



# A one-sided direct forcing immersed boundary method using moving least squares

Rahul Bale<sup>a,\*</sup>, Amneet Pal Singh Bhalla<sup>d,\*</sup>, Boyce E. Griffith<sup>c</sup>,  
Makoto Tsubokura<sup>a,b</sup>

<sup>a</sup> Riken Center for Computational Sciences, Japan

<sup>b</sup> Graduate School of System Informatics, Kobe University, Kobe, Japan

<sup>c</sup> Departments of Mathematics, Applied Physical Sciences, and Biomedical Engineering, University of North Carolina, Chapel Hill, NC, United States of America

<sup>d</sup> Department of Mechanical Engineering, San Diego State University, San Diego, CA, United States of America

## ARTICLE INFO

### Article history:

Available online 17 May 2021

### Keywords:

Backus-Gilbert MLS formulation

Fictitious domain method

Meshless methods

Vehicular aerodynamics

## ABSTRACT

This paper presents a one-sided immersed boundary (IB) method using kernel functions constructed via a moving least squares (MLS) method. The resulting kernels effectively couple structural degrees of freedom to fluid variables on only one side of the fluid-structure interface. This reduces spurious feedback forcing and internal flows that are typically observed in IB models that use isotropic kernel functions to couple the structure to fluid degrees of freedom on both sides of the interface. The method developed here extends the original MLS methodology introduced by Vanella and Balaras (2009) [27]. Prior IB/MLS methods have used isotropic kernel functions that coupled fluid variables on both sides of the boundary to the interfacial degrees of freedom. The original IB/MLS approach converts the cubic spline weights typically employed in MLS reconstruction into an IB kernel function that satisfies particular discrete moment conditions. This paper shows that the same approach can be used to construct one-sided kernel functions (kernel functions are referred to as generating functions in the MLS literature). We also examine the performance of the new approach for a family of kernel functions introduced by Peskin. It is demonstrated that the one-sided MLS construction tends to generate non-monotone interpolation kernels with large over- and undershoots. We present two simple weight shifting strategies to construct generating functions that are positive and monotone, which enhances the stability of the resulting IB methodology. Benchmark cases are used to test the order of accuracy and verify the one-sided IB/MLS simulations in both two and three spatial dimensions. This new IB/MLS method is also used to simulate flow over the Ahmed car model, which highlights the applicability of this methodology for modeling complex engineering flows.

© 2021 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Immersed boundary (IB) methods [1–3] are a class of fictitious domain methods [4] that can enable the efficient solution to complex moving domain problems. The IB methodology is now widely used in modeling large scale engineering [2,5–11]

\* Corresponding authors.

E-mail addresses: [rahul.bale@riken.jp](mailto:rahul.bale@riken.jp) (R. Bale), [asbhalla@sdsu.edu](mailto:asbhalla@sdsu.edu) (A.P.S. Bhalla).

<https://doi.org/10.1016/j.jcp.2021.110359>

0021-9991/© 2021 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

and biological [12–17] models that have large boundary deformations or displacements within the computational domain. IB methods have also seen substantial use for applications that require only fixed geometries because they simplify grid generation [18–21]. Compared to body-fitted and unstructured grid approaches, IB methods have lower memory footprint, are easier to parallelize, and allow for fast linear solvers.

In IB method, the structural displacement, velocity, and forces are described on a Lagrangian mesh, whereas the fluid velocity and pressure are described on a background Eulerian grid. The Lagrangian mesh is allowed to cut the background Cartesian grid arbitrarily, which allows the IB methodology to be flexible and efficient. Interactions between the Lagrangian and Eulerian variables are typically mediated by integral transforms with a smooth delta function kernel. The regularized delta function kernels effectively smear the interface over a few Eulerian grid cells. The smearing process reduces the order of accuracy of the methodology, which can necessitate using high grid resolution near the interface, particularly at higher Reynolds numbers. It, however, permits a continuous solution of velocity and pressure fields across the interface. A limitation of the standard diffuse-interface IB method is that for immersed bodies, the continuous solution can lead to spurious flow inside the structure [22]. The internal flow may not influence the external flow at moderate Reynolds numbers, but it can interfere with the external boundary layer at high Reynolds number (see Sec. 5.6).

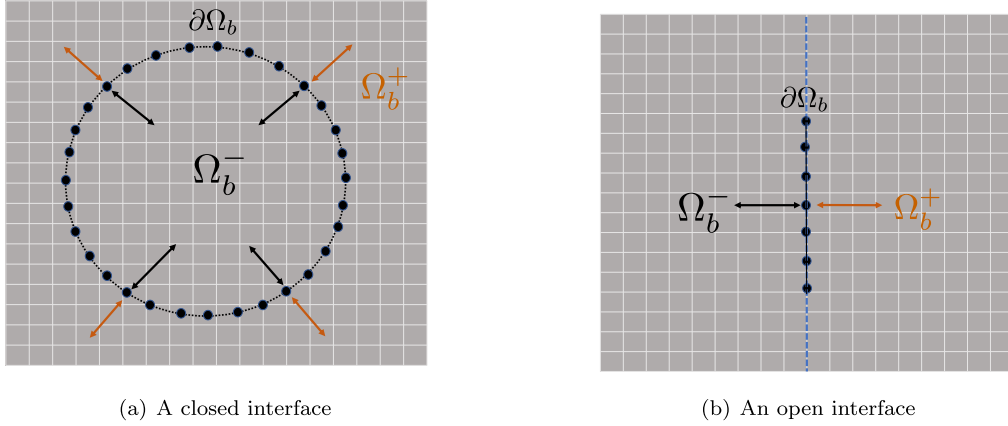
Sharp-interface versions of the IB method have also been developed that reconstruct the velocity along the interface using procedures that are not naturally expressed as integral transforms [23–25]. In some of these approaches, fluid boundary conditions are directly imposed along the immersed boundary, and the momentum and continuity equations are solved only on one side of the boundary. When the structure moves, fluid cells are “covered” and “uncovered”, and this can lead to spurious force oscillations in the time history of the integrated drag and lift quantities. Although these approaches can improve the order of accuracy as compared to formulations that use regularized forcing along the boundary, they require rich geometric information about the interface and its location relative to the background grid. Sharp IB methods also require the grid nodes to be classified as “IB”, “fluid”, and “solid” nodes, which increases the coding complexity, especially for large scale distributed computing models. In particular, for moving body problems, the node classification has to be reevaluated at every time step.

This work uses a direct forcing immersed boundary method [26,6,27] to model flows around complex bodies with prescribed kinematics. This is motivated by vehicular aerodynamics applications, for the which the “dirty” CAD geometry (which may include features such as sharp or thin edges, or artifacts such as holes) of the vehicle is relatively easier to handle via diffuse-interface IB methods than their sharp-interface counterparts [28,29].

Various approaches have been employed in the literature to construct regularized delta functions for diffuse IB methods. A common approach is to construct them by imposing certain discrete moment conditions. This approach was pioneered by Peskin [1] and has been extended in subsequent work, including by Yang et al. [30] and Stein et al. [31]. Another approach to constructing kernels functions is to employ the moving least squares (MLS) methodology [32]. The MLS technique can be used to construct interpolation kernels (which are referred to as generating functions in the MLS literature) for a particular Lagrangian point dynamically by solving a weighted least squares problem defined over a set of positions (interpolation points) where data are sampled for interpolation (to locations on the Lagrangian mesh). The generating functions produced by the MLS procedure are constructed to satisfy prescribed conditions, such as the polynomial reproduction constraints, which are equivalent to discrete moment conditions like those suggested by Peskin. One of the strengths of the MLS method is that it does not require the interpolation points to be arranged in a particular way around a Lagrangian point. This makes the MLS technique quite general and even applicable for unstructured grid IB methods [33,34]. As typically done in regular IB methods, prior IB/MLS works have also used a full support of interpolation points around a Lagrangian marker. In contrast, we employ the MLS technique to construct *one-sided* regularized delta functions for the direct forcing IB method in this work.

The MLS methodology has been extensively used in the mesh-free continuum mechanics literature [35–38]. Its use with the direct forcing IB method was introduced by Vanella and Balaras [27]. Recently, Li et al. [39] and Tullio and Pascasio [40] have also used the MLS technique of Vanella and Balaras to construct regularized IB kernels on structured Eulerian meshes. In prior studies [27,39,40], the MLS technique was used to transform cubic spline weight functions into regularized kernel functions that satisfy zeroth- and first-order discrete moment conditions. Because the fluid is described using a regular Cartesian grid, a Cartesian arrangement of interpolation points across the interface was used in defining the least squares problem. It is important to notice, however, that if the weight function already satisfies the conditions imposed by the MLS construction, then the generating function produced by the MLS procedure is the same as the weight function. For example, Peskin’s kernel functions and certain spline functions, already satisfy the reproducing properties, and hence are not modified by the MLS construction. We prove this property of MLS weight generation in Sec. 3. Consequently, there is no advantage to using a MLS construction with weighting functions that already satisfy the desired properties unless the construction aims to reduce the support of the weighting function. In contrast, if the MLS kernel is to be supported on a region that does not contain the full support of the basic kernel function, the MLS kernel will generally be different from the basic kernel. This approach allows us to interpolate velocity from and spread force to only one side of the interface (see Fig. 1).

We remark that although the one-sided generating functions will satisfy the desired moment conditions, they will generally include large over- and undershoots, even if the original weights are non-negative and monotonically decreasing away from the Lagrangian point. For diffuse-interface IB methods, the positive and negative values in the generating functions can produce oscillatory feedback through force spreading operation that can destabilize the FSI simulation. By construction, the velocity interpolation operation is not affected by (possibly larger-in-magnitude) negative weights. The source of the



**Fig. 1.** Demarcation of Eulerian domain  $\Omega = \Omega_b^- \cup \Omega_b^+$  based on the location of the interface  $\partial\Omega_b$ . Lagrangian marker points depicted as [●] on the interface interpolate velocity from and spread force to  $\Omega_b^-$  and  $\Omega_b^+$  regions separately. (a) A self-closing interface partitioning the domain into interior  $\Omega_b^-$  and exterior  $\Omega_b^+$  regions. (b) A hypothetical plane (shown with a dashed line) passing through an open interface and partitioning the domain into interior and exterior regions. The choice of the hypothetical plane for an open interface is arbitrary.

aforementioned instability can be understood as follows. For the direct forcing IB method, the feedback force is defined in terms of a slip velocity at a Lagrangian point. If the weights used in the force spreading operator are of similar magnitude, but with opposite sign, the Eulerian representation of the Lagrangian feedback force will be spatially oscillatory. This causes the numerical error to grow over time. We remark that for exactly-constrained IB methods, wherein the Lagrangian forces are computed by solving an extended saddle-point system [41,42], the (larger magnitude of) positive and negative weights do not affect the stability of the system — the IB force is dynamically adjusted to satisfy the imposed kinematic constraints. To overcome this issue, this paper introduces two shifting techniques to make weights non-negative and to reduce the overshoots for the one-sided kernels. A side-effect of the shifting procedure is that the first and other higher-order moment conditions are not satisfied in general, and as a result, the shifted one-sided kernels are limited to first-order accurate reconstructions.

One-sided MLS kernels have also been used by Le and Khoo [43] and Mohammadi et al. [44] in the context of sharp-interface IB methods. These developments are relatively recent. In their approach, velocity reconstruction at “IB forcing nodes” is done via an MLS method instead of grid-based interpolations. For a given IB forcing node (which is defined to be near the solid interface and towards the fluid side), nearby Eulerian fluid nodes and interfacial Lagrangian nodes are selected as a set of interpolation points to carry out the velocity reconstruction process. Internal solid nodes and neighboring IB forcing nodes are excluded from this interpolation set.<sup>1</sup> Le and Khoo [43] and Mohammadi et al. [44] found MLS-based reconstructions to be more robust than interpolating the velocity along a specific direction, such as grid-aligned directions or local normal directions and as commonly used with sharp-interface IB schemes. Their MLS-based sharp-interface IB method also retains second-order accuracy. We remark that there is a fundamental difference in the MLS interpolation procedure for sharp-interface IB methods and our current diffuse-interface IB method. For sharp-interface IB methods, the interfacial Lagrangian nodes participate in the interpolation process because they are not considered to be forcing points, and IB nodes near the solid interface and towards the fluid side act as forcing points. For diffuse-interface IB methods, the interfacial Lagrangian nodes are the forcing points, and therefore, only neighboring Eulerian fluid nodes are taken in the support of the interpolating kernel. The reason for exclusion stems from the fact that the range of spreading operator, which is typically constructed to be the adjoint of interpolation operator, is restricted to the fluid nodes. Conversely, the domain of the interpolation operator is also restricted to the same set of Eulerian fluid nodes.

Specialized forms of one-sided IB kernels dealing with interactions between immersed structures and the boundaries of the computational domain have also been proposed in the literature. For example, Griffith and co-workers algebraically derived one-sided regularized delta functions that continuously transition to Peskin’s four-point kernel as the structure position moves away from the boundary of the computational domain [45]. As in the simplest version of the MLS weighting functions constructed herein, this one-sided kernel included both positive and negative weights. An alternative approach was proposed by Yeo and Maxey to deal with boundaries of the computational domain [46]; see also Delong et al. [47]. However, these constructions do not deal with one-sided constructions along general (curved) internal boundaries, which is the focus of this paper. Moreover, the current MLS approach can also naturally handle immersed structures near the computational domain boundaries.

<sup>1</sup> These nodes are also excluded from grid-based interpolation schemes that are typically used in sharp IB methods.

## 2. Equations of motion

### 2.1. Immersed boundary method

We begin by describing the equations of motion of a rigid body moving in an incompressible viscous fluid of constant density  $\rho$  and viscosity  $\mu$  in a region  $\Omega \subset \mathbb{R}^d$ ,  $d = 2$  or  $3$ . In the immersed boundary formulation, a Lagrangian description is employed for the structural location and forces, and an Eulerian description is employed for the fluid velocity and pressure. We denote the position of the boundary of the Lagrangian structure  $\partial\Omega_b(t) \subset \Omega$  at time  $t$  by  $\mathbf{X}(\mathbf{s}, t)$ , in which  $\mathbf{s} = (s_1, \dots, s_{d-1}) \in U$  denotes a fixed material coordinate system attached to the boundary, and  $U \subset \mathbb{R}^{d-1}$  is the Lagrangian curvilinear coordinate domain. The fixed physical domain is fully described by the Cartesian coordinates  $\mathbf{x} = (x_1, \dots, x_d) \in \Omega$ . The open or closed boundary of the immersed structure partitions the Cartesian domain into two regions  $\Omega = \Omega_b^+(t) \cup \Omega_b^-(t)$  (see Fig. 1).

The combined equations of motion for fluid-structure system are [1,6]

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t) + \mathbf{u}(\mathbf{x}, t) \cdot \nabla \mathbf{u}(\mathbf{x}, t) \right) = -\nabla p(\mathbf{x}, t) + \mu \nabla^2 \mathbf{u}(\mathbf{x}, t) + \mathbf{f}(\mathbf{x}, t), \quad (1)$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0, \quad (2)$$

$$\mathbf{f}(\mathbf{x}, t) = \int_U \mathbf{F}(\mathbf{s}, t) \delta(\mathbf{x} - \mathbf{X}(\mathbf{s}, t)) d\mathbf{s}, \quad (3)$$

$$\mathbf{U}(\mathbf{s}, t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(\mathbf{s}, t)) d\mathbf{x}, \quad (4)$$

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{s}, t) = \mathbf{U}(\mathbf{s}, t). \quad (5)$$

Eqs. (1) and (2) are the incompressible Navier-Stokes equations written in Eulerian form, in which  $\mathbf{u}(\mathbf{x}, t)$  is the velocity and  $p(\mathbf{x}, t)$  is the pressure. Eq. (3) converts the Lagrangian force density  $\mathbf{F}(\mathbf{s}, t)$  into an equivalent Eulerian density  $\mathbf{f}(\mathbf{x}, t)$ . Conversely, Eq. (4) determines the physical velocity of each Lagrangian material point from the Eulerian velocity field, so that the immersed structure moves according to the local value of the velocity field  $\mathbf{u}(\mathbf{x}, t)$  (Eq. (5)). The Lagrangian and Eulerian quantities of Eqs. (3) and (4) are mediated by integral equations using Dirac delta function, which is taken to be  $d$ -dimensional tensor product of one-dimensional delta functions:  $\delta(\mathbf{x}) = \prod_{i=1}^d \delta(x_i)$ . In this formulation, the Lagrangian force density  $\mathbf{F}(\mathbf{s}, t)$  is a Lagrange multiplier that ensures that the body moves according to a prescribed velocity.

### 2.2. Discrete equations of motion

#### 2.2.1. Lagrangian discretization

In conventional IB methods, Eqs. (3) and (4) are approximated by replacing the singular delta function kernel by a regularized delta function  $\delta_h(\mathbf{x})$  that is supported on both sides of the interface  $\partial\Omega_b(t)$ . Using a short-hand notation, we denote the force spreading operation by  $\mathbf{f} = \mathcal{S}_{\text{IB}}[\mathbf{X}]\mathbf{F}$ , in which  $\mathcal{S}_{\text{IB}}[\mathbf{X}]$  is the *force-spreading operator* associated with the boundary configuration and regularized delta function. The velocity interpolation operation as defined in Eq. (4) is expressed as  $\frac{\partial \mathbf{X}}{\partial t} = \mathbf{U} = \mathcal{J}_{\text{IB}}[\mathbf{X}]\mathbf{u}$ , in which  $\mathcal{J}_{\text{IB}}[\mathbf{X}]$  is the *velocity-interpolation operator*. It can be shown that if  $\mathcal{S}_{\text{IB}}$  and  $\mathcal{J}_{\text{IB}}$  are adjoint operators, i.e.,  $\mathcal{S}_{\text{IB}} = \mathcal{J}_{\text{IB}}^*$ , then Lagrangian-Eulerian coupling conserves energy [1]. Later in Sec. 3.2 we will obtain spreading and interpolation operators using generating functions produced by the MLS technique, which we denote by  $\mathcal{S}_{\text{MLS}}$  and  $\mathcal{J}_{\text{MLS}}$ , respectively. Generic force-spreading and velocity-interpolation operators are denoted  $\mathcal{S}$  and  $\mathcal{J}$ , respectively.

