Journal of Computational Physics 228 (2009) 6617–6628

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Short Note

A moving-least-squares reconstruction for embedded-boundary formulations

Marcos Vanella [a], Elias Balaras [a,b,*]

[a] Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA
[b] Fischell Department of Bioengineering, University of Maryland, College Park, MD 20742, USA

ARTICLE INFO

Article history:
Received 16 January 2009
Received in revised form 25 April 2009
Accepted 7 June 2009
Available online 12 June 2009

Keywords:
Cartesian grids
Finite-difference method
Immersed boundary method

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Recently there has been a renewed interest on non-boundary conforming formulations, which have been emerging as a viable alternative to unstructured methods especially for complex geometries and fluid–structure interaction problems. To compute the flow around a complex body the equations of motion are usually solved on a fixed structured grid, which is almost never aligned with the body. Depending on the specifics of the formulation, boundary conditions are imposed by appropriately modifying the stencil in the neighborhood of the body [17], or by using a forcing function which can be derived either using physical arguments [15], or directly from the discrete problem [6]. The latter category of methods -also called direct-forcing methods- are particularly attractive for computing the flow around complex rigid bodies, since they can be implemented into existing finite-difference or finite-volume structured solvers in a straightforward manner. Boundary motion, however, introduces additional complications, and a trivial extension of the direct-forcing formulations designed for stationary boundaries (see for example [6,10,2]) to fluid–structure interaction problems, leads to hydrodynamic forces that lack smoothness and are a potential source of instabilities [18,19].

Yang and Balaras [19] suggested that the large fluctuations of the hydrodynamic forces on moving immersed bodies are due to the fact that, at any given timestep, some of the Eulerian grid points in the vicinity of the body will not have the correct velocity, pressure or their derivatives, due to their association with the solid in a previous timestep. The problematic cells as well as the appropriate treatment depends on the details of the implementation. Yang and Balaras [19], for example, proposed a field-extension procedure, where the solution is 'extended' into the body in a way that the cells that emerge into the fluid have the proper velocity and pressure at later timesteps. Mittal et al. [13] in their generalized ghost-cell formulation, assigned the proper values at the problematic cells by interpolating from their surroundings. Uhlmann [18] suggested an alternative direct-forcing scheme, where the force is computed on the Lagrangian markers rather than Eulerian points as

* Corresponding author. Address: Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA.
E-mail addresses: mvanella@umd.edu (M. Vanella), balaras@umd.edu (E. Balaras).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.06.003

it was done in all previous implementations, which resulted in much smoother hydrodynamic forces. For the purpose of illustrating the differences between the two strategies let us assume that u_i $(i = 1, 2, 3)$ is a discrete approximation of the velocity field and write the time-discretized form of momentum equation as:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = rhs^{n+1/2} + f_i^{n+1/2}, \tag{1}$$

where *rhs* contains all advective and diffusive terms, f_i is the direct-forcing function which is different from zero only at the grid points in the vicinity of the immersed body, and $n, n+1$ refer to the current and next timestep respectively. In the direct-forcing scheme proposed in [6] or [2], for example, for every point where $f_i \neq 0$, one can replace u_i^{n+1} in Eq. (1) with the desired velocity u_i^b (usually determined by means of interpolation from the surrounding nodes), and find:

$$f_i^{n+1/2} = \frac{u_i^b - u_i^n}{\Delta t} - rhs^{n+1/2}. \tag{2}$$

Substituting f_i back into Eq. (1) the proper boundary condition, $u_i^{n+1} = u_i^b$, is recovered. In the formulation proposed by Uhlmann [18] on the other hand, the direct-forcing function is computed on each Lagrangian marker, rather than on the Eulerian grid nodes, as follows:

$$F_i^{n+1/2} = \frac{U_i^b - U_i^n}{\Delta t} - RHS^{n+1/2}. \tag{3}$$

The upper case symbols in Eq. (3) denote the same variables as in Eq. (2), but at the Lagrangian points on the immersed body. Setting $\widetilde{U} = U_i^n + \Delta t\, RHS^{n+1/2}$, which is practically the Lagrangian counterpart of the predicted velocity, $\tilde{u}_i$, (see Eq. 5), we can rewrite Eq. (3) as:

$$F_i^{n+1/2} = \frac{U_i^b - \widetilde{U}_i}{\Delta t} \tag{4}$$

Uhlmann [18] computed the volume force from Eq. (4) and used the regularized delta functions introduced in [15] as kernels in the transfer of variables between the Eulerian and Lagrangian grids. The overall implementation was tailored to model suspended rigid spherical particles in laminar and turbulent flows, where it was demonstrated to be very efficient and robust. Direct extension to more complex fluid–structure interaction problems however, hinges upon the requirement to have uniform elements on the surface of the body, as well as on the fact that only integral hydrodynamic forces could be computed. In the present study, based on the ideas presented in [18], we propose a direct-forcing scheme that utilizes a versatile moving-least-square (MLS) approximation to build the transfer functions between the Eulerian and Lagrangian grids, and can be applied to arbitrary moving/deforming bodies. We will also propose a method to compute the local traction forces. The overall formulation utilizes very compact stencils and, without compromising accuracy and robustness, gives results that are identical to 'sharp' direct-forcing methods.

2. Methodologies

The proposed formulation will be discussed in the framework of a finite-difference, fractional-step, Navier–Stokes solver for incompressible flow. Both advective and diffusive terms are advanced explicitly using an Adams–Bashforth scheme, and all spatial derivatives are discretized using central, second-order, finite-differences on a staggered grid. Details on the basic solver together with applications in a variety of wall-bounded and free-shear flows can be found in [16,3,2]. In the following sections we will focus on the proposed direct-forcing scheme and computation of the hydrodynamic forces.

2.1. MLS reconstruction

In the framework of the above mentioned splitting scheme we first take a provisional step to compute the intermediate velocities, $\tilde{u}_i$, which do not satisfy the incompressibility constraint and the boundary conditions on the immersed body:

$$\tilde{u}_i = u_i^n + \frac{\Delta t}{2}\left(3H(u_i^n) - H(u_i^{n-1})\right) - \Delta t \frac{\partial p^n}{\partial x_i}, \tag{5}$$

where H is a discrete operator representing the spatially discretized convective and viscous terms. Next, we will build a direct-forcing function that will enforce the proper boundary conditions on all the Eulerian grid nodes influenced by the immersed body. As in [18] we will compute the forcing function on the Lagrangian markers and then transfer it to the Eulerian grid nodes. Our transfer operators, however, will be constructed using MLS shape functions with compact support [11,12]. To facilitate this process, for each Lagrangian marker we: (i) Identify the closest Eulerian grid node. Referring to Fig. 1(a), for example, the marker l_a is associated to the grid node x_a, which is in the center of a cell with dimensions h_x and h_y in the x and y directions respectively. Marker l_b is associated to the grid node x_b and so on. Note that more than one Lagrangian markers from the same, or different immersed bodies, can be associated with the same Eulerian grid node. (ii) Define a support-domain around each Lagrangian marker, in which the shape functions will be constructed. In our case the support

