A parallel and adaptative Nitsche immersed boundary method to simulate viscous mixing

Jeanne Joachim[a], Carole-Anne Daunais[a], Valérie Bibeau[a], Luca Heltai[b], Bruno Blais[a,*]

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

[b] *SISSA, International School for Advanced Studies, Via Bonomea 265, 34136, Trieste, Italy*

Abstract

In this study, we present a parallel immersed boundary strategy that uses Nitsche's method (noted NIB) to weakly impose on a given fluid the boundary conditions associated with a solid of arbitrary shape and motion. Specific details of the software implementation, as done in the software Lethe, are discussed. We verify the NIB method and compare it with other methods in the literature on the well-established test cases of Taylor-Couette flow and von Karman vortex street. Then, we validate the NIB method through simulations of the mixing of fluid in a stirred tank, which is a process central to industries as diverse as polymer manufacturing, food processing, pharmaceutical or chemicals. Simulation results show excellent agreement with experimental data available in the literature for a large range of Reynolds numbers ($Re \in \left[1\ ,\ 2 \times 10^3\right]$), for baffled and unbaffled tanks with a pitched-blade turbine (PBT) impeller. Lastly, the versatility of the NIB method is demonstrated with simulations of a mixing rig with two off-centered impellers with overlapping swept volumes, a case that is either unpractical or impossible to simulate with other largely used techniques. The Software as well as all the files that we used for the simulations are available in the public domain for ease of reproducibility.

Keywords: Computational fluid dynamics ; Immersed boundary method ; Finite Element methods ; Mixing

1. Introduction

Mixing of fluids in stirred tanks is central to many industries, with applications as diverse as polymer manufacturing [1], food processing [2], pharmaceutical or chemicals [3]. In particular, turbulent mixing is used in a wide variety of

*Corresponding authors

Email address: bruno.blais@polymtl.ca (Bruno Blais)

processes, such as blending, chemical reactions, and dispersing immiscible liquids [4]. To improve these processes, one needs access to flow information, which is very challenging to obtain experimentally via non-intrusive methods [3, 5]. Computational fluid dynamics (CFD) has then proven to be an inexpensive, fast and efficient tool to gain insight into the flow patterns [6]. CFD can be used to predict the residence time distribution, the mixing time, the occurrence of dead zones and the power consumption, among many other relevant quantities [7, 8]. The challenge in simulating stirred tanks resides in the presence of a single or multiple rotating impellers within the tank, and the possible presence of static baffles [9]. Indeed, multi-shaft mixers and mono-shaft mixers with stacked impellers are widely used for the mixing of non-Newtonian fluids [10]. Moreover, impellers can have complex geometries depending on the process for which they are designed, such as curved or tilted blades for turbine impellers, and helix or gate for paddle impeller [10].

Mainly two approaches that rely on conformal mesh techniques are used nowadays to simulate the flow generated by a rotating impeller in a baffled tank [8, 11]: Multiple Reference Frame (MRF) method and Sliding-Mesh (SM) (or sliding-grid, SG) strategies. In the MRF method, the fluid domain is divided in multiple mesh blocks: in the inner region that contains the impeller, considered in a rotating reference frame, which provide boundary conditions for the outer region that contains the baffles, considered in a stationary reference frame [12]. The MRF method is used to gain information on the steady state of the flow [8], and has been successfully applied even for multiphase flows [13, 14]. However, the MRF method requires that the flow variables does not change significantly with time or space at the surface between the inner and outer regions [7], so it cannot be used to model multiple impellers with different rotation axes or velocities. In SM strategies, the impeller and baffles are considered on two non-overlapping sub-domains, meshed as separate blocks [15, 7]. The mesh in the block containing the impeller is assumed to rotate with the impeller speed, and the two sub-domains are coupled at the interface, thus providing information on the transient state of the flow [8]. SM strategies are reliable and available in multiple commercial and open source software (such as Open∇FOAM [16]), which explains their wide use in the literature [17, 18, 19]. SM algorithms are an active field of research, with recent updates and improvements displaying excellent results for industrial applications such as stator-rotor-stator turbine [20] or wind turbine [20, 21] flow simulations. However, such algorithms require good quality meshes that can take substantial effort to generate, particularly for configurations in which the impeller have a complex geometry.

To discretize complex immersed boundaries, researchers have developed approaches that rely on non-conformal mesh techniques, such as extensions of SM algorithms [22] or the use of overset grids [23, 24, 25, 26]. Though such techniques enable to use complex geometries and have been used in a wide variety of applications, the handling of the interface between the subdomains is challenging and has an adverse impact on the overall accuracy and cost-efficiency of the method [27]. This is particularly true for configurations in which there are multiple moving subdomains, each rotating with their own velocities, as needed

to simulate common multi-shaft mixers. Furthermore, and like conformal mesh techniques, they do not allow for any overlap between the swept volumes of the subdomains.

To overcome the limitation of the conformal and overset mesh techniques, researchers use immersed boundary (IB) methods to model the moving impellers without generating a conformal mesh of its geometry. The impeller can be represented by discrete or continuous forcing terms that are applied on the fluid at the impeller location in forcing methods [28, 29, 30, 31, 32] and fictious domain method [33]. Alternative approach impose the immersed boundary using interface reconstruction [34] or sharp-edge boundary strategies [35]. These approaches give much more flexibility regarding the impeller geometry and localization in the tank, and allow for the use of dynamic mesh refinement strategies [36]. However, most of them are either only available in closed-source CFD code or in commercial software. Additionnaly, they generally lack the capability to carry out 3D simulations on high-performance computing environment while taking in to account dynamic mesh adaptation and load balancing mechanisms.

In this work, we present a parallel IB strategy that uses Nitsche's method to weakly impose the boundary conditions associated with the impeller [37, 38, 39]. This approach uses the Nitsche formulation to weakly impose Dirichlet boundary condition on rigid bodies. This work builds upon previous work which uses the Nitsche method for interface problems (cut-meshes, chimera meshes) [40], embedded domains [41] and deformable structures [42]. This Nitsche IB (NIB) method is implemented in a high-performance open-source FEM software: Lethe [43, 44]. Lethe has the capacity to carry out parallel simulations with dynamic mesh adaptation and load balancing, enabling the simulation of large systems.

We present our model formulation in Section 2. Then, we discuss the specific details of the software implementation of the NIB method in Section 3. In Section 4, we verify the accuracy of the NIB method using well-established test cases that generate sufficiently complex flows and that possess either analytical or well-documented solutions in the literature. In Section 5, we validate the method on a single impeller tank, with and without baffles, for which experimental and simulation results are available on the literature. Lastly, we showcase the versatility of the NIB method in Section 6 through the simulation of a mixing rig with two impellers, off-centered and with overlapping swept volumes.

2. Model formulation

The open-source CFD software Lethe [44, 43] solves the incompressible Navier-Stokes equations in a domain Ω:

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

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = \nabla p^* + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f} \tag{2}$$

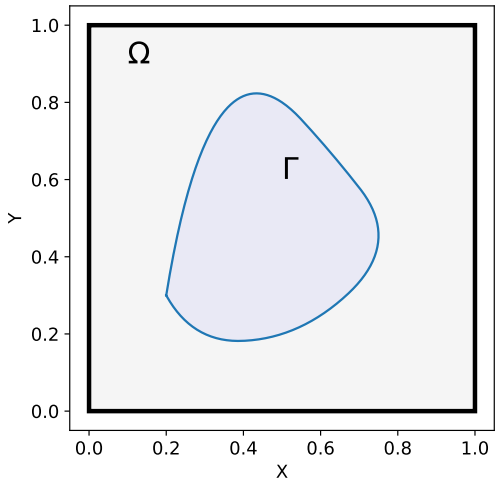


Figure 1: Illustration of a domain Ω within which there is an immersed solid Γ on which a Dirichlet boundary condition is applied.

with

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

with $\boldsymbol{u}$ the fluid velocity, $p^* = \frac{p}{\rho}$ is the pressure p divided by the density ρ, $\boldsymbol{\tau}$ the deviatoric stress tensor, $\boldsymbol{f}$ a source term and $\nu = \frac{\mu}{\rho}$ the kinematic viscosity (with μ the dynamic viscosity).

We are interested in solving Equations (1) and (2) in Ω where a subset Γ of the domain ($\Gamma \in \Omega$) is subjected to a Dirichlet boundary condition ($\boldsymbol{u} = \boldsymbol{u}_\Gamma(\boldsymbol{x}, t) \; \forall \; \boldsymbol{x} \in \Gamma$). This configuration is illustrated in 2D in Figure 1. First, we discuss the stabilized FEM solver used to solve this problem before discussing the specific details of the software implementation of the Nitsche immersed boundary (NIB) in Section 3.

Lethe uses an SUPG/PSPG (streamline-upwind/Petrov-Galerkin and pressure-stabilizing/Petrov-Galerkin) stabilized formulation to solve the incompressible Navier-Stokes equations [43]. This is essentially a Petrov-Galerkin formulation, stabilized by adding two terms to the regular Galerkin formulation of the incompressible Navier-Stokes problem.

To ensure that the no-slip boundary condition is respected on the immersed domain Γ, an additional restriction is added to the Navier-Stokes equations following Nitsche's method. This restriction takes the form of a forcing term which imposes, weakly, that the velocity of the fluid within Γ adheres to the velocity of the rigid body in motion. This restriction is weakly imposed, meaning that the velocity of the rigid body is only satisfied in the $\mathcal{L}^2$ norm.

The resulting weak form is:

$$\int_\Omega \nabla \cdot \boldsymbol{u} q \mathrm{d}\Omega + \underbrace{\sum_K \int_{\Omega_k} \left(\boldsymbol{\mathcal{R}}_{\boldsymbol{u}} \cdot \tau_u \nabla q\right) \mathrm{d}\Omega_k}_{\text{PSPG}} + \underbrace{\int_\Gamma \beta \left(\boldsymbol{u}_\Gamma - \boldsymbol{u}\right) \cdot \tau_u \nabla q \mathrm{d}\Gamma}_{\text{NIB}} = 0 \tag{4}$$

$$\int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{f}\right) \cdot \boldsymbol{v} \mathrm{d}\Omega + \int_\Omega \boldsymbol{\tau} : \nabla \boldsymbol{v} \mathrm{d}\Omega - \int_\Omega p^* \nabla \cdot \boldsymbol{v} \mathrm{d}\Omega + \underbrace{\int_\Gamma \beta \left(\boldsymbol{u}_\Gamma - \boldsymbol{u}\right) \cdot \boldsymbol{v} \mathrm{d}\Gamma}_{\text{NIB}}$$

$$+ \underbrace{\sum_K \int_{\Omega_k} \left(\boldsymbol{\mathcal{R}}_{\boldsymbol{u}}\right) \cdot \left(\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{v}\right) d\Omega_k}_{\text{SUPG}} + \underbrace{\int_\Gamma \beta \left(\boldsymbol{u}_\Gamma - \boldsymbol{u}\right) \cdot \left(\tau_u \boldsymbol{u} \cdot \nabla \boldsymbol{v}\right)}_{\text{NIB}} = 0 \tag{5}$$

where $(q, \boldsymbol{v})$ are the test function for pressure and velocity, respectively, β the Nitsche restriction parameter, and

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{u}} = \frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p^* - \nabla \cdot \boldsymbol{\tau} - \boldsymbol{f} \tag{6}$$

is the strong residual associated with the incompressible Navier-Stokes equations.

In the case of transient problems the stabilization parameters τ_u is defined as [43, 45]:

$$\tau_u = \left[\left(\frac{1}{\Delta t}\right)^2 + \left(\frac{2\left\|\boldsymbol{u}\right\|}{h_{\text{conv}}}\right)^2 + \left(\frac{4\nu}{3h_{\text{diff}}^2}\right)^2\right]^{-1/2} \tag{7}$$

whereas for stationary problems, τ_u is defined as [45]:

$$\tau_u = \left[\left(\frac{2\left\|\boldsymbol{u}\right\|}{h_{\text{conv}}}\right)^2 + \left(\frac{4\nu}{3h_{\text{diff}}^2}\right)^2\right]^{-1/2} \tag{8}$$

where Δt is the time step, h_{conv} and h_{diff} are the size of the element related to the convection transport and diffusion mechanism, respectively [46, 47]. In Lethe, both element sizes (h_{conv} and h_{diff}) are set to h, which is equal to the diameter of a sphere with a volume equivalent to that of the cell [48, 49].

This stabilized method adds three terms to the regular weak form of the Navier-Stokes equations. The first term added is the PSPG stabilization, that appears in Equation (4). This term allows the use of equal-order elements by adding an additional term to the continuity equation which depends on the residual of the strong form of momentum conservation [47]. Essentially, the aim is to relax the Ladyzhenskaya-Babuška-Brezzi (inf-sup) condition [50]. The second term added is the SUPG, that appears in Equation (5). It prevents oscillations in the velocity field by adding an additional term which depends on the residual of the strong form of the momentum conservation [51, 47]. This term is relevant in convection dominated flows, for which there are boundary layers where the velocity solution and its gradient exhibit rapid variation over short length scales [52]. Finally, the third term added is the NIB restriction on the immersed solid domain Γ, that appears in both Equations (4) and (5). The integrals in Equations (4) and (5) occur on a domain Γ for which the mesh does not coincide with the mesh of Ω. Furthermore, Γ may be translated, rotated or deformed

in time. Consequently, a special procedure discussed in Section 3 will be required to carry out the integrals over the immersed domain Γ.

