European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)

RECENT ADVANCES IN THE IMMERSED BOUNDARY METHOD

M. de Tullio*, A. Cristallo*, E. Balaras†, G. Pascazio*, P. De Palma*, G. Iaccarino‡, M. Napolitano*, and R. Verzicco*

*DIMeG and CEMeC, Politecnico di Bari Via Re David, 200, 70125, Bari, Italia
†Dept. of Mechanical Eng., Univ. of Maryland, College park, MD 20742, USA
‡Dept. of Mechanical Eng. Stanford University, Stanford, CA 94305, USA

Key words: Immersed boundaries, fluid/structure interaction, compressible flows.

Abstract. *In this paper we discuss some of the recent advances in the immersed boundary (IB) method. More specifically, we show the application of this technique to problems with strong fluid/structure interaction and to compressible flows using local mesh refinement. These extensions are a must for the success of the IB method in industrial applications.*

1 INTRODUCTION

Computer simulations are becoming an indispensable tool for the design and analysis of complex systems; performance can be tested in advance in an inexpensive manner and interactions among various independent parameters can be reproduced and studied in a very controlled environment. Fluid dynamic effects however, owing to their inherent complexity, have often been included using simplified models or empirical experimental correlations; flow simulations have been restricted to specific fields applications, e.g., aeronautics, and automotive engineering. The continuous growth of computing power is encouraging engineers and designers to include high-fidelity computational fluid dynamics (CFD) tools in the design and testing of new technological solutions. In spite of the widespread use of CFD, there are still a number of difficulties that prevent its use as a standard design tool. Among those, accurate predictions of turbulent flows and rapid analysis of complex systems are two of the most critical ones, which are addressed in this paper. For flows of industrial interest, usually around or inside bodies of complex shapes, the generation of a high quality grid fitted to a complex three–dimensional object can become extremely difficult and time-consuming; it can easily exceed the time needed to obtain the flow solution. In addition, the resulting meshes are typically not guaranteed to be orthogonal and this requires more complex solution algorithms. This results in a significant overhead in the per–cell operation count and, therefore, adds to the computational cost.

Moreover, in some applications, the inherent motion of the boundaries creates an additional complexity. If one or more parts of the object are in relative motion the solution

procedure requires deformating or regenerating the body fitted grid, an extremely time consuming process, which also requires the interpolation of the solution and might result in a decrease of accuracy and degradation of the conservation properties.

Therefore, it is clear that a numerical procedure capable of handling complex geometric configurations without resorting to body–fitted meshes would make a significant advance in the application of CFD to industrially relevant applications, especially in the initial design phase. The Immersed Boundary (IB) method has emerged in the last few years as a suitable candidate for such a goal.

In the IB approach the presence of a complex boundary is replaced by a time–spatially varying distribution of a forcing term which mimics the effect of the body on the flow. The main advantage of this approach is that the forcing can be prescribed on a simple regular mesh, thus retaining all the simplicity and efficiency of the methods developed in that framework. The IB method has presently become a well established approach for CFD analysis and several review papers [1, 2, 3] are now available in the literature showing its reliability and efficiency. Moreover, the method has been applied with success to compressible flows for a wide range of the Mach number [4]. In this paper we present the last developments of the IB method, namely: the solution of strongly coupled fluid/structure interaction problems and of 3D compressible flows using local mesh refinement.

2 FLUID/STRUCTURE INTERACTION

In order to obtain reliable numerical simulations of flows around complex objects with also fluid–structure interaction (like the flow inside a mechanical heart valve), four ingredients are needed: *i)* an efficient Navier–Stokes solver, *ii)* a reliable immersed boundary technique, *iii)* an accurate local force calculation and *iv)* an ODE (ordinary differential equation) solver coupled with the Navier–Stokes solver.

The first two points will only be shortly recalled here, having been already discussed elsewhere [5, 6] and they will be only shortly summarized. Here we discuss the method used to calculate the hydrodynamic loads (pressure and viscous forces exerted by the fluid on the mimicked body) and the procedure for the integration of the ODE governing the rigid body motion.

2.1 Navier–Stokes solver

The nondimensinal incompressible Navier–Stokes equations are discretized in space using second-order-accurate central differences in conservative form. The resulting semi-discrete system is discretized in time using a fractional–step method, where the viscous terms are computed implicitly and the convective terms explicitly, yielding the following equation:

$$\frac{\widehat{\mathbf{u}} - \mathbf{u^n}}{\Delta t} = -\alpha_n \nabla p^n - [\gamma_n H^n + \rho_n H^{n-1}] + \frac{\alpha_n}{2Re}\nabla^2(\widehat{\mathbf{u}} + \mathbf{u^n}) + \mathbf{f^{n+\frac{1}{2}}}, \tag{1}$$

where Δt is the time step, $\mathbf{u}$, p, and $\mathbf{f}$ are the nondimensional velocity, pressure, and body force, and Re is the Reynolds number. H contains the nonlinear terms and the large sparse matrix resulting from the implicit terms is inverted by an approximate factorization technique. According to equation (1), the momentum equation is provisionally advanced in time (predictor step) using the pressure at the previous level, giving an intermediate non–solenoidal velocity field ($\widehat{\mathbf{u}}$) which is projected onto a divergence-free vector field by a gradient correction:

$$\mathbf{u^{n+1}} = \mathbf{u^n} - \alpha_\mathbf{n}\mathbf{\Delta t}\nabla\phi, \tag{2}$$

the scalar ϕ being computed from the elliptic equation

$$\nabla^2\phi = -\frac{\nabla\cdot\widehat{\mathbf{u}}}{\alpha_n\Delta t}; \tag{3}$$

the pressure is then computed by

$$p^{n+1} = p^n + \phi - \frac{\alpha_n\Delta t}{2Re}\nabla^2\phi. \tag{4}$$

The large–banded matrix associated with the elliptic equation (3) is reduced to a penta–diagonal matrix using trigonometric expansions (FFT's) in one spatial direction; the resulting Helmholtz equations are then inverted using the FISHPACK package [7]. A hybrid low–storage third–order Runge–Kutta scheme, with coefficients α_n, γ_n, and ρ_n, in which each Δt is divided into three substeps, is used to advance the equations in time; this procedure has the advantage of requiring only a two–level storage, like common second–order schemes, while enjoying a stability limit up to $CFL = \Delta t U/\Delta x = \sqrt{3}$.

More details are given in [5, 6, 8], the last one providing also the source codes and several advanced tutorials.

2.2 Immersed boundary technique

The key idea of the immersed boundary (IB) method is the decoupling between the geometry of the problem and the computational grid. In more details, if the flow around the object $\mathbf{B}$ in figure 1 is to be computed in the rectangular domain Γ, a standard CFD approach would require the generation of a grid in the region bounded by $\partial\mathbf{B}$ and $\partial\Gamma$; a possible configuration is that of figure 1a. In the immersed boundary method, in contrast, an underlying grid is laid on Γ regardless of the shape of the object $\mathbf{B}$ which is *immersed* in the computational domain. The IB technique used in this work is based on that proposed in [6, 1]. In a preliminary step, the geometry under consideration, namely, the object B in figure 1, which is described by a closed curve in two dimensions (a closed surface in three dimensions), is overlapped onto a Cartesian (non uniform) grid. Using the ray tracing technique, the computational cells occupied entirely by the flow are tagged as *fluid* cells; those whose centres lie within the immersed body are tagged as *solid* cells; the remaining ones are finally tagged as *interface* cells. The effect of the body on the flow is

accounted for by computing at each boundary element a fictitious forcing term ($\mathbf{f}$) which enforces the correct flow boundary condition on $\partial\mathbf{B}$. The advantage of this approach is that $\mathbf{f}$ can be prescribed on the regular grid and the need of a body-conforming mesh is avoided. Let us define $\mathbf{f}(\mathbf{x}_s, t)$ as the force acting on the mimicked boundary element whose position at time t is $\mathbf{x}_s$. Since the force is dependent on the fluid velocity, which in general is unsteady, and the boundary itself could move, it turns out that both $\mathbf{f}$ and $\mathbf{x}_s$ are functions of time.

