Journal of Computational Physics **161,** 35–60 (2000)
doi:10.1006/jcph.2000.6484, available online at http://www.idealibrary.com on IDEAL®

Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations

E. A. Fadlun,*[,1] R. Verzicco,† P. Orlandi,* and J. Mohd-Yusof‡

*Dipartimento di Meccanica e Aeronautica, Università di Roma, "La Sapienza," via Eudossiana 18, 00184 Rome, Italy; †Politecnico di Bari, Istituto di Macchine ed Energetica, Via Re David 200, 70125, Bari, Italy; ‡Los Alamos National Laboratory, P.O. Box 1663 IGPP, MS C305, Los Alamos, New Mexico 87545

Received April 15, 1999; revised February 3, 2000

A second-order accurate, highly efficient method is developed for simulating unsteady three-dimensional incompressible flows in complex geometries. This is achieved by using boundary body forces that allow the imposition of the boundary conditions on a given surface not coinciding with the computational grid. The governing equations, therefore, can be discretized and solved on a regular mesh thus retaining the advantages and the efficiency of the standard solution procedures. Two different forcings are tested showing that while the quality of the results is essentially the same in both cases, the efficiency of the calculation strongly depends on the particular expression. A major issue is the interpolation of the forcing over the grid that determines the accuracy of the scheme; this ranges from zeroth-order for the most commonly used interpolations up to second-order for an *ad hoc* velocity interpolation. The present scheme has been used to simulate several flows whose results have been validated by experiments and other results available in the literature. Finally in the last example we show the flow inside an IC piston/cylinder assembly at high Reynolds number; to our knowledge this is the first example in which the immersed boundary technique is applied to a full three-dimensional complex flow with moving boundaries and with a Reynolds number high enough to require a subgrid-scale turbulence model.

Key Words: 3D complex flows; immersed boundaries; finite differences.

1. INTRODUCTION

The continuous growth of computer power strongly encourages engineers to rely on computational fluid dynamics for the design and testing of new technological solutions.

[1] Corresponding author: E. A. Fadlun, Dipartimento di Meccanica e Aeronautica, Via Eudossiana 18, 00184 Rome, Italy. E-mail: fadlun@orlandisun.ing.uniromal.it.

0021-9991/00 $35.00

Numerical simulations allow analysis of phenomena without resorting to expensive prototypes and difficult experimental measurements. On the other hand, while simple geometries discretized by regular grids are efficiently handled by currently available codes and hardware, complex geometry flows requiring body-fitted curvilinear or unstructured meshes are still challenging problems for todays computers.

In addition, in most industrial applications, geometrical complexity is combined with moving boundaries and high Reynolds numbers which considerably increase the computational difficulties since they require, respectively, regeneration or deformation of the grid and turbulence modeling. As a result, engineering flow simulations have large computational overhead and low accuracy owing to a large number of operations per node and high storage requirements in combination with low-order dissipative spatial discretization. Given the finite memory and speed of computers, these simulations are very expensive and time consuming, with discretizations that are generally limited to a maximum of 100^3 nodes.

In view of these difficulties it is clear that an alternative numerical procedure that can cope with the flow complexity but at the same time retain the accuracy and high efficiency of the simulations performed on fixed regular grids would represent a significant advance in the study of industrial flows.

One possibility for the solution of this problem is the introduction of a body-force field $\mathbf{f}$ such that a desired velocity distribution $\mathbf{V}$ can be assigned over a boundary S. In other words, we can add to the Navier–Stokes equations the body-force $\mathbf{f}$ and solve for $\mathbf{u}$ from

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{uu}) = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \qquad \nabla \cdot \mathbf{u} = 0. \tag{1}$$

In principle there are no restrictions for the velocity distribution $\mathbf{V}$ and for the shape and motion of S; therefore a wide variety of boundary conditions can be imposed. The main advantage of this approach is that $\mathbf{f}$ can be prescribed on a regular mesh (Fig. 1) so that the accuracy and efficiency of the solution procedure on simple grids are maintained.

Indeed this idea is not new since it has already been pursued by many researchers in the last three decades. Peskin [1, 2] reports at the beginning of the seventies simulations of the blood flow in the mitral valve and in the heart assuming a very low Reynolds number and two-dimensional flow. Three-dimensional heart flows were considered successively by McQueen and Peskin [3, 4] that included also the contractile and elastic nature of the boundary. In the above papers, the motion of the boundary was determined by the flow and only the forces occurring between boundary elements were known. Within this framework, solid boundaries were modeled by elements linked by very rigid springs even if the computation

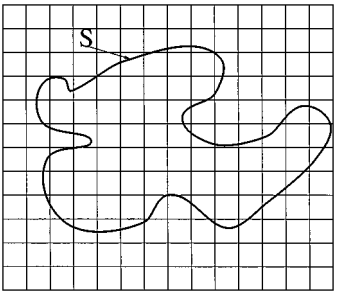

FIG. 1. Sketch of the immersed boundary.

of the force system remained as complex as in the case of elastic boundaries. In contrast if the boundary configuration is known the computation of **f** becomes much simpler since for every boundary element only local information is needed instead of the complete force distribution over the boundary.

Accordingly, Briscolini and Santangelo [5] used an immersed boundary approach (referred to as the mask method) to compute the unsteady two-dimensional flow around circular and square cylinders at Reynolds numbers up to 1000. Goldstein *et al.* [6] considered the two-dimensional startup flow around a circular cylinder and three-dimensional plane, and ribbed-turbulent channel flow. In the last two papers the immersed boundary technique was combined with spectral methods and spurious oscillations at the boundary appeared if the forcing was not spread over several (3–4) gridpoints across the boundary. Saiki and Biringen [7] used the forcing of [6] to compute the flow around steady and rotating circular cylinders using fourth-order central finite-difference approximations. Indeed the use of finite differences avoided the appearance of spurious flow oscillations at the boundary even if also in that case the forcing was spread across the boundary using a procedure that the authors refer to as "first-order accurate" similar to the delta-function of Peskin [1]. The main drawback of the forcing introduced in [6] is that it contains two free constants that need to be tuned according to the frequencies of the flow (see Subsection 2.1) and the higher the magnitude of the constants the stiffer the equations become. This implies that for strongly unsteady flows the time step size must be reduced to small values thus making the application of the method expensive. Even with this drawback, however, the method is always more convenient than the classic body-fitted mesh approach and it has been recently used for the simulation of three-dimensional flows by Goldstein *et al.* [8], Goldstein and Tuan [9], Saiki and Biringen [10], and Arthurs *et al.* [11].

Recently Mohd-Yusof [12] derived an alternative formulation of the forcing that does not affect the stability of the discrete-time equations. In addition, in this case there are no free constants, making the derivation of **f** flow independent. In [12] the new forcing was combined with B-splines to compute the laminar flow over a three-dimensional ribbed channel, showing substantial improvements with respect to the previous formulations. This discrete-time forcing scheme was originally developed in a spectral context and has also been successfully applied to flows around cylinders and spheres, at moderate Re.

In the present paper the forcings of [6, 12] are both implemented in a second-order finite-difference scheme with the aim of comparing their accuracy and efficiency. It is shown that while both methods give essentially the same results, the latter is substantially more efficient making it suitable for the simulation of complex three-dimensional flows. This remains true even after the first method is partially modified to make the time step limitation less restrictive.

An adapted version of the forcing of Mohd-Yusof [12] has been applied to the simulation of: (a) the vortex ring formation from a curvilinear nozzle, (b) the axisymmetric and three-dimensional flow around a sphere, and (c) the three-dimensional turbulent flow inside a motored IC piston/cylinder assembly. For the first flow an *ad hoc* experiment has been performed in order to validate the method thus proving the reliability of the results. In the second case we wanted to investigate the capability of the present approach to capture separation phenomena over a smooth surface even if the boundary of the sphere was mimicked over a rectangular mesh. Finally the results of the third flow were validated by available experiments in the literature and, to our knowledge, this is the first example of an immersed boundary simulation with an LES turbulence model.

2. FORCING METHODS

We can now specify the analytic expression of the force density $\mathbf{f}$. Let us define $\mathbf{f}(\mathbf{x}_s, t)$ as the force acting on the element of mimicked boundary 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. It must be noted that the location of $\mathbf{x}_s$ generally is not coincident with the position of the solution variables on the grid (in particular this never happens in the present case where a staggered grid is used) therefore the forcing must be interpolated in the nodes where unknowns are located. In the next two subsections we will specify the forcing disregarding the problem of the interpolation. Instead, Subsection 2.3 will be devoted to the interpolation procedure and its influence on the accuracy of the scheme.

2.1. Feedback forcing. According to the papers by Goldstein *et al.* [6] and Saiki and Biringen [7] the forcing $\mathbf{f}(\mathbf{x}_s, t)$ can assume the expression

$$\mathbf{f}(\mathbf{x}_s, t) = \alpha_f \int_0^t [\mathbf{u}(\mathbf{x}_s, t') - \mathbf{V}(\mathbf{x}_s, t')]\, dt' + \beta_f [\mathbf{u}(\mathbf{x}_s, t) - \mathbf{V}(\mathbf{x}_s, t)]. \tag{2}$$

α_f and β_f are *negative* constants whose dimensions are, respectively, $1/T^2$ and $1/T$. $\mathbf{V}(\mathbf{x}_s, t)$ is the velocity of the boundary that, through its space and time dependence, might mimic also deformable moving bodies. The above quantity is a feedback to the velocity difference $\mathbf{u}(\mathbf{x}_s, t) - \mathbf{V}(\mathbf{x}_s, t)$ and behaves in such a way to enforce $\mathbf{u} = \mathbf{V}$ on the immersed boundary. In fact the first term of Eq. (2) will decrease in time, in the sense that it becomes more negative as the velocity difference increases thus tending to annihilate any difference between $\mathbf{u}$ and $\mathbf{V}$. The second term, on the other hand, can be interpreted as the resistance opposed by the surface element to assume a velocity $\mathbf{u}$ different from $\mathbf{V}$.

An intuitive argument for understanding the action of the above forcing is the following: if from Eq. (1) we retain only the first term on the left hand side and the last term on the right hand side we have (in the case of $V = 0$ or V constant in time)

$$\frac{d\mathbf{q}}{dt} \approx \mathbf{f} = \alpha_f \int_0^t \mathbf{q}\, dt' + \beta_f \mathbf{q}, \tag{3}$$

with $\mathbf{q} = u - \mathbf{V}$. Equation (3) represents a simple damped oscillator with frequency $(1/2\pi)\sqrt{|\alpha_f|}$ and damping coefficient $-\beta_f/(2\sqrt{|\alpha_f|})$. This implies that as $\mathbf{u}$ on the boundary becomes different from $\mathbf{V}$ the forcing $\mathbf{f}$ "brings" $\mathbf{u}$ back to $\mathbf{V}$. In an unsteady flow the magnitude of α_f must be large enough so that the restoring force can react with a frequency which is bigger than any frequency in the flow. Unfortunately the value of the constants is flow dependent and, even if, when α_f and β_f are big enough, the flow becomes independent of their value, there is not a general rule for their determination. The major drawback of this forcing, however, is that big values of α_f and β_f render Eq. (1) stiff and its time integration requires very small time steps. Goldstein *et al.* [6] performed the stability analysis and they found that when all the forcing terms are computed explicitly, for an Adams–Bashforth scheme the stability limit is given by $\Delta t < (-\beta_f - \sqrt{(\beta_f^2 - 2\alpha_f k)})/\alpha_f$, k being a flow dependent constant of order 1. To have a flavor of how restrictive this constrain is we can refer to the simulations of Goldstein *et al.* [6] that were performed at a $CFL = O(10^{-3} - 10^{-2})$. Since the main aim of the present study is to develop a method for simulating three-dimensional

flows in complex geometries we found the above limitation too restrictive making the simulation of three-dimensional flows too expensive.