We impose velocity boundary conditions only along the fluid-structure interface, and the volume enclosed by the closed interface is not constrained. Discretely, Lagrangian markers with curvilinear mesh spacing  $(\Delta s_1, \Delta s_2)$  represent the codimension-1 interface  $\partial\Omega_b$ ; see Fig. 1. The Lagrangian markers are indexed by the tuple  $(l, m)$ . Based on our prior experience with direct forcing IB method, we typically take the Lagrangian marker spacing approximately equal to the Eulerian grid spacing on the finest grid level, i.e.,  $\Delta s_1 \approx \Delta s_2 \approx h$ , in which  $h$  is the uniform grid cell size on the finest level of the locally refined grid. A discrete approximation to any general quantity defined on marker points is described by  $\Phi_{l,m}^n \approx \Phi(\mathbf{s}_{l,m}, t^n) = \Phi(l\Delta s_1, m\Delta s_2, t^n)$  at time  $t^n$ . More specifically, the position, velocity, and force of a marker point are denoted as  $\mathbf{X}_{l,m}$ ,  $\mathbf{U}_{l,m}$ , and  $\mathbf{F}_{l,m}$ , respectively. Using the regularized IB kernel  $\delta_h$ , the discrete velocity interpolation of the collocated grid fluid velocity  $\mathbf{u} \equiv (u, v, w)$  onto a specific configuration of Lagrangian markers (i.e.,  $\mathbf{U} \equiv (U, V, W) = \mathcal{J}_{\text{IB}}[\mathbf{X}]\mathbf{u}$ ) reads

$$U_{l,m} = \sum_{\mathbf{x}_{i,j,k} \in \Omega} u_{i,j,k} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}_{l,m}) \Delta x \Delta y \Delta z, \quad (6)$$

$$V_{l,m} = \sum_{\mathbf{x}_{i,j,k} \in \Omega} v_{i,j,k} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}_{l,m}) \Delta x \Delta y \Delta z, \quad (7)$$

$$W_{l,m} = \sum_{\mathbf{x}_{i,j,k} \in \Omega} w_{i,j,k} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}_{l,m}) \Delta x \Delta y \Delta z, \quad (8)$$

in which the tuple  $(i, j, k)$  is the Cartesian grid cell index and  $\mathbf{x}_{i,j,k}$  is the spatial location of the cell centroid. Conversely, the discrete spreading of Lagrangian force density  $\mathbf{F} \equiv (F_1, F_2, F_3)$  onto cell centers of the collocated grid (i.e.,  $\mathbf{f} \equiv (f_1, f_2, f_3) = \mathcal{S}_{\text{IB}}[\mathbf{X}]\mathbf{F}$ ) reads

$$(f_1)_{i,j,k} = \sum_{\mathbf{x}_{l,m} \in \Omega_b} (F_1)_{l,m} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}_{l,m}) \Delta s_1 \Delta s_2, \quad (9)$$

$$(f_2)_{i,j,k} = \sum_{\mathbf{x}_{l,m} \in \Omega_b} (F_2)_{l,m} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}_{l,m}) \Delta s_1 \Delta s_2, \quad (10)$$

$$(f_3)_{i,j,k} = \sum_{\mathbf{x}_{l,m} \in \Omega_b} (F_3)_{l,m} \delta_h(\mathbf{x}_{i,j,k} - \mathbf{X}_{l,m}) \Delta s_1 \Delta s_2. \quad (11)$$

The regularized IB kernel  $\delta_h$  interpolates from and spreads to both sides of the interface  $\partial\Omega_b$ . Note that the arguments of an IB kernel  $\delta_h$  depend only on the difference between Eulerian and Lagrangian locations, i.e.,  $\delta_h = \delta_h(\mathbf{x} - \mathbf{X})$ , and not the locations themselves; see Eqs. (6)–(11). This property implies that the weights of an IB kernel are independent of the Lagrangian marker identity and the same kernel is used for each marker. In Sec. 3, we describe a procedure to obtain the one-sided version of IB kernel  $\psi_h(\mathbf{x}, \mathbf{X})$  using the moving least squares method whose weights depend upon the Lagrangian marker position explicitly. In particular, a different kernel function is generated for each marker.

### 2.2.2. Eulerian discretization and time-stepping scheme

In this work, a collocated grid discretization for the momentum and continuity equations is used, in which the Eulerian velocity, pressure, and force variables are defined at the centers of Cartesian grid cells of grid spacing  $h$ . Second-order finite differences are used to approximate the Eulerian equations on locally refined grids [28,6]. The spatially discretized cell-centered operators are denoted with a ‘cc’ subscript. To avoid the velocity and pressure decoupling on collocated grid [48], auxiliary face-centered variables and operators are introduced, which are distinguished using a ‘fc’ subscript.

A version of the second-order accurate pressure projection algorithm of Brown et al. [49] is used to solve the incompressible Navier-Stokes system, and the direct forcing approach of Bhalla et al. [6] is used to approximately impose the rigidity constraint of the immersed body. A fixed-point iteration time stepping scheme using  $n_{\text{cycles}} = 2$  cycles per time step is used to evolve quantities from time level  $t^n$  to time level  $t^{n+1} = t^n + \Delta t$ . A superscript “ $k$ ” is used to denote the cycle number of the fixed-point iteration. At the beginning of each time step, the solutions from the previous time step are used to initialize cycle  $k = 0$ :  $\mathbf{u}^{n+1,0} = \mathbf{u}^n$  and  $p^{n+\frac{1}{2},0} = p^{n-\frac{1}{2}}$ . At the initial time  $n = 0$ , the Eulerian velocity is prescribed via an initial condition, and the Eulerian pressure is taken to be zero.

With  $\mathbf{N} = (\mathbf{u} \cdot \nabla_h \mathbf{u})$  denoting the non-linear convective term, the equations of motion are

$$\rho \left( \frac{\mathbf{u}^{n+1,k+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}^{n+\frac{1}{2},k} \right) = -\nabla_h p^{n+\frac{1}{2},k+1} + \mu \nabla_h^2 \left( \frac{\mathbf{u}^{n+1,k+1} + \mathbf{u}^n}{2} \right) + \mathbf{f}^{n+\frac{1}{2},k+1}, \quad (12)$$

$$\nabla_h \cdot \mathbf{u}^{n+1,k+1} = 0, \quad (13)$$

and are integrated from time step  $n$  to  $n + 1$  in an operator-splitting manner as follows:

1. An intermediate velocity  $\tilde{\mathbf{u}}_*^{n+1,k+1}$  is obtained by integrating the advection-diffusion momentum equation, which ignores the pressure and the constraint forces

$$\rho \left( \frac{\tilde{\mathbf{u}}_*^{n+1,k+1} - \mathbf{u}^n}{\Delta t} + \mathbf{N}^{n+\frac{1}{2},k} \right) = \mu \nabla_h^2 \left( \frac{\tilde{\mathbf{u}}_*^{n+1,k+1} + \mathbf{u}^n}{2} \right). \quad (14)$$

A geometric multigrid solver using a Gauss-Seidel smoother is employed to solve for  $\tilde{\mathbf{u}}_*^{n+1,k+1}$  on a locally refined grid. For the first ( $k = 0$ ) cycle, the nonlinear convective term is discretized using the explicit Adams-Bashforth scheme, so that  $\mathbf{N}^{n+\frac{1}{2},0} = \frac{3}{2}(\mathbf{u} \cdot \nabla_h \mathbf{u})^n - \frac{1}{2}(\mathbf{u} \cdot \nabla_h \mathbf{u})^{n-1}$ , while for the remaining cycles, a midpoint approximation is used, so that  $\mathbf{N}^{n+\frac{1}{2},k} = (\mathbf{u} \cdot \nabla_h \mathbf{u})^{n+\frac{1}{2},k}$ , in which  $\mathbf{u}^{n+\frac{1}{2},k} = \frac{1}{2}(\mathbf{u}^{n+1,k} + \mathbf{u}^n)$ .

2. Impose the rigidity constraint of the immersed body by computing the constraint force  $\mathbf{F}$

$$\Delta \mathbf{U}^{n+\frac{1}{2},k+1} = \mathbf{U}_b^{n+\frac{1}{2}} - \mathcal{J}[\mathbf{X}] \left( \frac{\tilde{\mathbf{u}}_*^{n+1,k+1} + \mathbf{u}^n}{2} \right), \quad (15)$$

$$\tilde{\mathbf{u}}^{n+1,k+1} = \tilde{\mathbf{u}}_*^{n+1,k+1} + \mathcal{S}[\mathbf{X}] \Delta \mathbf{U}^{n+\frac{1}{2},k+1}, \quad (16)$$

$$\mathbf{F}^{n+\frac{1}{2},k+1} = \frac{\rho}{\Delta t} \Delta \mathbf{U}^{n+\frac{1}{2},k+1}, \quad (17)$$

$$\mathbf{f}^{n+\frac{1}{2},k+1} = \mathcal{S}[\mathbf{X}] \mathbf{F}^{n+\frac{1}{2},k+1}, \quad (18)$$

in which  $\mathbf{U}_b$  is the desired rigid body velocity of the structure, and  $\Delta \mathbf{U}$  is the slip velocity computed from the interpolated (unconstrained) fluid velocity on the Lagrangian markers. In our implementation, we directly update the velocity field  $\tilde{\mathbf{u}}^{n+1,k+1}$  using Eq. (16) on the Eulerian grid, without explicitly calculating  $\mathbf{f}^{n+\frac{1}{2},k+1}$  as given in Eq. (18). This equivalence can be readily verified by considering the operator-split between momentum and rigidity constraint equations [6]. However, Eq. (18) is useful for computing the net hydrodynamic forces on the immersed structure [50]. At this stage, the fluid velocity  $\tilde{\mathbf{u}}^{n+1,k+1}$  accounts for the rigidity of the structure, but not the incompressibility of the system. This is corrected in the next step.

3. Impose the incompressibility constraint by solving the pressure Poisson equation for the auxiliary variable  $\phi$  and estimate the fluid pressure

$$\tilde{\mathbf{u}}_{\text{fc}}^{n+1,k+1} = \mathbb{I}(\tilde{\mathbf{u}}^{n+1,k+1}), \quad (19)$$

$$-\nabla^2 \phi^{n+1,k+1} = -\frac{\rho}{\Delta t} \nabla \cdot \tilde{\mathbf{u}}_{\text{fc}}^{n+1,k+1}, \quad (20)$$

$$\mathbf{u}^{n+1,k+1} = \tilde{\mathbf{u}}^{n+1,k+1} - \frac{\Delta t}{\rho} \nabla \phi_{\text{cc}}^{n+1,k+1}, \quad (21)$$

$$\mathbf{u}_{\text{fc}}^{n+1,k+1} = \tilde{\mathbf{u}}_{\text{fc}}^{n+1,k+1} - \frac{\Delta t}{\rho} \nabla \phi_{\text{fc}}^{n+1,k+1}, \quad (22)$$

$$p^{n+\frac{1}{2},k+1} = \phi^{n+1,k+1} - \frac{\Delta t}{\rho} \frac{\mu}{2} \nabla^2 \phi_{\text{cc}}^{n+1,k+1}, \quad (23)$$

in which  $\mathbb{I}$  is the interpolation operator that determines the face-centered velocity field  $\mathbf{u}_{\text{fc}}^{n+1,k+1}$  by averaging the adjacent cell-centered velocity components.

We use adaptive mesh refinement (AMR) framework for some of the cases presented in Sec. 5. A grid with  $\ell_{\max}$  refinement levels has grid spacings  $\Delta x_0$ ,  $\Delta y_0$ , and  $\Delta z_0$  on the coarsest grid level and grid spacings  $\Delta x = \Delta x_0/n_{\text{ref}}^{\ell-1}$ ,  $\Delta y = \Delta y_0/n_{\text{ref}}^{\ell-1}$ , and  $\Delta z = \Delta z_0/n_{\text{ref}}^{\ell-1}$  on a grid level  $\ell$ , in which,  $n_{\text{ref}}$  is the integer refinement ratio. The Lagrangian mesh is embedded on the finest grid level to adequately resolve the thin boundary layers. For all of the cases considered in this work a constant time step size  $\Delta t = \min(\Delta t^\ell)$  is chosen, in which the time step size  $\Delta t^\ell$  on grid level  $\ell$  satisfies the convective CFL condition  $\Delta t^\ell \leq C \min\left(\frac{\Delta x}{\|u_x\|_\infty}, \frac{\Delta y}{\|u_y\|_\infty}, \frac{\Delta z}{\|u_z\|_\infty}\right)^\ell$ . In this work, the convective CFL number is set to  $C = 0.1$  unless otherwise stated.

### 3. Moving least squares method

#### 3.1. Review

The Backus-Gilbert formulation of the moving least squares method seeks the quasi-interpolant

$$\mathcal{P}g(\mathbf{X}) = \sum_i^N g(\mathbf{x}_i) \psi_i(\mathbf{X}), \quad (24)$$

in which  $\mathbf{g} = [g(\mathbf{x}_1), \dots, g(\mathbf{x}_N)]^T$  are given data at  $N$  interpolation points,  $\mathbf{X}$  is the evaluation point, and  $\mathcal{P}$  is the interpolation operator. The moving least squares method computes generating functions  $\psi_i(\mathbf{X}) = \{\psi(\mathbf{x}_i, \mathbf{X})\}$  subject to the polynomial reproduction constraints

$$\sum_{i=1}^N p(\mathbf{x}_i) \psi_i(\mathbf{X}) = p(\mathbf{X}), \quad \text{for all } p \in \Pi_d^s, \quad (25)$$

in which  $\Pi_d^s$  is the space of  $s$ -variate polynomials (centered around any arbitrary point) of total degree at most  $d$ . The polynomial reproduction constraints correspond to discrete moment conditions for the function  $\psi_i(\mathbf{X})$ . Eq. (25) can be written in a matrix form as

$$\mathcal{A}\Psi(\mathbf{X}) = \mathbf{P}(\mathbf{X}), \quad (26)$$

in which the entries of the polynomial matrix  $\mathcal{A} \in \mathbb{R}^{m \times N}$  are the values of the basis functions at the data point locations,  $\mathcal{A}_{ij} = p_i(\mathbf{x}_j)$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, N$ , and the right-hand side vector  $\mathbf{P} = [p_1, \dots, p_m]^T$  contains the values of the



polynomials at the evaluation point  $\mathbf{X}$ . The unknown generating function vector  $\Psi = [\psi_1, \dots, \psi_N]^T$  is obtained by solving a *least squares problem*. Since the set of generating functions changes by considering a different evaluation point  $\mathbf{Y} \neq \mathbf{X}$ ,  $\{\psi_i(\mathbf{Y})\} \neq \{\psi_i(\mathbf{X})\}$ , this least squares procedure is called the *moving least squares problem*.

Generally  $N \gg m$ , implying that Eq. (26) is an underdetermined system which can be solved in a weighted least squares sense with the help of Lagrange multipliers  $\lambda(\mathbf{X})$  to enforce the reproducing conditions. Specifically, the Backus-Gilbert MLS problem is solved by posing it as a constrained quadratic minimization problem, which reads as

$$\text{Backus-Gilbert MLS: } \begin{cases} \min_{\Psi(\mathbf{X}) \in \mathbb{R}^N} & J = \frac{1}{2} \Psi^T(\mathbf{X}) \mathcal{W}^{-1}(\mathbf{X}) \Psi(\mathbf{X}) \\ \text{subject to:} & \mathcal{A} \Psi(\mathbf{X}) = \mathbf{P}(\mathbf{X}). \end{cases} \quad (27)$$

Here,  $\mathcal{W}(\mathbf{X}) = \text{diag}(\mathbf{W})$  is the diagonal matrix containing weights of the interpolation points with respect to the evaluation point, and  $\mathbf{W} = (W(\mathbf{x}_1, \mathbf{X}), \dots, W(\mathbf{x}_N, \mathbf{X}))$  is the main diagonal of  $\mathcal{W}(\mathbf{X})$ . The weights are defined by a nonnegative weight function  $W(\mathbf{x}_i, \mathbf{X})$  that decreases in magnitude with distance away from the evaluation point  $\mathbf{X}$ . Eq. (27) is solved by minimizing the Lagrangian  $\mathcal{L}$  of the problem, defined using the Lagrange multiplier  $\lambda(\mathbf{X}) \in \mathbb{R}^m$  as

$$\mathcal{L} = \frac{1}{2} \Psi^T(\mathbf{X}) \mathcal{W}^{-1}(\mathbf{X}) \Psi(\mathbf{X}) - \lambda^T(\mathbf{X}) [\mathcal{A} \Psi(\mathbf{X}) - \mathbf{P}(\mathbf{X})]. \quad (28)$$

The Lagrangian is minimized by using the essential conditions of extrema,  $\frac{\partial \mathcal{L}}{\partial \Psi} = \mathbf{0}$  and  $\frac{\partial \mathcal{L}}{\partial \lambda} = \mathbf{0}$ , which yields

$$\mathcal{W}^{-1} \Psi - \mathcal{A}^T \lambda = \mathbf{0}, \quad (29)$$

$$\mathcal{A} \Psi - \mathbf{P} = \mathbf{0}. \quad (30)$$

Solving the above two equations, we obtain

$$\lambda = (\mathcal{A} \mathcal{W} \mathcal{A}^T)^{-1} \mathbf{P} = \mathcal{G}^{-1} \mathbf{P}, \quad (31)$$

$$\Psi = \mathcal{W} \mathcal{A}^T \lambda = \mathcal{W} \mathcal{A}^T (\mathcal{A} \mathcal{W} \mathcal{A}^T)^{-1} \mathbf{P} = \mathcal{W} \mathcal{A}^T \mathcal{G}^{-1} \mathbf{P}. \quad (32)$$

In Eqs. (31) and (32),  $\mathcal{G}(\mathbf{X}) = \mathcal{A} \mathcal{W} \mathcal{A}^T \in \mathbb{R}^{m \times m}$  is the symmetric positive-definite Gram matrix, whose entries are the weighted  $L^2$  inner products of the polynomials

$$\mathcal{G}_{jk}(\mathbf{X}) = \langle p_j, p_k \rangle_{W(\mathbf{X})} = \sum_{i=1}^N p_j(\mathbf{x}_i) p_k(\mathbf{x}_i) W(\mathbf{x}_i, \mathbf{X}), \quad j, k = 1, \dots, m. \quad (33)$$

It is also instructive to write the generating functions (obtained in Eq. (32)) in component form as

$$\psi(\mathbf{x}_i, \mathbf{X}) = \psi_i(\mathbf{X}) = W(\mathbf{x}_i, \mathbf{X}) \sum_{j=1}^m \lambda_j(\mathbf{X}) p_j(\mathbf{x}_i), \quad i = 1, \dots, N. \quad (34)$$

In matrix notation, Eq. (34) can be written as

$$\Psi(\mathbf{X}) = \mathcal{W}(\mathbf{X}) \odot \mathcal{L}(\mathbf{X}), \quad (35)$$

in which  $\mathcal{L}(\mathbf{X}) = \mathcal{A}^T \lambda(\mathbf{X})$  and  $\odot$  indicates the Hadamard (component-wise) product of two matrices.

