



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/17886>

To cite this version :

Mathieu SPECKLIN, Y. DELAURÉ - A sharp immersed boundary method based on penalization and its application to moving boundaries and turbulent rotating flows - European Journal of Mechanics - B/Fluids - Vol. 70, p.130-147 - 2018

Any correspondence concerning this service should be sent to the repository

Administrator : scienceouverte@ensam.eu



A sharp immersed boundary method based on penalization and its application to moving boundaries and turbulent rotating flows

M. Specklin^{a,b,*}, Y. Delauré^a

^a School of Mechanical and Manufacturing Engineering, Dublin City University, Glasnevin, Dublin, Ireland

^b Sulzer Pump Solutions Ireland Ltd., Wexford, Ireland

A B S T R A C T

This paper presents an Immersed Boundary Method (IBM) for handling flows in the presence of fixed and moving solids with complex geometries. The method is based on a penalization approach and designed to preserve the sharpness of the immersed boundaries. Corrections of the boundary conditions are implemented at the interface to improve the accuracy of the solution in comparison to first-order methods and avoid the rasterization issue on Cartesian grids.

The current IBM is developed in the OpenFOAM[®] library (-v 2.2) and its accuracy is first verified against the Wannier flow case. It is then applied to the flow in presence of fixed and moving circular obstacles. The computational results show good agreement with equivalent standard body-conforming simulations and other high order published IBM, and demonstrate as well that improvements can be achieved by correcting the boundary conditions for both velocity and pressure on the interface. Finally, the method is assessed by reference to a realistic engineering application involving rotating flow: a single-phase mixer. In this case, the method is coupled to a DES model for turbulence modeling, and results show again good comparison with experimental results.

1. Introduction

Although the accuracy of CFD methods handling complex moving geometries has increased in recent years, with new high order sliding mesh methods [1–3] for instance, or Arbitrary Lagrangian–Eulerian (ALE) techniques for Fluid–Structure–Interaction (FSI) [4–8], meshing complex realistic engineering systems remains a significant challenge in a wide range of applications. The additional mesh constraints imposed by sliding interfaces can make it especially difficult to achieve good quality refined mesh in particular in compact designs where thin gaps between moving boundaries are not uncommon. Immersed Boundary Methods (IBMs) have shown to be good alternatives for a broad range of problems. Their aim is to solve a single set of conservation equations for several physical components that can be gas, liquids or more commonly solids, without the need for a mesh which conform to the inter-phase boundaries. As a result, IBMs make it possible to handle multi-phase flow problems and fluid–structure interaction problems, both in a single-phase framework. One of the main interests of IBM lies thus in the simplification of the model. Of particular importance is the method’s suitability for use with simple Cartesian grids

which can guarantee second order accuracy and improve stability of the numerical solution. This strongly reduces the time required to setup the computational case, which can be significant with complex and moving geometries. Special care must then be taken to obtain the correct boundary conditions at the exact position of the interface, and to reproduce the effects of the real geometry. The imposition of the boundary condition is a key point of the methods and it is also what differentiate IBMs from each others.

The first documented IBM was implemented by Peskin [9] to model the interaction of the blood flow with an heart valve, modeled as non-conforming elastic solid. For this method, the effect of the solid is smeared on the cells in the vicinity of the interface, by introducing a fictitious force in the momentum equation, combined with a well chosen discretized Dirac delta function. The forcing term and solid velocity are calculated from constitutive law, as the Hook’s law if the immersed surface exhibits elastic deformation. Approaches of this type are named continuous forcing approach, as the Immersed Boundary (IB) force is added before the discretization of the equations. Extensions of this kind of methods have been proposed to deal with solid boundaries in the rigid limit. In this case, the forcing term can be modeled for example as a spring restoring force (see work of Lai and Peskin [10]), or can be obtained in a feedback way thanks to an already known solid velocity as proposed by Goldstein et al. [11]. However, these approaches are subject to stiffness problems in the limit of rigid

* Corresponding author at: School of Mechanical and Manufacturing Engineering, Dublin City University, Glasnevin, Dublin, Ireland.
E-mail address: mathieu@specklin.ie (M. Specklin).

solids, due to the high gradients of the forcing term. The Dirac delta function, used to map the variables from the fluid Cartesian grid to the solid Lagrangian grid, has the effect of blurring the location of the solid interface. High spatial resolution is thus required in order to retrieve a sharp representation of the interface.

In the same spirit of the method of Goldstein et al., the approach of Angot et al. [12] uses a forcing term based on the Brinkman equation for porous media, to obtain the desired no-slip velocity at the solid interface. This type of method belongs to the class of penalization methods, and has been widely used for immersed boundary problems, even with complex geometries as in fish-like swimming applications ([13,14] and [15]). However, most of published methods do not reach second order accuracy in space. To the author's knowledge, the Sub-Mesh Penalty Method of Sarthou et al. [16], the second order Penalty model of Introïni et al. [17], and the work of Chantalat et al. [18] are the only models based on penalization which offer improvements to tackle the rasterization effect on Cartesian grids. In addition, the active zone for the forcing term is generally also diffused by the use of mask functions combined with Heaviside functions, as it is the case in [19] and in [14]. A different type of penalization method has been proposed by Vincent et al. [20], where the stress tensor is penalized instead of the velocity, and has been applied to particle-laden flows. Concerning the application areas, penalization methods have been used for compressible flows ([21] and [22]), combined with level-set methods for interface tracking problems ([13,18] and [14]), used for fluid–structure interactions problems as in [23] or [24], or again used for highly turbulent flows ([13] and [25]).

The IBM presented in this article is based on a penalization approach as well, and is implemented in the Open Source solver library OpenFOAM[®]. Two penalization methods have already been implemented on this platform. Firstly, in order to study wettability conditions, the first order model of Horgue et al. [26] was combined to the Volume-Of-Fluid method. Secondly, Blais et al. [27] combined the pressure implicit with splitting of operators (PISO) scheme of OpenFOAM[®] with an IBM formulation based on volume correction. Although with a similar level of error, the latter method degrades the order from 2 to 1.33 in comparison to the standard PISO scheme on body-fitted mesh. Our penalization technique proposes to address both weaknesses of diffusion and low order of accuracy. The forcing term employed in the model is active strictly only inside the immersed solid, and does not use smearing functions. This feature allows a sharp representation of the geometry and its interface. In order to enhance the order of accuracy, the velocity is corrected inside the solid and in the vicinity of the interface, with a linear interpolation. Furthermore, the pressure is penalized on the same cells in order to avoid a spurious contribution to the pressure gradient in the momentum predictor and to the solution of the pressure equation.

In Section 2, the formulation of the present penalization approach is developed, including the corrections implemented for the velocity and the pressure near the interface. Particular attention is paid to (i) the estimation of the force exerted by the fluid on the solid, used for the calculation of the drag for instance, and (ii) the combination of the IBM with a DES turbulence model (DES-IBM), involving a specific treatment for the turbulent viscosity. The accuracy of the model is assessed by reference to the Wannier case, which is presented in Section 3, along with three other test cases involving moving and complex boundaries in laminar and turbulent rotating flows. For the turbulent case, a power-law based method for correcting the velocity at the immersed surface is introduced. This approach complies with the velocity profile of an attached boundary layer. The sensitivity of the results to the two types of velocity reconstruction is then discussed. The efficiency of the present IBM is demonstrated and compared to equivalent body-fitted mesh simulations. Finally, Section 4 summarizes and concludes the current formulation.

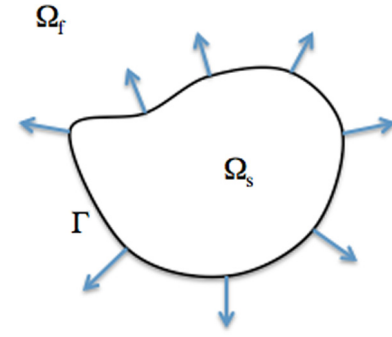


Fig. 1. Sketch of the different domains.

2. Numerical formulation

2.1. Immersed boundary formulation

Consider a rigid solid represented by Ω_s , immersed in a fluid domain Ω_f , as illustrated in Fig. 1. Γ is the interface between both domains. In the IBM presented in this study, the Navier–Stokes equations are solved in the entire domain $\Omega = \Omega_f \cup \Omega_s$.

2.1.1. Treatment of the solid geometry

The interface of the immersed solid Γ is represented by a discrete mesh which may be made of triangular or other polyhedral faces. The mesh is characterized by a set of Lagrangian points $x_{ib,n}$, with n from 0 to N , N being the number of solid points. These points represent the face centers of the surface mesh. The latter is used to identify the cell centers inside the solid domain from those inside the fluid domain, thanks to an octree partition of space.

2.1.2. Governing equations

An isothermal and incompressible flow is governed by the following equations of mass and momentum conservation:

$$\nabla \cdot \mathbf{u} = 0 \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nu \nabla^2 \mathbf{u} + \mathbf{f} = -\nabla p \quad (2)$$

where \mathbf{u} is the velocity, p the pressure, ν the kinematic viscosity and t the time. In Eq. (2), \mathbf{f} is the forcing term, or immersed boundary force, which is required to impose the correct boundary condition for the velocity at the interface with the immersed solid. This force is here strictly only active inside the solid, and its formulation is discussed below.

2.1.3. Penalization of the velocity

The immersed boundary method presented in this paper is based on the Penalty method, first introduced by Arquis [28]. In this approach, the immersed solid is considered as a porous media, with a very small permeability $0 < K \ll 1$. Hence the forcing term of Eq. (2) can be written in the following form:

$$\mathbf{f} = \chi \frac{\nu}{K} (\mathbf{u} - \mathbf{u}_{ib}) \quad (3)$$

where \mathbf{u}_{ib} is the velocity of the Immersed Boundary (IB) and χ is a characteristic function, which makes possible the localization of the solid. This function is defined in the equation below:

$$\chi(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_s(t) \\ 0 & \text{if } \mathbf{x} \in \Omega_f(t) \end{cases} \quad (4)$$

The forcing term is then not enabled in the fluid domain, and the original momentum equation is thus obtained. Angot et al. have

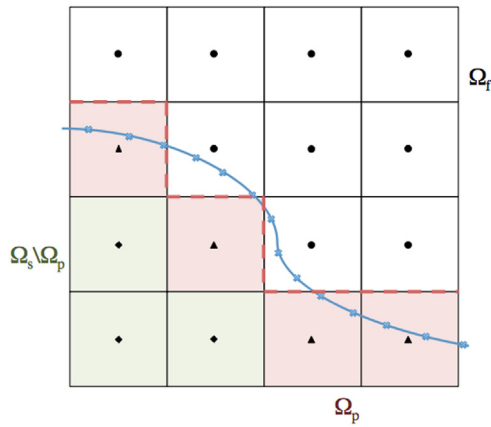


Fig. 2. Sketch of the discrete domains. The discrete solid domain is delimited by the dashed red line. • are the cell centers in the fluid domain Ω_f . ◆ are the cell centers in the solid domain, except the penalized domain, $\Omega_s \setminus \Omega_p$, in green, and ▲ are the cell centers in the penalized domain in red. × are the solid points (in blue). The grids considered in this study are made of cubic cells and uses the collocated storage of OpenFOAM®.

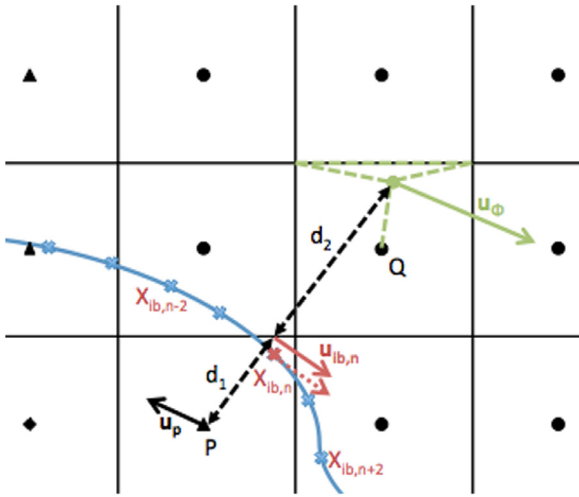


Fig. 3. Sketch of the velocity correction scheme on a 2D Cartesian mesh. The same nomenclature is used for the domains definition.

shown in [12] that this formulation is equivalent to the standard incompressible Navier–Stokes equations with a Dirichlet condition at the interface Γ for the velocity, and the authors have presented rigorous estimation of the errors introduced by this formulation as well as convergence theorems. In the cases considered in this document, the velocity of the immersed body \mathbf{u}_{ib} in Eq. (3) is always a geometrical velocity which is known and time-dependent.

2.1.4. Correction scheme for the velocity near the interface

For Cartesian grids, the forcing term presented in Eq. (3) leads to a stair-case definition of the solid geometry as shown in Fig. 2. Furthermore the cell centers, where the solid velocity is imposed, do not necessarily coincide with the solids points representing the immersed body. Two main errors are introduced with a simple Penalty approach. Firstly, the interface orientation is not accounted for, and secondly, the Dirichlet condition is not necessarily satisfied at the exact position of the interface.

In order to address these issues, Chantalat et al. [18] have proposed for elliptic problems an iterative correction of the normal derivative of the solution at the interface. A more intuitive approach is to impose a corrected velocity in the cells near the

interface, chosen to ensure that the Dirichlet condition $\mathbf{u} = \mathbf{u}_{ib}$ is satisfied at the exact interface position. Practically, this means that the corrected velocity is a function of both the solid velocity and the fluid velocity in the close neighboring area. In the Sub-Mesh Penalty Method of Sarthou et al. [16], this corrected velocity is obtained by linear interpolation between the solid and the fluid velocities. In [17] the corrected velocity is evaluated from an averaged reconstruction of the velocity gradient for the fluid contribution, and by means of a minimization problem for the solid contribution. The IB force is then scaled with the solid characteristic function or the cell volume ratio, and the modified Navier–Stokes equations are solved with a fractional-step method, which takes into account the forcing term in the correction step as well, as suggested by [29] for a consistent imposition of the boundary condition at the interface.

