An Immersed Boundary Method with Divergence-Free Velocity Interpolation

Yuanxun Bao*, Aleksandar Donev, David M. McQueen, Charles S. Peskin

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY, USA

Boyce E. Griffith

Departments of Mathematics and Biomedical Engineering, Carolina Center for Interdisciplinary Applied Mathematics, and McAllister Heart Institute, University of North Carolina, Chapel Hill, NC, USA

Abstract

The Immersed Boundary (IB) method is a mathematical framework for constructing robust numerical methods to study fluid-structure interaction in problems involving an elastic structure immersed in a viscous fluid. The IB formulation uses an Eulerian representation of the fluid and a Lagrangian representation of the structure. The Lagrangian and Eulerian frames are coupled by integral transforms with delta function kernels. The discretized IB equations use approximations to these transforms with regularized delta function kernels to interpolate the fluid velocity to the structure, and to spread structural forces to the fluid. It is well-known that the conventional IB method can suffer from poor volume conservation since the interpolated Lagrangian velocity field is not generally divergence-free, and so this can cause spurious volume changes. In practice, the lack of volume conservation is especially pronounced for cases where there are large pressure differences across thin structural boundaries. The aim of this paper is to greatly reduce the volume error of the IB method by introducing velocity-interpolation and force-spreading schemes with the property that the interpolated velocity field in which the structure moves is at least $\mathscr{C}^1$ and satisfies a continuous divergence-free condition, and the force-spreading operator is the adjoint of the velocity-interpolation operator. We confirm through numerical experiments in two and three spatial dimensions that our new IB method equipped with divergence-free velocity interpolation and force spreading is able to achieve substantial improvement in volume conservation compared to other existing IB methods, at the expense of a modest increase in the computational cost.

Keywords: Immersed boundary method, fluid-structure interaction, incompressible flow, volume conservation, velocity interpolation, force spreading

1. Introduction

The *Immersed Boundary* (IB) method [1] is a general mathematical framework for numerical solution of *fluid-structure interaction* problems arising from biological and engineering applica-

*Corresponding author

Email addresses: billbao@cims.nyu.edu (Yuanxun Bao), donev@courant.nyu.edu (Aleksandar Donev), mcqueen@cims.nyu.edu (David M. McQueen), peskin@cims.nyu.edu (Charles S. Peskin), boyceg@unc.edu (Boyce E. Griffith)

tions. The IB method was introduced to simulate flow patterns around the heart valves [2, 3], and since its success in modeling cardiac fluid dynamics [4, 5, 6, 7], it has been extended and applied to various other applications, including but not limited to motion of biological swimmers [8, 9], dynamics of red-blood cells [10, 11] and dry foam [12, 13], and rigid body motion [14, 15].

The essence of IB method as a numerical method lies in its simple way of coupling an Eulerian representation of the fluid and a Lagrangian representation of the structure, that is, *force spreading* from the structure to the fluid and *velocity interpolation* from the fluid to the structure are carried out via a regularized delta function δ_h. One effective way to construct δ_h is to require the regularized delta function to satisfy a set of moment conditions to achieve approximate grid translation-invariance and desired interpolation accuracy [16], thereby avoiding any complication of special grid treatment near the fluid-structure interface. In spite of its wide applicability and ease of implementation, the conventional IB method with collocated-grid discretization (IBCollocated) has two well-known shortcomings in accuracy: only first-order convergence for problems that possess sharp-interface solutions [17, 18] and lack of volume conservation in fluid regions enclosed by immersed structure [19]. Much research effort has been put into improving the convergence rate of the IB method to second order or even higher order for problems with singular forcing at the sharp interface, notably, the *Immersed Interface Method* (IIM) [20, 21], and more recently, a new method known as *Immersed Boundary Smooth Extension* [22, 23]. Our focus here, however, is on improving the volume conservation properties of the IB method.

As an immediate consequence of fluid incompressibility, which is one of the basic assumptions of the IB formulation, the volume enclosed by the immersed structure should be exactly conserved as it deforms and moves with the fluid. Thus, it is certainly a desirable feature of an IB method to conserve volume as nearly as possible. However, in practice, it is observed that, even in the simplest case of a quasi-static pressurized membrane [24], the IB method (regardless of collocated- or staggered-grid discretization) still produces volume error that persistently grows in time as if fluid "leaks" through the boundary. An intuitive explanation for this "leak" is that fluid is "squeezing" between the marker points used to discretize the boundary in a conventional IB method; however, this is *not* the full story, as refining the Lagrangian discretization does not improve the volume conservation of the method for a fixed Eulerian discretization. Peskin and Printz realized that the main cause of poor volume conservation of IBCollocated is that the interpolated velocity used to move the structure is not divergence-free in the continuous sense [19], despite that the discrete fluid velocity is enforced to be divergence-free with respect to the discrete divergence operator by the fluid solver. To improve volume conservation of the conventional IB method, Peskin and Printz proposed a modified finite-difference approximation to the discrete divergence operator to ensure that the *average* of the continuous divergence of the interpolated velocity is equal to zero in a small control volume with size of a grid cell [19]. Their IB method with modified finite-difference operators (IBModified) was applied to a heart model in two dimensions, and it achieved improvement in volume conservation by one-to-two orders of magnitude compared to IBCollocated. Nevertheless, a major drawback of IBModified that limits its use in applications is its complex, non-standard finite-difference operators whose coefficients are derived based on the choice of the regularized delta function (but see [12, 13] for applications). To address the issue of spurious currents across immersed structure supporting extremely large pressure differences, Guy and Strychalski developed a different extension of the IB method that uses non-uniform Fast Fourier Transform [25, 26] (NUFFT) to generate "spectral" approximations to the delta function [27], which also has superior volume conservation

property.

Over the past two decades, the staggered-grid (MAC) discretization has been widely adopted by the IB community [5, 7, 8, 14, 15, 28]. In addition to its most celebrated feature of avoiding the odd-even decoupling in the Poisson solver that can otherwise occur with collocated-grid discretization, which leads to "checkerboard" instability in the solutions, Griffith [24] concluded from his numerical studies that the improvement in volume conservation of the IB method with staggered-grid discretization (IBMAC) is essentially the same as that of IBModified. In practice, IBMAC is more favorable than IBModified in that the improvement in volume conservation directly comes as a byproduct of grid discretization without any modification to the finite-difference operators, and it is relatively straightforward to extend IBMAC to include adaptive mesh refinement [5, 29] and physical boundary conditions [30]. However, we emphasize that the nature of Lagrangian velocity interpolation of IBMAC remains the same as that of IBCollocated, and hence, there is much room for further improvement in volume conservation by ensuring that the interpolated velocity is constructed to be nearly or exactly divergence-free. We note that the methods designed to improve the convergence rate of IB methods, such as IIM [20, 21] and the Blob-Projection method [31] also improve volume conservation because the solution near the interface is computed more accurately. These methods, however, are somewhat more complex and less generalizable than the conventional IB method.

This paper is concerned with further improving volume conservation of IBMAC by introducing a velocity interpolation scheme that is divergence-free in the *continuum* sense. In every time step of the conventional IB method, the Lagrangian velocity by which the immersed structure moves is interpolated from a *discretely* divergence-free fluid velocity field obtained from the fluid solver. The key idea is first to construct a discrete *vector potential* that lives on an *edge-centered* staggered grid from the discretely divergence-free fluid velocity, and then to apply the conventional IB interpolation scheme to obtain a continuum vector potential, from which the interpolated velocity field is obtained by applying the *continuous* curl operator. Note that the existence of the discrete vector potential relies on the fact that the discrete velocity field is discretely divergence-free. The interpolated velocity field obtained in this manner is guaranteed to be continuously divergence-free, since the divergence of the curl of any vector field is zero. We also propose a new force-spreading operator that is defined to be the adjoint of velocity interpolation, so that Lagrangian-Eulerian interaction conserves energy. The Eulerian force density that is the result of applying this force-spreading operator to a Lagrangian force field turns out to be discretely divergence-free, so we refer to this new force-spreading operation as divergence-free force spreading. We name the IB method equipped with the new interpolation and spreading operators as the *Divergence-Free Immersed Boundary* (DFIB) method. In contrast to the local nature of interpolation and spreading in the conventional IB method, these operators of the DFIB method turn out to be non-local in that their construction requires the solution of discrete Poisson equations, which can be computed efficiently by the *Fast Fourier Transform* (FFT) or multigrid methods. Another new feature of our method is that transferring information between the Eulerian grid and the Lagrangian mesh involves derivatives of the regularized delta function $\nabla\delta_h$ instead of only δ_h. We confirm through various numerical tests in both two and three spatial dimensions that the DFIB method is able to reduce volume error by several orders of magnitude compared to IBMAC and IBModified at the expense of a modest increase in the computational cost. Moreover, we confirm that the volume error for DFIB decreases as the Lagrangian mesh is refined with the Eulerian grid size held fixed, which is not the case in the conventional IB method [19]. In addition to the substantial improvement in volume conservation, the DFIB

method is quite straightforward to realize from an existing modular IB code with staggered-grid discretization, that is, by simply switching to the new velocity-interpolation and force-spreading schemes while leaving the fluid solver and time-stepping scheme unchanged.

The rest of the paper is organized as follows. In Sec. 2, we begin by giving a brief description of the continuum equations of motion in the IB framework. Then we define the staggered grid on which the fluid variables live and introduce the spatial discretization of the equations of motion. Sec. 3 introduces the two main contributions of this paper: divergence-free velocity interpolation and force spreading. In Sec. 4, we present a formally second-order time-stepping scheme that is used to evolve the spatially-discretized equations, followed by a cost comparison of DFIB and IBMAC. Numerical examples of applying DFIB to problems in two and three spatial dimensions are presented in Sec. 5, where the volume-conserving characteristics of the new scheme are assessed.

