A Semi-Implicit Immersed Boundary Method and its Application to Viscous Mixing

Bruno Blais[a], Manon Lassaigne[a], Christoph Goniva[b], Louis Fradette[a], François Bertrand[a,*]

[a] *Research Unit for Industrial Flow Processes (URPEI), Department of Chemical Engineering, École Polytechnique de Montréal, P.O. Box 6079, Stn Centre-Ville, Montréal,QC, Canada, H3C 3A7*
[b] *DCS Computing GmbH, Altenbergerstraße 66a A-4040 Linz, Austria*

Abstract

Computational fluid dynamics (CFD) simulations in the context of single-phase mixing remain challenging notably due the presence of a complex rotating geometry within the domain. In this work, we develop a parallel semi-implicit immersed boundary method based on Open∇FOAM, which is applicable to unstructured meshes. This method is first verified on academic test cases before it is applied to single phase mixing. It is then applied to baffled and unbaffled stirred tanks equipped with a pitched blade impeller. The results obtained are compared to experimental data and those predicted with the single rotating frame and sliding mesh techniques. The proposed method is found to be of comparable accuracy in predicting the flow patterns and the torque values while

*Corresponding author
Email address: francois.bertrand@polymtl.ca (François Bertrand)

being straightforwardly applicable to complex systems with multiples impellers for which the swept volumes overlap.

Keywords: Computational fluid dynamics; Immersed boundary method; Single rotating frame technique; Sliding mesh technique; Mixing; Open∇FOAM

1. Introduction

In the process industry, mixing is often a critical unit operation that has a large impact on the yield, physical attributes and manufacturing cost of a product [1]. Single phase mixing, although relatively well understood, remains a challenging topic notably due to turbulence [2, 3] or rheology [4, 5] in the tank and to the arduous scale-up (or scale-down) of units, as is discussed in Machado *et al.* [6].

On the other hand, multiphase mixing, such as solid-liquid or gas-liquid mixing, is more complex due to the two-way coupling between the phases, which takes place at both the microscopic and mesoscopic flow scales [3]. This coupling usually has an impact on the global characteristic of the flow, the complexity of which can be exacerbated by rheology or turbulence. Furthermore, it renders highly challenging the prediction of macroscopic mixing quantities such as the

torque on the impeller, the just-suspended speed or the gas hold-up.

In both single-phase and multiphase mixing, computational fluid dynamics (CFD) has proven to be an inexpensive, fast and efficient tool to gain insight into the flow behavior in various set-ups [1]. For example, for solid-liquid mixing, two-fluid models have been solved via the finite volume method using FLUENT and to obtain, rather successfully in the turbulent regime, suspension curves [7], minimum speed for complete suspension [8] and solid particle distributions [9]. However, many challenges remain in solid-liquid mixing, such as the possibility of simulating suspensions of particles of different sizes and densities [10, 11], and suspensions in the laminar and transitional regimes [12, 13]. These are only tractable with complex models such as the CFD-DEM approach that combines a CFD technique for the fluid phase and the discrete element method (DEM) for the solid particles [14]. Similar problematic exist for the cases of gas-liquid and liquid-liquid (emulsion) mixing, for which non-trivial models such as population balance models [15] have the potential to shed light on the dynamics prevailing in this type of operations.

One of the recurring obstacle to the use of CFD in mixing, especially in complex multiphase models such as those based on CFD-DEM, is related to the presence of a single or multiple rotating impellers within the tank, with or

without baffles, combined with the unsteady character of the flow and the need for distributed memory parallelism owing to large computational times.

The three usual approaches to tackle this problem are : the sliding mesh (SM), multiple reference frame (MRF) and the single rotating frame (SRF) methods. However, they are known to suffer from limitations in terms of the type of geometry they can handle [3] and their capability to solve the unsteady- or steady-state Navier-Stokes equations [3, 16]. These issues, which provided the impetus for the present work, are discussed thoroughly in the next section.

The immersed boundary (IB) and fictious domain (FD) methods are interesting alternatives as they can handle complex geometries in a parallel and computationally efficient manner, without requiring an explicit mesh of the geometry. Moreover, they can be generalized to complex configurations, such as those involving multiple impellers for which the swept volumes overlap, and to full six degrees of freedom kinematics.

Although their use in mixing is not widespread, the IB and FD methods are efficient for many applications. For instance, a fictious domain based on the finite element method, developed by Bertrand *et al.* [17], was used to study numerous types of mixing configurations such as co-axial [18], rotor-stator [19] and planetary mixers [20]. However, for the study of single phase and multiphase

flows, the finite volume method has highly interesting local mass conversation properties that make it an adequate scheme, especially for hybrid methods such as CFD-DEM or for extensive turbulence modeling using large eddy simulation (LES). Consequently, there is a need for accurate finite volume based IB and FD methods applicable to the study of complex multiphase flow in mixing systems.

Most finite volume based IB and FD methods developed so far rely on fractional-step methods [21, 22] on structured Cartesian hexahedral meshes, which renders their implementation in the context of a generic cell-centered unstructured CFD solvers problematic. In the present work, we remedy this issue by means of the Open∇FOAM[23] library by developing a new semi-implicit immersed boundary method that brings into play a pressure implicit with splitting of operators (PISO) formulation. This immersed boundary method, which is directly integrated within the CFDEM framework [24, 25], has the following properties: it is efficient on unstructured meshes, compatible with static or dynamic mesh refinement, and requires only the use of a surface mesh to describe a moving object within the computational domain.

First, the main categories of approaches for conformal meshes or immersed boundaries are reviewed and their respective limitations are discussed. This review paves the way to the design of a new semi-implicit immersed boundary

method based on a PISO scheme, which is first verified using three academic test cases. Next, this method is applied to the study of the single phase mixing of a viscous fluid in a stirred tank provided with a pitched blade turbine, with and without baffles. The validity of the proposed IB approach is further assessed by comparing numerical results obtained with it to experimental data and other results coming from single rotating frame (SRF) and sliding mesh (SM) techniques. Finally, the accuracy of the proposed method and its potential for the study of complex mixing systems for both single and multiphase flows are discussed.

2. Strategies for moving boundaries

In this section, a brief review of the various approaches for the simulation of rotating objects is carried out with an emphasis on formulations and their theoretical limitations in the context of the finite volume method. The focus is geared towards the simulation of mixing flows using rigid impellers, and all issues related to fluid-structure interactions in the context of a flexible geometry are not addressed.

This review is split into two parts. The first one refers to conformal mesh techniques and the second to immersed boundary and fictitious domain meth-

ods.

2.1. Conformal mesh techniques

In conformal mesh methods, the impeller geometry is explicitly meshed and its motion is accounted for using surface boundary conditions. Semi-empirical methods such as those involving impeller boundary conditions (IBC) [26] are not discussed here, as their accuracy is highly dependent on the quality of local flow measurements and they cannot be extended to unsteady simulations. Consequently, they are not generic methods for moving boundaries and offer poor accuracy when compared to the methods presented in this section. For an extensive review of the results obtained via conformal mesh approaches in the context of mixing, the reader is referred to Brucato *et al.* [26].

2.1.1. Single rotating frame technique

The single rotating frame of reference (SRF) approach solves for the motion of the impeller by changing the formulation of the problem from the laboratory frame of reference to the Lagrangian frame of this impeller rotating at velocity $\boldsymbol{\omega}$. Consequently, zero-velocity no-slip boundary conditions are applied to the impeller whereas a the angular velocity imposed along the tank walls. The Navier-Stokes equations can be written for $\boldsymbol{u}$ in this rotating frame of reference,

which requires the addition of Coriolis and centrifugal forces:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1}$$

$$\partial_t \rho \boldsymbol{u} + \nabla \cdot \rho \boldsymbol{u} \otimes \boldsymbol{u} + 2\rho \boldsymbol{\omega} \times \boldsymbol{u} + \rho \boldsymbol{\omega} \times (\omega \times \boldsymbol{r}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2}$$

where $\boldsymbol{r}$ is the distance to the axis of rotation, $\boldsymbol{\omega}$ the angular velocity and p the pressure. The viscous stress tensor $\boldsymbol{\tau}$ for an incompressible flow is [27]:

$$\boldsymbol{\tau}_R = \mu \left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T \right) \tag{3}$$

where μ is the dynamic viscosity and ρ the density of the fluid.

Note that this method is valid for both transient and steady-state simulations. It is limited to geometries in which the static part of the tank is invariant by rotation. In the case of mixing, it is limited to set-ups comprising a single shaft (with one or many impellers) rotating in an unbaffled tank.

2.1.2. Sliding mesh technique

In the sliding mesh technique, the mesh is decomposed into $k + 1$ non-overlapping regions: a static region, the tank and k rotating regions, each of which rotates along with the impeller it encompasses. Consequently, the flow equations are solved for each rotating region and contain the acceleration terms corresponding to a the rotating grid, whereas flow in the static region is solved

in the laboratory frame of reference. The static region and each of the rotating subdomains regions are implicitly coupled at their interface by using face interpolation, to ensure a conservative distribution of the fluxes going from one subdomain to an adjacent one, as the mesh faces on each side do not usually coincide. Slightly different implementations of this technique exist and we refer the reader to the respective user's guides (such as [28] for FLUENT) for more details.

2.1.3. Multiple reference frame technique

In the multiple reference frame technique, the domain is decomposed into two subdomains, one using the Lagrangian frame of reference of the impeller it contains, and the other one in the (Eulerian) laboratory frame of reference. The Navier-Stokes equations are solved in each subdomain and the velocities at their interface are matched at each iteration of the underlying scheme. This method is limited to steady-state simulations as the coupling between the two subdomains assumes a steady flow condition at their interface [16].