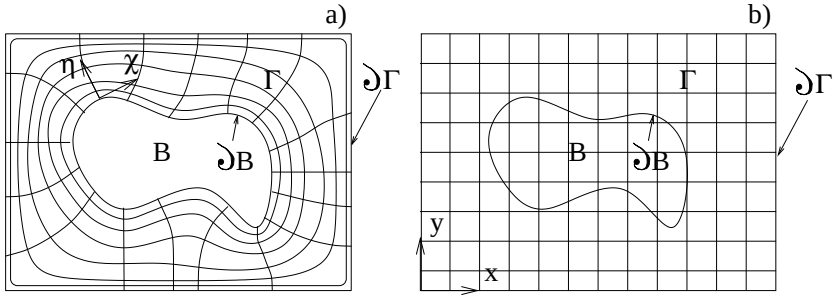

Figure 1: Sketch of the domain discretization with a body conformal mesh *a)* and with the immersed boundary method *b)*.

A convenient expression for the forcing $\mathbf{f}$ was derived by Mohd-Yusof [9]; in particular, if the momentum equation is discretized in time, we have

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \mathrm{RHS}^{n+1/2} + \mathbf{f}^{n+1/2}, \tag{5}$$

where $\mathrm{RHS}^{n+1/2}$ contains the convective and viscous terms as well as the pressure gradient. If now we ask which value of $\mathbf{f}^{n+1/2}$ will yield $\mathbf{u}^{n+1} = \mathbf{V}^{n+1}$ on the immersed boundary the answer is simply given by the equation below:

$$\mathbf{f}^{n+1/2} = -\mathrm{RHS}^{n+1/2} + \frac{\mathbf{V}^{n+1} - \mathbf{u}^n}{\Delta t}. \tag{6}$$

This forcing is direct in the sense that the desired value of velocity is enforced on the boundary without any dynamical process and does not require any free parameter. Therefore, at every time step, the boundary condition is satisfied regardless of the frequencies in the flow. In addition, the forcing in equation (6) does not require additional CPU time since it does not involve the computation of extra terms, and, when every term is computed at the appropriate time, it does not influence the stability of the time integration scheme.

As noted by Fadlun *et al.* [6], a possible problem is that the forcing $\mathbf{f}$ is applied to the provisional velocity $\hat{\mathbf{u}}$ which is afterward modified according to equation (2): while this step is mandatory in order to satisfy the continuity equation, such a correction could

change the velocity around the immersed boundary thus yielding an incorrect boundary condition. However, it has been shown that the correction of equation (2) at the immersed boundary is always small, $(\widehat{\mathbf{u}} - \mathbf{u}^{\mathbf{n+1}})/\widehat{\mathbf{u}} \approx \mathbf{10^{-3}}$–$10^{-4}$, and this error can be reduced to round–off by two or three iterations of the time advancement process, i. e., equations (1)–(4) (see Fadlun *et al.* [6], for more details). The problem was solved in a more elegant way by Kim *et al.* [10] who, by modifying the continuity equation in the nodes around the immersed boundary, added mass sources and sinks such as to cancel the correction of equation (2) at the forced grid-points.

It is noteworhty that the above problem is not related to the immersed boundary technique, but rather to its coupling with a projection method for the solution of the Navier–Stokes equations.

2.3 Boundary reconstruction

It is important to note that the above expression for the forcing term would be correct if the position of the unknowns on the grid coincided with that of the immersed boundary. This is not true in general, because it would require the boundary to lie on coordinate lines or surfaces, which is not the case for complex curvilinear geometries.

The different positions of the immersed boundary and unknown locations thus require an interpolation of the solution, often referred to as boundary reconstruction. A simple possibility is to compute the velocity value that, in a linear approximation, the point closest to the boundary would have if the boundary had the velocity $\mathbf{V}$. The corresponding forcing for the Navier–Stokes equations is simply given by the expression (6) with the imposed velocity $\overline{\mathbf{V}}$ instead of $\mathbf{V}$. This procedure is consistent with the overall second-order accuracy of the scheme. However, it is very satisfactory (exact) only if the first external points lie within the linear viscous sublayer. This is certainly true for a laminar flow or for direct numerical simulations (DNS) of turbulent flows, but it is unlikely to happen even for moderately high Reynolds number LES simulations. Possible strategies to alleviate this problem are described in [11, 3].

2.4 Moving boundaries

In the previous section we have shown that, either directly of through the forcing $\mathbf{f}$, the immersed boundary method essentially imposes the condition $\mathbf{u} = \mathbf{V}$ at the boundary of $\mathbf{B}$. Depending on the behavior of $\mathbf{V}$ and on the position of the body with respect to the grid, several additional difficulties might arise.

The simplest case is that of a fixed rigid body since it results $\mathbf{V} = 0$ on $\partial\mathbf{B}$, the computational nodes being tagged only once at the beginning of the simulation. It is noteworthy that this does not necessarily prevent the body from moving since an appropriate reference frame can still compensate for the motion of $\mathbf{B}$ with respect to the grid. This is the case of the unbaffled impeller stirred tank of Verzicco *et al.* [12] or the carangiform motion of an airfoil by Blondeaux *et al.* [13], where the Navier–Stokes equations in non

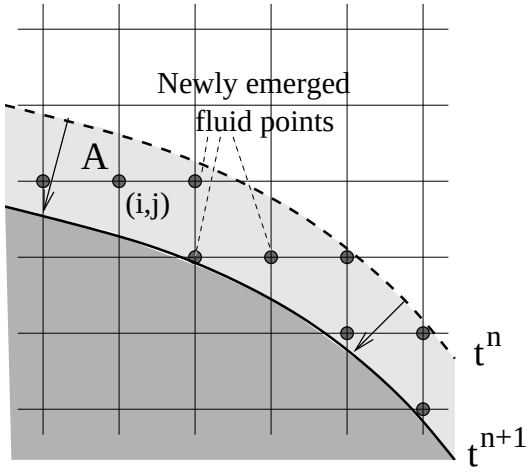

Figure 2: Illustration of the emergence of points from the solid to the fluid phase.

inertial reference frames were integrated.

The situation is different when the position of **B** changes relatively to the grid and when the solid body withdraws from the fluid, grid points inside the body emerging into the flow. The real difficulty is that some of the nodes that at the time t^n are inside the body, at the successive time step might become *fluid cells*, without having a valid time history for the correct integration of the equations. In particular, if in figure 2 the surface moves between t^n and t^{n+1} downward, there are some newly emerged *fluid* points for which the Navier–Stokes equations should be solved. For this integration to be possible, however, several terms of the Navier–Stokes equations at time t^n should be known and this is not the case, for example, for point A. Following [14, 15] the problem can be alleviated by a field extension strategy in which the interpolation procedure used in the previous section is extended not only to the fluid velocity but also to those quantities required for the integration of the equations (field reconstruction).

In particular, within the direct forcing approach [6], a newly emerged *fluid* point has a correct value of velocity due to the body velocity conditions imposed at the previous time step but an unphysical value of pressure. To overcome this problem, a particular care about the pressure at the interface points has to be taken. In this case the field extension would consists of a second-order geometrical reconstruction that involves the following steps: i) using a local search process starting from the *fluid* point closest to the boundary, find a test point, on the normal to the boundary, such that all surrounding nodes are in the fluid. ii) the value of the pressure at the test point is calculated by a standard trilinear interpolation involving the 8 surrounding points (in three–dimensions). iii) the pressure at the grid point closest to the boundary can then be computed by integrating $\partial p/\partial n = -\partial V_n/\partial t$.