2. Equations of motion and spatial discretization

2.1. Equations of motion

This section provides a brief description of the continuum equations of motion in the IB framework [1]. We assume a neutrally-buoyant elastic structure Γ that is described by the Lagrangian variables $\boldsymbol{s}$, immersed in a viscous incompressible fluid occupying the whole fluid domain $\Omega \subset \mathbb{R}^3$ that is described by the Eulerian variables $\boldsymbol{x}$. Eqs. (2.1) and (2.2) are the incompressible Navier-Stokes equations describing mass and momentum conservation of the fluid, in which $\boldsymbol{u}(\boldsymbol{x},t)$ denotes the fluid velocity, $p(\boldsymbol{x},t)$ is the pressure, and $\boldsymbol{f}(\boldsymbol{x},t)$ is the Eulerian force density (force per unit volume) exerted by the structure on the fluid. In this formulation, we assume that the density ρ and the viscosity μ of the fluid are constant. The fluid-structure coupled equations are:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) + \nabla p = \mu\nabla^2\boldsymbol{u} + \boldsymbol{f}, \tag{2.1}$$

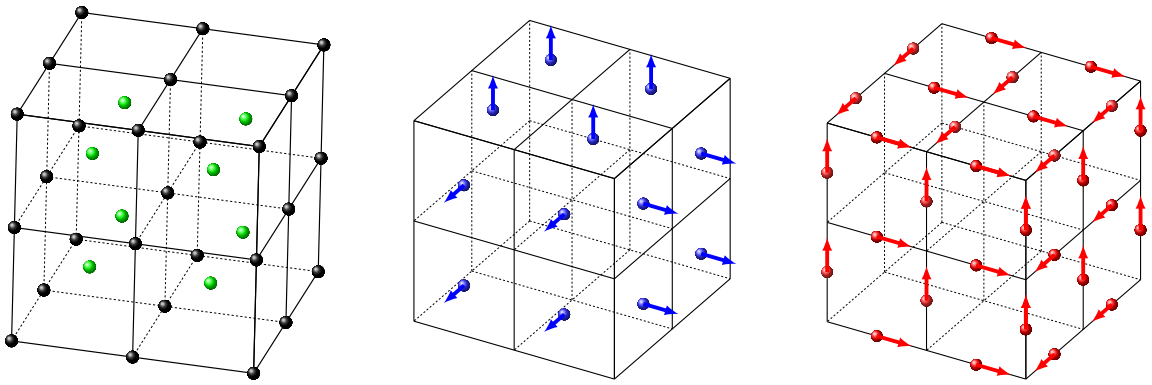
$$\nabla\cdot\boldsymbol{u} = 0, \tag{2.2}$$

$$\boldsymbol{f}(\boldsymbol{x},t) = \int_\Gamma \boldsymbol{F}(\boldsymbol{s},t)\,\delta(\boldsymbol{x}-\boldsymbol{X}(\boldsymbol{s},t))\,\mathrm{d}\boldsymbol{s}, \tag{2.3}$$

$$\frac{\partial \boldsymbol{X}}{\partial t}(\boldsymbol{s},t) = \boldsymbol{u}(\boldsymbol{X}(\boldsymbol{s},t),t) = \int_\Omega \boldsymbol{u}(\boldsymbol{x},t)\,\delta(\boldsymbol{x}-\boldsymbol{X}(\boldsymbol{s},t))\,\mathrm{d}\boldsymbol{x}, \tag{2.4}$$

$$\boldsymbol{F}(\boldsymbol{s},t) = \boldsymbol{\mathcal{F}}[\boldsymbol{X}(\cdot,t)\,;\boldsymbol{s}] = -\frac{\delta E}{\delta \boldsymbol{X}}(\boldsymbol{s},t). \tag{2.5}$$

Eqs. (2.3) and (2.4) are the fluid-structure interaction equations that couple the Eulerian and the Lagrangian variables. Eq. (2.3) relates the Lagrangian force density $\boldsymbol{F}(\boldsymbol{s},t)$ to the Eulerian force density $\boldsymbol{f}(\boldsymbol{x},t)$ using the Dirac delta function, where $\boldsymbol{X}(\boldsymbol{s},t)$ is the physical position of the Lagrangian point $\boldsymbol{s}$. Eq. (2.4) is simply the no-slip boundary condition of the Lagrangian structure, *i.e.*, the Lagrangian point $\boldsymbol{X}(\boldsymbol{s},t)$ moves at the same velocity as the fluid at that point. In Eq. (2.5), the system is closed by expressing the Lagrangian force density $\boldsymbol{F}(\boldsymbol{s},t)$ in the form of a force density functional $\boldsymbol{\mathcal{F}}[\boldsymbol{X}(\cdot,t)\,;\boldsymbol{s}]$, which in many cases can be derived from an elastic energy functional $E[\boldsymbol{X}(\cdot,t)\,;\boldsymbol{s}]$ by taking the variational derivative, denoted here by $\delta/\delta\mathbf{X}$, of the elastic energy. These functionals describe in two different ways the material properties of the immersed elastic structure.



(a) (b) (c)

Fig. 1: Staggered grids on which discrete grid functions are defined. (a) Cell-centered (green) and node-centered (black) grids for scalar functions. (b) Face-centered grid for vector grid functions. (c) Edge-centered grid for vector grid functions.

2.2. Spatial discretization

Throughout the paper, we assume the fluid occupies a *periodic* domain $\Omega = [0, L]^3$ that is discretized by a uniform $N \times N \times N$ Cartesian grid with meshwidth $h = \frac{L}{N}$. Each grid cell is indexed by (i, j, k) for $i, j, k = 0, \ldots, N-1$. For the Eulerian fluid equations, we use the staggered-grid discretization, in which the pressure p is defined on the cell-centered grid (Fig. 1a), denoted by $\mathbb{C}$, *i.e.*, at positions $\mathbf{x}_{i,j,k} = ((i+\frac{1}{2})h, (j+\frac{1}{2})h, (k+\frac{1}{2})h)$. The discrete fluid velocity $\mathbf{u}$ is defined on the face-centered grid (Fig. 1b), denoted by $\mathbb{F}$, with each component perpendicular to the corresponding cell faces, *i.e.*, at positions $\mathbf{x}_{i-\frac{1}{2},j,k}$, $\mathbf{x}_{i,j-\frac{1}{2},k}$ and $\mathbf{x}_{i,j,k-\frac{1}{2}}$ for each velocity component respectively. We also introduce two additional shifted grids: the node-centered grid (Fig. 1a) for scalar grid functions, denoted by $\mathbb{N}$, *i.e.*, at positions $\mathbf{x}_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}$, and the edge-centered grid (Fig. 1c) for vector grid functions, denoted by $\mathbb{E}$, with each component defined to be parallel to the corresponding cell edges *i.e.*, at positions $\mathbf{x}_{i,j-\frac{1}{2},k-\frac{1}{2}}$, $\mathbf{x}_{i-\frac{1}{2},j,k-\frac{1}{2}}$ and $\mathbf{x}_{i-\frac{1}{2},j-\frac{1}{2},k}$ for each component respectively. In Sec. 3, we will use these half-shifted staggered grids to construct divergence-free velocity interpolation and force spreading.

To discretize the differential operators in Eqs. (2.1) and (2.2), we introduce the central difference operators corresponding to the partial derivatives $\partial/\partial x_\alpha$,

$$D_\alpha^h \varphi := \frac{\varphi(\mathbf{x} + \frac{h}{2}\mathbf{e}_\alpha) - \varphi(\mathbf{x} - \frac{h}{2}\mathbf{e}_\alpha)}{h}, \quad \alpha = 1, 2, 3, \tag{2.6}$$

where φ is a scalar grid function and $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is the standard basis of $\mathbb{R}^3$. We can use D_α^h to define the discrete gradient, divergence and curl operators:

$$\mathbf{G}^h \varphi := (D_1^h \varphi,\ D_2^h \varphi,\ D_3^h \varphi), \tag{2.7}$$

$$\mathbf{D}^h \cdot \mathbf{v} := D_\alpha^h v_\alpha, \tag{2.8}$$

$$\mathbf{D}^h \times \mathbf{v} := \epsilon_{ijk} D_j^h v_k, \tag{2.9}$$

where $\mathbf{v}$ is a vector grid function, ϵ_{ijk} is the totally antisymmetric tensor, and the Einstein summation convention is used here. The discrete differential operators may be defined on different pairs of domain and range (half-shifted staggered grids), and therefore, in a slight abuse of

notation, we will use the same notation to denote the different operators,

$$\mathbf{G}^h : \varphi(\mathbb{C}) \longrightarrow \mathbf{v}(\mathbb{F}) \text{ or } \varphi(\mathbb{N}) \longrightarrow \mathbf{v}(\mathbb{E}), \tag{2.10}$$

$$\mathbf{D}^h\cdot : \mathbf{v}(\mathbb{E}) \longrightarrow \varphi(\mathbb{N}) \text{ or } \mathbf{v}(\mathbb{F}) \longrightarrow \varphi(\mathbb{C}), \tag{2.11}$$

$$\mathbf{D}^h\times : \mathbf{v}(\mathbb{E}) \longrightarrow \mathbf{v}(\mathbb{F}) \text{ or } \mathbf{v}(\mathbb{F}) \longrightarrow \mathbf{v}(\mathbb{E}). \tag{2.12}$$

Although the curl operator does not appear in the equations of motion explicitly, we define it here for use in Sec. 3. The discrete scalar Laplacian operator can be defined by $L^h = \mathbf{D}^h \cdot \mathbf{G}^h$, which yields the familiar compact second-order approximation to ∇^2:

$$L^h\varphi := \sum_{\alpha=1}^{3} \frac{\varphi(\mathbf{x}+h\mathbf{e}_\alpha) - 2\varphi(\mathbf{x}) + \varphi(\mathbf{x}-h\mathbf{e}_\alpha)}{h^2}. \tag{2.13}$$

Note that the range and domain of L^h are a set of grid functions defined on the same grid, and that grid can be $\mathbb{C}$ or $\mathbb{N}$ or any of the three subgrids of $\mathbb{E}$ or $\mathbb{F}$ on which the different components of vector-valued functions are defined. We will use the notation $\mathbf{L}^h$ to denote the discrete vector Laplacian operator that applies (the appropriately shifted) L^h to each component of a vector grid function.

We follow the same treatment of discretization of the advection term as in earlier presentations of the IB method [28]. From the incompressibility of the fluid flow $\nabla \cdot \boldsymbol{u} = 0$, we can write the advection term in the skew-symmetric form

$$[(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}]_\alpha = \frac{1}{2}\boldsymbol{u}\cdot(\nabla u_\alpha) + \frac{1}{2}\nabla\cdot(\boldsymbol{u}u_\alpha), \quad \alpha = 1,2,3. \tag{2.14}$$

Let $\mathbf{S}(\mathbf{u})\mathbf{u}$ denote the discretization of Eq. (2.14), and we define

$$[\mathbf{S}(\mathbf{u})\mathbf{u}]_\alpha = \frac{1}{2}\tilde{\mathbf{u}}\cdot\mathbf{G}^{2h}u_\alpha + \frac{1}{2}\mathbf{D}^{2h}\cdot(\tilde{\mathbf{u}}u_\alpha), \quad \alpha = 1,2,3, \tag{2.15}$$

where $\tilde{\mathbf{u}}$ denotes an averaged collocated advective velocity whose components all live on the same grid as u_α. In the work of [28], the advective velocity $\tilde{\mathbf{u}}$ is obtained by using the same interpolation scheme as the one used for moving the immersed structure. In our work, we simply take the average of $\mathbf{u}$ on the grid. For example, the three components of $\tilde{\mathbf{u}}$ in the x-component equation are

$$\begin{aligned}
\tilde{u}_1 &= u_1(\mathbf{x}_{i-\frac{1}{2},j,k}), \\
\tilde{u}_2 &= \frac{u_2(\mathbf{x}_{i,j-\frac{1}{2},k}) + u_2(\mathbf{x}_{i,j+\frac{1}{2},k}) + u_2(\mathbf{x}_{i-1,j-\frac{1}{2},k}) + u_2(\mathbf{x}_{i-1,j+\frac{1}{2},k})}{4}, \\
\tilde{u}_3 &= \frac{u_3(\mathbf{x}_{i,j,k-\frac{1}{2}}) + u_3(\mathbf{x}_{i,j,k+\frac{1}{2}}) + u_3(\mathbf{x}_{i-1,j,k-\frac{1}{2}}) + u_3(\mathbf{x}_{i-1,j,k+\frac{1}{2}})}{4}.
\end{aligned}$$

Note that in the y- and z-component equations, we need different averages of $\mathbf{u}$ to construct $\tilde{\mathbf{u}}$. We choose to use the wide-stencil operators in Eq. (2.15) so that the resulting grid functions are all defined on the same grid as u_α. A more compact discretization of the advection term has been previously described in [24, 30, 32].

