



HAL
open science

A volume-of-solid implicit volume penalty method for moving-body flows

Iason Tsetoglou, Mélody Cailler, Pierre Bénard, Ghislain Lartigue, Vincent Moureau, Julien Réveillon

► **To cite this version:**

Iason Tsetoglou, Mélody Cailler, Pierre Bénard, Ghislain Lartigue, Vincent Moureau, et al.. A volume-of-solid implicit volume penalty method for moving-body flows. *International Journal for Numerical Methods in Fluids*, 2024, 10.1002/fld.5334 . hal-04684737v1

HAL Id: hal-04684737

<https://hal.science/hal-04684737v1>

Submitted on 3 Sep 2024 (v1), last revised 10 Oct 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

RESEARCH ARTICLE

A volume-of-solid implicit volume penalty method for moving-body flows

Iason Tsetoglou*¹ | Mélody Cailler² | Pierre Bénard¹ | Ghislain Lartigue¹ | Vincent Moureau¹ | Julien Réveillon¹

¹CORIA, INSA Rouen Normandie,
Normandie Université,
Saint-Étienne-du-Rouvray, France
²Digital Sciences and Technologies
Department, Safran Tech, Châteaufort,
France

Correspondence

*Iason Tsetoglou, CORIA, INSA Rouen
Normandie, Normandie Université.
Email: iason.tsetoglou@gmail.com

Abstract

An original Immersed Boundary Method for solving moving body flows is proposed. This method couples i) a Lagrangian Volume-of-Solid description of the solid object avoiding conservation issues and ii) a robust implicit volume penalty forcing embedded in a low-Mach number projection method to account for the solid's impact on the fluid dynamics. A new composite velocity field is introduced to describe both solid and fluid domains in a single set of governing equations. The accuracy of the method has been assessed on several academic cases, involving stationary or moving bodies and with different mesh resolutions. The predicted forces on the solid are in excellent agreement with body-fitted reference cases. The system of equations is also proven to be fully mass conservative. Application of the method on a two-dimensional vertical axis turbine case shows a 30% reduction in computational cost compared to a body-fitted method.

KEYWORDS:

immersed boundary methods, implicit volume penalty, volume-of-solid, large-eddy simulation

1 | INTRODUCTION

Many engineering applications involve fluid-structure interactions (FSI). Such effects can be found in wind turbines, wind-induced vibrations of tall buildings, aeroelastic response of airplanes and even biological flows like blood flow in the arteries. In Computational Fluid Dynamics (CFD), for relatively simple geometries of the solid body the usual approach is to generate a regular Cartesian computational grid, and for more complex geometries, unstructured grids are preferred despite being more costly to generate. In both grid types, the computational nodes adopt the shape of the solid's surface, i.e. the fluid-solid interface, leading to an explicitly meshed surface and these approaches are referred to as body-fitted methods. The main argument that one can pin against them is the difficult treatment of moving geometries. Small displacements of the solid geometry can be

treated with the deformation of the grid and relying on the Arbitrary Lagrangian-Eulerian (ALE) method which is widely used for simulating FSI problems. However, grid deformations due to large displacements would degrade the quality of the mesh with stretched computational cells, hence a complete regeneration of a new grid at specific intervals is needed to follow the solid surface. Such procedures are extremely costly and can result in mesh distortions and mesh interpolation errors due to deforming mesh and re-meshing.

Immersed Boundary methods (IB or IBM) represent an attractive alternative for simulating FSI problems involving complex geometries and arbitrarily large movements. The term Immersed Boundary Method was first used as a reference to the method developed by Peskin (1972) to simulate cardiac mechanics and associated blood flow¹. The novel feature of that method lies in the use of a Cartesian grid for the entire simulation without conforming to the heart's geometry. A dedicated procedure was formulated to impose the effects of the immersed boundary on the flow. The basic idea of the IBM is to carry out the simulation on a relatively simple mesh which extends inside of the region where the solid geometry would be and introduces a forcing term in the fluid's governing equations to simulate the interaction between the solid and the fluid. Thus, one is able to impose boundary conditions on surfaces that are not aligned with the grid and to simulate arbitrarily large movements of complex solid boundaries on a fixed grid. The key point of these methods is the formulation of the forcing term imposed at the immersed interface so that it does not degrade the precision and the conservation properties of the solver while accurately representing the fluid-solid interaction.

Since first introduced by Peskin, IBM has been extended to various applications in scientific and engineering fields. Numerous modifications and refinements have been proposed and a number of variants of this approach now exist focusing on the definition of the forcing term. The IBM can be split into two main categories, being the continuous approach and the discrete forcing approach. An overview of the different approaches of IB methods can be found in numerous reviews^{2,3,4,5,6}.

Continuous forcing approaches integrate the forcing term into the continuous Navier-Stokes equations for the whole domain Ω before the discretization happens. They include the first IBM developed by Peskin¹ and other methods like the one described by Goldstein et al.⁷. The forcing term is usually based on a constitutive law, like Hook's law, for elastic deformable immersed bodies. For rigid bodies, a method similar to Goldstein's feedback approach can be used as presented by Angot et al.⁸. These approaches belong to the class of penalization methods, where the forcing term is based on the Brinkman equation of porous media. In discrete forcing approaches we introduce the forcing term to the already discretized momentum equation, acting only in cells near the fluid-solid interface. This is the so-called direct forcing IBM^{9,10,11,12}. The forcing term is computed in a way that it compensates the difference between the predicted velocity at the immersed boundary and the desired imposed velocity. The general advantage of continuous forcing is that the formulation is almost independent of the discretization scheme, which makes its integration easier to a pre-existing Navier-Stokes equations. The use of a forcing term to implicitly reach the target solid velocity at the interface weakens the precision on the predicted velocity fields. The opposites are true for the discrete forcing

methods since the forcing term is dependent on the spacial discretization making the implementation not trivial. Although, the solid velocity and the velocity gradient near the immersed boundary can usually be directly imposed leading to more accurate flow profiles in the fluid domain, in particular in high Reynolds number flows.

An essential part of the IBM is the procedure used to immerse the object into the computational mesh and the determination of the position of the nodes with respect to the object's surface in order to separate the computational cells into 'fluid cells' outside the object, 'solid cells' inside the object and 'interface cells' partially inside¹³. For the purpose of handling mobile interfaces on static meshes, two well-established numerical tools used are the Level-Set technique¹⁴ and the Volume-of-Fluid (VOF) approach¹⁵. Usually applied for tracking interfaces between fluids, these techniques may be adapted for solid-fluid interfaces. In that case, the term Volume-of-Solid (VOS) can be used instead of VOF to indicate the presence of solid-fluid interface. Liu et al.¹⁶ have used the VOS approach to represent the solid coupled with a direct forcing IB method, similar to the works of Ng¹⁷. In their work, the solid surface is represented by Lagrangian points and the cell type (fluid, solid or interface) is determined with a simple ray-tracing technique. The solid volume fraction field ϕ_s is then computed on the partially or fully immersed cells through a Gaussian quadrature integration. This volume fraction is then used to determine the value of the forcing term by correcting the velocity inside the solid region to a mean value of the predicted fluid velocity and the imposed solid velocity weighted by $(1 - \phi_s)$ and ϕ_s , respectively. Another feature of their method is the inclusion of the same forcing term to the pressure Poisson equation as a source term to impose the solid velocity at the same time as the divergence-free condition. Morente et al.¹⁸ have used a VOF approach coupled with a penalty IBM for the simulation of bubbly flows where spherical bubbles are considered as moving penalized obstacles interacting with the fluid. From the position of the bubble surface a volume fraction field is defined separating the two phases and acts as the penalty mask of the forcing term. However, instead of the usual one-fluid penalized momentum equation model⁸, a two-fluid Eulerian multi-phase frame is used where the momentum equations of both phases are coupled to a single equation. The recent work of Kemm et al.¹⁹ has also used the VOF method to represent the diffuse interface between the fluid and solid domains and to provide a reduced system of equations based on the gas phase with the addition of source terms related to the presence/movement of the solid body.

The IBM presented in this work²⁰ bares similar elements to the aforementioned methods. The solid is represented by a set of Lagrangian particles containing the information of the solid volume through which a scalar field of the solid volume fraction ϕ_s can be defined to separate the solid and fluid domains. The method couples i) this Lagrangian Volume-of-Solid (VOS) description of the body avoiding mass and momentum conservation issues and ii) a robust implicit volume penalty forcing⁸ embedded in a low-Mach number projection method to account for the solid's impact on the fluid dynamics. Incorporating the VOS approach directly into the governing equations results in extra source terms in the mass and momentum conservation equations. These terms represent the internal mass effect of a moving object as discussed in Suzuki and Inamuro²¹. This leads to a trivial yet accurate computation of the resistive forces exerted on the solid by the fluid through the IB forcing term.

This work is organized in the following manner. In Section 2 the coupling of a Lagrangian VOS approach with an implicit formulation of the penalty IBM is presented. To validate the accuracy of the method, flow problems involving stationary or moving solid geometries are examined in Section 3. In Section 4, the method is applied on a 2D vertical axis turbine case where the results and the computational performances of the method are compared to a reference body-fitted ALE solution. Finally, concluding remarks on the method and perspectives are given in Section 5.

2 | NUMERICAL METHOD

2.1 | Volume-of-solid implicit volume penalty method

To expand on the method's main components, three tasks can be defined:

- Coupling the VOS approach with an IBM: Incorporating the solid volume fraction field from the VOS approach into the Navier-Stokes equations allowing to define a composite velocity field, computed as the mean of the solid and fluid velocities, weighted by their respective volume fractions. Using the composite velocity leads to a new system of equations capable of describing the evolution of both fluid and solid domains at once. These resemble the pure fluid equations but additional mass and momentum source terms appear to represent the solid movement.
- Modelling the implicit penalty forcing term: Expressing the penalty term in an implicit manner so that its contribution can be split in the prediction step and the correction step of a projection scheme. The contribution in the correction stage serves to impose the solid velocity in the solid region at the same time as the incompressibility constraint is satisfied. This leads to the challenge of solving a modified pressure Poisson equation.
- Representing the solid body in a Lagrangian manner: Representing the solid object as a set of Lagrangian particles containing an elementary quantity of solid volume. First, the immersed object is displaced by moving the Lagrangian particles according to the prescribed motion and then projecting the solid volume onto the Eulerian grid creates the local solid volume fraction field.

Taking into account the techniques used, this method will be referred to as Volume-of-Solid Implicit Volume Penalty method (VOS-IVP). The method will be explained in detail in the following sections.

2.2 | Coupling the VOS approach with an IBM

2.2.1 | Immersed Boundary Method for Large-Eddy Simulations

The main advantage of the IBM is that body-conforming meshes are not necessary. Instead, the computational domain Ω includes both fluid Ω_f and solid Ω_s domains, so $\Omega = \Omega_f \cup \Omega_s$ (as seen in Figure 1). Thanks to the continuous mesh in the solid region,

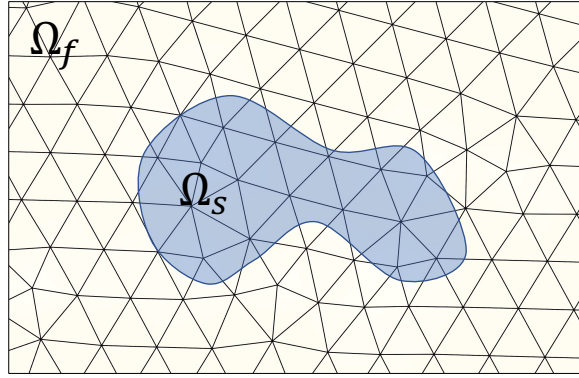


Figure 1 Mesh used with IBM including both fluid and solid domains.

there is no need for re-meshing in the case of a moving immersed object, which makes it an attractive solution for simulating moving bodies in fluid flow.

Let us introduce the scalar field of the local solid volume fraction $\phi_s(\mathbf{x}, t)$ as the fraction of the volume occupied by the solid in a computational cell $V_{s,i}$ over the total cell volume V_i at time t , defined as:

$$\phi_{s,i}(t) = \frac{V_{s,i}(t)}{V_i}, \quad (1)$$

where it takes the following values:

$$\phi_s = \begin{cases} 1 & \text{in } \Omega_s, \\ 0 & \text{in } \Omega_f. \end{cases} \quad (2)$$

Equivalently we can define the local fluid volume fraction as $\phi_f = 1 - \phi_s$.

This study is carried out with Large-Eddy simulations (LES) in which the smallest vortical structures are not resolved but modelled. Hence the physical quantities pass through a filtering operation. The filtering operator, which consists of projecting a field on the LES grid, is expressed as $\tilde{\cdot}$. Usually the LES Navier-Stokes equations are expressed in the case of a single fluid or phase occupying the whole computational domain. In our case, the domain is occupied by a fluid phase and a solid phase. Thus, the flow equations that describe the evolution of the fluid quantities in a control volume need to be modified to be applied to the fraction of the control volume occupied by the fluid. This can be achieved by using the finite volume integration with a phase indicator. For example, the transport of the physical property Ψ of a phase k , with the local volume fraction as a phase indicator, and advection velocity $\tilde{\mathbf{u}}_k$, and assuming that the phase indicator is transported at the same speed, can be written as:

$$\int_{V_k} \frac{\partial}{\partial t} (\tilde{\Psi}_k) dV_k + \int_{V_k} \nabla \cdot (\tilde{\Psi}_k \tilde{\mathbf{u}}_k) dV_k = 0, \quad (3)$$

$$\Rightarrow \int_V \frac{\partial}{\partial t} (\phi_k \tilde{\Psi}_k) dV + \int_V \nabla \cdot (\phi_k \tilde{\Psi}_k \tilde{\mathbf{u}}_k) dV = 0. \quad (4)$$

In our context, this manipulation introduces the local fluid volume fraction ϕ_f in front of the fluid quantities while integrating in the whole domain (like in the case of one-phase problems). The modified LES flow equations for the fluid phase with a constant density and with the addition of the IBM term, read as:

$$\frac{\partial}{\partial t} (\phi_f) + \nabla \cdot (\phi_f \tilde{\mathbf{u}}_f) = 0, \quad (5)$$

$$\frac{\partial}{\partial t} (\phi_f \tilde{\mathbf{u}}_f) + \nabla \cdot (\phi_f \tilde{\mathbf{u}}_f \otimes \tilde{\mathbf{u}}_f) = -\frac{1}{\rho} \nabla \tilde{P} + \frac{1}{\rho} \nabla \cdot (\phi_f \tilde{\boldsymbol{\tau}}_f) + \mathbf{f}_{\text{IB}}. \quad (6)$$

where $\tilde{\mathbf{u}}_f$ is the fluid velocity, \tilde{P} the pressure and ρ the constant fluid density. The viscous stress tensor can be expressed as:

$$\tilde{\boldsymbol{\tau}}_f = \mu_{\text{eff}} \left(\nabla \tilde{\mathbf{u}}_f + \nabla \tilde{\mathbf{u}}_f^T - \frac{2}{3} (\nabla \cdot \tilde{\mathbf{u}}_f) \mathbf{I} \right), \quad (7)$$

where \mathbf{I} is the identity tensor and μ_{eff} the effective fluid dynamic viscosity evaluated as the sum of the molecular and turbulent viscosities. The turbulent contribution in this study was obtained from the Dynamic Smagorinsky model^{22,23,24}. The last term in Equation 7 is equal to zero due to the incompressibility constraint $\nabla \cdot \tilde{\mathbf{u}}_f = 0$.

Finally, the term \mathbf{f}_{IB} represents any additional volumetric momentum sources such as the forcing term of the IBM, the expression of which will be explored in section 2.3.1. Note that the LES-filtering notation will be dropped for the rest of this work.

