INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids (2014)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/fld.3884

An immersed boundary method for fluid flows around rigid objects

A. Jendoubi, D. Yakoubi, A. Fortin*,† and C. Tibirna

GIREF, Département de mathématiques et de statistique, Pavillon Vachon, 1045 Avenue de la médecine, Université Laval, Québec, Canada

SUMMARY

In this paper, we present an immersed boundary method for solving fluid flow problems in the presence of static and moving rigid objects. A FEM is used starting from a base mesh that does not represent exactly rigid objects (non-body-conforming mesh). At each time step, the base mesh is locally modified to provide a new mesh fitting the boundary of the rigid objects. The mesh is also locally improved using edge swapping to enhance the quality of the elements. The Navier–Stokes equations are then solved on this new mesh. The velocity of moving objects is imposed through standard Dirichlet boundary conditions. We consider a number of test problems and compare the numerical solutions with those obtained on classical body-fitted meshes whenever possible. Copyright © 2014 John Wiley & Sons, Ltd.

Received 11 April 2013; Revised 26 September 2013; Accepted 19 January 2014

KEY WORDS: immersed boundary method (IB); finite element method; mesh subdivision; Navier–Stokes equations; non-body-fitted mesh

1. INTRODUCTION

Interactions between fluids and structures are frequently encountered in nature. The wings of a bird interacting with air, a fish swimming in water, heart valves interacting with the blood flow are just examples. The numerical simulation of such problems is still today a challenging task. One of the difficulties is managing the 'communications' between the solid objects, generally expressed in Lagrangian formulation, and the fluid, expressed in Eulerian coordinates.

The most common method to simulate the flow in a complex geometry is the body-fitted (BF) method where the mesh is built to fit as well as possible the geometry of the rigid objects. For moving or deforming objects, this requires a complete remeshing of the domain at each time step, which is a very difficult task in itself, and often leads to high computational cost and memory requirements.

The immersed boundary (IB) method was introduced by Peskin [1] to study flow patterns around heart valves. Its popularity is due to its ability to handle simulations with moving boundaries at low computational cost and memory requirements. The spatial discretization of the IB equations is based on a fixed Cartesian mesh for the Eulerian variables and a moving curvilinear mesh for the Lagrangian variables. The two types of variables are linked by interaction equations involving a smoothed approximation of the Dirac delta function.

This method is now widely used, and many variants can be found in the literature. Su *et al*. [2] have used in their formulation a mixture of Eulerian and Lagrangian variables, where the solid boundary is represented by Lagrangian markers exerting forces to the Eulerian fluid domain. The interactions between the Lagrangian markers and the fluid variables are enforced by simple discretized delta functions. Noor *et al*. [3] also developed the method of IB by using the method of direct forcing. A virtual force is added to the Navier–Stokes equations in order to impose the

*Correspondence to: A. Fortin, GIREF, Département de mathématiques et de statistique, Pavillon Vachon, 1045 Avenue de la médecine, Université Laval, Québec, Canada.
†E-mail: afortin@giref.ulaval.ca

interaction between the solid and the fluid. The interior and exterior of the solid are identified by a volume of solid method. Huang and Sung [4] also adopted the fictitious domain method for the simulation of the interaction between an incompressible fluid and a flexible solid. A review of different variants of IB methods can be found in the work of Mittal and Iaccarino [5].

Glowinski *et al*. [6] also proposed an IB method where the motion of the rigid object is imposed using Lagrange multipliers. The discretization of the Lagrange multipliers requires some care, but the method is quite general. Baaijens [7] used this approach for the simulation of the fluid–structure interaction between a Newtonian fluid and slender bodies. The method combines the ideas of a fictitious domain and the mortar methods (see Bernardi *et al.* [8]) by imposing continuity of the velocity field along the interface by means of Lagrange multipliers. Following the same lines, Van Loon *et al*. [9] proposed a fictitious domain method extended with a local mesh adaptation method, to provide the necessary flexibility with respect to the motion and deformation of heart valves and to ensure the ability to impose the pressure exerted by the solid on the fluid.

Ilinca *et al*. [10] present a finite element IB method where the mesh is dynamically modified to fit the solid object at each time step. Additional DOFs are introduced so that the modified mesh fits the rigid body. The additional DOFs are then eliminated through Dirichlet boundary conditions (for the velocity) and static condensation for the pressure, thanks to the introduction of a pressure discontinuity at the interface. In their paper, only static objects are considered.

Adaptive remeshing methods are now widely used for improving finite element solutions of fluid flow problems. Our mesh adaptation method (see Bois *et al*. [11, 12]) is based on the computation of an accurate finite element error estimator driving local operations on the mesh: edge division, edge swapping, node elimination, and node displacement. In the context of IB methods, estimating the error and then remeshing the whole domain at each time step would probably be prohibitive. We therefore propose a variant of the method introduced in [10] where the mesh is modified using some of the local mesh operators to fit the boundary of the rigid body. These modifications are performed only in the vicinity of the rigid object. New nodes (edge division) are introduced at the intersection of the mesh with the rigid object whose boundary is given by a level set function. This method can locally produce very poor quality elements. An essential ingredient of the method proposed in this paper is edge swapping, which is used to improve the quality of the elements in the neighborhood of the solid object. We show that this step is *absolutely* crucial if one is to compute quantities such as the drag or the lift on the rigid object. The mesh is thus modified only in the neighborhood of the rigid object and left untouched everywhere else. The velocity of the solid object is then imposed exactly through classical Dirichlet boundary conditions. Numerous examples will be presented and analyzed: problems with static and moving objects in both two-dimensional and three-dimensional flows.

2. METHODOLOGY

2.1. Governing equations

We consider the flow of an incompressible Newtonian fluid. The conservation equations of mass and momentum are written in non-dimensional form:

$$\begin{cases} \nabla \cdot \boldsymbol{u} = 0 \\ \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} \right) + \nabla p - \frac{2}{Re} \nabla \cdot (\dot{\gamma}(\boldsymbol{u})) = \boldsymbol{f} \end{cases}$$

where Re denotes the Reynolds number, $\boldsymbol{u}$ the velocity vector, p the pressure, $\dot{\gamma}(\boldsymbol{u})$ the strain-rate tensor defined as $\dot{\gamma}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right)$, and $\boldsymbol{f}$ a volumetric force vector (generally vanishing). The Reynolds number is usually defined as $Re = \frac{UL\rho}{\mu}$, where U and L are respectively the reference velocity and length, ρ is the density of the fluid, and μ its dynamic viscosity.

DOI: 10.1002/fld

2.2. Time derivative

Let $\Delta t > 0$ be the time step length and $t^n = n\Delta t, n \in N$. We denote $\boldsymbol{u}^n$ and p^n the finite element solutions at time t^n. For the discretization of the time derivative of each component of the velocity field u_i, we use an implicit backward finite difference scheme of order 2:

$$\frac{\partial u_i}{\partial t}(t^n) = \frac{\left(3u_i^n - 4u_i^{n-1} + u_i^{n-2}\right)}{2\Delta t} + \frac{(\Delta t)^2}{3}\frac{\partial^3 u_i}{\partial t^3}(\xi_i^n) \text{ with } \xi^n \in [t^{n-2}, t^n]$$

At time t^n, we thus have

$$\begin{cases} \nabla \cdot \boldsymbol{u}^n = 0 \\ \frac{\left(3\boldsymbol{u}^n - 4\boldsymbol{u}^{n-1} + \boldsymbol{u}^{n-2}\right)}{2\Delta t} + \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n - \frac{2}{Re}\nabla \cdot (\dot{\gamma}(\boldsymbol{u}^n)) + \nabla p^n = \boldsymbol{f} \end{cases}$$

