Development of a high-order continuous Galerkin sharp-interface immersed boundary method and its application to incompressible flow problems

Lucka Barbeau[a], Stéphane Étienne[b], Cédric Béguin[b], Bruno Blais[a,*]

[a] *Research Unit for Industrial Flows Processes (URPEI), Department of Chemical Engineering, École Polytechique de Montréal, PO Box 6079, Stn Centre-Ville, Montréal, QC, Canada, H3C 3A7*
[b] *Department of Mechanical Engineering, École Polytechique de Montréal, PO Box 6079, Stn Centre-Ville, Montréal, QC, Canada, H3C 3A7*

Abstract

The sharp-interface immersed boundary method is a strategy to impose boundary condition on complex geometries and, therefore, simplify the meshing process. This method can offer high-order of accuracy. Most sharp-interface methods have been developed in the context of the finite volume the finite difference methods. In this paper, we introduce, verify and validate a novel high-order sharp-interface immersed boundary method in the context of the finite element method. We apply this method to the incompressible Navier-Stokes equations using a pressure-stabilizing/Petrov-Galerkin (PSPG) stabilization. We verify that we obtain a high order of convergence using a Taylor-Couette flow. We validate the results obtained for the drag, lift and Strouhal number of the flow behind a cylinder at $Re = 200$. We investigate the flow around a sphere at $Re = 100$ and compare the drag force and the characteristics of the recirculating zones with experimental and numerical work obtained in the literature. Finally, the sharp-interface method is used to study a packing of 10 spheres at $Re = 50$ and the results are compared with those obtained with a conformal mesh. It is shown that the sharp-interface immersed boundary is high-order and accurately predicts the steady-state and transient flow around particles as well as the force evaluation.

Keywords: Sharp interface method, Finite Element Method (FEM), Immersed Boundary, High-Order methods, Navier-Stokes Incompressible flow

1. Introduction

Immersed boundary methods (IBM) are a well-established approach to impose boundary conditions on non-matching domains. This strategy has been originally introduced by Peskin in 1972 [1] and was used to analyze the blood flow around a valve within the human heart. This method has the great advantage of allowing the implementation of the boundary condition of a complex immersed (or embedded) topology on a Cartesian structured mesh. This allows the simulation of moving boundaries on a fixed computational domain. Additionally, it also allows the simulation

[*] Corresponding author
Email address: bruno.blais@polymtl.com (Bruno Blais)

of numerous, possibly deforming, objects that are embedded in the computational domain. Consequently, these methods not only greatly reduce the complexity of the mesh, thus simplifying the meshing process, but also open up a large range of possibilities for the simulation of moving and/or deforming domains. Even though IMB were mostly developed in the context of fluid-structure interaction (see [2] for an extensive review), they have also been applied to heat-transfer problems [3] [4] and mass-transfer problems [5]. Multiple formulations of the IBM have appeared over the years since the early work of Peskin. IBM have been applied and refined in all the classical finite numerical methods (finite difference, finite volume, and finite element methods (FEM)) and have been used in numerous engineering fields. Two main categories of IBM have emerged over the years: forcing (or continous) and reconstruction (e.g. sharp-interface) method [2]. Methods based on forcing function work by locally defining source terms at the boundary or within the immersed object. In fluid-structure interaction, the source term represents the momentum transferred to the fluid by the object immersed in it. The original method defined by Peskin in 1972 [1] falls in this category as well as the feedback approach proposed by Goldstein et al. in 1993 [6], or the Nitsche methods exposed by Freund et al. in 1995 [7]. The Lagrange multiplier method presented by Glowinski et al. in 1999 [7] also fall into this category. In contrast, the sharp method rely on reconstructing the boundary condition at the position of the degree of freedom (DOF) around the immersed boundary. The B-spline methods by Mohd-Yusof in 1997 [8] fall in this category and is, to our knowledge, the first usage of this kinds of methods. The sharp interface defined through ghost-cell by Mittal et al. in 2008 [9] also follow this approach.

The two main approaches defined previously have their limitations. The continuous methods are generally limited to low order of convergence as the solution (and its gradient) must be continuous at the boundary. This implies that the gradient is not properly defined at the location of the immersed boundary. As for the sharp methods, they allow a discontinuity in the gradient of the variables and, consequently, enable a higher order of convergence. However, they suffer from limitations when the boundary is moving since the nodes change from being subject to the fluid equation to being a boundary condition. This may lead to issues regarding mass conservation which, in turn, lead to spurious pressure oscillations on the boundary [10].

In recent years, there is an increasing interest in the development of high order methods as they may increase the precision obtained for the same amount of computation time [11, 12]. Most immersed boundary methods in the literature are limited to first or second order of accuracy and only a few allow for higher order [13, 14]. Methods with a high-order of convergence are mostly available in the context of the finite difference and finite volume methods. In FEM, most methods used to represent immersed boundary are based on forcing functions since they can be naturally introduced [7, 15]. These are generally first order methods. There are also a few methods that rely on cut cells and element enrichment to represent the immersed boundary directly [16, 17, 18] . The latter maintains second-order accuracy, but their implementation at higher-order or for elements which are not simplices (e.g. triangles or tetrahedron) is highly challenging.

To our knowledge, the sharp-interface method remains unused in the context of FEM. However, FEM has attractive properties, notably it easily allows for the use of higher order elements, is very suitable for the simulation of viscous flows and can easily be coupled with dynamic mesh (h) or order (p) adaptation.

In this article, we extend the work of Peller *et al.* [19] and Das *et al.* [20] to allow an application of the sharp-interface IBM in the framework of the finite element method within the open-source CFD code Lethe [11] which is based on the deal.II library [21]. Even though the method can be used to implement Dirichlet boundary conditions for most partial differential equations (PDE), we limit our application to the case of the incompressible Navier-Stokes equations and, in particular, to the flow around assembly of static or rotating circles (in 2D) or spheres (in 3D). First, we present the sharp-edge immersed boundary approach as a general approach to impose Dirichlet boundary conditions on immersed objects. Then, we introduce the GLS stabilized incompressible Navier-Stokes formulation with which the sharp-interface IBM is paired. This results in an immersed boundary scheme for which the order can be controlled at run time. This scheme is verified and validated in 2D using a Taylor-Couette flow and von Karman vortex street and in 3D using the flow around a single sphere or an assembly of spheres. We show that the sharp-edge approach introduced therein this work can reach high accuracy and can be used not only to predict the flow around spherical particles, but also precisely calculate the forces and torque acting on them.

2. Sharp-edge immersed boundaries

The objective of the sharp-interface IBM is to allow the solution of the immersed boundary problem to have a discontinuous gradient on either side of the immersed boundary on which a Dirichlet boundary condition is to be imposed. This enables the solution to have a higher order of convergence than traditional approaches. To achieve this, the interpolation function of the solution must be removed on cells cut by the boundary as it would enforce a continuous gradient at the location of the immersed boundary. After the cells cut by the boundary are identified, the contribution of the theses cells to the problem is removed. This disconnects both sides of the problem. The method then replaces the equation of the DOFs located in each cell cut by the boundary with an extrapolation function. The extrapolation function depends on the properties of the immersed boundary (velocity, shape, etc.).

We introduce the following notation:

- We define Ω as the domain in which the PDE is solved using the finite element method.
- We define $\Gamma \in \Omega$ as the immersed body on which a Dirichelt boundary condition is to be imposed.
- We define Ω_γ as a cell cut by the interface and Ω_Γ as the union of all cells which are cut by the interface.
- We define the effective domain Ω_E as the domain in which the finite element problem is not affected by Γ such that $\Omega_E = \Omega - \Omega_\Gamma$.
- We define d as a DOF that lies on the boundary of the effective domain Ω_E for which we want to define a sharp-interface boundary condition :
  - Let $\boldsymbol{x}_d$ be the location of d.
  - Let $\boldsymbol{n}$ be a vector normal to the boundary Γ that passes through $\boldsymbol{x}_d$. The definition of the magnitude of $\boldsymbol{n}$, $||\boldsymbol{n}||$, will be discussed when the sharp-interface method is explained.
  - Let $\boldsymbol{x}_\gamma$ be the intersection point of $\boldsymbol{n}$ with Γ and u_γ the value of the Dirichlet boundary condition at that location.
  - Let Ω_n be the cell which contains d and the point $\boldsymbol{x}_n = \boldsymbol{x}_\gamma + \boldsymbol{n}$.
  - Let e_i be the other degrees of freedom that are owned by Ω_n.

- Let $s(\boldsymbol{x}) = \sum_i \phi_i(\boldsymbol{x})u_i$ be the function for a field u calculated using the interpolation function ϕ_i within the element Ω_n.

Figure 1 presents the domain Ω and the various subdomains as well as their relation to each other.

