NRC Publications Archive Archives des publications du CNRC

A finite element immersed boundary method for fluid flow around moving objects

Ilinca, F.; Héту, J.-F.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1016/j.compfluid.2010.06.002>

Computers&Fluids, 39, 9, pp. 1656-1671, 2010-10-01

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=1dc42646-7caa-4ddd-8c2d-44dda35f48d7>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=1dc42646-7caa-4ddd-8c2d-44dda35f48d7>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.

A finite element immersed boundary method for fluid flow around moving objects

F. Ilinca and J.-F. Héту

National Research Council, 75 de Mortagne, Boucherville, Qc, Canada, J4B 6Y4

Abstract

This paper presents the extension of a recently proposed immersed boundary method to the solution of the flow around moving objects. Solving the flow around objects with complex shapes may involve extensive meshing work that has to be repeated each time a change in the geometry is needed. Mesh generation and solution interpolation between successive grids may be costly and introduce errors if the geometry changes significantly during the course of the computation. These drawbacks are avoided when the solution algorithm can tackle grids that do not fit the shape of immersed objects. This work presents an extension of our recently developed finite element Immersed Boundary (IB) method to transient applications involving the movement of immersed fluid/solid interfaces. As for the fixed solid boundary case, the method produces solutions of the flow satisfying accurately boundary conditions imposed on the surface of immersed bodies. The proposed algorithm enriches the finite element discretization of interface elements with additional degrees of freedom, the latter being eliminated at element level. The boundary of immersed objects is defined using a time dependent level-set function. Solutions are shown for various flow problems and the accuracy of the present approach is measured with respect to solutions on body-conforming meshes.

Keywords: Immersed boundary method, Finite elements, Moving interface, Non body-conforming mesh, Pressure enrichment

1. INTRODUCTION

Numerical solution of differential equations of fluid flow proceeds in two steps: temporal and spatial discretization. While time discretization is fairly simple, space discretization is much more challenging as it must account

for the complexity of the geometry (body-conforming grids and boundary conditions) and that of flow physics (upwinding, stabilization). While most industrial solvers use body conforming meshes, there is increasing interest for algorithms using non conforming grids. They are known under the names immersed boundary, immersed interface, embedded mesh and fictitious domain methods. For simplicity, we will use in the present work the immersed boundary (IB) term to identify a non body-conforming method. These techniques all rely on using simple grids on domains of simple shapes with the discretization being performed on a single domain enclosing both fluid and solid regions. It is understood that solid surfaces will intersect the mesh between grid points and that they will not be formally represented by the grid (i.e. by no-slip surfaces at the fluid-solid interface). These methods trade the geometrical complexity of the domain of interest for the simplicity of a regular parallelepiped with a relative uniform mesh. This necessitates abandoning a simple implementation of boundary conditions for a more complex one.

The original IB method was introduced by Peskin [1] and uses source terms distributed through the solid region to indirectly enforce the proper boundary conditions on the solid-fluid interface. IB methods received particular attention in recent years. Their applications cover a broad spectrum of fluid dynamics problems from flow through heart devices [1, 2], to the interaction of body movement and fluid flow [3], modeling anguilliform swimming [4] and 3-D parachute simulation [5]. Reviews of IB methods may be found in [3] and [6]. Most of these developments were made in a finite difference framework. IB methods using finite elements were developed more recently [7]. Zhang *et al.* [8] present an immersed finite element method using a Lagrangian solid mesh that moves on top of a background Eulerian fluid mesh. The background grid spans over the entire computational domain and a discretized delta function is used to describe the coupling between fluid and solid domains. Glowinski *et al.* [9] used a Lagrange multiplier-based fictitious domain method to obtain finite element solutions around moving rigid bodies. The method consists of filling a moving body by the surrounding fluid and then to impose rigid body motions to the fluid occupying the regions originally occupied by the rigid body. The rigid body motion constraint is then relaxed using distributed Lagrange multipliers and solving a flow problem over the entire domain. The finite element solution of the flow around immersed objects using a fixed mesh ALE approach is presented by Codina *et al.* [10]. In their work the solid boundary is represented using

a level-set function defined on a background fixed mesh, while the moving boundary is treated by an ALE technique in the region close to the immersed surface. Boundary conditions on the immersed boundary are imposed by a least-squares approximation.

In most IB methods, boundary conditions on immersed surfaces are handled either accurately by using dynamic data structures to add/remove grid points as needed, or in an approximate way by imposing the boundary conditions to the grid point closest to the surface or through least-squares. Our recently proposed approach [11] demonstrated on fixed immersed objects achieves the level of accuracy of cut cell dynamic node addition techniques with none of their drawbacks (increased CPU time and costly dynamic data structures). This approach is extended to moving fluid/solid interfaces in the present contribution.

In this work the flow around immersed bodies is solved using a 3-D finite element IB method. The solution is interpolated using linear elements and time integration is done by an implicit Euler scheme. The immersed boundary is represented using a time dependent level-set function using the same linear interpolation functions that are used to solve the flow problem. The proposed approach is verified on simple cases for which solutions on body-conforming grids exist and is then applied to more complex moving boundary flow problems.

The paper is organized as follows. First the model problem and the associated finite element formulation is presented. The IB formulation is discussed in section 3 and the procedure to impose the boundary conditions on moving solid surfaces is detailed. Section 4 illustrates the performance of the present IB method for a selection of 2-D and 3-D test problems. The paper ends with conclusions.

2. THE MODEL PROBLEM

We consider the incompressible fluid flow problem on a bounded computational domain Ω formed by the fluid region $\Omega_f(t)$ and the solid volume $\Omega_s(t)$ as shown in Figure 1. The fluid and solid volumes are time dependent but the total volume Ω formed by their reunion is not. The immersed interface $\Gamma_i(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$ represents a boundary for the fluid flow and is considered as being time dependent.

Figure 1: Computational domain formed by fluid region $\Omega_f(t)$ and solid region $\Omega_s(t)$.

