Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

On Reciprocity and Splitting Errors in Direct-Forcing Immersed Boundary Methods

Francesco Capuano

francesco.capuano@upc.edu

Universitat Politècnica de Catalunya

Research Article

Keywords: immersed boundary methods, direct forcing scheme, time integration, projection method, incompressible flow

Posted Date: March 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9107598/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

On Reciprocity and Splitting Errors in Direct-Forcing Immersed Boundary Methods

Francesco Capuano[1*]

[1]Department of Fluid Mechanics, Universitat Politècnica de Catalunya · BarcelonaTech, Av. d'Eduard Maristany, 16, Barcelona, 08019, Spain.

Corresponding author(s). E-mail(s): francesco.capuano@upc.edu;

Abstract

Direct-forcing immersed boundary methods (IBMs) are widely used to enforce no-slip boundary conditions in incompressible flow solvers owing to their flexibility and ease of implementation. In this work we introduce a semi-discrete framework that encompasses both Lagrangian and Eulerian direct-forcing IBMs. Within this formulation, classical algorithms can be interpreted as *approximate block-LU factorizations* of the fully coupled monolithic system. This perspective reveals two distinct sources of error in the enforcement of the no-slip condition: a reciprocity error, associated with the non-exact adjoint relationship between interpolation and spreading operators, and a splitting error, arising from the pressure predictor used in projection-based time integration. The framework clarifies how the relative importance of these mechanisms depends on the interpolation kernel, the time step, and the physical configuration. Representative numerical experiments illustrate the behavior of the two error mechanisms and confirm the predictions of the analysis. In particular, a simple oscillating-balloon configuration is proposed as a diagnostic benchmark that directly exposes inconsistencies between boundary motion and the Eulerian mass balance. The analysis provides a unified interpretation of previously observed inaccuracies in direct-forcing IBMs and suggests possible directions for improvement, including the development of stable higher-order pressure predictors and improved constructions of interpolation–spreading operators.

Keywords: immersed boundary methods, direct forcing scheme, time integration, projection method, incompressible flow

1 Introduction

This study is concerned with the numerical solution of the incompressible Navier–Stokes equations in domains with immersed surfaces, i.e. non-simply-connected regions such as the one sketched in Fig. 1(a). The domain of interest may be the external region Ω_e (e.g. external aerodynamics over a body), the internal region Ω_i (internal flows), or both, as in fluid–fluid or fluid–structure interaction problems. The continuous formulation reads

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u}\boldsymbol{u}) &= -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{u}, \\ \nabla \cdot \boldsymbol{u} &= 0 \quad \text{in } \Omega_*, \\ \boldsymbol{u}|_{S_i} &= \boldsymbol{u}_b(\xi, t), \end{aligned} \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ is the velocity field, p the pressure, ρ the density, ν the kinematic viscosity, and $\boldsymbol{u}_b(\xi, t)$ the velocity on the immersed surface $S_i = \partial\Omega_i$; the latter can be prescribed or come from the solution of a coupled system. For the sake of simplicity and without sacrificing generality, we will assume that the problem is governed by the Navier-Stokes equations in any subdomain Ω_*, with Ω_* corresponding to Ω_e, Ω_i or $\Omega_e \cup \Omega_i$[1].

A classical approach to this problem consists in generating a body-fitted mesh that conforms to S_i, see Fig. 1(b). This facilitates the direct imposition of boundary conditions, but is well known to suffer from severe drawbacks: mesh generation becomes a major bottleneck for complex geometries and often requires substantial manual intervention even for stationary bodies. On the other hand, moving bodies require costly and complex remeshing procedures. The Arbitrary Lagrangian–Eulerian (ALE) framework (see, e.g., Donea et al (2004) for a review) provides a systematic strategy to accommodate moving or deforming boundaries by allowing the computational mesh

[1] This is not necessarily true for certain multiphysics problems.

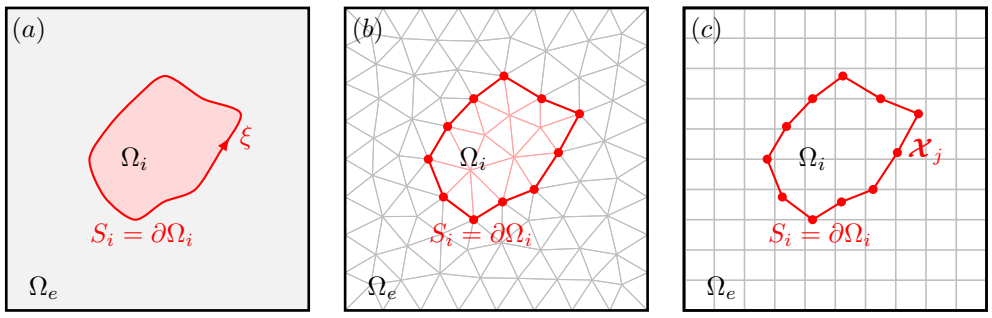


Fig. 1: Schematic representation of the problem setting. *(a)* Continuous problem: the fluid domain Ω_e surrounds the immersed region Ω_i, bounded by the surface S_i. *(b)* Discretization with a body-fitted mesh, where grid nodes conform to S_i and boundary conditions are directly imposed. *(c)* IBM discretization on a Cartesian grid, where Lagrangian markers j represent S_i.

to move independently of the fluid motion; nevertheless, significant mesh deformation or topological changes may still require re-meshing procedures.

An attractive alternative is the immersed boundary method (IBM), illustrated in Fig. 1(c), where the entire domain $\Omega_e \cup \Omega_i$ is discretized using a simple Cartesian grid. Since this Eulerian mesh does not conform to S_i, a set of auxiliary forcing points is introduced to enforce the no-slip condition locally. In so-called Lagrangian formulations, these forcing points coincide with discrete markers representing the surface, whereas in Eulerian formulations they correspond to selected grid nodes near the boundary. The reader is referred to Verzicco (2023), Mittal and Seo (2023) and Roy et al (2020) for a thorough historical outlook, complete taxonomy, and technical description of the IBM, particularly in the context of finite-difference (FD) and finite-volume (FV) methods. Here we focus on the broad distinction between continuous-forcing and discrete-forcing approaches. In the former category, pioneered by Peskin (1972), the body force is introduced directly at the level of the continuous equations, whereas in the discrete-forcing approach it is incorporated after their discretization. In this work, we study discrete (or direct) forcing methods, and consider

the following semi-discrete formulation:

$$\begin{aligned} \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + C(\mathbf{u}) &= -G\,\mathrm{p} + \nu L\mathbf{u} + H\boldsymbol{\mathfrak{F}} \\ D\mathbf{u} &= 0 \quad \text{in } \Omega_e \cup \Omega_i, \\ E\mathbf{u} &= \mathbf{u}, \end{aligned} \tag{2}$$

