Moving immersed boundary method for fluid–solid interaction

Shang-Gui Cai, Abdellatif Ouahsine, Yannick Hoarau

Moving immersed boundary method for fluid-solid interaction

Shang-Gui Cai (蔡尚炅),[1,2] Abdellatif Ouahsine,[1] and Yannick Hoarau[3]

[1)]*Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7337 Roberval, Centre de Recherche Royallieu, CS 60319, 60203 Compiègne Cedex, France*

[2)]*Aix Marseille Univ, CNRS, Centrale Marseille, M2P2 UMR 7340, Marseille 13451, France*

[3)]*Université de Strasbourg, CNRS, ICUBE UMR 7357 Strasbourg, 67000, France*

(*Electronic mail: shanggui.cai@univ-amu.fr)

(Dated: 2 April 2022)

A strongly coupled algorithm is presented for the incompressible fluid-rigid body interaction using the moving immersed boundary method. The pressure and the boundary force are treated as Lagrange multipliers to enforce the incompressibility and no-slip wall constraints. To compute the two unknowns from the velocity field, we adopt the fractional step algorithm and successively apply the two constraints. A Poisson equation and a moving force equation are derived for the pressure and the boundary force respectively. As both coefficient matrices are formulated to be symmetric and positive-definite, the resulting linear systems are solved efficiently with the conjugate gradient solver. The strongly-coupled nonlinear fluid-solid system is achieved by a fixed point iteration. To improve the computational efficiency, we only iterate the moving force equation with the rigid body motion equation and the time-consuming pressure Poisson equation is solved once the sub-iteration is finished. The proposed method is validated with various benchmark tests and the results compare well with the literature.

I. INTRODUCTION

One primary challenge of fluid-solid interaction (FSI) simulation is dealing of complicated and time-dependent interface between two physical fields. Body-fitted mesh methods, such as finite volume or finite element discretization of fluid equations in arbitrary Lagrangian–Eulerian formulation, need mesh-deforming or re-meshing regularly to accommodate the time-varying solid boundaries. It is very costly for large scale problems and prone to mesh interpolation errors and divergence due to severe mesh distortion. The Chimera/overset grid method[1] circumvents these issues by employing multiple computational fluid domains covered by independent meshes. One background mesh, usually a Cartesian mesh, is utilized to discretize the global domain, while the others are body-fitted meshes surrounding the objects. The Chimera/overset grid method is capable to capture the boundary layer well at moderate and high Reynolds number, comparable to classical body-fitted mesh methods, since grids can be allocated efficiently when aligning to solid boundaries. However constructing a body-fitted mesh of high quality for complex geometries is still non-trivial. Simple discretization method like finite difference may require a coordinate transformation to work on a curvilinear mesh. In such circumstance finite volume or finite element method is usually used and the resulting system is solved with sophisticated solvers, such as preconditioned Krylov subspace methods (e.g. conjugate gradient) or algebraic multigrid method. Moreover since the fluid equations are computed on different domains, it is difficult to ensure conservation.

Alternatively, the immersed boundary method (IBM)[2–8] adopts a surface mesh for the solid boundary, thereby the fluid is only simulated on the background mesh. Using a simple fluid mesh is advantageous not only for the ease of mesh generation but also for the accuracy matter. The Cartesian mesh produces the best mesh quality and hence high order accuracy can be achieved. The matrix assembly is also simplified and the matrix condition is well preserved. Most importantly, it facilitates the use of many simple and efficient methods developed for Cartesian mesh. The finite difference discretization can be constructed into high order accuracy easily and highly efficient solvers, such as geometrical multigrid method and fast Poisson solver, can be used straightforward. Parallelization of program is relatively simple to a good efficiency. However as no underlying mesh will conform to the solid boundary, the imposition of boundary condition at the interface can be tricky. Interpolation methods are often selected to overcome this problem, either by locally reconstructing the velocity [9,10] or by adopting a source forcing function in the

momentum equation [5–8]. In fact, the two approaches are essentially the same as they both rectify the velocity in the vicinity of immersed boundary to produce solid effect[11].

For the source forcing approach, it is crucial to compute the boundary force accurately to ensure a physical boundary condition. Uhlmann[3] developed an efficient direct forcing IBM for simulating particle suspensions, where an explicit formulation is used for the boundary force. As a result, the no-slip wall condition is not satisfied and large errors occur at the interface. Kempe and Fröhlich[4] reduced the error through an additional forcing loop, which is solved with only a few number of iterations without convergence. Therefore the no-slip wall condition will not be fully verified. Full convergence however requires considerable computational efforts, thus it is often relaxed in practice[12]. Actually, the boundary force can be viewed as a Lagrange multiplier for the no-slip constraint. an implicit immersed boundary projection method (IBPM) is proposed by Taira and Colonius[13] by integrating the boundary force and the pressure into a modified Poisson equation. The IBPM is very accurate since it enforces the divergence-free and the no-slip constraints simultaneously. In spite of the mathematical rigor and completeness, the IBPM is less efficient and stable than the explicit IBMs, as an enlarged system is formulated and evolves in time for moving bodies. Instead of solving the pressure and the boundary force in one single system, the moving immersed boundary method (MIBM) was proposed by Cai et al.[5–8] to decouple the pressure and the boundary force into two independent systems. The pressure Poisson coefficient matrix is kept unchanged and hence preconditioners only need to be constructed in the initial step. The boundary force is computed from a moving force equation after operator splitting. The size of the corresponding matrix is considerably reduced as it is only proportional to the number of Lagrangian markers defining the immersed boundary, which is much smaller than the fluid mesh numbers. The MIBM is found very efficient and accurate for moving boundary problems. In this work, we will extend the MIBM to fluid-solid interaction applications by incorporating the rigid body dynamics.

Generally, two approaches can be used for the fluid-solid interaction time coupling: the monolithic (direct) approach and the partitioned (segregated) approach. For the monolithic method, all the unknowns from the fluid and solid equations are formulated into a single large system along with interface conditions. While the partitioned method formulates the two subproblems into different systems and solve them separately. The monolithic approach is advantageous from the stability point of view, but it requires considerable efforts on code modifications. The partitioned approach can be further classified into two subgroups, namely the explicit coupling (weakly or

loosely coupled) and the implicit coupling (strongly or tightly coupled). Weak coupling is very efficient since it solves the two subproblems in a sequential manner, but it leaves the interface conditions unfulfilled. As a result, spurious energy is generated at the interface, which leads to unstable solutions, which is often called the added mass effect[14,15]. The stability criterion mainly depends on the density ratio between the solid and the fluid. Below a certain threshold the calculation diverges immediately for any chosen time step for incompressible fluids[16]. The strongly coupled method iterates the two subproblems until the interface conditions are satisfied, thus it is comparable to the monolithic method. In the present work, we will design a partitioned but strongly coupled scheme with the moving immersed boundary method.

The organization of this paper is as follows. First we present the general fluid-solid interaction problem in the immersed boundary formulation. In the following part the MIBM and the strongly coupled partitioned scheme are presented in detail. Numerical simulations are then performed with the proposed method and validated with benchmark tests. Conclusions are drawn in the final section.

II. METHODOLOGY

Figure 1 illustrates a general description of a rigid body motion in an incompressible fluid, where the fluid and the rigid body occupy the physical domains Ω_f and Ω_s respectively. The fluid-solid interaction takes place at the their common boundary $\partial\Omega_i = \Omega_f \cap \Omega_s$. The whole system is subjected to the gravitational acceleration $\mathbf{g}$.

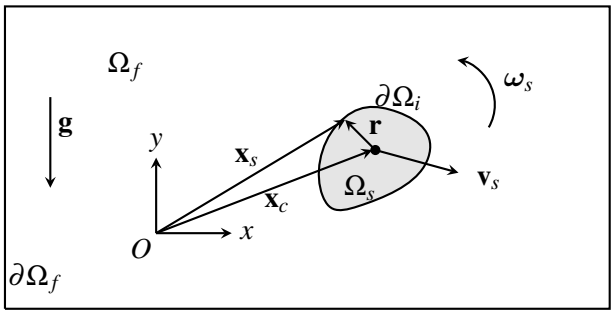


FIG. 1: Schematic representation of the fluid-solid interaction problem.