A partial improvement to the stability limit is obtained when the second term in the right hand side of Eq. (3) is computed implicitly in time,

$$\mathbf{f}^{l+1/2} = \alpha_f \left[\gamma_l \int_0^{t^l} \mathbf{q} \cdot dt + \rho_l \int_0^{t^{l-1}} \mathbf{q} \cdot dt \right] + \beta_f \left[\alpha_l \frac{\mathbf{q}^{l+1} + \mathbf{q}^l}{2} \right] \tag{4}$$

l being the discrete time level and γ_l, ρ_l, and α_l the coefficients of the time integration scheme.[2]

This modification allows us to increase considerably the time step with respect to a fully explicit treatment of the forcing. Once again the time step size depends on the values of α_f and β_f and also on the kind of flow and its Reynolds number. Nevertheless, the simulations that with the fully explicit scheme where performed at $CFL = 10^{-3}$ now were run at $CFL = 2.5 \cdot 10^{-2} - 10^{-1}$ which represents a substantial improvement. It is worth noting that this modification does not require extra memory; it only needs the change of the coefficients related to the implicitly treated terms that, in case of moving boundaries, must be recomputed every time step.

It should be stressed that the stability of the calculation (and therefore the time step size) depends not only on the values of α_f and β_f but also on the flow, i.e., on the details of the geometry to be mimicked. We have observed, for example, that the presence of sharp corners prevents the adoption of small values (in magnitude) of α_f and β_f. On the other hand, for smooth geometries small values of the constants can be used and simulations up to $CFL = 0.5$ can be run. It is also possible to relax the values of α_f and β_f during the "quiet" phases of the flow evolution, but there is no unique criterion for this and *ad hoc* judgments are needed.

2.2. Direct forcing. An alternative expression for the forcing $\mathbf{f}$ which, when properly computed, does not suffer from the limitations previously mentioned was derived by Mohd-Yusof [12]. If Eq. (1) is discretized in time, we have

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

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

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

This forcing is direct in the sense that the desired value of velocity is imposed directly on the boundary without any dynamical process. Therefore, at every time step, the boundary condition holds regardless of the frequencies in the flow. Another advantage is that there are

[2] For example, if an Adams–Bashforth scheme is used for the explicit term and a Crank–Nicolson scheme for the implicit term the coefficients in Eq. (4) are $l = 1$ and $\gamma_1 = 3/2$, $\rho_1 = -1/2$, and $\alpha_1 = 1$. The results shown in the present paper have been obtained using a hybrid third order Runge–Kutta Crank–Nicolson scheme for which we have $l = 1, 2, 3$ and the coefficients can be found in Verzicco and Orlandi [26].

no free constants to choose and the boundary conditions are exactly enforced. In addition the forcing of Eq. (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 (see the Appendix), it does not influence the stability of the time integration scheme.

2.3. Interpolation procedures. As mentioned at the beginning of Section 2, the expressions given for the forcing would be correct if the position of the unknowns on the grid coincided with that of the immersed boundary. This in general is not true because it would require the boundary to lie on coordinate lines or surfaces which is not the case for complex curvilinear geometries. In particular, in the present case, where a staggered grid is used, even if the boundary was coincident with the position where one velocity component was defined, this would not be so for the other components. Therefore, an interpolation procedure would be needed anyway. In order to test the importance of the interpolation, we have implemented three different procedures, and we have computed the effect on the accuracy of the scheme.

The simplest possibility is to select the gridpoints closest to the immersed boundary and to apply the forcing as if position of the unknown and the boundary were coincident. In fact, in this case there is no interpolation and the geometry is described in a stepwise way (Fig. 2a). Note also that the surface is somewhat diffused since the 3 velocity boundary conditions are applied at different locations.

The second procedure consists of computing for each cell crossed by the boundary the volume fraction occupied by the body ψ_b with respect to the volume of the cell ψ. The weight coefficient ψ_b/ψ is then used to scale the forcing applied to the unknowns closest to the boundary. For example, with reference to Fig. 2b for the u_i variable the forcing would be $f_i\psi_b/\psi$, f_i being the ith component of one of the forcings computed above (Eqs. (2), (4), (6)).

In the third case, instead of applying a scaling to the forcing, we compute the velocity value that, in a linear approximation, the point closest to the boundary would have if the boundary had the velocity $\mathbf{V}$. In Fig. 2c the procedure for one gridpoint is shown, and the forcing is simply given by one of the expressions above with the imposed velocity $\overline{\mathbf{V}}$ instead of $\mathbf{V}$.

In order to test these procedures, we computed the formation of a vortex ring by injecting a finite amount of fluid through a curvilinear nozzle (Fig. 3a). The flow and the geometry have been selected in such a way to reproduce an existing experimental apparatus whose flows have been used to validate the numerical results. A detailed comparison with the experiments will be shown in Subsection 3.1; here we only compare the different numerical results to

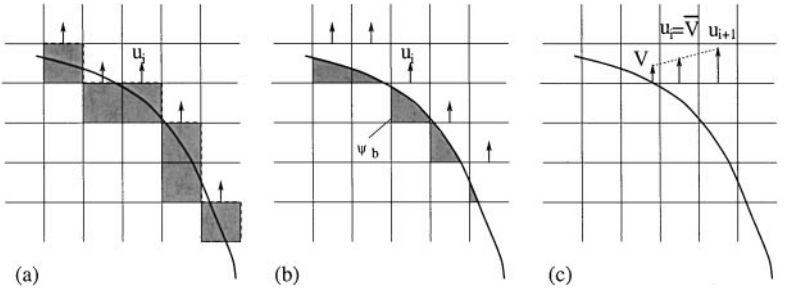

FIG. 2. Sketch of the interpolation procedures: (a) no interpolation ⇒ stepwise geometry, (b) volume fraction weighting, (c) velocity interpolation.

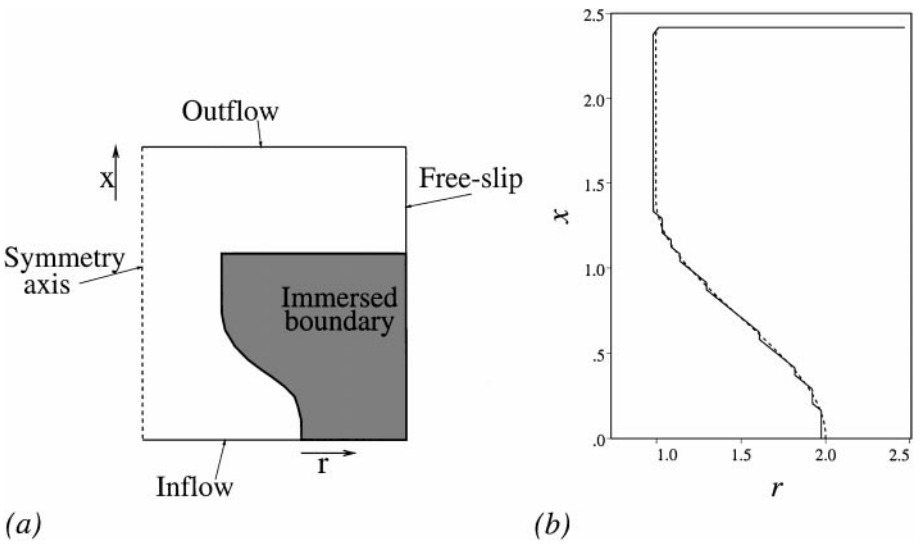

FIG. 3. (a) Sketch of the ring formation problem and its boundary conditions; (b) immersed boundary without interpolation (—) and with interpolation (- - -).

estimate the accuracy of the resulting scheme. The efficiency of the different forcings will be considered in the next subsection. Preliminary simulations have shown that the accuracy of the scheme is independent of weather the forcing of Eq. (4) or (6) is used (see also the next section); therefore, for the moment we will leave the forcing unspecified, and we focus only on the interpolation procedure. In Fig. 4, we show the evolution of the formation of the ring subsequent to the injection of fluid and the generation of secondary vorticity that eventually rolls up in a secondary vortex ring. In order to analyze the accuracy of each interpolation

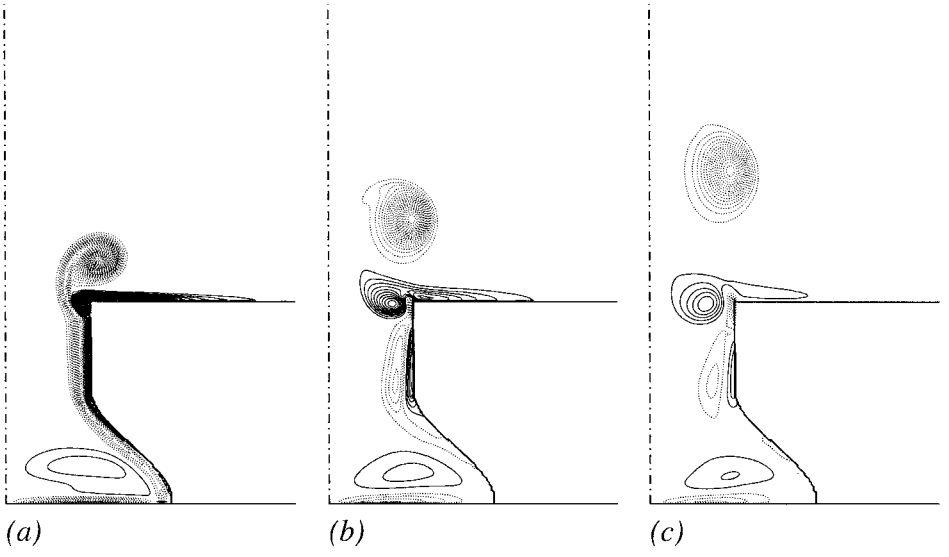

FIG. 4. Azimuthal vorticity contour plots for the vortex ring formation from a curvilinear nozzle, $Re = 1500$ (129×257 gridpoints, respectively, in the radial and axial directions); panels are at $t = 2.5$ (a), $t = 5.0$ (b), and $t = 7.5$ (c) time units. —, for positive; $\cdots$, for negative values ($\Delta\omega_\theta = \pm 2.$).