2.3. Treatment of convection term

Note that the problem is still non-linear because of the convective term and it is possible to use Newton's method to linearize the problem. It is however more efficient to use a Lagrange linear extrapolation from (t^{n-2}, u_i^{n-2}) and (t^{n-1}, u_i^{n-1}) to show that

$$u_i^n = 2u_i^{n-1} - u_i^{n-2} + \frac{\partial^2 u_i}{\partial t^2}(\eta_i^n)(\Delta t)^2 \text{ with } \eta_i^n \in [t^{n-2}, t^n]$$

and then replace $\boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^n$ by $(2\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2}) \cdot \nabla \boldsymbol{u}^n$, as proposed, for instance, in Turek [13]. The problem to solve at each time step is now linear and is given by:

$$\begin{cases} \nabla \cdot \boldsymbol{u}^n = 0 \\ \frac{3}{2\Delta t}\boldsymbol{u}^n + \left(2\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2}\right) \cdot \nabla \boldsymbol{u}^n - \frac{2}{Re}\nabla \cdot (\dot{\gamma}(\boldsymbol{u}^n)) + \nabla p^n = \boldsymbol{f} + \frac{1}{2\Delta t}\left(4\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2}\right) \end{cases}$$

In our numerical experiments, we have not seen significant differences between the solutions obtained with this extrapolation technique and those obtained using a fully implicit Newton's method. When combined with the backward finite difference scheme of order 2 for the time derivative, we get a second-order scheme in time ($O(\Delta t)^2$).

2.4. Finite element formulation

The Navier–Stokes equations are discretized using a second-order ($O(h^2)$) Taylor–Hood $P_2 - P_1$ element (see Brezzi and Fortin [14]). The Galerkin finite element formulation in the computational domain Ω (possibly including rigid objects), with boundary Γ, is given by

$$\begin{cases} \displaystyle\int_\Omega \left(\frac{3}{2\Delta t}\boldsymbol{u}^n \cdot \boldsymbol{w} + \frac{2}{Re}\dot{\gamma}(\boldsymbol{u}^n) : \dot{\gamma}(\boldsymbol{w}) - p\nabla \cdot \boldsymbol{w} + \left((2\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2}) \cdot \nabla \boldsymbol{u}^n\right) \cdot \boldsymbol{w}\right) dv = \\ \displaystyle\int_\Omega \boldsymbol{f} \cdot \boldsymbol{w}\, dv + \int_{\Gamma_1} \boldsymbol{t} \cdot \boldsymbol{w}\, dS + \frac{1}{2\Delta t}\int_\Omega (4\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2}) \cdot \boldsymbol{w}\, dv \\ \displaystyle\int_\Omega q\nabla \cdot \boldsymbol{u}^n\, dv = 0 \end{cases}$$

where symbols $\boldsymbol{w}$ and q denote the appropriate weighting functions and $\boldsymbol{t}$ stands for a possible natural Neumann boundary of the form $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}$ on Γ_1, where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{n}$ the unit normal vector. The global finite element approximation is therefore second order in both space and time.

DOI: 10.1002/fld

2.5. Immersed boundary method

Let us now consider the presence of a rigid (possibly moving) object denoted Ω_r in the computational domain. We will also suppose that the boundary of Ω_r is given by a level set of a function $\phi(\boldsymbol{x}, t)$, that is,

$$\phi(x_1, x_2, x_3, t) = 0$$

This function is commonly used in level set methods (see Sethian [15]) for the computation of free boundary problems. Typically, ϕ will be negative inside the object and positive outside. For simple shape objects, ϕ can be easily constructed. For more complex geometries, the boundary of the solid can take the form of a fine surface mesh, and ϕ can be defined as the signed distance function to the IB. In the examples of Section 3, the level set function will be given explicitly.

Now, starting with a non-body-fitting mesh covering the entire domain, we determine its intersection with the rigid boundary $\partial\Omega_r$. This is done by sweeping all the edges of the mesh and determining the intersection of each edge with the rigid boundary. The intersections are easily detected by a change of sign in ϕ. A new node is inserted at the intersection where $\phi = 0$ and new elements are created as illustrated in Figure 1. If the intersection of the rigid boundary with an edge is located near an existing node, an element with at least one very small edge could be created. A criterion is introduced to avoid that situation. A new node is created near an existing one only if it is located at a distance larger than 10^{-2} times the original length of the edge under scrutiny.

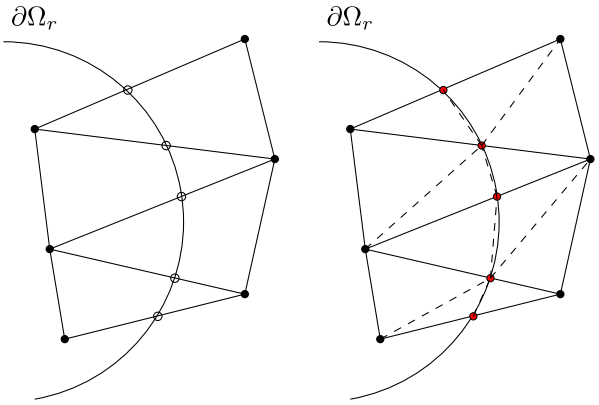
Despite this precaution, this approach inevitably creates poor-quality elements in some situations. Let us recall that for a tetrahedral element K with edge lengths $l_i, i = 1, \cdots, 6$ and with volume V_K, the quality can be defined as

$$Q_K^{3D} = \frac{12}{\sqrt{2}} \frac{V_K}{\sqrt{l_1 l_2 l_3 l_4 l_5 l_6}}.$$

It can be easily verified that this quantity is 1 for a regular tetrahedron with constant edge length. A similar definition exists for a triangle in the two-dimensional case:

$$Q_K^{2D} = \frac{27(l_1 + l_2 - l_3)(l_1 + l_3 - l_2)(l_2 + l_3 - l_1)}{(l_1 + l_2 + l_3)^3}.$$

which is obviously 1 for an equilateral triangle. Improving the quality of the elements is thus equivalent to trying to get as close as possible to equilateral triangles or regular tetrahedra. The quality of a mesh (or a submesh) is defined as the minimal quality of its elements.



(a) Intersection of the rigid object Ω_r with the mesh (b) Refinement of the mesh

Figure 1. Local refinement of the mesh.

DOI: 10.1002/fld

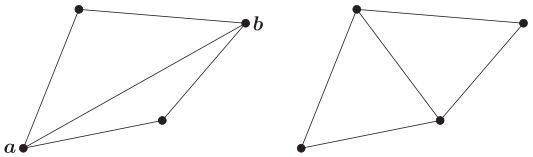


Figure 2. Edge swapping in 2D.

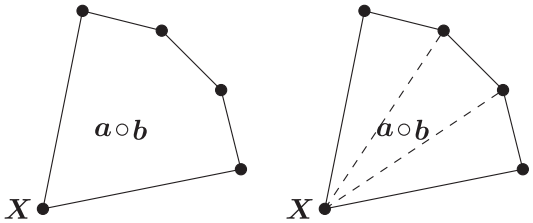


Figure 3. Edge swapping in 3D: N_e (left) and triangular mesh (right).

A fundamental aspect of our approach is edge swapping, which is illustrated in the two-dimensional case in Figure 2. For a given edge, we construct a submesh containing its two adjacent triangles. From the two illustrated submeshes in Figure 2, we choose the one with the highest quality. Note that there are obvious situations where edge swapping cannot be done.