The original VOS-IVP method was coupled with the incompressible solver of the YALES2 library²⁵ which solves the low-Mach number Navier-Stokes equations for turbulent flows on unstructured grids using a projection method for pressure-velocity coupling²⁶. A central 4th-order numerical scheme is used for spatial discretization and a 4th-order Runge-Kutta like scheme for the time integration. The Poisson equation is solved with a Deflated Preconditioned Conjugate Gradient (DPCG) solver²⁷. Concerning parallelization, YALES2 is based on a multi-level domain decomposition and can utilize both MPI and coarse-grain OpenMP paradigms²⁵. In this work, full MPI was used for the simulations as it was the most efficient. The domain decomposition relies on the graph coloring library METIS, while YALES2 manages the computational load balancing. For increased efficiency, cache-blocking is ensured by splitting the grid of each MPI rank into cell groups of a few thousand elements.

2.2.2 | Governing equations

In the previous section the flow equations were presented for the evolution of the fluid physical quantities such as the fluid velocity \mathbf{u}_f . We can also define a vector field \mathbf{u}_s containing information about the local solid velocity. This allows the introduction of a

new composite velocity field computed as the addition of the fluid and solid velocities weighted by the fluid and solid volume fraction, respectively, as follows:

$$\mathbf{u} = \phi_f \mathbf{u}_f + \phi_s \mathbf{u}_s. \quad (8)$$

Using the relation in Equation 8 and the fact that $\phi_s + \phi_f = 1$, we can rearrange the Navier-Stokes equations so that they describe the evolution of the composite velocity \mathbf{u} . The composite velocity allows to describe the evolution of both the fluid and the solid quantities on the Eulerian mesh via a single momentum conservation equation and improves numerical stability of the solver, since it does not need specific treatment at the interface between the two phases. For a sharp representation of the solid volume fraction the convective cross terms that include the product $\phi_s \phi_f (\mathbf{u}_s - \mathbf{u}_f)$ can be neglected since $\phi_s \phi_f = 0$ away from the solid/fluid interface and $\mathbf{u}_f \approx \mathbf{u}_s \approx \mathbf{u}$ at the interface (shown in Appendix A). This gives rise to a new conservative system of equations for both the fluid and the solid domains at once:

$$\nabla \cdot \mathbf{u} = \underbrace{\frac{\partial}{\partial t} (\phi_s) + \nabla \cdot (\phi_s \mathbf{u}_s)}_{\mathcal{Q}_s}, \quad (9)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho} \nabla P + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \underbrace{\frac{\partial}{\partial t} (\phi_s \mathbf{u}_s) + \nabla \cdot (\phi_s \mathbf{u}_s \otimes \mathbf{u}_s)}_{\mathcal{P}_s} + \mathbf{f}_{\text{IB}}. \quad (10)$$

Detailed derivation of Equation 9 and Equation 10 is shown in Appendix A. These equations are similar to the pure fluid equations except for the additional source terms on the right-hand side representing the solid movement and/or deformation. The mass source, noted as \mathcal{Q}_s , represents the change of solid volume in space and time, and the momentum source \mathcal{P}_s represents the solid acceleration projected onto the Eulerian non-conforming grid. In the context of the usual one-fluid immersed boundary model, the term \mathcal{P}_s would be equivalent to the momentum needed to move the fluid found at the interior of the immersed body according to the solid movement and will be further discussed in section 2.3.3.

2.2.3 | Discretized mass and momentum source terms

In the VOS-IVP method, the incompressible Navier-Stokes equations can be semi-discretized as follows:

$$\nabla \cdot \mathbf{u}^{n+1} = \mathcal{Q}^{n+1}, \quad (11)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla \cdot (\mathbf{u}^n \otimes \mathbf{u}^n) - \frac{1}{\rho} \nabla P^{n+1/2} + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau}^n + \mathcal{P}_s^{n+1} + \mathbf{f}_{\text{IB}}^{n+1}, \quad (12)$$

where n denotes the iteration of the previous time-step and $n + 1$ the current time-step.

The scalar quantities in the YALES2 solver are advanced from time-step $n - 1/2$ to $n + 1/2$ or from $n + 1/2$ to $n + 3/2$. The same happens to the solid particles, so after the re-localisation of the particles on the Eulerian mesh, the new solid volume fraction $\phi_s^{n+3/2}$ is computed according to the prescribed motion. Then it is computed at the time-step $n + 1$ by:

$$\phi_s^{n+1} = \frac{1}{2} [\phi_s^{n+1/2} + \phi_s^{n+3/2}] . \quad (13)$$

The new solid velocity field is computed directly at $n + 1$. This way, we can express the VOS mass and momentum sources at the time-step $n + 1$ as:

$$\mathcal{Q}_s^{n+1} = \nabla \cdot \mathbf{u}^{n+1} = \left[\frac{\phi_s^{n+3/2} - \phi_s^{n+1/2}}{\Delta t} \right] + \nabla \cdot (\phi_s^{n+1} \mathbf{u}_s^{n+1}) , \quad (14)$$

$$\mathcal{P}_s^{n+1} = \left[\frac{\phi_s^{n+3/2} \mathbf{u}_s^{n+1} - \phi_s^{n+1/2} \mathbf{u}_s^n}{\Delta t} \right] + \nabla \cdot (\phi_s^{n+1} \mathbf{u}_s^{n+1} \mathbf{u}_s^{n+1}) . \quad (15)$$

2.3 | Modelling the penalty forcing term

2.3.1 | Implicit volume penalisation and modified pressure Poisson equation

This new method utilises a volume penalty approach for the IB forcing term appearing in Equation 10. The penalty term guaranties that the composite velocity \mathbf{u} remains equal to the imposed solid velocity inside the solid region through a simple Dirichlet type boundary condition. At this time, no wall-law model has been implemented. The penalty forcing term reads:

$$\mathbf{f}_{\text{IB}} = \frac{\chi_s}{\eta} (\mathbf{u}_s - \mathbf{u}) . \quad (16)$$

The penalty mask is an Heaviside function of the solid fraction at each new time-step:

$$\chi_s = \begin{cases} 1 & \text{if } \phi_s > 0.5 , \\ 0 & \text{else.} \end{cases} \quad (17)$$

The penalty parameter is set as a function of the time-step:

$$\eta = \alpha \Delta t , \quad (18)$$

where $0 < \alpha \leq 1$ is called the penalisation time-step ratio.

Usually this penalty forcing term is applied solely on the intermediate velocity \mathbf{u}^* when a projection method is used²⁶, and the final velocity at time-step $n + 1$ is then modified by the correction step with the new pressure field, so the boundary condition and the continuity constraint are not satisfied at the same time. In the present method, as the method's name suggest, the penalty

term is expressed implicitly using the final unknown velocity field. This allows the forcing term to split into a contribution in the prediction step, \mathbf{f}^* , and a contribution in the correction step, \mathbf{f}^{corr} , as shown below:

$$\begin{aligned}\mathbf{f}_{\text{IB}}^{n+1} &= \frac{\chi_s}{\eta} (\mathbf{u}_s^{n+1} - \mathbf{u}^{n+1}) \\ &= \underbrace{\frac{\chi_s}{\eta} (\mathbf{u}_s^{n+1} - \mathbf{u}^*)}_{\mathbf{f}^*} + \underbrace{\frac{\chi_s}{\eta} (\mathbf{u}^* - \mathbf{u}^{n+1})}_{\mathbf{f}^{\text{corr}}}.\end{aligned}\quad (19)$$

The YALES2 solver relies on a modified projection method based on the Helmholtz-Hodge decomposition²⁶ to advance the Navier-Stokes equations in time. Including the previous pressure gradient term in the computation of the intermediate velocity field \mathbf{u}^* leads to a smaller error term in the prediction step making the correction step less computationally demanding²⁸. The intermediate velocity field is computed taking into account the prediction penalty force \mathbf{f}^* :

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\nabla \cdot (\mathbf{u}^* \otimes \mathbf{u}^n) - \frac{1}{\rho} \nabla P^{n-1/2} + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau}^n + \mathcal{P}_s^{n+1} + \frac{\chi_s}{\eta} (\mathbf{u}_s^{n+1} - \mathbf{u}^*) \quad (20)$$

$$\Rightarrow \mathbf{u}^* = \gamma_s^{-1} (\mathbf{u}^n - \Delta t (\nabla \cdot (\mathbf{u}^* \otimes \mathbf{u}^n) + \rho^{-1} \nabla P^{n-1/2} - \rho^{-1} \nabla \cdot \boldsymbol{\tau}^n - \mathcal{P}_s^{n+1})) + (1 - \gamma_s^{-1}) \mathbf{u}_s^{n+1}, \quad (21)$$

where the factor $\gamma_s = (1 + \chi_s \Delta t / \eta)$ is named the penalty density factor, with values of $\gamma_s = 1$ in the fluid and $\gamma_s > 1$ inside the solid.

Before the correction step, the old pressure gradient needs to be subtracted leading to the new intermediate velocity \mathbf{u}^{**} :

$$\mathbf{u}^{**} = \mathbf{u}^* - \left(-\frac{\Delta t}{\gamma_s \rho} \nabla P^{n-1/2} \right). \quad (22)$$

To find the irrotational part of the velocity field we correct the intermediate velocity with the addition of the new pressure term. The correction penalty term is also added:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{**}}{\Delta t} = -\frac{1}{\rho} \nabla P^{n+1/2} + \frac{\chi_s}{\eta} (\mathbf{u}^{**} - \mathbf{u}^{n+1}). \quad (23)$$

By factorisation we can rearrange the previous equation to make the penalty density factor γ_s appear:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{**}}{\Delta t} = -\frac{1}{\gamma_s \rho} \nabla P^{n+1/2}. \quad (24)$$

The new pressure term however needs to be computed first. To achieve this, the operator of divergence is applied to Equation 24 giving rise to a modified pressure Poisson equation:

$$\nabla \cdot \left(\frac{1}{\gamma_s \rho} \nabla P^{n+1/2} \right) = \frac{\nabla \cdot \mathbf{u}^{**}}{\Delta t} - \frac{\nabla \cdot \mathbf{u}^{n+1}}{\Delta t}. \quad (25)$$

Thanks to Equation 25 we manage to penalise the final velocity while satisfying the continuity constraint. We can also observe the appearance of the solid mass source term thanks to Equation 9, where $\nabla \cdot \mathbf{u}^{n+1} = \mathcal{Q}_s^{n+1}$. This guarantees a null divergence for the fluid velocity, $\nabla \cdot \mathbf{u}_f^{n+1} = 0$.

It is also interesting to note that γ_s acts as a modifier for the density, resulting in a higher effective density value inside the solid domain. In contrast to the variable density algorithms used in reactive or two-phase flows for instance, the modified density $\gamma_s \rho$ originates entirely from the penalty method, where only the fluid density ρ is defined. The implied solid density is directly proportional to the chosen penalisation time-step ratio $\alpha = \eta/\Delta t$.

2.3.2 | Discretized pressure Poisson equation

In order to update the pressure to correct the predicted velocity, a Poisson equation needs to be solved. In this method, taking into account the implicit penalty term and the added mass sources, the modified PPE of Equation 25 is obtained. The discretized form in time and space of this equation when integrated over the domain gives a linear system of the form $A p = B$ as follows:

$$\underbrace{\sum_{k \in \mathcal{N}_i} \frac{1}{\gamma_{ik} \rho} \frac{P_k - P_i}{\Delta \mathbf{x}_{ik}} \mathbf{dA}_{ik}}_{A p} = \underbrace{\frac{1}{\Delta t} \sum_{k \in \mathcal{N}_i} \mathbf{u}_{ik}^{**} \cdot \mathbf{dA}_{ik} - \frac{1}{\Delta t} \sum_{k \in \mathcal{N}_i} (\phi_s \mathbf{u}_s)_{ik}^{n+1} \cdot \mathbf{dA}_{ik} + \frac{1}{\Delta t^2} \left(\phi_{s,i}^{n+\frac{3}{2}} - \phi_{s,i}^{n+\frac{1}{2}} \right) dV_i}_{B}, \quad (26)$$

where ik indicates the pair index between two nodes i and k , \mathcal{N}_i is the set of nodes connected to node i , dV_i is the volume of the control volume around the node i , \mathbf{dA}_{ik} is the surface of contact of the control volumes defined by i and k . A schematic representation of the pair-based discretization of YALES2 can be viewed in Figure 2.

The penalty density factor of the pair ik is computed as:

$$\frac{1}{\gamma_{ik}} = \frac{1}{2} \left[\frac{1}{\gamma_i} + \frac{1}{\gamma_k} \right]. \quad (27)$$

2.3.3 | Estimation of the resistive force acting on the body

In the usual IBM framework, the resistive force \mathbf{F} acting on the solid body from the fluid can be computed as the negative sum of the applied forcing terms. This is true in the case of stationary bodies. In the case of a mobile body, the sum of the forcing terms is equal to the total force \mathbf{F}_{tot} experienced by the solid's immersed surface; this includes the forces from the external fluid, \mathbf{F} , and the internal fluid, \mathbf{F}_{in} , as explained by Suzuki et al.²¹. So, to compute the resistive force the following expression is used taking into account the internal force needed to move the fluid inside of the solid domain:

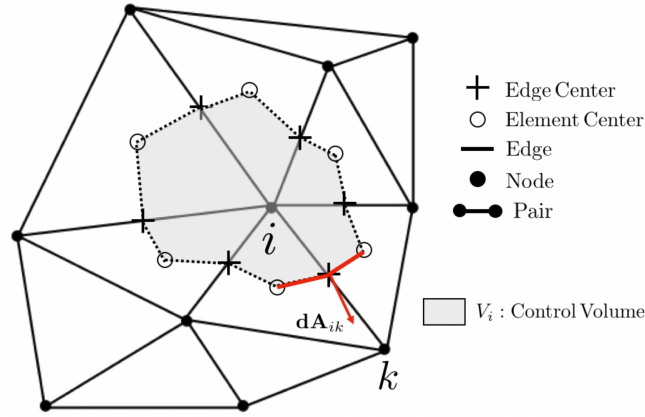


Figure 2 Finite-volume pair-based discretization in YALES2, using node-centered control volumes V_i .

$$\mathbf{F} = - \underbrace{\int_{\Omega} \rho \mathbf{f}_{\text{IB}} dV}_{\mathbf{F}_{\text{tot}}} + \underbrace{\int_{\Omega_s} \rho \frac{d\mathbf{u}}{dt} dV}_{\mathbf{F}_{\text{in}}}. \quad (28)$$

In the VOS-IVP method, as seen in Equation 10, we have the additional forcing term \mathcal{P}_s acting alongside the IB forcing term.

In other words the total force is:

$$\mathbf{F}_{\text{tot}} = - \int_{\Omega} \rho (\mathbf{f}_{\text{IB}} + \mathcal{P}_s) dV. \quad (29)$$

Furthermore, the term \mathcal{P}_s represents the force supplied to the fluid of the solid domain Ω_s so that it follows the prescribed motion of the solid body and is equivalent to the internal force:

$$\mathbf{F}_{\text{in}} = \int_{\Omega} \rho \mathcal{P}_s dV = \int_{\Omega_s} \rho \frac{d\mathbf{u}}{dt} dV. \quad (30)$$

By substituting the expressions of the total and internal forces from Equation 29 and Equation 30, respectively, into the Equation 28, it is shown that the resistive force can be computed by simply integrating the IB forcing term over the volume of the computational domain:

$$\mathbf{F} = - \int_{\Omega} \rho \mathbf{f}_{\text{IB}} dV. \quad (31)$$

2.4 | Representation of the solid body as a set of particles

2.4.1 | Lagrangian framework

For the representation of the immersed body in the method developed in this work, a discretized volume mesh of the solid geometry is needed in the pre-processing stage, which for simplicity is called ‘solid mesh’. This is an unstructured mesh consisting

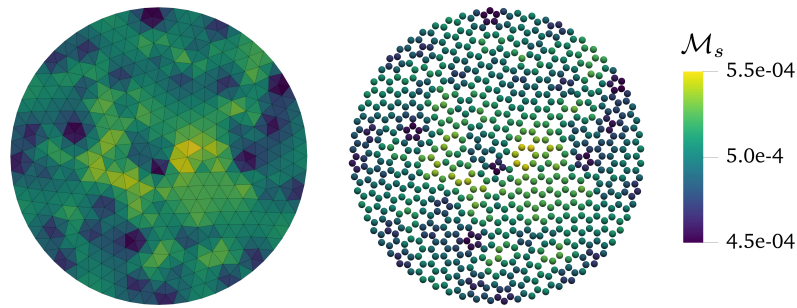


Figure 3 Pre-processing stage for solid body representation where a solid mesh is created and a Lagrangian particle is placed at each element storing the information of the elementary volume and the size of the element it represents.

of triangles in 2D and tetrahedra in 3D with a desired cell size. A set of Lagrangian particles is created by placing a particle at the centre of each cell E^s . In each particle p , the following data is stored:

- the volume V_p of the cell E^s they are placed in,
- the metric \mathcal{M}_s (indicator of the local element size) of the cell E^s they are placed in,
- the coordinates of the nodes, \mathbf{x}_n , at the N vertices of the cell,
- and the coordinates of the barycentre \mathbf{x}_p of the cell where the particle is placed computed as:

$$\mathbf{x}_p = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n. \quad (32)$$

Figure 3 shows an example of a discretized 2D cylinder, coloured by the metric, where the solid mesh on the left is replaced by the Lagrangian particles on the right.

