UNIVERSITÁ DEGLI STUDI DI TRIESTE

XXI^o Cycle
School of Doctorate Environmental and Industrial Fluid Mechanics

A LARGE EDDY SIMULATION TOOL FOR ENVIRONMENTAL AND INDUSTRIAL PROCESSES

(settore scientifico-disciplinare ICAR01)

DOTTORANDO:	TUTORE/SUPERVISORE:
Federico Roman	CHIAR.MO PROF. Vincenzo Armenio

ANNO ACCADEMICO 2007/2008

Contents

0.1 Introduction

Computational Fluid Dynamics (CFD) is an established tool for consulting and for basic research in fluid mechanics. CFD is required to provide information where analytical approaches or experiments would be impossible or too expensive. Most of the flows of engineering interest are turbulent.

Turbulence is an unresolved problem of classical physics. Because of the non linearity of the fluid motion equations, except for simple cases where simplifications can be made, there are no analytical solutions. The Navier-Stokes equations are one of the *seven millennium problems*, a list of problem for the mathematic community proposed in the 2000 by the Clay institute, analogous to that of Hilbert at the beginning of 1900. It is acknowledged that a more deep comprehension of Navier-Stokes equations or some analytical tool would have a great impact on the study of fluid motion. At the moment the only way to face them is numerically. In particular in the last years the increasing computer capabilities gave great possibility to study fluid motion with this approach.

Turbulence is composed of eddies in a broad range of size. These vortices are formed and are broken down continuously in a cascade from the largest to the smallest ones. Larger eddies break into smaller eddies and so on till their dimension is so small that they simply dissipate into heat because of viscosity.

To solve numerically the Navier-Stokes equations a very fine grid is necessary in order to catch also the smallest eddies. The consequent computational cost is enormous. From analytical reasoning the ratio between the largest to the smallest eddies can be computed, in one dimension it increases as $Re^{3/4}$ ($Re = ul/\nu$ is the Reynolds number with u and l a characteristic inertial velocity and length scale respectively and ν the kinematic viscosity). This means that in three dimensions and also considering time the computational cost increases as Re^3. Real flows are characterized by very large values of the inertial scales, *i.e.* very large Reynolds numbers. So the direct solution of Navier-Stokes equations (DNS) is not feasible for most of the practical flows. On the other hand DNS is a formidable tool to understand physics since it provides information (like two point spatial correlation, space-time evolution of coherent structures etc.) not obtainable in physical experiments.

With DNS all the scales of motion are directly solved, but in this way these simulations are confined just to low Reynolds number. In many applications it is not necessary to solve all the eddies, it can be sufficient to supply the effects of the turbulent motion to the mean flow. The feedback of the unresolved scales of motion is given through model. This operation brings to equations for averaged flow that are smoother, consequently the grid requirement becomes smaller making feasible also large Reynolds number flow simulations.

These models start from physical assumptions and are often calibrated through experiments. Reynolds Averaged Navier-Stokes equations (RANS) is one of these approaches, the flow is resolved for the mean flow and all the scales of motion are parametrized. It is an extremely cost saving procedure with respect to DNS. From the averaging procedure an additional stress appears due to the non linearity of the advective term of the momentum equations. Because of this additional stress the number of unknowns is larger than the number of equations and a *closure* is needed to obtain a solution. The fore mentioned model for the unresolved scales of motion has exactly the role to *close* the equations. However it is difficult to formulate a general model for this procedure. The smallest scales, far from the boundary, can be approximatively considered as isotropic and so they can be easily modeled in a general way. On the other hand the large scales are strongly influenced by the boundary conditions and it is impossible to derive a general model for them. However RANS can be very useful to obtain rough results on macroscopic quantities in reasonable time, also if it cannot give accurate details of the flow behavior.

As computers become powerful the range of turbulent scales that can be directly solved increases. In the past meteorologists started to solve only the large-scales of turbulent motion because they are fundamental for weather forecasting, while the smallest ones were modeled. These scales are important only in the way they affect the largest ones. This idea can be now applied also to complex environmental and industrial flows. This numerical approach is the Large Eddy Simulation (LES) and it can be located in between DNS and RANS.

In LES most of the scales of motion are directly solved, in particular all the large energy carrying scales, the ones that extract energy from the mean flow. These scales are influenced by the boundaries and they are strongly anisotropic. The smaller and dissipative scales must be modeled, but these scales loosing *memory* of the boundary conditions are more isotropic and hence formulating a general model that accounts for their effect is relatively easier. The scale separation is formally carried out by the application of a low-pass filter to the Navier-Stokes equations. The filter width is proportional to the grid resolution. The turbulent structures with dimension lower than this length scale represent the subgrid level whose effect must be supplied to the flow field. This means that asymptotically LES tends to DNS. There is a substantial conceptual difference between RANS and LES. At a certain level of grid refinement the results obtained through RANS become insensitive to other refinements, instead LES results become closer and closer to DNS results. This is because the dimension of the resolved size depends on the filter width, and all the scales of motion can be directly solved with a sufficient small filter. In this way LES provides good accuracy and its computational cost is reasonable in comparison to DNS.

Nowadays the use of parallel-architecture supercomputers makes Large Eddy Simulation a tool for investigation in real life problems, in particular when high detailed analysis is required. This is the case for many industrial and environmental processes. In general these flows are characterized by very complex physical phenomena that often cannot be caught in a proper way with simplified models, like the ones used in RANS.

Sea coastal flows are a clear example of this. These regions are in general shallow and characterized by complex geometry. The shallowness makes the bottom surface layer to encroache upon the Ocean Mixed Layer (OML, a review is in Phillips (1977), [53]), hence producing a single turbulent layer extending along the whole water column. The shallowness produces also wave breaking and generation of along-shore currents. The presence of the coast-line, rapid varying bathymetry and anthropic structures introduces complexity in the flow field, making it essentially three-dimensional. In semi-closed basins (*i.e.* bays) the interaction between the wind-driven surface current and the coast-line develops a mean circulation in the vertical planes, essentially characterized by the inversion of the mean velocity field in the bottom layers of the water column with respect to the upper layer field. This creates additional vertical shear resulting in enhanced turbulent production. Finally, buoyancy effects interacting with bathymetric gradients may favor additional circulation associated, for instance, to the downwelling of cold/salt water along the inclined bottom surface.

Traditionally, two-dimensional shallow-water approximation has been used in coastal problems, through the use of numerical models where three-dimensional effects are parameterized by means of calibration of empirical coefficients. Such approach has been shown to be effective in reproducing hydrodynamics characteristics of large scale shallow basins, where circulation in the vertical planes and buoyancy effects are negligible compared to mean horizontal transport (see [70]). In all other cases three-dimensional models must be used. Apart the numerical methods (time integration, space discretization, type of grid) these models mainly differ for the turbulence closure employed. This is a particularly suited configuration to take advantage of LES closure.

As highlighted before the main advantage of LES compared with classical RANS approach consists in the fact that the largest scales of the motion, which in a shallow-water three-dimensional basin are strongly anisotropic, are directly resolved in a three-dimensional time-dependent simulation, whereas the subgrid scales are parametrized through a closure model. Due to the geometric and physical complexities arising in coastal processes, a LES simulation is expected to give more accurate results compared to a RANS-like numerical model which generally uses turbulence models derived from industrial applications (mostly from the aerodynamic field) and calibrated on archetypal cases typically far from being representative of coastal applications.

The same reasoning applies to many industrial processes. RANS model can give reasonable values for the macroscopic variables, for example the pressure gradient in a pipe with a valve inside. On the other hand LES can furnish details on the characteristics of the flow, it can highlight if the flow goes under cavitation somewhere close to the valve.

For example, acoustic problems due to hydrodynamic noise are governed over a range of large scales which are easily reproduced by LES solution.

However in these types of flows many difficulties arise also for LES. In general these flows are characterized by high inertial scales values. These characteristic scales bring to high Reynolds number simulations. Wall-bounded flow at high *Re* requires high computational cost, in fact near solid boundaries also the small scales are anisotropic and energetic and in the end LES is constrained to be DNS-like. Besides complex geometries are often involved. Structured or Unstructured body-fitted grid can be very hard to made, moreover unstructured grid can be expensive and not suited for LES.

Scope of this thesis is to develop tools to apply LES to such configurations in order to make numerical simulation more adaptable to real life problems.

The work is structured in the following way: chapter 1 gives an introduction to the equation set; chapter 2 deals with complex geometry, here the well established Immersed Boundary Methodology is extended from Cartesian to curvilinear coordinates. Chapter 3 considers the Immersed Boundary in the contest of high Reynolds number flows, so the chapter is mainly focused on wall modeling. In chapter 4 a modified Smagorisky model for anisotropic grid is proposed. In chapter 5 particle dispersion is considered in stratified environmental flow. Finally in chapter 6 the developed tools are applied to an industrial and to an environmental problem.

Chapter 1

The governing equations

1.1 The governing equations

The Navier-Stokes equations govern the fluid motion. For many applications in industrial and environmental processes they can be written under the Boussinesq approximations considering the fluid as incompressible. In this way in the continuity equation the density variations are considered negligible compared to the magnitude of the velocity gradient. Also in momentum equations these variations are ignored except in the vertical momentum equation, where they affect the motion through the buoyancy term. This assumption is correct only if the density anomaly inside the fluid is just a fraction of the state of reference ρ_0. In a Cartesian frame of reference the equations read as:

$$\begin{aligned}
\frac{\partial u_i}{\partial x_i} &= 0 && (1.1)\\
\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} &= -\frac{1}{\rho_0}\frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\Delta\rho}{\rho_0} g \delta_{i,2} + B_i && (1.2)\\
\frac{\partial T}{\partial t} + \frac{\partial u_j T}{\partial x_j} &= k \frac{\partial^2 T}{\partial x_j \partial x_j} && (1.3)\\
\frac{\partial S}{\partial t} + \frac{\partial u_j S}{\partial x_j} &= k_S \frac{\partial^2 S}{\partial x_j \partial x_j} && (1.4)
\end{aligned}$$

In equations (1.1) and (1.2) u_i is the velocity component in direction i, x_i is the spatial coordinate in direction i, t is time, P is pressure, ν is the kinematic viscosity characteristic of the fluid, g is the gravitational acceleration, $\Delta\rho$ is the density anomaly and ρ_0 the bulk density, while the term B represents the body forces (other than gravity) acting over the fluid element as the Coriolis force. The effect of density variations is retained in the vertical momentum equation (here direction 2) through the buoyancy term $\Delta\rho/\rho_0 g\delta_{i,2}$.

Equations (1.3) and (1.4) represent the advection-diffusion of temperature and salinity respectively, k and k_s are the coefficients of thermal and salinity diffusion.

On the right hand side of eq. (1.2) the term ρ can be expressed in terms of temperature and salinity through the state equation:

$$\rho = \rho_0[1 - \alpha(T - T_0) + \beta(S - S_0)] \tag{1.5}$$

where α and β are the expansion coefficients for temperature and salinity respectively.

1.2 The equations set in curvilinear form

Equations (1.1) - (1.4) are written in a Cartesian frame of reference. In case of domain with complex geometry it is better to transform the equations in curvilinear coordinates ξ_i. Figure 1.1 shows an example of coordinates transformation between the physical and the computational space. The advantage of this transformation is that the new frame is able to follow physical shape if not too sharp. The coast line in the study of a sea coastal flow is an example. In this way the boundary conditions can be easily imposed on the geometrical contour.

Considering the relation:

$$\frac{\partial u_i}{\partial x_j} = \frac{\partial u_i}{\partial \xi_k}\frac{\partial \xi_k}{\partial x_j}$$

the equations of section 1.1 become:

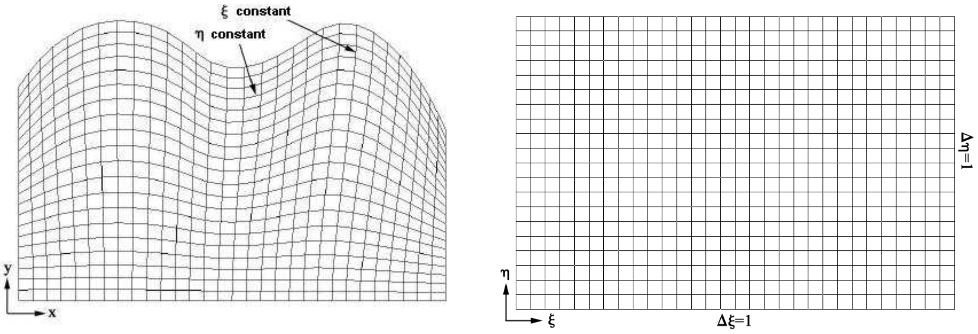

Figure 1.1: Coordinates transformation from the physical to the computational space.

$$\frac{\partial U_m}{\partial \xi_m} = 0 \tag{1.6}$$

$$\frac{\partial J^{-1}u_i}{\partial t} + \frac{\partial F_{im}}{\partial \xi_m} = J^{-1}B_i \tag{1.7}$$

where the gravitational term has been included in B and F_{im} is:

$$F_{im} = U_m u_i + J^{-1}\frac{\partial \xi_m}{\partial x_i}p - \nu G^{mn}\frac{\partial u_i}{\partial \xi_n} \tag{1.8}$$

ξ_m $(m = 1, 3)$ are the coordinates in the transformed (rectangular and regular) computational space, with J^{-1} the inverse of the Jacobian or the cell volume; U_m is the product between the contravariant velocity and the inverse of the Jacobian and represents the volumetric flux normal to the surface $\xi_m = cost$. The transformed pressure term represents the flux of the pressure gradient through the faces of the cell in the physical domain, while the third term on the right hand side of (1.8) is the transformed diffusive term, which represents the fluxes of the viscous stresses through the faces of the cell. The term G^{mn} is the "mesh skewness tensor".

The governing equations can be solved using a finite difference or a finite volume approach; in the latter case the metric terms appearing in eqs. (1.6) and (1.7) assume the meaning of distances, surfaces and volumes in the physical, curvilinear, space.

The terms in eqs. (1.6) - (1.8) are:

$$U_m = J^{-1}\frac{\partial \xi_m}{\partial x_j}u_j \tag{1.9}$$

$$J^{-1} = det(\frac{\partial x_i}{\partial \xi_j}) \tag{1.10}$$

$$G^{mn} = J^{-1}\frac{\partial \xi_m}{\partial x_j}\frac{\partial \xi_n}{\partial x_j} \tag{1.11}$$

Finally the equation for the advection and diffusion of a scalar can be written as:

$$\frac{\partial J^{-1}C}{\partial t} + \frac{\partial U_m C}{\partial \xi_m} = \frac{\partial}{\partial \xi_m}\left(kG^{mn}\frac{\partial C}{\partial \xi_n}\right) \tag{1.12}$$

where C can be the temperature or the salinity.

1.3 The equations set in discretized form

The equations are integrated in space and time using the fractional step as in [75]. If a non-staggered grid is used the pressure and the Cartesian velocity components are defined at the centroid of the cell while the contravariant fluxes are defined at the cell boundaries as in Fig. 1.2.

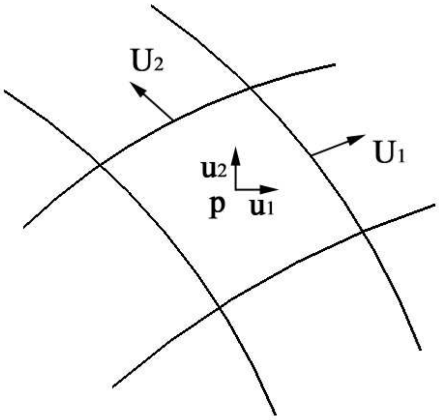

Figure 1.2: Distribution of velocities (Cartesian and contravariant) for a two dimensional case in a computational cell.

The equations can be integrated in time with an explicit or a semi-implicit scheme. In the latter case an Adam-Bashfort scheme can be used for the advective term, the forcing and the off diagonal diffusive term, while the diagonal diffusive term can be advanced in time with a Crank-Nicolson scheme. In this way the diffusive stability limit is overcome. Further discussion concerning the time advancement scheme will be presented in chapter 2 concerning the Immersed Boundary Method.

Once discretized, the momentum equation will read:

$$J^{-1}\frac{u_i^{n+1}-u_i^n}{\Delta t} = \frac{3}{2}\left(C(u_i^n)+D_E\left(u_i^n\right)+B_i^n\right)-\frac{1}{2}\left(C(u_i^{n-1})+D_E\left({u_i}^{n-1}\right)+{B_i}^{n-1}\right)$$

$$+R_i\left(p^{n+1}\right)+\frac{1}{2}\left(D_I\left(u_i^{n+1}\right)+D_I\left(u_i^n\right)\right) \qquad (1.13)$$

where C is the advective term and D_I and D_E are the diffusive terms treated in an implicit or explicit way respectively. Finally R_i is the gradient operator in curvilinear coordinates. Here a symbolic form is used to express the discretized terms, which is general for any mathematical formulation of the conservation laws.

Introducing an intermediate velocity u_i^* between u_i^{n+1} and u_i^n the momentum equation can be split into two steps, usually called *predictor* and *corrector*.

The *predictor* step is concerned with the time advancement of the advective and diffusive transport of momentum

$$\left(I-\frac{\Delta t}{2J^{-1}}D_I\right)\left(u_i^*-u_i^n\right)=$$

$$\frac{\Delta t}{J^{-1}}\left[\frac{3}{2}\left(C(u_i^n)+D_E\left(u_i^n\right)+B_i^n\right)-\frac{1}{2}\left(C(u_i^{n-1})+D_E\left({u_i}^{n-1}\right)+{B_i}^{n-1}\right)-D_I\left(u_i^n\right)\right] \tag{1.14}$$

while the *corrector* step adjusts the flow field through pressure.

$$u_i^{n+1}-u_i^{*}=\frac{\Delta t}{J^{-}1}\left[R_i\left(\psi^{n+1}\right)\right] \tag{1.15}$$

where ψ is a projector operator equal to the kinematic pressure in explicit scheme, see [1]. The procedure requires the solution of a Poisson equation for ψ, that enforces the continuity equation. The Poisson equation in the curvilinear coordinates form is:

$$\frac{\partial}{\partial \xi^m}\left(G^{mn}\frac{\partial \psi}{\partial \xi_n}\right)=\frac{1}{\Delta t}\frac{\partial U_m^*}{\partial \xi_m} \tag{1.16}$$

where the term $U_m^* = J^{-1}\partial\xi_m/\partial x_j \; u_j^*$ is the intermediate contravariant flux, built with the intermediate velocity field u_i^*. The Poisson equation for the pressure field is solved using a SOR iterative technique in conjunction with a V-cycle multigrid method.

The curvilinear coordinate fractional step method of [75] also requires the evaluation of the contravariant fluxes at the new time level:

$$U_m^{n+1}=U_m^{*}-\Delta t G^{mn}\frac{\partial \psi}{\partial \xi_n}. \tag{1.17}$$

After this procedure the time step $n+1$ is obtained and the next temporal step is advanced.

The equation for advection and diffusion of a scalar is treated as the predictor step.

1.4 The equations for LES

As introduced at the beginning of this thesis the LES is based on the direct solution of the energy carrying scales of motion, while the smaller scales are modeled. The scales separation is mathematically performed by the application of a low-pass filter to the variable:

$$\overline{u_i}=\int G(x,x')dx' \tag{1.18}$$

where $G(x,x')$ is the filter function. Every filter has an associated length scale Δ, for this reason the turbulent structures with a scale lower than Δ must be modeled.

After the filtering operation the equations set becomes:

$$\frac{\partial \overline{U_m}}{\partial \xi_m} = 0 \tag{1.19}$$

$$\frac{\partial \overline{J^{-1} u_i}}{\partial t} + \frac{\partial \overline{U_m u_i}}{\partial \xi_m} =$$

$$-\frac{\partial}{\partial \xi_m} \overline{\left(\frac{\partial J^{-1} \xi_m}{\partial x_i} p \right)} - \frac{\partial}{\partial \xi_m} \frac{\partial \sigma_m}{\partial x_i} + \frac{\partial}{\partial \xi_m} \overline{\left(\nu G^{mn} \frac{\partial u_i}{\xi_m} \right)} + \overline{J^{-1} B_i} \tag{1.20}$$

The effect of the unresolved scales of motion appears through the tensor σ_i^m:

$$\sigma_i^m = \overline{J^{-1} \frac{\partial \xi_m}{\partial x_i} u_j u_i} - \overline{J^{-1} \frac{\partial \xi_m}{\partial x_i} u_j} \overline{u_i} = \overline{U_m u_i} - \overline{U_m} \overline{u_i} \tag{1.21}$$

Further details on how to model this stress will be presented in chapter 3 and 4.

Finally the equation for the advection diffusion of a scalar can be written as:

$$\frac{\partial \overline{J^{-1} C}}{\partial t} + \frac{\partial \overline{U_m C}}{\partial \xi_m} = \frac{\partial}{\partial \xi_m} \left(\overline{k G^{mn} \frac{\partial C}{\partial \xi_n}} \right) - \frac{\partial \xi_m}{\partial x_j} \frac{\partial \lambda_i}{\partial \xi_m} \tag{1.22}$$

with λ_i the subgrid density fluxes.

Chapter 2

An improved Immersed Boundary Method for Curvilinear Grids

This chapter has been developed in a scientific cooperation with Prof. E. Napoli and Ph.D. B. Milici of the DIIAA of the University of Palermo. The work has been published in the journal: *Computer & Fluid*, [57].

Most engineering flows develop in complex geometrical domains. Numerical simulations of complex geometry flows require boundary-fitted unstructured grids or multiblock/multidomain curvilinear structured grids, which often have to move or to deform according to the velocity field. The use of unstructured grids, nevertheless, results in the increase of computational costs with respect to structured formulations, since a larger number of operations per grid point is required (see [6] for a discussion). On the other hand, curvilinear structured grids are suited for weak-to-moderate geometric complexity, whereas high level of geometric complexity (practical industrial and/or environmental flows) often requires computationally expensive multidomain approaches, and the use of highly distorted cells.

An effective and comparatively less expensive approach for handling complex geometry flows is the *immersed boundary method* (IBM), where the governing equations are discretized using a Cartesian grid system and the presence of a boundary surface within the computational domain is represented through a special treatment of the field equations. Specifically, the approach, which was first introduced by Peskin [52], mimics a boundary surface using a field of external forces applied in the interior of the computational domain.

After the seminal paper of Peskin [52], modifications and improvements were proposed. Among them, Goldstein et al. [28] proposed to obtain the external forces as a function of the difference between the calculated velocity at the boundary surface internal to the domain and the velocity of the boundary itself (*feedback forcing*). The main drawbacks of such approach are that it requires the tuning of two flow-dependent constants and that it has very restrictive stability limits for the time step. A different technique was successively proposed in [47], which allows to directly impose the desired velocity at the boundary (*direct forcing*), without introducing any additional parameter. The *direct forcing* approach also allows to partially alleviate the problem related to the stability limits, as extensively shown by Verzicco and co-authors in a series of papers ([22],[72],[17]). In both cases the use of interpolation procedures is required to enforce the desired solution at the immersed boundary, since the position of the unknowns on the grid in general does not coincide with that of the immersed boundary. Fadlun et al. [22] showed that the technique of Mohd-Yusof [47] allows obtaining very good results in the simulation of complex geometry flows, but some ambiguities remained associated to the suitable choice of the direction of interpolation. Interpolation procedures were thus successively improved by Kim et al. [34], Gilmanov et al. [26] and Balaras [6]. In particular, the latter proposed to interpolate the velocity along lines normal to the body surface. This modification improved the quality of the results in the near body region. Owing to the ability of the IBM in managing flows in complex geometries, the methodology has been recently extended to the relevant case of compressible flows (see for example [18] and [19]).

So far the IBM has been employed in conjunction with Cartesian-grid Navier-Stokes solvers. This allows modeling complex boundaries using Cartesian grids and thus retaining the accuracy and efficiency of regular-grid computations. However, there are a number of applications where the use of a Cartesian grid solver may result in high computational inefficiency, and the implementation of the IBM in curvilinear grid solvers may be beneficial. Figure 2.1 shows a typical example where a curvilinear structured grid used with immersed boundaries may produce high quality grids at a reduced computational cost when compared to the Cartesian counterpart. In other cases special constraints to the grid point distribution close to a solid surface have to be imposed. An example is given by the direct numerical simulation or *resolved* Large Eddy Simulation of turbulent fields, where at least 8 grid points should be placed within 10 wall units of a solid wall (see [54] and the literature therein cited). In these cases the use of Cartesian grids can result in a very high number of grid points near the boundaries, associated to a poor grid quality (see for example Fig. 2.2a), whereas the opposite is true when curvilinear grids are employed (Fig. 2.2b).

The importance of the extension of the immersed boundary technique to curvilinear structured Navier-Stokes solvers has been already highlighted in [46].

Kim et al.,[34] applied IBM to cylindrical coordinates. Cylindrical grid formulations, nevertheless, although not Cartesian, belong to the family of orthogonal coordinates schemes. Their use, moreover, remains confined to the study of a special class of problems, where a bulk cylindrical geometry

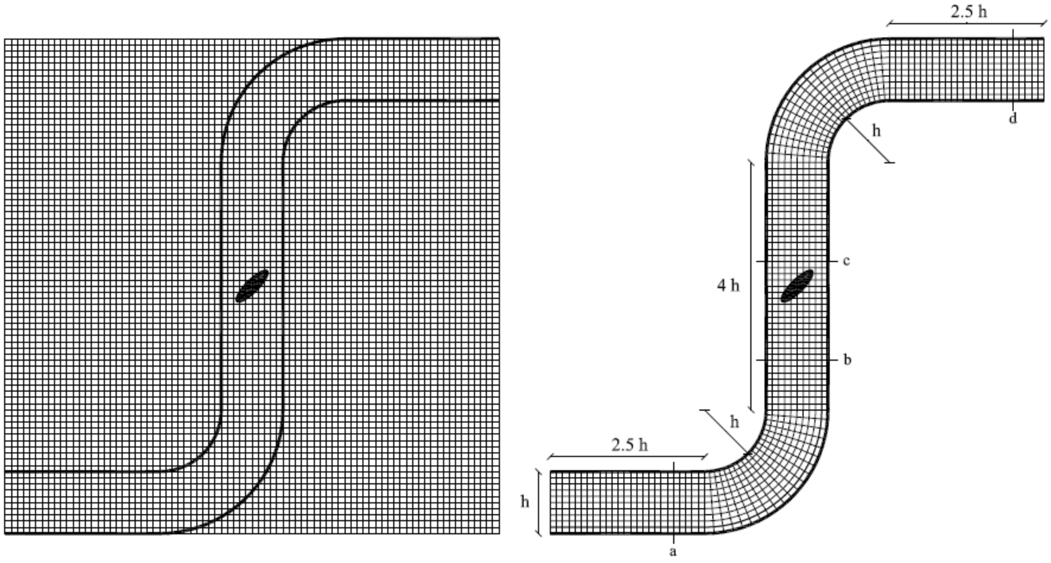

Figure 2.1: Examples of computational grids for the simulation of the flow around a valve placed within a S-shaped duct. a) Immersed boundary used with a Cartesian grid; b) Immersed boundary used in conjunction with a curvilinear grid. In the present example the advantage of using a curvilinear structured solver comes from the overall geometric characteristics of the duct.

is combined with obstacles to give an overall complex shape. The extension of the immersed boundary technique to non-orthogonal curvilinear grids can be thus considered as an alternative and more general approach which allows discretization of flow domains of general shape.
Another issue worth discussing is that, so far, the implementation of IBM in Navier-Stokes solvers has been mainly pursued using explicit schemes for the time-advancement of the solution, with the notable exception of Fadlun et al. [22]. Specifically the authors implemented an implicit direct forcing method in conjunction with a Cartesian solver that used a semi-implicit time-integration technique, with the aim to alleviate the limitation on the time step. The implicit calculation of the forcing terms, which is straightforward when the interpolation proposed in [22] is used, requires on the contrary the inversion of large sparse matrices or iterative methods when more accurate interpolation procedures, such as those proposed by Gilmanov et al. [26] and Balaras [6], are used. The explicit calculation of the forcing terms in conjunction with a semi-implicit solver was proposed in [34]. Nevertheless, the procedure requires to solve the momentum equation twice (the first one to obtain the forcing field, the second one to advance the solution in time).

The aim of the research presented in this chapter is to:

1. to extend the well established direct forcing, immersed boundary method [26, 6] to the curvilinear form of the Navier-Stokes equations;

2. to extend the IBM of [26] and [6] to any-shaped immersed surfaces, also including the case of sharp corners;

3. to propose and evaluate the accuracy of a simplified direct-forcing semi-implicit technique, which does not require the double resolution of the momentum equation at each time step.

This way it is possible to have a unique solver which can be efficiently used without limitations in geometrical complexity. A Cartesian-like geometry becomes just a particular case which can be easily afforded using the proposed technique.

Numerical tests are carried out to test the method, which is applied in steady and unsteady flow conditions considering 2-D as well as 3-D flows.

2.1 The direct forcing *Immersed Boundary* technique

The *Immersed boundary* methodology is based on the identification in the computational grid of a solid region delimited by an interface. When the direct forcing approach is employed, a forcing term and a mass source/sink

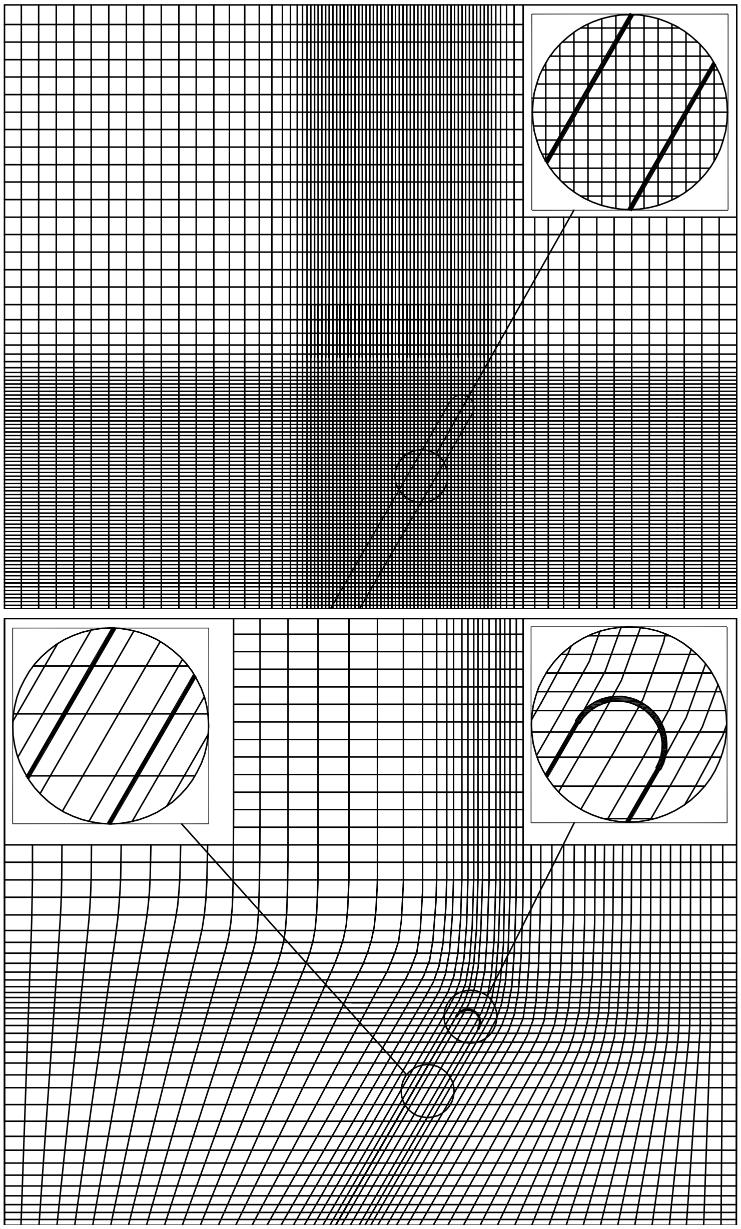

Figure 2.2: Computational grids for the simulation of the flow around a groin built on the side of an open-channel. a) Immersed boundary used with a Cartesian grid; b) Immersed boundary used in conjunction with a curvilinear grid. In the figures the distance between the internal boundary and the nearest grid point is maintained below $d/20$, where d is the width of the groin. In this example the choice of a curvilinear solver allows obtaining a high quality grid near the obstacle with a reduced number of grid points.

term are added on the right hand sides of the Navier-Stokes and continuity equations, respectively, aimed at miming the correct boundary condition at the interface. Working in the curvilinear-coordinate framework, the governing equations of section 1.2 read as:

$$\frac{\partial U_m}{\partial \xi_m} = J^{-1} g \tag{2.1}$$

$$J^{-1}\frac{\partial u_i}{\partial t} + \frac{\partial U_m u_i}{\partial \xi_m} = -J^{-1}\frac{\partial p}{\partial \xi_m}\frac{\partial \xi_m}{\partial x_i} + \frac{\partial}{\partial \xi^m}(\nu G^{mn}\frac{\partial u_i}{\partial \xi_n}) + J^{-1}B_i + J^{-1}f_i \tag{2.2}$$

where f_i, g are the forcing and mass source/sink terms discussed above. The additional terms g and f_i are built in such a way to obtain the desired boundary conditions at the interface. Apart the relevant paper [34], in the literature the mass source/sink term in the continuity equation g has not been considered. In the general cases, thus, the presence of an immersed boundary has been accounted for by means of the forcing term f_i in the momentum equation only. This technique has proven to give satisfactory results (see among the other the recent review of [45]). For this reason here the source/sink term g is neglected and the immersed boundary is thus considered through the forcing term f_i in the momentum equation. The term f_i is aimed at forcing velocities in the points surrounding the interface Ψ (in the nodes that will be referred as *IB points*) to assume values $u_{IB,i}$ compatible with the desired boundary condition on Ψ (a formal definition of IB points will be given in Section 2.1.1). Since IB points generally do not lie on the surface Ψ, velocities $u_{IB,i}$ must be obtained through suitable interpolation procedures among the surrounding nodes. As a consequence, the forcing term will depend on the velocities in these nodes as well.

