Immersogeometric analysis of moving objects in incompressible flows

Songzhe Xu[a], Fei Xu[b], Aditya Kommajosula[a], Ming-Chen Hsu[a], Baskar Ganapathysubramanian[a,*]

[a] *Department of Mechanical Engineering, Iowa State University, 2043 Black Engineering, Ames, IA 50011, USA*
[b] *Ansys Inc., 807 Las Cimas Parkway, Austin, TX 78746, USA*

Abstract

We deploy the immersogeometric approach for tracking moving objects. The method immerses objects into non-boundary-fitted meshes and weakly enforces Dirichlet boundary conditions on the object boundaries. The object motion is driven by the integrated surface force and external body forces. A residual-based variational multiscale method is employed to stabilize the finite element formulation for incompressible flows. Adaptively refined quadrature rules are used to better capture the geometry of the immersed boundaries by accurately integrating the intersected background elements. Treatment for the freshly-cleared nodes (i.e. background mesh nodes that are inside the object at one time step, but are in the fluid domain at the next time step) is considered. We assess the accuracy of the method by analyzing object motion in different flow structures including objects freely dropping in viscous fluids and particle focusing in unobstructed and obstructed micro-channels. We show that key quantities of interest are in very good agreements with analytical, numerical and experimental solutions. We also show a much better computational efficiency of this framework than current commercial codes using adaptive boundary-fitted approaches. We anticipate deploying this framework for applications of particle inertial migration in microfluidic channels.

Keywords: Moving objects, Immersogeometric method, Weakly enforced boundary conditions, Particle inertial migration, Fluid–structure interaction

Contents

[*] Corresponding author
Email addresses: jmchsu@iastate.edu (Ming-Chen Hsu), baskarg@iastate.edu (Baskar Ganapathysubramanian)

1. Introduction

Control and localization of finite-size particles (e.g., cells and precipitates) in aqueous flows are useful in biological processing, chemical reaction control, and for creating structured materials. Some examples include fast identification of *E. coli* in water, robust removal of circulating tumor cells from the blood plasma, and fast separation of cell types for rapid flow cytometry. The precise, efficient and cheap localization of a heterogeneous collection of cells in a fluid medium is an important challenge with multiple engineering and health applications. Computational modeling of particle motion, especially under inertial flow conditions, is a promising approach to understand, control and localize cells in microfluidic devices.

Tracking finite-size particle motion in inertial flows (especially in obstructed geometries) is a computationally challenging problem. Approaches to simulate such a system include boundary-fitted and non-boundary-fitted methods. In the boundary-fitted approach, the fluid problem is solved on a mesh that conforms to the fluid–object (particle) interface and deforms around it. The fluid problem on the deforming domain is written in an arbitrary Lagrangian–Eulerian (ALE) [1–3] or a space–time (ST) [4, 5] coordinate system. Boundary-fitted methods have the advantage of satisfying kinematic constraints, such as no-slip boundary conditions, by construction. However, for situations that involve large translational and/or rotational interface motions, the boundary-fitted mesh can become severely distorted if it is continuously deformed from a single reference configuration, harming both the conditioning of the discrete problem and the accuracy of its solution. Applying boundary-fitted methods to complex moving-interface problems may therefore require specialized solution strategies to maintain fluid mesh quality. One approach is remeshing, in which all or part of the fluid domain is re-discretized when mesh distortion becomes too extreme [6–8]. Mesh management is complicated further if the object moves into and out of contact with other objects, changing the topology of the fluid domain [9, 10].

For these reasons, non-boundary-fitted approaches have become a popular alternative for the simulation of moving-interface problems. Non-boundary-fitted methods approximate the solution of boundary value problems on analysis meshes that do not necessarily conform to the boundary of the domain. The analysis object is arbitrarily superimposed onto (or *immersed* into) a background fluid mesh. Such methods have greater geometric flexibility than their boundary-fitted counterparts. However, kinematic constraints such as Dirichlet boundary conditions at the immersed interface can no longer be imposed strongly on the discrete solution space. To apply interface conditions, one must devise a suitable method for weak enforcement. The first non-boundary-fitted approach that became widely known for computational fluid dynamics (CFD) was the immersed boundary method [11, 12]. Since then, immersed methods have been applied to a variety of flow problems [13–16]. In the context of finite elements, several adaptations of immersed methods were explored in the 2000s for the simulation of fluid interacting with moving objects. Glowinski and coworkers [17–19] simulated viscous flows interacting with rigid particles by enforcing the rigid-body-motion constraint on the overlapping fluid mesh through a distributed Lagrange multiplier field. Zhang, Liu and cowokers [20–23] developed the immersed finite element method (IFEM) to use a flexible Lagrangian solid mesh that moves on top of a background Eulerian fluid mesh. Casquero et al. [24] later enhanced IFEM by introducing non-uniform rational B-splines (NURBS) as the basis functions to improve the robustness and accuracy of the immersed method. In addition, Rüberg and Cirak [25] and Kadapa et al. [26] applied Nitsche's method [27] at the immersed interface with background B-spline finite elements for the simulation of moving-boundary problems.

While immersed methods show great flexibility in solving complex moving-boundary problems, they typically suffer from reduced accuracy of the solution near the immersed boundary. Kamensky et al. [28] and Xu et al. [29] found that the reduced accuracy is partially related to the representation of the geometry in the immersed domain. The immersogeometric analysis (IMGA) approach was proposed to alleviate this issue, by faithfully capturing the immersed geometry in the intersected elements. The method also alleviates the difficulties associated with CFD mesh generation around complex design geometries. The immersogeometric method is comprised of the following main components. A variational multiscale (VMS) formulation of incompressible flows

[30–33] is used, which provides accuracy and robustness in both laminar and turbulent flow simulations. The Dirichlet boundary conditions on the surface of the immersed geometry are enforced weakly using an extension of Nitsche's method [27, 34]. This weak boundary condition formulation can be integrated over the immersed object surface directly using its computer-aided design (CAD) boundary representation (B-rep) [35, 36]. An exact representation of the design geometry is therefore used in the simulation, sharing the same philosophy with isogeometric analysis [37]. Adaptively refined quadrature rules are used to accurately integrate the background elements cut by the immersed boundary. It was found in Xu et al. [29] that improved quadrature in intersected elements is critical for obtaining accurate flow solutions when using immersed methods. In this work, we apply the immersogeometric method to the analysis of moving-particle problems.

Applying the IMGA approach to problems with moving objects raises some challenges. The most significant one is the treatment of freshly-cleared nodes – the nodes that are previously inside the object at one time step, but are outside (i.e. in the fluid domain) at the next time step due to object motion. These nodes lack a time history of the fluid field. This may result in numerical issues arising from improper initial guesses, leading to numerical instabilities and error accumulation. Udaykumar et al. [38] approximated the velocity and pressure on those freshly-cleared nodes via interpolation of neighboring fluid nodes and immersed object surface points. We adopt this technique and extend it in the context of a finite element framework.

The paper is organized as follows. In Section 2, we summarize the formulation of the Navier–Stokes equations of incompressible flows posed on a non-boundary-fitted discretization. Section 3 describes the details of implementation of the framework. In Section 4, we verify and validate the developed method using several cases, including free falling 2D cylinder and 3D sphere in viscous fluids, and particle focusing and migration in obstructed and unobstructed channels. We demonstrate the capability of the developed framework on applications of fluid–structure interaction. Finally, we draw conclusions and motivate future research in Section 5.

2. Immersogeometric analysis

In this section, we summarize the variational formulation of the Navier–Stokes equations of incompressible flows and its spatial and temporal discretizations. We also emphasize the weak enforcement of boundary conditions, which is an essential component of the IMGA framework.

2.1. Governing equations of incompressible flows

The Navier–Stokes equations of the incompressible flows posed on a time-dependent fluid domain Ω_t are written as

$$\rho_f \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} - \mathbf{f} \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{0} \,, \tag{1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \,, \tag{2}$$

where ρ_f, $\mathbf{u}$, and $\mathbf{f}$ are the fluid density, the flow velocity and the external force per unit mass, respectively. We use $\frac{\partial(\cdot)}{\partial t}$ to denote a partial time derivative taken with respect to a fixed spatial coordinate in the referential domain. The stress and strain-rate tensors are defined respectively as

$$\boldsymbol{\sigma}\left(\mathbf{u}, p\right) = -p\,\mathbf{I} + 2\mu\,\boldsymbol{\varepsilon}(\mathbf{u}) \,, \tag{3}$$

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}\right) \,, \tag{4}$$

where p is the pressure, $\mathbf{I}$ is an identity tensor, and μ is the dynamic viscosity. The problem (1)–(4) is accompanied by suitable boundary conditions, defined on the boundary of the fluid domain, $\Gamma_t = \Gamma_t^{\mathrm{D}} \cup \Gamma_t^{\mathrm{N}}$:

$$\mathbf{u} = \mathbf{u}_g \qquad \text{on } \Gamma_t^{\mathrm{D}} \,, \tag{5}$$

$$-p\,\mathbf{n} + 2\mu\,\boldsymbol{\varepsilon}(\mathbf{u})\,\mathbf{n} = \mathbf{h} \qquad \text{on } \Gamma_t^{\mathrm{N}} \,, \tag{6}$$

where $\mathbf{u}_g$ denotes the prescribed velocity at the Dirichlet boundary Γ_t^{D}, $\mathbf{h}$ is the traction vector at the Neumann boundary Γ_t^{N}, and $\mathbf{n}$ is the unit normal vector pointing in the wall-outward direction.