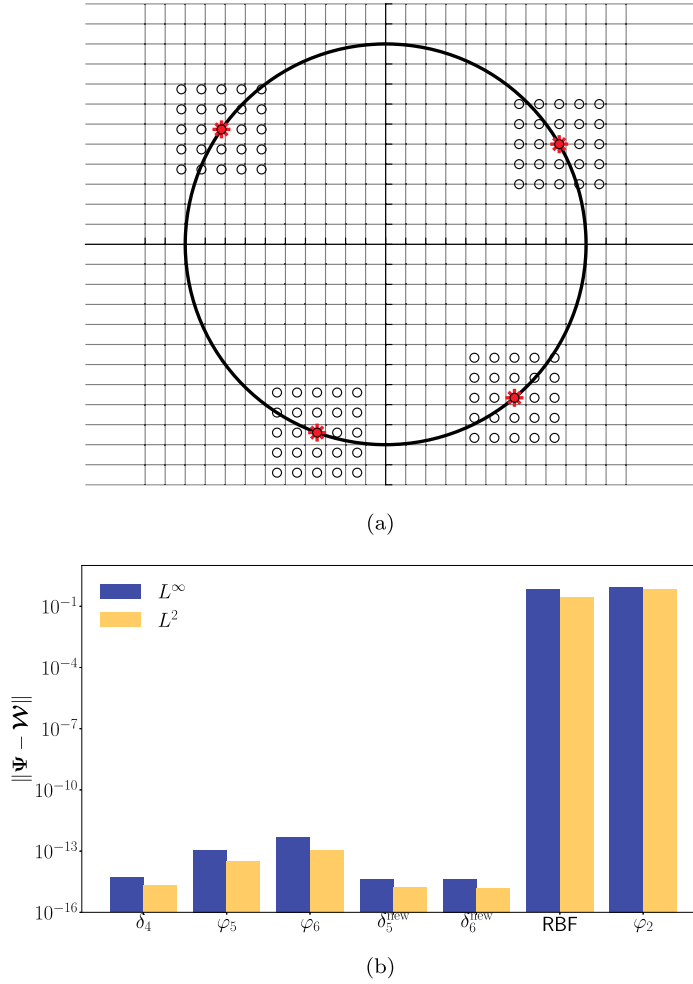
In the IB literature, the standard formulation of the MLS problem is more popular [27,40,33,34,43,44] compared to the Backus-Gilbert theory [51]. In the standard formulation of MLS, the quasi-interpolant  $\mathcal{P}g$  to  $g$  is expressed as

$$\mathcal{P}g(\mathbf{X}) = \sum_{j=1}^m c_j(\mathbf{X}) p_j(\mathbf{X}) = \mathbf{P}^T(\mathbf{X}) \mathbf{C}(\mathbf{X}), \quad (36)$$

and the unknown coefficient vector  $\mathbf{C}(\mathbf{X})$  appearing in Eq. (36) above is found by minimizing the weighted  $L^2$ -norm of the error function  $J(\mathbf{C})$ :

$$J(\mathbf{C}(\mathbf{X})) = \sum_{i=1}^N W(\mathbf{x}_i, \mathbf{X}) [\mathbf{P}^T(\mathbf{x}_i) \mathbf{C}(\mathbf{X}) - g(\mathbf{x}_i)]^2, \quad (37)$$

with respect to  $\mathbf{C}$ , i.e., by setting  $\frac{\partial J}{\partial \mathbf{C}} = 0$ . However, both formulations ultimately produce the same generating function  $\psi(\mathbf{x}_i, \mathbf{X})$ . There are two main advantages of the Backus-Gilbert MLS formulation: (i) the polynomial reproducing conditions are directly included in the problem formulation; and (ii) the unknown vector  $\Psi$  is independent of the data values  $\mathbf{g}$ , unlike the coefficient vector  $\mathbf{C}$ , which implicitly depends upon the sampled data values, i.e.,  $\mathbf{C} = \mathbf{C}(\mathbf{X}; \mathbf{g})$ .



**Fig. 2.** (a) Representative Lagrangian markers (\*) on a circular interface embedded in a Cartesian grid, and full support of the kernel (o) around marker points. (b) Error norms of  $\|\Psi - \mathcal{W}\|$  for various kernels.

As an example, consider the univariate polynomials in two spatial dimensions, i.e.,  $d = 2$  and  $s = 1$ , so that  $m = 3$ . The univariate basis functions defined relative to an evaluation point are  $p(\mathbf{x}) = (p_1(\mathbf{x} - \mathbf{X}), p_2(\mathbf{x} - \mathbf{X}), p_3(\mathbf{x} - \mathbf{X})) = (1, x - X, y - Y)$ , with  $\mathbf{X} = (X, Y)$  denoting the coordinates of the evaluation point  $\mathbf{X}$ . In the context of the immersed boundary method,  $p_1$  polynomial (constant) reproducing condition implies that Lagrangian and Eulerian forms of force and power are equivalent. Furthermore,  $p_2$  and  $p_3$  polynomial (linear) reproducing conditions make Lagrangian and Eulerian representations of torque equivalent [1]. Notice that defining the polynomial basis function relative to the evaluation point  $\mathbf{X}$  makes the right-hand of Eq. (26) a constant vector, i.e.,

$$\mathbf{P}(\mathbf{X}) = p(\mathbf{x})|_{\mathbf{x}=\mathbf{X}} = (1, 0, 0)^T, \quad (38)$$

independent of  $\mathbf{X}$ .

**Lemma 1.** *If the weight functions already satisfy the reproducing conditions then  $\mathcal{L}$  is a matrix of ones.*

**Proof.** In this case the weight functions are the generating functions  $\Psi(\mathbf{X}) = \mathcal{W}(\mathbf{X})$ . Therefore, from Eq. (35)  $\mathcal{L}_{ij} = 1$  follows.  $\square$

Next, we consider some commonly used weight functions  $\mathcal{W}$  in the IB literature to generate MLS weights  $\Psi$  for four representative points on a circular interface (see Fig. 2(a)). The chosen weight functions are: (i) Peskin's four-point delta kernel  $\delta_4$ ; (ii) new five-point  $\delta_5^{\text{new}}$  and (iii) six-point  $\delta_6^{\text{new}}$  delta kernels; (iv) five-point  $\varphi_5$  and (v) six-point  $\varphi_6$  spline functions; (vi) two-point cubic spline function  $\varphi_2$  as used in Vanella and Balaras [27]; and (vii) four-point radial basis function (RBF). The one-dimensional functional form of these kernels are provided in Appendix A. Out of these kernels, the



first five kernels satisfy the zeroth- and first-order moment conditions, when they have a full support around the Lagrangian marker — as in this case. Fig. 2(b) plots  $L^2$ - and  $L^\infty$ -norms of difference in  $\mathcal{W}$  and  $\Psi$  for the four Lagrangian points. As observed in the figure, the MLS procedure does not transform the weights of those kernels that already satisfy the moment conditions it tries to impose. The numerical results shown in Fig. 2(b) are consistent with Lemma 1.

### 3.2. One-sided IB kernels

To obtain one-sided IB kernel  $\psi_h$  using the MLS procedure, we use the Heaviside function  $H(\mathbf{x})$

$$H_{i,j,k} = \begin{cases} 0, & \Omega_b^-, \\ 1 & \Omega_b^+ \cup \partial\Omega_b, \end{cases} \quad (39)$$

to select the appropriate side of the interface  $\partial\Omega_b$  in order to define the domain of influence of the evaluation/Lagrangian point  $\mathbf{X}$ . This is achieved by multiplying the standard/unrestricted weights by the Heaviside function to obtain restricted weights  $W_{\text{MLS}}$

$$W_{\text{MLS}}(\mathbf{x}_i, \mathbf{X}) = W(\mathbf{x}_i, \mathbf{X})H(\mathbf{x}_i) \quad \text{s.t.} \quad W_{\text{MLS}}(\mathbf{x}_i, \mathbf{X}) = 0 \quad \forall \quad \mathbf{x}_i \in \Omega_b^-. \quad (40)$$

It can be easily verified that by using  $W_{\text{MLS}}$  in Eq. (34),  $\psi_h(\mathbf{x}_i, \mathbf{X}) = 0 \quad \forall \quad \mathbf{x}_i \in \Omega_b^-$ . Therefore, by using a simple weight manipulation process we can easily control the domain of influence or the interaction region of point  $\mathbf{X}$ .

Many choices for the weight function have been introduced in the MLS literature, including the Gaussian function, B-splines, Wendland function, inverse distance square function, and others. In this work we use four-point, and the new/modified five-point and the six-point Peskin's delta functions; the new kernels remove the negative tails of the original Peskin kernels by imposing a weaker second moment condition. Because Peskin's delta functions are specifically constructed to satisfy the zeroth- and first-order moment conditions, they automatically qualify as the generating function when the domain of influence of the evaluation point  $\mathbf{X}$  is not restricted. Furthermore, Peskin's delta functions show good grid translation invariance property in spite of a compact support [52]. In Sec. 5, we also compare the performance of Peskin's delta function weights to B-splines weights.

Finally, using the one-sided IB kernel  $\psi_h$ , the discrete velocity interpolation of the fluid velocity onto Lagrangian markers (i.e.,  $\mathbf{U} = \mathcal{J}_{\text{MLS}}[\mathbf{X}]\mathbf{u}$ ) reads

$$U_{l,m} = \sum_{\mathbf{x}_{i,j,k} \in \Omega} u_{i,j,k} \psi_h(\mathbf{x}_{i,j,k}, \mathbf{X}_{l,m}) \Delta x \Delta y \Delta z, \quad (41)$$

$$V_{l,m} = \sum_{\mathbf{x}_{i,j,k} \in \Omega} v_{i,j,k} \psi_h(\mathbf{x}_{i,j,k}, \mathbf{X}_{l,m}) \Delta x \Delta y \Delta z, \quad (42)$$

$$W_{l,m} = \sum_{\mathbf{x}_{i,j,k} \in \Omega} w_{i,j,k} \psi_h(\mathbf{x}_{i,j,k}, \mathbf{X}_{l,m}) \Delta x \Delta y \Delta z. \quad (43)$$

Conversely, the discrete spreading of Lagrangian force density  $\mathbf{F}$  onto cell centers using one-sided IB kernel (i.e.,  $\mathbf{f} = \mathcal{S}_{\text{MLS}}[\mathbf{X}]\mathbf{F}$ ) reads

$$(f_1)_{i,j,k} = \sum_{\mathbf{X}_{l,m} \in \Omega_b} (F_1)_{l,m} \psi_h(\mathbf{x}_{i,j,k}, \mathbf{X}_{l,m}) \Delta s_1 \Delta s_2, \quad (44)$$

$$(f_2)_{i,j,k} = \sum_{\mathbf{X}_{l,m} \in \Omega_b} (F_2)_{l,m} \psi_h(\mathbf{x}_{i,j,k}, \mathbf{X}_{l,m}) \Delta s_1 \Delta s_2, \quad (45)$$

$$(f_3)_{i,j,k} = \sum_{\mathbf{X}_{l,m} \in \Omega_b} (F_3)_{l,m} \psi_h(\mathbf{x}_{i,j,k}, \mathbf{X}_{l,m}) \Delta s_1 \Delta s_2. \quad (46)$$

Notice that the above formulas are same as those employed for regular IB interpolation and spreading operations, except that here we use MLS weighting functions.

### 3.3. Nullspace of the polynomial matrix

An interesting property of the MLS generating functions is that the odd-degree polynomial reproduction conditions are not affected by a constant shift, given a symmetric arrangement of the interpolation points  $\mathbf{x}$  around the evaluation point  $\mathbf{X}$ . The following Lemma makes it precise by considering two spatial dimensions and univariate polynomials.

**Lemma 2.** *With  $N$  denoting the number of Eulerian points in the support of the weight function in both  $\Omega_b^+$  and  $\Omega_b^-$  regions, and  $\mathbf{C} = c\mathbf{1}$  denoting a constant vector with entries  $c$ , the modified weights  $\Psi^m = (\Psi + \mathbf{C}) / \sum_{i=1}^N (\Psi + \mathbf{C}) = (\Psi + \mathbf{C}) / (1 + Nc)$  satisfy the first moment condition if the evaluation point is coincident with one of the interpolation points.*

**Proof.** Since  $\Psi$  satisfies the zeroth-order moment condition by construction,  $\sum \Psi = 1$  and since  $\mathbf{C}$  is a constant vector  $\sum \mathbf{C} = Nc$ . Therefore, the denominator of  $\Psi^m$  is  $\sum(\Psi + \mathbf{C}) = 1 + Nc$ . Next, considering Eq. (26), the modified weights satisfy the relation

$$\mathcal{A}\Psi^m = \mathbf{P}^m = (\mathbf{P} + \mathcal{A}\mathbf{C})/(1 + Nc). \quad (47)$$

In two spatial dimensions with univariate polynomials, the  $\mathcal{A}\mathbf{C}$  term of  $\mathbf{P}^m$  is

$$\mathcal{A}\mathbf{C} = c(\mathcal{A}\mathbf{1}) = \begin{bmatrix} Nc \\ c \sum_{i=1}^N (x_i - X) \\ c \sum_{i=1}^N (y_i - Y) \end{bmatrix} \quad (48)$$

If the evaluation point  $\mathbf{X} = (X, Y)$  is coincident with one of the interpolation points, odd-degree terms like  $\sum_{i=1}^N (x_i - X)$  or  $\sum_{i=1}^N (y_i - Y)$  evaluate to zero because of the symmetric arrangement. With  $\mathbf{P} = (1, 0, 0)^T$

$$\mathbf{P}^m = (\mathbf{P} + \mathcal{A}\mathbf{C})/(1 + Nc) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \mathbf{P}. \quad \square$$

Consequently,  $\Psi^m$  satisfies the same discrete moment conditions as  $\Psi$  for a symmetric arrangement of interpolation points centered around a moving point of interest  $\mathbf{X}$ . With a cell-centered discretization, this assumption holds for an IB point that is coincident with cell centroid. An analogous argument holds in three spatial dimensions with univariate polynomials. Notice that the zeroth-order moment condition of  $\Psi^m$  is satisfied due to the normalization factor  $(1 + Nc)$ , irrespective of the relative location of the evaluation point with respect to the background grid. We remark that higher-order moments, e.g., second-order moments with bivariate polynomials, won't be satisfied exactly even when  $\mathbf{X}$  is collocated on cell centroid for a constant shift. This is because even-degree entries of  $\mathbf{P}^m$ , such as  $\sum_{i=1}^N (x_i - X)^2$ , do not provide cancellations. However, odd-degree terms like  $\sum_{i=1}^N (x_i - X)(y_i - Y)$  would still equal zero.

### 3.4. Mollifying one-sided MLS kernels

In an essence, the moving least squares solution transforms the input weights  $W_{\text{MLS}}(\mathbf{x}_i, \mathbf{X})$  to  $\psi_h(\mathbf{x}_i, \mathbf{X})$  that satisfy the polynomial reproduction conditions. For IB spreading operation, it is desirable that generating functions  $\psi_h(\mathbf{x}_i, \mathbf{X})$  are monotonically decreasing and remain positive (or have negligible negative tails) for all evaluation points  $\mathbf{X}$ . In fact, the new/modified five- and six-point Peskin's delta functions were designed to eliminate the negative tails of the standard five- and six-point IB kernels, respectively.

The increased positivity of kernels also lead to better grid translation invariance property as shown empirically by Bao et al. [52]. In the context of one-sided IB kernels, the MLS procedure can generate larger (compared to the underlying weighting function) weights for nearby Eulerian grid nodes and negative weights for far-away grid nodes to satisfy the linear conditions. Fig. 3 in the next section (Sec. 3.5) describes this situation. We remark that large positive and negative weights in the kernel function do not pose stability concerns for certain IB methods that compute constraint forces exactly. It however, can induce flow instabilities in a direct forcing IB simulation through an oscillatory feedback. Indeed this was observed for test cases simulated using direct forcing IB method in this work. We also tested the one-sided MLS kernels with exactly-constrained IB methods [41,42]. Our preliminary tests suggest that the direct use of one-sided MLS kernels with exactly-constrained IB methods do not appear to produce flow instabilities; such specialized IB methods are not the focus of the current work and shall be explored more in the future.

We explore two *weight-shifting* strategies to mollify the generating functions  $\psi_h(\mathbf{x}_i, \mathbf{X})$ . The shifting procedure is motivated by the constant nullspace property defined in the previous section. We shall compare the two shifting strategies for flow past a sphere at  $Re = 10,000$  in Sec. 3.6 and make recommendations based on physical observations for this test case.