Although in principle the IBM may be used in conjunction with any algorithm which integrates the Navier-Stokes equations, as in [22] the implementation in the framework of fractional step methods is discussed, specifically the curvilinear-coordinate, fractional step method described in [75], a brief introduction is in section 1.3. This represents an extension of the semi-implicit fractional step method of [35]. Although the algorithm of [75] is semi-implicit, hereafter it is discussed the implementation of the IB methodology both for explicit and semi-implicit versions of the solver.

Consider now the discretized equations first introduced in section 1.3 but for an explicit time advancement:

$$\begin{aligned}\frac{u_i^* - u_i^n}{\Delta t} \quad + &\quad \alpha\left[C(u_i^n) - D(u_i^n) - B_i^n\right] + \\ &\quad \beta\left[C(u_i^{n-1}) - D(u_i^{n-1}) - B_i^{n-1}\right] - f_i = 0 \qquad (2.3)\\ \frac{u_i^{n+1} - u_i^*}{\Delta t} \quad = &\quad -GRAD_i(p) \qquad (2.4)\end{aligned}$$

the term $GRAD_i(p)$ denotes the discretised form of the pressure gradient. The parameters α and β depend on the adopted numerical scheme (for instance, $\alpha = 3/2$ and $\beta = -1/2$ when the Adams-Bashforth scheme is used as in 1.3). Eq. (2.3) must be solved to obtain the predictor-step velocity u_i^*.

The forcing field is required to give the desired boundary conditions on Ψ for the corrected velocity field u_i^{n+1}. Following a well-established procedure (see, for example, [34] and [6]), it is possible to enforce the boundary conditions on the intermediate time level u_i^* instead that on u_i^{n+1}. The forcing function can be thus obtained as [6]

$$\begin{aligned} f_i &= [\frac{u_{IB,i}^* - u_{IB,i}^n}{\Delta t} + \alpha(C(u_i^n) - D(u_i^n) - B_i^n) + \\ &\quad \beta(C(u_i^{n-1}) - D(u_i^{n-1}) - B_i^{n-1})] \quad \text{on} \quad \Psi \\ f_i &= 0 \quad \text{elsewhere} \end{aligned} \tag{2.5}$$

where $u_{IB,i}^*$ is the intermediate velocity in IB points compatible with the boundary conditions on Ψ. When IB points lie on Ψ the velocity $u_{IB,i}^*$ coincides with the immersed body velocity, while in the general cases interpolation procedures are required, as will described in section 2.1.2. It is simple to show that calculating f_i from eq. (2.5) and then resolving eq. (2.3) is equivalent to resolve eq. (2.3) with $f_i = 0$ and then directly impose $u_i^* = u_{IB,i}^*$ in IB points, without recurring to the explicit calculation of f_i.

Several Navier-Stokes solvers make use of semi-implicit approaches, where the convective terms are calculated explicitly and the diffusive terms implicitly, to remove the viscous stability limit for the time step. Specifically, the semi-implicit time-advancement of the predictor step gives rise to the following equation:

$$\begin{aligned} [1 \quad - \quad & \gamma \Delta t D](u_i^*) = u_i^n + \\ & \Delta t \left[\alpha[-C(u_i^n) + B_i^n] + \beta[-C(u_i^{n-1}) + B_i^{n-1}] + \delta D(u_i^n)\right] + \Delta t f_i \end{aligned} \tag{2.6}$$

where the weighting coefficients $\gamma = \delta = 0.5$ produce the 2nd-order accurate, implicit Crank-Nicolson scheme.

Nevertheless, the use of the forcing procedure described above in conjunction with semi-implicit approaches is not straightforward. Even using the approximation $u_{IB,i}^* = u_{IB,i}^{n+1}$, the forcing term in eq. (2.6) should be calculated as:

$$\begin{aligned} f_i &= \frac{u_{IB,i}^* - u_i^n}{\Delta t} - (\gamma D(u_{IB,i}^*) + \delta D(u_i^n)) + \\ &\quad \left[\alpha[C(u_i^n) - B_i^n] + \beta[C(u_i^{n-1}) - B_i^{n-1}]\right] \end{aligned} \tag{2.7}$$

where the calculation of the term $D(u_{IB,i}^*)$ would require the prior knowledge of the velocities u_i^* (that are still unknown) in the whole stencil of the operator

D centered on IB. The main difference with the explicit approach thus is that in the semi-implicit approach the velocities u_i^* in IB points depend on those in non-IB points through the second-order derivatives of the diffusive terms. Thus, the calculation of u_i^* with $f_i = 0$ in non-IB points and then the use of these values to obtain through suitable interpolation velocities $u_{IB,i}^*$ in IB points - as it is done in the explicit approach - would introduce additional errors. As a consequence, the calculation of the forcing field f_i would require the inversion of large sparse matrices, as also discussed in [6]. Note that this is not the case in the explicit approach.

Kim et al. [34] overcame the mentioned problem using a mixed explicit-implicit procedure. This procedure is based on: a) the resolution of the explicit eq. (2.3) with $f_i = 0$ to obtain a provisional velocity field $\tilde{u}_i^*$; b) the calculation of velocities $u_{IB,i}^*$ in every IB point through interpolation of the velocities $\tilde{u}_i^*$ in the surrounding computational nodes; c) the calculation of the forcing field using eq. (2.5); d) the solution of eq. (2.6) to obtain the predictor-step velocities u_i^*. The formulation of [34] does not introduce additional errors as discussed above, although it requires the subsequent resolution of eqs. (2.3) and (2.6).

Here a simplified and alternative procedure is proposed. First the implicit equation is solved:

$$\begin{aligned}[1 \quad - \quad & \gamma\Delta t D](\tilde{u}_i^*) = u_i^n - \\ & \Delta t \left[\alpha[C(u_i^n) + B_i^n] + \beta[C(u_i^{n-1}) + B_i^{n-1}] - \delta D(u_i^n)\right] \end{aligned} \tag{2.8}$$

without introducing any forcing term and using boundary conditions calculated at the intermediate time level, to obtain a preliminary velocity field $\tilde{u}_i^*$. Velocities $u_{IB,i}^*$ in IB points are then obtained through suitable interpolation of velocities $\tilde{u}_i^*$ and, finally:

$$\begin{aligned} u_i^* &= u_{IB,i}^* \quad \text{in IB points} \\ u_i^* &= \tilde{u}_i^* \quad \text{elsewhere} \end{aligned} \tag{2.9}$$

This procedure neglects the dependence of velocities $\tilde{u}_i^*$ in non-IB points on the forcing term f_i in IB points. It is worth noting, nevertheless, that the lack of the forcing term in IB points when resolving eq. (2.8) influences the velocity field $\tilde{u}_i^*$ everywhere through the implicit evaluation of diffusive terms. In order to evaluate the error introduced by this procedure, subtracting eq. (2.8) from eq. (2.6), we obtain:

$$[1 - \gamma\Delta t D](u_i^* - \tilde{u}_i^*) = \Delta t f_i \tag{2.10}$$

where $u_i^* - \tilde{u}_i^*$ is the difference between the *correct* and the *approximated* intermediate velocities. Since the forcing term f_i is non-zero in IB points only, the relation between u_i^* and $\tilde{u}_i^*$ in non-IB points is easily obtained from eq. (2.10) as:

$$u_i^* = \tilde{u}_i^* + \gamma\Delta t D(u_i^* - \tilde{u}_i^*) \tag{2.11}$$

Equation (2.11) clearly shows that although an error arises in calculating the velocities $\tilde{u}_i^*$, it is proportional to $\nu\Delta t$ (or to $\Delta t/Re$ when equations are made non-dimensional using suitable velocity and length scales $\hat{u}$ and $\hat{l}$ and Re $= \hat{u}\hat{l}/\nu$). Considering that the error arising in the implicit procedure here proposed decreases with the Reynolds number and that the time step is limited by the CFL condition, as will be extensively shown later on through the paper, the quality of the results is not appreciably affected by the time step of the simulation.

Once the intermediate velocity field is evaluated in all the active grid points, the velocities in the solid points are set equal to zero and the intermediate contravariant fluxes U_m^* are calculated. These fluxes are used for the solution of the curvilinear form of the Poisson equation (eq. (1.16)). Finally, according to the algorithm of [75], the new velocity field and the new contravariant fluxes are independently calculated using respectively eqs. (1.15) and (1.17). Similarly to the Cartesian-coordinate version of the direct forcing IBM of [22], in the numerical tests it has been observed that the results are insensitive to the values of the intermediate velocities and contravariant fluxes at the solid points.

2.1.1 Identification of *fluid* and *solid* points

Application of the *Immersed boundary* methodology preliminary requires the identification in the computational grid of *fluid nodes*, lying in the fluid phase, and *solid nodes*, which on the contrary are located in the solid phase. The fluid nodes surrounded by at least one point in the solid phase (see Fig. 2.3) are classified as *immersed boundary nodes* (IB).

Gilmanov et al. [26] proposed a procedure to identify *fluid* and *solid nodes* which can be applied to one immersed body only, having convex shape. More robust techniques are based on *ray-tracing* algorithms, similar to those used in 3D computer graphics (see for example [67, 27]). In this paper it is used a simple and robust ray-tracing procedure, which is able to address situations involving multiple and/or concave immersed boundaries. The surface of the immersed boundary Ψ bounding a closed body is discretised using an unstructured grid of plane triangular elements, as in [6]. The triangles are used since they are the plane geometrical figures better suited to adapt themselves to general shape surfaces and to univocally identify the normal direction. Then, the ray starting from one grid node and passing through the centroid of one arbitrary triangular element of the surface Ψ is considered. The ray will cross Ψ a finite number of times: the initial grid node is then classified as a *solid node* or a *fluid node* depending on whether the total number of these intersections is odd or even (see Fig. 2.4). The procedure is then repeated for each of the grid nodes, considering every time a new ray starting from the pivot node.

A particular situation occurs when the selected ray is tangent to the surface

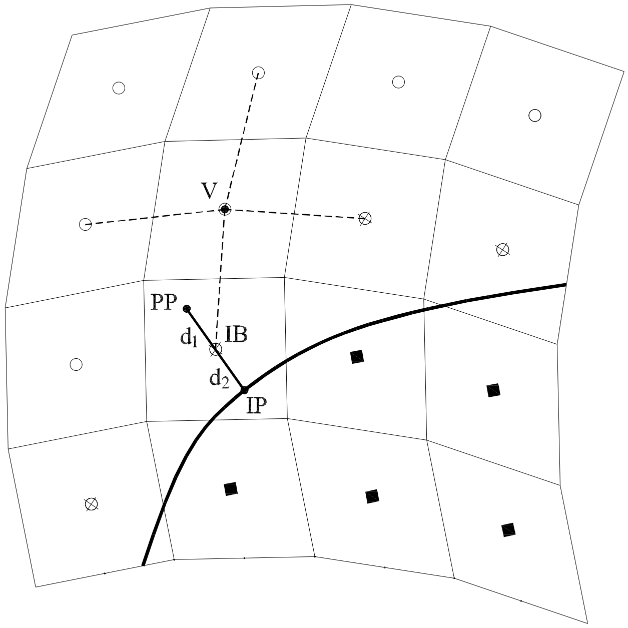

Figure 2.3: Identification, for a two-dimensional grid, of *solid nodes* (small solid squares), *fluid nodes* (small empty circles) and *immersed boundary nodes*, IB (small circles marked with a cross). For each IB node, identification of the *intersection point* (IP) and the *projection point* (PP), located at distances d_1 and d_2, respectively, from (IB). Here we choose $d_2 = d_1$. V is the pivot point for the interpolation of the fluid velocity upon the PP point.

Ψ, since the number of intersections of the ray with Ψ cannot be calculated and the procedure to identify *solid* and *fluid* nodes fails. To overcome this problem, whenever the ray intersects one of the triangles of the immersed boundary forming an angle with the normal to the triangle equal (or very close) to $\pi/2$ (see Fig. 2.5), the ray is replaced with a new ray obtained using a new surface element and the calculation of the number of intersections is restarted. This latter procedure is repeated if the new ray is again parallel to the surface. Since the surface of the immersed boundary is closed, there exist surface elements such that the ray joining its centroid with the grid point intersects the surface with an angle smaller than $\pi/2$. Finally, the grid nodes surrounding each *solid node* are classified as *IB points* if lying in the fluid phase.

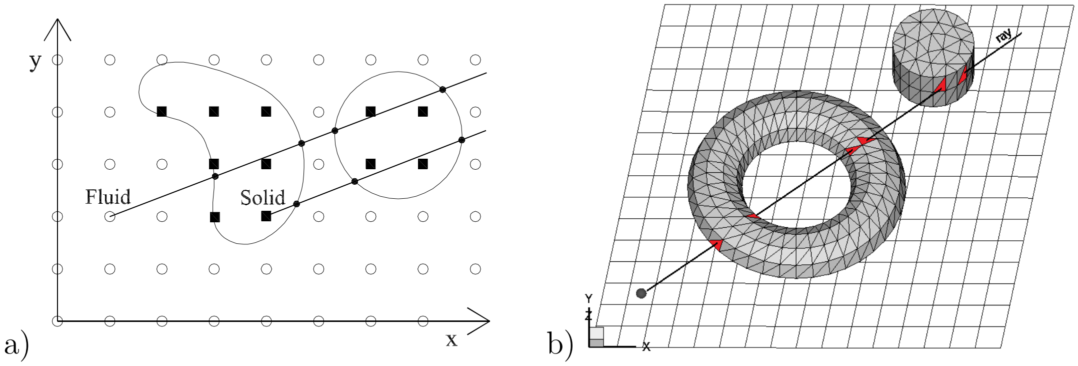

Figure 2.4: Identification of fluid and solid nodes, based on the ray-tracing technique. a) 2D scheme, where a Cartesian grid is showed for the sake of simplicity and the triangular surface elements collapse into segments. b) 3D example, showing a torus and a cylinder, whose surfaces are discretised using triangular elements.

Special situations may arise in presence of a moderate-to-strong concave surface. Figure 2.6 illustrates a typical case, where a cloud of IB points is located close to a concave surface. In these cases two strategies may be accomplished: 1) to increase the grid resolution in the concave region; 2) to switch off the internal IB points (namely the points which do not confine with fluid nodes) turning them into solid points. This operation corresponds to move the actual body surface over the IB nodes switched off.

2.1.2 Calculation of velocities in *IB* points, using curvilinear coordinates

Here, some modifications to the state-of-art techniques are proposed to extend the IBM to the curvilinear coordinate case. As discussed above, suitable interpolations are required to calculate the velocities in IB points whenever they

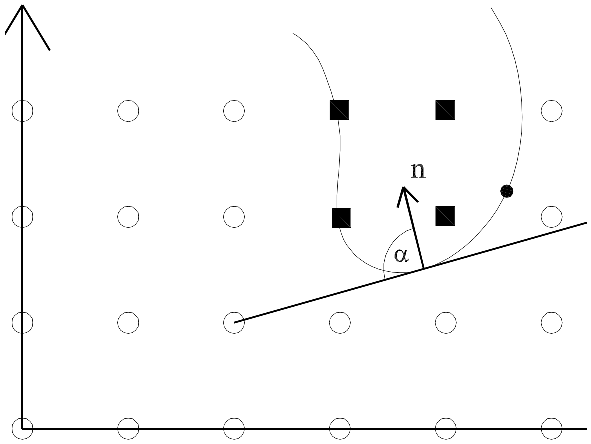

Figure 2.5: Occurrence of orthogonality condition ($\alpha \cong \pi/2$) between the current ray and the line normal to the immersed surface. A new ray must be chosen to identify solid and fluid nodes

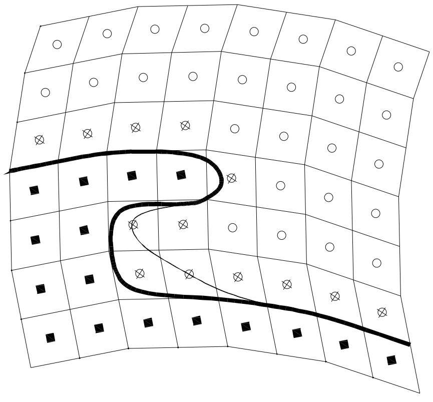

Figure 2.6: Concave surface where some IB points confine with IB and Solid points only (bold line). Turning internal IB points into Solid nodes corresponds to move outward the actual body surface (thin line).

do not lie on the immersed surface Ψ. Similarly to the proposal of [6], for each of the IB nodes two points (the *intersection point* IP and the *projection point* PP) are first identified. The point IP is placed at the intersection of the immersed boundary with a line l passing through the IB point and normal to the immersed boundary itself (see Fig. 2.3). Since the surface Ψ is discretised using triangular elements, the identification of lines normal to the immersed boundary is straightforward. The *projection point* (PP) is instead identified as a point belonging to the line l and standing on the opposite side of IP with respect to the current IB point. Specifically, PP is identified setting the distance d_2 between PP and IB equal to the distance d_1 between IP and IB (Fig. 2.3). After having identified the points IP and PP, the velocity in the current IB point (which is placed in the line l at the mid-point between IP and PP) can be obtained through the linear interpolation of the velocities u_{IP} in IP and u_{PP} in PP. Since points IP lie on the immersed boundary, velocities u_{IP} coincide with the immersed surface Ψ velocity and thus are known. The PP velocities can be obtained instead through the interpolation of velocities at the surrounding fluid nodes. This interpolation is straightforward when the fluid nodes are equispaced or when they belong to a Cartesian mesh. Typically, interpolation methods on Cartesian grids are linear and use different geometrical weighting procedures producing similar interpolation functions in terms of performance and accuracy. Some of these procedures make use of geometry-based algorithms which may be applicable only for regular meshes or may result in significant loss in accuracy when non-regular computational grids are used. Other procedures are based on iterative methods which require properly defined residuals to check the convergence of the solution (for a discussion see [42]). In order to extend the applicability of the IBM to curvilinear grids, th velocity u_{PP} is calculated using a very simple and effective interpolation technique suited for curvilinear meshes [42]. The technique is based on the use of the Taylor expansion of the function u_i about the *fluid node* closest to the PP point (referred to as node V in Fig. 2.3) up to the order of accuracy required by the NS solver. In order to deal with curvilinear meshes, the derivatives of the Taylor expansions are re-written in the computational space and calculated taking advantage of the metric terms already computed and stored for the space discretization of the governing equations. For second-order spatial accuracy we have:

$$u_{PP,i} = u_{V,i} + (x_{PP,i} - x_{V,i}) \left[\frac{\partial u_i}{\partial \xi_m} \frac{\partial \xi_m}{\partial x_i} \right]_V + O(\Delta x^2) \qquad (2.12)$$

Since the velocity derivatives in point V usually depend on velocities in IB points (see Fig. 2.3 for an example), we apply eq. (2.12) to all IB points and then iterate the procedure up to convergence. In the tests described in the next section, the iterative algorithm proved to converge to round-off error in three or four cycles. Although the application of this technique may look more expensive than simple interpolation between surrounding nodes, it requires a very small additional fraction of CPU time and is simpler and more accurate than linear interpolation applied to curvilinear meshes. When working with bodies with high curvatures or corners, the procedure described in this subsection may fail. In these cases, in fact, it may happen that no *intersection point* IP can be identified along the line normal to the surface Ψ (see for instance Fig. 2.7). In this case it is choosen as point IP the intersection between the line normal to the closest edge of the triangles and the edge itself (Fig. 2.7a). When even this intersection cannot be identified, as in Fig. 2.7b), the closest vertex of the triangle is used as point IP. The procedure then can be applied without any further change. In the next section the accuracy of the proposed method on curvilinear grids will be evaluated and its advantages with respect to applications on Cartesian grids will be shown.

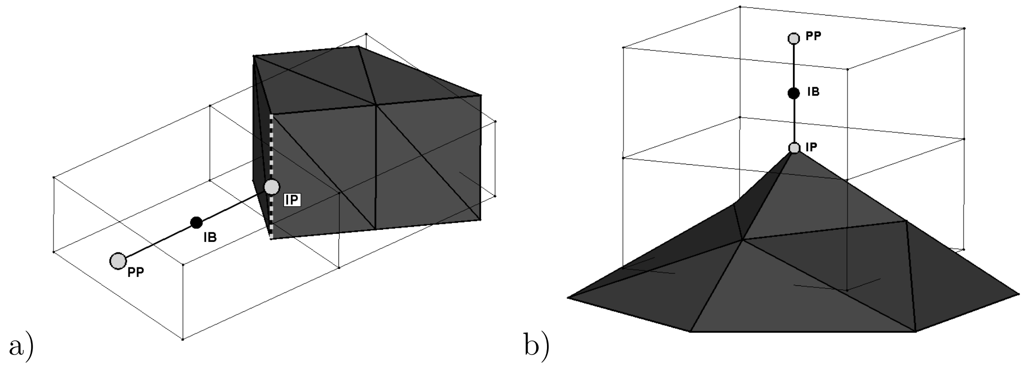

Figure 2.7: Examples of the proposed identification methodology of intersection (IP) and projection (PP) points for sharp or high curvature immersed surfaces.

2.2 Numerical tests

In the present section we show the results of three tests, aimed at the validation of the curvilinear coordinate, immersed boundary method (CC-IBM) herein proposed. First, in order to evaluate the accuracy of the method the results of simulations of a three-dimensional steady Stokes flow around a sphere are shown. The analytical solution is know in literature. Successively, in order to prove the reliability of the technique at larger values of the Reynolds number, the results of simulations of a two-dimensional, unsteady oscillatory flow around a circular cylinder at a finite value of the Reynolds number are reported. In particular it has been reproduced the test case used in the Priority Research Program 'Flow Simulation on High Performance Computers' of the German Research Association (DFG), whose results are extensively reported in Schäfer and Turek [62]. Finally it is quantified the superiority of the present approach with respect to a classical one which uses Cartesian grids in the problem illustrated in Fig. 2.1, which is representative of a class of applications where the use of a Cartesian grid may result in high computational inefficiency.

In the tests discussed in the present section the unsteady incompressible Navier-Stokes equations in curvilinear coordinates are solved using the finite-volume fractional-step method [75]. In the algorithm a collocated finite-volume method is implemented on a non-orthogonal structured grid consisting of non-overlapping hexahedral cells. Spatial derivatives are discretized using a second-order accurate centered scheme, whereas time-integration is carried out using an explicit second-order accurate Adams-Bashforth scheme.

The algorithm is described with details in [13] and [39] and is also used in [31].

2.2.1 Stokes flow around a sphere

The Stokes flow around a sphere is used to verify the accuracy of the proposed method in three-dimensional steady-flow under diffusion-dominated conditions. The well known analytical solution for a Reynolds number $Re = u_0 r_s/\nu < 1$ (where u_0 is the free-stream velocity, r_s the radius of the sphere and ν the kinematic viscosity) is, among others, in [38].

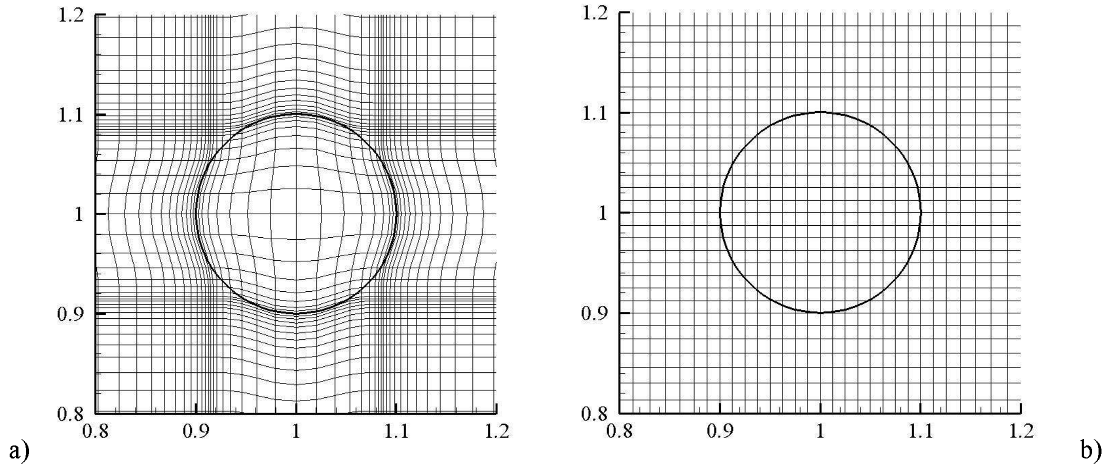

Figure 2.8: Curvilinear (a) and Cartesian (b) computational grids used to simulate the Stokes flow around a sphere. Detail of the xy symmetry plane.

The simulations are performed in a cubic domain extending from $-10r_s$ to $10r_s$ in all directions. The simulations are carried out using both Carte-

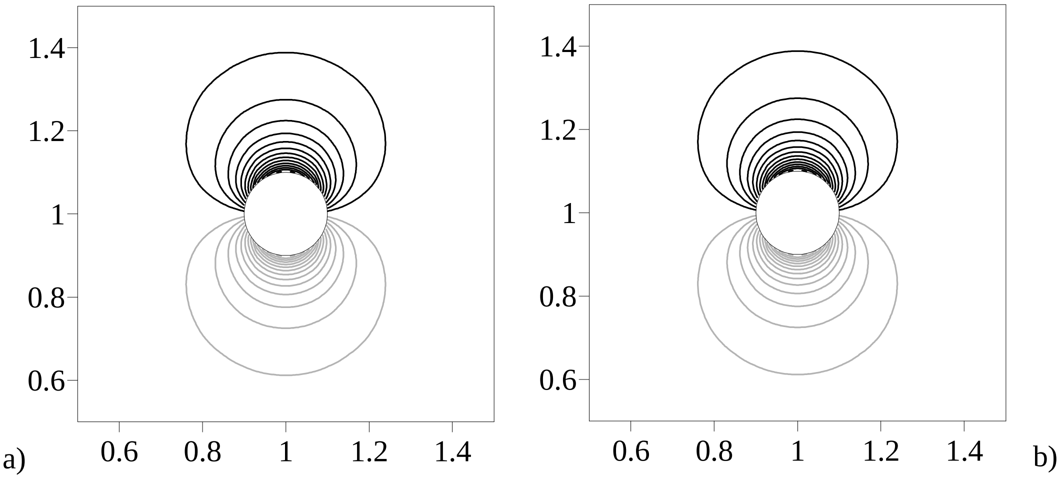

Figure 2.9: Stokes flow around a sphere for Re = 0.02. Comparison of analytical (left) and numerical (right) nondimensional vorticity field in the vertical streamwise midplane (xz). The quantity is made non dimensional with the ratio u_0/D. Positive values in black, negative values in gray. Steps = 0.05 (free-stream vorticity = 0). Numerical results obtained with the curvilinear grid.

sian and curvilinear grids as shown in Fig. 2.8. In both cases grid points are clustered in the sphere region to obtain a good near-body resolution. In the curvilinear simulation, additional clustering and deformation of the coordinate lines allows to better optimise the grid spacing when compared with the equivalent Cartesian grid. In both cases two grids are employed, namely a fine grid (80×80×80 cells) and a coarse grid (40×40×40 cells).

The inflow and lateral boundary conditions are assigned using the analytical solution, while stress-free conditions are imposed at the downstream outflow.

To quantify the accuracy of the numerical model, the root mean square error has been calculated as:

$$E_i = \frac{1}{u_0}\sqrt{\frac{\sum_{e,f,g} \Delta V_{e,f,g} \left(u^{num}_{i;e,f,g} - u^{an}_{i;e,f,g}\right)^2}{\sum_{e,f,g} \Delta V_{e,f,g}}} \tag{2.13}$$

where u_0 is the free-stream velocity, $\Delta V_{e,f,g}$ is the volume of the (e,f,g) computational cell, $u^{num}_{i;e,f,g}$ and $u^{an}_{i;e,f,g}$ are the numerical and analytical i-th components of velocity in the (e,f,g) cell and the sum is made over the whole domain. Application of eq. (2.13) to the numerical results gives the values 2.4, 1.35 and 1.36×10^{-6} for the curvilinear grid in the streamwise, spanwise and vertical directions, respectively, showing that a very good description of the velocity field is obtained using the proposed method. The analytical and numerical (curvilinear grid) vorticity fields in the horizontal midplane are shown

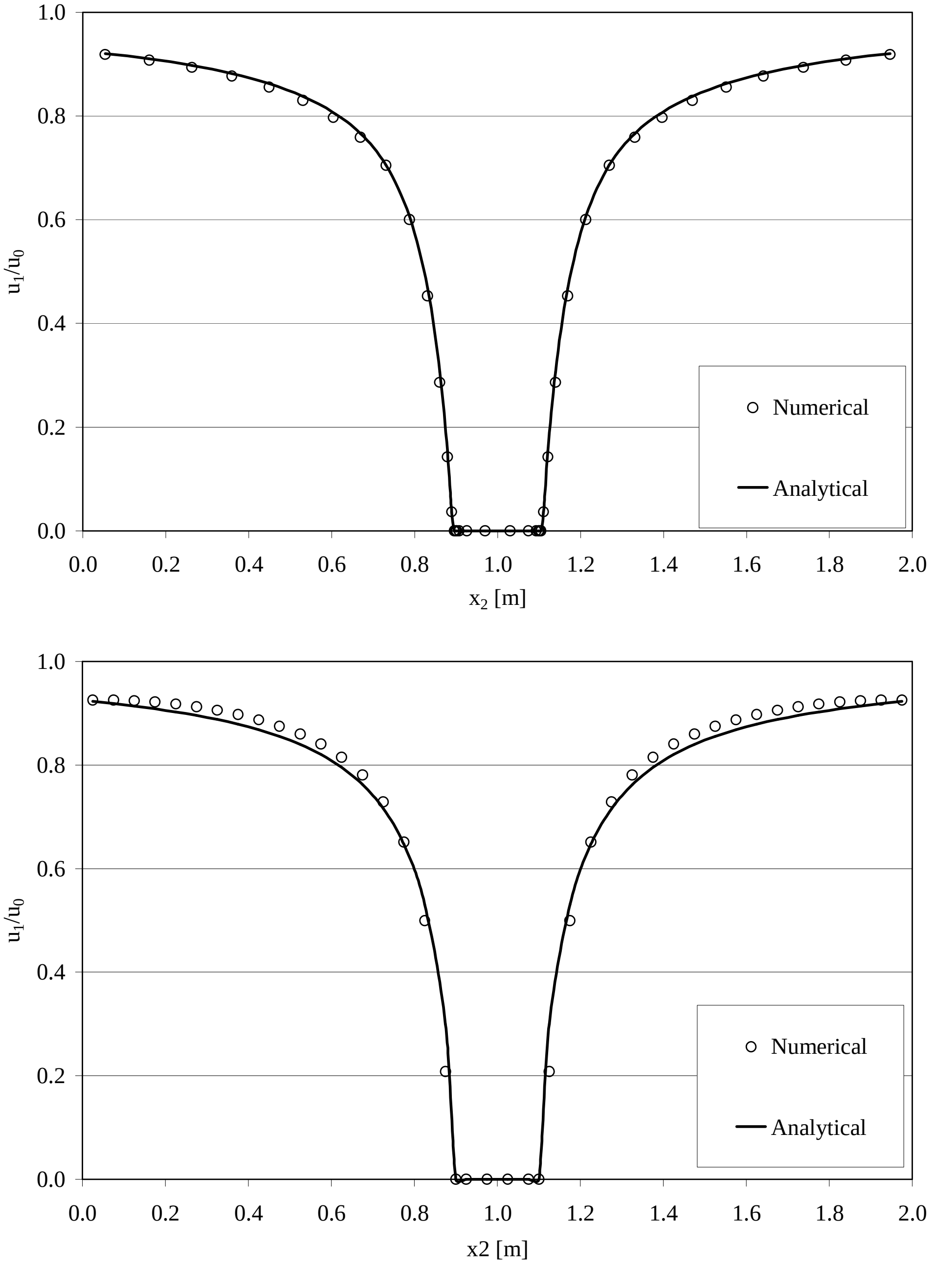

Figure 2.10: Stokes flow around a sphere for Re = 0.02. Comparison of analytical and numerical (40 × 40 × 40 cells) streamwise velocities in the spanwise axis of the horizontal midplane: curvilinear grid (top) and Cartesian grid (down).