where $\mathbf{u}(\mathbf{x},t) \in \mathbb{R}^{N_u}$ and $\mathrm{p}(\mathbf{x},t) \in \mathbb{R}^{N_p}$ are the discrete velocity and pressure defined on the Eulerian nodes $\mathbf{x}$, while $\boldsymbol{\mathfrak{F}} \in \mathbb{R}^{d\times N_f}$ and $\mathbf{u} \in \mathbb{R}^{d\times N_f}$ are the forces and desired velocities at a set of N_f forcing points whose location and number depend on the method; d is the dimension of the problem (hereinafter $d = 2$). In addition, $C : \mathbb{R}^{N_u} \to \mathbb{R}^{N_u}$ is the discrete convective term, $G : \mathbb{R}^{N_p} \to \mathbb{R}^{N_u}$ and $D : \mathbb{R}^{N_u} \to \mathbb{R}^{N_p}$ are the discrete gradient and divergence operators (such that $D = -G^\top$), and $L : \mathbb{R}^{N_u} \to \mathbb{R}^{N_u}$ is the discrete Laplacian operator. Finally, $E : \mathbb{R}^{N_u} \to \mathbb{R}^{2N_f}$ is an operator *transferring* Eulerian velocities to the forcing points, and $H : \mathbb{R}^{2N_f} \to \mathbb{R}^{N_u}$ is a spreading operator transferring forces on the full Eulerian grid. The operators E and H are constructed from discrete kernels, and in practice $H = E^\top$, with appropriate scaling factors to ensure momentum conservation. This condition also ensures semi-discrete kinetic energy conservation in the inviscid limit for stationary surfaces, provided that the operator $C(\mathbf{u})$ is skew-symmetric. The analysis presented in this paper is valid for any spatial discretization that can be expressed as in Eq. (2).

The details of how the operators H and E are constructed, and how $\boldsymbol{\mathfrak{F}}$ and $\mathbf{u}$ are determined, depend on the specific discrete-forcing strategy employed. In the FD/FV literature, these methods are typically divided into two broad categories:

- *Eulerian* schemes, in which the forcing is imposed at a subset of Eulerian grid nodes near the immersed surface, as in Fadlun et al (2000) (see Fig. 2a). In this case, E and H act as mask or selection operators on Eulerian degrees of freedom. The target

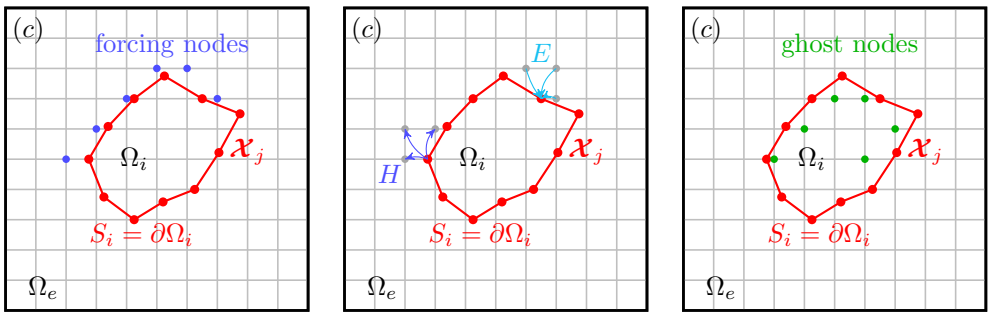


Fig. 2: Schematic representation of direct forcing approaches. *(a)* Eulerian discrete-forcing scheme: forcing is applied directly at selected Eulerian nodes near the immersed boundary. *(b)* Lagrangian discrete-forcing scheme, where interpolation (E) and spreading (H) operators couple Eulerian velocities and Lagrangian markers. *(c)* Ghost-cell method: discrete operators are modified at selected interior (ghost) nodes to enforce boundary conditions through stencil reconstruction.

velocities $\mathbf{u}$ are obtained through local interpolation or reconstruction procedures designed to enforce the no-slip condition;

- *Lagrangian* schemes, in which the force is computed at discrete Lagrangian markers representing the surface (see Fig. 2b), as originally proposed by Uhlmann (2005). Here N_f coincides with the number of Lagrangian marker points N_L, the velocity field $\mathbf{u}$ coincides with the prescribed surface velocity, and E and H are interpolation and spreading operators transferring information between Eulerian and Lagrangian locations.

Closely related sharp-interface approaches include ghost-cell methods (see Fig. 2c), in which the no-slip constraint is enforced at selected Eulerian nodes located inside the solid region through discrete reconstruction relations, as in Tseng and Ferziger (2003). From an algebraic standpoint, both direct-forcing Eulerian and ghost-cell schemes can be interpreted as enforcing the same discrete constraint at different sets of grid nodes.

It is important to observe that the IBM taxonomy presented above largely originates from the FD/FV community. In the finite-element method (FEM), immersed techniques are usually formulated within a variational framework and classified differently, e.g. as fictitious-domain or cut-element methods. The former category retains

the entire background mesh and imposes constraints variationally, in close analogy with FD/FV IBMs, see Glowinski et al (1999). In contrast, many *cut-FEM* formulations remove (or weakly constrain) the portion of the mesh lying outside the physical domain from the weak formulation, rather than treating it as an active fluid region; see the recent review by Burman et al (2025).

The objective of this study is to analyze the time integration of Eq. (2). From an algebraic standpoint, the forcing term can be interpreted as a Lagrange multiplier enforcing the no-slip constraint, in the same way that pressure enforces incompressibility. This variational interpretation was first proposed by Glowinski et al (1999) and later developed by Taira and Colonius (2007). In practice, however, most IBM implementations adopt a split time-advancement procedure, which may introduce temporal inconsistencies in the enforcement of the no-slip constraint. We show that Eq. (2) provides a unified algebraic framework encompassing both Lagrangian and Eulerian direct-forcing schemes. By introducing the forcing at the semi-discrete level, the accuracy of constraint enforcement in time can be systematically analyzed and the primary sources of error clearly identified. The main objective of the paper is therefore to assess the accuracy of different time-integration strategies for Eq. (2), with particular emphasis on the mechanisms responsible for constraint violation.

The paper is organized as follows. Section 2 introduces the mathematical formalism and the methods analyzed in this work. Results are reported and discussed in Section 3, while concluding remarks are drawn in Section 4.

2 Mathematical formulation

2.1 Monolithic time integration

In the absence of immersed surfaces, the semi-discrete system of Eq. (2) reduces to a differential–algebraic equation (DAE) of index 2, as thoroughly discussed by Sanderse and Koren (2012). When the IBM is included, the system remains of index 2,

with the pair $(\mathrm{p}, \boldsymbol{\mathfrak{F}})$ acting as algebraic multipliers that enforce, respectively, the incompressibility and no-slip constraints, as discussed by Liska and Colonius (2017).