The fluid subproblem is usually described in Eulerian coordinates, which is governed by the

incompressible viscous Navier-Stokes equations

$$\frac{\partial \mathbf{v}_f}{\partial t} + \nabla \cdot (\mathbf{v}_f \otimes \mathbf{v}_f) = \nabla \cdot \boldsymbol{\sigma}_f + \mathbf{g}, \tag{1a}$$

$$\nabla \cdot \mathbf{v}_f = 0, \tag{1b}$$

where $\mathbf{v}_f$ represents the fluid velocity vector and the fluid stress tensor $\boldsymbol{\sigma}_f$ (divided by ρ_f) is given by

$$\boldsymbol{\sigma}_f = -\frac{p}{\rho_f}\mathbf{I} + \nu \left(\nabla \mathbf{v}_f + (\nabla \mathbf{v}_f)^{\mathrm{T}}\right), \tag{1c}$$

where p designates the fluid pressure, ρ_f the fluid density, ν the fluid kinematic viscosity. Appropriate initial and boundary conditions are assumed to the fluid Navier-Stokes equations to ensure that the problem is well posed.

Contrarily the solid subproblem is often treated in a Lagrangian framework. The motion of a rigid body is governed by the Newton-Euler equations, which can be expressed as

$$m_s \frac{d\mathbf{v}_s}{dt} = \rho_f \int_{\partial\Omega_i} \boldsymbol{\sigma}_f \cdot \mathbf{n} ds + m_s(1 - \frac{\rho_f}{\rho_s})\mathbf{g}, \tag{2a}$$

$$I_s \frac{d\boldsymbol{\omega}_s}{dt} = \rho_f \int_{\partial\Omega_i} \mathbf{r} \times \left(\boldsymbol{\sigma}_f \cdot \mathbf{n}\right) ds, \tag{2b}$$

where m_s, ρ_s, I_s represent the solid mass, density and the moment of inertia respectively. $\mathbf{v}_s$, $\boldsymbol{\omega}_s$ designate the translational velocity and the angular velocity of the solid. $\mathbf{r} = \mathbf{x}_s - \mathbf{x}_c$ is the position vector of any surface point with respect to the solid mass center $\mathbf{x}_c$. $\mathbf{n}$ represents the outward-pointing normal vector to the surface $\partial\Omega_i$. The position of the rigid body can be obtained by integrating the kinematic equations

$$\frac{d\mathbf{x}_c}{dt} = \mathbf{v}_s, \tag{2c}$$

$$\frac{d\theta_c}{dt} = \boldsymbol{\omega}_s, \tag{2d}$$

where θ_c designates the rotation angle around the solid mass center.

On the fluid-solid interface $\partial\Omega_i$ the continuity of velocity

$$\mathbf{v}_f = \mathbf{v}_s + \boldsymbol{\omega}_s \times \mathbf{r}, \tag{3}$$

needs to be satisfied in order to take the fluid-solid interaction into account.

The immersed boundary method approximates the above fluid-solid interaction problem by replacing the solid domain with the surrounding fluid. To account for the presence of the immersed

solid, a boundary force $\mathbf{f}$ is introduced and added into the fluid momentum equation. Therefore the fluid is simply simulated in a fixed domain $\overline{\Omega} = \Omega_f(t) \cup \Omega_s(t)$ irrespective to the movement of the immersed solid. The fluid-solid interaction problem in the immersed boundary formulation becomes

$$\frac{\partial \mathbf{v}_f}{\partial t} + \nabla \cdot (\mathbf{v}_f \otimes \mathbf{v}_f) = -\frac{1}{\rho_f}\nabla p + \nu \nabla^2 \mathbf{v}_f + \mathbf{f} \quad \text{in } \overline{\Omega}, \tag{4a}$$

$$\nabla \cdot \mathbf{v}_f = 0 \quad \text{in } \overline{\Omega}, \tag{4b}$$

$$m_s \frac{d\mathbf{v}_s}{dt} = -\rho_f \int_{\Omega_s} \mathbf{f} dV + m_s(1 - \frac{\rho_f}{\rho_s})\mathbf{g}, \tag{4c}$$

$$I_s \frac{d\boldsymbol{\omega}_s}{dt} = -\rho_f \int_{\Omega_s} \mathbf{r} \times \mathbf{f} dV, \tag{4d}$$

$$\mathbf{v}_f = \mathbf{v}_s + \boldsymbol{\omega}_s \times \mathbf{r} \quad \text{on } \partial\Omega_i, \tag{4e}$$

$$\mathbf{f}(\mathbf{x},t) = \int_{\Gamma_s} \mathbf{F}(s,t)\delta(\mathbf{x} - \mathbf{X}(s,t)) ds, \tag{4f}$$

where $\mathbf{F}(s,t)$ represents the boundary force defined on the Lagrangian position $\mathbf{X}(s,t)$ and $\mathbf{f}(\mathbf{x},t)$ on the Eulerian frame respectively. δ designates the Dirac delta function. Note that the effect of gravity in the fluid momentum equation is from now on incorporated into the pressure.

The fluid governing equations (4) are discretized in time as

$$\frac{\mathbf{v}_f^{n+1} - \mathbf{v}_f^n}{\Delta t} + \frac{3}{2}\mathscr{N}(\mathbf{v}_f^n) - \frac{1}{2}\mathscr{N}(\mathbf{v}_f^{n-1}) = -\frac{1}{\rho_f}\mathscr{G} p^{n+1} + \frac{\nu}{2}\mathscr{L}(\mathbf{v}_f^{n+1} + \mathbf{v}_f^n) + \mathscr{S}\mathbf{F}^{n+1}, \tag{5a}$$

$$\mathscr{D}\mathbf{v}_f^{n+1} = 0, \tag{5b}$$

$$\mathscr{T}\mathbf{v}_f^{n+1} = \mathbf{v}_s^{n+1} + \boldsymbol{\omega}_s^{n+1} \times \mathbf{r}^{n+1} \quad \text{on } \partial\Omega_i^{n+1}, \tag{5c}$$

where second order semi-implicit time stepping schemes are utilized for the velocity, namely the explicit Adams-Bashforth scheme for the convection and the implicit Crank-Nicholson for the diffusion. Euler implicit scheme is applied to the two Lagrange multipliers, that is, the pressure and the boundary force. $\mathscr{N}$, $\mathscr{L}$, $\mathscr{D}$, $\mathscr{G}$ represent the convection, Laplacian, divergence, and gradient operators respectively. Since the fluid grid is not coincident with the solid node, $\mathscr{T}$ and $\mathscr{S}$ are the interpolation and spreading operators that are responsible for the field exchange between two grids (see Figure 2). The discrete delta function is employed for the construction of $\mathscr{T}$ and $\mathscr{S}$

$$\delta_h(\mathbf{x} - \mathbf{X}) = \frac{1}{h^2}\phi(\frac{x - X}{h})\phi(\frac{y - Y}{h}). \tag{6}$$

In this work, we select the three-point kernel proposed in Ref. 17 aimed for staggered grid, whose

one-dimensional form is

$$\phi(r) = \begin{cases} \frac{1}{3}\left(1+\sqrt{-3r^2+1}\right), & |r| < 0.5, \\ \frac{1}{6}\left(5-3|r|-\sqrt{-3(1-|r|)^2+1}\right), & 0.5 \leqslant |r| < 1.5, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

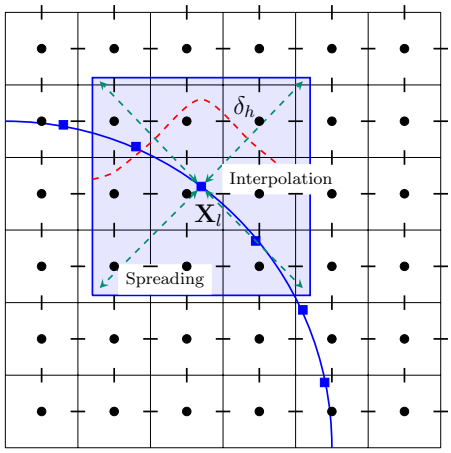


FIG. 2: Illustration of interpolation and spreading procedures with a discrete delta function.