Cases	N_x	N_y	$N_x \times N_y$
0	448	384	172032
1	224	192	43008
2	112	96	10752
3	56	48	2688

Table 2.1: Computational grids used for the simulation of the unsteady flow around the cylinder.

in Fig. 2.9, which gives visual evidence of the good performance of the proposed technique. The results obtained with the Cartesian grid are practically undistinguishable from those obtained using the curvilinear grid.

The simulations have been then repeated using coarser grids, composed of 40×40×40 cells. In this case the differences between the Cartesian and the curvilinear grid are clearly visible. In Fig. 2.10, analytical and numerical (Cartesian and curvilinear) streamwise velocity components along the axis $x_1 = x_3 = 1$ are compared, showing that the results obtained with the curvilinear grid are in better agreement with the analytical results than those obtained using the Cartesian grid. The rms of the numerical errors estimated using eq. (2.13) confirms this result, since the values obtained for the streamwise, spanwise and vertical velocity are, respectively, 0.75, 0.30 and 1.0×10^{-4} for the curvilinear grid and 1.6, 0.25 and $1.8(\cdot 10^{-4})$ for the Cartesian grid. The better performance of the curvilinear grid has to be attributed to the fact that it allows to obtain a more suitable distribution of the cells near the immersed surface, holding the level of accuracy of the IBM over Cartesian grids.

2.2.2 Unsteady flow around a cylinder

The unsteady two-dimensional flow around a cylinder with circular cross-section located between two infinite parallel walls is considered in this section. The aim of the test is to compare the performance of the algorithm in unsteady, finite Reynolds number, flows. As mentioned above, the test case herein considered was used in the Priority Research Program 'Flow Simulation on High Performance Computers' of the German Research Association (DFG) by 17 groups and the results collected in [62]. The results of the simulations by the different Research Groups gave a range of values for the *drag* and *lift* coefficients defined as:

$$c_D = \frac{2F_D}{\rho {U_m}^2 D} \qquad c_L = \frac{2F_L}{\rho {U_m}^2 D} \tag{2.14}$$

where U_m is the maximum value of the (time-dependent) cross-section averaged streamwise inflow velocity, D is the diameter of the cylinder and F_D and F_L are the components, in the horizontal and vertical directions, respectively, of the forces acting on the cylinder. Based on the values given in the database, results were considered to be accurate when lying within the following ranges:

$$2.93 \leq c_D \leq 2.97 \tag{2.15}$$

$$0.45 \leq c_L \leq 0.49 \tag{2.16}$$

The problem configuration is illustrated in Fig. 2.11a. The kinematic viscosity and the density of the fluid are respectively $\nu = 10^{-3}$ m^2 s^{-1}, and $\rho = 1$ kg m^{-3}. Due to the two-dimensionality of the geometry and of the flow field, the present case is studied using a two-dimensional version of the code. The time-dependent inflow boundary conditions are:

$$u(t,y) = 6U_m \frac{y(H-y)}{H^2} \sin\left(\pi \frac{t}{8}\right), \qquad v = 0 \tag{2.17}$$

where u and v are the horizontal (streamwise) and vertical velocity components, t is time, y is the vertical direction, $H = 0.41$ m is the height of the domain and $U_m = 1ms^{-1}$. The inflow velocity corresponds to a Poiseuille profile varying sinusoidally in time with a period $T = 16$ s. The Reynolds number of the flow, calculated using the velocity $U_m(t)$ and the cylinder diameter $D = 0.1$ m, varies between 0 and 100.

At the lateral boundaries no-slip conditions are used, whereas stress-free conditions are imposed at the outflow boundary. The initial condition corresponds to fluid starting from rest. The simulated time interval corresponds to half a period of the oscillatory flow given in eq. (2.17).

The simulations are performed using both Cartesian and curvilinear grids, with four different refinement levels (see Table 1). In both cases, the grid cells are clustered near the side walls and the cylinder surface. The differences between Cartesian and curvilinear grids are more evident in the region near the cylinder. Figs. 2.11b and 2.11c show a portion of the computational grids near the cylinder for Case 2 (see Table 1).

In Fig. 2.12 the vorticity fields around the cylinder obtained in Case 1 (224 x 192 cells) with the curvilinear grid are plotted. The figures refer to four phases along the oscillation, respectively $t/T = 1/8$, $1/4$, $3/8$ and $1/2$. Note that $t/T = 1/2$ corresponds to the time of inversion of the externally imposed motion. Fig. 2.12a shows that at $t/T = 1/8$, during the first stage of acceleration, a vorticity field develops at the solid walls and around the cylinder and is advected downstream by the inertial field. At this stage the symmetry of the flow is still retained and this has to be attributed to the choice of the initial conditions (fluid at rest at $t = 0$). At $t/T = 1/4$ (Fig. 2.12b) asymmetry is well evident in the wake region and at $t/T = 3/8$ (Fig. 2.12c) the alternate detachment of vortices downstream the cylinder is entirely developed. The near wall vorticity as well as the vortex structures in the near-cylinder region

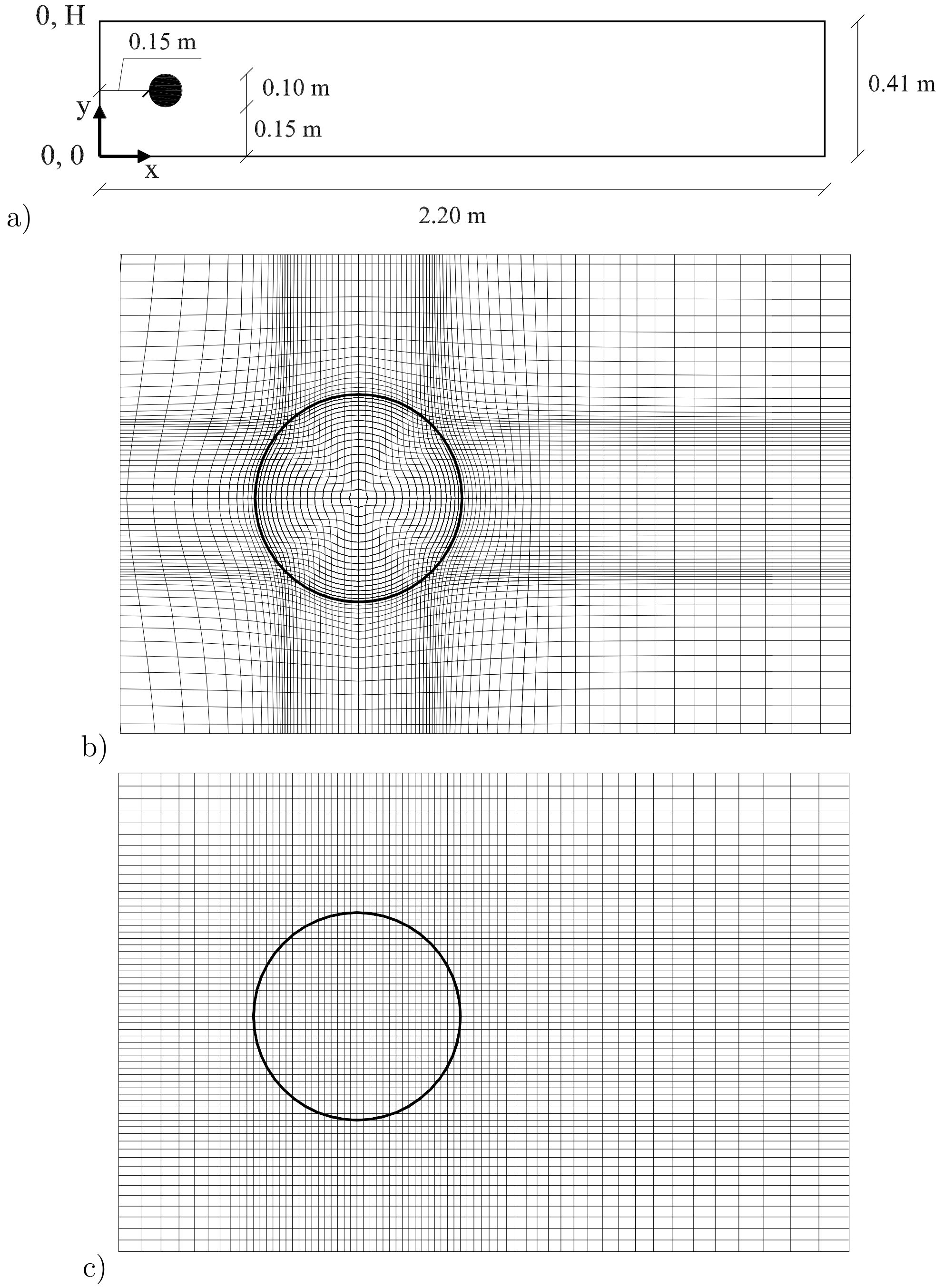

Figure 2.11: Unsteady plane flow around the cylinder confined between two parallel plane walls. a) Overall domain configuration; b) Near-cylinder curvilinear grid; c) Near cylinder Cartesian grid.

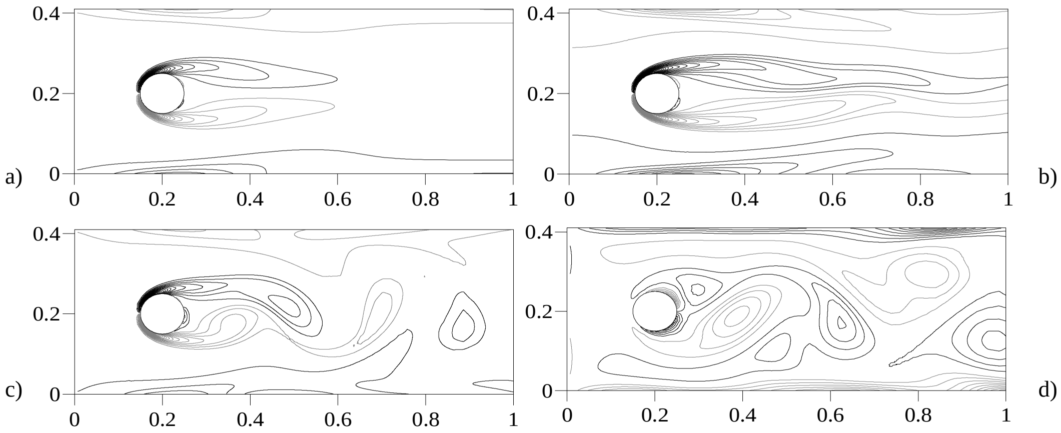

Figure 2.12: Unsteady plane flow around a cylinder. Nondimensional vorticity field (obtained using the curvilinear 224 × 192 grid (level 1)). The quantity has been made non dimensional with the ratio u_{max}/D where u_{max} is the maximum centerline velocity): (a) $t/T = 1/8$; (b) $t/T = 1/4$; (c) $t/T = 3/8$; (d) $t/T = 1/2$. Black and gray lines indicate positive and negative values, respectively (step 0.5 for (a), (b) and (c), 0.1 for (d))

.

are well visible at $t/T = 1/2$ (Fig. 2.12d) although the magnitude of vorticity is much smaller than before. As well known in literature, the mentioned effects typically occur in oscillatory flows. The presence of vorticity at the walls of the channel at the phase of inversion of the outer flow is due to the phase delay between the near wall velocity and the outer field occurring in an oscillatory boundary layer (see Salon et al. [61] and the literature reported therein). The presence of a wake around the cylinder during the inversion of the free-stream velocity in an oscillatory flow is discussed, among others, in [49] and in the literature reported therein. The numerical results show that the proposed methodology is able to capture the development of the unsteady flow around the cylinder. In particular the vorticity field, which often exhibits inaccuracies in the near-body region appears smooth and very well shaped in the whole fluid region. Note that similar results were also obtained with the Cartesian grid, thus showing that the quality of the results is independent on the grid distortion.

In order to quantify the differences in the results obtained with the Cartesian and the curvilinear grids, the *drag* and *lift* coefficients defined in eqs. (2.14) are calcolated. First we evaluate the forces F_D and F_L, whose definition is:

$$F_D = \int_S \left(\rho\nu \frac{\partial u_t}{\partial n} n_y - P n_x \right) dS \tag{2.18}$$

$$F_L = \int_S \left(\rho\nu \frac{\partial u_t}{\partial n} n_x + P n_y \right) dS \tag{2.19}$$

where n_x and n_y are the x- and y-component of the unit vector normal to the cylinder surface dS and u_t is the velocity component tangential to the cylinder. To improve accuracy in the calculation of the forces F_D and F_L we have preferred to calculate them by using the integral form of the momentum equation:

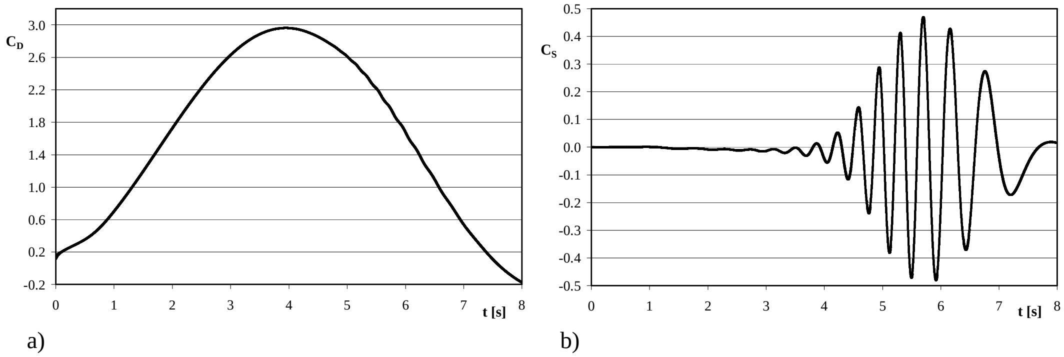

Figure 2.13: Unsteady plane flow around a cylinder. Time evolution of the maximum values of the drag (left) and lift (right) coefficients.

$$\mathbf{\Pi}_1 + \mathbf{\Pi}_2 + \mathbf{\Pi}_3 + \mathbf{\Pi}_4 + \mathbf{M}_1 - \mathbf{M}_2 + \mathbf{I} = 0 \tag{2.20}$$

where $\mathbf{\Pi}_1$, $\mathbf{\Pi}_2$, $\mathbf{\Pi}_3$ are the surface forces exerted on the fluid by the inflow, outflow and lateral boundaries, respectively, $\mathbf{\Pi}_4$ is the force exerted by the cylinder surface on the fluid, $\mathbf{M}_1$ and $\mathbf{M}_2$ are the momentum fluxes across the inflow and outflow boundaries and $\mathbf{I}$ is the local inertia term. F_D and F_L are the components in the horizontal and vertical direction of the term $-\mathbf{\Pi}_4$, which is easily obtained from eq. (2.20).

Figure 2.13 shows the evolution in time of the *drag* and *lift* coefficients. The *drag* coefficient grows with the mean velocity, but reaches the maximum value a few instants before the velocity; the *lift* coefficient is negligibly small during the initial phases when vortex shedding is symmetric and later on it oscillates (see [49] for a detailed discussion of the forces acting on a circular cylinder in an oscillatory flow).

In Fig. 2.14 the maximum values of the *drag* and *lift* coefficients obtained in the cases listed in Table 1 are plotted. Both Cartesian and curvilinear grids were used for each level of grid refinement. In Fig. 2.14 the lower and upper bounds of the coefficients are also plotted (see eqs. (2.15),(2.16)) as obtained using the data of the research groups involved in the aforementioned DFG Priority Research Program. The coefficients obtained in our computations fall within the prescribed range of values in cases C0 and C1, using both Cartesian and curvilinear grids. Conversely, when the number of grid cells is reduced (case C2), the results obtained with the curvilinear grid are still in the range of reference values, whereas when the Cartesian grid is used, this is no longer true. Again, this has to be attributed to the better grid quality obtainable with the

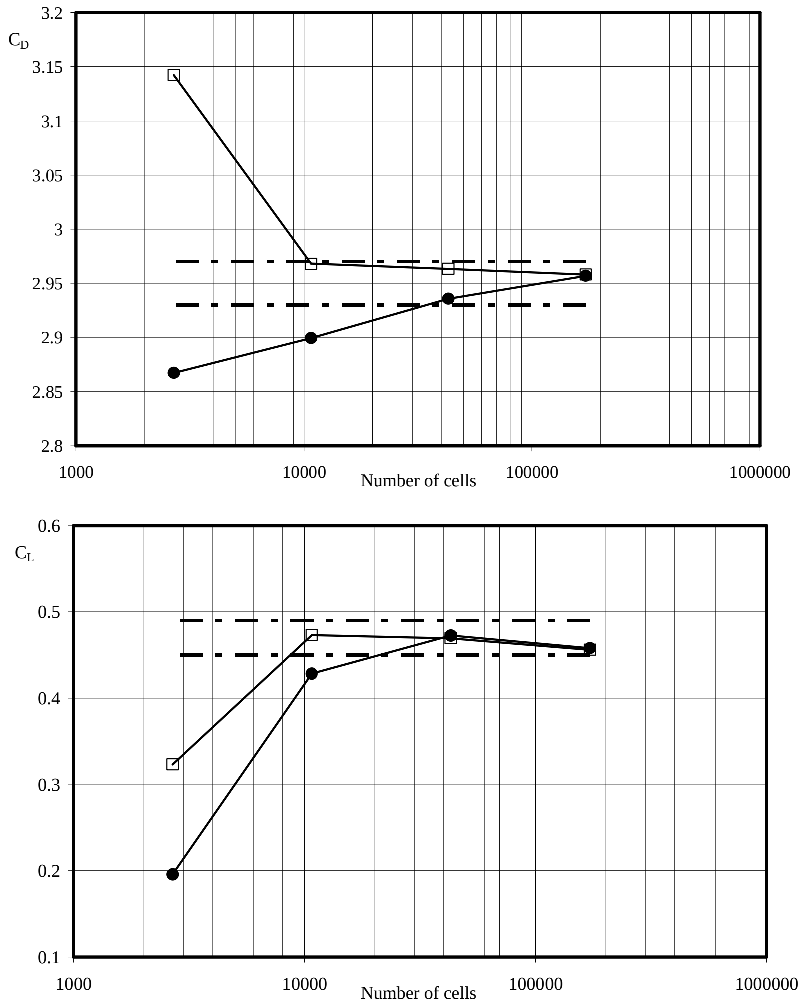

Figure 2.14: Unsteady plane flow around a cylinder. Maximum values of the drag (top) and lift (bottom) coefficients as a function of the number of computational cells. Continuous lines with empty squares: curvilinear grids. Continuous lines with circles: Cartesian grids. The dashed horizontal lines give the range of values of eqs. (2.15),(2.16).

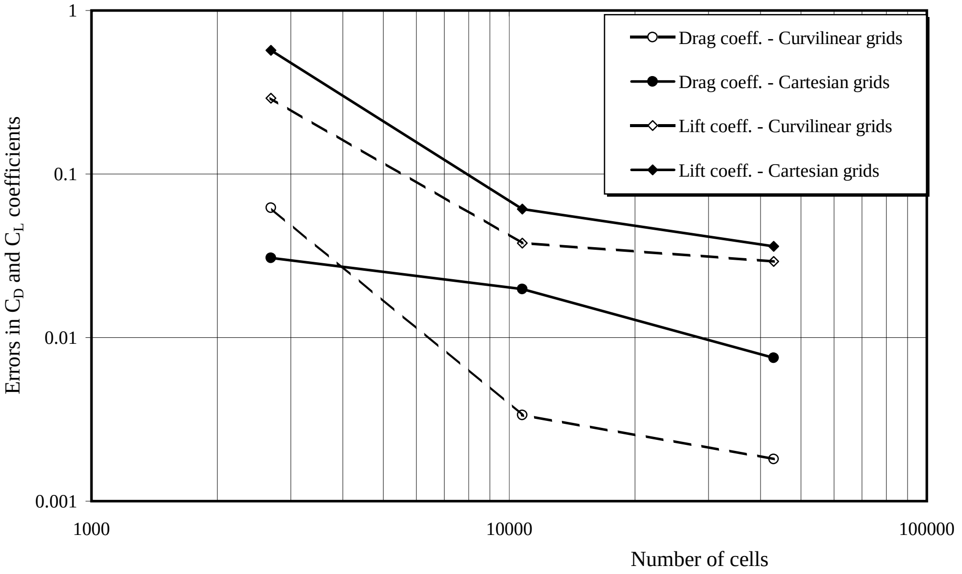

Figure 2.15: Relative error in the calculation of the coefficients C_D and C_L as a function of the number of computational cells, estimated using eq. (2.21). Empty symbols, dotted lines: curvilinear grid. Solid symbols, full lines: Cartesian grid.

curvilinear coordinates in the near-body region. Finally, the coarsest grid case (C3) gives the less accurate response in both cases (Cartesian and curvilinear grid respectively). In particular both the lift and the drag coefficients fall out of the range of values given in eqs. (2.15),(2.16).

In order to further confirm that the use of curvilinear grids gives more flexibility and consequently allows obtaining more accurate results with respect to Cartesian ones, in Fig. 2.15 it is shown the relative error in the calculation of the maximum values of the coefficients C_D and C_L for the cases C1, C2 and C3, taking as reference values those obtained on the finest grid C0:

$$Err = \frac{C_D^i - C_D^0}{C_D^0} \tag{2.21}$$

where C_D^i is the *drag* coefficient obtained in case (Ci) $(i = 1,2,3)$. The figure shows that the use of curvilinear grids systematically gives better prediction of the *lift* coefficient as well as of the *drag* coefficient, at least when a sufficient number of grid cells is used.

The Cartesian grid version of the curvilinear solver used in the paper allows to reduce the computational time by a factor of about 2.9, for a given number of cells. Although this result cannot be generalized (since major reduction can be achieved using different solvers, directly designed for Cartesian grids) we can argue that using curvilinear grids is more convenient when they allow reducing the number of computational cells more than three times (that is, about 30% in each direction for a 3D grid) with respect to a Cartesian grid.

2.2.3 Flow in a S-shaped duct with an internal valve

As previously discussed, there are problems where the use of a Cartesian grid may require an excessive number of solid nodes. A typical example is shown in Fig. 2.1, where an S-shaped duct with an internal valve is reported. Fig. 2.1a shows that the use of a Cartesian solver needs the treatment of most of the solid boundaries by IBM and that a large percentage of grid cells falls within the solid phase. On the other hand Fig. 2.1b clearly shows the advantage of using a curvilinear mesh, such that the external boundaries are aligned with the coordinate lines and the IBM is applied to account for the internal valve only. In this section it is calculated the flow field in laminar conditions in the duct of Fig. 2.1, and the results, obtained using the Cartesian and the curvilinear grids, and the computational requests are compared in the two cases.

The main dimensions of the duct are reported in Fig. 2.1b. The valve has an elliptic shape and its center is located in the middle of the symmetry axis of the vertical part of the duct. The major axis is inclined over the horizontal plane by $\theta = 45^o$ and its length is equal to $0.75h$; the minor axis is $0.25h$ long. The problem is two-dimensional and the inflow velocity profile is a Poiseuille flow with Reynolds number $Re = U_m h/\nu = 130$. The Cartesian grid has 256×256 grid cells respectively in the horizontal and vertical direction, whereas the curvilinear grid has 400×32 grid cells respectively in the streamwise and wall-normal direction. The choice of the number of grid cells in the curvilinear case has been made in such a way to minimize the differences in the cell dimensions between the two cases. Note that in the straight parts of the duct the grid cells dimension does not change in the two cases analyzed. A shear free boundary condition is employed at the outflow. It can be easily noticed that the Cartesian grid solver requires a number of grid cells as large as 5.12 times those used by the curvilinear solver. Note that this ratio is obtained with a uniform distribution of grid cells; clustering the grid cells near the walls will increase the ratio between the number of cells to be used by the Cartesian code when compared with that of an equivalent curvilinear code simulation.

Figure 2.16 shows the near-valve vorticity field obtained in the two simulations. Differences appear in the near-valve region, namely a spot-like behavior is observed in the Cartesian case, whereas smoother contours are obtained in the curvilinear case. The comparison between the velocity profiles at four meaningful sections is shown in Fig. 2.17. Some differences are observed between the two cases, the largest being present in the wake region (Fig. 2.17c). In order to give a comparison between dynamical quantities obtained in the two simulations, an overall drag coefficient has been calculated as:

$$C_{duct} = \frac{F_{P,1} - F_{P,2}}{\rho U_m^2} \qquad (2.22)$$

where $F_{P,1}$ and $F_{P,2}$ are respectively the pressure forces at the inflow and at

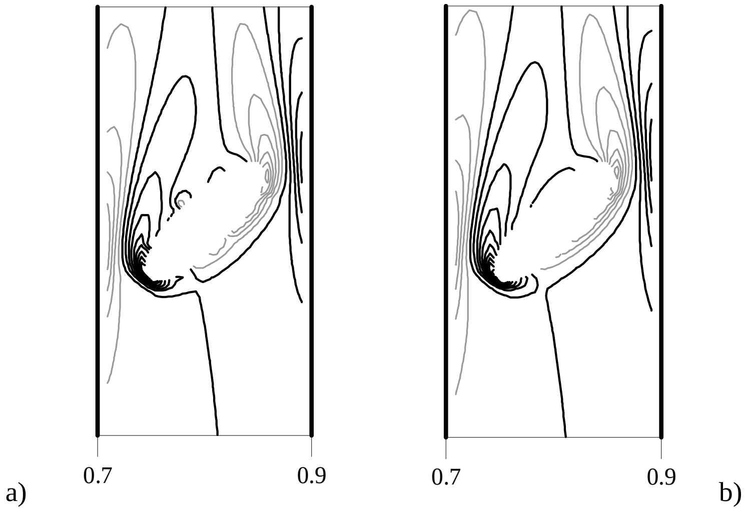

Figure 2.16: Nondimensional vorticity field in the near valve region. The quantity has been made non dimensional by the ratio U_m/h where U_m is the mean velocity. Positive values in black, negative values in gray. Steps=3.0. Left, Cartesian simulation; Right, curvilinear simulation.

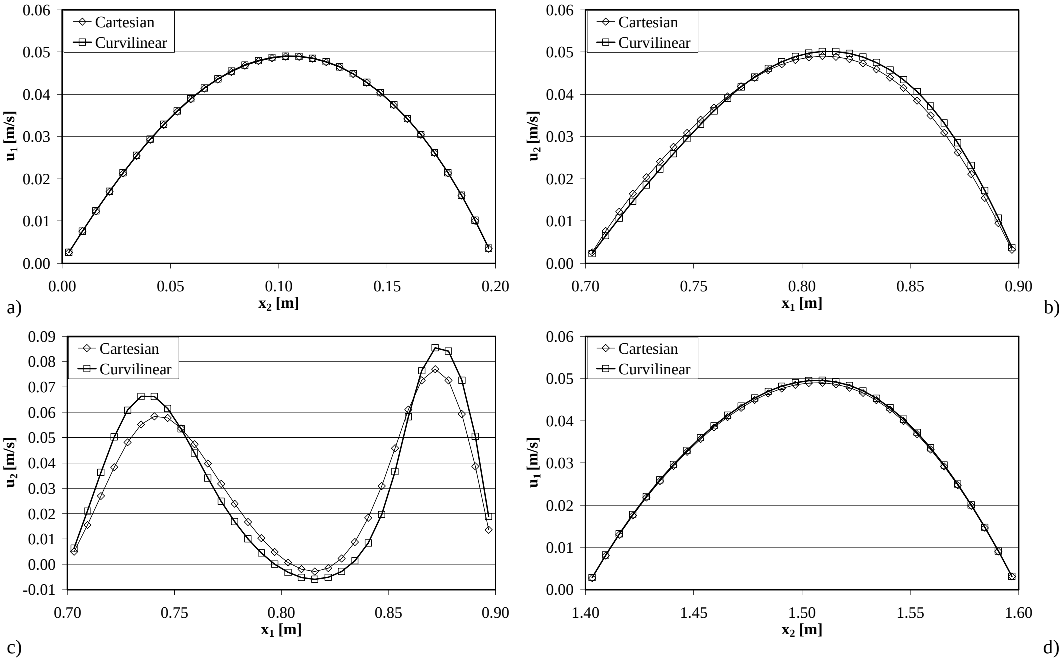

Figure 2.17: Comparison of streamwise velocity profiles at four wall-normal sections shown in Fig. 1b. Thin lines, Cartesian grid solution; thick lines, curvilinear grid solution.

the outflow sections. Since the outflow velocity profile is equal to the inflow one, due to the fact that a Poiseuille flow rapidly recovers after the upper curved part of the duct, by application of the integral form of the momentum equation it can be easily shown that the difference $F_{P,1} - F_{P,2}$ is equal to the total drag force within the duct. Again, some differences are obtained in the two cases: the Cartesian grid simulation gives $C_{duct} = 6.59 \cdot 10^{-4}$, whereas the curvilinear simulation gives a value $C_{duct} = 6.81 \cdot 10^{-4}$.

The results obtained with more refined grids (512 x 512 cells for the Cartesian grid, 800 x 64 for the curvilinear one) are indistinguishable from those shown above, demonstrating that the grid convergence has been reached in both cases. The slight differences between the results on Cartesian and curvilinear grids are thus due to the increased number of IB points required when working with the Cartesian grid with respect to the curvilinear case (where IB points are located near the valve only, while the curved part of the duct is discretised using boundary conforming cells).

The results of the present test show that more accurate results are obtained using the curvilinear strategy. In this case the number of grid cells required is less than 20% those needed by the Cartesian solver. If we consider that the use of the curvilinear version of the solver requires a CPU time per iteration as large as 2.9 times that required by the Cartesian version, we can be assert that in the present case the computational cost of the Cartesian solution is equal to 1.77 times that of the curvilinear solution. Note that the present test is relative to a two-dimensional geometrical configuration. For instance, in a case where the vertical part of the duct is inclined in the spanwise direction, the computational cost of the Cartesian simulation will get much higher than that of the curvilinear one.