In Sec. 3.3, we examined shifting the generating functions  $\Psi$  by a constant vector. It was noted that the constant shift preserves the linear polynomial reproduction constraint under certain conditions, although doing so eliminates the one-sided property of  $\psi_h(\mathbf{x}_i, \mathbf{X})$ ;  $\Psi^m(\mathbf{x}_i, \mathbf{X}) = c \neq 0$  for  $\mathbf{x}_i \in \Omega_b^-$ . To remedy this situation we propose two modifications:

1. a constant vector shift (CVS) based on a restricted version of constant vector  $\mathbf{C}_{\Omega_b^+} = c\mathbf{1}_{\Omega_b^+}$ ;
2. a more general non-constant vector shift (NCVS) based on MLS weights  $\mathbf{C}_{\Omega_b^+} = c\mathcal{W}_{\text{MLS}}$ .

The form of the modified generating functions remain the same, which read as

$$\Psi_{\text{CVS}}^m = \frac{\Psi + c\mathbf{1}_{\Omega_b^+}}{\sum(\Psi + c\mathbf{1}_{\Omega_b^+})}, \quad (49)$$

$$\Psi_{\text{NCVS}}^m = \frac{\Psi + c\mathcal{W}_{\text{MLS}}}{\sum(\Psi + c\mathcal{W}_{\text{MLS}})} = \frac{\mathcal{W}_{\text{MLS}} \odot (\mathcal{L} + c\mathbf{1})}{\sum(\Psi + c\mathcal{W}_{\text{MLS}})}. \quad (50)$$

Notice that both shifts are zero in the  $\Omega_b^-$  region which preserves the one-sidedness of  $\Psi^m$ .

We now consider the effect of two restricted shifts on the polynomial reproduction constraints. Let  $N = N^+ \cup N^-$  denote the number of Eulerian points in the support of the weight function, with  $N^+ \in \Omega_b^+$  and  $N^- \in \Omega_b^-$ . Using the zeroth moment condition of original  $\Psi$ , and the more general  $\mathbf{C}_{\Omega_b^+} = c\mathcal{W}_{\text{MLS}}$ , we have

$$\Psi^m = \frac{\Psi + \mathbf{C}_{\Omega_b^+}}{1 + c \sum_{i \in N^+} W_i}, \quad (51)$$

in which  $W_i = W_{\text{MLS}}(\mathbf{x}_i, \mathbf{X})$ . Using Eq. (26) for the modified weights  $\Psi^m$ ,

$$\mathcal{A}\Psi^m = \mathbf{P}^m = \frac{\mathbf{P} + \mathcal{A}\mathbf{C}_{\Omega_b^+}}{1 + c \sum_{i \in N^+} W_i}. \quad (52)$$

Again considering two spatial dimensions and univariate polynomials, the  $\mathcal{A}\mathbf{C}_{\Omega_b^+}$  term of  $\mathbf{P}^m$  is

$$\mathcal{A}\mathbf{C}_{\Omega_b^+} = c \begin{bmatrix} \sum_{i \in N^+} W_i \\ \sum_{i \in N^+} (x_i - X) W_i \\ \sum_{i \in N^+} (y_i - Y) W_i \end{bmatrix}. \quad (53)$$

With  $\mathbf{P} = (1, 0, 0)^T$ ,

$$\mathbf{P}^m = \begin{bmatrix} 1 \\ \frac{c \sum_{i \in N^+} (x_i - X) W_i}{1 + c \sum_{i \in N^+} W_i} \neq 0 \\ \frac{c \sum_{i \in N^+} (y_i - Y) W_i}{1 + c \sum_{i \in N^+} W_i} \neq 0 \end{bmatrix}. \quad (54)$$

Similarly, for  $\mathbf{C}_{\Omega_b^+} = c\mathbf{1}_{\Omega_b^+}$ ,  $\mathbf{P}^m$  can be obtained by simply setting  $W_i = 1$  in Eq. (54). We make several remarks about these two weight shifting strategies:

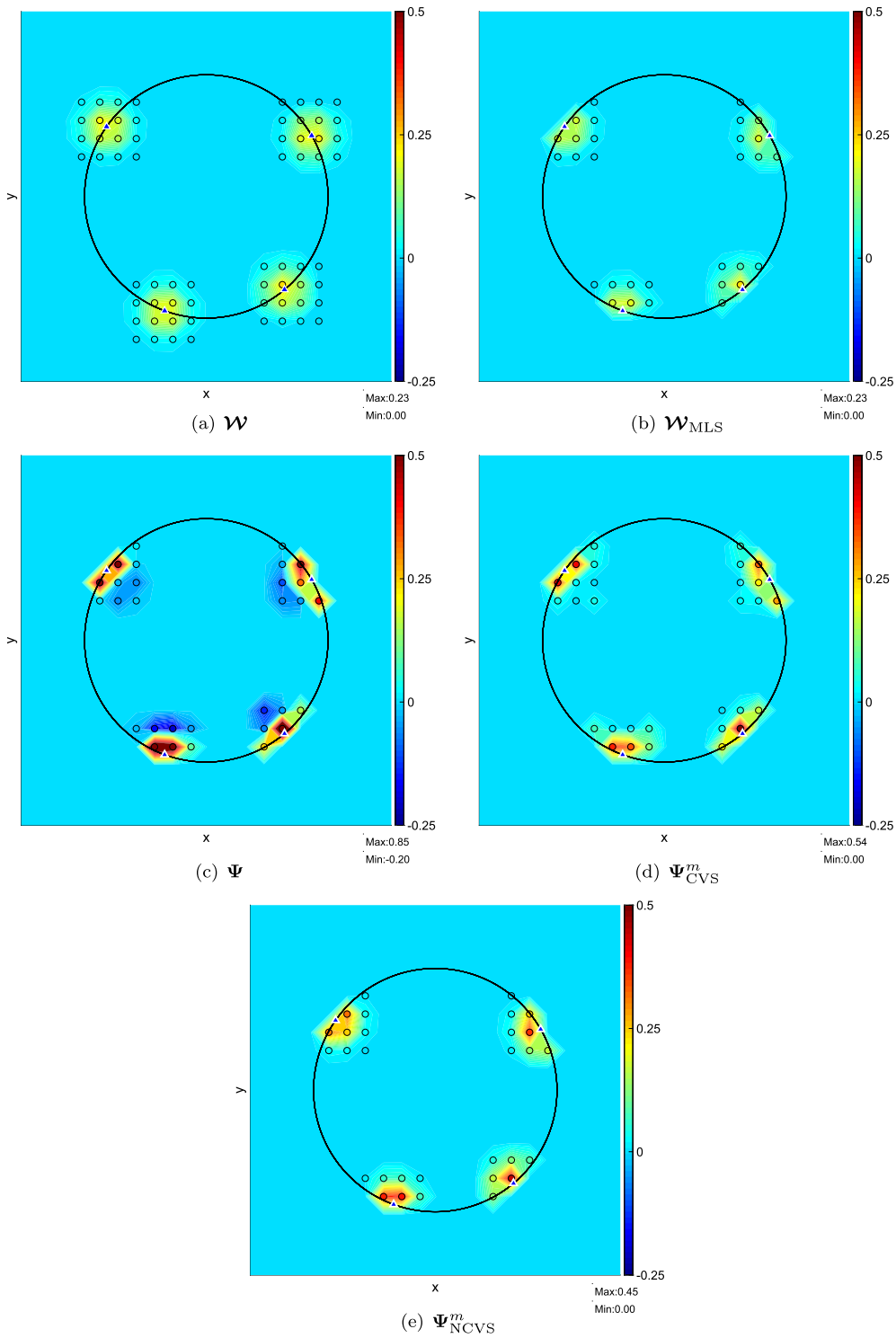
- For the Heaviside function,  $W_i = 0$  for  $\mathbf{x}_i \in \Omega_b^-$ , so only data in  $\Omega_b^+$  contribute to the summation. Hence, odd-degree polynomial terms may not cancel fully. Therefore, both strategies do not satisfy the linear polynomial reproducing condition in general. In practice, the IB marker points are not collocated on cell centroids, especially for moving bodies. Therefore, the modified weights do not satisfy the first moment condition even for the unrestricted constant shift  $\mathbf{C} = c\mathbf{1}$  for practical FSI applications.
- For the NCVS strategy, the shifts are inversely proportional to the distance between the evaluation and interpolation points. The CVS strategy is a special case of the NCVS strategy, with an equal shift applied to each interpolation point as  $W_i = 1$ . We will see the physical manifestation of these two shifting strategies in the context of flow past sphere in Sec. 3.6.
- The scalar  $c$  is a free-parameter, which can be chosen to enhance the positivity of  $\Psi$ . In this work we choose  $c = |\min(0, \min(\Psi))|$  and  $c = |\min(0, \min(\mathcal{L}))|$  for the CVS and NCVS strategy, respectively. From Eqs. (49) and (50), it can be easily verified that these are the minimum values of  $c$  that ensure  $\Psi^m \geq 0$ .
- The  $\Psi_{\text{NCVS}}^m$  weights in Eq. (50) can be viewed as a specific version of a more general combination of moment-satisfying one-sided MLS weights  $\Psi$  and a restricted version of two-sided weights  $\mathcal{W}_{\text{MLS}}$  (that may originally also satisfy the discrete moments),

$$\Psi_{\text{NCVS}}^m = \frac{\alpha\Psi + \beta\mathcal{W}_{\text{MLS}}}{\sum(\alpha\Psi + \beta\mathcal{W}_{\text{MLS}})}. \quad (55)$$

Here  $\alpha$  and  $\beta$  can be selected to satisfy the desired properties for the weights  $\Psi^m$ . The specific values of  $\alpha = 1$  and  $\beta = |\min(0, \min(\mathcal{L}))|$  considered here ensure that the weights are positive and revert to the MLS weights  $\Psi$  to achieve reproducing conditions, whenever possible.

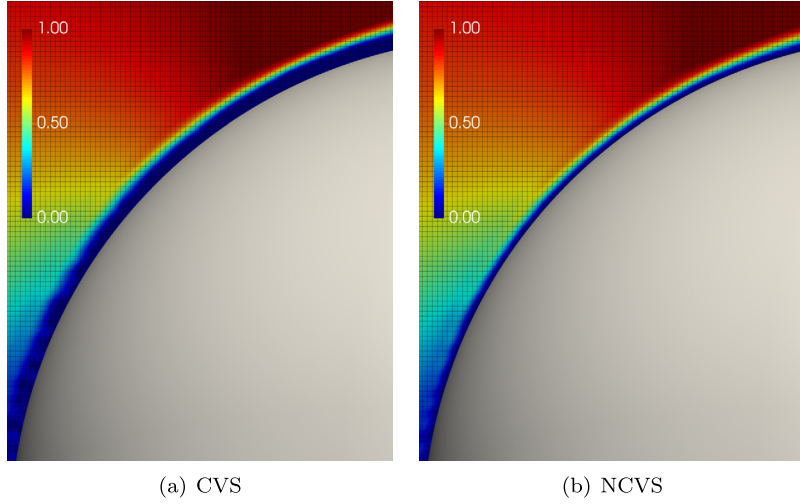
### 3.5. MLS procedure

In this section we take a simple case of a two-dimensional closed circular interface embedded in a uniform Cartesian grid to demonstrate the procedure of generating original and modified MLS weights,  $\Psi$  and  $\Psi^m$ , respectively. This example also serves to demonstrate that the MLS method can lead to over- and undershoots in  $\Psi$  for the restricted domain problem. It also serves to illustrate the process of remedying  $\Psi$  through CVS and NCVS shifting. For generality, the procedure is demonstrated for four Lagrangian points placed non-symmetrically along the interface:



**Fig. 3.** Color plot of (a) four-point delta function weights; (b) restricted four-point delta function weights; (c) MLS generated weights; (d) CVS shifted MLS weights; and (e) NCVS shifted MLS weights. Here, Eulerian grid points outside of the cylinder are masked. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

1. The first step involves generating the  $\mathcal{W}$  weights at the Eulerian grid points that are in the support of the Lagrangian material points. Fig. 3(a) shows the color plot of Peskin's four-point delta function weights. The Lagrangian points are highlighted with a filled triangle and the neighboring Eulerian grid points in the compact support are highlighted through black circles.



**Fig. 4.** Flow past a sphere at  $Re = 10,000$  using (a)  $\Psi_{CVS}^m$  and (b)  $\Psi_{NCVS}^m$  weights. The former approach leads to a thick numerical boundary layer, which is unphysical.

2. Next, the restricted weights  $\mathcal{W}_{MLS}$  are evaluated using a masking criterion. In this demonstration we mask the Eulerian grid points outside of the cylinder. The restricted weights are shown in Fig. 3(b).
3. The restricted weights  $\mathcal{W}_{MLS}$  are then used to produce the generating weights  $\Psi$ , which are plotted in Fig. 3(c). As mentioned in Sec. 3.4, and also for this example, the MLS construction produces large positive weights near the immediate vicinity of the interface and negative values near the tail-end region of  $\Psi$ ; compare the color scale of Figs. 3(a) and 3(c).
4. To mollify  $\Psi$ , the modified MLS weights  $\Psi^m$  are evaluated using CVS and NCVS approaches described in the prior Sec. 3.4. The modified weights  $\Psi_{CVS}^m$  and  $\Psi_{NCVS}^m$  are plotted in Fig. 3(d) and Fig. 3(e), respectively. It can be seen that the modified weights  $\Psi^m$  are positive towards the tail-end region and have mollified near the immediate vicinity of the interface.

### 3.6. CVS and NCVS comparison

While both approaches mollify  $\Psi$ , there is an important difference in their weight distribution.  $\Psi_{NCVS}^m$  has monotonically decreasing weights away from the evaluation point, which is also consistent with the distribution of the original weight  $\mathcal{W}$ . In contrast,  $\Psi_{CVS}^m$  has a weight distribution that does not decrease monotonically with distance. For IB simulations, it is desirable that weights have an inverse relationship with distance. If this condition is not met, the interpolation weights can artificially increase the boundary layer thickness at the fluid-IB interface.

To elucidate this effect, we consider flow past a sphere at a Reynolds number of 10,000. At  $Re = 10,000$ , the boundary layer is expected to be very thin and an IB simulation is expected to reflect this behavior. Peskin's four-point regularized delta kernel is used for evaluating  $\mathcal{W}_{MLS}$ . Velocity magnitude of the flow near the stagnation region of the sphere is shown in Fig. 4 for the two mollifying schemes. Because of the non-monotonically decreasing  $\Psi_{CVS}^m$  weights, grid nodes lying towards the tail-end region possess relatively large weights (see Fig. 3(d)). The resulting magnitude of the IB force at these grid nodes is large (compared to standard IB kernels), and consequently it leads to a thick numerical boundary layer as illustrated in Fig. 4(a). In contrast, flow at the fluid-IB interface corresponding to NCVS scheme is typical of an external flow around a sphere using a regular diffuse-interface IB method. The boundary layer over the sphere using the standard four-point kernel is quite similar to Fig. 4(b), and the comparison is omitted for brevity. Although this comparison is qualitative, and more systematic investigation is needed for high  $Re$  flows than what is presented in this section, quantitative comparison of CVS and NCVS approaches at moderate  $Re$  flows is demonstrated for few cases in Sec. 5. The results of Sec. 5 also favor the NCVS approach. Therefore, NCVS shifting scheme is preferred over CVS scheme, and all the results in what follows will be based on  $\Psi_{NCVS}^m$ , unless stated otherwise. As discussed in the introduction, the unmodified  $\Psi$  weights produce unphysical flow oscillations (due to force spreading) at the fluid-IB interface which destabilizes the direct forcing IB simulation; flow instability data not presented for this case.

## 4. Software implementation

The immersed boundary methods presented in this work are implemented in the solver CUBE [28], which is a multi-physics flow solver for massively parallel simulations. It is built on a hierarchical meshing technique known as the Building

Cube Method (BCM) [53]. CUBE supports hybrid parallelism through a combination of Message Passing Interface (MPI) and shared-memory parallelism using OpenMP.

## 5. Results

In this section, we first benchmark the one-sided IB kernels on problems from the IB literature to check the accuracy of FSI solutions. We also highlight the ability of one-sided kernels to avoid spurious flows inside the solid domain that are typically produced by standard diffuse IB kernels. Next, the MLS kernels are used to simulate flow past the Ahmed body [54], which demonstrates the potential of the one-sided IB approach to treat complex engineering geometries.

### 5.1. Taylor-Green vortex flow

In this test we follow Li et al. [39] and simulate the problem of Taylor-Green vortex flow with an embedded interface upon which exact velocity boundary conditions are imposed. Li et al. tested the accuracy of two-sided IB kernels using the MLS technique of Vanella and Balaras with direct forcing IB method. Here, we test the accuracy of one-sided MLS kernels for this problem. The analytical solution to the Taylor-Green vortex flow is

$$u(x, y, t) = -\cos(\pi x) \sin(\pi y) e^{-2\pi^2 t/Re}, \quad (56)$$

$$v(x, y, t) = \sin(\pi x) \cos(\pi y) e^{-2\pi^2 t/Re}, \quad (57)$$

$$p(x, y, t) = -\frac{1}{4} (\cos(2\pi x) + \cos(2\pi y)) e^{-4\pi^2 t/Re}. \quad (58)$$

The numerical simulation of the decaying vortex is performed with  $\Omega = [-2, 2] \times [-2, 2]$ , and a CFL number of  $C = 0.05$  is chosen. The exact solution at  $t = 0$  acts as the initial condition. A circular cylinder is chosen to represent the IB surface where a velocity based on the exact solution is imposed. The cylinder is located with its center at the origin of the domain, and its radius is taken to be 1. The Reynolds number of the flow is taken to be  $Re = 100$ , and the fluid density and viscosity are taken as,  $\rho = 1$  and  $\mu = 10^{-2}$ , respectively.