The three-dimensional case is much more complex and is worth a few comments. For a given edge $\boldsymbol{e}$ with nodes $\boldsymbol{a}$ and $\boldsymbol{b}$, we first construct the submesh Ω_e containing all tetrahedra sharing that edge. We then consider all the nodes in that submesh except $\boldsymbol{a}$ and $\boldsymbol{b}$:

$$N_e = \{\boldsymbol{x}_i \in \Omega_e | \boldsymbol{x}_i \neq \boldsymbol{a}, \boldsymbol{b}\}$$

This is illustrated in Figure 3, where the edge $\boldsymbol{e}$ is supposed perpendicular to the plane of the figure (with nodes $\boldsymbol{a}$ and $\boldsymbol{b}$ on either side superimposed in the figure). In order to simplify the explanation, we consider in the figure the special case where all nodes in N_e lie in the same plane. This is generally not the case, but the procedure is the same. Starting from one of the nodes in N_e denoted $\boldsymbol{X}$ in the figure, a star-shaped triangular mesh is created by adding new edges between that node and all the others (dashed lines in Figure 3). In the two-dimensional case, this operation is obvious, because there are only two nodes left, and only one edge can be added (Figure 2). In three dimensions, from each of the new triangles, two tetrahedra are then constructed by connecting them respectively to $\boldsymbol{a}$ (above the plane) and $\boldsymbol{b}$ (under the plane). In the illustrated case, six tetrahedra would be created. Different meshes can be obtained depending on the choice of the initial node $\boldsymbol{X}$. All these different configurations are explored and the quality of each submesh is computed. The configuration with the highest quality is conserved. All precautions are taken to avoid the creation of invalid elements (with negative volume for example). One important criterion is that the volume of the initial and final submeshes should be the same.

Edge swapping has the advantage of improving the quality of the elements without creating new nodes. These new edges, just created by swapping, are in turn intersected with $\partial\Omega_r$, thus creating a few more nodes. This is illustrated in Figure 4 for a circular (cylindrical) rigid object. Elements with an edge on the boundary of the rigid object are illustrated in red, before and after edge swapping, clearly showing an improvement of their geometric shape

The new mesh should now fit the boundary of Ω_r. Obviously, the more refined the base mesh is, the more accurate the reconstruction of the boundary of the rigid object will be. Note that any given element is either entirely in the fluid or in the solid region. The newly added nodes do not significantly increase the computational burden because they are located on the rigid object and are therefore eliminated when imposing the Dirichlet boundary conditions (the velocity of the rigid object). As a result, the geometry of the solid is accurately represented (up to the resolution of the base mesh), and the velocity of the rigid object is imposed exactly.

DOI: 10.1002/fld

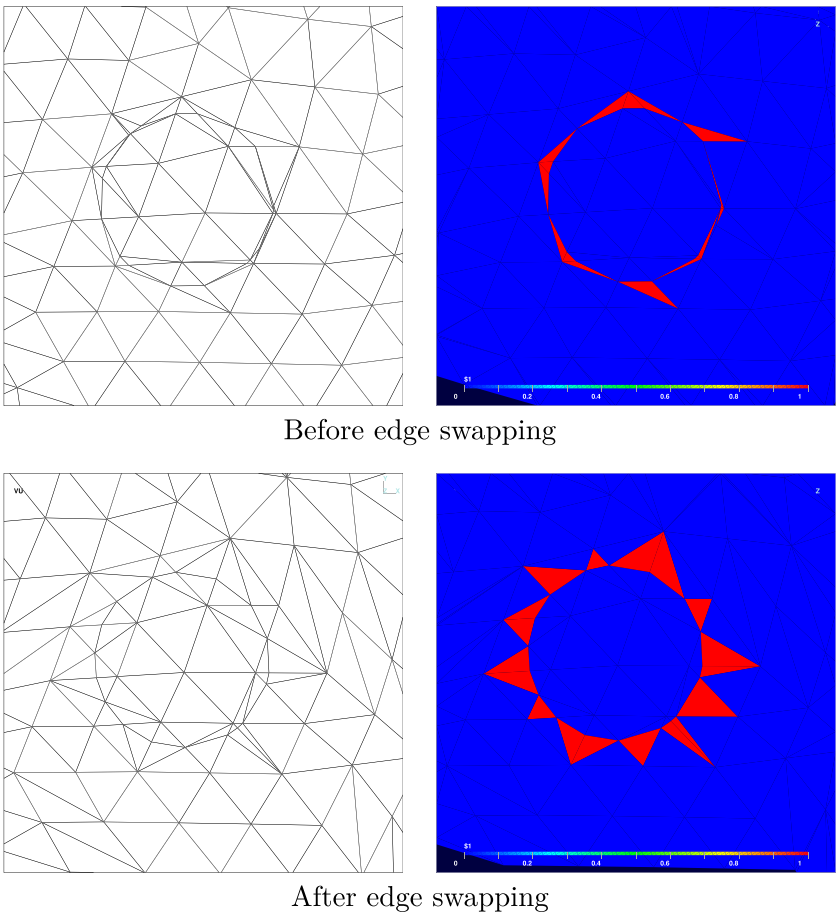


Figure 4. Elements around a circle before and after swapping edges.

The complete algorithm therefore requires a (constant) starting base mesh M_{base} and can be described as follows.

Algorithm

1. For a given time t^n:
    - 1.1 Using the known position of the rigid object at time t^n, perform local refinement of the base mesh M_{base}. We denote the new mesh M_{t^n}.
    - 1.2 Apply edge swapping and a new refinement iteration to improve M_{t^n}.
    - 1.3 Reinterpolate the velocity and pressure fields from previous time steps ($\boldsymbol{u}_{n-1}$, $\boldsymbol{u}_{n-2}$, and p_{n-1}) on the new mesh M_{t^n},
    - 1.4 Apply the appropriate boundary conditions at time t^n. In particular, Dirichlet boundary conditions are imposed inside the rigid object and on all nodes on its boundary, including the ones recently introduced.
    - 1.5 Solve the Navier–Stokes equations and compute the solution ($\boldsymbol{u}_n$, p_n)
2. Go to next time step.

Note that, at each time step, the local refinement technique (see step 1.1) is performed starting from the base mesh M_{base}. Otherwise, the mesh obtained would be more and more refined with each time step (and possibly completely distorted) throughout the region where the object moves. Obviously, if the rigid object is static, local refinement and edge swapping of the mesh are performed

DOI: 10.1002/fld

only once. Once the mesh M_{t^n} is created (step 1.3), the velocity and pressure solutions from previous time steps ($\boldsymbol{u}_{n-1}, \boldsymbol{u}_{n-2}$ and p_{n-1}) have to be reinterpolated on the new mesh. Note that most nodes are common to both meshes. Interpolation at these nodes is thus trivial. For the new nodes, interpolation is done in a classical manner. Using the coordinates of such a new node in mesh M_{t^n}, we determine which element of the old mesh contains it and then simply use the Lagrange basis functions to interpolate.

3. NUMERICAL RESULTS

We now present a number of numerical tests. We first consider static objects and the flows around a sphere and a circular cylinder in a channel. This will allow us to compare our results with those obtained on body-fitted meshes. Then, we will consider rotating and moving objects that represent the main goal of this work.

3.1. Flow around a sphere

To validate our method, we will start by studying the flow around a sphere. We will want to compute the drag and lift coefficients, which are obtained by first evaluating

$$\boldsymbol{F} = (F_1, F_2, F_3) = \int_{\partial\Omega_r} \boldsymbol{\sigma} \cdot \boldsymbol{n}\, ds \tag{1}$$

where $\boldsymbol{\sigma} \cdot \boldsymbol{n}$ is the total stress vector defined by

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \left(-p\boldsymbol{I} + \frac{2}{Re}\dot{\boldsymbol{\gamma}}(\boldsymbol{u}) \right) \cdot \boldsymbol{n}$$

The drag and lift forces use respectively the components of $\boldsymbol{F}$ parallel and orthogonal to the direction of the mean flow. The computation of these forces is difficult for classical IB methods because $\partial\Omega_r$ is not always clearly defined. In our approach, they are obtained rather easily because the modified mesh is body fitted.

The drag and lift coefficients are then defined by

$$C_D = \frac{F_1}{\frac{1}{2}\rho U^2 \pi R^2} \text{ and } C_L = \frac{F_2}{\frac{1}{2}\rho U^2 \pi R^2}$$

where U is the upstream mean velocity and R is the radius of the sphere.