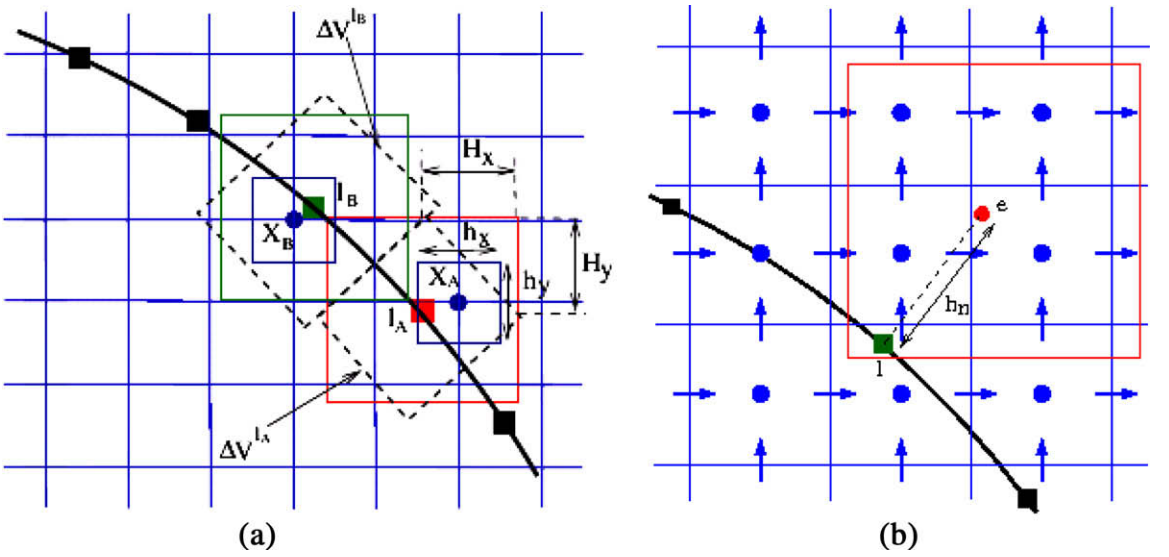

Fig. 1. (a) Definition of the support-domain for two neighboring Lagrangian markers, l_A and l_B, which are color coded for clarity. X_A and X_B denote the closest Eulerian nodes to l_A and l_B respectively. The corresponding volumes ΔV are also shown (dashed line). (b) The normal probe defined by the Lagrangian marker l and point, e is shown together with the support domain used in the MLS approximation.

domain is a rectangular box of size $2H_x \times 2H_y \times 2H_z$ centered at the location of the marker. H_x, H_y and H_z are different for each marker and are proportional to the local Eulerian grid. We found $H_x = 1.2h_x$, $H_y = 1.2h_y$ and $H_z = 1.2h_z$ (see Fig. 1(a)) to be sufficient for all cases considered in this study; (iii) Associate a volume, $\Delta V^l = A^l h^l$ (A^l is the area of the body surface associated to marker l, and h^l is a local thickness that depends on local grid size and will be defined in the following paragraphs) to each marker point. In Fig. 1(a) the volumes ΔV^{l_a} and ΔV^{l_b} for markers l_a and l_b respectively are shown. There is no overlapping between successive volumes, ΔV^l, and the sum of all local A^l is equal to the total area of the immersed object surface.

We can now define the transfer operator that will enable the computation of $\widetilde{U}_i$ from the corresponding velocities, $\tilde{u}_i$, given by Eq. (5). Using the MLS method, $\widetilde{U}_i$ for each Lagrangian marker, l, can be approximated in its support domain as follows:

$$\widetilde{U}_i(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \tag{6}$$

where $\mathbf{p}^T(\mathbf{x})$ is the basis functions vector of length m, $\mathbf{a}(\mathbf{x})$ is a vector of coefficients, and $\mathbf{x}$ is the position of the Lagrangian marker. We found that a linear basis, $\mathbf{p}^T(\mathbf{x}) = [1\ x\ y\ z]$, is a cost-efficient choice and would represent the field variation for all variables up to the accuracy of our spatial discretization scheme. To obtain the coefficient vector, $\mathbf{a}(\mathbf{x})$, the following weighted $L2$-norm is defined:

$$J = \sum_{k=1}^{ne} W(\mathbf{x} - \mathbf{x}^k)[\mathbf{p}^T(\mathbf{x}^k)\mathbf{a}(\mathbf{x}) - \tilde{u}_i^k]^2, \tag{7}$$

where $\mathbf{x}^k$ is the position vector of the Eulerian point k in the interpolation stencil, $\tilde{u}_i^k$ is the variable defined in Eq. (5) for grid point k, and $W(\mathbf{x} - \mathbf{x}^k)$ is a given weight function that will be defined below. ne is the total number of grid points in the interpolation stencil, which for the linear basis function above, involves five and seven points in two- and three-dimensions respectively. For simplicity we set the closest point to the Lagrangian marker to be the center point in the stencil. Minimizing J with respect to $\mathbf{a}(\mathbf{x})$ leads to the following set of equations:

$$\begin{aligned}
&\mathbf{A}(\mathbf{x})\ \mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\ \tilde{\mathbf{u}}_i^k \quad \text{with,}\\
&\mathbf{A}(\mathbf{x}) = \sum_{k=1}^{ne} W(\mathbf{x} - \mathbf{x}^k)\mathbf{p}(\mathbf{x}^k)\mathbf{p}^T(\mathbf{x}^k),\\
&\mathbf{B}(\mathbf{x}) = [W(\mathbf{x} - \mathbf{x}^1)\mathbf{p}(\mathbf{x}^1) \quad \ldots \quad W(\mathbf{x} - \mathbf{x}^{ne})\mathbf{p}(\mathbf{x}^{ne})], \quad \text{and}\\
&\tilde{\mathbf{u}}_i^k = [\tilde{u}_i^1 \quad \ldots \quad \tilde{u}_i^{ne}]^T.
\end{aligned} \tag{8}$$