2.5 Local load calculation

The computation of the surface pressure and shear stress is a key issue for non–boundary conforming formulations, especially when applied to fluid–structure interaction problems. In the present study, a linear interpolation strategy has been implemented resulting in very smooth surface pressure and wall shear stress distributions, as shown in [3].

The method starts with the geometrical description of the three-dimensional object, as a triangulated closed surface. Then, the centroid and the area of each triangle are computed and the position of the centroid is located with respect to the computational domain. Using a local search process, starting from each centroid, a probe point on the outgoing normal is selected so that all surrounding computational nodes are in the fluid. Again, this step can be iterative, adjusting the distance from the boundary until the above condition is met. The value of the pressure and velocity at the probe point are calculated by a standard trilinear interpolation involving the 8 surrounding nodes (in three dimensions) following exactly the field reconstruction procedure described in the previous section.

The pressure and velocity being known for all probe points (and at all centroids of the triangles discretizing the immersed boundary) the shear stress τ_w can be locally evaluated and the global loads computed by integrating it over the immersed surface. The whole procedure has been tested and validated versus well established results, see [3] for further details.

2.6 ODE Solver and strong coupling scheme

The equation of motion for a rigid body can be written as:

$$\mathbf{M}\ddot{\mathbf{x}}\left(t\right)+\mathbf{C}\dot{\mathbf{x}}(t)+\mathbf{K}\mathbf{x}\left(t\right)=\mathbf{f}\left[\mathbf{x}\left(\tau\right),\dot{\mathbf{x}}\left(\tau\right),\ddot{\mathbf{x}}\left(t\right)\right],\qquad\forall\tau\in0\leq\tau\leq t \tag{7}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is the damping matrix, $\mathbf{K}$ is the stiffness matrix, $\mathbf{f}$ is the vector of generalized hydrodynamic forces, and $\mathbf{x}$ is the vector of generalized structural displacements.

In the present study, a strong coupling scheme is employed, where the fluid and the structure are treated as elements of a single dynamical system, and all governing equations are integrated simultaneously in the time-domain. A fundamental complication with the application of a time-domain approach to two-way, fluid-structure interaction problems like the one described above, is that the prediction of the flow field and of the hydrodynamic loads requires the knowledge of the motion of the structure and vice-versa.

Equation (7) can be rewritten in non-dimensional form, as a system of $2n$ first-order, non-linear ODEs, n being the number of degrees-of-freedom of the structure:

$$\dot{\mathbf{y}}\left(t\right)=\mathbf{F}\left[\mathbf{y}\left(\tau\right),\dot{\mathbf{y}}\left(\tau\right)\right],\qquad\forall\tau\in0\leq\tau\leq t, \tag{8}$$

where half of the vector $\mathbf{F}$ represents generalized velocities and the other half represents generalized forces divided by the corresponding inertias. In general, the loads depend

on $\mathbf{y}$, explicitly, and on the history of the motion and the acceleration of the structure, implicitly . For this reason Hamming's 4^{th}-order predictor-corrector method is used to integrate equation (8) in time. The numerical procedure used to determine the current value of the vector $\mathbf{y}$ is:

1. Let $t_j = j\Delta t$ denote the time at the j-th time step and

$$\mathbf{y}^j = \mathbf{y}\left(t_j\right), \quad \dot{\mathbf{y}}^j = \dot{\mathbf{y}}\left(t_j\right), \quad \mathbf{F}^j = \mathbf{F}\left[\mathbf{y}\left(t_j\right)\right] \tag{9}$$

2. Compute the predicted solution, ${}^p\mathbf{y}^j$, and modify it, ${}^1\mathbf{y}^j$, using the local truncation error, $\mathbf{e}^{j-1}$, from the previous time step:

$${}^p\mathbf{y}^j = \mathbf{y}^{j-4} + \frac{4}{3}\Delta t \, \left(2\, \mathbf{F}^{j-1} - \mathbf{F}^{j-2} + 2\, \mathbf{F}^{j-3}\right), \quad {}^1\mathbf{y}^j = {}^p\mathbf{y}^j + \frac{112}{9}\, \mathbf{e}^{j-1} \tag{10}$$

3. Correct the modified, predicted solution as

$${}^{k+1}\mathbf{y}^j = \frac{1}{8}\left[9\, \mathbf{y}^{j-1} - \mathbf{y}^{j-3} + 3\Delta t \, \left({}^k\mathbf{F}^j + 2\, \mathbf{F}^{j-1} - \mathbf{F}^{j-2}\right)\right] \tag{11}$$

   where ${}^k\mathbf{F}^j = \mathbf{F}\left({}^k\mathbf{y}^j\right)$ and k is the iteration index. Convergence is achieved when the iteration error, $e^j = \left\| {}^{k+1}\mathbf{y}^j - {}^k\mathbf{y}^j \right\|_\infty$ is less than a prescribed tolerance ϵ.

4. Compute the local truncation error, $\mathbf{e}^j$, and the final solution, $\mathbf{y}^j$, at step j,

$$\mathbf{e}^j = \frac{9}{121}\left({}^{k+1}\mathbf{y}^j - {}^p\mathbf{y}^j\right), \quad \mathbf{y}^j = {}^{k+1}\mathbf{y}^j - \mathbf{e}^j \tag{12}$$

   and proceed to the next time step.

It is noteworthy that the procedure above requires information from four previous time steps. Therefore, the Euler, Adams-Moulton, and Adams-Bashforth methods were used, starting from the initial conditions, to provide such information.

The fluid solver is coupled with the structural solver as follows: i) Find the predicted location and velocity of the body using equation (10). ii) Find the predicted fluid velocity and pressure fields using the Navier–Stokes equations with the boundary conditions provided by step 1, and compute the resulting loads on the structure. iii) Compute the new location and velocity of the body using equation (11). iv) Check for convergence. If e^j is greater than the prescribed tolerance, ϵ, repeat steps 2 to 4. If convergence is achieved, find the final position and velocity of the body using equation (12), and the final fluid pressure and velocity fields from the Navier–Stokes equations. In all computations reported in this study, a tolerance of $\epsilon = 10^{-4}$ was used. The number of iterations required for convergence at each time step varied from 1 to 4, depending on the stiffness of the problem.

It is noteworthy that for the present application (the dynamics of a bileaflet mechanical heart valve) equation (7) has been solved with the matrices $\mathbf{C}$ and $\mathbf{K}$ set to zero. This makes the problem much stiffer, since the motion of the leaflets is not smoothed by friction or elastic bounds, leading to impulsive accelerations of the rigid bodies that need extremely small time steps to be tracked accurately.

2.7 Heart valve dynamics at Re=3200

In this study we have considered the flow in a St. Jude Medical Standard aortic bileaflet valve subjected to a pulsatile flow. The physiological pulsatile inflow was synthesized at the Georgia Tech by Yoganathan (*A.P. Yoganathan, personal communication*) using a left heart chamber simulator. The flow loop consisted of a pneumatic pulsatile system, a reservoir, an aortic valve-mounting chamber, a mitral valve-mounting chamber, resistance and compliance sections, a flow transducer, as well as ventricular and aortic pressure transducers. The compliance and resistance sections of the loop were adjusted to maintain the following conditions: heart rate of 70 beats/min, systolic duration of 300 ms, peak systolic flow rate of 25 ℓ/min, cardiac output of 5 ℓ/min, and aortic pressure of 80-140 mmHg. The resulting flow and pressure waveforms are shown in figure 3a.

