Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 215 (2006) 12–40

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries

Jianming Yang, Elias Balaras *

Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA

Received 31 October 2004; received in revised form 7 June 2005; accepted 19 October 2005
Available online 19 December 2005

Abstract

A non-boundary-conforming formulation for simulating complex turbulent flows with dynamically moving boundaries on fixed Cartesian grids is proposed. The underlying finite-difference solver for the filtered incompressible Navier–Stokes equations is based on a second-order fractional step method on a staggered grid. To satisfy the boundary conditions on an arbitrary immersed interface, the velocity field at the grid points near the interface is reconstructed using momentum forcing without smearing the sharp interface. The concept of field-extension is also introduced to treat the points emerging from a moving solid body to the fluid. Laminar flow cases and large-eddy simulations (LES) are presented to demonstrate the formal accuracy and range of applicability of the method. In particular, simulations of laminar flow induced by the harmonic in-line oscillation of a circular cylinder in quiescent fluid, and from a transversely oscillating cylinder in a free-stream are presented and compared to reference simulations and experiments. LES of turbulent flow over a traveling wavy wall and transitional flow through a bileaflet prosthetic heart valve are also shown. All results are in very good agreement with reference results in the literature.

Keywords: Large-eddy simulation; Immersed-boundary method; Fluid–structure interaction; Moving boundaries; Cartesian grid; Finite-difference method

1. Introduction

Recently there has been renewed interest in the development of non-boundary-conforming methodologies for the solution of the Navier–Stokes equations. In such methods the requirement that the grid conforms to a solid boundary is relaxed, and the effect of a complex object on the flow is introduced through proper treatment of the solution variables at the grid cells in the vicinity the body. The basic advantage of these formulations is the simplification of grid generation, especially in cases of moving boundaries where the need for regeneration or deformation of the grid is eliminated. In addition, efficient Cartesian solvers can be directly applied to complex flow problems. Both the above features are particularly attractive for Direct Numerical

* Corresponding author. Fax: +1 301 314 9477.
E-mail address: balaras@eng.umd.edu (E. Balaras).

doi:10.1016/j.jcp.2005.10.035

Simulations (DNS) or large-eddy simulations (LES) of turbulent and transitional flows, where the use of highly efficient, energy conserving solvers is imperative for accurate computations. It is therefore conceivable that successful integration of non-boundary-conforming strategies with robust Cartesian or cylindrical coordinate solvers developed for DNS/LES will open a wide new area of applications for these tools. Example applications include a variety of low and moderate Reynolds number turbulent flow problems from engineering, biology, and medicine, where fluid/structure interactions are central to the dynamics of the flow.

Over the past decades a variety of non-boundary-conforming methods with various degrees of accuracy and complexity have been proposed. The so-called immersed-boundary formulation pioneered by Peskin [19] represents a family of methods where a set of body forces is used to represent the effect of an object to the flow. Initially the method was used to study fluid–structure interaction problems in the cardiovascular circulation [20,21]. In these computations the vascular boundary was modeled as a set of elements linked by springs. As a result the forces required to enforce boundary conditions could be evaluated in a straightforward manner (i.e. Hooke's law). In the case of rigid immersed bodies, however, the corresponding forces are not known a priori and must be calculated by some feedback algorithm. Lai and Peskin [15] suggested a formulation where the body is allowed to move a little – rather than being fixed – by connecting it to a very stiff spring. They tested this approach for the flow around a cylinder with satisfactory results, although the specification of the stiffness constant is somehow ad hoc. Goldstein et al. [7] introduced an alternative approach where a feedback-forcing scheme is used to asymptotically enforce the desired boundary conditions on a solid boundary. Application of the method to three-dimensional computations of turbulent flow in plane and ribbed channels yield results in good agreement with reference data [8,9]. An advantage of immersed-boundary formulations is their straightforward implementation in existing solvers (modifications are confined to the RHS of the equations of motion). On the other hand, the need for a smooth transition between the fluid and the solid body spreads the forcing function over several grid cells and introduces some blurring between the two regions. This feature can decrease the order of accuracy of the scheme near the body or increase the resolution requirements making their use in turbulent flows problematic.

Another class of methods which does not suffer from the "blurring" mentioned above are the so-called Cartesian or cut-cell formulations. In this case the solid boundary is tracked as a sharp interface and the grid cells at the body interface are modified according to their intersections with the underlying Cartesian grid. The discrete operators at these cells are then modified to reflect the desired boundary conditions. Successful applications of such methods in two-dimensional flows with stationary boundaries can be found in [33,3,23,27,31]. However, due the large number of possible intersections between the grid and the boundary a variety of interface-cells is generated leading to an equally large number of special treatments. Also in complex configurations the unavoidable generation of irregularly shaped cells with very small size can have an adverse impact on the conservation and stability properties of the solver. Recently Ye et al. [31] suggested a cell merging scheme to address this problem. This formulation was also extended to treat moving boundaries with good results for a variety of two-dimensional problems [29]. The extension of the methodology in complex three-dimensional configurations remains to be investigated.

Recently, Fadlun et al. [6] introduced an embedded-boundary method, which shares a number of features with both approaches discussed above. As in immersed-boundary formulations the method still utilizes a force-field to enforce boundary conditions. In this case, however, the forces are not specified in the continuous space by means of some physical arguments, but rather in the discrete space by directly requiring the solution to respect the desired boundary conditions. This process is equivalent to a local reconstruction of the solution near the interface and enforces the desired boundary conditions "exactly", as in cut-cell formulations. The encouraging results reported in [30,6] together with the straightforward implementation of the method in existing Navier–Stokes solvers, motivated a number of recent studies, where alternative embedded-boundary formulations based on the principles outlined above have been proposed. The main difference between them is the way the solution is reconstructed near the interface. In [6,1], for example, the solution is reconstructed at the fluid nodes closest to the immersed boundary (fluid points with at least one neighbor in the solid phase). In the former study an one-dimensional interpolation scheme along an arbitrary grid line is used for this purpose, while in the latter the reconstruction is performed along the well-defined line normal to the interface. In [14,16], or [26] on the other hand, the solution is reconstructed at 'ghost-cells', which are points inside the solid phase with at least one neighbor in the fluid phase. Both the above strategies have the velocity boundary conditions ‘implicitly’ build into the reconstruction stencil, and therefore will potentially result in methodologies of comparable overall accuracy. The former strategy, however, which is the one adopted in the present work has some advantages in cases with moving boundaries as it will be discussed in the following sections.

A related issue, which also has implications on the overall accuracy of the approach, is the way the direct-forcing is incorporated into the time advancement scheme (fractional-step methods are almost exclusively used in the all the above studies). In [6], for example, the use of an one-dimensional interpolation strategy facilitates the imposition of boundary conditions to the predicted velocity by directly modifying the coefficients of the standard linear system. In [1] where a multi-dimensional reconstruction is used in the framework of an explicit time advancement scheme the same result can be simply achieved by directly modifying the predicted velocity field. As a result in both cases the actual value of the forcing function is not explicitly computed and the boundary conditions on the immersed interface are satisfied by a direct modification of the discrete operators. When a semi-implicit scheme (i.e. third-order Runge–Kutta scheme for advective terms, and Crank–Nicolson for the viscous terms) is used, however, the multi-dimensional reconstruction stencil involves values form the yet to be determined predicted velocity. To implicitly satisfy the boundary conditions, as in [6,1], one needs to modify the standard linear system of equations in the predictor step, which is now going to become a sparse system with a costly solution. To alleviate this problem, Kim et al. [14], proposed an approach where the forcing is estimated by taking a preliminary fully explicit ‘predictor’ step (a forward Euler step for the viscous terms in their case). The estimated forcing function is then incorporated into the RHS of the regular linear system to enforce the desired boundary conditions. Balaras and Yang [2] used a similar approach in the framework of a semi-implicit, finite-difference, fractional step method in cylindrical coordinates with good results.

The boundary motion/deformation over a fixed grid is usually the source of additional complications in most of the non-boundary-conforming strategies that were discussed in the previous paragraphs. In classical immersed-boundary formulations [19,15], such problems are usually minimal due to the smooth transition from the solid to the fluid. In cut-cell [33,3,23,27,31] or embedded-boundary formulations [6,14,26,1], however, complications are encountered due the fact that the role of the Eulerian grid points near the interface changes from timestep to timestep (for example a point in the solid body can became a point in the fluid at the next timestep and vise versa), as the body moves through the fixed grid. As a result the velocity and pressure for some points in the flow will get non-physical values due to their previous association with the solid phase. In the framework of cut-cell formulations Udaykumar et al. [29] proposed a cell merging scheme to handle the cells that emerge from the body which they call ‘freshly-cleared’ cells. In embedded-boundary formulations on the other hand, no systematic study that identifies and addresses problems associated with large boundary motions has been reported. Due to the explicit dependence of such methods on the details of the adopted numerical method it is difficult to formulate a general approach. In the case of ‘ghost-cell’ methods, for example, as the body moves through the fixed grid some of the ‘ghost-cells’ will emerge into the fluid and will became fluid cells. Since they were previously in the solid they have no history in the fluid phase and no physically realizable value for the velocity and pressure at the previous timestep(s). As a consequence their treatment to some degree would have to be ad hoc. In formulations where the solution is reconstructed at the fluid nodes closest to the boundary (i.e. [6]), the points that emerge from the solid become the boundary points that are central to the reconstruction procedure, and therefore their history in the fluid phase is irrelevant. The points that require special treatment in this case are the boundary points that move further into the fluid.

The latter approach is the one that will be discussed in detail in the present study, where a variance of the embedded-boundary formulation proposed in [1] will be extended to moving boundary problems. To address the problems due to boundary motion we will propose a field-extension strategy that practically extends the pressure and velocity fields into the solid body to implicitly treat problematic grid cells in a robust manner. We will also discuss a straightforward method to compute the local force distribution on a complex body that is based on the same reconstruction stencil used for the velocity field. As it will be demonstrated in the results section the reconstructed local forces are in good agreement with reference results obtained with boundary-conforming formulations. The paper is organized as follows: In Section 2.1 the basic Cartesian solver is described. The interface tracking scheme and solution reconstruction procedure near a solid boundary are given in Sections 2.2 and 2.3 respectively. The problems arising from the boundary motion and the proposed field-extension treatment are discussed in Section 2.4. In the results section a series of applications of increasing complexity are considered. First, the formal accuracy of the numerical method for moving interfaces is established. Then, laminar flow computations around a moving cylinder in a quiescent fluid and in a cross-flow are discussed. The main flow features and the distribution of forces on the cylinder's surface are compared to corresponding results from boundary-conforming formulations and experiments. Then, LES of flow over a traveling wavy wall are presented in comparison to reference DNS results, to establish the accuracy of the method in turbulent flows. To demonstrate the robustness and applicability of the method in highly unsteady turbulent flows that involve multiple moving complex boundaries, the flow around a bileaflet prosthetic heart valve is also presented. Finally, in Section 4 a summary and conclusions are given.

2. Numerical methods

2.1. Governing equations and basic solver

In the LES approach a spatial filtering operation separates the large, energy carrying eddies, which are directly resolved, from the small scales, which are modeled. In the present finite-difference implementation a top-hat filter is implicitly applied by the discrete operators. The resulting equations governing the evolution of the large scales for incompressible flow are

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \frac{1}{Re}\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} + f_i, \tag{1}$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \tag{2}$$

where x_i $(i = 1, 2, 3)$ are the Cartesian coordinates and $\bar{u}_i$ are the corresponding velocity components normalized by a reference velocity U; $\bar{p}$ is the pressure normalized by ρU^2, where ρ is the density of the fluid; f_i represents an external body force field. The Reynolds number is defined as $Re = UL/\nu$, where L is the reference length scale and ν is the kinematic viscosity of the fluid. The effect of the unresolved scales upon the resolved part of turbulence appears in the subgrid-scale (SGS) stress term, $\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j$, which needs to be parameterized. In all LES reported in the present study the Lagrangian dynamic SGS model is used for this purpose [17]. Details on the present implementation of the model in the framework of a non-boundary conforming formulation can be found [1].

The governing equations (1) and (2) are discretized using second-order central differences on a staggered grid and advanced in time using a fractional step method. The time advancement scheme is a low-storage third-order Runge–Kutta (RK3) scheme, where all terms in the RHS of momentum equation are advanced explicitly:

$$\begin{aligned} \frac{\hat{u}_i^k - u_i^{k-1}}{\Delta t} &= \mathrm{RHS}_i^k + f_i^k \\ &= \gamma_k H(u_i^{k-1}) + \rho_k H(u_i^{k-2}) - \alpha_k \frac{\partial p^{k-1}}{\partial x_i} + f_i^k, \end{aligned} \tag{3}$$

$$\frac{\partial^2 \phi^k}{\partial x_i \partial x_i} = \frac{1}{\alpha_k \Delta t}\frac{\partial \hat{u}_i^k}{\partial x_i}, \tag{4}$$

$$u_i^k = \hat{u}_i^k - \alpha_k \Delta t \frac{\partial \phi^k}{\partial x_i}, \tag{5}$$

$$p^k = p^{k-1} + \phi^k, \tag{6}$$

where k is the substep index, which ranges from 1 to 3, $\hat{u}_i^k$ is the intermediate velocity and ϕ is the scalar used to project $\hat{u}_i^k$ into a divergence-free space. H is a spatial operator containing the convective, viscous and SGS terms. f_i^k is the momentum forcing that will be used to enforce the desired boundary conditions on an immersed interface and will be discussed in the following sections. Δt is the timestep, and the RK3 coefficients are $\alpha_1 = 8/15$, $\gamma_1 = 8/15$, $\rho_1 = 0$; $\alpha_2 = 2/15$; $\gamma_2 = 5/12$, $\rho_2 = -17/60$; $\alpha_3 = 1/3$, $\gamma_3 = 3/4$, $\rho_3 = -5/12$. For all the cases considered in the present study the pressure Poisson equation (4) is reduced to a series two-dimensional problems using trigonometric expansions which are then solved with direct methods [25]. The direct solver is very efficient, but requires the use of a uniform grid in one coordinate direction. For variety of practical flow problems involving moving boundaries this may not a severe constraint since usually isotropic grids along the path of the body's motion need to be utilized anyway. However, there are applications – especially from the area of external flows – where the requirement of using uniform grids in one direction could make the computation prohibitively expensive. In such cases one can use cylindrical coordinates where still a direct pressure solver can be adopted (see for example [14,2]), or iterative solvers [26] without any further modifications to the proposed approach.