With respect to the solid equation, we choose a simple Euler implicit scheme

$$m_s \frac{\mathbf{v}_s^{n+1} - \mathbf{v}_s^n}{\Delta t} = -\rho_f \int_{\Omega_s} \mathbf{f}^{n+1} dV + m_s(1 - \frac{\rho_f}{\rho_s})\mathbf{g}, \tag{8a}$$

$$I_s \frac{\boldsymbol{\omega}_s^{n+1} - \boldsymbol{\omega}_s^n}{\Delta t} = -\rho_f \int_{\Omega_s} \mathbf{r} \times \mathbf{f}^{n+1} dV, \tag{8b}$$

$$\frac{\mathbf{x}_c^{n+1} - \mathbf{x}_c^n}{\Delta t} = \mathbf{v}_s^{n+1}, \tag{8c}$$

$$\frac{\boldsymbol{\theta}_c^{n+1} - \boldsymbol{\theta}_c^n}{\Delta t} = \boldsymbol{\omega}_s^{n+1}. \tag{8d}$$

The total amount of force and torque are not changed during the transfer between Lagrangian and Eulerian locations, owing to the partition of unity of the discrete delta function. The solid equations can be solved as follows

$$m_s \frac{\mathbf{v}_s^{n+1} - \mathbf{v}_s^n}{\Delta t} = -\rho_f \sum_{l=1}^{n_b} \mathbf{F}^{n+1} \Delta V_l + m_s(1 - \frac{\rho_f}{\rho_s})\mathbf{g}, \tag{9a}$$

$$I_s \frac{\boldsymbol{\omega}_s^{n+1} - \boldsymbol{\omega}_s^n}{\Delta t} = -\rho_f \sum_{l=1}^{n_b} \mathbf{r} \times \mathbf{F}^{n+1} \Delta V_l, \tag{9b}$$

where ΔV_l represents the area/volume associated with the Lagrangian marker. Therefore, provided the boundary force, the solid equations can be easily solved. For the fluid part, we employ an operator splitting scheme as

(1) Prediction step for $\hat{\mathbf{v}}_f^{n+1}$

$$\frac{\hat{\mathbf{v}}_f^{n+1}-\mathbf{v}_f^n}{\Delta t}+\frac{3}{2}\mathscr{N}(\mathbf{v}_f^n)-\frac{1}{2}\mathscr{N}(\mathbf{v}_f^{n-1})=-\frac{1}{\rho_f}\mathscr{G}p^n+\frac{\nu}{2}\mathscr{L}(\hat{\mathbf{v}}_f^{n+1}+\mathbf{v}_f^n). \tag{10}$$

(2) Immersed boundary forcing step for imposing the no-slip wall condition at the interface

$$\frac{\tilde{\mathbf{v}}_f^{n+1}-\hat{\mathbf{v}}_f^{n+1}}{\Delta t}=\mathscr{S}\mathbf{F}^{n+1}, \tag{11a}$$

$$\mathscr{T}\tilde{\mathbf{v}}_f^{n+1}=\mathbf{v}_s^{n+1}+\boldsymbol{\omega}_s^{n+1}\times\mathbf{r}^{n+1} \quad \text{on } \partial\Omega_i^{n+1}. \tag{11b}$$

Applying the no-slip constraint, we obtain

$$\mathscr{T}\mathscr{S}\mathbf{F}^{n+1}=\frac{\mathbf{v}_s^{n+1}+\boldsymbol{\omega}_s^{n+1}\times\mathbf{r}^{n+1}-\mathscr{T}\hat{\mathbf{v}}_f^{n+1}}{\Delta t}, \tag{12a}$$

$$\tilde{\mathbf{v}}_f^{n+1}=\hat{\mathbf{v}}_f^{n+1}+\Delta t\mathscr{S}\mathbf{F}^{n+1}. \tag{12b}$$

Given $\mathscr{M}=\mathscr{T}\mathscr{S}$ the moving force coefficient matrix, then we obtain a compact form of the moving force equation

$$\mathscr{M}\mathbf{F}^{n+1}=\mathbf{F}^e, \tag{13}$$

with

$$\mathbf{F}^e=\frac{\mathbf{v}_s^{n+1}+\boldsymbol{\omega}_s^{n+1}\times\mathbf{r}^{n+1}-\mathscr{T}\tilde{\mathbf{v}}_f^{n+1}}{\Delta t}, \tag{14}$$

where $\mathbf{F}^e$ is the explicit force used by Kempe and Fröhlich[4]. Given the solid velocity and position at time level $n+1$, the moving force equation can be solved with various linear system solvers. In the present work the moving force coefficient matrix $\mathscr{M}$ is found to be symmetric and positive-definite. Hence the moving force equation can be converged to the machine precision very quickly by using the conjugate gradient method, even without preconditioning. Moreover, the size of the moving force coefficient matrix is proportional to the number of nodes on the immersed boundary, which is in general much smaller than the size of the fluid matrix. Since the moving force coefficient matrix does not involve the fluid matrix, its update is computationally inexpensive in the case of moving boundaries.

(3) Projection step for obtaining a divergence free velocity $\mathbf{v}_f^{n+1}$

$$\frac{\mathbf{v}_f^{n+1} - \tilde{\mathbf{v}}_f^{n+1}}{\Delta t} = -\mathscr{G}\phi^{n+1}, \tag{15a}$$

$$\mathscr{D}\mathbf{v}_f^{n+1} = 0. \tag{15b}$$

Applying the divergence operator to (15a) along with the divergence free condition (15b), the projection step is actually performed as

$$\mathscr{L}\phi^{n+1} = \frac{1}{\Delta t}\mathscr{D}\tilde{\mathbf{v}}_f^{n+1}, \tag{16a}$$

$$\mathbf{v}_f^{n+1} = \tilde{\mathbf{v}}_f^{n+1} - \Delta t\mathscr{G}\phi^{n+1}. \tag{16b}$$

The final pressure is advanced by

$$p^{n+1} = p^n + \phi^{n+1} - \frac{\nu}{2}\mathscr{D}\hat{\mathbf{v}}_f^{n+1}. \tag{17}$$

Now we will present a novel coupling approach for the interaction of the two subproblems. Mathematically speaking, the FSI problem of equation (4) is accomplished by the Lagrange multiplier method to joint two subdomains on a common boundary. The unknowns of the entire system are ($\mathbf{v}_f^{n+1}$, p^{n+1}) for the fluid, ($\mathbf{x}_s^{n+1}$, $\dot{\mathbf{x}}_s^{n+1}$) for the solid, and $\mathbf{F}^{n+1}$ for the interface condition. The conventional strongly coupled method iterates those variables at each time step in order to match the interface condition. Even if it leads to accurate and stable solutions, solving implicit coupling usually exhibits a prohibitive computational cost. This becomes more unaffordable in ALE formulation, since the mesh and the associated matrices have to be updated regularly.

Fernández et al.[18] proposed an efficient coupling method by taking advantage of the projection method. The prediction (ALE-advection-viscous) step is ruled out from the FSI loop to reduce computational cost, so that the mesh and the associated matrices are computed once at each time step. Finally only the projection step is coupled implicitly to ensure stability. It has been shown that this semi-implicit scheme is stable for a reasonable range of the discretization parameters, compared to the explicit coupling approach.

We extend this idea to the moving immersed boundary method. However the projection step usually is the most time-consuming part in the projection method. In spite of various methods (e.g. Aitken relaxation) are available to accelerate the coupling procedure, the computational cost still remains high. We also notice that in equation (9) the solid is coupled to the fluid by the boundary force but not the pressure, and in equation (12) the boundary force is determined by

the solid velocity and position. Therefore, we can move out the time-consuming projection step from the FSI iteration, and replace it by the immersed boundary forcing step. Following Ref. 18, the prediction step is not included in the FSI iteration either, as it is performed on a stationary combined domain in MIBM and the boundary force is not incorporated in this step for the solid effects. The moving force equation can be considered as an implicit equation for the no-slip boundary condition for the fluid at the interface. Therefore the implicit coupling of the immersed boundary forcing step with the solid equations features a strongly coupled FSI method.