2.2. Immersed boundary and fictious domain methods

Immersed boundary and fictious domain methods refer to a very large class of methods used to discretize a static or moving, and flexible or rigid geometry

onto a grid without using a body-fitted mesh. The mathematical formulation of immersed boundary problems is the subject of many papers in the literature, and we refer the reader to Peskin [29] and Mitall and Iccarino [30] for extensive reviews on this topic. In the following description of the various approaches, we use the terminology from Mittal and Iccarino [30].

Two main categories of immersed boundary formulations exist, depending on whether the immersed boundary conditions are enforced into the Navier-Stokes equations in a discrete or continuous manner.

In the continuous forcing approach, as introduced by the work of Peskin [31], no-slip boundary conditions are enforced by introducing a forcing term directly into the continuous Navier-Stokes equations. This forcing term acts on so-called control points that discretize the fictitious part of the geometry. In the case of an impeller or another moving part, these points, which are located on its surface, move according to the corresponding kinematics. They can be viewed as Dirac distributions that are spread over the surface of the fictitious part using a smoothing kernel function [30]. As this method was originally developed for elastic bodies, its extension to rigid boundaries, as in the case of an impeller, requires the use of either a very stiff elastic modulus [30] or of an iterative forcing scheme such as the one proposed by Goldstein *et al.* [32].

An alternative consists of using direct penalization methods coupled with Navier-Stokes/Brinkman type models and a variable permeability to enforce the motion of a solid part [33, 34]. Two types of direct penalization can be distinguished: the velocity can be directly penalized in the Navier-Stokes equation by either adding a Brinkman permeability term or increasing the viscosity within the solid part. The reader is referred to Angot *et al.* [33] for a mathematical analysis and a comparison of these two approaches. A different alternative considers that the solid part is a fluid subjected to a rigidity constraint [35, 36, 37, 38]. In this technique, a continuous forcing term, the magnitude of which depends on the relative velocity between the fluid and the solid part, is applied to the fluid within or at the surface of this solid part to take into account its motion. These two alternate approaches are iterative in nature and require the integration of continuous terms within the Navier-Stokes equations, which makes them well-suited for a PISO scheme.

In the discrete approach, the regular Navier-Stokes equations are solved and the flow is modified *a posteriori* by taking into account the no-slip boundary conditions related to the moving part either directly [39] or indirectly through ghost cells [40] or a cut-cell approach [41].

3. PISO immersed boundary scheme

The PISO immersed boundary (PISO-IB) scheme developed in the present work can be seen as a semi-implicit PISO extension of the work of Glowinski [35], Patankar *et al.* [36] and, more recently, Shirgaonkar *et al.* [37] and Curet *et al.* [38]. The aim of the PISO-IB scheme is to add an immersed boundary method to the standard PISO scheme by using the current PISO loops to impose the velocity of the immersed rigid body while maintaining mass conservation. This means that no additional loops are required. Furthermore, it leads to a less stiff system than with a direct penalization approach, especially if the part is in motion. This scheme is tailored to have good parallel efficiency and can be used with unstructured polyhedral meshes. It is also suitable for hybrid meshes where the static geometry is discretized using a conformal mesh and the moving part (e.g an impeller) is taken into account through the immersed boundary method.

The PISO-IB scheme is presented in two steps. The discretization of the immersed body using cell center and vertex flagging is first introduced before the scheme itself is presented in detail. This stair-casing approach was chosen in the present work due to its speed, but the PISO-IB scheme is also compatible

with surface-based forcing using control points. The reader is referred to the paper by Peskin for a thorough description of the latter technique [29].

3.1. Discretization of the geometry using cell and vertices flagging

The PISO-IB method requires a surface mesh of the immersed body. The projection of this body onto the finite volume mesh is then carried out by generating a boolean list indicating which cell centers and vertices of this finite volume mesh intersect with it. This operation, which adds to the computational cost of the scheme as compared to the traditional PISO approach, is highly parallelizable as the corresponding operations, which are proportional to the number of cell centers and vertices in the mesh, can be done independently by many processors (or cores). Figure 1 illustrates schematically this cell center and vertex detection technique, which creates two lists of so-called fluid and solid nodes, respectively.

Using these two lists, a solid fraction is generated for each cell i by attributing an equal weight to the sum of the vertices in this cell and its center:

$$\beta_i = \frac{N_{vc,i} + N_{cc,i} N_{v,i}}{2 N_{v,i}} \tag{4}$$

where $N_{v,i}$ is the number of vertices in cell i (eg. 8 for a hexahedron), and $N_{vc,i}$ and $N_{cc,i}$ are the number of vertices and centers intersecting the immersed body,

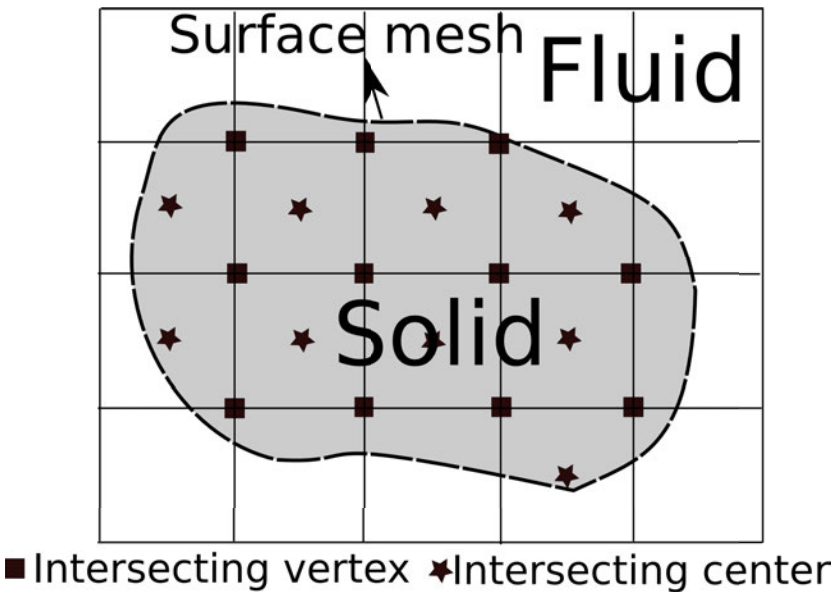

Figure 1: Illustration of the cell center and vertices flagging method.

respectively.

The vertices and the centers are also used to define the velocity of cell i that intersects the immersed body:

$$\boldsymbol{u}_{ib,i} = \boldsymbol{v}_{ib} + \frac{1}{N_{vc,\mathrm{i}} + N_{v,\mathrm{i}}} \left[\left(\sum_{j}^{N_{vc,i}} \boldsymbol{\omega} \times (\boldsymbol{x}_{v,j} - \boldsymbol{x}_{ib}) \right) + N_{v,i} \boldsymbol{\omega} \times (\boldsymbol{x}_{c,\mathrm{i}} - \boldsymbol{x}_{ib}) \right] \tag{5}$$

where $\boldsymbol{u}_{ib,\mathrm{i}}$ is the velocity of the cell i, $\boldsymbol{v}_{ib}$ is its translational velocity, $\boldsymbol{x}_{ib}$ its center of rotation, and $\boldsymbol{x}_{c,i}$ and $\boldsymbol{x}_{v,j}$ the coordinates of its center and vertices. Note that the expression within the brackets corresponds to the angular component of the velocity of the immersed body at the position of the cell i, which

is not know analytically due to the stair-casing. For a fully covered cell, this becomes equal to the angular velocity of the cell center.

Using this approach, the volume of the projected immersed body is not exactly the same as the volume the region delimited by its surface mesh. Consequently, the so-called halo layer, which corresponds to the cells i in which the body fraction $\beta_i \in]0, 1[$, is shrunk or expanded to correct the volume of the discretized immersed body. In particular, this ensures that this volume is not affected by cell alignment. This entire procedure is summarized in the block diagram of Figure 2.

3.2. The PISO-IB scheme

The PISO-IB scheme makes use of the intrinsic cycling within the PISO loop to iterate on the continuous forcing term added to the momentum equation to take into account the immersed body and its motion. We refer to the work of Issa [42] for a full presentation of the standard PISO scheme. In the present development, a notation close to that in the book by Ferziger and Perić [43] is used. For a more generic presentation of the cell-centered finite volume formulations available in Open∇FOAM, which is used in the current work, the reader is referred to Weller *et al.* [44] and Jasak *et al.* [45].