Low Reynolds number flows ($Re = \frac{UD\rho}{\mu} < 1$) are governed by a balance between viscous and pressure forces. For the flow around a sphere at low values of Re, Schiller and Naumann [16] have shown that the drag coefficient can be approximated by the relationship

$$C_D = \frac{24}{Re}(1 + 0.15\, Re^{0.687}) \tag{2}$$

We applied our IB method to calculate the drag coefficient for $Re = 0.1$ and $Re = 0.5$. The diameter of the sphere, centered at $(0, 0, 0)$, was set to 1 ($D = 1$) in a computational domain extending from 10 units upstream to 25 units downstream. The level set function is thus given by $\phi(x_1, x_2, x_3, t) = x_1^2 + x_2^2 + x_3^2 - 1/4$. This is illustrated in Figure 5 (which is out of scale). The boundary conditions consist of uniform flow $\boldsymbol{u} = (1, 0, 0)$ at the inflow plane and on the lateral boundaries, no slip velocity on the sphere, and a traction-free outflow boundary condition at the exit section.

The IB method was first applied on an unstructured tetrahedral mesh having $23,965$ elements (4496 nodes, 29,387 edges, and $\simeq 106,000$ DOFs). A second mesh with $124,407$ elements was also used to verify convergence with mesh size.

DOI: 10.1002/fld

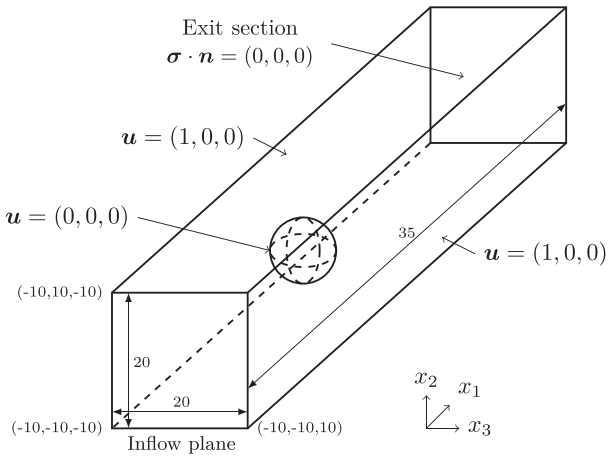


Figure 5. Flow around a sphere.

Table I. Drag coefficients on initial and (refined) meshes.

Re	Value from Equation (2)	Body-fitted method	IB
0.1	247	239 (253)	232 (250)
0.5	52.5	48.9 (52)	50 (53.1)

We first applied our IB method, starting from the coarse mesh, but without edge swapping. The computed drag value was a disappointing $C_D = 56$ instead of the 'theoretical value' 247 given by relation (2). The poor quality of the mesh in the region around the sphere made the computation very delicate. Swapping the edges gave higher quality elements around the sphere and a more accurate value $C_D = 232$ for the drag coefficient. This simple example shows that the global methodology works fine but also that the edge swapping step is necessary. Pursuing the computation on the finer mesh (with edge swapping), we obtained $C_D = 250$, a more accurate value. Similar results were obtained at $Re = 0.5$ as shown in Table I.

The same problem was also solved using a body-fitted method using two meshes having respectively $25,470$, and $128,161$ elements (numbers similar to those used for the IB method). The results are also presented in Table I and show that the computed drag values agree quite well with those given by formula 2. Refining the mesh decreases the error for both methods. This IB method can thus provide drag values very similar in precision to those obtained by a classical body-fitted mesh method.

3.2. Flow past a circular cylinder

A flow around a cylinder with circular cross section is now considered. The problem configurations and the boundary conditions are illustrated in Figure 6. A no-flow boundary condition is imposed on the cylinder and the lateral boundaries. The inflow Dirichlet boundary condition is given by

$$\boldsymbol{u}(0, x_2, x_3) = \left(\frac{72}{H^4} x_2 x_3 (H - x_2)(H - x_3), 0, 0 \right)$$

which produces a mean velocity $U = 2$. The height and width of the channel is $H = 4.1$ units, and the diameter of the cylinder is one unit ($D = 1$). The level set function is thus given by $\phi(x_1, x_2, x_3, t) = x_1^2 + x_2^2 - 1/4$. The Reynolds number is based on the diameter of the cylinder and on the mean velocity at the inlet. We consider the flow at $Re = 20$ for comparison purposes.

DOI: 10.1002/fld

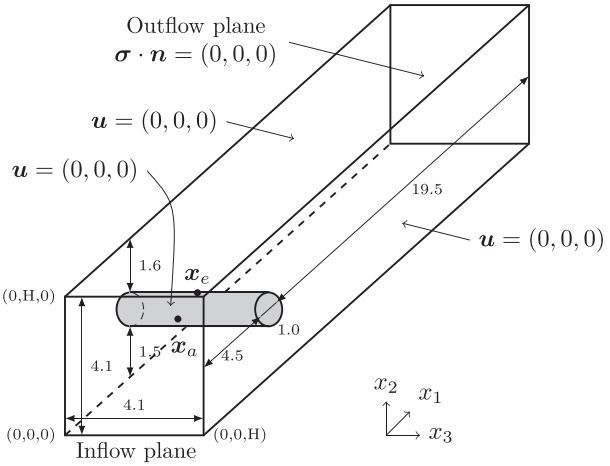


Figure 6. Flow past a circular cylinder.

Table II. Flow around a cylinder: comparison of the numerical results.

	Reference values in [17]	Body-fitted approach	IB approach
Δp	0.1694	0.1694	0.1693
C_D	6.1430	6.1124	6.0928
C_L	0.0084	0.0081	0.0079

For this problem, we compute both the drag C_D and lift C_L coefficients together with the pressure difference defined by $\Delta P = \Delta P(t) = P(\boldsymbol{x}_a, t) - P(\boldsymbol{x}_e, t)$. The points $\boldsymbol{x}_a$ and $\boldsymbol{x}_e$ are located on either side of the cylinder with coordinates $\boldsymbol{x}_a = (4.5, 2.0, 2.05)$ and $\boldsymbol{x}_e = (5.5, 2.0, 2.05)$) as illustrated in Figure 6.

Here, again, we compare the results obtained with our IB method with a standard body-fitted method and also with the numerical results of Schäfer and Turek [17]. Table II summarizes the different results. As can be seen, the IB and BF methods lead to very similar results.

3.3. Flow around a quadrifolium

We now consider moving objects. In this first example, the rotating rigid object is a quadrifolium (also known as four-leaved clover). The object can be expressed in polar coordinates (at time $t = 0$) by

$$\begin{cases} x_3^0(\theta) = (\alpha + \beta\cos(4\theta))\cos(\theta), \\ x_2^0(\theta) = (\alpha + \beta\cos(4\theta))\sin(\theta) \end{cases} \theta \in (0, 2\pi)$$

where $\alpha = \frac{1}{5}$ and $\beta = \frac{1}{10}$. In Cartesian coordinates, this corresponds to

$$\left(x_2^0\right)^2\left(x_3^0\right)^2 - \frac{5}{4}\left(\left(x_2^0\right)^2 + \left(x_3^0\right)^2\right)^2\left(0.3 - \sqrt{\left(x_2^0\right)^2 + \left(x_3^0\right)^2}\right) = 0 \tag{3}$$

which is represented in Figure 7. The width (in the x_1-direction) of the quadrifolium is set to 0.04 units.