2.2. *Semi-discrete variational multiscale formulation*

Consider a collection of disjoint elements $\{\Omega_t^e\}$, $\cup_e \Omega_t^e \subset \mathbb{R}^d$, $d = 2, 3$, with closure covering the fluid domain: $\Omega_t \subset \cup_e \overline{\Omega_t^e}$. Note that Ω_t^e is not necessarily a subset of Ω_t because of the non-conforming fluid–structure interface. Let $\mathcal{V}_u^h$ and $\mathcal{V}_p^h$ be the discrete velocity and pressure spaces of functions supported on these elements. The strong problem (1)–(6) may be recast in a weak form and posed over these discrete spaces to produce the following semi-discrete problem: Find $\mathbf{u}^h \in \mathcal{V}_u^h$ and $p^h \in \mathcal{V}_p^h$ such that for all $\mathbf{w}^h \in \mathcal{V}_u^h$ and $q^h \in \mathcal{V}_p^h$:

$$B^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) - F^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}\right) = 0 \,. \tag{7}$$

The bilinear form B^{VMS} and the load vector F^{VMS} are given as

$$\begin{aligned} B^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) = & \int_{\Omega_t} \mathbf{w}^h \cdot \rho_f \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{u}^h\right) d\Omega + \int_{\Omega_t} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}\left(\mathbf{u}^h, p^h\right) d\Omega \\ & + \int_{\Omega_t} q^h\, \boldsymbol{\nabla} \cdot \mathbf{u}^h \, d\Omega \\ & - \sum_e \int_{\Omega_t^e \cap \Omega_t} \left(\rho_f \mathbf{u}^h \cdot \boldsymbol{\nabla}\mathbf{w}^h + \boldsymbol{\nabla} q^h\right) \cdot \mathbf{u}' \, d\Omega \end{aligned}$$

$$
\begin{aligned}
& - \sum_e \int_{\Omega_t^e \cap \Omega_t} p' \, \nabla \cdot \mathbf{w}^h \, d\Omega + \sum_e \int_{\Omega_t^e \cap \Omega_t} \rho_f \mathbf{w}^h \cdot (\mathbf{u}' \cdot \nabla \mathbf{u}^h) \, d\Omega \\
& - \sum_e \int_{\Omega_t^e \cap \Omega_t} \rho_f \nabla \mathbf{w}^h : (\mathbf{u}' \otimes \mathbf{u}') \; d\Omega, \qquad (8)
\end{aligned}
$$

and

$$
F^{\mathrm{VMS}} \left(\{ \mathbf{w}^h, q^h \} \right) = \int_{\Omega_t} \mathbf{w}^h \cdot \rho \, \mathbf{f} \, d\Omega + \int_{\Gamma_t^{\mathrm{N}}} \mathbf{w}^h \cdot \mathbf{h} \, d\Gamma \, , \qquad (9)
$$

where $\mathbf{u}'$ is defined as

$$
\mathbf{u}' = -\tau_{\mathrm{M}} \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f} - \frac{1}{\rho_f} \nabla \cdot \boldsymbol{\sigma} \left(\mathbf{u}^h, p^h \right) \right) \, , \qquad (10)
$$

and p' is given by

$$
p' = -\rho_f \tau_{\mathrm{C}} \, \nabla \cdot \mathbf{u}^h \, . \qquad (11)
$$

Eqs. (8)–(11) feature the residual-based VMS formulation of Navier–Stokes equations of incompressible flows [33]. The additional terms added onto the standard weak Galerkin form can be interpreted as a combination of streamline/upwind Petrov Galerkin (SUPG) stabilization and VMS large-eddy simulation of turbulence modeling [31–33, 39–42]. The stabilization parameters are designed as

$$
\begin{aligned}
\tau_{\mathrm{M}} &= \left(\frac{C_t}{\Delta t^2} + \mathbf{u} \cdot \mathbf{G} \, \mathbf{u} + C_I \, \nu^2 \, \mathbf{G} : \mathbf{G} \right)^{-1/2} \, , \qquad (12) \\
\tau_{\mathrm{C}} &= \left(\tau_{\mathrm{M}} \operatorname{tr} \mathbf{G} \right)^{-1} \, , \qquad (13)
\end{aligned}
$$

where Δt is the time-step size, C_I is a positive constant [43–45], $\nu = \mu / \rho_f$ is the fluid kinematic viscosity, $\mathbf{G}$ is the element metric tensor calculated by the mapping from the isoparametric element to its physical counterpart $\mathbf{x}(\xi)$:

$$
G_{ij} = \sum_{k=1}^{d} \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j} \, , \qquad (14)
$$

$\operatorname{tr} \mathbf{G}$ is the trace of $\mathbf{G}$, and the parameter C_t is typically set to 4 [33, 41].

2.3. Variationally consistent weak boundary conditions

The standard way of strongly imposing Dirichlet boundary conditions in Eq. (7) is not feasible in IMGA. We thus impose the Dirichlet boundary conditions weakly in the sense of Nitsche's method [25, 27–29, 34]. The semi-discrete problem becomes

$$
\begin{aligned}
& B^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}, \{\mathbf{u}^h, p^h\}\right) - F^{\mathrm{VMS}}\left(\{\mathbf{w}^h, q^h\}\right) \\
& - \int_{\Gamma_t^{\mathrm{D}}} \mathbf{w}^h \cdot \left(-p^h\,\mathbf{n} + 2\mu\,\boldsymbol{\varepsilon}(\mathbf{u}^h)\,\mathbf{n}\right)\, d\Gamma \\
& - \int_{\Gamma_t^{\mathrm{D}}} \left(2\mu\,\boldsymbol{\varepsilon}(\mathbf{w}^h)\,\mathbf{n} + q^h\,\mathbf{n}\right) \cdot \left(\mathbf{u}^h - \mathbf{u}_g\right)\, d\Gamma \\
& + \int_{\Gamma_t^{\mathrm{D}}} \tau^B \mathbf{w}^h \cdot \left(\mathbf{u}^h - \mathbf{u}_g\right)\, d\Gamma \;=\; 0\,.
\end{aligned}
\tag{15}
$$

The detailed interpretation of different terms can be found in Bazilevs and Hughes [34]. The parameter τ^B is a penalty-like stabilization parameter that helps to satisfy the no-slip condition on the boundary and ensure the stability (or coercivity) of the formulation. If τ^B is too large, the penalty term dominates the formulation, overshadowing the variational consistency that is responsible for the good performance of the method, and can result in an ill-conditioned stiffness matrix. If τ^B is too small, on the other hand, the solution is not stable, and the solver may confront convergence issues. Following the suggestion and numerical experience in Kamensky et al. [46] and Wu et al. [47], we define $\tau^B = \max\{C^B_{\mathrm{inv}}\rho_f h/\Delta t, C^B_{\mathrm{vis}}\mu/h\}$, where C^B's are positive dimensionless constants. This definition ensures that the penalty does not disappear in the inviscid limit and the formulation remains stable in the viscous limit. For simplicity, we choose h to be the size of the whole cut element. Our numerical experiments reveal that the proposed formulation of penalty preserves the accuracy very well.

2.4. Time discretization and iterative method

We employ the backward Euler finite difference scheme to complete the discretization in time:

$$
\rho_f \frac{\partial \mathbf{u}}{\partial t} = \rho_f \frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\Delta t} = \mathcal{L}(\mathbf{u}^n, p^n),
\tag{16}
$$

where the operator $\mathcal{L}(\mathbf{u}^n, p^n)$ represents all the other terms except the time-dependent term in Eq. (1) evaluated at the current time step n. Δt may be selected to follow CFL condition. We implement the parallelized moving immersogeometric method within our in-house parallel finite element framework. The domain decomposition is achieved via ParMETIS [48]. The (non)linear system is solved via PETSc [49]. Specifically, we utilize the SNES construct (line search quasi-Newton), which uses the KSP construct (BCGS solver), for the linearized system. For some time steps, the SNES solver may not achieve a prescribed tolerance of the norm of absolute/relative residuals. For these time steps, we apply LU preconditioner to the sub-blocks of the matrix.