Time discretization of Eq. (2) can be written in compact form as a monolithic block-matrix system:

$$\begin{bmatrix} A & G & E^T \\ D & 0 & 0 \\ E & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ \phi^i \\ \boldsymbol{\mathfrak{f}}^i \end{bmatrix} = \begin{bmatrix} \mathbf{r}_u^i \\ 0 \\ \mathbf{u}^i \end{bmatrix}, \tag{3}$$

where $\mathbf{u}^i = \mathbf{u}(t^n + \tilde{c}_i \Delta t)$ and the scaled variables $[\phi^i, \boldsymbol{\mathfrak{f}}^i] = \tilde{c}_i \Delta t \cdot [\mathrm{p}^i, \boldsymbol{\mathfrak{F}}^i]$ have been introduced for convenience. Equation (3) is general and can represent multi-step or multi-stage schemes, either explicit or implicit. A specific method is recovered by specifying A, $\tilde{c}_i$, and $\mathbf{r}_u^i$ for $i = 1, \ldots, s$. In what follows, we focus on explicit schemes, for which $A = I$. Although semi-implicit methods (where viscous terms are treated implicitly) are frequently employed, restricting to the explicit case simplifies notation without affecting the main discussion.

For instance, an Adams–Bashforth (AB) method of order $k+1$ corresponds to $s = \tilde{c} = 1$, with $[\mathbf{u}^i, \phi^i, \boldsymbol{\mathfrak{f}}^i] = [\mathbf{u}^{n+1}, \phi^{n+1}, \boldsymbol{\mathfrak{f}}^{n+1}]$ and

$$\mathbf{r}_u = \mathbf{u}^n + \Delta t \sum_{j=0}^{k} \beta_j \left[C(\mathbf{u}^{n-j}) + \nu L \mathbf{u}^{n-j} \right]. \tag{4}$$

Similarly, for an explicit Runge–Kutta (RK) scheme with s stages, one obtains

$$\mathbf{r}_u^i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i} \tilde{a}_{ij} \left[C(\mathbf{u}^j) + \nu L \mathbf{u}^j \right], \tag{5}$$

where $[\mathbf{u}^s, \phi^s, \mathbf{f}^s] = [\mathbf{u}^{n+1}, \phi^{n+1}, \mathbf{f}^{n+1}]$. The *shifted* Butcher tableau notation for explicit schemes has been adopted:

$$\tilde{a}_{ij} = \begin{pmatrix} a_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{s1} & \dots & a_{s,s-1} & 0 \\ b_1 & \dots & b_{s-1} & b_s \end{pmatrix} \qquad \tilde{c}_i = \begin{pmatrix} c_2 \\ \vdots \\ c_s \\ 1 \end{pmatrix}.$$

Following Perot (1993) and Taira and Colonius (2007), LU decomposition of Eq. (3) yields a *half-explicit*, projection-like procedure:

$$\begin{gathered} \mathbf{u}^{i,*} = \mathbf{r}_u^i, \\ \begin{bmatrix} DG & -DH \\ -EG & EH \end{bmatrix} \begin{bmatrix} \phi^i \\ \mathbf{f}^i \end{bmatrix} = \begin{bmatrix} D\mathbf{u}^{i,*} \\ \mathbf{u}^i - E\mathbf{u}^{i,*} \end{bmatrix}, \\ \mathbf{u}^i = \mathbf{u}^{i,*} + H\mathbf{f}^i - G\phi^i. \end{gathered} \tag{6}$$

This algorithm is structurally equivalent to a classical projection method, with the key difference that the Lagrange multipliers consist of both the pressure and the forcing term. Solving the coupled system in Eq. (6) is generally impractical for three-dimensional simulations with a high number of Lagrangian markers; however, it enforces the no-slip condition *exactly*, and DAE theory guarantees that the solution retains the theoretical order of accuracy, see Hairer et al (2006).

2.2 Discrete-forcing schemes

In practice, discrete-forcing schemes are most often implemented according to the following *nested-projection* paradigm, see, e.g., Fadlun et al (2000) and Uhlmann (2005):

$$\mathbf{u}^{i,*} = \mathbf{r}_u^i - G\phi^{i,*} \tag{7a}$$

$$\mathbf{f}^i = \mathbf{u}^i - E\mathbf{u}^{i,*} \tag{7b}$$

$$DG\phi = D\underbrace{(\mathbf{u}^{i,*} + H\mathbf{f}^i)}_{\mathbf{u}^{i,**}} \tag{7c}$$

$$\mathbf{u}^i = \mathbf{u}^{i,**} - G\phi \tag{7d}$$

$$\phi^i = \phi + \phi^{i,*}, \tag{7e}$$

corresponding to: (7a) computation of a provisional velocity $\mathbf{u}^{i,*}$ using a pressure predictor $\phi^{i,*}$; (7b) explicit evaluation of the IBM forcing; (7c) correction of $\mathbf{u}^{i,*}$ to satisfy the no-slip condition and simultaneous solution of a Poisson problem for the differential pressure ϕ; (7d) projection of the velocity field onto the divergence-free space; (7e) incremental update of the pressure — using either AB or RK schemes.

The system in Eq. (7) can be equivalently recast in compact form as

$$\begin{gathered}
\mathbf{u}^{i,*} = \mathbf{r}_u^i \\
\begin{bmatrix} DG & -DH \\ 0 & I \end{bmatrix} \begin{bmatrix} \phi^i \\ \mathbf{f}^i \end{bmatrix} = \begin{bmatrix} D\mathbf{u}^{i,*} \\ \mathbf{u}^i - E(\mathbf{u}^{i,*} - G\phi^{i,*}) \end{bmatrix} \\
\mathbf{u}^i = \mathbf{u}^{i,*} + H\mathbf{f}^i - G\phi^i
\end{gathered} \tag{8}$$

The formulation of Eq. (8) makes the connection with Eq. (6) transparent. Specifically: (i) the Lagrange multiplier system is effectively decoupled by approximating the off-diagonal coupling term through an explicit pressure predictor, and (ii) the diagonal block EH is further approximated by the identity matrix, which assumes exact adjointness between E and H.

In this sense, the classical implementation of direct-forcing schemes can be reinterpreted as an *approximate LU factorization* of the original monolithic system.