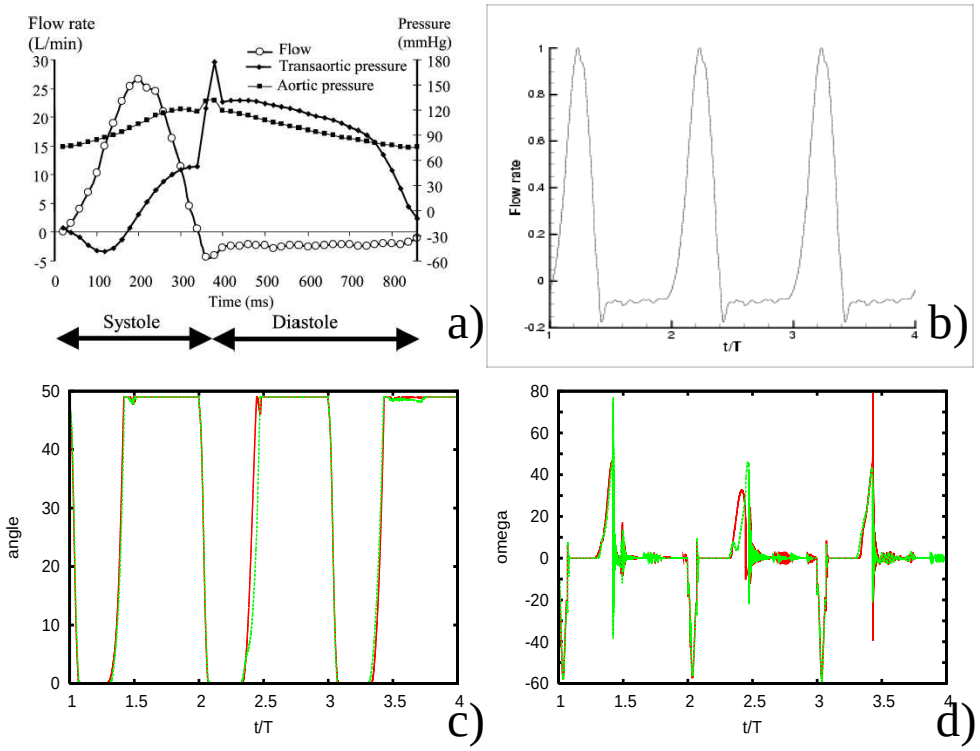

Figure 3: *a)* Aortic flow and pressure waveforms. *b)* Physiological flow-rate, $Re_{peak} = 3200$. *c)* Relative angular displacement: —— Upper leaflet, - - - - Lower leaflet. *d)* Angular velocity: —— Upper leaflet, - - - - Lower leaflet.

Starting from the inflow waveform of figure 3a, the peak bulk velocity and all of the nondimensional parameters were computed. In particular, considering the intake diameter $D = 0.0254\ m$ and the peak bulk velocity $U = 0.875\ m/s$, the peak Reynolds number is $Re = 6400$ (with a kinematic viscosity of $3.5 \cdot 10^{-6}\ m^2/s$). For computational economy, the direct numerical simulation was performed at half of the physiological Reynolds number, $Re = 3200$, using a mesh with $151x131x241$ nodes, in the azimuthal, radial and axial directions, respectively (a simulation at $Re = 6400$ on a twice finer grid is progress). The inflow waveform was scaled in order to have $U_{b,peak} = 1$ and the nondimensional period $T = 15$; figure 3b shows the inflow waveform for the three simulated cardiac cycles.

The leaflet dynamics is shown in figures 3c–d, where the relative angular displacement and the angular velocity of the leaflets are plotted versus time. It can be noticed that the motion of the leaflets is not always symmetric, especially during the closing phase, as observed also in experiments (*G.P. Romano, personal communication*). These asymmetries have a stochastic behavior that cannot be determined a priori and they are mostly due to the turbulent nature of the decelerating flow. In contrast, during the opening phase, the flow accelerates, turbulence is mostly suppressed, and the asymmetry in the leaflet motion disappears. This feature can be best appreciated from figure 3d, which shows the time evolution of the angular velocities of the leaflets.

The instantaneous streamlines and the vorticity for this simulation are given in figures 4 and 5. During the opening of the valve (figure 4), strong shear layers develop from the tips of the leaflets, since the gaps between the leaflet tips and the housing are smaller than in the case of a fully open valve. A strong shear layer is also generated from the central gap between the two leaflets. Although the opening time is short, the strong shear layers roll up quickly into two semitoroidal large vortices which move towards the sinus region. This allows a relatively large motion of the vortices attached to the downstream part of the valve. In the fully open position, a strong shedding from the leaflets is shown in figure 4d and 5a and small vortices are developed and shed. Moreover, the shear layers from the housing roll up and generate small vortices, which interact with the recirculation zone present in the sinus region. The closing phase is characterized by a different dynamics; in fact, as shown in figure 5c, a large vortex behind each leaflet is released and the leaflets reach the geometrically constrained fully closed position, the rebound from the housing produces the squeezing of the fluid from the central gap between the two leaflets evidenced by a mushroom–like structure. During the regurgitation phase (negative value of inflow rate) strong leakages through the small gaps between the leaflets and the housing are shown in figure 4a, which last until the accelerating phase starts and the valve opens again.

A more detailed analysis of the flow and comparisons with ongoing experiments are in progress. These preliminary results, however, already show how the flow (blood) dynamics is severely altered by the presence of a mechanical valve, which initiates the phenomenon of hemolysis and coagulation cascade; todate, prevention of these complications requires lifelong anticoagulation therapy.

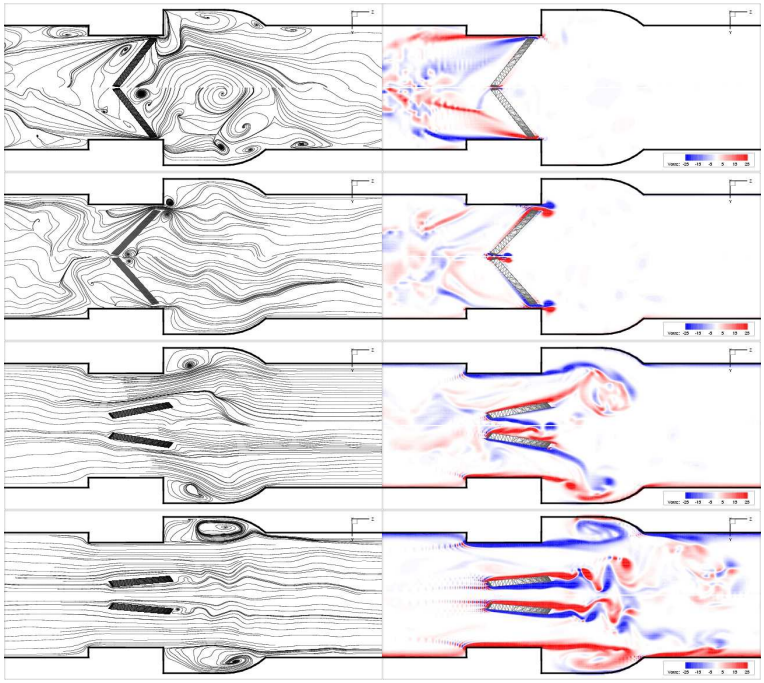

Figure 4: Time sequence of the streamlines (*left*) and azimuthal vorticity contours (*right*) for the opening phase. a) t/T=3.00, b) t/T=3.07, c) t/T=3.14, d) t/T=3.21.