2.1. Model equations and boundary conditions

The equations of motion are the incompressible Navier-Stokes equations:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{a} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right] + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

where ρ is the density, $\mathbf{u}$ the velocity vector, $\mathbf{a}$ the convection velocity vector, p the pressure, μ the viscosity, and $\mathbf{f}$ a volumetric force vector. The convection velocity is introduced to extend the formulation to the case of deforming meshes which will be used to validate the IB solution. For a fixed, non-deforming grid, the convection velocity is the same as the fluid velocity, whereas in the case of a deforming mesh the convection velocity is the relative velocity of the fluid with respect to that of the mesh:

$$\mathbf{a} = \mathbf{u} - \mathbf{u}_m \quad (3)$$

The interface Γ_i , between the fluid and solid regions, is specified using a level-set function ψ , which is defined as a signed distance function from the immersed interface:

$$\psi(\mathbf{x}, t) = \begin{cases} d(\mathbf{x}, \mathbf{x}_i(t)), & \mathbf{x} \text{ in the fluid region,} \\ 0, & \mathbf{x} \text{ on the fluid/solid interface,} \\ -d(\mathbf{x}, \mathbf{x}_i(t)), & \mathbf{x} \text{ in the solid region,} \end{cases} \quad (4)$$

where $d(\mathbf{x}, \mathbf{x}_i(t))$ is the distance between the point $P(\mathbf{x})$ and the fluid/solid interface $P_i(\mathbf{x}_i(t))$ at time t . Hence, points in the fluid region have positive values of ψ , whereas points in the solid region have negative ones. The

definition of the level-set function may be more complicated for complex 3-D geometries. In such a case, we may consider that the immersed boundary surface is provided in the form of a CAD file from which we generate a surface mesh with a mesh size sufficiently small to have a correct representation of the surface. Then, the level-set function is simply computed from the shortest distance between the nodes of the 3D mesh and those of the surface mesh. In this work only cases for which the level-set function is given analytically were considered.

The initial and boundary conditions associated to the momentum-continuity equations are

$$\mathbf{u} = \mathbf{U}_0(\mathbf{x}), \quad \text{for } t = t_0, \quad (5)$$

$$\mathbf{u} = \mathbf{U}_D(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma_D(t), \quad (6)$$

$$\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \hat{\mathbf{n}} - p \hat{\mathbf{n}} = \mathbf{t}(\mathbf{x}, t), \quad \text{for } \mathbf{x} \in \Gamma_t(t), \quad (7)$$

where Γ_D is the portion of the fluid boundary $\partial\Omega_f$ where Dirichlet conditions are imposed, and $\mathbf{t}$ is the traction imposed on the remaining fluid boundary $\Gamma_t = \partial\Omega_f \setminus \Gamma_D$. Dirichlet boundary conditions are imposed at the interface between fluid and solid regions, i.e. $\Gamma_i \subset \Gamma_D$. Because Γ_i is not represented by the finite element discretization, a special procedure is used to enforce velocity boundary conditions on this surface. This approach will be described in details in Section 3.

2.2. The Finite Element Formulation

The finite element formulation is the same as for the static IB case found in Ref. [11]. Time derivatives are computed using an implicit Euler scheme. Both velocity and pressure are discretized using linear continuous interpolants and the weak form of the equations corresponds to the GLS (Galerkin Least-Squares) method:

$$\begin{aligned} & \int_{\Omega} \rho \left(\frac{\mathbf{u} - \mathbf{u}_0}{\Delta t} + \mathbf{a} \cdot \nabla \mathbf{u} \right) N_i^u d\Omega - \int_{\Omega} p \nabla N_i^u d\Omega \\ & + \int_{\Omega} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \nabla N_i^u d\Omega - \int_{\Omega} \mathbf{f} N_i^u d\Omega \\ & + \sum_K \int_{\Omega_K} \left\{ \rho \left(\frac{\mathbf{u} - \mathbf{u}_0}{\Delta t} + \mathbf{a} \cdot \nabla \mathbf{u} \right) + \nabla p \right. \\ & \quad \left. - \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] - \mathbf{f} \right\} \tau \mathbf{u} \cdot \nabla N_i^u d\Omega_K = \int_{\Gamma_t} \mathbf{t} N_i^u d\Gamma, \quad (8) \end{aligned}$$

$$\int_{\Omega} \nabla \cdot \mathbf{u} N_i^p d\Omega + \sum_K \int_{\Omega_K} \left\{ \rho \left(\frac{\mathbf{u} - \mathbf{u}_0}{\Delta t} + \mathbf{a} \cdot \nabla \mathbf{u} \right) + \nabla p \right. \\ \left. - \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] - \mathbf{f} \right\} \tau \nabla N_i^p d\Omega_K = 0, \quad (9)$$

where $(\mathbf{u}, p)$ is the solution at the current time step t_n , $\mathbf{u}_0$ is the solution at the previous time step t_{n-1} and $\Delta t = t_n - t_{n-1}$ is the time step increment. N_i^u, N_i^p are continuous, piecewise linear test functions associated to the velocity and pressure equations. The first four integrals in the right hand side of equation (8) and the first integral in equation (9) correspond to the Galerkin formulation whereas the integrals over the elements interior are the GLS stabilization terms. The stabilization parameter τ is computed as from Refs. [12, 13]:

$$\tau = \left[\left(\frac{2\rho|\mathbf{a}|}{h_K} \right)^2 + \left(\frac{4\mu}{m_k h_K^2} \right)^2 \right]^{-1/2} \quad (10)$$

Here h_K is the size of the element K and m_k is a coefficient set to 1/3 for linear elements (see [12, 14]).

The nonlinear equations for the velocity and pressure, are solved with a few Picard steps followed by Newton-Raphson iterations. The resulting linear systems are generated directly in a compressed sparse row format [15], and solved using the bi-conjugate gradient stabilized (Bi-CGSTAB) iterative method [16] with an ILU preconditioner. An important reason for using the GLS formulation is that it also stabilizes the linear systems, making them tractable by iterative solvers [17]. This finite element algorithm was successfully used by the authors to solve casting applications [18], the filling and post-filling phases of the injection molding process [19], powder injection molding [20], gas-assisted injection molding [21] and co-injection [22].

3. THE IB METHOD

The algorithm used to treat the immersed boundary surface is the same as for the static IB method [11]. However, the change in position of the fluid/solid interface from one time step to another commands a special integration of time derivative terms.

In this section we use the nomenclature introduced in [11] and illustrated in Figure 2. The mesh is intersected by the interface $\Gamma_i(t_n)$ of the current time step t_n at points located along element edges and we consider those

Figure 2: Decomposition of elements cut by the immersed boundary.

points as additional degrees of freedom in the finite element formulation. Mesh nodes are separated depending on their relative location with respect to the immersed boundary as follows:

- nodes in the fluid region (N_f) - filled circles in Figure 2;
- additional nodes on the interface (N_i) - filled squares in Figure 2;
- nodes in the solid region, but connected with nodes in the fluid region (N_{sf}) - open circles in Figure 2;
- nodes in the solid region which are not connected with nodes in the fluid region (N_s) - filled diamonds in Figure 2.