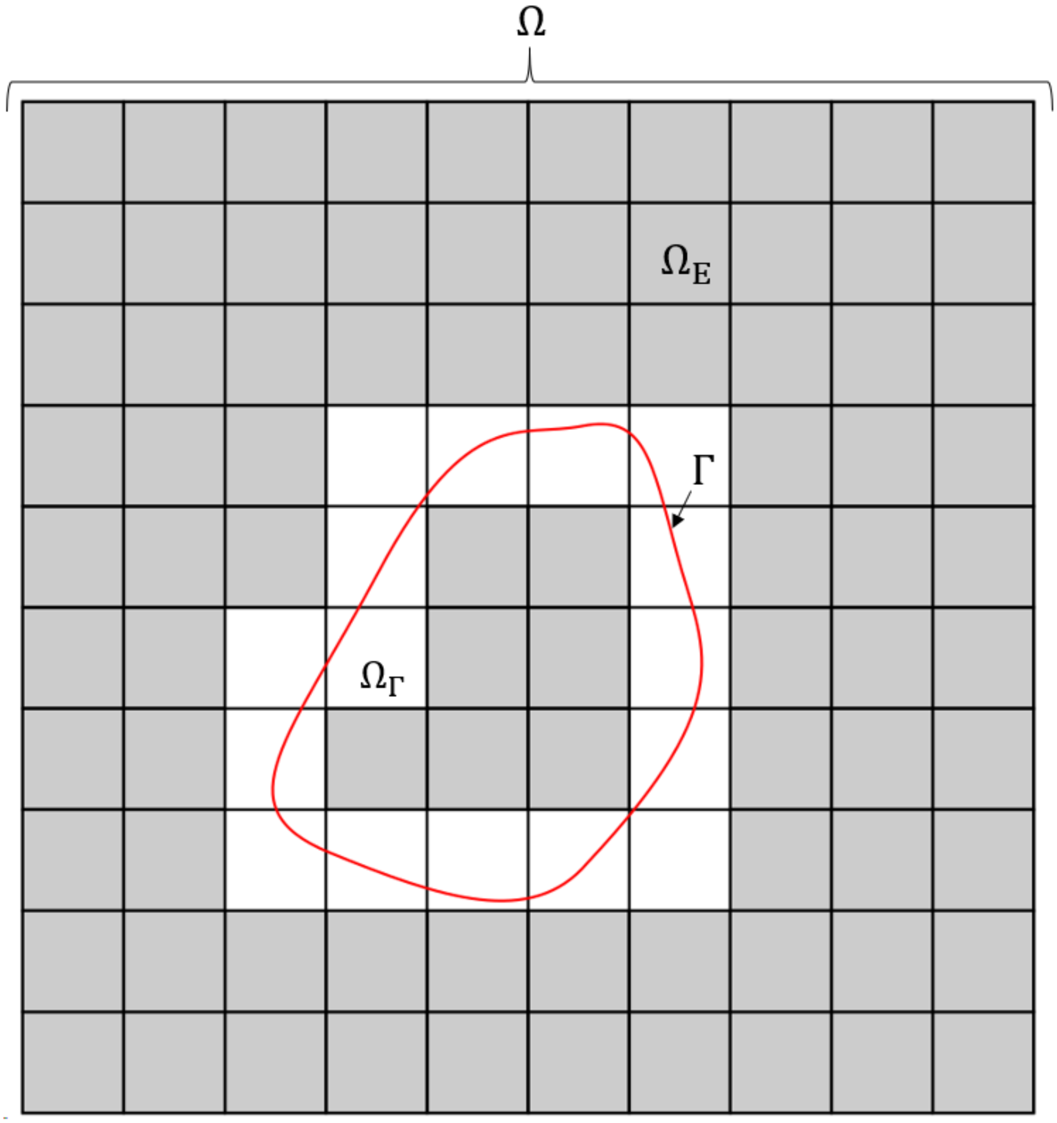

Figure 1: Representation of the domain Ω, the domain on which a PDE is solved, and its constituents : Ω_Γ the cells intersected by the immersed boundary Γ and $\Omega_E = \Omega - \Omega_\Gamma$ the effective domain.

The solution process is defined by the following step:

1. Define the finite element formulation of each cell of Ω_E using the weak formulation of the

differential equation to be solved.

2. If the cell is cut by the immersed body (Γ), modify the finite element problem using the sharp-interface method.
3. Solve the modified system of, possibly non-linear, equations.

As depicted in Figure 2, the sharp-interface method only changes the problem formulation for the cells cut by Γ. Consequently, the finite element problem is formulated as usual on Ω_E. The sharp-interface modification is applied on top of the system matrix obtained by a regular assembly. This linear system is solved and yields the solution to the problem directly if the problem is linear, or using an iterative process (e.g., Newton's method) in the case of a non-linear system of equations. In the case of a non-linear problem, the sharp-edge procedure must be carried out at each matrix assembly.

2.1. Sharp-interface method

For the elements in Ω_Γ, we apply the sharp-interface procedure.

For each d we:

1. Define the vector $\boldsymbol{n}$ that is normal to Γ and passes through $\boldsymbol{x}_d$ (the DOF location).
2. Find Ω_n using $\boldsymbol{x}_n$.
3. Define $s(\boldsymbol{x})^*$ as an approximation of $s(\boldsymbol{x})$ along $\boldsymbol{n}$ using $s(\boldsymbol{x})$.
4. Use $s(\boldsymbol{x})^*$ to define the constraint on the value associated with d using u_γ.

The only exception to this process is when d lies directly on Γ. In this case the value of d is imposed to be equal to u_γ and the sharp-interface modification is neglected for this DOF. All the other DOFs of Ω_Γ that do not satisfy the definition of d have their equation replaced by an identity equation fixing their value to a constant. Henceforth, we refer to those DOFs as dummy DOFs since they do not affect the problem to be solved and they are essentially discarded.

In the following, we describe in detail each step of the aforementioned procedure to establish the sharp-interface boundary method.

2.1.1. Definition of the vector normal to the boundary and finding $\boldsymbol{x}_n$

Using a vector normal to the boundary allows the use of one-dimensional extrapolation functions. These functions are simpler to implement and can appropriately be used to define a Dirichlet boundary condition [9, 22]. Defining a vector normal to a boundary and passing through a specific point is often not trivial since, in some cases, there are multiple acceptable vectors. Multiple methods can be used to define the vector $\boldsymbol{n}$. This choice depends on the strategy used to describe Γ. The most commonly used approaches to describe Γ in the case of immersed boundaries are unstructured meshes [9] and level-set functions [23, 17]. When an unstructured mesh is used to describe the boundary, the normal is found using a search algorithm to find the closest cell to x_d. As for the level-set method, the gradient of the distance function gives the orientation of the normal of the immersed boundary at any given point. In the present work, we are interested in the flow of *spherical* particles (circles in 2D , spheres in 3D). In this case, the orientation of the normal vector is directly defined using the center of the circle/sphere. After the orientation of the vector is established, $\boldsymbol{x}_\Gamma$ can be found easily, as it is the intersection point of the line parallel with n and the surface Γ. From there it's important to fix the norm of $\boldsymbol{n}$. The vector's norm is constrained by two factors: the point $\boldsymbol{x}_n$ must be found in a cell that contains d, and must be greater than $||\boldsymbol{x}_d - \boldsymbol{x}_\gamma||$. The following definition of $\boldsymbol{n}$ satisfies these requirements:

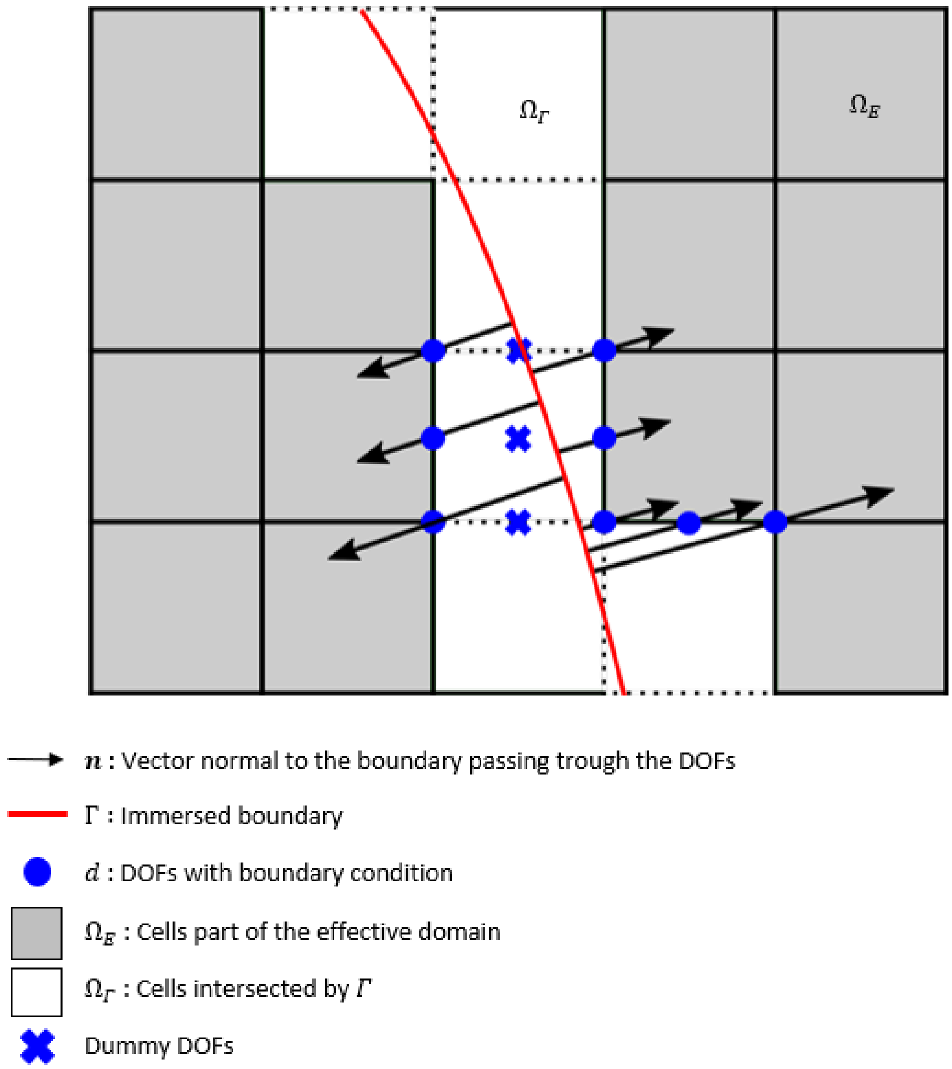
Ω_Γ
Ω_E
$\boldsymbol{n}$: Vector normal to the boundary passing trough the DOFs
Γ : Immersed boundary
d : DOFs with boundary condition
Ω_E : Cells part of the effective domain
Ω_Γ : Cells intersected by Γ
Dummy DOFs

Figure 2: Representation of a Q2 element being cut by an immersed boundary Γ. Present the normal to the boundary and d the DOFs that forms the boundary of the effective domain. The boundary value of those DOF allow the proper representation of the Boundary Γ that lies outside of the effective domain Ω_E.

$$\boldsymbol{n} = (\boldsymbol{x}_d - \boldsymbol{x}_\gamma)(1 + \varepsilon_n) \tag{1}$$

The maximum value of ε_n is specific to each mesh element. In this study we restrict ourselves to a Cartesian mesh using continuous Galerkin elements. This allows ε to be bound by the following inequality :

$$0 \leq \varepsilon_n \leq \frac{1}{2}\delta_q \tag{2}$$

where δ_q is the smallest distance between the DOFs. For example, in the case of Lagrange's polynomials built with equidistant points, δ_q is equal to $1/m$ with m the order of the element. In our case, the polynomials are built either using Lagrange polynomials with equidistant (degree up to 2) or Gauss-Lobatto (starting from degree 3) support points [21]. Choosing ε_n and, consequently, $\boldsymbol{n}$ in this fashion ensures that $\boldsymbol{x}_n$ lies in a cell which contains d if the mesh is sufficiently structured.