Similarly to the work of Sarthou et al., the correction scheme considered here is based on an interpolation of the velocity imposed in the vicinity of the fluid–solid interface. In this context, the forcing term introduced in Eq. (3) is re-defined as:

$$\mathbf{f}^* = \chi \frac{\nu}{K} (\mathbf{u} - \mathbf{u}_{ib}) + \chi^* \frac{\nu}{K} (\mathbf{u} - \mathbf{u}_p^*) \quad (5)$$

where χ^* stands for the characteristic function which represents an internal layer of cells, and where the modified velocity is applied \mathbf{u}_p^* . The new characteristic function defines a domain Ω_p , in red in Fig. 2, named here as the penalized domain and used in Eq. (6):

$$\chi^*(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_p(t) \\ 0 & \text{else.} \end{cases} \quad (6)$$

More precisely, each cell in Ω_s possessing at least one neighbor in Ω_f is included in Ω_p . In addition, the support of the characteristic function introduced in Eq. (4) for the imposition of non-modified solid velocity is changed from Ω_s to $\Omega_s \setminus \Omega_p$. It is important to note that the momentum forcing is enabled everywhere inside the solid domain, either to impose the interpolated velocity or the solid velocity. Thereby, the implementation of the correction scheme for the velocity does not introduce a reversed fictitious velocity inside the IB, as it is the case for IBMs in which the unmodified Navier–Stokes equations are solved inside the IB [30]. The imposed velocity \mathbf{u}_p^* in a cell of Ω_p is calculated from a linear extrapolation, using the closest solid point of the interface and a virtual point in the fluid domain on the normal direction, as described in Eq. (7):

$$\begin{aligned} \mathbf{u}_p^* &= \frac{d_1 + d_2}{d_2} \mathbf{u}_{ib,n} - \frac{d_1}{d_2} \mathbf{u}_\phi \\ &= \frac{d_1 + d_2}{d_2} \mathbf{u}_{ib,n} - \frac{d_1}{d_2} \left(\sum_k \alpha_k \mathbf{u}_{q,k} + \beta \mathbf{u}_q \right). \end{aligned} \quad (7)$$

In the above equation, d_1 and d_2 represent respectively the approximate distance from the cell center P to the interface, and the distance from the interface to a virtual point ϕ , along the interface normal. The procedure is illustrated in Fig. 3. Similar correction schemes have been employed for instance in the IBM of Tseng and Ferziger [31] and in the Penalty method of Sarthou et al. [16]. Two types of case studies will be discussed in the present manuscript. The first considers academic problems involving immersed geometries which can be defined analytically. In this case the distance to the interface can be determined exactly and the Dirichlet condition $\mathbf{u} = \mathbf{u}_{ib,n}$ is thus satisfied at the exact interface position. The second considers more realistic engineering systems with relatively complex immersed boundaries. In which case determining the exact interface position is not possible. Two options have been considered: estimate the position (i) with a dichotomy algorithm or (ii) simply using the closest solid point $\mathbf{x}_{ib,n}$. The former iterative approach can converge reasonably quickly but the increase in the computational is not negligible with high surface refinement. The error introduced in this case can be significantly

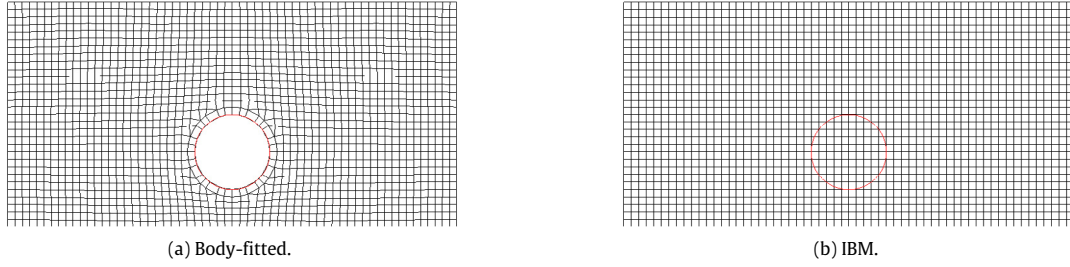


Fig. 4. Sketch of the grids used for the 2D Wannier test case, with 10 cells in the cylinder diameter for the IBM, and the equivalent body-conforming mesh. The surface mesh of the cylinder is represented in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

reduced with a sufficient refinement of the solid surface mesh, and was found negligible in the test cases considered in this work. This result is discussed in Section 3.4.2. It is important to note however that this discrepancy may prevent the model of achieving a formal second order accuracy in realistic application. Regarding the accuracy of the solution as a function of the parameters d_1 and d_2 , it was verified for the Wannier case (presented in Section 3.1) that $d_2 = d_1$ gives the best results. Finally, the velocity at the virtual point \mathbf{u}_ϕ is also obtained thanks to a linear interpolation of the velocity at the center of the cell Q , holding the virtual point, and velocities at different points k on the closest face of the considered cell. Thereby, α and β in Eq. (7) are the coefficients of the linear combination. The velocities at the face corners $\mathbf{u}_{i,k}$ are again estimated through interpolation from the neighboring cells.

2.1.5. Estimation of the immersed boundary force

The physical force \mathbf{F} exerted by the fluid on the solid can be decomposed into pressure and viscous terms:

$$\mathbf{F} = - \iint_{\Gamma} p \mathbf{dS} + \mu \iint_{\Gamma} (\mathbf{T} \cdot \mathbf{n}) \mathbf{dS} \quad (8)$$

where \mathbf{dS} is a small element of surface on the immersed boundary interface Γ , and \mathbf{n} and \mathbf{T} represent respectively the normal at the solid interface and the strain rate tensor. The estimation of this force requires additional and delicate computations. Indeed, as the surface of the immersed body does not necessarily coincide with the faces of the Eulerian grid, a reconstruction of the variables at the Lagrangian solid points is usually implemented [32,33]. In the present study, a different approach is adopted, which is based on the Gauss's theorem. Applying the latter to a domain $\tilde{\Omega}$ surrounding the immersed body allows to rewrite Eq. (8) as:

$$\mathbf{F} = - \iiint_{\tilde{\Omega}} \nabla p \mathbf{d}\tilde{\Omega} + \iint_{\tilde{\Gamma}} p \mathbf{dS} + \mu \iiint_{\tilde{\Omega}} \nabla^2 \mathbf{u} \mathbf{d}\tilde{\Omega} - \mu \iint_{\tilde{\Gamma}} (\mathbf{T} \cdot \mathbf{n}) \mathbf{dS} \quad (9)$$

where $\tilde{\Gamma}$ represents the boundaries of the domain $\tilde{\Omega}$, excluding the immersed interface Γ . With a relevant choice of $\tilde{\Omega}$, Eq. (9) can be discretized easily, in accordance with the considered mesh, in order to estimate the force applied by the fluid. Furthermore, the volume integral terms can also be replaced by the transient and convective terms thanks to the Navier–Stokes equations, as described in Eq. (10):

$$\mathbf{F} = \rho \iiint_{\tilde{\Omega}} \frac{\partial \mathbf{u}}{\partial t} \mathbf{d}\tilde{\Omega} + \iint_{\tilde{\Gamma}} p \mathbf{dS} + \rho \iiint_{\tilde{\Omega}} \nabla \cdot (\mathbf{u}\mathbf{u}) \mathbf{d}\tilde{\Omega} - \mu \iint_{\tilde{\Gamma}} (\mathbf{T} \cdot \mathbf{n}) \mathbf{dS}. \quad (10)$$

2.2. Discretization techniques and solution solver

The discretization procedure for the Navier–Stokes equations are based on the work of Jasak [34], following a Finite Volume formulation. All the variables are stored at the cell centers, where a non-staggered grid arrangement is used. The procedure of Jasak uses a predictor–corrector algorithm for the pressure–velocity coupling (PISO scheme), whose formulation is detailed below with combination to a pressure correction accounting for the immersed boundary presence. The equation are discretized in time with an Euler implicit scheme, and using second order schemes in space.

2.2.1. Momentum predictor

The first step of the method consists in estimating an intermediate velocity field, thanks to a semi-discretized momentum equation:

$$a_i \tilde{\mathbf{u}}_i = \mathbf{H}(\tilde{\mathbf{u}}) - \nabla p^n. \quad (11)$$

The predictor for the velocity is calculated from the pressure at the previous known time step n . In the matrix system (11), $\tilde{\mathbf{u}}_i$ represents the numerical estimation of the intermediate velocity at the center of the i th cell, and a_i is the diagonal matrix coefficient for the cell i . The term $\mathbf{H}(\tilde{\mathbf{u}})$, includes the contribution from neighboring cells, the explicit part of the transient term, and all other source terms. The forcing term f introduced in Eq. (2) includes an implicit contribution, involving the fluid velocity, and an explicit contribution, involving the solid or corrected velocity. The implicit part thus modifies directly the diagonal matrix coefficients a_i , while the explicit part is included in the term $\mathbf{H}(\tilde{\mathbf{u}})$. This term finally reads for a cell inside the solid domain:

$$\mathbf{H}(\tilde{\mathbf{u}}) = - \sum_j a_j \tilde{\mathbf{u}}_j + \frac{\mathbf{u}_i^n}{\Delta t} - \frac{\mathbf{u}_{ib}}{K}. \quad (12)$$

If one considers a cell inside the domain Ω_p , the solid velocity \mathbf{u}_{ib} is replaced by the corrected velocity \mathbf{u}_p^* in Eq. (12). In the latter, j stands for the neighboring cells, n represents the previous iteration time step and Δt is the value of the time step. The contribution of the neighboring cells for the term $\mathbf{H}(\tilde{\mathbf{u}})$ is obtained by a linearization of the quadratic convection term, which implies that a flux satisfying the mass conservation equation at time n is used to calculate the matrix coefficients a_i and a_j .

2.2.2. Pressure solution and PISO loop

The second step of the discretization procedure is to solve the pressure equation. This equation is derived from the momentum equation (11), by imposing mass conservation ($\nabla \cdot \mathbf{u} = 0$). Gradient and divergence terms are discretized using the Gauss theorem, providing an equation for pressure:

$$\sum_f \mathbf{S}_f \cdot \left[\left(\frac{1}{a_i} \right)_f (\nabla p^m)_f \right] = \sum_f \mathbf{S}_f \cdot \left(\frac{\mathbf{H}(\mathbf{u}^m)}{a_i} \right)_f \quad (13)$$

where \mathbf{S}_f is the outward normal surface vector for the face f of the i th cell. The resulting matrix for pressure is solved with an iterative method. One can note that the pressure does not depend directly on the intermediate velocity field, but only on its contribution through the term $\mathbf{H}(\mathbf{u}^m)$. Here, m stands for the increment of the PISO loop, whose initial conditions are given by $\mathbf{u}^m = \tilde{\mathbf{u}}$ and $p^m = p^n$ at the initial iteration $m = 0$.

The new face flux and cell center velocity are calculated from the new pressure field. Eq. (14) gives the new conservative fluxes, which will be used to build the matrix system for the momentum predictor at the next time step. The velocity is corrected explicitly with Eq. (15). The value at the faces are obtained using interpolation techniques.

$$F^{m+1} = \mathbf{S}_f \cdot \mathbf{u}_f^{m+1} = \mathbf{S}_f \cdot \left[\left(\frac{\mathbf{H}(\mathbf{u}^m)}{a_i} \right)_f - \left(\frac{1}{a_i} \right)_f (\nabla p^m)_f \right] \quad (14)$$

$$\mathbf{u}_i^{m+1} = \begin{cases} \frac{\mathbf{H}(\mathbf{u}^m)}{a_i} & \text{if } i \in \Omega_s \cup \Omega_p \\ \frac{\mathbf{H}(\mathbf{u}^m)}{a_i} - \frac{1}{a_i} \nabla p^m & \text{else.} \end{cases} \quad (15)$$

The velocity correction step, defined by Eq. (15) to account for the corrected pressure field, is modified in the solid and the penalized domains. As the intermediate velocity estimated from the momentum predictor step does not satisfy the boundary condition determined by the momentum forcing, this velocity correction step is not necessary in the latter domains. Because the velocity correction step is explicit, several iterations of pressure solutions and explicit velocity corrections are necessary (usually two or three) to take into account the transported influence of the corrections of neighboring velocities. After solving the velocity correction step for \mathbf{u}^{m+1} , the term $\mathbf{H}(\mathbf{u}^m)$ is recalculated and a new increment is performed as $m \rightarrow m + 1$. Once a predetermined convergence criterion is achieved, velocity and pressure are updated for the next time step with $\mathbf{u}^{n+1} = \mathbf{u}^{m+1}$ and $p^{n+1} = p^m$.

Finally, it is worth noting that the Rhie–Chow discretization procedure is adopted to avoid checker-boarding on the collocated grid [35]. The cell face fluxes are interpolated from the momentum equation (Eq. (14)) and then used to derive the pressure correction equation (Eq. (13)) from the continuity equation.

2.2.3. Boundary condition for pressure at the immersed interface

For continuous IBM with diffuse interface, no pressure boundary condition is usually needed on the immersed boundary [36], as the influence of the solid is smeared over several layers of cells. In the current sharp interface method however, the effect of the solid interface on local mass conservation must be taken into account in the solution of Eq. (13), for the pressure to be correct at the IB. The error introduced by a non-conservation of mass locally is furthermore accentuated by the mirrored velocities obtained from interpolation (see Section 2.1.4) and generates an un-physical pressure field near the IB. This issue has been considered in details by Kang et al. [37] for a discrete IBM, where a decoupling constraint for the pressure is proposed, and combined with a compatible interpolated velocity boundary condition related to mass conservation. Kim et al. [38] developed a method introducing a mass source/sink in addition to the momentum forcing, in order to satisfy the continuity in the cells containing the immersed boundary.

In the current approach, the pressure is corrected directly at the interface so that the local mass conservation error does not propagate into the fluid domain from the solid domain. Without correction, the spurious pressure in the domain Ω_p may influence the pressure solution in the neighboring fluid cells. Indeed in those fluid cells, both the evaluation of the pressure gradient in the momentum predictor and the discretization of the Laplacian term in the pressure solution step, involve the pressure in Ω_p . As

OpenFOAM[®] uses a small computational molecule, strictly only the cells inside the latter domain can influence the pressure in the fluid. Therefore, a correction of the pressure computed during the pressure equation step is adopted in the penalized domain. Namely, for a cell P of Ω_p the lower and upper triangles of the matrix build from Eq. (13), representing the contribution from neighboring cells, are canceled, while the source term is modified in order for the pressure to be imposed at the desired value. The latter value is chosen to be equal to the pressure at a virtual point along the interface normal \mathbf{n} , in accordance to a Neumann boundary condition ($\frac{\partial p}{\partial \mathbf{n}}|_f = 0$). The pressure at the virtual point p_ϕ is obtained by linear interpolation, similarly to the velocity corrections introduced earlier:

$$p_p = p_\phi = \sum_k \alpha_k p_k + \beta p_q. \quad (16)$$

2.3. Turbulence modeling

From an engineering point of view, a RANS model for turbulence can be sufficient to give good estimation of general integrated quantity. However, in order to capture the turbulent features of the flow, which can have a significant effect on the transport of a secondary phase in a multiphase flow, it is necessary to consider scale resolving simulation using for example Large Eddy Simulation (LES) models. The IBM presented here has been extended to include a hybrid RANS–LES turbulence model. A Detached-Eddy Simulation (DES) model [39] was coupled to the current penalization approach, taking advantage of the RANS modeling for the wall layer, while using a LES outside of this region to resolve all the scales of turbulence up to the filter size. The relative simplicity and robustness of the Spalart–Allmaras (SA) model makes it a good candidate for the RANS approach. In addition, the similarities of modeling between the SA model and a standard LES formulation for the turbulent viscosity simplifies the coupling of the RANS and LES models. The combinations of our IBM with the SA and the DES models are respectively discussed in Sections 2.3.1 and 2.3.2.