Mesh elements are also separated depending on their relative location with respect to the immersed boundary:

- elements in the fluid region (E_f);
- elements cut by the immersed boundary (E_i). These elements have nodes in both fluid and solid regions. However, the addition of nodes on the interface and the decomposition of elements cut by Γ_i yields to the formation of elements which are either entirely in the fluid region (E_{if}) or in the solid region (E_{is});
- elements in the solid region (E_s).

3.1. Imposition of boundary conditions at the interface

We consider that the solid embedded in the mesh has a prescribed velocity $\mathbf{u}_s(\mathbf{t})$. Velocity boundary conditions are imposed as follows:

- the velocity at nodes (Nf) is an unknown of the global system of equations (no boundary condition), except for nodes located on Γ_D for which a standard Dirichlet boundary condition is imposed;
- the velocity $\mathbf{u}_s(\mathbf{t})$ is imposed on nodes in the solid, i.e. for nodes (Ns), (Nsf) and (Ni).

By construction, the additional nodes (Ni) have the same velocity as the solid. Therefore, there are no additional equations associated to the velocity of those nodes. Only the right hand side of the fluid nodes connected to them will change and by these means take into account for the boundary as being located on Γ_i .

The pressure degrees of freedom are associated to the continuity equations. In order to enforce mass conservation in the entire fluid region, the continuity equations are solved on all fluid elements, i.e. in (Ef) and (Eif). The continuity equations are not solved in the solid elements (Es) and (Eis) and the pressure is set to a constant (say zero) on solid nodes (Ns). On nodes (Nsf) a mean pressure from the fluid nodes to which the solid node is connected is determined in order to improve the visual representation of the solution. Those values are not affecting the solution in the fluid region. As for the case of static immersed boundaries, the pressure discretization is discontinuous between interface elements and the additional pressure degrees of freedom corresponding to interface nodes are eliminated by static condensation. For more details the reader should consult reference [11].

One major impact of the moving fluid/solid interface consists in the fact that the solution discretization changes on interface elements from one time step to the next. At each time step interface elements are decomposed into sub-elements in order to add interface nodes into the finite element basis as shown in Figure 3. We show here one triangular element cut by both interfaces corresponding to time steps $(n - 1)$ and (n) . The three-dimensional procedure is similar but, for simplifying the illustration, only the 2D equivalent is shown. At time $(n - 1)$ the degrees of freedom are associated to fluid nodes (filled circles), solid nodes (open circles) and interface nodes (filled diamonds) and the solution is interpolated linearly inside sub-elements (Figure

Figure 3: Element decomposition for immersed boundary that changes with time: (a) discretization at time t_{n-1} ; (b) discretization at time t_n and (c) sub-elements for time integration at time t_n .

3(a)). Between time steps $(n-1)$ and (n) then the position of the interface has changed to another location, hence requiring a change in the finite element discretization. As seen in Figure 3(b), at time (n) the degrees of freedom are associated to the same fluid and solid nodes but the new interface nodes (filled squares) are located at a different position along the edges.

When integrating the time derivative terms in equations (8) and (9) one should take into account that discretizations at time steps $(n-1)$ and (n) are different. The proposed procedure is simple and consists in subdividing the sub-elements into new sub-elements conforming to both t_{n-1} and t_n interfaces. We first decompose interface elements into sub-elements which are either entirely in the solid region or in the fluid region at time (n) , as for the static boundary case [11] (see Figure 3(b)). Then, we determine which elements (or interface sub-elements) are intersected by the fluid/solid surface corresponding to time $(n-1)$ and apply the same decomposition procedure but this time using the level-set function at the previous time step $(n-1)$. Inside the newly formed sub-elements, shown in Figure 3(c), the solution at

both time steps, i.e. at t_n and t_{n-1} , is interpolated using linear functions and the usual element integration scheme is applied. This procedure allows an accurate interpolation of solutions both at t_{n-1} , using the computational domain 3(a), and at t_n using the computational domain 3(b).

Elementary systems are therefore constructed using the following algorithm:

- Identify elements (Ei) that are cut by the interface at t_n . Elements entirely in the solid region do not contribute to the global system, while elements entirely in the fluid region are computed as usual, ignoring the presence of the interface;
- Create the additional nodes (Ni) on the interface (filled squares in Figure 3(b));
- Decompose interface elements (Ei) into sub-elements which are either entirely in the fluid region (Eif - gray filled triangles in Figure 3(b)) or in the solid region (Eis);
- Determine the velocity at the previous time step t_{n-1} on interface nodes, i.e. $\mathbf{u}_0$ on nodes Ni1 and Ni2. When the interface moves towards the interior of the solid (as is the case for the configuration illustrated in Figure 3) this velocity is interpolated from the velocities in the solid region and therefore it is not exactly representative of the fluid field. It should be noted however, that this nodal velocity is used to determine the velocity $\mathbf{u}_0$ inside interface fluid elements Eif and therefore enters in the computation of velocities at nodes Nf1, Nf2 and Ni1, Ni2. The velocity at interface nodes Ni1 and Ni2 is determined by the velocity of the solid and is imposed as boundary condition. Moreover, velocities at fluid nodes Nf1 and Nf2 is little affected by $\mathbf{u}_0$ at interface nodes because of the upwinding scheme built inside the stabilization. In such a case, the flow speed is directed from within fluid elements and not from interface elements. Therefore the previous time step velocity inside interface elements has little influence on the solution;
- Identify elements (Ei_0) that are cut by the interface at t_{n-1} and the corresponding interface nodes (Ni_{01} , Ni_{02} and Ni_{03} in Figure 3(c)) where the interface at t_{n-1} intersects the grid;

- Decompose interface elements (Ei_0) into sub-elements which are either entirely in the fluid region or in the solid region with respect to the interface at t_{n-1} ;
- Determine the solution at interface nodes (Ni_0). For the solution at the current time step this is simply done by using the basis functions for the finite elements at t_n . For the solution at t_{n-1} only the velocity is needed and it is given by the solid velocity at the respective time;
- Integrate equations inside all fluid elements. For all elements cut by the interface at the previous time step integration is made on sub-elements separately and then assembled into the elementary system. For each interface element (Eif), the sub-element matrices are assembled into an augmented elementary system that also contains the velocity and pressure degrees of freedom associated to the interface nodes (Ni);
- The degrees of freedom associated to the interface nodes are eliminated at the element level by using the Dirichlet boundary conditions for the velocity and by static condensation for the pressure. The pressure degrees of freedom on interface nodes can also be computed using the flow solution at the previous iteration. The elimination of interface degrees of freedom is the same as for the static interface case and is described in details in [11].