3. Implementation of moving B-rep

3.1. Modeling the rigid body motion

The objects are modeled as rigid bodies. We denote the velocities associated with the objects as $\mathbf{v}_i$, with the subscript i indicating the i^{th} immersed object. The motion of the objects can be described in the Lagrangian reference frame by

$$\frac{d\mathbf{x}_i^{\mathrm{c}}}{dt} = \mathbf{v}_i^{\mathrm{c}}, \qquad \frac{d\mathbf{v}_i^{\mathrm{c}}}{dt} = \frac{\mathbf{F}_i}{m_i}, \tag{17}$$

$$\frac{d\boldsymbol{\theta}_i^{\mathrm{c}}}{dt} = \boldsymbol{\omega}_i^{\mathrm{c}}, \qquad \frac{d\left(\mathbf{J}_i\boldsymbol{\omega}_i^{\mathrm{c}}\right)}{dt} = \mathbf{T}_i, \tag{18}$$

where $\mathbf{x}_i^{\mathrm{c}}$ and $\boldsymbol{\theta}_i^{\mathrm{c}}$ are the linear and angular locations of the centroid of the i^{th} object, $\mathbf{v}_i^{\mathrm{c}}$ and $\boldsymbol{\omega}_i^{\mathrm{c}}$ are the linear and angular velocities of the centroid of the i^{th} object, $\mathbf{F}_i$ and $\mathbf{T}_i$ are the overall force and torque integrated over the surface of the i^{th} object, and m_i and $\mathbf{J}_i$[1] are the mass and moment of inertia tensor of the i^{th} object. $\mathbf{F}_i$ and $\mathbf{T}_i$ are computed from the solution of the fluid field, and defined as follows:

$$\mathbf{F}_i = \oint_{(\Gamma_t^{\mathrm{D}})_i} \boldsymbol{\sigma}_i(\mathbf{u},p)\,\mathbf{n}_i d\Gamma, \qquad \mathbf{T}_i = \oint_{(\Gamma_t^{\mathrm{D}})_i} \mathbf{r}_i \times (\boldsymbol{\sigma}_i(\mathbf{u},p)\,\mathbf{n}_i)\,d\Gamma, \tag{19}$$

where $\left(\Gamma_t^{\mathrm{D}}\right)_i$ is the boundary of the i^{th} object, $\boldsymbol{\sigma}_i(\mathbf{u},p)$ is the stress tensor acting on its surface, $\mathbf{r}_i$ is the distance vector from its centroid to its surface, and $\mathbf{n}_i$ is the unit normal vector of its surface pointing outward. The coordinates $\mathbf{x}_i$ and velocity $\mathbf{v}_i$ of the surface points of the i^{th} object are computed as

$$\mathbf{x}_i = \mathbf{x}_i^{\mathrm{c}} + \mathbf{r}_i, \qquad \mathbf{v}_i = \mathbf{v}_i^{\mathrm{c}} + \boldsymbol{\omega}_i^{\mathrm{c}} \times \mathbf{r}_i. \tag{20}$$

$\mathbf{v}_i$ is imposed as the prescribed velocity at the Dirichlet boundary of the i^{th} object. Assuming the overall force and torque acting on the i^{th} object surface are constant during one time step, we approximate the object motion as follows

$$\frac{(\mathbf{x}_i^{\mathrm{c}})^{n+1} - (\mathbf{x}_i^{\mathrm{c}})^{n}}{\Delta t} = \frac{(\mathbf{v}_i^{\mathrm{c}})^{n+1} + (\mathbf{v}_i^{\mathrm{c}})^{n}}{2}, \qquad \frac{(\mathbf{v}_i^{\mathrm{c}})^{n+1} - (\mathbf{v}_i^{\mathrm{c}})^{n}}{\Delta t} = \frac{(\mathbf{F}_i)^n}{m_i}, \tag{21}$$

$$\frac{(\boldsymbol{\theta}_i^{\mathrm{c}})^{n+1} - (\boldsymbol{\theta}_i^{\mathrm{c}})^{n}}{\Delta t} = \frac{(\boldsymbol{\omega}_i^{\mathrm{c}})^{n+1} + (\boldsymbol{\omega}_i^{\mathrm{c}})^{n}}{2}, \qquad \frac{(\boldsymbol{\omega}_i^{\mathrm{c}})^{n+1} - (\boldsymbol{\omega}_i^{\mathrm{c}})^{n}}{\Delta t} = \frac{(\mathbf{T}_i)^n}{J_i}. \tag{22}$$

[1] For particle shapes (2D cylinders or 3D spheres) considered in this work, the tensor $\mathbf{J}_i$ is reduced to a constant scalar J_i; we use J_i for simplicity in subsequent content.

Following the conservative traction definition for weakly imposed boundary conditions [50], we compute $(\mathbf{F}_i)^n$ and $(\mathbf{T}_i)^n$ as

$$(\mathbf{F}_i)^n = \int_{(\Gamma_t^{\mathrm{D}})_i} \boldsymbol{\sigma}_i(\mathbf{u}^n, p^n)\mathbf{n}_i d\Gamma - \int_{(\Gamma_t^{\mathrm{D}})_i} \tau^B(\mathbf{u}^n - \mathbf{v}_i^n) d\Gamma, \tag{23}$$

$$(\mathbf{T}_i)^n = \int_{(\Gamma_t^{\mathrm{D}})_i} \mathbf{r}_i \times (\boldsymbol{\sigma}_i(\mathbf{u}^n, p^n)\mathbf{n}_i)\, d\Gamma - \int_{(\Gamma_t^{\mathrm{D}})_i} \mathbf{r}_i \times \tau^B(\mathbf{u}^n - \mathbf{v}_i^n) d\Gamma. \tag{24}$$

Each object velocity is evaluated using an explicit forward Euler scheme, which requires small Δt to ensure accuracy and stability.[2] Each object location is then updated by the average velocity, which results in a more stable and accurate position evolution. Note that Eqs. (21) and (22) are essentially Newmark-beta method [51] with γ and β set to zero.

3.2. In-out test

The in-out test essentially returns a Boolean value depending on if a certain point is inside or outside the object. There are a variety of approaches for performing an in-out test, including ray-tracing algorithms. In this work, we only consider analytical shapes for our objects and can therefore determine the in-out result of a query point simply by using the geometry equation of the object. Note that the equation of the object geometry will change every time step because of the linear and angular motion of the object. We transform the query point into the Lagrangian reference frame of the object in the in-out test so that we can retain the consistency of the geometry equation to avoid its computation in Euler reference frame at every time step during the movement of the object.

3.3. Treatment of freshly-cleared nodes

Consider the nodes in the background mesh that are covered by an object at current time step. At the next time step, some of these nodes may no longer be covered by the object due to the motion of the object. We call these nodes “freshly-cleared”. Note that we have no fluid history of the flow field recorded on those nodes. In practice, most of the freshly-cleared nodes are on the intersected elements because Δt is small and the distance that an object can travel in one time step is also small. Those nodes may have solution values even if they are inside the object; however, the solutions on those nodes are for satisfying the weakly imposed boundary conditions on the object surface and achieving the accuracy of the solution on their neighbor nodes inside the fluid domain. Therefore, the values on the freshly-cleared nodes are not the correct solutions of the fluid field.

To obtain a reasonable solution on those nodes and complete the fluid field solution, we interpolate using the solution of their neighbor nodes in the fluid domain and boundary values of

[2] This is equivalent to employing a very small Courant number from the CFL condition.

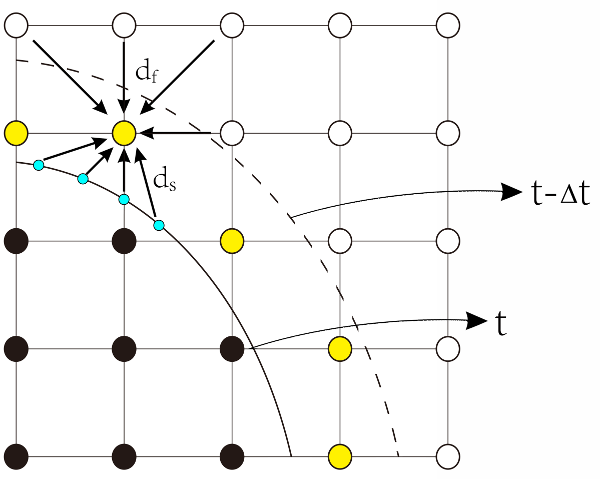


Figure 1: Schematic of the interpolation of the freshly-cleared nodes.

nearby points on the object surface. This provides a good initial guess to start the solution process at next time step. We denote the union of the variables in fluid field as $\mathbf{U} = \{\mathbf{u}, p\}$, and the union of the variables on the i^{th} object surface as $\mathbf{V}_i = \{\mathbf{v}_i, p_i\}$. The union of interpolated variables on the freshly-cleared nodes corresponding to the i^{th} object is denoted as $\mathbf{V}_i^{\text{fc}}$, which can be written as

$$\mathbf{V}_i^{\text{fc}} = \frac{\sum\limits_{k=1}^{N_\text{f}} \frac{1}{(d_\text{f})_k} \mathbf{U}_k + \sum\limits_{k=1}^{N_\text{s}} \frac{1}{(d_\text{s})_k} (\mathbf{V}_i)_k}{\sum\limits_{k=1}^{N_\text{f}} \frac{1}{(d_\text{f})_k} + \sum\limits_{k=1}^{N_\text{s}} \frac{1}{(d_\text{s})_k}}, \tag{25}$$

where d_f and d_s are the distances of neighbor nodes in fluid domain and nearby points on the object surface to the freshly-cleared nodes, respectively, and N_f and N_s are their respective numbers of nodes. All the nodes in fluid domain and points on object surface that are used to interpolate at one particular freshly-cleared node are set to be in the region centered at the freshly-cleared node with a radius of $O(h)$.[3] A schematic of this is shown in Figure 1.