2.2. Interface-grid relation

In the present study a front tracking scheme originally proposed for multi-phase flow problems is used for the description of the fluid/solid interface [28]. An example in two-dimensions is shown in Fig. 1a, where an immersed interface, Ψ, is represented by a series of interfacial marker particles defined by arc-length coordinates $\vec{X}(s,t)$. The marker particles are equally spaced on the interface with spacing approximately equal to the local grid size. For each marker particle a local description of the interface can be obtained by fitting quadratic polynomials to particle (i) and its two neighbors ($i-1$) and ($i+1$). Bi-splines can be used in three-dimensions.

Next, the relation between the interface and underlying Eulerian grid is established. Central to this process is the computation of the normal from each of a packet of grid points in the vicinity of a marker particle to the interface. This information is used to determine whether a grid point belongs to the fluid or solid phase. A detailed description of the above algorithm is given in [1]. An example of the result of such a tagging process is shown in Fig. 1b where all Eulerian grid points are split into three different categories: (a) *forcing points*, which are grid points in the fluid phase that have one or more neighboring points in the solid phase; (b) *solid points*, which are all the points in the solid phase; (c) *fluid points*, which are all the remaining points in the fluid phase. In the solution procedure, the *fluid points* are the unknowns, the *forcing points* are essentially boundary points, while the *solid points* do not influence the rest of the computation. For a stationary boundary the above tagging process is done only once at the beginning of the computation. For a moving body the process is repeated at each timestep.

2.3. Treatment of stationary immersed boundaries

The velocity boundary conditions on a complex body, which is not aligned with the grid, are imposed using a variance of the embedded-boundary approach discussed in the introduction. In this case a set of point forces directly derived from discrete problem is introduced for this purpose. To demonstrate the basic idea let us

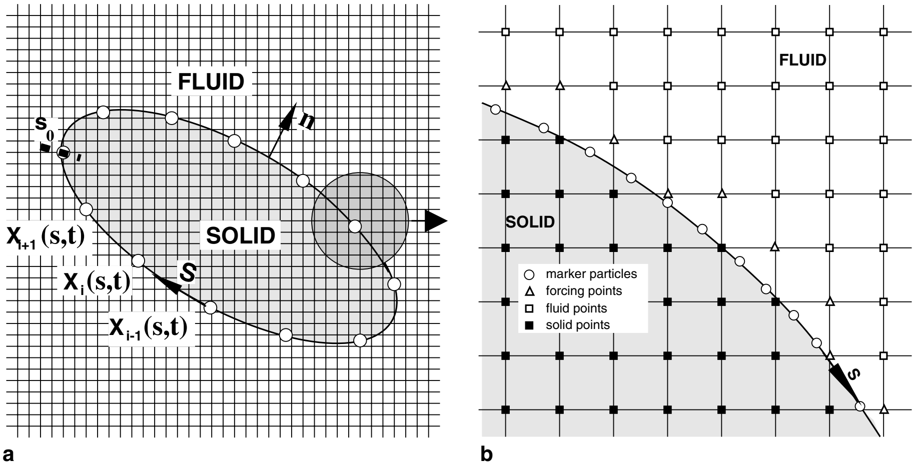


Fig. 1. (a) Parameterization of the immersed interface using marker particles; (b) zoom in the vicinity of interface where the different types of Eulerian grid nodes are shown.

examine the special case where an Eulerian grid point coincides with the immersed interface, Ψ, on which a Dirichlet boundary condition, u_Ψ, has to be imposed (see Case 2 in Fig. 2a). The magnitude of the forcing function that will enforce the above boundary condition can be obtained from Eq. (3) by simply setting, $\hat{u}_i^k = u_\Psi$, and solving for f_i^k:

$$f_i^k = \frac{u_\Psi - u_i^{k-1}}{\Delta t} - \mathrm{RHS}_i^k, \tag{7}$$

In the framework of the explicit fractional-step method discussed in the previous section, the use of f_i, given by Eq. (7) will enforce the proper boundary conditions on the predicted velocity, $\hat{u}_i^k$ (substitution of (7) into (3) gives $\hat{u}_i^k = u_\Psi$). This choice does not compromise the overall temporal accuracy of the splitting scheme, and u_i^k will also satisfy the same boundary condition to the order of Δt^2 [14]. We should also note that the computation of f_i in the case of a semi-implicit scheme (i.e. the commonly used RK3/Crank–Nicolson for the advective and viscous terms respectively) would required some additional steps due to its dependence on $\hat{u}_i^k$ that in this case is also an unknown. Following Kim et al. [14], a straightforward way to address this issue with a minimal increase in the overall cost is to take a preliminary explicit 'predictor' step for the purpose of evaluating the forcing function, f_i, which can then be incorporated into the RHS of the regular linear system to enforce the desired boundary conditions. Further details can be found in a recent study by Balaras and Yang [2], where a similar approach is used in the framework of a semi-implicit, finite-difference, fractional step method in cylindrical coordinates.

The above idealized case, although it demonstrates the basic approach, it rarely appears in practical applications, where the Eulerian grid nodes almost never coincide with the immersed boundary. In such case, f_i has to be computed at grid points near – and not exactly on – the interface. For reasons that will became apparent in the next section, we will compute f_i at all fluid points that have at least one neighbor in the solid phase (these points were labeled forcing points in the previous section). In this case, however, u_Ψ is not known and has to be reconstructed using information from the interface and the surrounding velocity field. The reconstruction scheme, that is used in the present study, is based on earlier work reported in [1], where u_Ψ at the forcing points is computed by means of linear interpolation along the well-defined line normal to the boundary. The approach is illustrated in Fig. 2a: initially a virtual point is located along the normal; then, the virtual point together with the point on the interface is used to perform the linear interpolation to find u_Ψ in Eq. (7) at the location of the forcing point. The velocity at the virtual point is computed from the surrounding fluid nodes using bi-linear interpolation. In this last step, one also has to impose the constrain that the stencil does not involve other forcing points, which can be easily achieved by gradually moving the virtual point further away from the boundary (see for example Case 1 in Fig. 2a). Implementation details and a series of applications in laminar and turbulent flows are given in [1].

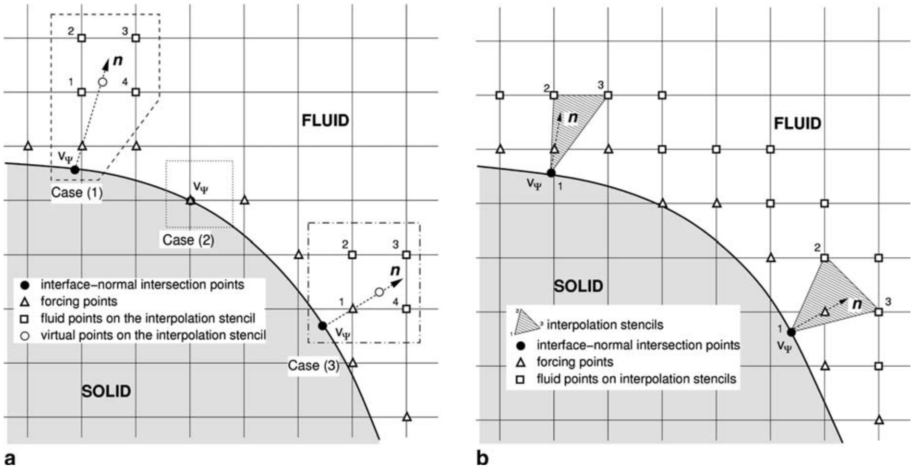

FLUID
SOLID
Case (1)
Case (2)
Case (3)
V_Ψ
n
1
2
3
4
● interface–normal intersection points
△ forcing points
□ fluid points on the interpolation stencil
○ virtual points on the interpolation stencil
a
FLUID
SOLID
interpolation stencils
● interface–normal intersection points
△ forcing points
□ fluid points on interpolation stencils
b

Fig. 2. (a) Interpolation stencil in [1]. Cases (1) and (3) illustrate two possible interpolation stencils depending on the interface topology and local grid size; (b) corresponding interpolation stencil in present approach.

In the present work we will use a variance of the above method that is better suited to moving boundary problems. As it will be demonstrated in the results section, the proposed method is as accurate and robust as the one in [1]. It utilizes, however, a more compact stencil and allows for the computation of all components of the strain rate tensor on the interface in a straightforward manner. The latter feature is particularly attractive in fluid/structure interaction problems, while the former simplifies parallelization. The similarities between the two approaches became apparent when comparing part (a) and (b) in Fig. 2. In the present approach we basically omit the virtual point and replace it with two points on an x-grid line, y-grid line, or along the diagonal. Consequently, the interpolation procedure is now a single-step process that involves two points from the fluid and one on the interface (shaded area in Fig. 2b). The interpolation coefficients for the case of linear interpolation can be computed as follows: lets us assume that any variable ϕ in the two-dimensional space can be written in the following form:

$$\phi = b_1 + b_2 x + b_3 y. \tag{8}$$

The coefficients b_1, b_2, and b_3 in Eq. (8) can be found by solving the following system:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}, \tag{9}$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3) in the 3×3 matrix $\mathbf{A}$ are the coordinates of the three points in the interpolation stencil shown in Fig. 2b. For the case of a stationary body, the inversion of matrix, $\mathbf{A}$, at every forcing point is performed in the beginning of the simulation and then stored in memory. For moving boundaries, however, the inversion is performed at every time-step, since new forcing points emerge as a consequence of the boundary motion. The extension of above interpolation procedure to three dimensions is straightforward (one needs to add a term of the form, $b_4\, z$, to the polynomial in Eq. (8) to reflect the dependence of the solution near the interface on an additional spatial dimension. Higher-order reconstructions can also be readily achieved using the same overall procedure (see for example [16,26]).

Another issue that is only relevant to the LES reported in this study, is the treatment of the turbulent viscosity near the immersed boundary. The methodology discussed above is directly applicable to the computation of laminar flows, or turbulent flows using DNS. In LES the computation of ν_T near the interface is not straightforward especially in the framework of the dynamic eddy viscosity models like the one used in the present work. This is due to the fact that the evaluation of all test-filtered quantities in the vicinity of the immersed boundary, involves points from the interior of the solid body. To avoid complex modifications of the filtering operator at these points, a reconstruction procedure similar to that used for the velocity field is also applied on ν_T. The implementation details together with extensive testing in LES over immersed boundaries are given in [1].

2.4. Treatment of moving immersed boundaries: field-extension

The general algorithm outlined in Sections 2.2 and 2.3 is directly applicable to moving boundary problems provided that the grid-interface relation and the interpolation coefficients are re-evaluated every time the location of the interface is updated. In moving boundary problems, however, complications that are usually related to the time advancement scheme can also arise. This is due to the fact that the role of the Eulerian grid points near the interface changes from timestep to timestep (for example a forcing point can became a fluid point at the next timestep and vise versa), as the solid body moves through the fixed grid. For the time advancement scheme used in the present study, the evaluation of the RHS of the momentum equation at substep k requires physical values of the velocity vector and pressure, as well as their derivatives from substep $k - 1$ at all fluid points (see Eq. (3)). Due to the fact that the interface changes locations, it is possible that some of the required values from substep $k - 1$ are not physical.

To identify the specific cases that are potential sources of error during the time advancement procedure, let us consider a simple, one-dimensional problem where the solid-phase: (1) encroaches upon the fluid phase, and (2) withdraws from the fluid phase as shown in Fig. 3a and b respectively. Due to the *CFL* restriction of the present scheme the interface cannot move by more than one gird cell in each substep, resulting to two possible changes in the flags of the points near the interface. For case 1: (a) forcing points ⇒ solid points, and (b) fluid points ⇒ forcing points. Both of the above do not cause problems to the time advancement scheme. For case 1a, the solution at t^k in all forcing points will be reconstructed anyway, and does not depend on the history of the velocity or pressure field. Case 1b, where Eulerian grid points move into the solid, also does not cause problems since the interior treatment of the body does not influence the solution.

In case 2, where the solid phase withdraws from the fluid (see Fig. 3b), the possible flag changes are the following: (a) solid points ⇒ forcing points; (b) forcing points ⇒ fluid points. The former case does not generate any problem for the same reason that was discussed above (case 1a). In case 2b, however, it is apparent from Eq. (3) that for all the newly defined fluid points the RHS which involves derivatives from previous substeps will not be correct. This is because these points at substep $k-1$ were forcing points and although they have the correct values for the velocity and pressure, their derivatives will not have physical values since they involve points from the solid phase. These unphysical values can introduce spurious vorticity near the boundary, leading to large errors.