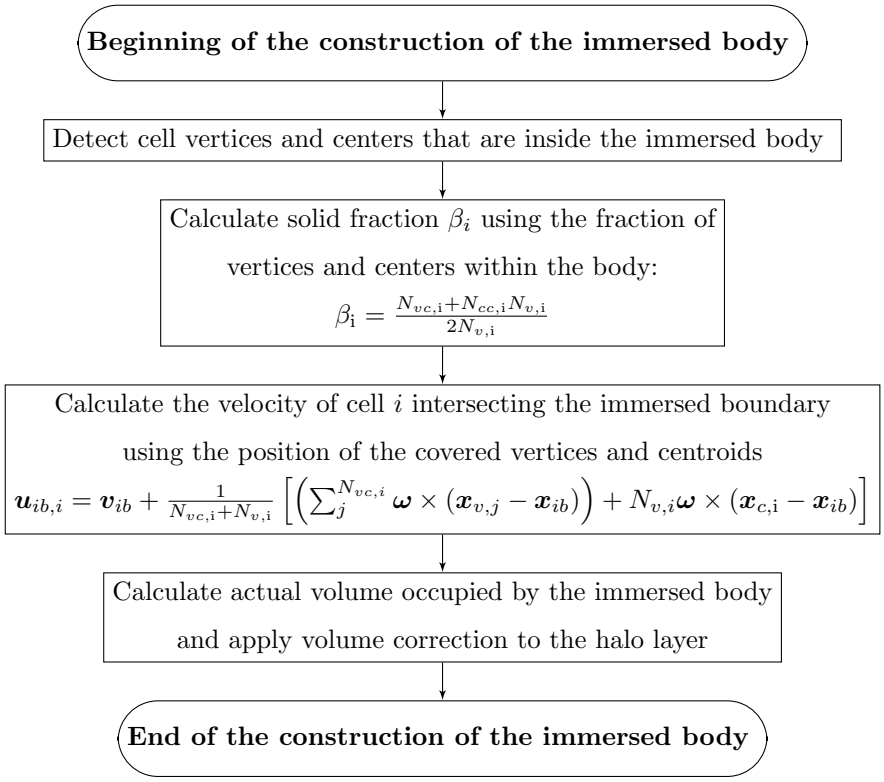

Figure 2: Block diagram for the construction of the immersed body and its velocity

The Navier-Stokes equations with the immersed boundary forcing term $\boldsymbol{H}$ are given by:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{6}$$

$$\partial_t \boldsymbol{u} + \nabla \cdot \boldsymbol{u} \otimes \boldsymbol{u} = -\frac{1}{\rho}\nabla p + \frac{1}{\rho}\nabla \cdot \boldsymbol{\tau} + \boldsymbol{H} \tag{7}$$

Time iteration begins with the update of the immersed body. The forcing term is then updated ($\boldsymbol{H}^{\mathrm{m}^*} = \boldsymbol{H}^{\mathrm{m}-1}$) so that it is zero outside of the immersed body. Next, the pressure-correction scheme starts with the solution of a predictor step for velocity $\boldsymbol{u}^{\mathrm{m}^*}$ using the pressure and velocity at time step $m-1$ (or the initial condition when $m = 1$) and the forcing term $\boldsymbol{H}^{\mathrm{m}^*}$:

$$A_{\mathrm{i}}\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^*} + \sum_{\mathrm{j}} A_{\mathrm{j}}\boldsymbol{u}_{\mathrm{j}}^{\mathrm{m}^*} = Q_{\boldsymbol{u},\mathrm{i}}^{\mathrm{m}-1} - \left(\frac{\delta p^{\mathrm{m}-1}}{\delta x}\right)_{\mathrm{i}} + \boldsymbol{H}_{\mathrm{i}}^{\mathrm{m}^*} \tag{8}$$

The content of A is linked to the discretization of the fluxes and velocity gradient whereas $Q_{\boldsymbol{u},\mathrm{i}}^{\mathrm{m}-1}$ can be deduced from the discretization of the time derivative in (7). The indices i and j refer to cell i and to the neighboring cells, respectively. The pressure term is given explicitly and the symbolic derivative is approximated by the centered scheme. The resulting velocity $\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^*}$ is not divergence free, hence the requirement of a pressure correction step.

First, a correction in the spirit of the Rhie and Chow interpolation procedure is applied to prevent the decoupling of the velocity from the pressure and the

forcing term is applied [46, 47]:

$$\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{**}} = \frac{Q_{\boldsymbol{u},\mathrm{i}}^{\mathrm{m}^{*}} - \sum_{\mathrm{j}} A_{\mathrm{j}} \boldsymbol{u}_{\mathrm{j}}^{\mathrm{m}^{*}}}{A_{\mathrm{i}}} \tag{9}$$

By means of a linear interpolation of the cell-centered velocity, the mass fluxes can be updated at the cell faces:

$$\phi_{\mathrm{F}}^{\mathrm{m}^{**}} = \langle \boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{**}} \rangle_{\mathrm{F}} \cdot \boldsymbol{S}_{\mathrm{F}} \tag{10}$$

where $\langle . \rangle_{\mathrm{F}}$ denotes the face interpolation of a variable from its value at the surrounding cell centers, and where $\boldsymbol{S}_{\mathrm{F}}$ is the surface normal vector.

A pressure correction equation that includes the forcing term $\boldsymbol{H}_{\mathrm{i}}^{\mathrm{m}^{*}}$ is then solved:

$$\sum_{\mathrm{F}} \langle \frac{1}{A_{\mathrm{i}}} \rangle_{\mathrm{F}} \boldsymbol{S}_{\mathrm{F}} \cdot \nabla p_{\mathrm{i}}^{\mathrm{m}^{**}} = \sum_{\mathrm{F}} \phi_{\mathrm{F}}^{\mathrm{m}^{**}} + \sum_{\mathrm{F}} \langle \frac{1}{A_{\mathrm{i}}} \rangle_{\mathrm{F}} \langle \boldsymbol{H}_{\mathrm{i}}^{\mathrm{m}^{*}} \rangle_{\mathrm{F}} \cdot \boldsymbol{S}_{\mathrm{F}} \tag{11}$$

which allows for the correction of the velocity using this new pressure as well as the forcing term:

$$\boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{***}} = \boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{**}} + \frac{1}{A_{\mathrm{i}}} \left(- \left(\frac{\delta p^{\mathrm{m}^{**}}}{\delta x} \right)_{\mathrm{i}} + \boldsymbol{H}_{\mathrm{i}}^{\mathrm{m}^{*}} \right) \tag{12}$$

Finally, this forcing term is corrected using the difference between the current velocity and the prescribed one within the immersed body:

$$\boldsymbol{H}_{\mathrm{i}}^{\mathrm{m}^{***}} = \boldsymbol{H}_{\mathrm{i}}^{\mathrm{m}^{*}} + \frac{\alpha \beta_{\mathrm{i}}}{\Delta t} \left(\boldsymbol{u}_{ib,\mathrm{i}} - \boldsymbol{u}_{\mathrm{i}}^{\mathrm{m}^{***}} \right) \tag{13}$$

where $\alpha \in]0, 0.9]$ is a relaxation coefficient. The algorithm then proceeds to another corrector step until the desired number of iterations or the convergence

of the forcing term is reached. The required number of iterations for convergence of either $\boldsymbol{H}$ or p is dependent on a number of factors such as the value of α and the Reynolds number of the flow.

The overall scheme is given as a block diagram in Figure 3.

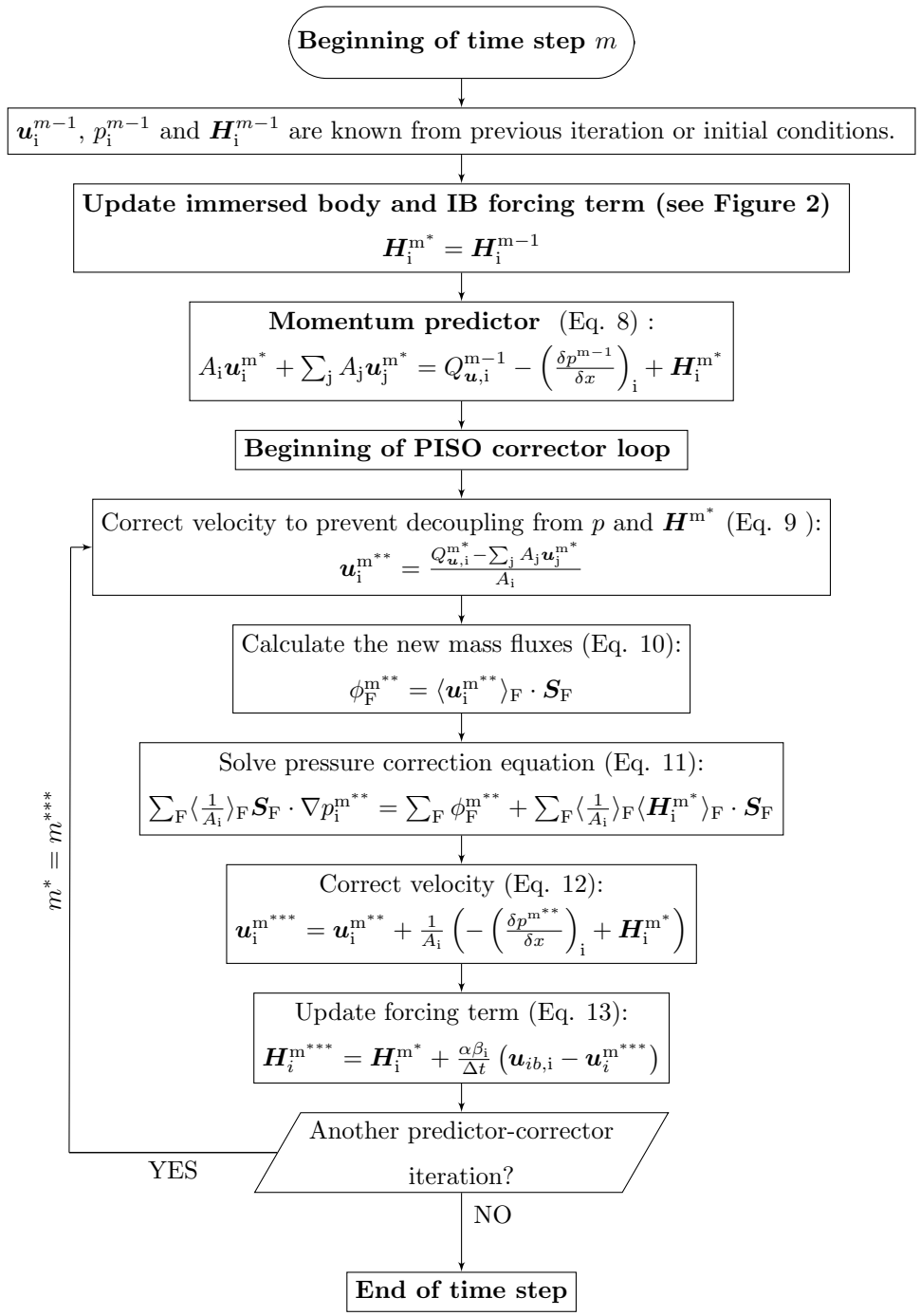

Figure 3: Block diagram for the PISO-IB scheme

4. Verification of the PISO-IB scheme

The PISO-IB scheme is first verified on three test cases: a Taylor-Couette flow, and the prediction of von Karman vortex streets past static and moving cylinders.