The size of matrix $\mathbf{A}(\mathbf{x})$ depends on the size of the basis vector, $\mathbf{p}(\mathbf{x})$, and it is 3×3 in two-dimensions and 4×4 in three-dimensions, while $\mathbf{B}(\mathbf{x})$ is of size $3 \times ne$ in two-dimensions or $4 \times ne$ in three-dimensions. Combining Eqs. (6) and (8) one can write $\widetilde{U}_i$ as follows:

$$\widetilde{U}_i(\mathbf{x}) = \sum_{k=1}^{ne} \phi_k^l(\mathbf{x})\tilde{u}_i^k = \mathbf{\Phi}^T(\mathbf{x})\tilde{\mathbf{u}}_i^k \tag{9}$$

where $\mathbf{\Phi}(\mathbf{x}) = \mathbf{p}(\mathbf{x})\,\mathbf{A}(\mathbf{x})^{-1}\,\mathbf{B}(\mathbf{x})$ is a column vector with length *ne*, containing the shape function values for marker point *l*. Cubic splines are used for the weight functions, $W(\mathbf{x}-\mathbf{x}^k)$, above, which can be written as:

$$W(\mathbf{x}-\mathbf{x}^k) = \begin{cases} 2/3 - 4\bar{r}_k^2 + 4\bar{r}_k^3 & \text{for} \quad \bar{r}_k \leqslant 0.5 \\ 4/3 - 4\bar{r}_k + 4\bar{r}_k^2 - 4/3\bar{r}_k^3 & \text{for} \quad 0.5 \leqslant \bar{r}_k \leqslant 1.0 \\ 0 & \text{for} \quad \bar{r}_k > 1.0 \end{cases} \tag{10}$$

where $\bar{r}_k = |\mathbf{x}-\mathbf{x}^k|/H_i$. These functions are monotonically decreasing and are sufficiently smooth in the support domain. The resulting shape functions reproduce exactly the linear polynomial contained in their basis and possess the partition of unity property $\sum_{i=1}^{ne}\phi_i(\mathbf{x}) = 1$ [12]. Also, the field approximation is continuous on the global domain as the *MLS* shape functions are compatible.

Eq. (9) will give $\widetilde{U}_i$, which can then be substituted into Eq. (4) to obtain the volume force F_i on all Lagrangian markers. To transfer F_i to the Eulerian points associated with each marker, *l*, the same shape functions used in interpolation procedure can be used if properly scaled by a factor c_l, which will be determined later. In such case, the final forces on the Eulerian grid nodes would be:

$$f_i^k = \sum_{l=1}^{nl} c_l \phi_k^l F_i^l, \tag{11}$$

where f_i^k is the volume force in the Eulerian point *k* in the direction *i*, ϕ_k^l is the shape function previously obtained relating variables between grid point *k* and marker *l*, and F_i^l is the force in marker *l*. Also, *nl* is the number of Lagrangian markers which are related to the grid point *k*. To properly rescale the shape functions we require that the total force acting on the fluid is not changed by the transfer:

$$\sum_{k=1}^{nte} f_i^k \Delta V^k = \sum_{l=1}^{ntl} F_i^l \Delta V^l \tag{12}$$

where, $\Delta V^k = (h_x \times h_y \times h_z)$ is the volume associated with the Eulerian grid point *k*, and $\Delta V^l = A^l h^l$ is the volume associated with the marker *l*, with $h^l = 1/3\sum_{k=1}^{ne}\phi_k^l(h_x + h_y + h_z)$. *nte* and *ntl* is the total number of forced grid points, and total number of Lagrangian markers respectively. As our surface is discretized using triangular elements, the area for marker *l* is obtained by a simple angle averaging process. Using (11) in (12) and rearranging the sums in the left hand side in terms of the total number of markers we get

$$\sum_{l=1}^{ntl}\sum_{k=1}^{ne} \phi_k^l \Delta V^k c_l F_i^l = \sum_{l=1}^{ntl} \Delta V^{El} c_l F_i^l = \sum_{l=1}^{ntl} F_i^l \Delta V^l \tag{13}$$

where ΔV^{El} is the averaged Eulerian grid volume associated to the Lagrangian marker *l*. For Eq. (13) to hold the scaling factor c_l needs to be set to:

$$c_l = \frac{\Delta V^l}{\Delta V^{El}} = \frac{A^l h^l}{\Delta V^{El}}, \tag{14}$$

One can also show that the above scheme guarantees the equivalence of total torque between the Eulerian and Lagrangian meshes:

$$\sum_{k=1}^{nte} \mathbf{x}_k \times \mathbf{f}_k \Delta V^k = \sum_{l=1}^{ntl} \mathbf{X}_l \times \mathbf{F}_k \Delta V^l \tag{15}$$

For simplicity we will provide a proof in two-dimensions, but the extension to three-dimensions is straightforward. In particular, the two-dimensional form of Eq. (15) can be written as:

$$\sum_{k=1}^{nte} (x^k f_y^k - y^k f_x^k)\Delta V^k = \sum_{l=1}^{ntl} (X^l F_y^l - Y^l F_x^l)\Delta V^l, \tag{16}$$

or as:

$$\sum_{k=1}^{nte} x^k f_y^k \Delta V^k = \sum_{l=1}^{ntl} X^l F_y^l \Delta V^l, \tag{17}$$

$$\sum_{k=1}^{nte} y^k f_x^k \Delta V^k = \sum_{l=1}^{ntl} Y^l F_x^l \Delta V^l, \tag{18}$$

Eq. (16) will hold if both (17) and (18) hold. In the following we will consider proof of Eq. (18), and similar arguments can be used for Eq. (17). For each Lagrangian marker, Y^l can be expressed in terms of the shape functions as follows:

$$Y^l = \sum_{k=1}^{ne} \phi_k^l y^k. \tag{19}$$

Substituting (19) and (11) into (18) and reordering the sums in the LHS:

$$\sum_{l=1}^{ntl} F_x^l c_l \sum_{k=1}^{ne} \phi_k^l y^k \Delta V^k = \sum_{l=1}^{ntl} F_x^l \sum_{k=1}^{ne} \phi_k^l y^k \Delta V^l. \tag{20}$$

Inspection of Eq. (20) confirms that the equivalence of total torque will be satisfied, if for each Lagrangian marker, l, the following is true:

$$c_l \sum_{k=1}^{ne} \phi_k^l y^k \Delta V^k = \sum_{k=1}^{ne} \phi_k^l y^k \Delta V^l \tag{21}$$

Given that $c_l = \Delta V^l / \sum_{k=1}^{ne} \phi_k^l \Delta v^k$, and assuming ΔV^k is constant on the Eulerian stencil, (21) is trivially satisfied. In summary, the proposed transfer operators, conserve momentum on both uniform and stretched grids. For torque to be conserved the cell volume across the stencil should be kept constant for each marker. This is satisfied in case of uniform grids. In other situations, the departure from equivalence for torque will depend on the amount of stretching of the grid. Numerical experiments on two dimensional meshes showed that this difference is small (less than 0.5%) for 10% grid stretching in each direction.

Using the forcing function from Eq. (11), we can now correct the intermediate velocity $\tilde{u}_i$ to respect the boundary conditions on the immersed body: $u_i^* = \tilde{u}_i + \Delta t f_i$. The resulting approximate velocity field, u_i^*, which is not divergence-free, can be projected into a divergence-free space by applying a correction of the form:

$$u_i^{n+1} = u_i^* - \Delta t \frac{\partial}{\partial x_i}(\delta p), \tag{22}$$

where $\delta p = p^{n+1} - p^n$ is the pressure correction, which satisfies the following Poisson equation:

$$\frac{\partial^2(\delta p)}{\partial x_i \partial x_i} = \frac{1}{\Delta t} \frac{\partial u_i^*}{\partial x_i}. \tag{23}$$

The velocity field, u_i^{n+1}, given by Eq. (22) is divergence-free and satisfies the boundary conditions to the order of $O(\Delta t^2)$ [10].