Once $\boldsymbol{n}$ is defined, $\boldsymbol{x}_n$ can be defined trivially and it is possible to find Ω_n using a simple search algorithm around the cells that contain d.

2.1.2. Definition and use of $s(\boldsymbol{x})^*$

As previously defined, $s(\boldsymbol{x})^*$ is the approximation of $s(\boldsymbol{x})$ along $\boldsymbol{n}$. Since the objective of the sharp-interface method is to conserve the order of the underlying FEM scheme, we constrain the order of $s(\boldsymbol{x})^*$ to be equal or higher to that of the finite element interpolation. Since the Lagrange elements used in this work are defined as the tensor product of polynomial function Q_m, we use a Lagrange polynomial of the same order (m). We note that alternative extrapolation schemes could be used without altering what follows as long as this constraint is satisfied. We also consider the case where an extrapolation with a higher order of accuracy than the underlying scheme is used to impose the sharp-interface boundary.

In this case, the Lagrange polynomial $s(\boldsymbol{x})^*$ is defined using m points along $\boldsymbol{n}$ comprised within the interval $[\boldsymbol{x}_d, \boldsymbol{x}_n]$. Here we define $\boldsymbol{x}_s^j$ as the jth sample point, with $j \in [1, m]$. From the solution at $s(\boldsymbol{x}_s^j)$ and its associated weight, a_s^j, we get the linear combination of the solution for $s(\boldsymbol{x}_\gamma)^*$.

The coefficients a_s^j can be obtained from the formulation of $\boldsymbol{n}$ and the Lagrange polynomial approximation of $s(\boldsymbol{x})$ to generate $s(\boldsymbol{x})^*$. Figure 3 present an example of the relationship between $\boldsymbol{x}_\gamma$, $\boldsymbol{x}_d$, $\boldsymbol{x}_n$ and the sampling point $\boldsymbol{x}_s^j$. As an example, Table 1 gives the a_s^j for $\varepsilon_n = \frac{1}{8}$. In this table, the sampling points are numbered from the closest to $\boldsymbol{x}_d$ to the farthest and it is assumed that the sampling points are uniformly distributed between $\boldsymbol{x}_d$ and $\boldsymbol{x}_n$. This value of ε_n is chosen since it is sufficiently small to allow for elements of interpolation order up to four.

Another option would be to use the finite element interpolation function of the cell $s(\boldsymbol{x})$ directly and impose that its extrapolation is equal to the boundary value at $\boldsymbol{x}_\gamma$. In this case, the support function of the cell is evaluated at the boundary and the solution at the boundary is set equal to u_γ. This means that $s(\boldsymbol{x})^* = s(\boldsymbol{x})$. For quadrilaterals elements in 2D and hexahedral elements in 3D, the support function along any vector that does not lie on a face is described by a polynomial function of order $m * dim$ where dim is the number of spatial dimensions in which the problem is solved. Since the order of the polynomial that defines $s(\boldsymbol{x})^*$ is much higher than the polynomial order of the scheme, $s(\boldsymbol{x})^*$ can be subject to Runge oscillations. Using a polynomial function that is susceptible to Runge oscillation for an extrapolation could affect the convergence and the stability of high order elements.

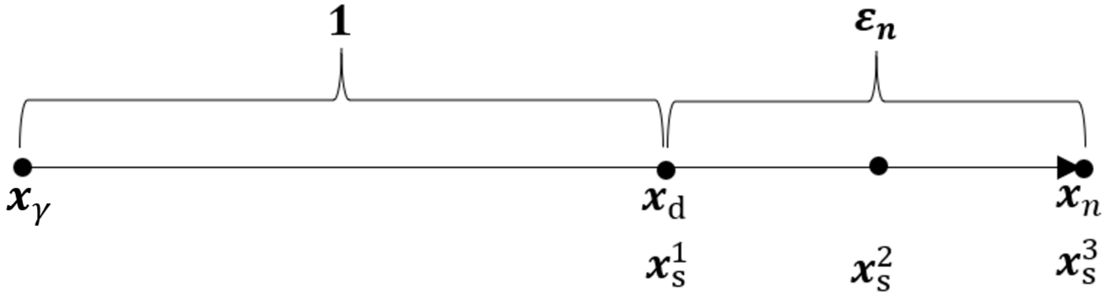
$\mathbf{1}$
$\boldsymbol{\varepsilon_n}$
$\boldsymbol{x}_\gamma$
$\boldsymbol{x}_{\mathrm{d}}$
$\boldsymbol{x}_n$
$\boldsymbol{x}_{\mathrm{s}}^1$
$\boldsymbol{x}_{\mathrm{s}}^2$
$\boldsymbol{x}_{\mathrm{s}}^3$

Figure 3: Representation of $\boldsymbol{n}$ and the key points along it. Here a second order Lagrange approximation of $s(\boldsymbol{x})$ is used to define $s(\boldsymbol{x})^*$. This function will then be used to define the boundary condition applied on d

Coefficient	Lagrange polynomial order for $s(\boldsymbol{x})^*$			
	1	2	3	4
a_s^1	9	153	2925	58905
a_s^2	-8	-288	-8424	-228480
a_s^3		136	8100	332640
a_s^4			-2600	-215424
a_s^5				52360

Table 1: Coefficients a_s^j that form $s(\boldsymbol{x})^*$ when n is defined based on equation 1 with $\varepsilon_n = 1/8$. The coefficients are given for a polynomial approximation of $s(\boldsymbol{x})^*$ of order lower or equal to 4.

Both formulations can be used to define $s(\boldsymbol{x})^*$ and we assess the impact of this choice within the present work.

Once $s(\boldsymbol{x})^*$ is defined, it can be used to impose the boundary condition at the location of the boundary $\boldsymbol{x}_\gamma$. To impose a Dirichlet boundary condition, we use the boundary value u_γ and impose the following :

$$u_\gamma = s(\boldsymbol{x_\gamma})^* \tag{3}$$

Since $s(\boldsymbol{x})^*$ can be expressed as a linear combination of e_i and d, the formulation of the boundary condition take the following form:

$$u_\gamma = \sum_{i=1}^{n_{dof}-1} a_i e_i + a_d d \tag{4}$$

where n_{dof} is the number of degree of freedom and a_i is the weight of the DOF e_i in $s(\boldsymbol{x})^*$ at the location of $\boldsymbol{x}_\Gamma$. The value of a_d is defined in the same way but is linked to the DOF d. In the case where $s(\boldsymbol{x})^* \neq s(\boldsymbol{x})$ we have :

$$a_i = \sum_{j=1}^{m} \phi_i(\boldsymbol{x}_s^j) a_s^j \tag{5}$$

In the case where $s(\boldsymbol{x})^* = s(\boldsymbol{x})$ we have :

$$a_i = \phi_i(\boldsymbol{x}_\gamma) \tag{6}$$

This equation is then used for d and, consequently, becomes the boundary condition for the domain Ω_E.

3. Application to the incompressible Navier-Stokes equations

This section links the sharp-interface immersed boundary method defined in Section 2 to its application to the incompressible Navier-Stokes (INS) equations. We recall the incompressible Navier-Stokes equations subject to a Dirichlet boundary condition on an immersed (or embedded) domain Γ:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{7}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\nabla p^* + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f} \tag{8}$$

with

$$\boldsymbol{\tau} = \nu \left((\nabla \boldsymbol{u}) + (\nabla \boldsymbol{u})^T \right) \tag{9}$$

and

$$\boldsymbol{u} = \boldsymbol{u_\gamma}(\boldsymbol{x}, t)\; on\; \boldsymbol{\Gamma} \tag{10}$$

where $\boldsymbol{u}$ is the velocity vector, $p^* = \frac{p}{\rho}$ with p the pressure and ρ the density, $\boldsymbol{\tau}$ the deviatoric stress tensor, ν the kinematic viscosity, $\boldsymbol{f}$ a body force, Γ the immersed boundary on which a Dirichlet boundary condition of value $\boldsymbol{u}_\gamma(\boldsymbol{x}, t)$ is imposed. In what follows, $\boldsymbol{u}_\gamma(\boldsymbol{x}, t)$ will simply be noted $\boldsymbol{u}_\gamma$.