2.3.1. IBM formulation of the Spalart–Allmaras model

The standard Spalart–Allmaras model assumes a zero Dirichlet condition on a wall surface for the turbulent viscosity ν_t [40]. In order to satisfy this boundary condition on the immersed boundary, the standard SA model has been modified to take into account the presence of the immersed body, using a penalization method similarly to the momentum equation. The Dirichlet condition is thus imposed for the turbulent viscosity inside the whole solid. Furthermore, for a better representation of the immersed interface, a corrected viscosity is imposed in its vicinity, similarly again to the correction scheme of the velocity.

In OpenFOAM[®], the SA model requires the solution of the transport equation for a modified kinetic viscosity $\tilde{\nu}$. The turbulent viscosity is obtained from the latter according to:

$$\nu_t = \tilde{\nu} f_{v1} \quad (17)$$

where:

$$\begin{cases} f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \\ \chi = \tilde{\nu}/\nu. \end{cases} \quad (18)$$

The penalized version of the transport equation for the modified turbulent viscosity finally reads:

$$\begin{aligned} \frac{D\tilde{\nu}}{Dt} = P - D + \frac{1}{\sigma} (\nabla \cdot (\nu + \tilde{\nu})(\nabla \tilde{\nu}) + c_{b2}(\nabla \tilde{\nu})^2) \\ + \frac{\chi}{K_{\tilde{\nu}}} (\tilde{\nu} - \nu_{ib}) + \frac{\chi^*}{K_{\tilde{\nu}}} (\tilde{\nu} - \nu_p^*). \end{aligned} \quad (19)$$

In Eqs. (18) and (19), P and D stands for the production and destruction terms which are implemented as defined by Spalart and Allmaras in [41], while c_{p1} and c_{p2} are constants of this model, and finally σ is the turbulent Prandtl number. Regarding the imposition of the boundary condition, v_{ib} stands for the zero immersed body viscosity, while ν_p^* is the corrected viscosity taking into account the position and the orientation of the interface. As for the penalization of the velocity, χ , χ^* and K_{ν_t} denotes here the characteristic functions and the penalization coefficient associated to the turbulent viscosity. Finally c_{b1} and c_{b2} are constants of the standard SA model, while σ is the turbulent Prandtl number. The corrected viscosity ν_p^* is estimated by an interpolation between v_{ib} at the position of the interface and the value of the turbulent viscosity in the neighboring fluid area, similarly to the algorithm used for the penalized velocity in Eq. (7). A similar type of correction has been used by Balaras in [42].

2.3.2. IBM-DES

In the DES model, the transition from RANS to LES is triggered by the following parameter:

$$\tilde{d} = \min(d_w, C_{DES}\Delta). \quad (20)$$

In Eq. (20), d_w stands for the distance to the wall, C_{DES} is a constant usually set to 1 and Δ defines the grid spacing as $\Delta = (\Delta x \Delta y \Delta z)^{1/3}$. When $d_w \leq \Delta$ the models acts as the SA model, while for $\Delta \leq d_w$ it acts as a LES model.

The IBM of this work is then combined with the DES model by merely changing d_w to d_w^* as detailed in Eq. (21).

$$d_w^* = \min(d_w, \psi) \quad (21)$$

where ψ is the distance function to the immersed boundary. Therefore, the model is acting as well as a SA model in the regions near the walls of the immersed solid.

3. Results and discussions

The current IBM has been validated against several test cases, covering different types of geometries for the solid and flow regimes. The improved penalization approach with corrections schemes is compared to a simple first order penalization approach, where no treatments are applied for velocity and pressure boundary conditions on the immersed surface. Numerical solutions were also computed on equivalent body-fitted grids for references. For all validation cases, the IBM solutions are obtained on Cartesian grids of perfect orthogonal quality. In addition, the penalization coefficients are the subject of a convergence study to confirm independence of the results with regards to the penalization coefficients. In the standard approach, the body-fitted grid is mainly Cartesian as well, and based on the same mesh size than the IBM grid, as illustrated in Fig. 4 for the Wannier case studied in Section 3.1. Then, in order to fit the Cartesian cells to the solid geometry, a so-called ‘‘Cut-Cell’’ method is used. Finally, an additional layer of identically sized boundary conforming cells is added on the surface of the solid, without any inflation layers. This was implemented in an attempt to minimize differences between the IBM and the body-fitted approaches. In this study, the computations were performed up to a residuals of 10^{-5} for both velocity and pressure. In OpenFOAM®, the residuals are normalized with the volume-weighted central coefficients from the discretized transport equation of the considered variable (where Upwind Differencing is used on the convection term) [34].

Table 1

Norm \mathcal{L}_1 of the errors obtained with the fine grid (Grid 3), and the associated order of accuracy, for both velocity components.

Model	$E_{\mathcal{L}_1,x}$	$E_{\mathcal{L}_1,y}$	Order in x	Order in y
Body-fitted	0.00104	0.00057	0.9	1.1
Simple Penalty	0.00251	0.00121	0.4	0.6
Penalty with corrections	0.00090	0.00051	1.4	1.4

Table 2

Norm \mathcal{L}_2 of the errors obtained with the fine grid (Grid 3), and the associated order of accuracy, for both velocity components.

Model	$E_{\mathcal{L}_2,x}$	$E_{\mathcal{L}_2,y}$	Order in x	Order in y
Body-fitted	0.00148	0.00078	0.9	1.2
Simple Penalty	0.00379	0.00166	0.4	0.6
Penalty with corrections	0.00142	0.00079	1.5	1.4

Table 3

Relative error $\bar{E}_{r,k}$ obtained with the fine grid (Grid 3), for both velocity components.

Model	$E_{r,x}$	$E_{r,y}$
Body-fitted	0.027	0.024
Simple Penalty	0.264	0.192
Penalty with corrections	0.088	0.056

Table 4

Comparison of the drag coefficient and re-circulation length obtained with the fine grid (Grid 3) in a steady-state with literature data.

Model	C_D	L_S
Coutanceau and Bouard (Exp.) [45]	–	2.13
Tritton (Exp.) [46]	1.59	–
Taira and Colonius [47]	1.54	2.30
Xu and Wang [48]	1.66	2.21
Parussini and Pediroda [49]	1.55	2.27
Body-fitted	1.60	2.31
Simple Penalty	1.58	2.21
Penalty with corrections	1.60	2.30

3.1. Accuracy study with a Wannier flow

The order of accuracy of our Penalty based IBM is first verified with a simple 2D flow in the Stokes regime. The analytical solution of a flow around a circular cylinder, in the vicinity of a moving wall, has been derived by Wannier [43], for very low Reynolds number. The existence of an analytical solution allows a precise comparison with the numerical estimations. The case considers a cylinder of diameter D , centered in a domain of length $6D$ and height $3D$. The dimensions of the problems are presented at the correct scale in Fig. 4. The bottom wall, located at a distance of $0.5D$ to the bottom of the cylinder, is a moving wall with a constant velocity $U = 1 \text{ m} \cdot \text{s}^{-1}$. On all other boundaries the velocity also satisfies a Dirichlet condition with the analytical solution. A zero gradient condition is imposed for the pressure at every boundary, except at the top wall for the sake of symmetry, where a Dirichlet condition is used.

As an analytical solution is known, the error between the numerical and analytical solutions can provide a good estimation of the accuracy of the method. The latter is computed through the norms \mathcal{L}_1 , \mathcal{L}_2 and $\bar{E}_{r,k}$, as defined in Eqs. (22)–(24) respectively.

$$E_{\mathcal{L}_1,k} = \frac{1}{N_c} \sum_i |u_{i,k} - v_{i,k}| \quad (22)$$

$$E_{\mathcal{L}_2,k} = \left(\frac{1}{N_c} \sum_i |u_{i,k} - v_{i,k}|^2 \right)^{0.5} \quad (23)$$

$$\bar{E}_{r,k} = \sum_{i \in \tilde{\Omega}} \left(\frac{|u_{i,k} - v_{i,k}|}{|v_{i,k}|} \right) s_i \quad (24)$$

where \mathbf{u}_i and \mathbf{v}_i stands respectively for the numerical and analytical solutions computed at the center of the i th cell, k is the velocity

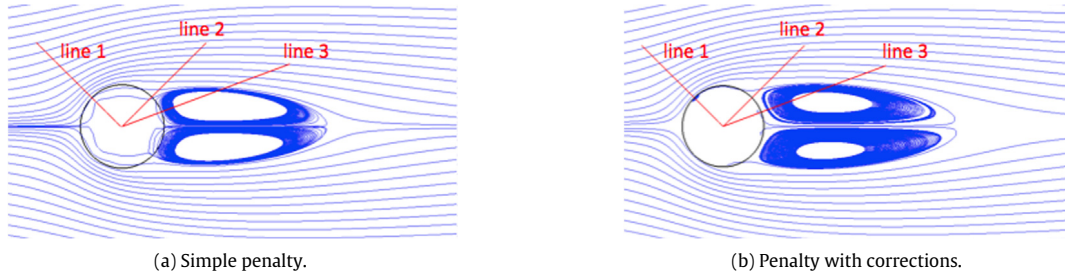


Fig. 5. Streamlines of the flow around a cylinder obtained with the coarse grid (Grid 1). The red lines show the position of the extracted velocity profiles.

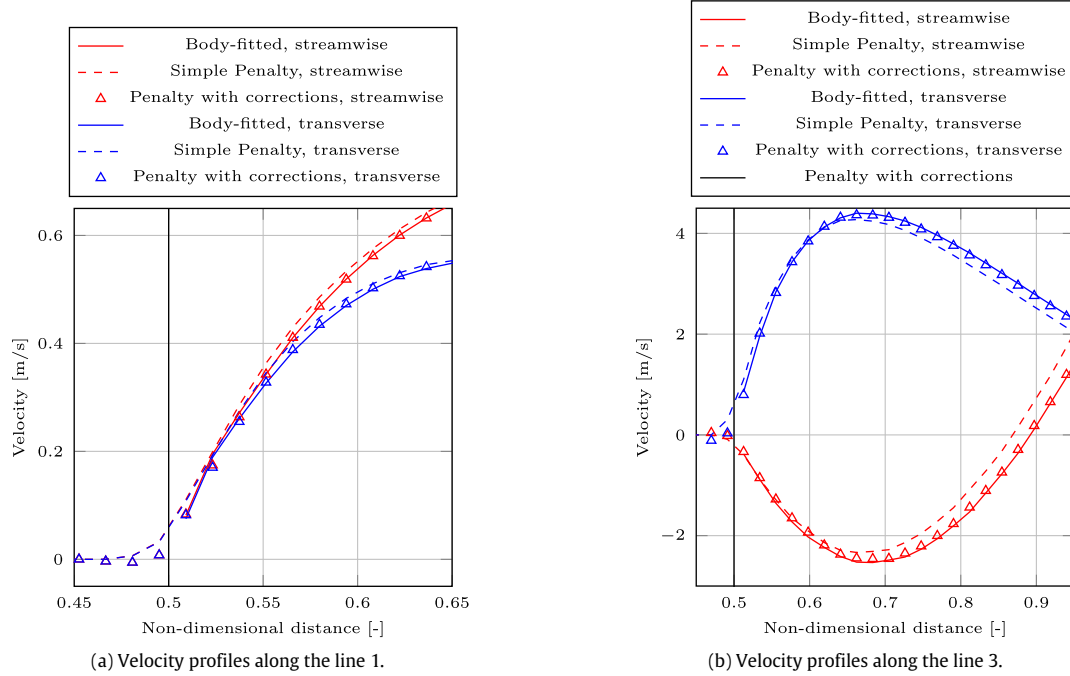


Fig. 6. Streamwise (in red) and transverse (in blue) velocity profiles along sampling lines 1 and 3 near the fixed cylinder. The results are obtained with the fine grid (Grid 3). The vertical black line depicts the position of the cylinder interface. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

component, and N_c is the total number of grid cells. The norms \mathcal{L}_1 and \mathcal{L}_2 represent global values computed on the whole domain, while the relative error $\bar{E}_{r,k}$ is averaged on a square domain $\bar{\Omega}$ of size $0.6D$ around the immersed object, and provides an estimate of the local error. In Eq. (24), s_i stands for the surface elements inside $\bar{\Omega}$. The norms \mathcal{L}_1 and \mathcal{L}_2 of the errors, obtained with different models, are listed in Tables 1 and 2 respectively, for both velocity components. The orders of the different models are estimated using the Richardson method [44]. This method is based on the ratios of the considered values obtained for three different grid refinements. The Grid 1, Grid 2 and Grid 3 considered here, include respectively 10, 20 and 40 cells in the cylinder diameter. The local errors $\bar{E}_{r,k}$ are listed in Table 3 for the finest grid.

The corrections of the boundary conditions for velocity and pressure at the interface are shown to achieve a significant decrease of the norms \mathcal{L}_1 and \mathcal{L}_2 of the global errors, which are brought to a similar level than body-fitted errors. The local error in Table 3 is also reduced from a factor 3 between the simple and the improved penalization methods. Although second-order schemes are used for all simulations, one can note that the overall order of accuracy for the global errors are relatively low. This is probably due to an over-constraint on the boundary conditions on the outer domain. However a qualitative comparison is still

relevant and shows an important increase in the order of the penalization method, which goes approximately from 0.5 to 1.5 when velocity and pressure corrections are accounted for. Finally, the fact that the corrected IBM method achieves a larger order than obtained with the body-fitted approach for a fairly similar error on the fine mesh, can be attributed to a lower accuracy on a coarse mesh.

3.2. Fixed cylinder in a cross flow

In this test case, a fixed cylinder in a 2D cross flow is considered. The domain is rectangular, with a length of $40D$ and a height of $30D$, where D is the diameter of the cylinder. The center of the cylinder is placed at $(10D, 15D)$. On the top and bottom walls a slip-condition is imposed, while inflow and outflow boundary conditions are fixed at the left and right boundaries respectively. A Reynolds number of 40 is considered. The same three grid refinements relatively to the cylinder, as for the Wannier flow, are used for this case (Grid 1, Grid 2 and Grid 3). This case has been widely studied and used for validation in literature.