4.1. Taylor-Couette Flow

The Taylor-Couette flow, or flow between two-concentric cylinders, is a well-established test case for immersed boundary conditions due to the fact that the geometry is not aligned with the mesh and that such flow possesses an analytical solution. In the present case, we consider two co-axial cylinders of radii $R_i = 0.25$m and $R_o = 0.1$m, with the inner cylinder rotating. The cylinders are both discretized using the immersed boundary method on a square domain $\Omega = [-0.11, 0.11] \times [-0.11, 0.11]$. Assuming that the flow is laminar, the azimuthal velocity profile between the two cylinders is given by [27]:

$$u_\theta\left(r\right) = \Omega_i \kappa R_o \frac{\left(\frac{R_o}{r} - \frac{r}{R_o}\right)}{\left(\frac{1}{\kappa} - \kappa\right)} \tag{14}$$

where Ω_i is the angular velocity of the inner cylinder and κ the ratio of the radius of the inner cylinder to the outer cylinder ($\kappa = \frac{R_i}{R_o}$). This test case can be used to carry out an order of convergence analysis by monitoring the decrease of the Euclidean norm of the error with the mesh size. The Euclidean norm of

the error on the velocity is defined as :

$$\|e_{\boldsymbol{u}}\|_2 = \sqrt{\frac{1}{a_T}\sum_j^N a_j \|\boldsymbol{u}_{\Delta x,j} - \boldsymbol{u}_j\|^2} \tag{15}$$

where a_j is the area of fluid cell j, a_T is the total area of the geometry, and $\boldsymbol{u}_{\Delta x,j}$ the numerical velocity. This definition is general enough to be used with both conformal (body-fitted) and Cartesian homogeneous meshes. In the case of a conformal mesh, the characteristic mesh size Δx is calculated as follows:

$$\Delta x = \sqrt{\frac{a_T}{N}} \tag{16}$$

The graph in Figure 4 shows the variation of the Euclidean norm of the error on $\boldsymbol{u}$ with respect to the mesh size, and the order of convergence for both the PISO-IB method and the standard PISO scheme with a body-fitted mesh. For the PISO-IB method, the simulations were carried out using meshes containing from 20x20 to 200x200 cells. For both approaches, the order of convergence was calculated using a linear least-square regression.

It can be noted that the PISO-IB scheme degrades the order of convergence of the method from 2 to 1.33. Secondly, the convergence is noisy and a refinement of the mesh does not always lead to a reduction of the error. This is due to the stair-casing approach that is used to discretize the immersed body. However, the envelope of the error decreases as the mesh size decreases. Finally, although

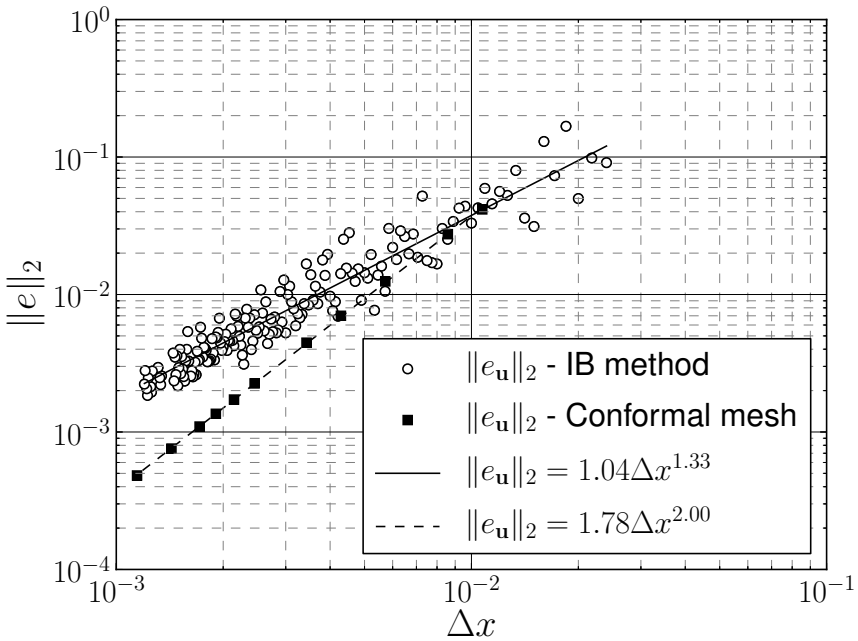

Figure 4: Euclidean norm of the error on $\boldsymbol{u}$ as a function of the mesh size, and order of convergence for the Taylor-Couette case.

the PISO-IB method degrades the order of convergence, the error obtained with it compares well with the error inherent to the standard PISO scheme and the body-fitted mesh.

4.2. Von Karman vortex street past static and moving cylinders

The unsteady flow behind a cylinder has been the topic of many experimental and numerical investigations due to the complexity of the hydrodynamics in its

unsteady oscillating wake [48]. This flow is an interesting test case because of the intrinsically periodic and complex phenomenological behavior within its vortex street. This is why it has been used by numerous authors such as [34, 49, 50, 51, 52, 53, 54, 55] for the verification of their implementation of an immersed boundary.

This test case can be studied in two different ways. The classical way is to simulate the flow past a static cylinder with a constant upstream fluid velocity, and to measure the perpendicular (lift) and parallel (drag) forces acting on it. This setup is illustrated in Figure 5. The second approach is to consider that the cylinder is moving at a constant velocity in a stagnant fluid. While this is nothing but the static problem with a simple change of reference frame, it entails a moving geometry. The setup for the latter case is identical to the static one, except that the length of the domain is increased to $L = 236$m, in order to allow sufficient time for the von Karman instability to develop. Consequently, this moving cylinder problem is more computationally demanding if the same mesh size is used, as the length of the domain is increased significantly. However, it is a pertinent test case because the moving geometry causes, at each time step, the generation of new solid and fluid nodes.

To assess the precision of the proposed immersed boundary method, both

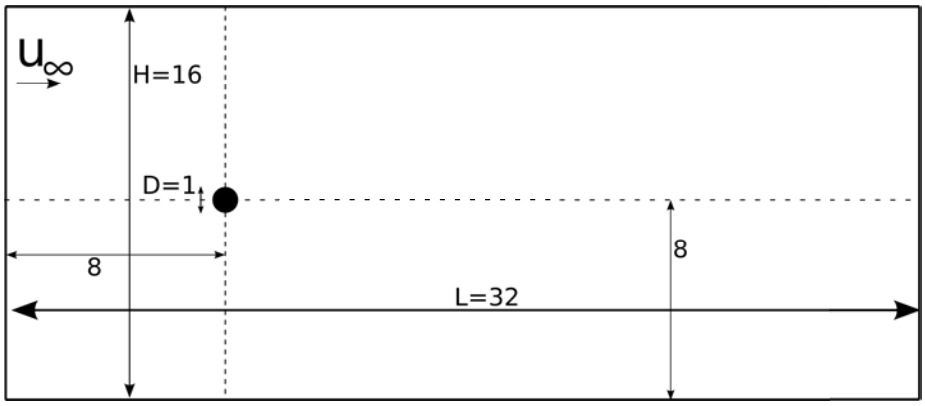

Figure 5: Von Karman vortex past a static cylinder: geometrical characteristics (dimensions in meters)

the static and moving cylinder cases were considered at $Re = 200$, which is sufficiently high to allow for vortex shedding at a constant frequency h. This frequency can be related to the upstream velocity u_∞ (or the velocity of the cylinder in the moving case) and the diameter of the cylinder (D), via the Strouhal number:

$$St = \frac{hD}{u_\infty} \tag{17}$$

The evaluation of the force $\boldsymbol{F}_C$ acting on the cylinder using our PISO-IB formulation is simple; it is the volume integral of the constraint acting on the

immersed cylinder:

$$\boldsymbol{F}_c = \int_{\Omega_c} \rho \boldsymbol{H} dV \tag{18}$$

where Ω_c is the domain occupied by this cylinder and $\boldsymbol{H}$ is the forcing term in (7).

Drag (C_D) and lift (C_L) coefficients can be defined using the x and y components of $\boldsymbol{F}_c$:

$$C_D = \frac{2F_{c,x}}{\rho u_\infty^2 D} \tag{19}$$

$$C_L = \frac{2F_{c,y}}{\rho u_\infty^2 D} \tag{20}$$

The simulations for both the moving and the static cases were carried out on Cartesian structured meshes. The corresponding number of cells and mesh size for both cases are presented in Table 1. For the static case, a convergence analysis (not shown here) revealed that the forces on the cylinder did not change significantly ($< 1\%$) if the mesh was further refined. The same mesh size was used in the dynamic case. Simulations were run for 200s to allow for a full development of the von Karman instability. The Strouhal number was calculated by obtaining the frequency of the shedding via a fast Fourier transform (FFT) of the lift coefficient. The time interval chosen for the FFT analysis was a subset of the simulation, within which the shedding frequency was constant, let alone

after around 100s (see Figure 7).

Table 1: Mesh size and number of cells used for the static and moving cylinder cases

Case	Δx	N_x	N_y
Static cylinder	0.0228	1400	700
Moving cylinder	0.0228	5600	700

Figures 6, 7 and 8 present the evolution of the drag and lift coefficients, and the amplitude of the FFT spectrum of the lift coefficient for the static and moving cylinder cases. In both cases, the drag coefficient C_D exhibits a similar behavior, decreasing steadily before the von Karman instability starts developing, and then increasing and leveling off to a slightly oscillating value. Once C_D has stabilized, it can be observed that the lift coefficient C_L oscillates between two values of opposite sign, as expected for this type of flow.