The implementation of the immersed boundary method is done using the Galerkin Least Square (GLS) approach as it allows the use of equal order finite elements for the pressure and the velocity components [24, 25, 11]. This results in the following weak formulation for the INS equations:

$$\int_\Omega \nabla \cdot \boldsymbol{u} q d\Omega + \sum_K \int_{\Omega_k} \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p^* - \nabla \cdot \boldsymbol{\tau} - \boldsymbol{f} \right) \cdot (\tau_u \nabla q) \, d\Omega_k = 0 \tag{11}$$

$$\begin{aligned} &\int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{f} \right) \cdot \boldsymbol{v} d\Omega + \int_\Omega \boldsymbol{\tau} : \nabla \boldsymbol{v} d\Omega - \int_\Omega p^* \nabla \cdot \boldsymbol{v} d\Omega \\ &+ \sum_K \int_{\Omega_k} \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p^* - \nabla \cdot \boldsymbol{\tau} - \boldsymbol{f} \right) \cdot (\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{v}) \, d\Omega_k = 0 \end{aligned} \tag{12}$$

The stabilization parameter τ_u can take 2 forms which depends on the fact that the problem is stationnary or transient. For the transient situation we get Eq. (**??**) and for the stationary case we get Eq. (13).

$$\tau_u = \left[\left(\frac{2 \left| \boldsymbol{u} \right|}{h_{conv}} \right)^2 + \left(\frac{4\mu}{3 h_{diff}} \right) \right]^{-1/2} \tag{13}$$

where Δt is the time step, h_{conv} and h_{diff} are the size of the element related to the convective transport and diffusion mechanism, respectively [26, 24]. In Lethe, both element size (h_{conv} and h_{diff}) are set to the diameter of a sphere of a volume equivalent to that of the cell [27, 25, 11].

Since the cells that are cut by Γ are not assembled, the weak formulation of the problem is not applied to those cells. This ensures that even when we replace the equation for the velocity DOF, the pressure on both sides of the immersed boundary is not linked through the pressure-stabilizing/Petrov-Galerkin (PSPG) stabilization. Consequently, both sides of the boundary are treated as two separate problems. The pressures DOF that are only part of cells cut are neglected and are treated as dummy DOFs as they have no impact on the problem. If the boundary is closed, a pressure reference must be given inside the domain delimited by the immersed boundary Γ.

One interesting use of the IBM is to model the fluid-structure interaction. This implies that it is often interesting to obtain the force on the boundaries that are represented by Γ. The calculation of flow proprieties at the location of Γ is not trivial since the solution of the velocity field and the pressure field at the location of the boundary are not properly defined. This is because they lie within cells that are cut and, consequently, the support function for these cells is not the finite element interpolation. As the boundary is defined by the extrapolation of the neighbouring cell on either side of it, a natural solution is to use those same cells to evaluate the solution at the location of the boundary through extrapolation. This allows the reconstruction of the pressure and the velocity gradient on the immersed boundary. The boundary is then discretized and sampled at multiple locations. Once the evaluation points are established, the normal to the boundary at these evaluation points is defined. Along the normal, the solution is sampled on the effective domain Ω_E.

From those sampling points, it is possible to get an approximation of the solution at the boundary evaluation point and to calculate the local force acting on the immersed object.

The steps to calculate the force/torque acting on an immersed objected are:

1. Discretize the boundary Γ into points $\boldsymbol{x}_\gamma$ that represent a part of Γ.
2. Find γ_n a vector normal to the boundary for each of the $\boldsymbol{x}_\gamma$.
3. Step in the direction of γ_n until the point $\boldsymbol{x}_\gamma^n = \boldsymbol{x}_\gamma + \gamma_n$ is found in Ω_E.
4. Sample the solution in the direction of γ_n the required number of time for the extrapolation function used.
5. Use the solution obtained for each of the $\boldsymbol{x}_\gamma$ to obtain the desired derived quantity.

This method does not use the same vector normal to Γ and the same associated points $\boldsymbol{x}_\Gamma$ that are used to define the sharp-edge boundary condition. This is due to the non-uniformity of their distribution on the boundary. Determining the surface area represented by a sampling point on the boundary when the distribution is not uniform is computationally inefficient. It is much more effective to discretize Γ into sampling points and to define the solution at those sampling points.

It is also important to note that the error made when evaluating proprieties at the location of Γ will be function of the discretization used for the evaluation points $\boldsymbol{x}_\gamma$ and the approximation function used to evaluate the solution at those points.

The sharp-interface immersed boundary methods and the GLS formulation of the INS equations are implemented in Lethe, an open source computational fluid dynamics (CFD) software based on the deal.II library [21]. The full definition of the methods and capability of this CFD solver, notably regarding the linear and non-linear solver used to solve the INS equations, are detailed in [11].

4. Applications examples

We now demonstrate the capacity of the GLS sharp-interface method by applying it to the study of various test cases. The goal is to verify the order of convergence of the proposed IBM and to validate the results obtained by comparing them to reference values in the literature obtained either through experimental means or other numerical simulations.

4.1. Taylor-Couette flow in 2D

The Taylor-Couette flow, or flow between two-concentric cylinders, is a well-established test case for immersed boundary conditions due to the fact that the geometry is not aligned with the mesh and that such flow possesses an analytical solution that is sufficiently rich. In the present case, we consider two co-axial cylinders of radii $R_i = 0.25$ and $R_o = 0.5$, with the inner cylinder rotating. The resulting Taylor number is $Ta = \frac{1}{256}$, which is much smaller than the critical $Ta_c \approx 1708$, ensuring that the solution those not present Taylor vortices. The cylinders are both discretized using the immersed boundary method on a square domain $\Omega = [-1, 1] \times [-1, 1]$. Assuming that the flow is laminar, the azimuthal velocity profile between the two cylinders is given by [28]:

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

Where $\kappa = \frac{R_i}{R_o}$

Since we have access to a non-trivial analytical solution, we can use it to demonstrate the order of convergence of the velocity field and the force evaluation. Both the inner and outer cylinders are defined by immersed boundaries. This problem is solved for the following finite element interpolation order for velocity-pressure : Q1-Q1, Q2-Q1, Q2-Q2, Q3-Q2, Q3-Q3. As a reminder, the expected convergence order p_f for a finite element scheme is function of the polynomial order used for the support function of the elements and follow $p_f = m + 1$ where m is the polynomial order. For each cell, two stencil formulations are used and compared. They are noted S_n where n is the order of the Lagrange polynomial used for the sharp-interface stencil. The non-linear equations are solved until the residual is inferior to 10^{-10}. The viscous force evaluation is realised for elements of order equal or lower to Q2-Q2 because the stencils used for force evaluation is limited to second order of convergence. The mesh size used for all of these cases are: 32X32, 64X64, 128X128, 256X256 elements. Figure 4 illustrates the norm of the velocity field for one of the solutions obtained on a 256X256 mesh. Also the figure 5 a comparison of the azimuthal velocity observed on the 64X64 mesh using Q1-Q1 S1, Q2-Q1 S2, Q3-Q2 S3 elements and the analytical solution. As we can see the error is relatively small for all of these cases and the results are in good agreement with the analytical solution. This leads to the results presented in the tables 2 and 3. These tables present the L2 norm of the error of the velocity for the various element types and mesh sizes. This table also presents the average convergence order observed and the convergence order of the last refinement as it is theoretically the closest to the asymptotic region. From these results, we draw two main conclusions.

First, the best average convergence order observed for Q1-Q1, Q2-Q1 and Q2-Q2 elements is within 10 % of the expected convergence order. This is considered acceptably close to the theoretical convergence order [29]. If we look at the best, last convergence order observed for each of the elements, all of them, except Q2-Q1, present a satisfactory convergence order for the velocity field. This means that almost all of the elements have a result that is acceptably close to the expected convergence order. We can conclude that the IBM presented seems to conserve the expected convergence order of the underlying finite element scheme.

Secondly, the choice of stencils used can have an impact on the error. When using the lower bound for the stencil order, the scheme tends to converge to a slightly lower order of convergence than expected. Q1-Q1 elements present significantly better results using a stencil of higher order. This indicates that the bi-linear interpolation function of the finite element interpolation has a significant impact on the accuracy due to it's higher isotropy. However, for Q2-Q1 and Q2-Q2 elements, the smallest error is extremely close between both stencils used. For both these elements, the convergence order was better when using the high order stencil. Q3-Q2 and Q3-Q3 elements both present a smaller error when using the lower order stencil. The convergence order seems to reach the asymptotic region faster while using a higher order stencil as for both Q3-Q2 and Q3-Q3 elements, the best last convergence order was obtained using the high order stencil. This leads to the conclusion that high order stencil lead to higher error for coarse meshes then lower order stencils, but have a convergence order that reach the asymptotic region faster. Indeed, only the Q1-Q1 elements present a lower error for the 32X32 mesh when using a high order stencil. The error difference between the low order stencil and the high order stencil increases with the element order.

It is important to note that the order of the polynomial increases rapidly with the dimension and the element order when using the direct extrapolation of an element (stencil of order higher or equal than $dim * m$). This can lead to Runge oscillations which create instability in the non-linear resolution process. As an example, major instabilities were observed when using Q4-Q3 and Q4-Q4