As illustrated in Figure 8, the object is rotating clockwise around the x_1-axis with angular velocity ω. At a given time t, the boundary of the quadrifolium is given by the level set function:

$$\phi\left(x_1, x_2, x_3, t\right) = \left(x_2^0\right)^2\left(x_3^0\right)^2 - \frac{5}{4}\left(\left(x_2^0\right)^2 + \left(x_3^0\right)^2\right)^2\left(0.3 - \sqrt{\left(x_2^0\right)^2 + \left(x_3^0\right)^2}\right) = 0$$

DOI: 10.1002/fld

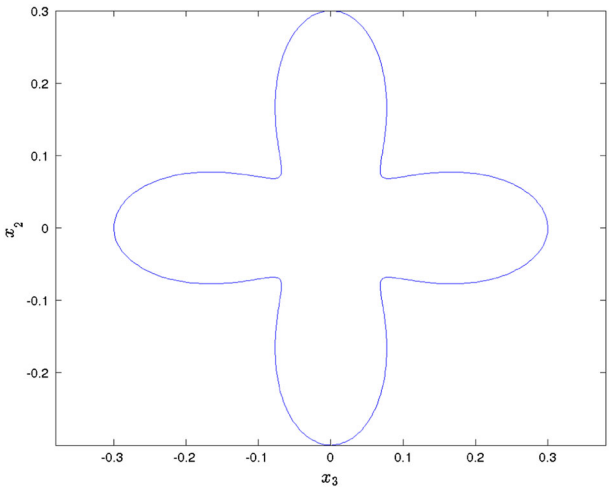


Figure 7. Quadrifolium or four-leaved clover.

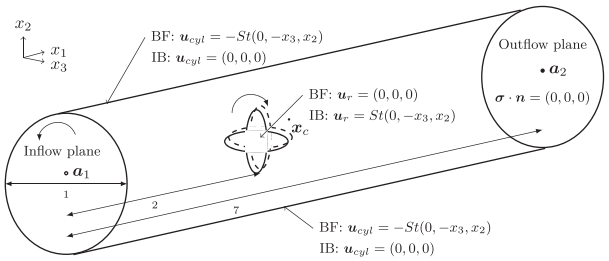


Figure 8. Laminar flow around a rotating quadrifolium.

where

$$\begin{aligned} x_3^0 &= x_3 \cos(\omega t) - x_2 \sin(\omega t) \\ x_2^0 &= x_3 \sin(\omega t) + x_2 \cos(\omega t) \end{aligned} \tag{4}$$

This last expression is a counterclockwise rotation of the point (x_3, x_2) to put it back in its original position. Equation (3) can then be used to determine the boundary of the quadrifolium. From Equation (4), one easily gets

$$\begin{aligned} x_3(t) &= x_3^0 \cos(\omega t) + x_2^0 \sin(\omega t) \\ x_2(t) &= -x_3^0 \sin(\omega t) + x_2^0 \cos(\omega t) \end{aligned}$$

which gives the position of a point on the quadrifolium at any time. By taking the derivative, we obtain the velocity that must be imposed on that point

$$\begin{aligned} x_3'(t) &= \omega(-x_3^0 \sin(\omega t) + x_2^0 \cos(\omega t)) = \omega x_2(t) \\ x_2'(t) &= \omega(-x_3^0 \cos(\omega t) - x_2^0 \sin(\omega t)) = -\omega x_3(t) \end{aligned}$$

This problem can be seen as a very crude approximation of a propeller and is presented to illustrate that the method also performs well in the case of moving objects. This 'propeller' is rotating inside a cylinder of diameter $D = 1$. A dimensional analysis shows that there are two dimensionless groups: the Reynolds number $Re = \frac{\rho U D}{\mu}$ and the Strouhal number $St = \frac{\omega U}{D}$, both defined using the mean inflow velocity U, the diameter D of the cylinder, and the angular velocity ω of the moving object. The Strouhal number represents the ratio of inertial forces due to the unsteadiness of the flow to the inertial forces due to changes in velocity from point to point in the flow field.

DOI: 10.1002/fld

Although the solid is rotating, this problem can still be solved using both a body-fitted mesh and the IB method if proper systems of coordinates are chosen. The geometry and boundary conditions for both methods are illustrated in Figure 8.

Immersed boundary (IB) method

In this case, we use a static system of coordinates. The inflow velocity is given by

$$\boldsymbol{u}_{inflow} = \left(\frac{2}{R^2} \left(R^2 - x_2^2 - x_3^2 \right),\ 0,\ 0 \right) \tag{5}$$

where R is the radius of the cylinder ($R = 0.5$), resulting in a mean velocity $U = 1$ and $St = \omega$. A no-slip condition is imposed on the cylinder ($\boldsymbol{u}_{cyl} = (0, 0, 0)$), and the rigid object is rotating clockwise with angular velocity St:

$$\boldsymbol{u}_r = St(0, -x_3, x_2)$$

Body-fitted (BF) method

In this case, we consider a system of coordinates rotating with the quadrifolium. The rigid object is therefore not moving ($\boldsymbol{u}_r = (0, 0, 0)$), but the cylinder is rotating counterclockwise:

$$\boldsymbol{u}_{cyl} = -St(0, -x_3, x_2)$$

Obviously, these two formulations are equivalent and one can show that

$$\boldsymbol{u}_{IB} = \boldsymbol{u}_{BF} + St(0, -x_3, x_2) \tag{6}$$

and that the velocity on the inflow section is

$$\boldsymbol{u}_{inflow} = \left(\frac{2}{R^2} \left(R^2 - x_2^2 - x_3^2 \right),\ 0,\ 0 \right) - St(0, -x_3, x_2)$$

Note that because we are in a rotating system of coordinates, Coriolis force terms have to be added to the momentum equations:

$$\begin{cases} \frac{3}{2\Delta t}\boldsymbol{u}^n + \left(\left(2\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2} \right) \cdot \nabla \boldsymbol{u}^n \right) - \frac{2}{Re} \nabla \cdot \left(\dot{\gamma}(\boldsymbol{u}^n) \right) + \nabla p^n + 2\left(\boldsymbol{w} \times \boldsymbol{u}^n \right) \\ \qquad = \boldsymbol{f} + \frac{1}{2\Delta t} \left(4\boldsymbol{u}^{n-1} - \boldsymbol{u}^{n-2} \right) - \boldsymbol{w} \times \left(\boldsymbol{w} \times \boldsymbol{r} \right) \end{cases}$$

where $\boldsymbol{w} = St(1, 0, 0)$ is the axis of rotation vector and $\boldsymbol{r} = (x_1, x_2, x_3)$ is the position vector.

Computations were carried out for $Re = 50$ and $St = 10$ for both IB and BF methods. The IB method was applied on an unstructured tetrahedral mesh having 115,372 elements (20,038 nodes and 136,629 edges). The BF method was also applied on an unstructured tetrahedral mesh having 104,001 elements (18,607 nodes and 125,033 edges). Cross sections of the two meshes on a plane

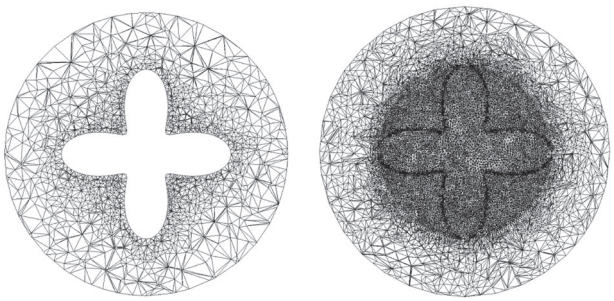


Figure 9. Meshes for quadrifolium for body fitted (left) and immersed boundary (right) methods.

DOI: 10.1002/fld

passing through the quadrifolium are presented in Figure 9. Note that these are not regular triangular meshes but merely the intersections of three-dimensional meshes with the plane. The mesh for the IB method is illustrated after edge division and is slightly more refined in the region where we need to rotate the quadrifolium. Note also that all nodes inside the quadrifolium will be eliminated from the linear systems, thus reducing the computational burden.

The time step was set to $\pi/180$, and consequently, a complete rotation is performed every 36 time steps. For the IB method, this means that 36 meshes have to be constructed from the base mesh and then be reused for the subsequent time steps. After transient effects have disappeared, we obtain the longitudinal velocity u_1 plotted along the x_1-axis in Figure 10. We also present, in Figure 11, the comparison between the pressure along the x_1-axis for both IB and BF methods. It is also possible to compare the transverse velocities (u_2) using relation (6) but not on the x_1-axis because they vanish. Figure 12 shows the comparison between the transverse velocities plotted from $\boldsymbol{a}_1 = (-2, 0.2, 0)$ to $\boldsymbol{a}_2 = (5, 0.2, 0)$. In all cases, the agreement is very good. The numerical solutions obtained with the two methods are similar. This is also the case for the computed drag and lift coefficients, which are given in Table III.

