X}_s + \begin{pmatrix} 0 \\ Y_c(t) - Y_c(t - \Delta t) \\ 0 \end{pmatrix} \tag{3.5}$$

ii. The calculation of the IBM force $\mathbf{F}_s$ is done on the Lagrangian markers using the interpolation of $\widehat{\mathbf{u}}$ (Eq. 2.7) :

$$\mathbf{F}_s^n = \frac{\mathbf{U}_s^d - \mathcal{I}[\hat{\mathbf{u}}]_s}{\Delta t} \tag{3.6}$$

where here $\mathbf{U}_s^d = U_{body}$, U_{body} being the velocity of the structure computed thanks to the Eq. 3.3.

iii. The values of the IBM force $\mathbf{F}_s$ is then spread on the Eulerian mesh to calculate $\mathbf{f}$ (Eq. 2.20).

c) A new velocity $\mathbf{u}^{\star,1}$ is calculated from the Navier-Stokes equations accounting now the immersed boundary force term f:

$$\frac{\partial \mathbf{u}^{\star,1}}{\partial t} + \nabla \cdot (\mathbf{u}^{\star,1}\mathbf{u}^{\star,1}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u}^{\star,1} + f(\widehat{\mathbf{u}}) \tag{3.7}$$

$\mathbf{u}^{\star,1}$ is the guess value of the velocity in the iterative PISO loop

The PISO loop and the final steps of the algorithm remain unchanged.

The cylinder oscillates here by the action of the flow at the natural frequency of the oscillator f_N, which depends on the mass, the rigidity and possibly the damping of the cylinder. The frequency of the cylinder oscillation is thus different from the Strouhal frequency of the fixed cylinder. This phenomenon occurs over a certain range of reduced velocities ($U_r = U_\infty / F_n D$), where the Strouhal is close to the natural frequency. We can then observe a peak of amplitude.

3.1.1.1. Structure motion equation

The motion of the cylinder is governed by the following equation of the damped oscillator:

$$m * [\frac{d^2 Y_c^*}{dt^{*2}} + 4\pi f_N^* \xi \frac{dY_c^*}{dt^*} + (2\pi f_N^*)^2 (Y_c^* - {Y^*}_0)] = F_L^* \tag{3.8}$$

where :

- ξ^* is the damping coefficient
- m^* is the mass of the structure
- F_L^* is the force vector of the flow exerted on the structure
- ${Y^*}_0$ is the position vector at rest of the spring on y-direction

The equations are presented for a structure moving freely in the cross-flow direction as the extense to a 2D model of freedom is trivial.

The variables are made dimensionless in the following way:

$$
\begin{aligned}
t &= \frac{U_\infty}{D} t^* \\
Y_c &= Y_c^*/D \\
Y_0 &= Y_0^*/D \\
\frac{d^2Y}{dt^2} &= \frac{D}{U_\infty^2} \frac{d^2Y^*}{dt^{*2}}
\end{aligned}
$$

$$
\begin{aligned}
\frac{dY}{dt} &= \frac{1}{U_\infty} \frac{dY^*}{dt^*} \\
f_N &= \frac{D}{U_\infty} f_N^* \\
M &= \frac{m^*}{\rho_f D^2} \\
C_L &= \frac{2F_L^*}{\rho_f U_\infty^2 D}
\end{aligned}
$$

with ρ_f the volumic mass of the fluid.

The dimensionless equation for the structure is introduced as:

$$
M\frac{d^2Y_c}{dt^2} + 4\pi f_N \xi M \frac{dY_c}{dt} + (2\pi f_N)^2 M(Y_c - Y_0) = \frac{C_L}{2} \tag{3.9}
$$

that can be rewriten as:

$$
M\frac{d^2Y_c}{dt^2} + B\frac{dY_c}{dt} + K(Y - Y_{c0}) = \frac{C_L}{2} \tag{3.10}
$$

with :

$$
\begin{aligned}
B &= 4\pi f_N^* \xi M^* \\
K &= (2\pi f_N^*)^2 M^*
\end{aligned}
$$

3.1.1.2. Structure motion solvers

Various explicit and implicit schemes have been tested to integrate time Eq. 3.10.

- **Backward Euler scheme**
  Using a finite-differences discretization, we obtain:

$$
\dot{Y}_c^{n+1} = \frac{1}{\Delta t}(Y_c^{n+1} - Y_c^n) \tag{3.11}
$$

$$
\ddot{Y}_c^{n+1} = \frac{1}{\Delta t^2}(Y_c^{n+1} - 2Y_c^n + Y_c^{n-1}) \tag{3.12}
$$

substituing the terms of Eq. 3.12 in 3.10 and re-arranging terms we obtain:

$$Y_c^{n+1} = \frac{C_L + (\frac{2M}{\Delta t^2} + \frac{B}{dt} - K)Y_c^n + KY_0 - \frac{M}{dt^2}Y_c^{n-1}}{\frac{M}{\Delta t^2} + \frac{B}{dt}} \tag{3.13}$$

This explicit solution gives the next position of the cylinder center Y_c in the y-direction.

- **Newmark scheme**
  We here introduce a predictor step such that:

$$\ddot{Y}_c^{n+1} = \frac{-\frac{C_L}{2} - B\dot{Y}_c - K(Y_c - Y_0)}{M} \tag{3.14}$$

Then, the Newmark scheme builds on Taylor developpements reads:

$$Y_c^{n+1} = Y_c^n + \dot{Y}_c^n \Delta t + \Delta t^2[(\frac{1}{2} - \beta)\ddot{Y}_c^n + \beta\ddot{Y}_c^{n+1}] \tag{3.15}$$

$$\dot{Y}_c^{n+1} = \dot{Y}_c^n + \Delta t[(1-\gamma)\ddot{Y}_c^n + \gamma\ddot{Y}_c^{n+1}] \tag{3.16}$$

with α and γ are two free parameters of the scheme, which control the implicit/explicit nature of the scheme and so its stability, as shown on Table 3.1.

Table 3.1.: *Stability conditions for the Newmark scheme*

Condition	Stability
$\gamma \leqslant 0.5$	unstable
$0.5 \leqslant \gamma$ & $2\beta \leqslant \gamma$	conditionnaly stable
$0.5 \leqslant \gamma$ & $\gamma \leqslant 2\beta$	unconditionnaly stable

- **HHT scheme (Newmark generalization)**

The HHT scheme is a generalization of the Newmark scheme (equivalent to a Newmark scheme with $\alpha = 0$). It comes from the same difference equations but the equation of motion is modified, using the parameter α as:

$$M\ddot{Y}_c^{n+1} + (1-\alpha)(B\dot{Y}_c^{n+1} + K(Y_c^{n+1} - Y_0) - \frac{C_L^{n+1}}{2}) + \alpha(B\dot{Y}_c^n + K(Y_c^n - Y_0) - \frac{C_L^n}{2}) = 0 \tag{3.17}$$

the stability conditions :

$$0 \leqslant \alpha \leqslant \frac{1}{3}$$

$$\beta = \frac{(1+\alpha)^2}{4}$$

$$\gamma = \frac{1}{2} + \alpha$$

we then obtain, by subsituing and rewritting the Newmark Eq. 3.16, the acceleration at the next step:

$$\begin{aligned}\ddot{Y}_c^{n+1} = & \\ & - \frac{\Delta t(1-\alpha)(1-\gamma)B + \Delta^2(1-\alpha)(\frac{1}{2}-\beta)K}{M + \Delta(1-\alpha)\gamma B + \Delta^2(1-\alpha)\beta K}\ddot{Y}_c^n \\ & - \frac{B + \Delta t(1-\alpha)K}{M + \Delta(1-\alpha)\gamma B + \Delta^2(1-\alpha)\beta K}\dot{Y}_c^n \\ & + \frac{-K(Y_c^n - Y_0) + (1-\alpha)\frac{C_L^{n+1}}{2} + \alpha\frac{C_L^n}{2}}{M + \Delta(1-\alpha)\gamma B + \Delta^2(1-\alpha)\beta K}\end{aligned} \tag{3.18}$$

The HHT scheme is at least 2^{nd}-order accurate and unconditionally stable.

3.1.1.3. Validation of the structural motion solver

The solver is validated whitout the fluid. The structure is placed away its equilibrium position in order to induce vibrations. In this case, the equation reads :

$$M\frac{d^2Y_c}{dt^2} + B\frac{dY_c}{dt} + K(Y_c - Y_0) = 0 \tag{3.19}$$

The parameters chosen for the simulation are $Y_0 = 0.5$, $K = 2$, $B = 0$ and $M = 0.25$. The resulting displacement in this case is known analytically as :

$$Y_c(t) = Y_0\cos(\sqrt{\frac{K}{M}}t) \tag{3.20}$$

Without any forcing, the system blows up when the time step us too coarse. A sub-iteration step has been then introduced to overcome this problem when the oscillator is not forced by the flow.

In order to get the solution, sub-iterations are done on variables for $m = 1, ...M$:

$$\ddot{Y}_c^{m+1} = F(\frac{\Delta t}{nb_{iter}}; \dot{Y}_c^m; Y_c^m; C_L^{n+1}; C_L^n) \tag{3.21}$$

$$\dot{Y}_c^{m+1} = G(\frac{\Delta t}{nb_{iter}}; \ddot{Y}_c^m; \dot{Y}_c^m; Y_c^m; C_L^{n+1}; C_L^n) \tag{3.22}$$

$$Y_c^{m+1} = H(\frac{\Delta t}{nb_{iter}}; \ddot{Y}_c^m; \dot{Y}_c^m; Y_c^m; C_L^{n+1}; C_L^n) \tag{3.23}$$

with F, G and H functions are determined according to the scheme chosen in Sec. 3.1.1.2.

Results on the displacement of the oscillator over time are succesfully improved, as it can be seen on Figure 3.1. Here, 10000 sub-iterations are used for the fluid. As the present configuration is simple and does not require too many Lagrangian markers, the additional computational time required is negligible.

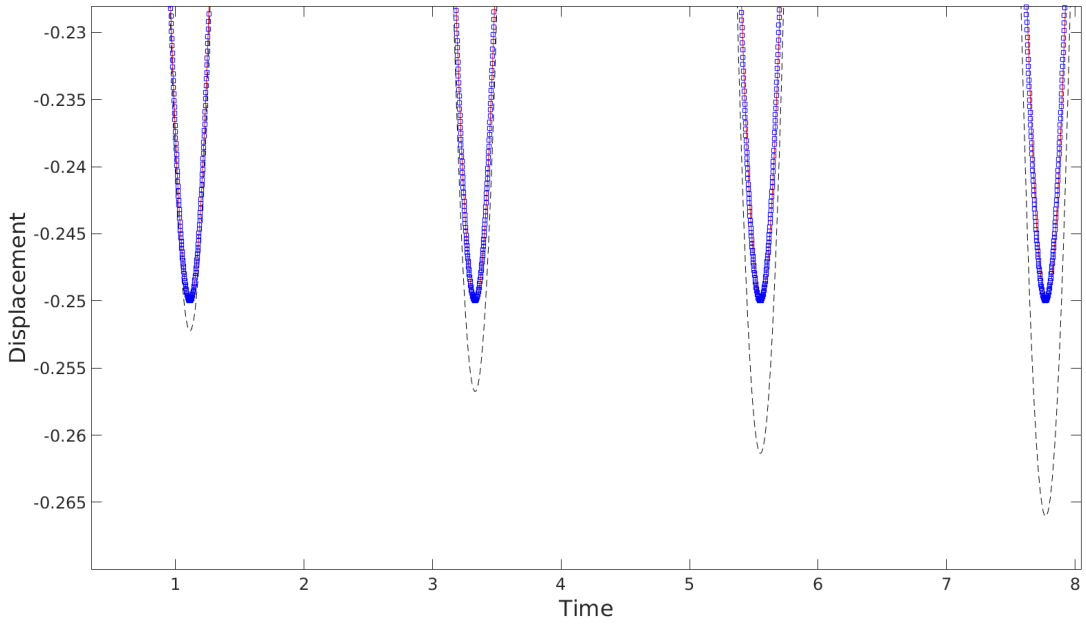


Figure 3.1.: *Displacement over time of the oscillator. Results without sub-iteration (dashed line), with sub-iterations (full line) and analytical solution (red).*

3.1.2. Solver coupling

In practice, the parameters $(M^*; B^*; K^*)$ are chosen to model the motion of the structure, but when fluid forces are not neglected, the actual oscillation frequency f_0 depends on them, and it is generally different from f_N.

When the fluid and structure motions are coupled, the lift force C_L influences the motion of the structure ($Y_c^*(t)$, Eq. 3.10), which influences in turn the flow field and the lift force. The aerodynamics forces and the motion of the structure are then coupled and need to be evaluated carefully. To do so, two approaches exist namely the weak and strong coupling approach.

In the weak approach, the fluid and structure parts are solved sequentially. This may lead, if the time step is too large to an increase of the numerical errors at the interface due to the time lag. In the strong coupling, iterations are performed between the fluid and structure solvers. Strong coupling improves the accuracy in satisfying coupling conditions, and thus reduces the occurence of eventual numerical instabilities (the incompatibility of the kinematic and dynamic quantities at the interface may generate artificial energy and cause numerical instabilities particularly when the structure density is closed or lighter than the fluid density). It is however more demanding in terms of numerical developments and CPU time. Hereafter both methods have been tested.

3.1.2.1. A weak coupling algorithm

The weak coupling method used here after is integrated into the PISO solver introduced in Sec 2.2.2. Only the predictor step has to be modified at each time step n as following:

a) **Predictor step:**

i. An estimate velocity $\widehat{\mathbf{u}}$ is obtained by solving the momentum Navier–Stokes equations without any force term, and using the pressure p computed at the previous time step $n-1$:

$$\frac{\partial \widehat{\mathbf{u}}}{\partial t} + \nabla \cdot (\widehat{\mathbf{u}}\widehat{\mathbf{u}}) = -\nabla p + \frac{1}{Re}\nabla^2 \widehat{\mathbf{u}} \tag{3.24}$$

ii. **Forcing and body motion step :**

A. The new body center position is computed thanks to the function $H(\mathbf{F}_s^{n-1})$ defined Eq. 3.23. Then all the Lagrangian markers are moved thanks to the equation :

$$\mathbf{X}_s^n = \mathbf{X}_s^{n-1} + \mathbf{X}_c^n - \mathbf{X}_c^{n-1} \tag{3.25}$$

B. The calculation of the IBM force $\mathbf{F}_s$ is done on the Lagrangian markers using the interpolation of $\widehat{\mathbf{u}}$ (Eq. 2.7) :

$$\mathbf{F}_s^n = \frac{\mathbf{U}_s^d - \mathcal{I}[\widehat{\mathbf{u}}]_s}{\Delta t} \tag{3.26}$$

where $\mathbf{U}_s^d = \mathbf{U}_{body}$, $\mathbf{U}_{body}$ being computed thanks to the function $G(\mathbf{F}_s^{n-1})$ defined Eq. 3.22.

C. The values of the IBM force $\mathbf{F}_s$ is then spread on the Eulerian mesh to calculate $\mathbf{f}$ (Eq. 2.20).

iii. A new velocity $\mathbf{u}^{\star,1}$ is calculated from the Navier-Stokes equations accounting now the immersed boundary force term f:

$$\frac{\partial \mathbf{u}^{\star,1}}{\partial t} + \nabla \cdot (\mathbf{u}^{\star,1}\mathbf{u}^{\star,1}) = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u}^{\star,1} + f(\widehat{\mathbf{u}}) \tag{3.27}$$

$\mathbf{u}^{\star,1}$ is the guess value of the velocity in the iterative PISO loop

3.1.2.2. A strong coupling algorithm

The strong coupling method used here after is integrated into the PISO solver introduced in Sec 2.2.2.

b) **Predictor step:**

i. An estimate velocity $\widehat{\mathbf{u}}$ is obtained by solving the momentum Navier–Stokes equations without any force term, and using the pressure p computed at the previous time step $n-1$:

$$\frac{\partial \widehat{\mathbf{u}}}{\partial t} + \nabla \cdot (\widehat{\mathbf{u}}\widehat{\mathbf{u}}) = -\nabla p + \frac{1}{Re}\nabla^2 \widehat{\mathbf{u}} \tag{3.28}$$

ii. **Predictor step of the strong coupling algorithm :**

A. The new body center position is computed thanks to the function $\mathbf{X}_c^{n,0} = H(\mathbf{F}_s^{n-1})$, Eq. 3.23. All Lagrangian markers are then moved thanks to the equation :

$$\mathbf{X}_s^{n,0} = \mathbf{X}_s^{n-1} + \mathbf{X}_c^{n,0} - \mathbf{X}_c^{n-1} \tag{3.29}$$

B. The velocity of the body $\mathbf{U}^0_{body}$ is computed thanks to the function $G(\mathbf{F}^{n-1}_s)$, Eq. 3.22. Then the target velocity of all Lagrangian markers are updated in order to take into account the motion of the structure into the N-S equations:

$$\mathbf{U}^{d,0}_s = \mathbf{U}^0_{body} \tag{3.30}$$

C. The calculation of the IBM force $\mathbf{F}^{n,0}_s$ is done on the new Lagrangian markers positions $\mathbf{X}^{n,0}_s$, using the interpolation $\mathcal{I}[\hat{\mathbf{u}}]_s$ (Eq. 2.7) done by the equation :

$$\mathbf{F}^{n,0}_s = \frac{\mathbf{U}^{d,0}_s - \mathcal{I}[\hat{\mathbf{u}}]_s}{\Delta t} \tag{3.31}$$

D. The values of the IBM force $\mathbf{F}^0_s$ is then spread on the Eulerian mesh to calculate $\mathbf{f^0}$ (Eq. 2.20). A new velocity $\mathbf{u}^1$ is calculated from the Navier-Stokes equations accounting now the immersed boundary force term f^0:

$$\frac{\partial \mathbf{u}^1}{\partial t} + \nabla \cdot (\mathbf{u}^1 \mathbf{u}^1) = -\nabla p^1 + \frac{1}{Re}\nabla^2 \mathbf{u}^1 + f^0(\widehat{\mathbf{u}}) \tag{3.32}$$

E. The IBM force $\mathbf{F}^{n,1}_s$ is updated on the Lagrangian markers $\mathbf{X}^{n,1}_s$ using the interpolation $\mathcal{I}[\mathbf{u^1}]_s$ (Eq. 2.7) done by the equation :

$$\mathbf{F}^{n,1}_s = \frac{\mathbf{U}^{d,0}_s - \mathcal{I}[\mathbf{u^1}]_s}{\Delta t} \tag{3.33}$$

F. The body center position is updated thanks to the function $\mathbf{X}^{n,1}_c = H(\mathbf{F}^{n,1}_s)$, Eq. 3.23. Then all Lagrangian markers are moved thanks to the equation :

$$\mathbf{X}^{n,1}_s = \mathbf{X}^{n-1}_s + \mathbf{X}^{n,1}_c - \mathbf{X}^{n-1}_c \tag{3.34}$$

G. The velocity of the body $\mathbf{U}^1_{body}$ is computed thanks to the function $G(\mathbf{F}^{n,1}_s)$, Eq. 3.22. Then the target velocity of all Lagrangian markers are updated in order to take into account the motion of the structure into the N-S equations by :

$$\mathbf{U}^{d,1}_s = \mathbf{U}^1_{body} \tag{3.35}$$

iii. **Corrector step of the strong coupling algorithm for the sub-iterations $a = 2$ to A, and up to convergence :**
The convergence criteria is reached if :

$$\| \sqrt{\frac{1}{K} \sum_{D_s} \mathbf{U}^{d,a}_k - \mathbf{U}^{d,a-1}_k \|^2} < e_{coupling} \tag{3.36}$$

with K the total number of Lagrangian markers and $e_{coupling}$ is set hereafter to 10^{-6}

A. The calculation of the IBM force $\mathbf{F}_s^{n,a-1}$ is done on the new Lagrangian markers $\mathbf{X}_s^{n,a-1}$ using the interpolation $\mathcal{I}[\mathbf{u}^{a-1}]_s$ (Eq. 2.7) done by the equation :

$$\mathbf{F}_s^{n,a-1} = \frac{\mathbf{U}_s^{d,a-1} - \mathcal{I}[\mathbf{u}^{a-1}]_s}{\Delta t} \tag{3.37}$$

B. The values of the IBM force $\mathbf{F}_s^{a-1}$ is then spread on the Eulerian mesh to calculate $\mathbf{f^{a-1}}$ (Eq. 2.20). A new velocity $\mathbf{u}^a$ is calculated from the Navier-Stokes equations accounting now the immersed boundary force term f^{a-1}:

$$\frac{\partial \mathbf{u}^a}{\partial t} + \nabla \cdot (\mathbf{u}^a \mathbf{u}^a) = -\nabla p^a + \frac{1}{Re}\nabla^2 \mathbf{u}^a + f^{a-1}(\mathbf{u}^{a-1}) \tag{3.38}$$

C. Then the IBM force $\mathbf{F}_s^{n,a}$ is updated on the Lagrangian markers $\mathbf{X}_s^{n,a}$ using the interpolation $\mathcal{I}[\mathbf{u^a}]_s$ (Eq. 2.7) done by the equation :

$$\mathbf{F}_s^{n,a} = \frac{\mathbf{U}_s^{d,a-1} - \mathcal{I}[\mathbf{u^a}]_s}{\Delta t} \tag{3.39}$$

D. The position of the body center is updated thanks to the function $\mathbf{X}_c^{n,a} = H(\mathbf{F}_s^{n,a})$, Eq. 3.23. Then all Lagrangian markers are moved thanks to the equation :

$$\mathbf{X}_s^{n,a} = \mathbf{X}_s^{n-1} + \mathbf{X}_c^{n,a} - \mathbf{X}_c^{n-1} \tag{3.40}$$

E. The velocity of the body $\mathbf{U}_{body}^a$ is computed thanks to the function $G(\mathbf{F}_s^{n,a})$, Eq. 3.22. Then the target velocity of all Lagrangian markers is updated in order to take into account the motion of the structure into the N-S equations by :

$$\mathbf{U}_s^{d,a} = \mathbf{U}_{body}^a \tag{3.41}$$

iv. The new guess value of the velocity $\mathbf{u}^{\star,1}$ in the iterative PISO loop is set to :

$$u^{\star,1} = \mathbf{u^A} \tag{3.42}$$

v. **PISO loop:**
For the sub-iteration $m = 1$ to $M - 1$, and up to convergence:

A. At each sub-iteration, a pressure field $p^{\star,m}$ is calculated from the following Poisson equation :

$$\nabla^2 p^{\star,m} = -\nabla \cdot (\mathbf{u}^{\star,m} \nabla \mathbf{u}^{\star,m}) + \nabla \cdot f(\widehat{\mathbf{u}}) \tag{3.43}$$

B. The velocity field is thus corrected using:

$$\mathbf{u}^{\star,m+1} = g\left(\mathbf{u}^{\star,m},\, \nabla p^{\star,m},\, f(\widehat{\mathbf{u}})\right) \tag{3.44}$$

where g as well as all discretized operators used in the algorithm are defined in Annex B.

vi. **Final step:**
The velocity and the pressure are finaly updated at time $n+1$:

$$\mathbf{u}^{n+1} = \mathbf{u}^{\star,M-1} \tag{3.45}$$
$$\mathbf{p}^{n+1} = \mathbf{p}^{\star,M-1} \tag{3.46}$$

3.2. Filtering

The issue is related here to the numerical noise that may occur with IBM when the structure moves on the Eulerian mesh. This noise is related to the Lagrangian markers which pass from the solid to the fluid, and inversely, when integrating forces.

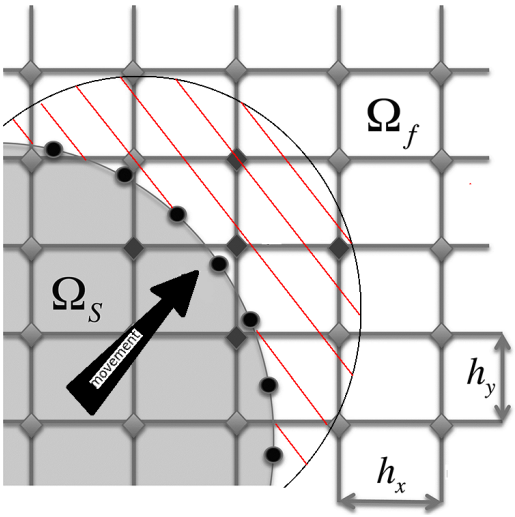


Figure 3.2.: *Sketch of the mesh showing Lagrangian markers and Eulerian point distribution when moving the structure with IBM.*

To overcome this issue a filtering of the solution has been used based on a 1^{st} order low pass filter with a window function. The filtering can be applied during post-processing, or during time integration loop. Effect of the filtering on the time evolution of the lift coefficient C_L is shown on Figure 3.3. We perform inside the code the FFT on the time evolution of the forces in order to determine the peak frequency in the Fourier space. A sample of the 2^N last points is chosen, N being chosen by the user but has to be large enough to allow an accurate estimation of the peak value. The frequency is then forced to be zero, if it is higher than the cut-off frequency, $Fc = f_{peak} + \Delta f_{cut\ off}$, with f_{peak} the peak frequency and $\Delta f_{cut\ off}$ a free parameter chosen by the user to keep the needed

frequencies around the peak (in the following $\Delta f_{cut\ off} = 4$ for the drag and lift signals). The filtered signals, $C_{D_{corrected}}$ and $C_{L_{corrected}}$, are now free of spurious oscillations, as shown on the example on Figure 3.3. The last term of the filtered signals are then used to compute the motion of the structure thanks to the functions F($C_{D_{corrected}}$, $C_{L_{corrected}}$), G($C_{D_{corrected}}$, $C_{L_{corrected}}$),H($C_{D_{corrected}}$, $C_{L_{corrected}}$), defined by Eqs. 3.21, 3.22 and 3.23, respectively.

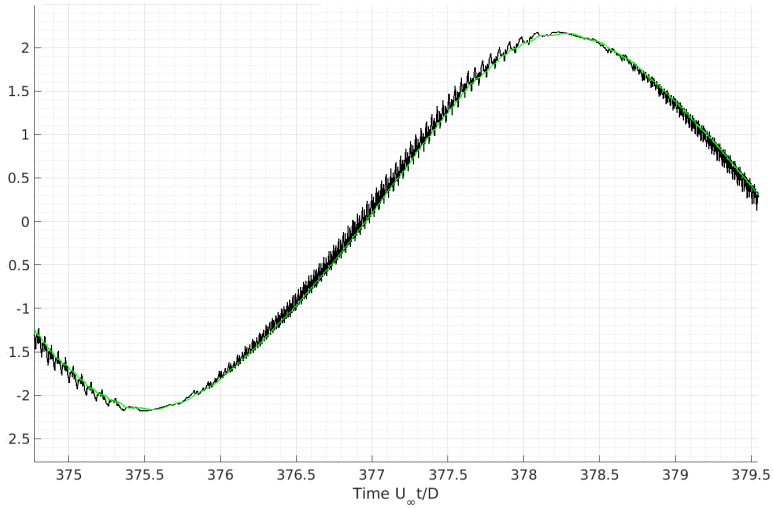


Figure 3.3.: *Effect of filtering on the time evolution of the lift coefficient C_L. Real IBM signal $C_L(t)$ (black) and filtered one $C_{L_{corrected}}(t)$ (green). Flow past an oscillating cylinder at $Re = 500$.*

3.3. Validation

The algorithm is validated by considering academic configurations of flow past a cylinder at low Reynolds numbers. Both forced ($Re = 500$) and free ($Re = 100$) oscillating cylinders are considered.

3.3.1. Computational domain

The computational domain is similar to the one considered for the flow past a $2D$ fixed cylinder, Sec. 2.4.1. The dimensions of the computational domain are $[-16D, 48D] \times [-16D, 16D] \times [-5.12D, 5.12D]$ in the streamwise (x), vertical (y) and spanwise (z) directions. Two resolutions have been considered in the vicinity of the cylinder corresponding to $\Delta x = \Delta y = 0.02D$ and $\Delta x = \Delta y = 0.01D$. Simulations have been performed on 12 CPU. The CFL has been fixed to 0.5 and the number of PISO loop to 3.

3.3.2. Flow past a forced oscillating cylinder

In order to compare our results with available data of literature, the cylinder is forced to oscillate in the vertical direction at a fixed amplitude ratio of $A = 0.25$ and with a frequency ratio of $F(= f_o/f_v) = 0.975$ (with f_o the frequency of the forced oscillation), following the work of [5].

The shedding frequency f_v is obtained from a preliminary flow simulation past a fixed cylinder at $Re = 500$.

Results favorably compare with literature, as shown on Table 3.2. With the finest grid, the predicted Strouhal number matches the value of Blackburn and Henderson [5] , and the drag coefficient is slightly overestimated with a difference of about 5%. A detailed description of the flow is also provided on Figure 3.4, with vorticity contours plotted at five instants spreading over half of the vortex shedding cycle. The comparison with the results of Blackburn and Henderson [5] provides good evidence that the spatial dynamics of the separation bubbles is well predicted by the new IBM solver. Starting from this established 2D shedding regime, the cylinder is set in motion.

Table 3.2.: : *Strouhal number and drag coefficient for the configuration of a fixed cylinder at $Re = 500$. Numerical data from literature are provided for comparison.*

		C_D	S_t
Present ($Re = 500$)	$\Delta x = \Delta y = 0.02D$	**1.547**	**0.225**
	$\Delta x = \Delta y = 0.01D$	**1.515**	**0.228**
Blackburn and Henderson [5]		1.445	0.228
Henderson [40]		1.445	-
Norberg [90]		-	0.205

A detailed description of the flow past the oscillating cylinder is provided on Figure 3.5 and shows a good agreement with results of Blackburn and Henderson [5] . The evolution of the lift coefficient as a function of the body displacement is shown over the 10 last periods of oscillations on Figure 3.6 and matches quite well the reference data of Blackburn and Henderson [5] .