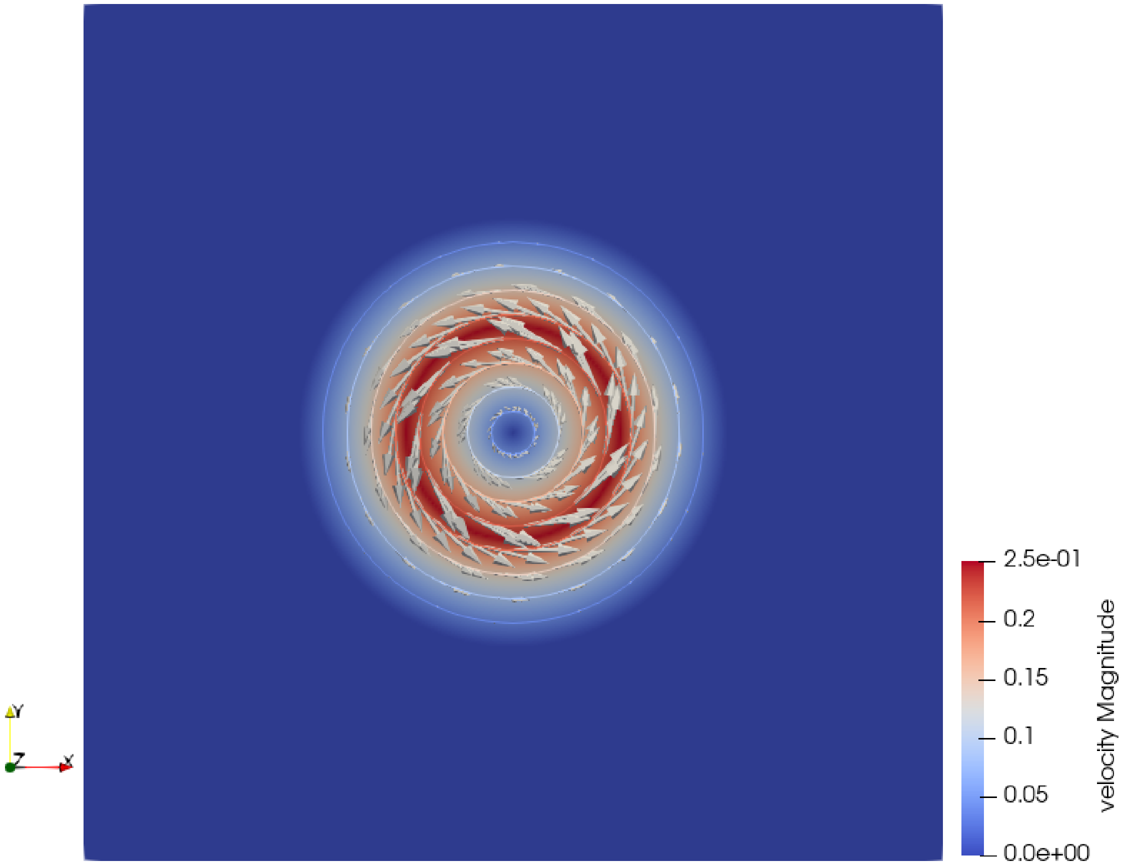

Figure 4: Norm of the velocity field for the Taylor-Couette flow in 2D using 2 sharp immersed boundaries. The flow is solved on both sides of the immersed boundary for both cylinders. The cylinder radius is $R_i = 0.25$ and $R_o = 0.5$ which gives a Taylor number $Ta = \frac{1}{256}$.

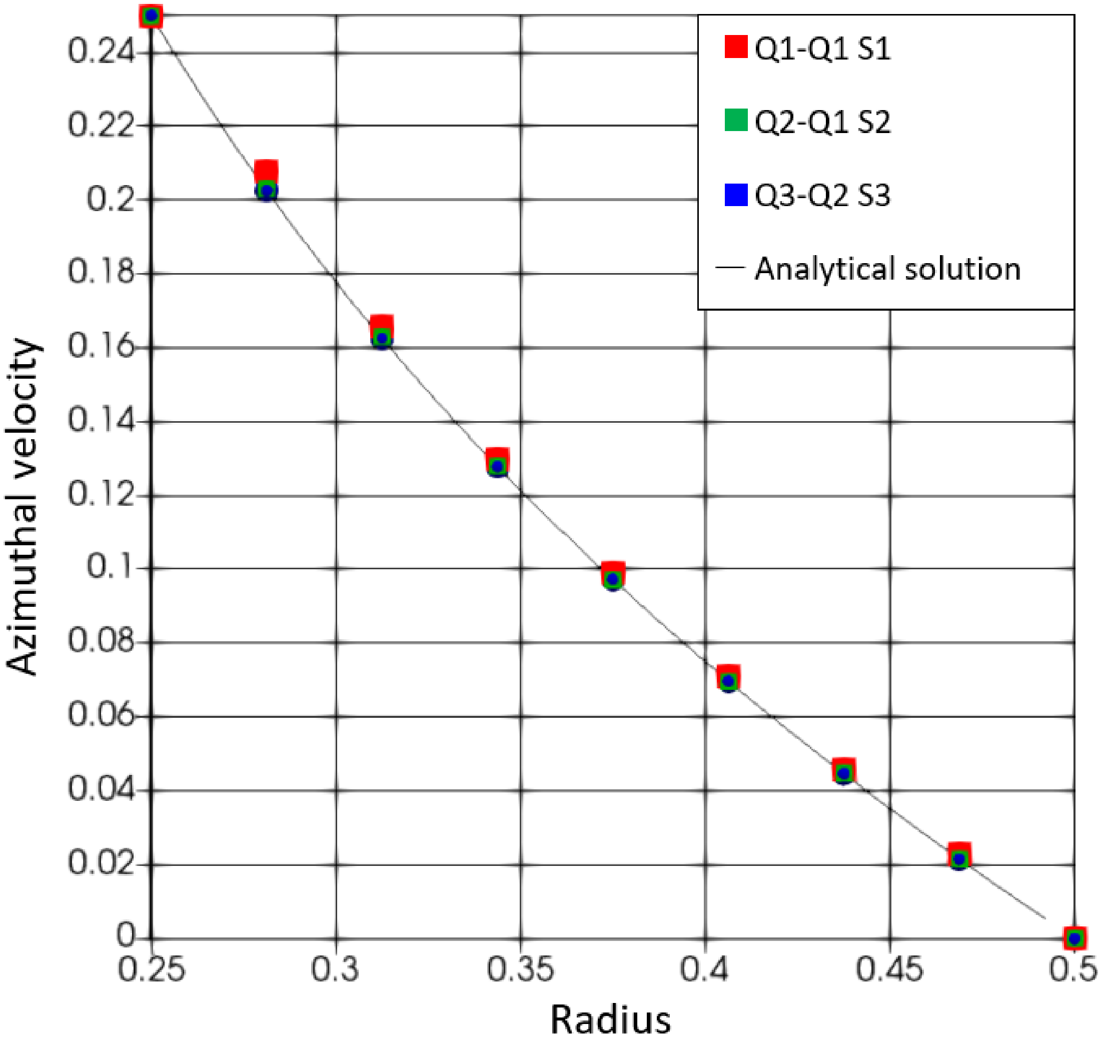

Figure 5: Comparison between the analytical solution and some simulation results of the azimuthal velocity in function of the radius of the Taylor-Couette case. The mesh used for comparison is 64X64 and the simulation results presented are Q1-Q1 S1, Q2-Q1 S2 and Q3-Q2 S3. Note : the results of the Q3-Q2 S3 simulation are difficult to distinguish from those of the Q2-Q1 S2 simulation since they both coincide on the analytical solution.

Mesh	**Q1-Q1**		**Q2-Q1**		**Q2-Q2**	
	S1	**S2**	**S2**	**S4**	**S2**	**S4**
32X32	5.73E-3	4.53E-3	6.37E-4	1.49E-3	7.72E-4	2.70E-3
64X64	1.72E-3	1.19E-3	1.41E-4	1.27E-4	1.75E-4	2.43E-4
128X128	4.67E-4	3.04E-4	2.16E-5	2.08E-5	2.74E-5	2.45E-5
256X256	1.29E-4	7.57E-5	3.48E-6	3.39E-6	4.48E-6	3.53E-6
Average convergence order	1.82	1.96	2.51	2.92	2.47	3.19
Last convergence order	1.85	2.00	2.63	2.62	2.61	2.79

Table 2: Error on the velocity field for the Taylor-Couette cases evaluated in the L2 norm for Q1-Q1, Q2-Q1 and Q2-Q2 elements, using both formulation of $s(\boldsymbol{x})^*$ given in section 2.1.2.

Mesh	**Q3-Q2**		**Q3-Q3**	
	S3	**S6**	**S3**	**S6**
32X32	2.61E-4	7.56E-4	1.62E-4	1.05E-3
64X64	1.85E-5	1.11E-4	1.93E-5	1.37E-4
128X128	2.14E-6	7.48E-6	8.21E-6	9.37E-6
256X256	1.75E-7	5.70E-7	3.12E-7	7.29E-7
Average convergence order	3.51	3.46	3.01	3.49
Last convergence order	3.61	3.71	4.72	3.68

Table 3: Error on the velocity field for the Taylor-Couette cases evaluated in the L2 norm for Q3-Q2, Q3-Q3 elements, using both formulation of $s(\boldsymbol{x})^*$ given in section 2.1.2

elements with a stencil of order 8. This is why the results of these two elements were not presented in this work.

The solution for the torque acting on the inner cylinder can also be obtained analytically from the expression of the velocity field [28]:

$$T_z = -4\pi\Omega_i\mu\frac{R_o^2}{\kappa^{-2}-1} \tag{15}$$

Using this solution, we now analyze the the capacity of the sharp-interface IB to accurately predict the torque acting on the cylinder. The torque were obtained using the procedure described in Section 3. The number of sampling points on the boundary used to evaluate the viscous torque was increased until the solution obtained was independent from it. Table 4 presents the error made on the torque evaluation on the inner cylinder. First we observe that we obtain the proper convergence order for the torque for Q1 elements. For Q2 element we observe a slightly lower then expected convergence rate (1.7 instead of the expected 2). The approximation function uses a three-point stencil with a non-uniform distribution of the points to retrieve the gradient at the boundary.