The parameter β in Equations (4) and (5) controls the stiffness of the Nitsche restriction:

$$\beta = \frac{\beta_0}{h^2} \tag{9}$$

where β_0 is a constant. In essence, this scales the Nitsche restriction with the cell in which it lies. Smaller cells receive a stronger restriction. We have found that this approach couples well with adaptative mesh refinement based on the fluid velocity. This generally leads to a stronger Nitsche restriction on the surface of immersed objects, leaving coarser cells and weaker restriction within the immersed body. Although β_0 can take any value, we have found in practice that taking $\beta_0 \in [10\ ,\ 100]$ leads to accurate results for all flows configurations, except for creeping flows.

Equations (4) and (5) are solved implicitely using Newton-Raphson's method [53].

3. Software implementation

Lethe is based on the `deal.II` framework [54], a well-established framework for finite element simulations. In particular, our implementation of the NIB uses the notion of particles to impose the Nitsche restriction. This leverages a lot of the particle-related features introduced in `deal.II 9.2` [55] and optimized in `9.3` [56], and `9.4` [54]. Some of these features are illustrated in the `step-68` and `step-70` tutorials of `deal.II` [57].

The addition of the Nitsche restriction due to the immersed boundary in Equations (4) and (5) leads to integral over the domain Γ. In general, Γ does not coincide with the triangulation of Ω. In particular, Γ may also be moving in an arbitrary fashion. Thus, to apply the restriction, it is necessary to know the fluid velocity $\boldsymbol{u}$ inside the non-matching solid Γ. The implementation presented in this work is generic: it supports any arbitrary solid geometry described by a triangulation (a mesh) in a fluid triangulation that may be unstructured, thus with irregular connectivity. To minimize the memory footprint, both the fluid and the solid triangulation are distributed among the computer cores using `MPI`. Indeed, for complex geometries, the solid triangulation may be a large object and keeping it on every core would severely increase the memory footprint of the solver. Additionally, mesh displacement of such large objects would become prohibitively expensive.

We need an efficient mechanism to carry out integrals over Γ, that requires information interpolated from Ω when both Γ and Ω are distributed over an arbitrarily large number of cores (n_p). Recalling that integrals over Γ are calculated using a discrete number of quadrature points, we treat the quadrature points in Γ as particles using the `ParticleHandler` class of `deal.II` [58]. This class manages particles on top of an existing grid by keeping track of the cells in which the particles reside. Thus, the particles represent the solid domain Γ. These particles can be advected using a defined velocity field, to simulate an arbitrary motion of the geometry. Using the underlying `deal.II` particle framework, the cell of the fluid triangulation in which the particles lie can be readily identified and thus, particles can freely migrate between processes as they switch cell. This approach follows in spirit the ideas presented by Becker *et al.* [59], by associating Lagrangian particles to quadrature points.

The following steps synthesize this approach, which is also illustrated in Figure 2:

1. Discretize Γ using a triangulation which is partitioned and distributed over n_p processes used to carry out the simulation. This generates a solid triangulation. Note that the partitioning of the solid triangulation can be arbitrary. A process may own cells in a solid triangulation that do not intersect the locally owned fluid cells.

2. Compute the locations associated with the quadrature points of the cells of the solid triangulation, and calculate the jacobian of the transformation and the quadrature weights (`JxW`). At this point, the solid triangulation could be discarded and is only kept for visualization purposes. This operation is local to each processor.

3. Locate the cells in which the points reside and insert particles at these location, and attach the `JxW` values to the particles.This step requires communication from some processors to others to ensure the localization of the particles.

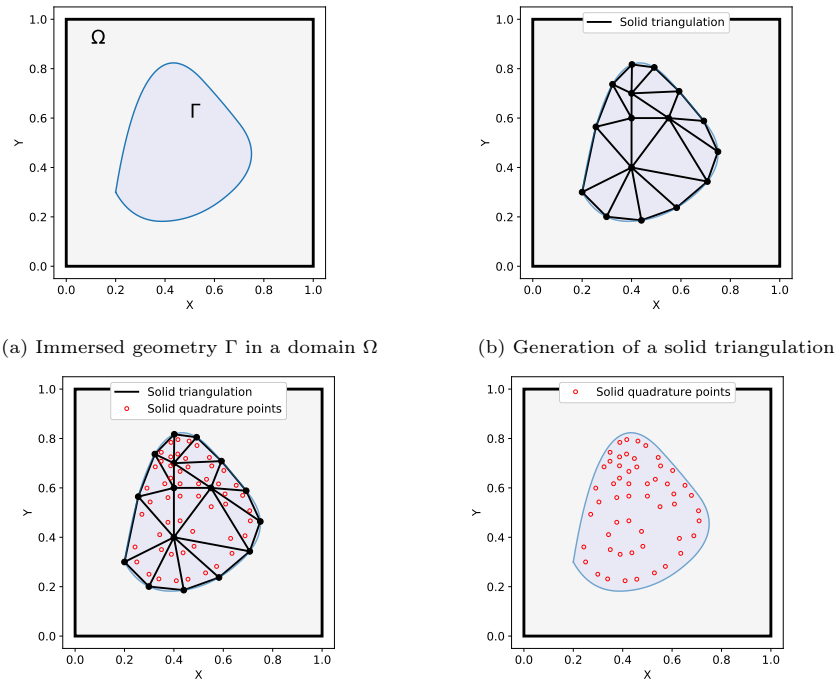
Using the `ParticleHandler` functionalities, particles belonging to a cell can be readily accessed when iterating over the cell and the fluid velocity or other properties can be interpolated to the particle location easily. Consequently, the application of the Nitsche restriction is fully parallel. The cost of assembling the Nitsche restriction is proportional to the number of particles which is, in turn, proportional to the number of cells in the solid triangulation. For large solid meshes, the assembly of the Nitsche's restriction can be a significant portion of the total assembly time. As such, it can be necessary to take into account the number of particles per cell as a load-balancing criterion when adapting the mesh and redistributing the cells.

This method provides many features which are available in various immersed boundary methods under the umbrella of a single unifying framework. The method allows for arbitary motion of the solid domain Γ (space and time-dependent) while ensuring mass conservation for the fluid phase since the Navier-Stokes equations are solved over the entire domain. It also has dynamic mesh adaptation and dynamic load balancing, ensuring that the computational load of the simulation is kept constant even if the geometry is significantly displaced. Finally, it preserves the order of convergence of the underlying FEM scheme, except in the cells which are at the solid-fluid interface.

4. Verification

We verify the accuracy of the Nitsche immersed boundary method implementation using test cases that generate sufficiently complex flows and with well-documented solutions in the literature. The first case, the Taylor-Couette flow, is used to establish the order of convergence of our immersed boundary method for both the velocity profiles and the torque in the case where the immersed solid is rigid. The Taylor-Green decaying vortex, assess the order of convergence of the scheme for a time-dependent flow. The third case, the von Karman vortex street behind a cylinder, is used to assess the capacity of the immersed boundary to predict the dynamic of transient flow.

For all cases which contain mesh adaptation, a Kelly error estimator [60] is used on the velocity field to identify cells which must be refined. Then, a fraction ($\epsilon \in]0\ ,\ 1]$) of the cells in which this error estimator is the largest are refined. Although this error estimator is not optimal for the combination of the Nitsche immersed boundary and the incompressible Navier-Stokes equations, we have found that a relatively robust and fast way to estimate the cells which should be refined.



(a) Immersed geometry Γ in a domain Ω

(b) Generation of a solid triangulation

(c) Insertion of quadrature points

(d) Solid triangulation is discarded

Figure 2: Process for the generation of the NIB. The immersed geometry Γ is first discretized to form a triangulation. The Gauss points of the elements of this triangulation are used to generate particles. The triangulation associated with Γ is discarded and the calculation is made with the particles associated with the Gauss points of the immersed triangulation.

4.1. Laminar Taylor-Couette Flow

The Taylor-Couette flow is the 2D flow that occurs between two-concentric cylinders when the inner, outer or both cylinders are rotating. Assuming that the flow is laminar, the azimuthal velocity profile between the two cylinders is [61]:

$$u_\theta(r) = \Omega_i \kappa R_o \frac{\left(\frac{R_o}{r} - \frac{r}{R_o}\right)}{\left(\frac{1}{\kappa} - \kappa\right)} \tag{10}$$

where Ω_i is the angular velocity of the inner cylinder and κ ($\kappa = \frac{R_i}{R_o}$) the ratio of the radius of the inner cylinder (R_i) to the outer cylinder (R_o). From this velocity profile, the z-component of the torque acting on the inner cylinder can be directly calculated [61]:

$$T_z = 4\pi\mu\Omega_i R_i^2 L \left(\frac{\kappa^2}{1-\kappa^2}\right) \tag{11}$$

with μ the dynamic viscosity and L the length of the cylinder, which we consider to be unitary since this case is simulated in 2D. At steady-state, the reciprocal of this torque is applied on the outer cylinder, ensuring conservation of angular momentum for the global system.

This test is highly relevant to verify an immersed boundary, because:

- the geometry is not aligned with the background mesh;
- the analytical solution cannot be represented exactly by the shape functions of the underlying FEM scheme;
- the velocity profile is fully two-dimensional;
- an analytical solution is known not only for the azimuthal velocity, but also for the torque acting on the cylinder.

We consider the Taylor-Couette problem in 2D with two co-axial cylinders of radii $R_i = 0.25\,\mathrm{m}$ and $R_o = 1\,\mathrm{m}$, with the inner cylinder rotating. The background mesh is a circular domain of radius 1 m and centered at the point $(0, 0)$, which is also the center of rotation. Only the inner cylinder is discretized using the immersed boundary method.

We carry an order of convergence analysis by monitoring the decrease of the $\mathcal{L}^2$ norm of the error on the velocity with the mesh size. The $\mathcal{L}^2$ norm of the error on the velocity is defined as :

$$\|e_{\boldsymbol{u}}\| = \sqrt{\int_\Omega \|\boldsymbol{u}_\mathrm{h} - \boldsymbol{u}_\mathrm{a}\|^2 \,\mathrm{d}\Omega} \tag{12}$$

where $\boldsymbol{u}_\mathrm{h}$ is the numerical solution and $\boldsymbol{u}_\mathrm{a}$ the analytical solution.

The graph in Figure 3 shows the variation of $\|e_{\boldsymbol{u}}\|$ with respect to the square root of the number of cells ($N_\mathrm{cells}^{1/2}$). It shows that, for a uniform mesh (black dots), the Nitsche IB reduces the order of the underlying GLS scheme to one instead of the expected two. This is due to the poor representation of the curved domain by the Nitsche IB. Similar results were noted in continuous forcing methods by Blais *et al.* [36]. Using adaptative mesh refinement on the same initial refinement than the

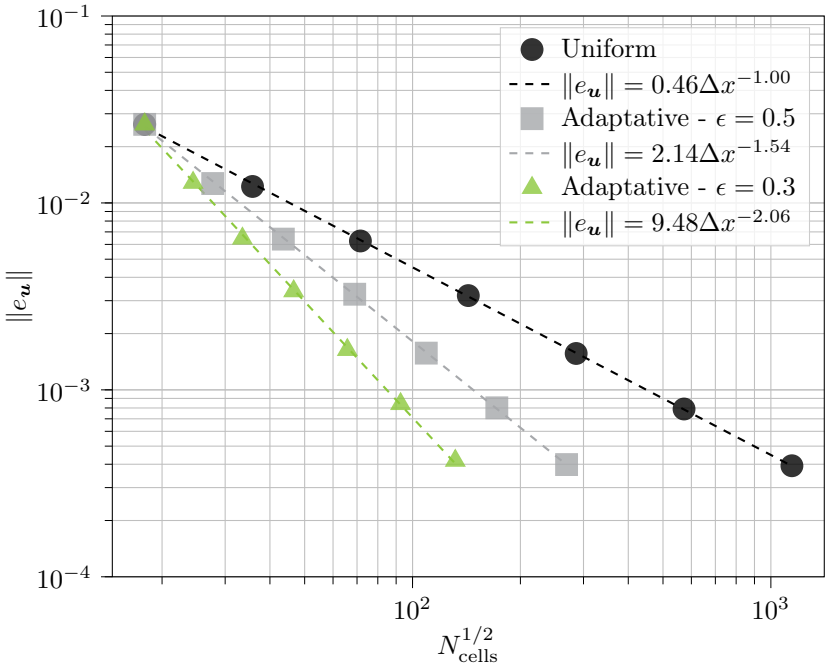


Figure 3: $\mathcal{L}^2$ norm of the error on the velocity $\boldsymbol{u}$ as a function of the number of cells ($N_{\text{cells}}^{1/2} \propto \Delta x$). Regressions show the apparent order of convergence. With uniform mesh refinement, the scheme degrades to first order. With adaptative mesh refinement, an apparent second order of convergence is recovered for a sufficiently low fraction of adaptated cell ($\epsilon = 0.3$).

uniform mesh, the apparent order of convergence becomes dependent on the fraction of cells refined by the adaptative mesh refinement (ϵ), as shown on the graph for $\epsilon = 0.5$ (gray squares) and $\epsilon = 0.3$ (green triangles). A sufficiently low fraction of adaptated cell ($\epsilon = 0.3$) can allow the scheme to recover an apparent second order of convergence.