Algorithm 1: Novel implicit coupling scheme

Given: $\mathbf{v}_f^n$, p^n, $\mathbf{x}_s^n$, $\dot{\mathbf{x}}_s^n$;

(Fluid) Predict the velocity $\tilde{\mathbf{v}}_f^{n+1}$ using (10);

Initialize values: $(\cdot)^{k=0,n+1} = (\cdot)^n$, where $(\cdot)$ includes $\mathbf{x}_s, \dot{\mathbf{x}}_s, \mathbf{F}$;

for $k = 0$ to $k_{\max}$ **do**

  (Fluid) Construct or update the interpolation operator matrix $\mathscr{T}(\mathbf{x}_s^{k,n+1})$ and the moving force coefficient matrix $\mathscr{M}(\mathbf{x}_s^{k,n+1})$;

  (Fluid) Interpolate the fluid velocity $\mathscr{T}(\mathbf{x}_s^{k,n+1})\tilde{\mathbf{v}}_f^{n+1}$;

  (Interface) Solve the moving force equation (12a) for $\mathbf{F}^{k+1,n+1}$ with $\dot{\mathbf{x}}_s^{k,n+1}$;

  (Solid) Compute the solid equations for ($\mathbf{x}_s^{k+1,n+1}$, $\dot{\mathbf{x}}_s^{k+1,n+1}$) using (9);

  if $||(\cdot)^{k+1,n+1} - (\cdot)^{k,n+1}||/||(\cdot)^{k+1,n+1}|| <$ tolerance **then**

    $(\cdot)^{n+1} = (\cdot)^{k+1,n+1}$;

    break;

  else

    $k = k + 1$;

  end

end

(Fluid) Correct the fluid velocity to $\hat{\mathbf{v}}_f^{n+1}$ with $\mathbf{F}^{n+1}$ using (12b);

(Fluid) Solve the pressure Poisson equation and compute the final velocity $\mathbf{v}_f^{n+1}$ and the pressure p^{n+1} using (16a), (16b), (17).

Note that the moving force equation is non-linear, because the interpolation, spreading op-

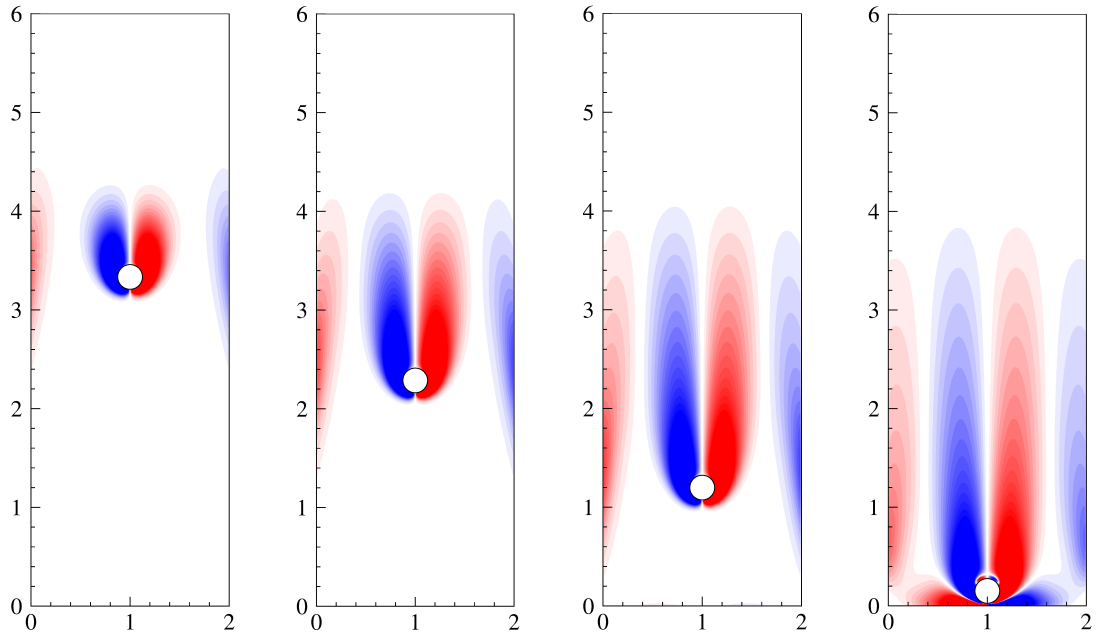
erators and the boundary force are functions of solid position $\mathbf{x}_s^{n+1}$, namely $\mathscr{M}^{n+1}\mathbf{F}^{n+1} = \mathscr{T}(\mathbf{x}_s^{n+1})\mathscr{S}(\mathbf{x}_s^{n+1})\mathbf{F}(\mathbf{x}_s^{n+1})$. We can linearize this equation by treating the moving force coefficient matrix explicitly $\mathscr{M}(\mathbf{x}_s^n)$, but this will decrease the overall accuracy due to the time lag, as indicated in Ref. 19. In order to preserve a high accuracy, fully implicit implementation of the moving force equation is considered in the Algorithm 1.

III. NUMERICAL RESULTS

A. Freely falling cylinder in a confined channel

We first investigate the motion of a circular cylinder falling freely between two parallel walls in a quiescent fluid. The width and height of the computational domain are chosen to be 2 cm and 6 cm. The circular cylinder with a diameter of $D = 0.25$ cm is released initially from the position (1 cm, 4 cm) and falls down because of gravity. The density of the cylinder and the surrounding fluid are $\rho_s = 1.25$ g/cm^3 and $\rho_f = 1.0$ g/cm^3 respectively. The fluid dynamic viscosity μ is set to 0.1 g/(cm·s).

The calculations are carried out on two different uniform meshes, i.e. $h = 1/48$ cm and $h = 1/96$ cm, to check the mesh sensitivity. The immersed cylinder surface is discretized evenly with a mesh size approximated to the size of surrounding fluid cells, due to the inherent limitation of the discrete delta function. Therefore the resulting Lagrangian marker numbers are 38 and 76 respectively. The time step is chosen to be $\Delta t = 0.001$ s and the resulting maximum CFL number is reported to be 0.46. The vorticity around the falling cylinder is shown in Figure 3 at different times t = 0.2 s, 0.4 s, 0.6 s, 0.8 s. The cylinder quickly reaches a uniform falling velocity until it hits the bottom of the channel. We plot the flow quantities as a function of time in Figure 4, including the longitude position y_c of the cylinder center, the vertical velocity v_c, the Reynolds number Re and the translational kinetic energy E_T. Here Re and E_T are defined as $Re = (\rho_s D\sqrt{u_c^2 + v_c^2})/\mu$ and $E_T = 0.5m_s(u_c^2 + v_c^2)$ respectively, where u_c is the horizontal velocity component. For comparison, the results of Ref. 20 are included in Figure 4, taken from $h = 1/96$ cm. Good agreements have been obtained.



(a) $t = 0.2$ s (b) $t = 0.4$ s (c) $t = 0.6$ s (d) $t = 0.8$ s

FIG. 3: Vorticity fields at different times t = 0.2 s, 0.4 s, 0.6 s, 0.8 s for the freely falling cylinder in a confined channel problem. The contour levels are set from -15 (blue) to 15 (red) with an increment of 1.

B. Freely falling and rising cylinder in an open domain

Next we consider an object freely falling and rising in an open domain as another test. This phenomenon happens frequently in nature and a large amount of work can be found in the literature. Here we compare to the numerical results of Namkoong et al.[21] and Lacis et al.[19], using the body-fitted ALE formulation and the immersed boundary projection method respectively.

Two density ratios are considered in this study, i.e. $\rho_s/\rho_f = 1.01$ for the falling case and $\rho_s/\rho_f = 0.99$ for the rising simulation. A large computational domain is taken as $[-5D, 5D] \times [-70D, 70D]$ with free-slip boundary conditions applied at all the exterior boundaries, where $D =$ 0.5 cm is the cylinder diameter. A uniform mesh is employed to cover the computational domain, and the mesh resolution is kept to $0.04D$ in order to compare with Lacis et al.[19]. Initially the cylinder is located at $\pm 65D$, depending on the situation ($65D$ for the falling case, $-65D$ for the rising case). The Reynolds number $Re = V_t D/\nu_f$ here is 156, where V_t is the terminal velocity. Note that the Reynolds number depends on the Galileo number $G = (|\rho_s/\rho_f - 1|gD^3)^{1/2}/\nu_f$ (the gravity force divided by the viscous force, $G = 138$) and the density ratio ρ_s/ρ_f.