4.2. Vortex shedding behind a cylinder

The flow around a cylinder is an interesting case to study since it leads to the generation of complex transient flow patterns known as von Karman vortex if the Reynolds number is sufficiently

Mesh	**Q1-Q1**		**Q2-Q1**		**Q2-Q2**	
	S1	**S2**	**S2**	**S4**	**S2**	**S4**
32X32	2.00E-1	1.76E-1	1.12E-1	7.76E-2	1.17E-1	4.50E-2
64X64	1.01E-1	7.95E-2	3.70E-2	2.86E-2	4.02E-2	2.08E-2
128X128	5.52E-2	4.03E-2	1.09E-2	8.55E-3	1.19E-2	7.49E-3
256X256	2.94E-2	1.95E-2	3.28E-3	2.48E-3	3.46E-3	2.42E-3
Average convergence order	0.92	1.05	1.70	1.65	1.69	1.40
Last convergence order	0.91	1.04	1.73	1.78	1.78	1.63

Table 4: Error on the torque evaluation, for the Taylor-Couette cases, on the inner cylinder for Q1-Q1, Q2-Q1 and Q2-Q2 elements, using both formulation of $s(\boldsymbol{x})^*$ given in the section 2.1.2. The analytical solution for this specific case is $T_z = -\frac{\pi}{3}$

high [30, 31] . The details of the geometry used in this case are described in Figure 6. We study the evolution of both components of the force (lift and drag) acting on the cylinder at $Re = 200$. We can relate the frequency of vortex shedding to the Strouhal (St) number using the following:

$$\mathrm{St} = \frac{hD}{u_\infty} \tag{16}$$

with u_∞ the upstream velocity, D the diameter of the cylinder and h the frequency of the shedding.

The values found for the Strouhal number as well as the drag and lift coefficient are compared with similar studies done with other numerical methods and by a large array of authors. This flow is studied in transient using a second-order backward differentiation formula (BDF2) time-stepping scheme and a constant time step 0,05 second for 200 seconds. The mesh is defined by a structured mesh of 128 by 64 cells, with 3 orders of refinement in a radius 1.5 times larger than the cylinder. This gives a total of 11396 elements. We use Q2-Q2 elements and a fourth order stencil to impose the sharp-interface immersed boundary.

This yields the C_D-C_L evolution presented in Figure 7. The evolution of the drag and lift is relatively standard. We see a decrease of the drag coefficient while the flow establish itself and as the instability starts to appear, the drag increases and oscillate around a fixed value. The lift coefficient is constant and null at first but starts oscillating around zero with increasing amplitude until it reaches a stable oscillation pattern once the von Karman alley is fully formed.

In Figure 8, we used the last 100 seconds of the simulation to define the Strouhal number using a fast Fourier transform on the lift coefficient. Table 5 present some of the various results found in the literature for the C_D,C_L and S_t and compares the value of these results to what we obtained using the sharp - interface methods described here. The results obtained are comparable to the results found in the literature for both the Strouhal number and the force applied to the cylinder. The results obtained using the sharp-interface approach are also very close to the results obtain using a conformal mesh with Lethe.

4.3. Flow around a sphere

Similarly to the case of the flow around a cylinder in 2D, the flow around a sphere in 3D has been extensively studied (e.g [39, 40]. We investigate the flow behavior at $Re = 100$ which is sufficiently low to ensure that the solution remains steady. The geometry of the domain is a 3D extension of

Study	C_D	C_L	S_t
Lethe sharp interface IBM	1.395 ±0.047	±0.71	0.200
Lethe Conformal mesh	1.37 ± 0.05	±0.69	0.200
Braza *et al* [30]	1.40 ± 0.05	±0.75	-
Bhalla *et al* [32]	1.39	-	0.2
Choi *et al* [31]	1.36 ± 0.048	±0.64	0.191
He *et al* [33]	1.356	-	0.198
Bergmann *et al* [34]	1.35±0.05	-	0.198
Henderson *et al* [35]	1.341	-	0.197
Wright *et al* [36]	1.33 ± 0.04	±0.68	0.196
Russel and Wang *et al* [37]	1.29 ± 0.022	±0.5	0.195

Table 5: Comparison table of C_L, C_D and S_t found in various numerical studies of flow around a cylinder at Reynolds = 200.

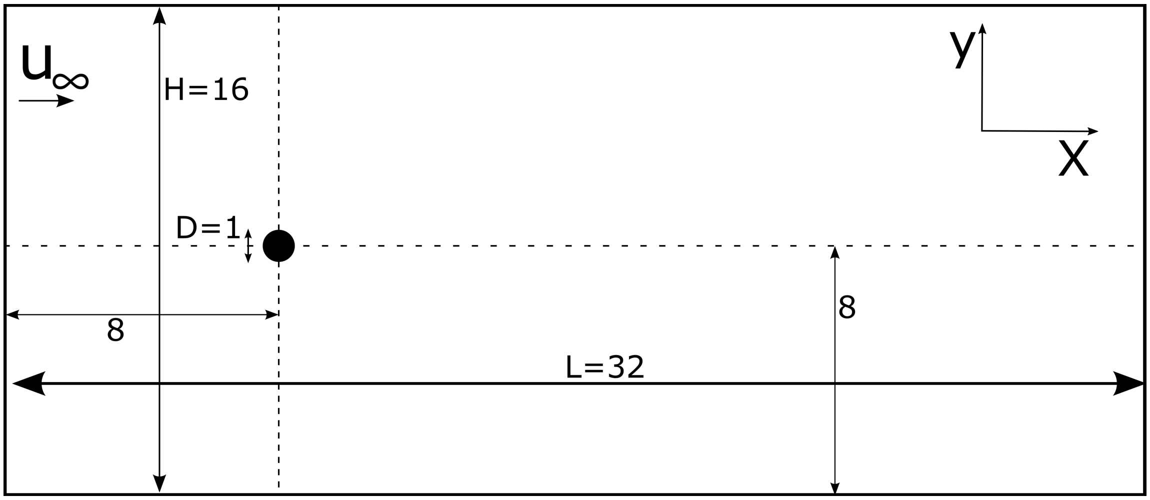

Figure 6: Geometry of the domain for the von Karman vortex analysis around a cylinder at $Re = 200$. Adapted from [38].

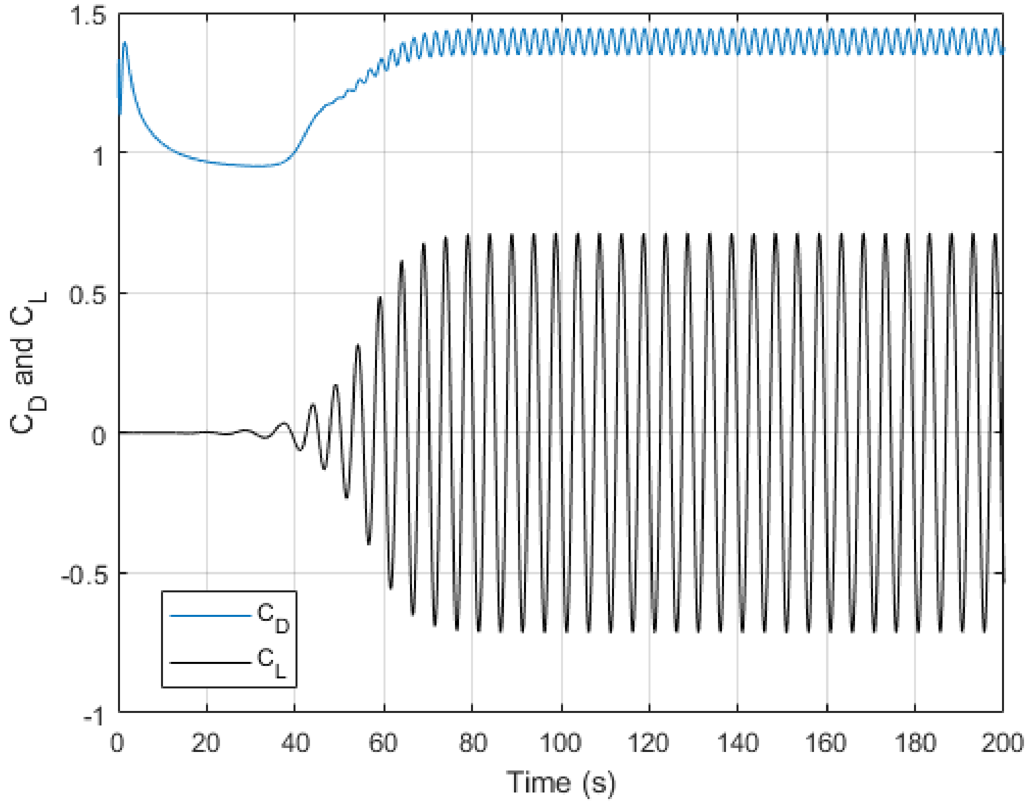

Figure 7: Evolution of the drag and lift coefficient for the vortex shedding cases behind a cylinder. The flow as a Reynold number $Re = 200$ and is simulated over 200 seconds.

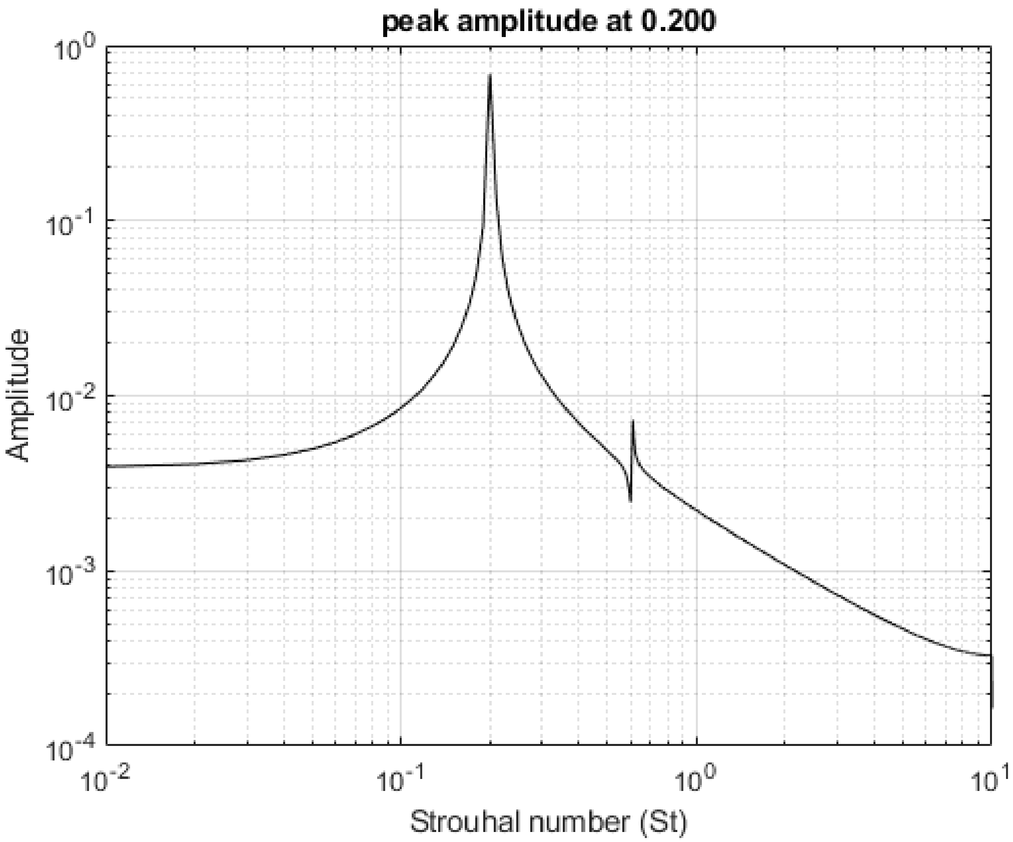

Figure 8: Strouhal number associated with the vortex shedding behind a cylinder obtained using a fast Fourier transform of the lift coefficient in the last 100 second of the simulation.