This set of Lagrangian particles is then imported to the simulations in order to represent the solid volume and the solid movement in the Eulerian computational domain. After applying a Lagrangian displacement of the particles, two new fields are computed on the Eulerian mesh. First, the volume contained by the particles is projected on the Eulerian mesh registering the local solid volume contained in each control volume. By dividing this quantity by the total volume of the cell, the local solid volume fraction field can be computed. Last, the solid velocity field is computed according to the prescribed motion of the solid and the solid volume fraction.

2.4.2 | Lagrangian movement of solid particles

When the particles are imported at the beginning of the simulation their initial position \mathbf{x}_p^0 is saved. Based on this initial position a set of transformations can be prescribed to the solid particles under the assumption of a rigid solid body. The current solid

movement types are: rotation, translation and oscillation. These operations can be applied at the same time by simply adding them. However, the rotation operation is always first.

1. **Rotation:** In the case of a prescribed rotary motion, the inputs required are the rotation axis $\mathbf{r} = (r_1 \ r_2 \ r_3)$ as a unit vector, the coordinates of the rotation centre \mathbf{x}^R and the rotational speed ω [rad.s⁻¹]. This way a rotation matrix \mathbf{R} can be defined as:

$$\mathbf{R} = \begin{pmatrix} r_1 r_1 (1 - c) + c & r_1 r_2 (1 - c) - r_3 s & r_1 r_3 (1 - c) + r_2 s \\ r_2 r_1 (1 - c) + r_3 s & r_2 r_2 (1 - c) + c & r_2 r_3 (1 - c) - r_1 s \\ r_3 r_1 (1 - c) - r_2 s & r_3 r_2 (1 - c) + r_1 s & r_3 r_3 (1 - c) + c \end{pmatrix}, \quad (33)$$

where t [s] is the current physical time, $c = \cos(\omega t)$ and $s = \sin(\omega t)$.

Thus, at each time-step the new coordinates of any particle p can be computed as:

$$\mathbf{x}_p = \mathbf{R}(\mathbf{x}_p^0 - \mathbf{x}^R) + \mathbf{x}^R, \quad (34)$$

or in more detail:

$$\mathbf{x}_p = \begin{pmatrix} R_{11}(x_{p,1}^0 - x_1^R) + R_{12}(x_{p,2}^0 - x_2^R) + R_{13}(x_{p,3}^0 - x_3^R) \\ R_{21}(x_{p,1}^0 - x_1^R) + R_{22}(x_{p,2}^0 - x_2^R) + R_{23}(x_{p,3}^0 - x_3^R) \\ R_{31}(x_{p,1}^0 - x_1^R) + R_{32}(x_{p,2}^0 - x_2^R) + R_{33}(x_{p,3}^0 - x_3^R) \end{pmatrix} + \begin{pmatrix} x_1^R \\ x_2^R \\ x_3^R \end{pmatrix}. \quad (35)$$

2. **Translation:** For a simple translation of the solid body at a constant speed the required inputs are the direction unit vector \mathbf{r}_t and the constant movement speed v [m.s⁻¹]. The new coordinates of a particle p at time t are computed as:

$$\mathbf{x}_p = \mathbf{x}_p^0 + (vt)\mathbf{r}_t. \quad (36)$$

3. **Oscillation:** For an oscillating solid body the required inputs are the oscillation axis as the unit vector \mathbf{r}_o , the oscillating amplitude A_o and the oscillating frequency f_o . The oscillation follows a sinusoidal evolution. The new coordinates of a particle p at time t are computed as:

$$\mathbf{x}_p = \mathbf{x}_p^0 + A_o \sin(2\pi f_o t) \mathbf{r}_o. \quad (37)$$

2.4.3 | Projection of Lagrangian solid volume to Eulerian VOS field

During the computation stage, before the advection of the velocity at each iteration, the particles are relocated on the Eulerian mesh according to the prescribed solid motion. Then the volume carried by the particles is projected onto the Eulerian mesh

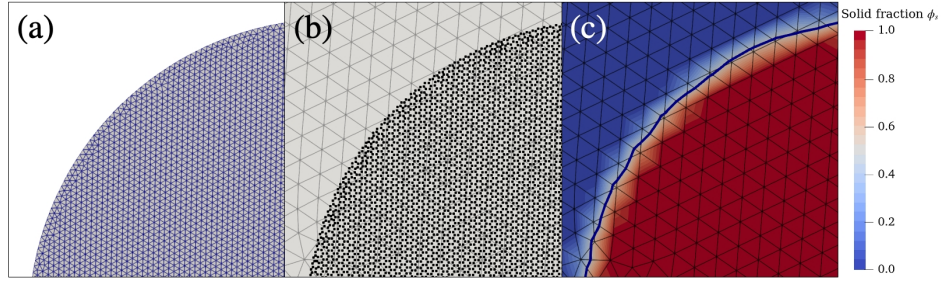


Figure 4 Pre-processing stage for solid body representation: (a) a solid mesh is created, (b) a Lagrangian particle is placed at each element and stores the information of the elementary volume and (c) the volume of the particles are interpolated onto the Eulerian mesh resulting in the computation of the solid volume fraction field ϕ_s .

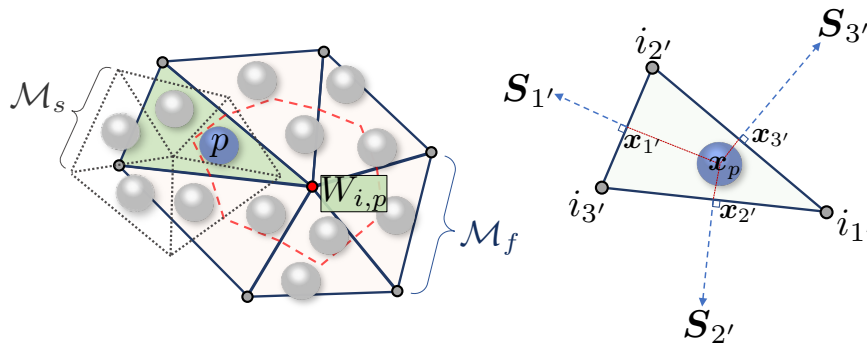


Figure 5 2D representation of six elements around the node i (red). The control volume of the node i is delimited by the red lines. A solid particle p is located in the green coloured element. The interpolation weight of the particle p at the node i is marked as $W_{i,p}$ and can be computed using the surface vectors $S_{1'}$, $S_{2'}$, $S_{3'}$, situated opposite of the element nodes $i_{1'}$, $i_{2'}$, $i_{3'}$, and the distance of the particle position x_p from the centres of the element edges $x_{1'}$, $x_{2'}$, $x_{3'}$, as expressed in Equation 39. The local solid and fluid mesh sizes, M_s and M_f respectively, are also indicated.

resulting in the computation of the local quantity of solid volume, and by extension, the local solid and fluid fractions ϕ_s and ϕ_f , respectively, as illustrated by Figure 4. The solid volume $V_{s,i}$ at grid node i is given by:

$$V_{s,i} = \sum_{p|x_p \in E_i} V_p W_{i,p}. \quad (38)$$

The subscript p denotes the properties of the p^{th} particle, E_i is the set of elements adjacent to the grid node i , and $W_{i,p}$ is the weight of the linear interpolation used. In our work, only triangular or tetrahedral elements are used in 2D and 3D cases, respectively, so the linear interpolation weights can be computed as:

$$W_{i,p} = \frac{|\mathbf{x}_p - \mathbf{x}_{fi}| \cdot \mathbf{S}_{fi}}{\sum_{i' \in \mathcal{N}(E_p)} |\mathbf{x}_p - \mathbf{x}_{fi'}| \cdot \mathbf{S}_{fi'}}, \quad (39)$$

where E_p is the element containing particle p , and $\mathcal{N}(E)$ is the set of nodes i' of the element E and $\mathcal{S}_{f_{i'}}$ the vector area of the face $f_{i'}$ opposite to the node i , as shown in Figure 5. By taking the ratio of the local solid volume over the total cell volume we can compute the local solid volume fraction ϕ_s as shown in Equation 1.

The benefits of the VOS representation of the immersed body is twofold. First, the penalty mask used, χ_s , is easily defined by a sharp Heaviside function of the solid volume fraction. Second, the rigid body movement is imposed on the Lagrangian particles and at each iteration the volume fraction is recomputed, hence the total volume of the solid in the Eulerian mesh is inherently conserved in time. Also, the added operations to represent the movement of the solid consist of Lagrangian displacement of the particles and their relocalisation on the processors in a parallel computing configuration. In terms of computational cost, these operations are less costly when compared to fully Eulerian approaches previously tested in YALES2 such as constructing a Level-Set function from a triangulated surface to represent the immersed surface and then displacing it. The last method would also need specific additional treatment to conserve the volume of the immersed solid.

2.4.4 | Solid velocity field

In contrast to the Lagrangian displacement of the solid particles, the solid velocity field is computed directly onto the Eulerian mesh of the computational domain based on the solid volume fraction field at the current time-step and the prescribed motion of the rigid solid. As explained for the operations for displacing the particles, two or more types of movement may occur at the same time, and the final velocity of the solid object is a combination of the velocities prescribed by each of the movements. The velocity of a solid body k can be computed in space and time as:

$$\mathbf{u}_{s,k} = \omega_k \mathbf{r}_k \times (\mathbf{x} - \mathbf{x}_k^R) + v_k \mathbf{r}_{t,k} + A_{o,k} 2\pi f_{o,k} \cos(2\pi f_{o,k} t) \mathbf{r}_{o,k}, \quad (40)$$

where the subscript ' k ' has been added to all the quantities associated to the displacement operations acting on the specific solid k .

In many applications it is possible to have multiple immersed bodies in the same simulation undergoing different displacement operations. In some cases, such as in gearboxes, the surfaces of two or more solids can come very close to each other. Depending on the size of the computational cells, there may be multiple solids present in a cell with different solid velocities. To solve this issue, we define the mean solid velocity field at each position in space (i.e. in each node) as the sum of the different solid velocities weighted by the local solid volume fraction of the solids:

$$\mathbf{u}_s = \frac{\sum_k \phi_{s,k} \mathbf{u}_{s,k}}{\sum_k \phi_{s,k}}. \quad (41)$$

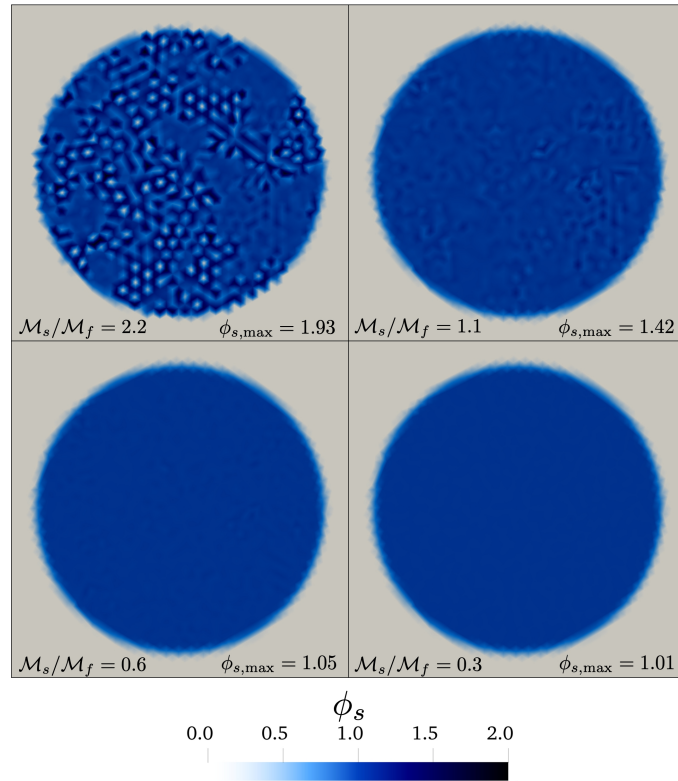


Figure 6 Solid volume fraction field and the maximum value for four different ratios of the solid mesh metric \mathcal{M}_s to the fluid mesh metric \mathcal{M}_f .

This mean solid velocity is used in the definition of the composite velocity shown in Equation 8 as it takes into account the existence of multiple solids in the same computational cell. This formulation also allows to set the solid velocity to zero in the fluid domain where $\phi_s = 0$.

2.5 | Restrictions on the solid cell size

The choice for the characteristic cell size of the solid mesh \mathcal{M}_s needs to be taken into consideration depending on the Eulerian fluid mesh cell size \mathcal{M}_f . From Figure 5 it can be understood that if \mathcal{M}_s is larger than \mathcal{M}_f , the solid particles will have more distance between them. This brings the risk of having some elements of the Eulerian mesh with no particles at all and some elements containing particles with a volume larger than the computational cell volume. In this case, the solid volume fraction field suffers from discontinuities in the form of ‘holes’ inside the solid region, with $\phi_s < 1$, accompanied by spots where the solid volume fraction overshoots, with $\phi_s > 1$. This is demonstrated in Figure 6 where four different ratios of $\mathcal{M}_s/\mathcal{M}_f$ were tested. The fluid metric remained unchanged, $\mathcal{M}_f = 2.5$, while the solid metric varied from $\mathcal{M}_s = 5.5$ to $\mathcal{M}_s = 0.7$. For ratios larger than 1 the discontinuities are clearly visible and the overshoots may even reach values of $\phi_s = 1.93$, which would not be acceptable from a physical point of view. For ratios smaller than 1, the peaks and troughs are greatly diminished and the

$$V_s = (1 - 0.0016) \times V_{cyl}$$

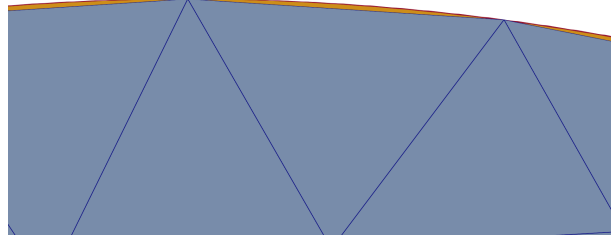


Figure 7 2D schematic of the difference between a continuous surface and its discretized shape. The total volume of the discretized solid, with solid mesh size of $\mathcal{M}_s = D/40$, is equal to V_s . The theoretical volume of a perfect 2D cylinder is equal to $V_{cyl} = \pi D^2/4$. A part of V_{cyl} , 0.16%, is lost due to discretization errors.

maximum value of ϕ_s is closer to the target value of unity inside the solid region. Thus the criterion for the solid mesh cell size that needs to be satisfied is the following:

$$\mathcal{M}_s / \mathcal{M}_f \leq 1. \quad (42)$$

3 | VALIDATION

In this section, several benchmark flow problems are solved using the VOS-IVP method to demonstrate its ability to obtain accurate results for different configurations. We examine the cases of flows around a cylinder of diameter D or an airfoil of chord c . The main quantities to compare between the numerical results with the reference data are the drag and lift coefficients (C_D and C_L) and the Strouhal number (St) defined as:

$$C_D = \frac{2F_x}{\rho S U_\infty^2}, \quad C_L = \frac{2F_y}{\rho S U_\infty^2}, \quad St = \frac{f_s D}{U_\infty}. \quad (43)$$

F_x and F_y are the stream-wise and cross-flow total forces, respectively. U_∞ is the free-stream velocity, ρ is the fluid density, S is the cross-sectional area of the body and f_s the vortex shedding frequency in unsteady flows. Numerically, we will be interested in the mean value ($\langle \bullet \rangle$) and the mean fluctuation (\bullet') of the variables. The shedding frequency f_s is computed through a Fast-Fourier Transform (FFT) analysis as the fundamental frequency of the lift's mean fluctuation.