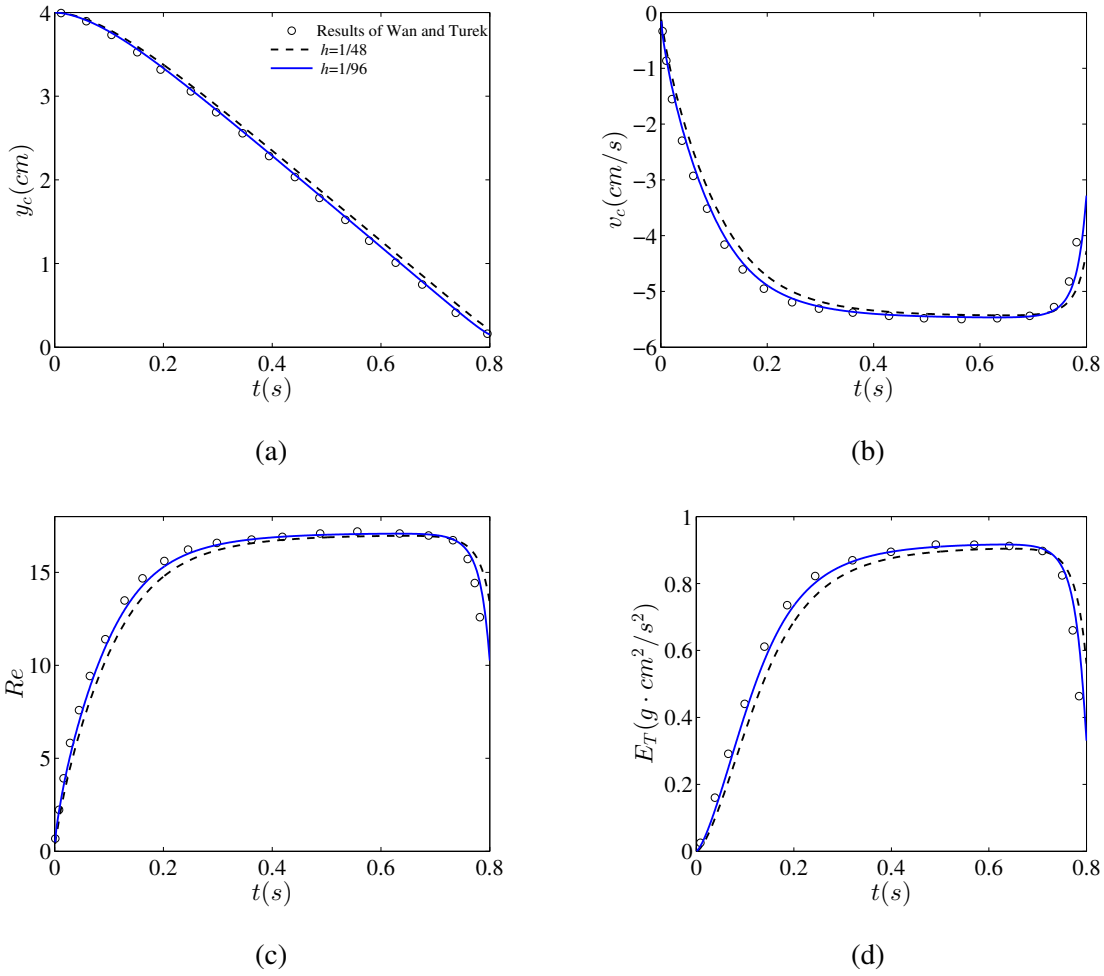
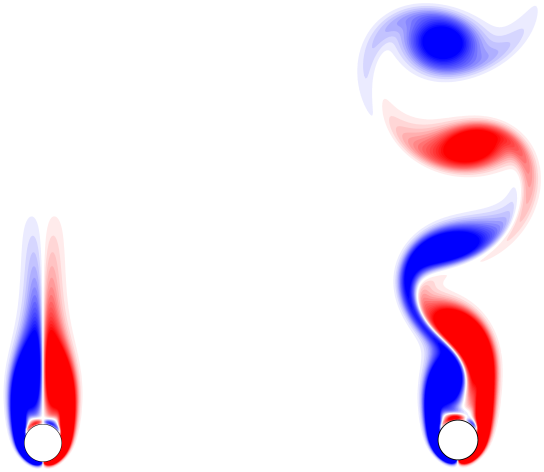


FIG. 4: Time evolution of longitude position y_c, velocity v_c of the cylinder center, the Reynolds number Re and the translational kinetic energy E_T for the freely falling cylinder in a confined channel problem. "○", result of Ref. 20; "—", present result with $h = 1/96$ cm; "- - - -", present result with $h = 1/48$ cm.

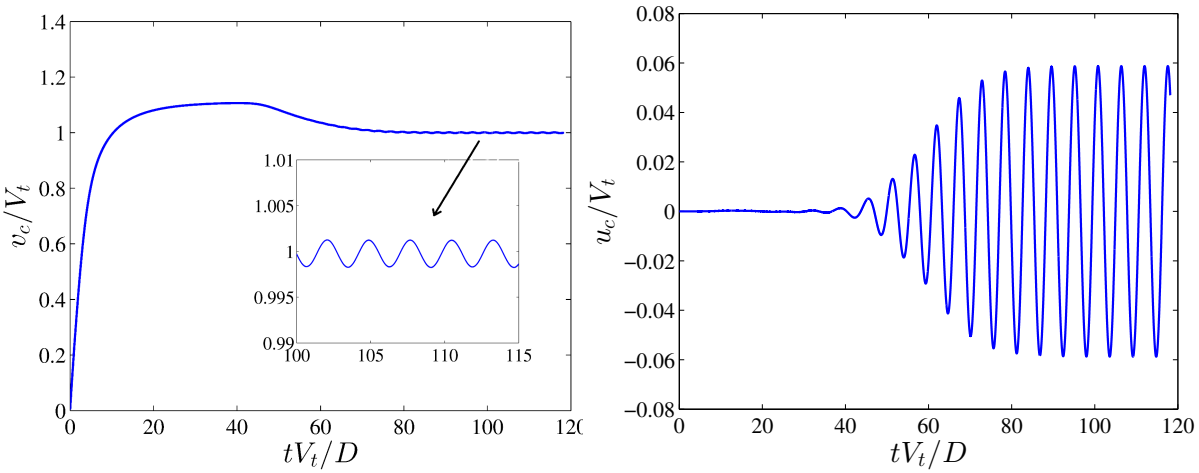
The instantaneous vorticity fields are presented in Figure 5 for the falling cylinder case. Initially symmetric vortex pair forms behind the cylinder in the beginning of falling. After that the numerical error accumulates and breaks the symmetry. At around $tV_t/D = 40$, the flow becomes unsteady and periodic vortex shedding occurs. The time histories of the velocity of the cylinder are plotted in Figure 6. Table I shows the Strouhal number $St = fD/V_t$ (f is the shedding frequency) and the coefficients of drag and lift. Present results are compared to those of Namkoong et al.[21] and Lacis et al.[19]. Good agreements have been obtained.

To illustrate the efficiency of the proposed coupling method, different strong FSI coupling



(a) $tV_t/D = 10$ (b) $tV_t/D = 90$

FIG. 5: Snapshots of vorticity fields for a freely falling cylinder in an open domain . The contour level is set from -6 (blue) to 6 (red) with an increment of 0.4.