2.2. Calculus of surface forces

In non-boundary conforming formulations the fact that the computational grid and the surface of the body are almost never aligned, introduces complications to the computation of hydrodynamic forces generated by the surrounding fluid. In the present formulation for the case of rigid bodies the distributed forcing function given by Eq. (11) can be utilized to compute the total hydrodynamic force on a solid object, provided that all interior points are properly treated (see for example [18]). Extension of this approach, however, to the general case of moving/deforming bodies is not trivial. In the present study we have developed a methodology where the local hydrodynamic force per unit area on a surface element,

$$f_i^H = \tau_{ji} n_j = \left[-p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] n_j, \tag{24}$$

is computed directly from the flow field around the body. In Eq. (24), f_i^H is the hydrodynamic surface force in x_i direction, τ_{ji} is stress tensor, and n_j is the direction cosine of the normal unit vector in x_j direction. The use of Eq. (24) requires knowledge of p and $\partial u_i / \partial x_j$ on the body surface. In the formulation outlined above the boundary is defined in a sharp manner, but the pressure and velocity fields are forced to vary smoothly through the surface of the body. Consequently the use of the same transfer functions to estimate p and $\partial u_i / \partial x_j$ at the Lagrangian markers would probably underestimate the actual traction forces. This was also verified by a series of numerical experiments we conducted for the case of the flow around an oscillating cylinder below.

To avoid such problems, for each Lagrangian marker, l, on the body we create a normal probe by locating an external point, e, at a distance, h_n, from the surface (see Fig. 1(b)). The distance h_n is proportional to the local grid spacing and is set to: $h_n = (h_x + h_y + h_z)/3$. To compute the surface pressure at marker, l, we first compute the pressure, p^e, at point e, using the MLS formulation described in the previous section. The support domain in this case is centered around point e as shown in Fig. 1(b). Next, the value of $\partial p / \partial n$ is obtained from the momentum equation normal to the boundary [19]:

$$\frac{\partial p}{\partial n} = -\frac{D\mathbf{u}}{Dt} \cdot \mathbf{n}, \tag{25}$$

where $\mathbf{n}$ is the normal unit vector passing through the marker l, and $\frac{D\mathbf{u}}{Dt}$ is the acceleration of the marker. The value of the pressure at the surface is then obtained from:

$$p^l = p^e - \frac{\partial p}{\partial n} h_n \tag{26}$$

The velocity derivatives, $\partial U_i/\partial x_j$, at the location e for each Lagrangian marker, l, are computed by differentiating Eq. (9):

$$\frac{\partial U_i}{\partial x_j} = \sum_{k=1}^{ne} \frac{\partial \phi_k}{\partial x_j} u_i, \tag{27}$$

where $\partial\phi_k/\partial x_j$ comes from the solution of an additional system of equations similar to (8) [12]. Given the fact that h_n is of the order of the local grid size, and assuming a linear variation of the velocity near the body, the derivatives, $\partial U_i/\partial x_j$, coming from Eq. (27) are good approximation for the derivatives $\partial u_i/\partial x_j$, at the surface. Higher-order reconstruction procedures could also be adopted, albeit at a higher cost. As we will demonstrate in the results section the above procedure reproduces the forces on the surface of an immersed body very accurately when compared to boundary-conforming methods at the same grid resolution.

3. Results

In this section we will present a series of test problems of increasing complexity to demonstrate the accuracy and robustness of the proposed formulation. First, the formal accuracy is examined for the case of the flow around a cylinder submerged in a driven cavity. Then, the flow around a cylinder oscillating in a cross flow is considered. Here we focus on the accuracy of the local force distribution on the surface of the cylinder. Finally, the robustness of the approach in complex fluid–structure interaction problems is demonstrated for the case of sphere-wall collisions.

3.1. Accuracy study

To evaluate the spatial accuracy of proposed algorithm we performed simulations of the flow around a cylinder immersed in a lid-driven cavity. Fig. 2(a) shows the geometry and a typical vorticity distribution. For all cases considered the cylinder diameter was set to $D = 0.4L_R$, and the Reynolds number, $Re = U_{lid}L_R/\nu = 1000$, where L_R is the cavity size and U_{lid} the velocity of the top boundary. The no-slip conditions on the surface of the cylinder were enforced using the proposed MLS reconstruction.

Although an analytical solution for this problem is not available, the overall accuracy of the scheme can be evaluated by comparing the solution among grids at different resolution. To facilitate this comparison on a staggered grid arrangement we considered meshes with $36^2, 60^2, 108^2, 180^2$ and 540^2 nodes. With this choice the finest one (540^2) is the reference solution, and the average and maximum errors on each of the coarser grids is computed without the need to interpolate. In Fig. 2(b)

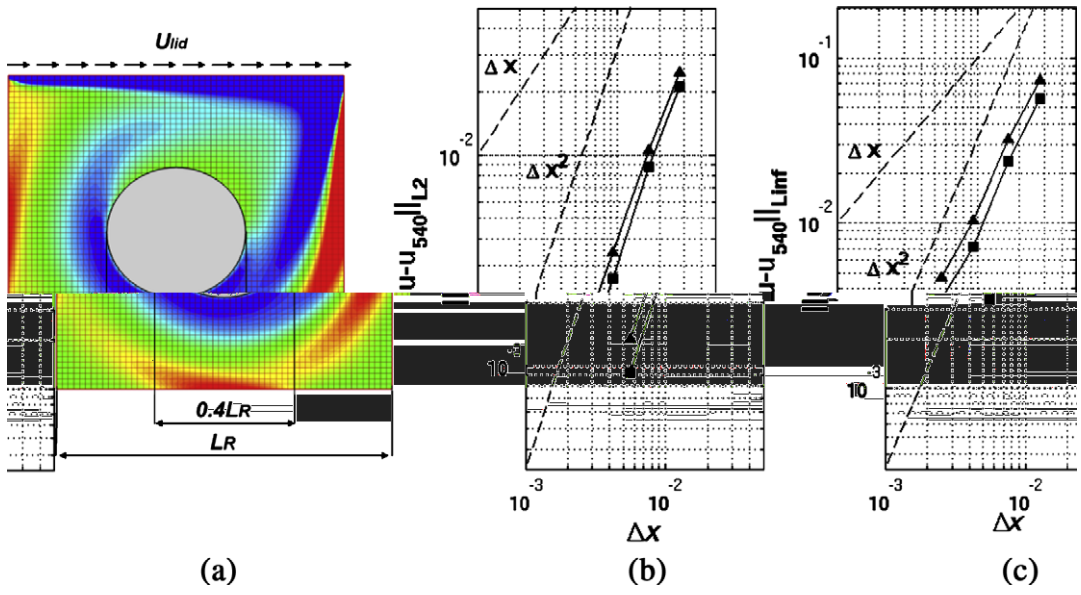

Fig. 2. The flow around a cylinder immersed in lid-driven cavity. (a) Computational set-up; (b) L_2 norm of the error and (c) L_{inf} norm of the error as a function of the cell size Δx. (▲) u velocity, (■) v velocity.

and (c), the L_2 and L_{inf} norms of the error are shown as a function the spatial resolution. Both errors decrease with a second-order slope, indicating that the second-order spatial accuracy of the Cartesian solver is maintained.