2.3 Explicit forcing with Semi-implicit time-advancement

As discussed in section 2.1, the proposed semi-implicit direct forcing IBM is a compromise approach that may introduce some inaccuracies with respect to the explicit formulation as well as to the implicit algorithm of [34]. The test case reported below is thus used to compare the performance of the proposed semi-implicit method with that of the explicit approach, whose accuracy was checked in the previous section. It has to be pointed out that this section is not devoted to discuss the formal accuracy of the semi-implicit IBM herein proposed against that of explicit schemes. Rather, in this section we intend to show that the simplifications introduced in our semi-implicit procedure do not appreciably alter the quality of the results for moderately large values of the time step. Specifically the main aims of these simulations are: a) to show that the accuracy of the proposed semi-implicit method is identical to that of the corresponding explicit one when the same time step is used; b) to prove that the increase of the time step within a reasonable range does not produce appreciable reduction of accuracy in our procedure; c) to show that the error decreases increasing the Reynolds number.

To these aims, here a numerical simulations is performed of the 2D flow around an infinite array of circular cylinders placed in a Poiseuille flow. The simulations are carried out at different values of the Reynolds number using both an explicit solver and a semi-implicit one. The solver herein employed is that described in [4] and it uses the curvilinear-coordinate fractional step method of [75]. In particular the explicit version uses a second-order Adams-Bashforth method, hence both the CFL and the diffusive conditions on the time step must be satisfied. In the semi-implicit algorithm, the diffusive terms only are advanced in time using the implicit Crank-Nicolson scheme, whereas the other terms are treated explicitly by means of the Adams-Bashforth method. This allows removing the diffusive restriction on the time step.

Note that the CFL and the diffusive conditions in curvilinear coordinates read as:

$$CFL = (|U^1|+|U^2|+|U^3|)\frac{\Delta t}{J^{-1}} < 1 \ , \ DL = \left(\frac{1}{G^{11}} + \frac{1}{G^{22}} + \frac{1}{G^{33}}\right) J^{-1}\nu\Delta t < 0.5$$

where U_m and G^{ii} were defined in section 1.2.

The geometrical characteristics of the computational domain and the boundary conditions are described below.

The channel height is $L_z = h$, the longitudinal and the spanwise extensions are respectively $L_x = 2.2h$ and $L_y = h$, the cylinder has a diameter equal to $D = 0.2h$ and its center is located at the center of a longitudinal vertical section. The domain is discretized using a Cartesian grid composed of $144 \times 8 \times 128$ cells in the streamwise, spanwise and wall-normal directions respectively. The grid cells are clustered in the region where the circular cylinder is located. No slip boundary conditions are imposed at the bottom and top solid walls, whereas periodicity is considered in the streamwise and in the spanwise directions. Periodicity in the spanwise direction, at the values of Re considered in our tests, ensures two-dimensionality of the flow field; as in [50] periodicity in the streamwise direction mimics an infinite array of cylinders with distance between the centers of two successive cylinders equal to L_x.

The simulations are performed for three values of the Reynolds number ($Re = 0.2, 20$ and 40) in order to compare the performance of explicit and implicit schemes in different flow conditions. For $Re << 1$ the flow field is dominated by the diffusion of momentum (Stokes regime); at $Re = 20$ two steady symmetric recirculation regions are present downstream the cylinder, whereas at $Re = 40$ the flow is characterised by the presence of the unsteady von Karman vortex street. Different values of the time step are considered to evaluate its own effect on the quality of the results.

Case	Re	Scheme	$\Delta t \cdot 10^{-3}$	CFL	DL	C_R
S1	0.2	Expl	0.0489	0.55×10^{-3}	0.089	108.483
S2	0.2	Impl	0.0489	0.55×10^{-3}	0.089	108.261
S3	0.2	Impl	0.1954	2.2×10^{-3}	0.780	107.523
S4	0.2	Impl	0.4873	5.48×10^{-3}	1.950	105.881
S5	0.2	Impl	4.7369	53.34×10^{-3}	18.95	89.919
E1	20	Expl	0.30	0.05	0.2	2.081
E2	20	Impl	0.30	0.05	0.2	2.077
E3	20	Impl	1.50	0.27	1.0	2.082
E4	20	Impl	2.40	0.43	1.6	2.086
K1	40	Expl	0.27	0.053	0.108	0.15343
K2	40	Impl	0.27	0.053	0.108	0.15341
K3	40	Impl	1.30	0.250	0.520	0.15485

Table 2.2: Computational parameters and drag coefficient for the periodic flow around a cylinder at three values of the Reynolds number.

In all simulations the drag coefficient

$$C_R = \frac{2\ D_{max}}{\rho U_m^2} \tag{2.23}$$

is calculated and reported in Table 2 together with the parameters of the simulations. In eq. (2.23) the drag coefficient is calculated using the maximum value D_{max} of the drag recorded in the simulation and the mean velocity in the channel, in a section which does not contain the cylinder.

As regards the low Reynolds number case $Re = 0.2$ five simulations are carried out with values of the time step spanning one order of magnitude. The cases with smallest value of the time step are run using both the explicit and the semi-implicit algorithms (S1,S2). The other values of the time step (S3-S5) are run with the semi-implicit code only. At $Re = 0.2$ (Stokes regime) the flow field around the cylinder is symmetric (see Fig. 2.18a). Table 2 shows that the drag coefficient is nearly unaffected by the time-advancement scheme for cases S1 and S2. This means that the explicit forcing method herein proposed is very robust and can work indifferently with explicit and semi-implicit time advancement schemes. Note that the drag coefficient $C_R = 108.45$ is larger than the value found in the case of a cylinder in a unbounded flow field (in this case at $Re = 0.2$, $C_R \sim 35$). The increase of the drag coefficient has to be attributed to the presence of the two walls and consequently to the appreciable increase of the mean velocity in the cylinder region due to the blocking effect ([77]). The progressive increase of the time step (cases S3 to S5) tends to deteriorate the quality of the results, although a very good evaluation of C_D (within 2.5%) is obtained with a time step as large as 10 times that of the reference simulations S1 and S2. The further increase of the time step (case S5, $DL = 18.9$) causes strong inaccuracies in the flow field, with a drag coefficient under-predicted by about 20%. This has to be attributed to the error introduced by the simplified semi-implicit procedure herein proposed which, as discussed in section 2.1, is proportional to $\Delta t/Re$. This error is likely to be non-negligible in low-Re calculations with high values of the time step. To summarize, in diffusion-dominated flow fields, the explicit treatment of the forcing term with semi-implicit time-advancement of the Navier-Stokes equations produces inaccuracies only when very large values of the time step are employed.

As discussed above, for $Re = 20$ the flow field exhibits two mirror-symmetrical recirculation bubbles in the wake region (see Fig. 2.18b). Four simulations (E1-E4) are performed. E1 and E2 respectively refer to the smallest time step, using explicit and semi-implicit time advancement. E3 and E4 refer to semi-implicit simulations with increasing values of the time step. The range of time steps considered spans one order of magnitude. For the range of values of Δt herein considered, we observe that the drag coefficient is nearly unaffected by the time advancement scheme as well as by the value of the time step.

Finally, the case $Re = 40$ exhibits the presence of a von Karman vortex street and the flow field is unsteady although it remains two-dimensional (Fig. 2.18c). Three simulations are carried out in this case: K1 and K2 using a small value of Δt and, respectively, the explicit and the semi-implicit algorithm; K3 where a value of Δt as large as about five times that used in K1 and K2 is considered. Similarly to the E-cases the increase of Reynolds number makes the results less and less sensitive to the employed value of Δt.

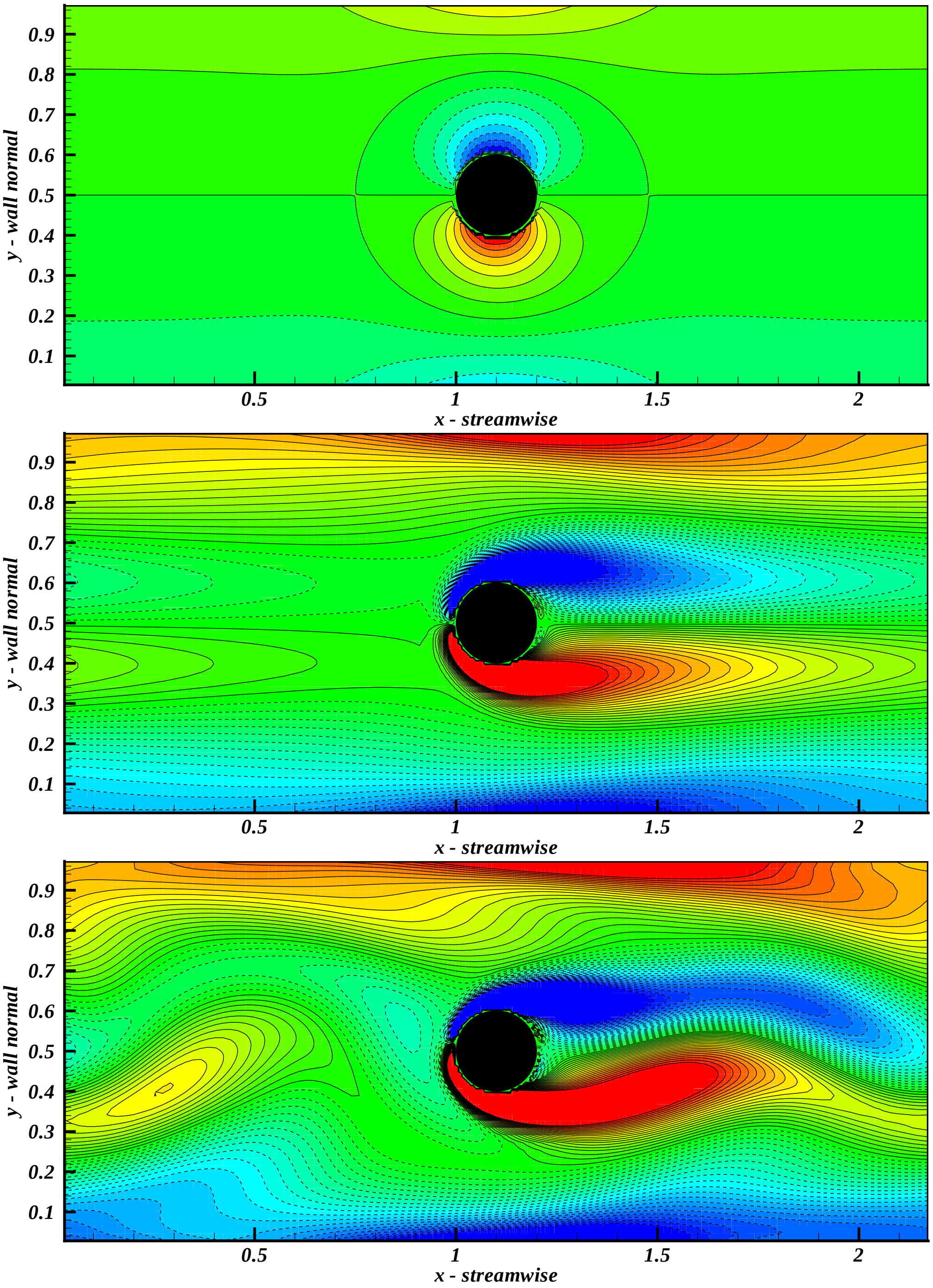

Figure 2.18: Vorticity field around the cylinder: a) Case S2, $Re = 0.2$; b) Case E2, $Re = 20$; c) Case K2, $Re = 40$.

Chapter 3

Wall modeling with an Immersed Boundary Method

The work presented in this chapter comes from a scientific collaboration with Prof. J. Frölich of Technical University of Dresden, the collaboration was partially supported by COST action P-20 [16].

In the present chapter the Immersed Boundary Methodology developed in chapter 2 will be extended to high Reynolds number flows. In the case of high Reynolds number flow high accuracy techniques as direct numerical simulation and *resolved* LES simulation (LES that resolves the viscous sub-layer) cannot be used, in particular when turbulence develops near solid walls.

As already discussed in chapter 1 in LES the large, energy-carrying, scales of motion are directly resolved through numerical integration of the three-dimensional, unsteady, filtered Navier-Stokes equations, while the small ones (those smaller than the size of the computational cell) are modeled through a subgrid-scale (SGS) model. The presence of solid walls makes the computational requests of LES more stringent with respect to the wall-free turbulent case. Solid walls or interfaces induce the formation of small elongated vortical structures (streaks) in their vicinity. These quasi-streamwise vortices are confined within the buffer layer and their dimensions scale with the Reynolds number. Their dynamics affect the flow and hence they must be properly resolved, thus requiring a large amount of computational nodes placed in a very narrow region of the domain. This makes the computational cost of *resolved* LES to scale with $Re^{2.5}$ where Re is a Reynolds number based on some inertial length scale and velocity scale (see [54]). LES are then required to be DNS-like near the wall. This implies that *resolved* LES is not feasible at very high Reynolds number. The computational cost confines these simulations far from typical real life problems.

The use of wall-layer models to skip the solution of the viscous layer is one of the possible strategies to overcome this limit (for a detailed review see [54]). Wall-layer models allow to place the first grid point at several wall units far from the wall, typically in the logarithmic region of the velocity profile.

The imposition of a wall-layer model is usually performed reconstructing the local wall shear stress from the knowledge of the tangential velocity at the first grid point. The imposition of the wall shear stress is straightforward when using body-fitted meshes; in this case the wall shear stress can be *directly* imposed as a flux at the wall and the boundary condition on the tangential velocity field is not required. An alternative way is to impose the wall shear stress by proper setting of the eddy viscosity at the first grid node.

The demand of numerical simulations of applicative high-Reynolds complex geometry cases, requires a proper wall-layer modeling to be used in conjunction of IBM. Although attempts have been done toward the solution of such a problem, it is still an open issue. The main problem with the immersed boundaries is that it is not possible to apply directly the computed stress on the boundary, because in the general cases the boundary does not coincide with the boundary of a computational cell. Even using the eddy viscosity setting at the cell neigbouring the immersed boundary surface, problems arise in the flux reconstruction. Tessicini et al. [69] used a boundary layer equation over a finer grid close to the immersed boundary; the same approach was used by Cristallo et al. [17]. In these cases the procedure consists in resolving a boundary layer equation along the wall-normal direction in an iterative way with boundary conditions obtained from the interior flow resolved by LES and the condition at the wall. An alternative approach was proposed by Choi et al. [12] who used an exponential law to scale velocity at IB points.

In this chapter a strategy is proposed to model the wall shear stresses at immersed boundary surfaces. The idea is to use a near wall equivalent eddy viscosity, obtained in an analytical way assuming a log-layer profile at the first grid point off the body and to scale the velocity at IB points with the logarithmic law. This idea was used by Cabot for wall modeling in Cartesian mesh without IBM. The model is tested in a variety of cases, employing different grids and positions of the cells with respect to the immersed surface. The model is implemented in the contest of the IBM developed in the previous chapter.

3.1 Wall modeling

The aim of wall functions is to skip the direct resolution of the viscous sublayer near the wall through proper parameterization. In this way the grid size can be related just to the large eddies developing in the fluid core and the required grid spacing becomes less dependent on the Reynolds number. This procedure however requires the parameterization of the effect of the wall-layer on the resolved flow. In the inner layer a logarithmic profile for the velocity can be derived [55]:

$$V^{+} = \frac{1}{\kappa} log\, y^{+} + B \tag{3.1}$$

where V^{+} is the tangential velocity scaled with the friction velocity $u_\tau = (\tau_w/\rho)^{1/2}$ (τ_w is the shear stress at the wall and ρ is the density). The wall-normal coordinate is $y^{+} = y/l_\tau$ with $l_\tau = \nu/u_\tau$. The von Karman constant κ is 0.41, while the offset B depends on the roughness of the wall. A classical approach is to place the first computational node within the logarithmic layer and to calculate the wall shear stress τ_w using eq. (3.1). In the seminal work of Schuman [63] the shear stress components at the wall $\tau_{yx,w}$ $\tau_{yz,w}$ are related to the velocity at the first grid point y_1 as:

$$\tau_{yx,w}(x,z) = \frac{\langle \tau_w \rangle}{\langle \overline{u}(x,z,y_1) \rangle} \overline{u}(x,z,y_1) \tag{3.2}$$

$$\tau_{yz,w}(x,z) = \nu \frac{\overline{w}(x,z,y_1)}{y_1} \tag{3.3}$$

In eqs. (3.2) and (3.3) x y and z are the streamwise, the wall-normal and the spanwise direction respectively, whereas y_1 is the distance of the first grid point from the wall. The over-bar denotes resolved quantities and $\langle \cdot \rangle$ a Reynolds-average operation. The mean stress $\langle \tau_w \rangle$ is obtained from an imposed pressure gradient, $\langle \bar{u}(x,z,y_1) \rangle$ is related to $\langle \tau_w \rangle$ by the logarithmic law (3.1). The spanwise component is calculated assuming a linear velocity profile and a constant eddy viscosity in the grid cell nearest to the wall. This parameterization considers a RANS-like model near the wall and a LES model in the core of the fluid. This approach assumes that the wall shear stress is instantaneously in phase with the local velocity. The drawbacks of this method are that mean values are required. In channel flow case it requires knowledge of the pressure gradient in order to deduce $\langle \tau_w \rangle$, in complex geometry problems it is difficult to obtain this mean value.

Based on [63] other approaches were proposed to treat the wall-layer, assuming the stress and the velocity being locally in phase and using logarithmic or exponential profile. For a review see [54].

Both in the Cartesian and in the curvilinear coordinate framework the imposition of the stress at the wall is easily accomplished replacing the discretized viscous wall flux with the actual value of the stress. This can be easily performed because the boundary of the computational domain coincides with a physical boundary. When using immersed-boundary methodologies this is not the case (for a general review on IBM see [45]).

Very few literature studies are available on wall modeling with immersed boundary. Tessicini et al.,[69] and Cristallo and Verzicco, [17] proposed a wall model in conjunction with the IBM. They used a boundary layer equation for the tangential velocity close to the Immersed Boundary:

$$\frac{\partial}{\partial \eta}\left((\nu+\nu_t)\frac{\partial u_i}{\partial \eta}\right) = F_i \;\; with \;\; F_i = \frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} + \frac{\partial p}{\partial x_i} \tag{3.4}$$

being η the wall-normal direction. This equation is solved in a layer between the external LES and the wall. Its solution provides the boundary condition for LES. The simplest way to approach eq. (3.4) is to use $F_i = 0$ obtaining the equilibrium stress balance model. Then the eddy viscosity is computed by a mixing length model, $\nu_t = \nu k \eta^+ (1 - e^{-\eta^+/A})^2$ with $A = 19$, η^+ is the distance from the wall in wall units. The computation of ν_t requires u_τ that comes from the solution of eq. (3.4) so the system must be solved in an iterative way.

An other approach is that of Choi et al. [12]. In order to interpolate velocity at the IB nodes they adopt a formula written in terms of a power law η^k with $k = 1/7 \;\; or \;\; 1/9$. In this way they approximate the logarithmic profile in the near wall region.

The method here proposed is closer to that of Choi. However the interpolation scheme at the Immersed Boundary interface is derived directly from the logarithmic law. Moreover a different approach is used to compute the eddy viscosity at this interface. The method is explained in the next section.

3.2 The proposed approach

In DNS or *resolved* LES the first off-the-wall computational node is placed at $y^+ = 1$ or even closer to the wall. At this location the velocity profile is linear. Then a linear scheme can be used to interpolate the required velocity at the Immersed Boundary interface from the interior resolved flow. Details are in section 2.1.2. However a possible scheme is the following:

$$V_{IB} = \frac{V_{PP} + V_{IP}}{2} \tag{3.5}$$

where V_{IP} is the velocity at the solid surface (intersection point) and V_{PP} is the velocity at the projection point PP (see chapter 2 for the nomenclature).

The velocity at PP is derived by interpolation from the interior resolved flow through a Taylor expansion (see Marchioli et al. [42]). If the body is at rest $V_{IP} = 0$ and $V_{IB} = V_{PP}/2$.

In high-Reynolds turbulent flows, as discussed above, the near-wall is not resolved and a wall-layer model must be implemented. With a coarse grid IB points can be far from the wall and a relation different from eq. (3.5) is needed in order to provide a turbulent velocity profile close to the wall.

If the logarithmic law (3.1) is assumed to hold both at IB point and at PP point:

$$V_{IB}^+ = \frac{1}{\kappa} log\,(d_{IB}^+) + B \tag{3.6}$$

$$V_{PP}^{+} = \frac{1}{\kappa} log\,(d_{PP}^{+}) + B \tag{3.7}$$

where d_{IB}^{+} and d_{PP}^{+} are the distance from the wall of the points IB and PP scaled with the length scale l_τ. Subtracting eq. (3.7) from eq. (3.6) a simple relation to derive velocity at IB point is obtained:

$$V_{IB} = V_{PP} - \frac{1}{\kappa}\sqrt{\frac{\tau_{wall}}{\rho}} log\left(\frac{d_{PP}}{d_{IB}}\right) \tag{3.8}$$

Then once the wall stress is known it is easy to compute the velocity at IB. The stress can be computed iteratively from eq. (3.1) using the tangential velocity at PP. After the velocity at IB is computed with eq. (3.8) the Cartesian velocity components at IB can be reconstructed assuming an alignment between the velocity at PP and IB.

As we will show in section 3.3 this procedure is not sufficient to provide good results even in simple case like a turbulent channel flow. If we adopt a Smagorinsky model to reproduce the SGS a different procedure at the IB points it is necessary.

The Reynolds stress can be reproduced through an eddy viscosity ν_t characteristic of the flow:

$$\frac{\tau_{ij}}{\rho} = 2\nu_t S_{ij} \tag{3.9}$$

In LES most of the Reynolds stress is directly resolved and only a small part of it must be modeled, the part lost because of the grid resolution. The use of a coarse grid close to the wall implies averaging over many turbulent structures and in the end LES is more close to a RANS than to a DNS. Then the eddy viscosity is required to be able to reproduce the entire Reynolds stress close to the wall. The mixing length approach in the boundary layer equation (3.4) uses such principle.

One of the common way to compute the eddy viscosity is to use a Smagorinsky model. In this model the eddy viscosity is expressed as the product of a length scale l_{nr} and a velocity scale v_{nr} related to the unresolved scales of motion. Using the turbulent kinetic energy equation and the equilibrium assumption it can be expressed as:

$$\nu_t = l_{nr} * v_{nr} = c^2 \Delta^2 |\overline{S_{ij}}| \tag{3.10}$$

where c is a constant, Δ is a characteristic length scale related to the cell dimension and $|\overline{S_{ij}}|$ is the contraction of the resolved strain rate tensor. This model has some drawbacks, the constant must be determined *a priori*, it is purely dissipative, it considers the stress aligned with the strain rate tensor, which in general is not true. On the other hand it is easy to implement and efficient in conjunction with wall modeling.

However its use just off the wall in a coarse grid is not correct and a different approach must be used, as discussed before.

An other issue is that the Smagorinsky model close to the wall does not work properly also because of the presence of the Immersed Boundary. If we use a Smagorinsky model at the IB points incorrect length and velocity scales are computed.

If one considers eq. (3.10), the stencil to compute $\overline{S_{ij}}$ takes into account also the computational nodes of the solid phase. It means that a non zero velocity is imposed at the Immersed Boundary surface (refer to Fig. 3.1a).

Besides the length scale of the model is related to the cell dimension by Δ but because of the presence of the immersed surface the real cell dimension can be smaller or larger depending on the fraction occupied by the body, see Fig. 3.1b.

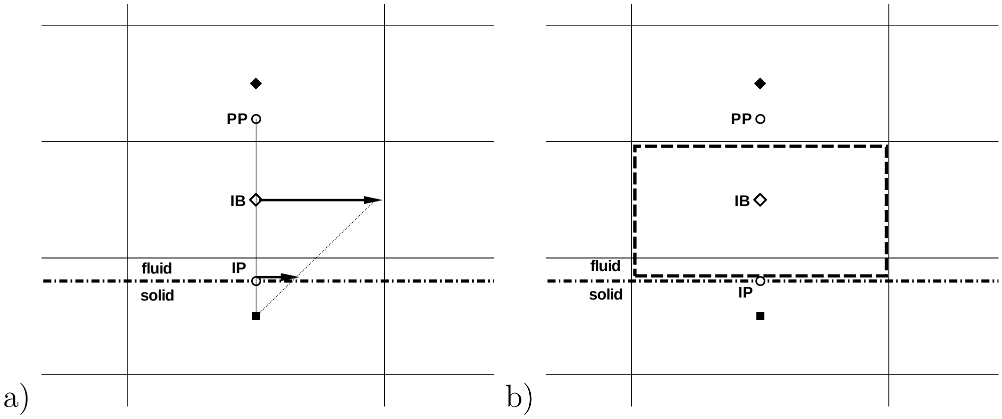

Figure 3.1: Square: solid nodes; Diamond: fluid nodes; Empty diamond: IB nodes; Continuous line: grid cell; line dot: immersed boundary surface. a) The strain rate tensor is computed also considering the solid nodes. b) Real cell dimension because of the presence of an Immersed Boundary.

To overcome these problems an alternative procedure is proposed to determine the eddy viscosity at the IB nodes.

Consider a shear flow of the type $u(y)$ with y the wall-normal direction. A Reynolds decomposition for the velocity is taken with u the mean velocity and u' the fluctuations. An instantaneous velocity field is of the type: $u + u'$, v', w' respectively for the streamwise wall-normal and spanwise direction. The quantity $-\rho < u'v' >$ represents the transfer rate of streamwise-momentum in the wall-normal direction through the turbulent fluctuations. Now using eq. (3.9) the shear stress can be expressed as:

$$\frac{- < u'v' >}{\rho} = \frac{\tau_{xy}}{\rho} = 2\nu_t \frac{1}{2}\left(\frac{du}{dy} + \frac{dv}{dx}\right) = \nu_t \frac{du}{dy} \tag{3.11}$$

The relation (3.11) is used to compute the eddy viscosity at IB points, using the tangential velocity V:

$$\nu_{t,IB} = \frac{\frac{\tau_{xy,IB}}{\rho}}{\frac{dV_{IB}}{dy}} \tag{3.12}$$

The velocity derivative in the wall-normal direction is determined through eq. (3.1):

$$\frac{dV}{dy} = \frac{u_\tau}{ky} \tag{3.13}$$

Using τ_w in eq. (3.12) instead of $\tau_{xy,IB}$, an error is introduced, but the procedure is simplified and the eddy viscosity can be computed as:

$$\nu_{t,IB} = cku_\tau d_{IB} \tag{3.14}$$

where c is a constant. Analytically this constant would be equal to 1. Different values can be necessary depending where the diffusive term is computed in a non staggered grid. Further investigations on it are ongoing. Equations (3.14) was also used by Cabot [10] and [11] in the contest of wall modeling with the boundary layer equation, here the constant c becomes a damping coefficient.

Using eq. (3.14) the problem related to the characteristics scales due to the Immersed Boundary presence is overcome. The eddy viscosity is strongly related to a length scale and a velocity scale characteristic of the physics of the problem close to the wall.

For clearness the overall procedure close to the wall is summarized:

- Compute the eddy viscosity at IB with eq. (3.14), using the wall stress computed at the previous time step.
- Compute the tangential velocity at PP from instantaneous values of the resolved flow.
- Compute the wall stress from the values at PP using eq. (3.1).
- Compute the tangential velocity at IB using eq. (3.8).
- Reconstruct the Cartesian velocity components for the points IB.

To verify this procedure several tests are made on a turbulent channel flow with an imposed pressure gradient. The configuration used and the results are shown in the next section.

3.3 Results

A test for the proposed wall model is performed on a plane channel flow. The flow is driven by a constant pressure gradient on the streamwise direction $d(p/\rho u_\tau^2)/dx_n = 1$, where x_n is the streamwise coordinate made non dimensional with half channel height δ. Two different Reynolds numbers are used: $Re_\tau = 1000$ and $Re_\tau = 2000$ based on half channel height and on the friction velocity u_τ. The results are compared with data from Del Alamo et al., (2004) and Hoyas and Jiménez, (2006).

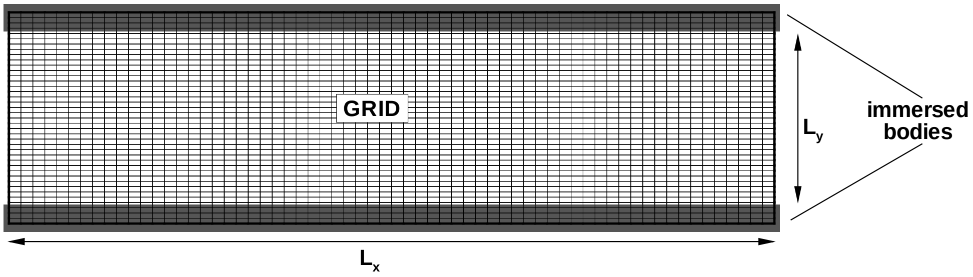

Figure 3.2: Plane channel flow configuration with immersed boundary.

The channel flow configuration is depicted in Fig. 3.2. The walls are modeled using the immersed boundary method.

Different computational grids are used and they are summarized in table 3.1. In the table l_x l_y l_z are the grid dimensions in streamwise, wall-normal and spanwise directions respectively. These are discretized with n_x n_y and n_z computational nodes.

We work on a non staggered grid and the position of the Immersed Boundaries are chosen to be not coincident with the computational nodes. In particular two configurations, namely A and B, with different position of the Immersed Boundary with respect to the grid are tested as in Figs. 3.3a and 3.3b. In configuration A the Immersed Boundary coincides with the grid cell boundary, while in configuration B the Immersed Boundary is very close to the solid nodes.

In both cases the distance L_y between the upper and lower boundaries is constant and equal to 2δ. This value is lower than the height of the computational domain l_y, so some computational nodes will fall in the solid phase. In table 3.1 $n_{fluid,y}$ represents the total grid nodes along the wall-normal direction in the fluid phase, so this is the actual discretization of the wall-normal direction.

Two resolutions are chosen in the wall-normal direction to test the grid independence of the proposed model. The fine grid uses 40 cells while the coarse one uses 20 cells.

The configuration A and B are chosen in order to test the eddy viscosity computation at the IB nodes.

grid	***configuration***	$\boldsymbol{l_x}$	$\boldsymbol{l_y}$	$\boldsymbol{l_z}$	$\boldsymbol{n_x}$	$\boldsymbol{n_y}$	$\boldsymbol{n_z}$	$\boldsymbol{n_{fluid,y}}$
***A*20**	A	$5/2\pi$	2.222	$\pi/2$	64	20	32	18
***A*40**	A	$5/2\pi$	2.222	$\pi/2$	64	40	32	36
***B*20**	B	$5/2\pi$	2.168	$\pi/2$	64	20	32	18
***B*40**	B	$5/2\pi$	2.168	$\pi/2$	64	40	32	36

Table 3.1: Discretization employed for the channel flow.

The simulations performed using these grids are shown in table 3.2. We perform two classes of simulations, namely L and LE. Simulations of type L use the logarithmic profile (see eq. (3.8)) between the PP and the IB nodes and the eddy viscosity at the IB nodes is computed with the Smagorinsky model as in the rest of the channel. In particular simulation $L1$ is performed on all the grids at $Re_\tau = 1000$. The LE simulations use the logarithmic profile as L but the eddy viscosity at IB nodes is computed with eq. (3.14). For LE we use all the grid types, in particular $LE1$ is simulated at $Re_\tau = 1000$ while $LE2$ is at $Re_\tau = 2000$. The distance of the IB points from the wall in wall units is reported as d_1^+. For all the possible configurations the IB points stay on the logarithmic profile. For the grid $A40$ at $Re_\tau = 1000$ we are on the lower limit of the logarithmic profile. The Smagorinsky model has a constant $c = 0.065$.

cases	***grid***	$\boldsymbol{Re_\tau}$	$\boldsymbol{d_1^+}$	***wall model***	$\boldsymbol{\nu_t}$ ***model***
$\boldsymbol{L1 - A20}$	A20	1000	55	*log*	SM
$\boldsymbol{L1 - A40}$	A40	1000	28	*log*	SM
$\boldsymbol{L1 - B20}$	B20	1000	78	*log*	SM
$\boldsymbol{L1 - B40}$	B40	1000	51	*log*	SM
$\boldsymbol{LE1 - A20}$	A20	1000	55	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE1 - A40}$	A40	1000	28	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE1 - B20}$	B20	1000	78	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE1 - B40}$	B40	1000	51	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE2 - A20}$	A20	2000	110	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE2 - A40}$	A40	2000	56	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE2 - B20}$	B20	2000	156	*log*	SM+$eq.$(3.14)
$\boldsymbol{LE2 - B40}$	B40	2000	102	*log*	SM+$eq.$(3.14)

Table 3.2: Simulations employed for the channel flow.