Likewise, the $\mathcal{L}^2$ norm of the error on the torque is defined as :

$$\|e_{\boldsymbol{T}_z}\| = \sqrt{\int_\Omega \|\boldsymbol{T}_{z\text{h}} - \boldsymbol{T}_{z\text{a}}\|^2 \, \mathrm{d}\Omega} \tag{13}$$

where $\boldsymbol{T}_{z\text{h}}$ is the numerical solution and $\boldsymbol{T}_{z\text{a}}$ the analytical solution.

The graph in Figure 4 shows the variation of $\|e_{\boldsymbol{T}_z}\|$ with respect to the square root of the number of cells ($N_{\text{cells}}^{1/2}$). Results obtained on the outer cylinder (black dots) and on the inner cylinder (square gray) are displayed for a uniform mesh refinement. It shows that first-order convergence is recovered on the torque for both cylinders, even if it is quantity-derived from the velocity field.

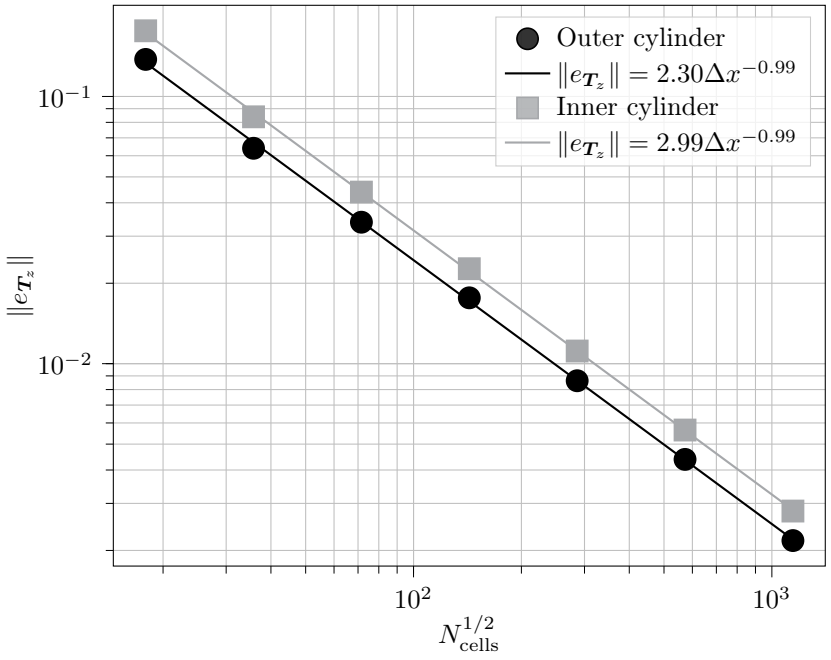


Figure 4: $\mathcal{L}^2$ norm of the error on the torque $\boldsymbol{T}_z$ induced by the flow on the inner and outer cylinder as a function of the number of cells with uniform mesh refinement ($N_{\text{cells}}^{1/2} \propto \Delta x$). Regressions show the apparent order of convergence. First-order convergence is recovered on the torque for both cylinders.

4.2. Taylor-Green decaying vortex

An alternative verification test-case, the Taylor-Green decaying vortex, was proposed by Kang and Hassan [62] to assess the order of convergence for immersed boundary methods on a more complex transient flow. In this simulation, a rigid circle with a radius of $L/2$ is embedded at the center of a square domain $[-L, -L] \times [L, L]$. The velocity $\boldsymbol{u} = [u, v]^T$ and pressure p follow the given analytical solution:

$$u = -u_0 \cos\left(\frac{\pi}{L}x\right) \sin\left(\frac{\pi}{L}y\right) \exp\left(-2\nu \left(\frac{\pi}{L}\right)^2 t\right) \tag{14}$$

$$v = u_0 \sin\left(\frac{\pi}{L}x\right) \cos\left(\frac{\pi}{L}y\right) \exp\left(-2\nu \left(\frac{\pi}{L}\right)^2 t\right) \tag{15}$$

$$p = p_0 - \frac{u_0^2}{4}\left[\cos\left(\frac{2\pi}{L}x\right) + \cos\left(\frac{2\pi}{L}y\right)\right] \exp\left(-4\left(\frac{\pi}{L}\right)^2 \nu t\right) \tag{16}$$

with L the dimension of the domain, ν the kinematic viscosity, and u_0 and p_0 constant values, here set to $u_0 = 1$ and $p_0 = 0$ without loss of generality for the verification.

The initial condition is obtained by setting $t = 0$ in (14), (15) and (16). Within the rigid circle, the analytical solution for the velocity is imposed using the Nitsche immersed boundary. Simulations are carried out using a BDF2 time-stepping scheme with a sufficiently low-time step value to ensure that the results are time-step independent. Figure 5 shows the evolution of the $\mathcal{L}^2$ norm of the error for the case without the immersed boundary ($\beta = 0$) and for two values of β which are consistent with common usage of the method. In this case, the mesh refinement applied is uniform.

It can clearly be seen that, for this test case, the Nitsche immersed boundary does not affect the underlying order of convergence of the scheme compared to the results obtained for the Taylor-Couette flow. This is because the analytical solution for this case has a continuous velocity gradient at the interface between the rigid disk and the fluid zone, contrarily to the Taylor-Couette case in which this gradient is discontinuous. Consequently, this exhibits the fact that the reduction in order of convergence observed for the Taylor-Couette case is caused by the incapacity of the NIB to model a discontinuous velocity gradient at the interface between solid and fluid.

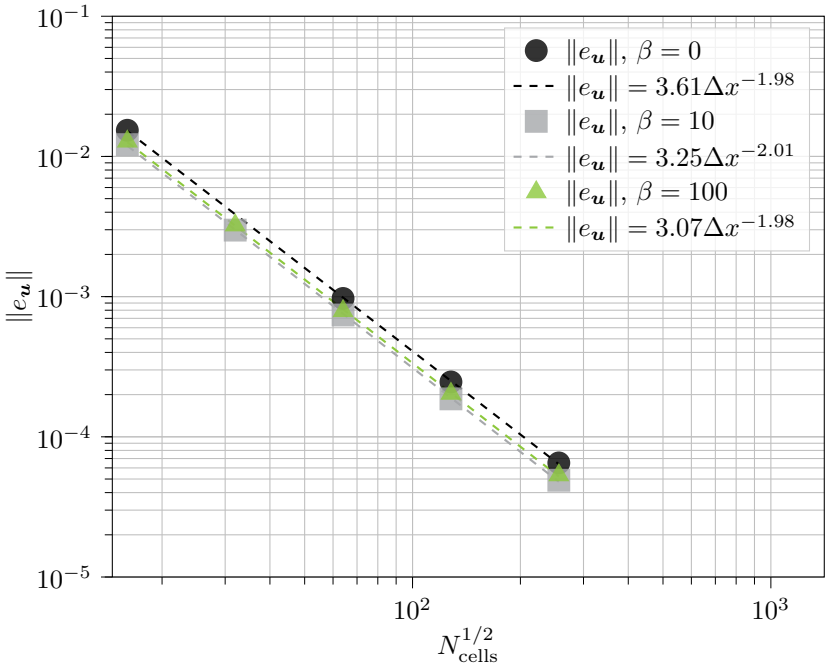


Figure 5: $\mathcal{L}^2$ norm of the error on the velocity $\boldsymbol{u}$ as a function of the number of cells ($N_{\text{cells}}^{1/2} \propto \Delta x$). Regressions show that for the verification test case proposed by Kang et al., the proposed Nitsche immersed boundary preserves the underlying second-order of the scheme with uniform mesh refinement. This is because the case defined by Kang has a continuous velocity gradient at the interface between solid and fluid.

4.3. Von Karman Vortex Street

The unsteady flow past a cylinder has been the topic of many experimental and numerical investigations due to the complexity of the hydrodynamics in its unsteady oscillating wake [63]. This flow is an interesting test case because of the intrinsically periodic and complex phenomenological behavior within its vortex street. It has been used by numerous authors [33, 29, 28, 64, 30, 31, 65, 66, 36] to verify the accuracy and the robustness of their implementation of various kind of immersed boundaries. By measuring the perpendicular (lift) and parallel (drag) forces acting on a cylinder, as well as the shedding frequency, an immersed boundary model can be validated on a complex time-dependent flow.

This case can be studied with two strategies. The first is to simulate the flow around a static cylinder. This problem can be simulated with a conformal grid, since the cylinder remains static. The static setup considered in the present work is illustrated in Figure 6. In this setup, the fluid inlet velocity is constant ($\boldsymbol{u} = [u_\infty, 0]^T$), slip boundary conditions are applied at the top and the bottom and a zero traction boundary condition is imposed at the right.

The second approach is to simulate the flow around the cylinder moving at a constant velocity (in this case, u_∞). This approach is a better assessment of the accuracy and the robustness of an immersed boundary method, even if both approaches should lead to the same forces acting on the cylinder, and the same shedding frequency. Indeed, when the rigid object is moving, computational cells transition from fluid to rigid solid throughout the simulation, which is a more realistic usage of the immersed boundary. The setup for the moving problem is identical to the static one (in Figure 6), except that the length of the domain is multiplied by 8 ($L = 256$), in order to allow sufficient time for the von Karman instability to develop.

The Reynolds number is defined as :

$$Re = \frac{\rho u_\infty D}{\mu} \tag{17}$$

With ρ the density of the fluid, u_∞ the upstream velocity (or the velocity of the cylinder in the moving case) and D the diameter of the cylinder. In this dimensionless problem, we set $\rho = 1$, $u_\infty = 1$ and $D = 1$. To assess the precision of the proposed immersed boundary method, the static and moving cylinder cases were considered at $Re = 200$ ($\mu = 5 \times 10^{-3}$), which is sufficiently high to allow for vortex shedding at a constant frequency h. This frequency can be related to u_∞ and D, via the Strouhal number:

$$St = \frac{hD}{u_\infty} \tag{18}$$

Using the x and y component of the force acting on the cylinder ($F_{c,x}$ and $F_{c,y}$ respectively), both the drag (C_D) and lift (C_L) coefficients can be calculated :

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

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

The Strouhal number was calculated by obtaining the frequency of the shedding via a fast Fourier transform (FFT) of the lift coefficient. The time interval chosen for the FFT analysis was a subset of the simulation, within which the shedding frequency

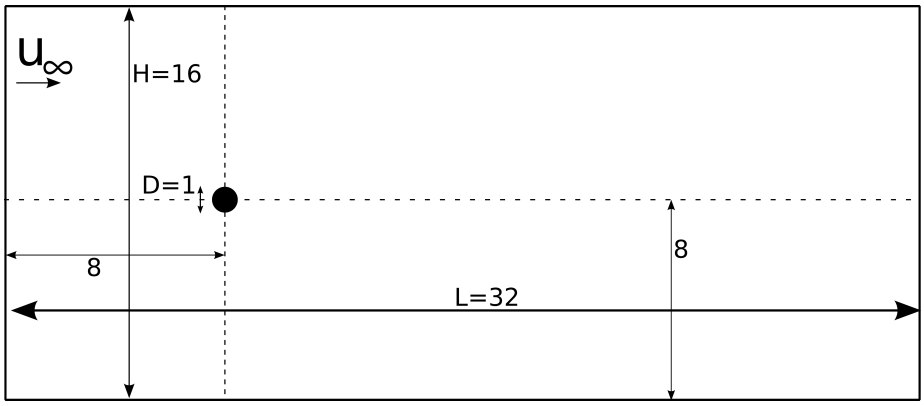


Figure 6: Von Karman vortex past a static cylinder: geometrical characteristics, adapted from Blais et al. [36].

was constant. The simulation was carried out using a time-step of $\Delta t = 0.005\,\mathrm{s}$ and a second-order BDF scheme (BDF2) [67]. This combination lead to time-step independent results. Dynamic mesh adaptation was used to control the mesh resolution. Results are presented for simulations in which the finest cell size allowed was $\Delta x_c = \frac{D}{64}$. To allow for a full comparison, simulations were also carried out using a very fine conformal mesh for which the results were found to be fully independent of the mesh.

The evolution of C_D and C_L is plotted for the static cylinder and with a converged conformal grid in Figure 7, and for the moving cylinder with dynamic mesh adaptation in Figure 8. For comparison purposes, Figure 8 also shows the envelope (range of variation) for C_D and C_L in the case of a static cylinder (extracted from Figure 7), for $t > 125\,\mathrm{s}$. The drag coefficient initially decreases until the onset of the von Karman instability. Then, the drag coefficient increases and the lift coefficient value is no longer zero. Once the flow becomes fully periodic, the drag and the lift coefficient oscillate around an average value, which is 0 for the lift coefficient. A comparison between Figure 7 and 8 also shows that the IB method introduces artificial oscillations on the drag and lift forces, though not clearly noticeable in this figure.