The immersed structure Γ is discretized by a Lagrangian mesh of M points or markers, and we denote the Cartesian position of the m^{th} Lagrangian marker $\mathbf{X}_m$ and the Lagrangian force

density $\mathbf{F}_m$, for $m = 1, \ldots, M$. The discretized interaction equations (Eqs. (2.3) and (2.4)) can be concisely written in the form

$$\mathbf{f}(\mathbf{x}) = \boldsymbol{\mathcal{S}}[\mathbf{X}]\mathbf{F}, \tag{2.16}$$

$$\mathbf{U}(\mathbf{X}) = \boldsymbol{\mathcal{S}}^*[\mathbf{X}]\mathbf{u}, \tag{2.17}$$

where $\boldsymbol{\mathcal{S}}$ denotes the force-spreading operator that spreads a collection of discrete Lagrangian force densities to the Eulerian fluid, and $\boldsymbol{\mathcal{S}}^*$ denotes the adjoint velocity-interpolation operator that interpolates the discrete fluid velocity $\mathbf{u}$ to obtain the Lagrangian velocity at $\mathbf{X}$. In the conventional IB method, these operators are simply discrete approximations of the surface and volume integrals in Eqs. (2.3) and (2.4), *i.e.*,

$$\boldsymbol{\mathcal{S}}[\mathbf{X}]\mathbf{F} := \sum_m^M \mathbf{F}_m \, \delta_h(\mathbf{x} - \mathbf{X}_m)\Delta\mathbf{s}, \tag{2.18}$$

$$\boldsymbol{\mathcal{S}}^*[\mathbf{X}]\mathbf{u} := \sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X})h^3. \tag{2.19}$$

Note that Eq. (2.19) is a vector equation. For each of the three components of the equation, the sum $\mathbf{x} \in \mathbb{F}$ is to be understood here and in similar expressions as the sum over the appropriate subgrid of $\mathbb{F}$. In Eqs. (2.18) and (2.19), the Dirac delta function is replaced by a regularized delta function δ_h to facilitate the coupling between the Eulerian and Lagrangian grids, which is taken to be of the tensor-product form

$$\delta_h(\mathbf{x}) = \frac{1}{h^3}\phi\left(\frac{x_1}{h}\right)\phi\left(\frac{x_2}{h}\right)\phi\left(\frac{x_3}{h}\right), \tag{2.20}$$

where $\phi(r)$ denotes the one-dimensional immersed-boundary kernel that is constructed from a set of moment conditions to achieve approximate grid translation invariance [1, 16]. The conventional IB interpolation Eq. (2.19) does not give a Lagrangian velocity that is continuously divergence-free, *i.e.*,

$$\nabla \cdot \mathbf{U}(\mathbf{X}) = -\sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x}) \cdot (\nabla\delta_h)(\mathbf{x} - \mathbf{X})h^3 \neq 0, \tag{2.21}$$

even though $\mathbf{u}$ is discretely divergence-free. In the following section, we develop a new velocity interpolation and force spreading scheme that produces an exactly divergence-free interpolated velocity field $\mathbf{U}(\mathbf{X})$ given a discretely divergence-free fluid velocity field $\mathbf{u}(\mathbf{x})$. Moreover, the new force-spreading operator that we shall describe is the adjoint of the new velocity-interpolation operator.

In summary, the spatially-discretized equations of motion are

$$\rho\left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{S}(\mathbf{u})\mathbf{u}\right) + \mathbf{G}^h p = \mu\mathbf{L}^h\mathbf{u} + \boldsymbol{\mathcal{S}}[\mathbf{X}]\mathbf{F}, \tag{2.22}$$

$$\mathbf{D}^h \cdot \mathbf{u} = 0, \tag{2.23}$$

$$\frac{\mathrm{d}\mathbf{X}_m}{\mathrm{d}t} = \mathbf{U}(\mathbf{X}_m, t) = \boldsymbol{\mathcal{S}}^*[\mathbf{X}_m]\mathbf{u}. \tag{2.24}$$

3. Divergence-free velocity interpolation and force spreading

This section presents the two main contributions of this paper: divergence-free velocity interpolation and force spreading. Familiarity with discrete differential operators on staggered grids and with some discrete vector identities, reviewed and summarized in Appendix A, will facilitate the reading of this section.

3.1. Divergence-free velocity interpolation

Here we introduce a new recipe for computing an interpolated velocity field $\mathbf{U}(\mathbf{X}) = \boldsymbol{\mathcal{S}}^*[\mathbf{X}]\mathbf{u}$, that is divergence-free in the continuous sense, *i.e.*, $\nabla \cdot \mathbf{U}(\mathbf{X}) = 0$ for all $\mathbf{X}$. For now we drop the dependence on time and assume $\mathbf{X}$ is defined everywhere in the domain $\Omega \subset \mathbb{R}^3$, not just on the Lagrangian structure Γ. The main idea is first to construct a discrete vector potential $\mathbf{a}(\mathbf{x})$ that is defined on the edge-centered staggered grid $\mathbb{E}$, and then to apply the conventional IB interpolation to $\mathbf{a}(\mathbf{x})$ to obtain a continuum vector potential $\mathbf{A}(\mathbf{X})$, so that the Lagrangian velocity defined by $\mathbf{U}(\mathbf{X}) = \nabla \times \mathbf{A}$ is automatically divergence-free.

Suppose the discrete velocity field $\mathbf{u}(\mathbf{x})$ is defined on $\mathbb{F}$ and is discretely divergence-free, *i.e.*, $\mathbf{D}^h \cdot \mathbf{u} = 0$. Let $\mathbf{u}_0$ be the mean of $\mathbf{u}(\mathbf{x})$,

$$\mathbf{u}_0 = \frac{1}{V} \sum_{\mathbf{x} \in \mathbb{F}} \mathbf{u}(\mathbf{x}) h^3, \tag{3.1}$$

where $V = \sum_{\mathbf{x} \in \mathbb{F}} h^3$ is the volume of the domain. Making use of the Helmholtz decomposition, we construct a discrete velocity potential $\mathbf{a}(\mathbf{x})$ for $\mathbf{x} \in \mathbb{E}$ that satisfies

$$\begin{cases} \mathbf{D}^h \times \mathbf{a} &= \mathbf{u} - \mathbf{u}_0, \\ \mathbf{D}^h \cdot \mathbf{a} &= 0, \end{cases} \tag{3.2}$$

where $\mathbf{a}(\mathbf{x})$ being divergence-free is an arbitrary gauge condition that makes $\mathbf{a}(\mathbf{x})$ uniquely defined up to a constant. Note that $\mathbf{D}^h \cdot \mathbf{a}$ is a scalar field defined on $\mathbb{N}$. In Appendix B, we prove that the the discrete vector potential $\mathbf{a}(\mathbf{x})$ defined by Eq. (3.2) exists (see Theorem 3). To determine $\mathbf{a}(\mathbf{x})$ explicitly, we take the discrete curl of the first equation in Eq. (3.2) and use the identity Eq. (A.3), which leads to a vector Poisson equation for $\mathbf{a}(\mathbf{x})$,

$$-\mathbf{L}^h \mathbf{a} = \mathbf{D}^h \times \mathbf{u}, \tag{3.3}$$

that can be efficiently solved. Note that the solution of the Poisson problem Eq. (3.3) determines $\mathbf{a}(\mathbf{x})$ up to an arbitrary constant. It is not necessary to uniquely determine $\mathbf{a}(\mathbf{x})$ because the constant term vanishes from differentiation later.

The next step is to interpolate the discrete vector potential $\mathbf{a}(\mathbf{x})$ in the conventional IB fashion to obtain the continuum vector potential

$$\mathbf{A}(\mathbf{X}) = \sum_{\mathbf{x} \in \mathbb{E}} \mathbf{a}(\mathbf{x})\, \delta_h(\mathbf{x} - \mathbf{X}) h^3. \tag{3.4}$$

Lastly, we take the continuum curl of $\mathbf{A}(\mathbf{X})$ with respect to $\mathbf{X}$,

$$(\nabla \times \mathbf{A})(\mathbf{X}) = \sum_{\mathbf{x} \in \mathbb{E}} \mathbf{a}(\mathbf{x}) \times (\nabla \delta_h)(\mathbf{x} - \mathbf{X}) h^3, \tag{3.5}$$

and our new interpolation is completed by adding the mean flow $\mathbf{u}_0$, that is,

$$\mathbf{U}(\mathbf{X}) = \boldsymbol{\mathcal{S}}^*[\mathbf{X}]\mathbf{u} = \mathbf{u}_0 + \sum_{\mathbf{x}\in\mathbb{E}} \mathbf{a}(\mathbf{x}) \times (\nabla\delta_h)(\mathbf{x}-\mathbf{X})h^3. \tag{3.6}$$

We note that the interpolation Eq. (3.4) is not performed in the actual implementation of the scheme but $\nabla\delta_h$ is computed on the edge-centered staggered grid $\mathbb{E}$ in Eq. (3.6). We emphasize again that, by construction, the interpolated velocity in Eq. (3.6) is exactly divergence-free.

There are two important features of our new interpolation scheme that are worth mentioning. First, in comparison to locally interpolating the velocity from the nearby fluid grid in the conventional IB method, our new interpolation scheme is non-local, in that it involves solving the discrete Poisson problem Eq. (3.3). Second, we note that the gradient of the regularized delta function $\nabla\delta_h$, instead of δ_h itself, appears in the new interpolation scheme in Eq. (3.6). We have numerically found that the improvement in volume conservation of the DFIB method depends on the smoothness of the IB kernel that is used to construct δ_h. With $\mathscr{C}^1$ IB kernels, such as the standard 4-point IB kernel [1] (denoted by ϕ_{4h}), we achieve little improvement in volume conservation of DFIB compared to IBMAC unless the Lagrangian mesh is discretized with impractically high resolution (4 to 8 markers per fluid meshwidth, see Fig. 5). On the other hand, we can obtain substantial reduction in volume error by using IB kernels that are at least $\mathscr{C}^2$ with only a moderate resolution of the Lagrangian mesh (1 to 2 markers per fluid meshwidth). We do not have a clear mathematical argument for the connection between volume conservation and smoothness of IB kernels. However, it is not surprising that higher regularity of the IB kernel certainly would be helpful since the derivative of δ_h is needed. The IB kernels that will be used later in Sec. 5 include the $\mathscr{C}^3$ 5-point and 6-point kernels [16] (denoted by ϕ^{new}_{5h} and ϕ^{new}_{6h} respectively), and the $\mathscr{C}^2$ 4-point B-spline kernel [33],

$$\phi^B_{4h}(r) = \begin{cases} \frac{2}{3} - r^2 + \frac{1}{2}r^3 & 0 \le |r| < 1, \\ \frac{4}{3} - 2r + r^2 - \frac{1}{6}r^3 & 1 \le |r| < 2, \\ 0 & |r| \ge 2, \end{cases} \tag{3.7}$$

and the $\mathscr{C}^4$ 6-point B-spline kernel,

$$\phi^B_{6h}(r) = \begin{cases} \frac{11}{20} - \frac{1}{2}r^2 + \frac{1}{4}r^4 - \frac{1}{12}r^5 & 0 \le |r| < 1, \\ \frac{17}{40} + \frac{5}{8}r - \frac{7}{4}r^2 + \frac{5}{4}r^3 - \frac{3}{8}r^4 + \frac{1}{24}r^5 & 1 \le |r| < 2, \\ \frac{81}{40} - \frac{27}{8}r + \frac{9}{4}r^2 - \frac{3}{4}r^3 + \frac{1}{8}r^4 - \frac{1}{120}r^2 & 2 \le |r| < 3, \\ 0 & |r| \ge 3. \end{cases} \tag{3.8}$$

These B-spline kernels are members of a sequence of functions obtained by recursively convolving each successive kernel function against a rectangular pulse (also known as the window function), starting from the window function itself [33]. The limiting function in this sequence is a Gaussian [34], which is exactly translation-invarant and isotropic. The family of IB kernels with nonzero even moment conditions, such as ϕ_{4h} and ϕ^{new}_{6h}, also have a Gaussian-like shape, but it is not currently known whether this sequence of functions also converges to a Gaussian.