2.3 No-slip error analysis

In the split nested-projection procedure described in Section 2.2 the divergence-free constraint is enforced exactly, whereas the no-slip constraint is satisfied only approximately. The error can be expressed as

$$\begin{aligned}\varepsilon_{\mathrm{IB}}^{n+1} &= \mathbf{u}^{n+1} - E\mathbf{u}^{n+1} = \mathbf{u}^{n+1} - E(\mathbf{u}^{*} + H\mathfrak{f}^{n+1} - G\phi^{n+1}) = \\ &= (I - EH)\mathfrak{f}^{n+1} + EG(\phi^{n+1} - \phi^{n+1,*}) = \\ &= \underbrace{\Delta t\,(I - EH)\mathfrak{F}^{n+1}}_{\text{reciprocity error}} + \underbrace{\Delta t\,EG(\mathrm{p}^{n+1} - \mathrm{p}^{n+1,*})}_{\text{splitting error}}.\end{aligned} \tag{9}$$

Two distinct error mechanisms can thus be identified:

- *Reciprocity* error, arising from the lack of exact adjointness between E and H. Although it scales linearly with Δt, it is primarily spatial in nature and depends on the choice of interpolation and spreading kernels;
- *Splitting* error, resulting from the explicit elimination of the off-diagonal term in the monolithic system and its replacement by a pressure predictor. This error is purely temporal and depends on the accuracy of the predictor in approximating the pressure at the correct time level. In particular, it scales with the order of the pressure predictor: without a predictor ($\mathrm{p}^{*} = 0$), the error is $\mathcal{O}(\Delta t)$; with $\mathrm{p}^{*} = \mathrm{p}^{n}$, it becomes $\mathcal{O}(\Delta t^{2})$.

It is important to observe that while Lagrangian schemes will generally be affected by both sources of error, in Eulerian-forcing schemes the reciprocity error is absent by definition. Indeed, since E and H are mask/selection operators, the identity $EH = I$ is always verified by construction.

Although it has long been recognized that direct-forcing immersed boundary methods introduce errors, their precise origin has not always been clearly identified. In fact, depending on the formulation, only one of the two mechanisms discussed above may be present. Purely Lagrangian direct-forcing schemes without projection suffer exclusively from reciprocity errors (for instance in many Lattice–Boltzmann implementations), whereas Eulerian direct-forcing schemes exhibit only splitting errors associated with the pressure–velocity coupling. In hybrid formulations, however, both mechanisms are present simultaneously, making it difficult to disentangle their respective contributions. As a consequence, although "slip" or "leakage" at the immersed boundary has long been observed, a unified algebraic framework explicitly identifying and distinguishing reciprocity and splitting errors has been lacking. To the best of the author's knowledge, the reciprocity error has only recently been clearly identified and named by Gsell and Favier (2021), from which we adopt the terminology used here. By contrast, the existence of an error associated with the pressure coupling was already noted by Fadlun et al (2000) in the context of Eulerian-forcing schemes, and later discussed by Domenichini (2008), among others. The present work aims to clarify this issue by introducing a simple semi-discrete framework that separates the two mechanisms and allows their effects to be examined independently.

2.4 Algorithms analyzed

Starting from Eq. (6) and Eq. (8), we introduce the following generalized class of algorithms:

$$
\begin{aligned}
\mathbf{u}^{i,*} &= \mathbf{r}_u^i, \\
\begin{bmatrix} DG & -DH \\ m_{21} & m_{22} \end{bmatrix}
\begin{bmatrix} \phi^i \\ \mathbf{f}^i \end{bmatrix}
&= \begin{bmatrix} D\mathbf{u}^{i,*} \\ \mathbf{u}^i - E(\mathbf{u}^{i,*} - G\phi^{i,*}) \end{bmatrix}, \\
\mathbf{u}^i &= \mathbf{u}^{i,*} + H\mathbf{f}^i - G\phi^i.
\end{aligned}
\tag{10}
$$

Within this framework, several relevant procedures can be obtained depending on the choice of the coefficients m_{21}, m_{22} and the pressure predictor ϕ^*. The algorithms considered in the present work are summarized in Table 1.

We restrict our attention to Lagrangian forcing schemes, which are affected by both splitting and reciprocity errors and are therefore particularly suitable to isolate and compare the two error sources. Four representative algorithms are considered:

(i) TC: the *exact* method of Taira and Colonius (2007), corresponding to the direct solution of the coupled system in Eq. (3);

(ii) SE: a family of schemes where the off–diagonal coupling term in the forcing equation is neglected. In SE1 the term is completely discarded, while in SE2 it is approximated using an explicit pressure predictor $\phi^* = \phi^n$. In both cases the forcing is still obtained implicitly;

(iii) RE: a scheme where the reciprocity relation is approximated by assuming $EH \approx I$, while the coupled system for the two Lagrange multipliers is retained;

(iv) Split: the fully split formulation obtained from Eq. (8), where both approximations are applied.

By construction, the TC method is free of both errors. The SE schemes are affected exclusively by splitting errors, the RE scheme only by reciprocity errors, while the Split formulation contains both contributions.

Three interpolation–spreading kernels are compared:

(i) *Collocated kernel*. The no-slip constraint is imposed directly at the nearest Eulerian grid node. This corresponds to a discrete Kronecker delta operator,

$$(E\mathbf{u})(\boldsymbol{\mathcal{X}}) = \mathbf{u}(\mathbf{x}_j),$$

while the spreading operator simply adds the Lagrangian force to that node,

$$(H\mathfrak{F})(\mathbf{x}_j) = \mathfrak{F}(\boldsymbol{\mathcal{X}}).$$

(ii) *Cell-average kernel.* The constraint is imposed through the average velocity over the control cell Ω_c containing the Lagrangian point,

$$(E\mathbf{u})(\boldsymbol{\mathcal{X}}) = \frac{1}{|\Omega_c|}\int_{\Omega_c} \mathbf{u}(\mathbf{x})\, d\mathbf{x}, \tag{11a}$$

$$(H\mathfrak{F})(\mathbf{x}) = \frac{1}{|\Omega_c|}\sum_{\boldsymbol{\mathcal{X}}\in\Omega_c} \mathfrak{F}(\boldsymbol{\mathcal{X}}). \tag{11b}$$

(iii) *Regularized delta function.* Interpolation and spreading are performed through a regularized discrete delta function,

$$(E\mathbf{u})(\boldsymbol{\mathcal{X}}) = \sum_{\mathbf{x}_i} \mathbf{u}(\mathbf{x}_i)\, \delta_h(\boldsymbol{\mathcal{X}} - \mathbf{x}_i), \tag{12a}$$

$$(H\mathfrak{F})(\mathbf{x}_i) = \sum_{\boldsymbol{\mathcal{X}}} \mathfrak{F}(\boldsymbol{\mathcal{X}})\, \delta_h(\boldsymbol{\mathcal{X}} - \mathbf{x}_i), \tag{12b}$$

where δ_h is defined as the tensor product

$$\delta_h(\mathbf{r}) = \frac{1}{h^2}\,\varphi\Big(\frac{r_x}{h}\Big)\,\varphi\Big(\frac{r_y}{h}\Big)$$

where h is the Eulerian grid spacing. For φ, we pick the following one–dimensional kernel proposed by Peskin (1972):

$$\varphi(r) = \begin{cases} \dfrac{1}{2b}\left(1 + \cos\left(\dfrac{\pi r}{b}\right)\right), & |r| \le b, \\[2ex] 0, & |r| > b\,, \end{cases}$$