Figure 9 presents the amplitude of the FFT spectrum of the lift coefficient for the moving cylinder case and compares it to the results obtained with a conformal mesh. Here, we clearly see that the Nitsche immersed boundary (NIB) method introduces high-frequency noise in the lift coefficient. This is a consequence of the motion of the immersed solid which changes dynamically the cells in which the Nitsche restriction is applied.

Table 1 compares the drag coefficient (C_D) and its fluctuation, the lift coefficient (C_L) and the Strouhal number for the NIB method (static and moving cylinder), with results obtained with a conformal mesh and extracted from the literature. We note that a very good agreement between the NIB method and the other methods, both in the static and moving cylinder cases. The variability of the data gathered in the literature highlights the sensitivity of this test case to mesh resolution, temporal accuracy and the accuracy of the representation of the boundary. In particular, we note that the NIB method significantly outperforms the continuous forcing method

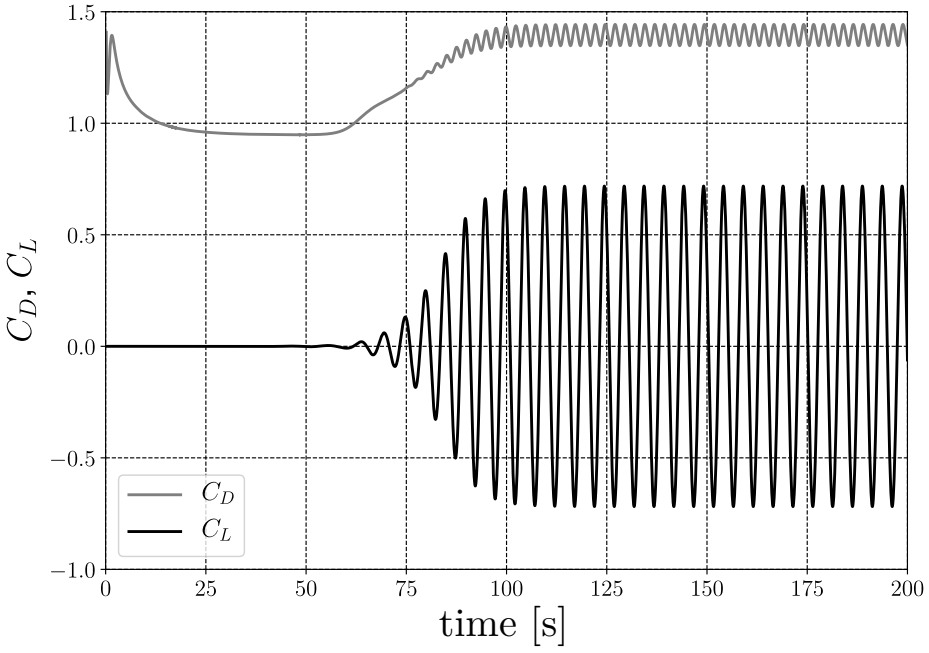


Figure 7: Drag and lift coefficients for the flow past a static cylinder simulated using a conformal mesh.

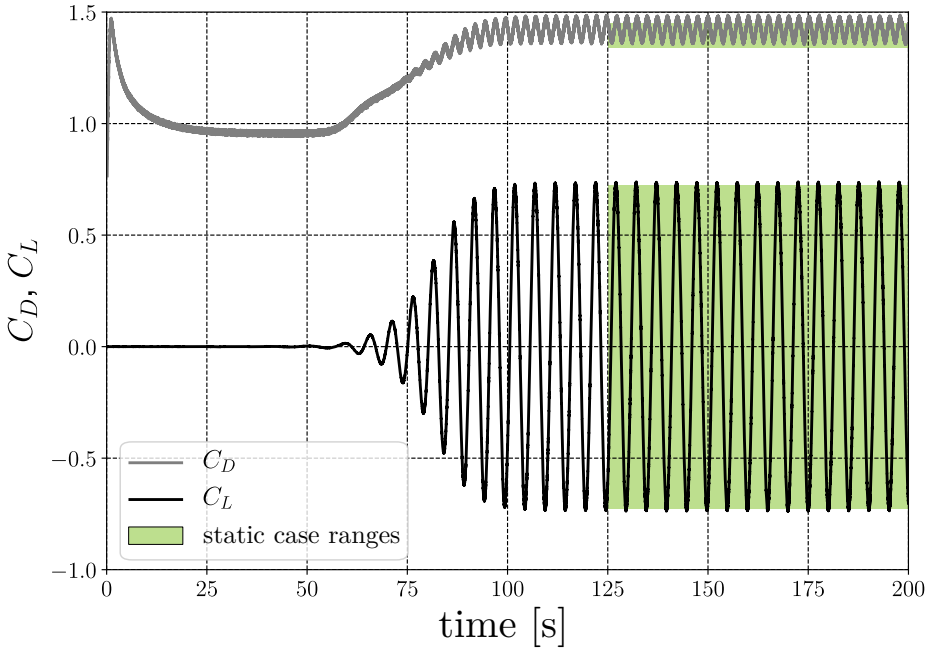


Figure 8: Drag and lift coefficients for the flow past a moving cylinder using dynamic mesh adaptation. The green area gives the envelope (range of variation) for C_D and C_L in the case of a static cylinder (extracted from Figure 7), for $t > 125\,\mathrm{s}$.

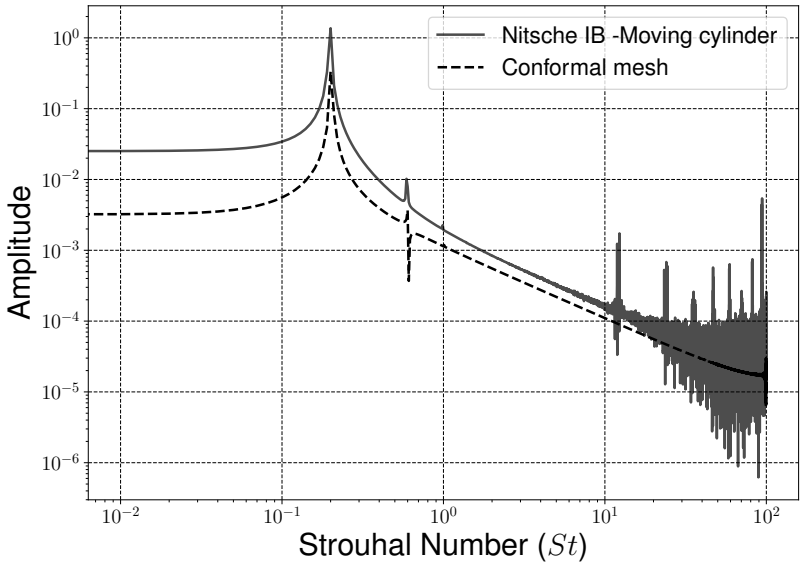


Figure 9: Amplitude of the FFT spectrum of the lift coefficient for the moving cylinder and the static conformal mesh cases once the instabilities were fully developed.

introduced by Blais *et al.* for the moving cylinder case [36]. This is largely due to the fully implicit formulation used in the NIB method for the velocity, the pressure and the Nitsche restriction. Thus, the no-slip boundary condition are better imposed on the cylinder as it moves from cell to cell and captures accurately the lift force on the cylinder.

From these results, it can be concluded that the proposed NIB method reproduces with very high accuracy the von Karman vortex street and is therefore a valid approach for unsteady flow in which the rigid body modeled by the NIB is moving.

Study	$\mathbf{C_D}$	$\mathbf{C_L}$	**St**
NIB (Lethe) - Static cylinder	1.42 ± 0.05	± 0.74	0.200
NIB (Lethe) - Moving cylinder	1.42 ± 0.07	± 0.74	0.200
Conformal mesh (Lethe)	1.40 ± 0.05	± 0.72	0.200
Blais *et al.* [36] - Moving cylinder	1.35 ± 0.1	± 0.51	0.200
Braza *et al.* [63]	1.40 ± 0.05	± 0.75	-
Choi *et al.* [28]	1.36 ± 0.048	± 0.64	0.191
Wright *et al.* [64]	1.33 ± 0.04	± 0.68	0.196
Bergmann *et al.* [30]	1.35	-	0.198
Russel and Wang *et al.* [31]	1.29 ± 0.022	± 0.50	0.195
Henderson *et al.* [65]	1.341	-	0.197
He *et al.* [66]	1.3560	-	0.198
Bhalla *et al.* [29]	1.39	-	0.200

Table 1: Comparison of the NIB (Nitsche Immersed Boundary) method results with conformal mesh results and literature data, for C_D, C_L and St

5. Validation

In this section, we will use the NIB method to simulate a mixing rig with a simple impeller, with and without baffles in the tank. Both of these cases are particularly appropriate to validate the model, as experimental and numerical (using a sliding mesh modeling technique) results are available in Blais *et al.* article [36].

Firstly, we present the methodology, with the problem geometry and dimensionless numbers of interest. Then, the sensitivity of the Nitsche immersed boundary method with regards to spacial and temporal discretization is assessed on the unbaffled tank case. Lastly, results for varying Re on both cases are compared with experimental and numerical data available in the literature.

5.1. Methodology

The mixing rig is identical to the one used in previous experiments and simulations [36]. The impeller is a Pitched-Blade Turbine (PBT45) axial impeller, which generates a flow pattern that amplifies axial circulation within the vessel [9, 8]. The tank is a simple cylinder, with its revolution axis matching the impeller rotation axis. Four baffles can be installed, distributed evenly near the edge of outer cylinder of the rig. The baffles are larger in the direction of the tank diameter, and they are closer to

the flat tank flat-bottom than the impeller. All dimensions are indicated explicitly in Figure 10, and their values and relationships with one another are given in Table 2.

Symbol	Description	Value
D_t	Tank diameter	$0.365\,\mathrm{m}$
H	Height of the fluid domain (liquid level)	D_t
C_i	Impeller off-bottom clearance	$D_\mathrm{t}/4$
D_i	Impeller diameter	$D_\mathrm{t}/3$
W_i	Impeller blade width	$D_\mathrm{i}/5$
T_i	Impeller blade thickness	$W_\mathrm{i}/10$
θ	Tilt angle of the impeller blades	$45°$
D_s	Shaft diameter	$W_\mathrm{i} \times \cos(\theta)$
D_h	Hub diameter	$1.4 \times D_\mathrm{s}$
H_h	Hub height	$W_\mathrm{i} \times \sin(\theta)$
C_b	Baffle off-bottom clearance	$C_\mathrm{i}/3$
S_b	Baffle minimal spacing with the outer cylinder	$19/20 \times D_\mathrm{t}$
W_b	Baffle width	$D_\mathrm{t}/10$
T_b	Baffle thickness	$W_\mathrm{b}/3$

Table 2: Relationships between the different dimensions of the fluid domain and the immersed solids (impeller and baffles), as shown in Figure 10

We compare simulation results with the experimental results obtained in [36] by establishing the dependency of the power number N_p on the Reynolds number Re through the power curve. The dimensionless numbers of interest on the context of mixing are [9]:

$$N_\mathrm{p} = \frac{2\pi T}{\rho N^2 D_\mathrm{i}^5} \tag{21}$$

$$Re = \frac{D_\mathrm{i}^2 N}{\nu} \tag{22}$$

with T the torque on the impeller at steady state (in $\mathrm{N\,m}$), ρ the volumetric mass density (in $\mathrm{kg\,m^{-1}}$), N the impeller angular velocity (in Hz, or $\mathrm{s^{-1}}$), D_i the impeller diameter (in m) and ν the fluid kinematic viscosity (in $\mathrm{m^2/s}$).

In the simulations, we use a Newtonian fluid with a unitary volumetric mass density ($\rho = 1\,\mathrm{kg\,m^{-1}}$). We vary its kinematic viscosity ν to study different Re values, using Equation 22. The impeller speed is constant, at $N = 16\,\mathrm{rad\,s^{-1}}$, so that the tangential velocity at the blade's tip $V_\mathrm{t} \approx 1\,\mathrm{m\,s^{-1}}$. The boundary conditions on the tank are set to no-slip on the hull and bottom wall, and slip on the upper wall to mimic the liquid's free surface. A zero traction boundary condition on the surface could be used since free surfaces have constant pressure, but the slip boundary condition is generally significantly more numerically robust. The no-slip condition on the Nitsche solids (impeller and baffles) is imposed through the Nitsche restriction (see Equation 5).