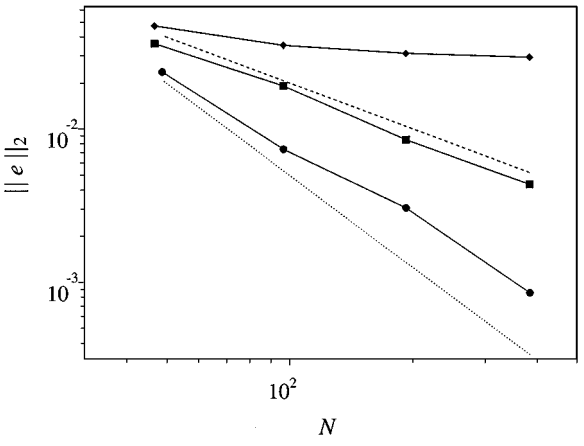

FIG. 5. L_2-norm error of the axial velocity component vs number of gridpoints: ●, interpolated velocities; ■, volume fraction; ◇, stepwise geometry; · · · , −2 slope; - - -, −1 slope.

procedure, the simulation of Fig. 4 has been repeated using different grids ranging from 385×769 down to 49×97 in the radial and axial directions, respectively. Regarding the solution on the finest grid as "exact" we have computed the errors of the solutions obtained on the coarser grids. Both the l_2 norm and maxima errors have been computed for the radial and axial velocity components always showing the same behaviour; therefore here we report the l_2 norm error for the axial velocity component as a representative case. In Fig. 5, it is shown that for the stepwise geometry (reported also in Fig. 3b with a solid line) the error decreases slower than first order. Better results are obtained by weighting the forcing by the volume fraction of the cell occupied by the body, improving the error to first order. Indeed preliminary simulations of flows over flat immersed boundaries have shown that this interpolation procedure underestimates the velocities at the boundary; therefore, even if the geometry is now smooth the velocity boundary conditions are not completely satisfactory yet. The best results are given by the third interpolation approach which yields a solution essentially accurate to second order. This means that within the overall accuracy of the scheme the immersed boundaries are mimicked "exactly." This result is not surprising since a linear interpolation has a second order accuracy; however, it must be noted that with this interpolation we do not solve any equation at the first gridpoint external to the boundary but we are rather imposing a linear velocity profile.

This interpolation scheme is more accurate than those proposed by Peskin [1] and Saiki and Biringen [7] that the authors refer to as first-order accurate; nevertheless in the present case, very close to a wall the velocity profile is assumed to be approximately linear, and this requires the grid to be fine enough in the region of the immersed boundary. Although some local refinement can be obtained using non-uniform orthogonal grids, this is not always possible for every complex geometry. This aspect could be improved on by combining immersed boundaries with embedded grids [14, 15], but a cost/benefit estimate should be performed to see if in this case the immersed boundary technique is still advantageous with respect to a scheme with boundary conforming meshes.

Before concluding this section we wish to stress the following points: in this technique there is no difference between steady or moving boundaries (at least if the motion of the wall is prescribed). In fact in both cases a velocity value coming from the interpolation of

Fig. 2c is assigned as boundary condition. Of course in the case of moving bodies the points over which this boundary condition is assigned must be recomputed every time step but this does not change the nature of the interpolation.

Sharp corners are treated as the other parts of the immersed boundary. In fact we are using a staggered grid; therefore we must compute for every velocity component its distance from the boundary. In this respect it does not make any difference if the boundary is smooth or a sharp corner.

2.4. Efficiency of the forcings. In the previous section, we have shown how the interpolation procedure affects the accuracy of the numerical scheme. In the discussion, we have left unspecified the particular expression of the forcing since we have found that the accuracy is independent of the forcing used. In contrast, now we are concerned with the efficiency of the method, and this proved to be strongly dependent on the forcing used.

We have again considered the flow of Fig. 4, and using the forcing of Eq. (4) we have repeated the simulation gradually increasing *CFL* up to the stability limit. It turned out that with the semi-implicit treatment of that forcing, the integration could be performed up to $CFL = 0.025$ which is already a substantial improvement with respect to the $CFL = O(10^{-3})$ imposed by the fully explicit treatment.

In order to compare the results of the two forcings, in another simulation, we have fixed the $CFL = 0.025$ and we have used the forcing of Eq. (6). The results obtained with the two forcings are compared in Fig. 6a and they prove to be indistinguishable from each other. Nevertheless, while the first forcing is already at its stability limit, the simulation with the second forcing can be run at much higher *CFL* without losing stability or accuracy. In Fig. 6b the time evolution of the peak vorticity for the flow in Fig. 4 is shown for several *CFL*s. It can be noted that the long term dynamics are well predicted even by the simulation at $CFL = 1.5$ which is very close to the theoretical stability limit $CFL = \sqrt{3}$ of the third order Runge–Kutta.

It is worth noting that while both forcings give the same results, since the *CFL* depends linearly on the time step size, the direct forcing allows a speed-up of about 60, making this technique very appealing for the computation of three-dimensional flows. Accordingly, all the numerical simulations shown in Section 3 have been performed using the forcing of Eq. (6) at a *CFL* of about 1.5.

Once again we wish to stress that better performances could have been obtained from the feedback forcing by tuning the α_f and β_f constants during the flow evolution; however, since a quantitative procedure is not available, this tuning would rely too much on the personal judgment of the researcher making the technique somewhat subjective.

2.5. Internal treatment of the body. A few words should be said about the internal treatment of the bodies since the forcings described in Subsections 2.1 and 2.2 only hold at the boundary. For the internal treatment of the body there are several possibilities, even if in our simulations we have found that the external flow is essentially independent of the internal conditions.

A first possibility is to apply the forcing inside the body without any smoothing. This is equivalent to imposing the velocity distribution inside the body with the pressure that adjusts accordingly.

An alternative approach consists of leaving the interior of the body free to develop a flow without imposing anything. Of course, in this case the flow pattern inside the body will be different from the previous case, but the external flow is unchanged.

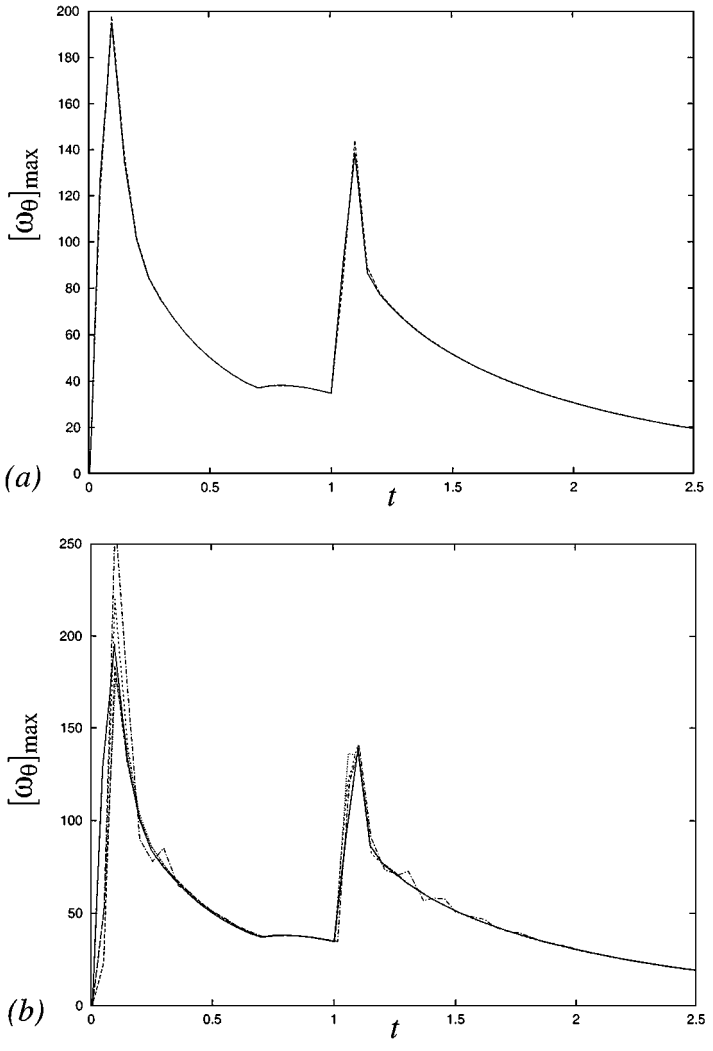

FIG. 6. Time evolution of the peak azimuthal vorticity for the vortex ring formation problem; (a) simulations at $CFL = 0.025$—forcing of Subsection 2.2; $\cdots$, forcing of Subsection 2.1; (b) simulations with forcing of Subsection 2.2; $-\cdot-$, $CFL = 1.5$; - - -, $CFL = 1.0$; $\cdots$, $CFL = 0.5$; —$CFL = 0.025$ (129×257 grid).

The last possibility we have investigated is to reverse the velocity at the first point inside the body in such a way that it still results $\mathbf{u} = \mathbf{V}$ on the boundary. This approach is equivalent to continuing the linear velocity profile of figure 2c up to the first gridpoint inside the body. Also in this case the internal flow pattern is different again, but the external flow is the same as before. Note that this internal treatment was required by [12] in spectral simulations to alleviate the problem of spurious oscillations near the boundary; this procedure was used also by Goldstein *et al.* [6] for their simulations.

Extensive testing of these procedures has been performed to check the influence of the internal treatment of the body on the accuracy and the efficiency of the scheme. We have found that when using the direct forcing of Subsection 2.2 there is essentially no influence. Therefore, depending on the particular flow, the easiest treatment can be used. On the other hand, the feedback forcing of Subsection 2.1 required smaller values of α_f and β_f (in

absolute value) when the velocity distribution inside the body was prescribed. Although this did not affect the external flow, lower values of α_f and β_f allowed the use of bigger time steps, thus improving the efficiency.

3. RESULTS

In this section we will show some applications of the immersed boundary technique in order to validate the procedure and the numerical results. We consider three different flows which can be considered as prototypes of interesting problems for practical applications. The first two examples are moderate-Reynolds flows; therefore, the simulations were carried out without a turbulence model (i.e., direct numerical simulations). In contrast the flow of Subsection 3.3 is the reproduction of an experiment and the high Reynolds number required a sub-grid-scale turbulence model (i.e., large-eddy simulation).

In particular, in the next subsection, we will show the formation of a vortex ring, and the results will be compared with laboratory experiments performed under the same conditions in order to validate the numerical results with flow visualization. In Subsection 3.2 we consider the flow around a sphere for which extensive data are available in the literature for the axisymmetric and the fully three-dimensional flow. In particular, we will show for the sphere that the present approach is capable of predicting not only the flow patterns but also the force coefficients, which are the most relevant quantities in practical applications. Finally in Subsection 3.3 we will mention some results obtained for the computation of the flow inside a motored IC piston/cylinder assembly. This example is important since the experimental data were obtained for a high Reynolds number and a turbulence model was needed in the computation. We will show that even in this case the results given by the immersed boundary technique are very good, and this makes this method very promising for industrial applications.