3.4. *Workflow of the framework*

We first solve for the fluid field solution $\mathbf{U}^n$ at current time step n, provided the updated coordinates of surface points of each object, $\mathbf{x}_i^n$, and the updated boundary velocities on surface points of each object, $\mathbf{v}_i^n$, and fluid field solution $\mathbf{U}^{n-1}$ from last time step $n-1$. We evaluate the overall force and torque over each object surface, $\mathbf{F}_i^n$ and $\mathbf{T}_i^n$, based on $\mathbf{x}_i^n$, $\mathbf{v}_i^n$, and $\mathbf{U}^n$. We then update the coordinates of surface points of each object, $\mathbf{x}_i^{n+1}$, at next time step $n+1$. We check if there are any freshly-cleared nodes in the fluid field, and interpolate for those nodes using $\mathbf{U}^n$ at neighbor

[3] We also ensure that N_f and N_s are comparable.

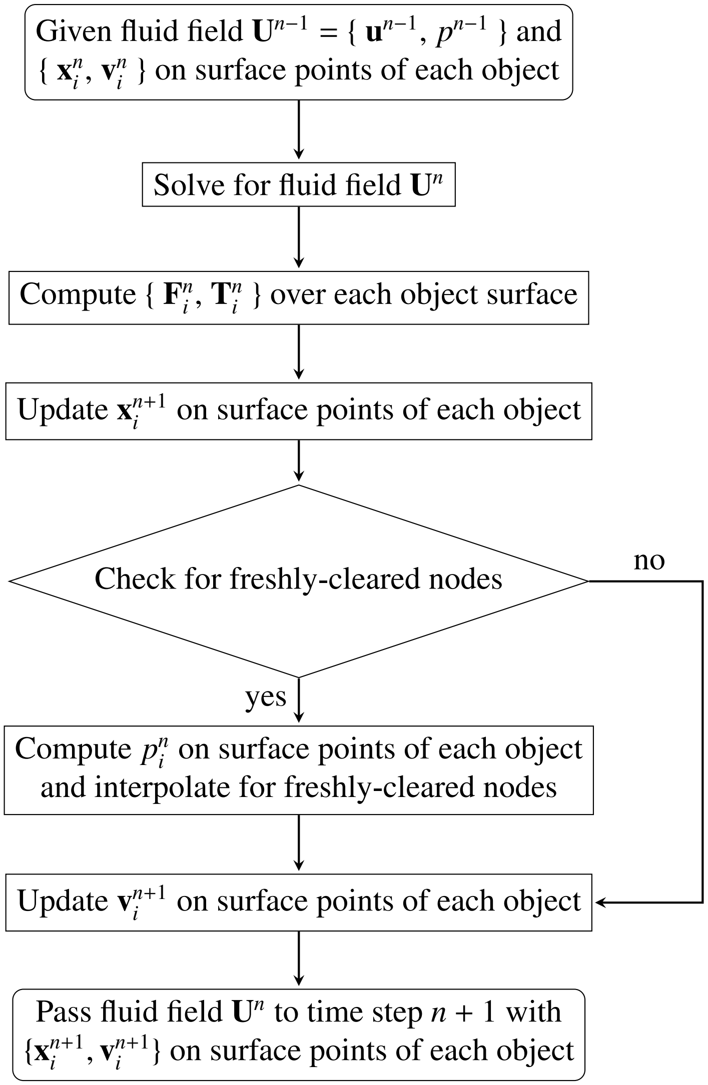

Given fluid field $\mathbf{U}^{n-1} = \{ \mathbf{u}^{n-1}, p^{n-1} \}$ and $\{ \mathbf{x}_i^n, \mathbf{v}_i^n \}$ on surface points of each object
Solve for fluid field $\mathbf{U}^n$
Compute $\{ \mathbf{F}_i^n, \mathbf{T}_i^n \}$ over each object surface
Update $\mathbf{x}_i^{n+1}$ on surface points of each object
Check for freshly-cleared nodes
no
yes
Compute p_i^n on surface points of each object and interpolate for freshly-cleared nodes
Update $\mathbf{v}_i^{n+1}$ on surface points of each object
Pass fluid field $\mathbf{U}^n$ to time step $n + 1$ with $\{\mathbf{x}_i^{n+1}, \mathbf{v}_i^{n+1}\}$ on surface points of each object

Figure 2: Flowchart of the process of moving IMGA.

nodes in fluid field, and $\mathbf{V}_i^n$ at nearby points on corresponding object surface. We need to compute the boundary values of pressure on surface points of each object, p_i^n, before the interpolation of pressure. Finally, we update the velocities on surface points of each object, $\mathbf{v}_i^{n+1}$, as boundary conditions at time step $n + 1$, and pass the fluid field $\mathbf{U}^n$, which is interpolated for freshly-cleared nodes, to solve for $\mathbf{U}^{n+1}$. This procedure is shown in Figure 2.

3.5. Non-dimensionalization

We non-dimensionalize the variables and parameters as follows:

$$\mathbf{x}^* = \frac{\mathbf{x}}{L_0}, \qquad \mathbf{u}^* = \frac{\mathbf{u}}{u_0}, \qquad t^* = \frac{t}{L_0/u_0}, \qquad \rho^* = \frac{\rho}{\rho_f},$$

with

$$p^* = \frac{p}{u_0\mu/L_0}, \qquad \mathbf{g}^* = \frac{\mathbf{g}}{u_0\mu/(\rho_f L_0^2)} \quad (\text{viscous scaling}), \tag{26}$$

or

$$p^* = \frac{p}{\rho_f u_0^2}, \qquad \mathbf{g}^* = \frac{\mathbf{g}}{u_0^2/L_0} \quad (\text{inertial scaling}), \tag{27}$$

where L_0 is the characteristic length, u_0 is the characteristic velocity depending on the problems of interest, and $\mathbf{g}$ is the gravitational acceleration. We employ viscous scaling when Reynolds number $Re = \frac{\rho_f u_0 L_0}{\mu} \ll 1$, and inertial scaling when Re is moderate.

4. Verification and validation

In this section, we illustrate the moving IMGA framework using several benchmark problems and results.

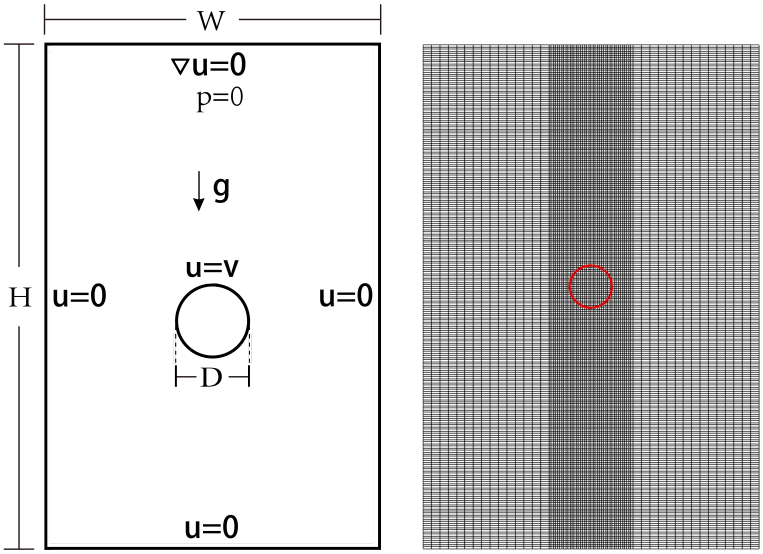
4.1. Free falling cylinder with low Reynolds number (2D)

A free dropping cylinder with diameter D and density ρ_s will reach an equilibrium state with a constant terminal velocity V_T in a sufficiently tall channel (channel height H) filled with a viscous fluid. Drag force F_{D}, buoyancy F_{b}, and gravitational force F_{g} will eventually reach an equilibrium state so that the net force is zero, and the cylinder will move with the constant terminal velocity.

For creeping flow with low Re number ($Re \ll 1$), we have an analytical solution for the terminal velocity of a cylinder with an infinite length [52]:

$$V_T = \frac{\left(\rho_s - \rho_f\right) g D^2}{16\mu}\left(\ln\left(\frac{W}{D}\right) - 0.9157 + 1.7244\left(\frac{D}{W}\right)^2 - 1.7302\left(\frac{D}{W}\right)^4\right), \tag{28}$$

where W is the channel width, and g is the magnitude of gravitational acceleration. The infinite cylinder length allows us to perform 2D simulations to verify our moving immersogeometric framework. A mesh convergence study is performed to show that the simulated terminal velocity converges to the analytical solution with increasing mesh density. We set $\rho_f = 10^3$ kg/m^3, $\rho_s = 1.25 \times 10^3$ kg/m^3, $g = 9.81$ m/s^2, $\mu = 0.5$ kg/(m·s), $D = 5 \times 10^{-3}$ m, $W = 0.04$ m and $H = 0.06$ m. Using these parameters, we get a terminal velocity $V_T = 9.12 \times 10^{-3}$ m/s, and $Re = 0.0912$ based on V_T and D. Since $Re \ll 1$, we non-dimensionalize this problem using a viscous scaling. Initial conditions are set as zero velocity and pressure in the whole fluid domain. No-slip boundary conditions are imposed on the lateral and bottom walls, and the traction-free boundary condition is imposed on the top wall. The problem setup is shown in Figure 3a.