We simulate the mixing rig for $0.2\,\mathrm{s}$, corresponding to 3.28 revolutions of the impeller. This simulation time ensures that the torque value reaches a pseudo-steady state while keeping the computational cost reasonable. The stabilized torque value T, as used in Equation (21), is computed as the mean of the instantaneous torque from $0.1\,\mathrm{s}$ to $0.2\,\mathrm{s}$ (that is plotted for the final simulation parameters at the end of the next

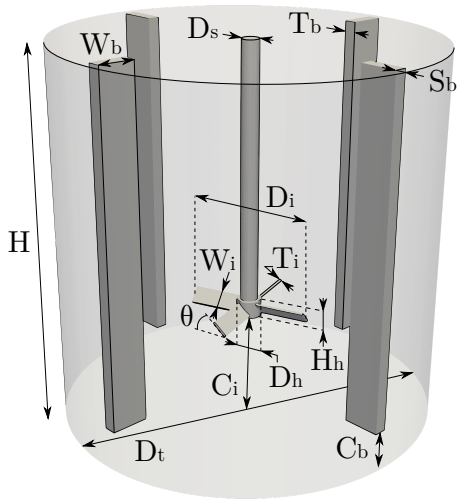


Figure 10: Dimensions necessary to define the fluid domain (liquid inside the tank) and the immersed solids (impeller and baffles) of the mixing rig.

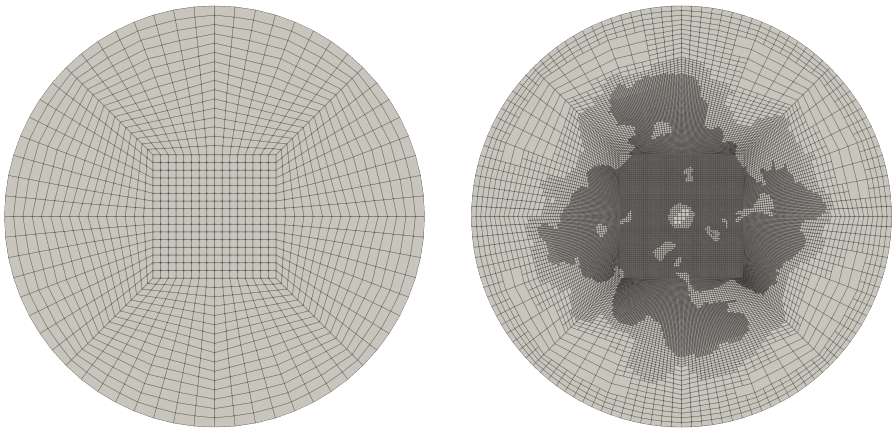
section, in Figure 16).

5.2. Simulation parameters

In order to set up the simulations, we carry out a sensitivity analysis of the NIB method to the spatial (fluid and solid mesh) and temporal discretizations, for the unbaffled mixing rig. This sensitivity analysis is crucial to properly choose the parameters for the upcoming simulations over a large Re range. The influence of parameters specific to the NIB method (namely the number of particles used to represent the solid mesh, and the restriction parameters) is discussed in this section.

5.2.1. Fluid and solid mesh parameters

We characterize the level of refinement of the fluid mesh by a parameter noted p_{fluid}. The fluid domain is a simple cylinder, generated through the `deal.II` grid generator `cylinder` [68], with hexahedral elements. Figure 11 shows an example of the mesh obtained at the beginning (Figure 11a) and at the end (Figure 11b) of the simulation, on the plane that passes through the middle of the impeller blades. For a uniform mesh, p_{fluid} corresponds directly to the initial refinement, and is constant throughout the simulation. For an adaptative mesh, the refinement of the cell is determined through a Kelly error estimator on the velocity: every 5 iterations, the mesh can be locally coarsened or refined depending on the velocity variation in the cell (Figure 11b). The initial refinement is set to 4 (Figure 11a) and a minimum refinement is set to 3. Then, p_{fluid} corresponds to the maximum refinement level allowed. Table 3 indicates the evolution of the number of cells (N_{cells}) with p_{fluid} for a uniform and an adaptative fluid mesh at the end of the simulation. In the case of adaptative mesh, N_{cells} at the beginning of the simulation corresponds to N_{cells} for a



(a) $t = 0\,\mathrm{s}$, initial coarse mesh (b) $t = 0.2\,\mathrm{s}$, refined mesh

Figure 11: Slice of the fluid mesh at the beginning and at the end of the simulation, when mesh adaptation is used ($p_{\mathrm{fluid}} = 7$), for $Re \approx 1200$ ($\nu = 2 \times 10^{-4}\,\mathrm{m}^2/\mathrm{s}$).

uniform mesh when $p_{\mathrm{fluid}} = 4$ (Figure 11a). With a uniform mesh, $p_{\mathrm{fluid}} = 8$ was not considered to limit the memory usage. It is then emphasized that adaptative mesh enables to have a finer fluid mesh close to the impeller while limiting N_{cells}, and thus decreasing the computational costs.

p_{fluid}	4	5	6	7	8
$\mathcal{O}(N_{\mathrm{cells}})$ uniform mesh	5×10^4	3×10^5	3×10^6	2×10^7	-
$\mathcal{O}(N_{\mathrm{cells}})$ adaptative mesh	3×10^4	1×10^5	3×10^5	2×10^6	2×10^7

Table 3: Orders of the number of cells ($\mathcal{O}(N_{\mathrm{cells}})$) obtained for different fluid mesh parameter (p_{fluid}) values.

We characterize the level of refinement of the solid mesh by a parameter noted p_{solid}, directly linked to the minimal number of elements on the blade thickness. The blade thickness is the smallest dimension of the impeller (see Table 2). The same parameter is used for the baffles mesh. Tetrahedral elements are used to better discretize the geometry, with the open source mesh generator Gmsh [69]. The files used to generate the mesh are available at the Lethe utilities repository [70]. Table 4 indicates the number of Nitsche particles used to represent each solid mesh (as defined in Section 3 and illustrated in Figure 2), and associated with the different values of p_{solid} studied, both for the impeller and the four baffles.

p_{solid}	0.5	1	2	3
$\mathcal{O}(n_{\mathrm{part}})$ impeller	4×10^3	2×10^4	1×10^5	3×10^5
$\mathcal{O}(n_{\mathrm{part}})$ four baffles	2×10^4	1×10^5	8×10^6	-

Table 4: Orders of the number of Nitsche particles ($\mathcal{O}(n_{\mathrm{part}})$) obtained for different solid mesh parameter (p_{solid}) values.

Loss of Nitsche particles can occur when the solid moves through multiple fluid cells during an iteration. The time step (Δt) should be taken to ensure that the magnitude of the particle displacement is not larger than a cell. To avoid unnecessary fluid iterations and thus to maintain a reasonable computational cost, we define sub-iterations for the Nitsche solid only. For all the simulations of this mixing rig, the Nitsche solid undergoes 20 sub-iterations per time step (Δt).

5.2.2. Sensitivity analysis for the unbaffled mixing rig simulations

We carry out sensitivity analyses on the simulation of the unbaffled mixing rig for the following parameters: fluid mesh parameter (p_{fluid}), solid mesh parameter (p_{solid}) for the impeller and time step (Δt). For all simulations, the Nitsche restriction parameter (β_0 in Equation (9)) is set to $\beta_0 = 10$. We indicate computation times for comparison purposes between different parameter values. However, note that the effective computation times are highly dependent on the computational setup available. We use 3 to 4 nodes on two Digital Alliance of Canada clusters (Niagara and Narval) for this study.

Fluid mesh. The power number (N_{p}) evolution with a fluid mesh parameter p_{fluid} is displayed in Figure 12, for $Re = 200$ and $\Delta t = 1 \times 10^{-3}$ s. For $p_{\text{fluid}} \leq 7$, N_{p} decreases as p_{fluid} increases. Then, N_{p} varies by less than 0.03 % between $p_{\text{fluid}} = 7$ and $p_{\text{fluid}} = 8$, which indicated that a convergence plateau is reached. When compared to the experimental values for $Re = 200$ [36], the computed N_{p} value is inside the experimental uncertainty range for $p_{\text{fluid}} \geq 6$. Lastly, results obtained with an adaptative fluid mesh follow closely those obtained for a uniform mesh. This highlights that adaptative mesh is a valid strategy to lower the computational cost (Table 3), without loss in accuracy.

Reaching convergence on the fluid mesh is more demanding in the case of turbulent flow. Figure 13 presents simulation results for $Re = 2000$, with $\Delta t = 1 \times 10^{-3}$ s and $p_{\text{solid}} = 2$. Though there is no experimental results for the unbaffled mixing rig to validate N_{p} values obtained, the N_{p} curve shows an inflection at $p_{\text{fluid}} \geq 7$. This corresponds to the beginning of a convergence with regards to p_{fluid}, though it should be noted that the plateau value is not reached yet. The computational time is displayed on the secondary axis for comparison purposes. It appears that $p_{\text{fluid}} = 7$ is the best compromise between convergence at high Re and computational costs, as this last increases exponentially with n_{dof} (Table 3).

Impeller mesh. Figure 14 displays N_{p} results for different p_{solid} of the impeller mesh, obtained for $Re = 200$, $p_{\text{fluid}} = 7$ and $\Delta t = 1 \times 10^{-3}$ s. First, we can confirm the closeness of results previously shown in Figure 12 for $p_{\text{solid}} = 1$ and $p_{\text{solid}} = 2$. Indeed, N_{p} appears less sensitive to p_{solid} than to p_{fluid}, and results are within the experimental uncertainty range for $p_{\text{solid}} \leq 0.5$. We reach numerical convergence for $p_{\text{solid}} \leq 1$. Computational times are displayed on the secondary axis. The high computational time associated with $p_{\text{solid}} = 0.5$ is due to convergence issue of the linear iterative solver in the generalized minimal residual method (GMRES), which then requires a higher incomplete LU (ILU) preconditioning fill level. This indicated that the linear system is poorly conditioned when there are fluid cells within the solid object without Nitsche restriction. From $p_{\text{solid}} = 1$, the computational time increases with the number of particles (Table 4), and thus $p_{\text{solid}} = 1$ is the best compromise between convergence and computational costs.

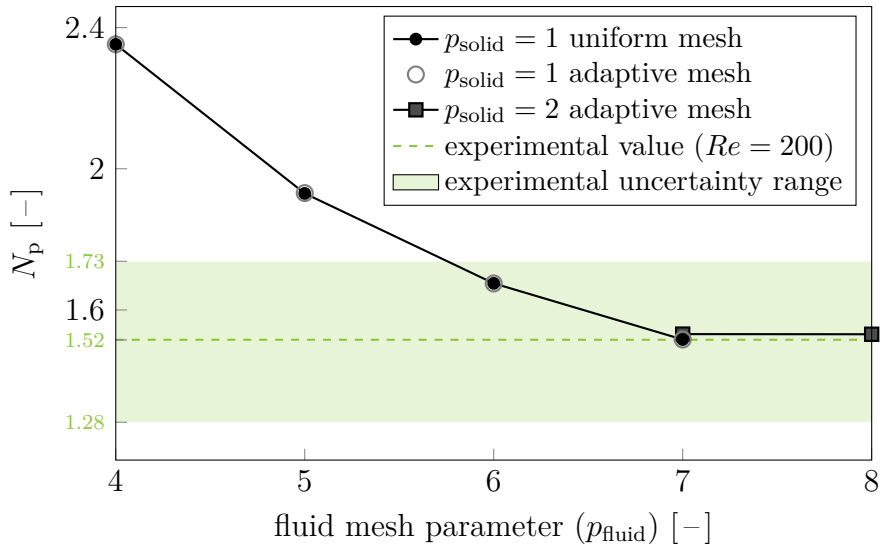


Figure 12: Power number (N_p) computed values for different fluid mesh parameter (p_fluid), for $Re = 200$ and $\Delta t = 1 \times 10^{-3}$ s, with uniform and adaptive mesh. Results are inside the experimental uncertainty range for $p_\mathrm{fluid} \geq 6$.

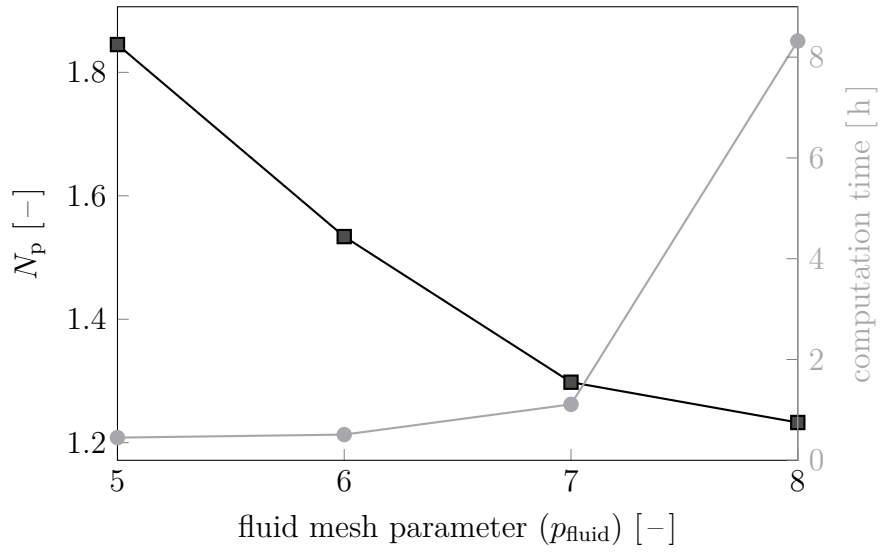


Figure 13: Power number (N_p) computed values for different fluid mesh parameter (p_fluid), for $Re = 2000$, with adaptive mesh. $p_\mathrm{fluid} = 7$ appear to be the best compromise between convergence and computational time.