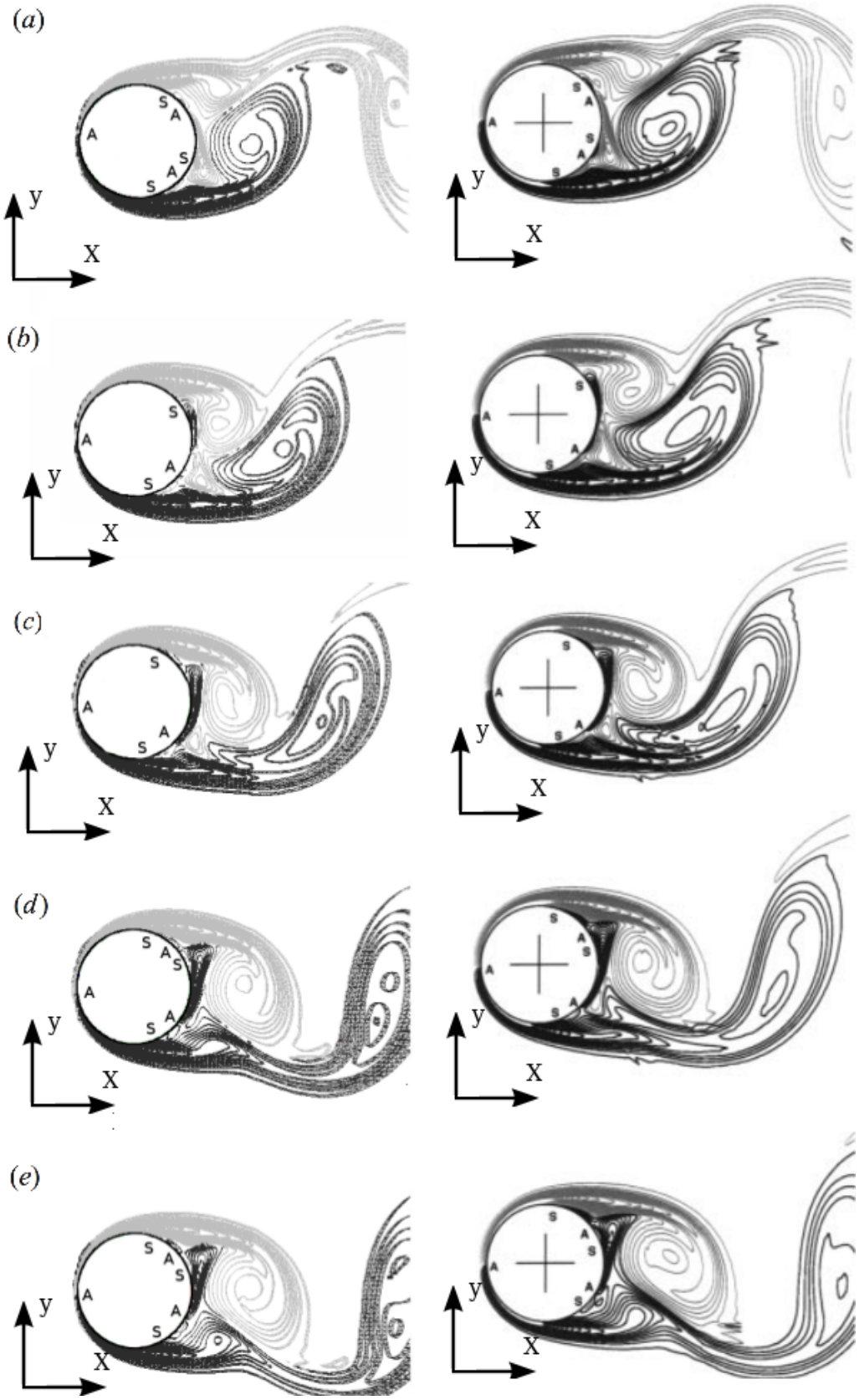


Figure 3.4.: : *Comparisons at five different instants (a to e) of the results of Blackburn and Henderson [5] (left column) and the present ones (right column). Instantaneous vorticity contours (black: positive values; grey: negative values) for the flow past a fixed cylinder at* $Re = 500$. *The attachment and separation points are labelled A and S, respectively.*

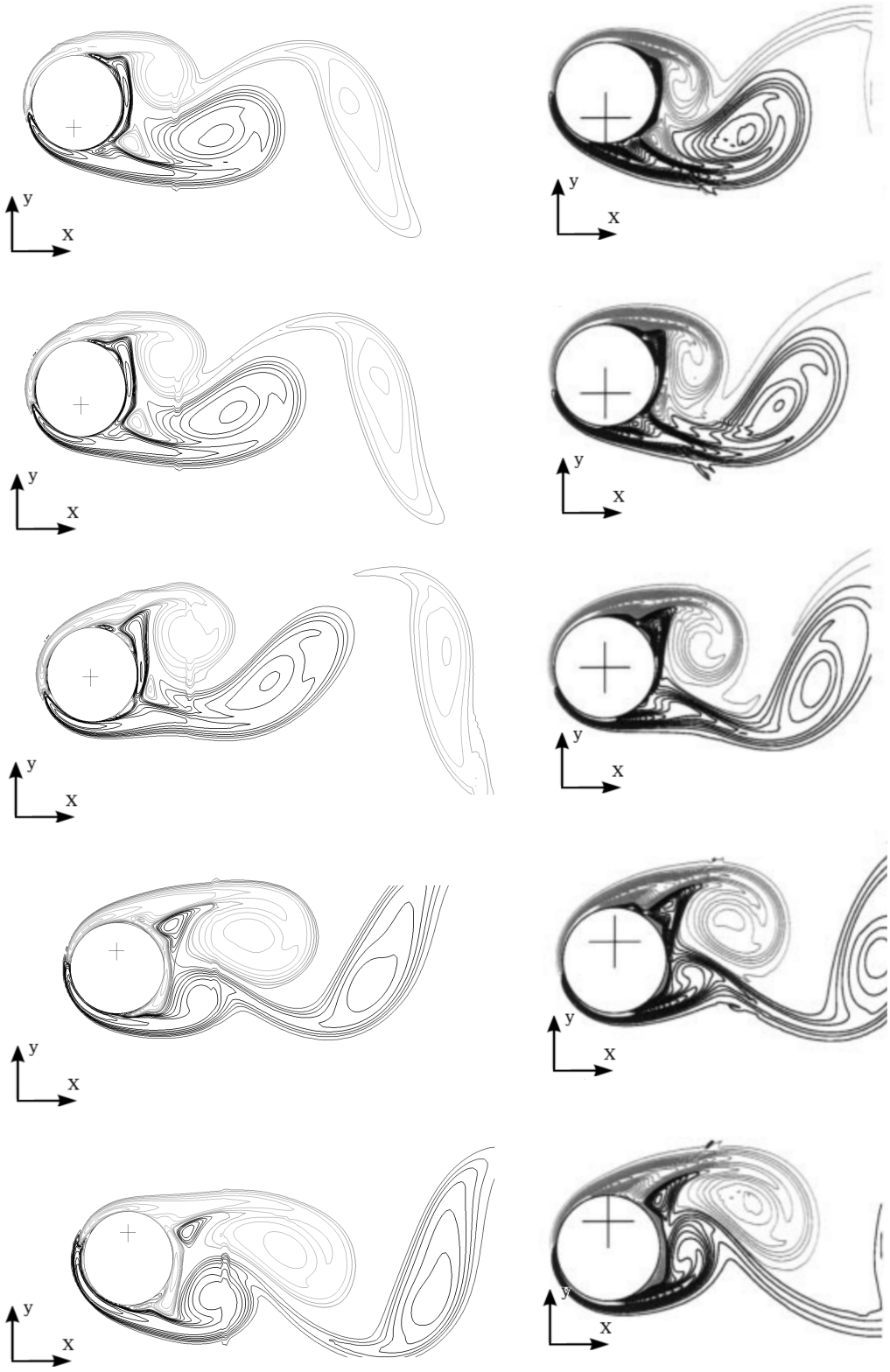


Figure 3.5.: *Comparisons at five different instants spreading over half of the shedding cycle of the present results (left columns) and the results of Blackburn and Henderson [5] (left column). Instantaneous vorticity contours (ranging from -1 (grey) to 1 (black)) for the flow past a oscillarting cylinder at* $Re = 500$.

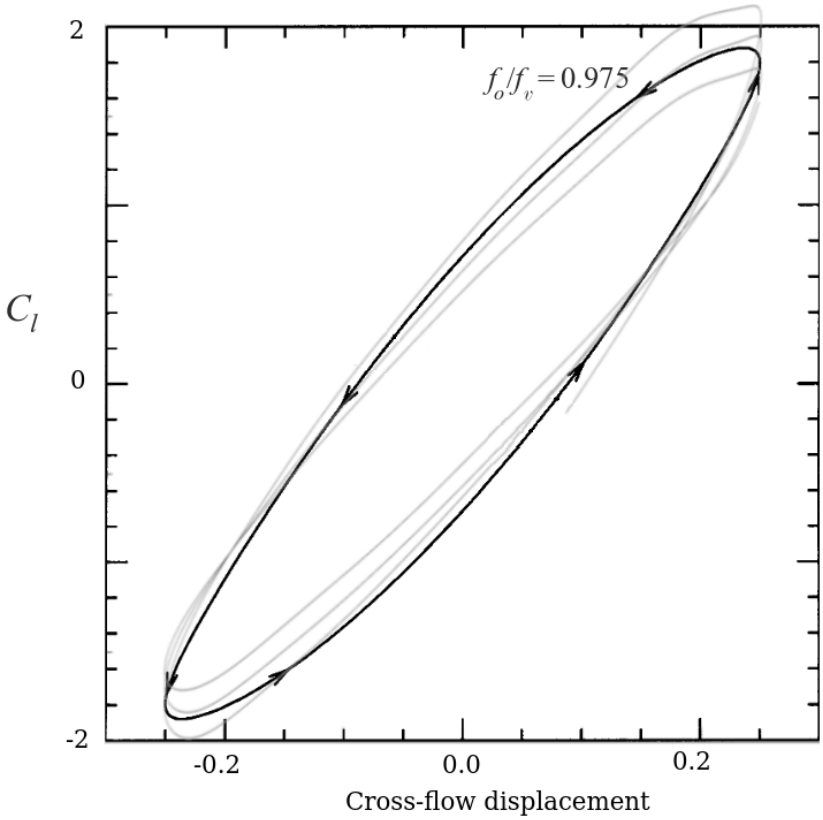


Figure 3.6.: *Lift coefficient C_L as a function of the cylinder displacement for the 2D flow past an oscillating cylinder at Re=500 : present results (grey line) vs. results obtained by Blackburn and Henderson [5] (black line).*

3.3.3. Flow past a freely moving cylinder: fluid/structure interactions

3.3.3.1. One degree of freedom

We consider here the flow past a moving circular cylinder at $Re = 100$ (see in Shiels et al. [117]). We are interested to emphasize VIV in the undamped ($B = 0$) oscillator system with a low non-dimensional mass $M = 2.5$, and a dimensionless spring constant of $K = 4.96$. At this Reynolds number, the flow and the structure are supposed to be in the lock-in regime, characterized by a large amplitude motion of the structure. Table 3.3 summarizes the results obtained together with data of reference from the literature. The mean drag $C_{D_{Mean}}$, the oscillation frequency f and the the maximum displacement of the structure y_{max}/D are rather well predicted, and are in good agreement with the data of Shiels et al. [117] and Shen and Lin [116] . The effective elasticity coefficient defined by Shiels et al. [117] as $k_{eff} = K - 4\pi^2 f^2 M$ is also in good agreements with the value of Shen and Lin [116] . As expected, the results computed using the Newmark and HHT schemes are globally closer to the reference results than the results computed using the Eulerian scheme, this latter being known to be less accurate. However, the maximum value of the lift coefficient is much larger than the value of reference.This overestimation is not yet fully explained and complementary numerical tests are still in progress. One of the issue when studying VIV is that this phenomenon is very sensitive to many variables. As a consequence, a small perturbation,

coming for instance from the discretization error when changing the numerical scheme or the code, can lead to rather different flow features and so to large changes on the values of the lift coefficient.

To complete the tests, simulations have been performed for various couples of parameters (K, M) corresponding to various effective elasticity coefficient k_{eff}. This parameter is used because it provides an unified scaling for an undamped system behaviour, compared to the reduced velocity usually used in the literature. That is to say that for any choices (K, M) couple which keep k_{eff} constant, a single response is consistent with the governing equation of motion at the contrary of the reduced velocity.

Simulations have been performed for $K_{eff} = -3.487$, $K_{eff} = 0.59$, $K_{eff} = 1.449$ and $K_{eff} = 1.763$, corresponding to the following pairs of (K, M)-parameters $(K = 0; M = 4)$, $(K = 4.74; M = 5)$, $(K = 4.96; M = 2.5)$ and $(K = 8.74; M = 5)$, respectively. Characteristic flow parameters of the simulations are plotted on Figure 3.8 as function of k_{eff}. With this parameter, instead of the reduced velocity, the simulations are found to predict the main trend of the parameters evolution as obtained in Ref. Shiels et al. [117] .

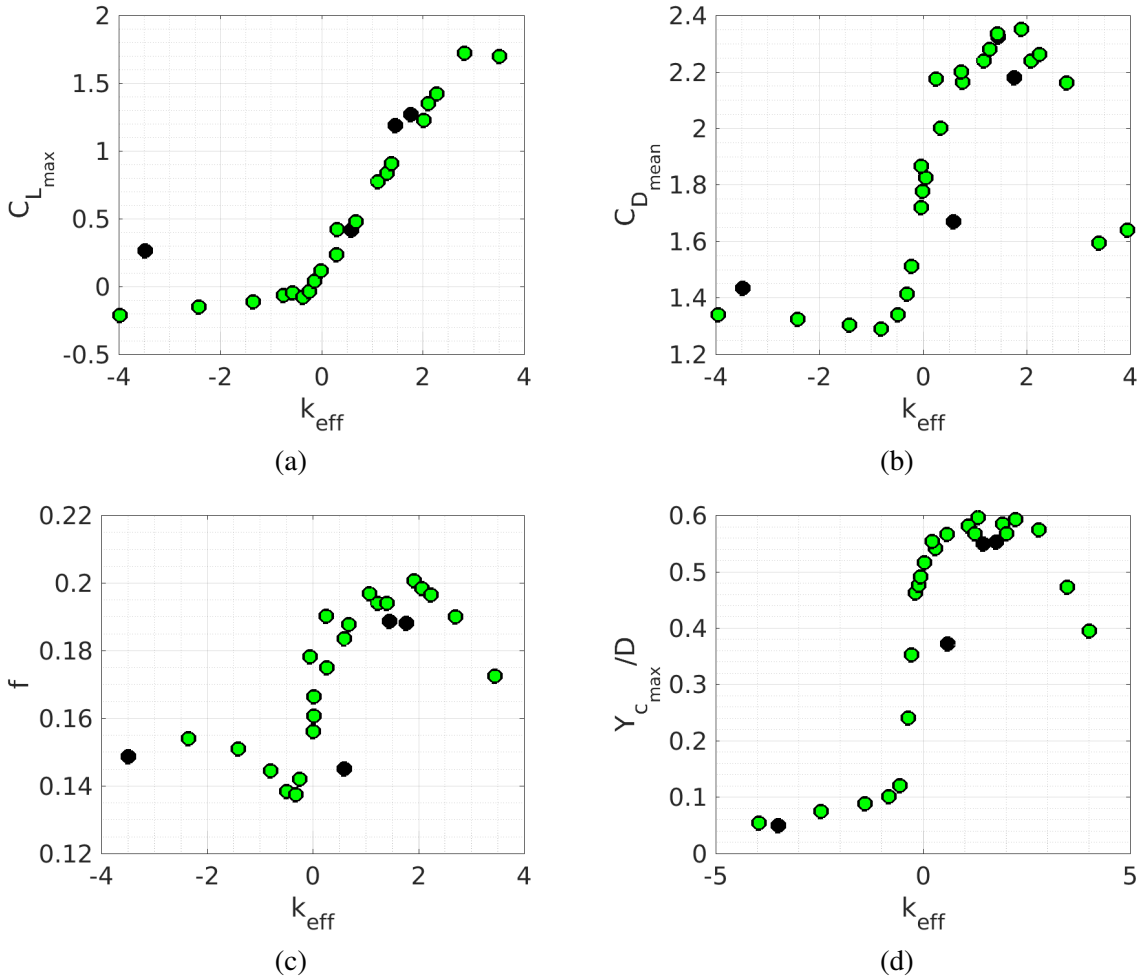


Figure 3.7.: *Maximum lift coefficients $C_{L_{max}}$ (a), mean drag $C_{D_{Mean}}$ (b), oscillation frequency f (c) and maximum vertical displacement y_{max}/D as function of the effective elasticity coefficient k_{eff}. (•) Present results, (•) results of Shiels et al. [117] . Flow past a freely moving cylinder at $Re = 100$.*

Table 3.3.: *Mean drag $C_{D_{mean}}$, maximum lift coefficient $C_{L_{max}}$, oscillation frequency f, maximum vertical displacement y_{max}/D and effective elasticity coefficient k_{eff}. Flow past a freely moving circular cylinder at $Re = 100$. Numerical and experimental data from the literature are provided for comparison.*

	$C_{D_{Mean}}$	$C_{L_{max}}$	f	y_{max}/D	k_{eff}
Present (Euler scheme)	**2.0779**	**1.2557**	**0.175**	**0.5397**	**1.9374**
Present (Newmark scheme)	**2.324**	**1.190**	**0.1886**	**0.5462**	**1.4493**
Present (HHT scheme)	**2.326**	**1.197**	**0.1883**	**0.5493**	**1.4493**
Shiels et al. [117]	2.22	0.77	0.196	0.58	1.17
Shen and Lin [116]	2.15	0.83	0.190	0.57	1.3970

3.3.3.2. Two degrees of freedom

Simulations at $Re = 150$

The cylinder is now able to move in two directions. The flow is simulated for Reynolds number $Re = 150$, a low non-dimensional mass $M = 2$, and for different effective elasticity coefficient k_{eff} of an undamped system ($B = 0$) (see in Zhou et al. [151]). The results shown on Figure 3.8 are in overall agreement with the data of Zhou et al. [151] .

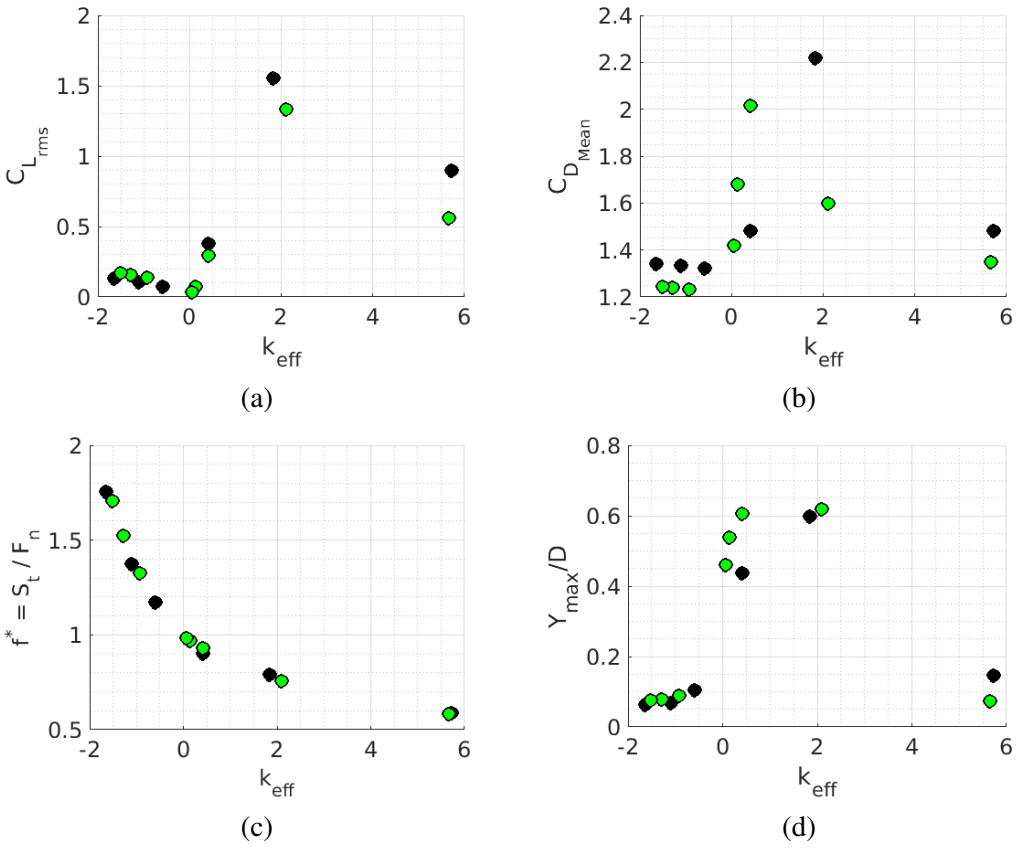


Figure 3.8.: *R.m.s. value of the lift signal (a), mean drag $C_{D_{mean}}$ (b), oscillation frequency f (c) and maximum vertical displacement y_{max}/D as function of the effective rigidity k_{eff}. (•) Present results, (•) results of Zhou et al. [151] . Flow past a freely moving cylinder at $Re = 150$.*

The peak values of the r.m.s. lift coefficient (Figure 3.8a), mean drag (Figure 3.8b) the dimensionless frequency signal $f* = \frac{S_t}{f_N}$ (Figure 3.8c), and the maximum vertical displacement y_{max}/D as function of the effective rigidity k_{eff} are close to the reference data.

Figure 3.9 shows the trajectories of the structure centerfor four values of the reduced frequency. The direction along the trajectories at the top position is marked by "C." and "C.C." meaning clockwise and counter-clockwise. Present IBM simulations and simulations of Zhou et al. [151] show the same behaviour and the same trajectory direction for each studied frequency. Effective rigidity k_{eff} coefficient is provided for each simulation. From figure 3.10 which present dimensionless frequency signal $f* = \frac{S_t}{f_N}$ as function of the reduced velocity U_r it can be seen that at this frequency the simulation is aligned with the results at $U_r = 8$ and $U_r = 10$ and doesn't form a lock-in region about $f* \sim 1$ with the result at $U_r = 6$. The simulations show the same path as the simulation of the reference excepted for $U_r = 7$ where the transition from lock-in regime to vortex shedding oscillations (see in Ref. Zhou et al. [151] and Figure Figure **??**). Our simulation predict a vortex shedding oscillations whereas the simulation of the reference predict a lock-in regime.

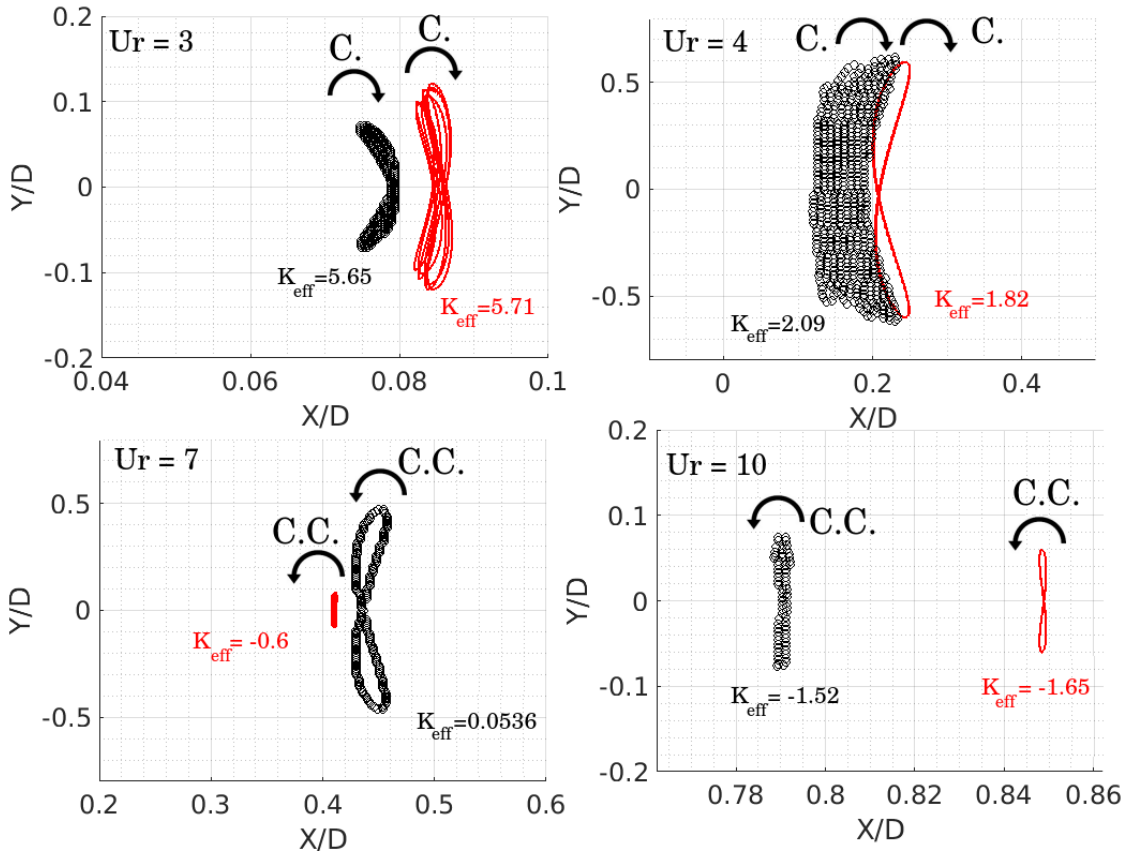


Figure 3.9.: *Path of the motion of the structure center for four values of reduced velocity U_r. (–) Present results, (∘) results of Zhou et al. [151] . "C" means clockwise and "CC"counterclockwise direction.*

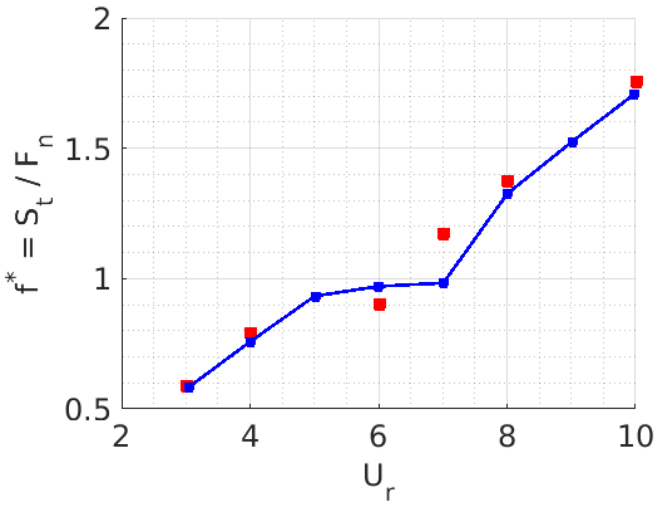


Figure 3.10.: *Dimensionless frequency signal* $f* = \frac{S_t}{f_N}$ *as function of the Reduced velocity* U_r. (■) *Present results, (–) Results of [151]*

Simulations when varying Reynolds number from $Re = 75$ **to** $Re = 130$

These simulations are proposed in Ref. [104]). The low non-dimensional mass is fixed at $M = 7.85398$, and the natural structural frequency of the oscillator f_N corresponds to the frequency of the vortex shedding past a fixed cylinder at $Re = 100$. This means that the natural frequency follows the expression $F_N = \frac{100 S_{t_{100}}}{Re}$. In this range of parameters, the flow is expected to be laminar and 2D. According to the literature, an hysteric cycle is expected for Reynolds numbers in a range $[129, 137.3]$, in which the flow can be leaded by a lock-in mechanism or oscillations related to vortex shedding for the same value of Reynolds number.

Present results are shown on Figure 3.11. The IBM solver predicts the general trend of the parameters evolution when varying Reynolds number. Peak values for the vertical displacement, the Strouhal number and the C_{D_rms} are well predicted. However, as mentionned before, the worst prediction is given for the peak value of the C_{L_rms}, with a value equal to twice the value of reference obtained in [104]. It is also noticeable that all peak values occur at a slightly lower Reynolds number ($Re = 82$) than in the reference ($Re = 86$).

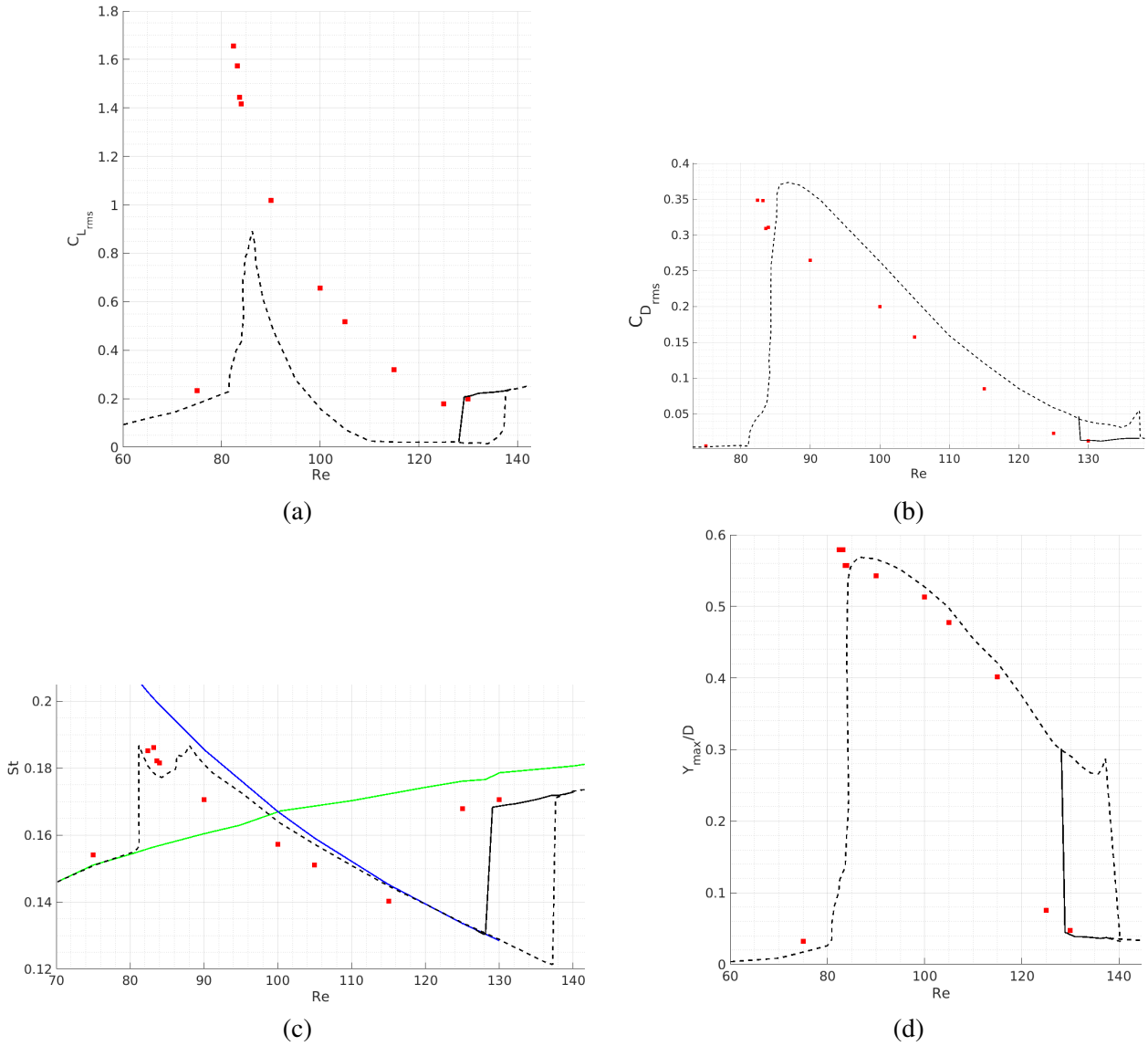


Figure 3.11.: *R.m.s. values of the lift coefficient (a) and of the drag (b), dimensionless oscillation frequency St (c), and maximum vertical displacement y_{max}/D as function of Reynolds number. (■) Present results, (–) lower branch results of [104] , (- -) upper branch results of [104]. (–)evolution of the dimensionless oscillation frequency of the fixed case. (–) evolution of the dimensionless natural frequency of the structure.*

Regarding the vortex shedding frequency, the lock-in regime occurs as expected at a frequency close to the non dimensional natural frequency of the structure (blue curve on Figure 3.11c), but in the range of Reynolds numbers slightly shitted to low Reynolds numbers, compared with results of reference obtained in Ref. [104]. At $Re = 125$ the solution bifurcates to the vortex shedding oscillations regime, with a Strouhal number close to the value obtained on the equivalent fixed case. The hysteric cycle, related to the sub- or suprecritcal nature of this bifurcation has not been investigated in the present computations.