3.2. Oscillating cylinder in a cross-flow

The ability of non-boundary conforming methods to properly capture the surface pressure and viscous stress distribution is of paramount importance, especially in fluid–structure interaction problems. To assess the performance of the proposed formulation we considered the case of a transversely oscillating cylinder in a cross-flow. The dominant parameters are the Reynolds number $Re = U_\infty D/\nu$(U_∞ is the inflow velocity), the forcing frequency, f_e, and amplitude, a_0, of the oscillation. When f_e varies around the natural shedding frequency, f_0, interesting phenomena occur due to the complex energy transfer between the fluid and the body [7,8].

Capturing the detailed flow physics for this problem requires an accurate reproduction of the vorticity dynamics on the surface of the body and is a stringent test for non-boundary-conforming schemes. The parametric space we considered is the one used in the experiments by Gu et al. [7], the boundary-conforming simulations of Guilmineau and Queutey [8], and computations by Yang and Balaras [19], where an embedded-boundary method with a direct-forcing scheme is used. The motion of the cylinder is given by $y(t) = a_0 sin(2\pi f_e t)$. We considered three cases with $Re = 185, a_0 = 0.2D$ and $f_e/f_0 = 1.0, 1.1, 1.2$ respectively. For all cases the computational domain was set to $50D \times 30D$ in the streamwise and cross-stream directions respectively, with the cylinder located at $10D$ from the inflow boundary. Free-slip conditions are used at the freestream boundaries and a convective condition at outflow boundary [14]. We considered two grids with 500×450 and 850×750 nodes, where the resulting cell size around the cylinder was $\Delta x = \Delta y = 0.008D$ and $\Delta x = \Delta y = 0.004D$ respectively. A series of tests for flow over a stationary cylinder was first conducted to examine the sensitivity of the results to the grid resolution. The predicted mean and root-mean-square (rms) values of the drag and lift coefficients on the fine grid were $\overline{C}_D = 1.377, C_D^{rms} = 0.296$ and $C_L^{rms} = 0.461$, and the corresponding values on the coarser grid are within 1.5% of the above, demonstrating the grid independency of the results.

The temporal evolution of the lift and drag coefficients for the case of the oscillating cylinder are shown in Fig. 3(a) and (b). It is evident that the proposed scheme results in a smooth variation of the force coefficients without special treatments. In Fig. 3(c) a comparison of $\overline{C}_D, C_D^{rms}$ and C_L^{rms} for the different excitation frequencies is shown with the corresponding results in the boundary conforming computations in [8] and the computations using a direct-forcing scheme in [19]. A similar comparison for the phase angle between the lift coefficient and transverse displacement of the cylinder is shown in Fig. 3(d). In general the agreement is excellent. The largest discrepancy appears in $\overline{C}_D$ and is of the order of 3.5%. We should also note that

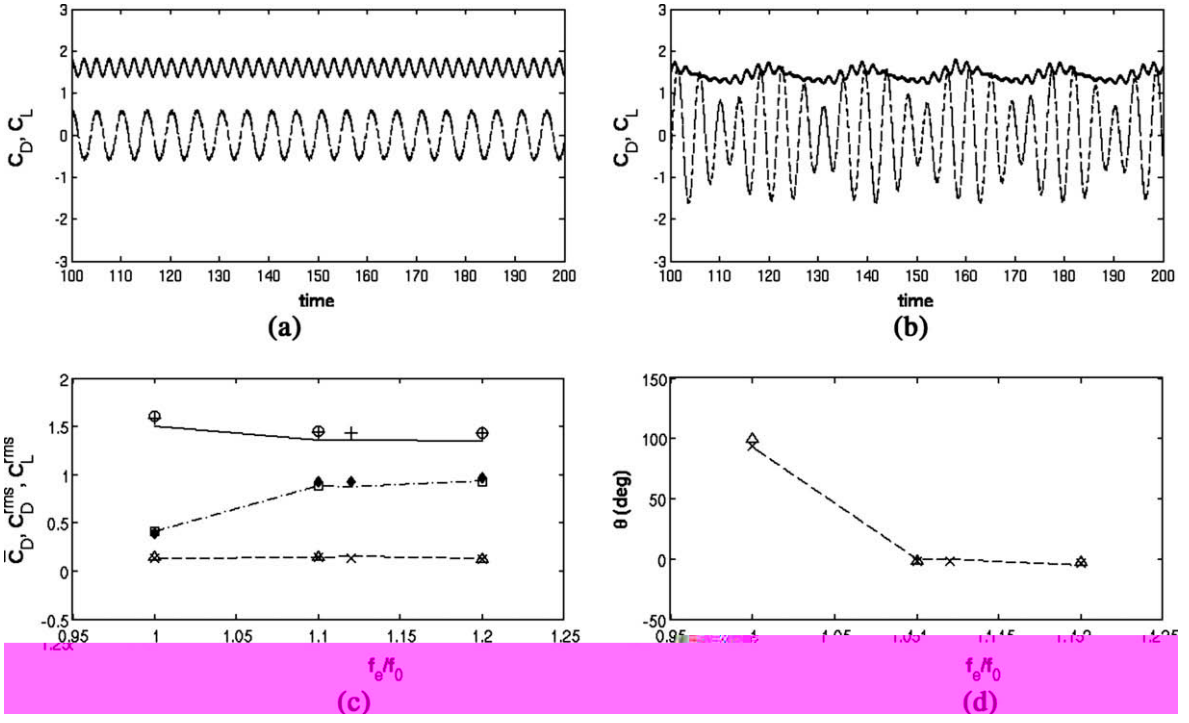

Fig. 3. Drag and lift coefficients as a function of time for the case of cylinder oscillating in a cross-flow for (a) $f_e/f_0 = 1.0$, and (b) $f_e/f_0 = 1.2$($-C_D$ and $-C_L$). (c) Comparison of force coefficients. $\circ\overline{C}_D, \triangle C_D^{rms}, \square C_L^{rms}$ are the present results for the fine grid; — $\overline{C}_D$, – – – – C_D^{rms}, –·– C_L^{rms} from reference [8], and + $\overline{C}_D, \times C_D^{rms}, \blacklozenge C_L^{rms}$ from [19]. (d) Phase angle between lift force and vertical displacement. $\triangle$ are the present results on the fine grid; - - [8] and $\times$ [19].

the numerical resolution around the cylinder in our computations is comparable to the one in the reference computations, where $\Delta x \sim 0.005D$.