The following observations can be made with respect to the proposed method:

Remark 1: The velocity and pressure discretization inside interface elements make use of the shape functions defined on the sub-elements determined by the cutting interface. Therefore, the resulting solution is similar to the one obtained on a grid having nodes on the fluid/solid interface.

Remark 2: The imposition of the solid velocity is made exactly on the interface nodes, thus resulting in an accurate representation of boundary conditions at this location.

Remark 3: The continuity equations are solved for elements covering the entire fluid region, including fluid elements (Ef) and interface/fluid sub-elements (Eif). Therefore mass conservation is enforced in the discrete form of the equations.

Remark 4: The pressure on interface nodes is not an unknown of the global system of equations, but it is determined at element level as function

of the solution inside the fluid. The pressure interpolation is continuous except for the element faces cut by the interface where it is discontinuous. The procedure results in the local enrichment of the pressure discretization. For other pressure enrichment techniques resulting in local discontinuous pressure interpolations the reader may consult the work of Minev *et al.* [23] for two-fluid flow with surface tension and that of Coppola-Owen and Codina [24, 25] for two-phase and free surface flows.

Remark 5: The procedure can be seen as a local refinement of the mesh that adds/removes grid nodes on the interface between the fluid and solid regions. However, it does not lead to the burden of changing dynamically the mesh as the additional degrees of freedom are eliminated at element level.

Remark 6: Subdivision of elements cut by the interface may generate very small or distorted sub-elements. The present finite element method behaves well even in such circumstances and no degradation in the conditioning of the system matrix was observed. The resulting linear system is solved by preconditioned iterative methods with a similar number of iterations as when solving on body conforming meshes.

Remark 7: Extension of the method to second order or multistage schemes may involve more than two levels of subdivisions, thus making the implementation more difficult.

Remark 8: The method can be applied to fluid-structure interaction when the solid is a non-deformable moving body. In such a case the velocity of the solid is used as boundary condition for the fluid at the fluid/solid interface. For deformable solids, the velocity at the interface depends on the equations describing the solid deformation and therefore avoiding the addition of interface degrees of freedom may not be possible.

3.2. Deforming mesh modeling

One way of validating the IB method for moving boundaries is to compare the solution with those obtained on body-conforming grids. Because the position of the immersed boundaries changes with time the body-conforming grid must follow the change in geometry. One way to achieve this goal is to deform the mesh in such a way that, at any time, the discretization corresponds to the computational domain. For simple test cases this can be done by considering an Arbitrary Lagrangian-Eulerian formulation (ALE) with the grid deformation given by simple relationships depending on the movement of immersed bodies. The change of the computational domain in time is illustrated in Figure 4 for a moving obstacle case that will be described

Figure 4: Computational domain change caused by the movement of the immersed body ($x_1^* = x_1 + u_s t$, $x_2^* = x_2 + u_s t$).

latter in the paper. The portion of the computational domain located between two selected locations $x = x_{01}$ and $x = x_{02}$ (shown in darker gray scale in Figure 4) changes in time to account for the actual position of the obstacle. This part of the mesh suffers a contraction/expansion with time and no re-meshing or insertion/removing of mesh elements is performed. Mesh deformation with time is given by the change of variables $x \rightarrow x^*$:

$$x^*(t) = x + (1 - \alpha(x)) u_s t \quad (11)$$

$$\alpha(x) = \begin{cases} \frac{x_1 - x}{x_1 - x_{01}} & \text{for } x_{01} < x < x_1 \\ 0 & \text{for } x_1 \leq x \leq x_2 \\ \frac{x_2 - x}{x_2 - x_{02}} & \text{for } x_2 < x < x_{02} \\ 1 & \text{elsewhere} \end{cases} \quad (12)$$

where u_s is the velocity of the moving immersed body, x is the coordinate of a mesh point in the initial undeformed mesh (at $t = 0$) and x^* is the coordinate of the respective point in the deformed mesh. As the mesh deforms, the ALE formulation of the conservation equations has to take into account for the mesh velocity given by:

$$u_m = (1 - \alpha(x)) u_s \quad (13)$$

The convection velocity is then computed using Eq. (3).

4. APPLICATIONS

4.1. 2-D channel flow over a moving obstruction

This first test problem represents a variation of the flow in an obstructed channel previously used to validate the IB method for the steady state case

Figure 5: 2-D flow in an obstructed channel.

[11]. The geometry illustrated in Figure 5 consists of a channel of height H which is partially obstructed by a fence of height S and thickness l located on the lower wall. Computations are carried out for a fence having $l/H = 0.2$ and a blockage ratio $S/H = 0.75$. The fence has round corners on the top side with a radius of $0.05H$. The fence and the bottom wall on which is attached move with constant speed $u_s = -U_0$ (in direction parallel to the channel from right to left), while the top wall is fixed.

Computations were first carried out by using a body-conforming mesh (BC) and two different approaches: (a) solve using a reference frame attached to the obstacle (BC-MRF), and (b) solve using an ALE formulation for a grid that deforms with time in order to follow the obstacle (BC-ALE). Mesh deformation parameters used in equation (12) are $x_1 = -0.3H$, $x_2 = 0.1H$, $x_{01} = -9H$, $x_{02} = 10H$. The IB method was then applied and computations were done with various time steps. The BC-MRF solution will be used in this case as a reference solution to validate both the deforming mesh procedure and the IB method. Since the reference frame moves at constant speed in the BC-MRF approach no inertial forces need to be added and the fixed reference and moving reference solutions only differ by the constant velocity of the obstacle. For all computations, the initial solution was the steady state solution obtained when imposing the velocity boundary condition $u = u_s$ on the bottom wall and obstacle. The initial solution for the BC-MRF approach was the same as the one for BC-ALE but corrected to take into account for the moving reference frame. Note that this solution is not the converged solution of the transient problem of interest because does not take into account geometry changes of computational domain caused by the obstacle movement. Simulations were carried out for a time $t = 5H/U_0$ and the final

Figure 6: Meshes for the obstructed channel flow.

solutions provided by the IB method and the two body-conforming methods are compared. The problem is two-dimensional and therefore only a slab was meshed with 3D tetrahedral elements.