Label	m_{21}	m_{22}	ϕ^*	Line style	No-slip errors	
					Splitting	Reciprocity
TC	$-EG$	EH	0		no	no
RE	$-EG$	I	0		no	yes
SE1	0	EH	0		yes	no
SE2	0	EH	ϕ^n		yes	no
Split	0	I	ϕ^n		yes	yes

Table 1: Summary of IBM algorithms analyzed in this work.

with $b = 2$. This kernel spreads the force radially in each coordinate direction and provides a smooth transfer between the Eulerian and Lagrangian grids. It is chosen for its simplicity; more sophisticated kernel functions may be employed, such as those proposed reported in Peskin (2002), or the moving-least square kernel by Vanella and Balaras (2009). However, this specific choice is supposed not to affect the analysis and results presented hereinafter.

In the following sections we present representative numerical results and discuss the relative importance of the two error sources, along with possible strategies to mitigate them.

3 Results and discussion

We consider two representative test cases: (i) the flow around an infinitely thin horizontal bar, and (ii) the flow generated by a balloon with an orifice that periodically contracts and expands. In both cases, the numerical solver employs a second-order staggered finite-difference discretization on a uniform Cartesian mesh with $N = 64$ grid points per side of a biperiodic domain of size $2\pi \times 2\pi$. Time advancement is performed using a third-order Adams–Bashforth scheme.

3.1 Stationary bar

The bar is located at the center of the domain, and the Reynolds number based on the bar length is $Re = 100$.

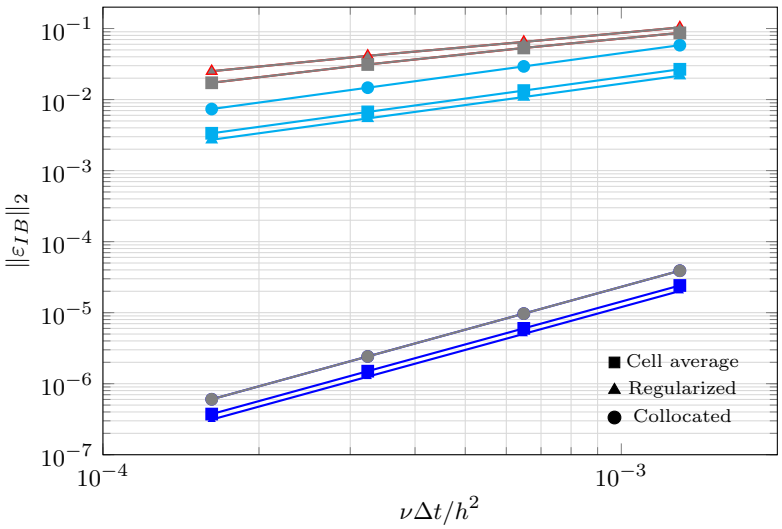


Fig. 3: L2-norm of the no-slip error for the five methods listed in Table 1, and for three interpolation kernels described in Section 2.4.

Results are reported in Fig. 3. The TC method produces only round-off errors ($\sim 10^{-15}$), which are therefore invisible at the plotting scale, for all interpolation/spreading kernels. A similar behaviour is observed for the RE scheme when using the collocated kernel, for which the operator product satisfies $EH = I$ exactly. In this particular configuration the no-slip constraint is enforced at the same Eulerian nodes where the forcing is applied, and the reciprocity condition is therefore satisfied to machine precision. While this behaviour is ideal in the present test, where the bar is perfectly aligned with the grid, it would generally deteriorate for curved or oblique geometries, where the collocated kernel provides only a crude geometric representation of the boundary. As an additional consequence, for this kernel the Split scheme becomes equivalent to SE2.

For the other kernels, the reciprocity error scales approximately as $\mathcal{O}(\Delta t)$, in agreement with the analysis of Gsell and Favier (2021). In practice we observe a slightly lower effective order (≈ 1.6), which is likely due to a residual interaction with spatial discretization errors. The splitting error, on the other hand, follows the expected

behaviour: it scales as $\mathcal{O}(\Delta t)$ in the absence of a pressure predictor and as $\mathcal{O}(\Delta t^2)$ when the predictor $\phi^* = \phi^n$ is employed.

For the Split scheme, and within the relatively small range of Δt considered here (chosen so as to clearly recover the expected temporal scalings), the reciprocity error dominates over the splitting error. Indeed, for the cell-average and Peskin kernels the results of the Split scheme essentially coincide with those of the RE method.

Although this example does not provide universal scaling laws, it illustrates that both error sources may become significant depending on the algorithmic choices, the physical configuration, and the selected time-step size. In the present case, increasing Δt (while remaining within the linear stability limit of the scheme) would likely lead to a regime where the splitting error becomes comparable to the reciprocity error.

3.2 Expansion/contraction of a balloon with orifice

The second test case is designed to assess the ability of the IBM formulation to correctly enforce a prescribed boundary motion while maintaining consistency with the Eulerian mass balance. In particular, the configuration allows a direct comparison between the rate of change of the volume enclosed by the moving boundary and the fluid flux through a prescribed opening. The test therefore provides a stringent verification of the coupling between the Lagrangian description of the moving boundary and the Eulerian representation of the flow field.

The main diagnostic quantity is the identity

$$\frac{dA}{dt} = -\int_{A_o} \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}A \equiv Q, \tag{13}$$

where $A(t)$ denotes the area of the fluid region enclosed by the balloon, A_o is the orifice area, and Q is the volumetric flux through the opening.

The balloon is centered at

$$(x_c, y_c) = \left(\pi,\, \pi - \sqrt{R(t)^2 - (W/2)^2}\right),$$

where W is the width of the orifice and $R(t)$ is the radius of the balloon, which varies in time according to

$$R(t) = R_0 + A\cos(\omega t), \tag{14}$$

where R_0 is the mean radius, A the oscillation amplitude, and ω the oscillation frequency. In the present test we take $R_0 = 1$, $A = 0.2$, and $\omega = 4\pi$.