(a) Problem setup.

(b) Cluster mesh.

Figure 3: Free falling 2D cylinder with low Reynolds number: Problem setup and cluster mesh.

The intersection region of the fluid domain and the object surface needs more refinement to capture the immersed boundary and resolve the no-slip boundary condition. In this case, knowing that the trajectory is a straight line, we refine the mesh all the way along the trajectory to avoid any remeshing and interpolation between two meshes. Hsu et al. [35] suggested that the ratio between the element size of the immersed surface and the element size of the background mesh that intersects the immersed surface needs to be at most $1/2$. We choose the ratio to be $1/2$ for all the simulations in this paper. In addition, we also keep the ratio between the sizes of the largest and smallest background elements around 4 as a reasonable maximum aspect ratio of any background elements in the fluid domain. The dimensionless element size across the fluid–structure interface is set to 0.1, 0.05, and 0.025 for three different mesh densities, which result in a total element number of 35×120, 70×240, and 140×480, respectively. An example of the fixed cluster mesh (70×240) is shown in Figure 3b. The dimensionless time step Δt is set to 1×10^{-3}.

The result of the mesh convergence study with the analytical terminal velocity is presented in Figure 4. We can clearly see that the numerically computed terminal velocity converges to the analytical result with increasing mesh density. The relative error with the analytical terminal velocity is 3.84%, 1.26%, and 0.1% for the mesh of 35×120, 70×240, and 140×480, respectively. The mesh density of 70×240 (dimensionless element size across the interface to be 0.05) produces a fairly accurate result. We also present the velocity magnitude contour along with the streamlines at time $t = 1$ after the cylinder reaches the terminal velocity in Figure 5. We can see two large vortices caused by the downward motion of the cylinder as physically expected in this case. Similar velocity magnitude contour and streamlines are also observed in Casquero et al. [24].

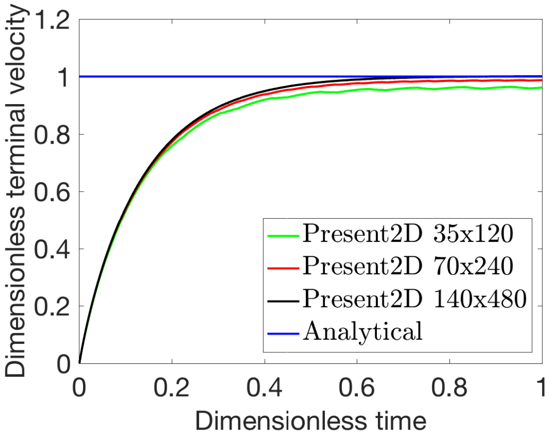


Figure 4: Free falling 2D cylinder with low Reynolds number: Mesh convergence result of the cylinder terminal velocity compared with the analytical solution.

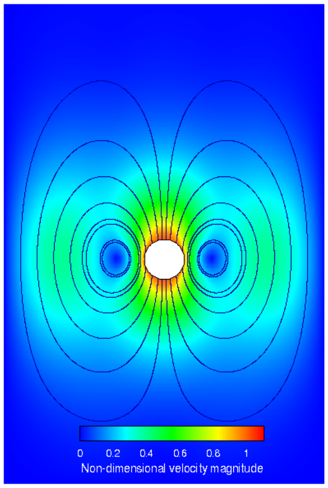
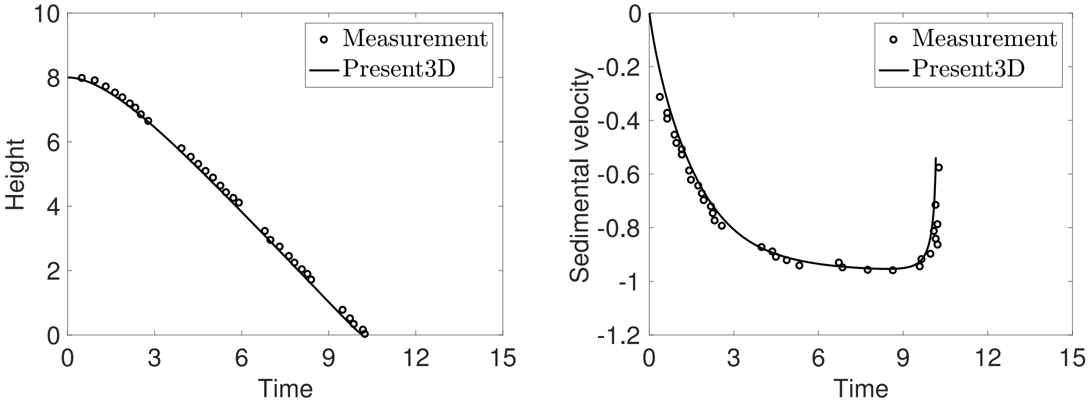


Figure 5: Free falling 2D cylinder with low Reynolds number: Fluid velocity magnitude contour and streamlines at $t = 1$.

4.2. Free falling sphere with moderate Reynolds number (3D)

The second benchmark problem we consider is a free falling 3D sphere in fluids with moderate Reynolds number. In this case, we compare the results from the moving immersogeometric framework with experimental benchmark results. We employ a similar case as in the last section, but replace the 2D cylinder with a 3D sphere, and simulate a free dropping sphere to compare with the experimental results in Ten Cate et al. [53]. The size of the domain is 0.1 m × 0.16 m × 0.1 m. The diameter of the sphere is $D = 0.015$ m, and it is released at rest at the height where its bottom apex is 0.12 m away from the bottom wall. The working fluid has a density $\rho_f = 960$ kg/m^3, and a dynamic viscosity $\mu = 0.058$ kg/(m·s). The density of the sphere is $\rho_s = 1120$ kg/m^3. This results



(a) Comparison of the sphere height.

(b) Comparison of the sphere sedimental velocity.

Figure 6: Free falling 3D sphere with moderate Reynolds number: Comparisons of the sphere trajectory and velocity.

in a Reynolds number $Re = 31.9$ based on $V_0 = 0.128$ m/s [53] and D. Since Re is moderate in this case, we non-dimensionalize it using inertial scaling. The dimensionless element size across the interface is set to 0.05, and the dimensionless Δt is set to 0.01. The same initial conditions and boundary conditions as in the last section are imposed.

We show the comparison of the dimensionless height of the sphere, as well as the dimensionless sedimental velocity of the sphere in Figure 6. We can see a very good agreement in both the trajectory and the sedimental velocity between our numerical results and the experimental results. In addition, we can observe that sedimental velocity magnitude drastically decreases as the sphere approaches the bottom wall. This is because of the so-called wall-effect force that appears when the sphere moves close towards the bottom boundary. Note that this wall-effect force is an important force in particle focusing problems, and we are able to capture its effect very well in this example. We also show the contours of the velocity magnitude at different heights (7.15, 4.88 and 0.41) in Figure 7. These results qualitatively match the contours in experimental studies [53].

4.3. Neutral buoyant circular particle focusing in a straight channel

In this case, we deploy our framework to model more complex flow physics. A circular particle moving in a straight channel will focus at unique locations in the channel cross-section. This is the so-called inertial focusing effect [54]. The motion of the particle is driven by a fully developed flow in the channel. The fluid velocity profile is disturbed locally by the particle, which leads to a more complicated velocity gradient (i.e. more complicated viscous force) on the particle surface, and therefore results in a more intricate interaction between the particle and fluid. According to Amini et al. [55], there are two dominant forces in this case – shear-gradient force and wall-effect force. The particle will focus to a unique position where these forces balance out. Feng et al. [56] showed

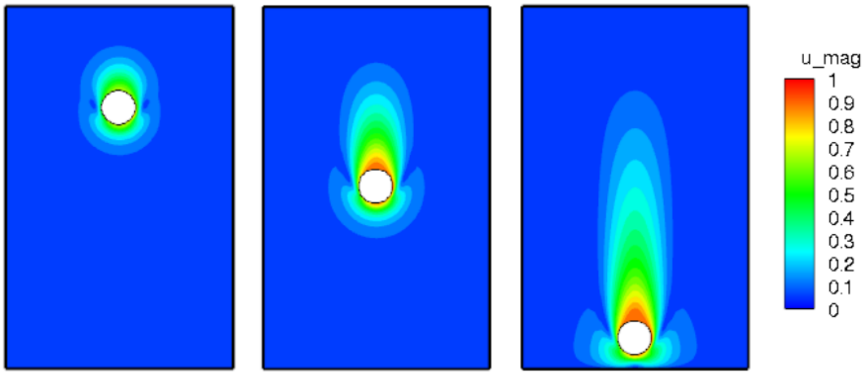
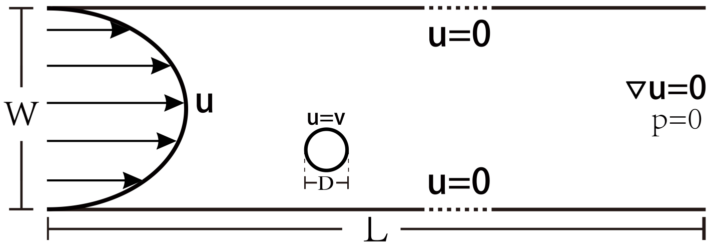


Figure 7: Free falling 3D sphere with moderate Reynolds number: Fluid velocity magnitude contours at different heights. The heights are 7.15, 4.88 and 0.41 (from left to right.)