3.4. Flow around an oscillating airfoil

We now investigate a two-dimensional motion of an oscillating NACA0015 profile. The geometry of the problem is presented in Figure 13, and the motion of the National Advisory Committee for Aeronautics (NACA) profile is described in Figure 14. The position of the point $\boldsymbol{x}^c(t) = (x_1^c(t), x_2^c(t))$ in Figure 13 and the angle $\theta(t)$ of the profile are given by

$$\begin{cases} \theta(t) = \theta_0 \sin(\omega t) \\ h(t) = h_0 \sin(\omega t) \\ \\ x_1^c(t) = 0 \\ x_2^c(t) = h(t) \end{cases}$$

The expression of the NACA profile at time $t = 0$ is

$$(x_2^0)^2 = \left(\frac{0.15c}{0.2}\right)^2 \left[0.2969\sqrt{\frac{x_1^0 - 1/3}{c}} - 0.1260\left(\frac{x_1^0 - 1/3}{c}\right) - 0.3537\left(\frac{x_1^0 - 1/3}{c}\right)^2 + 0.2843\left(\frac{x_1^0 - 1/3}{c}\right)^3 - 0.1015\left(\frac{x_1^0 - 1/3}{c}\right)^4\right]^2 \tag{7}$$

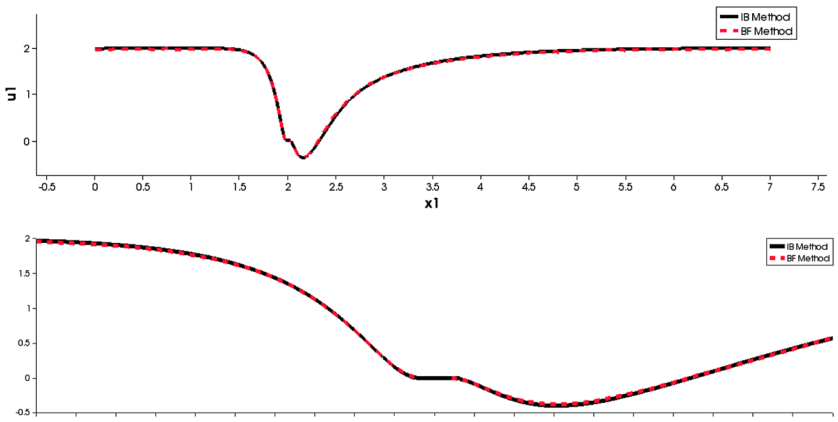


Figure 10. Longitudinal velocity along x_1-axis: comparison between BF and immersed boundary (IB) methods. Close view in the interval [1.5, 2.5] near the quadrifolium.

DOI: 10.1002/fld

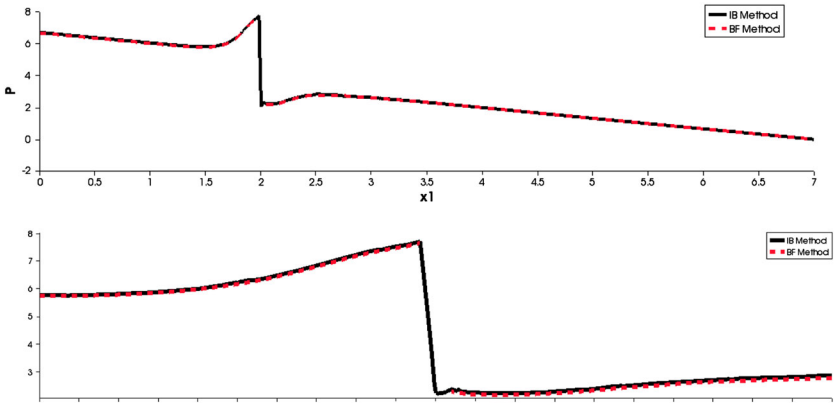


Figure 11. Pressure along x_1-axis: comparison between BF and immersed boundary (IB) methods. Close view in the interval [1.5, 2.5] near the quadrifolium.

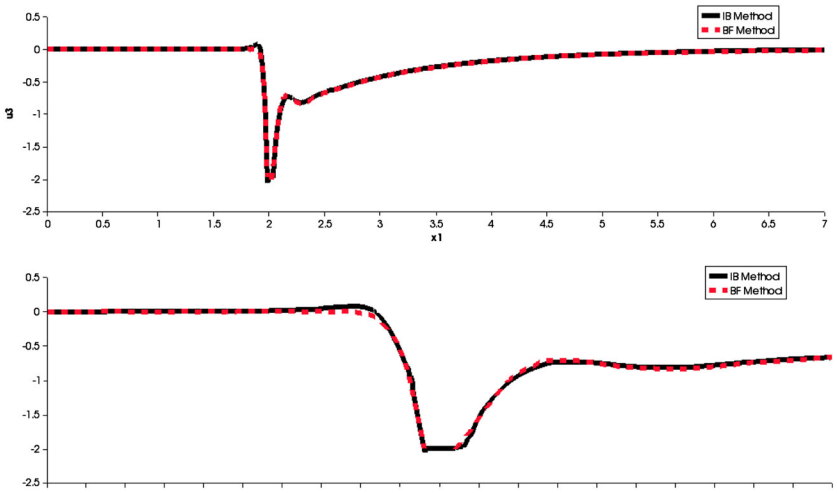


Figure 12. Transversal velocity from $\boldsymbol{a}_1$ to $\boldsymbol{a}_2$: comparison between BF and immersed boundary (IB) methods. Close view in the interval [1.5, 2.5] near the quadrifolium.

Table III. Numerical results and comparison.

	BF approach	IB approach
C_D	15.0592	14.9523
C_L	4.4326	4.2954

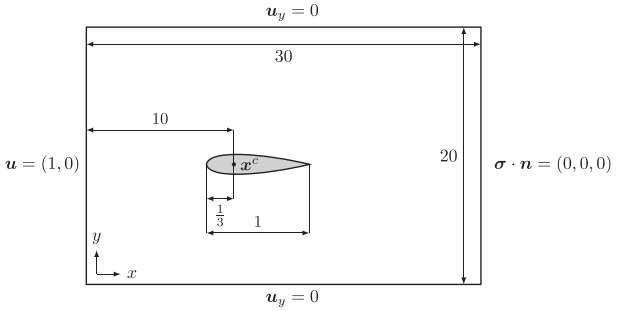


Figure 13. Domain definition for the modelization of flow around an oscillating airfoil.