3.2. The force-spreading operator

The force-spreading operator is constructed to be adjoint to velocity interpolation so that energy is conserved by the Lagrangian-Eulerian interaction, that is, the power generated by the

elastic body forces is transferred to the fluid without loss,[1]

$$\sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x})\cdot\mathbf{f}(\mathbf{x})\,h^3 = \sum_{m=1}^{M} \mathbf{U}_m\cdot\mathbf{F}_m\Delta\mathbf{s}, \tag{3.9}$$

where $\mathbf{U}_m = \mathbf{U}(\mathbf{X}_m)$ is the Lagrangian velocity, and $\mathbf{F}_m\Delta\mathbf{s}$ is the Lagrangian force applied to the fluid by the Lagrangian marker $\mathbf{X}_m$. Our goal is to find an Eulerian force density $\mathbf{f}(\mathbf{x})$ that satisfies the power identity Eq. (3.9). To see what Eq. (3.9) implies about $\mathbf{f}(\mathbf{x})$, we rewrite both sides in terms of $\mathbf{a}(\mathbf{x})$. On the left-hand side of Eq. (3.9), we use Eq. (3.2) to obtain

$$\begin{aligned}\sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x})\cdot\mathbf{f}(\mathbf{x})\,h^3 &= \mathbf{u}_0\cdot\sum_{\mathbf{x}\in\mathbb{F}}\mathbf{f}(\mathbf{x})h^3 + \sum_{\mathbf{x}\in\mathbb{F}}(\mathbf{D}^h\times\mathbf{a})(\mathbf{x})\cdot\mathbf{f}(\mathbf{x})\,h^3\\ &= \mathbf{u}_0\cdot\mathbf{f}_0 V + \sum_{\mathbf{x}\in\mathbb{E}}\mathbf{a}(\mathbf{x})\cdot(\mathbf{D}^h\times\mathbf{f})(\mathbf{x})\,h^3,\end{aligned} \tag{3.10}$$

where the average of $\mathbf{f}(\mathbf{x})$ over the domain is

$$\mathbf{f}_0 = \frac{1}{V}\sum_{\mathbf{x}\in\mathbb{F}}\mathbf{f}(\mathbf{x})\,h^3. \tag{3.11}$$

Note that we have used the summation-by-parts identity Eq. (A.5) to transfer the discrete curl operator $\mathbf{D}^h\times$ from $\mathbf{a}(\mathbf{x})$ to $\mathbf{f}(\mathbf{x})$, and thus, the grid on which the summation is performed in Eq. (3.10) is $\mathbb{E}$ not $\mathbb{F}$. On the the right-hand side of Eq. (3.9), we substitute for $\mathbf{U}_m$ by using the divergence-free velocity interpolation Eq. (3.6),

$$\begin{aligned}\sum_{m=1}^{M}\mathbf{U}_m\cdot\mathbf{F}_m\Delta\mathbf{s} &= \mathbf{u}_0\cdot\sum_{m=1}^{M}\mathbf{F}_m\Delta\mathbf{s} + \sum_{m=1}^{M}\sum_{\mathbf{x}\in\mathbb{E}}\mathbf{a}(\mathbf{x})\times(\nabla\delta_h)(\mathbf{x}-\mathbf{X})\cdot(\mathbf{F}_m\Delta\mathbf{s})\,h^3\\ &= \mathbf{u}_0\cdot\sum_{m=1}^{M}\mathbf{F}_m\Delta\mathbf{s} + \sum_{\mathbf{x}\in\mathbb{E}}\mathbf{a}(\mathbf{x})\cdot\sum_{m=1}^{M}(\nabla\delta_h)(\mathbf{x}-\mathbf{X})\times(\mathbf{F}_m\Delta\mathbf{s})\,h^3.\end{aligned} \tag{3.12}$$

Since $\mathbf{u}_0$ and $\mathbf{a}(\mathbf{x})$ are arbitrary (except for $\mathbf{D}^h\cdot\mathbf{a}=0$), the power identity Eq. (3.9) is satisfied if and only if

$$\mathbf{f}_0 = \frac{1}{V}\sum_{m=1}^{M}\mathbf{F}_m\Delta\mathbf{s} \tag{3.13}$$

and

$$\mathbf{D}^h\times\mathbf{f} = \sum_{k=1}^{M}(\nabla\delta_h)(\mathbf{x}-\mathbf{X})\times(\mathbf{F}_m\Delta\mathbf{s}) + \mathbf{G}^h\varphi,\ \text{ for all } \mathbf{x}\in\mathbb{E}, \tag{3.14}$$

where φ is an arbitrary scalar field that lives on the node-centered grid $\mathbb{N}$. Note that we have the freedom to add the term $\mathbf{G}^h\varphi$ in Eq. (3.14), since from the identity Eq. (A.4) and $\mathbf{D}^h\cdot\mathbf{a}=0$, we have

$$\sum_{\mathbf{x}\in\mathbb{E}}\mathbf{a}(\mathbf{x})\cdot\left(\mathbf{G}^h\varphi\right)h^3 = -\sum_{\mathbf{x}\in\mathbb{N}}\left(\mathbf{D}^h\cdot\mathbf{a}\right)(\mathbf{x})\,\varphi(\mathbf{x})\,h^3 = 0.$$

[1]Here and in similar expressions, $\sum_{\mathbf{x}\in\mathbb{F}}\mathbf{u}(\mathbf{x})\cdot\mathbf{f}(\mathbf{x})h^3$ is a shorthand for $\sum_{i=1}^{3}\sum_{\mathbf{x}\in\mathbb{F}}u_i(\mathbf{x})f_i(\mathbf{x})h^3$.

Indeed, we are required to include this term since the left-hand side of Eq. (3.14) is discretely divergence-free but there is no reason to expect the first term on the right-hand side of Eq. (3.14) is also divergence-free. Note that it is not required to find φ in order to determine $\mathbf{f}(\mathbf{x})$, because we can eliminate φ by taking the discrete curl on both sides of Eq. (3.14),

$$\mathbf{D}^h \times (\mathbf{D}^h \times \mathbf{f}) = \mathbf{D}^h \times \left(\sum_{m=1}^{M} (\nabla \delta_h)(\mathbf{x} - \mathbf{X}) \times (\mathbf{F}_m \Delta \mathbf{s}) \right), \text{ for all } \mathbf{x} \in \mathbb{E}. \tag{3.15}$$

By imposing the gauge condition

$$\mathbf{D}^h \cdot \mathbf{f} = 0, \tag{3.16}$$

we obtain a vector Poisson equation for $\mathbf{f}(\mathbf{x})$,

$$-(\mathbf{L}^h \mathbf{f})(\mathbf{x}) = \mathbf{D}^h \times \left(\sum_{m=1}^{M} (\nabla \delta_h)(\mathbf{x} - \mathbf{X}) \times (\mathbf{F}_m \Delta \mathbf{s}) \right), \text{ for all } \mathbf{x} \in \mathbb{E}. \tag{3.17}$$

Note again that $\nabla \delta_h$ is computed on $\mathbb{E}$, so that the cross-product with $\mathbf{F}_m$ is face-centered, which agrees with the left-hand side of Eq. (3.17). Note that the solution of Eq. (3.17) can be uniquely determined by the choice of $\mathbf{f}_0$. Like our velocity interpolation scheme, the new force-spreading scheme is also non-local because it requires solving discrete Poisson equations. We remark that the new force-spreading scheme is also constructed so that the resulting force density $\mathbf{f}(\mathbf{x})$ is discretely divergence-free. This means that $\mathbf{f}(\mathbf{x})$ includes the pressure gradient that is generated by the Lagrangian forces. We do not see a straightforward way to separate the pressure gradient from $\mathbf{f}(\mathbf{x})$ in case it is needed for output purposes.

4. Time-stepping scheme

In this section, we present a second-order time-stepping scheme, similar to the ones developed previously [7], that evolves the spatially-discretized system Eqs. (2.22) to (2.24). Let $\mathbf{u}^n, \mathbf{X}^n$ denote the approximations of $\boldsymbol{u}$ and $\boldsymbol{X}$ at time $t_n = n\Delta t$. To advance the solutions to $\mathbf{u}^{n+1}$ and $\mathbf{X}^{n+1}$, we perform the following steps:

Step 1. First, update the Lagrangian markers to the intermediate time step $n + \frac{1}{2}$ using the divergence-free velocity interpolation Eq. (3.6),

$$\widetilde{\mathbf{X}}^{n+\frac{1}{2}} = \mathbf{X}^n + \frac{\Delta t}{2} \boldsymbol{\mathcal{S}}^* \left[\mathbf{X}^n \right] \mathbf{u}^n. \tag{4.1}$$

Step 2. Evaluate the intermediate Lagrangian force density at $\widetilde{\mathbf{X}}^{n+\frac{1}{2}}$ from the force density functional or the energy functional, and spread it to the Eulerian grid using the force-spreading scheme Eq. (3.17) to get

$$\mathbf{f}^{n+\frac{1}{2}} = \boldsymbol{\mathcal{S}} \left[\widetilde{\mathbf{X}}^{n+\frac{1}{2}} \right] \mathbf{F}^{n+\frac{1}{2}}. \tag{4.2}$$

Step 3. Solve the fluid equations on the periodic grid [28],

$$\begin{cases} \rho \left(\dfrac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \widetilde{\mathbf{N}}^{n+\frac{1}{2}} \right) + \mathbf{G}^h p^{n+\frac{1}{2}} = \mu \mathbf{L}^h \left(\dfrac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + \mathbf{f}^{n+\frac{1}{2}}, \\ \mathbf{D}^h \cdot \mathbf{u}^{n+1} = 0, \end{cases} \tag{4.3}$$

	# of scalar Poisson solves				# of scalar interpolation/spreading			
	2D		3D		2D		3D	
	DFIB	IBMAC	DFIB	IBMAC	DFIB	IBMAC	DFIB	IBMAC
$\boldsymbol{\mathcal{S}}^*$ in Eq. (4.1)	1	-	3	-	2	2	6	3
$\boldsymbol{\mathcal{S}}$ in Eq. (4.2)	2	-	3	-	2	2	6	3
Fluid solver	3	3	4	4	-	-	-	-
$\boldsymbol{\mathcal{S}}^*$ in Eq. (4.5)	1	-	3	-	2	2	6	3
Total	7	3	13	4	6	6	18	9

Table 1: Cost of DFIB versus IBMAC in terms of the number of scalar Poisson solves and interpolation/spreading of a scalar from/to the Eulerian grid.

where the second-order Adams-Bashforth (AB2) method is applied to approximate the nonlinear advection term

$$\widetilde{\mathbf{N}}^{n+\frac{1}{2}} = \frac{3}{2}\mathbf{N}^n - \frac{1}{2}\mathbf{N}^{n-1}, \tag{4.4}$$

and $\mathbf{N}^n = \boldsymbol{S}(\mathbf{u}^n)\mathbf{u}^n$.