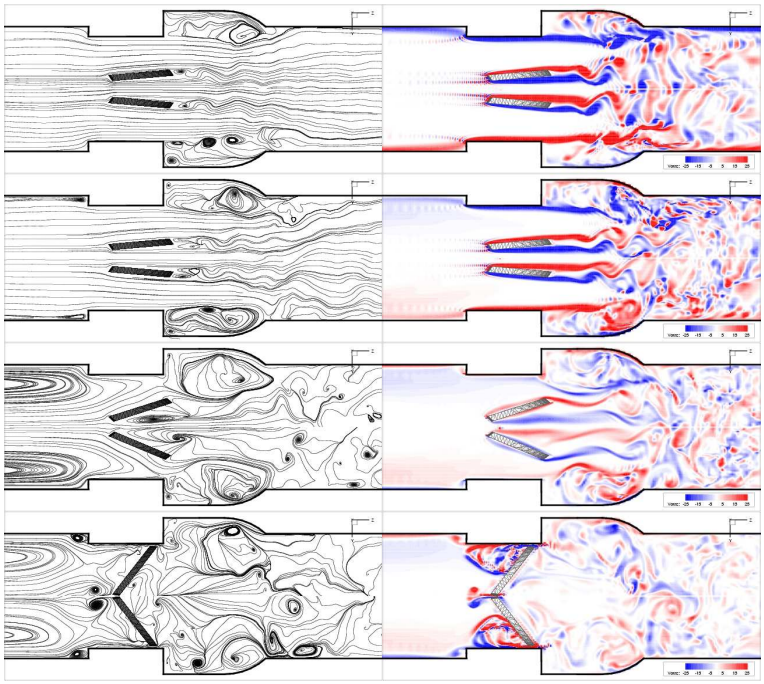

Figure 5: Time sequence of the streamlines (*left*) and azimuthal vorticity contours (*right*) for the closing phase. a) t/T=3.28, b) t/T=3.34, c) t/T=3.40, d) t/T=3.45.

3 PRECONDITIONED SOLVER FOR COMPRESSIBLE FLOWS

3.1 Governing equations and numerical method

In this work, the Reynolds Averaged Navier–Stokes (RANS) equations, written in terms of Favre mass-averaged quantities, are solved in conjunction with the low-Reynolds number $k-\omega$ turbulence model [16]. Such equations are given in compact form as

$$\mathbf{\Gamma}\frac{\partial Q_v}{\partial \tau}+\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}-\frac{\partial E_v}{\partial x}-\frac{\partial F_v}{\partial y}-\frac{\partial G_v}{\partial z}=D, \tag{13}$$

where Q is the conservative variable vector, E, F, G and E_v, F_v, G_v indicate the inviscid and viscous fluxes, respectively, and D is the vector of the source terms. A pseudo-time derivative for the primitive variable vector Q_v has been added to the left-hand-side in order to use a time marching approach for both steady state and unsteady problems. the preconditioning matrix, $\mathbf{\Gamma}$, proposed in [17] is finally used to premultiply the pseudo-time derivative in order to obtain an efficient iteration at all values of the Mach number.
Discretizing equation (13) by an Euler implicit scheme in the pseudo-time and approximating the physical-time derivative by a second-order-accurate three-point backward difference, the following equation in delta form is obtained:

$$\begin{aligned}\mathbf{S}\Bigg[\mathbf{I}+\Delta\tau\mathbf{M}_x\frac{\partial}{\partial x}\left(\mathbf{\Lambda}_x-R_x\mathbf{I}\frac{\partial}{\partial x}\right)\mathbf{M}_x^{-1}+\Delta\tau\mathbf{M}_y\frac{\partial}{\partial y}\left(\mathbf{\Lambda}_y-R_y\mathbf{I}\frac{\partial}{\partial y}\right)\mathbf{M}_y^{-1}\\ +\Delta\tau\mathbf{M}_z\frac{\partial}{\partial z}\left(\mathbf{\Lambda}_z-R_z\mathbf{I}\frac{\partial}{\partial z}\right)\mathbf{M}_z^{-1}\Bigg]\Delta Q_v=-\Delta\tau\left[\frac{3Q^r-4Q^n+Q^{n-1}}{2\Delta t}+\mathcal{R}^r\right].\end{aligned} \tag{14}$$

where r and $\Delta\tau$ indicate the pseudo-time level and step, whereas n and Δt indicate the physical-time level and step, respectively. The steady residual is given as:

$$\mathcal{R}^r=\frac{\partial(E^r-E_v^r)}{\partial x}+\frac{\partial(F^r-F_v^r)}{\partial y}+\frac{\partial(G^r-G_v^r)}{\partial z}-D^r, \tag{15}$$

and the delta unknowns to be annihilated at each pseudo-time level are

$$\Delta Q_v={Q_v}^{r+1}-{Q_v}^{r}. \tag{16}$$

The pseudo- and physical-time terms are grouped together into a new term $\mathbf{S}$,

$$\mathbf{S}=\mathbf{\Gamma}+\frac{3}{2}\frac{\Delta\tau}{\Delta t}\mathbf{P},\quad \mathbf{P}=\frac{\partial Q}{\partial Q_v}, \tag{17}$$

which has been factored out of the LHS in equation (14). The diagonalization procedure of Pulliam and Chausee [18] has been employed, so that

$$\mathbf{S}^{-1}\mathbf{A}_v=\mathbf{M}_x\mathbf{\Lambda}_x\mathbf{M}_x^{-1},\quad \mathbf{S}^{-1}\mathbf{B}_v=\mathbf{M}_y\mathbf{\Lambda}_y\mathbf{M}_y^{-1},\quad \mathbf{S}^{-1}\mathbf{C}_v=\mathbf{M}_z\mathbf{\Lambda}_z\mathbf{M}_z^{-1}, \tag{18}$$

where $\mathbf{A}_v = \partial E/\partial Q_v$, $\mathbf{B}_v = \partial F/\partial Q_v$, $\mathbf{C}_v = \partial G/\partial Q_v$. Equation (14) is factorized and solved by a BiCGStab method [19]. A collocated cell-centred finite volume space discretization is used. The convective terms at the RHS are discretized using either an upwind flux-difference-splitting scheme with first-, second- or third-order accuracy, or a second-order-accurate centred scheme. When computing flows with shocks, a total variation diminishing approach is employed using the minmod limiter function and either a second- or a third-order accurate upwind scheme. The viscous terms are discretized by second-order-accurate centred differences. The LHS convective term is always discretized using a first-order upwind scheme, according to a deferred-correction approach. Finally, the boundary conditions are treated explicitly.

3.2 Data structure

Local grid refinement (LGR) allows for efficient clustering of cells close to the immersed boundary. The basic idea was recently introduced by Durbin and Iaccarino [20] for a finite difference discretization and extended to a finite volume formulation by Iaccarino *et al.* [21]. The following description of the algorithm refers to two dimensions, the extension to three-dimensions being straightforward. An auxiliary structured grid is employed to handle the data structure of the semi-structured locally refined grid (shown in figure 6). The auxiliary grid covers the whole computational domain employing the finest mesh size on the semi-structured grid. Therefore, each cell of the semi-structured grid is bounded by the lines passing throught the vertices (i, j) and $(i + \Delta i, j + \Delta j)$, see figure 6, where the indeces $i = 0, \ldots, N_i$ and $j = 0, \ldots, N_j$ refer to the auxiliary (finest) grid and $\Delta i \geq 1$, $\Delta j \geq 1$ depend on (i, j). The major advantage of this approach with respect to classical OCTREE-based and fully-unstructured algorithms lies in the economy and flexibility of storing and retrieving connectivity informations, due to the presence of the auxiliary grid. In particular, using an $N_i \times N_j$ auxiliary grid, only $N < N_i \times N_j$ cells belonging to the semi-structured grid are defined using the two couple of indices (i, j) and $(\Delta i, \Delta j)$, with a total memory requirement of $4\,N$ integers. In addition, an array of integers, $ID(i, j)$ is needed to store the correspondence between the cells of the auxiliary and semi-structured grids. All of the cells of the auxiliary grid not employed in the actual mesh, namely, those included in the range $[i : i + \Delta i - 1]$ and $[j : j + \Delta j - 1]$, are tagged using the same cell number. The total storage required for allocating $ID(i, j)$ is, therefore, $N_i \times N_j$ integers. The connectivity information for each cell is retrieved by querying the array $ID(i, j)$.