The comparison focuses on the drag coefficient C_D and the re-circulation length L_S behind the cylinder evaluated from the steady-state solutions. The drag coefficient is computed from the

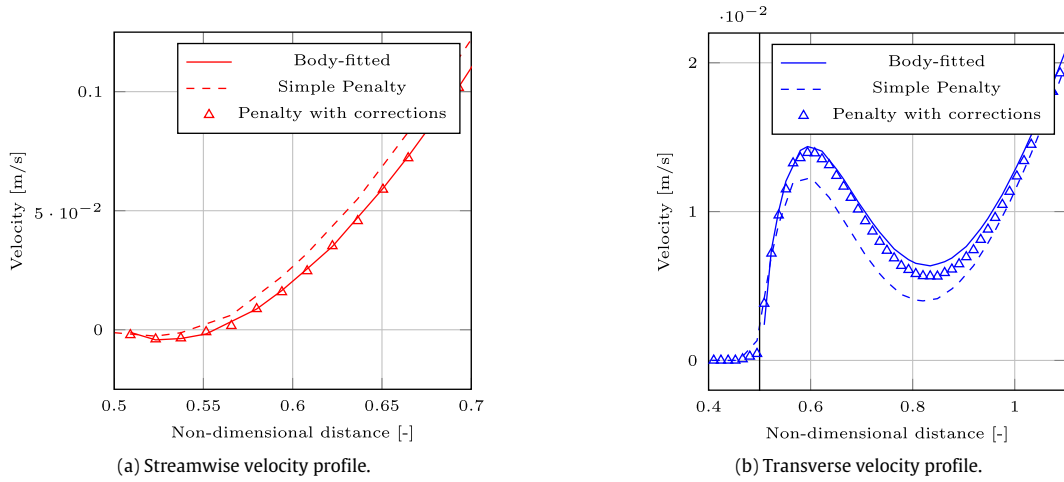


Fig. 7. Velocity profiles along the sampling line 2 near the fixed cylinder. Results obtained with the fine grid (Grid 3). The black line depicts the interface position.

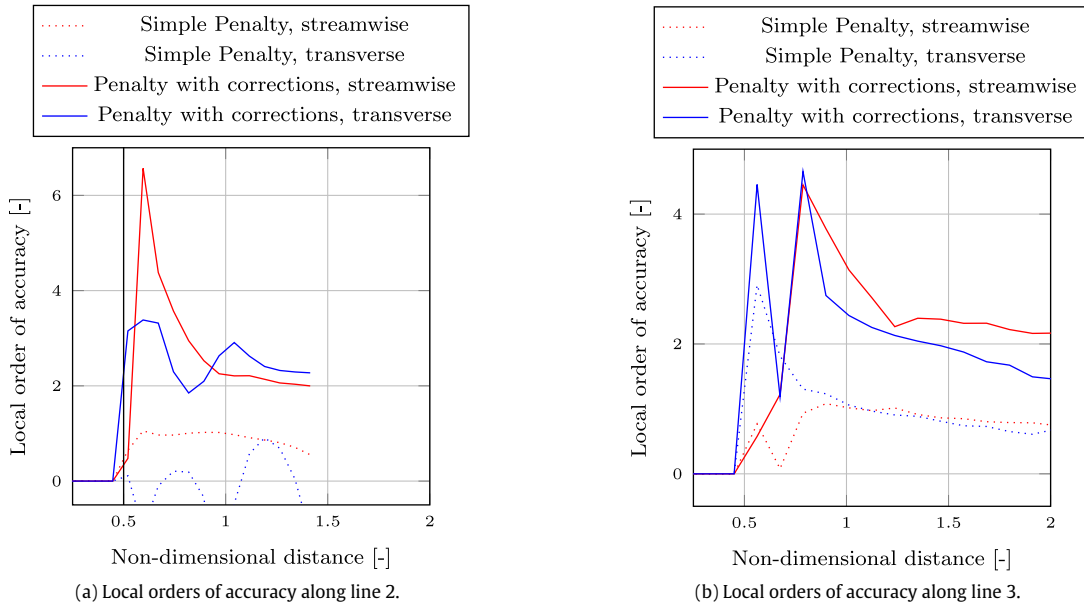


Fig. 8. Local order of accuracy for the velocity along the lines 2 and 3. 20 sampling points were considered for the estimation of the orders. The vertical black line depicts the position of the cylinder interface.

estimation of the force exerted by the fluid as described in Section 2.1.5. Both physical parameters obtained in previous published experimental and numerical work focusing on IBM are compared with results from the present penalization method and the equivalent body-fitted simulations (see Table 4). For a similar resolution near the interface ($\Delta x = 0.025D$ for grid 3, $\Delta x = 0.05D$ for Xu and Wang, and $\Delta x = 0.02D$ for Taira and Colonius), the two physical parameters obtained with the current penalization approaches are in good agreement with what can be found in literature. More precisely, one can note the improvements made by the corrections for the Penalty method, which increase both the drag and the specific length, in accordance with the body-fitted results.

The streamlines of the flow obtained with the present sharp penalization method are shown in Fig. 5. As in reality the penalization coefficient (in Eqs. (3) and (5)) is not exactly equal to 0, the zero Dirichlet condition on the immersed interface in this case is not satisfied up to the machine accuracy. As a result, the streamlines may penetrate the immersed solid as shown for the simple Penalty

approach. The improvements on the interface boundary conditions obtained with the corrected Penalty formulation are visible on the same Figure, as they reduce significantly the penetration of the streamlines inside the body. In addition, velocity profiles are extracted along the lines marked in red in Fig. 5. These lines make respectively an angle with the vertical axis of -45° , 45° and 70° , which correspond to a location where the boundary layer is developed (line 1), a location after the separation point (line 2), and finally a location crossing the re-circulation bubble (line 3). The corresponding velocity profiles are shown in Figs. 6 and 7, for both streamwise and transverse components. Moreover, for a sake of visibility in the comparison, some zooming were performed on the region close to the interface.

Fig. 6(a) shows the velocity profile before the separation point, where the boundary layer is still attached. As the main issue in this problem is actually for a numerical model to determine the right location of this separation point, all the approaches lead as expected to similar profiles at this location. After the separation points however, the profiles detailed in Figs. 7 and 6(b) show the significant

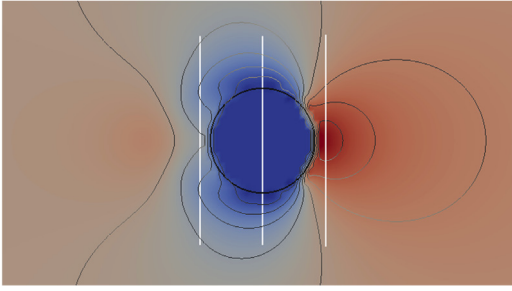


Fig. 9. Pressure contours around the oscillating cylinder for a phase angle $\omega t = 180^\circ$, obtained with the simple Penalty approach on the medium grid (Grid 2). The white lines show the position of the extracted velocity and pressure profiles.

improvements with the improved penalization compared to the first order approach. These conclusions are consistent with the numerical estimation of the re-circulation lengths, for which the simple Penalty method is giving a significant discrepancy with the body-fitted estimation (around 5%), while the Penalty approach, with both velocity and pressure corrections, has shown to be a more accurate model.

Fig. 8 shows locally the order of accuracy for the two velocity components computed with both simple and improved IBM. The orders are estimated along the lines 2 and 3 (see Fig. 5) with the Richardson method [44], using an additional level of refinement for the finest grid. The different curves exhibits a high variability of the order near the interface. Although the first sampling point shows a low order for the two Penalty approaches, the average order of accuracy is above 2 for the Penalty with corrections at a distance of one radius from the interface. Considering the streamwise component, away from the interface, the values seem to converge asymptotically towards 2 for the improved IBM, and towards 1 for the standard IBM. Similarly for the transverse component of the velocity, the order stabilizes between 1.5 and 2 for the improved IBM depending on the sampling line. One can note that the orders outline a higher variability for the transverse component. In addition, the local orders obtained for the simple penalty approach along line 2, which is in the vicinity to the separation point and on the edge of the recirculation bubble, are negatives in some parts. It could be the case that between two grid levels, the sampling line is passing inside or outside the recirculation zone, which would

affect significantly the extracted velocity magnitude. An extra level of refinement might be then necessary to obtain consistent orders of accuracy. This issue is corrected with the improved IBM. These results emphasize the improvements brought by the corrections. Furthermore, they are consistent with the orders of accuracy found for the recirculation length for instance (1.0 for the standard IBM and 1.5 for the improved IBM). The orders obtained with the current Penalty approach with boundary treatments are comparable to previously published sharp IBM offering improvements to tackle the rasterization effect. Hence, the penalization method of Introini et al. [17] leads to a global order of accuracy around 1.88 for the Taylor–Couette problem for both L_∞ and L_2 norms, while global orders between 1.27 and 2.31 are obtained for the flow around a static cylinder, or between 1.51 and 1.91 for the flow around a rotating cylinder. Similarly, the sharp IBM of Gilmanov et al. [50] reaches 1.48 and 1.74 as a global order for the norms L_∞ and L_2 respectively when studying the steady flow in a cubic lid-driven cavity. One can also mention that already existing high-order IBM can achieve perfect second-order or above for specific test cases [42,16,51].

3.3. In-line oscillating cylinder in a fluid at rest

A cylinder oscillating in a fluid at rest is considered in this case. The dimensions of the problem are similar to the previous test case, as well as the IBM and body-conforming grids used for the simulations. In the standard body-fitted approach, the cells are slightly and smoothly stretched in time, in order to follow the motion of the cylinder. The equation of motion reads: $x(t) = -A \sin(\omega t)$, where the frequency $f = \frac{\omega}{2\pi}$ is set to 0.2 Hz and the amplitude A is fixed to $\frac{5}{2\pi}$. This motion corresponds to a Reynolds number equal to 100 and a Keulegan–Carpenter number equal to 5.

The oscillating cylinder case is a validation case also frequently used in literature [51–54] for numerical model handling moving geometries. Laboratory experiments have been conducted as well by Dütsch et al. [55]. The oscillations of the cylinder lead to the development of lower and upper boundary layers, which separate and generate two counter-rotating vortices in the lee. This generation process stops when the cylinder starts to move backward and finally splits the vortex pair.

The pressure contours obtained with our simple Penalty approach are plotted in Fig. 9 for a phase position $\omega t = 180^\circ$, when the cylinder is in its central position and is moving to the right. The

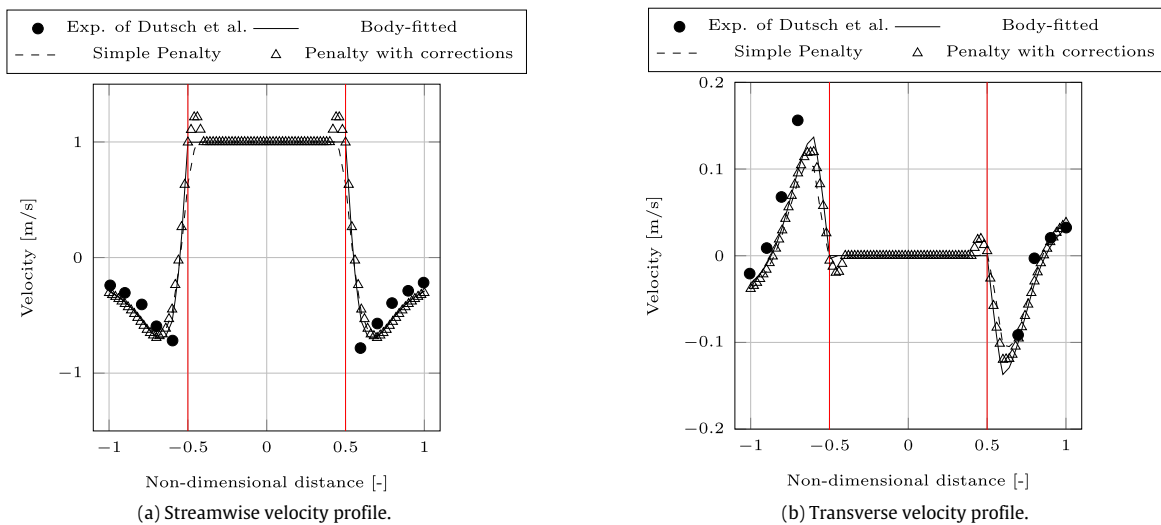


Fig. 10. Instantaneous velocity profiles near the oscillating cylinder, at $x = 0$ and for a phase angle $\omega t = 180^\circ$. The results are obtained with Grid 2. The vertical red lines depict the position of the cylinder interface.

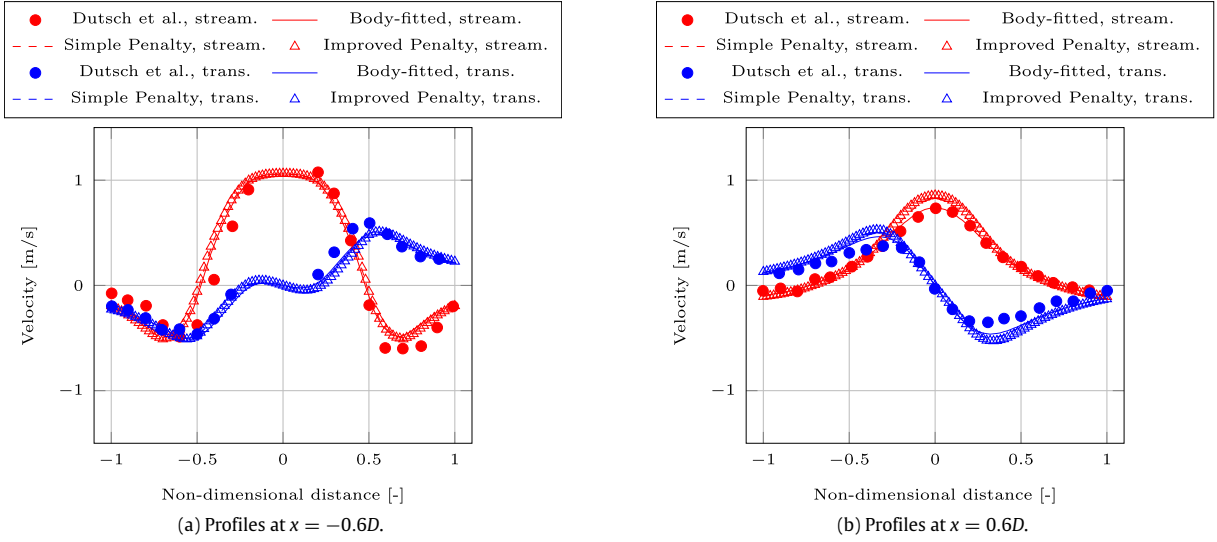


Fig. 11. Instantaneous velocity profiles near the oscillating cylinder, at $x = 0.6D$ and $x = -0.6D$ and for a phase angle $\omega t = 180^\circ$. The results are obtained with Grid 2 for streamwise (in red) and transverse (in blue) component of the velocity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