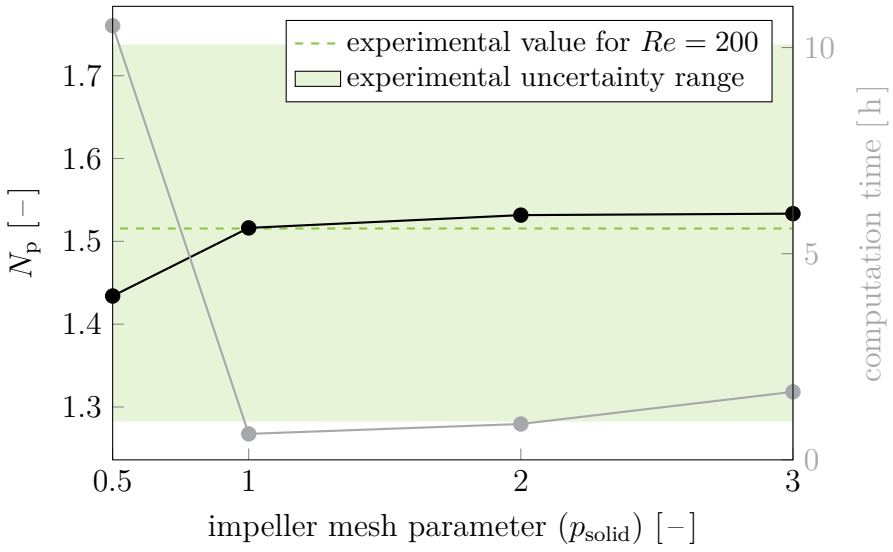


Figure 14: Power number (N_p) computed values for different solid mesh parameter (p_solid) of the impeller, with a fluid mesh parameter $p_\mathrm{fluid} = 7$.

Time step. Time integration is achieved through an implicit second order backward differentiation formula (BDF2). Figure 15 shows the evolution of N_p for different time steps Δt, on a semi-log scale, for $Re = 200$, $p_\mathrm{fluid} = 7$ and $p_\mathrm{solid} = 1$. It appears that although the value obtained for $\Delta t = 1 \times 10^{-3}$ s is closer to the experimental value, the value of N_p reaches a plateau only for $\Delta t \leq 2 \times 10^{-4}$ s. All values computed are within the experimental uncertainty range. The computational time, displayed on the secondary axis, are almost inversely proportional with Δt for $\Delta t \leq 4 \times 10^{-4}$ s. Hence, we chose the lowest Δt that guarantees that the results are independent from the time step: $\Delta t = 2 \times 10^{-4}$ s.

5.2.3. Parameters specific to the baffled mixing rig simulations

We use NIB method to model the four baffles added in the mixing rig, see Figure 10. Most of the parameters found with the sensitivity analysis of the unbaffled mixing rig simulations can be used also for baffled mixing rig simulations. We present in this subsection the parameters that must be set differently.

Fluid mesh. Modeling the baffles with NIB method simplifies the meshing procedure greatly, compared to creating a fluid mesh with the area corresponding to the baffles removed, and make the simulations much more adaptable. Indeed, the fluid mesh topology is not impacted by the baffle geometry, number and positioning, which is very advantageous for engineering applications.

However, the previous fluid mesh adaptation strategy, with an initial refinement of 4 and a minimum refinement of 3, is not fit anymore. Indeed, as adaptation is based on the fluid velocity gradient, it coarsens the cells near the hull of the tank and near the baffles. In our study, we found that the results obtained for a baffled tank with an

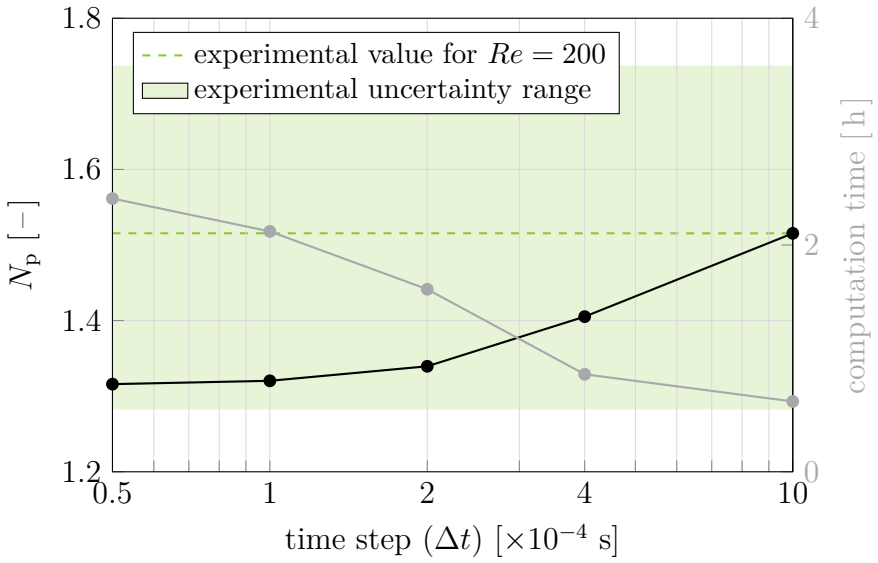


Figure 15: Power number (N_p) computed values for different time step (Δt), for $Re = 200$ and 2000, for adaptive mesh with $p_{\mathrm{fluid}} = 7$ and 8, and $p_{\mathrm{solid}} = 1$. Results at $t = 0.2\,\mathrm{s}$ are inside the experimental uncertainty range for all Δt values considered.

adaptative mesh differs greatly from results obtained with a uniform mesh (contrary to the unbaffled mixing rig simulations where both procedures give very close results, see Figure 12). In fact, results obtained for the baffled mixing rig with our previous adaptative mesh strategy are similar to the results obtained for an unbaffled maxing rig, indicating that a coarse mesh in the vicinity of the baffles, which are quite thin, results in them not being correctly accounted for in the flow. This has been observed both for laminar and turbulent flows.

Then, we found that using a uniform mesh with $p_{\mathrm{fluid}} = 6$ offers a good compromise between convergence of the results and computational costs for the baffled tank simulations.

Baffles mesh. The number of Nitsche particles for the baffles grows rapidly with p_{solid}, as given in Table 4, because of the large volume occupied by the four baffles (represented in Figure 10). To lower the memory consumption and computational costs, we use $p_{\mathrm{solid}} = 0.5$ for the baffles (we still use $p_{\mathrm{solid}} = 1$ for the impeller).

Moreover, because the baffles are static, the Nitsche restriction parameter (β_0 in Equation (9)) should be increased. For the impeller, we still use $\beta_0 = 10$, and for the baffles, we set $\beta_0 = 1000$. We choose this value of β_0 for the baffles among several value tested (10, 100, 1000 and 10 000), for various Re, as the best compromise between convergence of the results and a low computation time.

5.2.4. Summary

Table 5 summarizes the parameters for both the unbaffled and the baffled mixing rig, that we use to plot the power curve. As an indication, computational times that

we experienced on 3 nodes of the ComputeCanada cluster Narval range from 3.25 h to 6 h for the unbaffled mixing rig simulations, and from 7.5 h to 12.5 h for the baffled mixing rig. The computational time tends to increase with Re.

mixing rig	unbaffled	baffled
fluid mesh type	adaptative	constant
p_{fluid}	7	6
p_{solid} impeller	1	1
p_{solid} baffles	-	0.5
β_0 impeller	10	10
β_0 baffles	-	1000
Δt	2×10^{-4} s	2×10^{-4} s

Table 5: Parameters used in the simulations to plot the power curve, for both the unbaffled and baffled mixing rig simulations.

Figure 16 presents the time evolution of the torque on the impeller, T in Equation 21, obtained with these parameters, for the baffled and unbaffled mixing rig simulations at $Re = 200$. The torque T is on average stable starting from $t = 0.06$ s. We can then use the mean of the torque from $t = 0.1$ s to compute N_{p}. Moreover, the increase of T when baffles are added to the mixing rig is consistent with previous experimental observations [36].

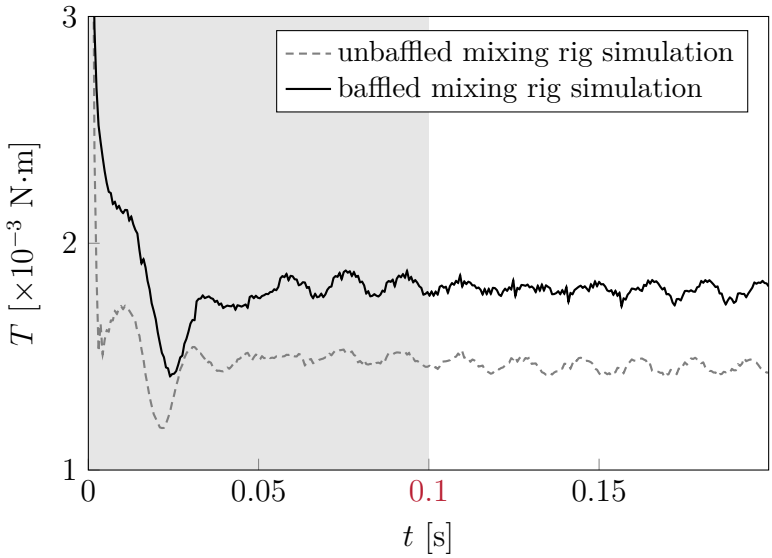


Figure 16: Torque (T) evolution in time (t), for baffled and unbaffled mixing rig simulations at $Re = 200$, showing the signal used to calculate the stabilized torque value ($t \geq 0.1$ s).

5.3. Results for varying Re

We carry out simulations of the mixing rig with the NIB method for 40 different Re values, with $Re \in \left[1\ ,\ 2 \times 10^3\right]$, with the parameters determined from the sensitivity analysis (Table 5). Figure 17 and Figure 18 show the power curve ($N_\mathrm{p} = f(Re)$) in a log scale, for the unbaffled mixing rig and the baffled mixing rig respectively. We compare simulation results with experimental results from [36]. Note that we were able to run simulations with $Re \geq 200$, which was the upper limit for the previous numerical study associated with the experimental results we use here [36].

For both cases, simulation results stay inside the experimental errors for the whole Re range tested. At low Re, simulation results for the baffled and unbaffled mixing rig are similar, and show a clear divergence from $Re = 200$ in Figure 18. At high Re, results show an excellent fit with the available experimental results for the baffled mixing rig. In particular, the plateau obtained on the experimental data for $Re \geq 500$ is well visible on the simulation results.

These results are consistent with the literature on baffled tanks: at low Re, N_p is independent of the presence of baffles, and at high Re, at which most mixing operations are performed, the N_p becomes independent of Re [71].

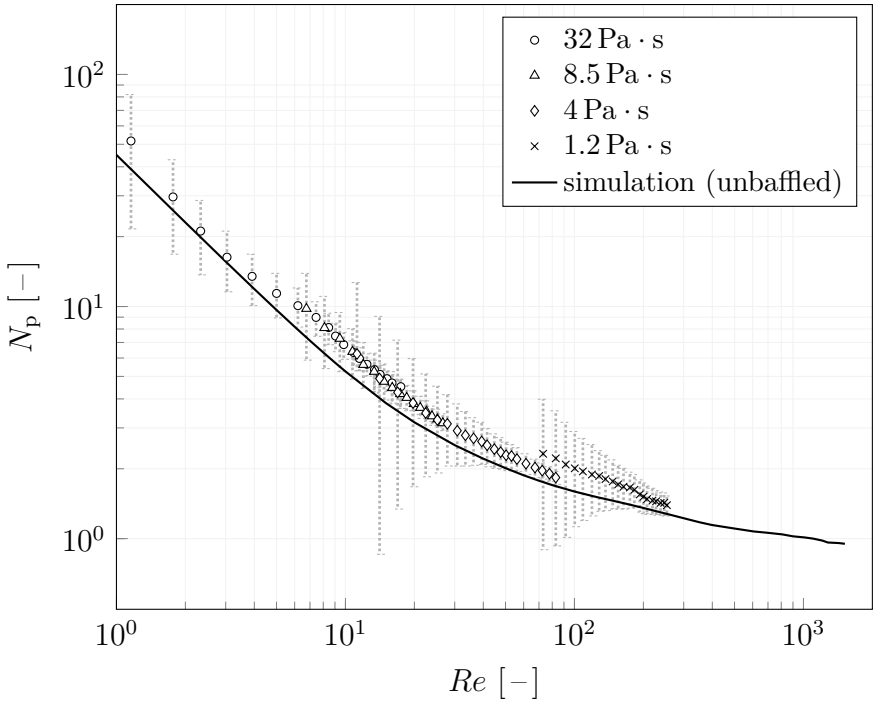


Figure 17: Power curve for unbaffled mixing rig. Experimental results, with error bars, come from [36]. Simulation results show a good agreement with experimental results.