Table 2 compares to literature data the C_D, C_L and St values obtained with the PISO-IB method for the moving and the static cases. First, it can be noticed that there are some discrepancies in the literature values for this problem. Although the reasons for this are hard to identify, it can be partially attributed to differences in the mesh size, and the channel width and length used for those simulations. Indeed, these parameters are not always given explicitely.

In the present work, the configuration of Bhalla *et al.* [49] was used so that similar results were expected.

It can be noticed that for C_D and St, both the moving and the static cases are in good agreement with themselves and the literature data. However, the immersed boundary method underpredicts the lift coefficient C_L in the case of the moving cylinder. This is *a priori* surprising, as the adequacy of the C_D and St results indicates that the flow features of the flows are well predicted. This underprediction of the lift coefficient is a direct consequence of the position update of the immersed cylinder at each time step, which results in the occurence of new fluid cells on which to apply the forcing term $\boldsymbol{H}$. It is interesting to note that these cells are also responsible for high-frequency peaks in the FFT spectrum of the lift coefficient in Figure 8.

From these results, it can be concluded that the proposed PISO-IB method reproduces with good accuracy the von Karman vortex street and is therefore a valid approach for periodic unsteady flows. This verification step highlights that it was important to solve this problem for both the moving and static frames of reference, as the motion of the immersed cylinder had an impact on the quality of the results.

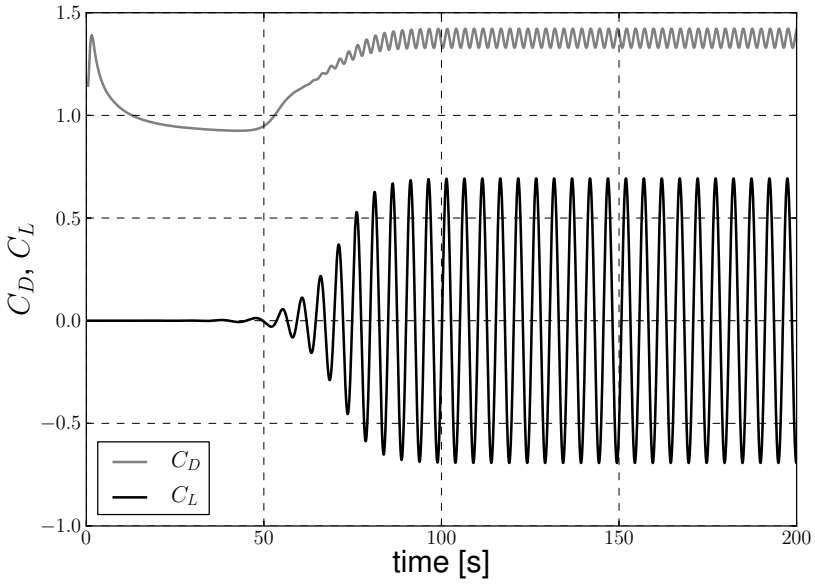

Figure 6: Drag and lift coefficients for the flow past a static cylinder.

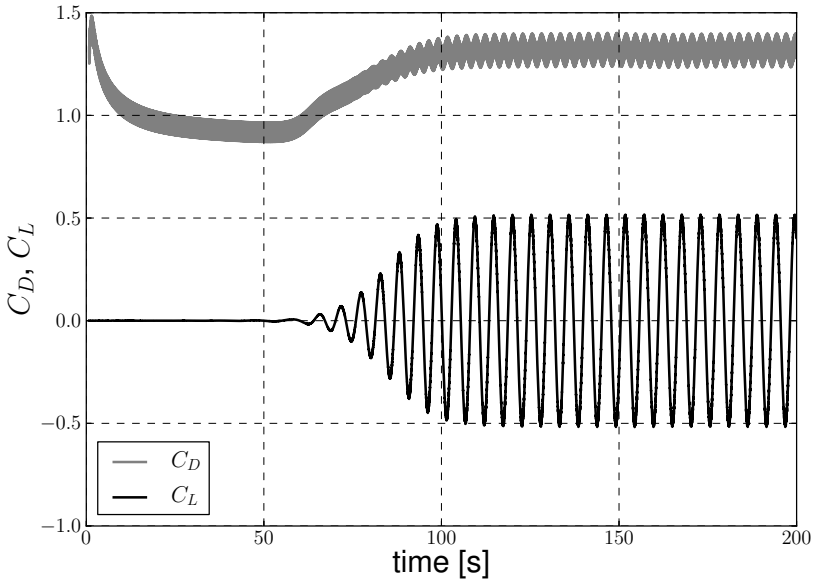

Figure 7: Drag and lift coefficients for the flow past a moving cylinder.

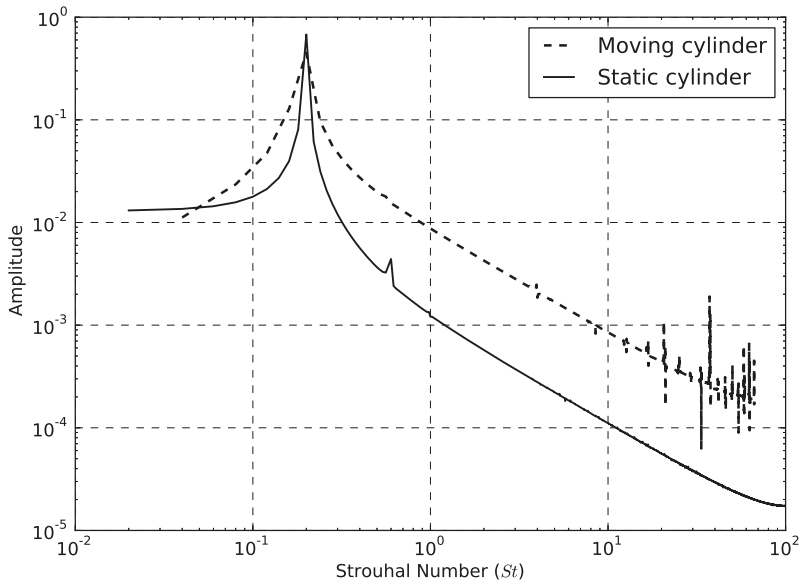

Figure 8: Amplitude of the FFT spectrum of the lift coefficient for the static and moving cylinder cases once the instabilities were fully developed.

Table 2: Comparison of the PISO-IB results to literature data, for C_D, C_L and St

Study	C_D	C_L	St
PISO-IB - Static cylinder	1.37 ± 0.05	± 0.69	0.200
PISO-IB - Moving cylinder	1.35 ± 0.1	± 0.51	0.200
Braza *et al.* [48]	1.40 ± 0.05	± 0.75	-
Choi *et al.* [50]	1.36 ± 0.048	± 0.64	0.191
Wright *et al.* [51]	1.33 ± 0.04	± 0.68	0.196
Bergmann *et al.* [52]	1.35	-	0.198
Russel and Wang *et al.* [53]	1.29 ± 0.022	± 0.50	0.195
Henderson *et al.* [54]	1.341	-	0.197
He *et al.* [55]	1.3560	-	0.198
Bhalla *et al.* [49]	1.39	-	0.200

5. Methodology for mixing experiments and simulations

A thorough verification of the proposed PISO-IB scheme was carried out in the previous section. The method is now validated in the context of mixing. This is accomplished by comparing its accuracy to that obtained with classical conformal mesh methods, namely the SRF and the SM techniques, by means of experimental data obtained in our lab.

The experimental set-up consisted of a 0.365 m diameter (T) flat-bottomed, cylindrical and transparent vessel equipped with a T/3 diameter pitched blade turbine (PBT). The tank was equipped with four removable baffles (W=T/10), and the clearance was set at a standard value of C=T/4. The height of the

liquid was fixed to that of the tank diameter (H=T). The dimensions of this mixing rig are shown in Figure 9 and their values are summarized in Table 5. The choice of a PBT is due to its widespread use in the chemical process industry. Furthermore, as the complexity of this impeller is such that it cannot be discretized with a structured mesh, it represents an excellent test case for the PISO-IB method.

The torque on the shaft was first measured for several single-phase experiments using a 0-5 N.m torque-meter provided by Ono Sukki, with a minimum resolution of 0.001 N.m and 0.2 % F.S. accuracy. These torque values were used to establish the dependency of the power number N_P on the Reynolds number Re, a relationship that can also be obtained with simulations. In the context of mixing, these two dimensionless number are defined as follows [1]:

$$N_p = \frac{P}{\rho N^3 D^5} = \frac{2\pi\Gamma}{\rho N^2 D^5} \tag{21}$$

$$Re = \frac{\rho D^2 N}{\mu} \tag{22}$$

with N the speed of the impeller, P the power consumption and Γ the torque on the impeller.