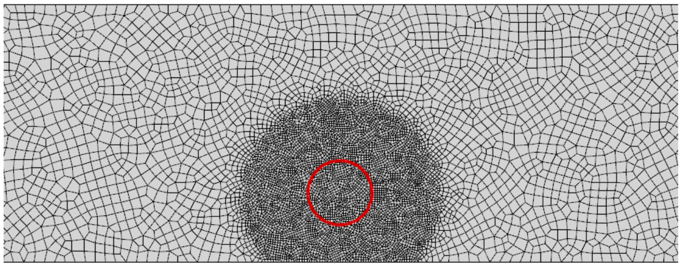
that a particle with diameter $D = 0.25$, released at a lateral position $y < 0.5$, in a channel at $Re = 40$ calculated based on channel width and maximum inlet velocity (parabolic profile), will focus at the lateral position $y = 0.272$. To observe this focusing phenomena, a sufficiently long channel is needed. An estimate of relation of the channel length and particle hydraulic diameter for focusing can be found in Di Carlo [57]. We choose a 2D channel with dimensionless sizes of length and width to be $L \times W = 40 \times 1$, and release the particle at the lateral position $y = 0.225$, with an initial dimensionless horizontal velocity $V_0 = 0.66$ [56]. The dimensionless element size across the interface is set to 0.05, and the dimensionless Δt is set to 5×10^{-4}. The initial conditions of the fluid are set as fully developed flow profiles of velocity and pressure in the absence of the particle. The boundary conditions are parabolic velocity profile at the inlet, and traction-free at the outlet. No-slip boundary condition is imposed on channel walls.

We perform remeshing and utilize the meshing framework Gmsh [58]. To minimize the number of remeshes required, we refine a large bounding circle that contains the particle. The radius of the bounding circle is set to be three times of the radius of the particle. This allows particle motion for several hundreds of time steps before remeshing. We remesh and interpolate when the particle is within a length of particle radius from the refinement boundary, and reset the bounding sphere center to be the particle centroid after remeshing. The problem setup and an unstructured mesh example are shown in Figure 8.

We plot the particle trajectory in Figure 9. The criterion we employ to verify inertial focusing is that the lateral position remains constant across 5 dimensionless seconds (out of totally simulated 32.8 dimensionless seconds). Figure 9 clearly illustrates that the framework is able to capture the focusing phenomena. Our result shows a focusing at the lateral position $y = 0.280$, which is slightly higher than the result in Feng et al. [56], with a relative error of 0.8% scaled by the channel width.



(a) Problem setup.



(b) A mesh example.

Figure 8: Neutral buoyant circular particle focusing in a straight channel: Problem setup and a mesh example.

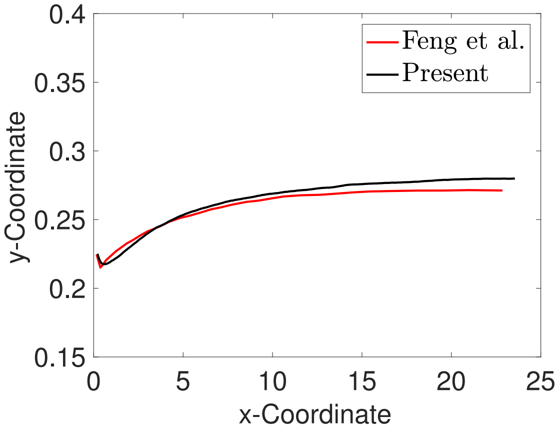
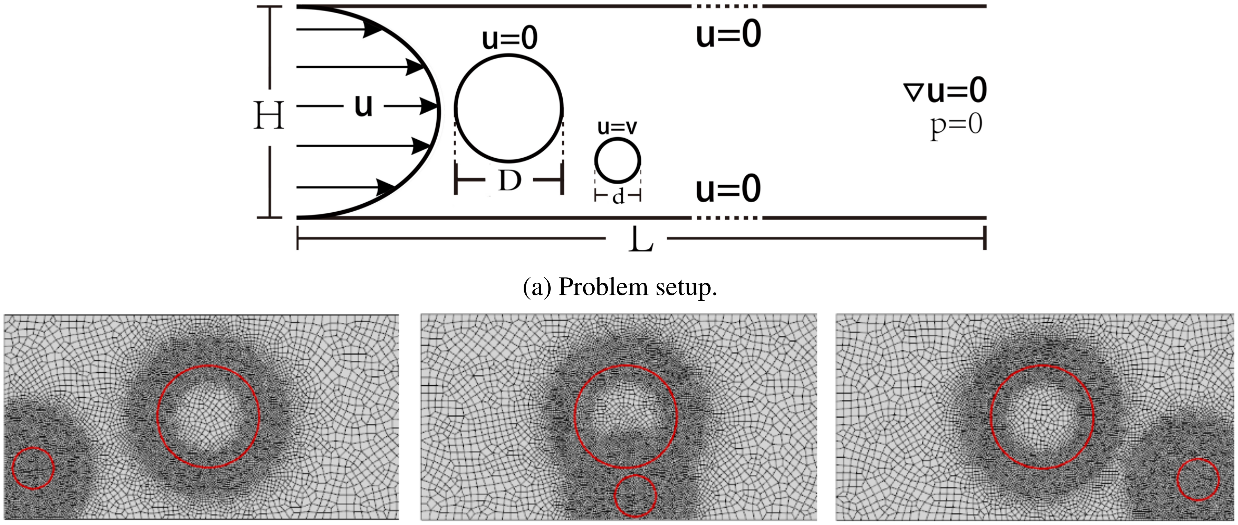


Figure 9: Neutral buoyant circular particle focusing in a straight channel: Particle trajectory.

4.4. Circular particle focusing in a straight channel with pillar

We simulate a 2D channel with a circular pillar as the obstruction in the channel. This is the canonical example of inertial interaction of particles in obstructed flow [59]. We define the domain with the origin at the bottom left corner of the channel. Buoyancy is considered in this case. All of the following parameters are in dimensionless form. The size of the channel ($L \times H$) is 30×5, where L is the length and H is the height of the channel. The diameter of the pillar is $D = 2.5$, and it is centered at $(5, 2.5)$. The diameter of the particle is $d = 1$, and it is released at $(0.6, 1.25)$ with zero initial velocity. The density ratio between particle and fluid is 1.01.



(a) Problem setup.

(b) The unstructured (non-body-fitted) meshes at different times.

Figure 10: Circular particle focusing in a straight channel with pillar: Problem setup and moving meshes at different times.

The Reynolds number is $Re = 20$, based on D and maximum inlet velocity (parabolic profile). Cut element size is again set to 0.05, and $\Delta t = 2 \times 10^{-3}$. Inertial scaling of non-dimensionalization is used. The initial conditions are fully developed flow profiles of velocity and pressure in the absence of the particle and obstruction. The boundary conditions are parabolic velocity at the inlet, and traction-free at the outlet. In this case, we have two objects – the moving particle, and the static pillar. We consider the immersed method for both objects. We use Gmsh to refine a ring-shaped region with a given center, radius, and band width to track the static pillar–fluid interface. The case setup and unstructured moving meshes at different times are shown in Figure 10.

For this case, since we lack other numerical results to verify our framework, we simulate an equivalent case using ANSYS Fluent 16.1 [60] with the body-fitted method. The detailed problem setup in Fluent is provided in the Appendix. We perform a mesh convergence study for the particle trajectory as the metric for both methods. The convergence result of moving IMGA is shown in Figure 11. Since the mesh changes during the particle motion, we report time-weighted average mesh densities for the moving IMGA. The three configurations that we evaluate consist of increasing average numbers of elements of 7332, 14608 and 28268. We notice that all three mesh densities produce similar results for the moving IMGA, which indicates the coarsest mesh would be able to reasonably predict the particle motion. In contrast, it is observed that using Fluent with body-fitted method, the particle trajectory is substantially more sensitive to the mesh density. We consider three configurations with increasing average numbers of elements of 7414, 15027 and 28182, with only the latter two configurations producing converging results (see Figure 15 in Appendix that shows this behavior). To ensure a fair assessment of computational efficiency, we

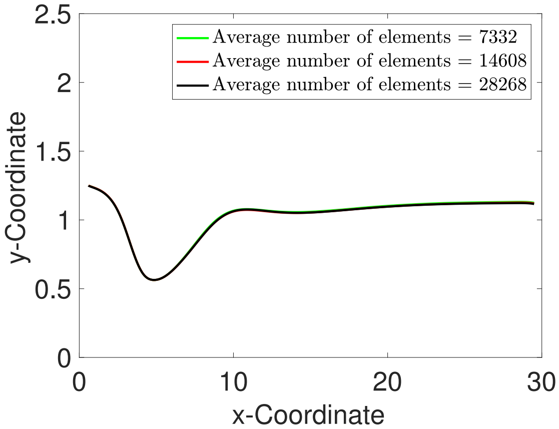


Figure 11: Circular particle focusing in a straight channel with pillar: Mesh convergence result of the particle trajectory for moving IMGA.