3.3 Semi-structured grid generation

Within the IB method, as described in section 2.2, the generation of semi-structured grids is carried out by firstly creating the auxiliary (fine) grid and coarsening it in the regions away from the immersed boundary. The advantage of this approach is that all the cell-tagging (ray-tracing) needed for using the IB technique can be performed on a structured grid, taking full advantage of the alignment of the cell centers and the grid

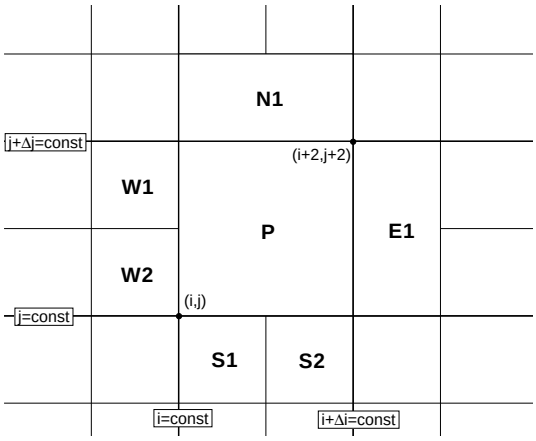

Figure 6: LGR grid showing a cell P and its neighbors.

nodes. In the present implementation, starting from an auxiliary grid with uniform mesh size, a structured grid is generated by recursively halving the mesh size at the immersed boundary region, until an assigned target value is reached. This automatic refinement is based on the following strategy. A tag funcion, generated using the ray tracing technique as mentioned before, is used to mark the cells inside and outside the immersed body: an integer value ± 1 is assigned to *fluid* and *solid* cells, respectively. The gradient of this function is different from zero only at the immersed boundary and depends on the local grid size. The components of this gradient in the x and y directions are used to select the rows of cells to be refined. The grid is refined until a user specified resolution is achieved at the boundary. A smoothing function can be applied on the ± 1 tagging function to obtain a smeared interface allowing a smoother transition between the coarse and the refined regions. By this procedure, the auxiliary grid is obtained. Then, starting from such a grid, the semi-structured mesh is obtained by coarsening the cells in the regions far from the boundaries until the maximum prescribed cell-size is achieved (see figure 7(a)). In addition to such an automatic refinement, it is possible to define regions of the computational domains to be refined, selecting the resolution of the refinement, like the wake region in figure 7(b) and, finally, it is possible to refine on *void* surfaces, namely, surfaces without *solid* or interface points, like the bow-shock shown in figure 7(c).

3.4 Interpolation method at the immersed surfaces

The application of the boundary conditions at the immersed surface is treated explicitly, by assigning the values of the variables at the interface cells. At *solid* cells, the velocity components are set to zero and, in the case of isothermal surfaces, the temperature is set equal to the wall value. At the interface cells, the nearby wall is modeled with an off wall boundary condition which consists of an interpolation of the flow variables, using the computed values of the surrounding *fluid* cells and the imposed values at the wall. For each interface cell it is possible to find N_{nbr} neighbouring *fluid* cells and N_{ib} intersections

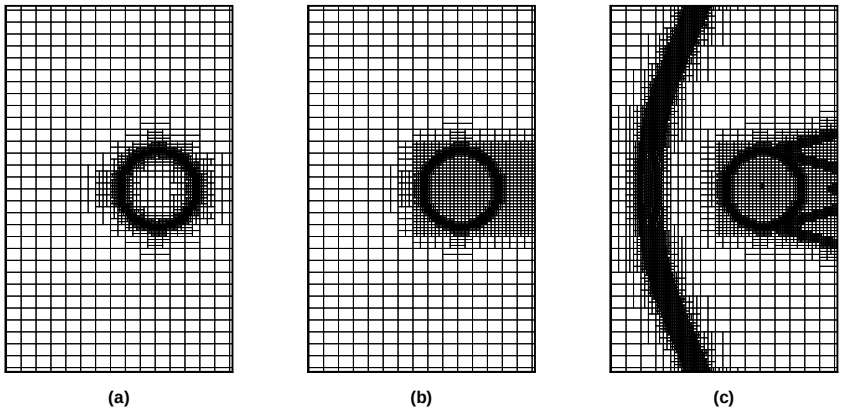

Figure 7: Different steps of refinement: (a) automatic on the immersed boundary; (b) specified window (wake); (c) external surface (shock).

of the faces of each cell with the immersed boundary. For the velocity components and the temperature in the case of isothermal surface, the following interpolation formula is used:

$$\phi_{int} = \sum_{i}^{N_{nbr}} \frac{\alpha_i}{q}\phi_i + \sum_{j}^{N_{ib}} \frac{\beta_j}{q}\phi_{j,wall}, \tag{19}$$

where $\phi_{j,wall}$ is the value of the flow variable to be imposed at the immersed surface,

$$q = \sum_{i}^{N_{nbr}} \alpha_i + \sum_{j}^{N_{ib}} \beta_j, \tag{20}$$

and α_i and β_j are the inverse distances between the surrounding cell centers and the interface cell center and between the wall intersections and the interface cell center, respectively. It can be shown that in the one-dimensional case, this procedure coincides with the linear interpolation scheme used in [6, 4]. The pressure gradient and the temperature gradient, in case of an adiabatic surface, along the normal to the immersed surface, are set to zero by assigning the corresponding flow field value at the interface cell.

3.5 Results

3.5.1 Unsteady flow past a heated circular cylinder

The unsteady two-dimensional low-Mach-number flow past a heated circular cylinder has been chosen in order to validate both the unsteady terms and the correct implementation of the energy equations, since experimental [22] and numerical [23] investigations indicate that the temperature fields has a significant influence on the flow pattern, expecially when the ratio between the cylinder wall temperature T_w and the free-stream one

T_∞, $T^* = T_w/T_\infty$ exceeds 1.1. It has been found that, for a given Re_∞, the vortex shedding frequency, f, and thus the Strouhal number $St = fD/U_\infty$, decreases for increasing values of T^*.
The computational domain has the inlet and outlet boundaries located at $x_i = -10D$ and $x_o = 40D$, and the far-field boundaries located at $y_w = \pm 15D$, the origin of the box coinciding with the centre of the cylinder. Standard characteristic boundary conditions have been imposed at inlet and outlet points, whereas free-shear wall boundary conditions are imposed at the far-field points. For such a low Re problem, centred differences are employed. Computations have been performed using a semi-structured, non uniform mesh, with 41509 cells and 293647 faces. The grid is highly refined on the cylinder surface, in order to solve the thermal boundary layer, and inside a box surrounding the cylinder and the wake, so as to obtain a satisfactory resolution of the shedding phenomenon. The auxiliary mesh is composed of 796×379 cells. The physical time step has been chosen in order to have about 500 steps per period; about 250 inner iterations are needed to reduce the residual to 10^{-6} at each time step. Figure 8 shows the computed values of the Strouhal number, St, for $Re_\infty = 100, 120, 140$ and $T^* = 1.0, 1.1, 1.5, 1.8$, as well as the experimental results provided by Wang *et al.* [22] and Sabanca & Durst [23]: a very good agreement is obtained; moreover the present results are comparable with those obtained using the uniform auxiliary grid [4].