To avoid forbiddingly complex special treatments of the computation of these derivatives every time such a change of flag is detected during the time advancement procedure, we propose a field-extension procedure, in which the velocity and pressure fields are "extended" in the solid phase at the end of each substep. Practically, with this procedure the velocity and pressure fields are extrapolated at the Eulerian nodes in the solid that have at least one neighbor in the fluid phase. For the purpose of discussion we will call these points pseudo-fluid points (see Fig. 4a). This way, not only the velocity and pressure at the forcing points at substep $k-1$, but also their derivatives will have physical values, eliminating problems in the computation of the RHS in the next substep. The values of the velocity and pressure at the pseudo-fluid points is reconstructed using a procedure which is similar to the one used for the forcing points that is described in Section 2.3. Central to this procedure is again the point on the interface where the normal from the corresponding pseudo-fluid point intersects it. In addition to this point the stencil involves two more points form the fluid phase as shown in Fig. 4b. The interpolation coefficients are computed by solving a system of equations identical to that given by Eq. (9). We should also note that due to the staggering of the mesh the selection of points on the pressure grid where the 'field-extension' is performed is not only based on their relative location with the interface but also on their association to velocity values that are unphysical. As a result the pressure is also 'extended' to some nodes on the pressure grid that would have been classified as 'boundary' nodes on the velocity grid. The overall field-extension procedure is similar to the generalized ghost-cell approach discussed in [13,26]. In both these cases, however, extrapolation is used to enforce boundary conditions on a stationary boundary.

Having established the treatment of all Eulerian points in the vicinity of a stationary or moving interface, we can summarize the overall algorithm as follows:

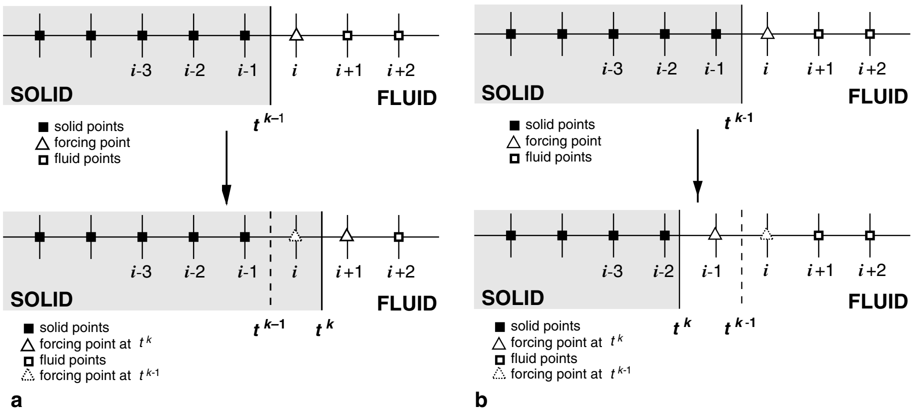


Fig. 3. A simplified one-dimensional model problem that demonstrates the possible changes of flags of the Eulerian grid nodes near a moving interface: (a) solid phase encroaches upon the fluid phase; (b) solid phase withdraws from the fluid phase.

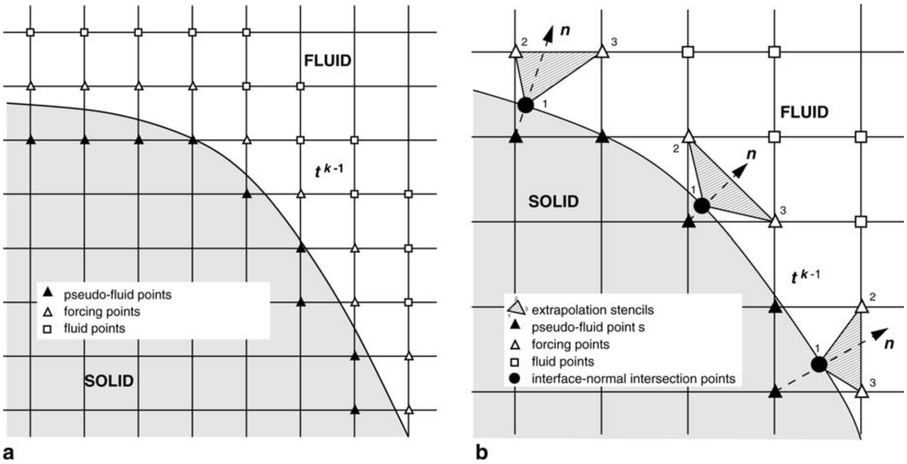


Fig. 4. The field-extension procedure: (a) identification of the pseudo-fluid points at substep t^{k-1} where the solution will be extended; (b) possible extrapolation stencils.

- Given the location of the interface at step k, identify fluid, forcing, solid, and pseudo-fluid points on the Eulerian grid. This procedure needs to be performed only once in the beginning of the computation for problems with stationary boundaries.
- Calculate the predicted velocity field $\hat{u}_i^k$ from Eq. (3).
- Reconstruct the predicted velocity $\hat{u}_i^k$ at the forcing points as discussed in Section 2.3.
- Solve the pressure Poisson equation (Eq. (4)).
- Update the velocity field to u_i^k (Eq. (5)) and pressure field to p^k (Eq. (6)).
- Perform the field-extension procedure described above for the pseudo-fluid points. This step is omitted for stationary boundary problems.

As it will be demonstrated in the results section the above algorithm is very robust and satisfies the boundary conditions on an immersed interface “exactly”, within the overall accuracy of the discretization scheme. Also, contrary to alternative reconstruction procedures (i.e. generalized ghost cell approaches) it does not require special treatment of the grid cells emerging from the solid, which is usually ad hoc (see [29] for a more detailed discussion on this issue). We should also note that in the above algorithm, boundary conditions for the pressure near the interface are not imposed directly, but they are essentially implicit into the RHS of the Poisson equation (4). To clarify this statement one can consider the projection of the momentum equation of a fluid particle on the solid boundary in the direction normal to the interface:

$$\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\cdot\mathbf{n} = -\nabla p\cdot\mathbf{n} + \frac{1}{Re}\nabla^2\mathbf{u}\cdot\mathbf{n}. \tag{10}$$

With the present reconstruction procedure the viscous term turns out to be zero as all velocity components are linearized in the vicinity of Ψ (see Sections 2.3 and 2.4), and the above reduces to

$$\frac{\partial p}{\partial n} = -\frac{\mathrm{D}\mathbf{u}}{\mathrm{D}t}\cdot\mathbf{n}, \tag{11}$$

Eq. (11) represents the boundary condition for the pressure which is usually enforced “directly” in boundary-conforming or cut-cell formulations involving moving boundaries (see for example [29]). In these cases, however, it is obtained from the projection of the inviscid momentum equation in the direction normal to the boundary. A discussion on the behavior of the pressure in embedded-boundary formulations is also given in [6]. In the results section, we will show that the above argument is sound and the pressure behaves correctly in the vicinity of the immersed boundaries.

2.5. Surface force calculation at the immersed interface

In the case of non-boundary-conforming formulations the computation of the local forces on the surface of the body is a non-trivial problem. A good discussion of this issue can be found in the study by Lai and Peskin [15], where the accuracy of several different force calculation algorithms is discussed in the framework of their immersed-boundary formulation. In the present work we propose a method to compute the local force distribution on a complex body that is based on the same reconstruction stencil used for the velocity field. In general, the force **f** per unit area on a surface element with an outward normal **n** can be written as

$$f_i = \tau_{ji}n_j = \left[-p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]n_j, \tag{12}$$

where f_i is the surface force component in x_i direction, τ_{ji} is stress tensor, and n_j is the direction cosine of **n** in x_j direction. To compute **f** from Eq. (12) one would need to find p and $\partial u_i/\partial x_j$ on the body surface. In the framework of the present scheme, the latter term can be computed in a straightforward manner using the stencil and interpolation coefficients that were utilized to reconstruct the velocity field near the interface (Eqs. (8) and (9)). Due to grid staggering, however, the derivatives for each velocity component are computed on different points on the interface. Fig. 5a, for example, shows the staggered grid arrangement of all flow variables in the vicinity of a solid body and the corresponding points on the interface where $\partial u_i/\partial x_j$ is computed for each velocity component. In a detailed view around a u velocity point (see Fig. 5b), one can identify the interpolation stencil that was used in the reconstruction step. It involves points (2) and (3) from the flow, and point (1) on the interface. Considering Eq. (8) at point (1) and differentiating with respect to x_j, we can compute the desired velocity derivatives as follows:

$$\frac{\partial u}{\partial x} = b_2 \quad \text{and} \quad \frac{\partial u}{\partial y} = b_3, \tag{13}$$

where the coefficients b_2 and b_3 have been already computed from Eq. (9). In a similar manner, the derivatives for v and w can be found at the interface-normal intersection points of each component using the corresponding interpolation stencils.

The computation of the surface pressure, p, is a little more complicated if one is to avoid simple extrapolation from the interior, which can result in large errors. In the present study we have tested several different strategies and found that an accurate and cost/efficient approach to compute the surface pressure is the one that utilizes the momentum equation normal to the boundary (Eq. (11)), in addition to information from the

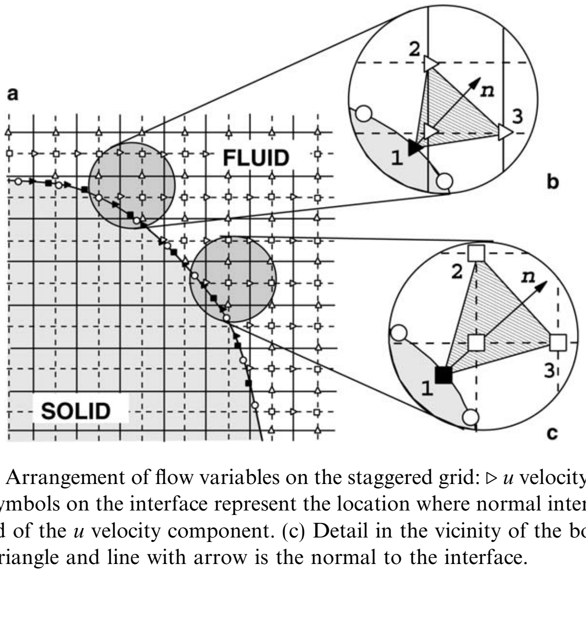


Fig. 5. Surface force calculation. (a) Arrangement of flow variables on the staggered grid: ▷ u velocity, △ v velocity, □ pressure, ○ marker particles. The corresponding filled symbols on the interface represent the location where normal intersects the boundary. (b) Detail in the vicinity of the boundary for the grid of the u velocity component. (c) Detail in the vicinity of the boundary for the grid of the pressure. Interpolation stencil is the shaded triangle and line with arrow is the normal to the interface.

interior of the flow. In particular, Eq. (11) can be used to obtain an estimate of $\partial p/\partial n$ on any point on the interface, since the material derivative in the RHS of can be computed directly from the known boundary velocities. The normal derivative of pressure can then be expressed as

$$\frac{\partial p}{\partial n}=\frac{\partial p}{\partial x}n_x+\frac{\partial p}{\partial y}n_y, \tag{14}$$

where n_x and n_y are the components of normal unit vector, $\mathbf{n}$, in x and y directions, respectively. Referring to the stencil shown in Fig. 5c – this stencil is build as if the pressure near the interface will be reconstructed in a way similar to the velocity field, even-thought it is only used for the surface pressure computation – differentiation of Eq. (8) at point (1) yields

$$\frac{\partial p}{\partial x}=b_2 \quad \text{and} \quad \frac{\partial p}{\partial y}=b_3. \tag{15}$$

Using Eqs. (14) and (15) we can rewrite Eq. (9) as follows:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}=\mathbf{A}^{-1}\begin{bmatrix} \frac{\partial p}{\partial n} \\ p_2 \\ p_3 \end{bmatrix}=\begin{bmatrix} 0 & n_x & n_y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}^{-1}\begin{bmatrix} \frac{\partial p}{\partial n} \\ p_2 \\ p_3 \end{bmatrix}. \tag{16}$$

We can now use Eq. (16) to compute b_1, b_2 and b_3, and then compute the surface pressure in a straightforward manner from Eq. (8). As it also mentioned above, due to the grid staggering the velocity derivatives and surface pressure have to be computed on different points on the interface. We then use cubic spline interpolation to compute the corresponding values at the locations of the marker particles. Finally the local forces can be computed from Eq. (12) at the marker particle location. Numerical integration is used to calculate integral quantities like drag and lift forces.

3. Results

To evaluate the accuracy of the proposed methodology several cases with increasing complexity have been considered. The first case, which simulates the two-dimensional flow around one element of a moving cylinder-array in a planar channel, is a demonstration of the formal accuracy of the method. Then, laminar flow problems involving prescribed motions of two-dimensional bodies are computed: the flow induced by the harmonic in-line oscillation of a circular cylinder in a quiescent fluid, and the flow from a transversely oscillating cylinder in a free-stream. Both cases are characterized by complex vortex–vortex and vortex–structure interactions and present a challenge to non-boundary conforming formulations. In addition, detailed experimental data and numerical results from boundary-fitted methods are available in the literature, allowing for a detailed validation. Then, LES of turbulent flow over a traveling wavy wall are also performed to demonstrate the accuracy and applicability of the method in LES of turbulent flows with dynamically moving boundaries. For this problem detailed DNS using a highly accurate boundary-fitted method will serve as the basis for the validation of the proposed approach. Finally to demonstrate the robustness and applicability of the method in complex turbulent flows that involve multiple moving boundaries, the flow around a bileaflet prosthetic heart valve is also presented.