3.1. Vortex ring formation. The generation of a vortex ring is generally obtained by pushing a finite amount of fluid through a nozzle of diameter D. In the present case, this was achieved experimentally by a piston/cylinder assembly driven by a AC motor/cam/electric clutch mechanism by which the (constant) velocity of the piston (U_p) and the stroke ($L = D/4$) could be controlled. The cylinder, of diameter $D_c = 2D$, was connected to a curvilinear nozzle like that shown in Fig. 3a. The nozzle provided an acceleration to the flow and at the end of the stoke the piston was flush with the bottom margin of the panels of Fig. 7 (i.e., at the position $x = 0$ in Fig. 3b). The ring was generated in water and injected fluid was coloured by di-sodium fluorescein illuminated by a laser sheet in a plane crossing the axisymmetric nozzle through a diameter. This provided flow visualizations of Fig. 7 that could be compared with the numerical simulations. Further details of the experimental apparatus can be found in Verzicco and Orlandi [13]; here we only wish to stress that the vortex ring was injected in a full tank of $50 \times 50 \times 70$ cm^3, thus containing 175 liters of water. This large amount of water was changed only after several rings were produced, and although we tried before each run to eliminate all the fluorescein from the previous shot, some small parcels were left inside the tank like the "wisps" of Fig. 7.

Let D be the exhaust diameter of the nozzle, $U \simeq 4U_p$ the mean flow velocity and ν the kinematic viscosity of the working fluid. Then a Reynolds number for the present flow can be defined by $Re = UD/\nu$. The dynamics of the ring generation have been already the subject of many papers; therefore, we will only summarize it here.

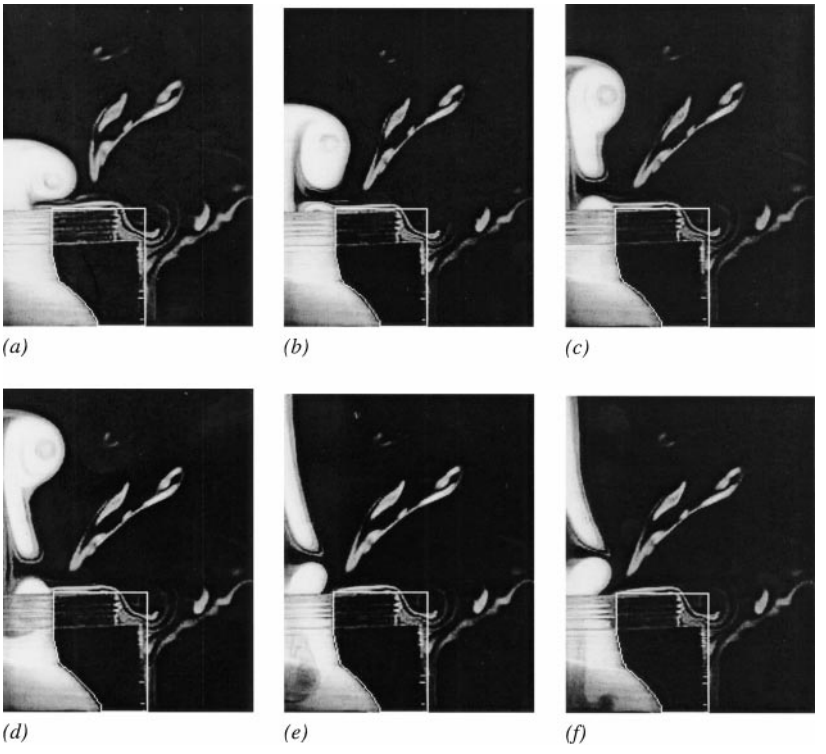

FIG. 7. *Experimental* flow visualizations of the vortex ring formation at $Re = 2500$; panels are at $t = 2.5$ (a), 5 (b), 7.5 (c), 10 (d), 40 (e), and 60 (f) time units.

As the fluid flows along the inner walls of the nozzle some vorticity is generated and this separates at the corner in the exhaust region. This vortex sheet is very unstable and starts rolling up forming a compact toroidal structure (Fig. 7a). During the roll-up some ambient irrotational fluid is entrained. The initial impulse is thus shared with a larger quantity of fluid, and this causes the ring to translate slower than the ejection velocity U. During the roll up, the primary vorticity induces at the external wall of the nozzle a secondary vorticity that also separates at the corner and forms a secondary vortex ring (Fig. 7b). This structure, however, has a circulation oppositely signed with respect to the primary ring, and thus propagates backward inside the nozzle (Figs. 7c–7d). During the descent, this ring interacts with the head of the piston following the dynamics of the ring/wall collision extensively described in Verzicco and Orlandi [13] (Fig. 7e). Depending on the Reynolds number, i.e., on the strength of the primary vortex ring, the secondary vortex can be strong enough to generate a tertiary vortex ring that propagates in the positive axial direction (Figs. 7e–7f).

The dynamics described above only occur if the Reynolds number is high enough. In particular in the present experiments, the tertiary ring propagated away from the nozzle only for $Re \geq 2500$, while $Re \geq 1500$ was necessary for the secondary ring to interact with the piston at its upper position. Finally, only for $Re \geq 500$ was the strength of the primary sufficient to generate a secondary ring.

We have repeated the above described laboratory experiment by numerical simulation with the shape of the curvilinear nozzle reproduced by the immersed boundary technique. The only relevant difference between the experiment and the simulation was that the motion of the piston was not computed. At the inflow of Fig. 3a, we prescribed a constant flow rate for

a time equal to that of the piston motion ($T = 2D/U$) in the experiment, and after the boundary was set as a no-slip wall to account for the presence of the piston at the end of its stroke.

The variation of flow behavior with Reynolds number has been confirmed by the numerical simulations, and in Fig. 8 evidence of the three flow regimes is given. That the

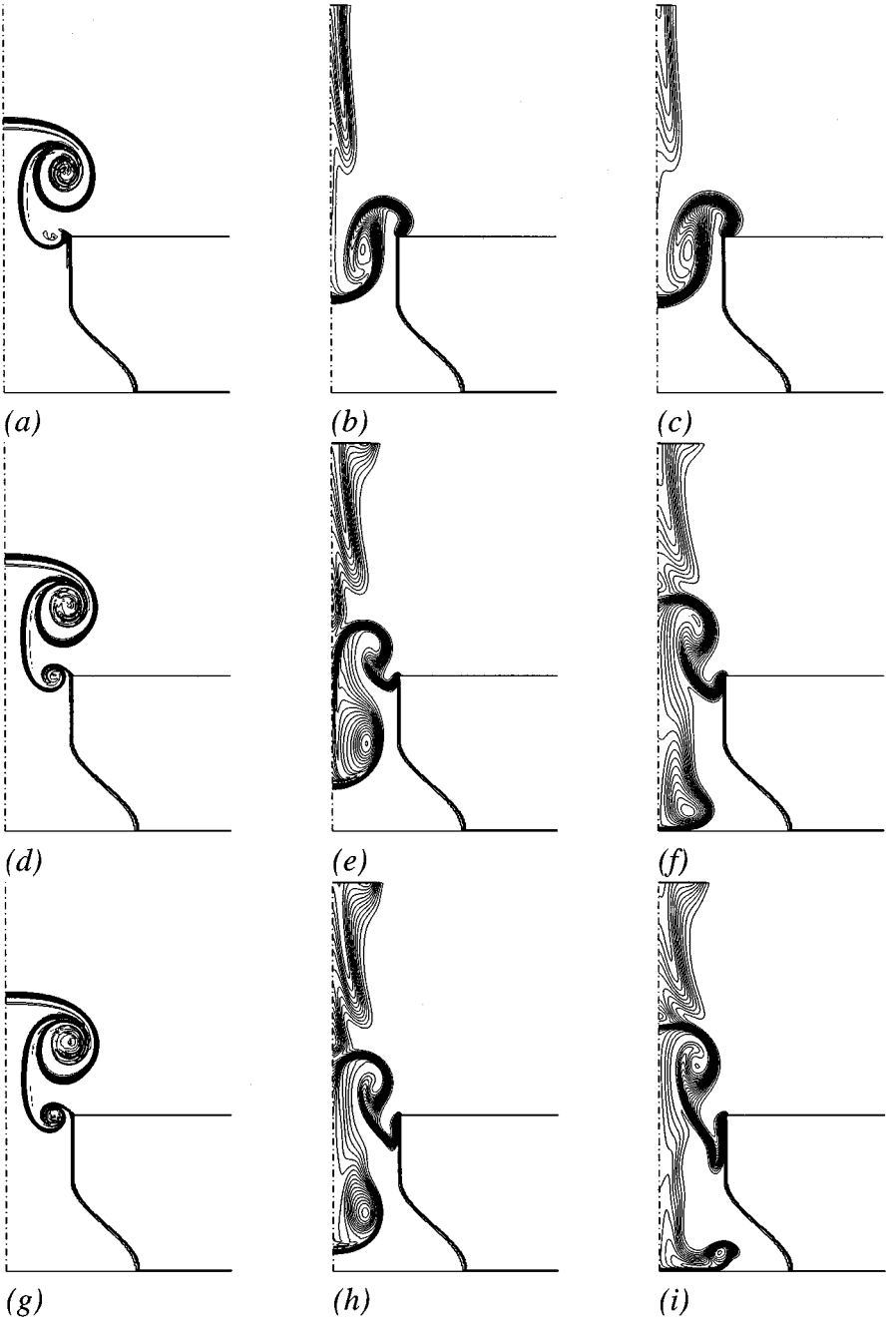

FIG. 8. Contour plots of passive scalar ($\Delta Q = 0.1$) at $t = 5$, $t = 40$, and $t = 60$ time units at $Re = 500$ (a)-(b)-(c) (129×257 grid); $Re = 1500$ (d)-(e)-(f) (129×257 grid); $Re = 2500$ (g)-(h)-(i) (193×385 grid). The Peclet number is $Pe = 10000$ in every simulation.