Figure 3.4 shows the velocity profile along the wall-normal direction for the streamwise velocity component u. It can be seen that the velocity profile is

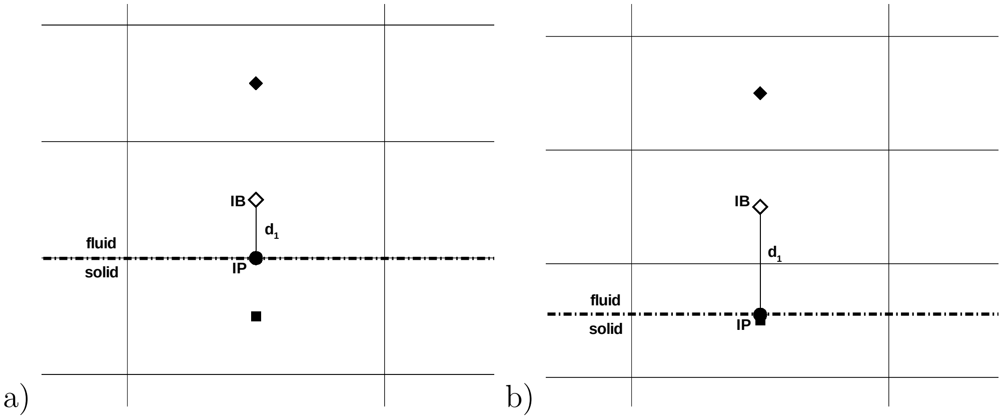

Figure 3.3: a) Configuration A for the channel flow with IBM. b) Configuration B for the channel flow with IBM. For both the figures square: solid nodes; diamond: fluid nodes; empty diamond: IB nodes; circle: IP nodes; continuous line: grid cell; line-dot: immersed boundary surface.

far from being logarithmic. The first two points outside the wall lie on the log profile as expected because of the velocity scaling that is applied to the IB points. The figure shows that this velocity scaling is not sufficient to obtain good results.

In Figs. 3.5 and 3.6 the velocity profiles for simulations LE1 are reported for grid type A and B respectively. In this case the eddy viscosity at the IB points is corrected as proposed before, using the friction velocity and the distance of the IB points from the Immersed Boundary as characteristic scales of the model. It can be seen that due to the eddy viscosity correction the results are in agreement with the reference data. All the profiles collapse on the logarithmic one, also with very coarse grids like $A20$ and $B20$ with only 20 points in the wall-normal direction. In this way the results appear to be grid independent.

Simulations LE2 are performed at $Re_\tau = 2000$. The results are shown in Figs. 3.7 and 3.8. Here the velocity profiles for simulations LE2 are reported for grid type A and B respectively. In this case as well the results are in agreement with the reference data.

For all the simulations shown, the shift of the third computational node from the logarithmic profile corresponds to the transition zone between the $RANS - area$ solution and the $LES - area$ solution.

The turbulent fluctuations and the Reynolds stress $< u'v' >$ are shown in Figs. 3.9 and 3.10. The figures refer to the grid $A40$ and $B40$ respectively, in the figures the values of the IB points are not reported because meaningless, in fact the velocities at these points are derived from the interior resolved flow using eq. (3.8). So their values are always lower than the closest resolved

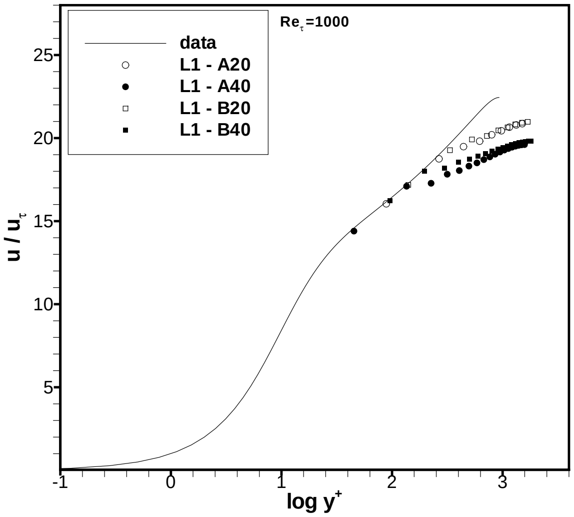

Figure 3.4: Simulations L1: streamwise velocity for a turbulent channel flow at $Re_\tau = 1000$.

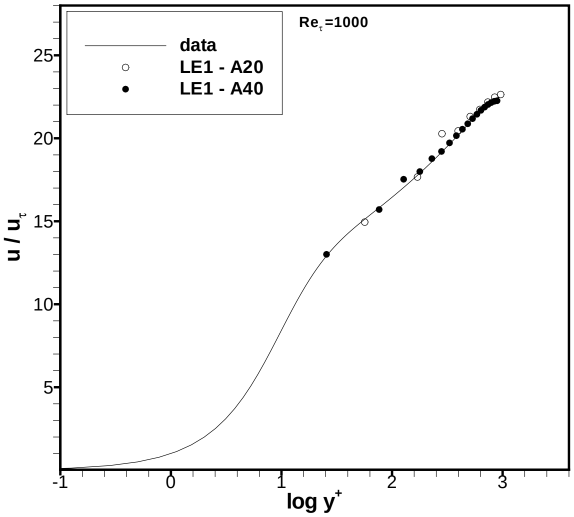

Figure 3.5: Simulations LE1 grid A: streamwise velocity for a turbulent channel flow at $Re_\tau = 1000$ on grid type A with 20 and 40 points on the wall-normal direction.

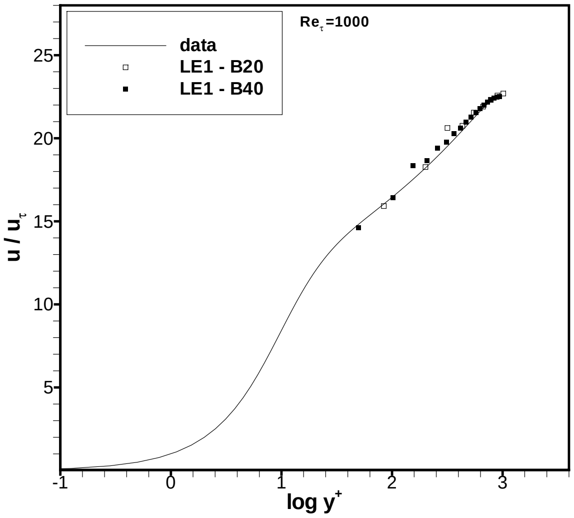

Figure 3.6: Simulations LE1 grid B: streamwise velocity for a turbulent channel flow at $Re_\tau = 1000$ on grid type B with 20 and 40 points on the wall-normal direction.

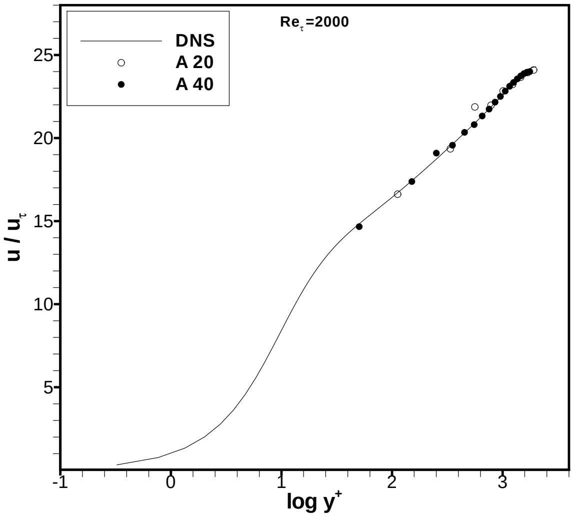

Figure 3.7: Simulations LE2 grid A: streamwise velocity for a turbulent channel flow at $Re_\tau = 2000$ on grid type A with 20 and 40 points on the wall-normal direction.

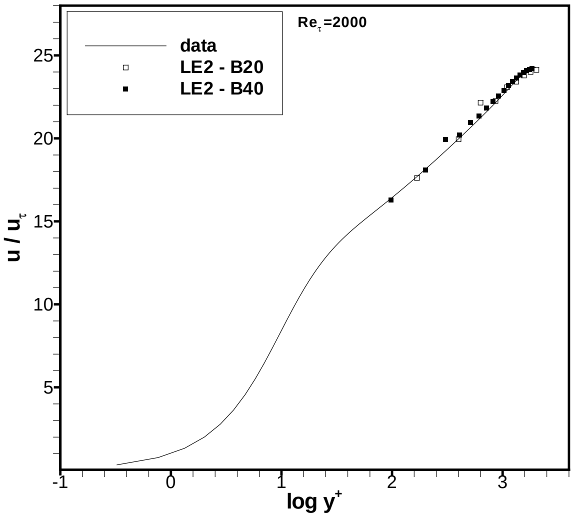

Figure 3.8: Simulations LE2 grid B: streamwise velocity for a turbulent channel flow at $Re_\tau = 2000$ on grid type B with 20 and 40 points on the wall-normal direction.

nodes. This is in contrast with the correct physical behavior, just out the viscous sub-layer higher fluctuations are expected. On the other hand in the wall modeling here proposed the boundary conditions are applied between the IB points and the interior resolved flow and not between the solid phase and the IB points.

In general because of the grid resolution the results are poor close to the wall, while they are quite good far from it. Conversely the spanwise fluctuation seems well reproduced everywhere. In particular the low fluctuations level for the v_{rms} determines a larger *unphysical* viscous layer, as it can be observed from the Reynolds stress. Besides the peak for the u_{rms} is quite far from the wall. In the end the streamwise and the wall-normal fluctuations near the wall behave as a simulation at a lower Reynolds number. An oscillation of the Reynolds stress can be observed at the first four points. The general poor behavior of the second order statistics near the wall can be related to the formation of unphysical structures and to a not correct resolution of the horizontal scales near the wall.

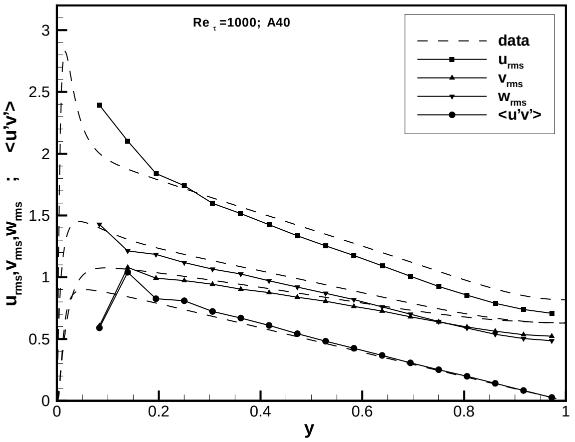

Figure 3.9: Simulations LE1 grid A40: *rms* quantities and Reynolds stress $< u'v' >$.

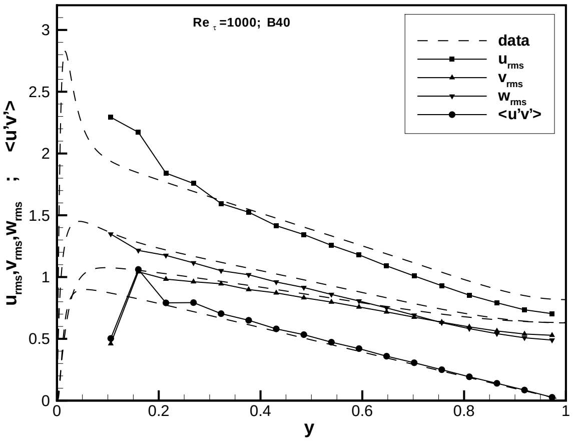

Figure 3.10: Simulations LE1 grid B40: *rms* quantities and Reynolds stress $< u'v' >$.

Chapter 4

A SGS model for sea coastal flow

In this chapter a SGS model for environmental hydrodynamics applications is developed. One of the most important aspects, when facing with sea coastal domains or lakes, is that there are typically two length scales, one for the horizontal dimensions (x and z or 1 and 3) of order of kilometers and one for the vertical direction (y or 2) of orders of ten meters. This difference (about two order of magnitude) creates problems when a characteristic length scale is required. In fact once the domain is discretized it is usual to have $pancake - type$ anisotropic cells, with an aspect ratio between the horizontal and the vertical direction of $10 : 1$. This can introduce inaccuracy when using classical subgrid stresses. In LES it is common to take, for cells with unequal sides, the Deardroff equivalent length scale $\Delta_{eq} = (\Delta_1 \Delta_2 \Delta_3)^{1/3}$, where $\Delta_{1,2,3}$ represent the cell sides. This is a good choice for weakly anisotropic grids or for grids where anisotropy is present close to solid boundaries, like in wall-bounded turbulence, but in the case of *cigar* cells or *sheet* cells, as in coastal flow, this determines anisotropic filtering on isotropic turbulence. This leads to inaccuracy in the evaluation of the turbulence statistics, inaccuracy which increases with anisotropy [32].

Scotti et al., (1993) investigated on the features of the Smagorinsky model with anisotropic grid in the case of inhomogeneous flow. They defined a single length scale obtaining a model close to that for isotropic case. Zahrai et al. (1995) proposed a model with different length scales to face this type of problem. To work on LES in sea coastal area with strong anisotropic grid, a modification on the Smagorinsky model similar to Zahrai et al. (1995) is proposed. These types of model are justified only for highly anisotropic filtering cells where the use of an unique characteristic length is no longer pertinent [59].

4.1 The mathematical model

The equations set is that shown in chapter 1 for LES. To model the SGS an eddy viscosity model is used, specifically as in chapter 3 a Smagorinsky model is considered. Literature studies have shown that the well established dynamic model is not suited for large-scale flows (see [9]). This is attributed to lacking of scale invariance of the SGS and subtest stresses in applicative large scale flows. Moreover, the explicit filtering operation required by the dynamic evaluation of the constant can be problematic when working with Immersed Boundaries. An alternative is to move back to the Smagorinsky model, which works well in conjunction with wall-layer models and Immersed Boundaries as shown before. Its formulation has been already introduced in chapter 3 in the contest of wall modeling. In this type of model the SGS is expressed as:

$$\tau_{ij}/\rho = -2\nu_t \overline{S}_{ij} \tag{4.1}$$

The eddy viscosity is evaluated as the product of a length scale $C\Delta$, proportional to the grid size, and a velocity scale $C\Delta|\overline{S}_{ij}|$, with C a constant (see eq. (3.10)).

The requirement of a single length scale in strongly anisotropic grid constitutes a problem, which can be overcome considering directional eddy viscosities, for the vertical and for the horizontal direction respectively. This approach is a standard technique used in large-scale ocean models. So far, to the best of our knowledge application in the LES contest has never been done. Two eddy viscosities $\nu_{t,h}$ and $\nu_{t,v}$ are commonly used in geophysical fluid dynamic [51], h denotes the horizontal direction and v the vertical one. Then the diffusive terms for the Navier-Stokes equations in a Cartesian framework read as:

$$D_i = \frac{\partial}{\partial x_1}\nu_h \frac{\partial \bar{u}_i}{\partial x_1} + \frac{\partial}{\partial x_2}\nu_v \frac{\partial \bar{u}_i}{\partial x_2} + \frac{\partial}{\partial x_3}\nu_h \frac{\partial \bar{u}_i}{\partial x_3} \tag{4.2}$$

where $\nu_h = \nu + \nu_{t,h}$ and $\nu_v = \nu + \nu_{t,v}$. Although widely used, this formulation is not mathematically consistent. It takes into account just deformation and not rotation to represent the stress. This can be done only assuming a linear proportionality with $\bar{S}_{ij}$, that is not true if we introduce directional eddy viscosities. A correct tensorial analysis brings to three coefficients for the eddy viscosity ([33], [44]): $\nu_{11} = \nu_{13} = \nu_{33}$, $\nu_{12} = \nu_{23}$ and ν_{22} with $\nu_{ij} = \nu_{ji}$. Then using a Smagorinsky model they can be expressed as:

$$\nu_{11} = (CL_h)^2|\overline{S}_h| \qquad \nu_{12} = (CL_v)^2|\overline{S}_v| \qquad \nu_{22} = (CL_v)^2|\overline{S}_r| \tag{4.3}$$

with L_h and L_v proper length scales for the cell in horizontal and vertical direction. The strain rate tensor is decomposed as follows:

$$|\overline{S}_h| = \sqrt{2(\overline{S}_{11}^2 + \overline{S}_{33}^2 + 2\overline{S}_{13}^2)} \tag{4.4}$$

$$|\overline{S}_v| = \sqrt{4\overline{S}_{12}^2 + 4\overline{S}_{23}^2} \tag{4.5}$$

$$|\overline{S}_r| = \sqrt{2\overline{S}_{22}^2} \tag{4.6}$$

If we consider $\nu_{11} = \nu_{t,h}$ and $\nu_{12} = \nu_{t,v}$ the diffusive terms for the horizontal plane can be written as in eq. (4.2). For the vertical direction 2 it becomes:

$$D_2 = \frac{\partial}{\partial x_1}\nu_v \frac{\partial \bar{u_2}}{\partial {x_1}^2} + \frac{\partial}{\partial x_2}\nu_v \frac{\partial \bar{u_2}}{\partial {x_2}^2} + \frac{\partial}{\partial x_3}\nu_r \frac{\partial \bar{u_2}}{\partial {x_3}^2} \tag{4.7}$$

where $\nu_r = \nu + \nu_{t,r}$ and $\nu_{t,r} = \nu_{11} - 2\nu_{12} + 2\nu_{22}$. A dimensional analysis shows that $\nu_{t,r}$ is of the same order of $\nu_{t,h}$. The coefficients of the model need calibration and this is still an open issue for lacking of proper test cases.

4.2 Adaptation to a curvilinear framework

In section 4.1 it has been shown that the introduction of directional eddy viscosities requires three terms to close the equations. At the moment, also if not properly correct, in the Navier-Stokes solver used to perform simulation on sea coastal flow, we have used the model with two eddy viscosity as in eq. (4.2). So:

$$\nu_{t,h} = (CL_h)^2|\overline{S}_h| \qquad \nu_{t,v} = (CL_v)^2|\overline{S}_v| \tag{4.8}$$

with $|\overline{S}_h|$ as in (4.4) and $|\overline{S}_v|$:

$$|\overline{S}_v| = \sqrt{4\overline{S}_{12}^2 + 2\overline{S}_{22}^2 + 4\overline{S}_{23}^2} \tag{4.9}$$

As discussed, this is a common practice in oceanography. The formulation proposed in the previous section is for a Cartesian frame of reference, in a curvilinear one it is necessary to see how directional viscosities combine with the mesh skewness tensor G^{mn}. Besides some simplifications are proposed.

Considering for example the diffusive term in direction 1 in Cartesian form, it can be written from eq. (4.2):

$$D_1 = \frac{\partial}{\partial x_1}\nu_h \frac{\partial \bar{u_1}}{\partial x_1} + \frac{\partial}{\partial x_2}\nu_v \frac{\partial \bar{u_1}}{\partial x_2} + \frac{\partial}{\partial x_3}\nu_h \frac{\partial \bar{u_1}}{\partial x_3} \tag{4.10}$$

The above diffusive term in curvilinear form becomes:

$$
\begin{aligned}
&\frac{\partial}{\partial \xi}\overbrace{(\nu_h \xi_x^2 + \nu_v \xi_y^2 + \nu_h \xi_z^2)}^{\nu G^{11}} u_\xi \\
&+\frac{\partial}{\partial \eta}\overbrace{(\nu_h \eta_x^2 + \nu_v \eta_y^2 + \nu_h \eta_z^2)}^{\nu G^{22}} u_\eta \\
&+\frac{\partial}{\partial \zeta}\overbrace{(\nu_h \zeta_x^2 + \nu_v \zeta_y^2 + \nu_h \zeta_z^2)}^{\nu G^{33}} u_\zeta \\
&+\frac{\partial}{\partial \xi}\overbrace{(\nu_h \xi_x \zeta_x + \nu_v \xi_y \zeta_y + \nu_h \xi_z \zeta_z)}^{\nu G^{13}} u_\zeta \\
&+\frac{\partial}{\partial \xi}\overbrace{(\nu_h \xi_x \eta_x + \nu_v \xi_y \eta_y + \nu_h \xi_z \eta_z)}^{\nu G^{12}} u_\eta \\
&+\frac{\partial}{\partial \eta}\overbrace{(\nu_h \xi_x \eta_x + \nu_v \xi_y \eta_y + \nu_h \xi_z \eta_z)}^{\nu G^{21}} u_\xi \\
&+\frac{\partial}{\partial \eta}\overbrace{(\nu_h \eta_x \zeta_x + \nu_v \eta_y \zeta_y + \nu_h \eta_z \zeta_z)}^{\nu G^{23}} u_\zeta \\
&+\frac{\partial}{\partial \zeta}\overbrace{(\nu_h \xi_x \zeta_x + \nu_v \xi_y \zeta_y + \nu_h \xi_z \zeta_z)}^{\nu G^{31}} u_\xi \\
&+\frac{\partial}{\partial \zeta}\overbrace{(\nu_h \eta_x \zeta_x + \nu_v \eta_y \zeta_y + \nu_h \eta_z \zeta_z)}^{\nu G^{32}} u_\eta
\end{aligned}
\tag{4.11}
$$

here to simplify the notation we write $\partial x/\partial \xi = x_\xi$.

In the discretization of the physical domain in sea coastal flows, a common choice is to use the coordinate y parallel to η, since the slope of the bathymetry is in general small. This is not true for the other directions. However in case of very large local slope of the bathymetry, rather than excessively deformate the grid cell, it is convenient to take advantage of the IBM previously discussed, maintaining a constant grid and modeling part of the bottom through an Immersed Boundary.

The parallelism between y and η brings to some simplifications for the composition of the metric terms with the viscosities (see the term (4.11)). Considering the plane zy and referring to Fig. 4.1:

$$z_\eta = x_\eta = 0 \tag{4.12}$$

since the derivatives ξ_y and ζ_y are written [23] as:

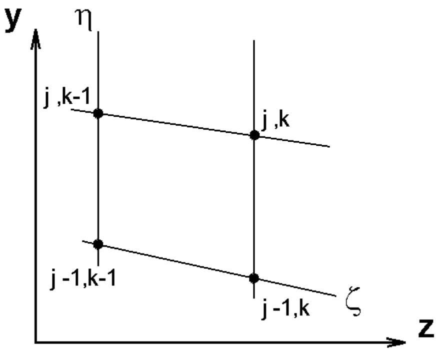

Figure 4.1: Curvilinear grid with η aligned with the vertical coordinate y.

$$\xi_y = \frac{x_\zeta z_\eta - x_\eta z_\zeta}{J^{-1}} \tag{4.13}$$

$$\zeta_y = \frac{x_\eta z_\xi - x_\xi z_\eta}{J^{-1}} \tag{4.14}$$

the alignment between y and η leads to:

$$\xi_y = \zeta_y = 0. \tag{4.15}$$

From eq. (4.15) the terms of the metric tensor G^{mn} multiplied by the vertical viscosities ν_v reduce to just the term G^{22}:

$$\frac{\partial}{\partial \eta}(\nu_h \eta_x^2 + \nu_v \eta_y^2 + \nu_h \eta_z^2) u_\eta \tag{4.16}$$

Collecting ν_v the term becomes:

$$\frac{\partial}{\partial \eta}\nu_v \left(\frac{\nu_h}{\nu_v}\eta_x^2 + \eta_y^2 + \frac{\nu_h}{\nu_v}\eta_z^2 \right) u_\eta \simeq \frac{\partial}{\partial \eta}\nu_v \eta_y^2 u_\eta \tag{4.17}$$

The expression (4.17) holds under the hypothesis:

$$\frac{\nu_h}{\nu_v}\eta_x^2 \ll \eta_y^2 \quad or \quad \frac{\nu_h}{\nu_v}\frac{\eta_x^2}{\eta_y^2} \ll 1 \tag{4.18}$$

on the plane xy (the same arguments apply to the plane zy).

The correctness of eq. (4.18) must be proved. The left hand side term is composed of two elements. The first term is the ratio between the eddy viscosities. The second term is the ratio between two metric terms, namely a term depending on the bottom slope.

The term η_x^2/η_y^2 is first considered (refer to Fig. 4.2a). For sake of simplicity we consider a 2D case, although the discussion holds for general 3D case. The

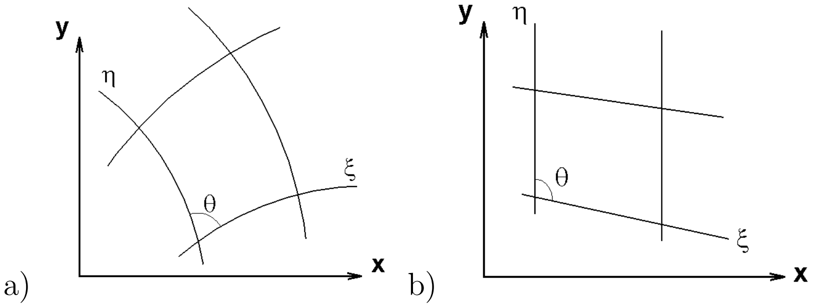

Figure 4.2: a) General curvilinear grid. b) Curvilinear grid in coastal flow.

angle θ represents the local distortion of the grid. It can be computed, see [23], from the metric tensor:

$$cos\theta = \frac{G^{12}}{(G^{11}G^{22})^{1/2}} = \frac{\xi_x\eta_x + \eta_y\xi_y}{[(\xi_x^2+\xi_y^2)(\eta_x^2+\eta_y^2)]^{1/2}} \tag{4.19}$$

Considering the case as in Fig. 4.2b and eq. (4.15), eq. (4.19) reduces to:

$$cos\theta = \frac{\xi_x\eta_x}{[\xi_x^2(\eta_x^2+\eta_y^2)]^{1/2}} = \frac{\eta_x}{(\eta_x^2+\eta_y^2)^{1/2}} \tag{4.20}$$

and after some trigonometric manipulation it can been obtained:

$$\frac{\eta_x^2}{\eta_y^2} = \frac{cos^2\theta}{1-cos^2\theta} = \frac{1}{tan^2\theta} \tag{4.21}$$

For weak bottom slope the ratio η_x^2/η_y^2 is very small.

This comes also from a simple dimensional analysis. As mentioned before, in a sea coastal domain the horizontal length scale L_H is of some order of magnitude larger than the vertical length scale L_V.

$$\frac{L_H}{L_V} \sim O(10^2) \tag{4.22}$$

This means that once the domain is discretized, *sheet like* cells are common and the cell ratio between L_h and L_v has order:

$$\frac{L_h}{L_v} \sim O(10) \tag{4.23}$$

From a dimensional analysis and considering (4.23), it follows:

$$\frac{\eta_x^2}{\eta_y^2} \sim \frac{(\Delta\eta/L_h)^2}{(\Delta\eta/L_v)^2} \sim \left(\frac{L_v}{L_h}\right)^2 \sim O(10^{-2}) \tag{4.24}$$

Now, in order to prove the correctness of eq. (4.18), it is necessary to show that the eddy viscosity ratio is of order $O(10)$. Consider two velocity scales U and V for the horizontal and the vertical direction. From continuity consideration and from the knowledge of marine circulation:

$$\frac{U}{V} \sim O(10) \tag{4.25}$$

A dimensional analysis of the strain rate tensor can be made. First consider the term $|S_v|$, looking at each element, neglecting the lower order terms and considering eqs. (4.25) and (4.23) it can be written

$$S_{12} \sim S_{32} \sim \frac{1}{2}\left(\frac{U}{L_v} + \frac{V}{L_h}\right) = \frac{1}{2}\frac{V}{L_v}\left(\frac{U}{V} + \frac{L_v}{L_h}\right) \sim \frac{1}{2}\frac{U}{L_v} \tag{4.26}$$

$$S_{22} \sim \frac{V}{L_v} \tag{4.27}$$

then:

$$|S_v| \sim \sqrt{2}\left[\left(\frac{U}{L_v}\right)^2 + \left(\frac{V}{L_v}\right)^2\right]^{1/2} = \sqrt{2}\frac{V}{L_v}\left[\left(\frac{U}{V}\right)^2 + 1\right]^{1/2} \sim \sqrt{2}\frac{U}{L_v} \tag{4.28}$$

For the term $|S_h|$:

$$S_{11} \sim S_{33} \sim S_{13} \sim \frac{U}{L_h} \tag{4.29}$$

so:

$$|S_h| \sim 2\sqrt{2}\frac{U}{L_h} \tag{4.30}$$

Then the ratio between the viscosities is:

$$\frac{\nu_h}{\nu_v} = \frac{L_h^2|S_h|}{L_v^2|S_v|} \sim \frac{L_h^2}{L_v^2}\frac{U2\sqrt{2}}{L_h}\frac{L_v}{U\sqrt{2}} = 2\frac{L_h}{L_v} \sim O(10) \tag{4.31}$$

and hence it follows that

$$\frac{\nu_h}{\nu_v}\frac{\eta_x^2}{\eta_y^2} \sim 2\frac{L_h}{L_v}\frac{L_v^2}{L_h^2} = 2\frac{L_v}{L_h} \sim O(10^-1) \ll 1 \tag{4.32}$$

Chapter 5

Dispersion of a vertical jet of buoyant particles in a stably stratified wind-driven Ekman layer

This chapter has been developed in a scientific cooperation with Ing. Valentina Stocca and dot. Roberto Inghilesi. The work was supported by ISPRA (ex APAT) and it has been published on the journal: *Heat and Fluid Flow*, [29].

An environmental problem that is becoming increasingly relevant nowadays is the prediction of dispersion of polluting particulate in a marine environment. Particles of buoyant fluid in a marine environment may be released for example when leakage occurs in submarine pipelines or other devices (oil spilling problems). In this kind of applications it is of significant interest the evaluation of the concentration of the released substance as well as the evaluation of the amount of particulate that reaches the free surface of the sea. In literature (see for example [58] for a general discussion) Eulerian models are commonly used, in which the plume of particles is described in an Eulerian way as a space-time distribution of their concentration. Although this approach is computationally inexpensive, it suffers from empiricism in particular when the concentration of the dispersed phase is small and it is mainly composed of an ensemble of separate particles traveling in the carrying fluid. Here it is studied the dispersion of a buoyant jet of particles released into a salty water basin. Note that since the time scale of diffusion of salinity is usually much larger than the buoyancy time scale, this analysis can also be applied to upwelling of fresh water particles in a salty water ambient.

In particular an archetypal problem is considered, representative of the upper part of the ocean forced by a constant wind stress and the release of the buoyant particles in a region below the free surface where turbulent mixing is negligible.

From a fundamental point of view, the top region of the sea can be treated as a wind driven Ekman layer, namely the boundary layer created by the tangential stress supplied by the action of the wind in a rotating environment. The Coriolis force causes a rotation of the velocity profiles and thus generates a component in the spanwise direction. In laminar conditions the penetration depth of the Ekman layer (the thickness on the boundary layer) is proportional to $\sqrt{2\nu/f_z}$ (the z-direction is vertical upward) where ν is the fluid viscosity and f_z is the Coriolis parameter. In the turbulent regime the penetration depth is found to be proportional to u_τ/f_z where $u_\tau = \sqrt{\tau_w/\rho_0}$ is the friction velocity associated to the free surface wind stress τ_w, and ρ_0 is the reference density of the water. A detailed discussion on the Ekman layer and its relevance in environmental fluid mechanics is in [56]. The main scope the present work is to understand how a cloud of buoyant particles released in the water column is dispersed in the wind-driven Ekman layer subject to different conditions of thermal stratification. To this aim an extension of the Maxey and Riley equation [43] for the particle motion is proposed, which takes into account the actual fluid density during the space-time evolution of the particle swarm. The study is performed numerically using a Lagrangian-Eulerian approach, which consists of moving Lagrangian particles in an Eulerian carrier phase.

The work is organized as follows: first a description of the problem investigated together with the mathematical formulation is done. Specifically the new form of the particle-motion equation is presented and discussed. Then the results of the simulations are discussed and concluding remarks are given in section 5.2.4.