Step 4. In the last step, update the Lagrangian markers $\mathbf{X}^{n+1}$ by using the mid-point approximation

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \Delta t\, \boldsymbol{\mathcal{S}}^* \left[\widetilde{\mathbf{X}}^{n+\frac{1}{2}}\right] \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2}\right). \tag{4.5}$$

Note that the time-stepping scheme described above requires two starting values because of the treatment of the nonlinear advection term using the AB2. To get the starting value at $t = \Delta t$, we can use the second-order Runge-Kutta (RK2) scheme described in [1, 28].

In Table 1 we compare the cost of DFIB and IBMAC for the above IB scheme in terms of the number of the two cost-dominating procedures: the scalar Poisson solver which costs $\mathcal{O}(N^d \log N)$ using FFT on the periodic domain, where $d \in \{2, 3\}$ is the spatial dimension, and spreading/interpolation of a scalar field between the Eulerian grid and the Lagrangian mesh which costs $\mathcal{O}(M)$. In summary, DFIB is only more expensive than IBMAC by 4 scalar Poisson solves for two-dimensional (2D) problems, and is more expensive by 9 scalar Poisson solves and 9 scalar interpolation and spreading for three dimensional (3D) problems. Therefore, the DFIB method is about two times slower than IBMAC per time step in 3D. We point out that if the RK2 scheme [1, 28] is employed rather than the scheme above, then we can save one interpolation step per time step, but the fluid equations need to be solved twice.

5. Numerical Results

This section presents numerical results of the DFIB method for various benchmark problems in 2D and 3D. We first consider in 2D a thin elastic membrane subject to surface tension of the membrane only. The continuum solution of this simple 2D problem has the special feature that the tangential component of the elastic force vanishes, and therefore, the normal derivative of the tangential fluid velocity does not suffer any jump across the immersed boundary. This has the effect that second-order convergence in the fluid velocity $\boldsymbol{u}$ and the Lagrangian deformation map $\boldsymbol{X}$ can be achieved [18]. In the second set of tests, we compare volume conservation in 2D, *i.e.*,

area conservation of DFIB and IBMAC by applying them to a circular membrane under tension, and we discuss the connection between area conservation and the choice of Lagrangian marker spacing relative to the Eulerian grid size. In the third set of computations, we apply the DFIB method to a problem in which a 2D elastic membrane actively evolves in a parametrically-forced system. In the last set of numerical experiments, we extend the surface tension problem to 3D, and compare volume conservation of DFIB with that of IBMAC.

5.1. A thin elastic membrane with surface tension in 2D

It is well-known that the solutions to problems involving an infinitely thin massless membrane interacting with a viscous incompressible fluid possess jump discontinuities across the interface in the pressure and in the normal derivative of the velocity due to singular forcing at the interface [21, 35]. These sharp jump discontinuities cannot be fully resolved by the IB method because of the use of the regularized delta function at the interface. Consequently, the numerical convergence rate for the Lagrangian deformation $\boldsymbol{X}$ is only first order even though the discretization is carried out with second-order accuracy. To achieve the expected rate of convergence, we consider problems with solutions that possess sufficient smoothness.

As a simple benchmark problem with a sufficiently smooth continuum solution we consider a thin elastic membrane that deforms in response to surface tension only. Suppose that the elastic interface Γ is discretized by a collection of Lagrangian markers $\{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$. The discrete elastic energy functional associated with the surface tension of the membrane is the total (polygonal) arc-length of the interface [12],

$$E[\mathbf{X}_1, \ldots, \mathbf{X}_M] = \gamma \sum_{m=1}^{M} |\mathbf{X}_m - \mathbf{X}_{m-1}|, \tag{5.1}$$

where $\mathbf{X}_0 = \mathbf{X}_M$ and γ is the surface tension constant (energy per unit length). The Lagrangian force generated by the energy functional at the marker $\mathbf{X}_m$ is

$$\mathbf{F}_m \Delta s = -\frac{\partial E}{\partial \mathbf{X}_m} = \gamma \left(\frac{\mathbf{X}_{m+1} - \mathbf{X}_m}{|\mathbf{X}_{m+1} - \mathbf{X}_m|} - \frac{\mathbf{X}_m - \mathbf{X}_{m-1}}{|\mathbf{X}_m - \mathbf{X}_{m-1}|} \right). \tag{5.2}$$

In our tests, we set the initial configuration of the membrane to be the ellipse

$$\boldsymbol{X}(s, 0) = L \cdot \left(\frac{1}{2} + \frac{5}{28}\cos(s),\ \frac{1}{2} + \frac{7}{20}\sin(s) \right), \quad s \in [0, 2\pi]. \tag{5.3}$$

The Eulerian fluid domain $\Omega = [0, L]^2$ is discretized by a uniform $N \times N$ Cartesian grid with meshwidth $h = \frac{L}{N}$ in each direction. The elastic interface Γ is discretized by a uniform Lagrangian mesh of size $M = \lceil \pi N \rceil$ in the Lagrangian variable s, so that the Lagrangian markers $\{\mathbf{X}_1, \ldots, \mathbf{X}_M\}$ are physically separated by a distance of approximately $\frac{h}{2}$ in the equilibrium circular configuration. In all of our tests, we set $L = 5,\ \rho = 1,\ \gamma = 1,\ \mu = 0.1$. The time-step size is chosen to be $\Delta t = \frac{h}{2}$ to ensure the stability of all simulations up to $t = 20$ when the elastic interface is empirically observed to be in equilibrium.

We denote by $\mathbf{u}^N(t)$ the computed fluid velocity field and by $\mathcal{I}^{2N \to N}$ a restriction operator from the finer grid of size $2N \times 2N$ to the coarser grid of size $N \times N$. The discrete l_p-norm of the successive error in the velocity component u_i is defined by

$$\varepsilon^N_{p,u,i}(t) = \left\| u_i^N(t) - \mathcal{I}^{2N \to N} u_i^{2N}(t) \right\|_p. \tag{5.4}$$

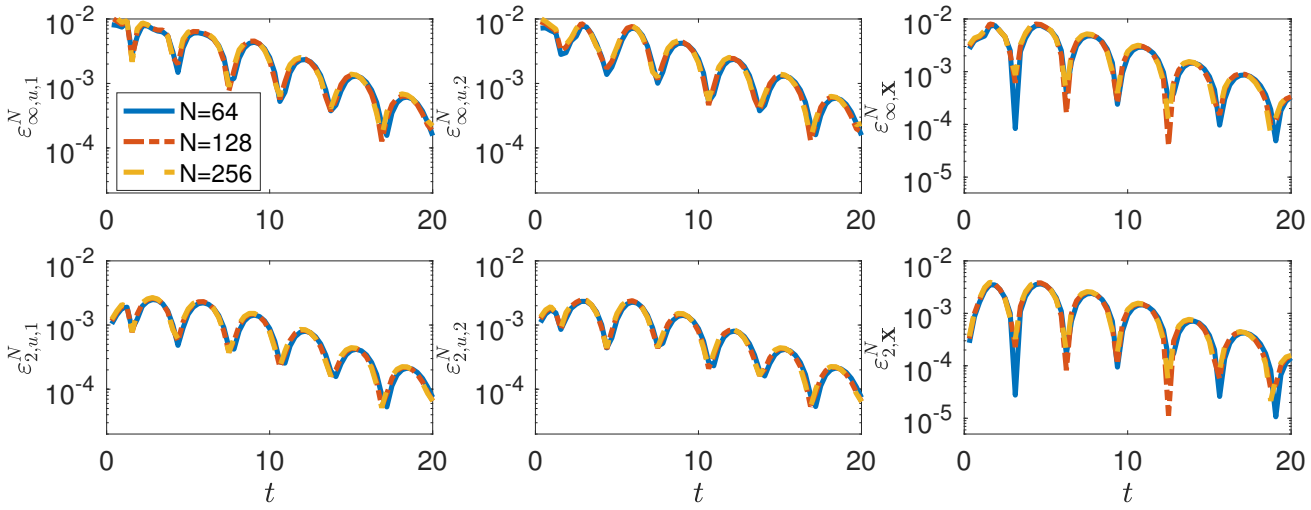


Fig. 2: Rescaled l_∞-norm (top panel) and l_2-norm (bottom panel) errors of the x, y-component of the fluid velocity defined by Eq. (5.4), and errors of the Lagrangian deformation map defined by Eq. (5.5) for the 2D surface tension problem are plotted as a function of time from $t = 0$ to $t = 20$. The left and middle columns show errors of the fluid velocity components and the right column shows errors of the Lagrangian deformation map. The Eulerian grid sizes are $N = 64, 128, 256$ and the corresponding Lagrangian mesh sizes are $M = 202, 403, 805$, so that the spacing between two Lagrangian markers is kept at a distance of approximately $\frac{h}{2}$ in the equilibrium configuration. For the finer grid resolution $N = 128, 256$, the errors in each norm are multiplied by a factor of 4 and 4^2 respectively. After rescaling, the error curves of the finer grid resolution almost align with the error curves of grid resolution $N = 64$, which indeed confirms that second-order convergence in $\mathbf{u}$ and $\mathbf{X}$ is achieved. For this set of computations, we use the $\mathscr{C}^3$ 6-point IB kernel in the discrete delta function, and the time-step size is chosen to be $\Delta t = \frac{h}{2}$.

To avoid artifacts in the error-norm computation because of Lagrangian markers getting too clustered during the simulation, we re-parametrize the interface (for the purpose of the error computation only) from the computed markers using periodic cubic splines after each time step, and compute the l_p-norm error of $\mathbf{X}$ based on a collection of M' uniformly sampled markers $\widetilde{\mathbf{X}}$ from the re-parametrized interface, that is,

$$\varepsilon^N_{p,\mathbf{X}}(t) = \left\| \widetilde{\mathbf{X}}^N(t) - \widetilde{\mathbf{X}}^{2N}(t) \right\|_p , \tag{5.5}$$

where M' does not change with N. We emphasize that the re-sampled markers are only used to compute the error norm and are discarded after each time step. In Fig. 2 the successive l_∞-norm and l_2-norm errors of the x, y-component of the fluid velocity and of the deformation map are plotted as a function of time from $t = 0$ to $t = 20$ for grid resolution $N = 64, 128$ and 256. The number of re-sampled markers for computing $\varepsilon^N_{p,\mathbf{X}}(t)$ is $M' = 128$. To clearly visualize that second-order convergence is achieved by our scheme, we multiply the computed errors for the finer grid resolution $N = 128$ and 256 by a factor of 4 and 4^2 respectively, and plot them along with errors for the coarser grid resolution $N = 64$ in Fig. 2. The observation that all three error curves almost align with each other (as shown in Fig. 2) confirms that second-order convergence in $\mathbf{u}$ and $\mathbf{X}$ is achieved.