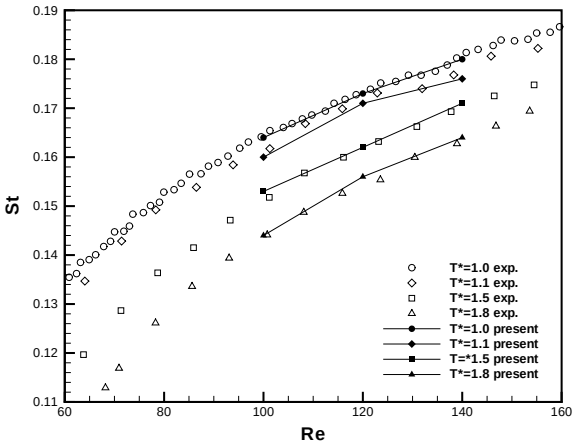

Figure 8: Strouhal number vs Reynolds number for the usteady flow past a heated circular cylinder.

3.5.2 Incompressible flow past a sphere

The incompressible flow past a sphere has been computed to test the three dimensional flow solver and the immersed boundary method. A single value of the free-stream Mach

number, $M_\infty = 0.03$, and four values of the Reynolds number (based on the sphere diameter, D, the free-stream velocity, U_∞), namely, 40, 60, 80, and 100, have been considered. The computational domain is a box; to reduce the influence of boundary conditions on the solution, the inlet and outlet boundary planes are located at $x_i = -40D$ and $x_o = 80D$ and the far-field boundaries are located at $y_w = \pm 40D$ and $z_w = \pm 40D$, the origin of the box coinciding with the centre of the sphere. Standard characteristic boundary conditions have been imposed at inlet and outlet points, whereas free-shear wall boundary conditions are imposed at the far-field boundary points. The numerical results are obtained using a 3rd-order accurate upwind spatial accuracy. Computations have been performed using a semi-structured mesh with 231031 cells and 2182547 faces. The grid is refined around the sphere surface, so as to have a good resolution of the boundary layer, and inside a box surrounding the sphere and the wake, in order to describe accurately the separation and wake regions. The auxiliary mesh contains $446 \times 199 \times 185$ cells. A local view of the mesh is given in figure 9. According to Batchelor [24], the flow around a sphere does

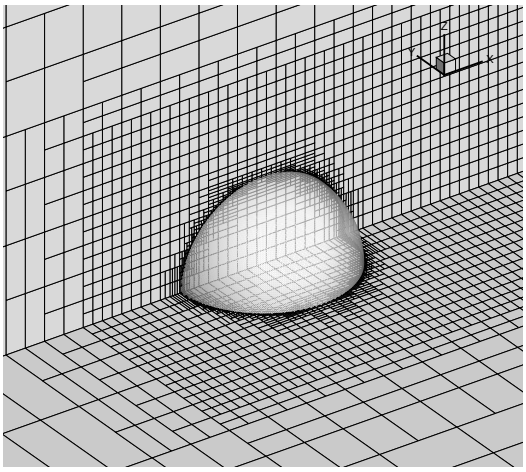

Figure 9: Local view of the grid.

not separate up to $Re \simeq 24$, and for increasing Reynolds number, the axial length of the separation bubble grows linearly up to $Re \simeq 100$, see figures 10 and 11, which provide the present results and experimental data for the length of the separation bubble and for the drag coefficient.

3.5.3 Supersonic flow past a circular cylinder

The steady turbulent supersonic flow past a circular cylinder has been considered as a suitable test case to validate the method for compressible flows with shocks. The flow with $M_\infty = 1.7$ and $Re_\infty = 2 \times 10^5$ has been computed, the inlet values of the turbulence kinetic energy and specific dissipation rate being $k/U_\infty^2 = 0.0009$ and $\omega D/U_\infty = 40$, respectively. For the considered values of M_∞, a bow shock is formed upstream of the

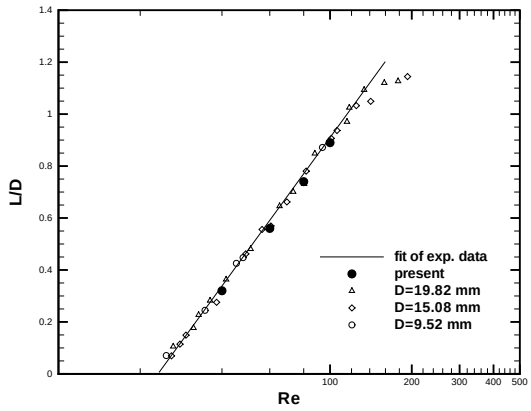

Figure 10: Length of the separation bubble vs. Re; experimental data from [24].

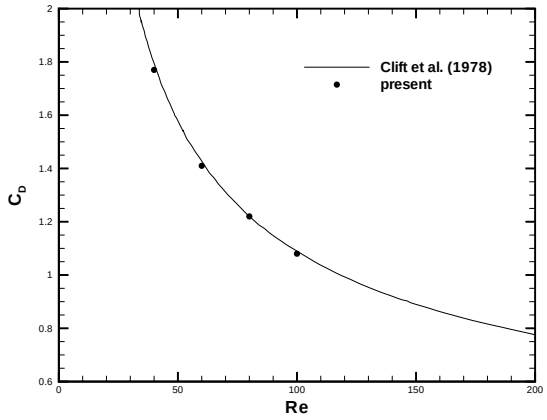

Figure 11: Drag coefficient vs. Re; eperimental data from [25].

cylinder; the subsonic flow behind the shock close to the cylinder accelerates along its surface forming a supersonic-flow region, enveloping the subsonic recirculation region behind the cylinder, and two symmetric tail shocks are formed at the end of the separation region. Results have been obtained using a rectangular computational domain with dimensions $[-10\,D; 15\,D] \times [-10\,D; 10\,D]$, D being the diameter of the cylinder centred at the origin. Standard characteristic boundary conditions have been imposed at the inlet and outlet surfaces and free-shear wall boundary conditions are imposed at the far-field boundaries. Numerical results are obtained using the third-order-accurate upwind scheme. Simulations on a coarse grid have been performed at first to locate the position of the shocks, approximatively. Based on such a solution, a new grid has been generated using the LGR at shock regions. The final semi-structured grid has 75556 cells and 545699 faces, corresponding to an auxiliary 1805×2159 mesh. A local view of the grid is shown in figure 12. The Mach number contours are given in figure 13, showing that a clear description of the shocks and of the wake are obtained, thanks to the local grid refinement in those regions. The computed separation angle, measured clockwise from the leading edge, is equal to 113° and agrees well with the corresponding experimental datum, 112° [26]. Moreover, the computed and experimental drag coefficients are equal to 1.41 and 1.43, respectively. Finally, the computed pressure coefficient distribution along the surface of the cylinder is provided in figure 14 together with the experimental data of [26]. All numerical results agree very well with the numerical solutions obtained using the same numerical method and a body-fitted grid [4] and reasonably well with the experimental.

4 CONCLUSIONS

In this paper the last developments of the immersed boundary method have been presented, namely, the solution of strongly coupled fluid/structure interaction problems

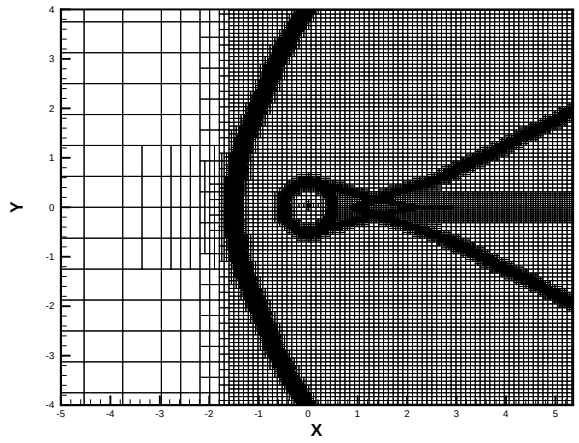

Figure 12: Local view of the grid for the case $M_\infty = 1.7$.