The vertical position of the circle center is chosen so that the endpoints of the orifice remain fixed during the motion, while the remainder of the boundary forms a circular arc of radius $R(t)$ passing through these endpoints. The arc is discretized using $N_L = 50$ Lagrangian markers with coordinates $\boldsymbol{\mathcal{X}}(t) = [\mathcal{X}_x(t),\, \mathcal{X}_y(t)]$. The marker positions are given by

$$\begin{aligned} \mathcal{X}_{x,i}(t) &= x_c + R(t)\cos(\theta_i), \\ \mathcal{X}_{y,i}(t) &= y_c + R(t)\sin(\theta_i), \end{aligned} \tag{15}$$

where $\theta_i = \theta_0 - s_i(\pi + 2\theta_0)$, with $s_i \in [0, 1]$ a curvilinear coordinate discretized with N_L points and

$$\theta_0(t) = \arccos\left(\frac{W}{2R(t)}\right)$$

the angle subtended by the orifice.

According to this configuration the orifice lies on the horizontal line $y = \pi$. Its width is chosen so that the endpoints coincide with Eulerian grid-cell edges, namely $W = m\Delta x$ for some integer m. In the present case $m = 10$, giving an opening width approximately equal to unity. Because of this alignment, the flux through the orifice can be computed directly from the staggered velocity field. Denoting by v_i the vertical

velocity components at the 10 Eulerian faces spanning the opening, the discrete flux is evaluated as

$$Q = \Delta x \sum_{i=1}^{10} v_i. \tag{16}$$

The enclosed area can be obtained analytically by subtracting the area of the circular segment subtended by the orifice from the full circle area,

$$A(t) = \pi R^2 - R^2(\alpha - \sin\alpha),$$

where $\alpha = \pi - 2\theta_0$. Differentiation of this expression yields the analytical value of dA/dt used for comparison.

The prescribed Lagrangian boundary velocity $\mathbf{u} = [\mathrm{u}_x, \mathrm{u}_y]$ is obtained by differentiating Eq. (15) with respect to time,

$$\begin{aligned} \mathrm{u}_{x,i}(t) &= \dot{R}\cos(\theta_i) - R\sin(\theta_i)\dot{\theta}_i, \\ \mathrm{u}_{y,i}(t) &= \dot{R}\sin(\theta_i) + R\cos(\theta_i)\dot{\theta}_i - \frac{R\dot{R}}{\sqrt{R^2 - (W/2)^2}}, \end{aligned} \tag{17}$$

where the overdot denotes differentiation with respect to time.

Ideally, the flux Q and the analytical value of dA/dt should coincide, with discrepancies arising only from spatial discretization and time-integration errors. The present test therefore provides a stringent verification of the consistency between Lagrangian boundary motion and Eulerian mass conservation.

The behaviour of the algorithms listed in Table 1 in enforcing Eq. (13) is shown in Fig. 4. The simulations were performed with $\nu\Delta t/h^2 \approx 6.5 \times 10^{-4}$; the Reynolds number based on the orifice width is approximately 1600 at the instnat of maximum expansion/contraction rate.

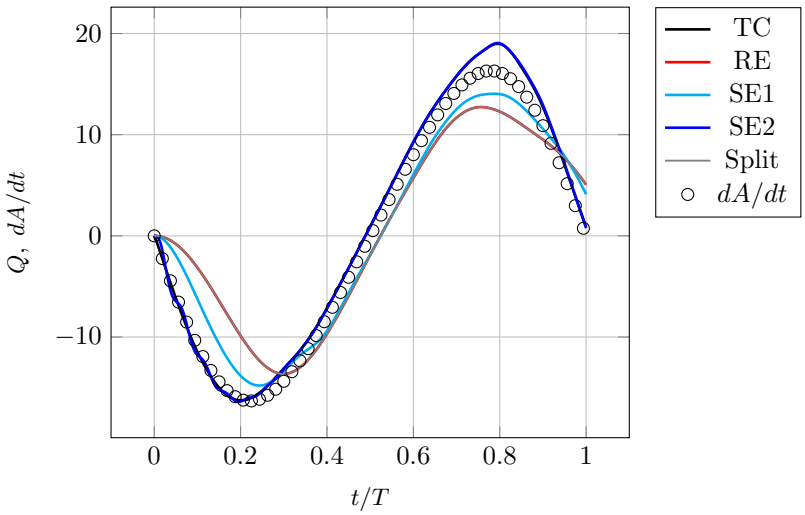


Fig. 4: Comparison between the flux through the orifice Q and the analytical rate of change of the enclosed area dA/dt for the balloon test case. Results are shown for the five IBM algorithms listed in Table 1.

Interestingly, although the TC method enforces the prescribed boundary velocity exactly (in the discrete sense), the computed flow rate still exhibits a noticeable discrepancy with the analytical curve for dA/dt. This mismatch is primarily attributable to spatial discretization effects and to the interpolation/spreading kernel. We also examined the influence of the number of Lagrangian markers (not shown): although some minor changes were observed, increasing the marker density did not eliminate the discrepancy. For $N_L = 50$, the ratio between the Lagrangian marker spacing h_L and the Eulerian grid spacing is $h_L/h \approx 0.82$ at the stasis time point (minimum balloon area). Furthermore, as the number of markers is increased, the matrix EH becomes severely ill-conditioned, as similarly noted by Vagnoli et al (2025) in the context of a moving-least square Lagrangian-forcing IBM. In this configuration the reciprocity error again appears to play a dominant role. In fact, the SE2 algorithm produces results that are very close to those of TC, apart from small oscillations—particularly during the contraction phase—which can be attributed to pressure splitting. By contrast, the Split scheme closely follows the behaviour of the RE method, indicating that

the reciprocity error largely determines the solution in this regime. These algorithms display a significant mismatch between Q and dA/dt, which reflects an inaccurate enforcement of the no-slip condition along the moving boundary. It is important to observe that, since the orifice endpoints coincide with Eulerian cell edges and the flux is evaluated directly from the staggered velocity components, the evaluation of Q is consistent with the finite–volume mass balance of the underlying solver; as a result, it introduces negligible additional numerical error.

Figure 5 shows three representative snapshots of the flow field inside and around the balloon during the oscillation cycle: maximum contraction rate ($t/T \approx 0.2$), stasis ($t/T = 0.5$), and maximum expansion rate ($t/T \approx 0.8$). During the expansion phase the flow field is only weakly affected by the choice of algorithm, likely because the strong outward acceleration dominates the dynamics. At $t/T = 0.5$, however, clear differences emerge, indicating that even relatively small discrepancies during the ejection phase propagate downstream and modify the flow structure. During contraction, the strength and size of the vortex pair forming inside the balloon depend directly on the incoming flow rate. In particular, the RE, SE1 and Split algorithms underestimate the inflow through the orifice, resulting in weaker vortex formation compared with the TC and SE2 solutions. These discrepancies ultimately originate from an imperfect enforcement of the no-slip condition at the immersed boundary. When the Eulerian velocity interpolated at the Lagrangian markers does not exactly match the prescribed boundary velocity, a small spurious flux is introduced across the boundary. Over time, this leakage manifests itself as a mismatch between the orifice flux and the analytical rate of change of the enclosed area.