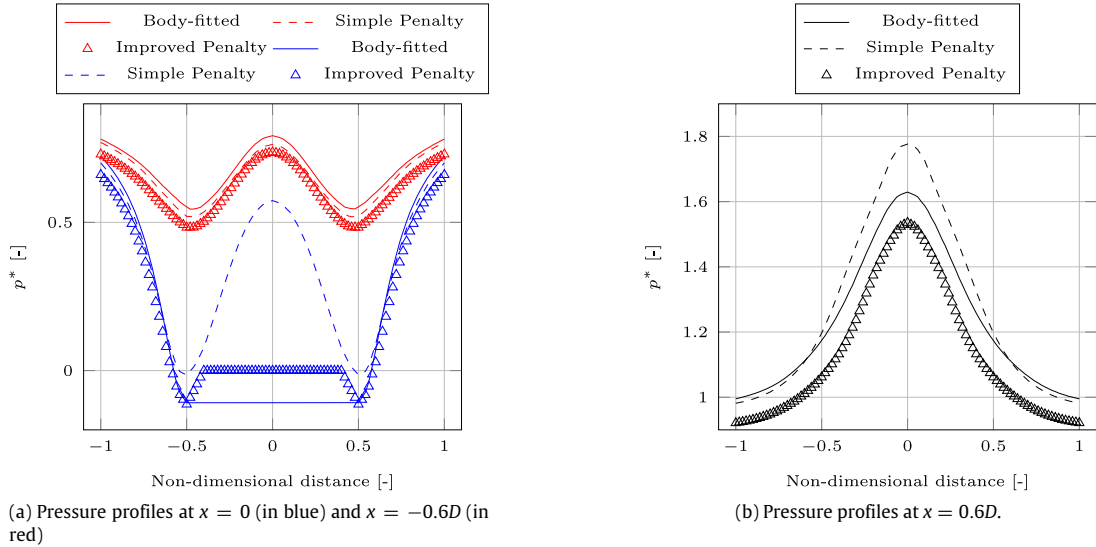


Fig. 12. Instantaneous pressure profiles along the three sampling lines near the oscillating cylinder for a phase angle $\omega t = 180^\circ$. The results are obtained with Grid 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

pressure is normalized in terms of the cylinder maximum speed U_m as $p^* = p/(\rho U_m^2)$. The pressure contour shows the symmetry obtained with our model, and are in good agreement with the results of Dutsch et al. [55]. On this figure, the white lines represent the position of the extracted velocity and pressure profiles at the same phase angle, respectively at a distance $x = -0.6D$, $x = 0$, and $x = 0.6D$ from the center of the cylinder. At this time, the cylinder is back to its initial central position, for which the body-fitted mesh is not stretched, allowing for a more relevant comparison with the IBM results.

The improvements brought by the corrected Penalty approach are especially visible near the interface for both streamwise and transverse velocity in Fig. 10, where a very good agreement with the equivalent body-fitted is obtained. This especially the case for the magnitude of the transverse velocity peaks. For the two other profiles on both sides of the cylinder in Fig. 11, the penalization

approach leads to satisfactory results as well. The experimental results of Dutsch et al. are not shown here for a quantitative validation, but only for comparison purpose. The profiles obtained with IBM are consistent with the experimental velocities and show thus that the qualitative features of the flow are well captured. Furthermore, similar level of discrepancies with these experimental data are obtained with other numerical models in literature [51,54].

Regarding the pressure, one can observe that the imposition of the Neumann condition improves the results near the interface on the profile crossing the cylinder ($x = 0$) in Fig. 12(a). For the two other profiles, although the magnitude is slightly under-estimated, Fig. 12 shows a better distribution of the pressure around the cylinder for the penalization model with corrections. This result supports the fact that the drag is correctly estimated in this case, and implies as well that the pressure gradient are very similar between the body-fitted and the corrected Penalty method.

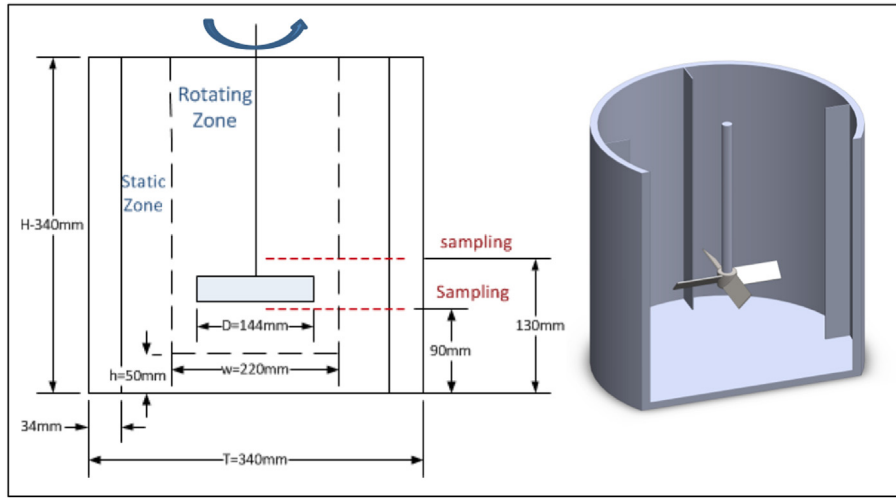


Fig. 13. Outline of mixer layout dimensions and CAD rendering view. The static and rotating zones are specified for the body-conforming approach.

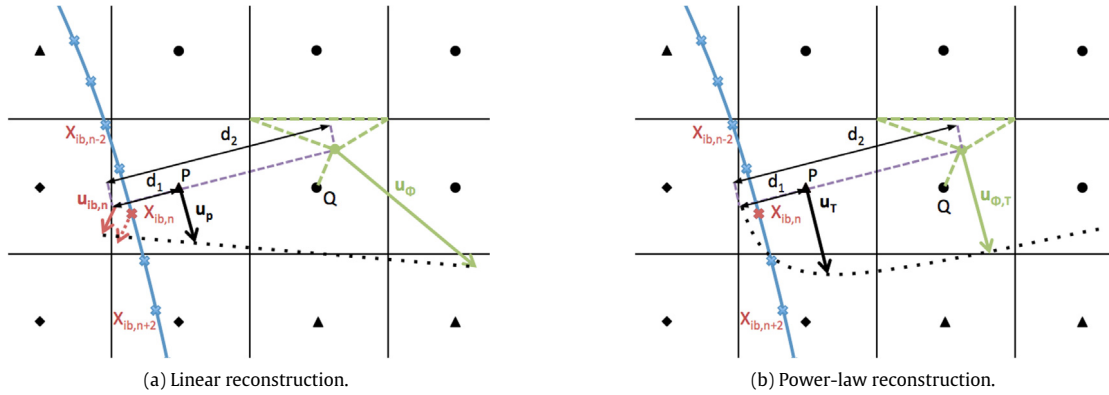


Fig. 14. Sketch of the velocity correction schemes used for the turbulent case. The same nomenclature is used for the cells definition.

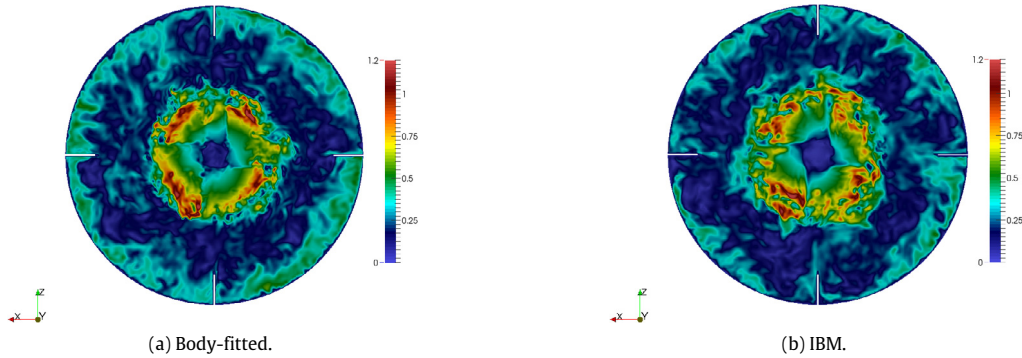


Fig. 15. Contour of the velocity magnitude on a cross-section downstream of the impeller at $y = 0.1$ m for both body-fitted and IBM computations.

3.4. Mixing in a stirred-tank

3.4.1. Case description

This case concerns the cylindrical mixing tank with 4 vertical baffles and the 4 pitched blade impeller studied by Ge et al. [56]. The results reviewed here refer to the 45° pitch impeller with rectangular blades. The tank and impeller assembly has a 90° rotational periodicity. Ge et al. study provides both experimental and computational results. Radial and axial velocities have been measured using PIV along two radial sampling lines located upstream and

downstream of the impeller at axial position $y = 0.09$ m and $y = 0.13$ m and offset by 5° from the vertical plane containing the baffles. In addition, the phase averaged turbulent kinetic energy is estimated from the 2D planar PIV measurement using the pseudo-isotropic assumption: $k = \frac{3}{4}(\bar{u}^2 + \bar{v}^2)$, where \bar{u} and \bar{v} are the turbulent radial and axial velocity fluctuations.

The design of the problem is shown in Fig. 13. The tank used in the experimental study was deeper than in the computational model where the height H corresponds to the level of the free surface in experiments. The impeller rotational speed is set at $N =$

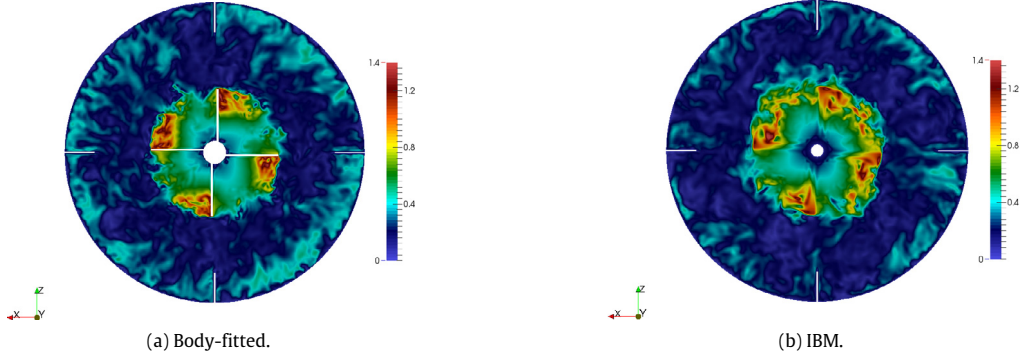


Fig. 16. Contour of the velocity magnitude on a cross-section cutting the impeller at $y = 0.11$ m for both body-fitted and IBM computations.

Table 5

Numerical settings for the different CFD models.

Model	Turbulence model	Max. grid res.	Impeller thickness
Ge et al.	RANS $k - \epsilon$	2 mm	0
Body-fitted	DDES	0.5 mm	2 mm
IBM	coupled IBM-DES	1 mm	1.6 mm

$\Omega/2\pi$ giving the Reynolds number $Re = ND^2/\nu = 5.18 \cdot 10^4$ and the Froude number $Fr = N^2D/g \simeq 3.6$. Under these conditions distortions of the free surface by the liquid stirring can be expected to have little influence [57]. Computational tests were conducted with a Volume of Fluid (VOF) model to account for the influence of the free surface and confirmed this to be the case.

3.4.2. Meshing details and accurate turbulence modeling

The computational results from [56] were obtained from a RANS $k - \epsilon$ model and a GGI method to handle the rotating zone and a hybrid mesh. Hexahedral cells were used for the stationary zone and tetrahedral cells ranging from 2 mm in the impeller discharge region to 6 mm elsewhere. Zero thickness baffles and impeller blades were used to avoid meshing complication. Similarly for the body-fitted simulations performed in this research, the Generalized Grid Interface (GGI) method was used to model the moving boundaries. However, a cubic mesh was employed for both the stator and the rotor parts. The edges of the quadrilateral cells are aligned with the coordinate axes everywhere except in the vicinity of inflation layers at wall boundaries where tetrahedral and pyramidal cells are used to make the transition. Local refinements are achieved by dividing cells into uniform octants. The body-fitted mesh is based in the end on 2 mm cells. A refined zone is defined in the rotating region, extending from its lowest point (see Fig. 13) to 0.19 impeller diameter. Finally, one additional refinement is applied in four layers adjacent to wall boundaries. Meshing of the impeller wall relies on a single quad based prism layer (inflation layer). Concerning the impeller itself, a 2 mm wall thickness was used instead of the zero thickness from Ge et al. With this settings, an average y^+ of 5.8 was obtained. A coarser mesh was also considered without the intermediate refined region. Finally for the IBM case, a similar cell size was considered in the tank. However, the refined region with 1 mm cells is smaller in this case, ranging from $y = 0.095$ m to $y = 0.13$ m. In this sense the equivalent IBM resolution is in-between both body-conforming grids. The IBM is using the physical thickness of the impeller (1.6 mm), hence its discrete representation is only one or two cells thick. The grids details are listed in Table 5.

The impeller surface mesh is composed of triangular elements of 0.25 mm in average, which is four times smaller than the size of the fluid cell. The approximation made for the estimation of the distance to the interface in arbitrary geometry, that is which cannot

Table 6

Volume averaged relative error as defined in Eq. (25) for different methods of computing the distance to the immersed interface.

Method	Dicho. 5 it.	Dicho. 10 it.	Dicho. 20 it.	surface mesh
Error \bar{E}_d	0.25	0.0059	$5.2e^{-6}$	0.0085

be defined analytically, is discussed here. We define the average error in the calculation of the distance between fluid cell center points and the interface as:

$$\bar{E}_d = \frac{1}{\int_{\Omega_p} d\Omega} \int_{\Omega_p} E_{d,i} d\Omega = \frac{1}{\int_{\Omega_p} d\Omega} \int_{\Omega_p} \left(\frac{d(\mathbf{x}_i) - d_{ref}(\mathbf{x}_i)}{d_{ref}(\mathbf{x}_i)} \right) d\Omega \quad (25)$$

where $E_{d,i}$ is the local relative error in cell i between the approximate distance $d(\mathbf{x}_i)$ and a reference distance $d_{ref}(\mathbf{x}_i)$. The reference distance is calculated with a dichotomy algorithm using 80 iterations. The average error is integrated over the penalized domain Ω_p . The average errors obtained with the considered surface mesh and with different dichotomy iterations are listed in Table 6. Using 20 iterations for the dichotomy algorithm is shown to give a negligible relative error. This support the argument that the error in the estimation of the distance, which converges in 2^k , k being the number of dichotomy iterations, is low enough to consider the distance obtained with 80 iterations as the real distance to the interface. From the table, we can see that approximating the distance to the interface by the distance to the closest solid points with a surface mesh of 0.25 mm elements leads to an average relative error below 1%. This can be considered as acceptable for our simulation case. Furthermore, it should be noted that the approximation made here reduces the CPU time for computing the distance from 8.87 s to 2.53 s in comparison to a dichotomy algorithm using 80 iterations. These CPU times were obtained for a sequential simulation and the distances were estimated for all cells in the penalized domains (approximately 20,000 cells in total). The same analysis on the error is performed for the Wannier case, described in Section 3.1. The medium mesh is considered (Grid 2) for the fluid domain and here also the surface mesh for the cylinder is composed of triangular elements four times smaller. The reference distance is calculated with the analytical distance to the interface. An average relative error around 2.5% was found in the penalized domain. The larger relative error obtained in this case can be explained by the use of larger volume and surface mesh.