5.1 The Problem Formulation

A Lagrangian-Eulerian approach is used, in which the dispersed phase is treated as an ensemble of Lagrangian particles moving in an Eulerian flow field. In order to evaluate the force field acting over the Lagrangian particles an interpolation of the Eulerian flow field onto the particle position is carried out. A detailed description of the mathematical method is here supplied.

5.1.1 The Eulerian field

A mid-latitude wind driven Ekman layer is considered, thus including both the vertical and the horizontal components of the rotation vector in the governing equations (see [14] and [60]). The relevant scales of the problem are: the already defined friction velocity u_τ, the time scale $T = 1/f_z$ associated to the Coriolis parameter and the penetration length $\delta = u_\tau/f_z$ which gives an estimation of the depth of the turbulent boundary layer. The stratification is considered as a variation of the density field with respect to a reference value ρ_0. Specifically the density field is $\rho_{tot,d}(x, y, z, t) = \rho_0 + \rho_d(x, y, z, t)$ with $\rho_d << \rho_0$ (hereafter the index d denotes dimensional quantities). In our numerical experiment the stratification comes from the imposition of a heat flux at the free surface, as in [68]; physically this corresponds to heating a fluid column by an incoming heat flux. Here we discuss two cases, respectively the case of neutral flow $Ri \rightarrow 0$ and the case of strongly stratified flow $Ri = 40$ where the Richardson number is $Ri = g/\rho_0 \; |d\rho/dz|_{fs}| \; \delta^2/u_\tau^2$. In the present case, the Richardson number is defined using a density scale $|d\rho/dz|_{fs}|\delta$ which is related to the free surface heat flux as follows $d\rho/dz|_{fs} = -\rho_0 \alpha dT/dz|_{fs}$, with α the thermal expansion coefficient. The wind stress acts in the direction south-north (x axis), the y-axis is directed from east to west and the z-axis is vertical upward. A mid-latitude case ($\theta = 45^o$ where θ is the latitude) is here considered: with the frame of reference herein used, the components of the rotation vector are the vertical one $f_z = 2\Omega_H sin\theta$ and the horizontal one $f_x = 2\Omega_H cos\theta$ where Ω_H is the earth rotation frequency. A typical full-scale value of $Re = u_\tau \delta/\nu$ is of the order of 7×10^5 considering the data of wind stress given in [56]. Although the approach we use, equivalent to that employed in the $Re \rightarrow \infty$ simulation of [76], can deal with applicative values of the Reynolds number, here we consider a moderate value of Reynolds number $Re = 10000$ (see Section 5.1.3). This value of Re is such to minimize Reynolds number effects on the dynamics of the flow field.

The equations of the Eulerian flow here considered are that of section 1.1, specifically the Cartesian form is taken, the equations are solved using LES. The non-dimensional filtered equations are:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{5.1}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_j \; \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} + \tag{5.2}$$

$$-\epsilon_{ijk} \frac{f_j}{|f_3|} \bar{u}_k - Ri \bar{\rho} \; \delta_{i3} - \frac{\partial \tau_{ij}}{\partial x_j}$$

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{u}_j \; \bar{\rho}}{\partial x_j} = \frac{1}{Re \; Pr} \frac{\partial^2 \bar{\rho}}{\partial x_j \partial x_j} - \frac{\partial \lambda_i}{\partial x_i} \tag{5.3}$$

In eqs. (5.1), (5.2) t is time made non dimensional with $1/f_z$, u_i is the velocity component in the i-direction made non dimensional with u_τ, x_i is the i-coordinate made non dimensional with δ, p is the pressure made non dimensional with $\rho_0 u_\tau^2 = \tau_w$, and ρ is the density made non-dimensional with the density scale $|\delta d\rho/dz|_{fs}|$. In this chapter directions 1,2,3 respectively correspond to x,y,z. The non-dimensional groups Re and Ri have been already defined, the Prandtl number is $Pr = \nu/k$ with k the thermal diffusivity of the medium. The quantities τ_{ij} and λ_i are respectively the subgrid-scale stresses and the SGS buoyancy fluxes. They were already defined in section 1.4 for a curvilinear framework, in the Cartesian form they read as $\tau_{ij} = \overline{u_i u_j} - \overline{u}_i\overline{u}_j$ and $\lambda_i = \overline{u_i\rho} - \overline{u}_i\overline{\rho}$.

Here a dynamic-mixed model is used for the SGS momentum fluxes, composed of a scale-similar part [8] and an eddy viscosity one:

$$\tau_{ij} = (\overline{\overline{u}_i\overline{u}_j} - \overline{\overline{u}}_i\overline{\overline{u}}_j) - 2C\overline{\Delta}^2|\overline{S}|\overline{S}_{ij} \tag{5.4}$$

In eq. (5.4) the first parenthesis on the right hand side is the scale similar part, C is the constant, $\bar{\Delta}$ is the filter width, $\overline{S}_{ij}$ is the resolved strain rate tensor and $|\overline{S}|$ is its contraction. The SGS density fluxes are parametrized using a dynamic eddy diffusivity model:

$$\lambda_i = -C_\rho\overline{\Delta}^2|S|\frac{\partial\bar{\rho}}{\partial x_i} \tag{5.5}$$

In eqs. (5.4) and (5.5) the constants C and C_ρ are calculated dynamically, as described with details in [5]. Specifically the authors have extensively shown that such a model is able to adjust automatically to the actual flow conditions and to predict the superlinear behavior of the SGS Prandtl number ($Pr_{SGS} = \nu_{sgs}/k_{sgs}$) which, as well known in literature, occurs in strongly stably stratified turbulence.

5.1.2 The Lagrangian phase

The dispersed phase is treated in a Lagrangian way using a simplified form of the Maxey and Riley equation (see [43]). Following [2] the Stokes drag is considered in the particle motion equation together with the gravity term. In the present work we consider the effect of the variation of density of the Eulerian field in the particle motion equation. A description of the proposed modification is given here.

The dimensional equation of the particle motion is:

$$\begin{aligned}\frac{dV_{p,i}}{dt_d} &= -\left(\frac{\rho_p - \rho_{tot,d}}{\rho_p}\right)g\delta_{i3} + \\ &\quad \frac{3}{4\,D_p}C_D\,|U_{p,i} - V_{p,i}|\,(U_{p,i} - V_{p,i}) - \\ &\quad 2\epsilon_{ijk}\Omega_j V_{p,k}\end{aligned} \tag{5.6}$$

In eq. (5.6) t_d is the dimensional time, $V_{p,i}$ and $U_{p,i}$ are respectively are the i-component of the dimensional particle velocity and of the dimensional fluid velocity at the particle location, ρ_p and D_p are respectively the density and the diameter of the particle, $C_D = 24/Re_p(1 + 0.15Re_p^{0.687})$ where $Re_p =$

$|U_{p,i} - V_{p,i}| \, D_p/\nu$ is the particle Reynolds number. Replacing the total density $\rho_{tot,d}$ with $\rho_0 + \rho_d$, and making eq. (5.6) non dimensional we obtain:

$$\begin{aligned}\frac{dv_{p,i}}{dt} &= \left(1 - \frac{1}{\Delta\rho}\right) \frac{1}{Fr^2} \delta_{i3} - \frac{Ri}{\Delta\rho} \rho \delta_{i3} + \\ & \frac{3}{4\, d_p} C_D \left|u_{p,i} - v_{p,i}\right| \left(u_{p,i} - v_{p,i}\right) - \\ & \epsilon_{ijk} \frac{\Omega_j}{\Omega_3} v_{p,k} \qquad (5.7)\end{aligned}$$

In eq. (5.7) $\Delta\rho = \rho_p/\rho_0$, $v_{p,i} = V_{p,i}/u_\tau$, $u_{p,i} = U_{p,i}/u_\tau$, $d_p = D_p/\delta$ and the buoyancy contribution appears split into two terms: the first one is the usual contribution proportional to the Froude number $Fr = u_\tau/(g\ \delta)^{0.5}$; the second one takes into account the density variation in the Eulerian field, and is proportional to the Richardson number. Note that $Fr^2 Ri = \rho_s/\rho_0$ where ρ_s is a typical density scale of the problem under investigation, *i.e.* $\delta |d\rho/dz|_{fs}|$ in this case, thus $Fr^2 Ri << 1$ under the Boussinesq approximation. The second term (II) on the RHS of eq. (5.7) gets important when it is comparable with the first term (I) on the RHS. It can be easily shown that $II/I = (\rho_s/\rho_0) \, / \, ((\rho_p - \rho_0)/\rho_0)$, thus II influences the particle motion when the particle density is very similar to the reference density of the carrying fluid. This is the case, for instance, of particles of fresh water released in a salty water environment.

Finally the position of the particles is advanced in time as follows:

$$\frac{dx_{p,i}}{dt} = v_{p,i} \qquad (5.8)$$

with $x_{p,i}$ the i-component of the particle position made non dimensional with δ.

5.1.3 The numerical method

The Lagrangian-Eulerian model herein employed works as follows: The LES field is first obtained through integration of the filtered equations (5.1)-(5.3). Such integration is carried out using the fractional-step described in 1.3.

Since the LES field does not contain the energy and vorticity contribution associated to the unresolved (small) scales of the motion, a SGS model for the motion of the Lagrangian particles is required. Note that in *resolved* LES , it was shown that the contribution of the small unresolved scales of motion on particle dispersion is small and the error in the dispersion coefficients is proportional to the dissipation rate of the SGS turbulent scales [3]. On the other hand, when dealing with large values of the Reynolds number (that require the use of a wall-layer model for the evaluation of the Eulerian field) neglecting the contribution of the small scales to the Lagrangian motion can lead to severe underestimation of the dispersion of the particle swarm. Since it is simulated a large Reynolds number flow, in order to overcome this problem the LES field is then reconstructed with the aim to recover the energy contribution associated to the unresolved scales. The approximate deconvolution technique, recently applied by [36] and [65] to the problem of particle dispersion, is used. Specifically, the following approximate deconvolution:

$$u_{r,i} = \overline{u}_i + \left(\overline{u}_i - \overline{\overline{u}}_i\right) + \left(\overline{u}_i - 2\overline{\overline{u}}_i + \overline{\overline{\overline{u}}}_i\right) + ... \qquad (5.9)$$

when used in conjunction with an equivalent model for the SGS turbulent stresses was proven to be able to recover the SGS turbulent contribution to the particle motion. In eq. (5.9) $u_{r,i}$ is the re-constructed i-component of the velocity field. Successive tests have shown that a first order re-construction (first two terms on the right hand side, RHS, of eq. (5.9)) gives accurate results when used in conjunction with a dynamic mixed model [41]. The reconstructed field is later on interpolated onto the particle position using a technique recently developed by [42]. This technique uses a Taylor expansion around the grid point (N) closest to the particle, to interpolate the Eulerian field onto the particle position (P):

$$u_{P,i} = u_{N,i} + \frac{\partial u_i}{\partial x_i}\left(x_{P,i} - x_{N,i}\right) + O(\Delta x^2) \qquad (5.10)$$

This technique is very simple and inexpensive, and allows obtaining second-order space-accuracy, as that achieved in the solution of the Eulerian field. The interpolated velocity is finally considered in the particle motion equation (eq. (5.7)) and the particle position is thus advanced in time using a second-order accurate Adams-Bashforth technique.

As regards the Eulerian field, the mathematical model herein employed has been validated in a number of flow conditions (see among the others [4], [5], [61]). Results of validation tests for the wind-driven Ekman layer are discussed in section 5.2.1.

5.1.4 Computational parameters

Following [76] the simulations are carried out over a rectangular box whose non-dimensional extensions are $L_x = 1$ and $L_y = 1$ in the horizontal directions, whereas $L_z = 1.5$. The domain size is large enough to reproduce the largest scales of motion both in the neutral case and in the stably stratified one. This occurs because stable stratification inhibits and destroys the largest scales of motion. The computational grid is uniform and has $64 \times 64 \times 200$ cells respectively in the x, y and z directions. The grid spacing in wall units (made non dimensional with ν/u_τ) is $\Delta x^+ = \Delta y^+ = 156$ and $\Delta z^+ = 75$. Such a grid size is typical of LES with wall-layer models (see [7]) and is also suited for stably stratified flows when dynamic SGS models are employed, due to the ability of the model constants to adapt to the actual flow conditions. The molecular Prandtl number is chosen equal to 5 which corresponds to thermally stratified water. Periodic boundary conditions are imposed over the horizontal directions, whereas at the bottom boundary of the computational domain we set $\rho = u_3 = \partial u_1/\partial x_3 = \partial u_2/\partial x_3 = 0$. This choice is justified by the fact that the domain depth is much larger than the turbulent penetration length and modifications of the flow variables are not expected at the bottom boundary. The *a posteriori* analysis of the numerical results confirmed the effectiveness of such a choice. Two cases are considered, a case of neutral flow ($Ri \to 0$) and a case of stable stratified flow ($Ri = 40$). The latter corresponds to a case of strong stratification. In both cases, a statistically steady state is first obtained, and later on the particles are released. In the two cases analyzed, 54000 buoyant particles have been released, with density ratio $\Delta\rho = \rho_p/\rho_0 = 0.976$, corresponding to fresh water particles released in a salt water environment and radius equal to $10^{-3}\delta$. The vertical plume is released continuously in time, up to $t = 1.1$, from a disc of radius 0.05δ, located at 0.5δ below the free surface and containing 900 particles. The particles have non zero initial vertical velocity $v_{p,3} = u_\tau$. As regards the choice of the parameters for the dispersed phase, the vertical location of the ejection disc was chosen in a turbulence-free region, close to the free surface in order to reduce the time needed to the particles to rise up to the surface. The radius of the disc was such to spread the particles over a number of grid cells and the choice of the initial velocity was made for having a typical velocity scale of the flow field. However tests have demonstrated that the memory effect of the initial velocity is lost after a short time, and the physics ruling the rise of the particles is mainly associated to buoyancy effects. Finally the number of the particles was such to ensure that all the particles left the disc at the final time of the simulation.

A sketch of the problem herein discussed is in Fig. 5.1.

5.2 Results

First the results of the validation tests are discussed, successively we show the statistics of the Eulerian field and finally we show the results of the Lagrangian phase.

5.2.1 Validation tests

To the best of our knowledge data of the wind-driven Ekman layer at $Re = 10000$ are not available. The only results available for the problem under investigation are those of the polar ($f_x = f_y = 0$) $Re \to \infty$ simulation of [76].

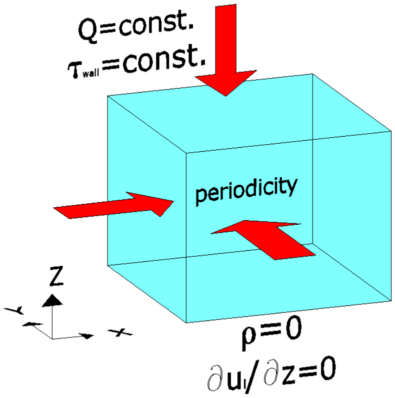

Figure 5.1: Schematic of the physical problem herein investigated. The quantity Q denotes the free surface heat flux.

A polar simulation at $Re = 10000$ has been run and first- and second-order statistics were compared with the reference data. Although the difference in the value of Re, the agreement between the results and the reference ones is pretty good, due to the fact that the Re number is large enough to minimize its own effect on the re-scaled velocity field. Figure 5.2a shows the comparison of the vertical profile of the mean horizontal velocity components. The notation $< . >$ denotes Reynolds averaged quantities. Small differences are detectable in the outer region, well below the free surface. Specifically, as expected, the comparison shows a larger penetration depth in the $Re \rightarrow \infty$ case with respect to the moderate Reynolds number case. This difference has to be attributed to the fact that a finite Re number implies a smaller penetration depth than the one observable at $Re = \infty$. In Fig. 5.2b the comparison of the rms of the three velocity components is shown. As for the mean velocity components, they are very close to each other at the upper surface while some differences are observable in the outer region. This is a Reynolds number effect well known in literature, namely the normal Reynolds stresses do not scale with the friction velocity far from the wall and, when they are made non-dimensional with wall variables they slightly increase with Re.

5.2.2 The Eulerian field

Here the results of the mid-latitude simulations are discussed in the two cases of stratification. Figure 5.3a shows the vertical profiles of the horizontal components of the mean velocity field in the two cases analyzed. In the neutral case the penetration length of the turbulent field is of the order of δ and, as expected, large gradients are present in the free surface region. The transversal velocity arising from the rotational motion is large and negative in the free surface region, indicating a rotation of the velocity vector toward the right

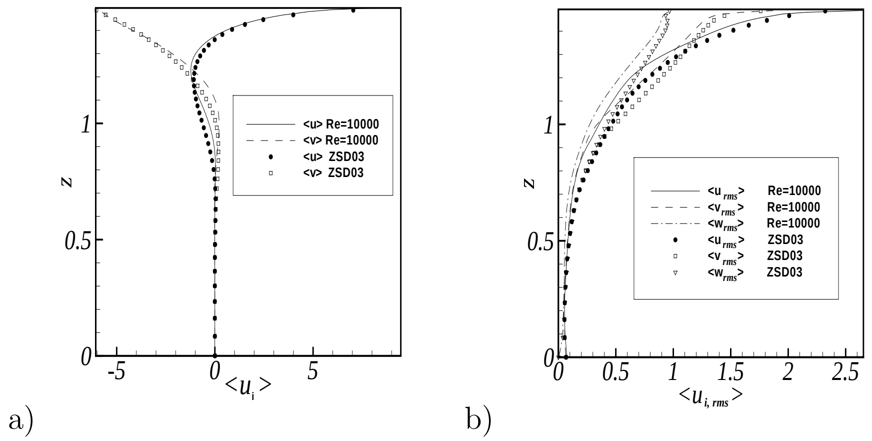

Figure 5.2: Comparison of a polar $Re = 10000$ simulation with the reference data of [76] at $Re \to \infty$: a) vertical profiles of the mean horizontal velocity components; b) vertical profile of the rms velocities.

direction as expected in the northern hemisphere. In the stratified case, the velocity profiles are dramatically affected by the presence of large incoming free surface heat flux. In the stratified case (Fig. 5.3a) it is observed a strong reduction of the penetration length of the boundary layer, a small increase of the streamwise velocity at the free surface and a noticeable increase of the spanwise component. Moreover, large velocity gradients are observed beneath the free surface. As a result, the angle between the wind stress and the velocity vector increases and, in agreement with relevant literature (see for instance [56], and [15]) the Ekman spiral appears strongly modified (Fig. 5.3b).

Stratification also affects the elements of the Reynolds stress tensor. Figure 5.4a shows the *rms* of the velocity components, namely the square root of the normal Reynolds stresses. As already observed by [5] in the analysis of a strongly stratified turbulent channel flow, stratification affects the normal Reynolds stresses in an anisotropic way. Specifically, the horizontal normal Reynolds stresses are enhanced whereas the vertical Reynolds stress is strongly inhibited. This is due to the fact that stratification suppresses directly vertical Reynolds stresses as well as the pressure-strain correlation which transfers turbulent kinetic energy (TKE) from the horizontal plane to the vertical direction. As a consequence the TKE produced in the horizontal directions tends to remain confined in the horizontal plane.

The non-dimensional Reynolds shear stresses are shown in Fig. 5.4b for the two cases studied. Stratification inhibits the amount of vertical mixing of momentum and reduces the thickness of the water column where significant mixing is present. In particular the spanwise-vertical Reynolds stress appears dramatically affected by stratification and this explains the strong increase of the mean spanwise velocity observed in Fig. 5.3a.

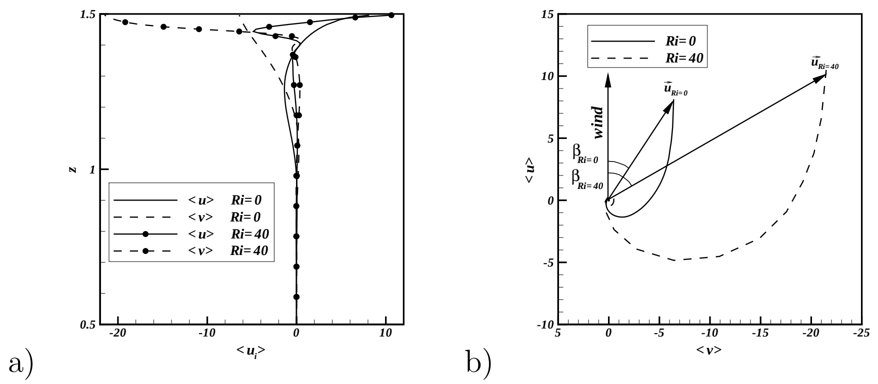

Figure 5.3: a)Vertical profile of the non dimensional mean horizontal velocity components. b) Hodograph of the horizontal components of the mean velocity.

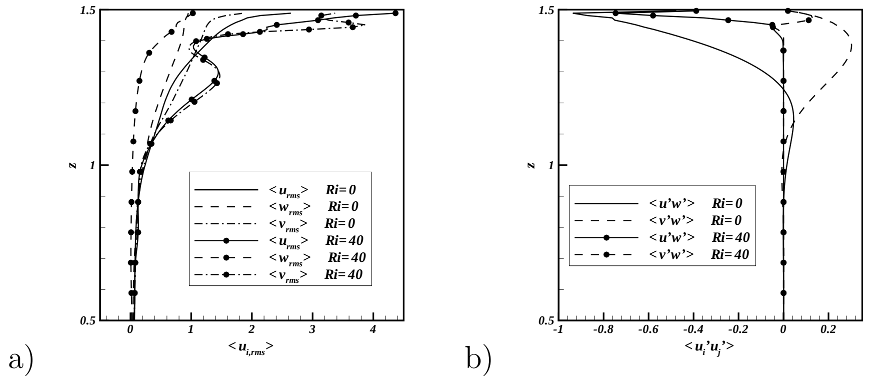

Figure 5.4: a) Vertical profile of the *rms* of the velocity components. b)Vertical profile of the Reynolds shear stresses.

The vertical profiles of the mean density and of its *rms* are shown in Figs. 5.5a and 5.5b respectively. In case of neutral flow, the density has to be considered as a passive scalar. From a physical point of view this represents a case where stratification is weak enough that it does not affect the velocity field ($Ri \rightarrow 0$).

In the $Ri \rightarrow 0$ case density slowly decreases going up toward the free surface and it appears well mixed in the upper part of the domain. Conversely, in case of strong stratification, the confinement of the mixed layer in the very upper region of the domain causes a strong decrease of the fluid density in the free surface region and the development of a picnocline (layer where the largest density gradient takes place) in a thin region below the free surface. A similar behavior is observed for the density fluctuations, quantified by ρ_{rms} (Fig. 5.5b): specifically they are very intense and spread over the fluid column in the neutral case, whereas they are confined within a thin region above the

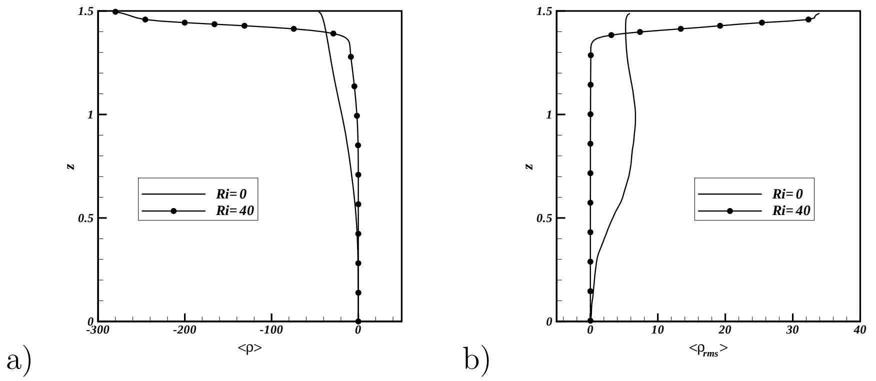

Figure 5.5: a) Vertical profiles of the mean density. b) Vertical profiles of the density root mean square.

picnocline in the stratified case. The analysis of the vertical profile of the gradient Richardson number $Ri_g = N^2(z)/S^2(z)$, where $N^2(z) = -g/\rho_0 d < \rho > /dz$ and $S = d < u > /dz$, in the stratified case, shows that its value is larger than 0.2 along the whole fluid column. According to [5] this indicates that the fluid column is in a *buoyancy dominated* regime, where turbulence is suppressed and internal waves are present in the picnocline region. In Fig. 5.6 it is shown the vertical buoyancy flux $< \rho' w' >$ which quantifies the attitude to vertical mixing of mass in the fluid column. The $Ri \rightarrow 0$ case shows that large vertical buoyancy fluxes are present in the free surface region where active turbulence is present, however noticeable activity is observable along a depth equal to 1.25δ, even larger than the vertical length scale δ.

The presence of strong stratification reduces the vertical buoyancy flux. In particular the maximum value is reduced by more than 20% and, more importantly, the region where appreciable values are recognized remains limited in the free surface region. Counter-gradient buoyancy fluxes are recorded in the region where the picnocline intensifies, and this is a typical feature of the *buoyancy dominated* regime described in [5].

Finally Fig. 5.7 illustrates two iso-density surfaces respectively in the neutral case and in the $Ri = 40$ one. In the neutral case (Fig. 5.7a) the iso-surface shows the presence of large mixing spread over a wide range of spatial scales and extending along the upper part of the water column. Conversely, in Fig. 5.7b it is possible to see the dramatic inhibition of vertical mixing caused by the stable stratification which is also associated with the presence of internal waves. This destruction of the vertical turbulent structures greatly affects the particle dispersion patterns as it will be shown in the next section.

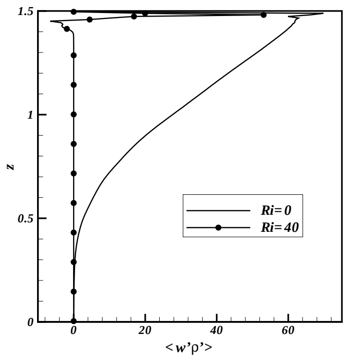

Figure 5.6: Vertical profiles of the vertical buoyancy flux.

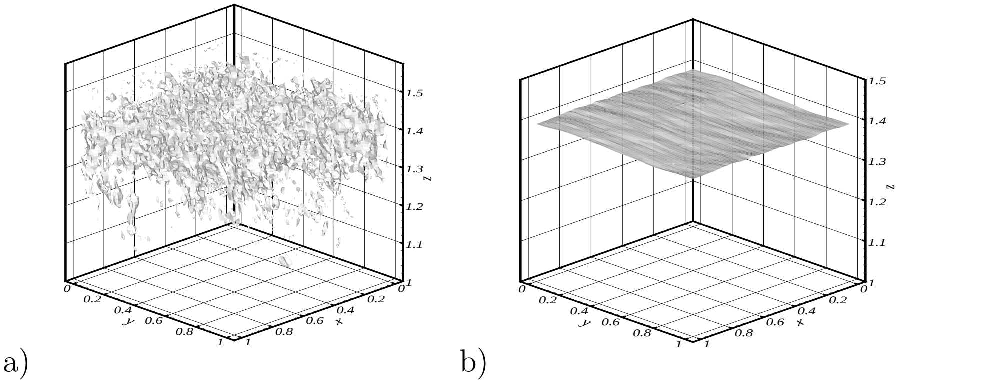

Figure 5.7: Iso-density surfaces in the two cases investigated: a) $Ri \to 0$; b) $Ri = 40$. Note the presence of internal waves in the $Ri = 40$ case.

5.2.3 Lagrangian particles

The dynamics above observed has a dramatic impact on the dispersion of the buoyant jet. Figs. 5.8 and 5.9 offer a $3D$ view of the plume of particles at $t = 1.10$. In case of neutral flow (Fig. 5.8) the particulate moves upward along a cylindrical path up to $z = 1.25$, in a region where the level of turbulent fluctuations is very small. Moving upward, the particulate enters a region characterized by the presence of non-zero horizontal velocity components and appreciable turbulent fluctuations, that cause the destruction of the cylindrical structure of the jet and the horizontal spreading of the particulate. This effect is enhanced when the particulate continues to move upward and once it reaches the free surface it is spread horizontally in the flow field.

A very different scenario is observable in the stratified case (Fig. 5.9). The suppression of turbulence in the region below the picnocline (discussed in the previous section) makes the particulate to travel along a well structured cylindrical path up to the picnocline. Due to the presence of a weak horizontal velocity field and of the Coriolis force, this path undergoes a weak distortion in the region $1.2 < z < 1.35$ still maintaining its own organized structure. However, once the plume reaches the picnocline, it remains confined in that region and oscillates according to a wave-like behavior, due to the presence of internal waves in the fluid column. The confinement of the buoyant jet in the picnocline region has to be attributed to the fact that the particles reach a condition of hydrostatic equilibrium below the free-surface, due to the strong reduction of the ambient fluid density (see Fig. 5.5a). This effect is well known in practical applications (see for example [58]) and is well captured by the mathematical model here proposed, which considers an extra-term proportional to the Richardson number in the particle motion equation.

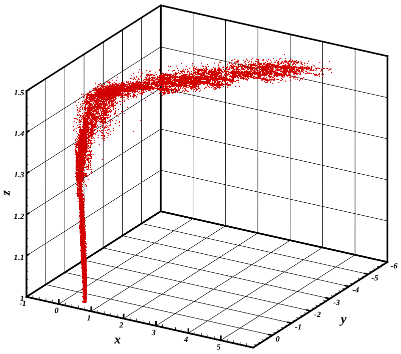

Figure 5.8: Three dimensional view of the plume of particles at the final time of simulation $t = 1.1$. Case $Ri \to 0$.

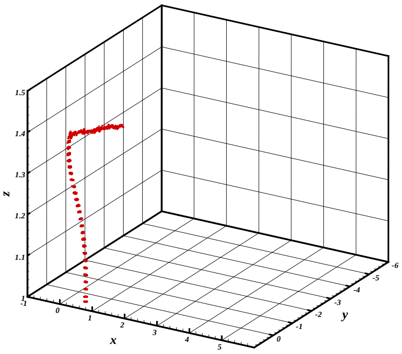

Figure 5.9: Three dimensional view of the plume of particles at the final time of simulation $t = 1.1$. Case $Ri = 40$.

The horizontal displacement and diffusion of the cloud of particulate appear to be inhibited by stratification (see Fig. 5.10a) for two reasons. The horizontal displacement is strongly reduced because in the stratified case the plume remains entrapped in the picnocline region, well below the free surface, in a region where the horizontal components of the velocity field are small. The horizontal diffusion of the plume is suppressed by stratification due to the inhibition of the turbulent fluctuations in the velocity field.

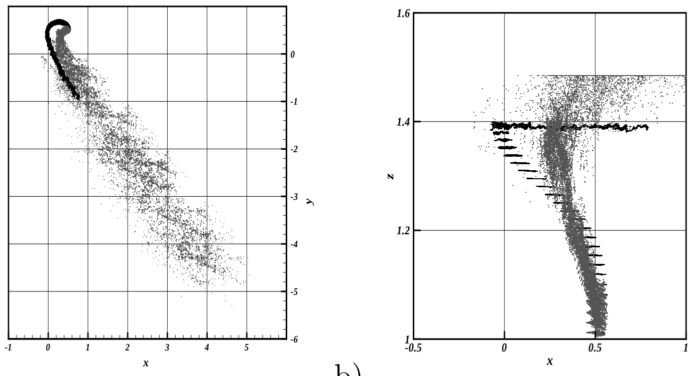

Figure 5.10: Two-dimensional views of the plume at $t = 1.1$. Grey particles, $Ri \to 0$; Black particles, $Ri = 40$. a) Top view; b) Side view.

A lateral view of the plume of particles (Fig. 5.10b) shows that in the stratified case the jet of particles conserves its shape up to the picnocline; in this region the plume moves horizontally according to the internal waves generated in that region. Conversely, the particles moving in the neutral flow experience a large horizontal spreading well before reaching the free surface.