DOI: 10.1002/fld

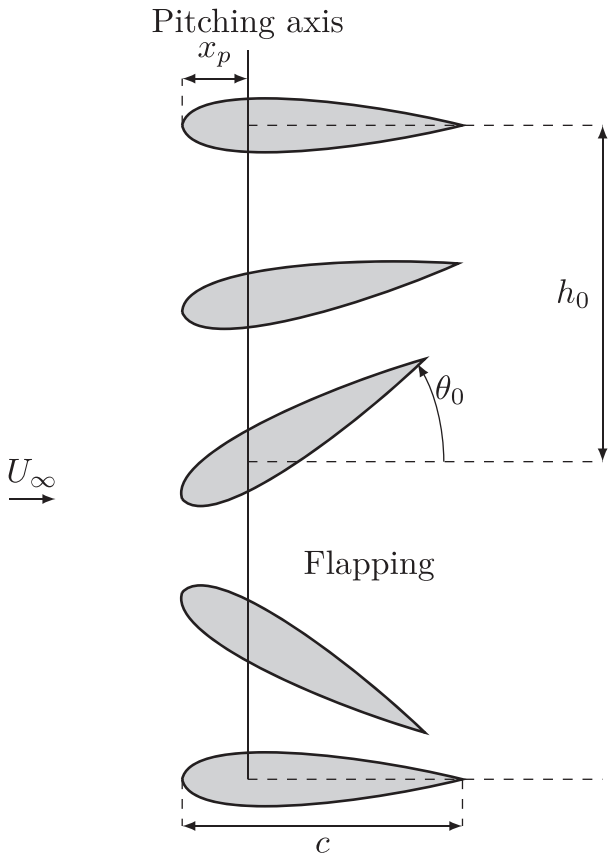


Figure 14. Movement of the airfoil.

which is represented in Figure 13. The movement is composed of a rotation with angle $\theta(t)$ followed by a translation of amplitude $(x_1^c(t), x_2^c(t))$. Thus, at a given time t, its boundary is given by the level set function:

$$\begin{aligned}\phi(x_1, x_2, t) \; = (x_2^0)^2 - \left(\frac{0.15c}{0.2}\right)^2 \Bigg[& 0.2969\sqrt{\frac{x_1^0 - 1/3}{c}} - 0.1260\left(\frac{x_1^0 - 1/3}{c}\right) \\ & - 0.3537\left(\frac{x_1^0 - 1/3}{c}\right)^2 + 0.2843\left(\frac{x_1^0 - 1/3}{c}\right)^3 \\ & -0.1015\left(\frac{x_1^0 - 1/3}{c}\right)^4 \Bigg]^2 = 0\end{aligned}$$

where, in order to return to the initial position, we remove the translation and rotate with an angle $-\theta$.

$$\begin{aligned} x_1^0 &= \big(x_1 - x_1^c(t)\big)\cos(\theta(t)) + \big(x_2 - x_2^c(t)\big)\sin(\theta(t)) \\ x_2^0 &= -\big(x_1 - x_1^c(t)\big)\sin(\theta(t)) + \big(x_2 - x_2^c(t)\big)\cos(\theta(t)) \end{aligned} \tag{8}$$

Inverting this last expression, we get

$$\begin{aligned} x_1(t) &= x_1^0\cos(\theta(t)) - x_2^0\sin(\theta(t)) + x_1^c(t) \\ x_2(t) &= x_1^0\sin(\theta(t)) + x_2^0\cos(\theta(t)) + x_2^c(t) \end{aligned}$$

DOI: 10.1002/fld

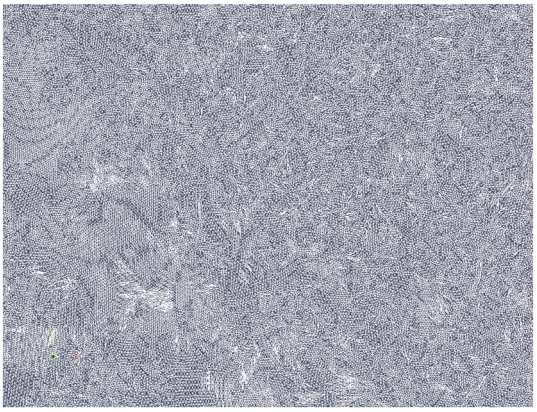


Figure 15. Initial grid for flow around an oscillating airfoil.

which gives the position of a point on the NACA profile at any time. Taking the derivative gives the velocity that must be imposed on that point:

$$x_1'(t) = \theta'(t)\left(-x_1^0\sin(\theta(t)) - x_2^0\cos(\theta(t))\right) + \left(x_1^c\right)'(t) = -\theta'(t)\left(x_2 - x_2^c\right) + \left(x_1^c\right)'(t)$$
$$x_2'(t) = \theta'(t)\left(x_1^0\cos(\theta(t)) - x_2^0\sin(\theta(t))\right) + \left(x_2^c\right)'(t) = \theta'(t)\left(x_1 - x_1^c\right) + \left(x_2^c\right)'(t)$$

As in Cori [18], we will consider the case where the chord length c is one unit, the heaving amplitude h is also one unit ($h_0 = 1$), the pitching angle $\theta_0 = \pi/3$, and the oscillating frequency is 0.18Hz. Consequently, we have $\omega = 2\pi \times 0.18 = 1.13$ and a corresponding period $T = 5.55$s. We consider the flow at $Re = 1100$ where $Re = \frac{\rho U_\infty c}{\mu}$. This is a difficult problem because the computational domain is very large with respect to the actual dimension of the airfoil. The initial (non-body-fitted) mesh is presented in Figure 15 and contains 86,580 nodes, a number favorably comparable with the meshes used in [18] (120,000 nodes) and [19] (more than 200,000 nodes). The time step was set to 0.05, and it takes 111 time steps to cover a complete oscillation of the airfoil. Consequently, 111 meshes were created for one cycle and used repeatedly. Starting from a null velocity field, transient effects disappeared after two complete cycles, and a fully established flow was obtained.

Here, again, special attention is given to the computation of the lift coefficient

$$C_L = \frac{F_2}{\frac{1}{2}\rho U_\infty^2 c}$$

which is a critical quantity in this type of engineering applications. The lift coefficient calculated over a complete period is presented in Figure 16 and compared with the results from [18]. The agreement is very satisfactory.

Finally, we present in Figure 17 isolines of the u_1-velocity component behind the airfoil at different times across one period T. A von Kármán vortex street can be easily seen.

3.5. Flow around two counter-rotating quadrifolia

In our last validation test, we present an example illustrating the flexibility of our approach. This problem cannot be solved using a standard body-fitted method. We consider a geometry similar to the one in Section 3.3, but we add a second counter-rotating quadrifolium with a different angular velocity. The computational domain and boundary conditions for this problem are shown in Figure 18. Computations were carried out for $Re = 50$ and for two different angular velocities : $\omega = 1$Hz for the first quadrifolium and $\omega = 10$Hz for the second one. The time step was set again to $\pi/180$.

DOI: 10.1002/fld

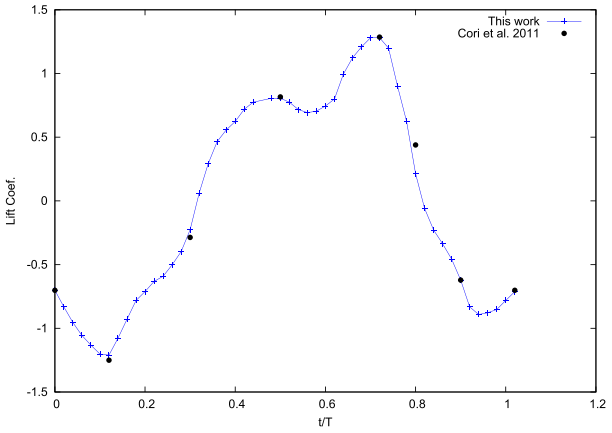
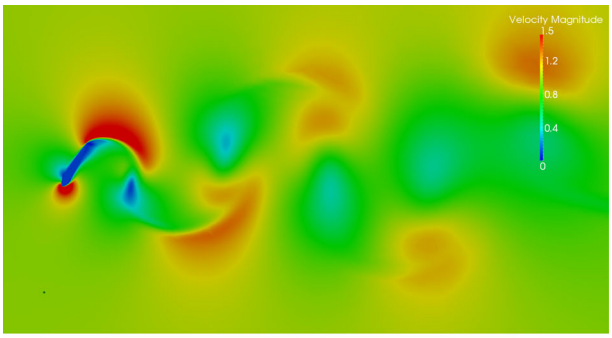
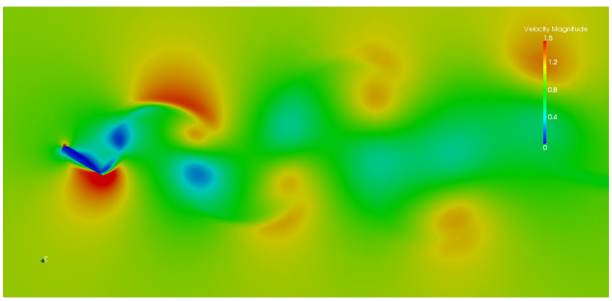


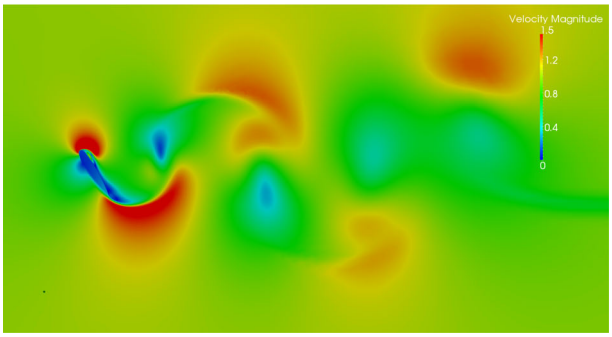
Figure 16. Evolution of the lift coefficient C_L over one period.