Because the Taylor-Green vortex problem is smooth, formulations using Peskin's second-order accurate delta functions provide high accuracy. This test also serves as a reference to compare the maximum order of accuracy for the proposed MLS kernels. In particular, more general models will typically include stress discontinuities along the fluid-structure interface, and in such conditions, all of the diffuse-interface methods, including the direct forcing IB method, will be only first-order accurate. Fig. 5 compares the numerical simulation at  $t = 1$  with the corresponding exact solution using  $L^1$ ,  $L^2$ , and  $L^\infty$  error norms. The  $u$  velocity error norms are shown with solid lines, and the pressure error norms are shown with dashed lines. As observed in Fig. 5(a), formulations using Peskin's four-point delta function in both  $\mathcal{J}$  and  $\mathcal{S}$  yields second-order accuracy for the velocity and pressure error norms. Similar results were obtained by Li et al. [39], who employed two-sided MLS technique of Vanella and Balaras with the five-point cubic spline function weight.

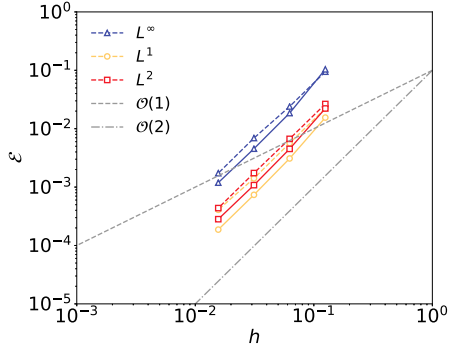
To evaluate the order of accuracy of the proposed MLS kernels, we consider two methods:

1. applying interior and exterior IB forcing to interior  $\Omega_b^-$  and exterior  $\Omega_b^+$  regions, respectively; and
2. applying exterior IB forcing to exterior  $\Omega_b^+$  region only.

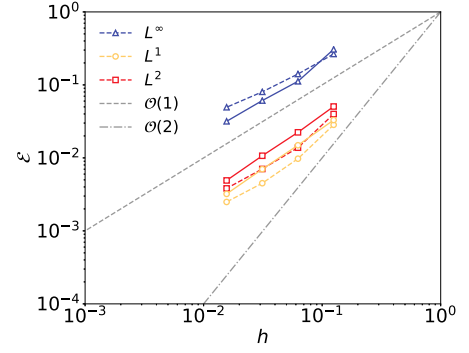
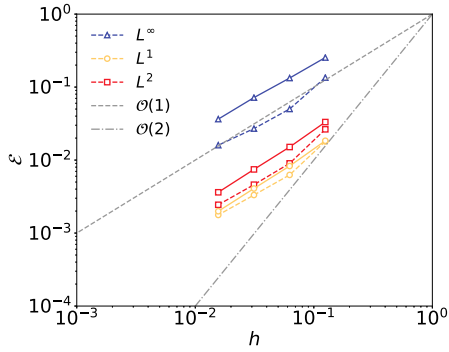
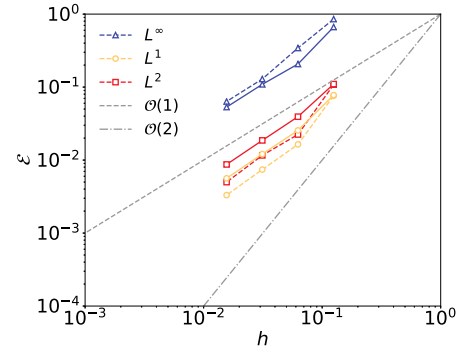
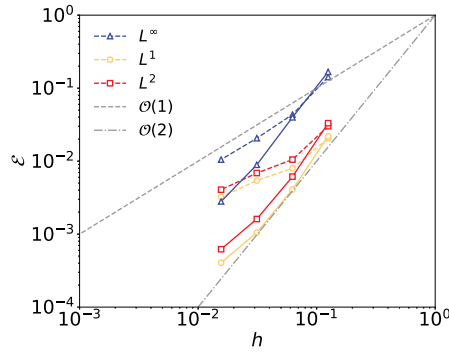
In these tests, Peskin's four-point delta function is chosen as the underlying delta-kernel for generating MLS weights. We first consider same weights  $\Psi_{\text{NCVS}}^m$  in the velocity interpolation and force spreading operators, so that  $\mathcal{S} = \mathcal{J}^*$ . In Figs. 5(b) and 5(c), the rate of convergence of  $L^1$ ,  $L^2$ , and  $L^\infty$  error norms with respect to the grid cell size for both methods of imposing IB forces are presented. The convergence rates are found to be approximately 1 for velocity and pressure for all error norms. This suggests that the two methods impose the same (velocity) boundary conditions on the two sides of the interface, and hence the order of accuracy of the solution is insensitive to the IB force location. The first-order convergence rates that occur when using modified weights  $\Psi_{\text{NCVS}}^m$  in  $\mathcal{S}$  and  $\mathcal{J}^*$  is also expected from the results of Sec. 3.4. For completeness, we also report in Fig. 5(d), the convergence rates obtained by using  $\Psi_{\text{CVS}}^m$  to construct  $\mathcal{J}$  and  $\mathcal{S}$ . IB forcing on both interior and exterior regions is considered (method 1). First-order accuracy is observed for this case as well.

Next, we compare the order of accuracy by employing non-shifted  $\Psi$  weights in  $\mathcal{J}$  and shifted  $\Psi_{\text{NCVS}}^m$  weights in  $\mathcal{S}$ . In this case  $\mathcal{S} \neq \mathcal{J}^*$ . IB forcing on both interior and exterior regions is considered (method 1). Fig. 5(e) shows super-linear accuracy in all velocity error norms and linear accuracy in the pressure error norm. This confirms that one-sided MLS weights satisfy the linear polynomial reproduction constraint, which improves the overall order of accuracy of the scheme. We note that the non-adjointness of  $\mathcal{S}$  and  $\mathcal{J}$  operators did not affect the stability of this scheme. However, using  $\Psi$  in  $\mathcal{S}$  resulted in flow instabilities for this case as well (data not shown).

Fig. 6 shows the vorticity plots for all of the MLS cases considered in this section. All cases show similar trends, except for the CVS strategy employed in Fig. 6(d), in which large spurious oscillations in the vorticity field are observed near the interface. This again highlights the fact that the proposed CVS approach to mollifying the MLS weights leads to unphysical flow fields. Mild oscillations are also observed in Figs. 6(b) and 6(c), which can be attributed to the first-order accuracy of  $\Psi_{\text{NCVS}}^m$  weights used in the interpolation operator  $\mathcal{J}$ .



(a) Interior and exterior IB forcing with Peskin's four-point delta function

(b) Interior and exterior IB forcing with  $\Psi_{\text{NCVS}}^m$  in  $\mathcal{T}$  and  $\mathcal{S}$ (c) Exterior IB forcing with  $\Psi_{\text{NCVS}}^m$  in  $\mathcal{T}$  and  $\mathcal{S}$ (d) Interior and exterior IB forcing with  $\Psi_{\text{CVS}}^m$  in  $\mathcal{T}$  and  $\mathcal{S}$ (e) Interior and exterior IB forcing with  $\Psi$  in  $\mathcal{T}$  and  $\Psi_{\text{NCVS}}^m$  in  $\mathcal{S}$ 

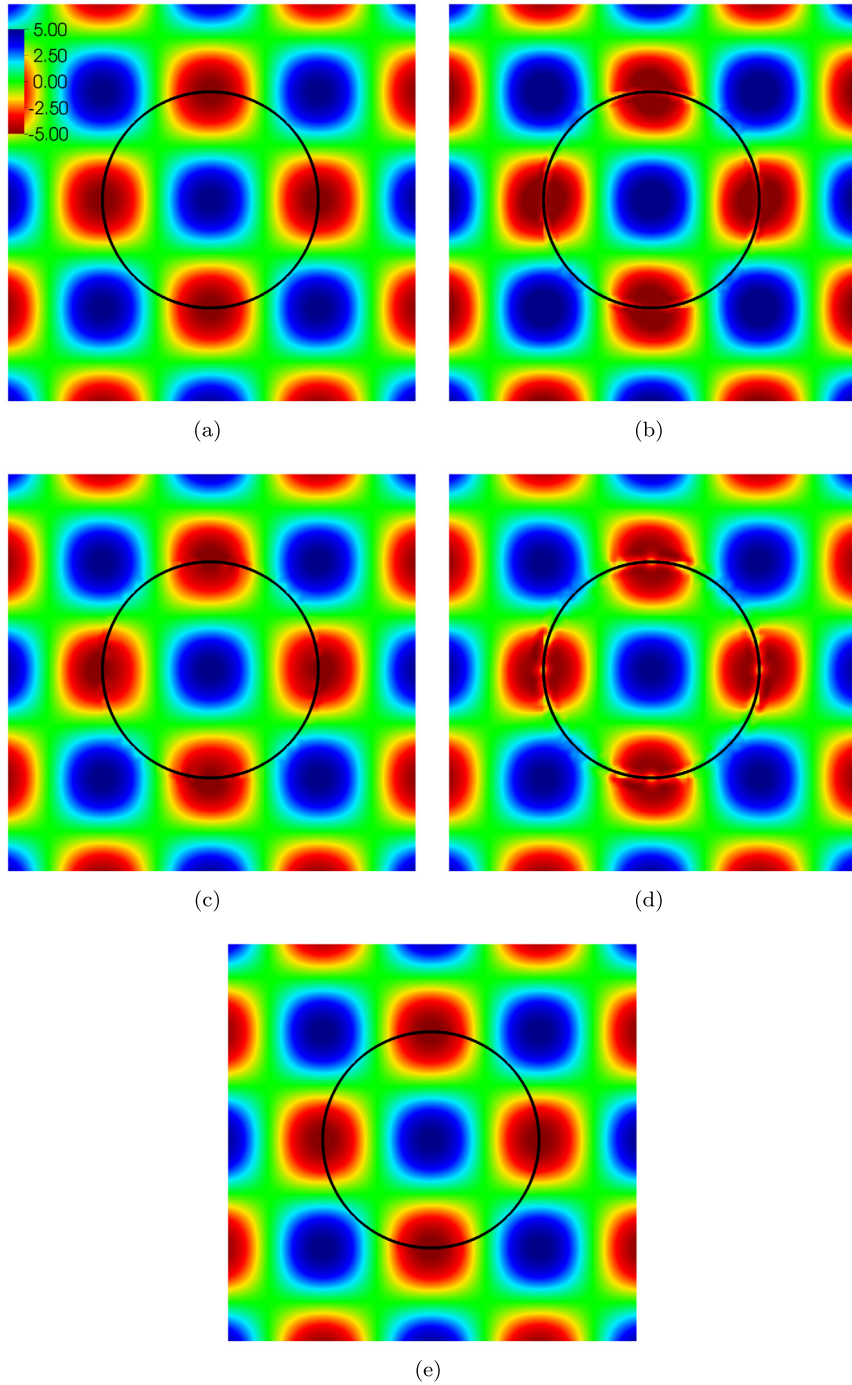
**Fig. 5.**  $L^1$ ,  $L^2$ , and  $L^\infty$  error norms of  $u$  velocity component (solid lines) and pressure (dashed lines) for the Taylor-Green vortex problem plotted against the grid cell size.

In what follows, we report results by employing  $\Psi_{\text{NCVS}}^m$  in both velocity interpolation and force spreading operators so that  $\mathcal{S} = \mathcal{T}^*$  in these cases. Although this coupling scheme reduces the order of accuracy as indicated by the results of this section, it will help bring-forth the worst-case scenario for one-sided IB/MLS simulations, if there is any.

## 5.2. Stokes' first problem

To validate that the current approach can capture time-dependent viscous boundary layers accurately, we consider the Stokes' first problem. The problem setup involves an infinitely long plate that is impulsively started with a velocity  $U_p$

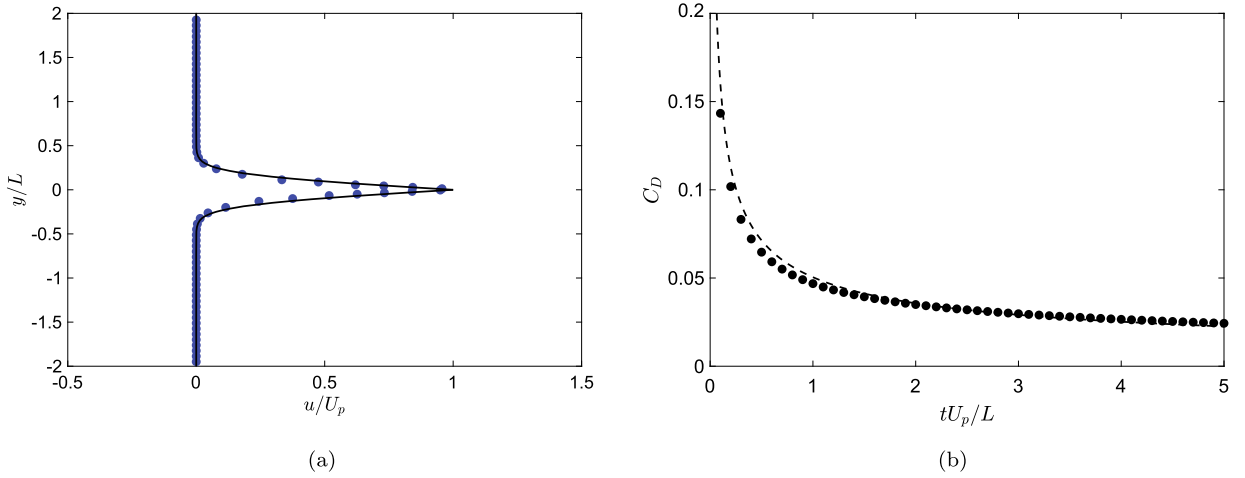




**Fig. 6.** Vorticity plots of the Taylor-Green vortex problem for (a) interior and exterior IB forcing with Peskin's four-point delta function; (b) interior and exterior IB forcing with  $\Psi_{\text{NCVS}}^m$  weights in  $\mathcal{J}$  and  $\mathcal{S}$ ; (c) exterior IB forcing with  $\Psi_{\text{NCVS}}^m$  weights in  $\mathcal{J}$  and  $\mathcal{S}$ ; (d) interior and exterior IB forcing with  $\Psi_{\text{CVS}}^m$  weights in  $\mathcal{J}$  and  $\mathcal{S}$ ; and (e) interior and exterior IB forcing with  $\Psi$  in  $\mathcal{J}$  and  $\Psi_{\text{NCVS}}^m$  in  $\mathcal{S}$ .

parallel to its axis oriented along the horizontal  $x$ -direction. Exact analytical solutions for the time evolution of the velocity profile and viscous drag coefficient are available for this problem. The transient velocity profile and drag coefficient are given by

$$u(y, t) = U_p \left[ \operatorname{erfc} \left( \frac{y}{2\sqrt{\nu t}} \right) \right], \quad (59)$$



**Fig. 7.** Comparison of analytical (solid line) and numerical (solid circle) solutions of the Stokes' first problem at  $Re = 500$ . (a) Vertical velocity profile  $u(y, t)$  at non-dimensional time  $T = tU_p/L = 5$ . (b) Time evolution of viscous drag coefficient.

$$C_D = \frac{2}{\sqrt{\pi t Re}}. \quad (60)$$

The Reynolds number is defined as  $Re = U_p L / \nu$ , in which  $L$  is the characteristic length and  $\nu = \mu / \rho$  is the kinematic viscosity of the fluid.

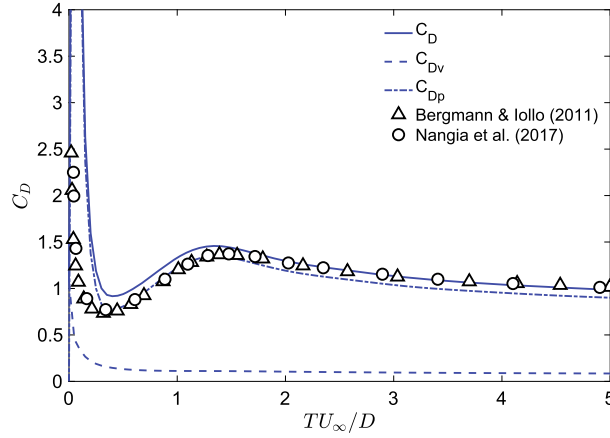
We carry out the simulation at  $Re = 500$  by setting  $U_p = 1$  and  $L = 1$ . The size of the computation domain is taken to be  $4L \times 4L$  with periodic boundary conditions along both  $x$  and  $y$  directions. A plate stretching across the domain along the  $x$ -axis is placed in the middle of the periodic box. The domain discretization is chosen such that the mesh spacing around the plate is  $\Delta x = \Delta y = 0.002L$ . Note that this problem setup corresponds to the open geometry configuration as shown in Fig. 1(b), and the top and bottom sides of the plate interact separately with the corresponding fluid domains. The velocity profile over the plate (on both sides) at non-dimensional time  $T = tU_p/L = 5$  is plotted in Fig. 7(a); the numerical solution agrees quite well with the analytical solution. Next, we compare the evolution of the drag coefficient  $C_D = F_x / \frac{1}{2} \rho U_p^2 A$  ( $F_x$  is the viscous force in the  $x$ -direction, and  $A$  is the area of the plate assuming a unit depth), with the analytical solution which is presented in Fig. 7(b). The agreement between the analytical and simulated drag coefficient profiles is also excellent. We also simulated this case using the standard Peskin kernel. The simulation results were identical to the ones shown in Fig. 7, and are not shown here for brevity.