3.3 Towards higher-order splitting schemes

In light of the error mechanisms identified above, we briefly review several recent approaches aimed at mitigating these effects and discuss their relation to the present framework.

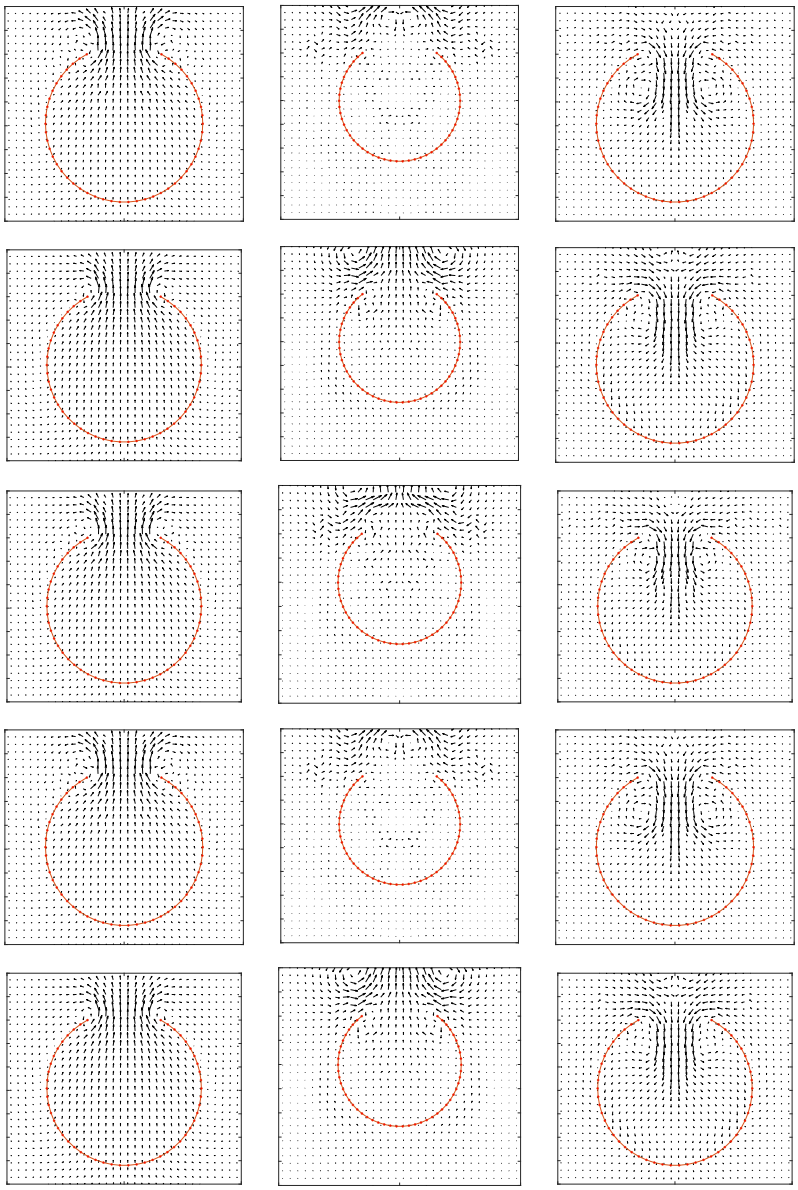


Fig. 5: Instantaneous velocity fields for the balloon test case. Rows correspond (from top to bottom) to the TC, RE, SE1, SE2 and Split algorithms. Columns correspond to three phases of the oscillation cycle: maximum contraction rate ($t/T \approx 0.2$), stasis ($t/T = 0.5$), and maximum expansion rate ($t/T \approx 0.8$). The displayed region is $x \in [0.5, 1.5]$ and $y \in [0.3, 1.2]$.

The reciprocity error depends strongly on the construction of the interpolation and spreading kernels. Gsell and Favier (2021) proposed approximating the diagonal of the inverse of the operator EH as

$$\mathrm{diag}(EH)^{-1} = \frac{1}{k}, \tag{18}$$

where the coefficient k depends only on the kernel function employed. This approximation effectively rescales the forcing and partially compensates for the non-reciprocity between interpolation and spreading. More recently, Vagnoli et al (2025) proposed a local correction strategy in which the scaling factor is determined dynamically for each Lagrangian marker at every time step. This procedure provides a spatially and temporally varying correction to the lack of reciprocity of the interpolation–spreading operators. Interestingly, these authors also tested an implicit forcing formulation that essentially coincides with the algorithm denoted SE2 in the present paper (although using a different kernel). Their results highlighted potential difficulties related to the conditioning of the matrix EH when the forcing is computed implicitly, consistent with the observations discussed in Section 3.2.

With regard to the splitting error, early efforts mainly focused on iterating the forcing calculation within the projection method until the no-slip condition was satisfied within a prescribed tolerance. More recently, it has been noted that this error can be particularly significant for thin shells and internal flows, such as the configurations analyzed in this work.

A possible strategy to reduce the splitting error follows from observing that

$$\varepsilon_{IB}^{\mathrm{split}} = \Delta t\, EG \underbrace{\left(\mathrm{p}^{n+1} - \mathrm{p}^{*}\right)}_{\mathcal{O}(\Delta t^{k})} = \mathcal{O}\left(\Delta t^{k+1}\right), \tag{19}$$

that is, a k-th order pressure predictor produces a $(k+1)$-th order splitting error. Equation (19) therefore suggests that higher-order pressure predictors constitute a natural path toward reducing this error. However, as emphasized by Sanderse and Koren (2012) and Capuano et al (2016), the *pseudo*-pressure appearing in projection schemes is not a pointwise quantity, but rather represents an integral over the time step. Consequently, pressure extrapolation should be performed at intermediate stages consistent with this interpretation. For instance, a second-order accurate scheme requires extrapolation at midpoints, and more generally an evaluation at $t^n + c_i\Delta t$ requires extrapolating the pressure at $t^n + \frac{c_i}{2}\Delta t$, as illustrated in Fig. 6.

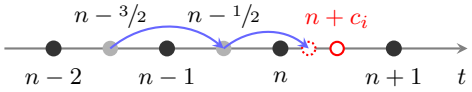


Fig. 6: Scheme for second-order extrapolation of the *pseudo*-pressure p at $t^n + c_i\Delta t$.

These extrapolation procedures are formally consistent and, in preliminary tests (not shown here), recover the expected order of accuracy for the pressure variable. However, incorporating them directly into the time advancement of Eq. (7) has so far led to unconditional instabilities in numerical experiments, regardless of the chosen time step. Ongoing work is therefore aimed at identifying the mechanisms responsible for these instabilities and at developing stable higher-order time-advancement strategies.