From these results, obtained with one (impeller only) or five (impeller and baffles) solids simulated with the NIB method, rotating or static, we can conclude that the NIB method is suitable for the simulation of mixing rigs for a large range of Re.

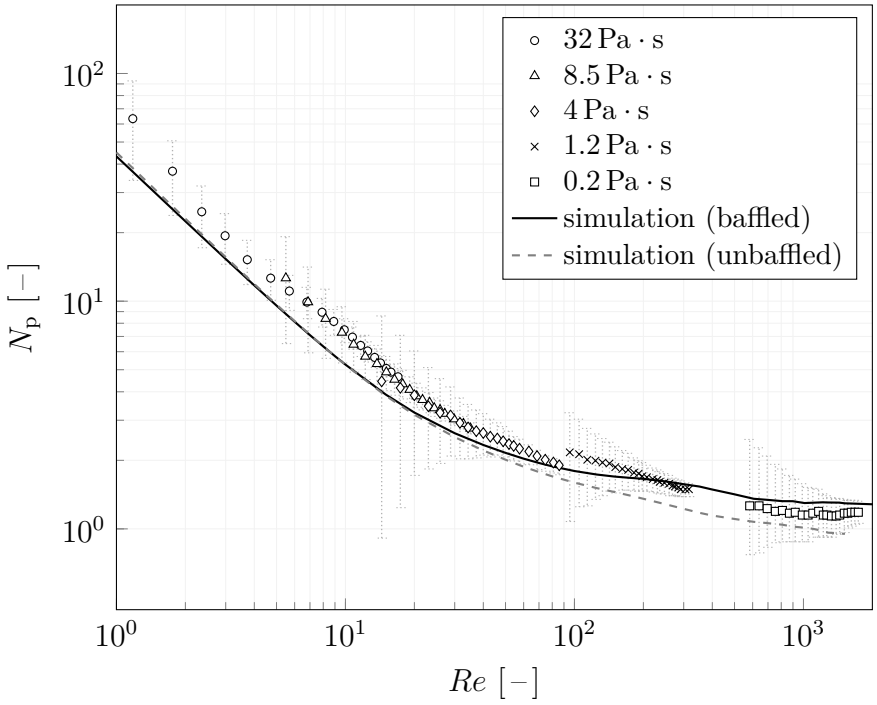


Figure 18: Power curve for a baffled mixing rig. Experimental results, with error bars, come from [36]. Simulation results show a good agreement with experimental results, particularly for $Re > 100$ where it differs from results obtained for the unbaffled mixing rig.

6. Simulation of a two-impeller mixing rig

With the NIB method, it is then possible to simulate cases that are not accessible to classical conformal mesh techniques. In this last section, we will showcase simulations of mixing rig with two impellers, with an overlap in the swept area of the impeller's blades.

We set the simulations with the same methodology and use the same parameters as for the unbaffled mixing rig (see Table 5). Both impellers have the geometry given in Figure 1 and Table 2. The files used to generate the mesh with the OpenSource mesh generator GMSH [69] are available at the Lethe utilities repository [70]. The impellers' axes of rotation are parallel to the tank revolution axis, with an offset of $\pm\frac{3D}{8}$, on the same diameter line. The impeller's speed is still set to $N = 16\,\mathrm{Hz}$, with one impeller turning clockwise and the other turning counter-clockwise.

Figure 19 shows the time-evolution of the torque on one of the impeller, for a kinematic viscosity $\nu = 1.2 \times 10^{-3}\,\mathrm{m}^2/\mathrm{s}$. For comparison purposes, we superimpose the torque signal previously obtained for a single impeller in an unbaffled mixing rig, for the same viscosity ($Re = 200$). Similarly to the one-impeller simulation, the signal for the two-impeller simulation presents small oscillations after the first impeller revolution ($t \geq 0.0625\,\mathrm{s}$). The amplitude of these oscillations are greater during the revolution 2, and though they dampen during the revolution 3 they are still greater than the oscillations during revolution 1 of the one-impeller simulation. This indicates that to draw the evolution of N_p with ν, a longer simulation time would be needed to reach the pseudo-permanent regime. Also, the frequency of the oscillations of T is affected by the presence of the second impeller, and is more closely correlated with the number of blades on each impeller than in the one-impeller simulation.

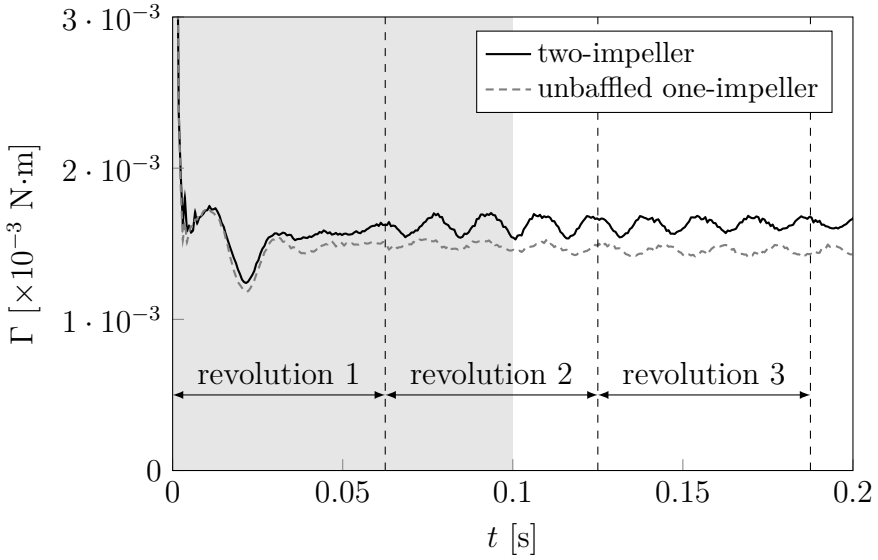
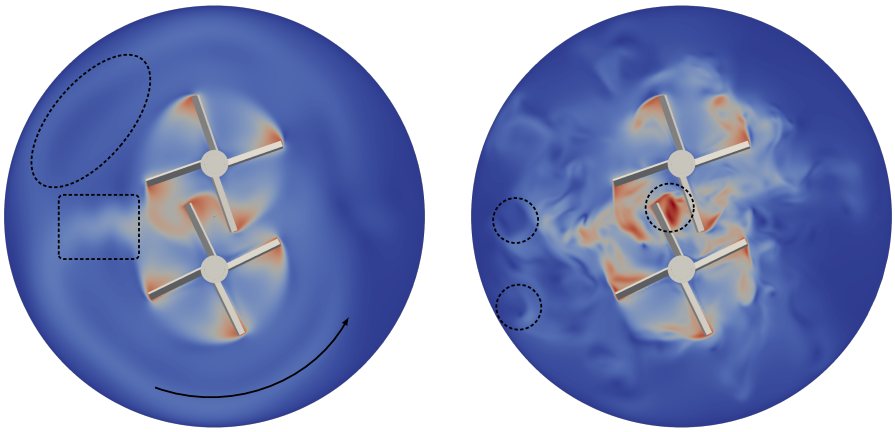


Figure 19: Torque (T) evolution with time (t) on one of the impeller, for the two-impeller mixing rig, compared to previous results obtained for the unbaffled mixing rig.



(a) $\nu = 1.2 \times 10^{-3}\,\mathrm{m}^2/\mathrm{s}$ ($Re \approx 200$) (b) $\nu = 2 \times 10^{-4}\,\mathrm{m}^2/\mathrm{s}$ ($Re \approx 1000$)

Figure 20: Fluid velocity magnitude profile, range from 0, in dark blue, to $7.4\,\mathrm{m\,s}^{-1}$ (tip-speed ratio of 1.25), in dark red, at $t = 0.2\,\mathrm{s}$, for the two-impeller mixing rig simulation. Flow patterns discernible through the velocity changes are indicated with black dashed shapes: oscillations (in the frame), rolls (in the ellipse) and eddies (in the circles) .

Figure 20 shows the fluid velocity magnitude profile on the plane that passes through the middle of the blades. It is clearly visible that the blades rotation pushes the fluid to the left, thus the velocity of the fluid is greater in the wake of the blades. For $\nu = 1.2 \times 10^{-3}\,\mathrm{m}^2/\mathrm{s}$ (which corresponds to $Re = 200$ for a single impeller mixing rig), the fluid oscillates in the wake of the overlapping region between the two impellers (black dashed frame in Figure 20a), which is expected. Then, the flow is slowed down in the vicinity of the tank and follows it (black curved arrow in Figure 20a) before entering the blades vicinity again. Even at this intermediate Re, the flow is 3-dimensional and moves in large rolls near the tank, as hinted by a low fluid velocity area between two higher velocity areas (black dashed ellipse in Figure 20a). For $\nu = 2 \times 10^{-4}\,\mathrm{m}^2/\mathrm{s}$ (which corresponds to $Re = 1000$ for a single impeller mixing rig), the flow is much more turbulent. The multiple regions of rapid velocity change, see black dashed circles in Figure 20b as examples, correspond to eddies that propagate throughout the vessel, which is expected for $Re = 1000$.

Thus, the Nitsche IB method is suitable to simulate a mixing rig with two impellers, with an overlap in the swept area of the impeller's blades, and provide physical results in a large range of Re.

7. Conclusion

In this work, we have introduced a restriction method to treat immersed boundary conditions, called the Nitsche immersed boundary (NIB) method. It has been implemented in the open-source CFD Software Lethe [44].

We have verified the NIB method using canonical test cases: the Taylor-Couette flow and the von Karman vortex street behind a cylinder. The Taylor-Couette flow shows that NIB intrinsically reduces the order of the underlying GLS scheme, but

this can be countered by an appropriate adaptative mesh refinement strategy. For the von Karman vortex street, we found a very good agreement between the NIB method results and conformal mesh results on the values of the lift and drag coefficient on the cylinder, which validates the capacity of the immersed boundary to predict the dynamic of transient flow.

We have then confronted simulation results with the NIB method on the case of a stirred tank, with and without baffles, for which we have experimental data available in the literature [36]. Both the rotating impeller and the four static baffles were modeled using the NIB method. We detailed the sensitivity analysis performed to set the various simulation parameters needed, such as the refinement of the fluid mesh, the refinement of the solid mesh and the time step. The power curves obtained for a large range of Reynolds numbers showed that simulation results match the experimental data very well. Therefore, the NIB method is suitable for the simulation of mixing rigs, from fully laminar to fully turbulent flows.

Finally, to showcase the versatility of the NIB method, we simulated a mixing rig with two impellers, with an overlap in the swept area of the impeller's blades. Though we do not have available experimental data for this case, simulation results (torque on one impeller and flow patterns) were realistic.

Thus, the NIB method offers a robust and flexible mesh generation technique for the solid immersed body, with no change in the topology of the fluid mesh. With this method, multiple solids with arbitrary geometry and moving in an arbitrary fashion can easily be modeled.

The Software as well as all the files that we used for the simulations are available in the public domain, under the LGPL version 2.1 [44, 70, 72].

8. Acknowledgements

Jeanne Joachim would like to acknowledge the financial support from Mitacs Elevate program, for her postdoctoral research with the industrial partner AMF Bakery (Sherbrooke, Qc).

Bruno Blais 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.

The authors would also like to acknowledge technical support and computing time provided by the Digital Research Alliance of Canada and Calcul Québec. Computations shown in this work were made on the supercomputer Narval and Niagara managed by Calcul Québec and the Digital Research Alliance of Canada. The operation of these supercomputers is funded by the Canada Foundation for Innovation (CFI), the Ministère de l'Économie, de la Science et de l'Innovation du Québec (MESI) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

The entire team would like to acknowledge support received by members of the `deal.II` user group.

References

[1] C. Rauwendaal, Mixing in polymer processing, Vol. 23, CRC Press, 1991.

[2] P. J. Cullen, Food mixing: Principles and applications, John Wiley & Sons, 2009.

[3] D. A. R. Brown, P. N. Jones, J. C. Middleton, Experimental Methods, Part A: Measuring Tools and Techniques for Mixing and Flow Visualization Studies, in: Handbook of industrial mixing, Wiley Online Library, 2004, Ch. 4, pp. 145–256.

[4] H. E. Van den Akker, The details of turbulent mixing process and their simulation, in: G. B. Marin (Ed.), Computational Fluid Dynamics, Vol. 31 of Advances in Chemical Engineering, Academic Press, 2006, pp. 151–229. doi:10.1016/S0065-2377(06)31003-4.

[5] M. Lassaigne, B. Blais, L. Fradette, F. Bertrand, Experimental investigation of the mixing of viscous liquids and non-dilute concentrations of particles in a stirred tank, Chemical Engineering Research and Design 108 (2016) 55–68. doi:10.1016/j.cherd.2016.01.005.

[6] G. Lane, M. Schwarz, G. Evans, Comparison of CFD methods for modelling of stirred tanks, in: 10th European Conference on Mixing, Elsevier, 2000, pp. 273–280. doi:10.1016/B978-044450476-0/50035-2.