Table 1: Circular particle focusing in a straight channel with pillar: Comparison of computational efforts.

	Moving IMGA	Body-fitted Fluent
Average number of elements	28268	28182
Number of cores	16	16
Total simulation time (min)	199	1446

utilize the finest meshes in both simulations with the same number of processors to compare the computation time. The details of this comparison are provided in Table 1. Our moving IMGA is about 8 times faster than body-fitted Fluent at the same mesh density. This is anecdotal evidence that the moving IMGA approach can produce good results at a substantially reduced computational cost.

We present the particle trajectory and velocity comparisons with the finest meshes in both simulations in Figures 12 and 13. The final vertical positions from present IMGA simulation and Fluent are 1.12 and 1.08, respectively. The relative error between them is 0.8% of the channel height. In addition, our framework shows the horizontal velocity eventually reaches a steady value of 0.646, which is 92.9 % of the undisturbed fluid velocity at the same height, and the vertical velocity is nearly zero. A more detailed comparison is shown in Figure 14. We plot the velocity contours and streamlines at three locations $(2.59, 1.05)$, $(4.95, 0.566)$, and $(7.17, 0.776)$. We can see the particle starts to disturb the symmetry of the downstream wake region when it passes below the pillar, and completely distorts the wake region during its subsequent movement. The lower circulation in the wake region almost disappears.

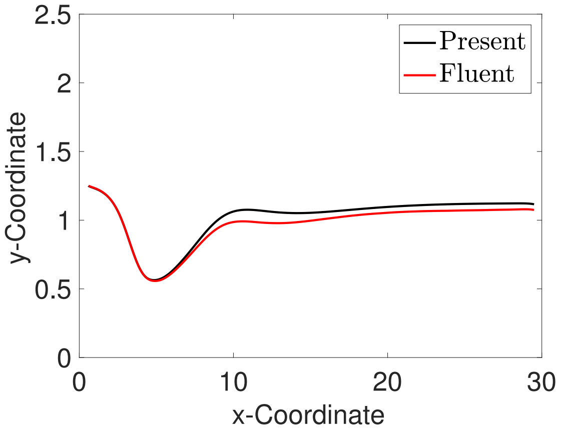
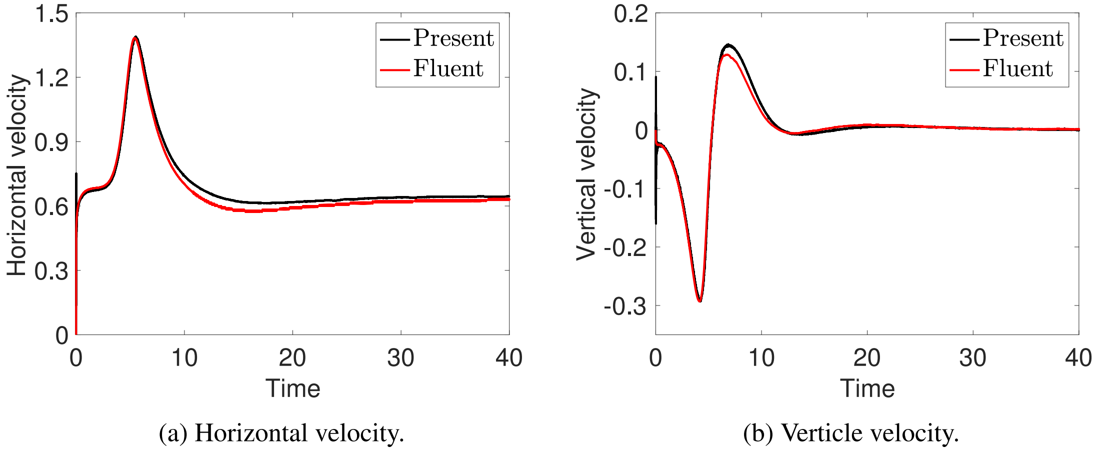


Figure 12: Circular particle focusing in a straight channel with pillar: Comparison of particle trajectory.



(a) Horizontal velocity.

(b) Verticle velocity.

Figure 13: Circular particle focusing in a straight channel with pillar: Comparison of particle velocities.

5. Conclusions and future work

We deployed the immersogeometric approach for tracking moving objects. We considered a numerical treatment for the freshly-cleared nodes (i.e. background mesh nodes that are currently in the fluid domain, but were inside the object at the last time step). We assessed the accuracy of the method by analyzing object motion in different flow structures including free dropping objects in viscous fluids and particle focusing in unobstructed and obstructed micro-channels. We showed that key quantities of interest are in very good agreement with analytical, numerical and experimental solutions. We also showed a much better computational efficiency of our framework than current commercial codes using adaptive body-fitted approaches. We conclude that the moving IMGA is a promising and viable approach for exploring the inertial focusing of particles in

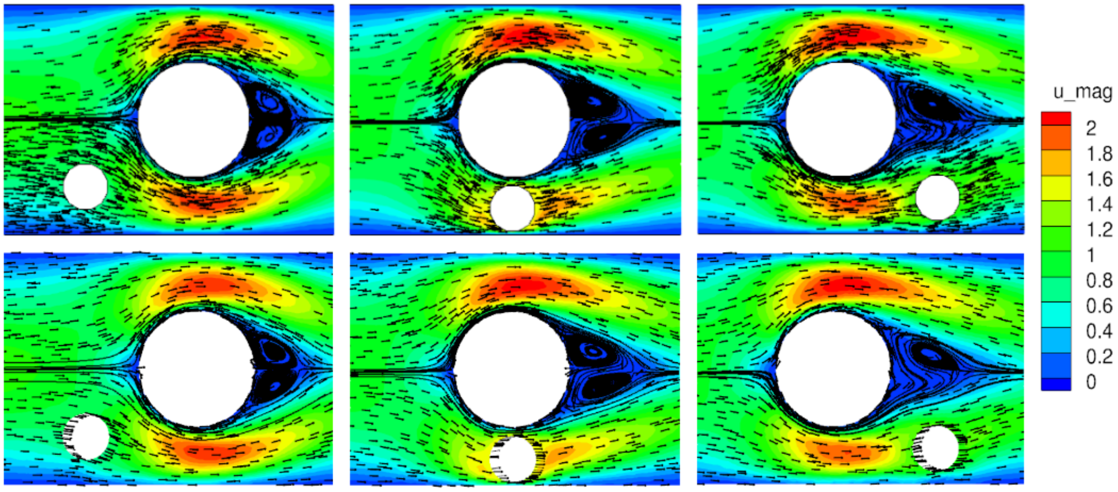


Figure 14: Circular particle focusing in a straight channel with pillar: Comparisons of fluid velocity magnitude contours and streamlines. (upper) Moving IMGA. (lower) Body-fitted Fluent.

complex micro-channels with obstructions. The interaction of the particle with the obstruction as mediated by the fluid, and the scaling behavior of this interaction with increasing particle diameter and Reynolds number will be the focus of subsequent studies using this framework.

Acknowledgment

We acknowledge funding from NSF 1149365, and computing support from XSEDE and Iowa State University. M.-C. Hsu was partially supported by NIH/NHLBI Grants R01HL129077 and R01HL142504.

Appendix

The simulation of particle moving in an obstructed (pillar) channel with body-fitted method was performed on ANSYS Fluent 16.1 using the built-in Dynamic Mesh (DM) module. The Navier–Stokes equations are solved by the default solver using the SIMPLE algorithm. The particle evolution is calculated in the *6 Degree-of-Freedom (6DOF)* solver for rigid-bodies. An implicit mesh update feature is enabled – the dynamic mesh is updated during the course (convergence) of the solution of flow fields in one time-step, which results in a stable mesh update and better tracking of the particle positions. The remeshing is automatically determined by the solver based on the worst element skewness due to the particle motion and controlled by given remeshing parameters. The remeshing parameters are guided by the default (minimum/maximum) mesh sizes from the initial undeformed domain. In addition, the spring constant factor is set to 0.5 using the spring-based smoothing method for the dynamic mesh. The flow fields are projected from the previous mesh to the current mesh upon remeshing. The flow fields are solved over a dimensional domain, 0.3

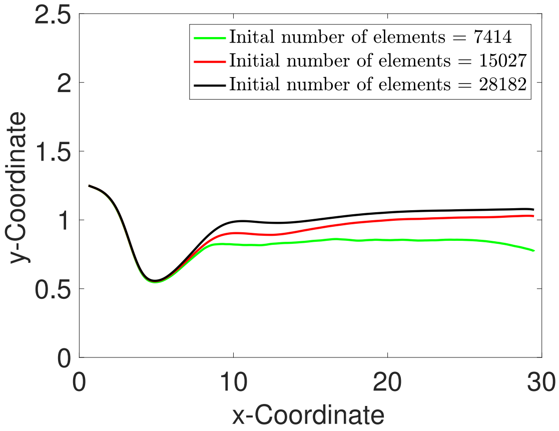


Figure 15: Circular particle focusing in a straight channel with pillar: Mesh convergence result of the particle trajectory using Fluent.

m ×0.05 m, where the fluid properties of density and dynamic viscosity measure, 1000 kg/m^3, and 0.1 kg/(m·s), respectively. The particle is released at $x = 0.006$ m, and, $y = 0.0125$ m, with the origin set at bottom left corner. The particle mass, and moment-of-inertia about the z-axis are externally supplied through a User-Defined Function (UDF) using the `DEFINE_SDOF_PROPERTIES` macro, and measure, 0.0793 kg, and 1×10^{-6} kg·m^2. The buoyant-force on the particle is imposed as an external force, `prop[SDOF_LOAD_F_Y]` = 0.770 N. The weight of the particle is computed by enabling the gravity option under the 6DOF solver with the value of 9.81 m/s^2. The initial conditions are taken as fully developed profiles of velocity and pressure, in the absence of the particle, and pillar, whose maximum values measure, 0.2 m/s, and 19.2 Pa. The boundary conditions include parabolic-velocity inlet, and zero-pressure outlet.