Figure 6 illustrates the two meshes in the region of the obstacle (the same grid was used for the two BC approaches). For the IB grid the solid region at $t = 0$ is shown with less intensity. A mesh convergence study was carried out and it was found that solutions on the final meshes shown in Figure 6 are grid converged. The Reynolds number used for the simulations is defined based on the obstacle height S and obstacle velocity U_0 as $Re_S = (\rho U_0 S)/\mu$. Computations were performed for $Re_S = 82.5$ (the equivalent Reynolds number based on the channel height is $Re_H = 110$).

The u -velocity distribution when the obstacle was displaced by $5H$ is shown in Figure 7. The time step for all cases was $\Delta t = 0.01H/U_0$, value for which the obstacle moves by one element during each time step in the IB method. In order to compare the solution in the moving reference frame with those obtained in the fixed reference frame, the BC-MRF solution was translated into the fixed frame by adding the velocity of the moving obstacle. As can be seen the IB solution compares very well with the two BC solutions which are almost identical.

The final u -velocity profile at various locations along the channel is shown in Figure 8. Note that at the end of the simulation the obstacle is located

(a) BC-MRF

(b) BC-ALE

(c) IB method

Figure 7: Velocity u/U_0 distribution at $t = 5H/U_0$ ($u = -U_0$ on the bottom wall and obstacle).

(a) $x/H = -5$

(b) $x/H = -4$

(c) $x/H = -3$

(d) $x/H = -2$

Figure 8: Velocity profile at various locations in the channel.

Figure 9: Pressure profile along the top and bottom walls.

between $x/H = -5.2$ and $x/H = -5$ (displaced to the left by $5H$). Figure 8 indicates that the solution from the IB method is in excellent agreement with those using body conforming grids. The solution at $x/H = -2$ experiences the largest differences with respect to the BC solutions. For $x/H = -1$ and higher (not shown) the velocity profile is almost linear across the channel and the agreement between the three solutions is excellent. The pressure profile on the top wall and on the bottom wall behind the obstacle are shown in Figure 9. Again excellent agreement is observed between the three solutions.

For this test problem, the IB method is at a disadvantage when compared with the BC solutions. For the BC-MRF approach the obstacle is fixed and the transient solution converges towards the steady state counterpart, thus having little sensitivity with respect to the time step. For the BC-MRF the obstacle is moving but the mesh in the region of the obstacle moves with the obstacle and hence the convection velocity is small. On the other hand, the IB solution has to deal with the continuous relocation of the obstacle, the change in the discretization caused by the moving fluid/solid interface and important changes in the solution at grid points located in the region traversed by the obstacle. In order to determine the effect of the time step on the solution accuracy we recomputed the solution with the three methods using time steps two times smaller ($\Delta t = 0.005H/U_0$) and four times smaller ($\Delta t = 0.0025H/U_0$) than in the first set of computations. Figure 10 shows the u -velocity profile at $x/H = -3$ and $x/H = -2$ for the solutions with $\Delta t = 0.0025H/U_0$. The solution at those two locations

Figure 10: Velocity profile for $\Delta t = 0.0025H/U_0$.

Figure 11: Length of the recirculation zone behind the obstacle.

Figure 12: Uniform flow around a circular cylinder: Domain and boundary conditions.

has proved to be more sensitive to the choice of the numerical approach than in the rest of the flow domain. The results indicate that the three solutions are practically superimposed one over each other. The length of the recirculation zone formed behind the obstacle when looking from a reference frame attached to the bottom wall is compared in Figure 11 for solutions obtained using different time steps. The difference between the prediction of the three different methods decreases when decreasing the time step. For the largest time step ($\Delta t = 0.01H/U_0$), the difference between the IB solution and the BC-ALE approach is about 0.8%. For the smallest time step ($\Delta t = 0.0025H/U_0$), the largest difference is observed between the two solutions using boundary conforming grids. Still, this difference is about 20 times smaller than the grid element size in this region and represents only 0.085% of the computed value of the recirculation length. We may conclude that the agreement is excellent, indicating that both the body-conforming deforming grid method and the IB method perform well.

4.2. In-line oscillating cylinder in uniform flow

4.2.1. Problem statement.

This problem consists in the flow around an in-line oscillating cylinder placed in uniform flow. The computational domain and boundary conditions are shown in Figure 12 with the flow entering the left side with uniform velocity. The cylinder is located at 10 diameters from the inlet and at 20 diameters from the outlet. Because the problem is two-dimensional, only a slab was meshed with 3D tetrahedral elements and the w component of the velocity was set to zero. The mesh is shown in Figure 13(a) and was designed to have smaller elements in the region of the immersed cylinder and

Figure 13: Mesh for the flow around an oscillating circular cylinder

in the wake of the cylinder where the solution is expected to exhibit larger time variations. The grid has 300 elements in flow direction, 200 elements in direction normal to the flow and only 2 elements are used in the width of the slab for a total of 600,000 tetrahedral elements. To assess the accuracy of the IB solution, comparison was made with the finite element solution computed on the body conforming grid (BC-ALE) shown in Figure 13(b). The performance of the procedure used to impose boundary conditions on the moving immersed interface can thus be estimated. Mesh deformation parameters used in equation (12) are $x_1 = -0.5d$, $x_2 = 0.5d$, $x_{01} = -1.5d$, $x_{02} = 1.5d$.

The numerical simulations were conducted using the flow parameters of Saiki and Biringen [26]. The Reynolds number is $Re = \rho U_0 d / \mu = 200$, where U_0 is the free-stream velocity and d is the cylinder diameter. The cylinder is oscillating parallel to the free-stream flow at a frequency $f_c = 1.88f_s$, with f_s being the Strouhal frequency for the stationary cylinder. For $Re = 200$ we have $St = f_s d / U_0 = 0.195$ [11] and therefore $f_c = 0.367U_0/d$. The center of the cylinder is considered initially at $x_c = 0$, $y_c = 0$ and is displaced in time following the relationship $x_c = A_c \sin(2\pi f_c t)$ with $A_c = 0.24$. Experimental data for the same conditions and $Re = 190$ are reported by Griffin and Ramberg [27] and similar experiments were made by Ongoren and Rockwell [28]. It is reported that there are three vortices shed during a cycle of the motion, two being clockwise and one counter-clockwise. This flow pattern is identified as anti-symmetrical mode A-III by Ongoren and Rockwell [28] and occurs when f_c/f_s is close to 2. Numerical solutions using an IB method for

(a) u -velocity at $x = 1.5d$

(b) v -velocity at $x = 1.5d$

(c) u -velocity at $x = 3.5d$

(d) v -velocity at $x = 3.5d$

Figure 14: Velocity evolution with time at $x = 1.5d$ and $x = 3.5d$.