An alternative interpretation views the splitting error as the consequence of missing boundary conditions on the immersed surface during the pressure Poisson solve, which introduces a local inconsistency. Building on this perspective, Yildiran et al (2024) added a pressure gradient predictor in Eq. (7c) projected along the local normal of the immersed surface. According to the ghost-fluid framework discussed by Lalanne et al (2015), this imposes a jump at the surface S_i such that $\partial p/\partial n = 0$; as a consequence, the correction in Eq. (7d) does not alter the normal velocity at the body. This

approach was shown to significantly reduce the transpiration error without additional computational cost, and could potentially be combined with higher-order extrapolation strategies to further improve robustness. Related ideas were also explored by Lauber et al (2022), who enforced pressure boundary conditions by modifying the Poisson operator.

4 Conclusions

In this work we have presented a unified algebraic framework for analyzing errors in immersed boundary methods, with particular focus on direct-forcing schemes coupled with projection methods. Within this formalism, two distinct sources of no-slip error can be clearly identified: (i) the reciprocity error, arising from the approximate adjoint relationship between the interpolation and spreading operators; and (ii) the splitting error, induced by the temporal decoupling of velocity and pressure inherent to projection schemes.

Although both effects have been recognized individually in previous studies, their coexistence and relative contributions had not been systematically distinguished. The framework introduced here isolates the two mechanisms and provides a simple interpretation of how they arise in common IBM formulations. Representative numerical experiments illustrate their behavior and highlight their dependence on the interpolation kernel, time-step size, and algorithmic formulation.

The results indicate that neither source of error can be considered negligible in general. Reciprocity errors can dominate for certain kernel constructions and marker arrangements, whereas splitting errors may become significant for larger time steps or in internal-flow configurations such as those examined here. In principle, higher-order pressure predictors provide a natural route to reducing splitting errors, but preliminary tests indicate that straightforward implementations lead to instabilities whose origin

remains to be clarified. Alternative strategies, including ghost-fluid–inspired corrections that enforce pressure compatibility at the immersed surface, appear promising in reducing splitting-related inconsistencies.

Overall, the present analysis clarifies the algebraic origin of long-recognized inaccuracies in direct-forcing immersed boundary methods and provides a framework for interpreting and comparing different algorithmic variants. It is hoped that this perspective will facilitate the systematic development of improved schemes. Future work will focus on identifying stable higher-order time-integration strategies and on exploring hybrid approaches that combine kernel-based corrections with improved pressure–velocity coupling.

Acknowledgments. The author is thankful to Elias Balaras, Javier Principe and Eric Neiva for insightful discussions on immersed boundary methods and projection schemes. The author gratefully acknowledges support from the Serra Húnter program of the Generalitat de Catalunya (Spain).

Declarations

Funding. This work was supported by the Agencia Estatal de Investigación (AEI, Spain) under the BREATHE project (Grant CNS2024-154759).

Conflict of Interest. The authors have no relevant financial or non-financial interests to disclose.

Ethical approval. Not applicable.

Informed consent. Not applicable.

References

Burman E, Hansbo P, Larson MG, et al (2025) Cut finite element methods. Acta Numerica 34:1–121

Capuano F, Coppola G, Chiatto M, et al (2016) Approximate projection method for the incompressible navier–stokes equations. AIAA journal 54(7):2179–2182

Domenichini F (2008) On the consistency of the direct forcing method in the fractional step solution of the navier-stokes equations. Journal of Computational Physics 227:6372–6384. https://doi.org/10.1016/j.jcp.2008.03.009

Donea J, Huerta A, Ponthot JP, et al (2004) Arbitrary l agrangian–e ulerian methods. Encyclopedia of computational mechanics pp 413–437

Fadlun EA, Verzicco R, Orlandi P, et al (2000) Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. Journal of Computational Physics 161:35–60. https://doi.org/10.1006/jcph.2000.6484

Glowinski R, Pan TW, Hesla TI, et al (1999) A distributed lagrange multiplier/fictitious domain method for particulate flows. International Journal of Multiphase Flow 25(5):755–794

Gsell S, Favier J (2021) Direct-forcing immersed-boundary method: A simple correction preventing boundary slip error. Journal of Computational Physics 435. https://doi.org/10.1016/j.jcp.2021.110265

Hairer E, Lubich C, Wanner G (2006) Geometric Numerical Integration. Springer

Lalanne B, Villegas LR, Tanguy S, et al (2015) On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method. Journal of Computational Physics 301:289–307

Lauber M, Weymouth GD, Limbert G (2022) Immersed boundary simulations of flows driven by moving thin membranes. Journal of Computational Physics 457. https://doi.org/10.1016/j.jcp.2022.111076

Liska S, Colonius T (2017) A fast immersed boundary method for external incompressible viscous flows using lattice green's functions. Journal of Computational Physics 331:257–279

Mittal R, Seo JH (2023) Origin and evolution of immersed boundary methods in computational fluid dynamics. Physical review fluids 8(10):100501

Perot JB (1993) An analysis of the fractional step method. Journal of Computational Physics 108(1):51–58

Peskin CS (1972) Flow patterns around heart valves: a numerical method. Journal of computational physics 10(2):252–271

Peskin CS (2002) The immersed boundary method. Acta numerica 11:479–517

Roy S, De A, Balaras E (2020) Immersed boundary method. Development and Applications 1

Sanderse B, Koren B (2012) Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations. J Comput Phys 231:3041–3063

Taira K, Colonius T (2007) The immersed boundary method: A projection approach. Journal of Computational Physics 225:2118–2137. https://doi.org/10.1016/j.jcp.2007.03.005

Tseng YH, Ferziger JH (2003) A ghost-cell immersed boundary method for flow in complex geometry. Journal of Computational Physics 192(2):593–623

Uhlmann M (2005) An immersed boundary method with direct forcing for the simulation of particulate flows. Journal of computational physics 209(2):448–476

Vagnoli G, Scarpolini MA, Verzicco R, et al (2025) A local and explicit forcing correction for lagrangian immersed boundary methods. Computer Physics Communications p 109741

Vanella M, Balaras E (2009) A moving-least-squares reconstruction for embedded-boundary formulations. Journal of Computational Physics 228:6617–6628. https://doi.org/10.1016/j.jcp.2009.06.003

Verzicco R (2023) Immersed boundary methods: Historical perspective and future outlook. Annual Review of Fluid Mechanics 55(1):129–155

Yildiran IN, Beratlis N, Capuano F, et al (2024) Pressure boundary conditions for immersed-boundary methods. Journal of computational physics 510:113057