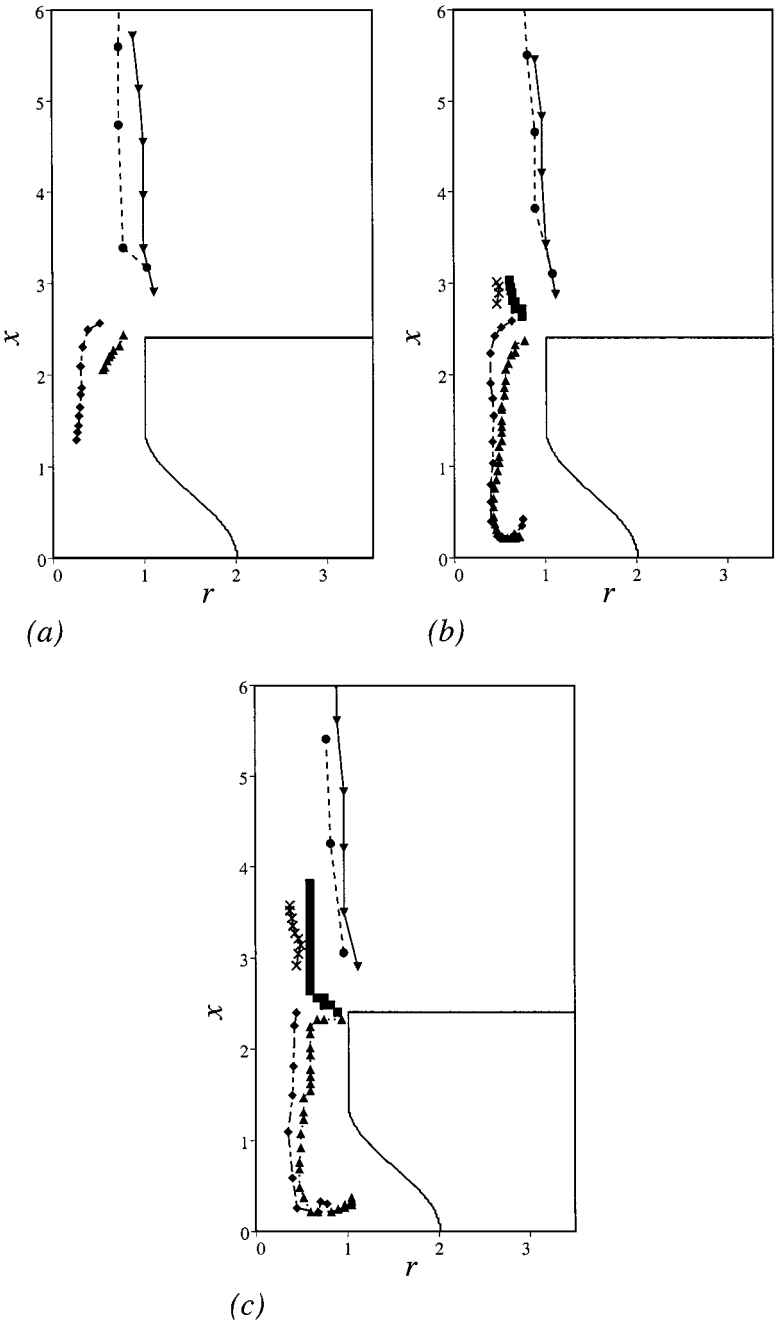

FIG. 9. Comparison between numerical and experimental vortex trajectories at $Re = 500$ (a), $Re = 1500$ (b), $Re = 2500$ (c). Primary ring: filled △, numerical; ●, experimental results. Secondary ring: filled ▽, numerical; filled ◇, experimental results. Tertiary ring: ■, numerical; ×, experimental results.

agreement is not only qualitative but also quantitative is shown in Fig. 9 where the numerical and experimental trajectories for the three flow regimes are given. When comparing the results, it must be noted that in the experiments the trajectories have been evaluated from frames like those of Fig. 7. Therefore some uncertainty in the evaluation of the centres of

the compact patches is possible. In addition the diffusivity of scalars in water is extremely small with Peclet number $Pe = O(10^6)$ (for $Re = O(10^3)$), and such a small value cannot be used in numerical simulations owing to resolution problems. This implies that the numerically simulated scalar is more diffusive than in reality and this could cause some additional mismatch.

Before concluding this section, we wish to stress that concerning the passive scalar transportation we were not able to impose exactly the condition $\nabla Q \cdot \mathbf{n} = 0$ along the immersed boundary and as a palliative we canceled out the viscous terms of the passive scalar equation inside the mimicked body. This prevents the scalar from diffusing inside the body and, since the velocity is zero at the wall, the convection is also inhibited. Although the resulting fields show a satisfactory behaviour, we are aware that it is not "exact" and it was only meant as a means to compare computer simulations and laboratory experiments within this study.

3.2. Flow around a sphere. In the previous section, we have seen that the immersed boundary technique can cope very well with flow in a complex geometry whose boundary crosses the coordinate lines. In that case, however, owing to the presence of a sharp corner the separation of the flow was naturally imposed by the geometry and an inaccurate treatment of the geometry probably would not affect the flow evolution. For this reason we will show here the results of the flow around a spheres since in this case the body is smooth everywhere and the flow separation is determined solely by the viscous processes at the wall. We believe that this flow should display every problem in the treatment of the boundary, and for this reason extensive analysis has been performed.

In this case, a uniform flow with velocity U is imposed at a certain distance from a sphere having a diameter D. The Reynolds number is now $Re = UD/\nu$, and, depending on its value, the flow exhibits different regimes ranging from steady to unsteady up to fully turbulent. Indeed, in the present study, the fully turbulent regime has not been considered since it would require a large computational effort, which is beyond the purposes of the present paper.

As a first test, we have reproduced the simulations carried out by Fornberg [16] that used boundary fitted meshes to simulate the axisymmetric flow around a sphere from $Re = 100$ up to $Re = 5000$. Those results were checked for grid-independence and domain size; therefore, they can be considered accurate enough to be used as reference solutions. We are aware that for $Re \geq 300$ the hypothesis of axial symmetry is not physical because the flow develops three dimensionality in the wake; however, the high-Re axisymmetric simulations were used to compare our results with those of Fornberg [16] obtained with a similar hypothesis. In the second part of this section, in contrast, we will simulate the full three-dimensional flow to capture the losses of symmetry occurring at $Re \simeq 300$ and $Re \simeq 450$ (Mittal [17]).

The simulations have been performed on a grid like that of Fig. 11a. Both in the radial and axial directions a non-uniform mesh has been used in order to cluster enough gridpoints in the region around the sphere. The spatial resolution of each simulation has been varied according to the Reynolds number and the grid refinement check for the most critical cases has been performed. At the external radial boundary a free slip condition has been imposed and this could cause a flow acceleration, due to blocking effects, if the radial boundary is placed too close to the sphere. Preliminary simulations have shown that a radial boundary at 4 sphere diameters induced negligible errors while an adequate radial stretching of the grid allowed the use of a sparse grid in the dynamically passive region. Another possible cause of error is the proximity of the sphere to the inflow which was assumed axially uniform;

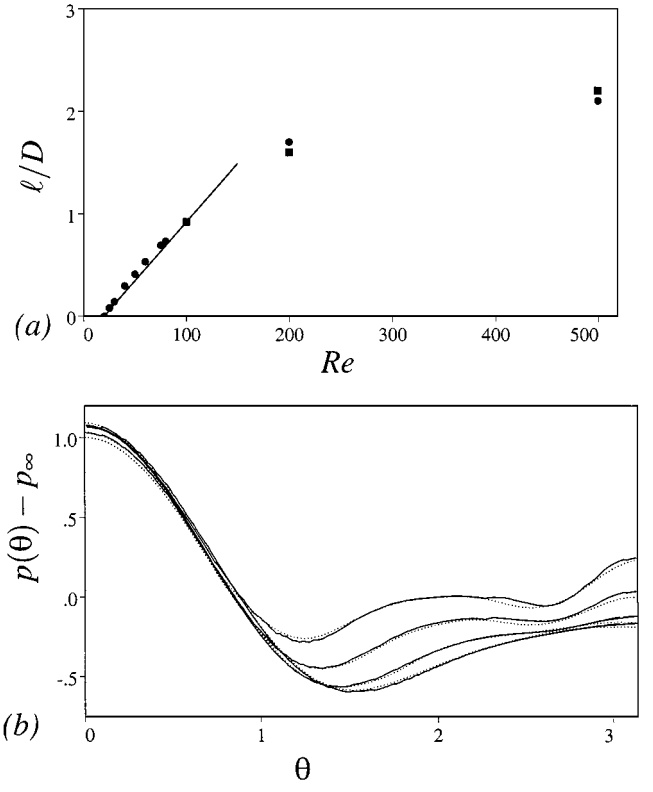

FIG. 10. (a) Length of the separation bubble vs Re; ●, present results; ■, results by [16]. —, linear fit from [18]. (The length at $Re = 1000$ is not reported to avoid the shrinking of the data at $Re \leq 100$, it results $\ell/D = 2.4$ in the present case and $\ell/D = 2.5$ by [16].) (b) Pressure profiles over the sphere surface at $Re = 100$, 200, 500, and 1000 ($Re = 100$ corresponds to the line with the most negative values and $Re = 1000$ is that with the less negative values); —, present results; $\cdots$, data from Fornberg [16].

also in this case some tests have been performed and a location of the sphere 3 diameters downstream of the inflow was found satisfactory. Downstream of the sphere the domain was 4 sphere diameters long, and at the outflow convective boundary conditions [20] were used.

According to Batchelor [18] the flow around a sphere does not separate up to $Re \simeq 24$, and for increasing Reynolds number the axial length of the separation bubble grows linearly up to $Re \simeq 100$. The same results have been found by the present numerical simulations, and the length of the separation bubble is very well predicted as shown in Fig. 10. Concerning the onset of the separation we have found that the flow remains attached up to $Re = 22$, while for $23 \leq Re \leq 25$ the separation length is of order of $10^{-2} D$. Using the results published by Fornberg [16], we could compare for $Re \geq 100$ the pressure distribution over the surface of the sphere and the force coefficients. Representative pressure fields for various Reynolds numbers are shown in Fig. 11, while the pressure profiles over the surface of the sphere are given in Fig. 10b together with the curves computed by Fornberg [16]. At first glance the correct behaviour of the pressure might appear very surprising since the boundary body force apparently does not imply any condition for the pressure. In the Appendix, however, it will be shown that indeed the correct boundary conditions for the pressure are automatically imposed by the forcing thus explaining this good agreement. From the pressure and velocity field around the sphere we have computed also the drag coefficient

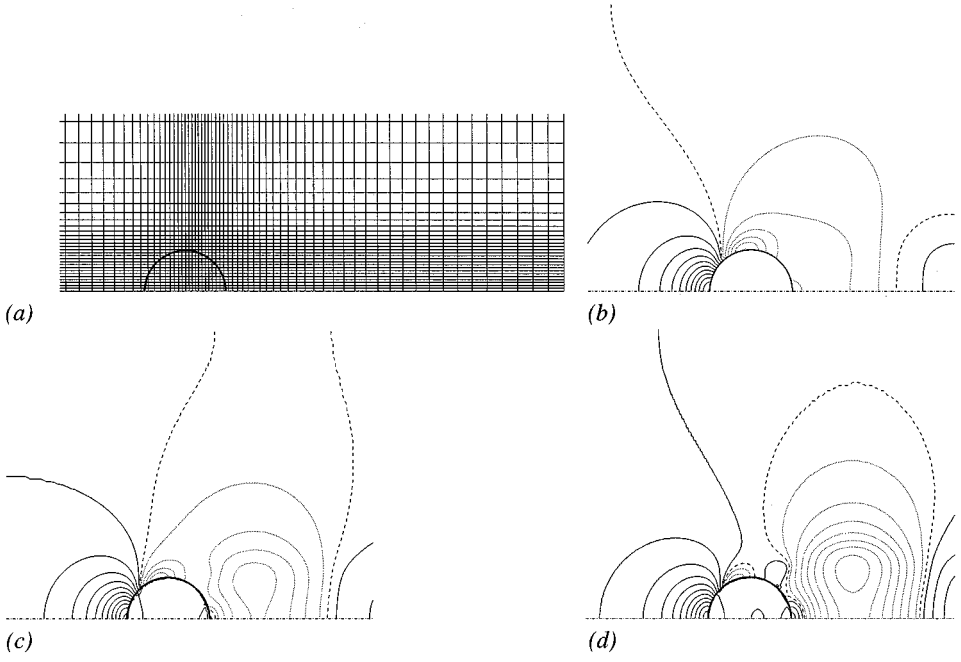

FIG. 11. (a) Stretching of the grid used for the computation of the flow around the sphere. Pressure contour plots at $Re = 200$ (b), 500 (c), and 1000 (d). ($\Delta p = \pm 0.05$, —, positive; · · · , negative, - - -, 0 value). Grid 129×257 gridpoints, respectively, in the radial and axial directions.