3.1. Accuracy study

There are no cases in fluid dynamics that involve complex moving boundaries and have analytical solutions, to serve as reference for the formal accuracy study of the proposed method. For this reason we have designed a test problem which involves a body moving through a fixed grid, and performed a detailed numerical accuracy study. In particular, we have considered the flow around periodic array of cylinders moving with constant velocity $W=-1$ in a planar channel. A sketch of the computational domain and boundary condition setup is shown Fig. 6a. Non-slip conditions are used on the walls and periodic boundary conditions in the direction of motion. When the reference frame moves with the cylinder, the case shown in Fig. 6a is recovered, where the cylinder is stationary and the walls have a velocity of $W = 1$. The Reynolds number based on the diameter and velocity of the cylinder (or the velocity of the moving wall) is 100. The corresponding flow fields are visualized in Fig. 6b where streamlines are shown. A recirculation bubble is formed and restricted between two cylinders, which keeps the flow two-dimensional and steady. Note that for the moving cylinder a reference frame following the cylinder is used. In both cases, the flow is identical even though we have two distinct computational configurations.

Calculations on four levels of uniform grids were carried out: 30×30, 90×90, 150×150, and 450×450. The above choice – there is not the usual doubling of grid points in the next finer grid level – is due to the staggered variable arrangement. We have selected the specific grids in a way that every node on coarser grids can be directly located on the reference, finest, grid to avoid interpolations when comparing the two solutions. The L_1 and L_2 norms of the error, which measure the difference between the solutions from coarser grids and the reference grid for the moving body case, versus the corresponding grid resolution are shown in Fig. 7b. In Fig. 7a, the results from the fixed cylinder case are also presented. Lines representing the first- and second-order accuracy slopes are included. It is clearly shown that the error decreases with Δx^2, verifying the second-order accuracy of the present method. This means that, within the order of the accuracy of the discretization scheme, the moving boundaries are represented "exactly". By comparing Fig. 7b with Fig. 7a, we can see that the error of the moving cylinder case is of the same order as the fixed cylinder case.

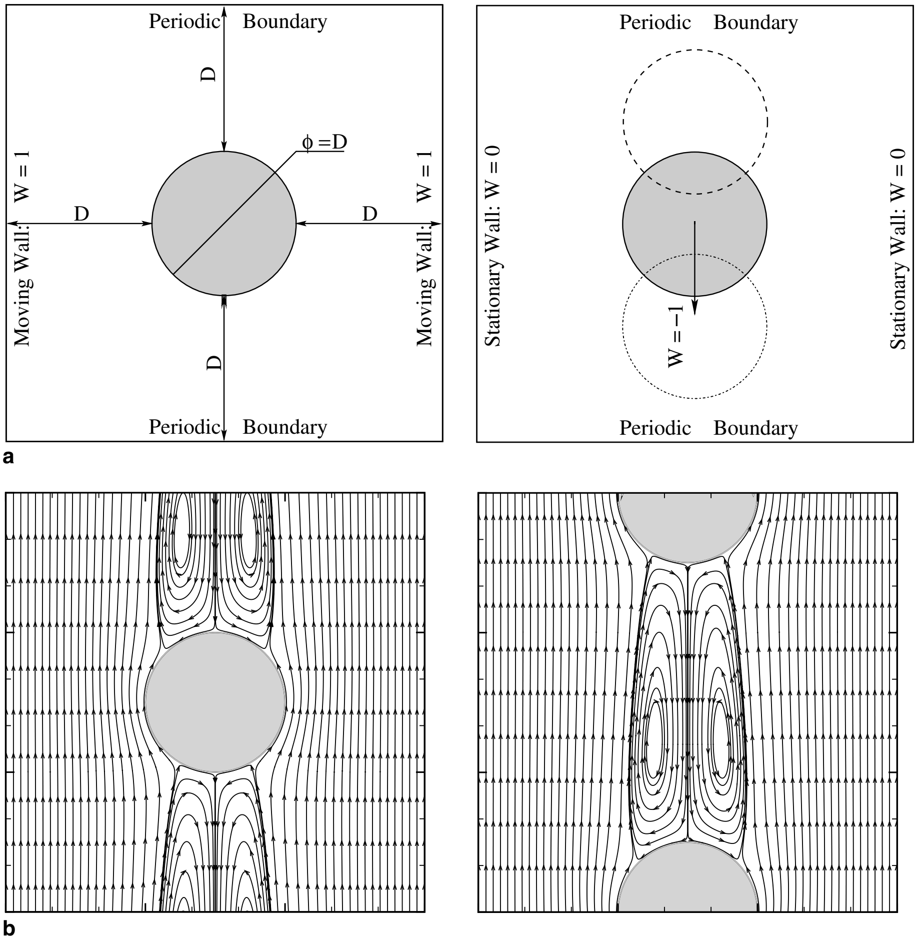


Fig. 6. Flow around a periodic array of cylinders in a plane channel: (a) sketch of the computational domain; (b) streamlines on a 150×150 uniform grid (a reference frame moving with the body is used for the moving cylinder case).

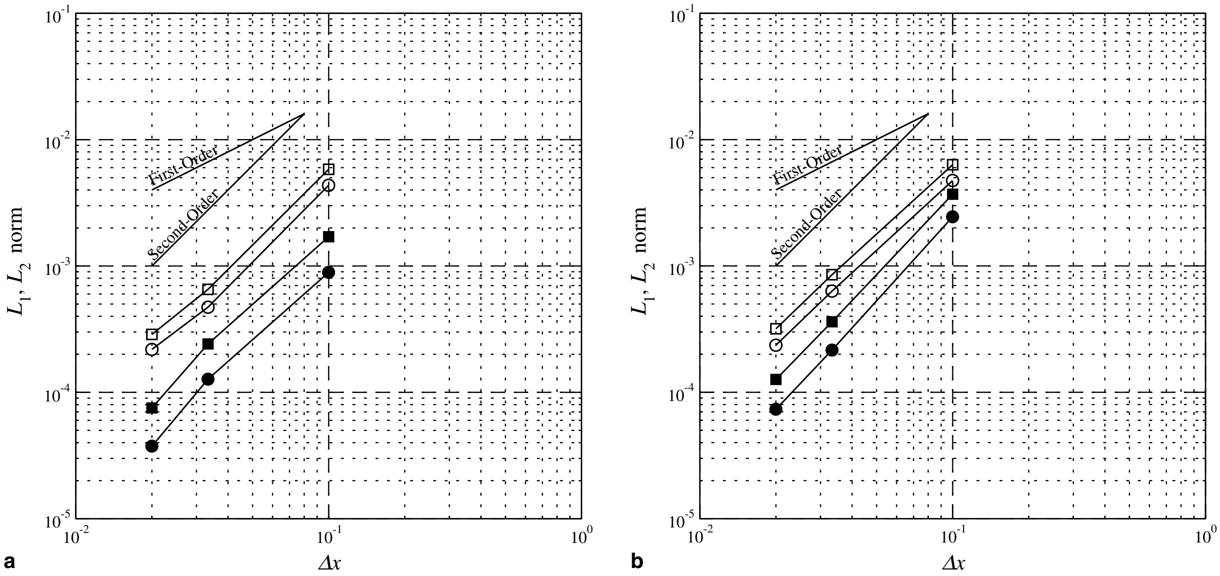


Fig. 7. (□) L_1 norm of the error; (○) L_2 norm of the error; open symbols are for w and filled for u: (a) fixed cylinder; (b) moving cylinder.

3.2. In-line oscillating cylinder in a fluid at rest

The interaction of an oscillating circular cylinder with a fluid at rest is a well-documented model-problem in the literature because of its relevance to a variety of applications. The two key parameters in this flow are the Reynolds number, $Re = U_{\max}D/\nu$, and the Keulegan–Carpenter number, $\mathrm{KC} = U_{\max}/fD$ ($U_{\max}$ is the maximum velocity of the cylinder, D is the diameter of the cylinder, ν is the kinematic viscosity of the fluid, and f is the characteristic frequency of the oscillation). The parametric space we selected in the present work corresponds to the LDA experiments and numerical simulations reported by Dütsch et al. [5]. In particular, the cylinder's translational motion is described by a simple harmonic oscillation, $x(t) = -A\sin(2\pi ft)$, where A is the amplitude of the oscillation. The Reynolds and Keulegan–Carpenter numbers were set to $Re = 100$ and $\mathrm{KC} = 5$ respectively. For this setup the flow remains two-dimensional with periodic vortex shedding. The size of computational domain is $50D \times 30D$ in x and z respectively, with the cylinder located at the center. Radiative boundary conditions are applied to all the far-field boundaries. Three different grid levels were used (240×120, 480×240, and 960×480) to investigate the sensitivity of the results to numerical resolution. Near the cylinder the grid is approximately uniform in both directions with local spacing of $0.05D$, $0.025D$, and $0.0125D$ respectively.

All computations were started from a quiescent field and the integration in time was performed until periodic vortex shedding was established. In Fig. 8 the pressure and vorticity contours at four different phase-angles are shown. One can observe the two fixed stagnation points at both ends of the cylinder that also define the axis of symmetry of the flow. As the cylinder moves to the left, two thin boundary layers develop on the upper and lower sides (see Fig. 8a), that eventually separate giving rise to two counter-rotating vortices of exactly the same strength. The vortex generation procedure stops as the body reaches its extreme left location, as shown in Fig. 8b. Then the cylinder moves backwards, and the same process takes place on its right side. During this part of the cycle, the vortex pair generated earlier is split and pushed away by the cylinder (see Fig. 8c). These results are in very good agreement with the corresponding results reported in [5], demonstrating that the present method can properly capture the dynamics of the vorticity field.

To further examine the accuracy of the proposed method, quantitative comparisons with the experimental results are considered. Fig. 9 shows the computed velocity profiles on the intermediate grid, at four x locations and three different phase-angles, in comparison with the experiments. The agreement is very good. We should also note that the results on the two finest grids (only the intermediate is shown here) are almost indistinguishable,

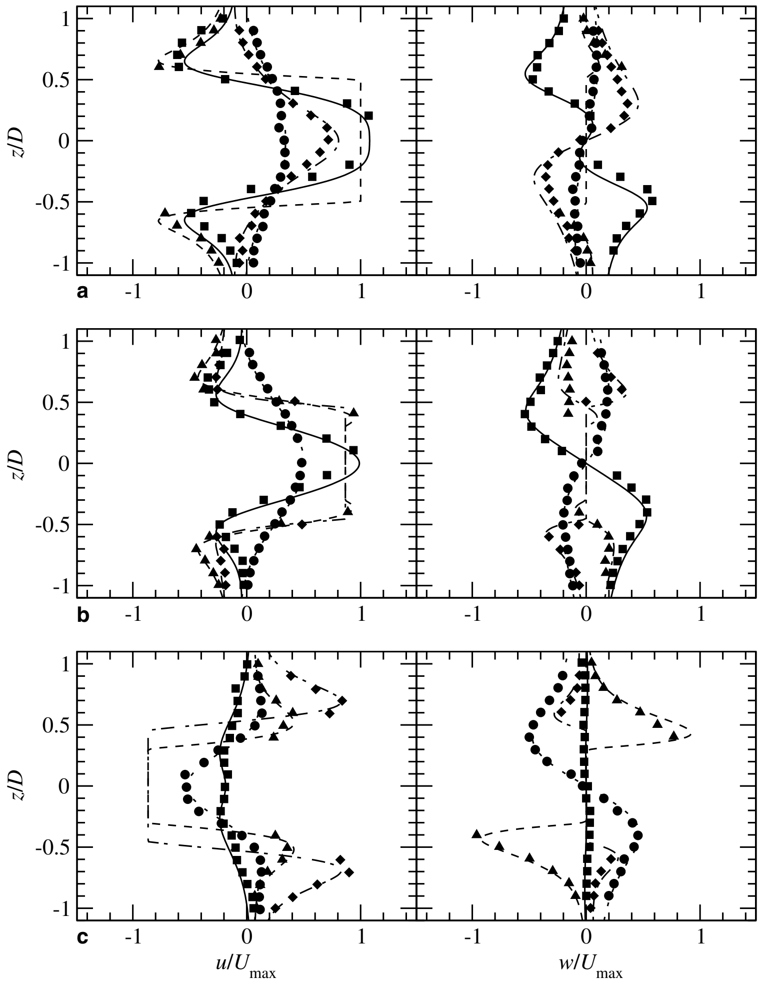


Fig. 9. In-line oscillating cylinder in a fluid at rest ($Re = 100$ and KC = 5). Velocity profiles at: (a) 180°; (b) 210°; (c) 330°. Lines are the present computation on the 480×240 mesh; symbols are the experimental data in [5]: (—- and ■) $x = -0.6D$; (--- and ▲) $x = 0.0D$; (◆ and –·–) $x = 0.6D$; (··· and ●) $x = 1.2D$.

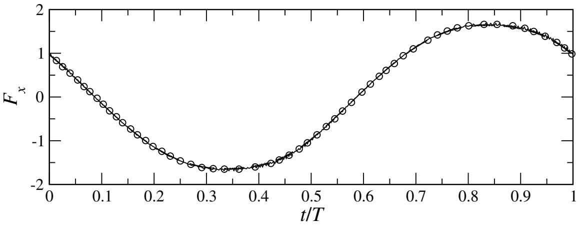


Fig. 10. Evolution in time of the in-line force acting on a cylinder oscillating in a fluid at rest at $Re = 100$ and KC = 5: (○) boundary-conforming simulation [5]; (—) present computation 480×240; (---) present computation 960×480.