The liquid used consisted of Newtonian glucose solutions. Its viscosity was modified by varying the proportion of glucose and water within the tank. Four

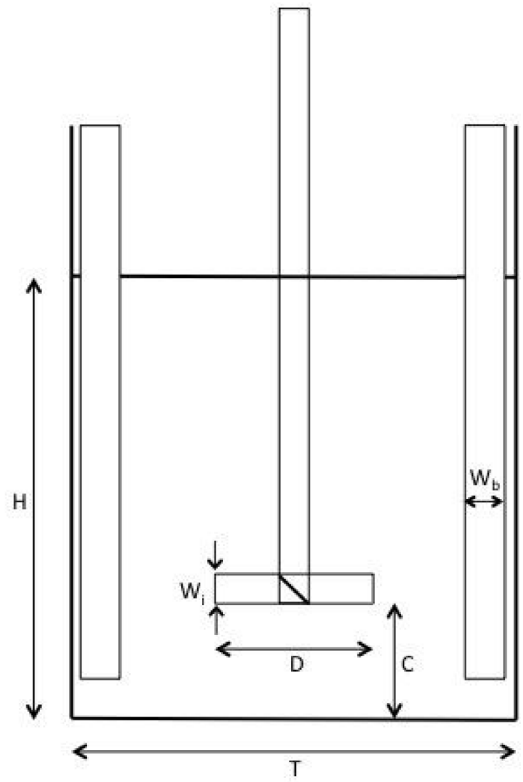

Figure 9: Scheme of the mixing rig used for the experiments and the simulations.

Table 3: Dimensions of the mixing rig

Symbol	**Name**	**Dimension**
T	Tank diameter	0.365m
D	Impeller diameter	T/3
H	Liquid level	T
C	Off-bottom clearance	T/4
W_i	Blade width	D/5
W_b	Baffle width	T/10

viscosities were considered: 32, 8.5, 4 and 0.84 Pa.s, at a temperature of 25°C. They were measured using an Anton Paar MCR 502 rheometer with a cone and plate geometry. Silicone oil was used to prevent evaporation in the rheometer. For all samples, the viscosity was found to be independent of the shear rate from 1 to 10 s^{-1}. Note that the viscosity of the solutions in the tank was found to be sensitive to temperature. An Arrhenius function was therefore used to obtain an expression for $\mu(T)$ and a thermocouple was installed in the tank to monitor the temperature of the solutions during the experiments. Using the fit for $\mu(T)$, the right process viscosity could be recovered from the temperature measurements. The underlying hypothesis is that the temperature was assumed

to be homogenous throughout the tank during the experiments.

For each experiment, the speed of the impeller was gradually increased with 30 RPM steps from 30 RPM to 800 RPM, which was the maximum velocity that could be reached without air entrapment or surface instabilities. The torque measurements for the impeller without load was removed from the raw values, and the power number resulting from these torque measurements was plotted agaisnt the Reynolds number to produce a power curve. The curves obtained were superimposed to sweep a large range of Re values and confirm the consistency of the torque values. Both baffled and unbaffled configurations of the tank were investigated. It must be noted that some bubbles were observed, mainly in the case of the 32 Pa.s solution, which were entrapped during the filling of the tank. Although larger bubbles were given sufficient time to exit the tank, a small amount of small bubbles could not be removed due to the relatively large viscosity of the solutions.

Other possible sources of uncertainty in the experiments are related to the torque and impeller speed measurements and the vibration of the shaft. Only the error on the torque measurements was taken into account and the other sources of uncertainty were considered negligible.

5.1. Simulation set-up

Simulations for the unbaffled tank baffles were carried out using the SM, SRF and PISO-IB methods. For the baffled configuration, only the SM and PISO-IB methods were investigated because the SRF model would not work in such a case. The simulations were carried out at a constant viscosity ($\mu = 0.5$ Pa.s) and at a large range of impeller speeds N = 1 ($Re = 0.75$), 2 (1.5), 5 (3.8), 10 (7.5), 25 (19), 50 (38), 100 (75), 150 (113), 200 (150), 250 (188) and 300 (225) RPM to capture both the laminar and the transitional regimes. More points were taken in the transitional region to achieve greater accuracy in capturing the non-linear character of N_p. For each method, the mesh was refined progressively until no significant differences ($< 1\%$) could be observed in the measured torque.

For the SRF and SM methods, a hexahedral background mesh of the tank was first generated with 33x88x60 (r,θ,z) cells. The impeller was then integrated to it using the snappyHexMesh utility of OpenFOAM which led to a combination of around 95% of hexahedra and 5% of polyhedra. In the case of the SM method, the same resulting mesh was further split into two regions, one encompassing the impeller and the other one complementing the computational domain. In all cases, the meshes were locally refined in the regions of the impeller and the baffles, yielding more than 210k cells.

In the case of the PISO-IB method, only the tank was meshed. The impellers and the baffles, when present, were taken into account using the IB forcing scheme described in Section 3. A mesh refinement procedure was applied using the dynamic mesh objects of OpenFOAM in order to refine the mesh in the volume swept by the impeller (Figure 10) and in the region of the baffles. Although this procedure increased the number of cells compared to the use of a dynamic mesh refinement technique, it was found to be much more efficient because it allowed for static memory allocation and efficient load balancing between the processors. The background hexahedral mesh consisted of 33x88x60 (r,θ,z) cells and, following the mesh refinement in the swept volume of the impeller and near the baffles, the final mesh contained more than 368k cells.

All simulations were unsteady. A centered scheme and a second-order Crank-Nicholson method were used for the space and time discretizations, respectively. For the SRF and the SM methods, a standard two-loop PISO scheme was used. For each simulation, a Courant-Friedrichs-Lewy stability condition $CFL = 0.5$ was considered. The simulation time was 40s, which was sufficient in all cases to reach convergence for the torque as well as for the flow patterns.

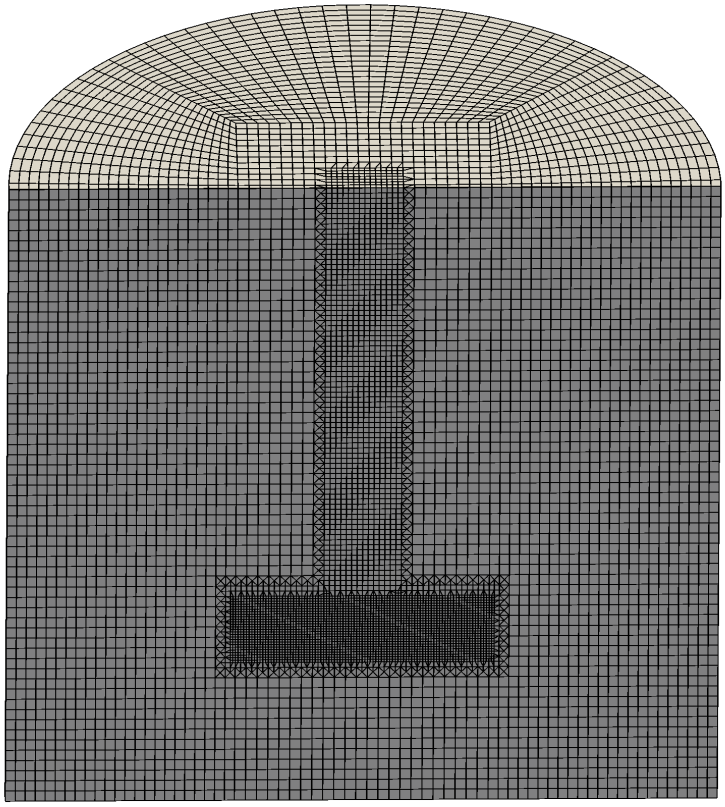

Figure 10: Slice of the hexahedral mesh with refinement in the volume swept by the impeller for the PISO-IB method

6. Results and discussion

Figure 11 present a slice of the axial velocity contours obtained with the SRF, SM and PISO-IB methods at $Re = 250$ for the case without baffles. Apart from minor differences among these contours, there is a good agreement between all three approaches. Comparisons (not shown here) for other planes and other components of the velocity field, for any of the values of Re considered (see Secton 5) led to the same conclusion.

The variation of the power number with respect to the Reynolds number is given in Figures 12 and 13 for unbaffled and baffled tanks, respectively. The simulations results are in good agreement with the experimental data and lie within the experimental error bars. In particular, one can note that the numerical models capture the early transitional regime ($Re \in [10, 100]$) with adequate accuracy. In fact, all three SRF, SM and IB methods give indistinguishable results, except for a slight deviation in the case of the SRF method at the highest Reynolds number investigated ($Re = 225$). This slight gap is most likely due to differences in the resolution of the large-scale unsteady structures that develop in the flow. Furthermore, we recall that in the laminar regime, $Np \propto Re^{-1}$ [1]. Using a linear regression on the data for which $Re \leq 5$, a slope of -1.0 was

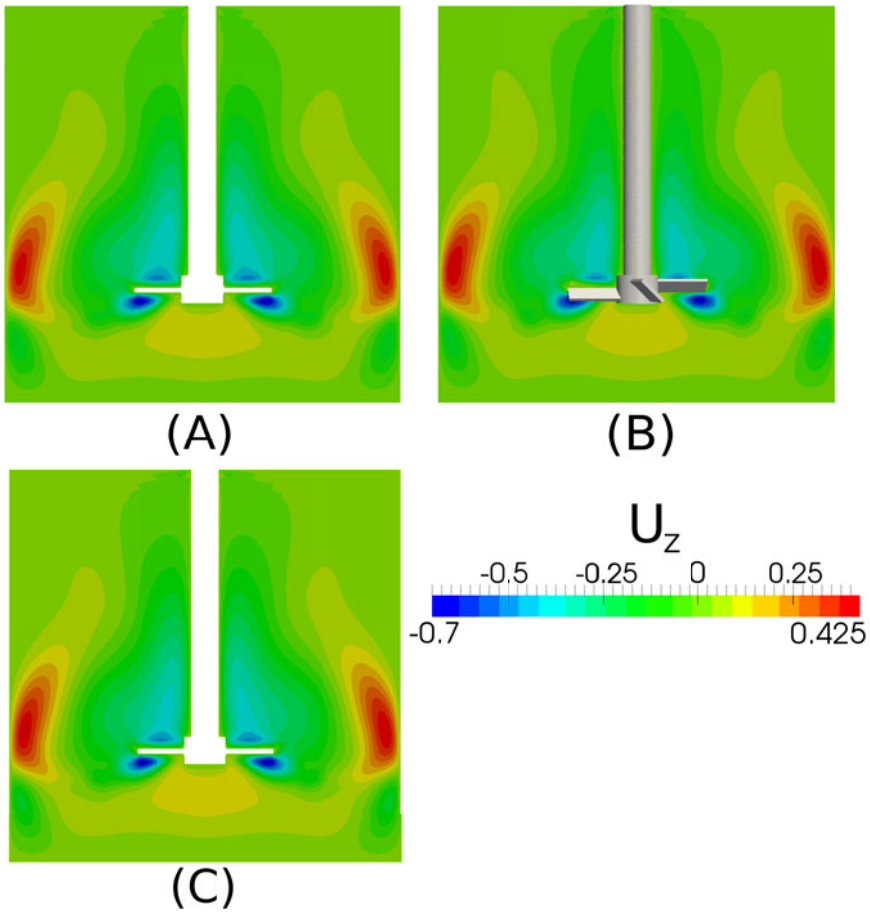

Figure 11: Axial velocity contours on a cross-section plane after $30s$ for $Re = 250$: (A) SRF technique, (B) PISO-IB technique and (C) SM technique

obtained with a R^2 coefficient equal to 0.9999 for all three methods.