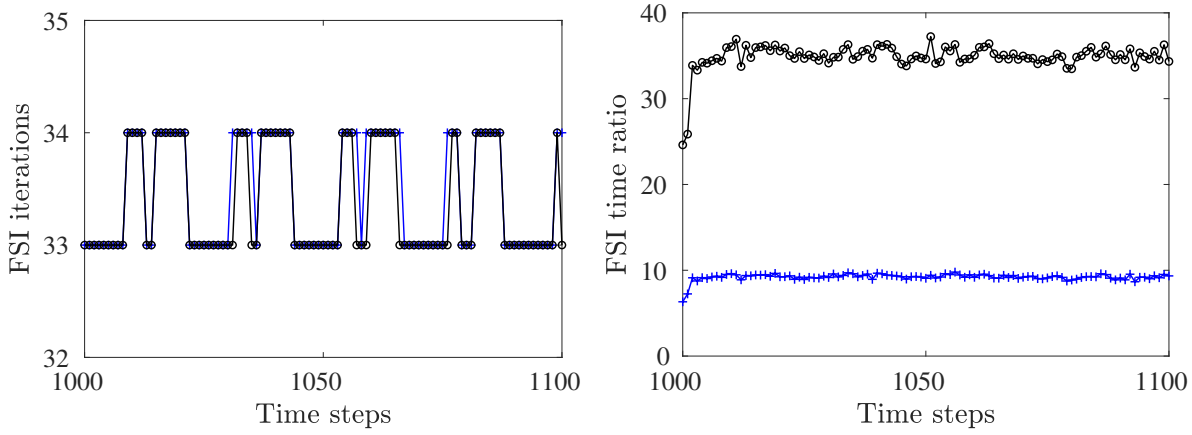
(a) (b)

FIG. 6: Time histories of the vertical and horizontal velocity for the freely rising cylinder $\rho_s/\rho_f = 0.99$.

methods are compared in Figure 7. The time consumption of the traditional strong FSI coupling method, where the pressure Poisson equation is involved in the sub-iteration, almost scales linearly with the number of sub-iterations, which is due to the fact that the solution of the pressure Poisson equation is the most dominating part. It can be seen that the proposed method reduces up to 75%

ρ_s/ρ_f	Methods	C_D	$\lvert C_L\rvert_{\max}$	St
1.01	Present	1.35	0.10	0.189
	Lacis et al.[19]	1.29	0.14	0.17185
	Namkoong et al.[21]	1.23	0.15	0.1684
0.99	Present	1.35	0.10	0.189
	Lacis et al.[19]	1.29	0.14	0.17188
	Namkoong et al.[21]	-	-	0.1687

TABLE I: The drag, lift coefficients and the Strouhal number for the freely falling and rising circular cylinder in an open domain.



(a) (b)

FIG. 7: Comparison of different strong FSI coupling strategies. (a) The number of sub-iterations in the strong FSI coupling; (b) Time consumption ratio of various strong coupling methods compared to the weak coupling method. “○", strong coupling with pressure Poisson equation in the sub-iteration; “+", the proposed strong coupling.

of the FSI coupling time within nearly the same number of sub-iteration.

C. Elliptical particle sedimentation in a confined channel

In this example, we consider the sedimentation of an elliptical particle in a narrow channel, to demonstrate the ability of current FSI algorithm for non-circular object. This example was stud-

ied previously by Xia et al.[22] for the boundary effects on the sedimentation mode. Five distinct modes of sedimentation have been found ranging from oscillating, tumbling along the wall, vertical sedimentation, horizontal sedimentation to an inclined mode. In their work, a multi-block lattice Boltzmann method is used and compared to the traditional ALE formulation.

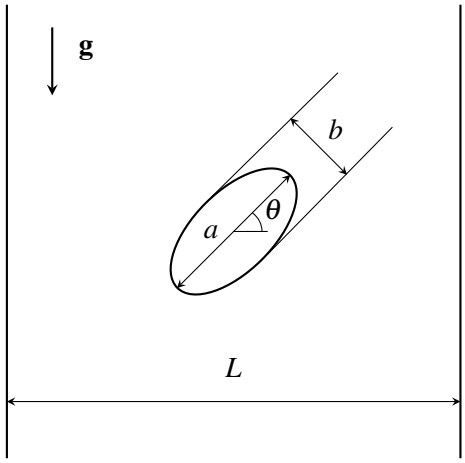


FIG. 8: Computational domain of the elliptical particle sedimentation problem.

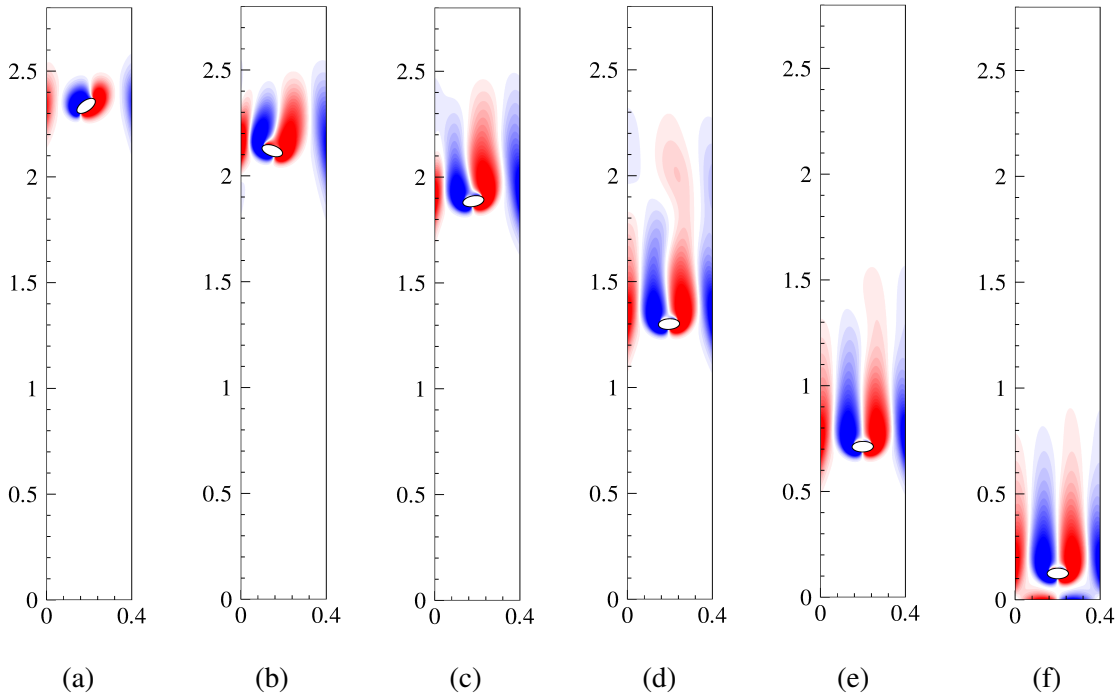


FIG. 9: Vorticity fields at different times: (a) $t = 0.1$ s ; (b) $t = 0.3$ s; (c) $t = 0.5$ s; (d) $t = 1.0$ s; (e) $t = 1.5$ s; (f) $t = 2.0$ s. The contour levels are set from -15 to 15.

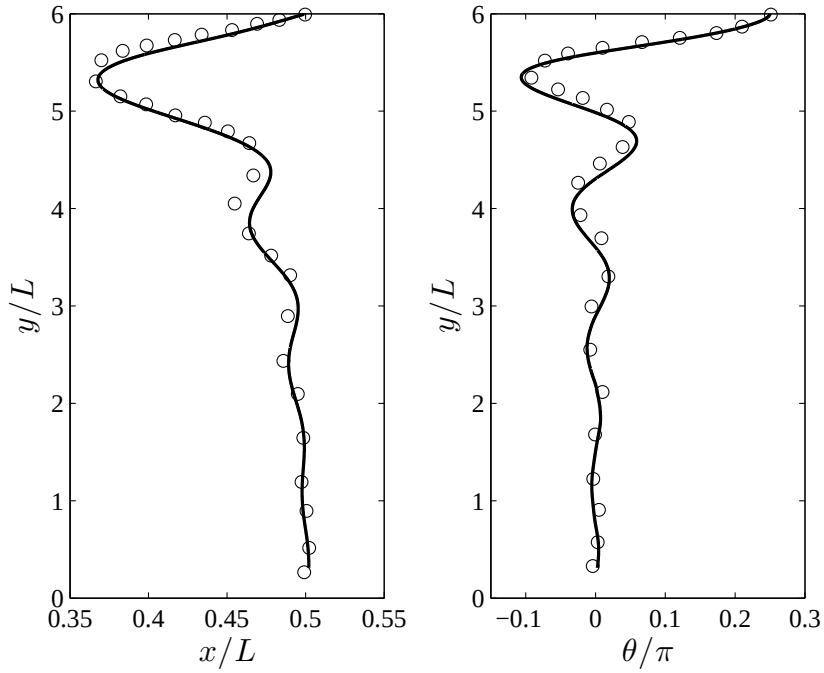


FIG. 10: Particle trajectory and orientation of the elliptical particle. "—", present results; "○", results of Xia et al.[22].