Both body-fitted and IBM simulations performed for this study have used hybrid RANS-LES type of turbulence model. While the IBM is coupled to a DES model (see Section 2.3), the body-fitted case used instead the Delayed Detached Eddy Simulation (DDES), which reduces incorrect behavior triggered by a grid spacing parallel to the wall smaller than the boundary-layer thickness [58]. The resolutions of the two body-fitted grids have been assessed in

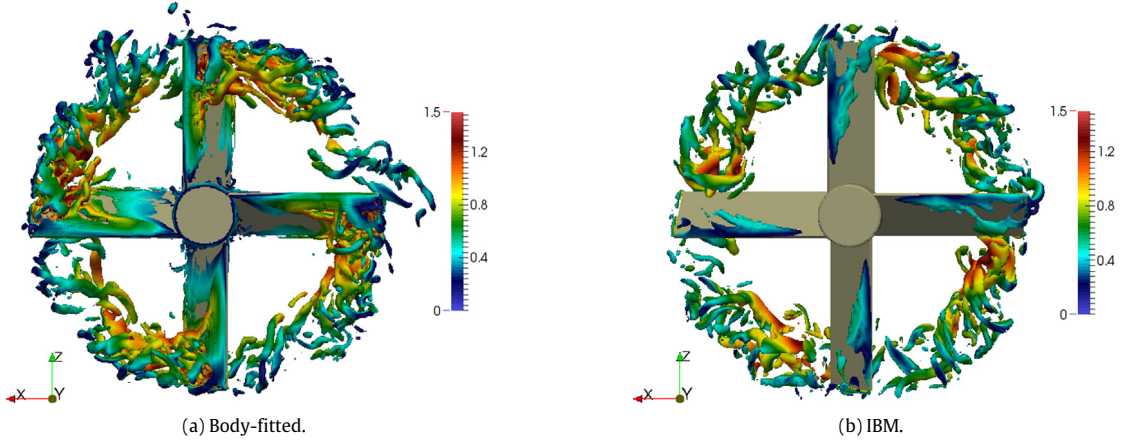


Fig. 17. Q -criterion iso-contours with $Q = 10^4$ colored by the magnitude of the velocity.

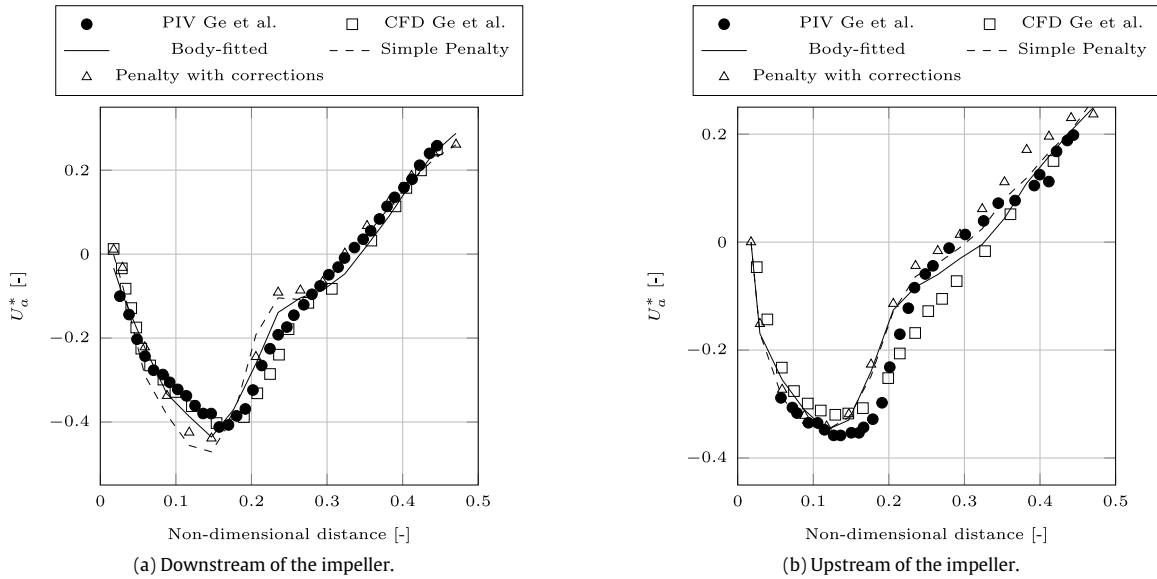


Fig. 18. Phase and time averaged axial velocity along the two sampling lines.

terms of the kinetic energy spectra. The results have shown that both mesh are fine enough to fully resolved the inertial range, with a spectrum characterized by a curve of slope $-5/3$. For this type of transient problem involving high speed flows, the PIMPLE scheme is used to solve the pressure-velocity coupling, instead of the PISO scheme introduced before. The PIMPLE scheme is a combination of both PISO and semi-implicit method for pressure-linked equations (SIMPLE) schemes, that enables looping over the entire system of equations and allows the use of larger time step.

The correction scheme for the velocity boundary condition on the interface, introduced in Section 2.1.4, has been modified for the mixer case. Due to the small thickness of the impeller blade relatively to the mesh size, the velocity corrections are applied within an Eulerian cell layer outside of the solid domain. In this case, the penalized domain Ω_p refers then to the cells in Ω_f possessing at least one neighbor in Ω_s . In this configuration, the reconstruction of the penalized velocity in Ω_p is based on a linear interpolation in the same spirit as the previously employed extrapolation. Both inner [31,16] and outer [50,42] types of corrections have been considered in IBM studies. However, given the mean y^+ obtained with the body-fitted simulation, it can be assumed that cells adjacent to the impeller boundaries will typically be placed in the

logarithmic layer provided that the flow remains attached, when modeled with IBM. For this reason, a velocity correction scheme based on a power-law is also considered in order to better describe the turbulent velocity profile of the boundary layer. Similar types of corrections have been proposed by Choi et al. [59] and Chang et al. [60] for turbulent flows. A 1/7 power law was firstly proposed by Werner and Wengle [61]. In [60], Chang et al. have considered and compared both linear and power-law based reconstruction of the velocity. Their results suggest that the former method gives slightly better results, and the authors argue that this is most likely due to the high level of grid refinement used, which is fine enough to reach the viscous sub-layer in some areas.

The velocity correction scheme based on linear interpolation is formulated in Eq. (26). This correction is performed in the same spirit as the inner extrapolation scheme described in Section 2.1.4. The definition of d_1 , d_2 and \mathbf{u}_ϕ are identical. The linear interpolation is illustrated in Fig. 14(a).

$$\mathbf{u}_p^* = \frac{d_2 - d_1}{d_2} \mathbf{u}_{ib,n} + \frac{d_1}{d_2} \mathbf{u}_\phi. \quad (26)$$

For the power-law interpolation, the reconstruction is formulated similarly to the *tangency correction* of Choi et al. [59], by decomposing the velocity into its tangential and normal components

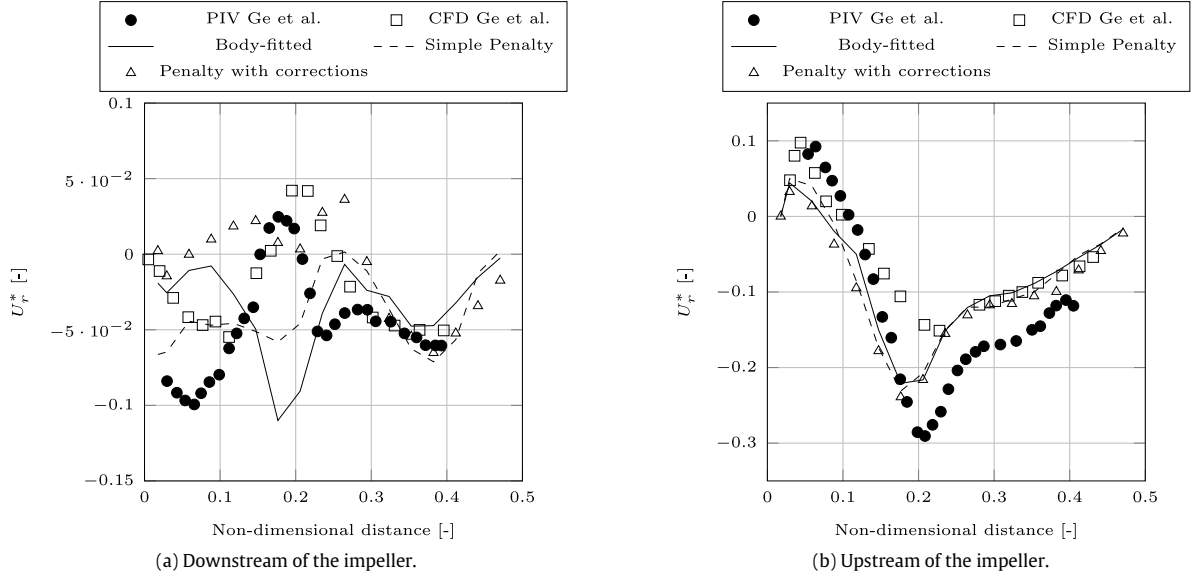


Fig. 19. Phase and time averaged radial velocity along the two sampling lines.

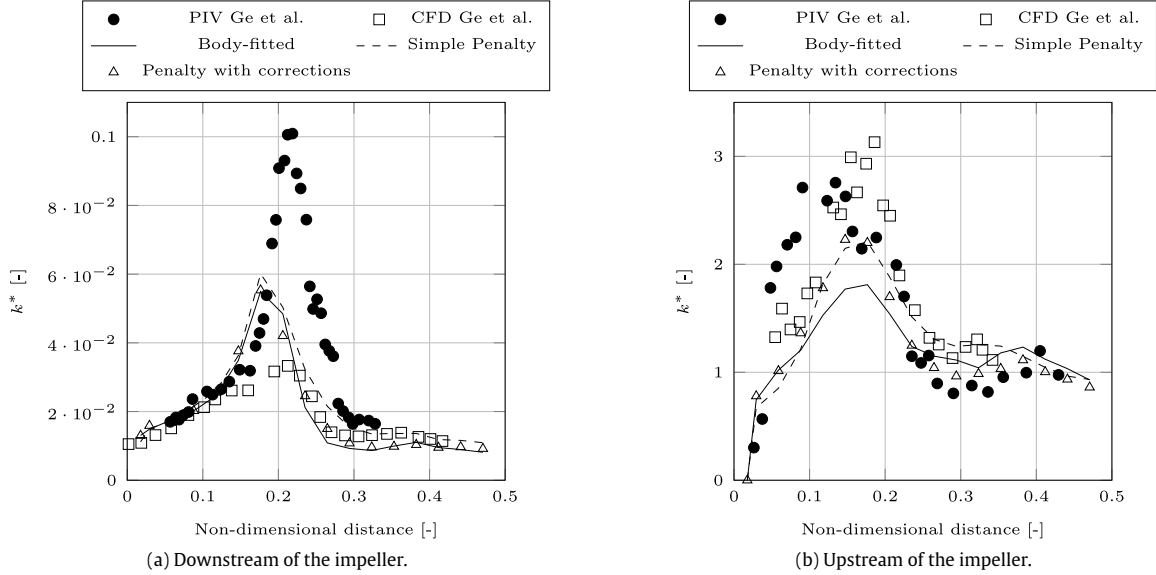


Fig. 20. Phase and time averaged turbulent kinetic energy along the two sampling lines.

relatively to the immersed surface: $\mathbf{u}_p^* = \mathbf{u}_T + \mathbf{u}_N + \mathbf{u}_{ib,n}$. In this formulation, \mathbf{u}_T and \mathbf{u}_N are the components of the fluid velocity relative to the wall velocity. The dependence of the tangential component \mathbf{u}_T to the normal distance to the wall is assumed to follow a power-law function. In this case, the tangential velocity at the cell center P can be derived from the tangential velocity at a virtual point ϕ as:

$$\mathbf{u}_T = \left(\frac{d_1}{d_2}\right)^k \mathbf{u}_{\phi,T} \quad (27)$$

where small values of k (usually 1/7) can approximate the logarithmic distribution expected in the near wall region for an attached turbulent flow. The normal component of the velocity \mathbf{u}_N is assumed to follow a linear distribution, i.e. using $k = 1$. The reconstruction of the tangential velocity component is illustrated in Fig. 14(b). The body fitted simulations indicate that the wall adjacent cells along the impeller lie well within the inner layer of

an attached boundary layer with an average y^+ of 5.8. As the IBM mesh size is twice that of the body fitted version, the penalized point P determined in terms of the distance d_1 from the immersed interface can be expected to be typically located somewhere in the buffer layer, but this will change as the interface moves over the underlying Eulerian grid. Rather than reproducing the different parts of a fully-developed turbulent boundary layer, the inner layer (from the viscous sub layer to the overlap layer) is approximated by the power-law as implemented for example in [61] for $y^+ > 11.8$ or in [59] for the whole layer. This means that the boundary correction does not impose the turbulent boundary layer profile as a function of y^+ and a single reference point is sufficient to reconstruct the full profile irrespective of its location within the boundary layer. The main assumption used in this case is that the velocity at the virtual point (at a distance d_2) does fit the power law, which is incorrect if the flow departs from the zero pressure gradient boundary layer condition. Imposing this reconstruction does in

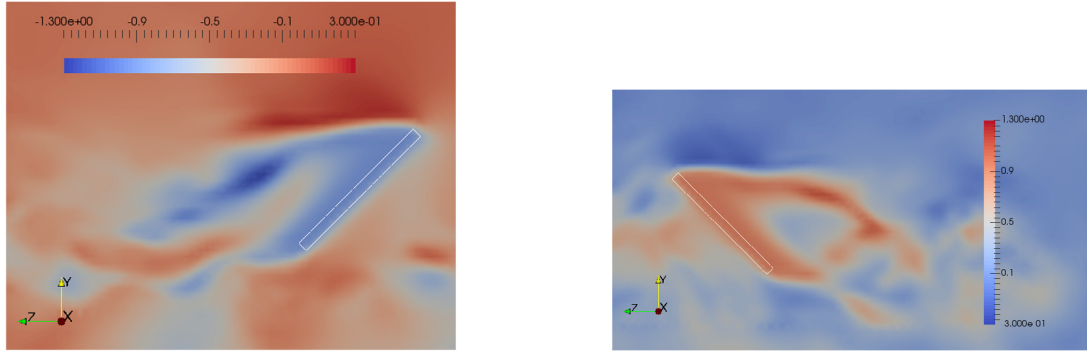


Fig. 21. Contours of the velocity, projected in the direction of the impeller rotational speed (+z), obtained with a linear reconstruction at the immersed surface. The contours are shown on cross-sections perpendicular to two impeller blades diametrically opposed. The white contours depict the impeller surface.