The topology of the flow solutions are investigated showing the instantaneous vorticity fields on Figure 3.12. As expected from literature in this range of Reynolds numbers([149]), the cylinder wakes exhibit the two flow patterns, called 2S and C(2S), for classical von Karman vortex street

and coalescing wake, respectively. The (2S)→(C(2S)) and the (C(2S))→(2S) transitions are rather well predicted by the IBM solver at critical Reynolds numbers in a range $75, 82.4$ and in the range $90, 105$, respectively. In Ref. [104], the same transitions are obtained at critical Reynolds numbers in a range $84.2, 86, 82.4$ and in the range $100, 110$, respectively. The transition between different regime are observed between vortex shedding oscillations and lock-in regimes.

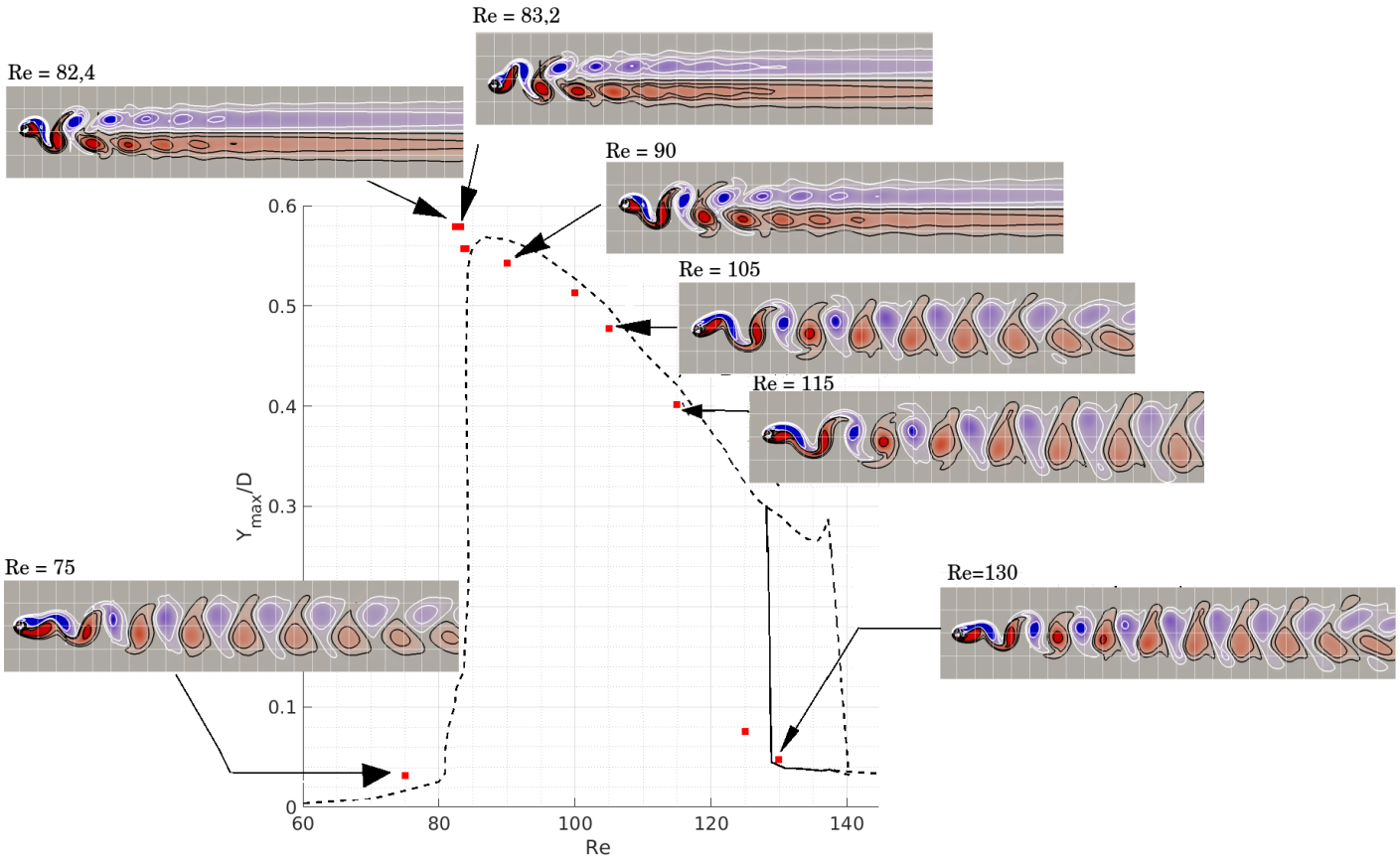


Figure 3.12.: *Maximum values of the vetical displacement as function of the Reynolds number and corresponding characteristicwake flow patterns. (■) Present results, (–) results of [104] lower branch, (- -) results of [104] upper branch*

Chapter 4

Simulations of turbulent flows

In this chapter we extend the method presented in the former chapter to turbulence modelling. Hybrid models have received attention in the last decade for industrial studies. Native OpenFOAM DDES and IDDES models are incorporated into the new IBM PISO algorithm. As a first attempt to model the thin boundary layers around the obstacles without resolving the viscous laminar near-wall region, wall-functions are integrated, and the IBM modified accordingly. The wall distance, which is not straightforwardly available when using IBM, is derived from the location of the Lagrangian points. Simulations of turbulent flows past academic configurations is used to validate the methodology.

4.1. The available turbulence models in OpenFOAM

To allow a large diffusion of the work into the OpenFOAM users community, OpenFOAM turbulence models have been considered. More specifically hybrid models based on DDES and IDDES have been considered in this work. The aim of this section is to present the main characteristics of these models. For further details, the reader is invited to refer to the review paper of Spalart [124].

4.1.1. The Delayed Detached Eddy simulation model (DDES)

As mentioned in the Introduction, the high numerical cost associated with the use of LES models to simulate boundary-layer flows has led to the development of hybrid models. These models attempt to combine both aspects of RANS and LES methodologies. In openFOAM, the model is the DES of Spalart et al. [126], later extended to DDES. The DES model combines, a RANS model within the regions near solid boundaries where the turbulent length scale is smaller than the grid size, with a LES model as the turbulent length scale exceeds the grid size. It is here based on the RANS Spalart-Allmaras model [123], in which the distance function d, defined as a turbulence length scaleis a RANS model close to walls, otherwise a Smagorinsky-like LES closure is employed.

In particular, the Spalart Allmaras model was specifically designed fo aerospace applications involving wall-bounded flows, and has been shown to provide good results for boundary-layers with adverse pressure gradients.

The model is relatively simple and solves the following modeled transport equation for a viscosity-like variable $\tilde{\nu}$ proportional to the kinetic turbulent eddy viscosity ν_t:

$$\frac{\partial \tilde{\nu}}{\partial t}+u_j\frac{\partial \tilde{\nu}}{\partial x_j}=c_{b1}(1-f_{t2})\tilde{S}\tilde{\nu}-[c_{w1}f_w-\frac{c_{b1}}{\kappa^2}f_{t2}](\frac{\tilde{\nu}}{d})^2+\frac{1}{\sigma}[\frac{\partial}{\partial x_j}((\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j})+c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}] \quad (4.1)$$

with :

$$\chi = \frac{\tilde{\nu}}{\nu} \tag{4.2}$$

$$\tilde{S} = f_{v3}\Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \tag{4.3}$$

$$\Omega = \sqrt{2W_{ij}W_{ij}} \tag{4.4}$$

$$W_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}) \tag{4.5}$$

$$f_{v2} = \frac{1}{(1 + frac\chi c_{v2})^3} \tag{4.6}$$

$$f_{v3} = \frac{(1 + \chi f_{v1})(1 - f_{v2})}{(\chi)} \tag{4.7}$$

$$f_w = g[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}]^{\frac{1}{6}} \tag{4.8}$$

$$g = r + c_{w2}(r^6 - r) \tag{4.9}$$

$$r = \min[\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10] \tag{4.10}$$

$$f_{t2} = c_{t3} e^{-c_{t4}\chi^2} \tag{4.11}$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b3}}{\sigma} \tag{4.12}$$

and where the turbulent eddy viscosity is derived from $\mu_t = \rho\tilde{\nu} f_{v1}$, with

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3} \tag{4.13}$$

The constants are fixed to : $c_{v2} = 5$, $c_{b1} = 0.1355$, $\sigma = \frac{2}{3}$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$, $c_{t3} = 1.2$, $c_{t4} = 0.5$

The boundary conditions are defined as :

$$\nu_{t,wall} = 0 \tag{4.14}$$

$$\tilde{\nu}_{farfield} = 3\nu_\infty : to : 5\nu_\infty \tag{4.15}$$

In DES, the distance function is modified as :

$$\tilde{d} = \min[d, C_{DES}\Delta] \tag{4.16}$$

where C_{DES} is a constant taken equal to 0.65, and Δ is the largest dimension of the local grid cell. For structured grids, Δ is the largest grid spacing over all three directions.

This approach has been later improved by Spalart et al. [125] for turbulent flows with thick boundary-layers and mild separation regions. For these flows, the wall distance can be much larger than the cell dimension (and then the LES mode is activated), but the cell is still within the boundary layer. This can occur when the mesh is gradually refined, and the dimensions of the cell parallel to

the wall becomes smaller than the wall distance. The LES, not adapted to solve the equations on the stretched grid of the boundary layer (suited for the RANS model) then reduces the eddy viscosity below the RANS level, and solves the Reynolds stresses deriving from velocity fluctuations. These stresses reduce the skin friction, which can lead to a too early separation of the flow. In order to make the model less dependent on the mesh, Spalart et al. [125] proposed a new DES model, called DDES for Delayed DES, which rely on a modification of the DES length scale. The main idea of DDES is to include the molecular and turbulent viscosity information into the switching mechanism to delay this switching in boundary layers.

The new DES length scale reads :

$$r_d = \frac{\nu_t + \nu}{\sqrt{\partial_i u_j \partial_i u_j} \kappa^2 d^2} \tag{4.17}$$

$$f_d = 1 - \tanh((8r_d)^3) \tag{4.18}$$

$$\tilde{d} = d - f_d \max(0, d - C_{DES}\Delta) \tag{4.19}$$

This function has been calibrated on a flat-plate boundar- layer (see in Spalart et al. Spalart et al. [125]) in order to have a RANS behaviour in the boundary layer and a LES behaviour elsewhere.

The calculation of d in the IBM framework will be detailed thereafter.

4.1.2. The improved DDES model (IDDES)

The Improved DDES model developped by Shur et al. [118] is also proposed in OpenFOAM. It offers a way to combine the capabilities of the wall modeled LES (WMLES) and the DDES defined above with the goal to guarantee a correct matching between the modelled and the simulated log-layer.

Central to this model is a new definition of Δ, which includes explicitly a wall distance dependency (through wall-normal direction) and not only the local characteristics of the cells like in DDES. The modification tends to depress Δ near the wall, and gives it a steep variations, which induces instabilities, increasing the resolved Reynolds stress. Other features of the model are related with the introduction of new empirical functions which address log-layer mistmatch, and the bridge between wall-resolved and wall-modeled DES.

The new functions read:

$$\tilde{d} = \tilde{f}_d(1 + f_e)d + (1 - \tilde{f}_d)C_{DES}\Delta \tag{4.20}$$

$$\Delta = \min\{\max[C_w \max(\Delta x, \Delta y, \Delta z), C_w d, \max(\Delta \mathbf{x}_f \cdot \mathbf{n})], \max(\Delta x, \Delta y, \Delta z)\} \tag{4.21}$$

$$\tilde{f}_d = \max(1 - f_d, f_B) \tag{4.22}$$

$$f_d = 1 - \tanh((8r_d)^3) \tag{4.23}$$

with $C_w = 0.15$ is a constant of the model (based on wall-resolved LES of the channel flow with the Smagorinsky SGS model), $\mathbf{n}(n_x, n_y, n_z)$ is the normal to the wall, $\Delta \mathbf{x}_f$ is the vector composed by the distances between each surface center of the cell, and f_B a blending function which varies from 0 to 1 in order to switch between RANS ($f_B = 1.0$) and LES ($f_B = 0$).

4.2. IBM and turbulence models implementation

4.2.1. The IBM PISO solver for the Reynolds-averaged Navier-Stokes equations

The 4 steps procedure of PISO algorithm introduced in Chapter 2 are modified at each time step n as follows ($\overline{\cdot}$ denotes averaged quantities):

i. **Predictor step:**

A. An estimate averaged velocity $\widehat{\overline{\mathbf{u}}}$ is obtained by solving the Reynolds-averaged Navier–Stokes equations (including the Reynolds stress tensor τ, see chapter1) but without any IBM force term, and using the averaged pressure $\overline{p}$ computed at the previous time step $n-1$:

$$\frac{\partial \widehat{\overline{\mathbf{u}}}}{\partial t} + \nabla \cdot (\widehat{\overline{\mathbf{u}}}\, \widehat{\overline{\mathbf{u}}}) = -\nabla \overline{p} + \frac{1}{Re}\nabla^2 \widehat{\overline{\mathbf{u}}} - \nabla \cdot \tau \tag{4.24}$$

B. The calculation of the IBM force $\mathbf{F}_s$ is detailed in 2.2.2. It is calculated on the Lagrangian markers using $\widehat{\overline{\mathbf{u}}}$ (Eq. 4.24), and its values spread on the Eulerian mesh to calculate $\mathbf{f}$ (Eq. 2.20).

C. A new averaged velocity $\overline{\mathbf{u}}^{\star,1}$ is calculated from the Reynolds-averaged equations accounting now the immersed boundary force term f calculated using the estimate averaged velocity $\widehat{\overline{\mathbf{u}}}$:

$$\frac{\partial \overline{\mathbf{u}}^{\star,1}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}^{\star,1} \overline{\mathbf{u}}^{\star,1}) = -\nabla \overline{p} + \frac{1}{Re}\nabla^2 \overline{\mathbf{u}}^{\star,1} + f(\widehat{\overline{\mathbf{u}}}) - \nabla \cdot \tau \tag{4.25}$$

$\overline{\mathbf{u}}^{\star,1}$ is the guess value of the averaged velocity in the iterative PISO loop

ii. **PISO loop:**
For the sub-iteration $m = 1$ to $M-1$, and up to convergence:

A. At each sub-iteration, an averaged pressure field $p^{\star,m}$ is calculated from the following Poisson equation :

$$\nabla^2 \overline{p}^{\star,m} = -\nabla \cdot (\overline{\mathbf{u}}^{\star,m} \nabla \overline{\mathbf{u}}^{\star,m}) + \nabla \cdot f(\widehat{\overline{\mathbf{u}}}) \tag{4.26}$$

B. The velocity field is thus corrected using:

$$\overline{\mathbf{u}}^{\star,m+1} = g\left(\overline{\mathbf{u}}^{\star,m},\ \nabla \overline{p}^{\star,m},\ f(\widehat{\overline{\mathbf{u}}})\right) \tag{4.27}$$

where g as well as all discretized operators used in the algorithm are defined in Annex B.

iii. **End of the PISO loop:**
The averaged velocity and pressure are finaly updated at time $n+1$:

$$\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^{\star,M-1} \tag{4.28}$$

$$\overline{p}^{n+1} = \overline{p}^{\star,M-1} \tag{4.29}$$

iv. **Final step:**
The Reynolds stress tensor, τ, is recomputed according to the new value of the averaged velocity, $\overline{\mathbf{u}}^{n+1}$. This is done using the Boussinesq hypothesis (Eq. 1.48), where ν_t is derived from the Spallart-Allmaras transport equation for $\tilde{\nu}$ (Eq. 4.1).

4.2.2. IBM - DDES and IBM - IDDES implementation

The DES model introduced in Sec. 4.1.1 requires the calculation of the distance to the wall, d, for each Eulerian grid point of the computational domain. In IBM, d becomes d_{IBM}.

4.2.2.1. Estimate of the wall distance d_{IBM}

In a flow configuration where the obstacle is fixed, this calculation can be done once at the beginning of the simulation (before the time loop). For each Eulerian cell j, d_{IBM_j} is defined as a minimum distance between this point and all Lagrangian markers S as:

$$d_{IBM_j} = \min_S |\mathbf{x}_j - \mathbf{X}_S| \tag{4.30}$$

When the obstacle is moving, this would have to be done at each time step. To save CPU time, we slightly improved the procedure by limiting this calculation to a set of Eulerian points within a flow region, denoted (D_x, D_y, D_z), and defined by the user around the obstacle, as shown on Figure 4.1. For the Eulerian points located outside this flow region, the distance is approximated by the distance to the center of the structure X_0, avoiding thus the loop on the set of Lagrangian markers. The same procedure is applied in DDES.

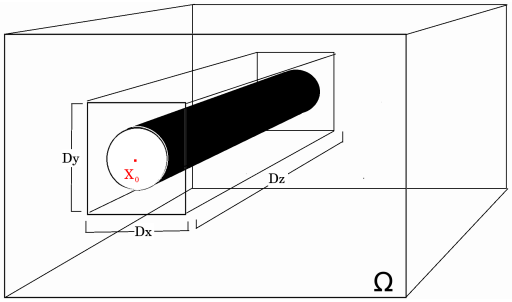


Figure 4.1.: *Sketch of the zone around the obstacle involved in the calculation of the distance to the wall..*

4.2.2.2. IDDES implementation

In IDDES, the normal to the wall has also to be defined at each Eulerian point (Sec.4.1.2). To do so in IBM, a connectivity table is created, and the normal n_S is calculated on each Lagrangian marker S such that:

$$\mathbf{n}_S = \sum_s^N (\mathbf{X}_{s+2} - \mathbf{X}_S) \otimes (\mathbf{X}_{s+1} - \mathbf{X}_S) \tag{4.31}$$

with N the number of neighboring points of the Lagrangian marker S, and $\otimes$ defines the cross product. The calculation of different normals to the wall is illustrated on Figure 4.2.

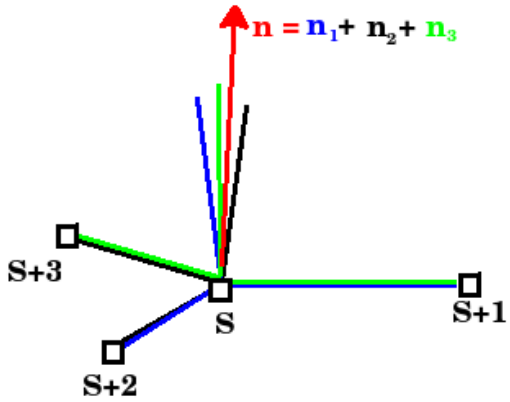


Figure 4.2.: *Normal to the wall.*

To estimate this normal on the Eulerian point, a loop over all Lagrangian markers is used to find the closest one. The normal is finally normalized, and incorporated into the computation of Δ (Eq. 4.21). As in the computation of d, this loop is limited to the Eulerian points located within the flow region (D_x, D_y, D_z). Outside, we have $\max(\Delta \mathbf{x}_f \cdot \mathbf{n}) << C_w d$, and the computation of the normal is not required.

4.2.3. Wall function implementation

The Spalart-Allmaras model implemented in OpenFOAM is a high-Reynolds number model that requires to solve the near-wall boundary layer developping around the obstacle. That requires small values of y^+ , typically $y^+ \lesssim 4$, leading to a high concentration of grid cells in the vicinity of the wall, at moderate to high Reynolds numbers. For complex geometries, that leads to design meshes as resolved as a for a body-fitted approach (see an example on Figure 4.3), without however reaching the accuracy of this approach. Furthermore, configurations with moving structures become easily unaffordable in terms of cpu cost.

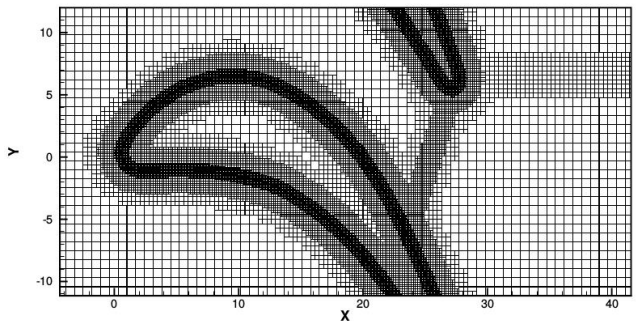


Figure 4.3.: *Example of refined mesh around a VKI-LS59 turbine-rotor cascade using Immersed Boundary Method (see Ref [17])*

In this context, we decided, as a first attempt, to implement a wall function technique (see Chapter

1), based on the law of the wall proposed by Launder and Spalding [61] . However, this approach being not suited to take in account pressure gradient effect, it will not be used in the configurations of flows around solid obstacles. It is presented here as a proof of concept showing the capacity of the method to deal with such a technique.

Using each Lagrangian marker and its associated normal (as shown on Figure 4.4), a point is defined inside the log-layer thanks to an offset (defined by the user). This new set of points will be used to interpolate variables, and thus to impose the boundary conditions. We also define 2 tangents in order to implement the wall function. In particular, in 3D, we need to define two tangents at each Lagrangian marker S. The first one, $\mathbf{t_1}$, is defined as the distance vector between the Lagrangian marker S and its first neighbourg in the connectivity table described in Sec. 4.2.2. The second one, $\mathbf{t_2}$, is defined as the cross product between the normal $\mathbf{n}$ and $\mathbf{t_1}$.

$$\mathbf{t}_{1,S} = \mathbf{X}_{S+1} - \mathbf{X}_S \tag{4.32}$$

$$\mathbf{t}_{2,S} = \mathbf{n}_S \otimes \mathbf{t}_{1,S} \tag{4.33}$$

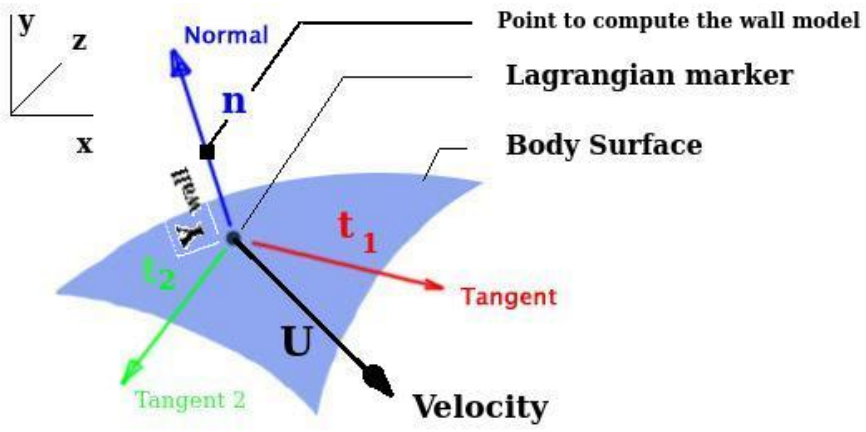


Figure 4.4.: *Sketch showing the normal, n, and the tangents, t_1 and t_2, at the surface at the Lagrangian marker.*

The PISO algorithm is modified to accomodate the wall function in the computation. It reads for each time step n as :

i. **Predictor step:**

A. An estimate averaged velocity $\widehat{\overline{\mathbf{u}}}$ is obtained by solving the Reynolds-averaged equations without any force term, and using the averaged pressure $\overline{p}$ computed at the previous time step $n-1$:

$$\frac{\partial \widehat{\overline{\mathbf{u}}}}{\partial t} + \nabla \cdot (\widehat{\overline{\mathbf{u}}}\widehat{\overline{\mathbf{u}}}) = -\nabla \overline{p} + \frac{1}{Re}\nabla^2 \widehat{\overline{\mathbf{u}}} - \nabla \cdot \tau \tag{4.34}$$

B. Predictor step for the wall function:

C. An interpolate velocity on the s^{th} Lagrangian markers inside the log layer is calculated thanks to the interpolation function 2.8 $\mathbf{U}_{log,s} = \mathcal{I}[\overline{\mathbf{u}}^n]_s$

D. We define the following gradients at the wall :

$$\frac{dU_{t1}}{dx_{\mathbf{n}}} = \frac{\mathbf{U}_{log,s}.\mathbf{t}_{1,s} - U_{wall,s}.\mathbf{t}_{1,S}}{\parallel (\mathbf{X}_s + y_{wall} \cdot \mathbf{n}_s) - \mathbf{X}_s \parallel} \tag{4.35}$$

$$\frac{dU_{t2}}{dx_{\mathbf{n}}} = \frac{\mathbf{U}_{log,s}.\mathbf{t}_{2,s} - U_{wall,s}.\mathbf{t}_{2,S}}{\parallel (\mathbf{X}_s + y_{wall} \cdot \mathbf{n}_s) - \mathbf{X}_s \parallel} \tag{4.36}$$

with:

- $\mathbf{U}_{log,s}$ and $U_{wall,s}$ the velocities at the s^{th} Lagrangian marker inside the log-layer and on the surface of the structure, respectively
- y_{wall} is the cell dimension in the normal direction $(\mathbf{n}_s \cdot \mathbf{\Delta})$, and corresponds to the distance from the wall where the model will be computed. This point must be located within the log-layer region ($y^+ \geq 30$) and thus the grid size is chosen accordingly.

E. The wall shear stress τ_{wall} and friction velocity are computed as:

$$\tau_{wall} = \nu * \sqrt{(\frac{dU_{t1}}{dx_{\mathbf{n}}})^2 + (\frac{dU_{t2}}{dx_{\mathbf{n}}})^2} \tag{4.37}$$

$$U_\tau = \sqrt{\frac{\tau}{\rho}} \tag{4.38}$$

F. Then the predicted value of $\widehat{y}^+ = \frac{y_{wall} * U_{\tau_{wall}}}{\nu}$ is computed

G. Corrector step for the wall function, for the sub-iterations $a = 1$ to A, and up to convergence:

H. τ^a_{wall} is re-evaluated thanks to the $y^{+,a}$ value is computed assuming that the point used to compute the wall model is located inside the log-layer thanks to the following formula:

$$\tau_{wall} = \frac{\kappa * \mathbf{U}_{log,s}}{log(E * y^+)} \tag{4.39}$$

where :

- $\kappa = 0.41$ is the Von Kármán constant.
- $E = e^{\kappa * B}$ is a constant with $B \simeq 5$.

I. Then the value of $y^{+,a} = \frac{y_{wall} * U_{\tau^a_{wall}}}{\nu}$ is updated

J. The calculation of the IBM force $\mathbf{F}_s$ is then done using the following expression:

$$\mathbf{F}^n_s = \frac{\mathbf{U}^d_s - \mathcal{I}[\overline{\hat{\mathbf{u}}]_s}}{\Delta t} \cdot \mathbf{n}_s + \frac{\tau^A_{wall}}{\parallel (\mathbf{X}_s + y_{wall} \cdot \mathbf{n}_s) - \mathbf{X}_s \parallel} \cdot \frac{\mathbf{U}_{log,s}.\mathbf{t}_{1,s} + \mathbf{U}_{log,s}.\mathbf{t}_{2,s}}{\parallel \mathbf{U}_{log,s}.\mathbf{t}_{1,s} + \mathbf{U}_{log,s}.\mathbf{t}_{2,s} \parallel} \tag{4.40}$$

K. The calculation of the IBM force is spread on the Eulerian mesh to calculate $\mathbf{f}$ (Eq. 2.20).

L. A new averaged velocity $\overline{\mathbf{u}}^{\star,1}$ is calculated from the Reynolds averaged equations accounting now the immersed boundary force term f:

$$\frac{\partial \overline{\mathbf{u}}^{\star,1}}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}^{\star,1} \overline{\mathbf{u}}^{\star,1}) = -\nabla \overline{p} + \frac{1}{Re} \nabla^2 \overline{\mathbf{u}}^{\star,1} + f(\widehat{\overline{\mathbf{u}}}) - \nabla \cdot \tau \tag{4.41}$$

$\overline{\mathbf{u}}^{\star,1}$ is the guess value of the velocity in the iterative PISO loop

ii. **PISO loop:**
For the sub-iteration $m = 1$ to $M - 1$, and up to convergence:

A. At each sub-iteration, a pressure field $p^{\star,m}$ is calculated from the following Poisson equation :

$$\nabla^2 \overline{p}^{\star,m} = -\nabla \cdot (\overline{\mathbf{u}}^{\star,m} \nabla \overline{\mathbf{u}}^{\star,m}) + \nabla \cdot f(\widehat{\overline{\mathbf{u}}}) \tag{4.42}$$

B. The velocity field is thus corrected using:

$$\overline{\mathbf{u}}^{\star,m+1} = g\left(\overline{\mathbf{u}}^{\star,m},\, \nabla \overline{p}^{\star,m},\, f(\widehat{\overline{\mathbf{u}}})\right) \tag{4.43}$$

where g as well as all discretized operators used in the algorithm are defined in Annex B.

iii. **End of the PISO loop:**
The velocity and the pressure are finaly updated at time $n + 1$:

$$\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^{\star,M-1} \tag{4.44}$$

$$\overline{p}^{n+1} = \overline{p}^{\star,M-1} \tag{4.45}$$

iv. For the points inside the log layer we correct the turbulent kinematic viscosity as following:

$$\nu_t = \frac{\tau^A_{wall}}{\frac{\partial u}{\partial y}} = \kappa y_{wall} \sqrt{\tau^A_{wall}} \tag{4.46}$$

v. **Final step:**

The Reynolds stress tensor, τ, is recomputed according to the new value of the averaged velocity, $\overline{\mathbf{u}}^{n+1}$. This is done using the Boussinesq hypothesis (Eq. 1.48), where ν_t is derived from the Spallart-Allmaras transport equation for $\tilde{\nu}$ (Eq. 4.1).

4.3. Validation test-cases

The numerical methodology presented above is validated on well documented academic test cases.

4.3.1. Near-wall resolved simulations of the turbulent flow past a fixed cylinder

We consider the flow around a circular cylinder in the subcritical regime at Reynolds number $Re = 3900$. This flow configuration corresponds to the well-documented ERCOFTAC benchmark as detailed in Breuer [7] . The von Karman vortex street at this Reynolds number already exhibits most of the characteristic features of industrial applications. Even though this test case is defined by a simple geometry, it is fully three-dimensional and unsteady, including transition regions to turbulence as well as flow separations along the sidewall. Therefore, it is identified as a relevant test case for the assessment of the IBM solver to perform simulations of complex turbulent flows. At this moderate Reynolds number, the flow is subcritical i.e., the boundary layers at the cylinder exhibit laminar separation and the transition takes place in the free shear layers. Therefore any DES model, which works in RANS mode in the near wall region, is expected to provide reliable results. Here, the Delayed Detached Eddy Simulation with the OpenFOAM implementation presented above is used.

4.3.1.1. Computational details

The center of the cylinder is the origin of the domain at $(0, 0)$. The dimensions of the computational domain are $[5D, 15D] \times [-10D, 10D] \times [-1.57D, 1.57D]$ in the streamwise (x), vertical (y) and spanwise (z) directions, Figure 4.5. The flow periodicity was assumed to be in the spanwise direction. Inflow, outflow, upper and lower sides of the domain are defined as shown Figure 2.1.