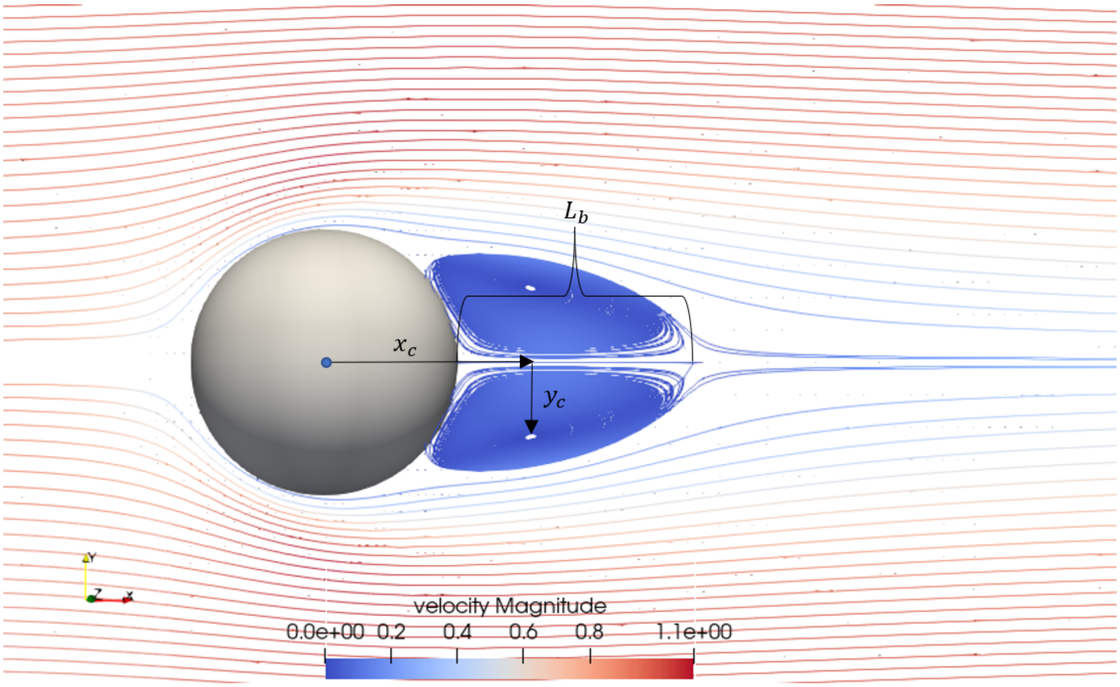

Figure 9: Streamlines for the flow around a sphere at $Re = 100$. Also present the key variable for the geometry of the re-circulation zone.

Study	method	x_c/d	y_c/d	L_b/d	C_d
Present	NIB	0.78	0.28	0.88	1.07
Mittal (1999)[39]	NCM	-	-	0.87	1.09
Bagchi et al [42]	NCM	-	-	0.87	1.09
Johnson and Patel [40]	NCM	0.75	0.29	0.88	1.09
Marella et al[43]	NIB	-	-	0.88	1.06
Mittal (2008) [9]	NIB	0.742	0.278	0.84	-
Cliff et al [44]	EXP	-	-	-	1.09
Hao Zhang et al [45]	NCM	-	-	-	1.10

Table 6: Table of key results for a flow around a sphere at Re=100, Methods use are classified as follow : simulation with a conformal mesh (NCM), simulation with immersed boundary (NIB), experimental (EXP).

the domain presented in Figure 6. This give us a hyper-rectangle domain defined by the diagonal points $(-8, -8, -8)$ and $(24, 8, 8)$ with a sphere of radius 0.5 centered at $(0, 0, 0)$. The solution is obtained using an adaptive mesh based on a Kelly error estimator [41]. The final mesh has as 107 732 Q1-Q1 elements with the smallest element having a length of $\frac{1}{32}$ the diameter of the sphere. The number of forces evaluation on the surface of the boundary was increased until independence of the solution obtain for the drag coefficient is obtained.

The flow pattern obtained is presented in Figure 9. The value measured for the center of the re-circulation zone and the drag are in agreement with the value found in the literature as can be seen in Table 6. This demonstrates the ability of the sharp-interface IBM to accurately describe 3D curved geometries using a Cartesian mesh.

4.4. Flow through packing of sphere

The flow through swarms of particles is highly relevant to numerous engineering application related to mass and energy transfer that occur in unit operations such as packed bed reactors. Mesh generation for such cases is often troublesome because of the necessity of meshing the gap between the particles.

We consider an assembly of 10 spheres. Table 7 presents the disposition of the spheres in a rectangular domain defined by the diagonal points $(-6, -3, -3)$ and $(6, 3, 3)$. The spheres have a radius of 0.5. The Reynolds number is set to 50 using a unit speed at the inlet (Y, Z plane at $x = -6$). The flow is solved using Q1-Q1 element in steady state. We solve the following case on a conformal mesh and using the sharp interface IBM. The conformal mesh is unstructured and has 647488 cells with a minimal characteristic length around a sphere of 0.05. The mesh used for the immersed boundary case is a structured mesh with local refinement containing 407710 cells with a minimal cell length of 0.046875.

Figure 10 presents the results obtained using the sharp interface IBM. In this figure the purple surface represents isosurface at 5% of the upstream velocity. It demonstrates the ability of this method to represent the presence of multiple objects in a flow without the need to define a complex mesh.

From these results we can analyze the force obtained on each of the particle and compare it to the results obtained using immersed boundaries. This case presents a symmetry on each side of the X, Z and X, Y planes. This allows us to look for symmetries in the results for the force. For example, the spheres 2 and 5 should have equal and opposite force in the Y, Z plane.

Table 8 present the force obtained with both approaches (conformal and IBM). We observe that both solutions are in agreement and there is no major difference in the results obtained. The solution obtained using the immersed boundary seems to respect more the symmetries of the problem. For example, the force of the spheres 2 and 5 are equal and opposite at 2 decimal places which is not the case for the results obtain using a conformal unstructured mesh.

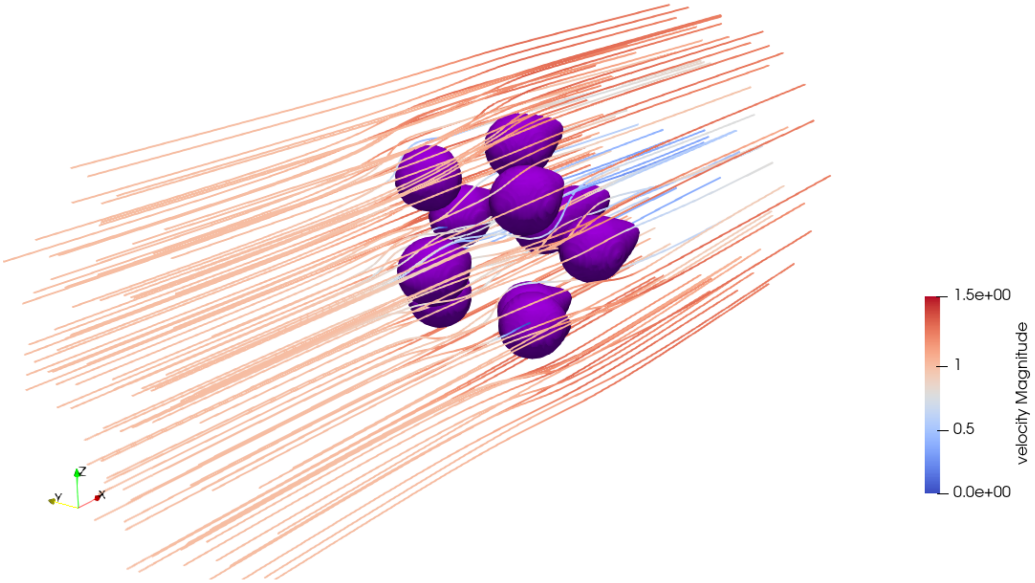

Figure 10: Streamline of a flow through the packing of 10 sphere at $Re_p = 50$. The purple surface is obtained based on the iso-surface of the norm of the velocity field at $||u|| = 0.05$

Sphere ID	P_x	P_y	P_z
1	-1	0	0
2	0	1	1
3	0	-1	1
4	0	1	-1
5	0	-1	-1
6	1	1.414	0
7	1	0	1.414
8	1	0	-1.414
9	1	-1.414	0
10	1.5	0	0

Table 7: Position of the center of the spheres of radius $R = 0.5$ for the case of a flow throw the packing of 10 spheres .