3.1 | Solid volume conservation

Table 1 Mean and r.m.s. values of the relative error in solid volume of a curved solid geometry for two coarse meshes.

\mathcal{M}_f	$\langle \epsilon_s \rangle$ [%]	ϵ'_s [%]
$D/5$	0.16	$2.11 \cdot 10^{-14}$
$D/10$	0.16	$1.38 \cdot 10^{-14}$

To demonstrate that the Lagrangian particles conserve the solid mass independently of the fluid mesh resolution and the movement of the solid, we solved the flow past a 2D cylinder of diameter D and volume V_{cyl} , oscillating with amplitude $y_{max} = D$ and frequency $f = 2.2$ Hz.

First, the theoretical value of the cylinder volume (equivalent to a circular surface in 2D) is computed as $V_{cyl} = \frac{\pi D^2}{4}$. The volume $V_{s,p}$ of the discretized solid mesh of $\mathcal{M}_s = D/40$ can be computed as the sum of the volume information of each particle $V_{s,p} = \sum_p V_p = 0.9984 V_{cyl}$. It slightly underestimates the theoretical value by 0.16% due to curvature discretization errors as demonstrated in the schematic of Figure 7.

During the simulation, as the solid is represented through a VOS approach, by integrating the solid volume fraction over the domain one should obtain the total volume of the immersed object $V_s = \int_{\Omega} \phi_s(t) dV$. The error of the computed solid volume relative to the theoretical one can be computed as:

$$\epsilon_s(t) = \frac{|V_{cyl} - V_s|}{V_{cyl}}. \quad (44)$$

For the fluid mesh, two coarse grids were tested with $\mathcal{M}_f = D/5$ and $\mathcal{M}_f = D/10$. Table 1 shows the mean and the r.m.s. values of the relative error after four oscillations of the cylinder. The mean value remains constant for both grids giving the same 0.16% error as the one of the discretized solid mesh with respect to the theoretical value. This affirms the conservative nature of the solid volume fraction field, where $\int_{\Omega} \phi_s(t) = V_s$. The r.m.s. value decreases with finer meshes thanks to lower interpolation errors. Despite that, both values are essentially near machine precision at an order of magnitude of 10^{-14} %.

The same oscillatory movement was imposed on a 2D square, i.e. a shape with no curves, of sides equal to the cylinder diameter D on the grid of cell-size $\mathcal{M}_f = D/5$. The 2D volume of the square is equal to $V_{square} = D^2$. When projected, the mean and r.m.s. values of the volume relative error are 2.7×10^{-14} % and 5.9×10^{-14} %, respectively. Both values are essentially zero, further proving the fact that in the case of the cylinder, the errors originated from the discretization of the curves in the solid mesh.

This study shows that the whole discretized solid volume V_s , held by the Lagrangian particles, is successfully projected onto the Eulerian mesh. This remains unchanged irrespective of the movement of the solid and the resolution of the fluid mesh used. The 0.16% difference from the theoretical value observed for the cylinder case does not come from our volume projection algorithm described in Section 2.4.1, but from the initial discretized solid mesh used to create the Lagrangian particles in the pre-processing stage as described in Section 2.4.3.

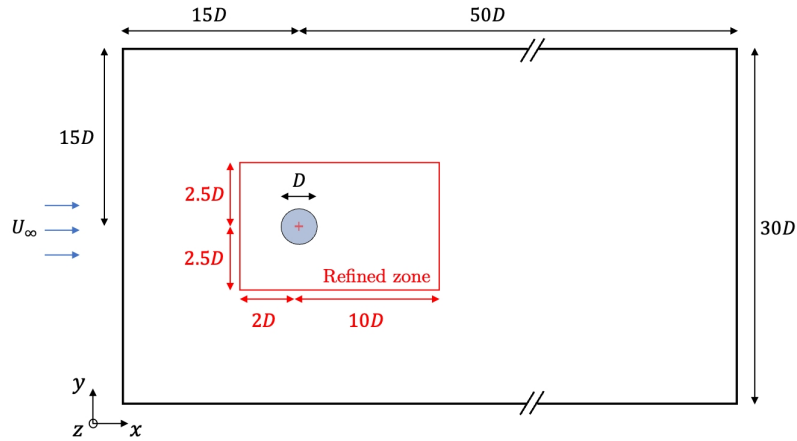


Figure 8 Sketch of the domain for the case of the stationary 2D cylinder.

3.2 | Laminar flow around a stationary cylinder

3.2.1 | Flow at Reynolds number of 100

A well-researched benchmark problem for many years, the laminar unsteady flow past a two-dimensional stationary cylinder of diameter D is studied. The results of this problem are known to be sensitive to the size of the computational domain, particularly for relatively small values of Reynolds number. The computational domain, shown in Figure 8, extends $15D$ upstream of the solid and $50D$ downstream. The top and bottom boundaries of the domain are placed $15D$ from the centre of the solid, sufficiently far to limit blockage effects, with slip-wall boundary conditions. The inlet velocity U_∞ is kept constant and the Reynolds number, computed as $Re = U_\infty D/\nu$, is imposed by changing the value of the kinematic viscosity ν . The cylinder is placed in a refined zone of dimensions $[-2D, 10D] \times [-2.5D, 2.5D]$ where the grid-spacing Δx corresponds to $D/\Delta x = 50$. The downstream length of the refinement zone covers a sufficient portion of the wake to assure that all the attached vortical structures to the cylinder are well-resolved, because according to Kang et al.²⁹, for incompressible flow with a discrete wake, it is proven that the total force on the body is solely determined by the body-connected vortical structures. The mesh is composed of 3.73×10^5 elements. The time-step of the simulations is determined by the CFL condition $CFL = 0.9$. The penalisation time-step ratio is $\alpha = 1$. The simulated physical time covers 1000 non-dimensional periods ($t^* = tU_\infty/D$) and all simulations run on 20 CPU cores. The VOS-IVP method is validated for $Re = 100$ against body-fitted (BF) simulations, also carried out with YALES2, and reference data obtained from numerical simulations from the literature^{30,31,32}.

In Figure 9 the time series of the drag and lift coefficients are shown and the numerical results of the VOS-IVP appear to be smooth. The frequency of the drag coefficient fluctuation is double the one of the lift fluctuation due to the contribution of the alternating upper and lower vortices to the drag force.

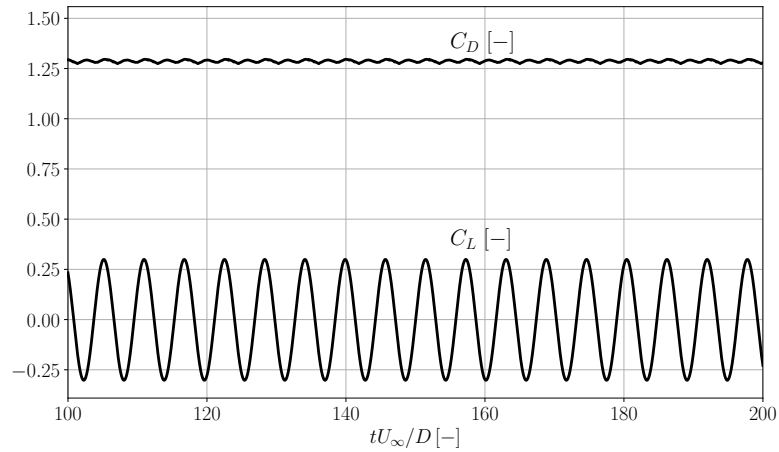


Figure 9 Time-series of the drag and lift coefficient predicted with the VOS-IVP method in the case of of $Re = 100$.

Table 2 Mean drag coefficient $\langle C_D \rangle$, mean lift fluctuation C'_L , Strouhal number St , wake closure length L_c and time-step Δt for the case of $Re = 100$.

Cases	$\langle C_D \rangle [-]$	$C'_L [-]$	$St [-]$	$L_c [-]$	$\Delta t \times 10^{-4} [s]$
Qu et al. ³⁰	1.317	0.222	0.165	1.41	–
Park et al. ³¹	1.330	0.235	0.165	1.42	–
Kravchenko et al. ³²	1.320	0.222	0.164	1.45	–
Body-fitted (BF)	1.335	0.237	0.167	1.38	1.012
VOS-IVP	1.300	0.215	0.171	1.32	1.072

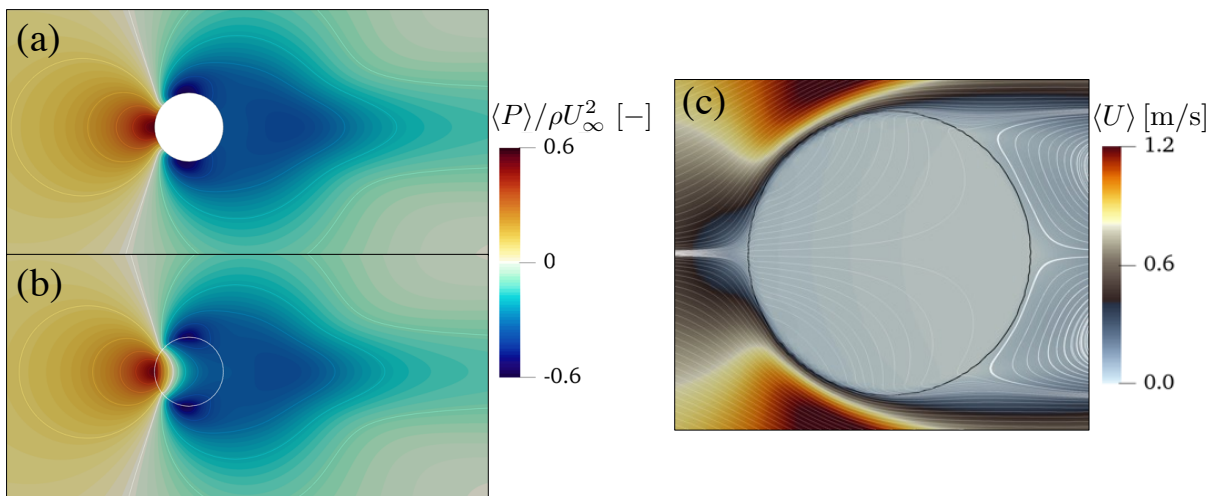


Figure 10 On the left figure, time-averaged dynamic pressure fields of the (a) body-fitted method and (b) the VOS-IVP method for $Re = 100$. On the right figure, (c) time-averaged mean composite velocity field of the VOS-IVP method and streamlines near the cylinder.

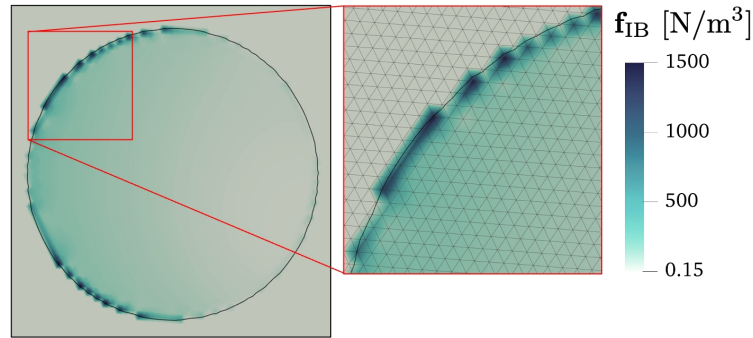


Figure 11 Local volumetric penalty forces applied in the solid region. A closer view near the fluid-solid interface is also provided with the grid visible.

Table 2 shows that the mean drag coefficient, the mean fluctuation of lift coefficient, and the Strouhal number match very well with the body-fitted case and the reference data. The wake closure length, i.e. the distance along the wake centre line from the cylinder to the point of zero velocity, is very close to the body-fitted case, but overall underestimated in our simulations compared to the literature. This can be seen in Figure 10 where the time-averaged fields of the pressure are shown for the body-fitted method and the VOS-IVP method. The fields match very well with slight variations near the cylinder. The two zones of low pressure at the upper and lower parts of the cylinder appear smaller in the VOS-IVP results. This shows that the fluid does not decelerate as much when passing the cylinder as in the body-fitted case. The difference stems from the sharp penalty mask used. As seen in Figure 11, the first points where a volumetric forcing term is applied are well within the solid region. Effectively, the cross-sectional surface of the body seen by the fluid is smaller than the real one. This fact also explains the lower values in the aerodynamic quantities for the present method. [Another factor for the lower drag coefficient prediction is the penetration of the solid region by some fluid. This is evident from the streamlines based on the mean values of the composite velocity field in Figure 10 \(c\), whose value is non-zero inside the solid. Liu et al.¹⁶ and Wu & Shu³³ have managed to resolve the streamline penetration problem by calculating the IBM forcing contribution via iterative processes. In the former work, the force is added as an unknown source term to the Pressure Poisson Equation. In the latter, the force is determined in such a way that the velocity at the immersed boundary point interpolated from the velocity field satisfies the non-slip boundary condition. In the current work, such iterative methods were not considered to avoid extra computational costs. The effect of the presented implicit volume penalty method and the influence of its parameters to the better solid velocity imposition inside the immersed body can be found in the following two sections.](#)

The velocity profiles in Figure 12 show the mean and r.m.s. values of the stream-wise and cross-flow velocity components at three different positions ($x = [1D, 2D, 5D]$) for the body-fitted and the VOS-IVP methods. The profiles match very well between the two methods but we can still notice marginally higher velocity values in both directions for the VOS-IVP method, further supporting the previous observations from the mean pressure fields. Figure 13 shows the mean stream-wise velocity along the

centre line of the wake. There is a noticeable difference in the region between $x = 1D$ and $x = 3D$ where the wake recovery is slightly faster in the VOS-IVP method. The wake closure length L_c was determined as the point of vanishing $\langle \mathbf{u} \rangle$, where there is a change from negative to positive values, along the wake centre line downstream of the solid's surface. The VOS-IVP measures the wake closure at $L_c = 1.32D$ downstream of the object, 4% shorter compared than the body-fitted case.