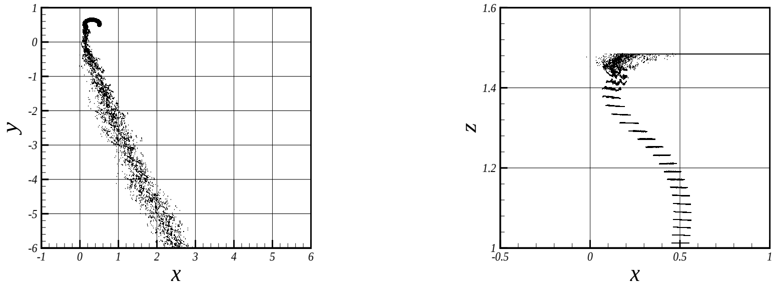

Figure 5.11: Two-dimensional views of the plume at $t = 1.1$ evaluated without the Ri term in eq. (5.7). a) Top view; b) Side view.

With the aim to show the effect of the extra-term proportional to Ri on the dynamics of the plume, we have also run a simulation for the $Ri = 40$

case, not considering the variation of the fluid density in the particle-motion equation, thus omitting the second term of the right hand side of eq. (5.7). Figure 5.11 shows the top and the lateral view of the plume at $t = 1.1$, when obtained omitting the variable-density term. We observe that if the variation of density is not considered in the particle motion equation, the dynamics of the plume is still affected by the stratification effect on the flow field, in that the strong inhibition of turbulence makes the plume to hold its own cylindrical structure up $z = 1.45$. A spreading occurs above $z = 1.45$ and finally the particles reach the free surface and are spread horizontally according to the velocity field. This happens because the particles are not sensible to the local variation of the ambient density and therefore the entrapment effect below the picnocline is not reproduced. As a consequence the vertical upwelling as well as the horizontal transport are dramatically overpredicted (compare Fig. 5.11 to Fig. 5.10).

5.2.4 Concluding remarks

In this chapter the dispersion of a plume of buoyant particles in a stratified wind-driven Ekman layer was investigated. The density ratio herein employed is representative of particles with density comparable with that of the ambient fluid (for example, fresh water particles released into a salty water environment). The analysis was carried out in two different conditions, namely a case of neutral flow and a case of strongly stratified flow. Stable stratification was supplied by incoming heat flux at the free surface of the domain. The study was carried out using LES for the Eulerian phase, whereas the dynamics of the plume of particles was simulated using a Lagrangian technique. In order to make the particles sensitive to the actual fluid density, the particle motion equation was improved and an extra buoyancy term, proportional to the Richardson number, was considered. Because of the Reynolds number considered in our numerical experiment, a wall model approach was employed, directly imposing the wind stress and the heat flux at the free-surface of the domain, not resolving the near-wall structures. The results of the numerical simulations have shown that under stable stratification, the penetration depth of the boundary layer as well as the vertical mixing in the water column decrease. In case of strong stratification, turbulence is almost completely suppressed in the water column and internal waves are observed in the picnocline region. The dispersion of the buoyant jet of particles is dramatically affected by stratification. In the neutral case, once the plume reaches the turbulent region while continuing to go up, it is spread in the horizontal direction by turbulent mixing; afterward, when the particles approach the free surface they are dispersed horizontally according with the local velocity field. In the stratified case, the particles remain entrapped in the picnocline region and are not able to reach the free surface region, characterized by large values of the horizontal velocity components. As a result, both the horizontal displacement and the horizontal spreading of the plume appear strongly inhibited by stratification. In order to check the importance of the extra-term proportional to Ri in the particle-motion equation, it has been run a simulation considering constant ambient density in the particle motion equation. The simulation has clearly shown that neglecting the variation of density leads to a strong overestimation of the vertical as well of the horizontal transport of the plume.

Chapter 6

Applications

Here the tools developed in the previous chapters have been applied to two practical studies, specifically an environmental flow and an industrial one. The idea is to face real life problems with the methodology here proposed. The first study regards an environmental flow, specifically the study of turbulent mixing due to a river merging into the sea (estuarine flow). This case is suited to the application of the IBM of chapter 2 and the SGS model developed in chapter 4. The second study concerns an inertial filter to collect paint droplets. This is an industrial application where the use of IBM with curvilinear grid is particularly suited.

6.1 Simulation of an estuarine flow

In this section the mathematical model is applied to the study of an estuarine flow. The study concerns the mixing effects due to the incoming flow of a river in the sea, specifically the Tevere river that goes into the Tirreno sea. A sketch of the area can be seen in Fig. 6.1. This type of simulation can be useful to better understand the sea dynamics, in particular to study the effect of river pollution on a touristic area.

The main forcing considered are a sea current coming from south and the river inflow. These forcing terms come from an analysis of the characteristics of the sea currents in this area. Since the domain in the horizontal plane is large and the characteristic velocities, both for the river and for the sea current, are small, the Coriolis effect cannot be neglected. Because of this sea current and the Coriolis force the river stream is expected to deviate toward north and so the area investigated is located at the north with respect to the river mouth. The incoming flow from the river has the density of fresh water, this light fluid goes into a salt environment, that is heavier. We consider a winter condition, and consequently the stratification comes from the merging of the two currents only. In other words density anomaly is just due to difference in salt concentration. Under these conditions Reynolds analogy holds. So for a turbulent flow we can consider that the rate of transport of a scalar (*i.e.*, a concentration) is the same of that of momentum. In the advection-diffusion equation (1.12) for salinity concentration we can consider a turbulent Schmidt number $Sc_t = 0.5$ (where $Sc_t = \nu_t/k_t$, with k_t the eddy diffusivity). Density is treated as an active scalar in momentum equation, it affects the motion through the buoyancy term as in eq. (1.20), then we expect that the incoming jet will rise on the surface because of buoyancy effects, spreading over the horizontal plane (in agreement with literature results).

The domain considered for the numerical simulation covers an area of $5km$ x $6km$ in the horizontal plane and it is discretized with 385 grid points both for x direction and z direction, while 33 points are taken in the vertical direction y where the maximum depth is 50 meters. This determines cells of about ten meters in the horizontal plane and of 0.4 meters in the vertical direction close to the coastline. Because of the characteristic length scales in the horizontal and vertical plane the grid is strongly anisotropic. This domain is particularly suited for the use of the SGS model developed in the chapter 4 and so different eddy viscosities are considered, namely one for the vertical direction and one for the horizontal one.

A sketch of a horizontal plane of the grid can be seen in Fig. 6.2a, for simplicity a coarser one is shown. The horizontal plane is a circular sector with an angle of 20 degrees. The computational grid is constructed from the bathymetry. In Fig. 6.2b the bottom of the grid can be seen, in the figure the vertical direction is magnified with respect to the horizontal ones. The sea depth reach a maximum of 50 meters at the corner denoted as SO (see Fig. 6.1). Then the bottom smoothly reaches the coast line with a depth of about one meters.

As suggested in chapter 4 some simplifications are possible in the computation of the metric terms if the vertical curvilinear coordinate η is taken aligned with the Cartesian vertical direction as it is done in the present case, see Fig. 6.3a. For these simplifications to hold, a weak slope of the grid bottom is required. In order to satisfy this requirement part of the bottom and of the coastline is modeled using the IBM as described in chapter 2. This avoids the use of a grid with an excessive distortion. A sketch of the use of IBM is given in Fig. 6.3b.

The boundary conditions are chosen in the following way. At the sea bottom and at solid walls (*i.e.* the Immersed Boundaries) a wall function is used to avoid the resolution of the near wall structures in order to maintain a coarser grid near walls and to save cpu time, see chapter 3 for details. At the boundary SE-SO a sea current parallel to the coast is taken, with a value typical for sea environment of 0.02 m/s. Few real data are known for the region considered, but the general dynamics of Tirreno sea exhibits a near coast current going toward the north. At the boundary NE-SE corresponding to the mouth of the Tevere river we consider an inflow condition. This inflow comes from a

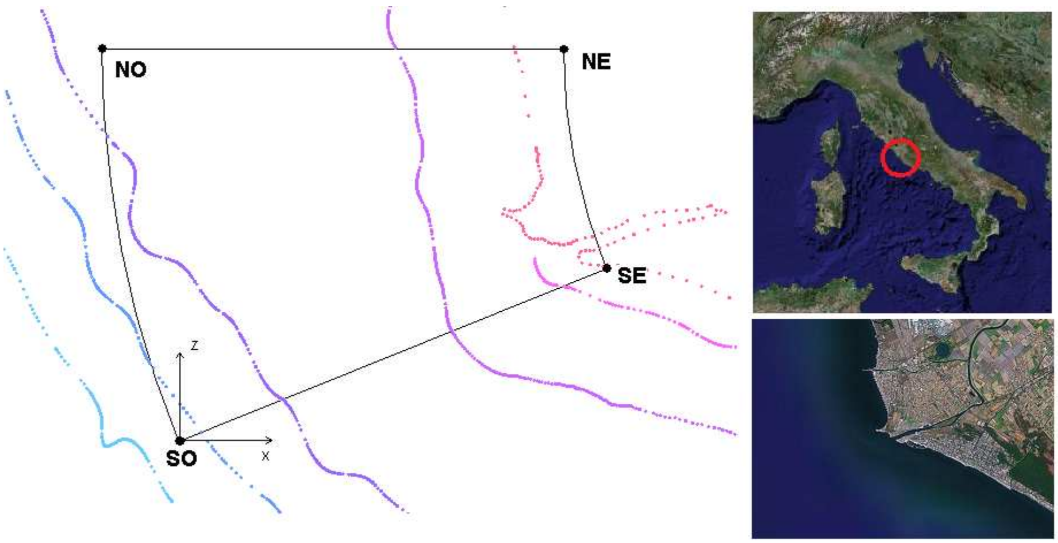

Figure 6.1: Estuarine flow. Physical domain for Tevere river area. Bathymetric lines and area analyzed.

pre-simulation of a turbulent channel flow with the characteristics river cross-stream geometry and mass transport. The average velocity is about 0.3 m/s while the inflow is of about 300 m^3/s. This value is typical for Tevere in winter time. The channel flow is simulated considering the Coriolis effect, so its profile is not symmetric, and it provides the turbulent kinetic energy for the simulation.

A free-slip condition is considered on the sea surface, while on the remaining boundaries an Orlanski condition is taken. A simulations with density as a passive scalar (SDP) is performed in order to highlight the difference with respect to the use of density as an active scalar (simulation SDA).

Figures 6.4a and 6.4b show a contour for the horizontal velocities u and w respectively for the simulation SDA. The plots refer to an horizontal plane just below the free surface. It can be seen that the river flow once in the sea tends to go toward the north direction due to the presence of the meridional sea current and because of the Coriolis effect. It remains far from the coastline and a large recirculation area is trapped between the main flow and the coast line. In particular we can observe the asymmetric profile of the river inflow due to the Coriolis effect, namely higher velocity is close to north bank. Many small recirculation areas can be seen where the river flow meets the meridional sea current. Figure 6.4c shows a plot for the vertical velocity component. As can be seen from the contour legend, the vertical velocity is of one order of magnitude smaller than the horizontal velocity components, this difference in the magnitude is typical for sea flow.

The Figs. 6.4a, 6.4b, 6.4c show a contour for the velocity component u

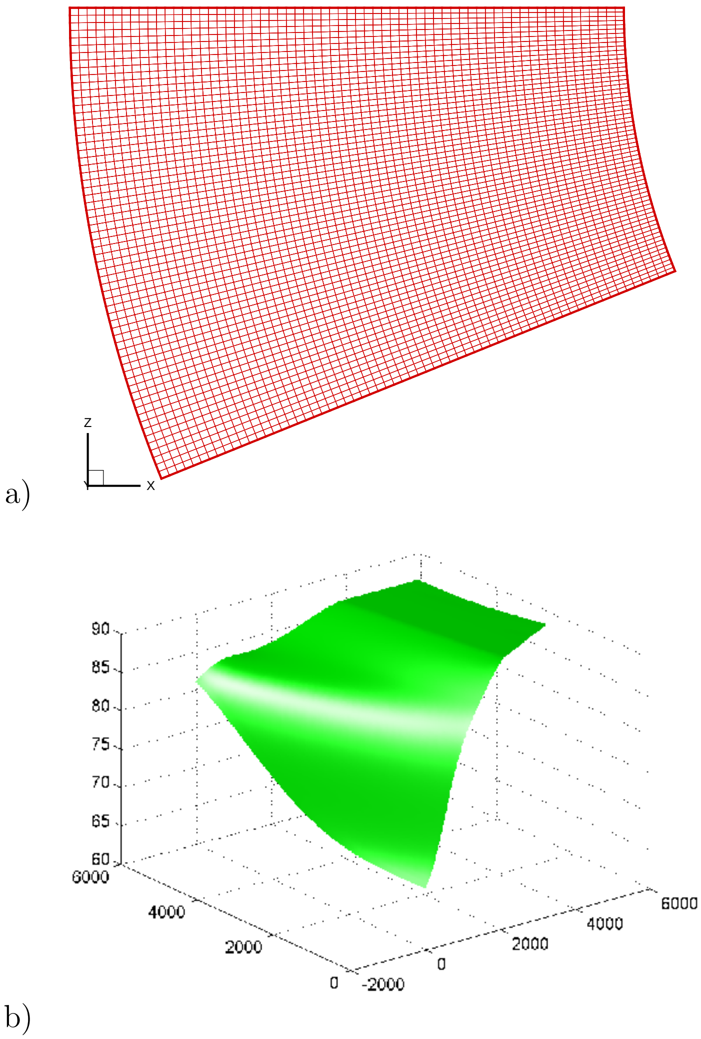

Figure 6.2: Estuarine flow. a) Sea bottom for Tirreno sea in front of Tevere mouth interpolated from bathymetric data with a biharmonic spline. b) Horizontal plane for a coarse computational grid in the case of Tirreno simulation.

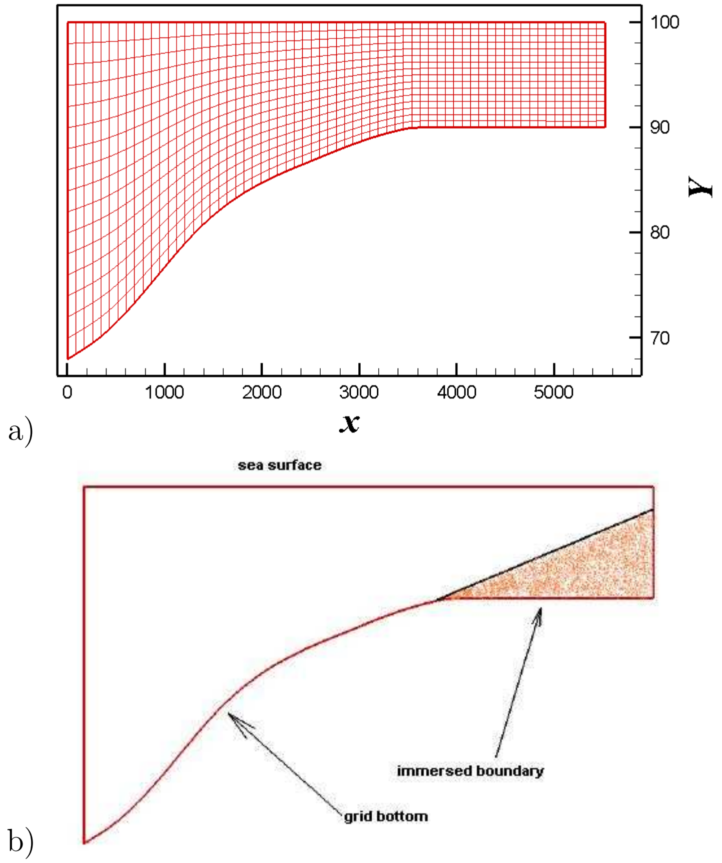

Figure 6.3: Estuarine flow. a) Lateral view for the computational grid. b) Example of sea domain partially modeled with IBM.

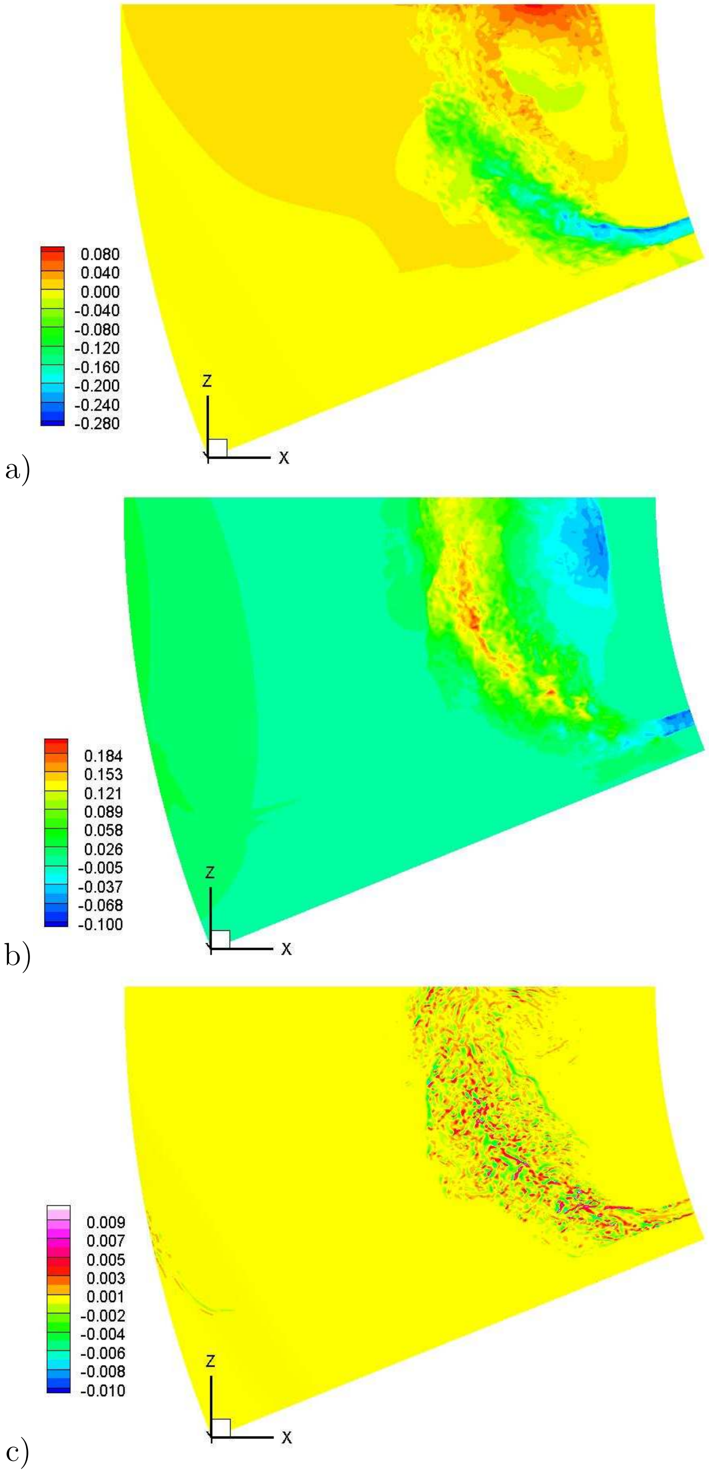

Figure 6.4: Estuarine flow simulation SDA. a) Contour for u velocity at one meter below the surface. b) Contour for w velocity at one meter below the surface. c) Contour for v velocity at one meter below the surface.

w v respectively, for the simulation SDP. The general behavior is the same of simulation SDA, because of the forcing considered in both the simulations, but the velocity magnitude close to the surface is lower and the river stream is more confined. This is due to the fact that in SDP momentum just transports the salinity and the flow is not affected by buoyancy, on the other hand in SDA the river flow, because of the gravitational term, tend to rise over the heavier sea fluid and to spread on the horizontal plane. The vertical velocity activity for simulation SDA is larger than simulation SDP. This can be explained with the intrusion of salt water in the fresh water layer. This will be more clear looking at density contour plots.

In Figs. 6.6a, 6.6b, 6.6c a contour plot of the density anomaly ρ'/ρ_0 can be seen. The figures show instantaneous values at different levels on vertical direction for simulation SDA. The red color distinguishes the salt water while fresh water is in blue. The light water coming from the river, because of buoyancy tends to rise over the salt water and to spread in the horizontal direction. This is due to the fact that density is treated as an active scalar in momentum equations. Then the fresh water is transported by the mean current and so go toward the north without reaching the coast far from the river's mouth. The fresh water concentration rapidly decreases far from the river mouth. Figure 6.7 shows the formation of a salt edge inside the river mouth. On the lower levels the heavier salt water tends to compensate the rise of fresh water. The salt wedge can be much larger than the one observed in the figure, but here cannot be simulated properly because of the boundary conditions considered. Specifically the salt intrusion can move upward in the river for hundred meters. The proper simulation would require a very long river channel, or, better, the knowledge of the actual salinity profiles near the mouth. In Figs. 6.8a, 6.8b, 6.8c a contour plot for the density anomaly ρ'/ρ_0 can be seen. The figures show instantaneous values at different levels along the vertical direction for simulation SDP. In this case the behavior is completely different compared to the simulation SDA. Here density is just transported by momentum equation and the fresh water remains confined far from the surface with an unphysical behavior.

Figures 6.9a and 6.9b show a contour plot for the eddy viscosities, the horizontal and the vertical component respectively, for the simulation SDA. As expected the two values differ approximatively of one order of magnitude. Larger values are observed where shear phenomena are present, so close to the river banks and where the river flow meets the sea water.

Finally Fig. 6.10 shows a velocity field, here represented through a coarse vectors field, qualitatively compared to an image from a satellite. The simulation seems in agreement with real observation, considering that the sea meridional current intensity is arbitrary and the river inflow considered was for a winter time configuration.

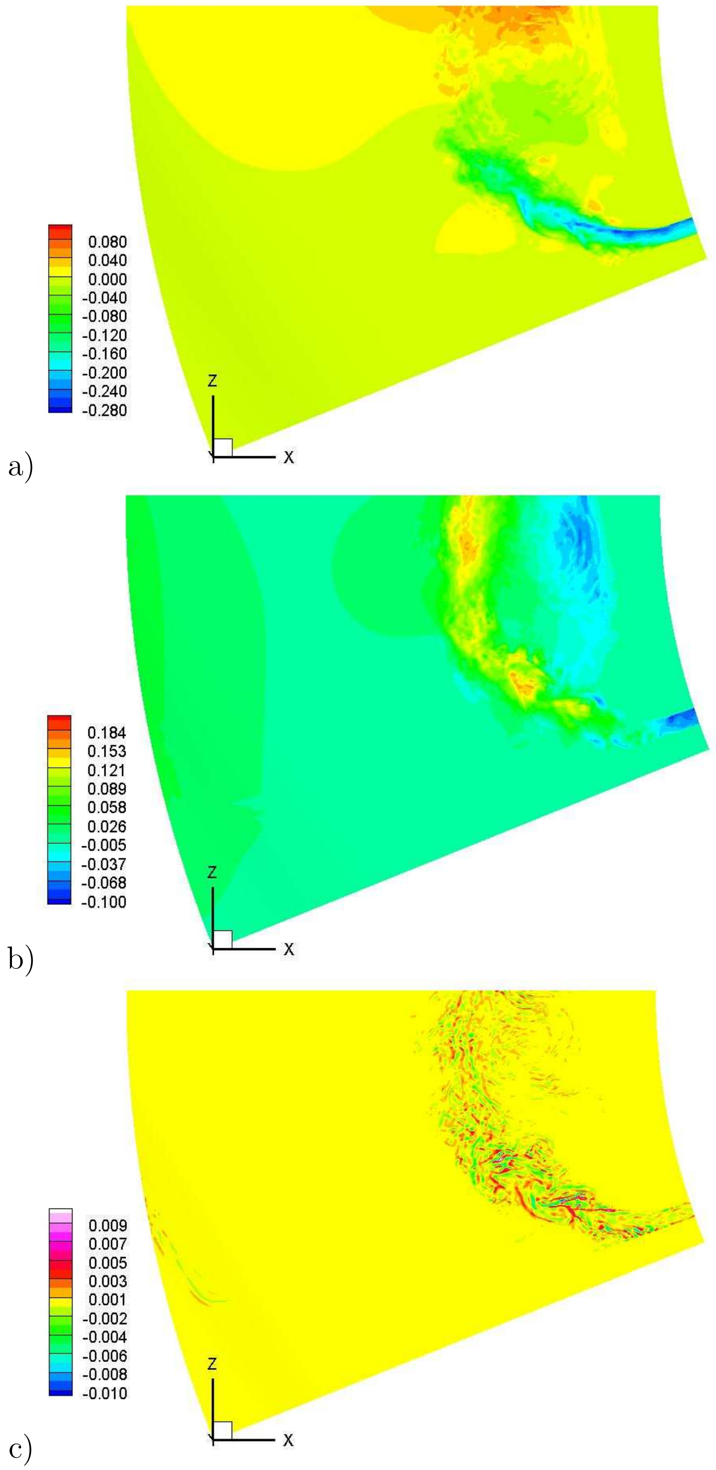

Figure 6.5: Estuarine flow simulation SDP. a) Contour for u velocity at one meter below the surface. b) Contour for w velocity at one meter below the surface. c) Contour for v velocity at one meter below the surface.

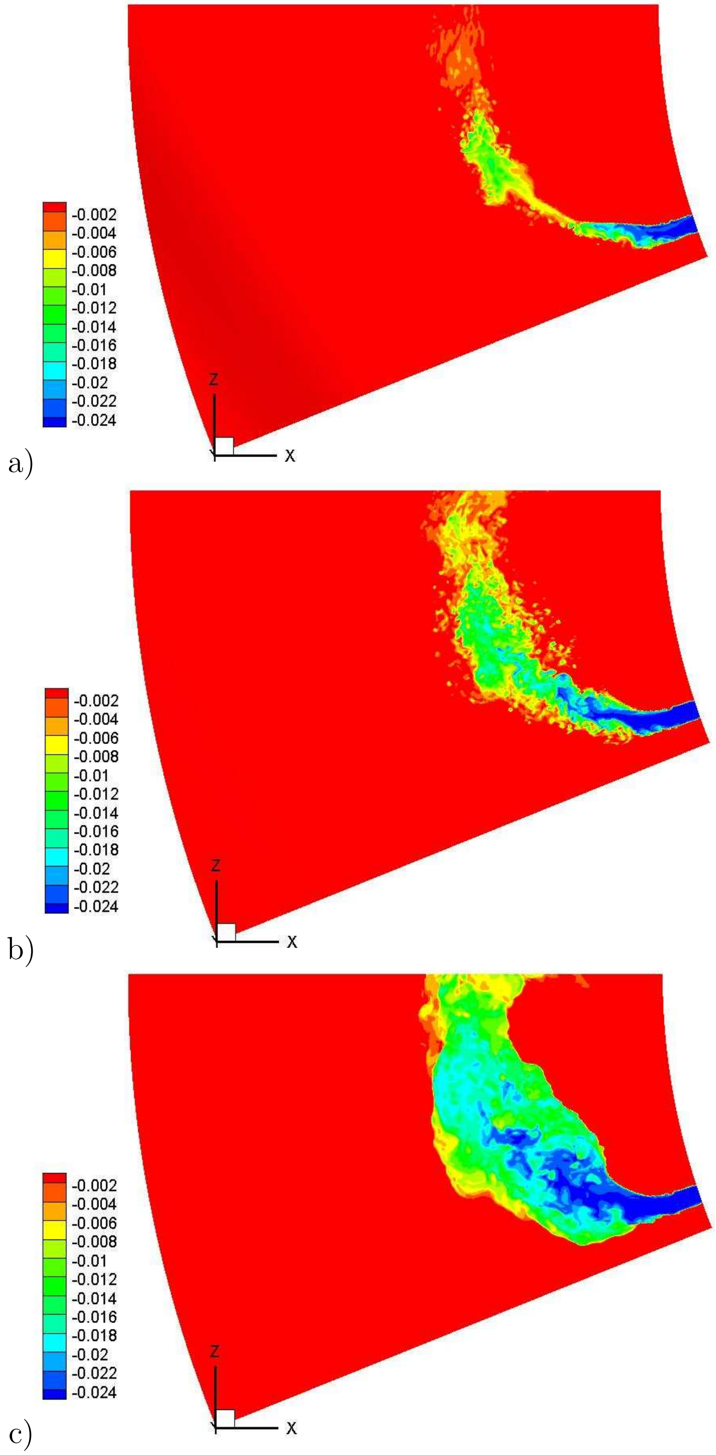

Figure 6.6: Estuarine flow simulation SDA. a) Contour plot for ρ'/ρ_0 at the bottom. b) Contour plot for ρ'/ρ_0 at 5 meters from the sea surface. c) Contour plot for ρ'/ρ_0 at the sea surface.

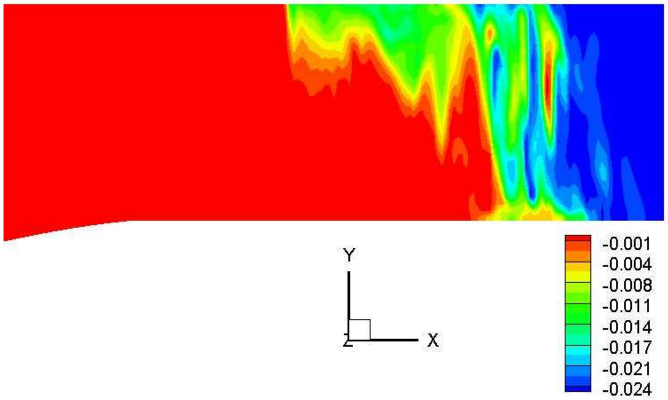

Figure 6.7: Estuarine flow simulation SDA. Contour plot for ρ'/ρ_0 in the river mouth.

6.2 Simulation of a practical industrial problem

Here we discuss the application of the numerical tools to an industrial problem, related to the efficiency of an inertial filter. These filters are widely in use to reduce the concentration of fine painting droplets in air, during a painting process in industrial applications. This problem shows the advantage of using an Immersed Boundary Methodology in conjunction with curvilinear grids.

A typical filter geometry is shown in Fig. 6.11a (see [78]). The filter is composed of a sequence of identical elements (Fig. 6.11b). Air enters through a hole (indicated with 1 in Fig. 6.11b), then air flows within the cavity and goes out from an outflow hole (2 in Fig. 6.11b). The inflow and outflow holes are staggered between each other to ensure air mixing within the chamber. The filter is made of paper and the painting droplets tend to attach over the paper surface during the air flow. This allows to filter out a large amount of particles from air. The increase of turbulent mixing within the chamber enhances the adherence of particles to the paper surface and consequently the efficiency of the device. Due to the modularity of the filter, the computational domain here is limited to a single element of the panels, as shown in Fig. 6.12, taking advantages of the periodicity of the problem. In Fig. 6.12 the computational setup together with a plane section are plotted. It is possible to observe that the use of a non-orthogonal grid obtained using diamond-like cells allows modeling the lateral sides of the filter (side AD of the actual chamber and side BC of the successive one) as body fitted surfaces, whereas the other surfaces are modeled using immersed boundaries. This combination reduces the number of surfaces to be modeled with IBM and allows to optimize the number of grid cells needed for the simulation. In practical applications the incoming air velocity u_{air} is

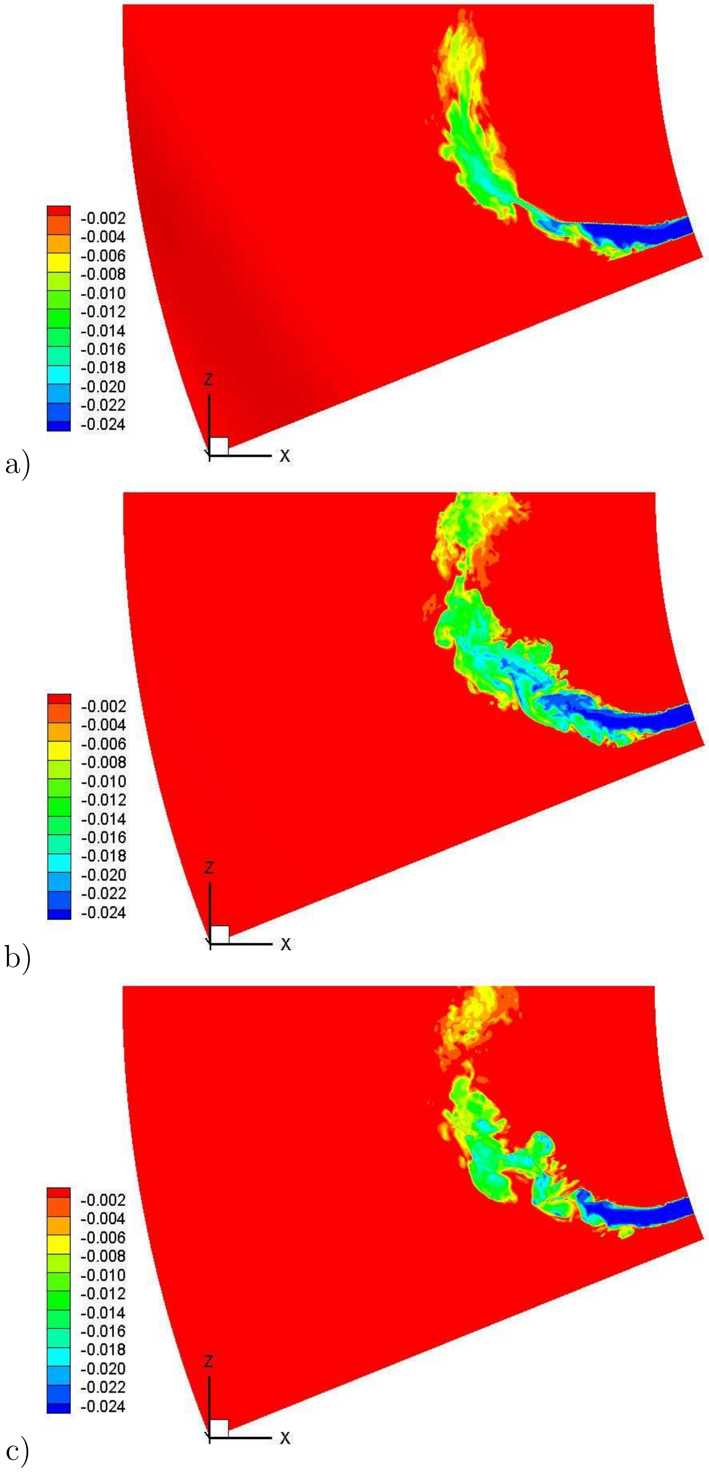

Figure 6.8: Estuarine flow simulation SDP. a) Contour plot for ρ'/ρ_0 at the bottom. b) Contour plot for ρ'/ρ_0 at 5 meters from the sea surface. c) Contour plot for ρ'/ρ_0 at the sea surface.