Sphere ID	F_x		F_y $\cdot 10^{-2}$		F_z $\cdot 10^{-2}$	
	CM	SI IB	CM	SI IB	CM	SI IB
1	0.64	0.63	0.03	0.00	0.01	0
2	0.78	0.78	9.35	9.44	9.37	9.51
3	0.78	0.78	-9.15	-9.37	9.18	9.51
4	0.78	0.78	9.28	9.43	-9.36	-9.51
5	0.78	0.78	-9.28	-9.44	-9.29	-9.51
6	0.73	0.73	8.55	8.88	0.04	0
7	0.72	0.73	0.15	0	8.68	8.83
8	0.72	0.73	-0.04	0.02	-8.52	-8.86
9	0.72	0.73	-8.60	-8.88	0.09	0
10	0.52	0.53	-0.04	0	0	0

Table 8: Comparison table of the force on each sphere from table 7 using a conformal mesh (CM) and the sharp interface immersed boundary method (SI IB).

5. Conclusion

In this work we presented a sharp-interface immersed boundary method in the context of the continuous Galerkin finite element method. This method was implemented in a GLS formulation of the incompressible Navier-Stokes equations within the open source CFD software Lethe. Multiple cases were studied and analyzed in order to verify and validate the model. As can be seen in the previous section, the sharp-interface IBM developed is able to reach high order of convergence. The results obtain using different stencils for the extrapolation function where compared. Using a high-order stencil formulation for the extrapolation function enables reaching the asymptotic region of the convergence order faster (i.e using a coarser mesh) but leads to higher error for coarser meshes. The validation case demonstrated that using a sharp-interface immersed boundary give results that are in correspondence with the results found in the literature both in the case of unsteady 2D flow and 3D steady flow. The implementation in 3D also demonstrated that the method was easily generalized to higher dimensions. We demonstrated that this method can also lead to simplification in the mesh generation process in the case where multiple particles need to be represented.

The sharp-interface method presented in this work has a few limitations.

Two different immersed boundaries cannot be contained in two adjacent elements or in a single cell as it leads to an over-constraint of the system and a singular matrix. There are a few solutions to this problem. The simplest approach would be to prioritize one of the boundaries and only implements the equation related to this boundary in cells that are in conflict. Thus, this is only a limitation of the implementation and is not inherent to the method.

In this work we focused on cases where the immersed boundary are stationary. Even though there are a multiple application for stationary immersed boundaries in cases where the mesh generation can be complex, moving immersed boundaries are also relevant to multiple applications. However, mass conservation poses additional challenges when the immersed body is in motion. We consider that the static sharp-interface approach is highly relevant for numerous engineering application. It's extension to moving geometries be the main topic of future work done on this method.

6. Acknowledgements

The authors would like to acknowledge the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) through the RGPIN-2020-04510 Discovery grant. In particular, Lucka Barbeau would like to acknowledge an NSERC CREATE Simulation Based Engineering Science (SBES) scholarship. The authors would also like to acknowledge technical support and computing time provided by Compute Canada and Calcul Québec.

References

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

[2] W. Kim, H. Choi, Immersed boundary methods for fluid-structure interaction: A review, International Journal of Heat and Fluid Flow 75 (2019) 301–309.

[3] J. Lu, X. Zhu, E. Peters, R. Verzicco, D. Lohse, J. Kuipers, Moving from momentum transfer to heat transfer–a comparative study of an advanced graetz-nusselt problem using immersed boundary methods, Chemical engineering science 198 (2019) 317–333.

[4] A. Pacheco-Vega, J. R. Pacheco, T. Rodić, A general scheme for the boundary conditions in convective and diffusive heat transfer with immersed boundary methods (2007).

[5] X. Gong, Z. Gong, H. Huang, An immersed boundary method for mass transfer across permeable moving interfaces, Journal of Computational Physics 278 (2014) 148–168.

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

[7] J. Freund, R. Stenberg, On weakly imposed boundary conditions for second order problems, in: Proceedings of the Ninth Int. Conf. Finite Elements in Fluids, Venice, 1995, pp. 327–336.

[8] J. Mohd-Yusof, Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries, Center for turbulence research annual research briefs 161 (1) (1997) 317–327.

[9] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. Von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, Journal of computational physics 227 (10) (2008) 4825–4852.

[10] J. Lee, J. Kim, H. Choi, K.-S. Yang, Sources of spurious force oscillations from an immersed boundary method for moving-body problems, Journal of computational physics 230 (7) (2011) 2677–2695.

[11] B. Blais, L. Barbeau, V. Bibeau, S. Gauvin, T. El Geitani, S. Golshan, R. Kamble, G. Mikahori, J. Chaouki, Lethe: An open-source parallel high-order adaptative cfd solver for incompressible flows, SoftwareX 12 (2020) 100579.

[12] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845. `doi:10.1002/fld.3767`.
URL `http://doi.wiley.com/10.1002/fld.3767`

[13] J. H. Seo, R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-mach number flow-induced sound in complex geometries, Journal of computational physics 230 (4) (2011) 1000–1019.

[14] M. Linnick, H. Fasel, A high-order immersed boundary method for unsteady incompressible flow calculations, in: 41st Aerospace Sciences Meeting and Exhibit, 2003, p. 1124.

[15] L. Heltai, F. Costanzo, Variational implementation of immersed finite element methods, Computer Methods in Applied Mechanics and Engineering 229 (2012) 110–127.

[16] F. Ilinca, J.-F. Hétu, A finite element immersed boundary method for fluid flow around rigid objects, International Journal for Numerical Methods in Fluids 65 (7) (2011) 856–875.

[17] F. Ilinca, J.-F. Hétu, Solution of flow around complex-shaped surfaces by an immersed boundary-body conformal enrichment method, International journal for numerical methods in fluids 69 (4) (2012) 824–841.

[18] J.-F. Hétu, F. Ilinca, Immersed boundary finite elements for 3d flow simulations in twin-screw extruders, Computers & Fluids 87 (2013) 2–11.

[19] N. Peller, A. L. Duc, F. Tremblay, M. Manhart, High-order stable interpolations for immersed boundary methods, International Journal for Numerical Methods in Fluids 52 (11) (2006) 1175–1193.

[20] S. Das, A. Panda, N. Deen, J. Kuipers, A sharp-interface immersed boundary method to simulate convective and conjugate heat transfer through highly complex periodic porous structures, Chemical Engineering Science 191 (2018) 1–18.

[21] D. Arndt, W. Bangerth, B. Blais, T. C. Clevenger, M. Fehling, A. V. Grayver, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, R. Rastak, I. Thomas, B. Turcksin, Z. Wang, D. Wells, The `deal.II` library, version 9.2, Journal of Numerical MathematicsIn press (2020). `doi:10.1515/jnma-2019-0064`.
URL `https://dealii.org/deal92-preprint.pdf`

[22] A. De, et al., A robust and efficient sharp interface immersed boundary framework for viscous flow simulations at arbitrary mach number involving complex and moving boundaries, arXiv preprint arXiv:1912.11908 (2019).

[23] J. Yang, F. Stern, Sharp interface immersed-boundary/level-set method for wave–body interactions, Journal of Computational Physics 228 (17) (2009) 6590–6616.

[24] T. E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, in: Advances in applied mechanics, Vol. 28, Elsevier, 1992, pp. 1–44.
URL `http://files/748/WQKE8BY4.pdf`

[25] F. Ilinca, K. R. Yu, B. Blais, The effect of viscosity on free surface flow inside an angularly oscillating rectangular tank, Computers and Fluids (2019). `doi:10.1016/j.compfluid.2019.02.021`.
URL `https://doi.org/10.1016/j.compfluid.2019.02.021`

[26] T. E. Tezduyar, S. Mittal, S. E. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, Computer Methods in Applied Mechanics and Engineering 95 (2) (1992) 221–242. `doi:10.1016/0045-7825(92)90141-6`.

[27] B. Blais, F. Ilinca, Development and validation of a stabilized immersed boundary CFD model for freezing and melting with natural convection, Computers and Fluids (2018). `doi:10.1016/j.compfluid.2018.03.037`.

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

[29] W. L. Oberkampf, C. J. Roy, Verification and validation in scientific computing, Cambridge University Press, 2010.

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

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

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

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

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

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

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

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

[38] B. Blais, M. Lassaigne, C. Goniva, L. Fradette, F. Bertrand, A semi-implicit immersed boundary method and its application to viscous mixing, Computers and Chemical Engineering 85 (2016). `doi:10.1016/j.compchemeng.2015.10.019`.

[39] R. Mittal, A fourier–chebyshev spectral collocation method for simulating flow past spheres and spheroids, International journal for numerical methods in fluids 30 (7) (1999) 921–937.

[40] T. Johnson, V. Patel, Flow past a sphere up to a reynolds number of 300, Journal of Fluid Mechanics 378 (1999) 19–70.

[41] D. Kelly, J. De SR Gago, O. Zienkiewicz, I. Babuska, A posteriori error analysis and adaptive processes in the finite element method: Part i—error analysis, International journal for numerical methods in engineering 19 (11) (1983) 1593–1619.

[42] P. Bagchi, M. Ha, S. Balachandar, Direct numerical simulation of flow and heat transfer from a sphere in a uniform cross-flow, J. Fluids Eng. 123 (2) (2001) 347–358.

[43] S. Marella, S. Krishnan, H. Liu, H. Udaykumar, Sharp interface cartesian grid method i: an easily implemented technique for 3d moving boundary computations, Journal of Computational Physics 210 (1) (2005) 1–31.

[44] R. Clift, J. R. Grace, M. E. Weber, Bubbles, drops, and particles, Courier Corporation, 2005.

[45] H. Zhang, B. Xiong, X. An, C. Ke, G. Wei, Prediction on drag force and heat transfer of spheroids in supercritical water: A pr-dns study, Powder Technology 342 (2019) 99–107.