3.3. Transversely oscillating cylinder in a free-stream

The fluid–structure interaction problem becomes more complicated when the oscillating cylinder is in a free-stream. In the present study we only consider cases of transverse oscillation. The relevant parameters for such a problem are the Reynolds number $Re = U_\infty D/\nu$ (U_∞ is the free-stream velocity), the excitation frequency, f_e, and the oscillation amplitude, A. When A is larger than a threshold value the synchronization condition can appear – manifested by a jump in the phase-angle, ϕ, between the lift force and the body motion – when the frequency of the oscillation is varied around the natural shedding frequency, f_0, of the stationary cylinder. The variations in this phase angle are associated with the change in sign of the energy transfer between the fluid and the cylinder. Capturing the physics of these flows, even at low Reynolds numbers, requires a very accurate prediction of the vorticity production and dynamics on the surface of the body, which makes their simulation especially using non-boundary-conforming formulations a very challenging task.

The choice of parameters in this flow reproduces the conditions in the experiments by Gu et al. [10] and the subsequent simulations by Guilmineau and Queutey [11], where an accurate boundary-conforming numerical method is used. To conduct quantitative comparisons, however, we will be mainly using the numerical results since the amount of quantitative information reported in the experiments is limited. To match the above conditions the prescribed motion cylinder is chosen to be $z(t) = A\cos(2\pi f_e t)$, where $z(t)$ is the cross-stream location of the cylinder as a function of time. Six different cases were considered at $Re = 185$, $A = 0.2D$ and $0.8 \leqslant f_e/f_0 \leqslant 1.2$. This way we cover a wide spectrum of the conditions in [10,11].

The computational domain is $50D \times 30D$, in the streamwise and cross-stream directions respectively. The cylinder is located at a distance of $10D$ from the inlet boundary, and $15D$ from the upper and lower boundaries. At the outflow boundary a convective boundary condition is used, and radiative conditions are imposed at the free-stream boundaries. The grid is stretched in both directions to cluster points in the vicinity of the body. To investigate the dependency of our results to grid resolution three different meshes with an increasing number of nodes are considered: 200×160, 400×320, and 800×640 points in the streamwise and cross-stream directions respectively. For all three grids the flow past a fixed cylinder at $Re = 185$ is initially computed, to obtain the natural shedding frequency, f_0. As for the previous test problem the results converge with grid refinement. For example, $\overline{C}_D = 1.366$, $C_D^{\text{rms}} = 0.029$, and $C_L^{\text{rms}} = 0.461$, on the finest grid, which are within 1% of the values on the previous grid level. The Strouhal number is 0.197 for all the three grids.

Fig. 11 shows the instantaneous streamline patterns and vorticity contours on the finest grid when the cylinder is at the extreme upper position. Four cases are shown, where f_e/f_0 is gradually increased from 0.8 to 1.2. A sudden change in the topology of the near wake can be observed at $f_e/f_0 = 1.10$ (see Fig. 11c), as manifested by the appearance of concentric, closed streamlines, and remains the same up to the highest value of excitation frequency. This topology suggests the existence of concentrated vorticity in the near wake, which can be verified from the vorticity isolines shown in the same figure. The different wake structure is also reflected in the lift and drag coefficients shown in Fig. 12. For values of f_e/f_0 greater than 1.0, both the drag and lift exhibit signs of the influence of a higher harmonic. This behavior can be attributed to the strengthening of the opposite-sign vorticity formed at the base of the cylinder as f_e/f_0 increases, which alters the vorticity in the top and bottom shear layers. The above results are in excellent agreement with reference data in the literature [10,11].

Quantitative comparisons of the predicted drag and lift coefficients for all frequencies with the corresponding values from the simulations by Guilmineau and Queutey [11] are shown in Fig. 13a. The variation of the average drag coefficient, $\overline{C}_D$, with f_e/f_0 is in agreement with the reference data, although it is a little higher in our simulations (approximately 5%) for all frequency ratios. In the reference simulations, however, $\overline{C}_D$ was found to be very sensitive to grid resolution and they reported higher values when the grid was refined for the case of $f_e/f_0 = 1.10$. The *r.m.s* values of the drag and lift coefficients are less sensitive to grid resolution and are in excellent agreement with the reference data. The same applies to the phase angle, ϕ, between the lift coefficient and the vertical displacement of the cylinder that is shown in Fig. 13b. Both ϕ and C_L^{rms} exhibit a sudden change at $f_e/f_0 = 1.10$ when the vortex switching occurs as shown in [10,11].

Next, we will examine how accurately the local forces on the cylinder's surface are predicted. The fact that $\overline{C}_D, C_D^{\text{rms}}$, and C_L^{rms} above are in good agreement with the reference simulations, does not guarantee the same agreement on the local skin friction distribution, for example, which contributes a small percentage to the above quantities. Guilmineau and Queutey [11] reported the local distribution of surface pressure and vorticity, when the cylinder is located at the extreme upper position. A comparison with our results on all grids for a range of frequencies is shown in Fig. 14. The angle θ on the x axis is measured clockwise from the stagnation point. It can be seen that our results follow the reference data very well especially on the finest grid. As expected the skin friction is more sensitive to grid resolution and is under-predicted near the peaks on the coarser grids. The pressure coefficient on the other hand is fairly insensitive and almost identical on all grids. We should also note that the body-fitted grid that is used in [11] is much finer near the cylinder's surface compared

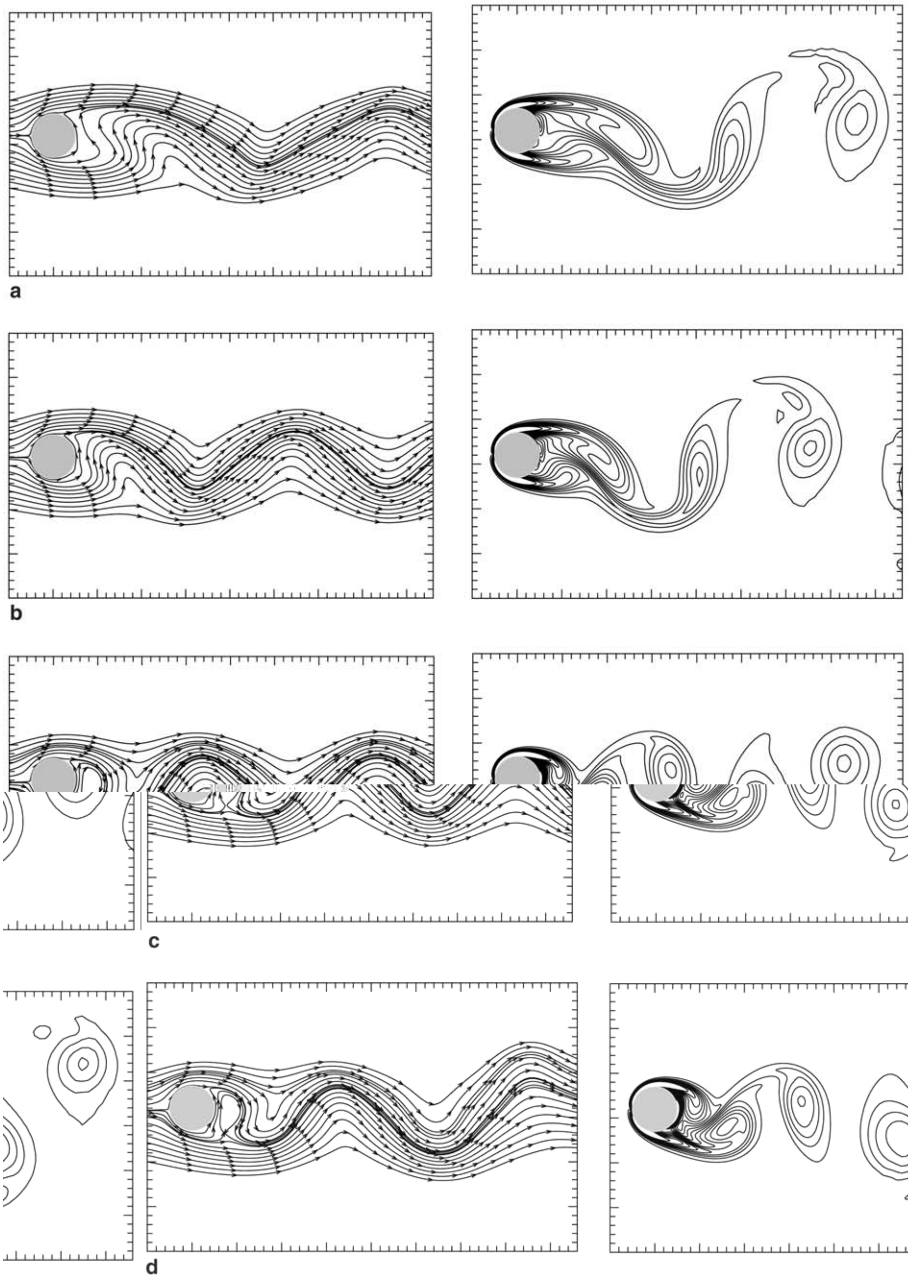


Fig. 11. Cylinder oscillating in a free-stream: instantaneous streamline patterns and vorticity contours when cylinder is located at its extreme upper position; $-6 < \omega_y < 6$ with intervals of 0.68. (a) $f_e/f_0 = 0.8$; (b) $f_e/f_0 = 1.0$; (c) $f_e/f_0 = 1.1$; (d) $f_e/f_0 = 1.2$.

to our finest grid. The local grid spacing normal to the boundary is $0.001D$ in [11] and $0.005D$ for our 800×640 grid, which further demonstrates the accuracy of the overall approach.

3.4. LES of complex turbulent and transitional flows

3.4.1. Turbulent flow over a traveling wavy wall

In this section we will investigate the accuracy and efficiency of the proposed approach in the framework of LES of turbulent flows. In particular, we will perform LES for the case of turbulent flow over a flexible wall

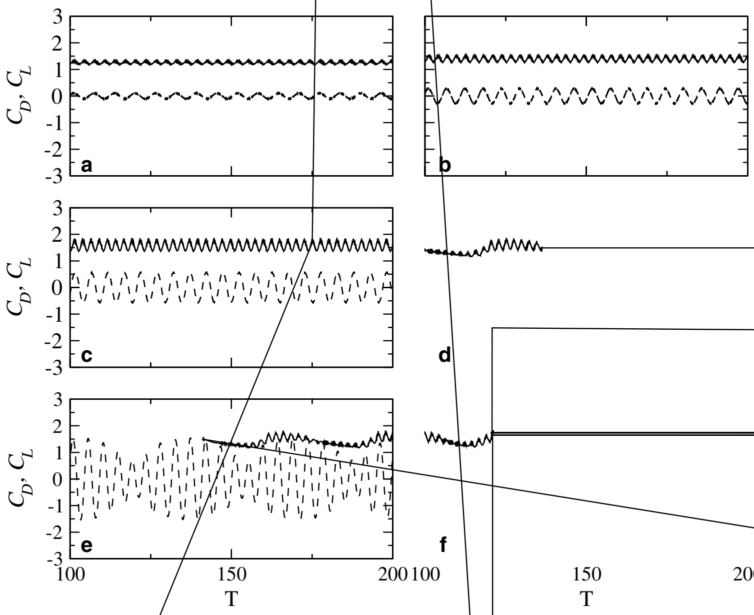


Fig. 12. Evolution of the drag and lift coefficients in time for the case of a cylinder oscillating in a free-stream: (—) C_D; (---) C_L. (a) $f_e/f_0 = 0.8$; (b) $f_e/f_0 = 0.9$; (c) $f_e/f_0 = 1.0$; (d) $f_e/f_0 = 1.1$; (e) $f_e/f_0 = 1.12$; (f) $f_e/f_0 = 1.2$.