$t = T/2$



$t = 3T/4$



$t = T$

Figure 17. Isolines of the velocity component u_1.

DOI: 10.1002/fld

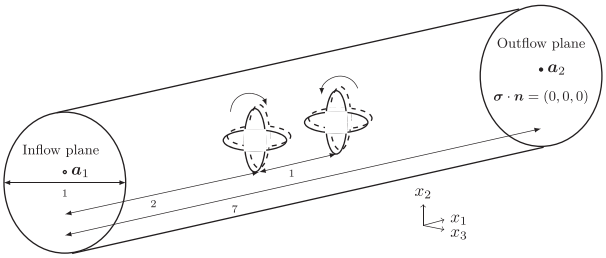


Figure 18. Laminar flow around a two counter-rotating quadrifolium.

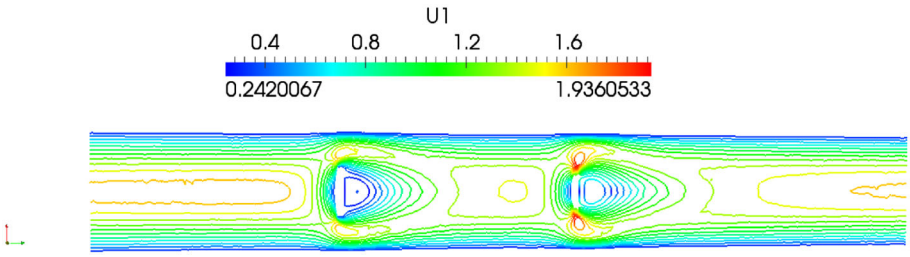


Figure 19. Isolines of u_1 in the plane $x_2 = 0$.

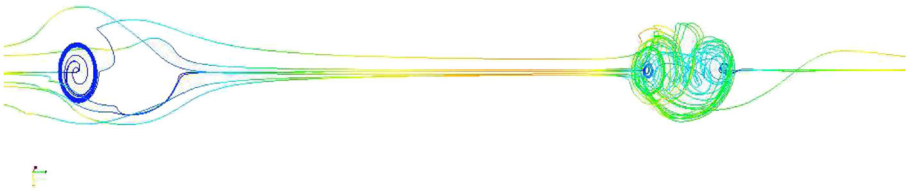


Figure 20. Streamlines for the flow with two quadrifolia.

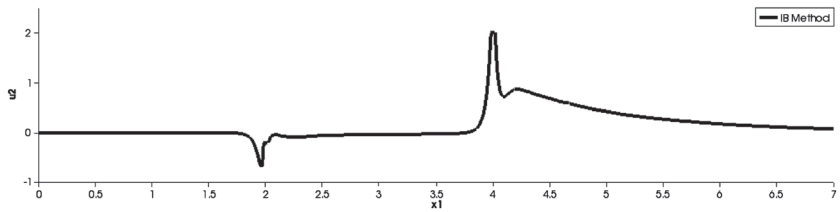


Figure 21. Transversal velocities from $\boldsymbol{a}_1$ to $\boldsymbol{a}_2$ for the flow with two quadrifolia.

A cross section of the first component of the velocity field on the plane $x_2 = 0$ is presented in Figure 19 showing the presence of two recirculation zones behind the rotating objects. Figure 20 shows selected streamlines in the cylinder. Figure 21 presents the transversal velocity u_2 between the same points $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ used in Section 3.3. The transversal velocities induced by the rotating objects are clearly seen with a factor of roughly 10 between the two quadrifolia.

4. CONCLUSIONS

We have presented an IB method where a base mesh is locally modified in order to fit the boundary of the rigid object. The finer the base mesh is, the more accurate the reconstruction of the rigid object is. Edge swapping is used to enhance the quality of the elements and to improve the accuracy of quantities such as drag and lift coefficients. The results show that the method works remarkably well in various situations, including static and moving rigid objects.

DOI: 10.1002/fld

ACKNOWLEDGEMENT

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for its financial support.

REFERENCES

1. Peskin CS. The immersed boundary method. *ACTA Numerica* 2002; **11**:479–517.
2. Su SW, Lai MC, Lin CA. An immersed boundary technique for simulating complex flows with rigid boundary. *Computers and Fluids* 2007; **36**:313–324.
3. Noor DZ, Chern MJ, Horng TL. An immersed boundary method to solve fluid-solid interaction problems. *Computational Mechanics* 2009; **44**:447–453.
4. Huang WX, Sung HJ. An immersed boundary method for fluid-flexible structure interaction. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**:2650–2661.
5. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; **37**:239–261.
6. Glowinski R, Pan TW, Périaux J. A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1994; **112**:133–148.
7. Baaijens FPT. A fictitious domain/mortar element method for fluid-structure interaction. *International Journal for Numerical Methods in Fluids* 2001; **35**:743–761.
8. Bernardi C, Maday Y, Patera AT. A new nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear Partial Differential Equations and Their Applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991)*, Vol. 299 of *Pitman Research Notes in Mathematics Series*. Longman Sci. Tech.: Harlow, 1994; 13–51.
9. Van Loon P, Anderson PD, Van de Vosse FN. A fluid-structure interaction method with solid-rigid contact for heart valve dynamics. *Journal of Computational Physics* 2006; **217**:806–823.
10. Ilinca F, Hétu JF. A finite element immersed boundary method for fluid flow around rigid objects. *International Journal for Numerical Methods in Fluids* 2011; **65**(7):856–875.
11. Bois R, Fortin M, Fortin A. A fully optimal anisotropic mesh adaptation method based on a hierarchical error estimator. *Computer Methods in Applied Mechanics and Engineering* February 2012; **209-212**:12–27.
12. Bois R, Fortin M, Fortin A, Couët A. High order optimal anisotropic mesh adaptation using hierarchical elements. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 2012; **21**(1-2):72–91.
13. Turek S. *Efficient Solvers for Incompressible Flow Problems*, Vol. 6 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag: Berlin, 1999.
14. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*, Vol. 15 of *Springer Series in Computational Mathematics*. Springer-Verlag: New York, 1991.
15. Sethian JA. *Level Sets Methods and Fast Marching Methods.* Number 3 in Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press: Cambridge, 1996.
16. Schiller L, Naumann Z. A drag coefficient correlation. *Zeitschrift des Vereins Deutscher Ingenieure* 1935; **77**: 318–320.
17. Schäfer M, Turek S. Benchmark computations of laminar flow around a cylinder. In *Flow Simulation with High-Performance Computers II*, Vol. 52 of *Notes on Numerical Fluid Mechanics*, Hirschel E (ed.). Vieweg: Weisbaden, 1996; 547–566.
18. Cori JF. Analyse et caractérisation d'interactions fluide-structure instationnaires en grands déplacements. *Ph.D. Thesis*, Département de génie mécanique, École Polytechnique de Montréal, 2011.
19. Kinsey T, Dumas G. Parametric study of an oscillating airfoil in a power-extraction regime. *AIAA Journal* 2008; **46**(6):1318–1330.

DOI: 10.1002/fld