To compare with Xia et al.[22], the computational domain is selected to be $[0,L]\times[0,7L]$ with $L=0.4$ cm, as shown in Figure 8. The aspect ratio of the ellipse is $\alpha=a/b=2$, where a and b are the major and minor axes respectively. The blockage ratio is defined as $\beta=L/a=4$. The density ratio is $\rho_s/\rho_f=1.1$. The kinematic viscosity of fluid is set to $\nu=0.01$ cm^2/s. The particle starts falling in a quiescent fluid from the centroid at $(0.5L, 6L)$ with an initial angle of $\pi/4$ to break the symmetry.

No-slip boundary conditions are applied at four boundaries. A uniform mesh is employed with a gird resolution of 0.0027 cm. The time step is chosen such that the CFL condition is satisfied. Figure 9 shows the vorticity fields at different times at t = 0.1 s, 0.3 s, 0.5 s, 1.0 s, 1.5 s, 2.0 s. The trajectory and orientations are compared to the results of Xia et al.[22] in Figure 10. Good agreements have been obtained.

D. Flow around a rotating NACA0012 airfoil

The rigid objects simulated in this section so far have been the circular or elliptical particles. In this example, we consider the incompressible viscous flow over a NACA0012 airfoil to demon-

strate the capacity of current MIBM for handling sharp geometries. The shape of the NACA0012 airfoil is given by

$$Y = \pm 0.6 \cdot (0.2969\sqrt{X} - 0.1260X - 0.3516X^2 + 0.2843X^3 - 0.1015X^4), \tag{18}$$

where $X \in (0, 1.009)$ cm. The characteristic length, i.e. the airfoil length, is 1.009 cm. Following Glowinski et al.[23] and Wan and Turek[20], we select a computational domain of [-4 cm,16 cm] $\times$ [-2 cm, 2cm] with the airfoil centered at (0.42 cm,0), as shown in Figure 11.

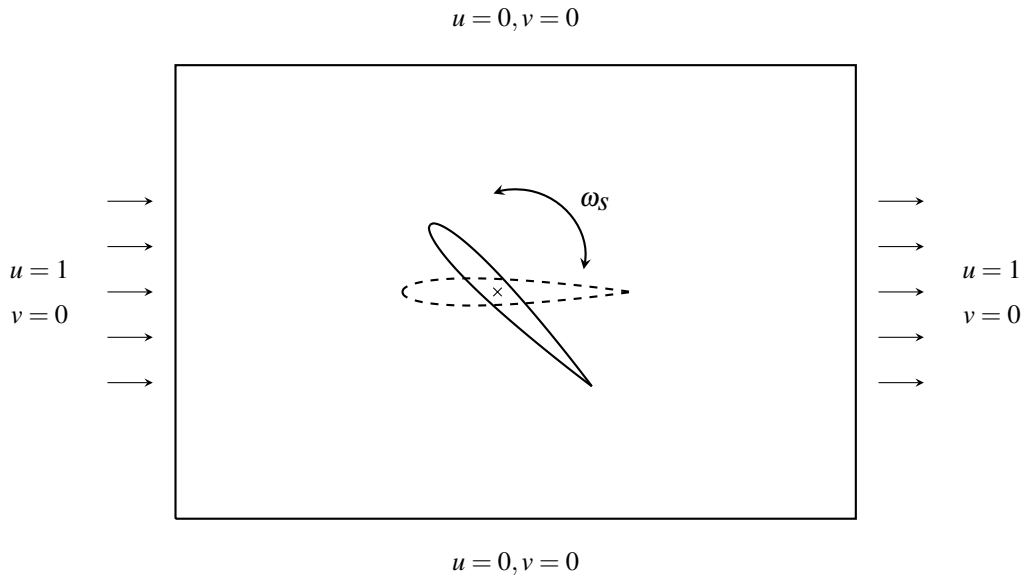
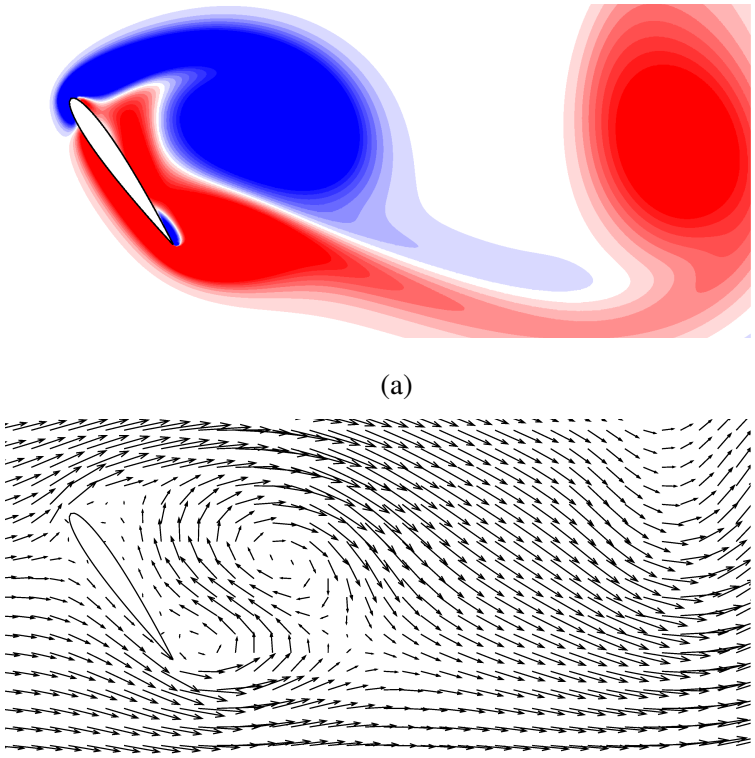


FIG. 11: Computational domain of the flow past a rotating NACA0012 airfoil. The dashed lines represent the initial position of the airfoil.

The airfoil is fixed at its mass center and is free to rotate due to hydrodynamic forces. The density of fluid is taken as $\rho_f = 1.0$ g/cm^3 and the density of solid is $\rho_s = 1.1$ g/cm^3 in this simulation. The viscosity of the fluid is $\nu_f = 0.01$ cm^2/s. Initial angular velocity and incident angle of the airfoil are set to zero. The boundary conditions of flow are given as $\mathbf{u} = (0,0)$ at y=-2 cm, 2 cm and $\mathbf{u} = (1,0)$ cm/s at x = -4 cm, 16 cm. Those boundary conditions are used in Glowinski et al.[23] and are adopted here in order to compare the results of two methods. The Reynolds number is around 101 based on airfoil length and the maximum inflow speed.

This flow is quite challenging as the leading edge of the airfoil has very small radius of curvature. To resolve the flow near the leading edge, a good resolution of the Cartesian mesh is required. Two sets of grids are chosen here to test the grid sensitivity, namely $h = 1/96$ cm and $h = 1/64$

cm. The same time step is used in both cases ($\Delta t = 0.002$ s). The resulting CFL numbers are 0.40 and 0.25 respectively.