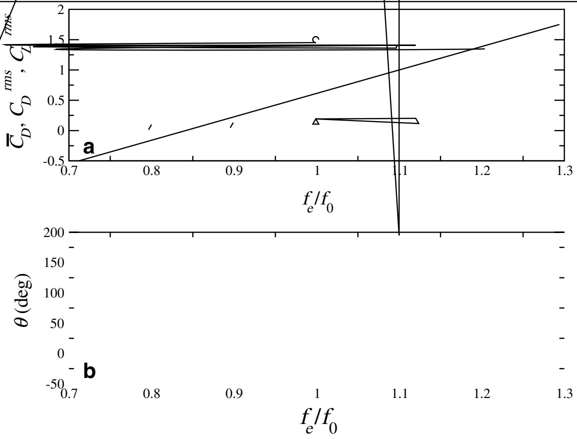


Fig. 13. (a) Comparison of force coefficients for the case of the cylinder oscillating in a free-stream: (— and ○) $\overline{C}_D$; (--- and △) C_D^{rms}; (–·– and □) C_L^{rms}; lines are present results on the 800 × 640 grid and symbols are the corresponding data from [11]. (b) Comparison of the phase angle between the lift coefficient and the vertical displacement: (—) present results; (□) [11].

undergoing streamwise traveling wave motion. A key parameter in this flow is the ratio of the wave speed, c, to the external mean flow, U, which has a substantial effect on turbulence production near the surface. Depending the value of c/U turbulence can be enhanced or fully suppressed leading to relaminarization. The parametric space in our computations is selected to match the conditions in the DNS by Shen et al. [24]. These simulations were performed using a pseudospectral method in the horizontal directions, and second-order finite-differences in the wall-normal direction. They also used a coordinate transformation to map the physical space into a computational space that eliminates the need to deform the grid with the wave motion. A sketch of the computational domain is shown in Fig. 15. The wavy wall is undergoing a vertical oscillation in the form of a streamwise-traveling wave. The location of the wall boundary as a function of time is $y_w(t) = a \sin k(x - ct)$, where a is the magnitude of the oscillation, $k = 2\pi/\lambda$ is the wavenumber with λ the wavelength,

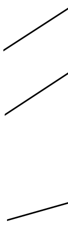
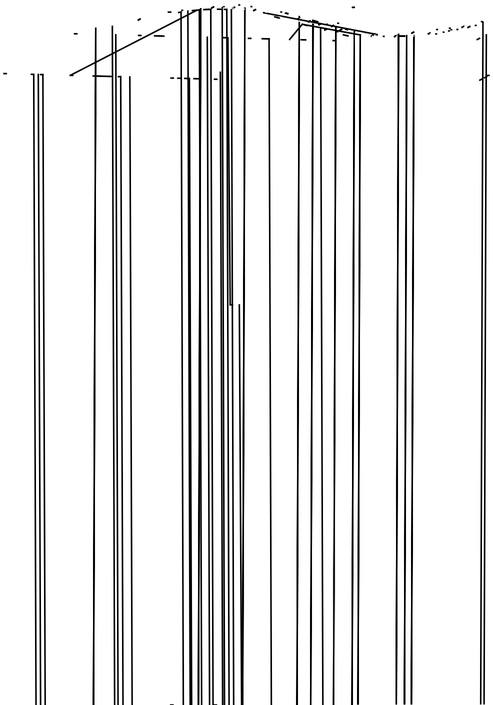


Fig. 14. D
present co
computatio



and *c*
velocit
quenc
(wall-

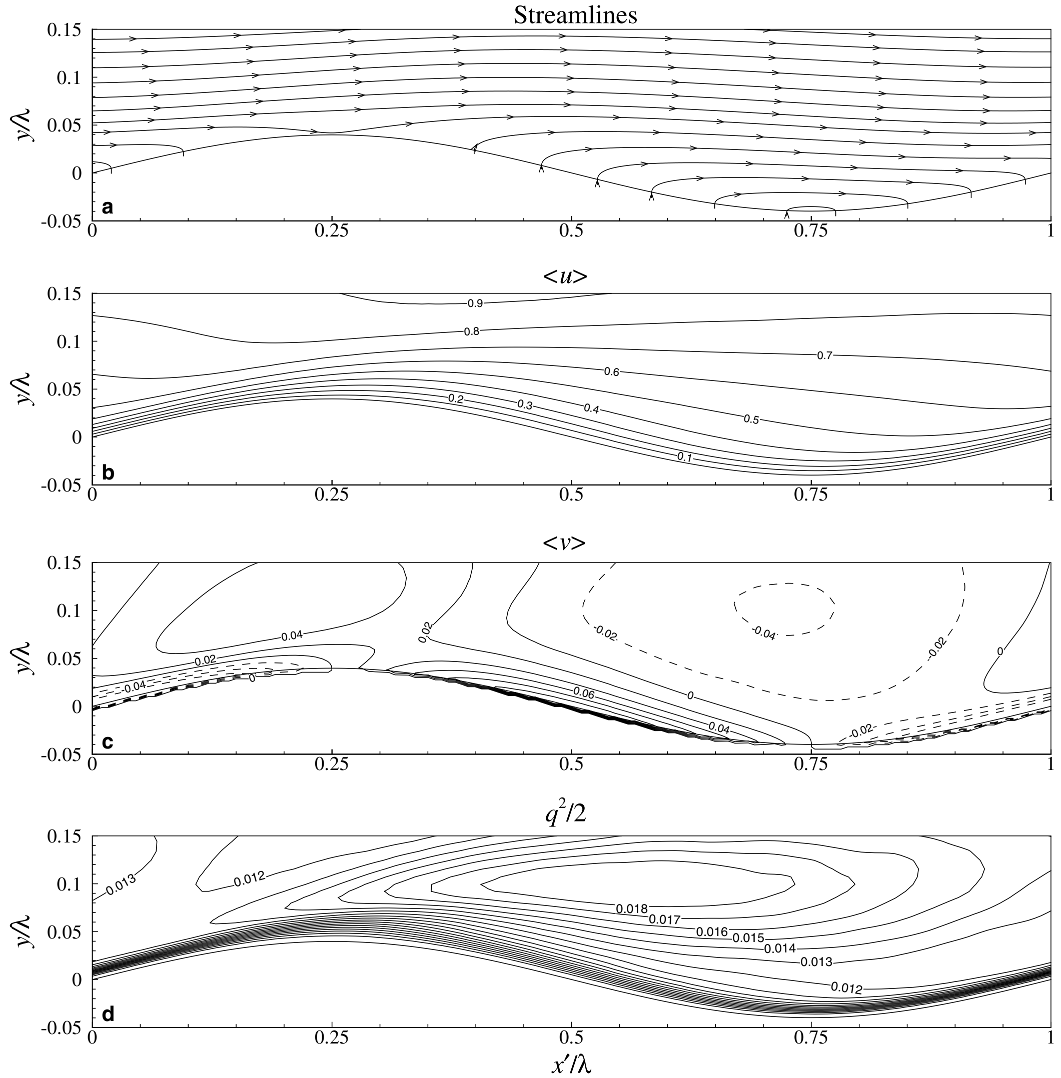


Fig. 17. Phase-averaged statistics for $c/U = 0.4$.

and 18c). This has an effect on the shape of the streamlines which is concave for $c/U = 0$ (see Fig. 16a), and becomes flat for $c/U = 0.4$ and convex for $c/U = 1.2$. As expected, c/U, also affects the turbulent fluctuations. The turbulent kinetic energy is dramatically decreased and the maximum is relocated further downstream when c/U increases from 0.4 to 1.2, as shown in part (d) of both figures. A comparison of Figs. 17 and 18 to the corresponding ones in [24] – the scale and contour intervals are selected appropriately to facilitate direct comparisons – shows very good agreement with the reference results.

To further investigate the accuracy of the proposed approach the forces on the wavy boundary are computed, and then compared to the corresponding reference data. These quantities are very sensitive to the adopted numerical methodology and are a stringent test for the accuracy of the proposed formulation. Following Shen et al. [24], the local friction force, f_x^f, and local pressure force, f_x^p, on an element, ds, on the solid boundary can be written as

$$\begin{cases} f_x^f = \mu\left[-2\frac{\partial u}{\partial x}\frac{\mathrm{d}y_w}{\mathrm{d}x} + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right]\frac{1}{\mathrm{d}s}, \\ f_x^p = p\frac{\mathrm{d}y_w}{\mathrm{d}x}\frac{1}{\mathrm{d}s}. \end{cases} \tag{17}$$

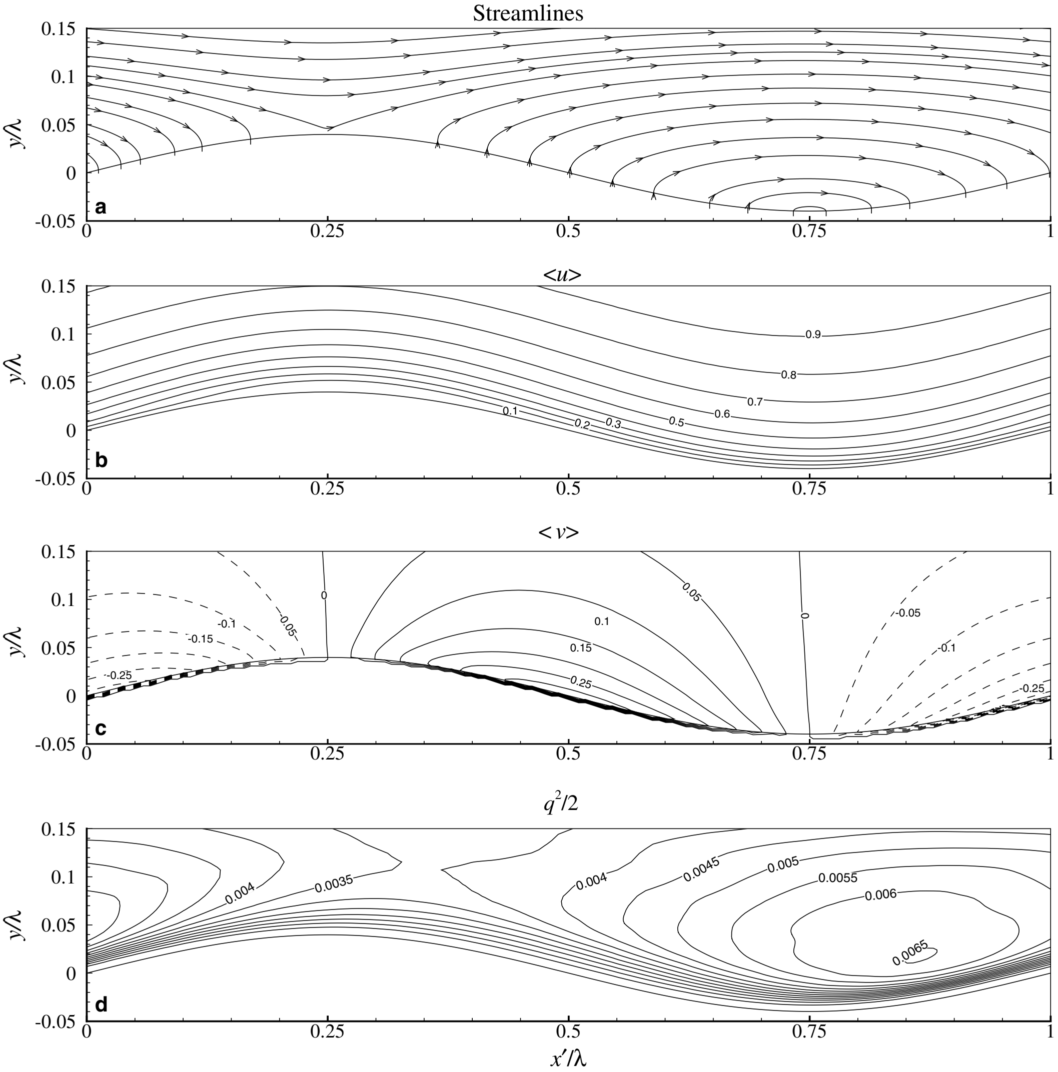


Fig. 18. Phase-averaged statistics for $c/U = 1.2$.

The velocity derivatives and surface pressure in Eq. (17) are computed using the method discussed in Section 2.5. Then, one can integrate f_x^f and f_x^p over the wavy surface to obtain the total friction force, F_f, and total pressure force, F_p, respectively. A comparison of these quantities with the reference data is shown in Fig. 19. Note that we also include a simulation at $c/U = 0.8$, where all other parameters are the same as in the cases discussed above. Our results are in excellent agreement with DNS results. As in the reference simulations the variation of the friction force is relatively small and the value is always above zero, while the pressure force decreases as c/U increases and becomes negative around $c/U = 1.0$.

Finally, a series of instantaneous realizations from all cases were carefully examined to verify the smoothness of the velocity, vorticity and pressure fields near the moving boundary, and visualize the characteristic coherent vortical structures present in this flow. An example is shown in Fig. 20 where isosurfaces of the second invariant of the velocity gradient tensor, Q, are used to identify the near wall vortices (for details on the identification of coherent structures using the "Q-criterion" the reader is referred to [12]). Three cases are included: $c/U = 0.0$, $c/U = 0.4$, and $c/U = 1.2$. For all cases the isosurfaces are colored with the streamwise vorticity values. For the stationary wavy boundary ($c/U = 0.0$) the presence of strong quasi-streamwise vortices that begin in the upslope area of the wave and extend over the trough is evident. As c/U increases, there

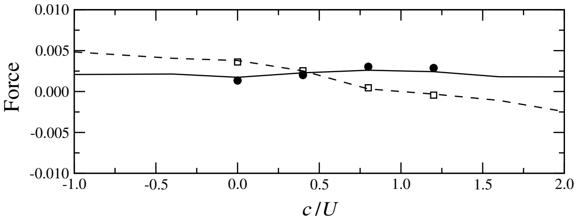


Fig. 19. Mean force acting on the traveling wavy boundary as a function of c/U. —, ● is total friction force, F_f; - - -, □ is the total pressure force, F_p. Lines are the DNS results in [24], and symbols are the present results.