$Re = 100$ and $f_c/f_s = 2$ are provided by Liao *et al.* [29].

4.2.2. Flow solution.

The initial conditions are obtained from the steady state solution around a stationary cylinder at $Re = 125$ for which a symmetric recirculation region is formed behind the cylinder. Following the work of Sohankar [30], the time-step was set to $\Delta t = 0.025d/U_0$, but simulations were also performed with smaller time steps, $\Delta t = 0.0125d/U_0$ and $\Delta t = 0.00625d/U_0$ respectively.

Both the IB and BC-ALE solutions resulted in a periodic flow with a time period twice that of the oscillating cylinder ($T = 2T_c$) in agreement with the experimental observations. The flow signal at points located on the symmetry axis downstream from the cylinder at $x/d = 1.5$ and $x/d = 3.5$ is shown in Figure 14. For $\Delta t = 0.025d/U_0$ there are small differences in the u -velocity predicted by the two methods. This is more apparent at the point located farther from the cylinder. However, these differences decrease signif-

Figure 15: Maximum and minimum u -velocity as function of the number of steps par period.

icantly when the time step increment decreases and the agreement between the two sets of results is excellent at $\Delta t = 0.0125d/U_0$ and $\Delta t = 0.00625d/U_0$. For the v -component of the velocity the two solutions agree well for all values of the time step. Remark that the solution exhibits important changes when the time step decreases from $\Delta t = 0.025d/U_0$ to $\Delta t = 0.0125d/U_0$ and then smaller differences are observed between $\Delta t = 0.0125d/U_0$ and $\Delta t = 0.00625d/U_0$. We may conclude that the solution is very sensitive to the time step increment. However, both methods behave in exactly the same way indicating that the immersed boundary algorithm performs very well.

The velocity signal presents one maximum and one minimum of the v -velocity during each period of the flow, but 2 local maximums and 2 local minimums are observed for the axial velocity u . This is determined by the fact that the axial velocity varies in phase with the cylinder velocity. The two maximums and and two minimums of the u -velocity are shown in Figure 15 and the maximum and minimum of the v -velocity are shown in Figure 16 for the point located at $x = 1.5d$, $y = 0$. We observe that when decreasing the time step, the difference between the two local maximums and the difference between the two local minimums increases as shown in Figure 15. Also the difference in the absolute values of the maximum and minimum v -velocity increases when decreasing the time step thus indicating deviation from a symmetrical vortex shedding. This behavior coincides with the flow pattern having more intense vortices and the accentuation of the anti-symmetrical mode A-III (one vortex clockwise and two counter-clockwise, [28]) observed

Figure 16: Maximum and minimum v -velocity as function of the number of steps par period.

for $\Delta t = 0.0125d/U_0$ and $\Delta t = 0.00625d/U_0$ (see Figure 17). The IB and BC-ALE solutions are almost identical when solving with these two smaller time steps. This flow pattern agrees well also with the numerical solution of Saiki and Biringen [26].

Vorticity contours during a vortex shedding period are shown in Figure 18 for the IB solution and $\Delta t = 0.00625d/U_0$. Recall that the time period of vortex shedding T is twice that of the cylinder oscillation T_c . Three vortices shed during a cycle of the motion (composed of two complete oscillations of the cylinder), one being clockwise (red in Figure 18) and two counter-clockwise (blue in Figure 18).

The vortex contours at different times during the vortex shedding cycle are compared with those provided by the BC-ALE method in Figure 19. Both solutions were computed with the smaller time step increment. Results indicate that the two methods produce very similar solutions, thus indicating that the IB method performs well.

4.3. Flow inside a single screw mixer

4.3.1. Problem definition.

This problem is inspired from extrusion and injection molding applications in which plastic pellets are mixed in a screw plasticizing unit prior to being injected into a mold. We propose to simulate the flow inside a single

Figure 17: Vorticity contours: IB and BC-ALE solutions at $t = 2nT_c$ for various time step increments.

Figure 18: Vorticity contours for flow around an in-line oscillatory circular cylinder.

Figure 19: Vorticity contours: Comparison of IB (left) and BC-ALE (right) solutions.

Figure 20: Screw profile for the single screw mixer.

screw mixer, for which the screw rotates at a constant angular speed ω inside a fixed barrel of radius R . The screw profile, illustrated in Figure 20, has an inner radius $r_i = 0.4R$, outer radius $r_o = 0.8R$ and pitch $L = R$. The profile has a trapezoidal shape with round corners having a radius $r_r = 0.05R$. A non-dimensional solution is obtained by defining reference values for the variables describing the geometry and the flow. The reference length is taken equal to the barrel radius R and the reference velocity is given by $U_0 = \omega R$. Computations were performed for a Reynolds number $Re = \rho U_0 R / \mu = 100$.

Numerical simulations were carried out considering a total screw length $L_s = 4L$. The computational domain spans from $x = 0$ to $x = 4L$ and from $r = 0.3R$ to $r = R$, where $r = (y^2 + z^2)^{1/2}$. Figure 21(a) shows the computational domain, with the fluid region shown in transparent gray. A cross-section of the grid at $x = 2L$ is shown in Figure 21(b) with the solid region of the grid plotted with lighter gray scale. The mesh has a total of 344,544 nodes and 1,658,880 tetrahedral elements.

The following boundary conditions are imposed:

- Non-slip boundary conditions on the barrel surface, $u = v = w = 0$;
- Imposed axial velocity $u = 0$ and zero-traction condition for the v and w components at $x = 4L$;
- Zero-traction condition for all velocity components at $x = 0$. This condition will set the zero pressure level on this surface;

Figure 21: Computational domain and screw geometry.

- Screw velocity $\mathbf{u} = \omega \times \mathbf{r}$ for points inside the screw. Screw rotation speed is $\omega = -\omega \mathbf{i}$, which gives $u = 0$, $v = \omega z$, $w = -\omega y$.

The apparent velocity of the screw profile is $\mathbf{u}_a = -L\omega/(2\pi)\mathbf{i}$, which means that the profile is moving in the negative direction of the x -axis by a distance equal to the screw pitch L in a time interval equal to the rotation period $T = 2\pi/\omega$.