An important point in the computation of the hydrodynamics forces, especially in fluid–structure interaction problems, is consistency on the total force and moment exchanged between the fluid and solid systems (action–reaction). Uhlmann [18], for example, who utilizes a similar forcing scheme, proposes a force computation approach that results in equivalence of integral forces. For the case of the oscillating cylinder we compared the force coefficients obtained by direct integration of the local stresses resulting from the normal probe approach, to ones obtained using the approach in [18]. The agreement is very good and the maximum difference is 2.3% for the coarse grid ($\Delta x = 0.008$) and 1.4% for the fine one ($\Delta x = 0.004$) indicating the consistency of our approach.

While mean force predictions is a good indicator of the overall performance of the method, they do not necessarily translate into an accurate representation of the local forces. In Fig. 4 the distributions of pressure coefficient, C_p, and the skin friction coefficient, C_f, on the cylinder's surface are shown for the time instance corresponding to the extreme upper position. Results for both grids are included from our computations, and are compared with the corresponding results by Guilmineau and Queutey [8] and Yang and Balaras [19]. The higher sensitivity of C_f to the grid resolution results in slightly lower peak values on our coarse grid computations. The results on finer grid agree very well with the reference data. C_p is less sensitive to the grid resolution the results on the different grids are almost indistinguishable.

3.3. Three dimensional example: sphere-wall collision

To investigate the robustness and accuracy of the proposed method in three dimensional configurations we performed computations of a rigid sphere bouncing off a wall. Problems involving collisions between immersed bodies are particularly challenging for direct-forcing schemes, since the presence of two or more Lagrangian markers from different bodies in the proximity of the same Eulerian grid cell is usually the source of ambiguity. The proposed forcing scheme treats such situations in a robust manner without the need for special treatments.

The particular configuration we selected has applications to particulate flows, and a number of experimental (i.e. [5,9]) and numerical (i.e. [1]) results are available in the literature for comparison. The dominant parameter in the collision process is the Stokes number, $St = 1/9(\rho_b/\rho_f)Re$, where ρ_b and ρ_f are the particle and fluid densities respectively, and Re is the Reynolds number based on the particle diameter, D, and the translational velocity, U_f, an instant before impact. For low values of the Stokes number ($St < 10$) no rebound will occur, even if the dry restitution coefficient, e_{dry}, is different from zero. For $St > 10$ rebound occurs, and the total restitution coefficient, e_T, is lower than e_{dry}. For large values of the Stokes number ($St > 500$) the total restitution coefficient, e_T, approaches e_{dry}. If $e_{dry} = 0$, then the Reynolds number is the only dominant parameter.

Below we will present results from two different configurations: (i) a case where no rebound occurs, which resembles the conditions in the experiments by Eames and Dalziel [5]; (ii) a case where rebound is allowed, which resembles the conditions in the axisymmetric Navier–Stokes computations by Ardekani and Rangel [1]. In Fig. 5 a sketch of the computational domain is shown. Both the sphere and the wall are immersed into a locally refined Cartesian mesh [4], and are represented by an unstructured Lagrangian grid with 2.7×10^5 and 2.9×10^5 markers respectively (see Fig. 5(b)). The Eulerian grid is arranged in a way that the resolution around the sphere is $\Delta x = 0.01D$.

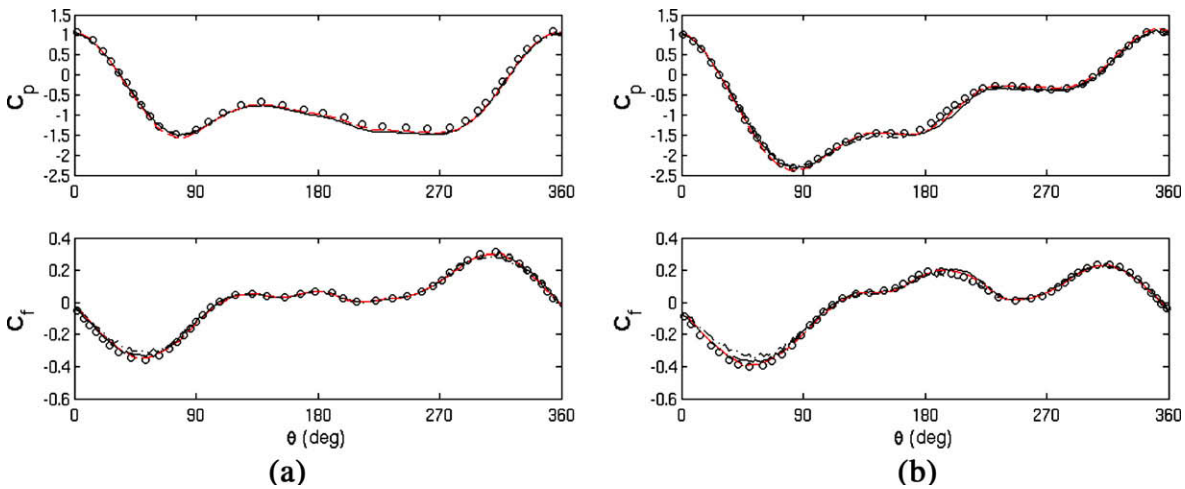

Fig. 4. Distribution of the pressure and skin friction coefficients C_p and C_f for the case of a cylinder oscillating in a cross-flow. The cylinder is located at the extreme upper position. -.- present results for $\Delta x = 0.008D$, — present results for $\Delta x = 0.004D$, ∘ body-fitted computations in [8], - - non boundary-conforming computations in [19]. (a) $f_e/f_0 = 1.0$; (b) $f_e/f_0 = 1.2$.