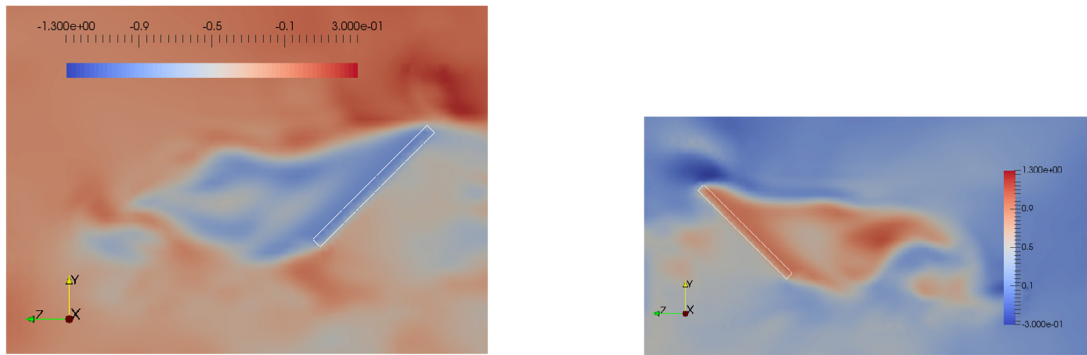


Fig. 22. Contours of the velocity, projected in the direction of the impeller rotational speed (+z), obtained with a power-law ($k = 1/7$) reconstruction at the immersed surface. The contours are shown on cross-sections perpendicular to two impeller blades diametrically opposed at a distance of 0.06 m from the impeller center of rotation. The white contours depict the blades surface.

such a case promote boundary layer attachment by imposing the attached velocity profile at the immersed boundary cell. The next sections present the results obtained with a linear reconstruction of the velocity, while Section 3.4.5 discusses the sensitivity to the type of reconstruction.

Finally, regarding the dependence of the turbulence viscosity to the distance to the wall, Prandtl's assumption [62] is usually used for the logarithmic layer, for which the non-dimensional turbulent viscosity is directly proportional to y^+ . In this context, the present linear reconstruction of the turbulent viscosity in the penalized domain, as used also by Balaras [42] is relevant. For the body-fitted simulations, Spalding's law [63] is used to compute the local turbulent viscosity in order to get the proper wall shear stress. This wall function is valid for all- y^+ and was developed for attached boundary layers.

3.4.3. General results

The initial conditions for the simulation were based on a fluid at rest with uniform static pressure distribution gravity being neglected. The velocity contour obtained after 5 impeller revolutions are shown in Figs. 15 and 16. The accuracy of this variable is meaningful for mixing problems. In the considered areas, the mesh size is $\Delta x = 1$ mm in a circular region centered in the impeller with a radius of respectively 0.11 m for the body-fitted case and 0.08 m for the IBM case, while the mesh size is $\Delta x = 2$ mm outside these regions. Although the regions of high velocity magnitude look more heterogeneous directly below the impeller in Fig. 15, the current penalization approach is leading to a similar level of mixing in the tank, as confirmed by the time averaged flow velocity

profiles discussed in the next section (Section 3.4.4). The size of the dead zone directly under the impeller as well as the extend of the zone characterized by high downstream velocity are in good agreement with the body-fitted case. For the plane cutting the impeller in Fig. 16, the extend of the region of high velocity swept by the impeller also show generally similar characteristics in terms of their spread and diameter. There is however some non negligible difference in the intensity and length of the wake trailing behind each impeller blade. The high velocity region is shown to spread further radially downstream of the impeller (at $y = 0.1$ m) in the IBM case with a lower velocity magnitude, while the flow remains attached to the impeller for longer with the body-fitted mesh allowing for higher velocity flow to evolve which then diffuses more rapidly in the wake, as it can be seen in Fig. 17. This figure highlights the turbulent structures predicted by the models in terms of the Q -criterion. Q is defined by $1/2(\Omega^2 - S^2)$, where $S = \|\mathbf{1}/2(\nabla\mathbf{u} + \nabla^T\mathbf{u})\|$ and $\Omega = \|\text{tr}(\nabla\mathbf{u})\|$ are the strain rate and vorticity rate magnitudes, so that positive iso-contours represent vortices defined as a region where the vorticity is greater than the magnitude of the strain rate.

Concerning turbulence modeling, it is worth noting that the penalization of the turbulent viscosity is a necessity when using a full RANS model. Without a correct boundary condition on the immersed interface, the turbulent viscosity computed inside the solid from the Spalart-Allmaras equation can be very high, leading to an overestimated diffusion in the flow. The influence of this penalization process is however found to have a negligible influence when a DES turbulence model is used. Results confirm that with DES models a correct estimation of the turbulent variables in the

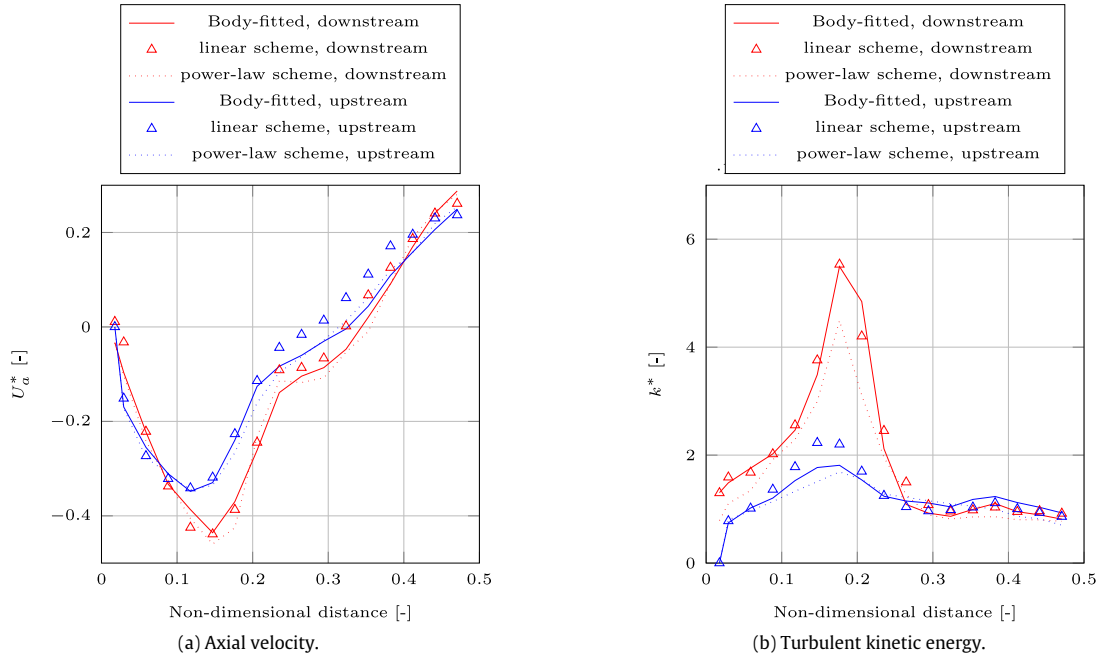


Fig. 23. Dependence of the averaged quantities in the velocity reconstruction scheme.

core is more important, and the latter is not significantly affected by what happens near the boundary layer, in this type of problem where there is significant repeated mixing. A possible reason is that a lot of the turbulence is determined by the interaction of larger turbulent scale with the impeller blade and subsequently the turbulent cascade. This is an important conclusion regarding the relatively low resolution of the impeller wall with the immersed boundary formulation. Fig. 17 illustrates the similarity of the vortical structures around the impeller for both numerical methods. The absence of inflation layers in the IBM can explain the failure of the method to resolve the boundary layer growth and separation. This clearly impacts on the formation of vortices at the impeller blade tips. The type and shape of the vortices obtained with IBM however appear to be similar, and the comparison of flow velocity profiles in Section 3.4.4 shows that this effect on time average velocity mixing is not significant.

3.4.4. Averaged data

All data averaging were performed over 20 revolutions, after an initial start-up period of 20 revolutions. LES computations reported in [57], and based on similar Reynolds number and averaging procedure, confirmed that the initial start-up phase was sufficient to filter out transients and to provide statistically averaged data. For comparison, 15 impeller revolutions were reported in [64] and 15–20 revolutions in [65]. The results discussed in this section concern phase resolved velocity measured along the sampling lines defined earlier and averaged over these 20 revolutions. The discussion also considers the turbulent kinetic energy which is both phase and time averaged. The phase averaging is performed over 5° angular steps covering one quadrant of the flow domain (i.e. covering the flow domain between two baffles). The axial and radial velocities and the turbulent kinetic energy are normalized in terms of the impeller tip velocity $U_{tip} = \pi ND$ as $U_{a,r}^* = U_{a,r}/U_{tip}$ and $k^* = k/U_{tip}^2$. The total turbulent kinetic energy reported here is determined from the resolved flow velocity components u_i and the sub-grid scales contribution $k_{sgs} = \nu_t/(0.07\bar{d})$ with $k = (u_i^2 - \bar{u}_i^2)/2 + k_{sgs}$ [64].

The phase resolved axial and radial velocities and the phase averaged turbulent kinetic energy distributions along the two sample

lines are compared against experimental and simulation profiles from [56] in Figs. 18–20. Although the general trends and peak velocity values are similar to the experimental data, significant differences are observed over parts of the profiles. The difference is particularly noticeable in the case of the radial velocity downstream of the impeller (Fig. 19(a)). However, the experimental velocities in this case experience relatively small variations around the average value of 0.045 m/s which is approximately 5% of the axial velocity range. Both body-fitted and IBM models predict generally similar results by comparison with the reference profiles obtained with the $k - \epsilon$ RANS model, with two notable differences, (i) gradients are generally better resolved with the DES models and (ii) the peak in turbulent kinetic energy downstream of the impeller, although still significantly dampened is closer to experimental observations. The $k - \epsilon$ model does seem to achieve more accurate kinetic energy predictions upstream of the impeller (Fig. 20(b)). This however is in conjunction with significant under prediction of the radial velocity which, in the case of the DDES simulation, impacts on the determination of the kinetic energy $k = (u_i^2 - \bar{u}_i^2)/2 + k_{sgs}$. The IBM is shown to be in very good agreement with the body-fitted simulations for most of the profiles. Regarding the turbulent kinetic energy, one can note a better estimation of the energy peak upstream of the impeller with the IBM. This overestimation compared to the body-fitted case may be due to the rasterization of the velocity at the immersed boundary artificially increasing stresses. Finally, it is worth mentioning the improvements achieved with the corrected Penalty method, obtained for the axial velocity downstream of the impeller (Fig. 18(a)), which reduces the velocity peak in accordance to the other data.

3.4.5. Sensitivity to the velocity reconstruction scheme

The influence of the type of velocity correction at the immersed interface is discussed in this section. Fig. 21 shows the velocity contour in the wake of two blades on opposite sides of their axis of rotation, when a linear correction is applied. The velocity is projected in the tangential direction of the impeller rotational speed. Fig. 22 shows a similarly projected velocity at the same location when a correction based on a power-law $k = 1/7$ is applied. These results are shown after 5 impeller revolutions.

Different profiles of separation can be observed between the two types of velocity reconstruction. With the linear correction, the wake behind the blade appears broader, while on the contrary the power-law correction generates a narrower wake with a smaller angle of separation from the blades corner. In addition, along the blades, the separation appears closer to the impeller center of rotation with the linear correction. The non-transient nature of these flow characteristics were verified. The same conclusions can be drawn at different locations and different times, although the differences between the two methods slightly reduce with time probably because of the increasing mixing. These results are consistent with the power-law distribution, which enforces more flow tangency to the immersed surface. Similar observations on the delayed separation of the boundary layer in external flow past bluff bodies and on the wake size were made by Choi et al. [59], when they compared the two types of velocity reconstruction.

Lower level of turbulent energy are obtained with the power-law approach, as depicted by Fig. 23(b). This was also observed in [59]. The higher turbulent intensity generated with a linear reconstruction is due to the earlier flow detachment, which destabilizes the different boundary layers and promotes mixing. The phase and time averaged mean velocity, differences are also visible for the axial velocity in Fig. 23(a) but appear less significant. The same conclusion can be drawn for the radial velocity. Both instantaneous and time averaged flow characteristics have been found to be sensitive to the velocity reconstruction scheme. It can be argued that the linear correction gives predictions which are closer to the body-fitted simulation, especially in terms of turbulent quantities and structures. It is argued that this is due to the effect that the power law reconstruction has on promoting attachment of the boundary layer.

4. Conclusion

A penalization based immersed boundary method has been presented, as an alternative to standard body fitted simulation based on the GGI method. The formulation preserves the sharpness of the immersed solid and relies on correction schemes for both velocity and pressure to impose consistent boundary conditions on the immersed interface. Although a second order accuracy is not achieved locally at all boundary points, the corrections have been shown to increase both the accuracy of the results and the order of convergence in comparison to a simple Penalty approach, and to minimize the rasterization effect inherent to Cartesian grids. The computational results presented here confirm that the IBM can produce integrated as well as local velocity and pressure values that are in close agreement with experimental data and standard body-fitted CFD simulations. The validation includes both low and high Reynolds flows, and both attached and separated flows past fixed and oscillating obstacles have been considered. For turbulent flows, the present IBM has been coupled to an hybrid DES model, that uses a LES modeling outside the boundary layer and a modified Spalart–Allmaras model within this layer, accounting for the Dirichlet condition of the turbulent viscosity on the interface. Although some details of the boundary layer formation and separation leading to vortex shedding are lost when the grid resolution is not sufficient at the walls, results have confirmed the ability of the IBM to capture equivalent turbulent structures and to predict averaged flow profiles in similar agreement with experimental result by comparison with body-conforming simulations. The linear reconstruction of the velocity has been compared to a power-law based reconstruction, which complies with the logarithmic profile of the boundary layer. Both instantaneous and averaged quantities have been found to be sensitive to the type of reconstruction. The linear corrections scheme leads to a better prediction of the energy and mixing levels. In the light of these results, the current approach

can be considered as applicable to flows with particle transports or chemical reactions, when time averaged characteristics are of interest. For applications where a fine resolution of the boundary layer is necessary, it is recommended to use adaptive refinement to achieve the desired resolution of boundary layer flow.

Acknowledgments

The financial support of the Irish Research Council (IRC) under its Employment Based Research Scholarship programme (Grant No. EBPPG/2013/63) is acknowledged. Computation for all numerical simulation were carried out using the FIONN cluster of the SFI/HEA Irish Centre for High-End Computing (ICHEC) through dceng002b and dceng005b Class B Projects.