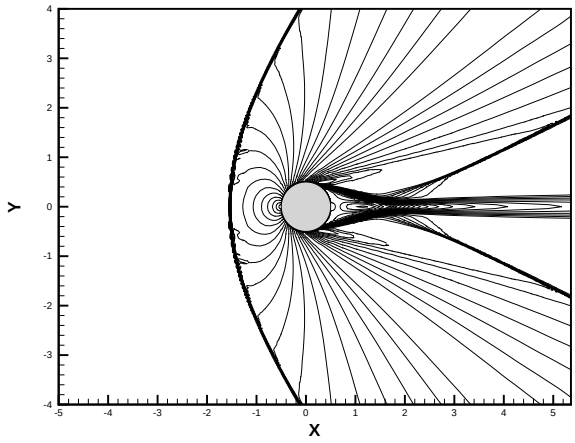

Figure 13: Supersonic flow past a circular cylinder at $M_\infty = 1.7$: Mach number contours ($\Delta M = 0.08$).

and of compressible flows using local mesh refinement. In all cases, the comparison of the numerical results with experimental data demonstrates the effectiveness and versatility of this approach.

REFERENCES

[1] Iaccarino G, Verzicco R. 2003. Immersed boundary technique for turbulent flow simulations. *Appl. Mech. Rev.* 56:331–347.

[2] R. Mittal and G. Iaccarino. Immersed Boundary Methods. *Annu. Rev. Fluid Mech.* 37:239–261, 2005.

[3] Cristallo A, Verzicco R. 2006. Combined Immersed Boundary/Large–Eddy–Simulations of Incompressible Three Dimensional Complex Flows. *Flow Turb. and Comb.* to appear.

[4] P. De Palma, M. de Tullio, G. Pascazio, M. Napolitano An immersed boundary method for compressible viscous flows *Comput. Fluids*, to appear, 2006.

[5] R. Verzicco and P. Orlandi. A finite difference scheme for three–dimensional incompressible flows in cylindrical coordinates *J. Comp. Phys.*, 123:402–413, 1996.

[6] E. A. Fadlun, R. Verzicco, P. Orlandi and J. Mohd–Yusof. Combined immersed–boundary/finite–difference methods for three–dimensional complex flow simulations. *J. Comp. Phys.*, 161:35–60, 2000.

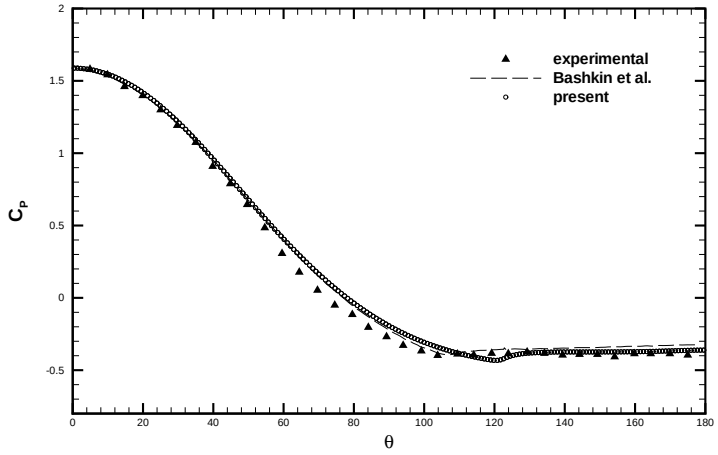

Figure 14: Pressure coefficient distribution along the surface of the cylinder: comparison between experimental and numerical results for $M_\infty = 1.7$.

[7] P. N. Swartzrauber A direct method for the discrete solution of separable elliptic equations. *SIAM J. Numer. Anal.*, 11:1136–1150, 1974.

[8] P. Orlandi. Fluid flow phenomena: a numerical toolkit. Kluwer Acad. Pub., pp. 355.

[9] Mohd-Yosuf J. 1997. Combined immersed boundary/B-spline methods for simulation of flow in complex geometries. *Annual Research Briefs, Center for Turbulence Research*. 317–328.

[10] J. Kim, D. Kim and H. Choi. An Immersed–Boundary finite–volume method for simulation of flow in complex geometries. *J. Comput. Phys.*, 171:132–150, 2001.

[11] S. Majumdar, G. Iaccarino and P. A. Durbin. RANS solver with adaptive structured non–conforming grids. *Annual Res. Briefs*, Center for Turbulence Research, 353–364, 2001.

[12] R. Verzicco, G. Iaccarino, M. Fatica & P. Orlandi. Flow in an Impeller Stirred Tank Using an Immersed Boundary Technique. *AIChE Journal*, 50(6):1109–1118, 2004.

[13] P. Blondeaux, F. Fornarelli, L. Guglielmini, M.S. Triantafyllou and R. Verzicco. 2005. Numerical experiments on flapping foils mimicking fish-like locomotion. *Phys. of Fluids*, **17**, 113601.

[14] J. Yang, S. Preidikman & E. Balaras. A strong coupling scheme for fluid–structure interaction problems in viscous incompressible flows. In M. Papadrakakis, E. Oñate

and B. Schreffler, editors, *Proceedings of Int. Conf. on Comput. Methods for Coupled Problems in Sci. and Eng.*, 2005.

[15] J. Yang & E. Balaras. An embedded–boundary formulation for large–eddy simulation of turbulent flows interacting with moving boundaries. *J. Comp. Phys.* to appear, 2006.

[16] D. C. Wilcox. Turbulence models for CFD. DCW Industries, Inc., 1998.

[17] C. L. Merkle Preconditioning methods for viscous flow calculations In M. Hafez and K. Oshima editors, *Computational Fluid Dynamics Review 1995*, 419–436, 1995.

[18] T. H. Pulliam & D. S. Chaussee A diagonal form of an implicit factorization algorithm. *J. Comp. Phys.*, 39:347, 1981.

[19] H.A. van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM J.Sci.Statist.Comput.*, 13:631, 1992.

[20] P. A. Durbin & G. Iaccarino Adaptive grid refinement for structured grids. *J. Comp. Phys.*, 128:110, 2002.

[21] G. Iaccarino, G. Kalitzin, P. Moin and B. Khalighi Local grid refinement for an immersed boundary RANS solver Paper AIAA-2004-0586, 2004.

[22] A.-B. Wang, Z. Trávníĉek, K.-C. Chia On the relationship of effective Reynolds number and Strouhal number for the laminar vortex shedding of a heated circular cylinder. *Phys. Fluids*, 12:1401, 2000.

[23] M. Sabanca and F. Durst Flows past a tiny circular cylinder at high temperature ratios and slight compressible effects on the vortex shedding. *Phys. Fluids*, 15:1821, 2003.

[24] G. K. Batchelor An Introduction to Fluid Mechanics. Cambridge Univ. Press, 1967.

[25] R. Clift, J. R. Grace, M.E. Weber Bubbles, Drops and Particles. Academic Press, 1978.

[26] V. A. Bashkin, A. V. Vaganov, I. V. Egorov, D. V. Ivanov, G. A. Ignatova Comparison of calculated and experimental data on supersonic flow past a circular cylinder. *Fluid Dynamics*, 37:473, 2002.