5.2. Area conservation and IB marker spacing

As an immediate consequence of fluid incompressibility, the volume enclosed by the immersed structure should be exactly conserved as it deforms and moves with the fluid. However, it is

observed that even in the simplest scenario of a pressurized membrane in its circular equilibrium configuration [24], the volume error of an IB method with conventional interpolation and spreading systematically grows at a rate proportional to the pressure jump across the elastic interface [19]. In this set of tests, we demonstrate that, the "volume" or area enclosed by a 2D membrane is well-conserved by the DFIB method when the Lagrangian interface is sufficiently resolved.

We follow the same problem setup as in the test described in [24]. A thin elastic membrane $\boldsymbol{X}(s,t)$, initially in a circular equilibrium configuration,

$$\boldsymbol{X}(s,0) = \left(\frac{1}{2} + \frac{1}{4}\cos(s), \frac{1}{2} + \frac{1}{4}\sin(s)\right), \quad s \in [0, 2\pi], \tag{5.6}$$

is immersed in a periodic unit cell $\Omega = [0,1]^3$ with zero initial background flow. The Lagrangian force density on the interface is described by

$$\boldsymbol{F}(s,t) = \kappa \frac{\partial^2 \boldsymbol{X}}{\partial s^2}, \tag{5.7}$$

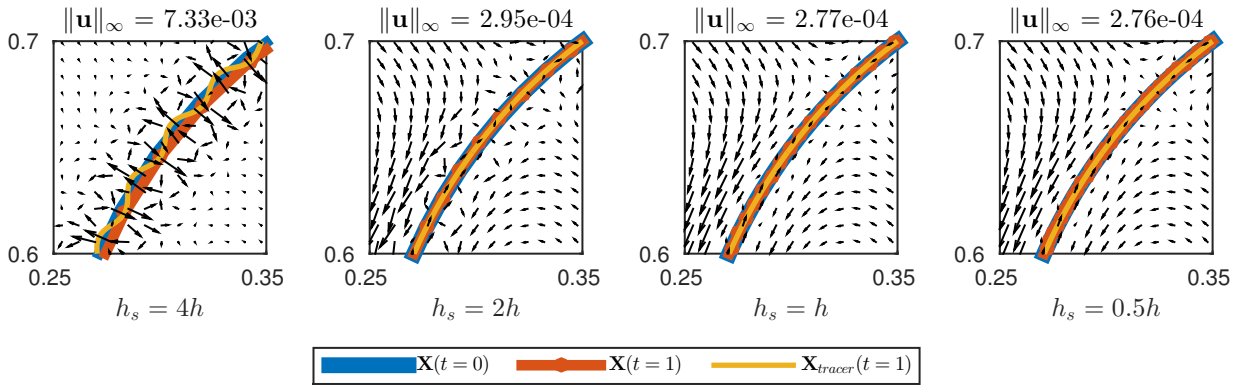
in which κ is the uniform stiffness coefficient. The elastic membrane is discretized by a uniform Lagrangian mesh of M points in the variable s. We approximate the Lagrangian force density by

$$\mathbf{F}_m = \frac{\kappa}{(\Delta s)^2}\left(\mathbf{X}_{m+1} - 2\mathbf{X}_m + \mathbf{X}_{m-1}\right), \tag{5.8}$$

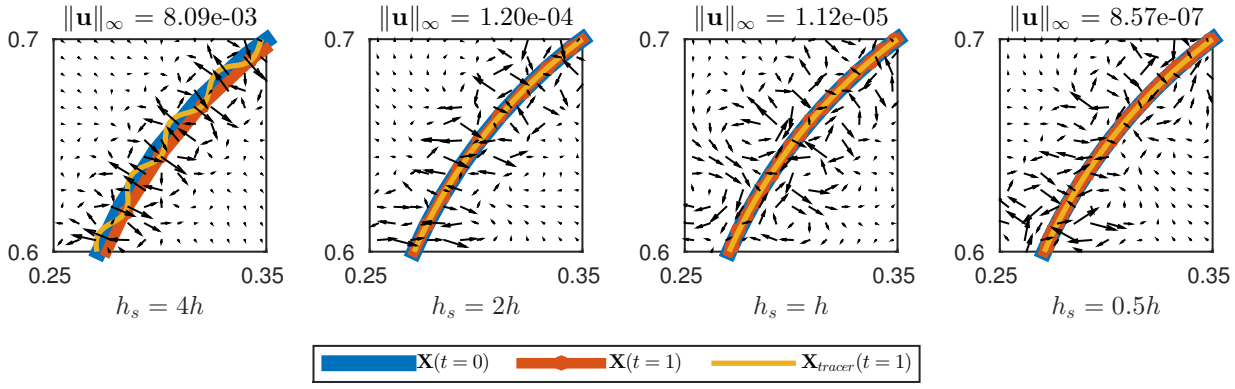
which corresponds to a collection of Lagrangian markers connected by linear springs of zero rest length with stiffness κ. For this problem, since the elastic interface is initialized in the equilibrium configuration with zero background flow, any spurious fluid velocity and area loss incurred in the simulation are regarded as numerical errors.

In our simulations, we set $\rho = 1,\ \mu = 0.1,\ \kappa = 1$. The size of the Eulerian grid is fixed at 128×128 with meshwidth $h = \frac{1}{128}$. The size of the Lagrangian mesh M is chosen so that two adjacent Lagrangian markers are separated by a physical distance of h_s in the equilibrium configuration, that is, $M \approx 2\pi R/h_s$, where R is the radius of the circular membrane. The time-step size is set to be $\Delta t = \frac{h}{4}$ for stability of computation. In all computations, we use the $\mathscr{C}^3$ 6-point IB kernel to form the regularized delta function δ_h.

In Fig. 3 we compare the computational results of DFIB with those of IBMAC for different $h_s = 4h$, $2h$, h and $\frac{h}{2}$ (from the left to right in Fig. 3). Each subplot of Fig. 3 shows a magnified view of the same arc of the circular interface along with its nearby spurious fluid velocity field. The interface represented by the Lagrangian markers $\mathbf{X}(t=1)$ is shown in red and the initial configuration $\mathbf{X}(t=0)$ is shown in the blue curve. In addition, we also include 1024 passive tracers $\mathbf{X}_{\text{tracer}}(t=1)$ that move with the interpolated velocity associated with the corresponding IB method in the yellow curve. In the first column of Fig. 3 in which $h_s = 4h$, we see that the maximum spurious velocity $\|\mathbf{u}\|_\infty$ of IBMAC is of the same magnitude as that of DFIB. At such coarse resolution in the Lagrangian mesh, fluid apparently leaks through the gap between two adjacent markers, as can be observed by the wiggly pattern in the passive tracers. As the the Lagrangian mesh is refined gradually from $h_s = 4h$ to $\frac{h}{2}$ (from left to right in Fig. 3), we see that $\|\mathbf{u}\|_\infty$ decreases from 10^{-3} to 10^{-7} in the DFIB method, whereas $\|\mathbf{u}\|_\infty$ stops improving around 10^{-4} in IBMAC. Moreover, in the columns where $h_s = 2h$, h, $\frac{h}{2}$, we see a clear global pattern in the spurious velocity field in IBMAC, while the spurious velocity field of DFIB appears to be much smaller in magnitude and random in pattern.



(a) IBMAC



(b) DFIB

Fig. 3: A magnified view of the quasi-static circular membrane and its nearby spurious velocity field for different Lagrangian mesh spacing $h_s = 4h$, $2h$, h and $\frac{h}{2}$, as indicated below each figure panel, while keeping $h = \frac{1}{128}$ fixed. The top panel (a) shows the computational results from IBMAC, and the bottom panel (b) shows the results from DFIB. The interface represented by the Lagrangian markers $\mathbf{X}(t = 1)$ is shown in red, the initial configuration $\mathbf{X}(t = 0)$ is shown in blue, and the interface represented by 1024 passive tracers is shown in yellow. The time-step size is set to be $\Delta t = \frac{h}{4}$ for stability. In the above computations, the $\mathscr{C}^3$ 6-point IB kernel ϕ_{6h}^{new} is used in IBMAC and DFIB.

We define the normalized area error with respect to the initial configuration

$$\Delta A(t;\mathbf{X}) := \frac{|A(t;\mathbf{X}) - A(0;\mathbf{X})|}{A(0;\mathbf{X})}, \tag{5.9}$$

where $A(t;\mathbf{X})$ denotes the area of the polygon enclosed by the collection of Lagrangian markers $\{\mathbf{X}_1,\ldots,\mathbf{X}_M\}$ at time t. Fig. 4 and Fig. 5 show normalized area errors defined by Eq. (5.9) for DFIB and IBMAC with different choices of the IB kernels: $\phi_{4h} \in \mathscr{C}^1$, $\phi^{\mathrm{B}}_{4h} \in \mathscr{C}^2$, $\phi^{\mathrm{new}}_{5h} \in \mathscr{C}^3$, $\phi^{\mathrm{new}}_{6h} \in \mathscr{C}^3$ and $\phi^{\mathrm{B}}_{6h} \in \mathscr{C}^4$. For the coarse Lagrangian marker spacings, for example, when $h_s = 2h$, $4h$, the area errors for IBMAC and DFIB have similar orders of magnitude (compare Fig. 4a, 4b to Fig. 5a, 5b). As the Lagrangian marker spacing is reduced from $2h$ to h, we see a decrease in $\Delta A(t;\mathbf{X})$ for IBMAC by approximately a factor of 10 (see Fig. 4b, 4c) for all the IB kernels we consider in this set of tests. In contrast, the area errors for DFIB improve by at least a factor of 10^3 for the IB kernels that are at least $\mathscr{C}^2$ (see Fig. 5b, 5c), and in the best scenario, $\Delta A(t;\mathbf{X})$ for ϕ^{B}_{6h} decreases from 10^{-4} to 10^{-9}. Moreover, as the Lagrangian mesh is refined from h to $\frac{h}{8}$, area errors for DFIB keep improving, even to the machine precision for ϕ^{B}_{6h} at $h_s = \frac{h}{4}, \frac{h}{8}$ and for ϕ^{new}_{6h} at $h_s = \frac{h}{8}$ (see Fig. 5e, 5f). For a moderate Lagrangian marker spacing, such as $h_s = h$ and $\frac{h}{2}$, area errors for DFIB are several orders of magnitude smaller than those of IBMAC. On the other hand, area errors for IBMAC stop improving around 10^{-5} for $h_s \le h$, no matter how densely the Lagrangian mesh is refined (see Fig. 4c to Fig. 4f). We remark that the smoothness of the IB kernel appears to play an important role in volume conservation of DFIB. In this study DFIB achieves the best volume conservation result for $h_s \le h$ with ϕ^{B}_{6h}, and this kernel also has the highest regularity of the kernel functions considered in this work.