### 5.3. Impulsive flow over a cylinder

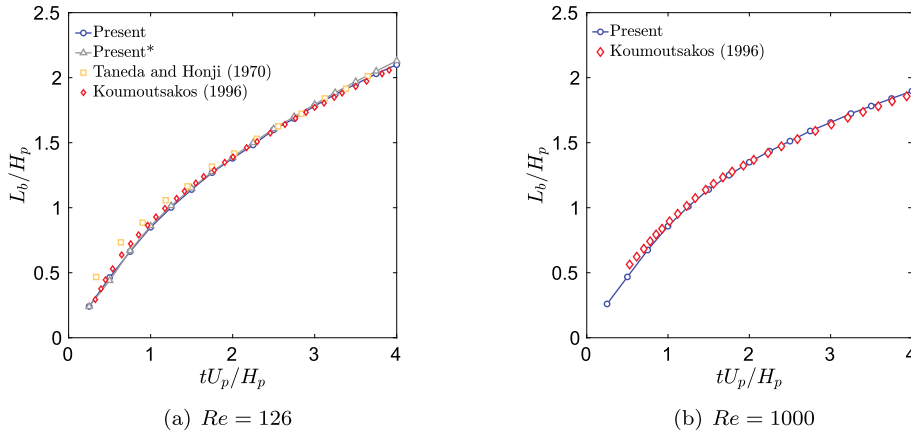
Next, we consider an impulsively started flow over a 2D circular cylinder of diameter  $D$ , which also serves to quantify the transient boundary layer development. The simulation is performed in the computation domain  $\Omega = [-4D, 12D] \times [-16D, 16D]$ . The diameter of the cylinder is  $D = 1$ , and it is placed with its center at  $(x, y) = (0, 0)$ . The Reynolds number is  $Re = DU_\infty / \nu$ , and is set to 500, the flow velocity in the horizontal  $x$ -direction is  $U_\infty = 1$ , and the kinematic viscosity of the fluid is  $\nu = 1/Re = 0.002$ . The mesh resolution on the cylinder surface is approximately  $0.01D$ . We validate the results of our simulation by comparing the evolution of the drag coefficient with the numerical results of Bergmann and Iollo [55] and Nangia et al. [50]. The drag coefficient along the flow direction is defined as  $C_D = F_x / \frac{1}{2} \rho DU_\infty^2$ . The comparison of  $C_D$  with data from the literature plotted in Fig. 8 shows a good agreement. In the same plot, we also present evolution of the viscous ( $C_{Dv}$ ) and pressure contribution ( $C_{Dp}$ ) to the total drag. These coefficients are defined analogous to  $C_D$  by using horizontal component of viscous and pressure forces, respectively. Note that both prior studies [55,50] compute the net hydrodynamic force on the body in an extrinsic manner, i.e., indirectly through momentum-conservation principle, whereas we compute it directly in an intrinsic manner by integrating the hydrodynamic stress tensor on the surface of the body. The difference in the initial transient drag profile in Fig. 8 is attributed to the manner in which hydrodynamic forces are evaluated in the current and the prior works.

### 5.4. Impulsively started plate

We consider the flow over an impulsively started, infinitesimally-thin, two-dimensional plate to validate the present numerical method for moving and finite-sized non-closed geometry. The flow resulting from the plate motion is characterized by two counter-rotating vortices in the wake which are fed by the shear layer at the tip of the two ends of the plate. The size of the wake vortices gradually increases before eventual saturation. Our numerical method is validated by capturing the dynamics of the time evolution of the wake vortices. We perform two-dimensional simulation at Reynolds numbers of



**Fig. 8.** Comparison of time variation of drag coefficient  $C_D$  of cylinder in an impulsively started flow with Bergmann and Iollo [55] and Nangia et al. [50]. Also plotted are the time evolution of skin friction and pressure coefficients,  $C_{Dv}$  and  $C_{Dp}$ , respectively.



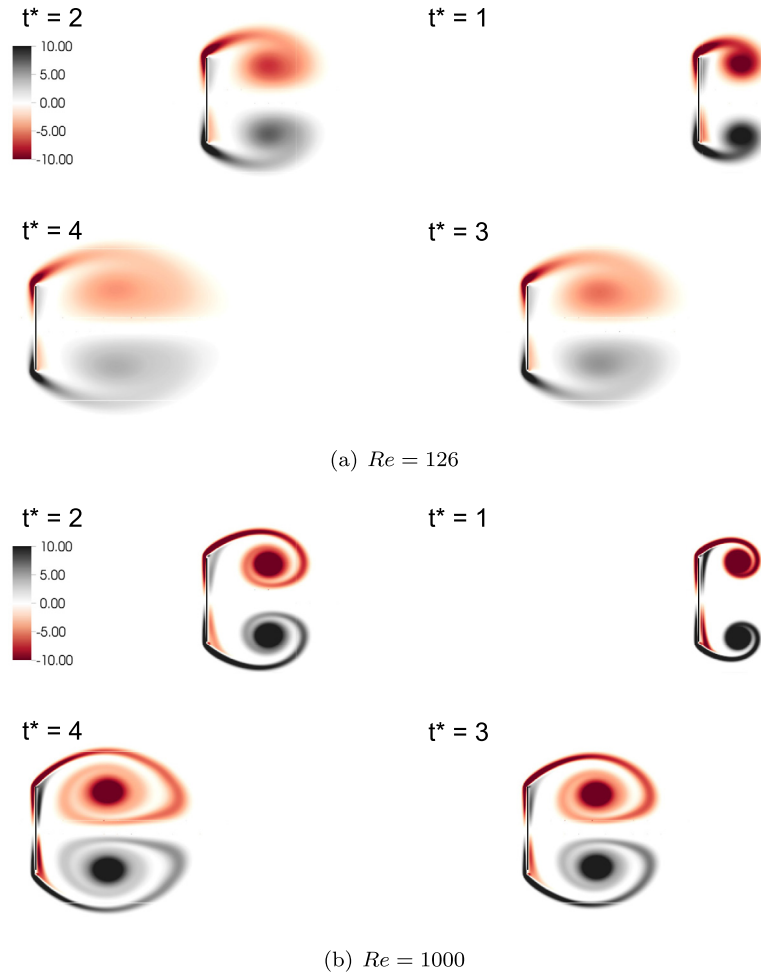
**Fig. 9.** The wake bubble size plotted against time in dimensionless form at (a)  $Re = 126$ , and (b)  $Re = 1000$ . Results of the present numerical method are compared against reported data in the literature [56,57]. In (a), the curve (—○—) corresponds to using  $\Psi_{NCVS}^m$  weights in both  $\mathcal{T}$  and  $\mathcal{S}$ , whereas the curve (—△—) corresponds to using  $\Psi$  weights in  $\mathcal{T}$  and  $\Psi_{NCVS}^m$  weights in  $\mathcal{S}$ .

126 and 1000, in which the Reynolds number is based on the height  $H_p$  and speed  $U_p$  of the plate. The  $Re$  is chosen to enable comparison with experimental and numerical results reported in the literature [56,57]. A computational domain of size  $10H_p \times 10H_p$  with no-slip boundary condition on the domain boundaries in the lateral direction and periodic boundary condition along the axial (corresponding to the direction of plate motion) is used. A uniform mesh resolution of  $0.01H_p$  is chosen for both  $Re$  cases. This resolution is consistent with the that of Mittal and coworkers [58]. At  $t = 0$  the plate is placed at the center of the computational domain and impulsively started with a velocity of  $U_p$  in the axial direction.

The height and velocity of the plate are used as characteristic length and velocity scale, respectively, to define the nondimensional wake bubble size  $L_b^* = L_b/H_p$  and the nondimensional time  $t^* = tU_p/H_p$ . We measure the evolution of the size of counter-rotating vortices in the wake bubble. The length of the wake bubble  $L_b$  is the normal distance from the plate to the end of the region of reverse flow behind the plate. Fig. 9 shows the temporal evolution of the nondimensional wake bubble size, and we compare our results against the reported data in the literature. We find that our results are in good agreement with the experimental results of Taneda and Honji [56] at  $Re = 126$ , and with numerical results of Koumoutsakos and Sheils [57] at  $Re = 126$  and  $Re = 1000$ . In Fig. 9(a), the results obtained by using non-shifted  $\Psi$  weights in  $\mathcal{T}$  and shifted  $\Psi_{NCVS}^m$  weights in  $\mathcal{S}$  are compared against those obtained by using shifted  $\Psi_{NCVS}^m$  weights in both  $\mathcal{S}$  and  $\mathcal{T}$  operators. The results are found to be virtually indistinguishable. The progression of vorticity development behind the plate is visualized in Fig. 10 for the two  $Re$  cases.

### 5.5. Flow around an oscillating cylinder

In order to further investigate the robustness of the current numerical method in handling moving geometries, we consider the flow around an oscillating cylinder, which involves a time-varying motion of the body. The problem setup



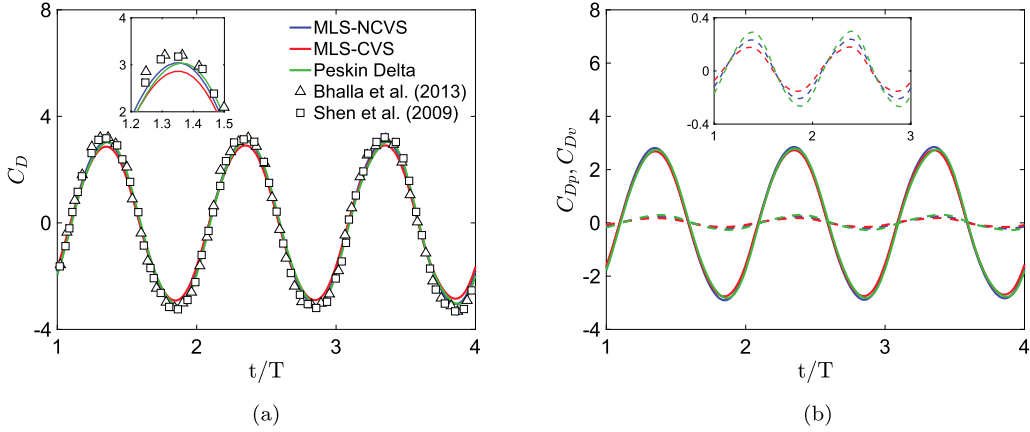
**Fig. 10.** Evolution of the separation bubble in the wake of an impulsively started plate at different instants of time for (a)  $Re = 126$ ; and (b)  $Re = 1000$ .

involves a circular cylinder oscillating about a mean position in a quiescent fluid. Assuming that the cylinder is oscillating along the  $x$ -axis, the velocity of oscillation of the cylinder is

$$U = U_m \cos\left(\frac{2\pi}{T}t\right), \quad (61)$$

in which  $U_m$  and  $T$  are the peak velocity and time period of oscillation, respectively. The flow generated by the oscillating cylinder is characterized by two dimensionless parameters, namely, the Keulegan-Carpenter number  $KC = U_m T/D$  and the Reynolds number  $Re = U_m D/\nu$ . Here,  $D$  is the diameter of the cylinder. The case has been well studied in the literature, for example [59,6]. We compare our computations against the results of Shen et al. [59] and Bhalla et al. [6] by setting  $D = 1$  and  $U_m = 1$ ,  $KC = 5$ , and  $Re = 100$ . The computational setup consists of a cylinder placed at the center of a domain at  $t = 0$ , and a square domain that spans a length of  $32D$  along both  $x$  and  $y$  directions. The mesh spacing on the cylinder surface and along its path of oscillation is chosen to be  $0.008D$ , which also matches the spacing employed in [6].

To elucidate the differences between the CVS and NCVS mollification strategies, the current validation case is performed using both approaches and compared to results obtained using the standard four-point Peskin kernel. The drag coefficient,  $C_D = F_x / \frac{1}{2} \rho U_m^2 D$ , computed from the simulation is presented in Fig. 11(a), and compared against the literature. There is good agreement between the present results and those reported previously [59,6]. However, closer inspection reveals that near the peaks of the  $C_D$  curve, the two-sided Peskin kernel and the one-sided NCVS kernel show better agreement with the literature data. Differences in the approaches are further investigated by comparing the total skin friction coefficient  $C_{Dv}$  and pressure coefficient  $C_{Dp}$ , against those obtained using the two-sided delta function; see Fig. 11(b). Whereas the differences in the pressure coefficient  $C_{Dp}$  are negligible, we find that the CVS approach underestimates the skin friction coefficient to a greater extent than the NCVS approach; see the inset of Fig. 11(b). We attribute this difference to the nature of the magnitude and (lesser) decay of the CVS weights away from the evaluation point (cf. Fig. 3), which leads to smaller velocity gradients.



**Fig. 11.** Flow around an oscillating cylinder at  $Re = 100$  and  $KC = 5$ . (a) Comparison of the drag coefficient  $C_D$  as a function of time with the literature. (b) Comparison of pressure  $C_{DP}$  (solid lines) and skin friction  $C_{DV}$  (dashed lines) coefficients using one-sided CVS and NCVS mollification strategies with those obtained using the two-sided four-point Peskin kernel. The color coding of the curves in (a) is same as (b).

**Table 1**

Comparison of steady state wake characteristics for flow over a sphere at  $Re = 100$  with data from the literature.

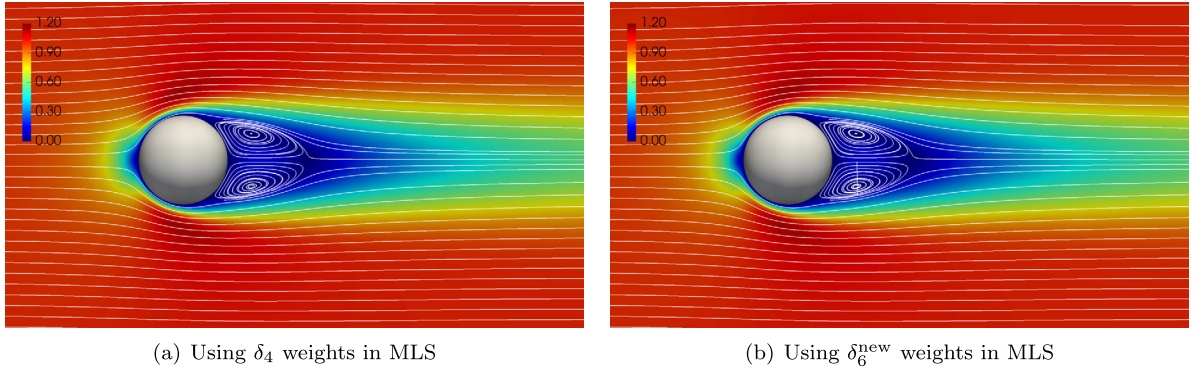
	$y_c/D$	$x_c/D$	$L_b/D$
$\delta_4$	0.29	0.766	0.89
$\varphi_5$	0.29	0.76	0.9
$\varphi_6$	0.29	0.764	0.9
$\delta_5^{\text{new}}$	0.294	0.764	0.9
$\delta_6^{\text{new}}$	0.29	0.762	0.905
Mittal et al. [58]	0.278	0.742	0.84
Johnson and Patel [60]	0.29	0.75	0.88
Taneda [63]	0.28	0.74	0.8
Tomboulides and Orszag [61]	-	-	0.88

## 5.6. Flow past a sphere

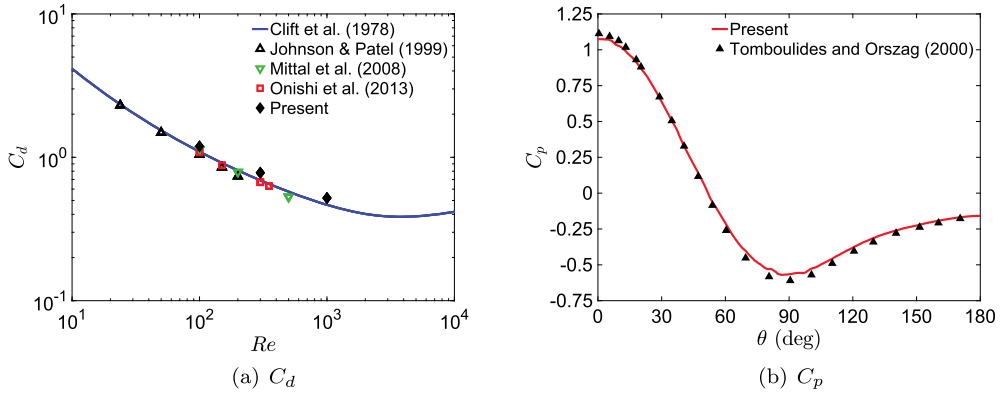
For validating three-dimensional flows with one-sided kernels, we consider the canonical case of flow past a sphere. The availability of detailed local, global and topological flow characteristics via several experimental and direct numerical simulation based studies makes flow over a stationary sphere an excellent candidate for validation [60,58,61,62]. The flow around a sphere exhibits steady axisymmetric, steady non-axisymmetric and unsteady non-axisymmetric behavior depending on the Reynolds number of the flow. For the present study, we choose Reynolds numbers of 100, 300, and 1000, which cover both steady and unsteady regimes. The computational domain is  $\Omega = [-40D, -40D, -40D] \times [120D, 40D, 40D]$  for  $Re = 100$  and  $Re = 300$  cases, and  $\Omega = [-25D, -60D, -60D] \times [100D, 65D, 65D]$  for the  $Re = 1000$  case. In all of the cases, a sphere of diameter  $D$  is placed with its center at  $(x, y, z) = (0, 0, 0)$ . Local mesh refinement is used to enhance the mesh resolution around the sphere. The near-surface mesh resolution for  $Re = 100$  and  $Re = 300$  is  $0.01D$ , and for  $Re = 1000$  is  $0.008D$ . A uniform inflow boundary condition and a homogeneous Neumann boundary condition are imposed at the inflow and outflow boundaries, respectively, and free-slip boundary conditions are imposed on the vertical and lateral walls of the computational domain.

As discussed in Sec. 3, the number of interpolation points in an MLS procedure depends upon the choice of the weighting function. Kernels with very narrow width can result in a near-singular Gram matrix when used for one-sided interpolation. Using flow around a sphere at  $Re = 100$ , we investigate the stability and accuracy of various kernels. Through this study, we can understand the smallest stable and accurate weighting kernel suitable for the MLS methodology.