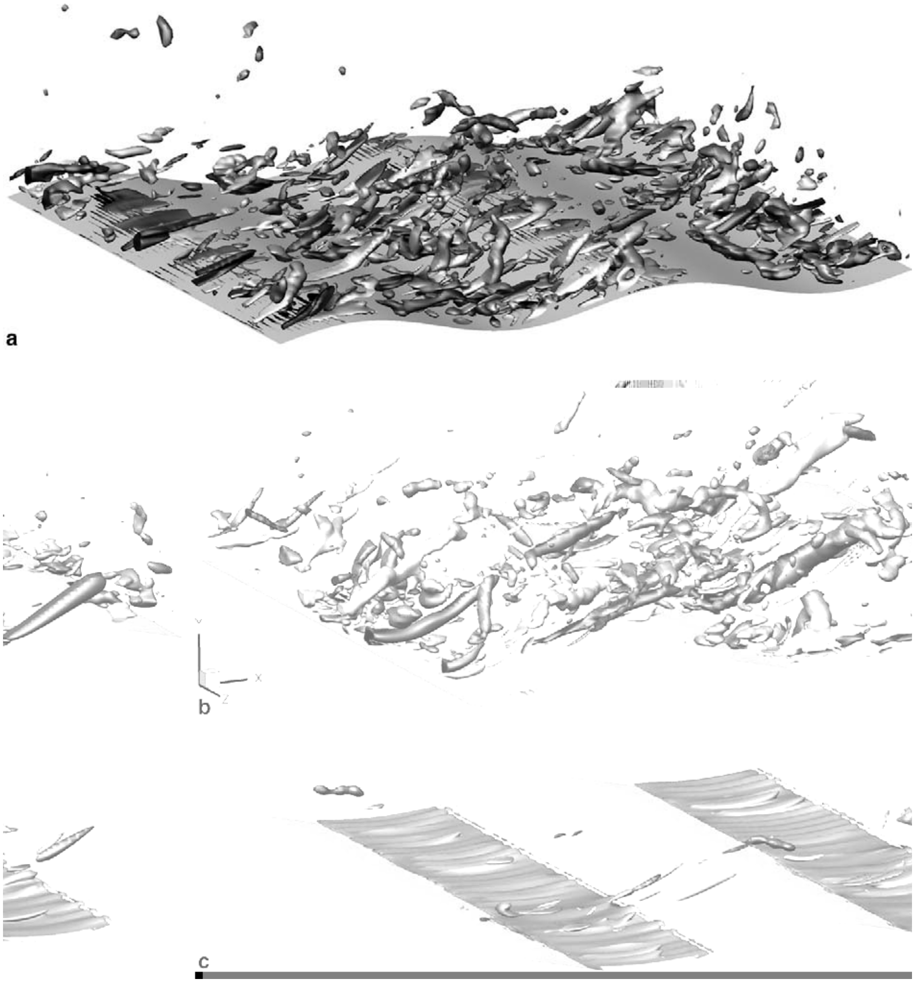


Fig. 20. Isosurfaces of $Q = 8.0$ colored with the streamwise vorticity at an instant in time. (a) $c/U = 0.0$; (b) $c/U = 0.4$; (c) $c/U = 1.2$.

are fewer and more elongated vortices. For the maximum ratio considered in the present study ($c/U = 1.2$) these structures are fully suppressed. This trend is in agreement with the phase-averaged turbulent statistics shown in Figs. 17 and 18, and the results reported in [24].

3.4.2. Pulsatile flow past a prosthetic bileaflet heart valve

To further demonstrate the capabilities of the present method in handling realistic turbulent and transitional flow problems that involve complex three-dimensional boundaries consisting of multiple moving parts, we have computed the flow past a mechanical bileaflet heart valve in the aortic position. The geometry of the valve is shown in Fig. 21. The shape and size of the leaflets roughly mimics the St. Jude Medical (SJM) Standard bileaflet, which is commonly used in clinical practice. In our case, however, the hinge region has been simplified to only allow one degree of freedom for each leaflet (rotation around a fixed axis in the *y* direction). Also, the leaflets contact the housing wall tightly and there is no gap between the hinge and the housing wall. The housing is also shown in Fig. 21, where a straight pipe with rigid walls expands and then contracts to mimic the geometry of the aortic root. The overall setup resembles the ones commonly used in in-vitro

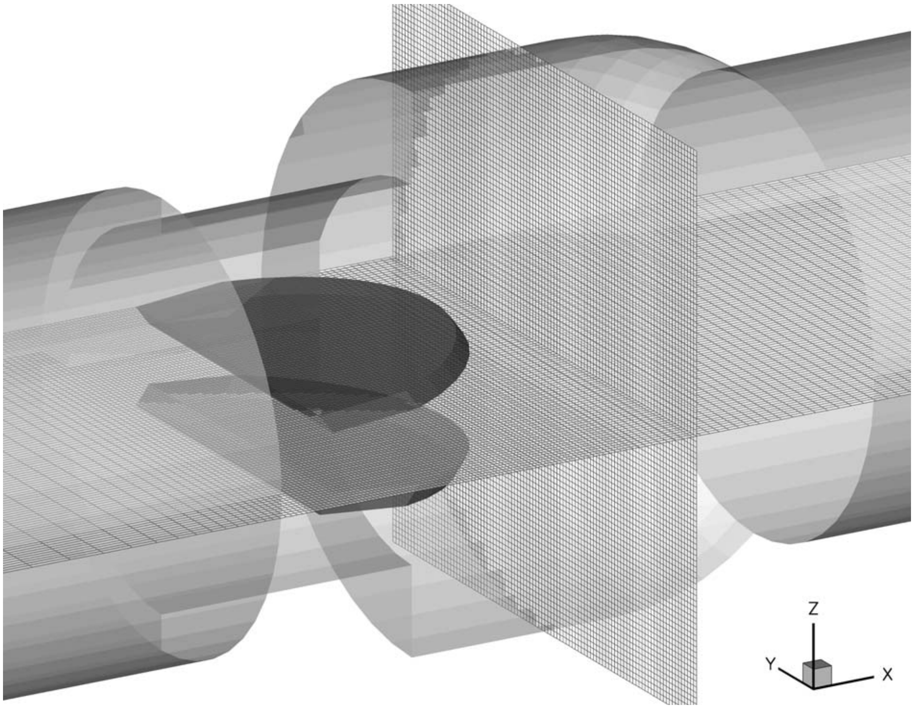


Fig. 21. Geometry of the simplified bileaflet prosthetic heart valve placed in a rigid wall model of the aortic root. The leaflets are shown in their fully open position, and part of the housing wall has been removed for clarity. Two planes of the Cartesian grid are also shown.

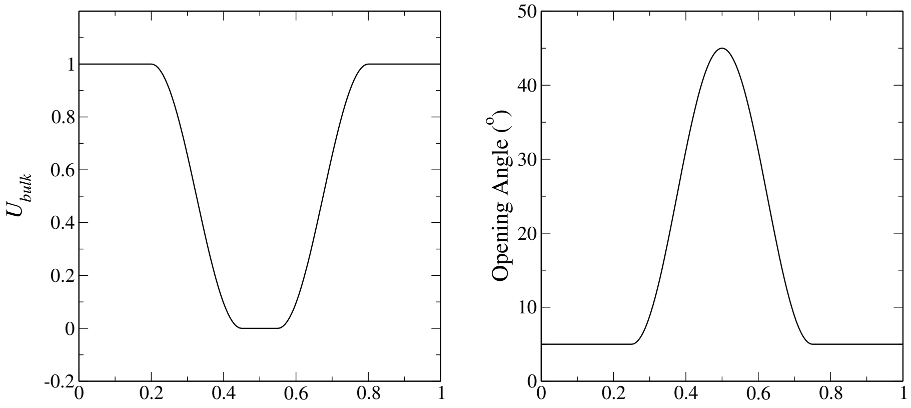


Fig. 22. Imposed flow rate and leaflet kinematics for the heart valve problem: (a) bulk velocity at the inlet; (b) opening angle of the leaflets, as functions of time.

experiments to test the hemodynamic performance of prosthetic valves in the aortic position (see for example [32] for a recent review).

In the actual case the motion of the leaflets is a product of their interaction with the pulsatile blood flow: the valve opens at the beginning of the systole and closes before the start of the diastole. When the leaflets are fully open the blood flow rapidly accelerates through the valve and reaches its peak velocity. Since the objective of the present simulations, however, is to demonstrate the applicability of the method to complex moving boundaries, the movement of the leaflets is prescribed according to a simplified law that roughly resembles the movement of the leaflets as determined by their interaction with the blood flow. At the fully open position of each

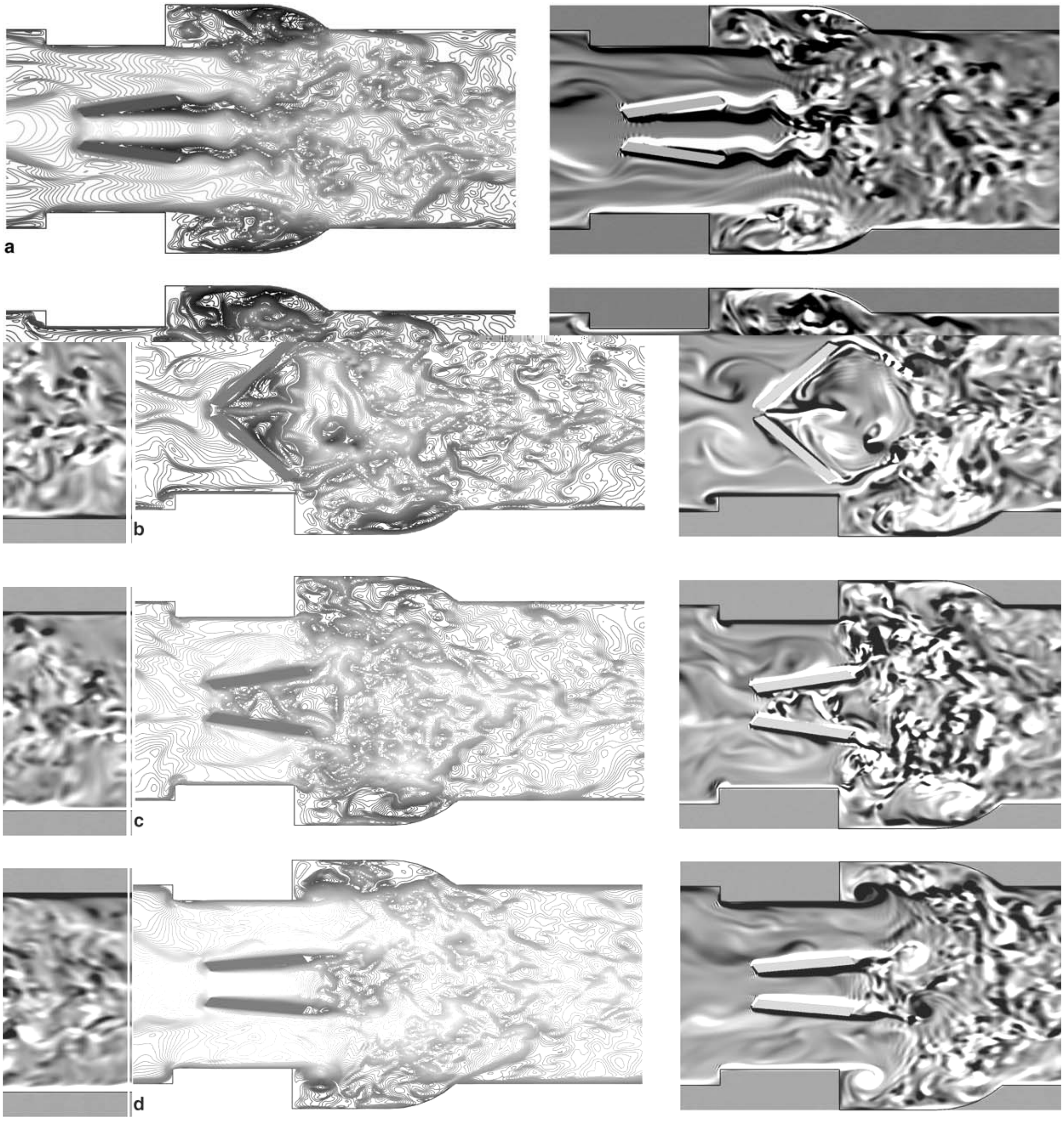


Fig. 23. Instantaneous flow fields at an x–z plane cutting through the middle of the leaflets. Left: streamwise velocity ($-1 < u/U_b^{\max} < 2$, intervals are 0.035); Right: spanwise vorticity ($-10 < \omega_y D/U_b^{\max} < 10$, intervals are 0.2). Phase reference, t/T, from Fig. 22. (a) $t/T = 0.3$; (b) $t/T = 0.5$; (c) $t/T = 0.7$; (d) $t/T = 0.9$.

leaflet forms an 85° angle with the y–z plane, while a rotation of 53° towards the housing wall closes the valve completely. In Fig. 22 the variation of the bulk velocity and the corresponding angle of the leaflets are shown for a full pulsatile cycle. The Reynolds number based on the pipe diameter and the maximum bulk velocity was set to 4000, which is a realistic value for this application.