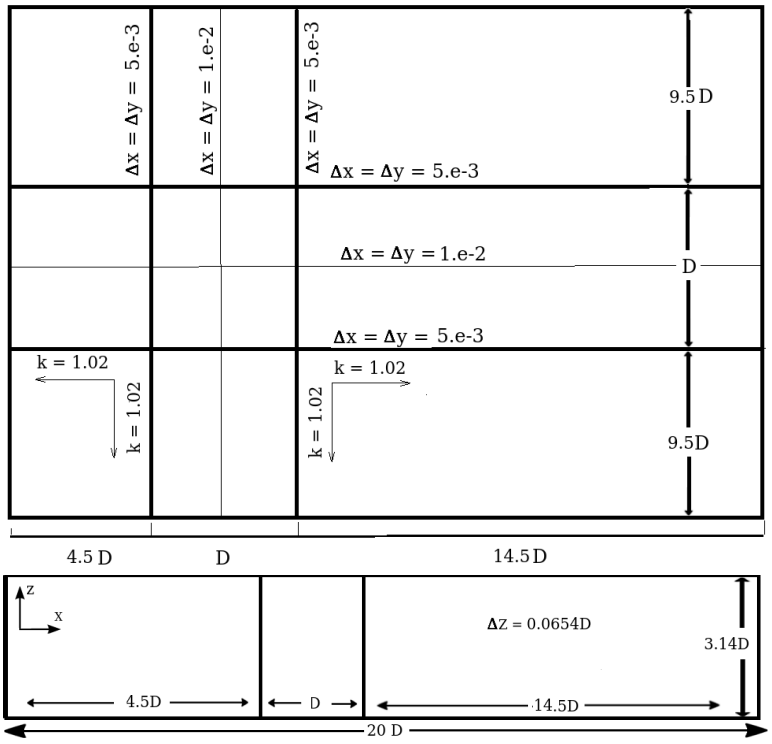


Figure 4.5.: *Computational domain decomposition and grid spacings (with k the cell-to-cell stretching ratio). (x, y)-plane (top) and spanwise direction (bottom).*

A rather coarse mesh of 4 millions grid cells has been considered (most of the CFD references of the literature consider between 5 to 10 millions of grid cells) in order to assess the capabilities of the code in capturing the main coherent structures developing in the wake. The domain has been discretized in 15 elements in the spanwise direction, leading to very elongated computational cells in the z-direction. No wall function is used, so the mesh is refined near the cylinder wall ($y^+ \lesssim 4$ in wall units) and in the wake, as shown on Figure 4.5. The time-step is equal to $10^{-3}D/u_\infty$ and computations have been run over a time interval of $460D/u_\infty$ corresponding to about 90 shedding periods.

DDES computations have been performed both with the new IBM PISO solver and with the classical body-fitted approach. In somes cases, additional computations have been carried out using the native IDDES model of OpenFOAM to show the impact of the turbulence modelling on the simulation.

4.3.1.2. Results

The analysis of the main flow features indicates that a reasonably high level of precision in the flow prediction has been achieved using the new IBM solver.

Figure 4.6 shows the contours of the vorticity magnitude predicted by the new IBM solver (Figures 4.6 a, b) and the classical body fitted approach (Figure 4.6 c). These results are compared to results of reference obtained from LES (Figure 4.6 d) and by PIV in experiment (Figure 4.6 e). Although being rapidly dissipated in the wake due to the stretching of the mesh , the IBM computations exhibit the first vortices of the well-known von Karman vortex street with periodic vortex shedding. The picture clearly shows the transition to turbulence takes place in the free shear layers in the very near wake. The IBM DDES results are qualitatively very closed to those obtained using the body-fitted solver. By comparing now with the results of reference, the IDDES model clearly improves the quality of the results as shown on Figure 4.6.

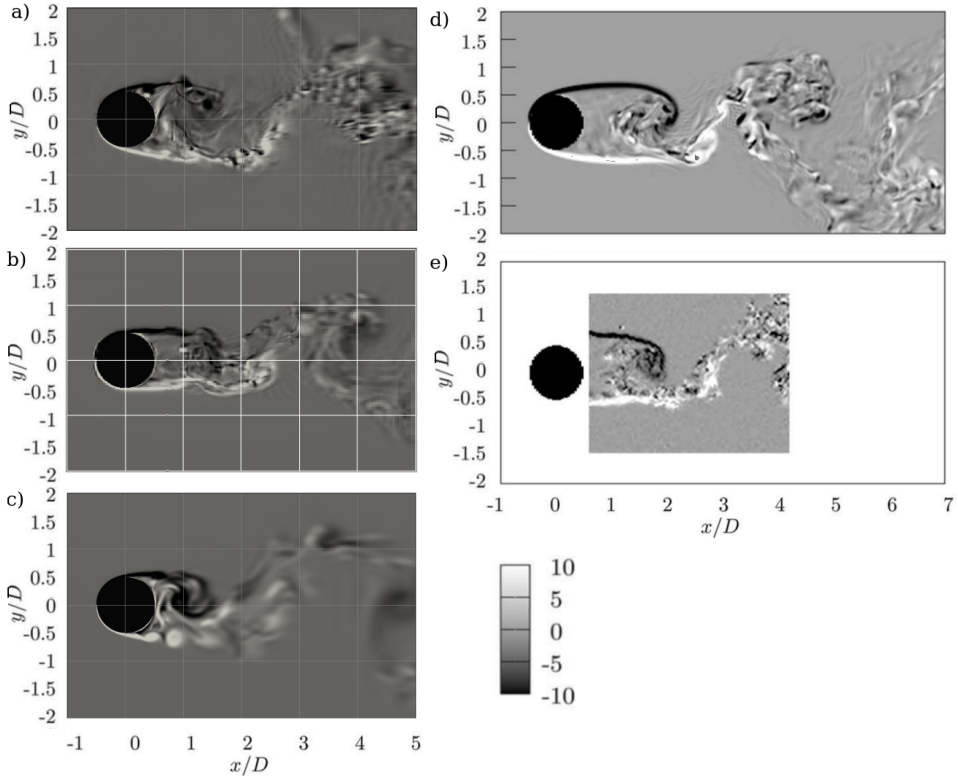


Figure 4.6.: *Snapshot of instantaneous contour of the vorticity magnitude ranging from -10 to 10 at* $Re = 3900$. *IBM DDES (a), IBM IDDES (b), Body fitted DDES (c), LES from Parnaudeau et al. [91] (d), PIV from Parnaudeau et al. [91] (e).*

Typical time histories of the computed lift coefficient C_L and drag coefficient C_D are plotted on Figure 4.7. Both show cyclic oscillations due to the vortex shedding phenomenon and high-frequency turbulent fluctuations. The mean drag and rms lift coefficients as well as the Strouhal number are in agreement with numerical and experimental data available in the literature, as shown on Table 4.1. The mean separation angle is also very well predicted by the model that shows its capacity to deal with boundary layer separation.

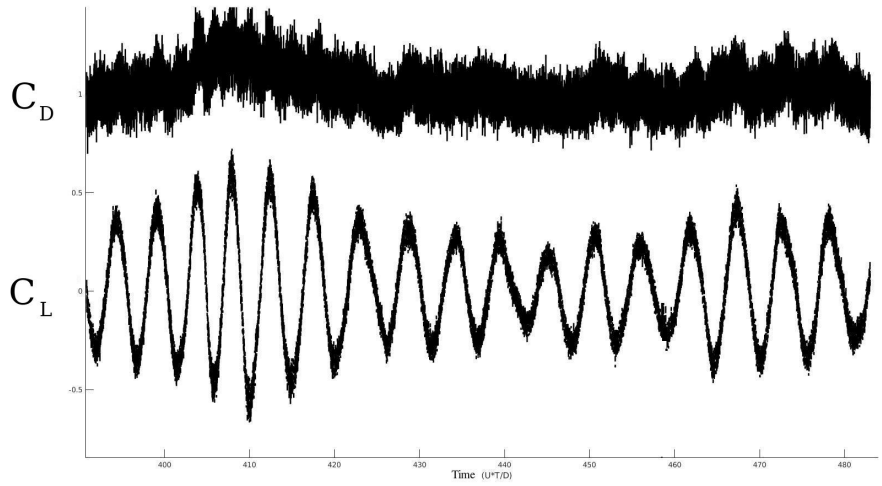


Figure 4.7.: *Lift C_L and drag C_D coefficients time histories. IBM IDDES results at $Re = 3900$.*

Table 4.1.: *Rms lift, mean drag coefficients, Strouhal number and mean separation angle for 3D flow past a fixed cylinder at $Re = 3900$. Numerical and experimental data from the literature are provided for comparison.*

	C_{Lrms}	C_{Dmean}	St	θ_{mean}
Present DDES-SA($Re = 3900$)	**0.2593**	**1.0389**	**0.2108**	**89.89**
Present IDDES-SA($Re = 3900$)	**0.1119**	**0.9424**	**0.221**	**89.92**
Parnaudeau et al. [91] PIV	-	-	0.21	-
Lourenco and Shi [67] PIV	-	0.99	0.21	-
D'Alessandro et al. [15] [*OpenFOAM* SA-IDDES]	0.1458	1.0235	0.222	87
D'Alessandro et al. [15] [*OpenFOAM* NLDES]	0.3832	1.1751	0.217	88.99
D'Alessandro et al. [15] [*OpenFOAM* SA-DES]	0.4248	1.2025	0.215	89.28
D'Alessandro et al. [15] [*OpenFOAM* v2-f DES]	0.1088	0.9857	0.214	86.40
Mittal and Moin [82] LES	-	1	0.22	87
Kravchenko and Moin [60]	-	1.04	0.21	88

First and second-order flow statistics are shown together with experimental data and other *OpenFOAM* results of the literature. The overall agreement is very good as shown by the iso-contours of the mean streamwise and cross-flow velocity on Figures 4.8 and 4.9. In particular, comparisons between the new IBM solver and the original solver of OpenFOAM based on a body-fitted approach show an excellent agreement. Both models well predict the peak of the mean streamwise velocity on the centerline but slightly more upstream behind the cylinder than in the LES simulations and experiments of Parnaudeau et al. [91] . This feature with OpenFOAM simulations is already mentionned in D'Alessandro et al. [15] . We can assume that it is due to the native DDES model which predicts an earlier breakdown to turbulence than in the predictions of Parnaudeau et al. [91] , leading on thus an earlier vortex shedding. Unfortunately, IBM IDDES statistics seem to be not converged yet.

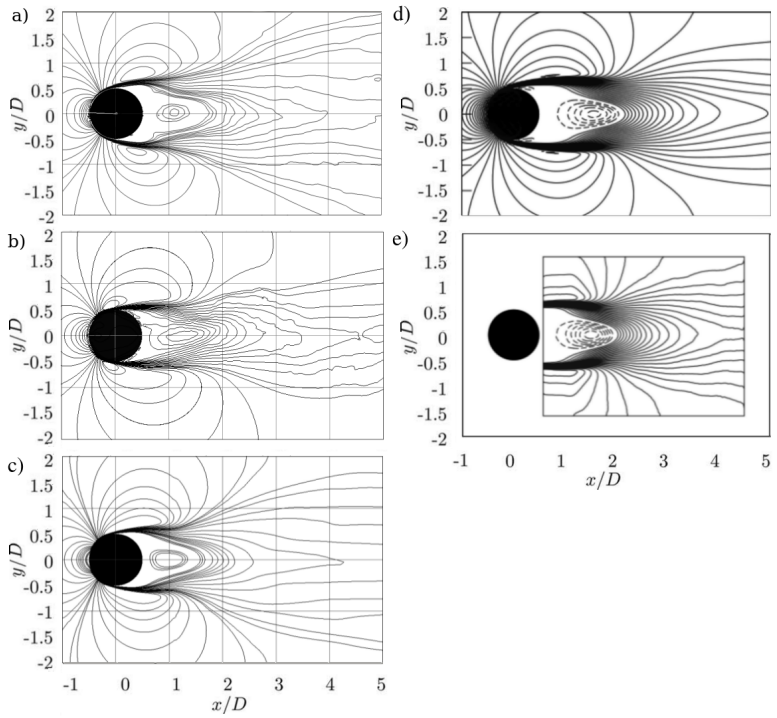


Figure 4.8.: *Snapshot of the mean velocity contours on streamwise direction u_x at $Re = 3900$. IBM DDES (a), IBM IDDES (b), Body fitted DDES (c), LES Parnaudeau et al. [91] (d), PIV Parnaudeau et al. [91] (e).*

As confirmed by the variance of the streamwise velocity and the covariance of the velocity fluctuations contours $< u'u' >$ and $< u'v' >$ respectively, on Figures 4.10 and 4.11. Both Figures show a symmetry around the centerline with two saddle points in eithers sides of the centerline (with different signs on each sides of the centerline for u_y and $< \overbrace{u'v'} >$). As previously mentionned, the peaks are located behind the cylinder but more upstream than in the data of the literature. This feature was also observed on the vorticity snapshots Figure 4.6, where the DDES models predict an earlier shedding whereas, the IDDES model better compares with the predictions of Parnaudeau et al. [91] .

Despite the lack of convergence the mean streamwise velocity profiles (Figure 4.13) are rather well predicted, both in the wake centerline and at three locations in the wake. The results show a particularly satisfactory agreement with the PIV measurements of Lourenco and Shi [67] , with a good estimate of the mean streamwise length of the recirculation zone (corresponding to negative velocity values) and of the U-shape of the profiles. It is noteworthy to notice that the present results match well the body-fitted results obtained with the same turbulence model meaning that the IBM correctly reproduces the solid boundary. The same overall agreement is obtained on the streamwise Reynolds stress in the near wake, Figure 4.12. The overestimate of the streamwise velocity fluctuations at $x = 1.06D$ and around $y = 0$, and so of the turbulence intensity, should be related to an early numerical transition to turbulence in the thin shear layers behind the cylinder leading to their shortening. This is supported by the snapshot of vorticity on Figure 4.6.

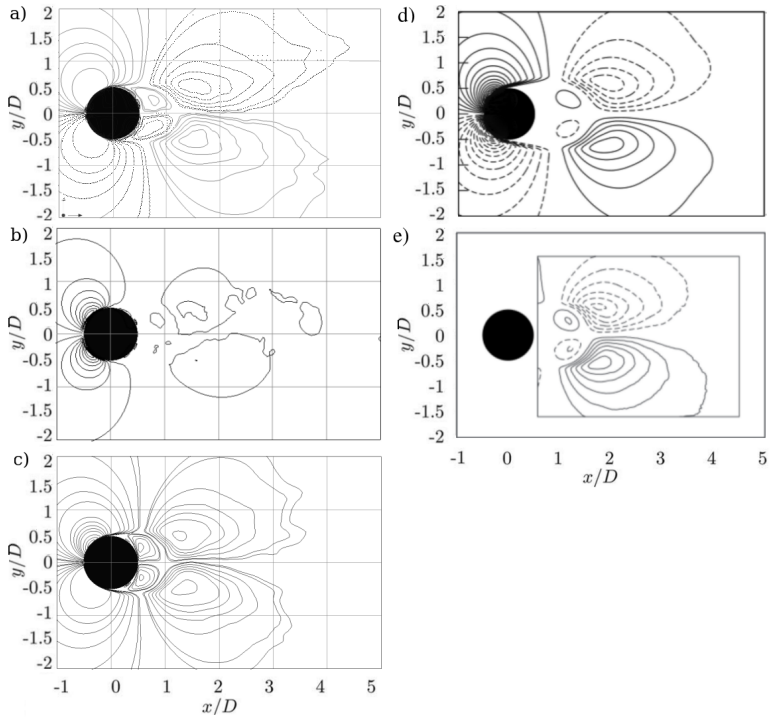


Figure 4.9.: *Snapshot of the mean velocity contours on cross-flow direction u_y at $Re = 3900$. IBM DDES (a), IBM IDDES (b), Body fitted DDES (c), LES Parnaudeau et al. [91] (d), PIV Parnaudeau et al. [91] (e).*

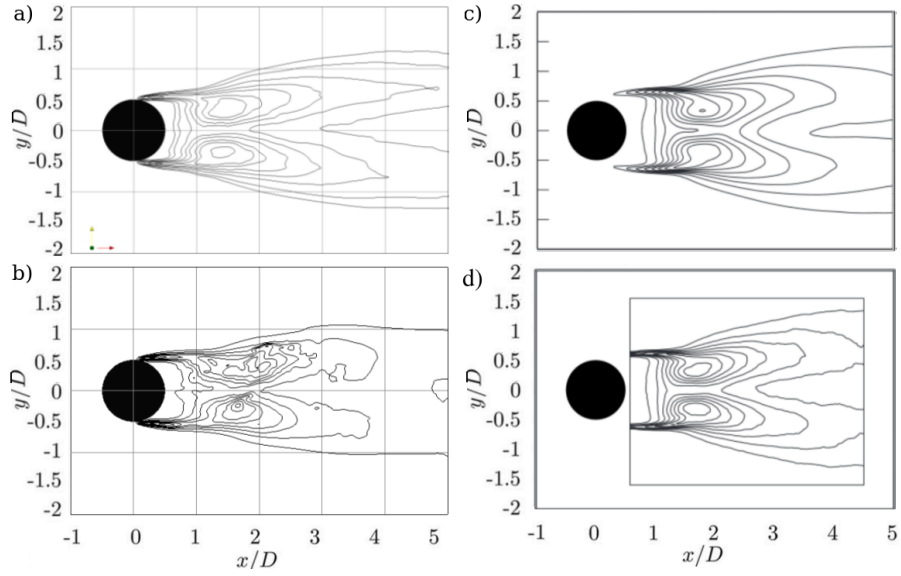


Figure 4.10.: *Snapshot of the variance of the streamwise velocity fluctuation contours $< u'u' >$ at $Re = 3900$. IBM DDES (a), IBM IDDES (b), LES Parnaudeau et al. [91] (c), PIV Parnaudeau et al. [91] (d).*

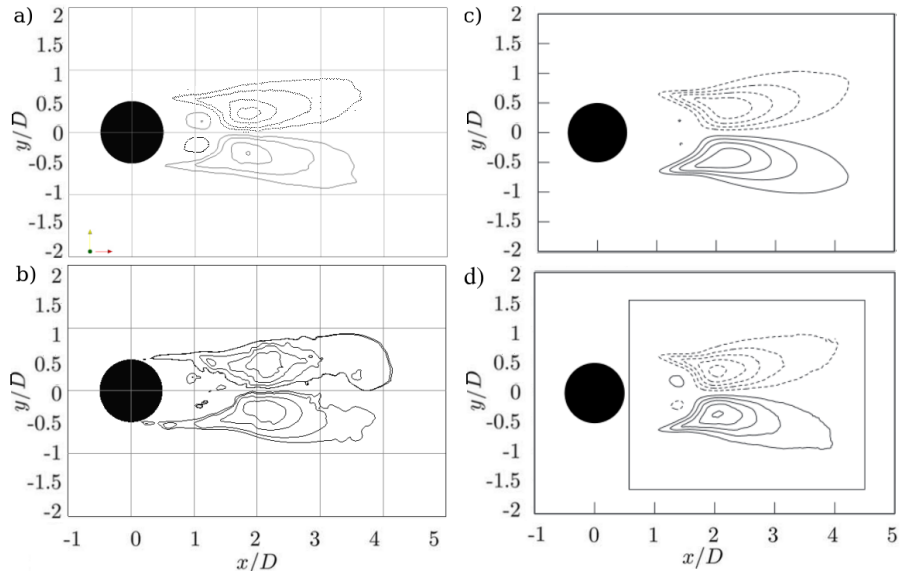


Figure 4.11.: *Snapshot of the covariance of the velocity fluctuation contours* $< uv >$ *at* $Re = 3900$. *Present IBM DDES (a), present IBM IDDES (b), LES Parnaudeau et al. [91] (c), PIV Parnaudeau et al. [91] (d).*

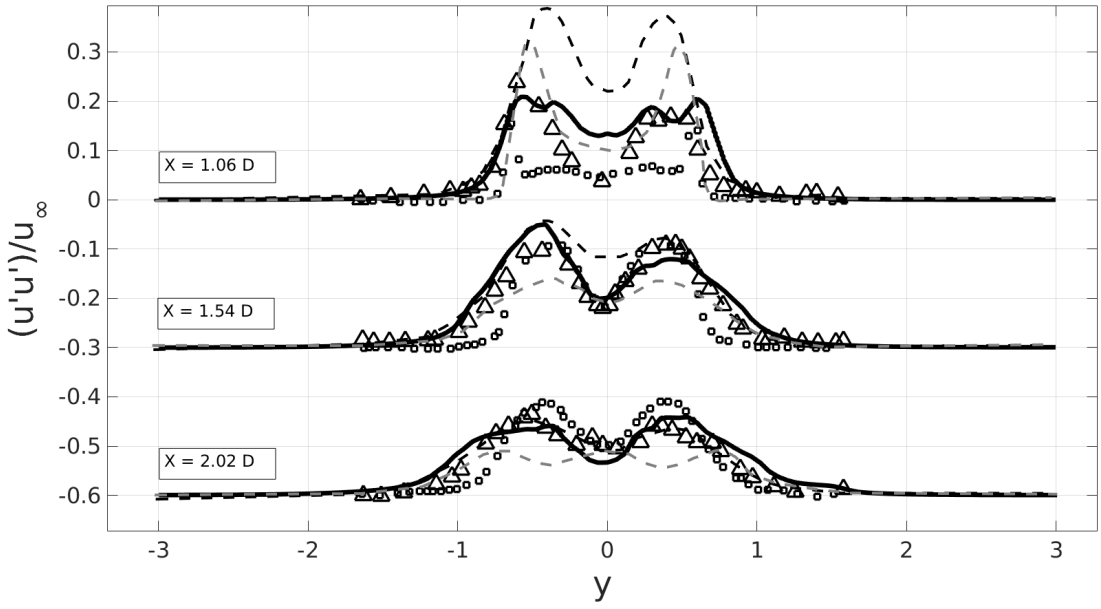


Figure 4.12.: *Resolved streamwise Reynolds stresses at different locations in the wake region. (–) Present simulation IBM-DDES, (□) PIV Parnaudeau et al. [91] , (△)PIV Lourenco and Shi [67] , (- -)OpenFOAM DES D'Alessandro et al. [15] , (- -) OpenFOAM IDDES D'Alessandro et al. [15]*

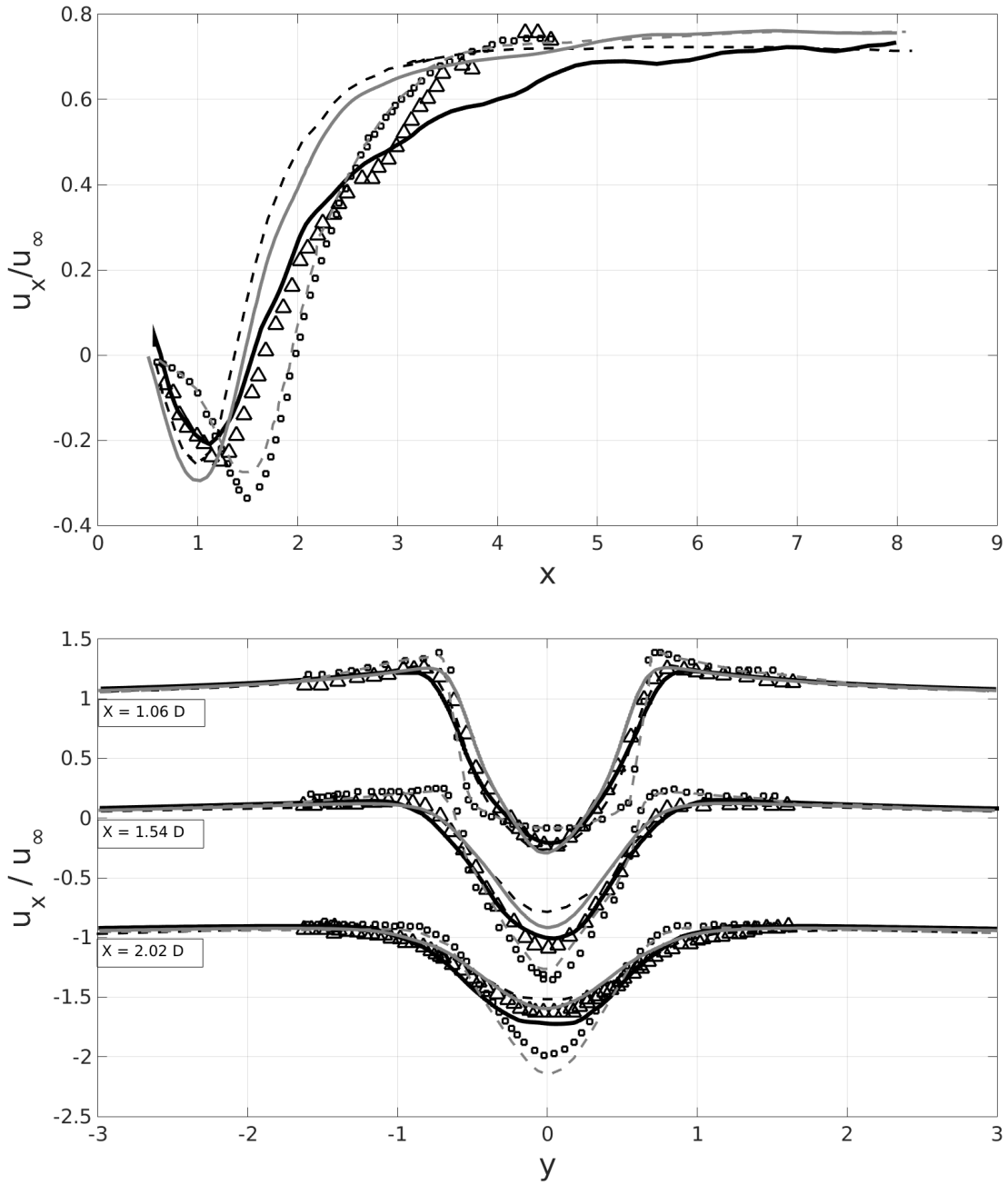


Figure 4.13.: *Mean streamwise velocity profiles in the wake centerline (top) and at three locations in the near wake (bottom). (–) Present simulation IBM-DDES, (–) Present simulation DDES body-fitted, (□) PIV Parnaudeau et al. [91] , (△)PIV Lourenco and Shi [67] , (- -)OpenFOAM DES from D'Alessandro et al. [15] , (- -) OpenFOAM IDDES from D'Alessandro et al. [15]*

4.3.2. Simulation past a sphere

To test the method in a more challenging 3D geometry, the flow past a fixed sphere has been simulated at $Re = 10000$ using the new IBM IDDES solver implemented in OpenFOAM At this

Reynolds number, the turbulent flow is subcritical in the sense that the shear layers remain laminar at separation whereas the flow becomes turbulent in the wake.

The flow dynamics is known to be complex. Hairpin-shaped vortices are periodically shed with a shedding orientation depending on the time and thus the flow becomes asymmetric.

4.3.2.1. Computational details

The mesh used is described Sec. 2.4.2.1, and refers to the domain 2 with$H = 32$.

4.3.2.2. Results

The analysis of the main flow features indicates that a reasonably high level of precision in the flow prediction has been achieved using the new IBM solver.

Figure 4.14 shows the topology of the vortex shedding predicted by the new IBM IDDES solver (Figure 4.14a), by experiment (Figure 4.14b), by LES (Figure 4.14c) and finally by DNS (Figure 4.14d).

Although being less resolved than the data of reference of the literature, the IBM-IDDES solution qualitatively shows the same features than the experiment of Sakamoto [113] .

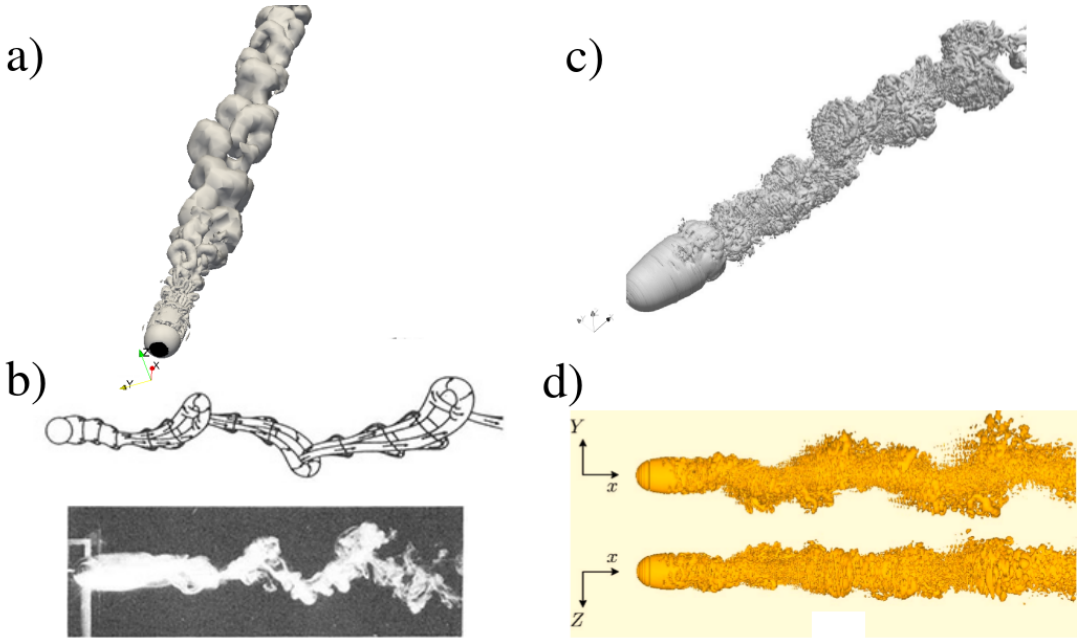


Figure 4.14.: *Instantaneous patterns of vortex shedding in the wake region of a sphere at* $Re = 10^4$. *a) Present simulation IBM-IDDES (Q criterion), b) Sakamoto [113] Exp., Rodriguez et al. [108] DNS (Q criterion) and d) Yun et al. [150] LES (using vortex identification method by Jeong and Hussain [52])*

More quantitatively, the main flow parameters along with the experimental measurements and simulations results of the literature are summarized in table 4.2. A satisfactory agreement is found. The mean drag coefficient C_{Dmean} is only 1% larger than the DNS value.The Strouhal number is in the range of the values obtained from experiments, and close to the value obtained by the DES simulation of Constantinescu and Squires [11] using a 2^{nd}-order scheme for the convective terms similar to ours. The recirculation length L/D is also found in agreement with a value l 5% lower than the LES of Yun et al. [150] .

Table 4.2.: *Mean drag coefficient C_{Dmean}, Strouhal number St, and mean recirculation length $L/D$for 3D flow past a fixed sphere at $Re = 10^4$. Numerical and experimental data from the literature are provided for comparison.*

	C_{Dmean}	St	L/D
Present IDDES-SA($Re = 10^4$)	**0.4051**	**0.1886**	**1.3**
Achenbach [1] Exp.	-	0.15	-
Sakamoto [113] Exp.	-	0.19	-
Rodriguez et al. [108] DNS	0.402	0.195	1.657
Stadler et al. [127] DNS	-	0.21	1.475
Yun et al. [150] LES	0.393	0.17	1.364
Constantinescu and Squires [11] LES	0.393	0.195	1.7
Constantinescu and Squires [11] DES $2^{nd}order$	0.44	0.185	-
Constantinescu and Squires [11] DES $5^{th}order$	0.397	0.2	-

The time histories of the computed lift coefficient C_L and drag coefficient C_D are plotted on Figure 4.15 . Both show cyclic oscillations due to the vortex shedding phenomenon.