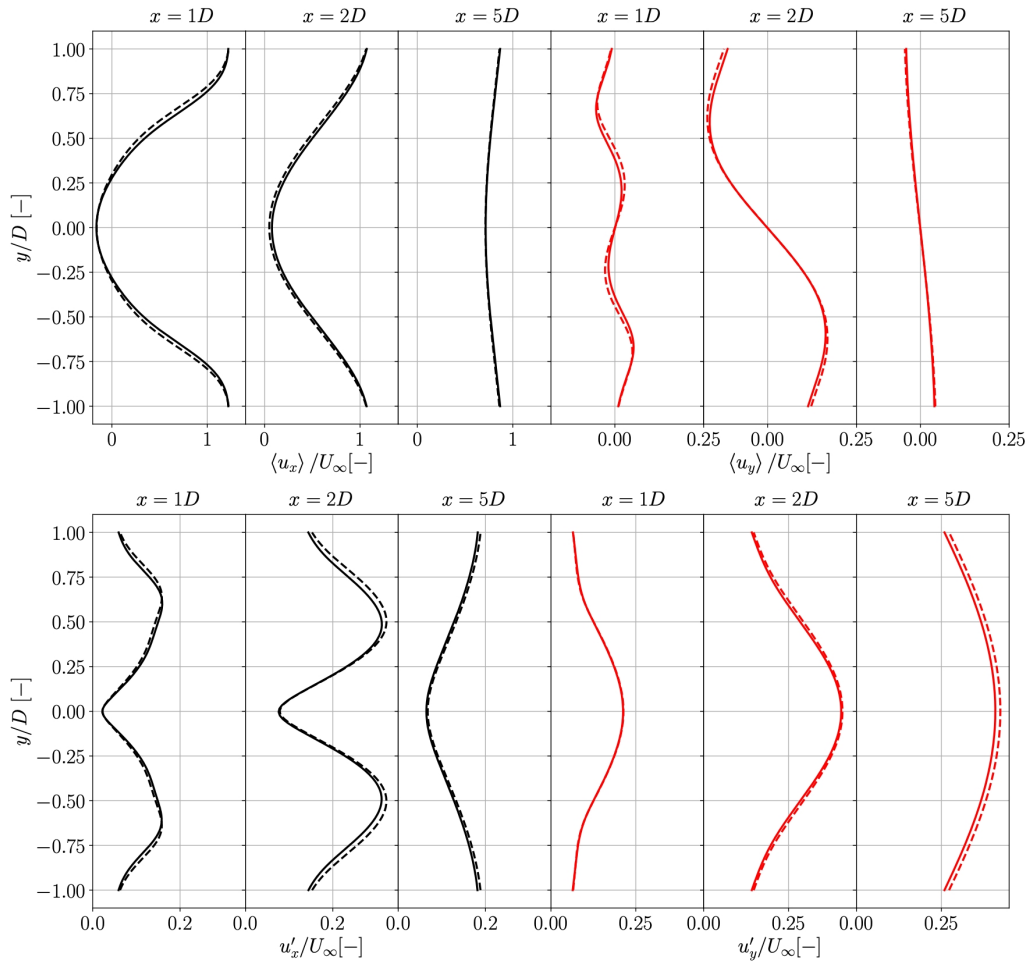


Figure 12 Vertical profiles of the mean (top) and r.m.s. (bottom) velocity components in stream-wise (black) and cross-flow (red) directions at three different positions, $x = [1D, 2D, 5D]$, for $Re = 100$. Dashed lines: body-fitted case; solid lines: VOS-IVP.

3.2.2 | Mesh dependency study and influence of the implicit penalty term

A mesh dependency study is conducted on the case of section 3.2.1 by coarsening the computational grid. Three mesh sizes were tested: $D/\Delta x = [10, 25, 50]$. The purpose of this study is to examine the influence of the mesh size to i) the aerodynamic

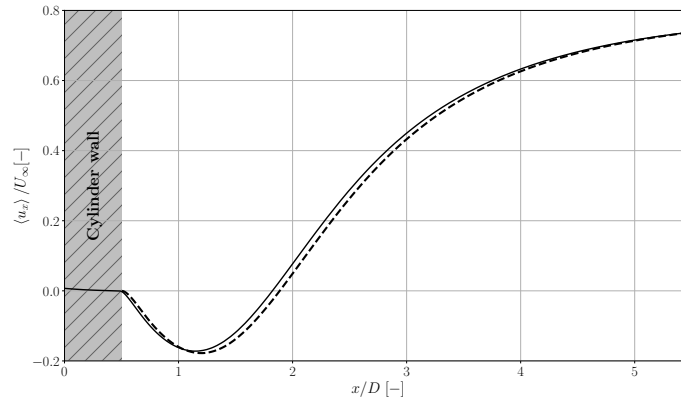


Figure 13 Mean stream-wise velocity normalised by the free flow velocity along the centre line of the wake, for $Re = 100$. Dashed line: body-fitted case; solid line: VOS-IVP.

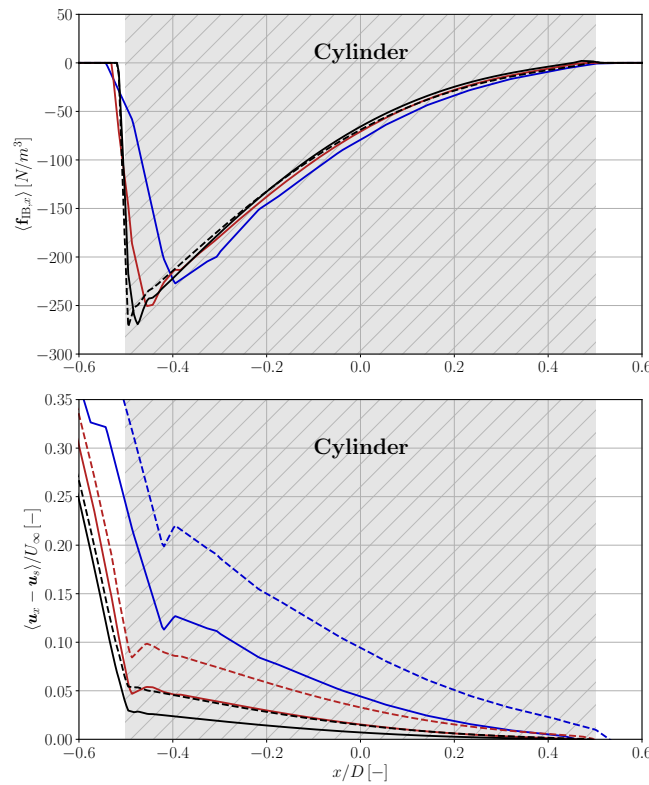


Figure 14 Comparison between the application and effectiveness of the usual volume penalty method and the implicit formulation for different grid sizes at the region of the immersed solid. The top figure shows the profile of the mean stream-wise component of the penalty force along an horizontal line passing by the cylinder centre. The bottom figure shows the mean stream-wise component of the composite velocity along the same line. Bold lines: IVP; dashed lines: VP. Blue colour: $D/\Delta x = 10$; red colour: $D/\Delta x = 25$; black colour: $D/\Delta x = 50$.

Table 3 Mean drag coefficient $\langle C_D \rangle$, mean lift fluctuation C'_L and Strouhal number St for different mesh sizes in the case of $Re = 100$.

Mesh	$\langle C_D \rangle [-]$	$C'_L [-]$	$St [-]$
$D/\Delta x = 10$	1.314	0.217	0.178
$D/\Delta x = 25$	1.297	0.221	0.177
$D/\Delta x = 50$	1.300	0.215	0.171

coefficients, ii) the penalty force distribution inside the solid and iii) the velocity profile inside the solid when using the VOS-IVP method.

Table 3 shows the mean drag coefficient, the r.m.s. of the lift coefficient and the Strouhal number for different mesh sizes. The aerodynamic coefficients do not show any particular trends when changing the mesh size, their values remain unchanged. This can be attributed to the relatively simple form of the bluff body. In the case of the Strouhal number, its value decreases with a smaller grid spacing approaching the values found in literature (Table 2).

The mesh size influences strongly the accurate imposition of the prescribed solid velocity $\mathbf{u}_s = 0$ inside the solid volume due to the change in distribution of the penalty force which serves to bring the fluid at rest. The top of Figure 14 shows the profile of the mean stream-wise component of the penalty force $\langle \mathbf{f}_{IB,x} \rangle$ along a horizontal line passing by the cylinder centre ($y = 0$). As a reminder, the free stream flows from left to right. For the finest grid, the peak force value is inside the solid volume near the left solid-fluid interface at $x/D = -0.5$. As the grid coarsens, the peak force value weakens in magnitude and is applied further inside the solid region. The profile also loses its initial sharpness but further inside the solid, $x/D > -0.2$, all the profiles converge towards the same values. The direct impact of the change in force distribution with the different mesh sizes can be seen at the bottom of Figure 14, which shows the profile of the mean stream-wise component of the composite velocity. It is evident that with coarser grids, the solid velocity is not well imposed. The penalty force is not sufficient to decelerate the fluid fast enough and fairly high positive values of $\langle \mathbf{u}_x \rangle$ persist inside the solid. With refining the mesh, the velocity values drop significantly approaching the target value.

Let us now examine the influence of the implicit penalty term of the IVP compared to the forcing of the usual volume penalty method (VP). The same simulations were carried out but with the correction contribution of the penalty force deactivated. Their respective velocity profiles can also be seen in Figure 14. It is evident that the simple penalty method is not as effective as the implicit penalty method developed in this work and for the same mesh size the positive velocity values are higher. For $D/\Delta x = 10$, the velocity fails to reach the 0 value. It is interesting to look at the penalty force profiles for $D/\Delta x = 50$. The VP method gives a sharper profile and the force is applied closer to the solid surface, but this influences only the intermediate velocity field \mathbf{u}^* . In the correction step the velocity will be modified due to the new pressure gradient in order to satisfy only the incompressibility constraint. This results in smoothing the velocity gradient at the solid surface and thus the higher velocity

values. On the contrary, in the IVP method the penalty force comprises a contribution in the prediction step and in the correction step. The correction contribution counteracts with the pressure gradient and results in a smoother penalty force profile. However, the boundary condition of the velocity is imposed in the immersed volume at the same time as the incompressibility constraint, resulting in velocity values closer to the imposed solid velocity.

Figure 15 shows, for both VP and IVP methods, the evolution of the mean stream-wise velocity value at the first node inside the solid domain for the different mesh sizes. There appears to be a linear relationship between them for both methods: $\langle \mathbf{u}_x \rangle \propto \Delta x$. Here again, the error committed by the IVP approach is lower than the VP approach.

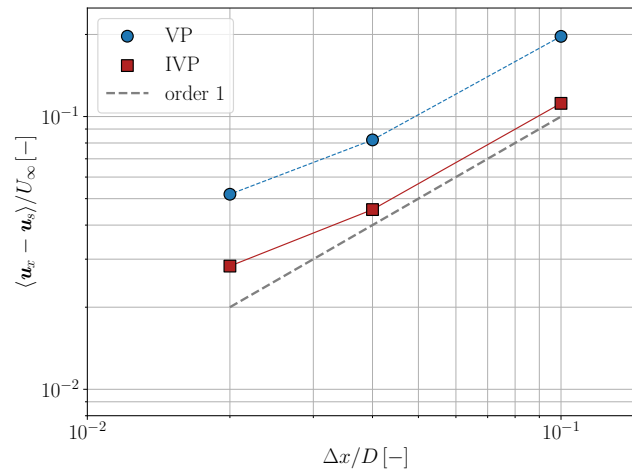


Figure 15 Evolution of the mean stream-wise velocity value at the first node inside the solid domain for the different mesh sizes.

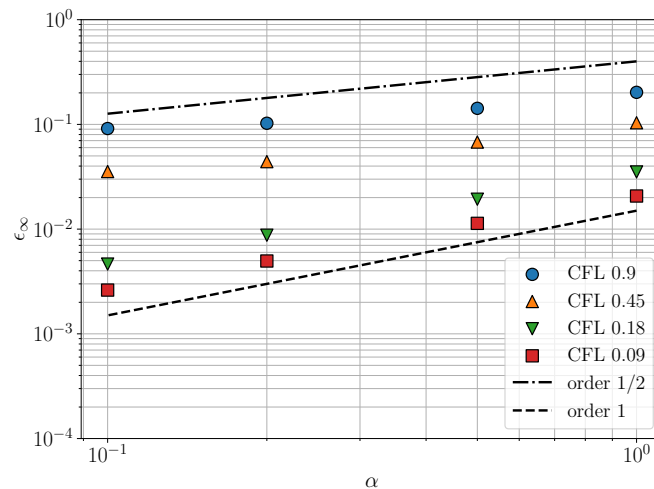


Figure 16 Evolution of maximum velocity error inside the solid with the penalisation time-step ratio for different CFL values.

3.2.3 | Influence of time-step and penalty parameter

A parametric study has been conducted to evaluate the influence of the time-step Δt and the penalty parameter η on the force estimation and the imposition of the solid velocity inside the immersed body. Using the same computational domain as before, with $D/\Delta x = 50$ at the immersed solid, and keeping the same Reynolds number of $Re = 100$, 200 non-dimensional periods were simulated. The time-step was determined by the CFL condition and the penalty parameter by the penalisation time-step ratio $\alpha = \eta/\Delta t$. Four values of each parameter were tested: $CFL = [0.09, 0.18, 0.45, 0.9]$ and $\alpha = [0.1, 0.2, 0.5, 1]$. The main tools of comparison are the relative error of the mean drag coefficient compared to the previously mentioned body-fitted case and the L_∞ norm of the velocity magnitude in the region where the penalty term is applied. These quantities are computed as:

$$\varepsilon_{\langle C_D \rangle} = \frac{|\langle C_D \rangle_{\text{VOS-IVP}} - \langle C_D \rangle_{\text{BF}}|}{\langle C_D \rangle_{\text{BF}}}, \quad (45)$$

$$\varepsilon_\infty = \max(\chi_s \phi_s \| \mathbf{u} \|_2). \quad (46)$$

Figure 16 shows the maximum velocity error in the solid region depending on the CFL and α . The error diminishes with smaller time-step values and with smaller penalty parameter values. Furthermore, with smaller time-steps, the order of convergence of ε_∞ with respect to η increases from $O(\eta^{1/2})$ to almost $O(\eta^1)$. This result agrees with the findings of Angot et al.⁸ who have rigorously shown using asymptotic analysis that the solution of the penalized velocity converges to the exact Navier-Stokes equations at the immersed boundaries as $\eta \rightarrow 0$ with a global error of $O(\eta^{3/4})$.

When examining the value of $\varepsilon_{\langle C_D \rangle}$ for the different combinations of CFL and α values, there is small variation in the error with the minimum and maximum values of 2.6% and 3.6%, respectively, and no clear tendency can be observed. With higher time-steps, a value of $\alpha = 1$ gives a closer estimation to the body-fitted result. But for the smaller values of Δt , smaller values of α are preferred. From a computational point of view, decreasing the value of α with a constant CFL doesn't have an important influence on the computational cost in the case of stationary solids. However, reducing the CFL limit and forcing smaller time-steps the computational cost increases considerably. The average cost for $CFL = 0.9$ was 20 hCPU while for $CFL = 0.09$ the average cost was 80 hCPU. This overcost is due to more calls to the pressure Poisson equation solving even if each solve is less costly at smaller CFL number. Hence, there's a compromise to be made between the accurate velocity imposition and the computational cost.

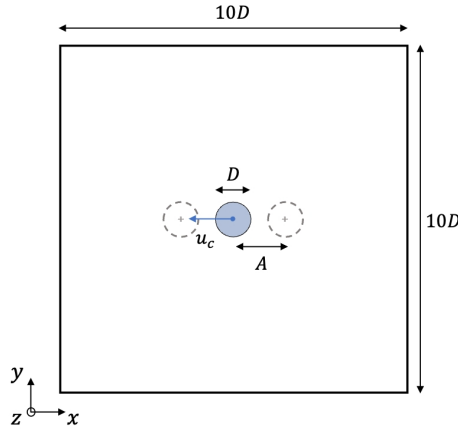


Figure 17 Sketch of the domain for the case of the in-line oscillating cylinder in a quiescent fluid.

3.3 | Oscillating cylinder in a quiescent fluid

To validate the present method for a moving solid body problem, the case of an oscillating cylinder in a fluid at rest was examined. This case involves the simple harmonic motion in the x -direction of a 2D cylinder with diameter $D = 1\text{m}$ placed at the centre of a square domain as shown in Figure 17, where periodic boundary conditions were applied. The cylinder oscillation leads to the development of boundary layers on the upper and lower sides, which separate from the body generating two counter-rotating vortices. When the cylinder starts to move in the opposite direction the vortex generation stops and the body splits the previously created vortex pair. The periodic motion of the cylinder is described by the following equations:

$$x_c(t) = -A \sin(2\pi f t), \quad (47)$$

$$u_c(t) = -2\pi f A \cos(2\pi f t), \quad (48)$$

$$v_c(t) = 0. \quad (49)$$

where x_c , u_c and v_c are the position, the horizontal velocity and the vertical velocity of the cylinder centre, respectively. The frequency of the oscillation is expressed as f and the amplitude of oscillation A . The maximum velocity of the oscillation is

Table 4 Mean fluctuation of the in-line force over 20 periods, relative error of the in-line force compared to the body-fitted case and fundamental frequency of the predicted force for all the simulations.