C_d that, as shown in Table I, agrees very well with the results of Fornberg [16] and the error never exceeds the 2%. Considering that the resolution of the boundary layer used by Fornberg [16] with the body fitted mesh was incomparably better than ours, we can consider the present values of the C_d essentially correct. It must be noted that the drag coefficient is made by two contributions: pressure and viscous drag. For each simulation we have checked that each separate term compares well with the results given by Fornberg [16]. In addition, in Fig. 12, we show that the rate of convergence of the present drag coefficient to the value of Fornberg [16] is second order for the finer grids and even third order for the coarser grids; this behaviour is not surprising since the drag coefficient is an integrated quantity, and its convergence rate is consistent with the results of Fig. 5. This result is very important for the validation of the immersed boundary approach because we have shown that all the viscous processes and the pressure dynamics are correctly handled even if these simulations are much more inexpensive than those performed on a curvilinear grid (see Section 4 for quantitative comparisons).

TABLE I
Comparison of the Drag Coefficient for the Flow around the Sphere at Different Reynolds Numbers

Re	C_D (Fornberg [16])	C_D (Present results)	% error
100	1.0852	1.0794	0.53
200	0.7683	0.7567	1.50
500	0.4818	0.4758	1.24
1000	0.3187	0.3209	0.69

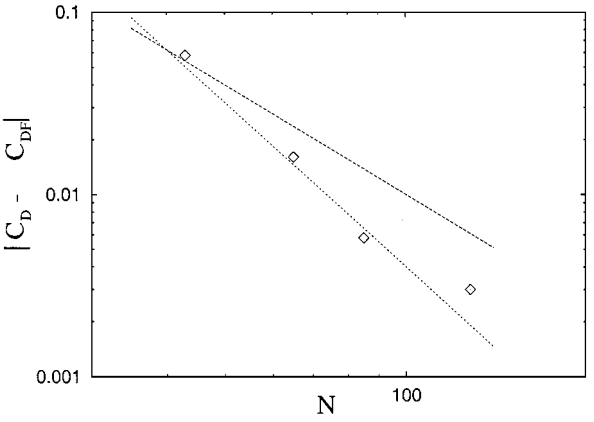

FIG. 12. Convergence of the drag coefficient C_D to the value given by Fornberg [16] C_{DF} vs number of grid points. $\cdots$, slope -3; ---, slope -2 given for comparison. Flow at $Re = 100$.

In order to test the capability of the code to cope with fully three-dimensional flows and to capture flow transitions, we have performed some simulations in which the hypothesis of axial symmetry was removed. Similar simulations were performed by Mittal [17] using boundary fitted grids and he found that the flow preserves symmetry for $Re \leq 270$. When the Reynolds number is in between 270 and 300 the wake becomes wavy but it still preserves symmetry about a plane crossing the sphere in the centre. Finally for $Re \geq 450$ the flow becomes fully three dimensional and all symmetry is lost.

In the present simulations, we have initiated the three dimensionality as follows: over a developed axisymmetric flow, we have imposed a perturbation of the inflow according to $U(\theta) = U(0)[1 + A\sin(\theta)]$ with $A = 0.25$ for all simulations. This perturbation was applied for a window of time of $5D/U$ units and then stopped. The response of the flow to this perturbation was analysed by looking at the azimuthal energy modes. If $\mathbf{u}(r, \theta, x)$ is the velocity field, after a Fourier transform in the azimuthal direction, we have $\tilde{\mathbf{u}}(r, k, x)$ with k the azimuthal wavenumber. Indicating by $\tilde{\mathbf{u}}^*$ the complex conjugate of $\tilde{\mathbf{u}}$ we compute $E(k) = \int_0^{L_r}\int_0^{L_x} \tilde{\mathbf{u}}^* \cdot \tilde{\mathbf{u}}\, dr\, dx$, L_r and L_x being the dimensions of the computational domain in the radial and axial directions, respectively. $E(k)$ is proportional to the kinetic energy of the field in the azimuthal wavenumber k and its evolution is shown in Fig. 13 for three different Reynolds numbers. We can see that when the Reynolds number is below the first critical value, the perturbation gradually dies out and the flow recovers its initial axial symmetry (Fig. 13a). When the Reynolds is slightly above the threshold, in contrast,

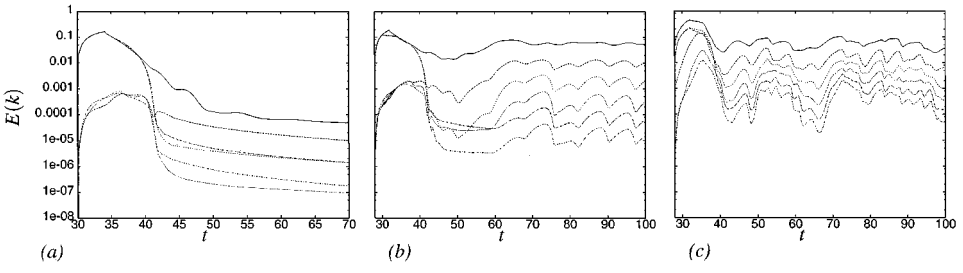

FIG. 13. Time evolution of the azimuthal energy modes for the three-dimensional flow around the sphere at $Re = 200$ (a), 300 (b), and 500 (c). —, $k = 1$; thick ---, $k = 2$; ---, $k = 3$; $\cdots$, $k = 4$; – – –, $k = 5$; and – · –, $k = 6$. (In the long term evolution of panels (b) and (c) the lines for the modes 1 to 6 are monotonically decreasing in amplitude.)

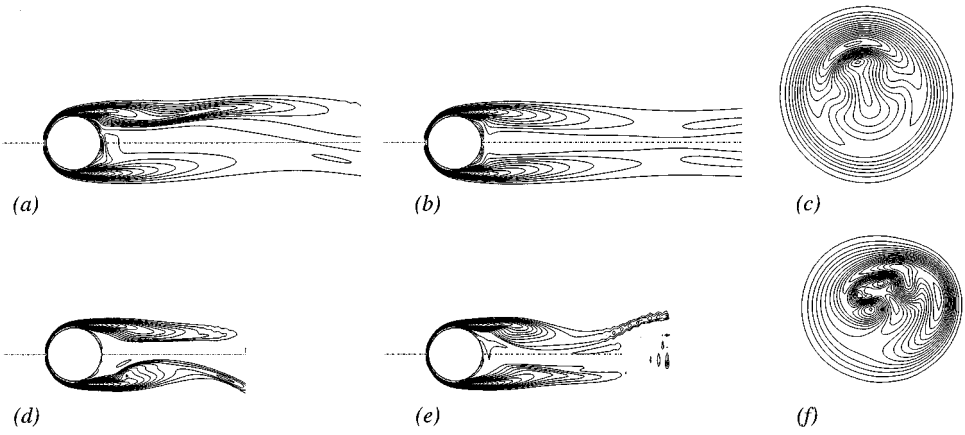

FIG. 14. Contour plots of azimuthal vorticity at $Re = 300$ (a)–(c) and $Re = 500$ (d)–(f) through orthogonal sections. The $r - \theta$ sections (c) and (f) are taken in both cases at a distance $x/D = 2$ downstream of the sphere ($\Delta\omega_\theta = \pm 2$). Grid $97 \times 97 \times 193$ gridpoints in the azimuthal, radial, and axial directions.

the perturbation is sustained indicating that the flow has some three dimensionality. It is worth noting that the immediate response of the energy modes to the given perturbation displays the growth only of the odd modes thus confirming that a symmetry about one plane is retained. Later on, however, the even modes start growing owing to the nonlinear interaction of the odd ones. It might seem that the energy spreading among all the azimuthal modes evidences the loss of symmetry about a plane; however, this is true only when the flow remains centered at the axis (i.e., the $k = 1$ mode has zero energy). In this flow this certainly does not happen since the wake oscillates off axis and this explains why the flow is still symmetric even if all azimuthal modes contain energy. This is confirmed by the vorticity contour plots of Fig. 14 showing the oscillation of the wake only in one plane. It might be objected that this behaviour is a consequence of the perturbation which is also symmetric. However, while the symmetry of the perturbation was about the $\theta = \pm\pi/2$ plane the symmetry of the wake is about a different plane thus excluding the objection. In addition, Mittal [17] obtained essentially the same results starting from a random perturbation without any symmetry. The last simulation is performed at $Re = 500$ which is above the second critical Reynolds number. Now all energy modes gain energy directly form the perturbation indicating the loss of the previous symmetry. Again this is confirmed by the flow visualizations of Fig. 14 showing the waviness of the wake in two orthogonal planes.

3.3. Flow inside a IC piston. In this last example we show the capability of the method to simulate high-*Re* turbulent flows in complex geometries for which a turbulence model is needed. In addition, this example includes a moving boundary, a feature which was missing in the two previous flows.

The configuration chosen is a simplified axisymmetric piston-cylinder assembly with a fixed central valve. For a configuration like that of Fig. 15a, experimental measurements are available for the validation of the numerical results. In particular, Morse *et al.* [19] used laser-Doppler anemometry to measure phase averaged mean and rms radial profiles of axial velocity. The profiles are available at 10-mm increments starting from the cylinder head for crank angles 36° and 144° after top dead center. In the experiment, the piston was externally