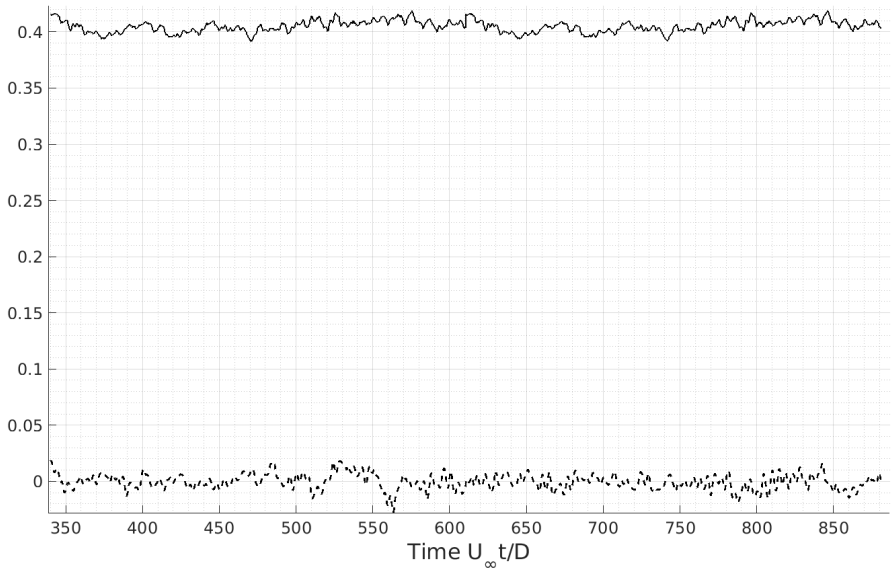


Figure 4.15.: *Temporal evolutions of C_D (full line) and C_L (dashed line) for the 3D flow past a sphere at $Re = 10^4$*

In order to analyse the flow more in details, the mean streamwise velocity u, the streamwise Reynolds stress $< u'u' >$ and the primary Reynolds shear stresses $< u'v' >$ profiles are plotted at

different locations in the wake on Figures 4.16, 4.18 and 4.17, respectively. They are compared to the simulations of Constantinescu and Squires [11] , Rodriguez et al. [108] and Stadler et al. [127] .

If the IBM IDDES solver is able to reproduce the good trend in the profiles, there is in this stage of development no agreement. At $x/D = 0.6$ and $x/D = 0.83$, there is a switch upstream in the flow of the locations of both the velocity drop Figures 4.16 and the turbulent peak Figures 4.18 and 4.17, from $y/D \simeq 0.58$ for the present simulations to $y/D \simeq 0.64$ on the simulations of reference. This difference is most likely coming from the rather poor resolution used in the present simulation in the near wake, associated to differences in the discretization of the convective terms, which is of 5^th-order in the simulations of reference, compared to $2^{nd} - order$ in the present solver.

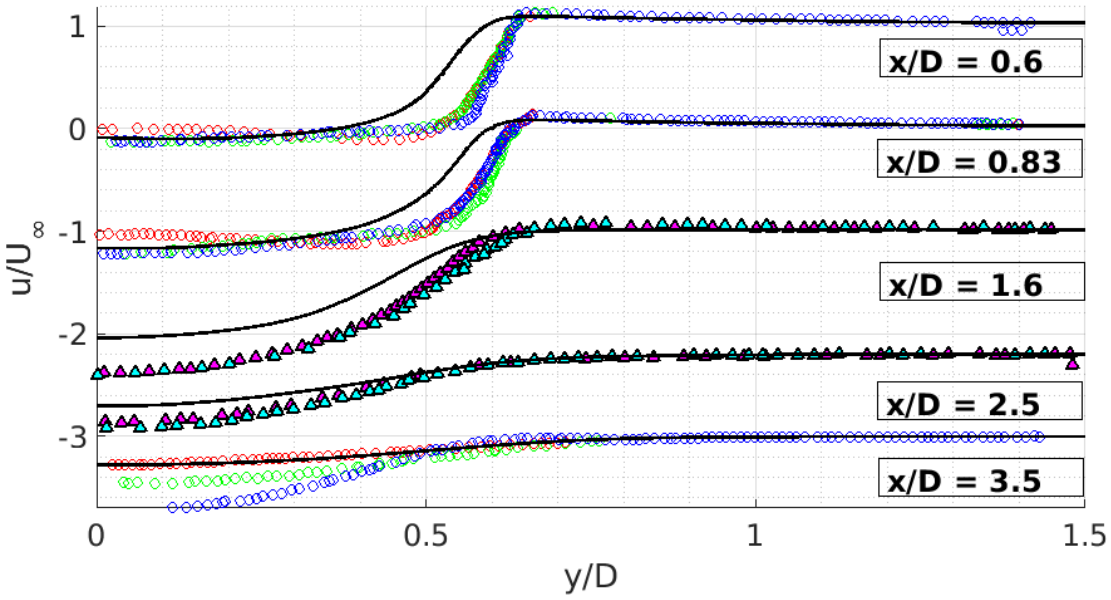


Figure 4.16.: *Streamwise velocity at different locations in the wake . (-) Present IBM-IDDES, (○) DES from Constantinescu and Squires [11] , (○) LES from Constantinescu and Squires [11] , (○) RANS-SA from Constantinescu and Squires [11] , (▲) DNS from Rodriguez et al. [108] , (▲) DNS from Stadler et al. [127] . Flow past a sphere at* $Re = 10000$.

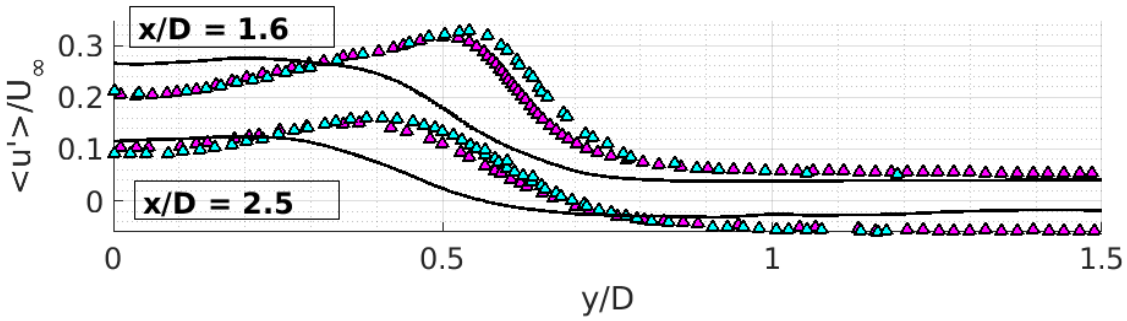


Figure 4.17.: *Streamwise Reynolds stresses at different locations in the wake. (-) Present IBM-IDDES, (▲) DNS from Rodriguez et al. [108] , (▲) DNS from Stadler et al. [127] . Flow past a sphere at* $Re = 10000$.

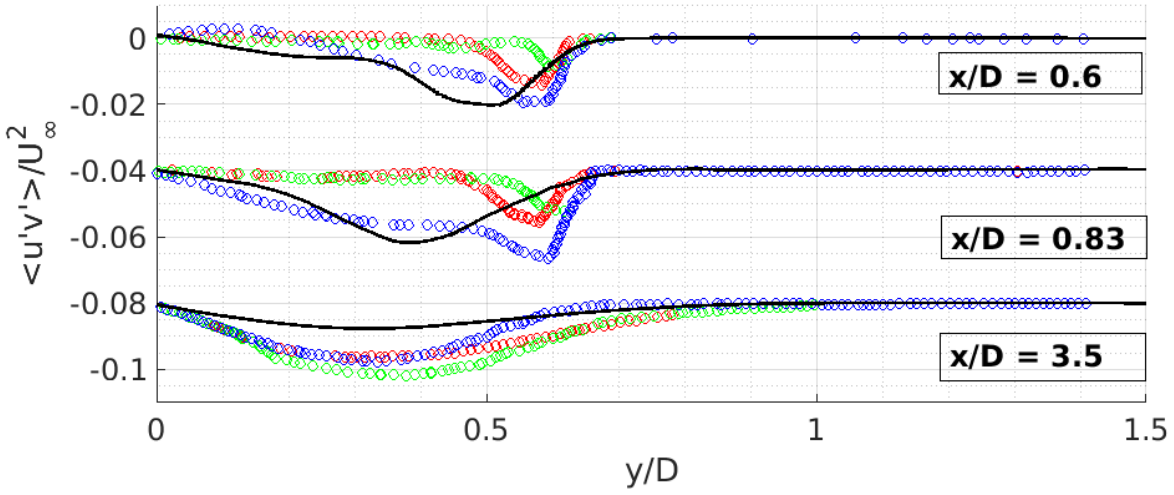


Figure 4.18.: *Primary Reynolds shear stresses at different locations in the wake. (-) Present IBM-IDDES, (○) DES from Constantinescu and Squires [11] , (○) LES from Constantinescu and Squires [11] , (○) RANS-SA from Constantinescu and Squires [11] . Flow past a sphere at* $Re = 10000$.

4.3.3. Validation of the wall-function approach

In this thesis, we lay the foundations for the coupling of a wall boundary-layer model with the new IBM solver. The basic wall-function model presented in Chapter 1 is known to be not suited for the flows under consideration in this work, but it allows us to investigate such kind of coupling in the IBM framework.

To validate our implementation, we consider hereafter the simple flow (without pressure gradients) over a flat plate, as illustrated on Figure 4.19. The Reynolds number based on the plate length L is equal to $Re = 10^6$.

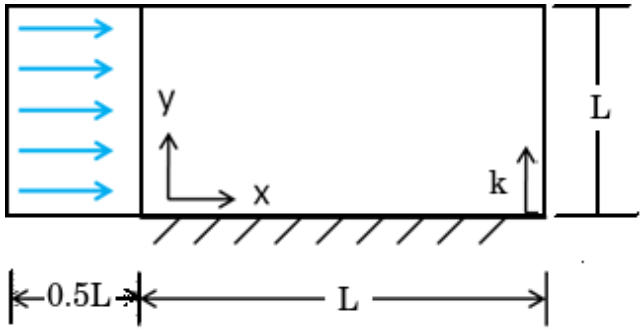


Figure 4.19.: *Sketch of the flow configuration over a flat plate.*

4.3.3.1. Computational details

The computational domain sizes as $[0, 1.5L] \times [0, L]$ in the streamwise (x) and vertical (y) directions. It is extended of $0.5L$ upstream to the plate in the streamwise direction with a slip condition to avoid a strong perturbation at the edge of the plate.

IBM simulations with or without a wall-function are performed on four grids shown Figure 4.19 corresponding to various resolutions of the boundary layer: $\Delta y = 0.0032$ for the grid 1 ($G1$), $\Delta y = 0.02$ for the grid 2 ($G2$) and $\Delta y = 0.0016$ for the grid 3 ($G3$) near the wall with a cell to cell stretching factor of $k = 1$ on the y direction and a $\Delta x = 0.02$. The grid $G4$ has a resolution of $\Delta y = 2.353e^{-5}$ near the wall with a cell to cell stretching factor of $k = 1.02$ on the y direction and a $\Delta x = 0.02$.

For comparisons, computations are also performed using the body-fitted approach of OpenFOAM. In these computations, the boundary layer is either modeled, using the OpenFOAM native wall-function (called $nutUSpaldingWallFunction$) for the grids ($G1, G2$) or is resolved, using a very fine mesh ($G4$) corresponding to $y^+ \simeq 1$ at the plate.

Figure 4.20 shows the streamwise evolution of y^+ for various simulations. In the classical body-fitted model, the wall-function is applied at the center of the cell whereas in our IBM it is applied at the side of the first cell, corresponding to $y_{wall} = \Delta y$, Δy being the vertical dimension of the first cell close to the wall. As a consequence, the same grid does not lead to the same values of y^+ independing on the numerical model which is used (body-fitted versus IBM). With the grids $G1$ and $G2$, Figure 4.20 shows that the wall-function is applied on a point located within the log-layer region, and outside the inner boundary-layer. However, IBM simulations performed with or without wall-function on these grids lead to the same values of y^+. On the finer $G3$ grid, the IBM leads to similar values of y^+ obtained with the body-fitted model on the $G2$ grid. Finally, body-fitted and IBM simulations on the finest grid $G4$ without wall-function lead to similar values of y^+, all smaller than 1. Thus, in these simulations, the inner boundary layer is well-resolved.

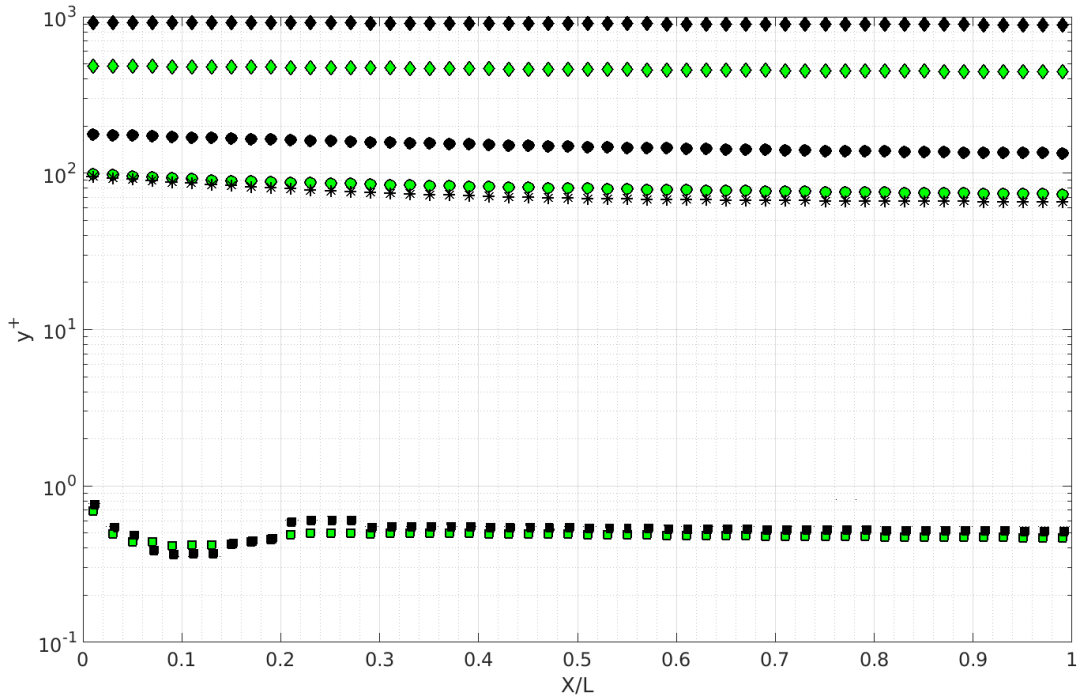


Figure 4.20.: *Streamwise evolutions of y^+ for the flow over the flat plate at $Re = 10^6$ and for different grids. Simulations with wall-function on $G1$ and $G2$ grids: with IBM (● and ⧫) and body-fitted model (● and ⧫). IBM simulation with wall-function on $G3$,(⁎). Simulations without wall-function on $G4$: with IBM (■) and body-fitted model (■).*

4.3.3.2. The flow structure

The evolution of the friction coefficient and velocity profile along the plate are plotted for the different simulations together with data of reference of the literature on Figures 4.21 and 5.4, respectively.

For $x/L \gtrsim 0.2$ all simulations tend to the turbulent skin friction evolution given by Schlichting [114] . Even for coarse meshes simulations (first point within the log-layer), the C_f evolution is rather well predicted. In addition, at the same resolution, IBM and body-fitted approach with wall function provide very close results showing the correct implementation of the wall-function in the IBM solver.

It is worth to note that in wall-resolved simulations ($G4$ grid, without wall-function), body-fitted and IBM behave similarly wi . In an upstream part of the floand delay the breakdown to turbulence and delay the breakdown to turbulenceand delay the breakdown to turbulencew $(x/L \lesssim 0.15)$, the friction coefficient is close to the laminar value provided by the blasius friction law.

Finally and as expected, simulations on the coarse $G1$ grid without wall function predict an unconsistent result (Figure 4.22), confirming the necessity to use such an approach when the boundar- layer is not resolved.

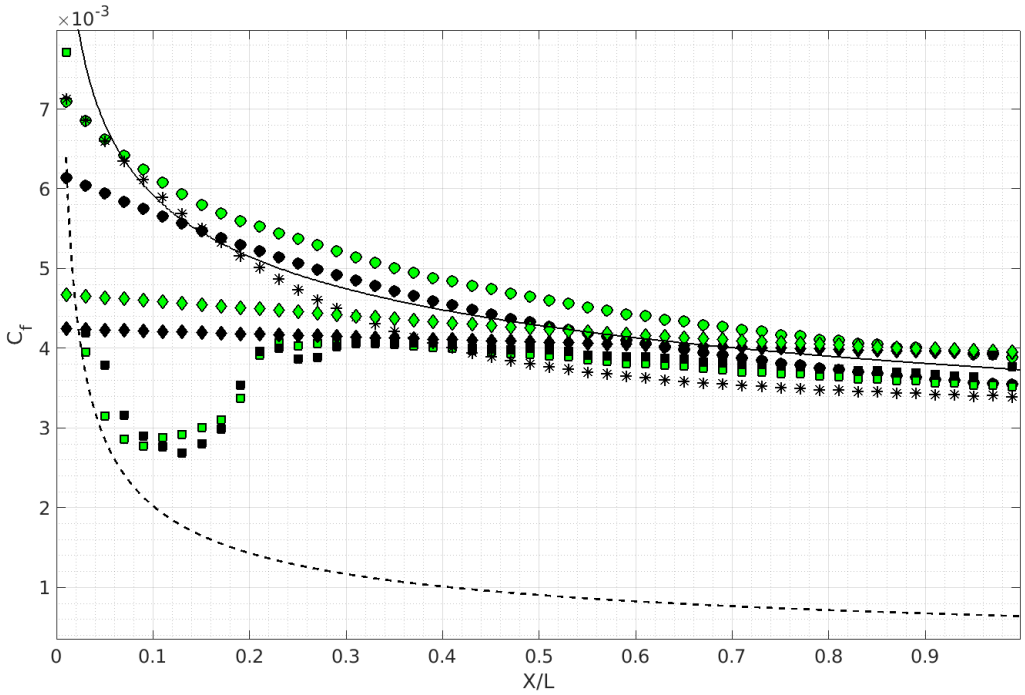


Figure 4.21.: *Streamwise evolution of the friction coefficient along the plate at* $Re = 10^6$. *Simulations with wall-function on* $G1$ *and* $G2$ *grids: IBM (● and ♦), and body-fitted model (● and ♦). IBM simulation with wall function on* $G3$ *grid (⁎). Simulations without wall function on* $G4$ *grid: IBM (■) and body-fitted model (■). Turbulent skin friction from Schlichting [114]* $C_f = \frac{0.0592}{Re_x^{1/5}}$ *(-). Laminar skin friction (Blasius friction law)* $C_f = \frac{0.64}{Re_x^{1/2}}$ *(- -) .*

The same trend is observed on the vertical velocity profiles, Figure 5.4. All simulations are able to reasonnably predict the velocity profile shape. When the viscous sublayer is not resolved ($G1, G2$ and $G3$ grids) the velocity profile is rather well predicted within the log layer. In the near wall resolved simulations ($G4$ grid), both the IBM and body-fitted model are able to describe the inner sublayer.

Former simulations lead to different values of turbulent viscosity across the boundary layer, as shown on Figure figure 4.24 at $x/L = 0.95$. The disagreement between body-fitted and IBM simulations is large when wall-functions are used with the coarsest grids. On the finest $G4$ grid, both simulations are closer but the body-fitted approach continues to predict a higher peak of ν_t within the boundary layer.

As a conclusion, wall-function has to be impemented into the IBM solver when the mesh is coarse. A rather fine mesh is however required as the first point must be located into the log region of the boundary layer. In this case, present results show that a rather accurate estimate of the skin friction through the IBM forcing is then possible.

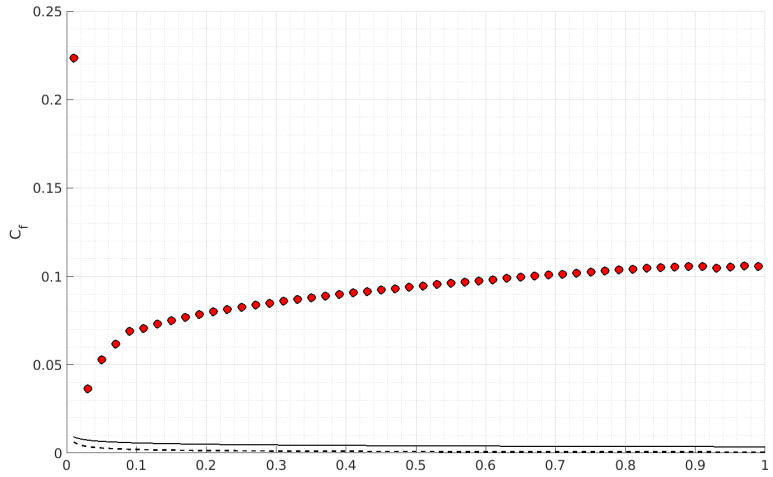


Figure 4.22.: *Streamwise evolution of the friction coefficient along the plate at* $Re = 10^6$ *on the coarsest* $G1$ *grid: IBM without wall-function (●), turbulent skin friction from Schlichting [114]* $C_f = \frac{0.0592}{Re_x^{1/5}}$ *(-) and laminar skin friction (Blasius friction law)* $C_f = \frac{0.64}{Re_x^{1/2}}$ *(- -).*

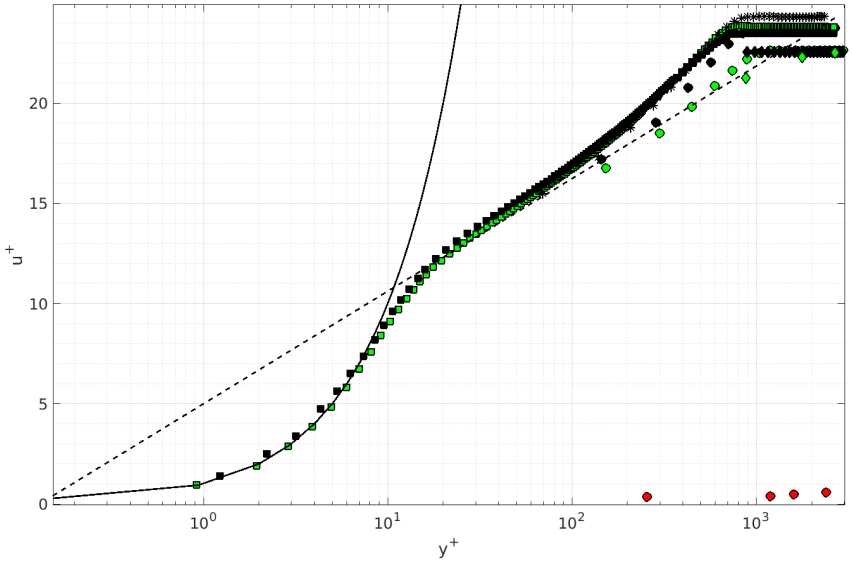


Figure 4.23.: *Vertical velocity profiles at* $x/L = 0.95$ *for the flow over the plate at* $Re = 10^6$*. Simulations with wall-function on* $G1$ *and* $G2$ *grids: IBM (● and ⧫), and body-fitted model (● and ⧫). IBM simulation with wall function on* $G3$ *grid (⁎). Simulations without wall function on* $G4$ *grid: IBM (■) and body-fitted model (■). Sub layer profile* $u^+ = y^+$ *(-) and log-layer profile* $u^+ = \frac{1}{\kappa} log(y^+) + C$ *(–).*

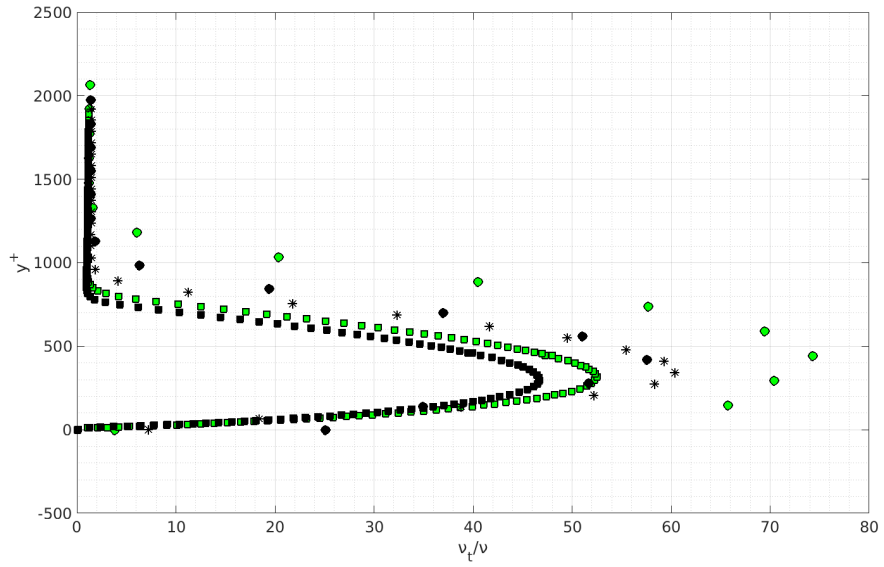


Figure 4.24.: *Vertical turbulent viscosity profiles for the flow over the plate at* $Re = 10^6$ *and* $x/L = 0.95$*. Simulations with wall-function on* $G1$ *grid: IBM (●) and body-fitted model (●). IBM simulation with wall-function on* $G3$ *grid (⋇). Near wall-resolved simulations (without wall-function) on* $G4$ *grid: IBM (■) and body-fitted model (■).*

Chapter 5

Linearized direct and adjoint solvers in the IBM PISO of OpenFoam

---

The control of VIV will require the determination of the the sensitivity of the flow to base-flow and force modifications. In this chapter, we describe the implementation and the validation of a linearized Navier-Stokes and of a discrete adjoint solver in the new IBM PISO algorithme developped in OpenFOAM. As a step towards the theoretical control of VIV, this chapter discusses the implementation and the validation of linearized direct/adjoint Navier-Stokes solvers developed in OpenFOAM and building on the novel IBM PISO algorithm. The related solutions are used to assess the sensitivity to a small external forcing of the instability mechanism responsible for the onset of VIV. For the flow past a fixed cylinder at $Re = 100$, the obtained results are in good agreement with the reference results of the literature.

5.1. Implementation of direct and adjoint equations in the IBM PISO solver

It is generally accepted that the origin of VIVs is in a global instability of the coupled flow-cylinder system arising at low Reynolds number (see Ref [12] ,[77],[83] among others). A possible approach to control VIVs is thus to manipulate the instability properties to either mitigate VIVs (which can be done either by reducing the related growth rate, or by bringing the instability eigenfrequency closer to the structural frequency) or enhance them (either by increasing the growth rate or shifting the instability eigenfrequency away from the structural frequency). For pure flow systems, there exists rigorous theoretical framework to so, that build on the adjoint method to predict where and how to control without having to span exhaustive parameter ranges, hence a tremendous reduction in the computational costs (Ref [41] ,[34],[71] [76] and see Ref [69] for a detailed review).

Generally speaking, the approach require :

i. computing a steady solution to the Navier-Stokes equations,

ii. computing the leading eigenvectors of the direct/adjoint Navier-Stokes operators linearized about this steady solution (i.e. those eigenvectors whose growth rate is the largest),

iii. recombining the direct/adjoint eigenvectors to grant access to the sensitivity. The main steps are reviewed in the following together with the specifics of the numerical developments performed in OpenFOAM.

5.1.1. Steady solution to the Navier-Stokes equations

Velocity and pressure fields are decomposed into a 2D base flow denoted $Q_0 = (U_0, p_0)$. This base flow is solution of the stationnary 2D incompressible Navier-Stokes equations in which a body force f is assumed to be steady and to act only on the base flow.

Since the flow past a cylinder at $Re > 47$ is intrinsically unstable (see Figure 1.3), a dedicated steady solver has been developed. The steady solution being supposed to be symmetric along the x-axis, the idea is to constrain the simulation into the half-domain past the cylinder, in order to kill any instability related to the Von Karman street.

To do so, the immersed boundary near the symmetry axis ($y = 0$) has to be adapted. New ghost cells are needed, because the supporting box of the Lagragian markers being divided in 2 parts, Eulerian points are missing to perform the interpolation. The N-S solver on the half domain is the IBM PISO algorithm described in Sec. 2.2.1. The solution is reconstructed at the end on the whole domain.

5.1.2. Linearized direct solver

Linearized Navier-Stokes equations around the base flow for 2D infinitesimal perturbations $\mathbf{q}' = (u', p')$ write:

$$\frac{\partial(\mathbf{u}')}{\partial t} + \nabla(\mathbf{U_0}) \cdot \mathbf{u}' + \nabla(\mathbf{u}') \cdot \mathbf{U_0} = -\nabla \mathbf{p}' + \frac{\mathbf{1}}{\mathbf{Re}} \nabla^{\mathbf{2}} \mathbf{u}' \quad (5.1)$$

$$\nabla \cdot (\mathbf{u}') = 0, \tag{5.2}$$

the associated boundary conditions consisting of Dirichlet condition $\mathbf{u}' = 0$.

The IBM PISO algorithm is first run for the state variables (u', p').

i. **Predictor step:**

A. An estimate velocity $\widehat{\mathbf{u}}$ is obtained by solving the momentum Navier–Stokes equations without any force term, and using the pressure perturbation p' computed at the previous time step $n-1$:

$$\frac{\partial(\widehat{\mathbf{u}'}^n)}{\partial t} + \nabla(\mathbf{U_0}) \cdot \widehat{\mathbf{u}'}^n + \nabla(\widehat{\mathbf{u}'}^n) \cdot \mathbf{U_0} = -\nabla \widehat{\mathbf{p}'}^{n-1} + \frac{1}{Re} \nabla^2 \widehat{\mathbf{u}'}^n + f(\widehat{\mathbf{u}'}^n) \tag{5.3}$$

B. The calculation of the IBM force $\mathbf{F}_s$ is detailed in 2.2.2. It is calculated on the Lagrangian markers using $\widehat{\mathbf{u}\text{'}}$ (Eq. 2.7), and its values are spread on the Eulerian mesh to calculate $\mathbf{f}$ (Eq. 2.20).

C. A new velocity $\mathbf{u}'^{\star,1}$ is calculated from the Navier-Stokes equations accounting now the immersed boundary force term $\mathbf{f}$:

$$\frac{\partial \mathbf{u}'^{\star,1}}{\partial t} + \nabla(\mathbf{U_0}) \cdot \mathbf{u}'^{\star,1} + \nabla(\mathbf{u}'^{\star,1}) \cdot \mathbf{U_0} = -\nabla p^{n-1} + \frac{1}{Re} \nabla^2 \mathbf{u}'^{\star,1} + f(\widehat{\mathbf{u}\text{'}}) \tag{5.4}$$

$\mathbf{u}'^{\star,1}$ is the guess value of the velocity perturbation in the iterative PISO loop

ii. **PISO loop:**
For the sub-iteration $m = 1$ to $M-1$, and up to convergence:

A. At each sub-iteration, a pressure field $p'^{\star,m}$ is calculated from the following Poisson equation :

$$\nabla^2 p'^{\star,m} = -\nabla \cdot (\mathbf{u}'^{\star,m} \nabla \mathbf{u}'^{\star,m}) + \nabla \cdot f(\widehat{\mathbf{u}\text{'}}) \tag{5.5}$$

B. The velocity field is thus corrected using:

$$\mathbf{u}'^{\star,m+1} = g\left(\mathbf{u}'^{\star,m},\ \nabla p'^{\star,m},\ f(\widehat{\mathbf{u}\text{'}})\right) \tag{5.6}$$

where g as well as all discretized operators used in the algorithm are defined in Annex B.

iii. **Final step:**
The velocity and the pressure are finaly updated at time $n+1$:

$$\mathbf{u}'^{n+1} = \mathbf{u}'^{\star,M-1} \tag{5.7}$$
$$\mathbf{p}'^{n+1} = \mathbf{p}'^{\star,M-1} \tag{5.8}$$

Assuming the perturbations is sought in the form of global modes $(u', p')(x, y, z, t) = (\tilde{u}', \tilde{p}')(x, y, z)e^{\lambda t}$, Eq. 5.1 leads to the eigenvalues problem:

$$L(Q_0)\tilde{\mathbf{q}}^{'} = \lambda\tilde{\mathbf{q}}^{'} (5.9)$$

where $\lambda = \sigma + i\omega$ is the complex eigenvalue associated to the eigen vector $\tilde{\mathbf{q}}^{'}(\tilde{\mathbf{u}}', \tilde{\mathbf{p}}^{'})$. σ and ω, are the linear growth rate and the pulsation of the mode, respectively.