5.3. A thin elastic membrane with parametric resonance in 2D

In many biological applications, the immersed structure is an active material, interacting dynamically with the surrounding fluid and generating time-dependent motion. It has been reported that the simulation of active fluid-structure interactions using the conventional IB method may suffer from significant loss in the volume enclosed by structure [19]. A simple prototype problem for active fluid-structure interaction is a thin elastic membrane that dynamically evolves in a fluid in response to elastic forcing with periodic variation in the stiffness parameter [36, 37], that is,

$$\boldsymbol{F}(s,t) = \kappa(t)\frac{\partial^2 \boldsymbol{X}}{\partial s^2}, \tag{5.10}$$

where $\kappa(t)$ is a periodic time-dependent stiffness coefficient of the form

$$\kappa(t) = K_c(1 + 2\tau\sin(\omega_0 t)). \tag{5.11}$$

It is quite remarkable that such a purely temporal parameter variation can result in the emergence of spatial patterns, but that is indeed the case. We assume that the immersed structure is initially in a configuration that has a small-amplitude perturbation from a circle of radius R,

$$\boldsymbol{X}(s,0) = R(1 + \epsilon_0\cos(ps))\ \hat{\boldsymbol{r}}(s), \tag{5.12}$$

where $\hat{\boldsymbol{r}}(s)$ denotes the position vector pointing radially from the origin. For certain choices of parameters, the perturbed mode in the initial configuration may resonate with the driving frequency ω_0 in the periodic forcing, leading to large-amplitude oscillatory motion in the membrane. The stability of the parametric resonance has been studied in the IB framework using

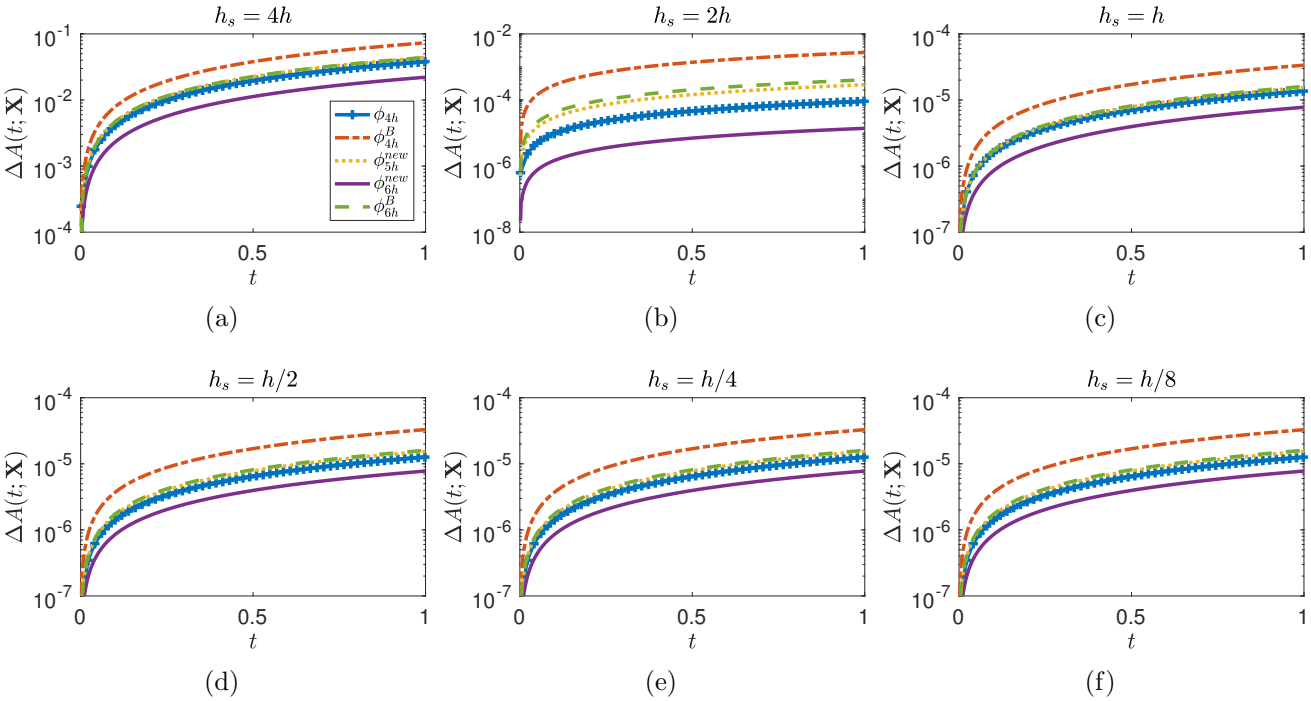


Fig. 4: Normalized area errors of the pressurized circular membrane simulated by IBMAC with the IB kernels: $\phi_{4h} \in \mathscr{C}^1$, $\phi^{\mathrm{B}}_{4h} \in \mathscr{C}^2$, $\phi^{\mathrm{new}}_{5h} \in \mathscr{C}^3$, $\phi^{\mathrm{new}}_{6h} \in \mathscr{C}^3$ and $\phi^{\mathrm{B}}_{6h} \in \mathscr{C}^4$, and with different Lagrangian marker spacings $h_s \in \left\{4h, 2h, h, \frac{h}{2}, \frac{h}{4}, \frac{h}{8}\right\}$ indicated above each figure panel. Note that area errors for IBMAC stop improving around 10^{-5} for $h_s \leq h$, no matter how densely the Lagrangian mesh is refined.

ρ	μ	L	R	K_c	ω_0	p	ϵ_0	τ
1	0.15	5	1	10	10	2	0.05	0.4 (damped oscillation) 0.5 (growing oscillation)

Table 2: Parameters used to simulate the motion of the 2D membrane with parametric resonance.

Floquet linear stability analysis for a thin elastic membrane in 2D [36, 37], and recently for an elastic shell in 3D [38]. Motivated by the linear stability analysis of [36, 37], we consider two sets of parameters listed in Table 2 for our simulations. The first set of parameters with $\tau = 0.4$ leads to a stable configuration in which the membrane undergoes damped oscillations (Fig. 6a), and the second set with $\tau = 0.5$ leads to an unstable configuration in which the membrane oscillates with growing amplitude (Fig. 6b).

The computational domain $\Omega = [0, L]^2$ is discretized by a 128×128 uniform Cartesian grid with meshwidth $h = \frac{L}{128}$. The number of Lagrangian markers is determined so that the distance between the Lagrangian markers is $h_s \approx \frac{h}{2}$ in the initial configuration. The discretization of the Lagrangian force density Eq. (5.10) is constructed in the same way as Eq. (5.8). The time-step size is $\Delta t = \frac{h}{10}$ to ensure the stability of computation. On the left panel of Fig. 6 we show snapshots of the membrane configuration for each case, and on the right panel we plot the time-dependent amplitude $\epsilon(t)$ of the ansatz

$$\boldsymbol{X}(s,t) = R(1 + \epsilon(t)\cos(ps))\ \hat{\boldsymbol{r}}(s) \tag{5.13}$$

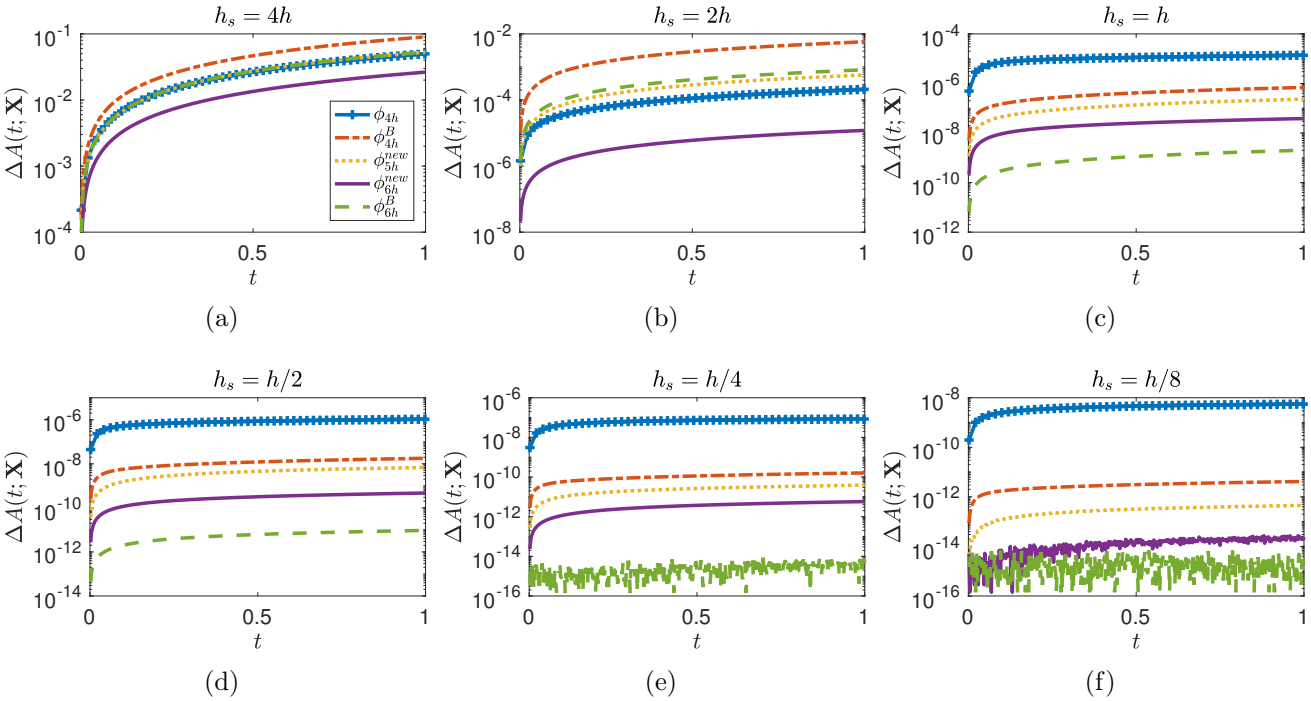
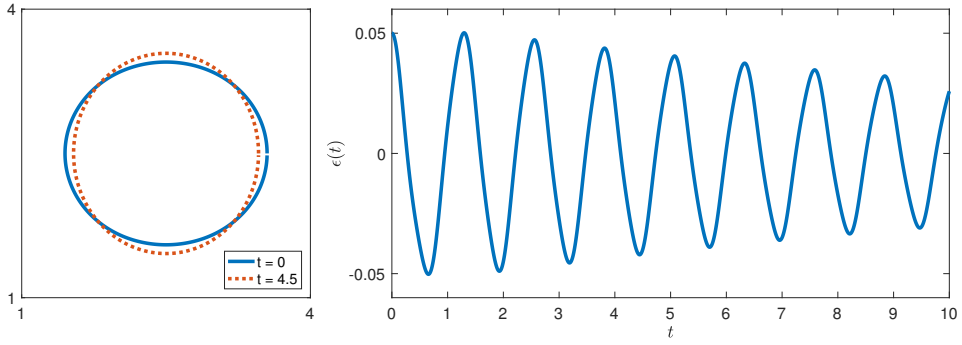


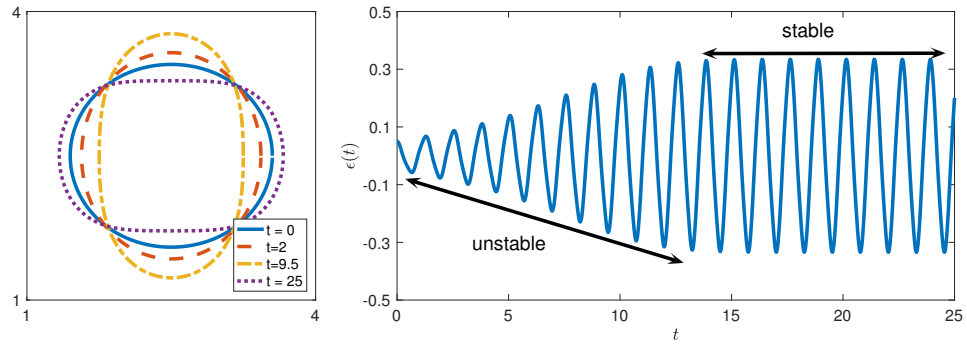
Fig. 5: Normalized area errors of the pressurized circular membrane simulated by DFIB with the IB kernels: $\phi_{4h} \in \mathscr{C}^1$, $\phi^{\mathrm{B}}_{4h} \in \mathscr{C}^2$, $\phi^{\mathrm{new}}_{5h} \in \mathscr{C}^3$, $\phi^{\mathrm{new}}_{6h} \in \mathscr{C}^3$ and $\phi^{\mathrm{B}}_{6h} \in \mathscr{C}^4$, and with different Lagrangian marker spacings, $h_s \in \left\{4h, 2h, h, \frac{h}{2}, \frac{h}{4}, \frac{h}{8}\right\}$ indicated above each figure panel. As the Lagrangian mesh is refined, area errors for DFIB keep improving, even to the machine precision for ϕ^{B}_{6h} at $h_s = \frac{h}{4}, \frac{h}{8}$ and for ϕ^{new}_{6h} at $h_s = \frac{h}{8}$.

by applying the FFT to the Lagrangian marker positions $\mathbf{X}$. In the case of growing oscillation (Fig. 6b), the amplitude of the perturbed mode increases from 0.05 to 0.3 until nonlinearities eventually stabilize the growing mode and the membrane starts to oscillate at a fixed amplitude.