4.3.2. Numerical results.

A first computation was carried out using a time step increment $\Delta t = T/40$ and then by reducing successively Δt by a factor of two: $\Delta t = T/80$, $T/160$ and $T/320$. Note that the mesh has 96 elements in circumferential direction, resulting in a CFL number on elements close to the screw surface between 2.4 and 0.3 for the various time steps considered. The flow solution was computed for a time interval corresponding to 5 complete screw rotations and it was found that the solution seen from a reference frame rotating with the screw remains practically unchanged after two screw rotations. However, the solution is different when using different values for the time step increment Δt because the fluid/solid interface is relocated continuously as the screw rotates and at each grid point the solution changes with time. One way to assess the effect of the time step is to compute the flow in a reference frame tied to the screw. In this case we use the same IB method and the same grid as for the rotating screw solution, but impose homogeneous

Figure 22: v -velocity component at $x = 2L$, $y = 0$.

Dirichlet conditions for the velocity inside the screw, rotating instead the barrel in the opposite direction, $\omega_b = -\omega$. In this case the fluid/solid interface remains fixed and the solution converges towards the same steady-state solution for all values of the time step. In the rotating reference frame (RRF) momentum equations should include a source term that takes into account for solving in a non-inertial reference frame. The force $\mathbf{f}$ in equation (1) is given by

$$\mathbf{f} = -2\rho\boldsymbol{\omega} \times \mathbf{u} - \rho\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x}) \quad (14)$$

where $-2\rho\boldsymbol{\omega} \times \mathbf{u}$ is the Coriolis force and $-\rho\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{x})$ is the centrifugal force.

The rotating screw IB solutions for different time step increments are compared with the RRF solution in Figures 22 and 23. Figure 22 shows the v -velocity component (a) and the error with respect to the RRF solution (b) at $x = 2L$ and $y = 0$. Figure 23 shows the w -velocity component (a) and the error with respect to the RRF solution (b) at $x = 2L$ and $z = 0$. Note that the points $(x = 2L, y = 0, z = 0.5)$ and $(x = 2L, y = 0.4, z = 0)$ are on the screw surface as seen in Figure 21(b). Results indicate that the agreement with the RRF solution is very good and improves when decreasing the time step increment. This can be also seen in Figure 24 presenting the mean error of the solution for the two cross-sections considered in Figures 22 and 23. The slope of the curve $\log(\text{error}) - \log(\Delta t)$ is very close to the theoretical value of 1 of a first-order Euler implicit scheme.

Figure 23: w -velocity component at $x = 2L$, $z = 0$.

Figure 24: Mean error of velocity components with respect to the RRF solution.

Figure 25: Computational domain and mesh for twin-screw mixer.

4.4. Flow inside a co-rotating twin screw mixer

4.4.1. Problem definition.

In this application, we consider a co-rotating twin screw mixing device in which each screw has the same geometry as the one defined in the previous example. This problem is hard to solve using body-conforming grids, as one should deal with complex computational domain changes resulting in frequent remeshing or mesh adaptation. The IB method does not have this drawback as the computational grid remains the same during the course of a simulation.

The screws are located at a distance $\Delta z = 1.3R$ (measured between their axis of rotation) and the computational domain has a length $L_s = 3L$. Figure 25(a) shows the computational domain, with the fluid region shown in transparent gray. A cross-section of the grid at $x = 1.25L$ is shown in Figure 25(b) with the solid region of the grid plotted with lighter gray scale. The mesh has a total of 451,067 nodes and 2,177,280 tetrahedral elements. Both screws rotate in the same direction at angular speed ω . Solutions are computed using the same non-dimensional form the equations and boundary conditions as for the previous example and the Reynolds number is $Re = \rho\omega R^2/\mu = 100$.

4.4.2. Numerical results.

The problem was solved using the proposed IB method and various time step increments: $\Delta t = T/40, T/80, T/160, T/320$ and $T/640$, where T is the rotation period of the two screws $T = 2\pi/\omega$, $\omega = 1$. For this problem there

Figure 26: v -velocity component at $x = 1.25L$, $y = 0$.

is no way to make a change in the reference frame such as to have the screw surface stationary. Therefore we have used the solution for the smallest time step increment ($\Delta t = T/640$) as a reference with which the other solutions are compared.

Figure 26 shows the v -velocity component (a) and the error with respect to the reference solution (b) at $x = 1.25L$ and $y = 0$. Figure 27 shows the w -velocity component (a) and the error with respect to the reference solution (b) at $x = L$ and $z = 0$. As can be seen, the agreement with the reference solution is very good and improves when decreasing the time step increment. This can be also seen in Figure 28 presenting the mean error of the solution with respect to the reference one for the two cross-sections considered in Figures 26 and 27. The slope of the curve representing the error as a function of the time step increment is very close to the theoretical value of 1 of a first-order Euler implicit scheme.

5. CONCLUSIONS

A new IB finite element method that accurately imposes the boundary conditions on the immersed interface is presented for the case of moving boundaries. The procedure consist of incorporating into the grid the points where the mesh intersects the boundary. The degrees of freedom associated with the additional grid points are then eliminated either because the velocity is known or by static condensation in the case of the pressure.

(a) w -velocity profile

(b) Error with respect to the reference solution

Figure 27: w -velocity component at $x = L$, $z = 0$.

Figure 28: Mean error of velocity components with respect to the reference solution.

Application of the proposed method to the flow in an obstructed channel shows an excellent agreement with body-conforming grid solutions. The agreement improves when the time step increment used for time integration is decreased.

For the application to transient flow past a circular cylinder oscillating in-line with the free-stream the results compare well with experimental data and previous numerical results. The solution shows the same sensitivity with respect to the time step increment for the IB method as for the body conforming grid solution. The IB solution recovers the anti-symmetric vortex shedding pattern observed experimentally for a frequency of the cylinder oscillation near twice that of the vortex shedding behind the stationary cylinder

The three-dimensional IB finite element method was also used to solve the flow inside a single-screw mixer and the solution was compared with the one obtained in a reference frame rotating with the screw. The IB solution is accurate and again improves when the time step decreases. The flow inside a co-rotating twin-screw extruder is more complex and in the absence of a reference solution on a body-conforming grid we used the solution for the smallest time step as reference. The method performs well and the accuracy improves by decreasing the time step.