The results obtained show that all three methods give qualitative (velocity profiles) and quantitative (power curves) results of comparable quality. The use of local mesh refinement to guarantee accurate results with the PISO-IB scheme add to its computational cost. Taking the SRF technique as a reference and considering meshes that yield the same level of accuracy for the torque, we found that the SM and the IB methods increased their computational cost by a factor of 1.2 and 2, respectively, when 6 Intel Westmere 2.67 Ghz processor cores were used. When 12 cores were used, this factor increased to 1.4 for the SM method, but remained 2 for the PISO-IB method. This indicates that, although the PISO-IB method is *a priori* more expensive, it has better parallel scaling properties than the SM technique.

The PISO-IB method also benefits from other advantages. First, the method does not require a body-fitted mesh, the generation of which can be time-consuming. Next, it can be used for geometries within which the volume swept by multiple impellers overlap. Secondly, contrary to the SM method, it entails the use of a static mesh. For solid-liquid simulations, this enables the fast detection of the mesh cells in which particles reside, a feature essential for an efficient CFD-DEM model such as the one developed by our group [56].

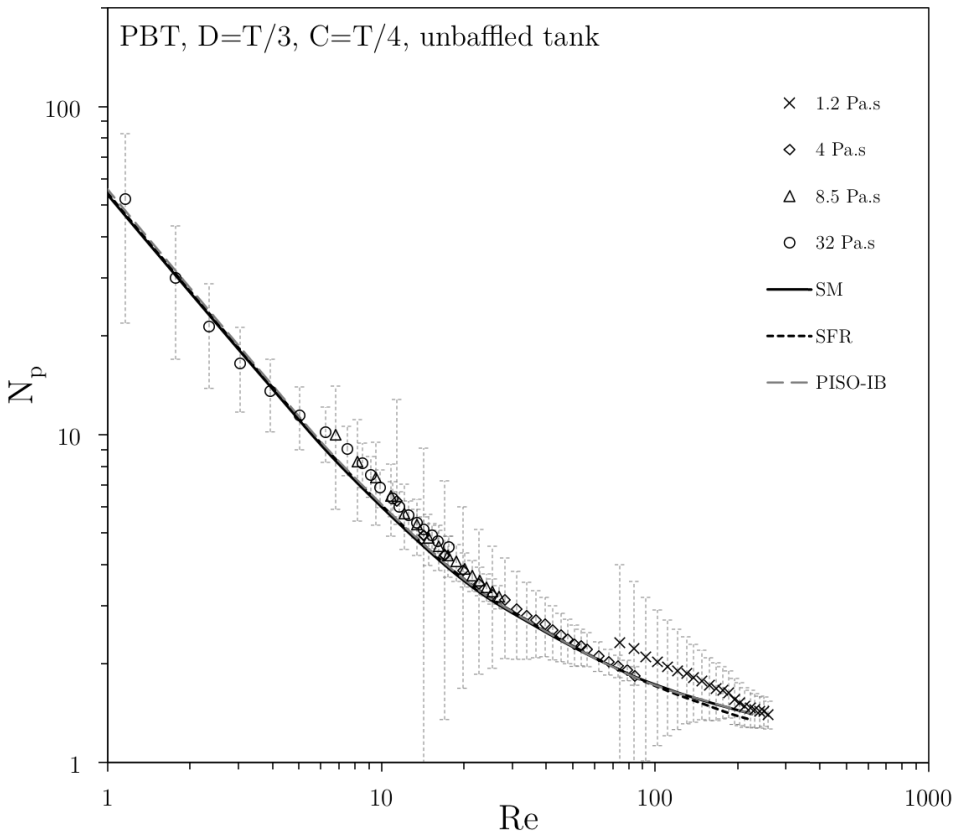

Figure 12: Power curve for the PBT in an unbaffled tank.

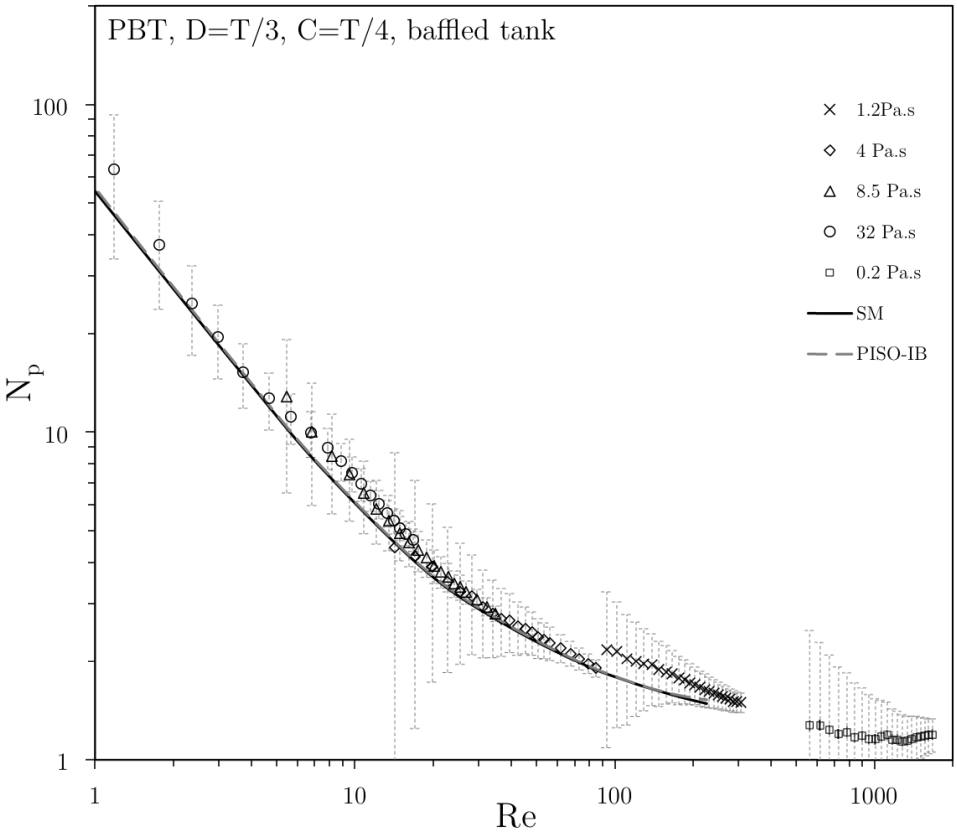

Figure 13: Power curve for the PBT in a baffled tank. Note the superposition of the PISO-IB and SM results.

7. Conclusion

The modeling of fluid flow in stirred tanks may still be challenging due to the impeller geometry and kinematics, the presence of baffles and configurations in which the volume swept by the mutiple impellers overlap. In the latter case, the use of an immersed boundary method is essential.

In this work, we introduced a novel semi-implicit immersed boundary (PISO-IB) method based on the addition of a forcing term to a PISO finite volume solver that is applicable to unstructured meshes. With this method, both static and dynamic mesh refinement strategies are tractable. Furthermore, this method works with a static mesh, which in the context of solid-liquid mixing enables the fast detection of particles in the cells of this mesh, an essential characteristic of CFD-DEM and the likes.

This method was implemented in the open-source CFDEM framework, which is based on Open∇FOAM for the liquid phase and LIGGGHTS for modeling the flow of particles when they are present. Our implementation was first verified by performing an order of convergence analysis using the Taylor-Couette test case and by comparing to literature data numerical results obtained for the flow around static and moving cylinders. Although a degradation of the order of

convergence of the method was observed, it was found that the method yields very good velocity profiles and can be used to accurately measure the forces acting on a moving body. The method was then validated in the context of single-phase mixing, more precisely in the case of a pitched blade turbine in baffled and unbaffled tanks. The torque measurements obtained via the PISO-IB method are in good agreement with those predicted by the SRF and the SM techniques as well as with experimental data, despite the challenge posed by the non-alignment of the moving pitched blade turbine with the mesh cells.

In future work, our immersed boundary method will be used in combination with the CFD-DEM model of the CFDEM framework to study solid-liquid mixing in baffled stirred tanks, including tanks provided with more complex configurations such as planetary mixers.

8. Acknowledgements

The financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) is gratefully acknowledged. In particular, Bruno Blais is thankful for the NSERC Vanier Scholarship. The authors would like to thank Gilles Lenfant for measuring the viscosity of the glucose solutions.

Computations were made on supercomputer Briaree from University of Mon-

treal, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), the Ministère de l'Économie, de l'Innovation et des Exportations du Qubec (MEIE), RMGA and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

References

[1] E. L. Paul, V. A. Atiemo-Obeng, S. M. Kresta, Handbook of industrial mixing - science and practice (2004).