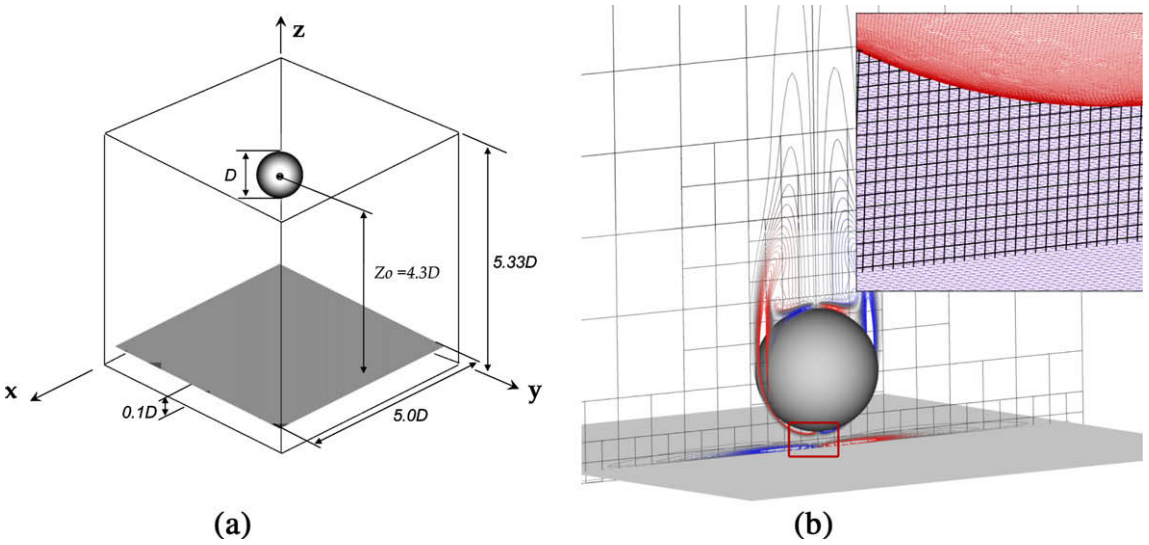

Fig. 5. Sphere-wall interaction problem. (a) Computational setup; (b) Lagrangian and Eulerian grids.

In both cases the horizontal displacements and rotations are constrained and the vertical displacement, $z_s(t)$, is governed by:

$$m_s\ddot{z}_s(t) = -m_s g + f_z(z_s, \dot{z}_s, t), \tag{28}$$

where m_s is the mass of the sphere, g is the acceleration of gravity and f_z is the hydrodynamic force on the sphere in the vertical direction. The Navier–Stokes equations governing the dynamics of the fluid, and Eq. (28) governing the dynamics of the sphere are solved as a coupled system using the predictor–corrector strategy proposed in [20].

Initially the sphere is located at a distance of $4.3D$ from the horizontal wall, and is impulsively started to reach a velocity corresponding to an initial Reynolds number of $Re_i = 510$. The density ratio is fixed to $\rho_s/\rho_f = 3.2$. The resulting Reynolds number just before impact is $Re = 830$ and the corresponding Stokes number is $St = 295$. Contact is assumed to take place when sphere and floor are within a distance of one cell size. In the experiments by Eames and Dalziel [5] the Reynolds number before impact was $Re = 850$, but the motion of the sphere was prescribed and no bounce was allowed to occur. To simulate these conditions in our computations the dry restitution coefficient, e_{dry}, was set to zero (see Eq. (29) below).

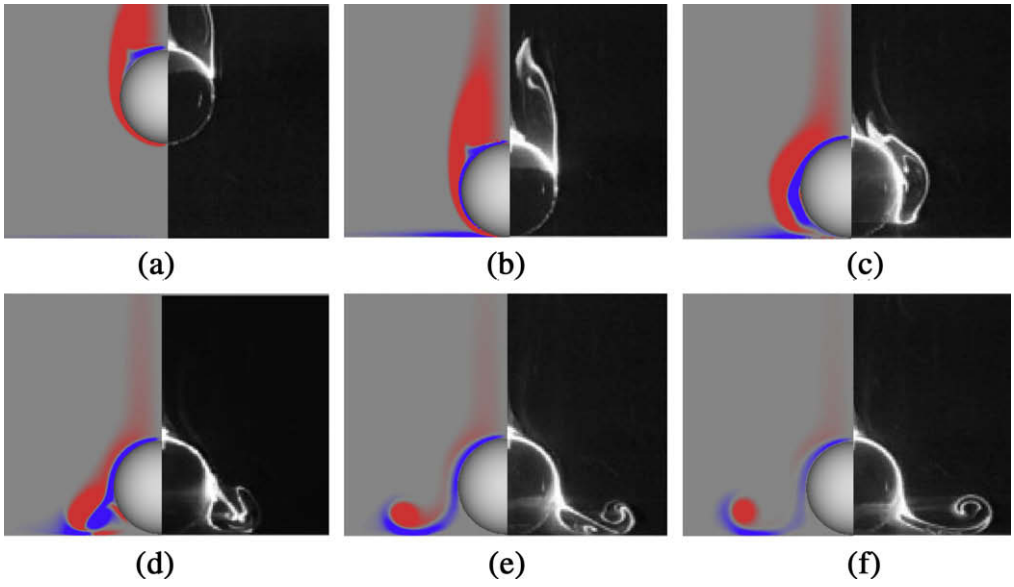

Fig. 6. Sphere-wall interaction with $e_{dry} = 0.0$. The left half in all figures are azimuthal vorticity isolines from the present computations, and the right half are snapshots from the dye visualizations in [5] at $Re = 850$. (a) $-\tau_f$, (b) 0, (c) τ_f, (d) $2\tau_f$, (e) $3\tau_f$, (f) $4\tau_f$, where $\tau_f = D/U_f$.

In Fig. 6 dye flow visualizations form the experiments in [5] are directly compared to our computations, where the flow patterns are visualized using azimuthal vorticity isolines. The computed flow patterns are in very good qualitative agreement with the experiment. Direct quantitative comparisons are not possible due to the fact that vorticity and scalars, such as dye, do not have the same dynamics as a result of their different diffusivities and the absence of vortex stretching in the case of scalars. As the sphere approaches the wall the detached shear layers and the small recirculating areas behind the sphere are evident in both experiments and computations (Fig. 6(a) and (b)). Just after impact the vorticity in the shear layers moves towards the wall generating a layer of vorticity with opposite sign on the sphere's surface (see Fig. 6(c)). As soon as this layer separates (Fig. 6(d)), a vortex dipole is formed and moves away from the sphere (Fig. 6(e) and (f)). By locating the center of these vortices in the computations and the experiments we found that in their trajectories through time are always within 5%.

Next, we considered the bouncing sphere problem. The dry restitution coefficient was the one used in the axisymmetric calculations of Ardekani and Rangel [1], $e_{dry} = 0.97$, which is typical for steel-sphere and glass-wall collisions. The collision process starts when the distance between the particle and the wall is equal to the roughness height, h_r. We assume that rough surface has a negligible effect on the viscous force until the gap between the smooth portions of surfaces becomes equal to the size of largest roughness element, h_r. This is also the moment the impact is assumed to occur. Details can be found in [1]. Just after the collision we define a new set of initial conditions for Eq. (28) as follows:

$$z_{s2} = z_{s1}; \quad \dot{z}_{s2} = -e_{dry}\dot{z}_{s1} \tag{29}$$

where $z_{s1}, \dot{z}_{s1}$ and $z_{s2}, \dot{z}_{s2}$ are the sphere's vertical position and velocity before and after the impact respectively.