REFERENCES

- [1] C.S. Peskin, “Flow patterns around heart valves: a numerical method,” *Journal of Computational Physics*, vol. 10, pp. 252–271, 1972.
- [2] C.S. Peskin, “Numerical analysis of blood flow in the heart,” *Journal of Computational Physics*, vol. 25, pp. 220–252, 1977.
- [3] R. Mittal and G. Iaccarino, “Immersed boundary methods,” *Annu. Rev. Fluid Mech.*, vol. 37, pp. 239–261, 2005.
- [4] R. Tyson, C.E. Jordan and J. Hebert, “Modelling anguilliform swimming at intermediate Reynolds number: A review and a novel extension of immersed boundary method applications,” *Comput. Methods Appl. Mech. Engrg.*, vol. 197, pp. 2105–2118, 2008.
- [5] Y. Kim and C.S. Peskin, “3-D parachute simulation by the immersed boundary method,” *Computers & Fluids*, vol. 38, pp. 1080–1090, 2009.

- [6] F. Margnat and V. Morinière, “Behaviour of an immersed boundary method in unsteady flows over sharp-edged bodies,” *Computers & Fluids*, vol. 38, pp. 1065–1079, 2009.
- [7] R. Glowinski, T.-W. Pan and J. Périaux, “A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations,” *Comput. Methods Appl. Mech. Engrg.*, vol. 112, pp. 133–148, 1994.
- [8] L. Zhang, A. Gerstenberger, X. Wang and W.K. Liu, “Immersed finite element method,” *Comput. Methods Appl. Mech. Engrg.*, vol. 193, pp. 2051–2067, 2004.
- [9] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph and J. Périaux, “A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: application to particulate flows,” *International Journal for Numerical methods in Fluids*, vol. 30, pp. 1043–1066, 1999.
- [10] R. Codina, G. Houzeaux, H. Coppola-Owen and J. Baiges, “The fixed mesh ALE approach for the numerical approximation of flows in moving domains,” *Journal of Computational Physics*, vol. 228, pp. 1591–1611, 2009.
- [11] F. Ilinca, J.-F. Héту, “A finite element immersed boundary method for fluid flow around rigid objects,” *Int. Journal for Numerical Methods in Fluids*, in press, 2009.
- [12] T.E. Tezduyar, R. Shih, S. Mittal and S.E. Ray, “Incompressible flow using stabilized bilinear and linear equal-order-interpolation velocity-pressure elements,” *Research Report UMSI 90/165*, University of Minnesota/Supercomputer Institute, Minneapolis, 1990.
- [13] F. Ilinca, D. Pelletier, and A. Garon, “An adaptive finite element method for a two-equation turbulence model in wall-bounded flows,” *International Journal for Numerical Methods in Fluids*, vol. 24, pp. 101–120, 1997.
- [14] L.P. Franca and S.L. Frey, “Stabilized finite element methods: II. The incompressible Navier-Stokes equations,” *Comp. Methods Appl. Mech. Engrg.*, vol. 99(2-3), pp. 209–233, 1992.

- [15] Y. Saad, "Iterative Methods for Sparse Linear Systems," PWS Publishing Company, Boston, MA, 1996.
- [16] H. Van der Vorst, "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM J. Sc. Stat. Comp.*, vol. 13, pp. 631–644, 1992.
- [17] F. Ilinca, J.-F. Hétu, M. Audet and R. Bramley, "Simulation of 3-D Mold-Filling and Solidification Processes on Distributed Memory Parallel Architectures," *ASME Winter Annual Meeting, Proc. Fluids Eng. Div.*, Dallas, TX, vol. 244, pp. 477–484, 1997.
- [18] F. Ilinca, and J.-F. Hétu, "Finite element solution of three-dimensional turbulent flows applied to mold-filling problems," *Int. J. Num. Methods Fluids*, vol. 34, pp. 729–750, 2000.
- [19] F. Ilinca and J.-F. Hétu, "Three-dimensional filling and post-filling simulation of polymer injection molding," *Int. Polymer Proc.*, vol. 16, p. 291–301, 2001.
- [20] F. Ilinca F, J.-F. Hétu, A. Derdouri and J. Stevenson, "Metal injection molding: 3D modeling of nonisothermal filling," *Polymer Eng. Sc.*, vol. 42, pp. 760–770, 2002.
- [21] F. Ilinca and J.-F. Hétu, "Three-dimensional finite element solution of gas-assisted injection molding," *Int. J. Num. Methods Engng.*, vol. 53, pp. 2003–2017, 2002.
- [22] F. Ilinca, J.-F. Hetu, and A. Derdouri, "Numerical investigation of the flow front behavior in the co-injection molding process," *Int. J. Num. Methods Fluids*, vol. 50, pp. 1445–1460, 2006.
- [23] P.D. Minev, T. Chen and K. Nandakumar, "A finite element technique for multifluid incompressible flow using Eulerian grids," *J. Comput. Physics*, vol. 187, pp. 255–273, 2003.
- [24] A.H. Coppola-Owen and R. Codina, "Improving Eulerian two-phase flow finite element approximation with discontinuous gradient shape functions," *Int. J. Num. Methods Fluids*, vol. 49, pp. 1287–1304, 2005.

- [25] A.H. Coppola-Owen and R. Codina, “A finite element model for free surface flows on fixed meshes,” *Int. J. Num. Methods Fluids*, vol. 54, pp. 1151–1171, 2007.
- [26] E.M. Saiki and S. Biringen, “Numerical Simulation of a Cylinder in Uniform Flow: Application of a Virtual Boundary Method,” *J. Comp. Phys.*, vol. 123, pp. 450–465, 1996.
- [27] O.M. Griffin and S.E. Ramberg, “Vortex shedding from a cylinder vibrating in line with an incident uniform flow,” *J. Fluid Mech.*, Vol. 75, pp. 257–271, 1976.
- [28] A. Ongoren and D. Rockwell, “Flow structure from an oscillating cylinder, Part 2. Mode competition in the near wake,” *J. Fluid Mech.*, Vol. 191, pp. 225–245, 1988.
- [29] C.-C. Liao, Y.-W. Chang, C.-A. Lin and J.M. McDonough, “Simulating flows with moving boundary using immersed-boundary method,” *Computers & Fluids*, vol. 39, pp. 152–167, 2010.
- [30] A. Sohankar, C. Norberg and L. Davidson, “Low-Reynolds-Number Flow Around a Square Cylinder at Incidence: Study of Blockage, Onset of Vortex Shedding and Outlet Boundary Condition,” *Int. J. Num. Methods Fluids*, Vol. 26, pp. 39–56, 1998.