[2] J. Derksen, H. E. A. Van den Akker, Large eddy simulations on the flow driven by a rushton turbine, AIChE Journal 45 (2) (1999) 209–221.

[3] H. E. Van den Akker, The details of turbulent mixing process and their simulation, Advances in Chemical Engineering 31 (2006) 151–229.

[4] V. Stobiac, L. Fradette, P. A. Tanguy, F. Bertrand, Pumping characterisation of the maxblend impeller for Newtonian and strongly non-Newtonian fluids, The Canadian Journal of Chemical Engineering 92 (4) (2014) 729–741.

[5] Y. Guntzburger, A. Fontaine, L. Fradette, F. Bertrand, An experimental

method to evaluate global pumping in a mixing system: Application to the Maxblend for Newtonian and non-Newtonian fluids, Chemical Engineering Journal 214 (2013) 394–406.

[6] M. B. Machado, K. J. Bittorf, V. T. Roussinova, S. M. Kresta, Transition from turbulent to transitional flow in the top half of a stirred tank, Chemical Engineering Science 98 (2013) 218–230.

[7] A. Tamburini, A. Cipollina, G. Micale, A. Brucato, M. Ciofalo, CFD simulations of dense solid-liquid suspensions in baffled stirred tanks: Prediction of suspension curves, Chemical Engineering Journal 178 (2011) 324–341.

[8] A. Tamburini, A. Cipollina, G. Micale, A. Brucato, M. Ciofalo, CFD simulations of dense solid-liquid suspensions in baffled stirred tanks: Prediction of the minimum impeller speed for complete suspension, Chemical Engineering Journal 193 (2012) 234–255.

[9] A. Tamburini, A. Cipollina, G. Micale, A. Brucato, M. Ciofalo, CFD simulations of dense solid–liquid suspensions in baffled stirred tanks: Prediction of solid particle distribution, Chemical Engineering Journal 223 (2013) 875–890.

[10] I. Ayranci, S. M. Kresta, Design rules for suspending concentrated mixtures

of solids in stirred tanks, Chemical Engineering Research & Design 89 (10A) (2011) 1961–1971.

[11] I. Ayranci, S. M. Kresta, J. J. Derksen, Experiments and simulations on bidisperse solids suspension in a mixing tank, Chemical Engineering & Technology 36 (11) (2013) 1957–1967.

[12] S. B. Ibrahim, A. W. Nienow, The effect of viscosity on mixing pattern and solid suspension in stirred vessels, Eighth European Conference on Mixing (136) (1994) 25–32.

[13] S. Ibrahim, A. W. Nienow, Comparing impeller performance for solid-suspension in the transitional flow regime with newtonian fluids, Chemical Engineering Research and Design 77 (8) (1999) 721–727.

[14] Z. Y. Zhou, S. B. Kuang, K. W. Chu, A. B. Yu, Discrete particle simulation of particle-fluid flow: model formulations and their applicability, Journal of Fluid Mechanics 661 (2010) 482–510.

[15] D. L. Marchisio, R. O. Fox, Computational models for polydisperse particulate and multiphase systems, Cambridge University Press, 2013.

[16] J. Luo, R. Issa, A. Gosman, Prediction of impeller induced flows in mix-

ing vessels using multiple frames of reference, in: Institution of Chemical Engineers Symposium Series, Vol. 136, Hemsphere publishing corporation, 1994, pp. 549–549.

[17] F. Bertrand, P. A. Tanguy, F. Thibault, A three-dimensional fictitious domain method for incompressible fluid flow problems, International Journal for Numerical Methods in Fluids 25 (6) (1997) 719–736.

[18] F. Thibault, P. Tanguy, Power-draw analysis of a coaxial mixer with Newtonian and non-Newtonian fluids in the laminar regime, Chemical Engineering Science 57 (18) (2002) 3861–3872.

[19] F. Barailler, M. Heniche, P. A. Tanguy, CFD analysis of a rotor-stator mixer with viscous fluids, Chemical Engineering Science 61 (9) (2006) 2888–2894.

[20] P. Tanguy, F. Thibault, C. Dubois, A. Ait-Kadi, Mixing hydrodynamics in a double planetary mixer, Chemical Engineering Research and Design 77 (4) (1999) 318–324.

[21] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier-Stokes equations, Journal of Computational Physics 59 (2) (1985) 308–323.

[22] J. H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, J Comput Phys 230 (19) (2011) 7347–7363.

[23] OpenCFD, OpenFOAM - The Open Source CFD Toolbox, URL : http://www.openfoam.com, 2014.

[24] CFDEM, CFDEM- Open Source CFD, DEM and CFD, URL : http://www.cfdem.com, 2014.

[25] C. Goniva, C. Kloss, N. G. Deen, J. A. M. Kuipers, S. Pirker, Influence of rolling friction on single spout fluidized bed simulation, Particuology 10 (5) (2012) 582–591.

[26] A. Brucato, M. Ciofalo, F. Grisafi, G. Micale, Numerical prediction of flow fields in baffled stirred vessels: A comparison of alternative modelling approaches, Chemical Engineering Science 53 (21) (1998) 3653–3684.

[27] R. B. Bird, W. E. Stewart, E. N. Lightfoot, Transport Phenomena, second edition Edition, John Wiley & Sons, 2007.

[28] Ansys Fluent, 12.0 theory guide, Ansys Inc (2009).

[29] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[30] R. Mittal, G. Iaccarino, Immersed boundary methods, Annual Review of Fluid Mechanics 37 (1) (2005) 239–261.

[31] C. S. Peskin, Flow patterns around heart valves: a numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[32] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, Journal of Computational Physics 105 (2) (1993) 354–366.

[33] P. Angot, C.-H. Bruneau, P. Fabrie, A penalization method to take into account obstacles in incompressible viscous flows, Numerische Mathematik 81 (4) (1999) 497–520.

[34] K. Khadra, P. Angot, S. Parneix, J. Caltagirone, Fictitious domain approach for numerical modelling of Navier-Stokes equations, International Journal for Numerical Methods in Fluids 34 (8) (2000) 651–684.

[35] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, A distributed la-

grange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow 25 (5) (1999) 755–794.

[36] N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, T.-W. Pan, A new formulation of the distributed lagrange multiplier/fictitious domain method for particulate flows, International Journal of Multiphase Flow 26 (9) (2000) 1509–1524.

[37] A. A. Shirgaonkar, M. A. MacIver, N. A. Patankar, A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion, Journal of Computational Physics 228 (7) (2009) 2366–2390.

[38] O. M. Curet, I. K. AlAli, M. A. MacIver, N. A. Patankar, A versatile implicit iterative approach for fully resolved simulation of self-propulsion, Computer Methods in Applied Mechanics and Engineering 199 (37) (2010) 2417–2424.

[39] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, Journal of Computational Physics 161 (1) (2000) 35–60.

[40] S. Majumdar, G. Iaccarino, P. Durbin, RANS solvers with adaptive struc-

tured boundary non-conforming grids, Annual Research Briefs, Center for Turbulence Research, Stanford University (2001) 353–466.

[41] H. Udaykumar, W. Shyy, M. Rao, Elafint: a mixed Eulerian-Lagrangian method for fluid flows with complex and moving boundaries, International journal for numerical methods in fluids 22 (8) (1996) 691–712.

[42] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational physics 62 (1) (1986) 40–65.

[43] J. H. Ferziger, M. Perić, Computational methods for fluid dynamics, Vol. 3, Springer Berlin, 1996.

[44] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in Physics 12 (6) (1998) 620–631.

[45] H. Jasak, A. Jemcov, Z. Tukovic, OpenFOAM: A C++ library for complex physics simulations, in: International workshop on coupled methods in numerical dynamics, pp. 1–20.

[46] C. Rhie, W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, Aiaa Journal 21 (11) (1983) 1525–1532.

[47] F. P. Kärrholm, Rhie-Chow interpolation in OpenFOAM, Appendix from Numerical Modelling of Diesel Spray Injection and Turbulence Interaction at Chalmers University. http://www. tfd. chalmers. se/hani/kurser/OS CFD.

[48] M. Braza, P. Chassaing, H. H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, Journal of fluid mechanics 165 (1986) 79–130.

[49] A. P. S. Bhalla, R. Bale, B. E. Griffith, N. A. Patankar, A unified mathematical framework and an adaptive numerical method for fluid-structure interaction with rigid, deforming, and elastic bodies, J. Comput. Physics 250 (2013) 446–476.

[50] J.-I. Choi, R. C. Oberoi, J. R. Edwards, J. A. Rosati, An immersed boundary method for complex incompressible flows, Journal of Computational Physics 224 (2) (2007) 757–784.

[51] J. A. Wright, R. W. Smith, An edge-based method for the incompressible NavierStokes equations on polygonal meshes, Journal of Computational Physics 169 (1) (2001) 24–43.

[52] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, Journal of Computational Physics 230 (2) (2011) 329–348.

[53] D. Russell, Z. Jane Wang, A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, Journal of Computational Physics 191 (1) (2003) 177–205.

[54] R. D. Henderson, Nonlinear dynamics and pattern formation in turbulent wake transition, Journal of Fluid Mechanics 352 (1997) 65–112.

[55] J. W. He, R. Glowinski, R. Metcalfe, A. Nordlander, J. Periaux, Active control and drag optimization for flow past a circular cylinder, Journal of Computational Physics 163 (1) (2000) 83–117.

[56] B. Blais, M. Lassaigne, C. Goniva, L. Fradette, F. Bertrand, On the extension of CFD-DEM to viscous suspensions and its application to solid-liquid mixing, in preparation.