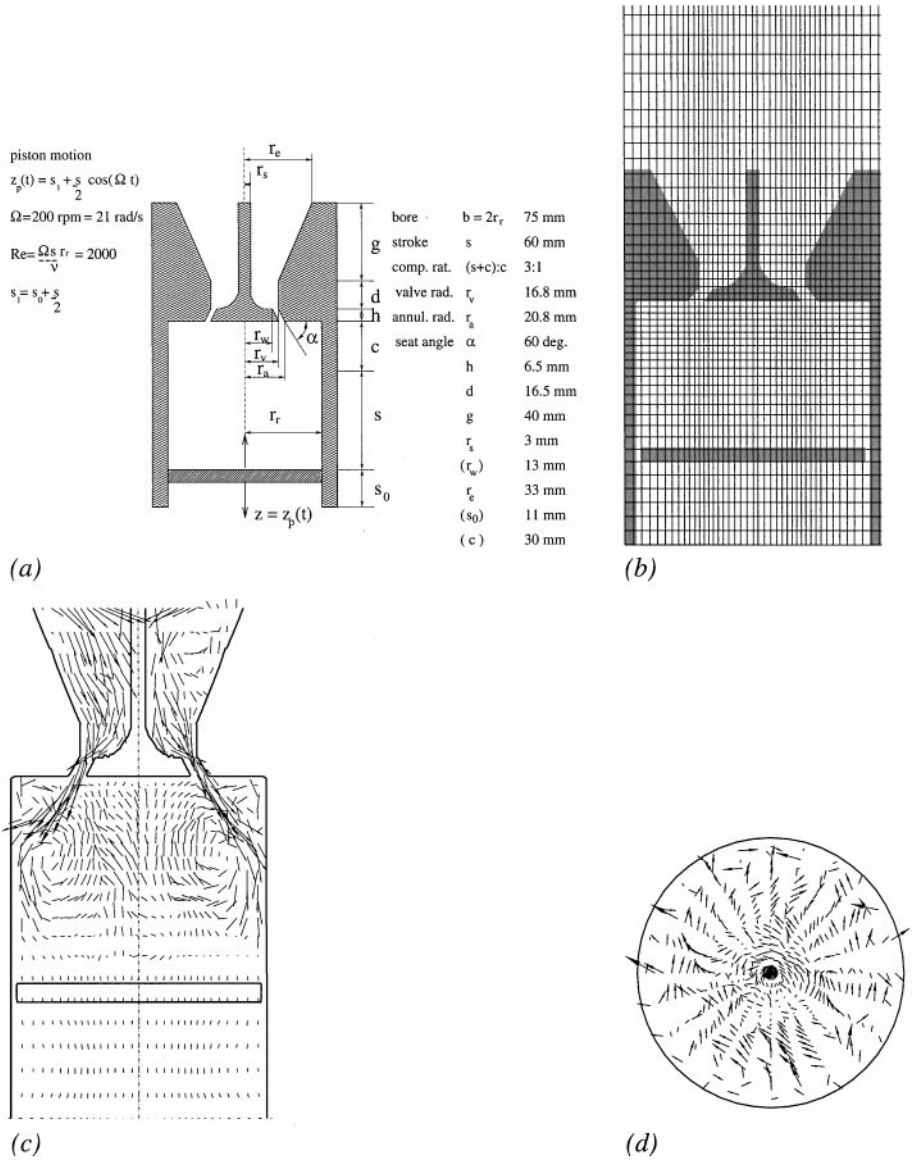

FIG. 15. Flow inside the IC piston: geometry details (a) and grid with the immersed boundary (b). Vector plots at $t = \pi/2$ of the oscillation cycle (c) and (d). The $r - \theta$ section (d) is taken 10 mm below the ceiling of the cylinder. Grid $65 \times 65 \times 151$.

driven so that the fluid flowed into the cylinder from outside during the downward piston motion and vice versa when the piston moved up. Since the valve was fixed and a tiny annular gap was left open between the valve and the cylinder head, no compression phase was included in the flow dynamics. The piston was driven by a simple harmonic motion at a speed of 200 rpm $\simeq$ 21 rad/s which for the present geometry yields a mean piston speed of $\bar{V}_p = 0.4$ m/s (when averaged over half cycle). The Reynolds number of the flow based on $\bar{V}_p$ and on the piston radius is $Re = 2000$ in air.

Details of the subgrid-scale model used in this computation are given in Verzicco *et al.* [20]. Here it suffices to mention that a dynamic Smagorinsky model was used (Germano

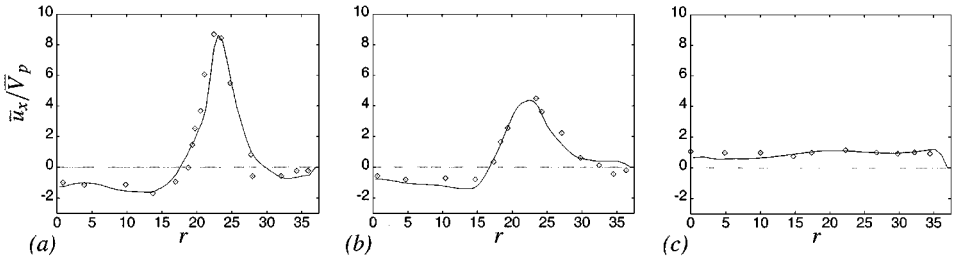

FIG. 16. Comparison between numerical and experimental averaged vertical velocity profiles for the crank angle 36° respectively 10 (a), 20 (b), and 30 mm (c) below the cylinder ceiling.—, present results; ◇, experiments by [19]. The abscissa is expressed in millimeters. The velocity profiles drop to zero for $r \geq 37.5$ mm when the immersed boundary region is crossed.

et al. [21] and Lilly [22]) and all the points with total viscosity smaller than zero were clipped; their number never exceeded the 3% of the total points. In Verzicco *et al.* [20] all the details of the computation and the boundary conditions are given; in this paper we only mention that all the solid surfaces are obtained by immersed boundaries (Fig. 15) while the lateral surface is free-slip. At the lower surface a prescribed mass flow is assigned with a constant axial velocity profile in such a way as to preserve the free divergence in the region between the lower boundary and the piston. At the upper boundary, convective boundary conditions are used as extensively explained in [20].

In Verzicco *et al.* [20] extensive analysis of the flow inside the piston has been performed showing the changes in the dynamics when the axial symmetry was enforced, when three-dimensionality developed, and when the turbulence sets in. In this paper we only show the turbulent case for which the comparison with the experiments was made.

In Fig. 15 snapshots during one instant of the oscillating cycle are given and the high three dimensionality of the flow can be appreciated from the vector plots in orthogonal sections. Radial profiles of axial velocity were obtained by phase averaging the fields over four piston cycles and then averaging in the azimuthal direction. These profiles are shown in Fig. 16 for several distances from the top of the cylinder at two different crank angles and the comparison with the experimental data shows that the agreement is always very good. In Verzicco *et al.* [20] profiles for additional sections and rms profiles of axial velocity are also given showing that the quality of the agreement is always very good. Finally in Haworth [23] the present results are compared also with the data obtained by a code using an unstructured boundary fitted deformable mesh. In that paper it is shown that the quality of the results is the same even though the immersed boundary technique is much less expensive (see Section 4 for a quantitative comparison).

4. CLOSING REMARKS

In Section 3 we have discussed three numerical examples showing the capability of the present method to deal with different complex geometry flows. We have seen that, although the flows are quite complex, the simulations are in fact performed on simple grids. Therefore, two main advantages immediately arise. The first is that the computer codes are much simpler than the curvilinear coordinate counterparts since the former do not need metric arrays and the solution procedure requires less operations per node. This implies that immersed-boundary codes have in general low RAM and CPU-time requirements making

possible three-dimensional complex flow simulations on PCs or small workstations. For example, for the flow around the sphere, Fornberg [16] reports that his code required 708 Mb of memory for a grid 487×163 on a 64-bit precision machine. The present code, for the same grid in double precision, has a size of only 20 Mb; the comparison of the CPU time is not given since Fornberg's simulations were run in 1988 and the actual processor speeds are not comparable.

A better comparison can be performed for the IC cylinder flow of Subsection 3.3 which was simulated at the same time using body-fitted and immersed boundary codes. The present incompressible immersed-boundary three-dimensional turbulent case was run on a PC-like workstation (500-MHz processor) with 128 Mb of RAM in about one week using a grid of approximately $6 \cdot 10^5$ nodes (about 24 CPU-hours per engine cycle). By contrast, the compressible unstructured boundary-fitted deforming-grid case by Haworth and Jansen [24] ($\approx 1.5 \cdot 10^5$ nodes) was run either on a single processor of a Cray T-90 or on eight processors of a SGI Origin 2000, occupied 600 Mb of RAM, and required 30–40 CPU-hours per engine cycle. Further simulations were performed by Haworth and Jansen [24] with increased spatial resolution and results of the same quality as the present required a grid of $5 \cdot 10^5$ nodes with a consistently higher memory and CPU time requirement (Haworth, personal communication).

On the other hand, the present immersed boundary technique has some disadvantages that should be considered when deciding which approach is the most convenient for a given problem. In fact, the points inside the mimicked boundary are passive concerning the flow dynamics but the equations of motion are integrated in that region anyway. For this reason the choice of the "simple grid" over which equations are integrated must be a compromise between the need of having an efficient computer code and a low percentage of the total gridpoints inside the boundaries. For the examples shown in this paper a polar-cylindrical grid was a quite natural choice, but this is not always the case. For example, if one had to compute the flow inside a 90°-bent circular pipe neither a rectangular nor a polar-cylindrical grid would be appropriate since most of the points would be in the passive region making the use of the immersed boundary technique uneconomical. Similarly, flows around bodies at very high Re would be problematic since, while only a small portion of the computational volume would be in the body, the grid stretching required to resolve the boundary layers would place a large number of wasted gridpoints in that region.

Another drawback is related to the interpolation procedure of the forcing; in fact, as mentioned in Subsection 2.3 at the first gridpoint outside the boundary we are not solving the Navier–Stokes equations but we enforce a velocity coming from a linearization. This requires the grid to be fine enough close to the boundary so that the linearized velocity is accurate. This has been obtained in the present paper by clustering the gridpoints in the boundary regions. For simple geometries, like the sphere of Subsection 3.2 this has been very easy even if many points have been concentrated also inside the sphere thus in the passive region. For a more complex geometry, like the IC cylinder, this strategy was only partially successful, but the results are still satisfactory. There might be cases, however, in which a nonuniform grid is not enough to have an accurate description of the boundary, and in those cases a body fitted mesh would be perhaps preferable.

A remedy for both the above problems could be the combination of the immersed boundaries with the embedded grids. This possibility has been pursued by [14, 15] for the standard body-fitted approach and only recently by Roma *et al.* [25] for the immersed boundary technique applied to two-dimensional flows. If the results obtained for the B-splines [15] would

be confirmed also for finite-difference/immersed boundary approximations the above combination promises to be very powerful.

APPENDIX: SOME COMPUTATIONAL DETAILS

In this appendix, we describe in particular how the integration of Eq. (1) is changed by the presence of the forcing **f**. The integration procedure is essentially that described in Verzicco and Orlandi [26] and it consists of a fractional-step method in combination with a hybrid third-order Runge–Kutta explicit scheme for the convective terms and an implicit Crank–Nicolson for the viscous terms. The advantage of the third order Runge–Kutta scheme with respect to other time-integration procedures is the improved stability condition that, with the help of the viscous terms, results in $CFL \simeq 2$. Indeed in this scheme each time step is advanced through three substeps thus implying three times more operations than the common second-order schemes; nevertheless the third-order Runge–Kutta is still advantageous since it does not require extra storage and at the same computational cost as other second-order schemes yields an error which is generally smaller.

Concerning the implicit treatment of the viscous terms, this is done in order to avoid the restrictive viscous stability condition $\Delta t \leq \Delta x^2 Re/(2n)$ being n the number of spatial dimensions (Ferziger and Peric [27, p. 135]). It might be thought that for high-Re flows the condition becomes less restrictive; however, as the Reynolds number increases the grid spacing must be decreased as well making the stability condition always more restrictive than the CFL condition. This is true even when performing large-eddy simulations. In fact for the case of Subsection 3.3 it has been verified that while the turbulent stresses only play a marginal role in the stability of the integration, the viscous stresses are the most restrictive and, if they had not been computed implicitly, the integration time step would have been about ten times smaller than the limit given by the CFL condition.