[7] J. B. Joshi, N. K. Nere, C. V. Rane, B. Murthy, C. S. Mathpati, A. W. Patwardhan, V. V. Ranade, CFD simulation of stirred tanks: Comparison of turbulence models. Part I: Radial flow impellers, The Canadian Journal of Chemical Engineering 89 (1) (2011) 23–82. doi:10.1002/cjce.20446.

[8] J. B. Joshi, N. K. Nere, C. V. Rane, B. Murthy, C. S. Mathpati, A. W. Patwardhan, V. V. Ranade, CFD simulation of stirred tanks: Comparison of turbulence models. Part II: Axial flow impellers, multiple impellers and multiphase dispersions, The Canadian Journal of Chemical Engineering 89 (4) (2011) 754–816. doi:10.1002/cjce.20465.

[9] E. L. Paul, V. A. Atiemo-Obeng, S. M. Kresta, Handbook of Industrial Mixing: Science and Practice, John Wiley & Sons, 2004. doi:10.1002/0471451452.

[10] P. R. Gogate, A. A. Beenackers, A. B. Pandit, Multiple-impeller systems with a special emphasis on bioreactors: a critical review, Biochemical Engineering Journal 6 (2) (2000) 109–144. doi:10.1016/S1369-703X(00)00081-4.

[11] A. Brucato, M. Ciofalo, F. Grisafi, G. Micale, Numerical prediction of flow fields in baffled stirred vessels: A comparison of alternative modelling approaches, Chemical Engineering Science 53 (21) (1998) 3653–3684. doi:10.1016/S0009-2509(98)00149-3.

[12] J. Y. Luo, A. D. Gosman, Prediction of impeller-induced flow in mixing vessels using multiple frames of reference, Vol. 136, Institute of Chemical Engineers Symposium Series, 1994.

[13] G. L. Lane, M. P. Schwarz, G. M. Evans, Predicting gas-liquid flow in a mechanically stirred tank, Applied Mathematical Modelling 26 (2) (2002) 223–235. doi:10.1016/S0307-904X(01)00057-9.

[14] A. Tamburini, A. Cipollina, G. Micale, A. Brucato, M. Ciofalo, CFD Simulations of Dense Solid-Liquid Suspensions in Baffled Stirred Tanks: Prediction of Suspension Curves, Chemical Engineering Journal 178 (2011) 324–341. doi:10.1016/j.cej.2011.10.016.

[15] J. Y. Murphy, S. R. Mathur, D. Choudhary, CFD Simulation of Flows in Stirred Tank Rectors using a Sliding Mesh Technique, in: Proc. 8th European Conference on Mixing, 1994, pp. 155–162.

[16] OpenFOAM website, https://www.openfoam.com/.

[17] G. Montante, K. Lee, A. Brucato, M. Yianneskis, Numerical simulations of the dependency of flow pattern on impeller clearance in stirred vessels, Chemical Engineering Science 56 (12) (2001) 3751–3770. doi:10.1016/S0009-2509(01)00089-6.

[18] A. Tamburini, A. Cipollina, G. Micale, CFD simulation of solid liquid suspensions in baffled stirred vessels below complete suspension speed, Chemical Engineering Transactions 24 (2011) 1435–1440. doi:10.3303/CET1124240.

[19] B. Blais, M. Lassaigne, C. Goniva, L. Fradette, F. Bertrand, A semi-implicit immersed boundary method and its application to viscous mixing, Computers & Chemical Engineering 85 (2016) 136–146. doi:10.1016/j.compchemeng.2015.10.019.

[20] J. Dürrwächter, M. Kurz, P. Kopper, D. Kempf, C.-D. Munz, A. Beck, An efficient sliding mesh interface method for high-order discontinuous galerkin schemes, Computers & Fluids 217 (2021) 104825. doi:10.1016/j.compfluid.2020.104825.

[21] J. Gao, A sliding-mesh interface method for three dimensional high order spectral difference solver, Journal of Computational Physics 454 (2022) 110988. doi:10.1016/j.jcp.2022.110988.

[22] S. P. Domino, Design-order, non-conformal low-mach fluid algorithms using a hybrid cvfem/dg approach, Journal of Computational Physics 359 (2018) 331–351. doi:10.1016/j.jcp.2018.01.007.

[23] H. Tang, S. C. Jones, F. Sotiropoulos, An overset-grid method for 3d unsteady incompressible flows, Journal of Computational Physics 191 (2) (2003) 567–600. doi:10.1016/S0021-9991(03)00331-0.

[24] I. Borazjani, L. Ge, T. Le, F. Sotiropoulos, A parallel overset-curvilinear-immersed boundary framework for simulating complex 3d incompressible flows, Computers & fluids 77 (2013) 76–96. doi:10.1016/j.compfluid.2013.02.017.

[25] W. J. Horne, K. Mahesh, A massively-parallel, unstructured overset method to simulate moving bodies in turbulent flows, Journal of Computational Physics 397 (2019) 108790. doi:10.1016/j.jcp.2019.06.066.

[26] M. Hedayat, A. M. Akbarzadeh, I. Borazjani, A parallel dynamic overset grid framework for immersed boundary methods, Computers & Fluids 239 (2022) 105378. doi:10.1016/j.compfluid.2022.105378.

[27] A. Posa, A. Lippolis, R. Verzicco, E. Balaras, Large-eddy simulations in mixed-flow pumps using an immersed-boundary method, Computers & Fluids 47 (1) (2011) 33–43. doi:10.1016/j.compfluid.2011.02.004.

[28] 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. doi:10.1016/j.jcp.2006.10.032.

[29] 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. doi:10.1016/j.jcp.2013.04.033.

[30] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, Journal of Computational Physics 230 (2) (2011) 329–348. doi:10.1016/j.jcp.2010.09.017.

[31] D. Russell, Z. Jane Wang, A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow, Journal of Computational Physics 191 (1) (2003) 177–205. doi:10.1016/S0021-9991(03)00310-3.

[32] D. Boffi, L. Gastaldi, L. Heltai, C. S. Peskin, On the hyper-elastic formulation of the immersed boundary method, Computer Methods in Applied Mechanics and Engineering 197 (25-28) (2008) 2210–2231. doi:10.1016/j.cma.2007.09.015.

[33] K. Khadra, P. Angot, S. Parneix, J. Caltagirone, Fictitious domain approach for numerical modelling of Navier-Stokes equations, International Journal for Numerical Methods in Fluids 34 (8) (2000) 651–684. doi:10.1002/1097-0363(20001230)34:8<651::AID-FLD61>3.0.CO;2-D.

[34] S. Kang, G. Iaccarino, P. Moin, Accurate immersed-boundary reconstructions for viscous flow simulations, AIAA journal 47 (7) (2009) 1750–1760. doi:10.2514/1.42187.

[35] L. Barbeau, S. Etienne, C. Beguin, B. Blais, Development of a high-order continuous galerkin sharp-interface immersed boundary method and its application to incompressible flow problems, Computers & Fluids 239 (2022) 105415. doi:10.1016/j.compfluid.2022.105415.

[36] B. Blais, M. Lassaigne, C. Goniva, L. Fradette, F. Bertrand, A semi-implicit immersed boundary method and its application to viscous mixing, Computers & Chemical Engineering 85 (2016) 136–146. doi:10.1016/j.compchemeng.2015.10.019.

[37] P. Hansbo, J. Hermansson, Nitsche's method for coupling non-matching meshes in fluid-structure vibration problems, Computational Mechanics 32 (1-2) (2003) 134–139. doi:10.1007/s00466-003-0467-7.

[38] E. Burman, M. A. Fernández, An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes, Computer Methods in Applied Mechanics and Engineering 279 (2014) 497–514. doi:10.1016/j.cma.2014.07.007.

[39] A. Massing, M. G. Larson, A. Logg, M. E. Rognes, A stabilized nitsche overlapping mesh method for the stokes problem, Numerische Mathematik 128 (1) (2014) 73–101. doi:10.1007/s00211-013-0603-z.

[40] P. Hansbo, Nitsche's method for interface problems in computational mechanics, GAMM-Mitteilungen 28 (2) (2005) 183–206. doi:10.1002/gamm.201490018.

[41] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. part i: Poisson and stokes problems, Journal of Computational Physics 372 (2018) 972–995. doi:10.1016/j.jcp.2017.10.026.

[42] L. Zhang, A. Gerstenberger, X. Wang, W. K. Liu, Immersed finite element method, Computer Methods in Applied Mechanics and Engineering 193 (21-22) (2004) 2051–2067. doi:10.1016/j.cma.2003.12.044.

[43] 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. doi:10.1016/j.softx.2020.100579.

[44] Lethe-cfd git repository, https://github.com/lethe-cfd/lethe.

[45] T. E. Tezduyar, S. Sathe, Stabilization parameters in SUPG and PSPG formulations, Journal of Computational and Applied Mechanics 4 (1) (2003) 71–88.

[46] 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.

[47] T. E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, Advances in applied mechanics 28 (1991) 1–44. doi:10.1016/S0065-2156(08)70153-4.

[48] 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.

[49] F. Ilinca, K. R. Yu, B. Blais, The effect of viscosity on free surface flow inside an angularly oscillating rectangular tank, Computers & Fluids 183 (2019) 160–176. doi:10.1016/j.compfluid.2019.02.021.

[50] D. Boffi, F. Brezzi, M. Fortin, et al., Mixed finite element methods and applications, Vol. 44, Springer, 2013. doi:10.1007/978-3-642-36519-5.

[51] J. Donea, A. Huerta, Finite element methods for flow problems, John Wiley & Sons, 2003.

[52] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, T. Coupez, Stabilized finite element method for incompressible flows with high Reynolds number, Journal of Computational Physics 229 (23) (2010) 8643–8665. doi:10.1016/J.JCP.2010.07.030.

[53] M. G. Larson, F. Bengzon, The finite element method: theory, implementation, and applications, Vol. 10, Springer Science & Business Media, 2013.

[54] D. Arndt, W. Bangerth, M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Sticko, B. Turcksin, D. Wells, The deal.II library, version 9.4, Journal of Numerical Mathematics (2022). doi:10.1515/jnma-2022-0054.

[55] 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. Tomas, B. Turcksin, Z. Wang, D. Wells, The deal.II library, version 9.2, Journal of Numerical Mathematics 28 (3) (2020) 131–146. doi:10.1515/jnma-2020-0043.

[56] D. Arndt, W. Bangerth, B. Blais, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, U. Köcher, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Proell, K. Simon, B. Turcksin, D. Wells, J. Zhang, The deal.II library, version 9.3, Journal of Numerical Mathematics 29 (3) (2021) 171–186. doi:10.1515/jnma-2021-0081.

[57] L. Heltai, B. Blais, R. Gassmöller, The deal. II tutorial step-70: Massively parallel non-matching grid simulations of fluid structure interaction problems (May 2020). doi:10.5281/zenodo.3829064.

[58] deal.ii documentation for the ParticleHandler class, https://www.dealii.org/current/doxygen/deal.II/classParticles_1_1ParticleHandler.html.

[59] P. Becker, S. R. Idelsohn, E. Oñate, A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh, Computational Mechanics 55 (6) (2014) 1091–1104. doi:10.1007/s00466-014-1107-0.

[60] D. W. Kelly, J. De SR Gago, O. C. 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. doi:10.1002/nme.1620191103.

[61] R. B. Bird, W. E. Stewart, E. N. Lightfoot, Transport Phenomena, second edition Edition, John Wiley & Sons, 2007. doi:10.1115/1.1424298.

[62] S. K. Kang, Y. A. Hassan, A comparative study of direct-forcing immersed boundary-lattice boltzmann methods for stationary complex boundaries, International Journal for Numerical Methods in Fluids 66 (9) (2011) 1132–1158. doi:10.1002/fld.2304.

[63] 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. doi:10.1017/S0022112086003014.

[64] 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. doi:10.1006/jcph.2001.6705.

[65] R. D. Henderson, Nonlinear dynamics and pattern formation in turbulent wake transition, Journal of Fluid Mechanics 352 (1997) 65–112. doi:10.1017/S0022112097007465.

[66] 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. doi:10.1006/jcph.2000.6556.

[67] A. Hay, S. Etienne, D. Pelletier, A. Garon, hp-adaptive time integration based on the BDF for viscous flows, Journal of Computational Physics 291 (2015) 151–176. doi:10.1016/j.jcp.2015.03.022.

[68] deal.II documentation for the GridGenerator namespace, https://www.dealii.org/current/doxygen/deal.II/namespaceGridGenerator.html.

[69] C. Geuzaine, J.-F. Remacle, A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309—-1331. doi:10.1002/nme.2579.

[70] Lethe-utils git repository, folder containing the .geo files used for this article, https://github.com/lethe-cfd/lethe-utils/tree/master/gmsh/mixers/NIB_article.

[71] J. Rushton, E. Costich, H. Everett, Power characteristics of mixing impeller Part II, Chem. Eng. Prog. 46 (1950) 467–476.

[72] Lethe-utils git repository, folder containing the .prm files used for this article, https://github.com/lethe-cfd/lethe-utils/tree/master/cases/nitsche.