References

- [1] E. Ferrer, R.H.J. Willden, A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes, *J. Comput. Phys.* 231 (21) (2012) 7037–7056. <http://dx.doi.org/10.1016/j.jcp.2012.04.039>.
- [2] L. Ramirez, C. Foulquié, X. Nogueira, S. KHELLADI, J.-C. Chassaing, I. Colominas, New high-resolution-preserving sliding mesh techniques for higher-order finite volume schemes, *Comput. & Fluids* 118 (2015) 114–130. <http://dx.doi.org/10.1016/j.compfluid.2015.06.008>.
- [3] B. Zhang, C. Liang, A simple, efficient, and high-order accurate curved sliding-mesh interface approach to spectral difference method on coupled rotating and stationary domains, *J. Comput. Phys.* 295 (2015) 147–160. <http://dx.doi.org/10.1016/j.jcp.2015.04.006>. arXiv:1503.02198v2.
- [4] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods, *J. Comput. Phys.* 230 (17) (2011) 6664–6687. <http://dx.doi.org/10.1016/j.jcp.2011.05.003>.
- [5] M. Meldi, E. Vergnault, P. Sagaut, An arbitrary Lagrangian–Eulerian approach for the simulation of immersed moving solids with Lattice Boltzmann Method, *J. Comput. Phys.* 235 (2013) 182–198. <http://dx.doi.org/10.1016/j.jcp.2012.10.014>.
- [6] B. Erzincanlı, M. Sahin, An arbitrary Lagrangian–Eulerian formulation for solving moving boundary problems with large displacements and rotations, *J. Comput. Phys.* 255 (2013) 660–679. <http://dx.doi.org/10.1016/j.jcp.2013.08.038>.
- [7] S. Basting, A. Quaini, S. Čanić, R. Glowinski, Extended ALE Method for fluid–structure interaction problems with large structural displacements, *J. Comput. Phys.* 331 (2017) 312–336. <http://dx.doi.org/10.1016/j.jcp.2016.11.043>.
- [8] Y. Wu, X.-C. Cai, A fully implicit domain decomposition based ALE framework for three-dimensional fluid–structure interaction with application in blood flow computation, *J. Comput. Phys.* 258 (2014) 524–537. <http://dx.doi.org/10.1016/j.jcp.2013.10.046>.
- [9] C.S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* 10 (1972) 252–271.
- [10] M.-C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* 160 (2) (2000) 705–719. <http://dx.doi.org/10.1006/jcph.2000.6483>.
- [11] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* 105 (2) (1993) 354–366. <http://dx.doi.org/10.1006/jcph.1993.1081>.
- [12] P. Angot, C.-h. Bruneau, P. Fabrie, Numerische Mathematik A penalization method to take into account obstacles in incompressible viscous flows, *Numer. Math.* 81 (1999) 497–520.
- [13] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, *J. Comput. Phys.* 230 (2) (2011) 329–348. <http://dx.doi.org/10.1016/j.jcp.2010.09.017>.
- [14] M. Gazzola, P. Chatelain, W.M. van Rees, P. Koumoutsakos, Simulations of single and multiple swimmers with non-divergence free deforming geometries, *J. Comput. Phys.* 230 (19) (2011) 7093–7114. <http://dx.doi.org/10.1016/j.jcp.2011.04.025>.
- [15] S.A. Ghaffari, S. Viazzo, K. Schneider, P. Bontoux, Simulation of forced deformable bodies interacting with two-dimensional incompressible flows: Application to fish-like swimming, *Int. J. Heat Fluid Flow* 51 (2015) 88–109. <http://dx.doi.org/10.1016/j.ijheatfluidflow.2014.10.023>.
- [16] A. Sarthou, P. Angot, J.-p. Caltagirone, A. Sarthou, P. Angot, J.-p.C. T. Sub-mesh, The sub-mesh penalty method in: 5th International Symposium on Finite Volumes for Complex Applications, 2009, p. 633.
- [17] C. Introini, M. Belliard, C. Fournier, A second order penalized direct forcing for hybrid Cartesian/immersed boundary flow simulations, *Comput. & Fluids* 90 (2014) 21–41. <http://dx.doi.org/10.1016/j.compfluid.2013.10.044>.

- [18] F. Chantalat, C.H. Bruneau, C. Galusinski, A. Iollo, Level-set, penalization and cartesian meshes: A paradigm for inverse problems and optimal design, *J. Comput. Phys.* 228 (17) (2009) 6291–6315. <http://dx.doi.org/10.1016/j.jcp.2009.05.017>.
- [19] N.K. Kevlahan, J.-M. Ghidaglia, Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization, *Eur. J. Mech. B Fluids* 20 (2001) 333–350.
- [20] S. Vincent, J.C. Brändle de Motta, A. Sarthou, J.-L. Estivaleres, O. Simonin, E. Climent, A Lagrangian VOF tensorial penalty method for the DNS of resolved particle-laden flows, *J. Comput. Phys.* 256 (2014) 582–614. <http://dx.doi.org/10.1016/j.jcp.2013.08.023>.
- [21] O. Boiron, G. Chiavassa, R. Donat, A high-resolution penalization method for large Mach number flows in the presence of obstacles, *Comput. & Fluids* 38 (3) (2009) 703–714. <http://dx.doi.org/10.1016/j.compfluid.2008.07.003>.
- [22] A. Piquet, O. Roussel, A. Hadjadj, A comparative study of Brinkman penalization and direct-forcing immersed boundary methods for compressible viscous flows, *Comput. & Fluids* 136 (2016) 272–284. <http://dx.doi.org/10.1016/j.compfluid.2016.06.001>.
- [23] W.X. Huang, C.B. Chang, H.J. Sung, An improved penalty immersed boundary method for fluid-flexible body interaction, *J. Comput. Phys.* 230 (12) (2011) 5061–5079. <http://dx.doi.org/10.1016/j.jcp.2011.03.027>.
- [24] T. Engels, D. Kolomenskiy, K. Schneider, J. Sesterhenn, Numerical simulation of fluid-structure interaction with the volume penalization method, *J. Comput. Phys.* 281 (2015) 96–115. <http://dx.doi.org/10.1016/j.jcp.2014.10.005>.
- [25] S. Vincent, A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février, C. Mignot, G. Pianet, Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns, *J. Comput. Phys.* 230 (4) (2011) 956–983. <http://dx.doi.org/10.1016/j.jcp.2010.10.006>.
- [26] P. Horgue, M. Prat, M. Quintard, A penalization technique applied to the Volume-Of-Fluid method: Wettability condition on immersed boundaries, *Comput. & Fluids* 100 (2014) 255–266. <http://dx.doi.org/10.1016/j.compfluid.2014.05.027>.
- [27] B. Blais, M. Lassaingne, C. Goniva, L. Fradette, F. Bertrand, A semi-implicit immersed boundary method and its application to viscous mixing, *Comput. Chem. Eng.* 85 (2016) 136–146. <http://dx.doi.org/10.1016/j.compchemeng.2015.10.019>.
- [28] E. Arquis, J.P. Caltagirone, Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: application à la convection naturelle, *C. R. Acad. Sci. Paris* 2 299 (1984) 1–4.
- [29] T. Ikeno, T. Kajishima, Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations, *J. Comput. Phys.* 226 (2) (2007) 1485–1508. <http://dx.doi.org/10.1016/j.jcp.2007.05.028>.
- [30] A. Mark, B.G. van Wachem, Derivation and validation of a novel implicit second-order accurate immersed boundary method, *J. Comput. Phys.* 227 (13) (2008) 6660–6680. <http://dx.doi.org/10.1016/j.jcp.2008.03.031>.
- [31] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2) (2003) 593–623. <http://dx.doi.org/10.1016/j.jcp.2003.07.024>.
- [32] I. Borazjani, L. Ge, F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies, *J. Comput. Phys.* 227 (16) (2008) 7587–7620. <http://dx.doi.org/10.1016/j.jcp.2008.04.028>.
- [33] A. Calderer, S. Kang, F. Sotiropoulos, Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures, *J. Comput. Phys.* 277 (2014) 201–227. <http://dx.doi.org/10.1016/j.jcp.2014.08.010>.
- [34] H. Jasak, *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows* (Ph.D. thesis), Imperial College London, 1996.
- [35] C.M. Rhie, W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, *Am. Inst. Aeronaut. Astronaut.* 21 (1983) 1525–1532.
- [36] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (1) (2005) 239–261. <http://dx.doi.org/10.1146/annurev.fluid.37.061903.175743>.
- [37] S. Kang, G. Iaccarino, F. Ham, P. Moin, Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method, *J. Comput. Phys.* 228 (18) (2009) 6753–6772. <http://dx.doi.org/10.1016/j.jcp.2009.05.036>.
- [38] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *J. Comput. Phys.* 171 (1) (2001) 132–150. <http://dx.doi.org/10.1006/jcph.2001.6778>.
- [39] P.R. Spalart, W.-H. Jou, M. Strelets, S.R. Allmaras, Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach, in: *Proceedings of the first AFOSR International Conference on DNS/LES*, 1997.
- [40] S.R. Allmaras, F.T. Johnson, P.R. Spalart, Modifications and clarifications for the implementation of the Spalart–Allmaras turbulence model, in: *Seventh International Conference on Computational Fluid Dynamics, ICCFD7-1902*, 2012, pp. 1–11.
- [41] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: *30th Aerospace Sciences Meeting and Exhibit*, Vol. 1 (1), American Institute of Aeronautics and Astronautics, Reston, Virginia, 1992, pp. 5–21. <http://dx.doi.org/10.2514/6.1992-439>. arXiv: 1011.1669v3.
- [42] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, *Comput. & Fluids* 33 (3) (2004) 375–404. [http://dx.doi.org/10.1016/S0045-7930\(03\)00058-6](http://dx.doi.org/10.1016/S0045-7930(03)00058-6).
- [43] G.H. Wannier, A contribution to hydrodynamics of lubrication, *Quart. Appl. Math.* 8 (1950) 1–32.
- [44] L.F. Richardson, The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a Masonry Dam, *Trans. R. Soc. London Ser. A* 210 (1910) 307–357.
- [45] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, *Fluid Mech.* 79 (1977) 231–256.
- [46] B.D.J. Tritton, Experiments on the flow past a circular cylinder at low Reynolds numbers, *J. Fluid Mech.* 6 (1959) 547–567.
- [47] K. Taira, T. Colonius, The immersed boundary method: A projection approach, *J. Comput. Phys.* 225 (2) (2007) 2118–2137. <http://dx.doi.org/10.1016/j.jcp.2007.03.005>.
- [48] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, *J. Comput. Phys.* 216 (2) (2006) 454–493. <http://dx.doi.org/10.1016/j.jcp.2005.12.016>.
- [49] L. Parussini, V. Pediroda, Fictitious domain approach with hp -finite element approximation for incompressible fluid flow, *J. Comput. Phys.* 228 (10) (2009) 3891–3910. <http://dx.doi.org/10.1016/j.jcp.2009.02.019>.
- [50] A. Gilmanov, F. Sotiropoulos, E. Balaras, A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids, *J. Comput. Phys.* 191 (2) (2003) 660–669. [http://dx.doi.org/10.1016/S0021-9991\(03\)00321-8](http://dx.doi.org/10.1016/S0021-9991(03)00321-8).
- [51] C.C. Liao, Y.W. Chang, C.A. Lin, J.M. McDonough, Simulating flows with moving rigid boundary using immersed-boundary method, *Comput. & Fluids* 39 (1) (2010) 152–167. <http://dx.doi.org/10.1016/j.compfluid.2009.07.011>.
- [52] 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. <http://dx.doi.org/10.1016/j.jcp.2011.06.003>. arXiv: NIHMS150003.
- [53] J. Yang, F. Stern, A simple and efficient direct forcing immersed boundary framework for fluid-structure interactions, *J. Comput. Phys.* 231 (15) (2012) 5029–5061. <http://dx.doi.org/10.1016/j.jcp.2012.04.012>.
- [54] R.-Y. Li, C.-M. Xie, W.-X. Huang, C.-X. Xu, An efficient immersed boundary projection method for flow over complex/moving boundaries, *Comput. & Fluids* 140 (2016) 122–135. <http://dx.doi.org/10.1016/j.compfluid.2016.09.017>.
- [55] H. Dütsch, F. Durst, S. Becker, H. Lienhart, Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan-Carpenter numbers, *Fluid Mech.* 360 (1998) 249–271.
- [56] C.Y. Ge, J.J. Wang, X.P. Gu, L.F. Feng, CFD simulation and PIV measurement of the flow field generated by modified pitched blade turbine impellers, *Chem. Eng. Res. Des.* 92 (6) (2014) 1027–1036. <http://dx.doi.org/10.1016/j.cherd.2013.08.024>.
- [57] J.J. Derksen, M.S. Doelman, H.E.A. van den Akker, Three-dimensional measurements in the baffle region of a turbulently stirred tank, *Exp. Fluids* 27 (1999) 522–532.
- [58] P.R. Spalart, S. Deck, M.L. Shur, K.D. Squires, M.K. Strelets, A. Travin, A new version of detached-eddy simulation, resistant to ambiguous grid densities, *Theor. Comput. Fluid Dyn.* 20 (3) (2006) 181–195. <http://dx.doi.org/10.1007/s00162-006-0015-0>.
- [59] J.-I. Choi, R.C. Oberoi, J.R. Edwards, J.A. Rosati, An immersed boundary method for complex incompressible flows, *J. Comput. Phys.* 224 (2) (2007) 757–784. <http://dx.doi.org/10.1016/j.jcp.2006.10.032>.
- [60] P.H. Chang, C.C. Liao, H.W. Hsu, S.H. Liu, C.A. Lin, Simulations of laminar and turbulent flows over periodic hills with immersed boundary method, *Comput. & Fluids* 92 (2014) 233–243. <http://dx.doi.org/10.1016/j.compfluid.2013.10.043>.
- [61] H. Werner, H. Wengle, Large-Eddy simulation of turbulent flow over and around a cube in a plate channel, in: *Turbulent Shear Flows 8*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993, pp. 155–168.
- [62] H. Schlichting, K. Gersten, *Boundary-Layer Theory*, McGraw-Hill, Springer Berlin Heidelberg, Berlin, Heidelberg, 2017. <http://dx.doi.org/10.1007/978-3-662-52919-5>. arXiv: 1011.1669v3.
- [63] D.B. Spalding, A single formula for the “Law of the Wall”, *J. Appl. Mech.* 28 (3) (1961) 455. <http://dx.doi.org/10.1115/1.3641728>.
- [64] H. Hartmann, J. Derksen, C. Montavon, J. Pearson, I. Hamill, H. van den Akker, Assessment of large eddy and RANS stirred tank simulations by means of LDA, *Chem. Eng. Sci.* 59 (12) (2004) 2419–2432. <http://dx.doi.org/10.1016/j.ces.2004.01.065>.
- [65] R. Verzicco, M. Fatica, G. Iaccarino, P. Orlandi, Flow in an impeller-stirred tank using an immersed-boundary method, *AIChE J.* 50 (6) (2004) 1109–1118. <http://dx.doi.org/10.1002/aic.10117>.