Case	F'_x [N]	$\epsilon(F'_x)$ [%]	f_{fund} [Hz]
VOS-IVP, $D/\Delta x = 10$	1.24	8.77	0.200 ± 0.005
VOS-IVP, $D/\Delta x = 25$	1.07	6.14	0.200 ± 0.005
VOS-IVP, $D/\Delta x = 50$	1.08	5.26	0.200 ± 0.005
VOS-IVP, $D/\Delta x = 100$	1.11	2.63	0.200 ± 0.005
BF, $D/\Delta x = 100$	1.14	—	0.200 ± 0.005

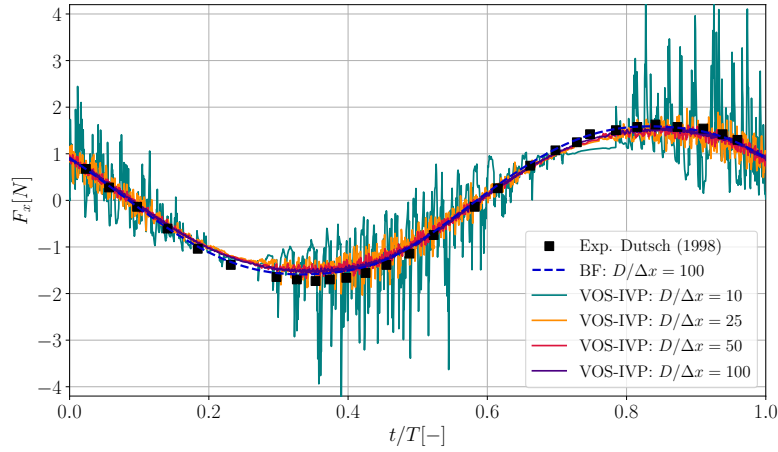


Figure 18 In-line force acting on the body during one oscillation period.

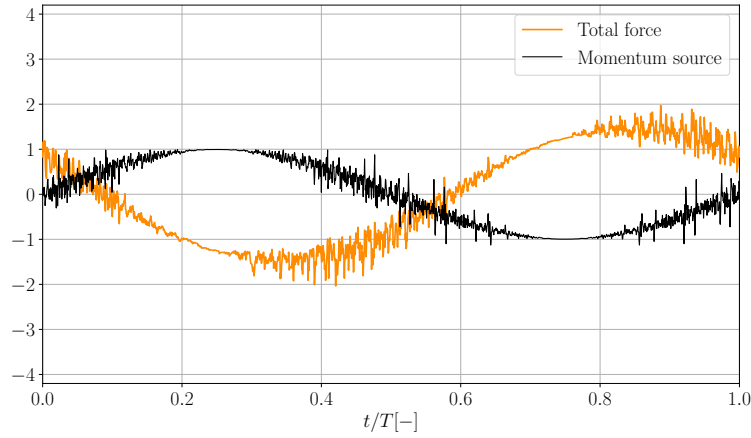


Figure 19 In-line force acting on the body (orange) and solid momentum source (black) over one period for $D/\Delta x = 25$.

defined as $U_{\max} = 2\pi f A$. The Reynolds number characterising this flow problem is calculated from this velocity value and the cylinder diameter. Another useful non-dimensional quantity is the Keulegan-Carpenter (KC) number, describing the relative importance of the drag forces over the inertia forces for a bluff body in an oscillatory flow, defined as:

$$KC = \frac{U_{\max}}{fD}. \quad (50)$$

To compare our results with the experimental data we match the two key numbers to the values of $Re = 100$ and $KC = 5$. The maximum velocity is set to $U_{\max} = 1$ m/s with the frequency of oscillation set to $f = 0.2$ Hz and the amplitude to $A = 5/2\pi$ m. The simulations ran for $T = 200$ s (corresponding to 40 oscillation periods) with a time-step of $\Delta t = 0.0025$ s. The present

method was tested on four unstructured meshes where $D/\Delta x = [10, 25, 50, 100]$, Δx being the grid-spacing near the solid body, and the body-fitted case was conducted on a mesh of $D/\Delta x = 100$. The penalisation time-step ratio was set to $\alpha = 1$.

Our computational results are compared to the experimental data of Dutsch et al.³⁴ and resolved body-fitted simulations with imposed moving reference frame. As shown in Table 4, the mean fluctuation of the in-line force is very well predicted with the present method even for the coarser grid resolution. Its relative error compared to the body-fitted case remains under 9%. Similarly the fundamental frequency of the force's fluctuation is well predicted for all the grid resolutions and its value is $f_{\text{fund}} = 0.2$ Hz, matching the frequency of the body's periodic motion. Figure 18 shows the evolution of the in-line force over one oscillation period compared to the results of Dutsch et al.³⁴ and the body-fitted simulation. The predicted force is in excellent agreement with the reference data for a grid resolution of $D/\Delta x = 25$ or higher. It is important to note, however, that due to the sharp penalisation mask used in the present method, high frequency (HF) noise is observed on the predicted forces for the case of a moving immersed body, a known problem with IB methods as reported in literature^{35,2}. These HF oscillations originate from the solid momentum source term \mathcal{P} , as seen in Figure 19. The local value of the source term at the solid-fluid interface spikes when that interface traverses a grid node and changes from a solid node to a fluid one and vice versa. In fact, this change leads to sharp variations of the time-derivative of $\phi_s \mathbf{u}_s$ in Equation 15, which is the main contributor of the HF oscillations. That is why at the phase angles of 90° and 270° where the solid slows down approaching the maximum displacement, the HF noise disappears. Figure 20 shows the frequency spectra of the in-line force for all simulations carried out. One can see that with the reduction of grid-spacing, the HF noise rapidly decreases in strength and starts at a higher frequency.

In Figure 21 the velocity profiles along the y-direction at three different locations, those being $x = [-0.6D, 0.0D, 0.6D]$, for the phase-angle of 180° are shown compared to the reference data. At that phase, the cylinder passes from its initial position with a positive velocity in the x-direction. The present method reproduces well the velocity field around the solid since both the x-component and y-component of the velocity are in very good agreement with the experimental data. The velocity profile at $x = 0$ shows the smooth transition from the velocity of the fluid to the correct solid velocity inside the body ($-0.5 < y/D < 0.5$) imposed thanks to solid momentum source term \mathcal{P} . We can also observe that with a grid-spacing of $D/\Delta x = 25$ or higher, the profiles are identical to those of the body-fitted simulation.

4 | APPLICATION TO A VERTICAL AXIS TURBINE (VAT) UNDER LAMINAR FLOW

A more complex case to test the capabilities of the VOS-IVP method is the study of the unsteady incompressible fluid flow through a vertical axis turbine. More specifically, a two-dimensional study is conducted on a three-bladed Darrieus type turbine subjected to laminar flow. The results of the VOS-IVP method, as well as the computational performances, are compared to

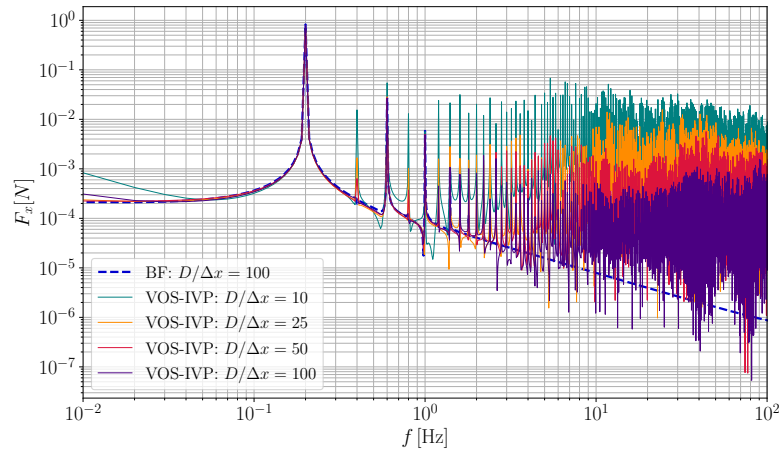


Figure 20 Frequency spectra of the in-line force.

body-fitted simulations using the ALE method on YALES2 and other numerical studies found in the literature using either rotating sliding body-fitted meshes^{36,37} or direct IBM forcing³⁸.

4.1 | Case description

The case setup in this work is the same as the one defined in Ferrer et al.³⁶. The turbine consists of three NACA 0015 airfoils of chord length $c = 1$ m as blades. They are placed at a radius of $R = 2c$ from the rotation centre and the radius connects to the airfoil at the quarter-length ($c/4$) of the chord. The blades are equally spaced from each other in the radial direction (at 120° angles).

The operating conditions are presented in Table 5. The free-stream velocity is set to $U_\infty = 0.5$ m/s and the prescribed rotational speed of the blades is $\omega = 0.5$ rad/s. Hence, the tip-speed ratio, computed as $\lambda = \omega R/U_\infty$, is 2. The chord-based Reynolds number is $Re_c = 100$. The computational domain is shown in Figure 22. It extends $5R$ upstream from the rotor centre, $12.5R$ downstream and $5R$ in each cross-flow direction. The domain is divided into 3 regions of different element sizes. The element size at the blades and the interior of the rotor is $\Delta x = 0.01c$, while in the near wake region $\Delta x = 0.02c$ and for the rest $\Delta x = c$. The mesh contains 0.44×10^6 elements. The simulations are driven by a CFL condition, $CFL = 0.9$. The penalty time-step ratio is $\alpha = 0.1$ for a better solid velocity imposition. Each blade is represented by 15×10^3 Lagrangian particles, as shown in Figure 23. The particles are obtained from a 2D solid mesh of the NACA 0015 airfoil with a cell-size of $\mathcal{M}_s = 0.004c$. The physical time simulated covers 10 rotor revolutions τ ($T_{\max} = 10\tau$). The body-fitted simulation follows the same setup.

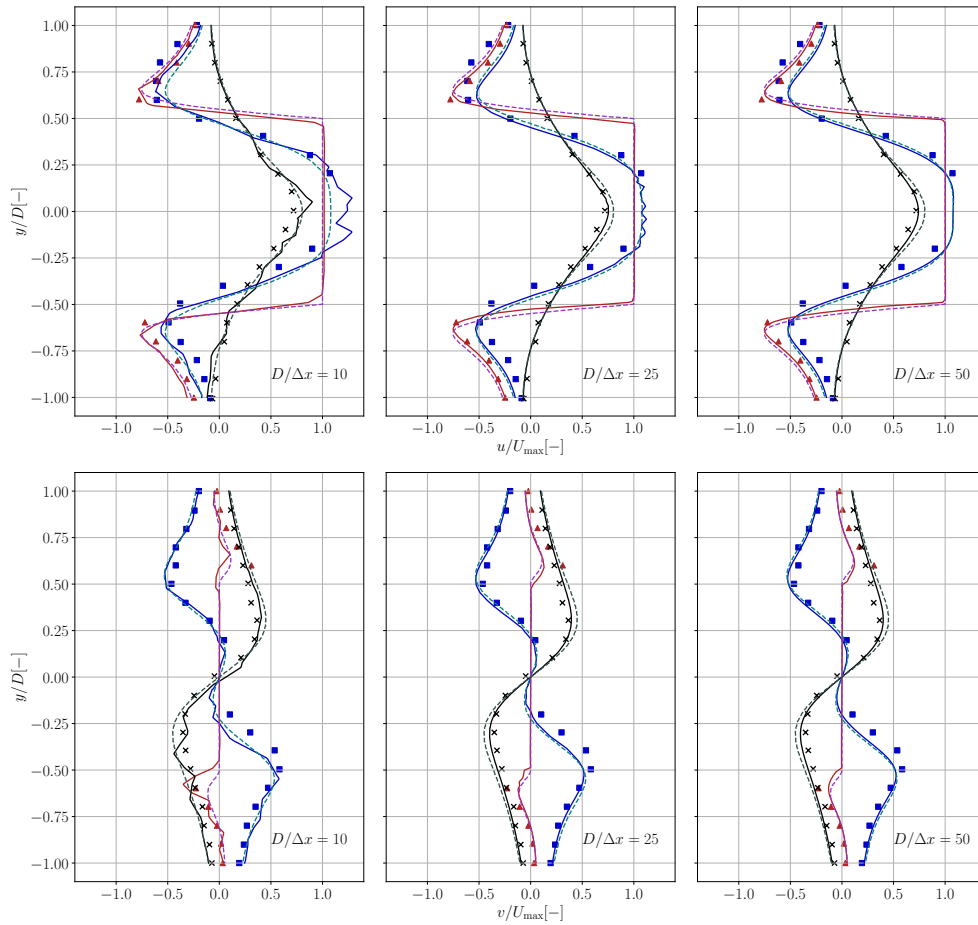


Figure 21 Velocity profiles at different locations when the phase angle is 180° . Solid lines represent VOS-IVP results, dashed lines BF results and symbols experimental data³⁴. Blue shades correspond to $x = -0.6D$, red shades to $x = 0.0D$ and black shades to $x = 0.6D$.

Table 5 Operating conditions of VAT simulation.

Blade profile	NACA 0015
N_{blades}	3
Blade chord c [m]	1
Rotor radius R	$2c$
Free-stram velocity U_∞ [m.s^{-1}]	0.5
Chord-based Reynolds Re_c	100
Rotation speed ω [rad.s^{-1}]	0.5
TSR $\lambda = \omega R / U_\infty$	2

A 2D graphic presenting the rotor movement can be seen in Figure 24. The rotor moves in the counter clock-wise direction around the z -axis at the rotational speed ω . The angle swept by the blades in time t is $\theta = \omega t$. The rotation matrix defining the rotor motion is computed as:

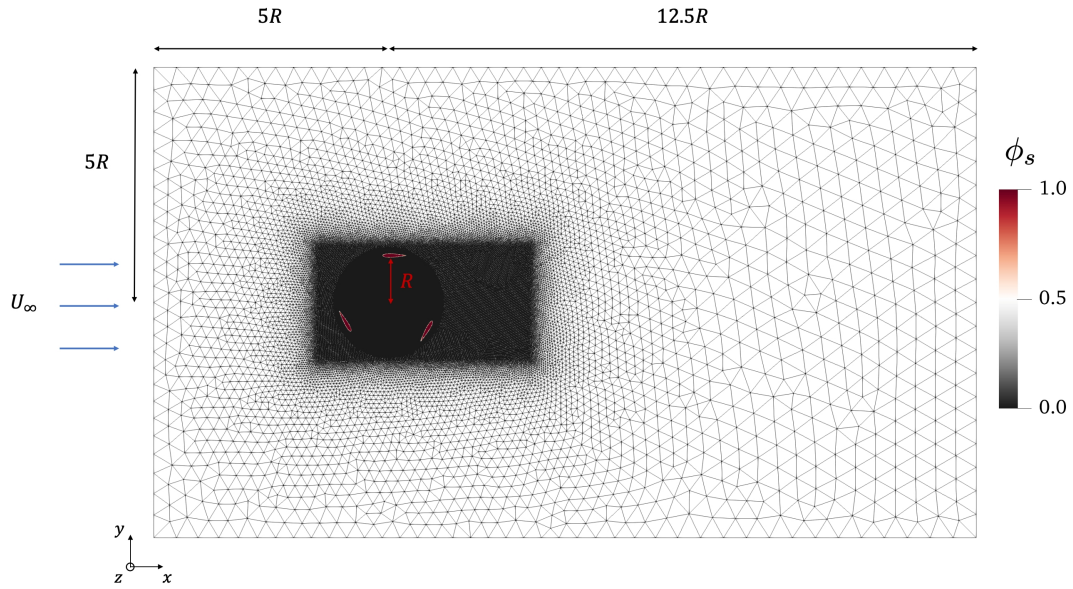


Figure 22 Dimensions of the computational domain relative to the rotor radius. The mesh is coloured by the value of the solid volume fraction at the nodes making the blades visible.