References

[1] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[2] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1-3):689–723, 1982.

[3] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodriguez-Ferran. Arbitrary Lagrangian–Eulerian methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.

[4] T. E. Tezduyar, M. Behr, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure:

I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, 94(3):339–351, 1992.

[5] T. E. Tezduyar, M. Behr, S. Mittal, and J. Liou. A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space–time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, 94(3):353–371, 1992.

[6] A. A. Johnson and T. E. Tezduyar. Parallel computation of incompressible flows with complex geometries. *International Journal for Numerical Methods in Fluids*, 24:1321–1340, 1997.

[7] A. A. Johnson and T. E. Tezduyar. 3D simulation of fluid-particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering*, 145:301–321, 1997.

[8] A. A. Johnson and T. E. Tezduyar. Advanced mesh generation and update methods for 3D flow simulations. *Computational Mechanics*, 23:130–143, 1999.

[9] K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. Space–time interface-tracking with topology change (ST-TC). *Computational Mechanics*, 54:955–971, 2013.

[10] K. Takizawa, T. E. Tezduyar, A. Buscher, and S. Asada. Space–time fluid mechanics computation of heart valve models. *Computational Mechanics*, 54:973–986, 2014.

[11] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 2002.

[12] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[13] D. M. McQueen and C. S. Peskin. Computer-assisted design of butterfly bileaflet valves for the mitral position. *Scandinavian journal of thoracic and cardiovascular surgery*, 19(2): 139–148, 1985.

[14] R. P. Beyer. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics*, 98(1):145–162, 1992.

[15] R. Dillon, L. Fauci, A. Fogelson, and D. Gaver III. Modeling biofilm processes using the immersed boundary method. *Journal of Computational Physics*, 129(1):57–73, 1996.

[16] L. J. Fauci and C. S. Peskin. A computational model of aquatic animal locomotion. *Journal of Computational Physics*, 77(1):85–108, 1988.

[17] R. Glowinski, T.-W. Pan, T. I. Hesla, and D. D. Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999.

[18] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Périaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *Journal of Computational Physics*, 169(2):363–426, 2001.

[19] R. Glowinski and Y. Kuznetsov. Distributed Lagrange multipliers based on fictitious domain method for second order elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 196:1498–1506, 2007.

[20] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193:2051–2067, 2004.

[21] W. K. Liu, D. W. Kim, and S. Tang. Mathematical foundations of the immersed finite element method. *Computational Mechanics*, 39(3):211–222, 2007.

[22] X. S. Wang, L. T. Zhang, and W. K. Liu. On computational issues of immersed finite element methods. *Journal of Computational Physics*, 228(7):2535–2551, 2009.

[23] X. Wang and L. T. Zhang. Modified immersed finite element method for fully-coupled fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 267:150–169, 2013.

[24] H. Casquero, C. Bona-Casas, and H. Gomez. A NURBS-based immersed methodology for fluid–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 284: 943–970, 2015.

[25] T. Rüberg and F. Cirak. Subdivision-stabilised immersed B-spline finite elements for moving boundary flows. *Computer Methods in Applied Mechanics and Engineering*, 209–212:266–283, 2012.

[26] C. Kadapa, W. G. Dettmer, and D. Perić. A stabilised immersed boundary method on hierarchical b-spline grids for fluid–rigid body interaction with solid–solid contact. *Computer Methods in Applied Mechanics and Engineering*, 318:242–269, 2017.

[27] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36:9–15, 1971.

[28] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes. An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053, 2015.

[29] F. Xu, D. Schillinger, D. Kamensky, V. Varduhn, C. Wang, and M.-C. Hsu. The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, 141:135–154, 2016.

[30] T. J. R. Hughes, L. Mazzei, and K. E. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3:47–59, 2000.

[31] T. J. R. Hughes, L. Mazzei, A. A. Oberai, and A. Wray. The multiscale formulation of large eddy simulation: Decay of homogeneous isotropic turbulence. *Physics of Fluids*, 13: 505–512, 2001.

[32] T. J. R. Hughes, G. Scovazzi, and L. P. Franca. Multiscale and stabilized methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.

[33] Y. Bazilevs, V. M. Calo, J. A. Cottrel, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[34] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36:12–26, 2007.

[35] M.-C. Hsu, C. Wang, F. Xu, A. J. Herrema, and A. Krishnamurthy. Direct immersogeometric fluid flow analysis using B-rep CAD models. *Computer Aided Geometric Design*, 43:143–158, 2016.

[36] C. Wang, F. Xu, M.-C. Hsu, and A. Krishnamurthy. Rapid B-rep model preprocessing for immersogeometric analysis using analytic surfaces. *Computer Aided Geometric Design*, 52: 190–204, 2017.

[37] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[38] H. S. Udaykumar, R. Mittal, and P. Rampunggoon. Interface tracking finite volume method for complex solid–fluid interactions on fixed meshes. *International Journal for Numerical Methods in Biomedical Engineering*, 18(2):89–97, 2002.

[39] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[40] T. E. Tezduyar. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics*, 28:1–44, 1992.

[41] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[42] M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. *Computer Methods in Applied Mechanics and Engineering*, 199:828–840, 2010.

[43] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Sweden, 1987.

[44] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods, 2nd ed.* Springer, Berlin, 2002.

[45] A. Ern and J. L. Guermond. *Theory and Practice of Finite Elements*. Springer, Berlin, 2004.

[46] D. Kamensky, M.-C. Hsu, Y. Yu, J. A. Evans, M. S. Sacks, and T. J. R. Hughes. Immersogeometric cardiovascular fluid–structure interaction analysis with divergence-conforming B-splines. *Computer Methods in Applied Mechanics and Engineering*, 314:408–472, 2017.

[47] M. C. H. Wu, D. Kamensky, C. Wang, A. J. Herrema, F. Xu, M. S. Pigazzini, A. Verma, A. L. Marsden, Y. Bazilevs, and M.-C. Hsu. Optimizing fluid–structure interaction systems with immersogeometric analysis and surrogate modeling: Application to a hydraulic arresting gear. *Computer Methods in Applied Mechanics and Engineering*, 316:668–693, 2017.

[48] K. Schloegel, G. Karypis, and V. Kumar. ParMETIS - Parallel Graph Partitioning and Fill-reducing Matrix Ordering , Version 4.0.3. http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview, 2013. University of Minnesota, Minneapolis, MN.

[49] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.10, Argonne National Laboratory, 2018. http://www.mcs.anl.gov/petsc.

[50] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual-based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.

[51] N. M. Newmark. A method of computation for structural dynamics. *ASCE Journal of the Engineering Mechanics Division*, 85:67–94, 1959.

[52] J. Happel and H. Brenner. *Low Reynolds number hydrodynamics: with special applications to particulate media*. Springer Science & Business Media, 2012.

[53] A. Ten Cate, C. H. Nieuwstad, J. J. Derksen, and H. E. A. Van den Akker. Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity. *Physics of Fluids*, 14(11):4012–4025, 2002.

[54] G. Segre. Radial particle displacements in poiseuille flow of suspensions. *Nature*, 189:209–210, 1961.

[55] H. Amini, W. Lee, and D. Di Carlo. Inertial microfluidic physics. *Lab on a Chip*, 14(15): 2739–2761, 2014.

[56] J. Feng, H. H. Hu, and D. D. Joseph. Direct simulation of initial value problems for the motion of solid bodies in a newtonian fluid. Part 2. Couette and Poiseuille flows. *Journal of fluid mechanics*, 277:271–301, 1994.

[57] D. Di Carlo. Inertial microfluidics. *Lab on a Chip*, 9(21):3038–3046, 2009.

[58] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[59] A. J. Chung, D. Pulido, J. C. Oka, H. Amini, M. Masaeli, and D. Di Carlo. Microstructure-induced helical vortices allow single-stream and long-term inertial focusing. *Lab on a Chip*, 13(15):2942–2949, 2013.

[60] *ANSYS Fluent Theory Guide. Release 16.1.* Ansys, Inc., Canonsburg, PA, 2015.