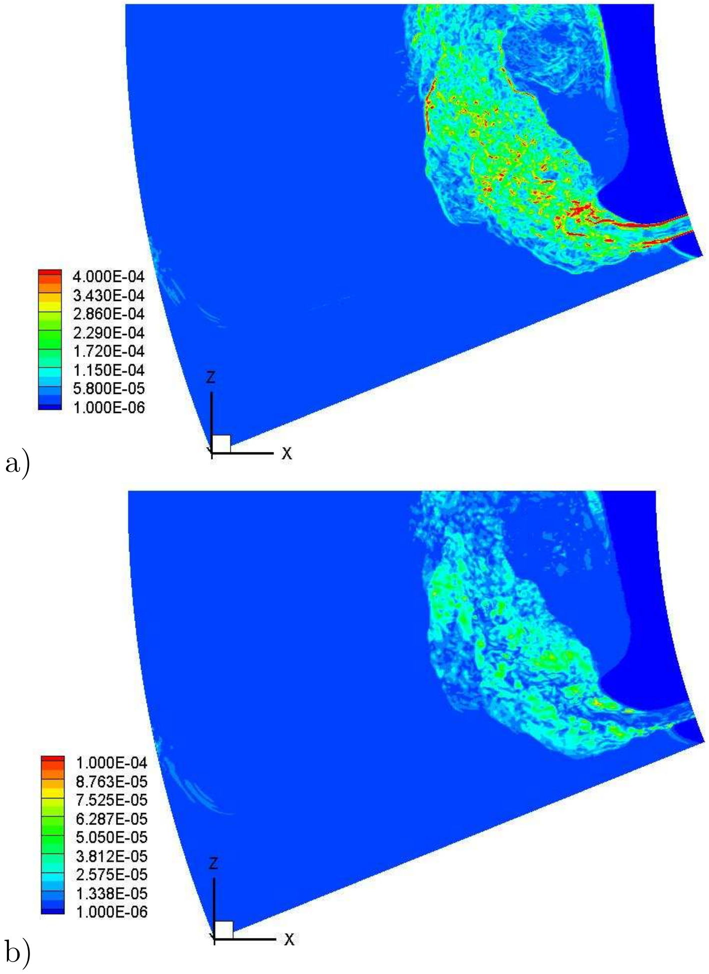

Figure 6.9: Estuarine flow simulation SDA. a) Contour plot for the horizontal eddy viscosity at one meter below the surface. b) Contour plot for the vertical eddy viscosity at one meter below the surface.

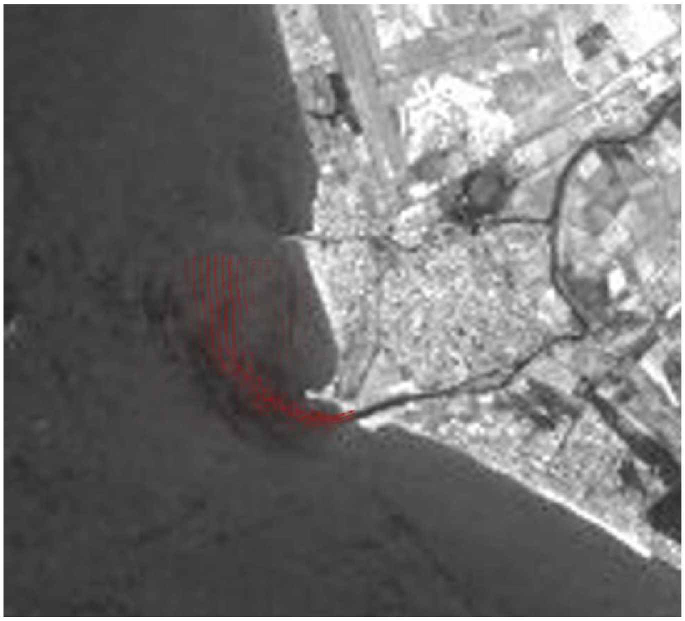

Figure 6.10: Estuarine flow SDA. Comparison of a coarse instantaneous vector plot with an image from satellite

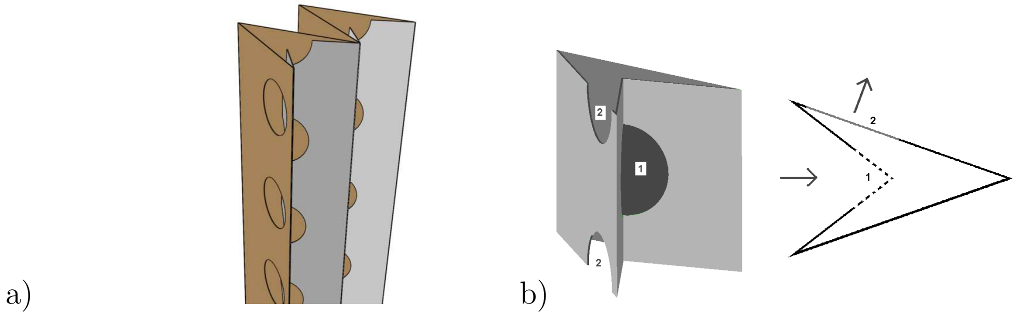

Figure 6.11: Geometry of the filter (a) with identification of the modular element (b)

of the order of $0.5m/s$ giving $Re = u_{air}l_{AB}/\nu = 2000$ where $l_{AB} = 0.04m$ is the distance between the upstream edges of the filter (Fig. 6.12b) and $\nu = 1.0 \times 10^{-5}$. At this Reynolds number a direct numerical simulation (DNS) can be performed and we use 192 grid cells in the streamwise direction x, 160 cells in the y direction and 128 in the z direction. The inflow velocity at the surface represented by the segment AB of Fig. 6.12 is a series of turbulent planes generated with a pre-simulation of a turbulent plane channel flow. At the outflow surface represented by the segment CD of Fig. 6.12 a radiative condition is imposed. No-slip conditions are imposed over the filter surfaces represented by the segments BC and AD and over the immersed surfaces AE, EB and BD. Finally, periodicity is imposed along the z-direction. Here some details are shown of the velocity field. Figure 6.13a shows the instantaneous horizontal vector field at a section intersecting the diameter of the inflow hole. It can be observed that the level of turbulence is very low at the inflow due to the relatively small value of the Reynolds number, and vortex structures are generated beyond the inflow hole. Air is thus forced to move along the vertical direction and flows out through the outflow section. Figure 6.13b shows that air going out through the outflow hole splashes against the surface of the successive chamber and moves downstream along the surface creating a number of small scale structures with enhanced mixing. In this region a large accumulation of painting droplets is expected. On the other hand other surfaces of the chamber (see the surface represented by the segment AE of Fig. 6.12b at the height of the outflow section) bounds a stagnation region where probably particles are not accumulated. Figure 6.14 shows a zoom of the near-surface instantaneous velocity field of Fig. 6.13b. It is possible to observe that the curvilinear-grid IBM method, as discussed in chapter 2, correctly reproduces the near-wall velocity which appears to be tangent to the body surface. Note

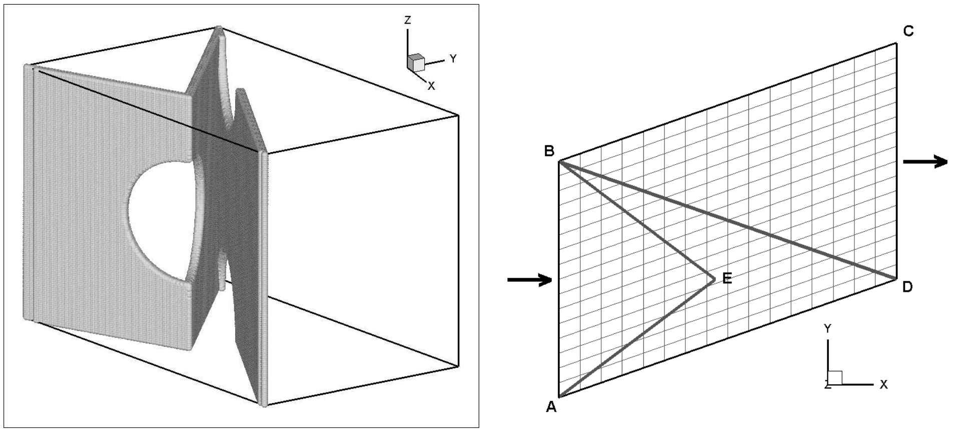

Figure 6.12: a) Three-dimensional view of the computational domain; b) xy-section of the computational domain.

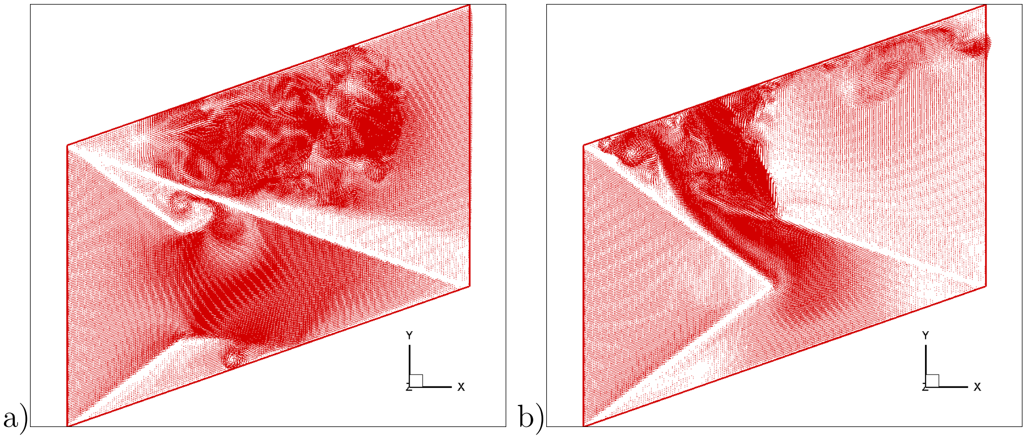

Figure 6.13: Vector velocity field in two xy sections: a) section intersecting the diameters of the inflow hole; b) section intersecting the diameters of the outflow hole.

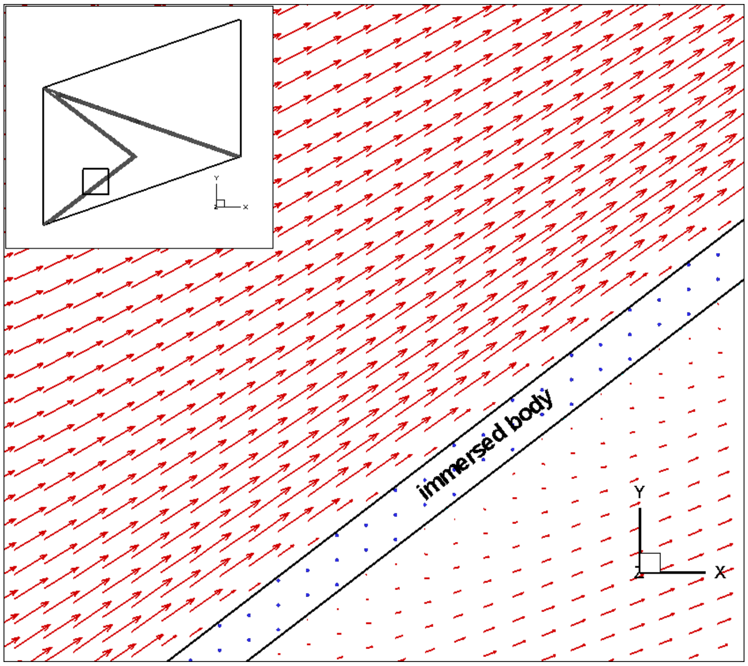

Figure 6.14: Particular of the velocity field shown in Fig. 6.13b.

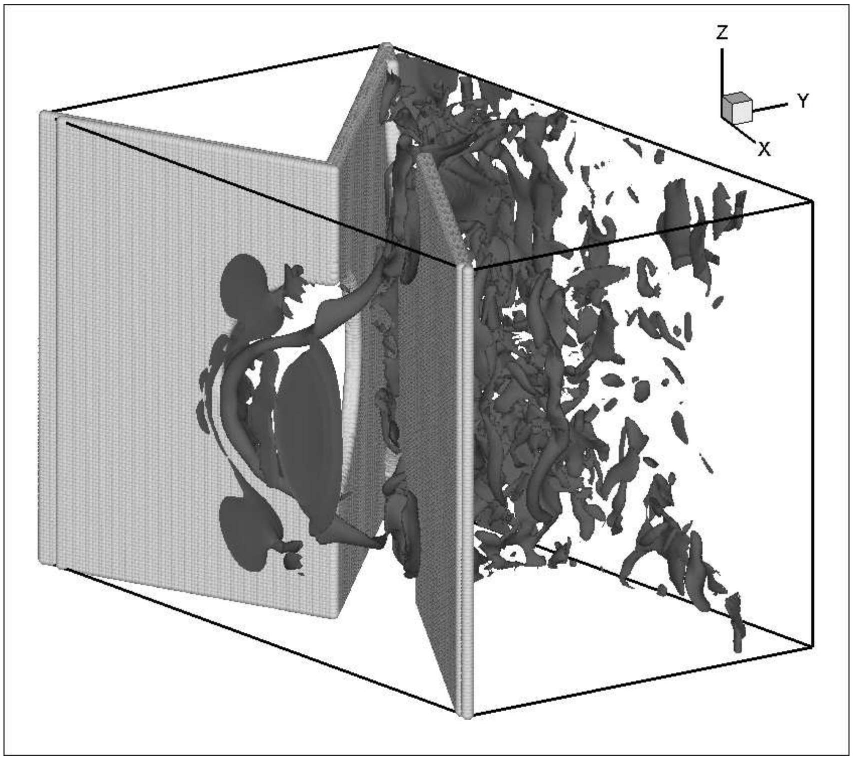

Figure 6.15: Isosurface of instantaneous z-component of vorticity. The value $\omega_z l_{AB}/u_{air} = 144$ is plotted.

the absence of spurious velocity components crossing the immersed body surface even in case of very thin bodies at large value of Re. Finally, Fig. 6.15 gives a three-dimensional illustration of the instantaneous vertical component of vorticity. Small organized structures appear downstream the inflow hole, representative of the vortices visualized in Fig. 6.13a. Conversely once air goes out from the chamber through the outflow hole, due to the already discussed splashes, a large number of small scale structures are generated. A more detailed analysis of the turbulent field together with the investigation of the Lagrangian dispersion of particles is currently ongoing. However, the example herein reported clearly shows the ability of the methodology here presented to efficiently afford applicative complex geometry problems at practical values of the Reynolds number.

Chapter 7

Conclusions

In this thesis different tools were presented with the aim to develop a methodology for LES in real-life problems. In particular the work has focused on: complex geometry, high Reynolds number flow, turbulence model for anisotropic grid, particle dispersion. All these aspects can be fundamental to work in real applications. Below the achievments of the present work are reported for each argument investigated.

An improved Immersed Boundary Method for Curvilinear Grids

In chapter 2 the direct forcing, immersed boundary technique has been extended to the general case of curvilinear-coordinate, structured, Navier-Stokes solvers. Modifications to the available techniques have been proposed with the aim to extend the applicability of the techniques to any-shaped single/multi body configurations also having local high curvatures or sharp corners. In this way a general technique is available to deal with any-shaped multiple-body complex geometry.
Four different numerical examples have proven that the technique is able to accurately simulate a wide class of complex-geometry flow fields. Four main conclusions can be drawn:

- when the geometry is well suited for both Cartesian and curvilinear meshing, the advantage of using curvilinear grids is that the cell distribution in the body region may be better shaped to the body geometry and consequently a smaller number of grid cells may be used when compared to those needed with a Cartesian mesh.

- when the overall geometry of the problem to be studied is not suited for Cartesian meshing, the use of a curvilinear solver allows obtaining more accurate results with a considerable reduction of the computational cost of the simulation. For the S-duct example treated in section 2.2.3, the

Cartesian simulation gave less accurate results with a computational cost equal to 1.77 times that of the curvilinear one; however the increase of the computational cost of the Cartesian simulation can be even larger for more complex situations;

- in a wide class of problems (see the S-shaped duct and the inertial filter discussed in section 6.2) the use of body-fitted curvilinear (or in general non-orthogonal) meshes allows to considerably reduce the amount of body surface to be reproduced by means of immersed boundary, thus improving accuracy;
- a simplified curvilinear-coordinate, direct-forcing, semi-implicit fractional step algorithm has been proposed. Results clearly show that no additional complications are introduced and that the error related to the use of this simplified technique is negligible in a wide range of computational time steps and rapidly decreases with the Reynolds number of the simulation.

Wall modeling with an Immersed Boundary Method

In chapter 3 the Immersed Boundary Method developed in chapter 2 has been extended to high Reynolds number flow introducing a wall model approach. It has been highlight that the usual way of implementing wall modeling is not suited for the IBM and an alternative procedure is required. The proposed method uses the logarithmic profile close to the wall in conjunction with an eddy viscosity mixing length approach at the Immersed Boundary interface.

The method has been proved in a turbulent plane channel flow, with different grid configurations and at different Reynolds number up to $Re_\tau = 2000$. The results obtained are satisfactory. Further investigations on the approach are ongoing. In particular the extension of the method to curvilinear grid will be the next achievment.

A SGS model for sea coastal flow / Simulation of an estuarine flow

In chapter 4 a SGS model for anisotropic grid has been proposed. The model is suited for environmental flow, in particular for sea coastal flow or lakes, where the discretized computational domain could present $sheet-like$ cells. The model has been used in a practical application, specifically it has been applied in the simulation of turbulent mixing for an estuarine flow, see section 6.1. Proper test cases are not available to prove the model, but the results obtained in section 6.1 are in good agreement with real observations.

Dispersion of a vertical jet of buoyant particles in a stably stratified wind-driven Ekman layer

In chapter 5 the dispersion of a plume of buoyant particles in a stratified wind-driven Ekman layer was investigated. The equation for the particle motion was modified in order to make the particle sensitive to actual fluid density. So an extra buoyancy term proportional to the Richardson number was considered.

The results have shown that stratification inhibits the vertical mixing, under strong stratification turbulence is almost suppressed in the water column and internal waves are observed. Also the particles spreading strongly decreases in stratified flow. Besides without the extra term in the particle motion the vertical and horizontal transport is overpredicted.

Bibliography

[1] Armenio V. An improved MAC method (SIMAC) for unsteady high-Reynolds free surface flows. Int. Jou. for Num. Meth. in Fluids 1997;24(2):185–214.

[2] Armenio, V., Fiorotto, V. The importance of the forces acting on particles in turbulent flows. Physics of Fluids 2001;13:2437-2440.

[3] Armenio V., Piomelli U., Fiorotto V. Effect of the subgrid scales on particle motion. Physics of Fluids 1999;11:3030–3042.

[4] Armenio V., Piomelli U. A Lagrangian mixed subgrid-scale model in generalized coordinates. Flow Turb. & Comb. 2000;65:51–81.

[5] Armenio, V., Sarkar, S. An investigation of stably stratified turbulent channel flow using large-eddy simulation. Journal of Fluid Mechanics 2002;459:1–42.

[6] Balaras E. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. Computers & Fluids 2004;33:375–404.

[7] Balaras E., Benocci C., Piomelli U. Finite-Difference Computations of High Reynolds Number Flows Using the Dynamic Subgrid-Scale Model. Theoretical and Computational Fluid Dynamics 1995;7(3):207–216.

[8] Bardina J., Ferziger J.H., Reynolds W.C. Improved Subgrid-Scale Models for Large-Eddy Simulations. AIAA Paper 1980.

[9] Bou-Zeid E., Meneveau C., Parlange M.B. A scale-dependent Lagrangian dynamic model for large eddy simulation of complex turbulent flows. Physics of Fluids 2005;17.

[10] Cabot W. Large-eddy simulations with wall models. In: Annual Research Briefs. Center for Turbulence Research 1995;41–50.

[11] Cabot W. Near-wall models in large eddy simulations of flow behind a backward-facing step. In: Annual Research Briefs. Center for Turbulence Research 1996;199–210.

[12] Choi J., Oberoi R.C., Edwards J.R., Rosati J.A. An immersed boundary method for complex incompressible flows. Journal of Computational Physics 2007;224:757–784.

[13] Cioffi F, Gallerano F, Napoli E. Three-dimensional numerical simulation of wind driven flows in closed channels and basins. J. Hydraul Res 2005;43(3):290–301.

[14] Coleman G.N., Ferziger J.H., Spalart P.R. A numerical study of the turbulent Ekman layer. Journal of Fluid Mechanics 1990;213:313–348.

[15] Coleman, G.N., Ferziger, J.H., Spalart, P.R. Direct simulation of the stably stratified turbulent Ekman layer. Journal of Fluid Mechanics 1992;244:677–712.

[16] http://www.cost.esf.org/

[17] Cristallo A., Verzicco R. Combined Immersed Boundary/Large-Eddy-Simulations of Incompressible Three Dimensional Complex Flows. Flow Turb. & Comb. 2006;77(1-4):3–26.

[18] De Palma P., de Tullio M. D., Pascazio G., Napolitano M. An immersed-boundary method for compressible viscous flows. Computers & Fluids 2006;35:693–702.

[19] De Tullio M.D., De Palma P., Iaccarino G., Pascazio G., Napolitano M. An immersed boundary method for compressible flows using local grid refinement. Journal of Computational Physics 2007;225:2098–2117.

[20] Del Alamo J.C., Jimenez J., Zandonade P., Moser R.D. Scaling of the energy spectra of turbulent channels. Journal of Fluid Mechanics 2004;500:135–144.

[21] Domenichini F., Pedrizzetti G., Baccani B. Three-dimensional filling flow into a model left ventricle. Journal of Fluid Mechanics 2005;539:179-198

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

[23] Fletcher C.A.J. Computational techniques for Fluid Dynamics. vol.II Springer-Verlag 1988.

[24] Fröhlich J., Mellen C.P., Rodi W., Temmerman L., Leschziner M. Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. Journal of Fluid Mechanics 2004;526:19–66.

[25] Ge L., Sotiropoulos F. A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries. Journal of Computational Physics 2007;225:1782–1809.

[26] Gilmanov A., Sotiropoulos F., Balaras E. A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids. Journal of Computational Physics 2003;191:660–69.

[27] Glassner A.S. Space subdivision for fast ray tracing. IEEE CG & A 1984;4(10):15–22.

[28] Goldstein D., Handler R., Sirovich L. Modeling a no-slip flow boundary with an external force field. Journal of Computational Physics 1993;105:354–66.

[29] Inghilesi R., Stocca V., Roman F., Armenio V. Dispersion of a vertical jet of buoyant particles in a stably stratified wind-driven Ekman layer. Int. J. Heat and Fluid Flow 2008;29:733–742.

[30] Hoyas S., Jiménez J. Physics of Fluids 2006;18:011702.

[31] Józsa J, Milici B, Napoli E. Numerical simulation of the internal boundary–layer development and comparison with atmospheric data. Boundary–Layer Meteorol. 2007;123:159–175.

[32] Kaltenbach H.J. Cell aspect ratio dependence of anisotropy measures for resolved and subgrid scale stresses. Journal of Computational Physics 1997;136:399–410.

[33] Kamenkovich V.M. Fundamentals of ocean dynamics. Elsevier 1977.

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

[35] Kim J., Moin P. Application of a fractional step to incompressible Navier-Stokes equations. Journal of Computational Physics 1985;59:308-323.

[36] Kuerten, J.G.M. Subgrid modelling in particle-laden channel flow. Physics of Fluids 2006;18:025108.

[37] Kundu P. Fluid Mechanics. Academic Press 2002.

[38] Lamb H. Hydrodynamics. 6th Edition, Dover Publications, New York, 1945.

[39] Lipari G., Napoli E. The impacts of the ALE and hydrostatic-pressure approaches on the energy budget of unsteady free-surface flows. Computer & Fluids 2008;37:656–673.

[40] Loth F., Fischer P.F., Bassiouny H.S. Blood Flow in End-to-Side Anastomoses. Annual Review of Fluid Mechanics 2008;40:367–393

[41] Lupieri G., Armenio V. An MPI code for Lagrangian dispersion in a turbulent Eulerian field. Science and Supercomputing at Cineca 2005;378–382.

[42] Marchioli C., Armenio V., Soldati A. Simple and accurate scheme for fluid velocity interpolation for Eulerian-Lagrangian computation of dispersed flows in 3D curvilinear grids. Computers & Fluids 2007;36:1187–1198.

[43] Maxey R., Riley J. Equation of motion for a small rigid sphere in a nonuniform flow. Physics of Fluids 1983;26:883–889.

[44] Mills J., On transversely isotropic eddy viscosity. Journal of Physical Oceanogrphy 1994;24:1077–1079.

[45] Mittal R, Iaccarino G. Immersed Boundary Methods. Annual Review of Fluid Mechanics 2005; 37:239–261.

[46] Moin P. Advances in large eddy simulation methodology for complex flows. Int. J. Heat and Fluid Flow 2002;23:710–720.

[47] Mohd-Yusof J. Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries. In: Annual Research Briefs. Center for Turbulence Research 1997.

[48] Napoli E., Armenio V., De Marchis M. The effect of the slop of irregularly distributed roughness element on turbulent wall-bounded flows. Journal of Fluid Mechanics 2008;459:385–394.

[49] Nehari D., Armenio V., Ballio F. Three-dimensional analysis of the unidirectional oscillatory flow around a circular cylinder at low Keulegan-Carpenter and beta numbers. Journal of Fluid Mechanics 2004;520:157–186.

[50] Nehari D., Armenio V., Ballio F. Gap effect on transversal force acting on infinite array of cylinders at low KC and beta. Int J Offshore Polar Eng 2005;15:249–256.

[51] Pedlosky J. Geophysical fluid dynamics. Springer 1987.

[52] Peskin C.S. Flow patterns around heart valves: a numerical method. Journal of Computational Physics 1972;10:252–71.

[53] Phillips O. M. The dynamics of the Upper Ocean. 2nd Edition, Cambridge University Press 1977.

[54] Piomelli U., Balaras E. Wall-layer models for large-eddy simulations. Annual Review of Fluid Mechanics 2002;34:349–74.

[55] Pope S. Turbulent flows. Cambridge University Press 2000.

[56] Price J.F., Sundermeyer M.A. Stratified Ekman layers. J. Geophys. Res. 1999;104:20467–20494.

[57] Roman F., Napoli E., Milici B., Armenio V. An improved Immersed Boundary method for curvilinear grids. Computers & Fluids 2009, in press.

[58] Rubin H., Atkinson J.F. Environmental Fluid Mechanics. Marcel Dekker Inc., New York, 2001.

[59] Sagaut P. Large Eddy Simulations for incompressible flows, an introduction. Springer Verlag 1998.

[60] Salon S., Armenio V., Crise A. Large-Eddy Simulation of an oscillating-rotating turbulent flow. Proceedings, Fourth International Symposium on Turbulence and Shear Flow Phenomena - TSFP4, Williamsburg, VA, USA. 2005;2:383–388.

[61] Salon S., Armenio V., Crise A. A numerical investigation of the Stokes boundary layer in the turbulent regime. Journal of Fluid Mechanics 2007;570:253–296.

[62] Schäfer M., Turek S. The benchmark problem flow around a cylinder. In Flow Simulation With High-Performance Computers II. Hirschel EH (ed.). Notes on Numerical Fluid Mechanics 1996;52:547–66.

[63] Schuman U. Subgrid scale models for finite difference simulations of turbulent flows in plane channel and annuli. Journal of Computational Physics 1975;18:376–404.

[64] Scotti A., Meneveau C., Lilly D.K. Generalized Smagorinsky model for anisotropic grids. Physics of Fluids 1993;5(9):2306–2308.

[65] Shotorban B., Mashayek F. Modeling Subgrid-scale effects on particles by approximate deconvolution. Physics of Fluids 2005;17:081701.

[66] Suo J., Oshinski J., Giddens D. Entrance flow patterns in the coronary arteries: A computational study. Proceedings of the 2005 Summer Bioengineering Conference, Key Biscayne, Florida 2005;191–192.

[67] Sutherland I.E., Sproull R.F., Schumacker R.A. A characterization of ten hidden-surface algorithms. Comput Surv 1974;6(1):1–35.

[68] Taylor J.R., Sarkar S., Armenio V. Large Eddy simulation of stably stratified open channel flow. Physics of Fluids 2005;17:11660.

[69] Tessicini F., Iaccarino G., Fatica M., Wang M., Verzicco R. Wall modeling for Large Eddy simulation using an immersed boundary method. In: Annual Research Briefs. Center for Turbulence Research 2002;181–187.

[70] Tsanis I.K., Wu J., Shen H., Valeo C. Environmental Hydraulics. Developments in Water Sciences. Elsevier 2007, Vol. 56.

[71] Tseng Y., Ferziger J.H. A ghost-cell immersed boundary method for flow in complex geometry. Journal of Computational Physics 2003;192:593–623.

[72] Verzicco R., Fatica M., Iaccarino G., Moin P., Khalighi B. Large Eddy Simulation of a Road Vehicle with Drag-Reduction Devices. AIAA Journal 2002;40(12):2447–2455.

[73] Wang M., Moin P. Dynamic wall modeling for LES of complex turbulent flows. Physics of Fluids 2002;14:2043–2051.

[74] Zahrai S., Bark F.H., Karlsson R.I. On anisotropic subgrid modeling in Large Eddy Simulations. Eur. J. of Mech. B/Fluids 1995;14:459–486.

[75] Zang J., Street R.L., Koseff J.R. A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates. Journal of Computational Physics 1994;114:18–33.

[76] Zikanov O., Slinn D.N., Dhanak M.R. Large-Eddy simulations of the wind-induced turbulent Ekman layer. Journal of Fluid Mechanics 2003;495:343–368.

[77] Zovatto L., Pedrizzetti G. Flow about a circular cylinder between parallel walls. Journal of Fluid Mechanics 2001;440:1–25.

[78] http://www.raccoglivernice.it/AER_IT/ST_FIP.html