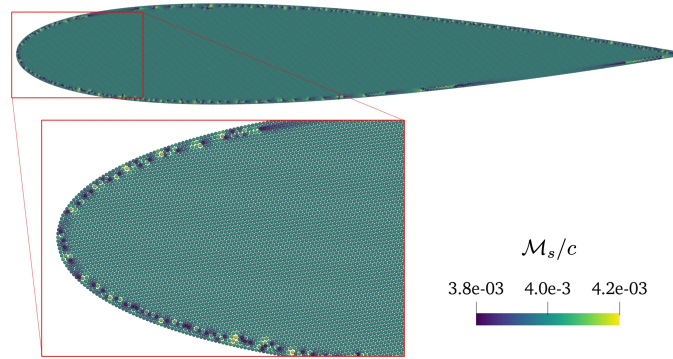


Figure 23 Solid particle set of a NACA 0015 airfoil profile. The particles are coloured by their solid mesh size compared to the airfoil chord.

$$\mathbf{R} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (51)$$

Thus, the position of a blade b at a given instant is computed as:

$$\mathbf{X}_b = \mathbf{R}\mathbf{X}_b^0, \quad (52)$$

where \mathbf{X}_b and \mathbf{X}_b^0 are the current and initial positions, respectively, of the blade.

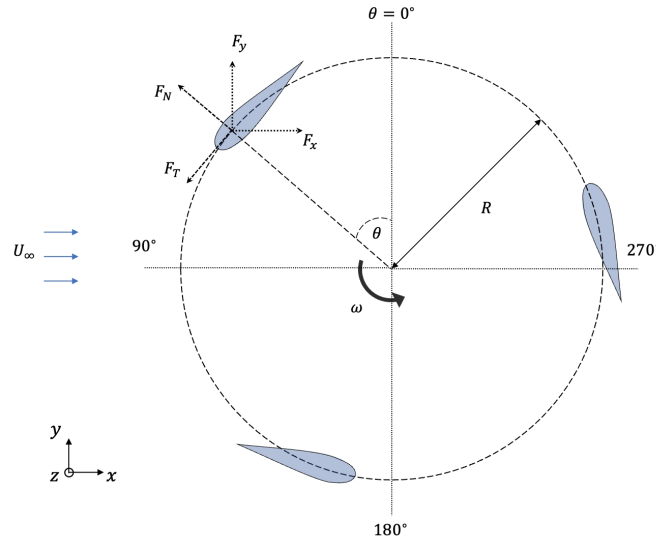


Figure 24 A 2D schematic of the turbine geometry and the main parameters for the rotor movement and the computation of the forces.

As discussed in section 2.3.3, the resistive force on a blade is computed as the negative sum of the penalty forces in the volume V_b occupied by the blade:

$$\mathbf{F} = - \int_{V_b} \rho \mathbf{f}_{IB} dV, \quad (53)$$

where the components of the force follow the x and y directions, i.e. $\mathbf{F} = (F_x, F_y)$, and ρ is the fluid density. Due to the assumption of a two-dimensional flow, the turbine is considered infinitely long and the force in the z direction is ignored.

However, the quantities of interest are the torque, F_T , and the normal forces, F_N , acting on the blades, i.e. the resistive forces in the parallel and perpendicular directions with respect to the blade movement. These can be computed directly from the force estimation in Equation 53 and the position angle θ of the blades. This leads to the expression:

$$\begin{pmatrix} F_T \\ F_N \end{pmatrix} = \begin{pmatrix} -\cos(\theta) & -\sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} F_x \\ F_y \end{pmatrix}. \quad (54)$$

The torque and normal coefficients can then be calculated as:

$$C_T = \frac{2F_T}{\rho c U_\infty^2}, \quad C_N = \frac{2F_N}{\rho c U_\infty^2}. \quad (55)$$

From the torque, one can also compute the power coefficient of the VAT, C_p , as the ratio between the power generated by a turbine blade and the available power in the fluid:

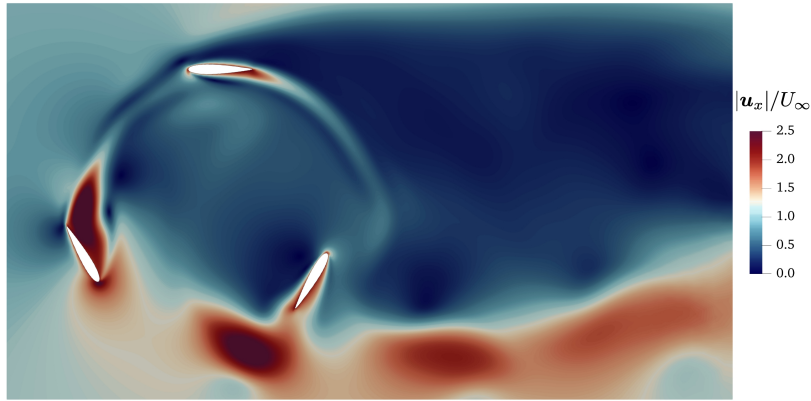


Figure 25 Streamwise velocity magnitude at $\theta = 720^\circ$.

$$C_P = \frac{2\omega R F_T}{\rho A U_\infty^3} \quad (56)$$

where $A = 2RH$ is the turbine's projected area, with $H = 1$ due to the 2D assumption.

4.2 | Results

For this section, the results presented concern the last two revolutions of the simulation, ($8\tau - 10\tau$). Figure 25 shows the streamwise velocity magnitude field when the turbine is at $\theta = 720^\circ$, i.e. the end of the tenth revolution. The velocity contours bear very close resemblance to the ones found by Ramirez et al.³⁷ and Ouro and Stoesser³⁸. One can notice classical features of vertical axis turbines such as the high velocity wakes behind the lower part of the turbine and the velocity deficit at the largest part of the wake. Furthermore, one can observe the vastly different near-field aerodynamics around the blades at different angles θ . Upstroke ($270^\circ < \theta < 90^\circ$), the airfoils encounter much lower velocity values compared to the downstroke region ($90^\circ < \theta < 270^\circ$). Finally, the velocity field is smooth, even near the moving immersed boundaries.

Figure 26 shows the time-history of the coefficients of the torque, the normal force and the power from the blade whose initial position was at $\theta = 0^\circ$, during the last two revolutions. The VOS-IVP coefficient predictions are compared against the ALE results and the results of Ramirez et al.³⁷. Concerning the torque coefficient C_T , the VOS-IVP and ALE predictions are in very good agreement. Although, both methods give higher torque values than the ones found in Ramirez et al.³⁷, especially during the upstroke movement. The normal force coefficient C_N , appears to follow better the values found in Ramirez et al.³⁷. It also follows the same trend as the ALE approach but presents a continuous underestimation. Finally, the power coefficient C_P is also shown between the two YALES2 methods. The power coefficient value is systematically higher in the VOS-IVP method than the ALE approach.

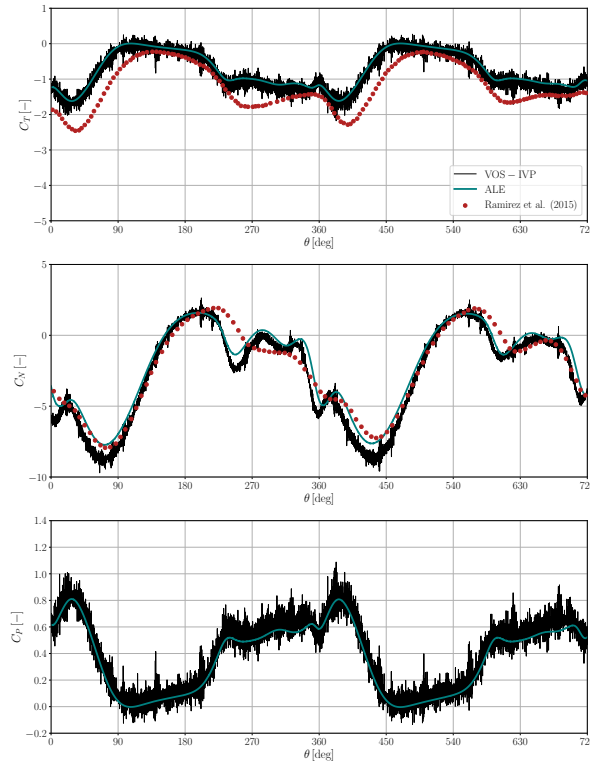


Figure 26 Aerodynamic coefficients of the turbine under laminar flow between the present and blade resolved reference data³⁷: Torque coefficient C_T (top), normal coefficient C_N (middle) and power coefficient C_P (bottom).

Usually, the important quantity in VAT studies is the torque, thanks to which the turbine power can be found. The time-average values of the torque and power coefficients are computed as:

$$\langle C_i \rangle = \frac{1}{2\tau} \int_{8\tau}^{10\tau} C_i dt, \quad \text{where } i = (T, P). \quad (57)$$

The average torque coefficient $\langle C_T \rangle$ predicted by the VOS-IVP method is -0.814 , while the ALE approach gives -0.776 , showing a 5% difference with respect to the body-fitted case. The average power coefficients $\langle C_P \rangle$ take the values 0.407 and 0.368 for the VOS-IVP and ALE cases, respectively. This results in a 10% difference between the two methods.

Overall, the predicted aerodynamic quantities show very good agreement with the body-fitted method, despite the fact that the VOS-IVP time-series suffer from high frequency noise, as this artefact was established in section 3.3.

4.3 | Computational performance

The numerical details concerning the computational performance of both YALES2 simulations are presented in Table 6. In both cases, the simulations are driven by $CFL = 0.9$ and the cell-size at the airfoil surfaces is $\Delta x = 0.01c$, but the time-step is higher in the VOS-IVP method. This results in fewer iterations to cover 10 rotor revolutions compared to the body-fitted case. A possible

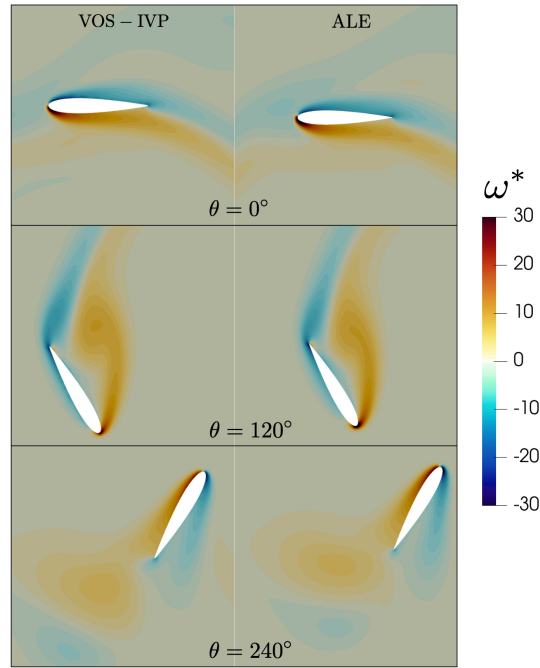


Figure 27 Non-dimensional vorticity fields around the blades at positions $\theta = [0^\circ, 120^\circ, 240^\circ]$ for the VOS-IVP case on the left and the ALE case on the right.

explanation of this difference is the fact that in the body-fitted method we observe slightly higher velocities at the leading edge of the airfoils, with stronger local vortices as shown in Figure 27, leading to bigger restrictions in the global time-step.

To quantify the computational cost of the simulations we need to multiply the total time of the simulation, WCT, by the number of CPU cores used, $N_{\text{cores}} = 32$. The computational cost in CPU hours per one rotor revolution in the VOS-IVP case is 5.82 hCPU/revolution and in the ALE case it reaches 8.50 hCPU/revolution. We achieve a speed-up factor of 1.46 with our method compared to the body-fitted simulation. This means that the VOS-IVP method costed 30% less than the ALE method.

Table 7 shows in more detail the reduced computational times (RCT) of the different processes used in both approaches. The three most costly processes in VOS-IVP are the pressure correction stage, 51% of the cost, the update of the data/variables on the grid nodes, 23%, and the relocation of the solid particles according to the prescribed rotor motion, 22%. For the ALE case the three most costly processes are the velocity advection, 40%, the mesh adaptation, 37%, and the pressure correction, 11%.

The pressure correction step included the process of solving the elliptic pressure Poisson differential equation (PPE). In both cases, the Deflated PCG algorithm was used with a convergence criterion of 10^{-7} , i.e. a diminution of the infinity-norm residual of seven orders of magnitude. However, in the VOS-IVP case, due to the implicit penalty term, we have a variable coefficient in

Table 6 Computational performances of VAT simulations.

	VOS-IVP	ALE
Δt [ms]	4.92	3.48
$N_{\text{iterations}}$	27.1×10^3	32.8×10^3
WCT [s]	6.55×10^3	9.56×10^3
N_{cores}	32	32
hCPU	58.2	85.0
hCPU/revolution	5.82	8.50
speed – up	1.46	1.00

Table 7 Reduced computational times of VAT simulations.

RCT [$\mu\text{s}/\text{iter}/\text{nodes}$]	VOS-IVP	ALE
Total	35.1	37.5
Update of grid variables	8	3.1
Advection	1.3	15.1
Pressure correction	18	4.3
Post-processing	3.3	0.7
VOS-IVP pre-processing	1.7	–
Relocate solid particles	7.8	–
Mesh adaptation	–	13.9

front of the density, known as the penalty density factor, as explained in section 2.3.2. This increases the computational effort needed to solve the PPE, explaining the $18 \mu\text{s}/\text{iter}/\text{nodes}$ RCT compared to the $4.3 \mu\text{s}/\text{iter}/\text{nodes}$ RCT in the ALE case. The data update on the grid costs more in the VOS-IVP method due to the increased number of variables that need to be stored for the penalty parameters and the computation of the mass and momentum source terms, as shown in section 2.2.3. Comparing the advection step, the ALE seems to cost more due to the treatment of the moving mesh nodes. The last differences between the methods are the cost of the particle relocation present in the VOS-IVP case, $7.8 \mu\text{s}/\text{iter}/\text{nodes}$, and the mesh adaptation cost in the ALE case, $13.9 \mu\text{s}/\text{iter}/\text{nodes}$.

To conclude, in this two-dimensional flow around a moving complex geometry, the costs associated with the VOS-IVP processes (PPE and solid particle relocation) seem to be smaller than the costs of the ALE processes (mesh adaptation and mesh movement). The results show a great speed-up when using our method. This is particularly promising for future three-dimensional studies where the mesh adaptation cost in the ALE method increases exponentially.

5 | CONCLUSION AND PERSPECTIVES

In this work, a new immersed boundary method for solving moving body flows has been proposed. The Volume-of-Solid Implicit Volume Penalty method shows very promising results. The conservative properties of the equations prove to be independent

of the fluid mesh resolution thanks to the use of a Lagrangian solid volume fraction field. The benefits of using an implicit formulation for the penalty term are also shown by the faster imposition of the target solid velocity inside the immersed volume. The method was validated on flow problems involving both stationary and moving solids. The predicted forces acting on the solids are in excellent agreement with the reference values, with the relative errors always remaining under 10%. The fluid flow behaviour near the immersed solid is reproduced very well even for very coarse grids. The computational performance of the method was particularly examined on a vertical axis turbine flow problem and compared against a body-fitted simulation. For the same physical time simulated, our method achieves a speed-up factor of 1.46, reducing the total computational cost by 30%.