The steady axisymmetric flow at  $Re = 100$  is characterized by a recirculation bubble in the immediate wake of the sphere. We identify the key geometric characteristics of the recirculation bubble, namely its center and length and compare them against the reported data in the literature. The center of the recirculation  $(x_c, y_c)$  is measured with respect to the center of the sphere, and the bubble length  $L_b$  is measured from the leeward end of the sphere. In Table 1 we report the center and length of the wake bubble nondimensionalized by  $D$  for all of the kernels, except the smoothed three-point IB kernel. Our tests found that the masking procedure using the smoothed three-point IB kernel results in a near-singular Gram matrix, and leads to numerical divergence. This test suggests that kernels whose width is three grid point or less are not suitable for the one-sided IB/MLS method presented in this work. The wider IB kernels are stable and pose no stability issues.



**Fig. 12.** Steady state streamlines and velocity magnitude for flow past a sphere at  $Re = 100$ .



**Fig. 13.** (a) Time-averaged drag coefficient plotted against Reynolds number and compared with literature [62,58,60,64]. (b) Comparison of steady state surface pressure coefficient at  $Re = 100$ , plotted against polar angle and compared with the reported results of Tomboulides and Orszag [61].

The comparison of geometric wake characteristics in Table 1 shows that all of the kernels reported yield virtually identical results. Based on this, it can be concluded that the IB kernel with the smallest width that is stable and accurate is the four-point IB kernel. In what follows, the four-point IB kernel will be used unless stated otherwise. With regards to the wake characteristics, the results presented in Table 1 show that our simulations agree well with the reported data in the literature. The streamlines and velocity magnitude of the flow in the vicinity of the sphere are shown in Fig. 12. Results obtained from  $\delta_4$  and  $\delta_6^{\text{new}}$  IB kernels are essentially indistinguishable.

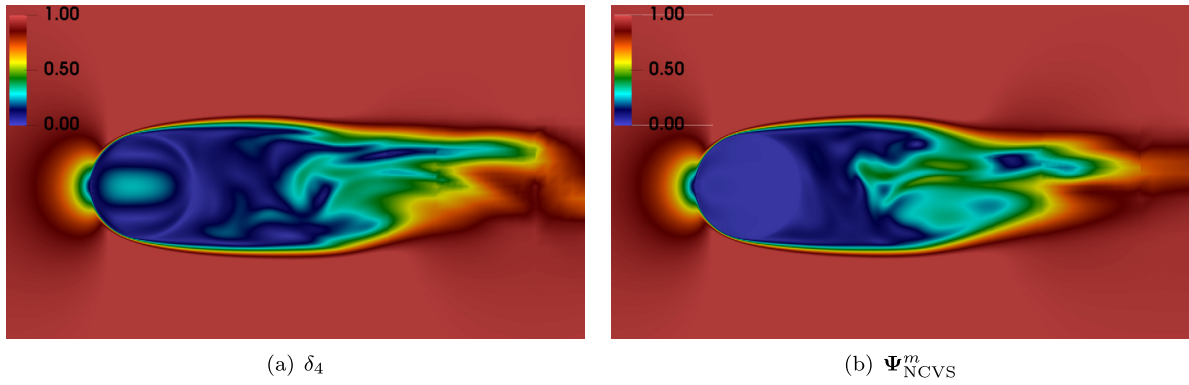
The time-averaged drag coefficients for the three  $Re$  cases considered are plotted in Fig. 13(a) and compared with experimental and numerical data from literature. A good agreement is obtained. The surface pressure coefficient for the  $Re = 100$  case, as shown in Fig. 13(b), agrees very well with that reported in Tomboulides and Orszag [61], who used a body-fitted finite volume solver in their study. The dimensionless dominant frequency of the unsteady wake, i.e., the Strouhal number of the flow for  $Re = 300$  and  $Re = 1000$  cases are compared with the literature in Table 2. For both cases the agreement is good. We conclude this section with a comparison of velocity magnitude of the flow inside the sphere geometry at  $Re = 1000$  produced by using one-sided and regular IB kernels. As shown in Fig. 14 the two-sided four-point IB kernel leads to sloshing flow inside the sphere geometry, which can interact and influence the external flow through the discrete diffusion and advection operators. In contrast, the internal flow using the current one-sided IB kernel is insignificant. Hence, it results in a better separation of internal and external flow domains.

### 5.7. Ahmed body

The study of the aerodynamics of road vehicles is challenging both numerically and experimentally due to the complexity of road vehicle geometries and the resulting high-speed flows. To enable characterization of the complex three-dimensional flow around road vehicles a simplified geometry, known as the Ahmed vehicle model or the Ahmed body has been used [54]. The Ahmed vehicle model retains the main characteristic of the flow features around real vehicles such as the three-dimensional regions of separation and unsteady coherent vortex structures. Owing to its ability to mimic real vehicle flows, the Ahmed vehicle model has been the de-facto standard to study the road vehicle aerodynamics.

The applicability of the present method to moderately high  $Re$  flows and complex geometries is demonstrated by using the flow around an Ahmed body as the final validation case. The Ahmed vehicle model considered for present case has a



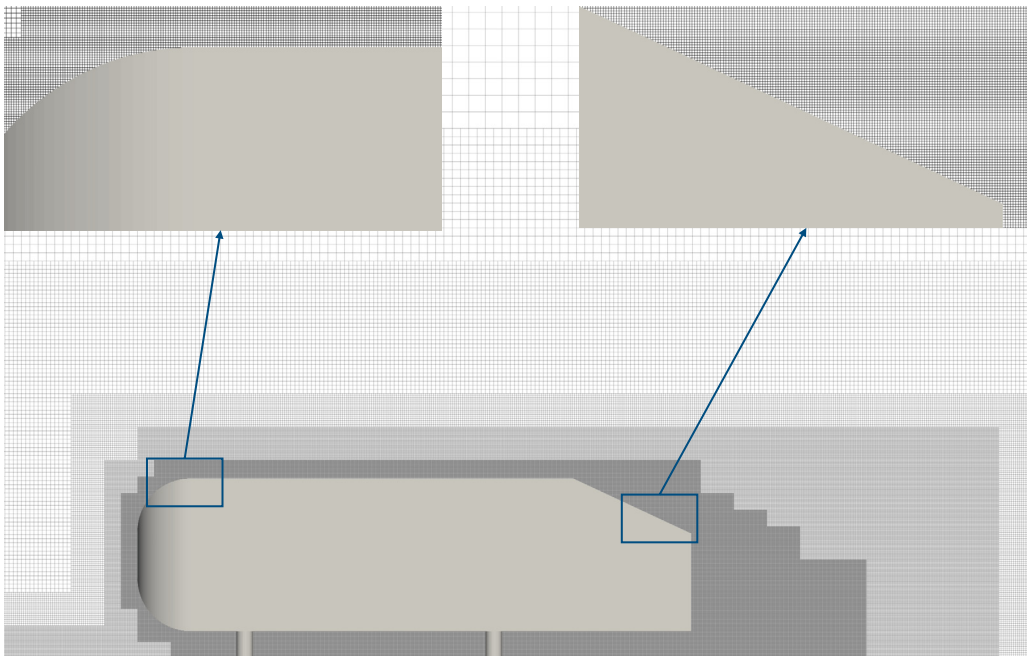


**Fig. 14.** Comparison of internal flow (shown through velocity magnitude colorplot) inside the sphere at  $Re = 1000$  using: (a) two-sided  $\delta_4$  IB kernel; and (b) one-sided  $\Psi^m_{NCVS}$  MLS kernel for the direct forcing IB method at  $t^* = tU_\infty/D = 50$ , in which  $U_\infty$  is the uniform inlet velocity.

**Table 2**

Strouhal number of flow over sphere at  $Re = 300$  and  $Re = 1000$ .

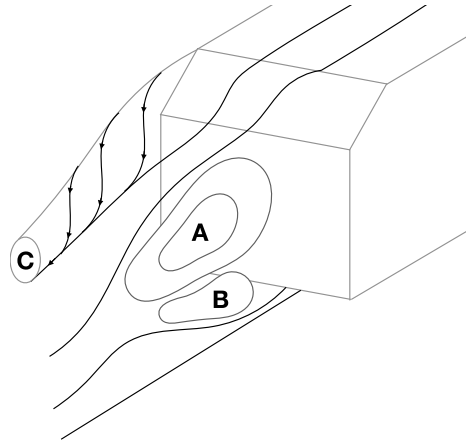
	$Re = 300$	$Re = 1000$
Present	0.127	0.195
Tomboulides and Orszag [61]	0.136	0.195
Mittal et al. [58]	0.137	–
Johnson and Patel [60]	0.135	–



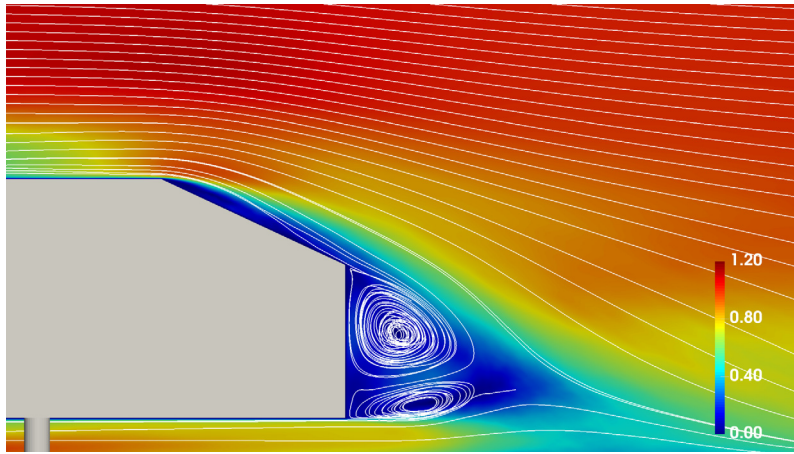
**Fig. 15.** Ahmed vehicle model and the locally refined mesh used in the numerical simulation. The insets show a magnified view of the mesh near the geometry surface.

rear slant surface that makes an angle of  $25^\circ$  with the top horizontal surface. Our simulations are validated against the experimental data of Moghimi and Rafee [66]. Moghimi and Rafee report drag and lift coefficient at  $Re_L = 9.31 \times 10^4$ , in which the Reynolds number is based on the length  $L$  of the Ahmed body. The computational domain and the boundary conditions are set to match the experimental setup of [66]. A uniform inflow boundary condition and a homogeneous Neumann boundary condition are imposed at the inflow and outflow boundaries, respectively, and a no-slip condition is imposed on the boundaries along the lateral and vertical direction. To ensure that the boundary layer encountered over the Ahmed body is resolved by our computational mesh, a near geometry mesh resolution of  $0.001L$  is employed. Local mesh refinement is used to restrict the fine mesh close to the geometry surface to reduce the computational cost of uniform

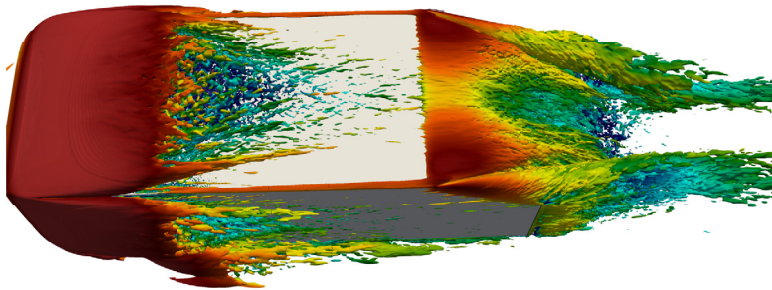




(a)

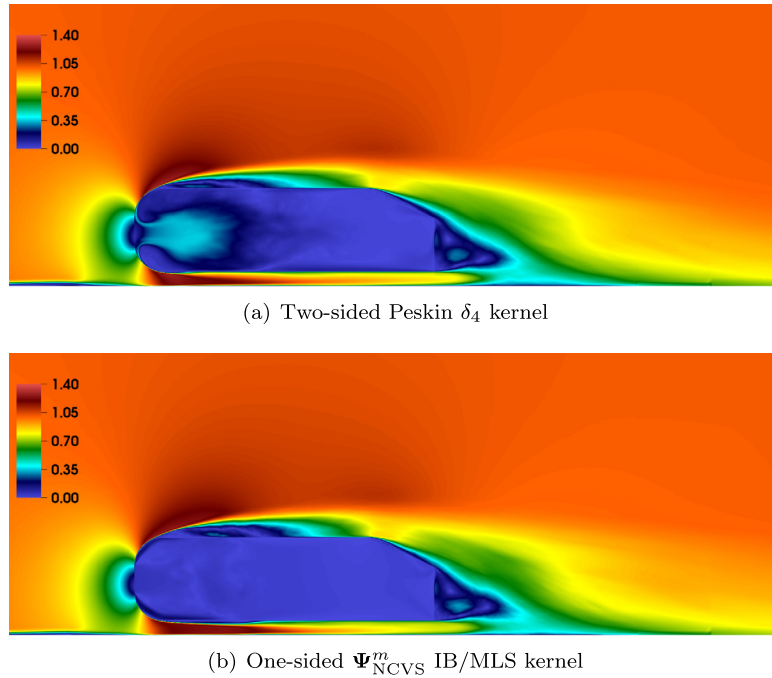


(b)



(c)

**Fig. 16.** Flow past an Ahmed body at  $Re_L = 9.31 \times 10^4$ . (a) Schematic of time-averaged wake structures behind an Ahmed body when the slant angle is between  $12.5^\circ$  and  $30^\circ$  [54,65]; (b) Visualization of the characteristic wake structures and the separation bubble over the slant surface of the Ahmed vehicle model through streamlines and velocity magnitude on a vertical plane; (c) Iso-surfaces of Q-criterion with dimensionless  $Q^* = QL^2/U^2 = 10$  of the time-averaged flow.



**Fig. 17.** Time-averaged velocity magnitude of flow over the Ahmed vehicle model using (a) the standard two-sided and (b) the present one-sided IB method.

**Table 3**  
Drag and lift coefficient of the Ahmed body.

	$C_d$	$C_L$
Present	0.4	0.29
Moghimi and Rafee [66]	$0.44 \pm 0.04$	$0.25 \pm 0.03$

grids. The locally refined mesh used for the present case is shown in Fig. 15. Considering the Blasius solution over a flat plate, the laminar boundary layer thickness is expected to scale as  $\delta_{lam} \sim 5LRe_L^{-\frac{1}{2}}$ , which gives a boundary layer thickness of  $\delta_{lam} \sim 0.016L$ . The chosen near wall mesh resolution results in at least 15 cells inside the boundary layer. The drag and lift coefficients from our results are compared with the wind tunnel data of [66] in Table 3. The results of our simulation are in good agreement with the reported data of Moghimi and Rafee within the error range of the experimental data.

Although the wake of the flow over a bluff-body is typically unsteady, the time-averaged flow over an Ahmed body shows persistent large-scale wake structures. Several experimental studies have characterized these flow structures based on the rear slant angle [54,65,67]. These studies have identified critical slant angles of  $12.5^\circ$  and  $30^\circ$  which appear to govern the topological structures of the time-averaged flow. When the slant angle is between these critical angles, three predominant three-dimensional vortical structures occur in the wake accompanied by a separation bubble on the slant surface. A schematic of these structures is shown in Fig. 16(a). Right behind the rear end of the Ahmed body, two counter-rotating vortices A and B, one larger than the other can be found. The third vortical structure C originates at the intersection of the slant surface and the side wall, which is stretched and elongated in the wake while it feeds the two primary vortices A and B. The characteristic wake signatures of the Ahmed body are identified in the results of our numerical simulation for qualitative validation. The near-wake time-averaged flow and streamlines on a vertical plane are presented in Fig. 16(b) in which the primary vortices A and B can be identified and the separation bubble over the rear slant surface can also be seen. The elongated helical flow structure C can be identified through the visualization of the time-averaged Q-criterion. In Fig. 16(c), iso-surfaces of Q-criterion with dimensionless  $Q^* = 10$  are plotted. The helical wake structure C can be seen on the two sides of the rear end of the Ahmed body. Fig. 17 shows the side view of the velocity magnitude over the entire Ahmed car model. It is clearly observed in the figure that the two-sided IB method leads to substantial flows within the car interior, which are largely eliminated using the current one-sided approach.

## 6. Summary and conclusions

This work develops a one-sided IB kernel approach for diffuse-interface IB methods. For IB models using thin structure or interface representations, the two sides of the flow region often require segregation and separate treatments. This is necessary to avoid spurious flows inside closed geometries and to avoid interaction of flow on the two sides of an IB sur-

face resulting from velocity interpolation and force spreading operations. To realize this, this paper introduces an approach that uses the moving least squares (MLS) method to dynamically generate one-sided IB kernels from standard two-sided IB kernels. We find that the stability of direct forcing IB methods benefit from kernels that are monotonically decreasing and positive (or have negligible negative tails) to avoid oscillatory feedback through force spreading. A straightforward application of the one-sided MLS construction can generate larger weights for nearby Eulerian grid nodes and negative weights for far-away grid nodes to satisfy the linear conditions. Two weight-shifting approaches, NCVS and CVS, are proposed to alleviate this issue. It was shown that the CVS approach can generate numerically thick boundary layers and spurious flow oscillations near the interface, whereas the NCVS approach produced physically correct flow features around the structure. Therefore, based on these results, the NCVS mollification approach is recommended.

The order of accuracy of one-sided IB kernels is tested through the Taylor-Green vortex flow problem. It was shown that by employing mollified weights in both velocity interpolation and force spreading operators the order of accuracy of the solution reduces to one. However, the accuracy of the scheme is improved (becomes super-linear) by adopting the original non-shifted MLS weights in the velocity interpolation operator. For more general problems, which typically include stress discontinuities along the fluid-structure interface, the present method will be only first-order accurate even with non-shifted MLS weights in the interpolation operator. This also holds true for regular diffuse-interface IB methods.