The above configuration which involves the interaction of the flow with a stationary (the aortic chamber) and two moving (the leaflets) boundaries is a great challenge for any structured or unstructured boundary-conforming method. In the present computation the overall geometry is immersed into a Cartesian grid and the boundary conditions on the stationary and moving boundaries are imposed using the method described in the previous sections. The computational grid involves 420 × 200 × 200 nodes in the streamwise, spanwise (direction parallel to the rotation axis of the leaflets), and transverse directions, respectively. The mesh is stretched in the x–y plane and is kept uniform in the z direction (see Fig. 21), in a way that the leaflets

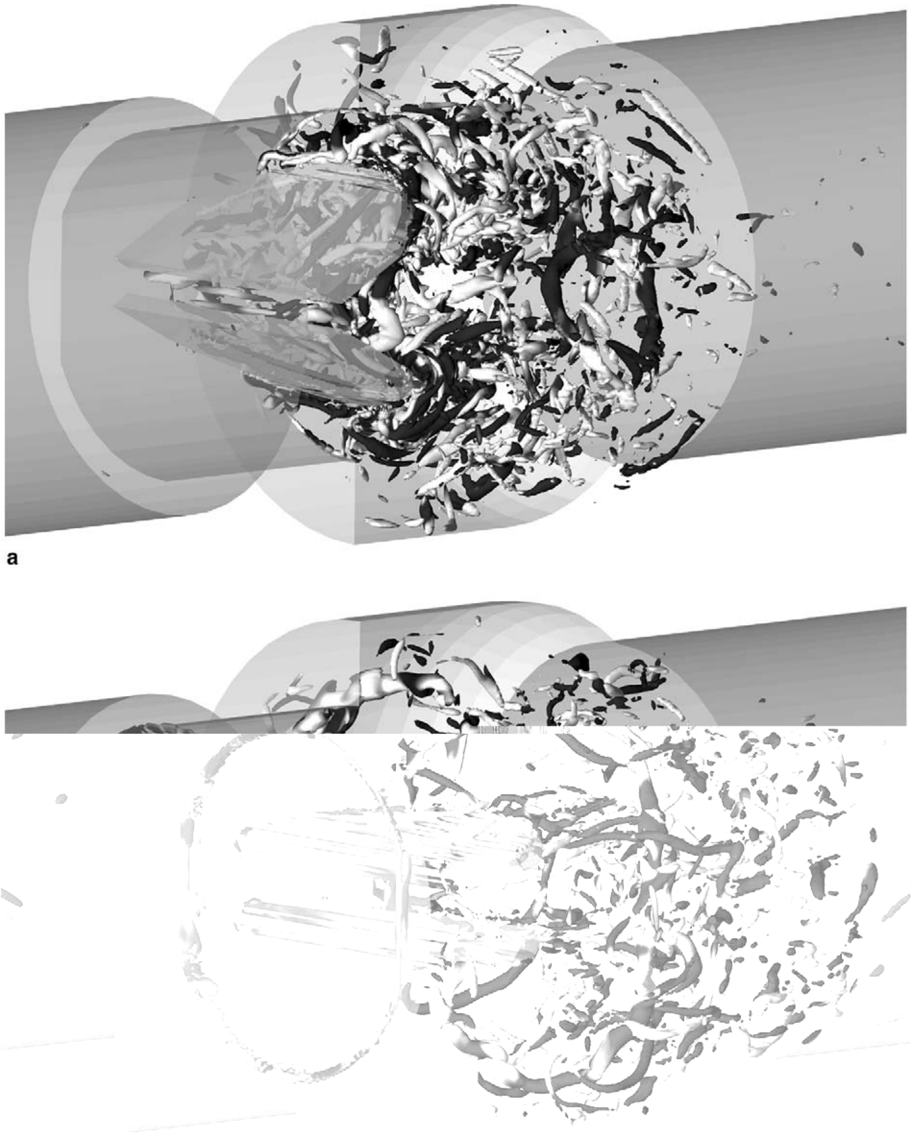


Fig. 24. Instantaneous vortical structures visualized using isosurfaces of $Q = 600$ colored by the streamwise vorticity. Phase reference, t/T, is from Fig. 22. (a) $t/T = 0.6$; (b) $t/T = 0.8$.

are moving in an approximately isotropic grid with average dimension of $0.005D$. At the inflow boundary, located $4D$ upstream of the valve, fully developed flow corresponding to the flowrate variation shown in Fig. 22 is imposed. At the outflow boundary ($8D$ downstream of the valve) a convective boundary condition is used.

The flow near the leaflets is very complicated, and it is dominated by intricate vortex–leaflet and vortex–vortex interactions. To illuminate basic flow patterns and demonstrate that the present method can accurately capture the thin shear layers that form on both moving and stationary immersed boundaries, instantaneous contours of the streamwise velocity and spanwise vorticity are shown in Fig. 23 for a few characteristic phase angles during the pulsatile cycle. When the leaflets are fully open, a strong jet is formed from the central orifice which interacts with the wake of the leaflets as it can be seen from the velocity and vorticity contours in Fig. 23a. Two shear layers are generated on each leaflet and roll-up into large vortices which are interacting with other vortices inside the chamber. A similar flow pattern has been observed downstream of a bileaflet valve in a recent highly resolved PIV experiment where a much more detailed of in a model of the left ventricle was used [22]. There are several other locales where fairly strong shear layers are present, like the edge of the leaflet housing for example. In general, at this stage in the pulsatile cycle the flow has similar characteristics with what has been observed in steady experiments with the leaflets at the fully open position [4].

As the flow rate is decreasing and the leaflets begin to move towards the fully closed position (see Fig. 23b), the central orifice shrinks and the jet becomes thinner. The interaction of this jet and the ambient decelerating flow generates pockets of reverse flow on the downstream side of the leaflets. The flow rate keeps decreasing until it reaches zero, and the leaflets reach their maximum closing angle. The central jet during this stage becomes weaker and is finally dissipated as there is no flow through the central orifice. At the same time the large coherent vortices in the aortic chamber break down generating smaller structures. When the leaflets begin to open again and the flow rate starts to increase, all these structures are convected downstream (see Fig. 23c). The flow accelerates until it reaches the maximum flow rate, and the leaflets reach their fully open position again as shown in Fig. 23d.

To better illuminated the highly complex and three-dimensional nature of the flow just downstream of the leaflets, isosurfaces of Q are shown in Fig. 24. Two instantaneous realizations during the opening phase of the valve have been selected. In Fig. 24a the complex interactions of vortices originating from the leaflets, the hinge area, and the housing can be observed. At a later time these structures are washed out of the aortic root and the formation of large ring vortex at the expansion plane is evident (see Fig. 24b).

4. Summary and conclusions

The aim of the present work is the extension of embedded-boundary formulations to laminar and turbulent flows interacting with moving boundaries. Despite the increased attention that these methods have been receiving in recent years, systematic studies examining their accuracy and applicability in cases of moving boundaries have not been reported. In the present paper we initially identified problems that are unique to these formulations, and are a consequence of the boundary motion on a fixed grid. In particular, we have shown that in a typical fractional-step scheme, when a point near the interface (forcing point) moves into the flow (fluid point), carries with it unphysical information related to the pressure and velocity derivatives. As a result, the evaluation of the RHS of the momentum equations at this splitting cycle is problematic. To address this issue we proposed a robust field-extension procedure, where the velocity and pressure fields are "extended" in the solid phase at the end of each substep. This way the pressure and velocity derivatives at the problematic points have values that reflect the proper boundary conditions. The overall extension procedure, although it is tailored to our reconstruction scheme that utilizes the well-defined normal to the interface [1], is applicable in a straightforward manner to any embedded-boundary formulation that reconstructs the solution around points in the fluid phase [6]. In cases where some type of a generalized ghost-cell approach is used [14,26], then the grid points emerging from the solid need to be treated in a manner similar to that suggested, for example, in [29].

In terms of validating the proposed methodologies, a variety of problems with increasing complexity were considered. First, the formal accuracy of the method (second order) was established for the problem of an array of cylinders moving in a plane channel. Then, computations for two cases of laminar flow interacting with moving boundaries were conducted. The flow induced by the harmonic in-line oscillation of a circular cylinder in a quiescent fluid, and the flow from a transversely oscillating cylinder in a free-stream. Both cases are well documented in the literature, and experimental and numerical results are available for comparison. The proposed method was found to reproduce all features of the flow, including the detailed force distribution on the body with the same degree of accuracy as the reference boundary-conforming computations. Computations of the flow over a traveling wavy wall served a validation of the method in LES of turbulent flows with moving boundaries. Finally LES of transitional flow around a prosthetic heart valve with moving leaflets demonstrated the robustness and applicability of the method in turbulent/transitional flows with multiple moving boundaries. Despite the simplification on the geometry the obtained results revealed flow patterns that have been reported in the literature.

In general the results in this study show that embedded-boundary formulations can be cost–efficient strategies especially for moving boundary problems without sacrificing accuracy. A limiting factor in all these methods, however, is the inflexibility in clustering grid points near a complex body. As the Reynolds number increases and more points need to be clustered near the body to resolve the thinner boundary layers, this can result in very large grids. To investigate if they are still cost/efficient compared to boundary conforming formulations, one needs to look at the rate of increase of the total number of grid points for both categories of methods and the cost per node for each method as a function of the Reynolds number. This is a fairly difficult task since it depends on a variety of problem specific parameters. In general for the case of a single body, when using boundary-fitted grids, an increase in the Reynolds number requires refinement of the mesh in the wall-normal direction only; in non-boundary-conforming formulations like the present one, however, to achieve refinement normal to the body the grid usually needs to be refined in two or three directions. As a result the required number of grid points as a function of the Reynolds number increases faster for non-boundary-conforming formulations compared to boundary conforming ones. Actually, for the flow over a cylinder be shown that the ratio of the boundary-conforming to the non-boundary-conforming grid size increases approximately as $Re^{1.5}$ [18]. However, given that methods like the present one are substantially less expensive per node than boundary-conforming ones (especially in case of boundaries undergoing large motions and/or deformations where grid re-generation is a major obstacle that compromises the accuracy and efficiency of boundary fitted approaches) they can be cost/efficient for a fairly wide range of Reynolds numbers. The computation of the flow around the heart valve for example, has been performed on a Linux desktop workstation equipped with four AMD Opteron 846 2 GHz processors (each with 1M cache, and 8GB memory in total) takes about 6 CPU hours for one pulsatile cycle. A 10-cycle simulation which with 17 million points can be completed in less than 3 days. For a variety of problems, especially from biology where only low/moderate Reynolds numbers are encountered methods such the present one have a lot of advantages. For high Reynolds numbers, however, some type of local grid refinement would be necessary.

Acknowledgments

The authors are grateful to Mr. A. Cristallo for generating the geometry for the heart valve computation. Financial support from the National Institutes of Health (Grant R01-HL-07262) and the National Science Foundation (Grant CTS-0347011) is gratefully acknowledged.

References

[1] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, Comput. Fluids 33 (2004) 375–404.

[2] E. Balaras, J. Yang, Non-boundary conforming methods for large-eddy simulations of biological flows, ASME J. Fluids Eng. 127 (2005) 851–857.

[3] S.A. Bayyuk, K.G. Powell, B. van Leer, A simulation technique for 2-D unsteady inviscid flows around arbitrary moving and deforming bodies of arbitrary geometry, AIAA Paper 93-3391-CP, 1993.

[4] P. Browne, A. Ramuzat, R. Saxena, A.P. Yganathan, Experimental investigation of the steady flow downstream of the St. Jude bileaflet heart valve: a comparison between LDV and PIV techniques, Ann. Biomed. Eng. 28 (2000) 39–47.

[5] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers, J. Fluid Mech. 360 (1998) 249–271.

[6] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.

[7] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (1993) 354–366.

[8] D. Goldstein, R. Handler, L. Sirovich, Direct numerical simulation of turbulent flow over a modeled riblet covered surface, J. Fluid Mech. 302 (1995) 333.

[9] D. Goldstein, T.C. Tuan, Secondary flow induced by riblets, J. Fluid Mech. 363 (1998) 115.

[10] W. Gu, C. Chyu, D. Rockwell, Timing of vortex formation from an oscillating cylinder, Phys. Fluids 6 (1994) 3677–3682.

[11] E. Guilmineau, P. Queutey, A numerical simulation of vortex shedding from an oscillating circular cylinder, J. Fluids Struct. 16 (2002) 773–794.

[12] J.C.R. Hunt, A.A. Wray, P. Moin, Eddies, streams and convergence zones in turbulent flows, Center for Turbulence Research, Proceedings of the Summer Program, 1988, p. 193.

[13] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (2003) 331–347.

[14] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (2001) 132–150.

[15] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.

[16] S. Majumdar, G. Iaccarino, P. Durbin, RANS solvers with adaptive structured boundary non-conforming grids, Center for Turbulence Research, Annual Research Briefs, 2001, pp. 353–366.

[17] C. Meneveau, C.S. Lund, W.H. Cabot, A Lagrangian dynamic subgrid-scale model of turbulence, J. Fluid Mech. 319 (1996) 353–385.

[18] R. Mittal, J. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[19] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.

[20] C.S. Peskin, D.M. McQueen, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1980) 220–252.

[21] C.S. Peskin, D.M. McQueen, Cardiac fluid dynamics, Crit. Rev. Biomed. Eng. 20 (1992) 451–459.

[22] O. Pierrakos, P.P. Vlachos, D.P. Telionis, Time-resolved DPIV analysis of vortex dynamics in a left ventricular model through bileaflet mechanical and porcine heart valve prostheses, J. Biomech. Eng. 126 (2004) 714–726.

[23] J.J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrary two-dimensional bodies, Comput. Fluids 23 (1994) 125.

[24] L. Shen, X. Zhang, D.K.P. Yue, M.S. Triantafyllou, Turbulent flow over a flexible wall undergoing a streamwise traveling wave motion, J. Fluid Mech. 484 (2003) 197–221.

[25] P.N. Swartzrauber, Direct method for the discrete solution of separable elliptic equations, SIAM J. Numer. Anal. 11 (1974) 1136.

[26] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.

[27] H.S. Udaykumar, W. Shyy, M.M. Rao, Elafint: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, Int. J. Numer. Methods Fluids 22 (1996) 691.

[28] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, J. Comput. Phys. 153 (1999) 535–574.

[29] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (2001) 345–380.

[30] R. Verzicco, J. Mohd-Yusof, P. Orlandi, D. Haworth, Large-eddy simulation in complex geometric configurations using body forces, AIAA J. 38 (2000) 427.

[31] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.

[32] A.P. Yoganathan, Z. He, C.S. Jones, Fluid mechanics of heart valves, Annu. Rev. Biomed. Eng. 6 (2004) 331–362.

[33] D. De Zeeuw, K.G. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, AIAA Paper 91-1542-CP, 1991.