The aforementioned results of using the VOS-IVP method make it an attractive option for solving incompressible flow problems with moving solid geometries. There are opportunities for further improvement by eliminating the high-frequency oscillations in the force signals. One can also extend the method for high Reynolds number turbulent flows by incorporating wall-models in the penalty term.

ACKNOWLEDGMENTS

This work was granted access to the HPC resources of IDRIS and TGCC under the allocation 2021-A0112B06880 made by GENCI and to CRIANN resources under the allocation 2012006. The authors would like to thank Prof. Grégoire Winckelmans for the fruitful discussions on immersed boundary methods and his suggestion of the implicit form of the penalization term.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

How to cite this article: Tsetoglou I, Cailler M, Bénard P, Lartigue G, Moureau V, and Réveillon J (2024), A volume-of-solid implicit volume penalty method for moving-body flows, *Int J Numer Meth Fluids*, 2024;00:1–6.

APPENDIX

A DERIVATION OF THE VOS-IVP GOVERNING EQUATIONS

In section 2.2.1 the initial mass and momentum conservation equations were shown for the fluid phase and in section 2.2.2 the final equations of the VOS-IVP method, describing both fluid and solid phases through the use of the composite velocity, were shown. In this section the process of passing from the former set of equations to the latter shall be shown step by step.

The two relations used to derive the final set of equations are:

$$\phi_s + \phi_f = 1, \quad (\text{A1})$$

$$\mathbf{u} = \phi_s \mathbf{u}_s + \phi_f \mathbf{u}_f, \quad (\text{A2})$$

where, ϕ_s and ϕ_f are the solid and fluid volume fractions, respectively, \mathbf{u}_s and \mathbf{u}_f are the solid and fluid velocities, respectively, and \mathbf{u} is the composite velocity.

- **Mass conservation equation**

The mass conservation equation was derived by simple substitutions of the fluid quantities:

$$\begin{aligned} \frac{\partial}{\partial t} (\phi_f) + \nabla \cdot (\phi_f \mathbf{u}_f) &= 0 \\ \Rightarrow \frac{\partial}{\partial t} (1 - \phi_s) + \nabla \cdot (\mathbf{u} - \phi_s \mathbf{u}_s) &= 0 \\ \Rightarrow \nabla \cdot \mathbf{u} &= \frac{\partial}{\partial t} (\phi_s) + \nabla \cdot (\phi_s \mathbf{u}_s). \end{aligned} \quad (\text{A3})$$

- **Momentum conservation equation**

The initial fluid momentum equation is:

$$\underbrace{\frac{\partial}{\partial t} (\phi_f \mathbf{u}_f)}_{\text{I}} + \nabla \cdot \underbrace{(\phi_f \mathbf{u}_f \otimes \mathbf{u}_f)}_{\text{II}} = -\frac{1}{\rho} \nabla P + \frac{1}{\rho} \nabla \cdot \underbrace{(\phi_f \boldsymbol{\tau}_f)}_{\text{III}} + \mathbf{f}_{\text{IB}}. \quad (\text{A4})$$

Let us examine term by term how the composite velocity appears. The time derivative term changes as follows:

$$\text{I} : \frac{\partial}{\partial t} (\phi_f \mathbf{u}_f) = \frac{\partial \mathbf{u}}{\partial t} - \frac{\partial}{\partial t} (\phi_s \mathbf{u}_s). \quad (\text{A5})$$

For the convective term, the following operations take place:

$$\begin{aligned}
\Pi : \phi_f \mathbf{u}_f \otimes \mathbf{u}_f &= \mathbf{u} \otimes \mathbf{u}_f - \phi_s \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \mathbf{u} \otimes (\phi_f + \phi_s) \mathbf{u}_f - \phi_s \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \mathbf{u} \otimes \phi_f \mathbf{u}_f + \mathbf{u} \otimes \phi_s \mathbf{u}_f - \phi_s \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \mathbf{u} \otimes \mathbf{u} - \mathbf{u} \otimes \phi_s \mathbf{u}_s + \mathbf{u} \otimes \phi_s \mathbf{u}_f - \phi_s \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \mathbf{u} \otimes \mathbf{u} - \phi_s \underbrace{(\mathbf{u} \otimes \mathbf{u}_s - \mathbf{u} \otimes \mathbf{u}_f + \mathbf{u}_s \otimes \mathbf{u}_f)}_{\text{IV}}.
\end{aligned} \tag{A6}$$

Developing further the term IV we get:

$$\begin{aligned}
\text{IV} : \mathbf{u} \otimes \mathbf{u}_s - \mathbf{u} \otimes \mathbf{u}_f + \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \phi_f \mathbf{u}_f \otimes \mathbf{u}_s + \phi_s \mathbf{u}_s \otimes \mathbf{u}_s - \mathbf{u} \otimes \mathbf{u}_f + \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \phi_f \mathbf{u}_f \otimes \mathbf{u}_s + \phi_s \mathbf{u}_s \otimes \mathbf{u}_s - \phi_f \mathbf{u}_f \otimes \mathbf{u}_f - \phi_s \mathbf{u}_s \otimes \mathbf{u}_f + \mathbf{u}_s \otimes \mathbf{u}_f \\
&= \phi_f \mathbf{u}_f \otimes \mathbf{u}_s + \phi_s \mathbf{u}_s \otimes \mathbf{u}_s - \phi_f \mathbf{u}_f \otimes \mathbf{u}_f - (1 - \phi_f) \mathbf{u}_s \otimes \mathbf{u}_f + \mathbf{u}_s \otimes \mathbf{u}_f \\
&= 2\phi_f \mathbf{u}_f \otimes \mathbf{u}_s + (1 - \phi_f) \mathbf{u}_s \otimes \mathbf{u}_s - \phi_f \mathbf{u}_f \otimes \mathbf{u}_f \\
&= \mathbf{u}_s \otimes \mathbf{u}_s + \phi_f (2\mathbf{u}_f \otimes \mathbf{u}_s - \mathbf{u}_s \otimes \mathbf{u}_s - \mathbf{u}_f \otimes \mathbf{u}_f) \\
&= \mathbf{u}_s \otimes \mathbf{u}_s - \phi_f (\mathbf{u}_s - \mathbf{u}_f) \otimes (\mathbf{u}_s - \mathbf{u}_f).
\end{aligned} \tag{A7}$$

Injecting this formula back to term II gives:

$$\Pi : \phi_f \mathbf{u}_f \otimes \mathbf{u}_f = \mathbf{u} \otimes \mathbf{u} - \phi_s \mathbf{u}_s \otimes \mathbf{u}_s + \phi_s \phi_f (\mathbf{u}_s - \mathbf{u}_f) \otimes (\mathbf{u}_s - \mathbf{u}_f). \tag{A8}$$

The last term can be neglected for a sharp representation of the solid volume fraction. The product $\phi_s \phi_f (\mathbf{u}_s - \mathbf{u}_f)$ can be neglected since $\phi_s \phi_f = 0$ away from the solid/fluid interface and $\mathbf{u}_f \approx \mathbf{u}_s \approx \mathbf{u}$ at the interface. Hence, the convective term in the momentum equation can be expressed as:

$$\nabla \cdot (\phi_f \mathbf{u}_f \otimes \mathbf{u}_f) = \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\phi_s \mathbf{u}_s \otimes \mathbf{u}_s). \tag{A9}$$

For the diffusive term we first define a composite dynamic viscosity in the same manner as the composite velocity:

$$\mu = \phi_f \mu_f + \phi_s \mu_s, \tag{A10}$$

but we set the solid viscosity to zero to avoid shear flows inside the solid domain and so that only the pressure term would counteract the forcing terms of the VOS-IVP method. So, the composite viscosity μ used in the solver is equal to $\phi_f \mu_f$.

Term III can be re-arranged in the following manner:

$$\begin{aligned} \text{III} : \phi_f \boldsymbol{\tau}_f &= \mu \left(\nabla(\phi_f \mathbf{u}_f) + \nabla(\phi_f \mathbf{u}_f)^T \right) \\ &= \mu \left(\nabla(\mathbf{u} - \phi_s \mathbf{u}_s) + \nabla(\mathbf{u} - \phi_s \mathbf{u}_s)^T \right), \end{aligned} \quad (\text{A11})$$

where since the viscosity is a multiple of the fluid volume fraction, the cross terms including the product $\mu \phi_s$ can be neglected. Thus, a new viscous stress tensor $\boldsymbol{\tau}$ can be computed from the composite fields, where:

$$\text{III} : \phi_f \boldsymbol{\tau}_f = \mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) = \boldsymbol{\tau}. \quad (\text{A12})$$

Finally, replacing all the terms containing the fluid velocity with those containing the composite velocity gives the final form of the momentum conservation equation describing both phases at once:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho} \nabla P + \frac{1}{\rho} \nabla \cdot \boldsymbol{\tau} + \frac{\partial}{\partial t} (\phi_s \mathbf{u}_s) + \nabla \cdot (\phi_s \mathbf{u}_s \otimes \mathbf{u}_s) + \mathbf{f}_{\text{IB}}. \quad (\text{A13})$$

References

1. Peskin CS. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics* 1972; 10: 252-271. doi: 10.1016/0021-9991(72)90065-4
2. Mittal R, Iaccarino G. Immersed boundary methods. *Annual Review of Fluid Mechanics* 2005; 37: 239-261. doi: 10.1146/annurev.fluid.37.061903.175743
3. Sotiropoulos F, Yang X. Immersed boundary methods for simulating fluid-structure interaction. *Progress in Aerospace Sciences* 2014; 65: 1-21. doi: 10.1016/j.paerosci.2013.09.003
4. Huang WX, Tian FB. Recent trends and progresses in the immersed boundary method. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 2019; 0: 1-20. doi: 10.1177/0954406219842606
5. Kim W, Choi H. Immersed boundary methods for fluid-structure interaction: A review. *International Journal of Heat and Fluid Flow* 2019; 75: 301-309. doi: 10.1016/j.ijheatfluidflow.2019.01.010
6. Roy S, De A, Balaras E. *Immersed Boundary Method - Development and Applications*. Springer Singapore . 2020

7. Goldstein D, Handler R, Sirovich L. Modeling a No-Slip Flow Boundary with an External Force Field. *Journal of Computational Physics* 1993; 105: 354-366. doi: 10.1006/jcph.1993.1081
8. Angot P, Bruneau CH, Fabrie P. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik* 1999; 81: 497-520. doi: 10.1007/s002110050401
9. Mohd-Yusof J. Combined immersed boundary/B-spline methods for simulation of flow in complex geometries. *Center for Turbulence Research Annual Research Briefs* 1997: 317-328.
10. Udaykumar HS, Mittal R, Rampungoon P, Khanna A. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries. *Journal of Computational Physics* 2001; 174: 345-380. doi: 10.1006/jcph.2001.6916
11. Kim J, Kim D, Choi H. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *Journal of Computational Physics* 2001; 171: 132-150. doi: 10.1006/jcph.2001.6778
12. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *Journal of Computational Physics* 2000; 161: 35-60. doi: 10.1006/jcph.2000.6484
13. Tullio MDD, Verzicco R, Iaccarino G, Iaccarino G. Immersed Boundary Technique for Large-Eddy-Simulation. tech. rep., von Karman Institute for Fluid Dynamics; : 2022.
14. Sethian JA, Smereka P. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics* 2003; 35: 341-372. doi: 10.1146/annurev.fluid.35.101101.161105
15. Hirt C, Nichols B. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; 39: 201-225. doi: 10.1016/0021-9991(81)90145-5
16. Liu RKS, Ng KC, Sheu TWH. A volume of solid implicit forcing immersed boundary method for solving incompressible Navier-Stokes equations in complex domain. *Computers and Fluids* 2021; 218: 104856. doi: 10.1016/j.compfluid.2021.104856
17. Ng KC. A collocated finite volume embedding method for simulation of flow past stationary and moving body. *Computers and Fluids* 2009; 38: 347-357. doi: 10.1016/j.compfluid.2008.04.009
18. Morente A, Laviéville J, Legendre D. A penalization method for the simulation of bubbly flows. *Journal of Computational Physics* 2018; 374: 563-590. doi: 10.1016/j.jcp.2018.07.042
19. Kemm F, Gaburro E, Thein F, Dumbser M. A simple diffuse interface approach for compressible flows around moving solids of arbitrary shape based on a reduced Baer–Nunziato model. *Computers & fluids* 2020; 204: 104536.

20. Tsetoglou I. *Advanced immersed boundary methods for wind turbine wake modeling*. PhD thesis. Normandie Université, Rouen, France; 2023.
21. Suzuki K, Inamuro T. Effect of internal mass in the simulation of a moving body by the immersed boundary method. *Computers and Fluids* 2011; 49: 173-187. doi: 10.1016/j.compfluid.2011.05.011
22. Smagorinsky J. General circulation experiments with the primitive equations I. the basic experiment. *Monthly weather review* 1963; 91: 99-164.
23. Germano M, Piomelli U, Moin P, Cabot WH. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A : Fluid Dynamics* 1991; 3: 1760-1765.
24. Lilly DK. A proposed modification of the Germano subgrid-scale closure method. *Phys. Fluids A* 1992; 4.
25. Moureau V, Domingo P, Vervisch L. Design of a massively parallel CFD code for complex geometries. *Comptes Rendus - Mecanique* 2011; 141-148. doi: 10.1016/j.crme.2010.12.001
26. Chorin AJ. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation* 1968; 22: 745-745. doi: 10.1090/S0025-5718-1968-0242392-2
27. Malandain M, Maheu N, Moureau V. Optimization of the deflated Conjugate Gradient algorithm for the solving of elliptic equations on massively parallel machines. *Journal of Computational Physics* 2013; 238: 32-47. doi: 10.1016/j.jcp.2012.11.046
28. Klainerman S, Majda A. Compressible and incompressible fluids. *Communications on Pure and Applied Mathematics* 1982; 35: 629-651. doi: 10.1002/cpa.3160350503
29. Kang LL, Liu LQ, Su WD, Wu JZ. Minimum-domain impulse theory for unsteady aerodynamic force. ; 30(1): 016107.
30. Qu L, Norberg C, Davidson L, Peng SH, Wang F. Quantitative numerical analysis of flow past a circular cylinder at Reynolds number between 50 and 200. *Journal of Fluids and Structures* 2013; 39: 347-370. doi: 10.1016/j.jfluidstructs.2013.02.007
31. Park J, Kwon K, Choi H. Numerical Solutions of Flow Past a Circular Cylinder at Reynolds Numbers up to 160. *1200 KSME International Journal* 1998; 12: 1200-1205.
32. Kravchenko AG, Moin P, Shariff K. B-Spline Method and Zonal Grids for Simulations of Complex Turbulent Flows. *Journal of Computational Physics* 1999; 151: 757-789.
33. Wu J, Shu C. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *Journal of Computational Physics* 2009; 228(6): 1963-1979. doi: doi:10.1016/j.jcp.2008.11.019

34. Dutsch H, Durst F, Becker S, Lienhart H. Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers. *Journal of Fluid Mechanics* 1998; 360: 249-271. doi: 10.1017/S002211209800860X
35. Kumar M, Roy S. A sharp interface immersed boundary method for moving geometries with mass conservation and smooth pressure variation. *Computers and Fluids* 2016. doi: 10.1016/j.compfluid.2016.07.008
36. Ferrer E, Willden RH. A high order Discontinuous Galerkin - Fourier incompressible 3D Navier-Stokes solver with rotating sliding meshes. *Journal of Computational Physics* 2012; 231: 7037-7056.
37. Ramírez L, Foulquié C, Nogueira X, Khelladi S, Chassaing JC, Colominas I. New high-resolution-preserving sliding mesh techniques for higher-order finite volume schemes. *Computers and Fluids* 2015; 118: 114-130. doi: 10.1016/j.compfluid.2015.06.008
38. Ouro P, Stoesser T. An immersed boundary-based large-eddy simulation approach to predict the performance of vertical axis tidal turbines. *Computers and Fluids* 2017; 152: 74-87. doi: 10.1016/j.compfluid.2017.04.003