Equation (1) is provisionally advanced in time using the pressure at the previous time step. Thus we obtain a non-solenoidal velocity field $\hat{\mathbf{u}}$ given by

$$\frac{\hat{\mathbf{u}} - \mathbf{u}^l}{\Delta t} = -\alpha_l \nabla p^l - [\gamma_l \nabla(\mathbf{uu})^l + \rho_l \nabla(\mathbf{uu})^{l-1}] + \frac{\alpha_l}{2Re}\nabla^2(\hat{\mathbf{u}} + \mathbf{u}^l) + \mathbf{f}^{l+\frac{1}{2}}. \tag{7}$$

$\hat{\mathbf{u}}$ is then projected onto a solenoidal field by

$$\mathbf{u}^{l+1} = \hat{\mathbf{u}} - \alpha_l \Delta t \nabla \phi^{l+1}, \tag{8}$$

with the scalar ϕ^{l+1} computed applying the divergence to the above equation thus solving

$$\nabla^2 \phi^{l+1} = -\frac{\nabla \cdot \hat{\mathbf{u}}}{\alpha_l \Delta t}. \tag{9}$$

Once the scalar ϕ^{l+1} is known the velocity field is updated using Eq. (8) and the new pressure field is obtained from

$$p^{l+1} = p^l + \phi^{l+1} - \frac{\alpha_l \Delta t}{2Re}\nabla^2 \phi^{l+1}. \tag{10}$$

The first point to be noted is that when comparing Eq. (7) with Eq. (5) (with $\hat{\mathbf{u}}$ instead of $\mathbf{u}^{l+1}$) we see that, owing to the implicit treatment of the viscous terms, $\mathrm{RSH}^{l+1/2}$ is in fact a function of $\hat{\mathbf{u}}$ therefore the evaluation of $\mathbf{f}^{l+1/2}$ through Eq. (6) is not explicit but it requires

the inversion of a discrete system of equations coupled with that of the discrete momentum equation. In practice the inversion of this additional system is not performed because Eqs. (7) and (6) can be properly recast in such a way that only the standard Eq. (7) is inverted. Since the notation is the easiest in one dimension we will show the procedure only for one-dimensional equations, the extension to two and three dimensions being straightforward. First we note that Eq. (7) holds both in the fluid and on the immersed boundary resulting in the first case $\mathbf{f}^{l+1/2} \equiv 0$. Accordingly, indicating by $\delta\mathbf{u} = \hat{\mathbf{u}} - \mathbf{u}^l$ Eq. (7) inside the flow reads

$$\left(1 - \frac{\alpha_l \Delta t}{2Re}\frac{d}{dx^2}\right)\delta\mathbf{u} = \mathrm{RHS}^{l+1/2}, \tag{11}$$

or when d/dx^2 is discretized spatially by a three-point stencil approximation

$$a_i \delta\mathbf{u}_{i-1} + (1 + b_i)\delta\mathbf{u}_i + c_i \delta\mathbf{u}_{i+1} = \mathrm{RHS}_i^{l+1/2}, \tag{12}$$

where a_i, b_i, and c_i depend on the particular approximation used for the spatial derivative. Note that now $\mathrm{RHS}_i^{l+1/2}$ contains only quantities known from the previous step therefore its evaluation is straightforward.

For Eq. (7) on the immersed boundary the computation of the forcing should be performed; however, even if we do not know the value of $\mathbf{f}^{l+1/2}$ we know that at the first gridpoint outside the boundary we must have $\hat{\mathbf{u}}_i = \overline{\mathbf{V}}_i$ (see Subsection 2.3). But $\overline{\mathbf{V}}_i$ is just a linear interpolation between $\mathbf{V}_i$ and $\hat{\mathbf{u}}_{ie}$ with $ie = i + 1$ or $ie = i - 1$ depending on the direction of the outward normal of the body (for example, for the case of Fig. 2c we have $ie = i + 1$). This implies that $\hat{\mathbf{u}}_i$ can be written as $\hat{\mathbf{u}}_i = d_i \mathbf{V}_i - e_i \hat{\mathbf{u}}_{ie}$ with d_i and $-e_i$ the coefficients of the linear interpolation. By adding and subtracting the appropriate terms the previous relation can be rewritten as

$$\delta\mathbf{u}_i + e_i \delta\mathbf{u}_{ie} = d_i \mathbf{V}_i - \mathbf{u}_i^l - e_i \mathbf{u}_{ie}^l = \mathrm{RHS}_i^{l+1/2}. \tag{13}$$

The above equation has a structure similar to Eq. (12) therefore they can be solved together without changing the solution procedure and without requiring the explicit evaluation of the forcing.

Another point that needs to be clarified is the imposition of the forcing in the equation for $\hat{\mathbf{u}}$ instead of $\mathbf{u}^{l+1}$. Although this problem does not pertain directly to the immersed boundary technique but rather to its combination with the fractional step method, we believe that some discussion is necessary since, if the correction step (8) changed the velocities on the immersed boundary, the treatment of the body would be incorrect. Indeed for every numerical simulation we have monitored the errors at the immersed boundaries and we have seen that the modification produced by the correction step (8) was always negligibly small ($O(10^{-3} - 10^{-4})$) when compared to the velocity values themselves. A possible explanation for this behaviour is the following: the velocity boundary conditions in a frame moving with the mimicked (plane) surface would be $u_\tau = u_n \equiv 0$, with τ and n the tangential and normal directions, respectively. If it happened that $u_\tau = u_n \equiv 0$ then the forcing would vanish since there is nothing to enforce and the momentum equation normal to the wall would read

$$\frac{\partial u_n}{\partial t} + u_n \frac{\partial u_n}{\partial n} + u_\tau \frac{\partial u_n}{\partial \tau} = -\frac{\partial p}{\partial n} + \frac{1}{Re}\frac{\partial^2 u_n}{\partial n^2} + \frac{1}{Re}\frac{\partial^2 u_n}{\partial \tau^2}. \tag{14}$$

Because of the boundary conditions, all the terms disappear except for the first and the last on the left hand side. However, if the velocities are linearized at the immersed boundary the term $1/Re\,\partial^2 u_n/\partial\tau^2$ also vanishes and the above equation yields $\partial p/\partial n = 0$. The same arguments could be repeated for the tangential velocity component u_τ showing that *along* the boundary it must result $\partial p/\partial\tau = 0$. Note that this occurs only on the mimicked boundary, while inside and outside the body the pressure behaves according to the flow dynamics. This explains why, even if nothing is explicitly imposed on the pressure, it behaves correctly at the walls as shown in Fig. 11.

On the other hand, given the relation (10) between the p and ϕ we can argue that $\nabla\phi \approx 0$ at the immersed boundary, explaining why in this region the velocity field remains essentially unchanged by the correction step (8).

Before concluding this appendix we wish to mention an additional numerical experiment in which for every Δt the steps 7–10 where iterated until the difference between $\mathbf{u}^{l+1}$ and $\mathbf{V}$ was reduced to round-off error. This experiment was run for the case of the vortex ring formation and, although the computational time considerably increased (about of a factor 6), the results were indistinguishable from those previously obtained.

ACKNOWLEDGMENTS

We thank G. Carnevale for having revised the final version of the manuscript. The problem of the IC piston was solved during the CTR Summer Program 1998 (NASA Ames/Stanford University to which the authors are grateful for the financial support. The help of M. Fatica during the CTR Summer Program is gratefully acknowledged. We are also indebted to D. C. Haworth for providing us with the geometry of the piston and the experimental data. We thank S. Leonardi for the discussion of the results and for having tested a problem in the "volume fraction" interpolation procedure. The Italian authors aknowledge the grants of the the MURST (Ministero dell'Università e della Ricerca Scientifica e Tecnologica). Finally one of us R.V. thanks ERO (European Research Office) for a financial support under Contract AMXIP-ER-RI/R&D 8609-AN-06 (70-1s).

REFERENCES

1. C. S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* **10**, 252 (1972).
2. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
3. D. M. McQueen and C. S. Peskin, Shared-memory parallel vector implementation of the immersed boundary method for the computation of blood flow in the beating mammalian heart, *J. Supercomp.* **11**, 213 (1997).
4. D. M. McQueen and C. S. Peskin, A three-dimensional computational method for blood flow in the heart. II. Contractile fibers, *J. Comput. Phys.* **82**, 289 (1989).
5. M. Briscolini and P. Santangelo, Development of the mask method for incompressible unsteady flows, *J. Comput. Phys.* **84**, 57 (1989).
6. D. Goldstein, R. Handler, and L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* **105**, 354 (1993).
7. E. M. Saiki and S. Biringen, Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method, *J. Comput. Phys.* **123**, 450 (1996).
8. D. Goldstein, R. Handler, and L. Sirovich, Direct numerical simulation of the turbulent flow over a modelled riblet covered surface, *J. Fluid Mech.* **302**, 333 (1995).
9. D. Goldstein and T.-C. Tuan, Secondary flow induced by riblets, *J. Fluid Mech.* **363**, 115 (1998).
10. E. M. Saiki and S. Biringen, Spatial numerical simulation of boundary layer transition: Effects of a spherical particle, *J. Fluid Mech.* **345**, 133 (1997).
11. K. M. Arthurs, L. C. Moore, C. S. Peskin, F. B. Pitman, and others, Modeling arteriolar flow and mass transport using the immersed boundary method, *J. Comput. Phys.* **147**, 402 (1998).

12. J. Mohd-Yusof, *Combined Immersed Boundaries/B-Splines Methods for Simulations of Flows in Complex Geometries*, CTR Annual Research Briefs, NASA Ames/Stanford University, 1997.
13. R. Verzicco and P. Orlandi, Wall/vortex ring interactions, *App. Mech. Rev. ASME* **49**, 447 (1996).
14. K. Shariff and R. D. Moser, Two-dimensional mesh embedding for B-spline methods, *J. Comput. Phys.* **145**, 471 (1998).
15. A. P. Kravchenko, P. Moin, and R. D. Moser, Zonal embedded grids for numerical simulations of wall-bounded turbulent flows, *J. Comput. Phys.* **127**, 412 (1996).
16. B. Fornberg, Steady viscous flow past a sphere a high Reynolds number, *J. Fluid Mech.* **190**, 471 (1988).
17. R. Mittal, Planar symmetry in the unsteady wake of a sphere, *AIAA J.* **37**, 388 (1999).
18. G. K. Batchelor, *An Introduction to Fluid Mechanics* (Cambridge Univ. Press, Cambridge, UK, 1967).
19. A. P. Morse, J. H. Whitelaw, and M. Yanneskis, *Turbulent Flow Measurement by Laser Doppler Anemometry in a Motored Reciprocating Engine*, Report FS/78/24, Department Mech. Eng., Imperial College, 1978.
20. R. Verzicco, J. Mohd-Yusof, P. Orlandi, and D. C. Haworth, LES in complex geometries using boundary body forces, in *Proc. of the 1998 CTR Summer Program, VII, 1999*, p. 171.
21. M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, A dynamic subgrid-scale eddy viscosity model, *Phys. Fluids A* **3**, 1760 (1991).
22. D. K. Lilly, A proposed modification of the Germano subgrid-scale closure method, *Phys. Fluids A* **4**, 633 (1992).
23. D. C. Haworth, Large-eddy-simulation of in-cylinder flows, in *Multidimensional Simulation of Engine Internal Flows* (IFP, Rueil-Malmaison, France, 1998).
24. D. C. Haworth and K. Jansen, Large-eddy-simulation on unstructured deforming meshes: Towards reciprocating IC engines, *Comput. & Fluids*, in press.
25. A. M. Roma, C. S. Peskin, and M. J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* **153**, 509 (1999).
26. R. Verzicco and P. Orlandi, A finite-difference scheme for three dimensional incompressible flows in cylindrical coordinates, *J. Comput. Phys.* **123**, 403 (1996).
27. J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics* (Springer-Verlag, New York, 1996).