The evolution equation for the perturbations being written in the complex space, the OpenFOAM solver could not compute the complex solution. The obtained solution is real and is assumed to be the real part of the infinetisimal perturbations $q^{'}$. We now integrate these quantities in time over $T/4$, $T = \frac{2\pi}{\omega}$ in order to obtain the pure imaginary part of the perturbations. As the solution has been integrated over $T/4$, the imaginary part has a growth factor $e^{\sigma\frac{T}{4}}$ compared to the real one. We then reconstruct the complex solution as :

$$\mathbf{q}' = \mathbf{q}'_{\mathbf{Re}} + i\frac{\mathbf{q}'_{\mathbf{Im}}}{e^{\sigma\frac{T}{4}}} \tag{5.10}$$

Finally, the velocity perturbation vector is normalized with respect to the instantaneous value of the energy of the perturbation :

$$\mathbf{u}' = \frac{\mathbf{u}'}{(\mathbf{u}', \mathbf{u}')} \tag{5.11}$$

where the inner product $(.,.)$ is defined by :

$$(\mathbf{u_A}, \mathbf{u_B}) = \int_\Omega (\mathbf{u_A^*} \bullet \mathbf{u_B}) d\Omega \tag{5.12}$$

with u_A and u_B, two complex vector, * the complex conjugate operator.

5.1.3. Linearized adjoint solver

The methodology leads to solve the following adjoint equation (the reader is refered to the work of Marquet et al. [71] for more details):

$$\frac{\partial(\mathbf{u}^+)}{\partial t} + \nabla^T(\overline{\mathbf{U}}_\mathbf{0})\cdot\mathbf{u}^+ - \nabla(\mathbf{u}^+)\cdot\overline{\mathbf{U}}_\mathbf{0} = -\nabla\mathbf{p}^+ + \frac{1}{Re}\nabla^2\mathbf{u}^+ \tag{5.13}$$

$$\nabla\cdot(\mathbf{u}^+) = 0 \tag{5.14}$$

As the OpenFoam solver could not solve the system with the associated boundary contditions described in Marquet et al. [71] the equation is computed with the following boundary conditions :

$$u_y^+ = 0 \; on \; \partial\Omega \tag{5.15}$$

$$u_x^+ << U_\infty \; on \; \partial\Omega_{u,d,i} \tag{5.16}$$

$$p^+ = 0 \; on \; \partial\Omega_{u,d,i} \tag{5.17}$$

$$p^+.n + Re^{-1}\partial_x u_x^+.n = u_x^+ U_{0x}.n \; on \; \partial\Omega_o \tag{5.18}$$

$$u_x'(0) = u_y'(0) = u_x^+(T) = u_y^+(T) = 0, \tag{5.19}$$

with n the normal to the surface of the boundary.

The same procedure has been applied for the adjoint equation for the perturbations, Eq. 5.13. The IBM PISO algorithm described above is used for the adjoint perturbation variable q^+.

As for the direct solver, the imaginary part is obtained by integrating the final adjoint variables over $T/4$. However, as the time integration in OpenFOAM does not work backward in time, the temporal derivative is computed as a positive derivative. Thus, the imaginary part the complex solution q^+ has the opposite sign. The reconstruction of the complex adjoint perturbation $\overline{\mathbf{q}}^+$ writes thus:

$$\mathbf{q}^+ = \mathbf{q}^+_{Re} - \frac{\mathbf{q}^+_{Im}}{e^{\sigma \frac{T}{4}}} \tag{5.20}$$

The adjoint velocity perturbation is finally renormalized in order to get $(\mathbf{u}^+, \mathbf{u}') = 1$ that is required by the sensitivity analysis (Ref [71]). This leads to:

$$\mathbf{u}^+ = \frac{\mathbf{u}^+}{(\mathbf{u}^+, \mathbf{u}')} \tag{5.21}$$

5.1.4. Sensitivity of the flow to base-flow and Force modifications

Following the work of Marquet et al. [71] we can establish the expression of the sensibility to the base-flow modifications $\nabla_{\mathbf{U_0}} \lambda$ which reads :

$$\nabla_{\mathbf{U_0}} \lambda = \nabla(\mathbf{u}^+)\overline{\mathbf{u}}' - \nabla^T(\overline{\mathbf{u}}')\mathbf{u}^+ \tag{5.22}$$

with ^{T} and $\overline{...}$ the transpose and conjugate operators, respectively

Using the base flow equation we then reconstruct an adjoint equation (details in Ref [41] ,[71]):

$$\nabla^T(\overline{\mathbf{U}}_\mathbf{0}) \cdot \mathbf{U}^+_\mathbf{0} - \nabla(\mathbf{U}^+_\mathbf{0}) \cdot \overline{\mathbf{U}}_\mathbf{0} - f = -\nabla \mathbf{p}^+_\mathbf{0} + \frac{1}{Re}\nabla^2 \mathbf{U}^+_\mathbf{0} + \nabla(\mathbf{u}^+)\overline{\mathbf{u}}' - \nabla^T(\overline{\mathbf{u}}')\mathbf{u}^+ \tag{5.23}$$

$$\nabla \cdot (\mathbf{U}^+_\mathbf{0}) = 0, \tag{5.24}$$

with the following boundary conditions :

$$U_0^+{}_y = 0 \; on \; \partial\Omega \tag{5.25}$$

$$U_0^+{}_x << U_\infty \; on \; \partial\Omega_{u,d,i} \tag{5.26}$$

$$p_0^+ = 0 \; on \; \partial\Omega_{u,d,i} \tag{5.27}$$

$$p_0^+ . n + Re^{-1} \partial_x U_0^+{}_x . n = U_0^+{}_x U_{0x} . n \; on \; \partial\Omega_o, \tag{5.28}$$

where U_0^+, the adjoint complex variable of U_0, is the sensibility field to force modifications as detailed in Ref [41] ,[71].

5.2. Validations

Validations have been performed in the configuration of the flow past a fixed cylinder.

5.2.1. The computational domain

The computational domain is the same as in Sec. 2.4.1 with $\Delta x = \Delta y = 0.02D$. The mesh presented in Sec. 5.2.1, has been clipped on the y-axis at $y = 0$.

5.2.2. Steady solutions validation

Several simulations have been performed for steady flows for Reynolds numbers ranging between 15 and 200. For Reynolds numbers smaller than $Re = 47$, the flow is steady. It becomes unsteady at larger values. However, whatever the Reynolds number, the flow has been solved in half the domain in order to constrain the solution to be steady. For $Re < 47$, where the flow is naturally steady, such computations allow us to validate the procedure by comparing with experimental data of the literature. Results on the evolution of integrated quantities, C_D and L (length of the recirculation buble) are shown on Figure 5.1 together with experimental data and show a good agreemen for all Reynolds numbers considered in this study.

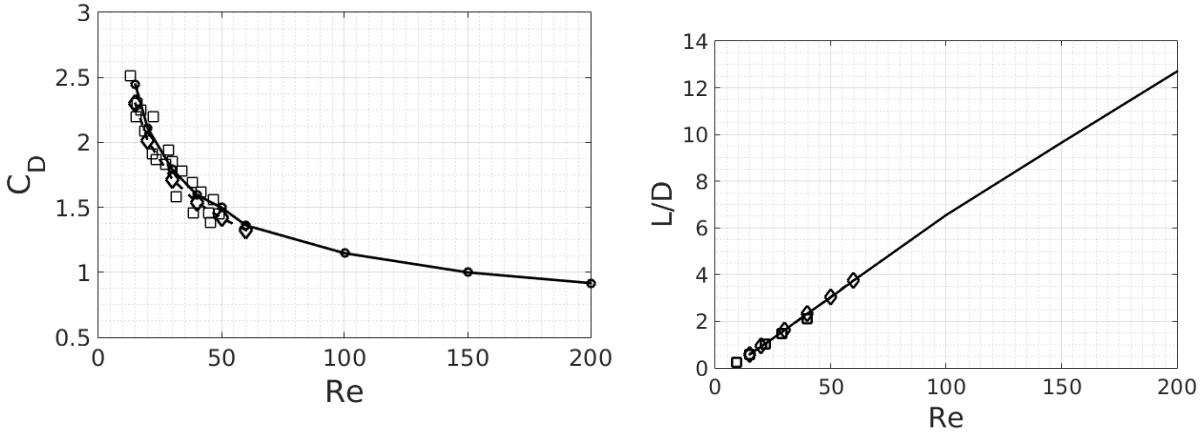


Figure 5.1.: *: Evolutions of the drag doefficient C_D (left) and the recirculation length L (right) as function of Reynolds number Re. $(-)$ present result, $(\square)$ experimental results of [139] and Taneda [135] , and $(\diamond)$ simulation results of Takami and Keller [133] .*

5.2.3. Sensitivity analysis to base-flow and to force modifications

The cylinder flow being known to become unstable at a critical Reynolds number, $Re = 47$, the global stability of the base flow is investigated here in the range $Re \in [60, 200]$.

Figure 5.2 shows the solutions for the linearized Navier-Stokes equations and for the adjoint equation of the growth rate σ and the pulsation ω as function of Reynolds number. A comparison with data of Meliga (private communication) show a good agreement.

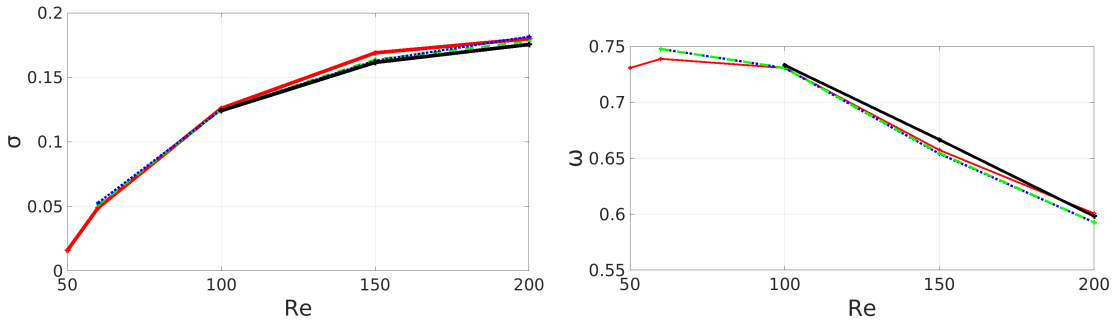


Figure 5.2.: *: Evolutions of the growth rate σ (top) and the pulsation ω (bottom) of the linearised Navier-Stokes and adjoint. (– –) present solution of the linearised Navier-Stokes, (....) present solution for the adjoint , (–) solution of Meliga (private Communication),(–) solution of the full Navier-Sokes equations.*

The direct and adjoint velocity perturbation fields are shown on Figure 5.3. The prediction of the IBM solver implemnted in OpenFOAM is also in good agreement with the results of Meliga (private communication).

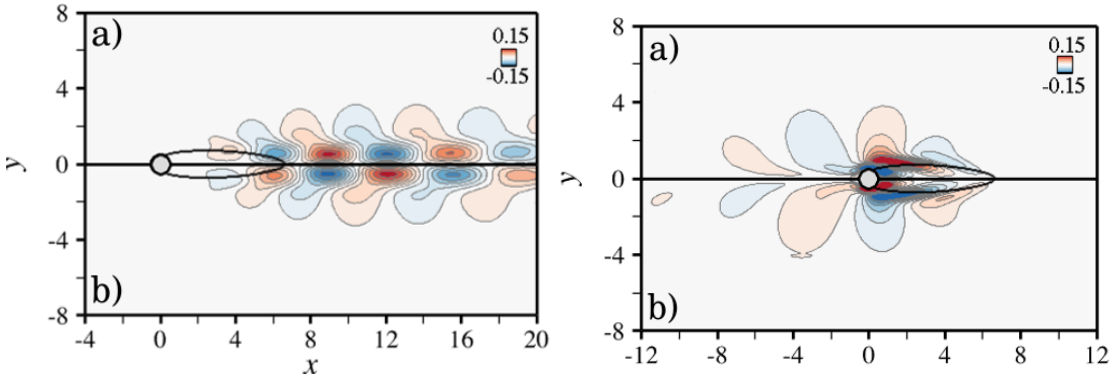


Figure 5.3.: *: Direct (top) and adjoint (bottom) velocity perturbation fields compared to the literature at $Re = 100$. (a) Present solution, (b) solution of Meliga (private Communication)*

At $Re = 100$, the sensitivity of the base flow to force modifications has been used to identify the flow regions where they produce the largest eigenvalue variations. Results predicted by the IBM solver of OpenFOAM show again a good agreement with results of Meliga (private communication), Figure 5.4 .

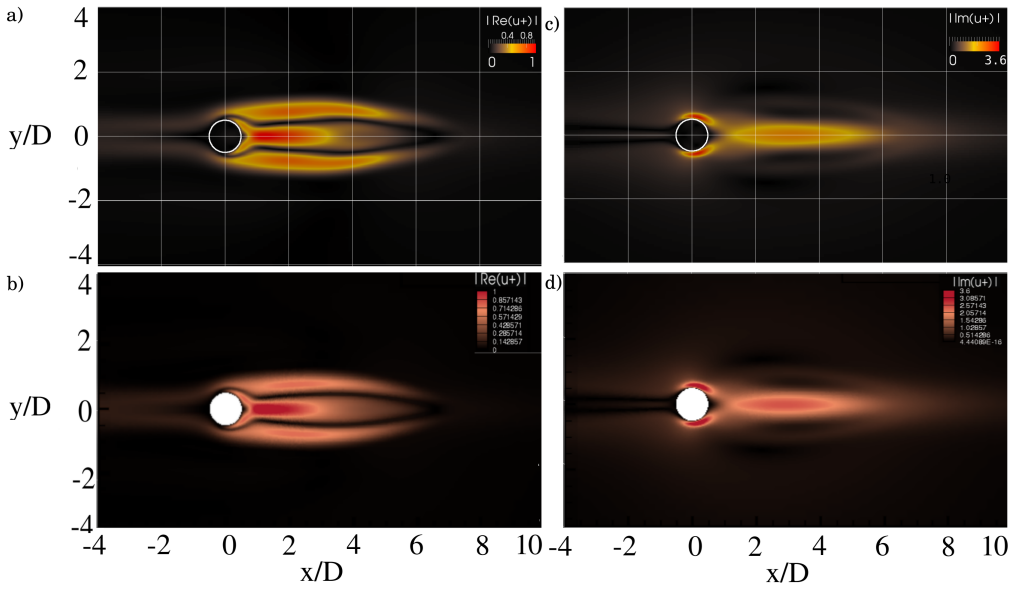


Figure 5.4.: *: Growth rate σ and pulsation ω sensitivities to force modifications compared to the literature. (a) Present growth rate sensibility, (b) growth rate sensibility of Meliga, (c) Present pulsation sensibility, (d) pulsation sensibility of Meliga. Flox past a fixed cylinder at $Re = 100$.*

Conclusion

In this thesis we have proposed a new immersed boundary method in OpenFOAM to simulate incompressible flows past bluff bodies. This IBM, originally proposed by Pinelli et al. [99] , is accurate and versatile for the study of unsteady/deforming structures, as it relies only on the accuracy of the interpolation and spreading steps, which are independent of the complexity of the geometry. The IBM has been incorporated into the native PISO solver of OpenFOAM in particular the changes to satisfy the two-constraints problem involved by the imposition of both the no-slip and the divergence free conditions on the velocity at the solid boundary. A careful and original verification study has been provided using a manufactured solution, which may be applied in a more general context for algorithms using IBM. The efficiency and the accuracy of the new algorithm has been shown on various 2D and 3D well-documented test cases of the literature, for flows around fixed cylinders and sphere, and for Reynolds numbers ranging from $Re = 30$ to $Re = 300$, i.e. from 2D steady to 3D unsteady regimes. These validation tests have shown a good agreement with available numerical and experimental results of the literature.

Validation tests have been further extended to turbulent configurations. Details on the forcing term definition and on the treatment of the immersed surfaces have been discussed in this framework. The IBM PISO algorithm has been extended to hybrid DDES and IDDES models, native of OpenFOAM. Comparison with available data of the litrature for the flow past a fixed cylinder at $Re = 3900$ and past a sphere ar $Re10000$ have confirm the reliability of the OpenFOAM simulations in this context. The formalism has been also extended to boundary-layer model by incorporating a wall function. The aim in the future is to be able to use a coarse mesh in the near wall region able to reproduce accurately the turbulent skin friction C_f and the velocity profile u^+. Despite the rather good agreement with data of the the literature (encouraging the use of the present approach for complex industrial applications), a number of points still need a further investigation, as the account for force-field effects on the mean velocity profile described in the work of Gerasimov [32] or Craft et al. [14] .

The capability of the new solver has been further extended to deal with fluid structure interaction. The flow past a circular fixed and moving cylinder has been investigated. Weak and strong coupling have been tested to couple the motion of the cylinder with the motion of the fluid in a fixed computational mesh. The validation of the methods was conducted in a laminar flow from $Re = 75$ to $Re = 150$ on cross flow and in-line oscillations. The lock-in regime was observed both with forced oscillation and free oscillations. We concluded that this new IBM PISO algorithm is able to reproduce many of the phenomena observed during the tests, at least at low Reynolds numbers. Further studies will focus on three-dimensional simulations in order to assess the effect of cylinder motion on the flow. It is then planned to extend the range of Reynolds number by introducing turbulence modeling at moderate (see Ref [44]) and high Reynolds number in order to tackle industrial problems.

Finally, the IBM-PISO solver has been modified in order to perform sensibility analysis for two-dimensional base-flow and force modifications in laminar regime. The flow past a cylinder at $Re = 100$ has been considered. A linearized and adjoint Navier-Stokes solvers have been developped in this framework. Validation tests have shown the IBM PISO algorithm is able to predict the main features as the sensitivity field of a perturbation to force modifications. Along with these results we

are envisioning as a future challenges the analysis of three-dimensional turbulent flows (following the work of Meliga et al. [75] [78]) and the vortex induced vibration sensibility analysis (following the methodology described in Ref [83]) for more complex geometries together with the actual IBM method.

Bibliography

[1] E. Achenbach. "Vortex shedding from spheres". In: *Journal of Fluid Mechanics* 62 (2) (1974), pp. 209–221. DOI: 10.1017/S0022112074000644 (cit. on p. 108).

[2] E. Balaras. "Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations". In: *Computers and Fluids* 33.3 (2004), pp. 375–404 (cit. on p. 29).

[3] D. Barkley and R.D. Henderson. "Three-dimensional Floquet stability analysis of the wake of a circular cylinder". In: *J. Fluid Mech.* 322 (1996), pp. 215–241 (cit. on p. 57).

[4] R.P. Beyer and R.J. LeVeque. "Analysis of a One-Dimensional Model for the Immersed Boundary Method". In: *SIAM Journal on Numerical Analysis* 29.2 (1992), pp. 332–364 (cit. on p. 28).

[5] H.M. Blackburn and R.D. Henderson. "A study of two-dimensional flow past an oscillating cylinder". In: *J. Fluid Mech.* 385 (1999), pp. 255–286 (cit. on pp. 55, 57, 78–82).

[6] R. D. Blevins. "The effect of sound on vortex shedding from cylinders". In: *J. Fluid Mech.* 161 (1985), pp. 217–237 (cit. on p. 35).

[7] M. Breuer. "Large eddy simulation of the subcritical flow past a circular cylinder : numerical and modeling aspects". In: *Physics of Fluids* 12 (2) (2000), pp. 403–417 (cit. on p. 99).

[8] F.J. Gómez Carrasco. "Matrix-free time-stepping methods for the solution of TriGlobal instability problems". PhD thesis. School of Aeronautics, Universidad Politécnica de Madrid, Sept. 2013 (cit. on p. 34).

[9] J. Chattot and Y. Wang. "Improvement treatment of intersecting bodies with the chimera method and validation with a simple fast flow solver". In: *Computer Fluids* 27 (1998), pp. 721–740 (cit. on p. 26).

[10] Y. Cheny and O. Botella. "Set Method for the Computation of Incompressible Viscous Flows in Complex Moving Geometries with Good Conservation Properties". In: *Journal of Computational Physics* 229 (2010), pp. 1043–1076 (cit. on p. 27).

[11] G.S. Constantinescu and K.D. Squires. "LES and DES Investigations of Turbulent Flow over a Sphere at Re = 10,000". In: *Flow, Turbulence and Combustion* 70 (2003), pp. 267–298. DOI: 10.1023/B:APPL.0000004937.34078.71 (cit. on pp. 108–110).

[12] C. Cossu and L. Morino. "On the instability of a spring-mounted circular cylinder in a viscous flow at low Reynolds numbers." In: *J. Fluids Struct.* 1 (14) (2000), pp. 183–196 (cit. on p. 117).

[13] M. Coutanceau and 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". In: *J. Fluid Mech* 79 (1977), pp. 231–256. DOI: 10.1017/S0022112077000135 (cit. on p. 55).

[14] T.J. Craft, S.E. Gant, A.V. Gerasimov, H. Iacovides, and B.E. Launder. "Development and application of wall-function treatments for turbulent forced and mixed convection flows". In: *Fluid Dynamics Research* 38.2-3 (2006), p. 127. URL: http://iopscience.iop.org/1873-7005/38/2-3/A06 (cit. on p. 124).

[15] V. D'Alessandro, S. Montelpareb, and R. Riccia. "Detached–eddy simulations of the flow over a cylinder at Re = 3900 using OpenFOAM". In: *Computers and Fluids* 136 (2016), pp. 152–169 (cit. on pp. 102, 105, 106).

[16] A.W. Date. "Solution of Navier—Stokes equations on non-staggered grid". In: *Int. J. Heat and Mass transfer* 36 (1993), pp. 1913–1922 (cit. on p. 22).

[17] M.D. Detullio. "Development of an Immersed Boundary method for the solution of the preconditioned Navier-Stokes equations". PhD thesis. Bari, Italy: POLITECNICO DI BARI, 2006 (cit. on p. 95).

[18] E. DeVilliers. "The potential of large eddy simulation for the modeling of wall bounded flows". In: *PhD thesis, Imperial college of science, technology and medicine* (2006) (cit. on p. 39).

[19] A. Dipankar, T.K. Sengupta, and S.B. Talla. "Suppression of vortex shedding behind a circular cylinder by another control cylinder at low Reynolds numbers". In: *J. Fluid Mech.* 573 (2007), pp. 171–190 (cit. on p. 34).

[20] F. Domenichini. "On the consistency of the direct forcing method in the fractional step solution of the Navier-–Stokes equations". In: *Journal of Computational Physics* 227.12 (2008), pp. 6372–6384 (cit. on p. 30).

[21] J.P. Van Doormaal and G.D. Raithby. "Enhancements of the SIMPLE method for predicting incompressible fluid flows". In: *Numerical Heat Transfer* 7 (1984), pp. 147–163. DOI: 10.1080/01495728408961817 (cit. on p. 24).

[22] F. Duarte, R. Gormaz, and S. Natesan. "Arbitrary Lagrangian-Eulerian method for Navier-Stokes equations with moving boundaries". In: *Computer Methods in Applied Mechanics and Engineering* 193 (2004), pp. 4819–4836 (cit. on p. 26).

[23] ESI-OpenCFD. "http://wwww.openfoam.com". In: *OpenFoam* (2011) (cit. on p. 21).

[24] E.A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations". In: *Journal of Computational Physics* 161.1 (2000), pp. 35–60 (cit. on pp. 29, 30).

[25] J. Favier, A. Dauptain, and A. Bottaro. "Passive separation control using a self-adaptative hairy coating". In: *Journal of Fluids Mechanics* 627 (2009), pp. 451–483. DOI: 10.1017/S002211200900611 (cit. on p. 34).

[26] J. Favier, C. Li, L. Kamps, A. Revell, J. O'Connor, and C. Bruecker. "PELskin project - part I - Fluid-structure interaction in a row of flexible flaps : a reference study in oscillating channel flow". In: *Meccanica* 52 (8) (2017), pp. 1767–1780. DOI: 10.1007/s11012-016-0521-0 (cit. on p. 34).

[27] F.N. Felten and T.S. Lund. "DNS/LES Progress and Challenges". In: *Proceedings of the Third AFOSR International Conference on DNS/LES*. 2001 (cit. on p. 22).

[28] J.H. Ferziger and M. Peric. *Computational Methods in Fluid Dynamics*. New-York : Springer-Verla, 1996. ISBN: 978-3-642-56026-2. DOI: 10.1007/978-3-642-56026-2 (cit. on pp. 17, 25, 39).

[29] F. Flores, R. Garreaud, and Muñoz R.C. "CFD simulations of turbulent buoyant atmospheric flows over complex geometry: Solver development in OpenFOAM". In: *Computers and Fluids* 82 (2013), pp. 1–13. DOI: 10.1016/j.compfluid.2013.04.029 (cit. on p. 21).

[30] R.M. Franck and R.B. Lazarus. “Mixed Eulerian-Lagrangian method”. In: *Methods in Computational Physics : Fundamental methods in Hydrodynamics*. Ed. by B. Alderand S. Fernbach and M. Rotenberg. Vol. 3. Academic Press: New York, 1964, pp. 47–67 (cit. on p. 26).

[31] L. Gao, J. Xu, and G. Gao. “Numerical Simulation of Turbulent Flow past Airfoils on OpenFOAM”. In: *Computers and Fluids* 31 (2012), pp. 756–761. DOI: 10.1016/j.proeng.2012.01.1098 (cit. on p. 21).

[32] A.V. Gerasimov. “Development and application of an analytical wall-function strategy for modelling forced, mixed and natural convection flows”. PhD thesis. Manchester: Department of Mechanical, Aerospace and Manufacturing Engineering, UMIST, 2003 (cit. on p. 124).

[33] M. Giacobello, A. Ooi, and S. Balachandar. “Wake structure of a transversely rotating sphere at moderate Reynolds numbers”. In: *Journal of Fluid Mechanics* 621 (2009), pp. 103–130 (cit. on pp. 61, 63, 64).

[34] F. Giannetti and P. Luchini. “Structural sensitivity of the first instability of the cylinder wake”. In: *J. Fluid Mech.* 581 (2007), pp. 167–197 (cit. on pp. 36, 117).

[35] R. Glowinski, T.-W. Pan, and T.I. Hesla. “A distributed Lagrange multiplier/fictitious domain method for particulate flows”. In: *International Journal of Multiphase Flow* 25 (1999), pp. 755–794. DOI: 10.1016/S0301-9322(98)00048-2 (cit. on p. 27).

[36] D. Goldstein, R. Handler, and L. Sirovich. “Modeling a No-Slip Flow Boundary with an External Force Field”. In: *Journal of Computational Physics* 105.2 (1993), pp. 354–366 (cit. on p. 28).

[37] E. Guilmineau and P. Queutey. “A numerical simulation of vortex shedding from an oscillating circular cylinder”. In: *J. Fluids Struct.* 16.6 (2002), pp. 773–794 (cit. on pp. 56, 57).

[38] K. Hanjalic and B.E. Launder. “A Reynolds stress model of turbulence and its application to thin shear flows”. In: *Journal of Fluid Mechanics* 52 (4) (1972), pp. 609–638 (cit. on p. 32).

[39] M. Heil, A. Hazel, and J. Boyle. “Solvers for large-displacement fluid-structure interaction problems: Segregated versus monolithic approaches”. In: *Computational Mechanics* 43 (1) (2008), pp. 91–101. DOI: 10.1007/s00466-008-0270-6 (cit. on pp. 25, 26).

[40] R.D. Henderson. “Details of the drag curve near the onset of vortex shedding”. In: *Phys.Fluids* 7.9 (1995), pp. 2102–2104 (cit. on pp. 57, 79).

[41] C. Hill. “A theoretical approach for analyzing the restabilization of wakes”. In: *AIAA Paper* 92 (1992), p. 0067 (cit. on pp. 36, 117, 120).

[42] C.W. Hirt, A.A. Amsden, and J.L. Cook. “An arbitrary Lagrangian-Eulerian computing method for all flow speeds”. In: *J. Comput. Phys.* 14 (1974), pp. 227–253 (cit. on p. 26).

[43] J. Hoeppfner, A. Bottaro, and J. Favier. “Mechanisms of non-modal energy growth in a channel flow between compliant walls”. In: *Journal of Fluids Mechanics* 642 (2010), pp. 489–507. DOI: 10.1017/S0022112009991935 (cit. on p. 34).

[44] F. S. HOVER, A. H. TECHET, and M. S. TRIANTAFYLLOU. “Forces on oscillating uniform and tapered cylinders in cross flow”. In: *Journal of Fluid Mechanics* 363 (1998), pp. 97–114. DOI: 10.1017/S0022112098001074 (cit. on p. 124).

[45] H.H. Hu, D.D. Joseph, and M.J. Crochet. “Crochet Direct Simulation if fluid particle motions”. In: *Theoretical and Computational Fluid Dynamics* 3 (1992), pp. 285–306 (cit. on p. 26).

[46] T. Ikeno and T. Kajishima. “Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations”. In: *Journal of Computational Physics* 226.2 (2007), pp. 1485–1508 (cit. on p. 30).

[47] L. Isoardi, G. Chiavassa, and G. Ciraolo. “Penalization modeling of a limiter in the Tokamak edge plasma”. In: *Journal of Computational Physics* 229 (2010), pp. 2220–2235 (cit. on p. 27).

[48] R.I. Issa. “Solution of the implicitly discretised fluid flow equations by operator-splitting”. In: *Journal of Computational Physics* 62 (1985), pp. 40–65. DOI: 10.1016/0021-9991(86)90099-9 (cit. on pp. 24, 25).

[49] H. Jasak. “Error analysis and estimation for the Finite Volume method with applications to fluid flows”. In: *PhD. Thesis, Imperial College, University of London* (1996) (cit. on p. 39).

[50] H. Jasak. “Dynamic mesh handling in OpenFoam”. In: *47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2009 (cit. on p. 26).

[51] H. Jasak, D. Rigler, and Z. Tukovic. “Design and implementation of Immersed Boundary Method with discrete forcing approach for boundary conditions”. In: *In proceedings of 6th European Congress on Computational Fluid Dynamics - ECFD VI Barcelona, Spain* ISBN: 978-849428447-2 (2014) (cit. on pp. 29, 30).

[52] J. Jeong and F. Hussain. “On the identification of a vortex”. In: *Journal of Fluids Mechanics* 285 (1995), pp. 69–94. DOI: 10.1017/S0022112095000462 (cit. on p. 107).

[53] T.A. Johnson and V.C. Patel. “Flow past a sphere up to a Reynolds number of 300”. In: *Journal of Fluid Mechanics* 378 (1998), pp. 19–70 (cit. on pp. 60–66).