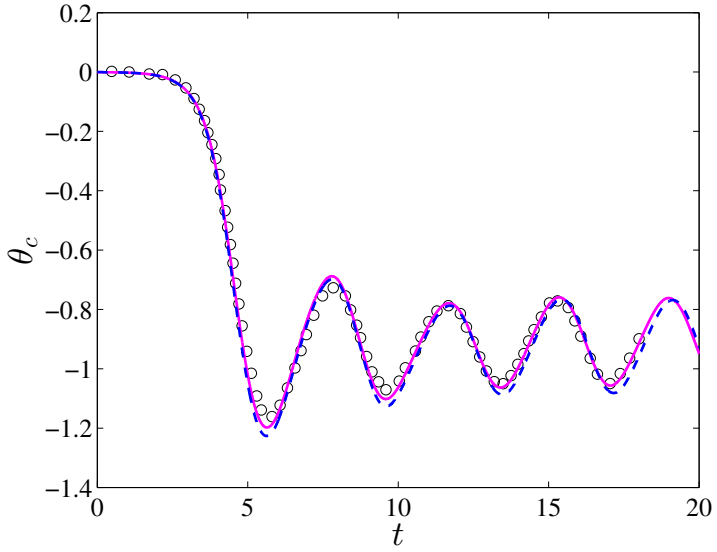


(a)

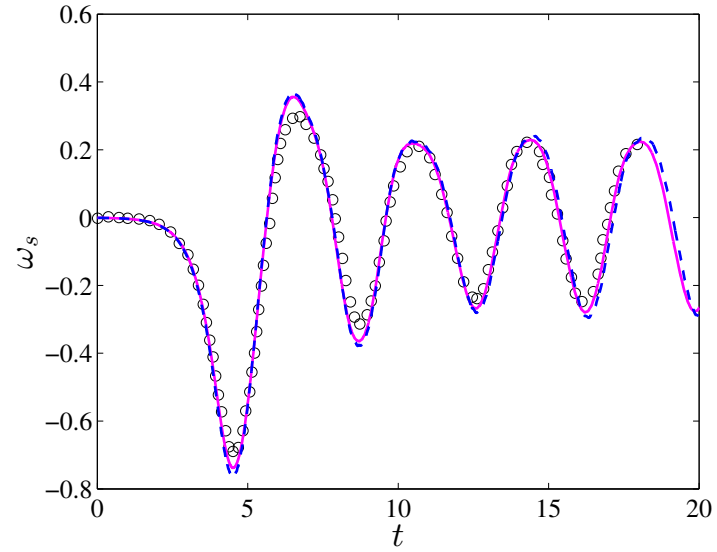
(b)

FIG. 12: Instantaneous vorticity (a) and velocity (b) of the flow over a rotating NACA0012 airfoil.

The flow fields are shown in Figure 12. The airfoil keeps a stable position with its broadside perpendicular to the in-flow direction in the beginning and finally reaches a periodic motion of oscillation. The time histories of the rotational angle and the angular velocity are plotted in Figure 13. The results are pretty smooth. For comparison, we also include the results of Glowinski et al.[23] in Figure 13 obtained by the distributed Lagrange multiplier method. Presents results match well those of Glowinski et al.[23]. We observe that the direction of oscillation depends on the numerical errors. The oscillating direction changes over different calculation parameters. An opposite oscillating case can be found in Wan and Turek[20]. Present direction of oscillation is in accord with Glowinski et al.[23].



(a)



(b)

FIG. 13: Time histories of the angle (a) and the angular velocity (b) of the rotating NACA0012 airfoil. $h = 1/96$ cm, solid line; $h = 1/64$ cm, dashed line. The measures are in rad and rad/s respectively. The results of Glowinski et al.[23] are marked with circles.

IV. CONCLUSIONS

A revised algorithm for the fluid-solid interaction was presented in this paper based on the moving immersed boundary method. Operator splitting is successively employed to achieve computational efficiency and modularity. First the pressure-velocity coupling is handled by a second order fractional step method, breaking the Navier-Stokes equations into a Helmholtz-type equation for the velocity and a Poisson equation for the pressure; Followed by the immersed boundary force splitting, a moving force equation is derived to account for the solid effect; At last a partitioned approach separates the fluid field from the solid field. By doing this, all the variables are decoupled and computed effectively with their own favored solvers. To achieve a more stable FSI coupling, a strongly coupled scheme is employed through a fixed point iteration. For better performance, we have removed the time-consuming pressure Poisson solver out from the FSI iteration and the computational time has been considerably reduced. Various cases have been tested, ranging from circular to non-circular objects with large displacements and rotations. The numerical results are compared to the benchmark, which demonstrates a good accuracy of the proposed method.

ACKNOWLEDGMENTS

This work was part of the PhD study of the first author, where the financial support of the China Scholarship Council is acknowledged. The computation was performed on the platform PILCAM2 at the Université de Technologie de Compiègne and the HPC at the Université de Strasbourg.

REFERENCES

[1]T. Deloze, Y. Hoarau, and J. Dusek, "Transition scenario of a sphere freely falling in a vertical tube," J. Fluid Mech. **711**, 40–60 (2012).

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

[3]M. Uhlmann, "An immersed boundary method with direct forcing for the simulation of particulate flows," J. Comput. Phys. **209**, 448–476 (2005).

[4]T. Kempe and J. Fröhlich, "An improved immersed boundary mehod with direct forcing for the simulation of particle laden flows," J. Comput. Phys. **231**, 3663–3684 (2012).

[5]S.-G. Cai, *Computational fluid-structure interaction with the moving immersed boundary method*, Ph.D. thesis, Université de Technologie de Compiègne (2016).

[6]S.-G. Cai, A. Ouahsine, J. Favier, and Y. Hoarau, "Improved implicit immersed boundary method via operator splitting," (Springer Verlag, 2016) Chap. 3, pp. 49–66.

[7]S.-G. Cai, A. Ouahsine, J. Favier, and Y. Hoarau, "Implicit immersed boundary method for fluid-structure interaction," La Houille Blanche **1**, 33–36 (2017).

[8]S.-G. Cai, A. Ouahsine, J. Favier, and Y. Hoarau, "Moving immersed boundary method," Int. J. Numer. Meth. Fluids **85**, 288–323 (2017).

[9]S.-G. Cai, J. Degrigny, J.-F. Boussuge, and P. Sagaut, "Coupling of turbulence wall models and immersed boundaries on Cartesian grids," J. Comput. Phys. **429**, 109995 (2021).

[10]J. Degrigny, S.-G. Cai, J.-F. Boussuge, and P. Sagaut, "Improved wall model treatment for aerodynamic flows in LBM," Comput. Fluids , 105041 (2021).

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

[12]W.-P. Breugem, "A second-order accurate immersed boundary method for fully resolved simulations of particle-laden flows," J. Comput. Phys. **231**, 4469–4498 (2012).

[13]K. Taira and T. Colonius, "The immersed boundary method: A projection approach," J. Comput. Phys. **225**, 2118–2137 (2007).

[14]P. Causin, J.-F. Gerbeau, and F. Nobile, "Added-mass effect in the design of partitioned algorithms for fluid–structure problems," Comput. Methods Appl. Mech. Engrg. **194**, 4506–4527 (2005).

[15]C. Förster, W. A. Wall, and E. Ramm, "Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows," Comput. Methods Appl. Mech. Engrg. **196**, 1278 – 1293 (2007).

[16]C. Kassiotis, A. Ibrahimbegovic, R. Niekamp, and H. G. Matthies, "Nonlinear fluid–structure interaction problem. Part I: implicit partitioned algorithm, nonlinear stability proof and validation examples," Comput. Mech. **47**, 305–323 (2011).

[17]A. M. Roma, C. S. Peskin, and M. J. Berger, "An adaptive version of the immersed boundary method," J. Comput. Phys. **153**, 509–534 (1999).

[18]M. A. Fernández, J.-F. Gerbeau, and C. Grandmont, "A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid," Int. J. Numer. Meth. Eng. **69**, 794–821 (2007).

[19]U. Lacis, K. Taira, and S. Bagheri, "A stable fluid–structure-interaction solver for low-density rigid bodies using the immersed boundary projection method," J. Comput. Phys. **305**, 300–318 (2016).

[20]D. Wan and S. Turek, "Direct numerical simulation of particulate flow via multigrid fem technique and the fictitious boundary method," Int. J. Numer. Meth. Fluids **51**, 531–566 (2006).

[21]K. Namkoong, J. Yoo, and H. Choi, "Numerical analysis of two-dimensional motion of a freely falling circular cylinder in an infinite fluid," J. Fluid Mech. **604**, 33–53 (2008).

[22]Z. Xia, K. Connington, S. Rapaka, P. Yue, J. Feng, and S. Chen, "Flow patterns in the sedimentation of an elliptical particle," J. Fluid Mech. **625**, 249–272 (2009).

[23]R. Glowinski, T. Pan, T. Hesla, D. Joseph, and J. Periaux, "A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow," J. Comput. Phys. **169**, 363–426 (2001).