In numerical simulations of contact problems it is important that the lubrication layer between the bodies is resolved. In all our computations the surface roughness, which practically determines how close the bodies can come, and numerical resolution were selected in a way that a minimum of 5–6 Eulerian grid points were present between the bodies during impact. We first considered the same case as in the above example ($Re = 830$ and $St = 295$) with a non-zero restitution coefficient, $e_{dry} = 0.97$, in order to compare the vortex dynamics with the no-bounce case. In Fig. 7 azimuthal vorticity isolines of the no-bounce (left) and bouncing sphere (right) are shown. Just before impact (Fig. 7(a)) the flow for both cases is identical since we start from the same initial conditions. At a later time and after the first impact (Fig. 7(b) and (c)), the layers of vorticity with alternating sign that were observed in the case with $e_{dry} = 0.0$ can also be seen in the bouncing sphere problem. In the latter case, however, the primary vortex originating in the wake is weaker and the upward motion of the sphere causes the shear layer at the surface to roll-up into a strong secondary vortex. As the downward motion of the sphere starts the secondary vortex pinches-off (see Fig. 7(d)) and by the time the second bounce occurs it is dissipated. As a result the wake and secondary vortices do not form the dipole structure seen in the no bounce case.

Ardekani and Rangel [1] defined a total restitution coefficient, $e_T = U_a/U_f$, where U_a is the velocity of the sphere at $tU_f/D = 0.07$ after the impact time, t_c, which measures the dissipative effect of the fluid, as it is drained and subsequently reenters the layer between sphere and wall. For the present case we found $e_T = 0.63$. A direct comparison with the computations reported in [1], where they reported $e_T = 0.92$, is not possible because of the differences in the Reynolds numbers.

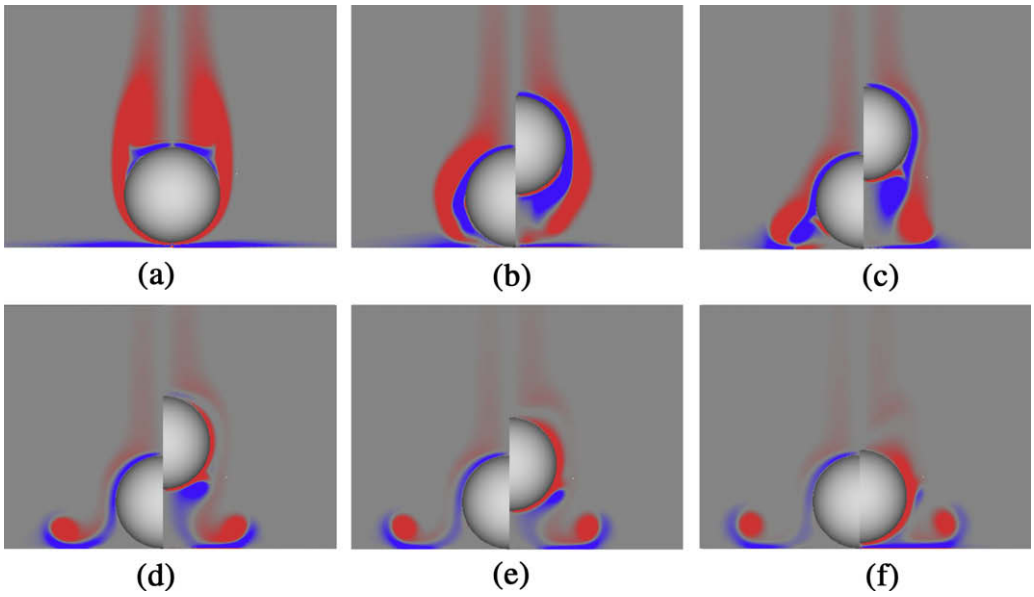

Fig. 7. A comparison of $e_{dry} = 0.00$ (left) and $e_{dry} = 0.97$ (right) computations for the sphere-wall interaction at $Re = 830$. (a) $-0.1\tau_f$, (b)τ_f, (c)$2\tau_f$, (d) $3\tau_f$, (e) $3.5\tau_f$, (f) $4.1\tau_f$, where $\tau_f = D/U_f$.

Our e_T value, however, is consistent with the trend reported in [1] where a decrease in e_T was observed with increasing Reynolds number and constant Stokes number. In particular they found a decrease of 5.0% for e_T at $St = 301$ when Re increases from 35 to 162. To further verify the accuracy of our formulation we also conducted a computation that closely matches the low Reynolds number conditions in the simulation by Ardekani and Rangel [1]. The Reynolds Number before impact was $Re = 76.8$ and the Stokes number $St = 299$. In this case we computed an $e_T = 0.91$, which is in very good agreement with the reference results of $e_T = 0.92$.

4. Summary and conclusions

In the present work a MLS reconstruction procedure for immersed-boundary type methods is proposed. The overall method is a generalization of the formulation initially proposed by Uhlmann [18], where the main difference with existing direct-forcing schemes (i.e. [6,10,2]) is that the evaluation of the forcing function is done on the Lagrangian nodes rather than the Eulerian points. The main advantages of this strategy compared to existing direct-forcing schemes can be summarized as follows:

1. It is much more versatile since it decouples the local disctetization from the computation of the forcing function and, therefore, can be implemented into structured or unstructured codes in a straightforward manner. Most of the available direct-forcing schemes have been developed in the framework of finite-difference or finite-volume formulations on Cartesian grids.
2. It is very robust in dealing with collisions of multiple bodies. The forcing function is built and appropriately scaled based on the contributions of all bodies in the vicinity of an Eulerian point without special treatments. In most direct-forcing methods the presence of two or more Lagrangian markers from different bodies in the proximity of the same Eulerian grid point is usually the source of ambiguity.

The method was also found to maintain the second-order spatial accuracy of the underlying finite-difference solver. Most importantly, we have demonstrated that when combined with the scheme we propose to compute the surface forces, it has a sharp-like behavior similar to sharp Eulerian direct-forcing schemes and boundary conforming methods. The overall computational cost of the proposed formulation is comparable to other 'direct' forcing approaches available in the literature (i.e. [10,2]). In the case of stationary bodies the interface tracking step, as well as the computation of the shape functions can be done in a pre-processing module and stored in memory. Then, the computational effort associated with the forcing step reduces to the evaluation of the weighted sums for interpolating the intermediate velocities Eq. (9) and extrapolating the force Eq. (11)

[15] C. Peskin, Flow patterns around heart valves – numerical method, J. Comput. Phys. 10 (2) (1972) 252.
[16] U. Piomelli, E. Balaras, A. Pascarelli, Turbulent structures in accelerating boundary layers, J. Turbul. 1 (2000) 001.
[17] H. Udaykumar, W. Shyy, M. Rao, Elafint: a mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, Int. J. Numer. Meth. Fl. 22 (8) (1996) 691–712.
[18] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476.
[19] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, J. Comput. Phys. 215 (2006) 12–40.
[20] J. Yang, S. Preidikman, E. Balaras, A strongly coupled embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies, J. Fluid Struct. 24 (2) (2008) 167–182.