[54] C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H.G. Mathies. “Non-linear fluid-structure interaction problem. Part I: implicit partitioned algorithm, nonlinear stability proof and validation examples.” In: *Computational Mechanics* 47 (2011), pp. 305–323 (cit. on p. 26).

[55] J. Kim, D. Kim, and H. Choi. “An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries”. In: *Journal of Computational Physics* 171.1 (2001), pp. 132–150 (cit. on p. 29).

[56] A.N. Kolmogorov. “A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high Reynolds number”. In: *Journal of Fluid Mechanics* 13 (1962), pp. 82–85. DOI: 10.1017/S0022112062000518 (cit. on p. 30).

[57] A.N. Kolmogorov. *Turbulence and stochastic processes: Kolmogorov’s ideas 50 years on - Dissipation of energy in the locally isotropic turbulence*. 1962. DOI: 10.1098/rspa.1991.0076 (cit. on p. 30).

[58] E. Komena and A. Shamsa. “Quasi-DNS capabilities of OpenFOAM for different mesh types”. In: *Computers and Fluids* 96 (2014), pp. 87–104. DOI: 10.1016/j.compfluid.2014.02.013 (cit. on p. 21).

[59] I. Korkischko and J. Meneghini. “Suppression of vortex-induced vibration using moving surface boundary-layer control”. In: *J. Fluids Struct.* 34 (2012), pp. 259–270. DOI: 10.1016/j.jfluidstructs.2012.05.010 (cit. on p. 35).

[60] A. Kravchenko and P. Moin. "Numerical studies of flows over a circular cylinder at $Re_D = 3900$". In: *Physics of Fluids* 12 (2) (2000), pp. 403–417 (cit. on p. 102).

[61] B.E. Launder and D.B. Spalding. "The numerical computation of turbulent flows". In: *Comp. Methods Appl. Mech. Eng.* 3 (1974), pp. 269–289. DOI: 10.1016/0045-7825(74)90029-2 (cit. on pp. 33, 34, 96).

[62] J.C. Lecordier, L. Hamma, and P. Paranthoen. "The control of vortex shedding behind heated circular cylinders at low Reynolds numbers". In: *Exp. Fluids* 10 (1991), pp. 224–229 (cit. on p. 35).

[63] E. Lefrançois and J.P. Boufflet. "An introduction to fluid-structure interaction: Application to the piston problem". In: *SIAM Review* 52 (2003), pp. 747–767 (cit. on p. 26).

[64] P. LeGrand. "Formes curvilin eaires avanc ees pour la mod elisation centr ee objet des ecoulements souterrains par la m ethode des el ements analytiques." PhD thesis. Universit e Jean Monnet - Saint-Etienne: Ecole Nationale Superieure des Mines de Saint-Etienne, July 2003 (cit. on p. 26).

[65] Z. Li, J. Favier, U. D'Ortona, and S. Poncet. "A numerical approach to combine immersed boundary method and lattice Boltzmann model for single- and multi-component fluid flows". In: *Journal of Computational Physics* in press (2015) (cit. on p. 42).

[66] W.K. Liu, S. Jun, and Y.F. Zhang. "Reproducing kernel particle methods". In: *Int. J. Num. Meth. Fuids* 20 (8) (1995), pp. 1081–1106 (cit. on p. 42).

[67] L. Lourenco and C. Shi. *Characteristics of the plane turbulent near wake of a circular cylinder, a particle image velocimetry.* Ed. by P. Beaudan and P. Moin. 1993 (cit. on pp. 102, 103, 105, 106).

[68] X.Y. Lu and C. Dalton. "Calculation of the timing of vortex formation from an oscillating cylinder". In: *J. Fluids Struct.* 10.5 (1996), pp. 527–541 (cit. on p. 57).

[69] P. Luchini and A. Bottaro. "Adjoint Equations in Stability Analysis". In: *Annual Review of Fluid Mechanics* 46.1 (2014), pp. 493–517. DOI: 10.1146/annurev-fluid-010313-141253 (cit. on p. 117).

[70] D.A. Lysenko, I.S. Ertesvåg, and K.E. Rian. "Modeling of turbulent separated flows using OpenFOAM". In: *Computers and Fluids* 80 (2013), pp. 408–422. DOI: 10.1016/j.compfluid.2012.01.015 (cit. on p. 21).

[71] O. Marquet, D. Sipp, and L. Jacquin. "Sensitivity analysis and passive control of cylinder flow". In: *J. Fluid Mech.* 615 (2008), pp. 221–252. DOI: 10.1017/S0022112008003662 (cit. on pp. 36, 117, 119, 120).

[72] R.L. Meakin. "Moving body overset grid methods for complete aircraft tiltrotor simulations". In: *AIAA paper* 3350 (1993) (cit. on p. 26).

[73] H.F. Meier, J.J.N. Alves, and M. Mori. "Comparison between staggered and collocated grids in the finite-volume method performance for single and multi-phase flows". In: *Computers and Chemical Engineering* 23 (1999), pp. 247–262. DOI: 10.1016/S0098-1354(98)00270-1 (cit. on p. 22).

[74] M. Meldi, M.V. Salvetti, and P. Sagaut. "Quantification of errors in large-eddy simulations of a spatially evolving mixing layer using polynomial chaos". In: *Physics of Fluids (1994-present)* 24.3 (2012), p. 035101 (cit. on p. 21).

[75] P. Meliga, E. Boujo, G. Pujals, and F. Gallaire. “Sensitivity of aerodynamic forces in laminar and turbulent flow past a square cylinder”. In: *Physics of Fluids* 26 (2014), p. 104101. DOI: 10.1063/1.4896941 (cit. on p. 125).

[76] P. Meliga and D. Sipp J.M. Chomaz. “Open-loop control of compressible afterbody flows using adjoint methods”. In: *Physics of Fluids* 22 (5) (2010), pp. 137–167. DOI: 10.1063/1.3425625 (cit. on p. 117).

[77] P. Meliga and J.M. Chomaz. “An asymptotic expansion for the vortex-induced vibrations of a circular cylinder”. In: *Journal of Fluid Mechanics* 671 (2011), pp. 137–167. DOI: 10.1017/S0022112010005550 (cit. on p. 117).

[78] P. Meliga, G. Pujals, and E. Serre. “Sensitivity of 2-D turbulent flow past a D-shaped cylinder using global stability”. In: *Physics of Fluids* 24.6 (2012), p. 061701. DOI: 10.1063/1.4724211 (cit. on p. 125).

[79] M. Minguez, R. Pasquetti, and E. Serre. “High-order large-eddy simulation of flow over the “Ahmed body” car model”. In: *Phys Fluids* 20 (2008) (cit. on p. 27).

[80] R. Mittal and S. Balachandar. “Generation of streamwise vortical structures in bluff body wakes”. In: *Physique Review Letter* 75 (1995), pp. 1300–1303 (cit. on p. 59).

[81] R. Mittal and G. Iaccarino. “Immersed boundary methods”. In: *Annual Review of Fluid Mechanics* 37 (2005), pp. 239–261 (cit. on p. 29).

[82] R. Mittal and P. Moin. “Suitability of upwind-biased finite difference schemes for Large-Eddy simulation of turbulent flows”. In: *AIAA J.* 35 (8) (1997), pp. 1415–1417 (cit. on p. 102).

[83] S. Mittal. “Lock-in in vortex-induced vibration”. In: *Journal of Fluid Mechanics* 794 (2016), pp. 565–594. DOI: 10.1017/jfm.2016.157 (cit. on pp. 117, 125).

[84] J. P. Mohd-Yusof and R.J. LeVeque. “Combined Immersed-Boundary/B-spline methods for simulations of flow in complex geometries”. In: *Center for Turbulence Research - Annual REsearch Briefs* (1997), pp. 317–327 (cit. on p. 29).

[85] S.A. Morton, R.B. Melville, and M.R. Visbal. “Accuracy and Coupling Issues of Aeroelastic Navier-Stokes Solutions on Deforming Meshes”. In: *Journal of Aircraft* 35 (5) (1998), pp. 798–805. DOI: 10.2514/2.2372 (cit. on p. 25).

[86] I. Nakamura. “Steady wake behind a sphere”. In: *Phys Fluids* 19 (1976), p. 5 (cit. on p. 61).

[87] S. Nix, Y. Imai, T. Ishikawa, and T. Yamaguchi. “Boundary Element Analysis Of Deformation And Movement Of A Capsule And A Red Blood Cell Close To The Wall”. In: *Computer Methods in Applied Mechanics and Engineering* 53 (2012), pp. 191–199. DOI: 10.2495/BE120171 (cit. on p. 26).

[88] W.F. Noh. “CEL:A time-dependent two-space dimensional coupled Eulerian-Lagrangian code”. In: *Methods in Computational Physics*. Ed. by B. Alderand S. Fernbach and M. Rotenberg. Vol. 3. Academic Press: New York, 1964, pp. 117–179 (cit. on p. 26).

[89] C. Norberg. “Pressure forces on a circular cylinder in cross flow”. In: *Bluff-Body Wakes, Dynamics and Instabilities* (1993), pp. 275–278 (cit. on p. 59).

[90] C. Norberg. “An experimental investigation of the flow around a circular cylinder: influence of aspect ratio”. In: *J. Fluid Mech* 258 (1994), pp. 287–316 (cit. on pp. 56, 57, 79).

[91] P. Parnaudeau, J. Carlier, D. Heitz, and E. Lamballais. “Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900”. In: *Physics of Fluids* 20 (6) (2008), pp. 441–453 (cit. on pp. 101–106).

[92] S.V. Patankar. CRC Press, 1980. ISBN: 0-89116-522-3 (cit. on pp. 22, 24).

[93] S.V. Patankar and D.B. Spalding. “A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows”. In: *International Journal of Heat and Mass Transfer* 15 (1972), pp. 1787–1806. DOI: 10.1016/0017-9310(72)90054-3 (cit. on p. 24).

[94] G.A. Patino, R.S. Gioria, and J.R. Meneghini. “Evaluating the control of a cylinder wake by the method of sensitivity analysis”. In: *Phys. of Fluids* 29 (2017), p. 044103 (cit. on pp. 35, 36).

[95] M. Pepona. “Modèle de frontières immergées pour la simulation d’écoulements de fluide en interaction avec des structures poreuses.” PhD thesis. Marseille: Aix-Marseille Université, Nov. 2016 (cit. on p. 42).

[96] Charles S Peskin. “Flow patterns around heart valves: A numerical method”. In: *Journal of Computational Physics* 10.2 (1972), pp. 252–271 (cit. on p. 27).

[97] Charles S Peskin. “Numerical analysis of blood flow in the heart”. In: *Journal of Computational Physics* 25.3 (1977), pp. 220–252 (cit. on pp. 27, 28).

[98] C.S. Peskin. “The immersed boundary method”. In: *Acta Numerica* 11 (2002), pp. 1–39 (cit. on p. 27).

[99] A. Pinelli, I.Z. Naqavi, U. Piomelli, and J. Favier. “Immersed-boundary methods for general finite-difference and finite-volume Navier–Stokes solvers”. In: *Journal of Computational Physics* 229.24 (2010), pp. 9073–9091. DOI: 10.1016/j.jcp.2010.08.021 (cit. on pp. 30, 37, 38, 40, 42, 43, 54–57, 124, 138).

[100] S. Piperno and C. Farhat. “Partitioned procedures for the transient solution of coupled aeroelastic problems Part II: energy transfer analysis and three-dimensional applications”. In: *Computer Methods in Applied Mechanics and Engineering* 190 (2425) (2001), pp. 3147–3170. DOI: 10.1016/S0045-7825(00)00386-8 (cit. on p. 25).

[101] S.B. Pope. *Turbulent Flow*. Cambridge University Press. book, 2000 (cit. on p. 32).

[102] C. Pozrikidis. Cambridge University Press, 1992. ISBN: 0-521-40502-5 (cit. on p. 26).

[103] J.O. Pralits, L. Brandt, and F. Giannetti. “Instability and sensitivity of the flow around a rotating circular cylinder”. In: *J. Fluid Mech.* 650 (2010), pp. 513–536 (cit. on p. 36).

[104] T.K. Prasanth and S. Mittal. “Vortex-induced vibrations of a circular cylinder at low Reynolds numbers”. In: *J. Fluid Mech.* 594 (2008), pp. 463–491. DOI: 10.1017/S0022112007009202 (cit. on pp. 86–88).

[105] L. Qu, C. Norberg, L. Davidson, S.H Peng, and F. Wang. “Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200”. In: *Journal of Fluids and Structures* 39 (2013), pp. 347–370 (cit. on p. 59).

[106] B.N. Rajani, A. Kandasamy, and M. Sekhar. “Numerical simulation of laminar flow past a circular cylinder”. In: *Applied Mathematical Modelling* 33 (2009), pp. 1228–1247 (cit. on p. 59).

[107] C.M. Rhie and W.L. Chow. "Numerical study of the turbulent flow past anairfoil with trailing edge separation". In: *AIAA Journal* 21 (11) (1983), pp. 1525–1532. DOI: 10.2514/3.8284 (cit. on p. 22).

[108] I. Rodriguez, O. Lehmkuhl, R Borrell, and A. Oliva. "Flow dynamics in the turbulent wake of a sphere at sub-critical Reynolds numbers". In: *Computers and Fluids* 80 (2013), pp. 233–243. DOI: 10.1016/j.compfluid.2012.03.009 (cit. on pp. 107–109).

[109] W. Rojanaratanangkule and A. Hokpunna. "Performance of high-order schemes on collocated and staggered grids". In: *Journal of Research and Applications in Mechanical Engineering* 3 (1) (2015), pp. 22–28. DOI: 10.14456/jrame.2015.2 (cit. on p. 22).

[110] A.M. Roma, C.S. Peskin, and M.J. Berger. "An adaptive version of the immersed boundary method". In: *Journal of Computational Physics* 153 (1999), pp. 509–534 (cit. on pp. 42, 51).

[111] F.W. Roos and W.W. Willmarth. "Some experimental results on sphere and disk drag". In: *AIAA Journal* 9 (2) (1971), pp. 285–291. DOI: 10.1063/1.3425625 (cit. on p. 64).

[112] E.M. Saiki and S. Biringen. "Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method". In: *J. Comput. Phys.* 123 (1996), p. 450. DOI: 10.1006/jcph.1996.0036 (cit. on p. 29).

[113] H. Sakamoto. "A Study on Vortex Shedding From Spheres in a Uniform Flow". In: *Journal of Fluids Engineering* 112 (4) (1990), pp. 386–392. DOI: 10.1115/1.2909415 (cit. on pp. 107, 108).

[114] H. Schlichting. Vol. 7. 1979. ISBN: 0-07-055334-3 (cit. on pp. 112–114).

[115] B. Selma, M. Désilets, and P. Proulx. "Optimization of an industrial heat exchanger using an open-source CFD code". In: *Applied Thermal Engineering* 69.1–2 (2014), pp. 241–250 (cit. on p. 21).

[116] L. Shen and E.S. Chan P. Lin. "Calculation of hydrodynamic forces acting on a submerged moving object using immersed boundary method". In: *Computers and Fluids* 38 (3) (2009), pp. 691–703. DOI: 10.1016/j.compfluid.2008.07.002 (cit. on pp. 82, 84).

[117] D. Shiels, A. Leonard, and A. Roshko. "Flow-induced vibration of a circular cylinder at limiting structural parameters". In: *J. Fluids Struct.* 15 (2001), pp. 3–21. DOI: 10.1006/jfls.2000.033 (cit. on pp. 82–84).

[118] M.L. Shur, M.K. Strelets P.R.Spalart, and A.K. Travina. "A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities". In: *International Journal of Heat and Fluid Flow* 29 (6) (2008), pp. 1638–1649. DOI: 10.1016/j.ijheatfluidflow.2008.07.001 (cit. on p. 92).

[119] A. Skillen, A. Revell, A. Pinelli, U. Piomelli, and J. Favier. "Flow over a wing with leading-edge undulations". In: *AIAA Journal* 53 (2) (2015), pp. 464–472. DOI: 10.2514/1.J053142 (cit. on p. 34).

[120] J. Smagorinsky. "General circulation experiments with the primitive equations I. the basic experiment". In: *Mon. Weather Rev.* 91 (3) (1963), pp. 99–164. DOI: 10.1175/1520-0493(1963)091<0099\%3AGCEWTP>2.3.CO\%3B2 (cit. on p. 32).

[121] J. Smagorinsky. "The large reynolds number, asymptotic theory of turbulent boundary layers". In: *International Journal of Engineering Science* 10 (10) (1972), pp. 851–873. DOI: 10.1016/0020-7225(72)90055-9 (cit. on p. 33).

[122] J. Smagorinsky. “Strategies for turbulence modelling and simulations”. In: *Int. J. Heat Fluid Flow* 21 (2000), pp. 252–263. DOI: 10.1016/S0142-727X(00)00007-2 (cit. on p. 32).

[123] P.R. Spalart and S.R. Allmaras. “A One-Equation Turbulence Model for Aerodynamic Flows”. In: *Recherche Aerospatiale* 1 (1994), pp. 5–21 (cit. on p. 90).

[124] P.R. Spalart and S.R. Allmaras. “Detached-Eddy Simulation”. In: *Annual Review of Fluid Mechanic* 41 (2009), pp. 181–202. DOI: 10.1146/annurev.fluid.010908.165130 (cit. on p. 90).

[125] P.R. Spalart, S. Deck, M.L. Shur, K.D. Squires, M.K. Strelets, and A. Travin. “A new version of detached-eddy simulation, resistant to ambiguous grid densities”. In: *Theor. Comput. Fluid Dyn.* 20 (2006), pp. 181–195. DOI: 10.1007/s00162-006-0015-0 (cit. on pp. 91, 92).

[126] P.R. Spalart, W.H. Jou, and S.R. Allmaras. “Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach”. In: *In: Liu, C., Liu,Z. (Eds.), Advances in DNS/LES, 1st AFOSR Int. Conf. On DNS/LES, Greyden Press, Louisiana Tech University* (1994) (cit. on pp. 33, 90).

[127] M.B. Stadler, N.R. Rapaka, and S. Sarkar. “Large eddy simulation of the near to intermediate wake of a heated sphere at $Re = 10,000$”. In: *International Journal of Heat and Fluid Flow* 49 (2014), pp. 2–10. DOI: 10.1016/j.ijheatfluidflow.2014.05.013 (cit. on pp. 108, 109).

[128] J. Steger and F.D.J. Benek. “A chimera grid scheme”. In: *Advances in Grid GEneration* 5 (1983), pp. 59–69 (cit. on p. 26).

[129] O.D.L. Strack. Ed. by Englewood Cliffs. New Jersey: Prentice Hall, 1989. ISBN: 0-13-365412-5 (cit. on p. 26).

[130] P.J. Strykowski and K.R. Sreenivasan. “On the formation and suppression of vortex ‘shedding’ at low Reynolds numbers”. In: *J. Fluid Mech.* 218 (1990), pp. 71–107 (cit. on pp. 34, 36).

[131] G.R. Tabor and M.H. Baba-Ahmadi. “Inlet conditions for large eddy simulation: A review”. In: *Computers & Fluids* 39.4 (2010), pp. 553–567 (cit. on p. 21).

[132] K. Taira and T. Colonius. “The immersed boundary method: A projection approach”. In: *Journal of Computational Physics* 225.10 (2007), pp. 2118–2137 (cit. on pp. 29, 30).

[133] H. Takami and H.B. Keller. “Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder”. In: *Physics of Fuids* 12 (12) (1969), pp. II–51. DOI: 10.1063/1.1692469 (cit. on p. 121).

[134] S. Taneda. “Experimental investigation of the wake behind a sphere at low Reynolds numbers”. In: *J. Phys. Soc. Japan* 11 (1956), p. 1104 (cit. on p. 61).

[135] S. Taneda. “Experimental Investigation of the Wakes behind Cylinders and Plates at Low Reynolds Numbers”. In: *Journal of the Physical Society of Japan* 11 (3) (1956), pp. 302–307. DOI: 10.1143/JPSJ.11.302 (cit. on p. 121).

[136] A.G. Tomboulides and S.A. Orszag. “Numerical investigation of transitional and weak turbulent flow past a sphere”. In: *Journal of Fluid Mechanics* 416 (2000), pp. 45–73 (cit. on p. 61).

[137] A.G. Tomboulides, S.A. Orszag, and G.E. Karniadakis. “Direct and large-eddy simulation of axisymmetruc wakes”. In: *AIAA Paper* 93 (1993), p. 0546 (cit. on p. 64).

[138] M. Towara, M. Schanen, and U. Naumann. “MPI-Parallel Discrete Adjoint OpenFOAM”. In: *Computers and Fluids* 51 (2015), pp. 19–28. DOI: 10.1016/j.procs.2015.05.181 (cit. on p. 21).

[139] D.J. Tritton. “Experiments on the flow past a circular cylinder at low Reynolds numbers”. In: *J. Fluid Mech* 6 (1959), pp. 547–567. DOI: 10.1017/S0022112059000829 (cit. on pp. 55, 121).

[140] J.G. Trulio. *Theory and Structure of the AFTON Codes*. Tech. rep. AFWL-TR-66-19. Air Force Weapons Laboratory: Kirtland Air Force Base, 1966, pp. 227–253 (cit. on p. 26).

[141] M. Uhlmann. “An immersed boundary method with direct forcing for the simulation of particulate flows”. In: *Journal of Computational Physics* 209.2 (2005), pp. 448–476 (cit. on p. 29).

[142] M. Vanella and E. Balaras. “A moving-least-squares reconstruction for embedded-boundary formulations”. In: *Journal of Computational Physique* 228.18 (2009), pp. 6617–6628 (cit. on pp. 54, 57).

[143] H.K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics : The finite volume method*. Vol. 2. Pearson Education, 2007. ISBN: 978-0-13-127498-3 (cit. on p. 24).

[144] V. Vuorinen, J.P. Keskinen, and Duwig C. “On the implementation of low-dissipative Runge-Kutta projection methods for time dependent flows using OpenFOAM”. In: *Computers and Fluids* 93 (2014), pp. 153–163. DOI: 10.1016/j.compfluid.2014.01.026 (cit. on pp. 21, 25).

[145] H. Wan and S.S. Patnaik. “Suppression of vortex-induced vibration of a circular cylinder using thermal effects”. In: *Phys. of Fluids* 28 (2016), p. 123603 (cit. on p. 35).

[146] C. Wieselsberger. “New data on the law of hydro and aerodynamic resistance”. In: *NACA TN* 84 (1922) (cit. on p. 59).

[147] C.H.K. Williamson. “Defining a universal and continuous Strouhal–Reynolds number relationship for the laminar vortex shedding of a circular cylinder”. In: *Phys. Fluids* 31.10 (1988), pp. 2742–2744 (cit. on pp. 56, 57).

[148] C.H.K. Williamson. “Vortex dynamics in the cylinder wake”. In: *Annu. Rev. Fluid. Mech.* 28 (1996), pp. 477–539 (cit. on pp. 57–59).

[149] C.H.K. Williamson and A. Roshko. “Vortex formation in the wake of an oscillating cylinder”. In: *J. Fluids Struct.* 2 (1988), pp. 355–381. DOI: 10.1016/S0889-9746(88)90058-8 (cit. on p. 87).

[150] G. Yun, D. Kim, and H. Choi. “Vortical structures behind a sphere at subcritical Reynolds numbers”. In: *Physics of Fluids* 18 (2006), p. 015102. DOI: 10.1063/1.2166454 (cit. on pp. 107, 108).

[151] D. Zhou, J. Tu, and Y. Bao. “Two degrees of freedom flow-induced vibrations on a cylinder”. In: *The Seventh International Colloquium on Bluff Body Aerodynamics and Applications (BBAA7)*. China, Sept. 2012 (cit. on pp. 26, 84–86).

[152] M. Zijlema, A. Segal, and P. Wesseling. “Finite volume computation of incompressible turbulent flows in general coordinates on staggered Grids”. In: *Int. J. Numer. Meth. Fluids* 20 (1995), pp. 621–640 (cit. on p. 22).

!

ANNEXES

A. IBM Implementation

A.1. Overall description of the interpolation process

The lagrangians points of the mesh are soart out by the processors. If the number of lagrangian points in a cell exeed one, the nearest lagrangian point to the eulerian point is chosen and the other one is desactivated. This is to ensure a well-conditioned system (equation **??**) and a smooth and positive resulting solution ϵ at all lagrangian nodes according to [99].

A supporting box, used to spread the body force on the eulerian mesh, is created around each lagrangian point thanks to the nearsest eulerian point selected. The dimensions of the box are calculated thanks to all the neighbours of the selected eulerian point.

A C++ structure is created for each lagrangian node and contain the coordinates of the supporting box and the labels of the eulerian points includes in this box.

Each lagrangian node have a processor owner which look for the the eulerians points to include in the grid through an algorithm which start with the neighbours of the selected cell. The algorithm go through the faces of the selected cells to test if the neighbours cells are also in this supporting box.

Then the kernel interpolation δ_h (equation 2.9) is calculated independently on each processor. The ϵ of equation 2.21 is determined by the method of the biconjugate gradient. At each iteration, the value of the matrix is evaluated by the processors which builds together the matrix of the equation 2.21. Once the value of the ϵ is found for each lagrangian point, the force can be interpolate thanks to the equation 2.7. Each "node owner" gather all the velocities values on differents lagrangian points to calculate the body force value and share the information with the "ghost owners". Then the spreading on the eulerian nodes can be achieved thanks to the equation 2.20 independently on each processor.

A.2. Ghost points

The algorithm go through the faces of the selected cells to test if the neighbours cells are also in this supporting box. If the lagrangian marker is located near a boundary the algorithm would then face a boundary face and will not be able to construct correctly a 3 points supporting box. Then a special treatment is required involving the creation of ghost points. If the algorithm go through a boundary face :

i. If the boundary face is tagged as a processor boundary, the actual structure is tagged as "Ghost point owner". Then the actual processor or "node owner" activate the concerned neighbour processor as a "ghost owner". The "node owner" send to the ghost owner the coordinates of the box of the actual lagrangian point and save the number of the processor as "ghost owner". The "ghost owner" activate one structure for this lagrangian point and run the algorithm to find points in the box.

ii. If the boundary face is tagged as a field boundary, the processor check if the boundary is tagged as a periodic boundary. If so then the algorithm continue to look for cell on the other side of the field. These points will be tagged as "periodic points" and instead of the coordinates of the box and the lagrangian point they refer to delayed coordinates :

$$X_i^{ref} = X_i^{real} \pm |X_i^{max} - X_i^{min}| \tag{.29}$$

with X_i the coordinate on the i-th component (component of the periodic boundary). The Figure .5 present how the actual implementation of the immersed boundary deal with the borders of the field in OpenFoam.

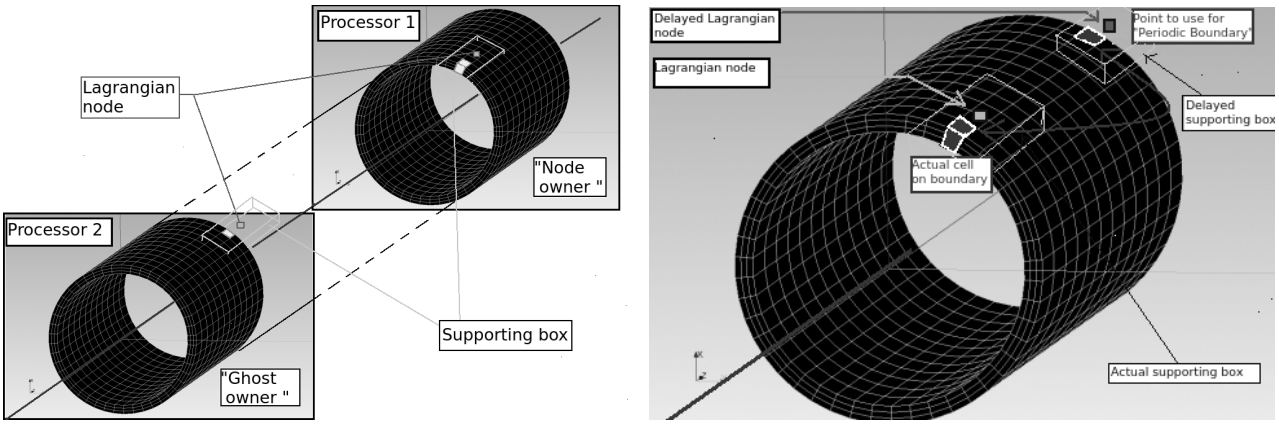


FIGURE .5. : *Boundary implementation in the solver.* ***Left :*** *Processor Boundary ;* ***Right :*** *Periodic boundary*

A.2.1. Scability

The performance of this implementation on several processors has been tested and is shown Figure .6 .

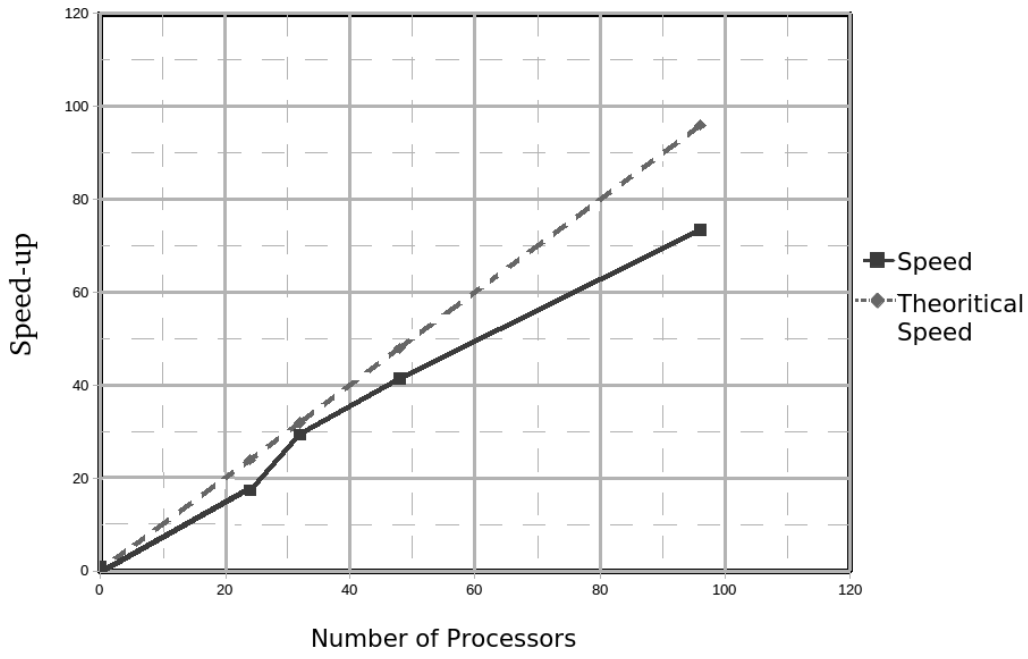


FIGURE .6. : *Performance of the new solver on several processors for a 2D calcul of a flow at Re = 30 past a fix cylinder with 10^6 eulerian points and 312 lagrangian points*

B. Discretized operators in the modified PISO Loop

At each PISO loop iteration m, let's define $[S]$ such that :

$$[S] = \frac{\partial u^{\star,m}}{\partial t} + \nabla \cdot (u^{\star,m} u^{\star,m}) - \frac{1}{Re} \nabla^2 u^{\star,m} \tag{.30}$$

with A its diagonal part and H its extra-diagonal part, i.e. :

$$[S] = [A] - [H] \tag{.31}$$

with :

$$[A] = \begin{pmatrix} a_{i-1;j-1} u^{\star,m}_{i-1;j-1} & 0 & 0 \\ 0 & a_{i;j} u^{\star,m}_{i;j} & 0 \\ 0 & 0 & a_{i+1;j+1} u^{\star,m}_{i+1;j+1} \end{pmatrix} = \{a_{ii}\} \times [U^{\star,m}_{ii}] \tag{.32}$$

and

$$[H] = \begin{pmatrix} 0 & a_{i;j-1} u^{\star,m}_{i;j-1} & a_{i+1;j-1} u^{\star,m}_{i+1;j-1} \\ a_{i-1;j} u^{\star,m}_{i-1;j} & 0 & a_{i+1;j} u^{\star,m}_{i+1;j} \\ a_{i-1;j+1} u^{\star,m}_{i-1;j+1} & a_{i;j+1} u^{\star,m}_{i;j+1} & 0 \end{pmatrix} \tag{.33}$$

Taking into account the (IBM) force term spread on the Eulerian mesh $f(\widehat{u})$ and the unknown pressure $p^{\star,m}$, we get :

$$[A] - [H] = -\nabla p^{\star,m} + f(\widehat{u}) \tag{.34}$$

$$[A] = \{a_{ii}\} \times [U^{\star,m}_{ii}] = -\nabla p^{\star,m} + f(\widehat{u}) + [H] \tag{.35}$$

$$[U^{\star,m}_{ii}] = \{a^{-1}_{ii}\}(-\nabla p^{\star,m} + f(\widehat{u}) + [H]) \tag{.36}$$

Taking the divergence of equation (.36) and using the discrete continuity condition $\nabla \cdot [u^{\star,m}_{ii}] = 0$ we get the Poisson equation for the pressure :

$$\nabla \cdot (\{a^{-1}_{ii}\} \nabla p^{\star,m}) = \nabla \cdot (\{a^{-1}_{ii}\} f(\widehat{u})) + \nabla \cdot (\{a^{-1}_{ii}\}[H]) \tag{.37}$$

The term $\nabla \cdot \{a^{-1}_{ii}\} f(\widehat{u})$ corresponds to the divergence of the force term. The velocity is then corrected, using the new pressure $p^{\star,m}$ such that :

$$[u^{\star,m+1}_{ii}] = g(u^{\star,m}, p^{\star,m}, f(\widehat{u}))$$

with

$$g \equiv \{a^{-1}_{ii}\}(-\nabla p^{\star,m} + f(\widehat{u}) + [H])$$

In order to update the flux, the force should be interpolated on the surface, introducing the issue discussed in section 2.2. Thus, the equation is written as :

$$F = S \cdot [\{a_{ii}^{-1}\}(-\nabla p + f(\widehat{u}) + [H])]_{faces} \tag{.38}$$

with $f(\widehat{u})_{faces}$ the immersed boundary force is calculated analytically on the surface.