The dependency of accuracy and stability of an MLS kernel on the underlying weighting function is also analyzed. The one-sided MLS kernel was found to be stable for all weighting functions tested in this work except for the narrower three-point kernel. Our observation is that a basic regularized delta function kernel appears to require at least a support of four grid cells in each coordinate direction. A wider kernel avoids (near) singular Gram matrix in the one-sided moving least squares problem. When considering the problem of flow past a sphere at  $Re = 100$ , all weighting kernels were found to produce essentially equally accurate one-sided kernels. The proposed IB/MLS method was further validated through the cases of flow past an oscillating cylinder, Stokes' first problem, flow over a sphere, impulsively started plate/cylinder, and the Ahmed vehicle model. In all cases, our results are in excellent agreement with the reported data in the literature. We also demonstrate that the present method can effectively eliminate the spurious internal flow that is typically generated by simulating flow past bluff bodies with regular diffuse-interface IB methods.

### CRedit authorship contribution statement

Rahul Bale: Data generation, Investigation, Methodology, Resources, Software, Validation, Visualization, Data analysis, Writing – original draft preparation, review & editing, Funding acquisition.

Amneet Pal Singh Bhalla: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Data Analysis, Project administration, Supervision, Writing – original draft preparation, review & editing.

Boyce E. Griffith: Conceptualization, Methodology, Funding acquisition, Writing – editing & review.

Makoto Tsubokura: Supervision, Funding acquisition.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgements

A.P.S.B acknowledges support from NSF award OAC 1931368. R.B acknowledges the support for this work by JSPS KAKENHI Grant Number 20K19503. B.E.G acknowledges support from NSF awards OAC 1450327, OAC 1652541, OAC 1931516, and DMS 1664645.

### Appendix A. Kernel functions

Here we write the one-dimensional form of some of the kernels used in this work. In multiple dimensions, a tensor product of the one-dimensional kernels is formed. The functional form of the kernel is expressed in terms of  $r = (x - X)/h$ .

$$\text{Smoothed three-point IB kernel [30]: } \delta_3(r) = \begin{cases} \frac{3}{4} - r^2, & 0 \leq |r| < 0.5, \\ \frac{1}{2}(\frac{9}{4} - 3|r| + r^2), & 0.5 \leq |r| < 1.5, \\ 0, & 1.5 \leq |r|. \end{cases} \quad (\text{A.1})$$

$$\text{Peskin's four-point IB kernel [1]: } \delta_4(r) = \begin{cases} \frac{1}{8} \left( 3 - 2|r| + \sqrt{1 + 4|r| - 4r^2} \right), & 0 \leq |r| < 1, \\ \frac{1}{8} \left( 5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2} \right), & 1 \leq |r| < 2, \\ 0, & 2 \leq |r|. \end{cases} \quad (\text{A.2})$$

$$\text{Radial basis function: RBF}(r) = \begin{cases} e^{-2r^2}, & |r| < 2, \\ 0, & 2 \leq |r|. \end{cases} \quad (\text{A.3})$$

$$\text{Two-point cubic spline kernel [27]: } \varphi_2(\bar{r}) = \begin{cases} \frac{2}{3} - 4\bar{r}^2 + 4\bar{r}^3 & 0 \leq |\bar{r}| < 0.5, \\ \frac{4}{3} - 4\bar{r} + 4\bar{r}^2 - \frac{4}{3}\bar{r}^3 & 0.5 \leq |\bar{r}| < 1, \\ 0, & 1 \leq |\bar{r}|. \end{cases} \quad (\text{A.4})$$

$$\text{Five-point spline kernel: } \varphi_5(r) = \begin{cases} \frac{1}{24}(6\kappa^4 - 60\kappa^3 + 210\kappa^2 - 300\kappa + 155) & 0 \leq |r| < 0.5, \\ \frac{1}{24}(-4\kappa^4 + 60\kappa^3 - 330\kappa^2 + 780\kappa + 655) & 0.5 \leq |r| < 1.5, \\ \frac{1}{24}(\kappa^4 - 20\kappa^3 + 150\kappa^2 - 500\kappa + 625) & 1.5 \leq |r| < 2.5, \\ 0, & 2.5 \leq |r|, \end{cases} \quad (\text{A.5})$$

$$\text{Six-point spline kernel: } \varphi_6(r) = \begin{cases} \frac{1}{60}(-5\kappa^5 + 90\kappa^4 - 630\kappa^3 + 2130\kappa^2 - 3645\kappa + 2193) & 0 \leq |r| < 1, \\ \frac{1}{120}(5\kappa^5 - 120\kappa^4 + 1140\kappa^3 - 5340\kappa^2 + 12270\kappa - 10974) & 1 \leq |r| < 2, \\ \frac{1}{120}(-\kappa^5 + 30\kappa^4 - 360\kappa^3 + 2160\kappa^2 - 6480\kappa + 7776) & 2 \leq |r| < 3, \\ 0, & 3 \leq |r|. \end{cases} \quad (\text{A.6})$$

Here  $\bar{r} = r/1.2$  for the two-point (more specifically 2.4 grid cells wide) cubic spline function used in Vanella and Balaras [27], and  $\kappa = |r| + 2.5$  and  $\kappa = |r| + 3$  for the five-point and six-point spline function, respectively. In addition to the aforementioned kernels, we also consider the new five- and six-point IB kernel,  $\delta_5^{\text{new}}$  and  $\delta_6^{\text{new}}$ , respectively. These new kernels remove the negative tail of the standard five- and six-point IB kernels by imposing a weaker second moment condition. We refer readers to Bao et al. [52] for their functional form.

## References

- [1] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [2] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.
- [3] B.E. Griffith, N.A. Patankar, Immersed methods for fluid–structure interaction, *Annu. Rev. Fluid Mech.* 52 (2020) 421–448.
- [4] N.A. Patankar, P. Singh, D.D. Joseph, R. Glowinski, T.-W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, *Int. J. Multiph. Flow* 26 (9) (2000) 1509–1524.
- [5] I. Borazjani, L. Ge, T. Le, F. Sotiropoulos, A parallel overset-curved-immersed boundary framework for simulating complex 3D incompressible flows, *Comput. Fluids* 77 (2013) 76–96.
- [6] A.P.S. Bhalla, R. Bale, B.E. Griffith, N.A. Patankar, A unified mathematical framework and an adaptive numerical method for fluid–structure interaction with rigid, deforming, and elastic bodies, *J. Comput. Phys.* 250 (1) (2013) 446–476, <https://doi.org/10.1016/j.jcp.2013.04.033>, <http://linkinghub.elsevier.com/retrieve/pii/S0021999113003173>.
- [7] M. Gazzola, O.V. Vasilyev, P. Koumoutsakos, Shape optimization for drag reduction in linked bodies using evolution strategies, *Comput. Struct.* 89 (11–12) (2011) 1224–1231.
- [8] N. Nangia, N.A. Patankar, A.P.S. Bhalla, A DLM immersed boundary method based wave–structure interaction solver for high density ratio multiphase flows, *J. Comput. Phys.* 398 (2019) 108804.
- [9] C.-G. Li, M. Tsubokura, R. Bale, Framework for simulation of natural convection in practical applications, *Int. Commun. Heat Mass Transf.* 75 (2016) 52–58.
- [10] K. Khedkar, N. Nangia, R. Thirumalaisamy, A.P.S. Bhalla, The inertial sea wave energy converter (ISWEC) technology: Device-physics, multiphase modeling and simulations, *Ocean Eng.* 229 (2021) 108879.
- [11] A.P.S. Bhalla, N. Nangia, P. Dafnakis, G. Bracco, G. Mattiazzi, Simulating water-entry/exit problems using Eulerian–Lagrangian and fully-Eulerian fictitious domain methods within the open-source IBAMR library, *Appl. Ocean Res.* 94 (2020) 101932.
- [12] B.E. Griffith, X. Luo, D.M. McQueen, C.S. Peskin, Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method, *Int. J. Appl. Mech.* 1 (01) (2009) 137–177.
- [13] B.E. Griffith, X. Luo, Hybrid finite difference/finite element immersed boundary method, *Int. J. Numer. Methods Biomed. Eng.* 33 (12) (2017) e2888.
- [14] A.P.S. Bhalla, B.E. Griffith, N.A. Patankar, A forced damped oscillation framework for undulatory swimming provides new insights into how propulsion arises in active and passive swimming, *PLoS Comput. Biol.* 9 (6) (2013) e1003097.
- [15] H. Luo, R. Mittal, X. Zheng, S.A. Bielamowicz, R.J. Walsh, J.K. Hahn, An immersed-boundary method for flow–structure interaction in biological systems with application to phonation, *J. Comput. Phys.* 227 (22) (2008) 9303–9332.
- [16] N.K. Patel, A.P.S. Bhalla, N.A. Patankar, A new constraint-based formulation for hydrodynamically resolved computational neuromechanics of swimming animals, *J. Comput. Phys.* 375 (2018) 684–716.
- [17] B. Sprinkle, R. Bale, A.P.S. Bhalla, M.A. MacIver, N.A. Patankar, Hydrodynamic optimality of balistiform and gymnotiform locomotion, *Eur. J. Comput. Mech.* 26 (1–2) (2017) 31–43.
- [18] R. Thirumalaisamy, N. Nangia, A.P.S. Bhalla, Critique on “volume penalization for inhomogeneous Neumann boundary conditions modeling scalar flux in complicated geometry”, *J. Comput. Phys.* 433 (2021) 110163, <https://doi.org/10.1016/j.jcp.2021.110163>, <https://www.sciencedirect.com/science/article/pii/S0021999121000553>.
- [19] R. Thirumalaisamy, N.A. Patankar, A.P.S. Bhalla, Handling Neumann and Robin boundary conditions in a fictitious domain volume penalization framework, preprint, arXiv:2101.02806.
- [20] A. Helgadottir, Y.T. Ng, C. Min, F. Gibou, Imposing mixed Dirichlet–Neumann–Robin boundary conditions in a level-set framework, *Comput. Fluids* 121 (2015) 68–80.

- [21] M. Chai, K. Luo, H. Wang, S. Zheng, J. Fan, Imposing mixed Dirichlet-Neumann-Robin boundary conditions on irregular domains in a level set/ghost fluid based finite difference framework, *Comput. Fluids* 214 (2021) 104772.
- [22] M. Jiang, Z. Liu, A boundary thickening-based direct forcing immersed boundary method for fully resolved simulation of particle-laden flows, *J. Comput. Phys.* 390 (2019) 203–231.
- [23] R. Mittal, H. Dong, M. Bozkurtas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (10) (2008) 4825–4852.
- [24] H. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.* 174 (1) (2001) 345–380.
- [25] I. Borazjani, L. Ge, F. Sotiropoulos, Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies, *J. Comput. Phys.* 227 (16) (2008) 7587–7620.
- [26] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2) (2005) 448–476.
- [27] M. Vanella, E. Balaras, Short note: a moving-least-squares reconstruction for embedded-boundary formulations, *J. Comput. Phys.* 228 (18) (2009) 6617–6628.
- [28] N. Jansson, R. Bale, K. Onishi, M. Tsubokura, Cube: a scalable framework for large-scale industrial simulations, *Int. J. High Perform. Comput. Appl.* 33 (4) (2019) 678–698.
- [29] R. Bale, N.A. Patankar, N. Jansson, K. Onishi, M. Tsubokura, Stencil penalty approach based constraint immersed boundary method, *Comput. Fluids* 200 (2020) 104457.
- [30] X. Yang, X. Zhang, Z. Li, G.-W. He, A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations, *J. Comput. Phys.* 228 (20) (2009) 7821–7836.
- [31] D.B. Stein, R.D. Guy, B. Thomases, Immersed boundary smooth extension: a high-order method for solving pde on arbitrary smooth domains using Fourier spectral methods, *J. Comput. Phys.* 304 (2016) 252–274.
- [32] H. Wendland, Local polynomial reproduction and moving least squares approximation, *IMA J. Numer. Anal.* 21 (1) (2001) 285–300.
- [33] S. Krishnan, E.S. Shaqfeh, G. Iaccarino, Fully resolved viscoelastic particulate simulations using unstructured grids, *J. Comput. Phys.* 338 (2017) 313–338.
- [34] A. Saadat, C.J. Guido, G. Iaccarino, E.S. Shaqfeh, Immersed-finite-element method for deformable particle suspensions in viscous and viscoelastic media, *Phys. Rev. E* 98 (6) (2018) 063316.
- [35] W.-K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *Int. J. Numer. Methods Fluids* 20 (8–9) (1995) 1081–1106.
- [36] W.-K. Liu, S. Li, T. Belytschko, Moving least-square reproducing kernel methods (i) methodology and convergence, *Comput. Methods Appl. Mech. Eng.* 143 (1–2) (1997) 113–154.
- [37] D. Mirzaei, R. Schaback, M. Dehghan, On generalized moving least squares and diffuse derivatives, *IMA J. Numer. Anal.* 32 (3) (2012) 983–1000.
- [38] N. Sukumar, R. Wright, Overview and construction of meshfree basis functions: from moving least squares to entropy approximants, *Int. J. Numer. Methods Eng.* 70 (2) (2007) 181–205.
- [39] D. Li, A. Wei, K. Luo, J. Fan, An improved moving-least-squares reconstruction for immersed boundary method, *Int. J. Numer. Methods Eng.* 104 (8) (2015) 789–804.
- [40] M.D. de Tullio, G. Pascasio, A moving-least-squares immersed boundary method for simulating the fluid–structure interaction of elastic bodies with arbitrary thickness, *J. Comput. Phys.* 325 (2016) 201–225.
- [41] B. Kallemov, A.P.S. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, *Commun. Appl. Math. Comput. Sci.* 11 (1) (2016) 79–141.
- [42] F. Balboa Usabiaga, B. Kallemov, B. Delmotte, A.P.S. Bhalla, B. Griffith, A. Donev, Hydrodynamics of suspensions of passive and active rigid particles: a rigid multiblob approach, *Commun. Appl. Math. Comput. Sci.* 11 (2) (2017) 217–296.
- [43] D.-V. Le, B.-C. Khoo, A moving-least-square immersed boundary method for rigid and deformable boundaries in viscous flow, *Commun. Comput. Phys.* 22 (4) (2017) 913–934.
- [44] M. Haji Mohammadi, F. Sotiropoulos, J. Brinkerhoff, Moving least squares reconstruction for sharp interface immersed boundary methods, *Int. J. Numer. Methods Fluids* 90 (2) (2019) 57–80.
- [45] B.E. Griffith, X. Luo, D.M. McQueen, C.S. Peskin, Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method, *Int. J. Appl. Mech.* 1 (01) (2009) 137–177.
- [46] K. Yeo, M.R. Maxey, Dynamics of concentrated suspensions of non-colloidal particles in Couette flow, *J. Fluid Mech.* 649 (2010) 205.
- [47] S. Delong, F.B. Usabiaga, R. Delgado-Buscalioni, B.E. Griffith, A. Donev, Brownian dynamics without Green's functions, *J. Chem. Phys.* 140 (13) (2014) 134110.
- [48] S. Patankar, *Numerical Heat Transfer and Fluid Flow*, CRC Press, 2018.
- [49] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 168 (2) (2001) 464–499.
- [50] N. Nangia, H. Johansen, N.A. Patankar, A.P.S. Bhalla, A moving control volume approach to computing hydrodynamic forces and torques on immersed bodies, *J. Comput. Phys.* 347 (2017) 437–462.
- [51] G. Backus, F. Gilbert, The resolving power of gross earth data, *Geophys. J. Int.* 16 (2) (1968) 169–205.
- [52] Y. Bao, J. Kaye, C.S. Peskin, A gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance, *J. Comput. Phys.* 316 (2016) 139–144.
- [53] K. Nakahashi, Building-cube method for flow problems with broadband characteristic length, in: *Computational Fluid Dynamics 2002*, Springer, 2003, pp. 77–81.
- [54] S.R. Ahmed, G. Ramm, G. Falin, Some salient features of the time-averaged ground vehicle wake, *Tech. rep.*, SAE Technical Paper, 1984.
- [55] M. Bergmann, A. Iollo, Modeling and simulation of fish-like swimming, *J. Comput. Phys.* 230 (2) (2011) 329–348.
- [56] S. Taneda, H. Honji, Unsteady flow past a flat plate normal to the direction of motion, *J. Phys. Soc. Jpn.* 30 (1) (1971) 262–272, <https://doi.org/10.1143/JPSJ.30.262>.
- [57] P. Koumoutsakos, D. Shiels, Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate, *J. Fluid Mech.* 328 (1996) 177–227, <https://doi.org/10.1017/S0022112096008695>.
- [58] R. Mittal, H. Dong, M. Bozkurtas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *J. Comput. Phys.* 227 (10) (2008) 4825–4852.
- [59] L. Shen, E.-S. Chan, P. Lin, Calculation of hydrodynamic forces acting on a submerged moving object using immersed boundary method, *Comput. Fluids* 38 (3) (2009) 691–702.
- [60] T. Johnson, V. Patel, Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.* 378 (1999) 19–70.
- [61] A.G. Tomboulides, S.A. Orszag, Numerical investigation of transitional and weak turbulent flow past a sphere, *J. Fluid Mech.* 416 (2000) 45–73.
- [62] R. Clift, J. Grace, M. Weber, *Bubbles, Drops, and Particles*, Dover Civil and Mechanical Engineering Series, Dover Publications, 2005.
- [63] S. Taneda, Experimental investigation of the wake behind a sphere at low Reynolds numbers, *J. Phys. Soc. Jpn.* 11 (10) (1956) 1104–1108.
- [64] K. Onishi, S. Obayashi, K. Nakahashi, M. Tsubokura, Use of the immersed boundary method within the building cube method and its application to real vehicle cad data, AIAA paper AIAA-2013-2713, 2013.

- [65] G. Vio, S. Watkins, P. Mousley, J. Watmuff, S. Prasad, Flow structures in the near-wake of the Ahmed model, *J. Fluids Struct.* 20 (5) (2005) 673–695.
- [66] P. Moghimi, R. Rafee, Numerical and experimental investigations on aerodynamic behavior of the ahmed body model with different diffuser angles, *J. Appl. Fluid Mech.* 11 (4) (2018) 1101–1113.
- [67] A. Spohn, P. Gilliéron, Flow separations generated by a simplified geometry of an automotive vehicle, in: *IUTAM Symposium: Unsteady Separated Flows*, Citeseer, 2002, pp. 8–12.