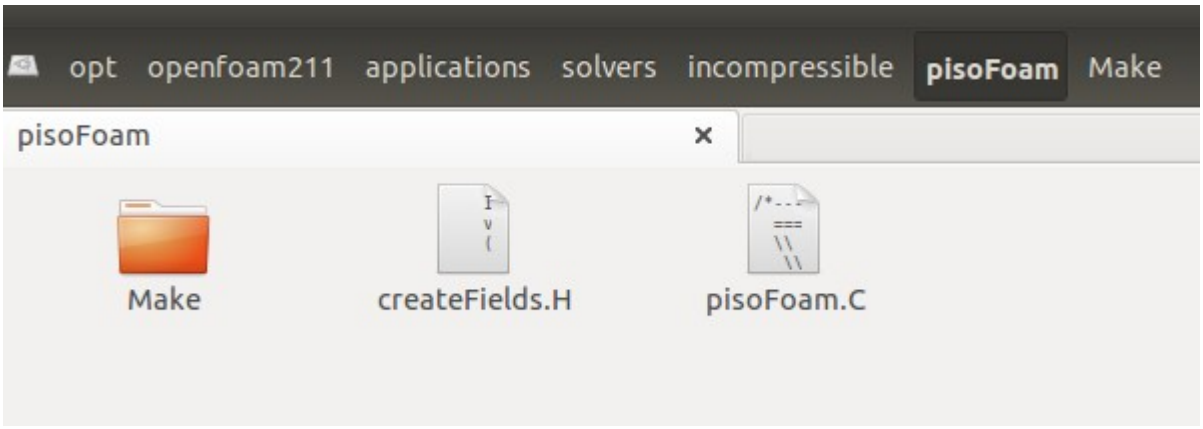
C. Library User guide

USER GUIDE

How to import IBM library in a solver

Example PisoFoam :

Copy the desired solver :



Solver modification :

Add 2 includes

```
34 #include "fvCFD.H"
35 #include "singlePhaseTransportModel.H"
36 #include "turbulenceModel.H"
37 #include "Interpolation_IBM.C"
38 // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
39
40 int main(int argc, char *argv[])
41 {
42     #include "setRootCase.H"
43
44     #include "createTime.H"
45     #include "createMesh.H"
46     #include "createFields.H"
47     #include "initContinuityErrs.H"
48     #include "Set_Immersed_Boundary.H"
49     // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
50
```

Before Main to declare the IBM structure

After main to declare the variable and read in the controlDict file

Add 2 command line

```
50
51  interpolation_ibm.Create_IBM_case( &mesh, runTime,Info);
52
53      Info<< "\nStarting time loop\n" << endl;
54
55      while (runTime.loop())
56      {
57          Info<< "Time = " << runTime.timeName() << nl << endl;
58
59          #include "readPISOControls.H"
60          #include "CourantNo.H"
61
62          // Pressure-velocity PISO corrector
63          {
64              // Momentum predictor
65
66              fvVectorMatrix UEqn
67              (
68                  fvm::ddt(U)
69                + fvm::div(phi, U)
70                + turbulence->divDevReff(U)
71              );
72
73              //UEqn.relax();
74
75
76                  solve(UEqn == -fvc::grad(p));
77
78
79
80                interpolation_ibm.Force_Velocity_Interpolation( &mesh, runTime,Info, U, p, forceIB,div_kernel_force,d);
81
```

Add command line before the loop of the solver to create the structure IBM

Add command line during the loop to calculate the forceIB (interpolation of the force), div_kernel_force (analytical gradient of the force) and d, the divergence of the field

Can be any scalar field to calculate on the wall

Use the VolVectorField forceIB and div_kernel_force in your solver equation to take in account the Immersed boundaries.

Make file modification (to load the IBM library) :

modify the option file to load the libraries needed (2 lines needed)

```
1 EXE_INC = \
2     -I$(LIB_SRC)/turbulenceModels/incompressible/turbulenceModel \
3     -I$(LIB_SRC)/transportModels \
4     -I$(LIB_SRC)/transportModels/incompressible/singlePhaseTransportModel \
5     -I$(LIB_SRC)/finiteVolume/lnInclude \
6     -I$(LIB_SRC)/meshTools/lnInclude \
7     -I$(LIB_SRC)/IBM_lib/lnInclude
8
9 EXE_LIBS = \
10    -lincompressibleTurbulenceModel \
11    -lincompressibleRASModels \
12    -lincompressibleLESModels \
13    -lincompressibleTransportModels \
14    -lfiniteVolume \
15    -lmeshTools
```

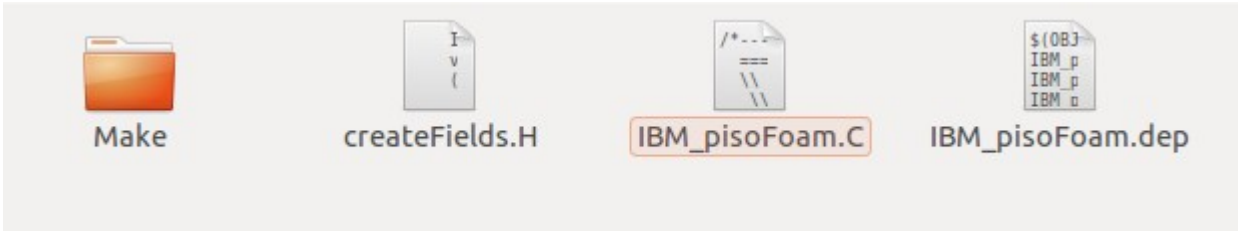
OpenFoam tool for research

IBM library to add in the folder : LIB_SRC the library IBM_lib

Modify the file to compile your new solver (here pisoFoam → IBM_pisoFoam)

```
1 IBM_pisoFoam.C
2
3 EXE = $(FOAM_USER_APPBIN)/IBM_pisoFoam
```

The new Solver folder is now ready to compile with the command wmake :



Input/Output

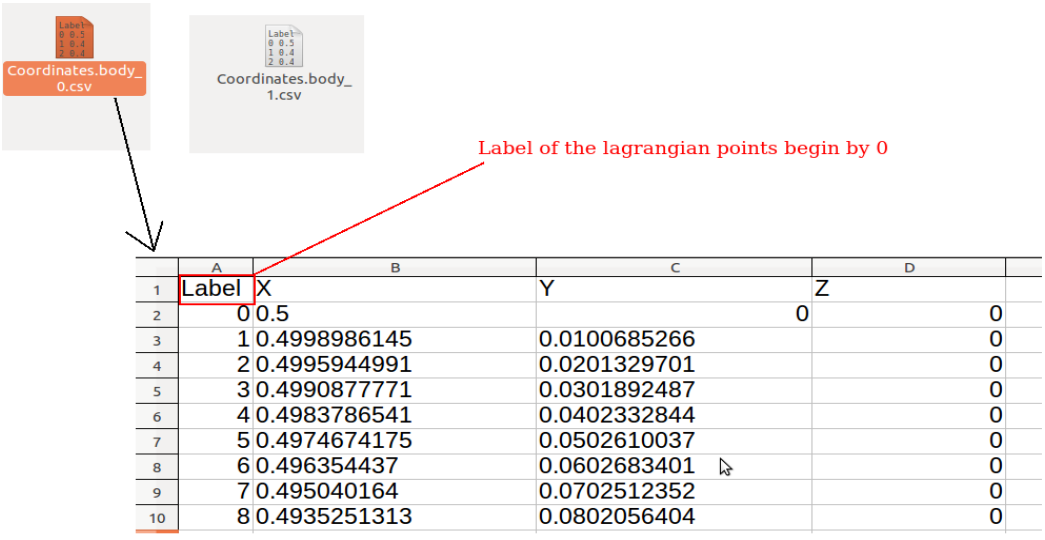
1) controlDict commands

Several options can be added to the controlDict file to control the IBM simulation :

```
49 //****************************** Immersed Boundary Options *******************************************
50
51 Two_D_simulation true; //to perform a 2D simulation : Default false
52
53 moving_object true;//if one object is moving : Default false
54
55 startTime_movement_body_no_0 0.25;//starTime of the movement for the body no 0 : Default startTime of the simulation (only if moving_object true)
56
57 movement_type_body_no_0 "User_define";// take the movement define in the function User_define_equation of the object Movement_equation in the IBM_lib for the body no 0 : Default
   "Forced_oscillation" (only if moving_object true)
58                                       //"Forced_oscillation" / "Free_oscillation" will be implemented soon /  "User_define" to implement by the user
59
60 Fq_X_oscillation_body_no_0 0.0;// oscillation frequency on x component for the body no 0 : Default 0.0 (only if moving_object true & movement_type_body_no_0  "Forced_oscillation" )
61 Fq_Y_oscillation_body_no_0 0.1825;
62 Fq_Z_oscillation_body_no_0 0.0;
63
64
65 Amplitude_X_oscillation_body_no_0 0.0;// oscillation amplitude/2 for the body no 0 : Default 0.0 (only if moving_object true & movement_type_body_no_0  "Forced_oscillation" )
66 Amplitude_Y_oscillation_body_no_0 0.25;
67 Amplitude_Z_oscillation_body_no_0 0.0;
68
69 Tolerance_bic_support 1e-9;// tolerance of the biconjugate gradient to calculate the epsilons : Default 1e-2
70
71 Diagnosctic_IBM false; // Launch the diagnostic of the IBM support : Default false
72 No_equidistante_points true; // Soart the lagrangian points to prevent 2 lagrangian points too close
73
74 //****************************************************************************************************
```

2) Inputs

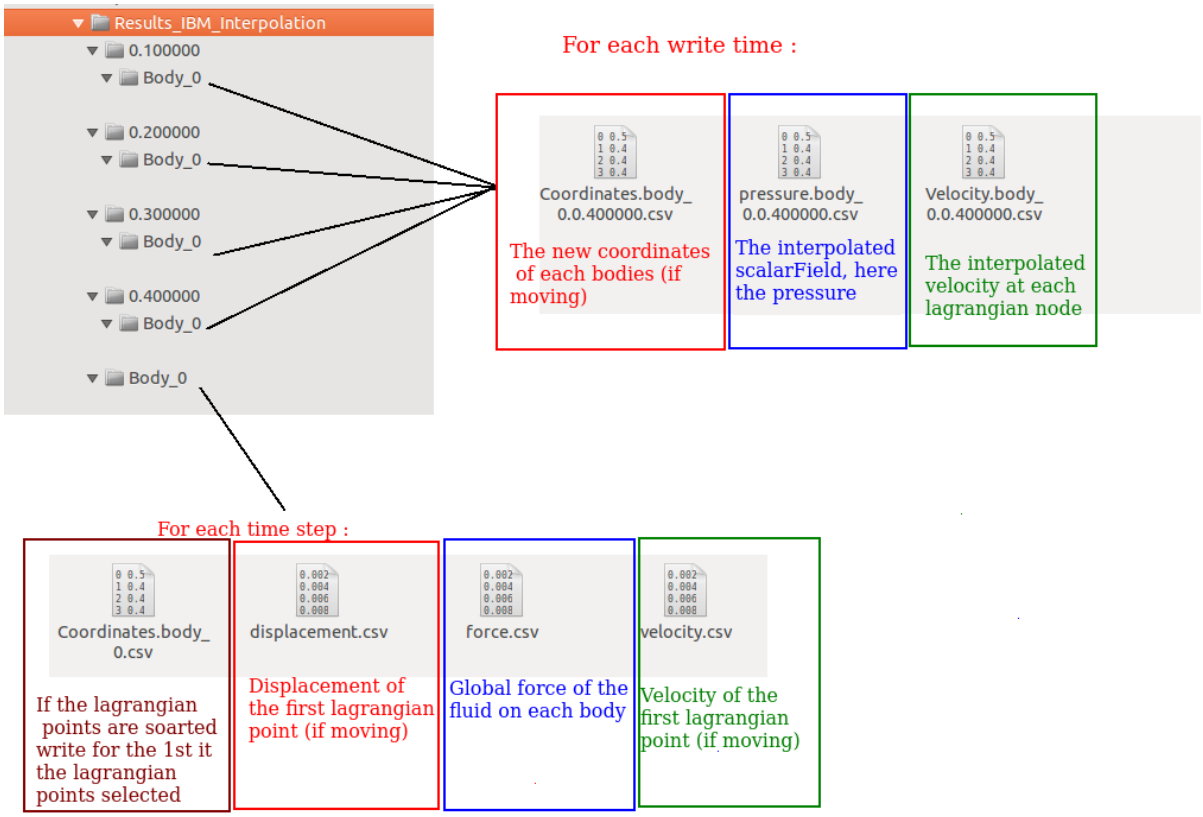
For each body a file Coordinates.body_X.csv with the coordinates of the lagrangian points :



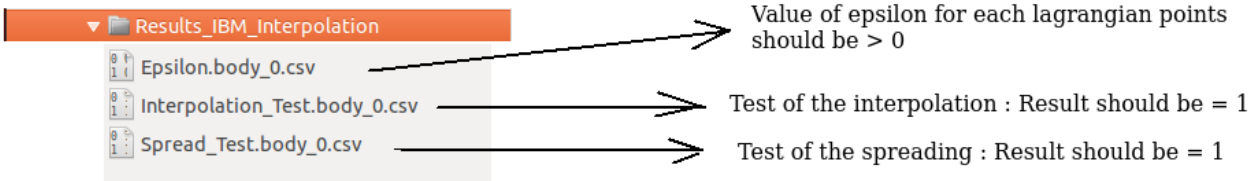
	A	B	C	D
1	Label	X	Y	Z
2	0	0.5	0	0
3	1	0.4998986145	0.0100685266	0
4	2	0.4995944991	0.0201329701	0
5	3	0.4990877771	0.0301892487	0
6	4	0.4983786541	0.0402332844	0
7	5	0.4974674175	0.0502610037	0
8	6	0.496354437	0.0602683401	0
9	7	0.495040164	0.0702512352	0
10	8	0.4935251313	0.0802056404	0

Ouputs :

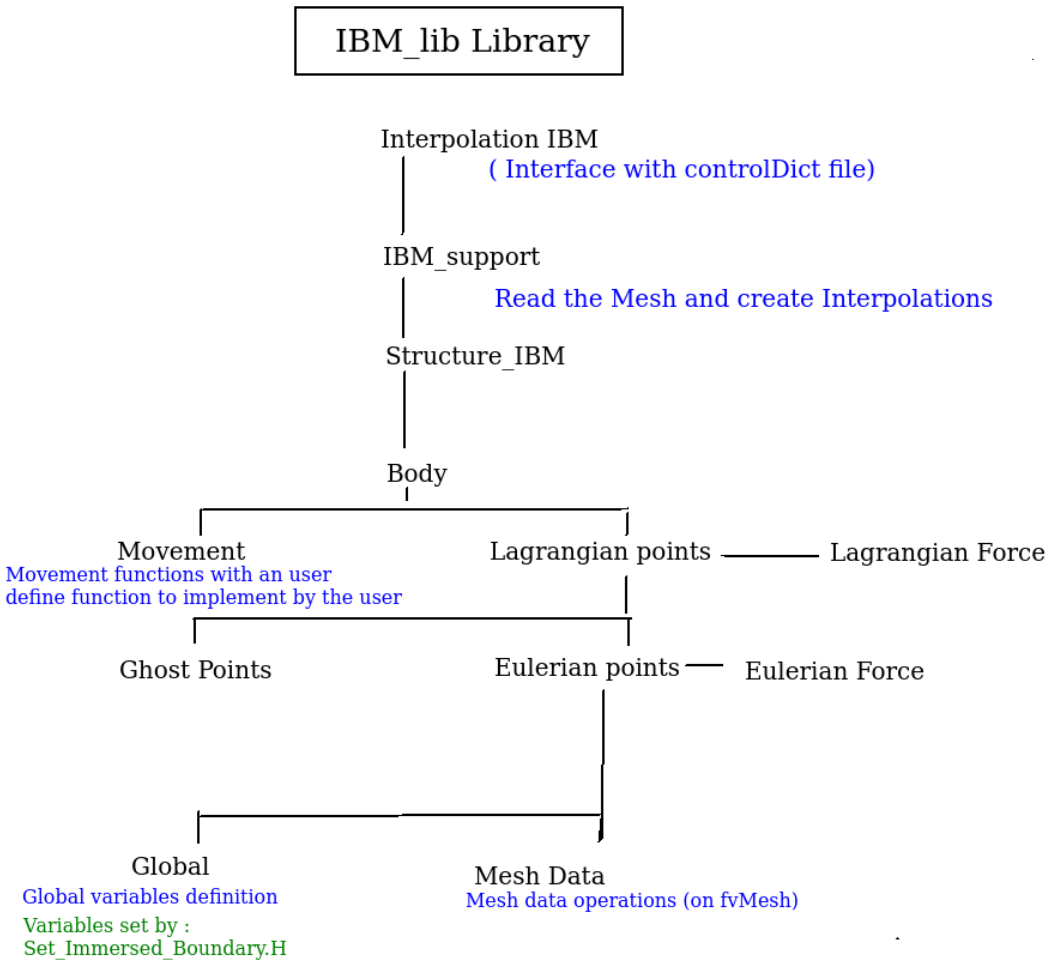
The Simulation create a file « Results_IBM_Interpolation » with :



If a diagnostic is launched (option : Diagnostic_IBM true ; in the controlDict) then the code return :



Scheme of the IBM Library:



Code Structure

(Object description)

Interpolation IBM

var : IBM_Support

function :

- **Create_IBM_Case()**
  - Input : fvMesh* (mesh ptr), runTime object, Info object
  - Output : /
  - Description :
    - Read the mesh in the Coordinate file
    - Soart the lagrangian points if specify in the controlDict
    - Save the mesh in a structure ibm_supp
    - Define the type of movement of each body
  - function used :
    - ControlMesh (in IBM_Support obj)
    - ReadMesh (in IBM_Support obj)
    - ReadandSave (in IBM_Support obj)
    - Set_structure (in IBM_Support obj)

- **Force_Velocity_Interpolation()**
  - Input : fvMesh* (mesh ptr), runTime object, Info object,Velocity, Pressure, forceIB, divergence of the forceIB, divergence of the field
  - Output : /
  - Description :
    - Intepolate the velocity field and calculater the associated force and the divergence of the force
    - Move the structure if one body is moving
  - function used :
    - Force_Interpolation (in IBM_Support obj)

- Move_structure (in IBM_Support obj)

IBM_Support

var : Structure_IBM

function :

- **ReadMesh()**
    - Input : /
    - Output : List of List of vector

    - Description :
        - Read the mesh in the Coordinate file
        - Return the mesh in a List of List of vector
        - Set the proc domain size (in global variables)

    - function used :
        - Set_domain (in Mesh_data_Eul obj)

- **ReadandSaveMesh()**
    - Input : fvMesh*
    - Output : /

    - Description :
        - Read the mesh in the Coordinate file
        - Save the mesh in a structure obj
        - Set the proc domain size (in global variables)

    - function used :
        Set_domain (in Mesh_data_Eul obj)

- **ControlMesh()**
    - Input : List of List of vector , fvMesh*
    - Output : /

    - Description :
        - Soart the points in the List of List of vector
        - Save the mesh in a structure obj
        - Set the proc domain size (in global variables)

- function used :
  Set_domain (in Mesh_data_Eul obj)

- **Set_Mesh()**
  - Input : fvMesh*
  - Output : /

  - Description :
    - Create the supporting box of all lagrangian points

  - function used :
    Create_support (Structure_IBM obj)

- **Force_Interpolation()**
  - Input : runTime object, Info object,Velocity, Pressure, forceIB, divergence of the forceIB, divergence of the field
  - Output : /

  - Description :
    - Intepolate the velocity field and calculater the associated force and the divergence of the force
    - Launch the diagnostic if asked in controlDict
    - Write the output data in the folder « Results_IBM_Interpolation »

  - function used :
    - Test_Force_Interpolation (in Structure_IBM obj)
    - Force_Interpolation (in Structure_IBM obj)
    - Scalar_Interpolation (in Structure_IBM obj)
    - Vector_Interpolation (in Structure_IBM obj)

- **Move_Lag_points()**
  - Input : mesh*, t, dt, Info
  - Output : /

  - Description :
    - move the lagrangian points
    - create new supports

- Write the output data for the moving cases in « Results_IBM_Interpolation »

- function used :
  - Move_Lag_points (in Structure_IBM obj)
  - Create_support (in Structure_IBM obj)

Structures_IBM

var :

- num_struct
- List of Body
- Mesh_data_eul

function :

- **Create_support()**
  - Input : mesh* , bool is_not_first_creation (defaut false)
  - Output : /
  - Description :
    - Add eulerian points to the lagrangian points (if a construction of the supporting box is needed)
  - function used :
    - Add_point_supp (in Body obj)
- **Force_Interpolation()**
  - Input : Velocity, Pressure, forceIB, divergence of the forceIB, divergence of the field,dt
  - Output : Global Force
  - Description :
    - Intepolate and spread the force and div(force)
  - function used :
    - InterpolSpread (in Body obj)
- **Test_Force_Interpolation()**
  - Input : Velocity, Pressure, forceIB, divergence of the forceIB, divergence of the field,dt
  - Output : /
  - Description :

        - Intepolate and spread the field of value 1
        - Write the value of the Interpolation and the Spreading and the epsilon in the file « Results_IBM_Interpolation »
    - function used :
        - Test_InterpolSpread (in Body obj)

- **Scalar_Interpolation()**
    - Input : Scalar, runTime
    - Output : /
    - Description :
        - Intepolate the given scalar
    - function used :
        - Interpol_scalar (in Body obj)

- **Vector_Interpolation()**
    - Input : Vector, runTime
    - Output : /
    - Description :
        - Intepolate the given vector
    - function used :
        Interpol_vector (in Body obj)

- **Move_Lag_points()**
    - Input : t, dt
    - Output : /
    - Description :
        - move the lagrangian points regarding to the motion defined in the controlDict
    - function used :
        - Forced_oscillation (in Movement_equation)
        - Free_oscillation (in Movement_equation)
        - User_define_equation (in

Movement_equation)

Movement_equation

var : /

function :

- **Forced_oscillation()**
  - Input : vector position ptr, t,dt,amplitude,frequency,vector velocity ptr, startTime of movement
  - Output : /
  - Description :
    - modify the given position and velocity regarding to the time, startTime, amplitude and frequency entered in the controlDict
  - function used :
    - /
- **Free_oscillation()**
  - Input : Body ptr
  - Output : /
  - Description :
    - To implement (should modify the body to compute a new velocity target and a new position regarding to the force
  - function used :
    - /
- **User_define_equation()**
  - Input : Body ptr
  - Output : /
  - Description :
    - To implement by user
  - function used :
    - /

Body

var :

- label number of lagrangian nodes
- List of Point_Lag
- Fq_oscillation
- Amplitude_oscillation
- boolean if the body is moving
- startTime of the movement
- string type of movement

function :

- **Add_point_supp()**
    - Input : mesh ptr, bool is it first creation of supporting box
    - Output : /

    - Description :
        - Add the first point for all lagrangian point and his neighbour (research made by findnearestpoint, if no results is found then the fincell() function which is slower is performed)
        - find the node owner of each lagrangian point
        - if one neighbour is missing found this neighbour through another processor (proc exchange)
        - Construct the supporting box
        - fill the supporting box with eulerian points (proc exchange)
        - Track if one eulerian cell associated with one lagrangian node is also associated with another one
        - Initialisation of epsilon if it is the first calculation of epsilon
        - Calculation of Epsilon

    - function used :
        - Add_1st_point_supp (in Point_Lag obj)
        - Proc_exchange (in Body obj)
        - Create_supp_box (in Point_Lag obj)
        - Fill_sup_box (in Point_Lag obj)

- Proc_exchange_fill_supp_box(in Body obj)
- Calculate_Same_points (in Body obj)
- Init_Epsilon (in Body obj)
- Calculate_Epsilon (in Body obj)

- **Calculate_Same_points()**
  - Input : /
  - Output : /
  - Description :
    - Track if one eulerian cell associated with one lagrangian node is also associated with another one
    - Save the same point adress in the structure
  - function used :
    - /
- **Proc_exchange_fill_supp_box()**
  - Input : mesh ptr, matrix of processor send map, face on the border to send to another border, corresponding lagrangian point, corresponding boundary condition no, correspond boolean is cyclic or not, corresponding face coordinate.
  - Output : /
  - Description :
    - Create the send map for each processor and send all the faces on a border to the other border proc regarding to send map.
    - Save the received point if they enter in the support box
  - function used :
    - Already_count_cell(Point_Lagobj)

- **Proc_exchange()**
  - Input : mesh ptr, matrix of processor send map, face on the border to send to another border, corresponding lagrangian point,

corresponding boundary condition no, correspond boolean is cyclic or not, corresponding face coordinate.

- Output : /

- Description :
  - Create the send map for each processor and send all the faces on a border to the other border proc regarding to send map.
  - Save the received point
  - Send back the coordinates of the neighbour cell to create the support dimensions

- function used :
  - Already_count_cell(Point_Lag obj)

- **Calculate_Epsilon()**
  - Input : /
  - Output : /

  - Description :
    - Use the biconjugate gradient to calculate epsilon

  - function used :
    - bicgstab_ju(Body obj)
- **Init_Epsilon()**
  - Input : /
  - Output : /

  - Description :
    - initialization of epsilon to accelerate the first calculation of epsilon
- **InterpolSpread()**
  - Input :dt
  - Output : /

  - Description :
    - Interpolate the velocity on lagrangian points
    - Calculate the force on lagrangian points
    - Spread the force on eulerian points

- **Test_InterpolSpread()**
  - Input:/
  - Output : /

  - Description :
    - Interpolate a field of value 1 and analyse of the interpolation
    - Spread a field of value 1
    - Interpol the field spreaded and analyse of the spreading
- **Interpol_Scalar()**
  - Input:/
  - Output : /

  - Description :
    - Interpolate a scalar field
- **Interpol_Vector()**
  - Input:/
  - Output : /

  - Description :
    - Interpolate a vector field

Point Lag

var :

- List of Point Eul
- nb of eulerian points
- nb of eulerian point temporary
- epsilon
- boolean if have periodic points
- h the dimensions of the supporting box
- boolean if the node is activated
- label of the processor node owner
- Force_point_Lag object
- List of ghost point eulerian

function :

- **Add_1st_point_supp()**
    - Input : fvMesh* (mesh ptr)
    - Output : label
    - Description :
        - Determine if the lagrangian node is part of the processor domain
        - find the nearest point of the lagrangian point
        - Determine if the first lagrangian point is near to a border
        - If the first lagrangian point is close to a border register the border face in the ghost point structure
    - function used :
        - getcelldim (in Mesh_data_Eul obj)

- **Create_supp_box()**
    - Input : /
    - Output : /

    - Description :

        - Calculate the supporting box dimension
- **Already_count_cell()**
    - Input : cell number
    - Output : boolean

    - Description :
        - Determine if the cell no is already counted in the supporting box
- **In_sup_boxl()**
    - Input : position, shifted position if cyclic point
    - Output : boolean

    - Description :
        - Determine if the cell is inside the supporting box
- **Fill_sup_box()**
    - Input : mesh ptr
    - Output : /

    - Description :
        - Calculate the kernel function of each eulerian point
        - Fill the supporting box with eulerian points
    - function used :
        - Calculate_Kernel_function (in Point_Eul obj)

Force Point Lag

var :

- V_lag speed of the fluid interpolated
- F lag force of the fluid on the Body
- U_target Dirichlet Boundary condition on the Body (= speed of the body)

Point Eul

- var :
- Position
- no of cell
- Force Eul obj
- boolean if periodic point
- List of similar points in the structure
- Kernel function
- Derivative of Kernel function
- Area or Volume of the cell (2/3D)

function :

- **Calculate_Kernel_function()**
    - Input : Position, dimensions of the supporting box
    - Output : /

    - Description :
        - Calculate the Kernel function and the derivative kernel function for the current point

Ghost Point Eul

- var :
- face label
- face no in the current boundary patch
- Position of face center
- Proc no adressing
- bolean is cyclic if the patch is connected to another processor through a cyclic condition
- bolean is just cyclic if the patch is connected to a cyclic condition without processor adressing
- number of the equivalent patch connection in another processor

Force Point Eul

var :

- Force spreaded
- gradient of the force
- velocity on eulerian point

Mesh Data Eul

var : /

function :

- **Set_domain()**
  - Input : mesh ptr
  - Output : Matrix of processor min and max dimensions
  - Description :
    - Calculate the Processor domain dimensions
- **getcelldim()**
  - Input : mesh ptr, label cell
  - Output : Matrix of processor min and max dimensions
  - Description :
    - Calculate the cellule dimensions
- **Find_min_dim()**
  - Input : mesh ptr,
  - Output : vector of minimum dimension
  - Description :
    - Calculate the minimum cellule dimensions