We next give a direct comparison of area conservation of IBModified (with $\phi^{\cos}_{4h}$ [19]), IBMAC and DFIB with the IB kernels: $\phi_{4h} \in \mathscr{C}^1$, $\phi^{\mathrm{B}}_{4h} \in \mathscr{C}^2$, $\phi^{\mathrm{new}}_{5h} \in \mathscr{C}^3$, $\phi^{\mathrm{new}}_{6h} \in \mathscr{C}^3$ and $\phi^{\mathrm{B}}_{6h} \in \mathscr{C}^4$. In Fig. 7 and Fig. 8 we show time-dependent area errors enclosed by the parametric membrane for the damped-oscillation and the growing-amplitude cases respectively. For the damped-oscillation case (Fig. 7), we see that area errors for DFIB are at least two orders of magnitude smaller than those of IBMAC and IBModified for IB kernels that are at least $\mathscr{C}^2$. The volume conservation of IBModified and IBMAC was not directly compared in the previous work [24], but it was anticipated that they are similar. In our comparison, we find that IBModified is only slightly better than IBMAC in volume conservation, yet IBMAC is much simpler to use in practice. In this set of tests, the choice of IB kernel also plays a role in affecting area conservation. In particular, the area errors for DFIB using ϕ^{new}_{6h} and ϕ^{B}_{6h} are smaller than those of ϕ^{B}_{4h} and ϕ^{new}_{5h} by approximately one order of magnitude. Additionally, the error curves of DFIB with ϕ^{new}_{6h} and ϕ^{B}_{6h} remain oscillating below 10^{-7} while apparent growth of error in time is observed with ϕ^{B}_{4h} and ϕ^{new}_{5h}, and in the other IB methods. We select two timestamps $t = 4$ and 10, at which we report the numerical values of $\Delta A(t;\mathbf{X})$ of the three IB methods, as well as the ratio of improvement with respect to IBCollocated with ϕ_{4h} as the benchmark in Table 3. The improvements in area conservation of DFIB is consistently more than 10^4 times over IBCollocated and about 10^3 times over IBMAC. Similar results are obtained for the growing-amplitude case (see Fig. 8 and Table 4) except that the parametrically-unstable membrane has experienced some area loss due to the



(a)



(b)

Fig. 6: Left panel: snapshots of the 2D membrane with parametric resonance. Right panel: the time-dependent amplitude $\epsilon(t)$ of the perturbed mode in Eq. (5.13). (a) Damped oscillation (b) Growing oscillation.

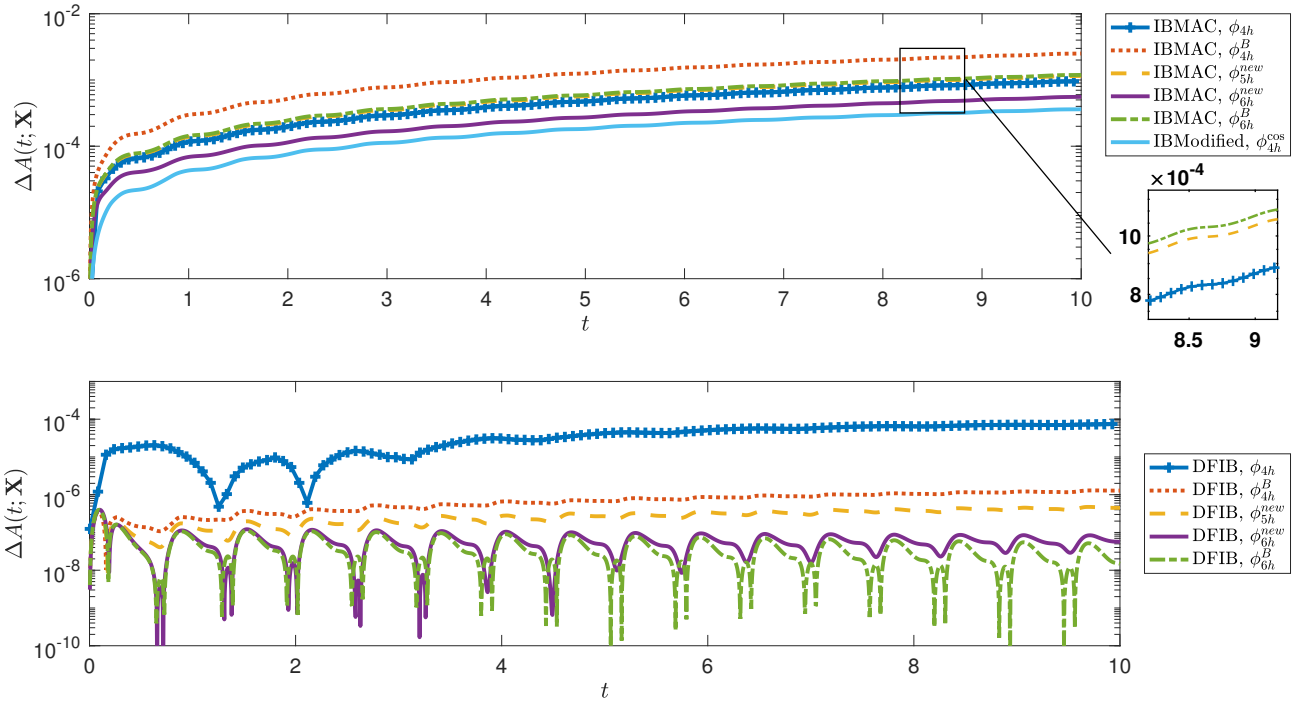


Fig. 7: Normalized area errors $\Delta A(t;\mathbf{X})$ of the 2D parametric membrane undergoing damped oscillatory motion (corresponding to the motion shown in Fig. 6a) are plotted on the semi-log scale. Computations are performed using IBMAC and DFIB with the IB kernels: $\phi_{4h} \in \mathscr{C}^1$, $\phi^{\mathrm{B}}_{4h} \in \mathscr{C}^2$, $\phi^{\mathrm{new}}_{5h} \in \mathscr{C}^3$, $\phi^{\mathrm{new}}_{6h} \in \mathscr{C}^3$ and $\phi^{\mathrm{B}}_{6h} \in \mathscr{C}^4$, and IBModified with $\phi^{\cos}_{4h} \in \mathscr{C}^1$. The top panel shows area errors for IBMAC and IBModified, and the bottom panel shows area errors for DFIB.

growing-amplitude oscillation before its motion is stabilized by the nonlinearities.

5.4. A 3D thin elastic membrane with surface tension

In our final test problem, we examine volume conservation of the DFIB method by extending the surface tension problem to 3D. We consider in 3D a thin elastic membrane that is initially in its spherical equilibrium configuration. The spherical surface of the membrane is discretized by a triangulation consisting of approximately equilateral triangles with edge length approximately equal to h_s, constructed from successive refinement of a regular icosahedron by splitting each facet into four smaller equilateral triangles and projecting the vertices onto the sphere to form the refined mesh (see Fig. 9 for the first two levels of refinement). We use $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_M\}$ and $\{\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_P\}$ to denote the vertices (Lagrangian markers) and the triangular facets of the mesh respectively. The generalization of discrete elastic energy functional of surface tension in 3D is the product of surface tension constant γ (energy per unit area) and the total surface area of the triangular mesh [13], that is,

$$E[\mathbf{X}_1, \ldots \mathbf{X}_M] = \gamma \sum_{p=1}^{P} |\mathbf{T}_p| \,, \tag{5.14}$$

where $|\mathbf{T}_p|$ is the area of the p^{th} triangle. The Lagrangian force $\mathbf{F}_k \Delta\mathbf{s}$ at the k^{th} vertex is minus the partial derivative of $E[\mathbf{X}_1, \ldots, \mathbf{X}_M]$ with respect to $\mathbf{X}_k$,

$$\mathbf{F}_k \Delta\mathbf{s} = -\frac{\partial E}{\partial \mathbf{X}_k} = -\gamma \sum_{l \in \mathrm{nbor}(k)} \frac{\partial |\mathbf{T}_l|}{\partial \mathbf{X}_k} \,, \tag{5.15}$$

Method	**IB Kernel**	$\Delta A(t=4;\mathbf{X})$		$\Delta A(t=10;\mathbf{X})$	
		(damped)	ratio	(damped)	ratio
IBCollocated	ϕ_{4h}	9.965e-03	-	2.394e-02	-
IBMAC	ϕ^{B}_{4h}	1.033e-03	9.6	2.508e-03	9.5
	ϕ_{4h}	3.888e-04	25.6	9.574e-04	25.0
	ϕ^{new}_{5h}	4.673e-04	21.3	1.150e-03	20.08
	ϕ^{new}_{6h}	2.229e-04	44.7	5.566e-04	43.0
	ϕ^{B}_{6h}	4.853e-04	20.5	1.119e-03	21.4
IBModified	ϕ^{cos}_{4h}	1.510e-04	66.0	3.616e-04	66.2
DFIB	ϕ_{4h}	3.076e-05	323.9	7.465e-05	320.7
	ϕ^{B}_{4h}	6.597e-07	1.5e+04	1.285e-06	1.9e+04
	ϕ^{new}_{5h}	2.741e-07	3.6e+04	4.491e-07	5.4e+04
	ϕ^{new}_{6h}	9.401e-08	1.1e+05	5.659e-08	4.2e+05
	ϕ^{B}_{6h}	8.078e-08	1.2e+05	1.484e-08	1.6e+06

Table 3: Normalized area errors $\Delta A(t;\mathbf{X})$ of the 2D parametric membrane undergoing damped oscillatory motion (Fig. 6a) are reported for $t = 4$ and 10. The ratios are computed using the area error for IBCollocated with ϕ_{4h} as the benchmark.

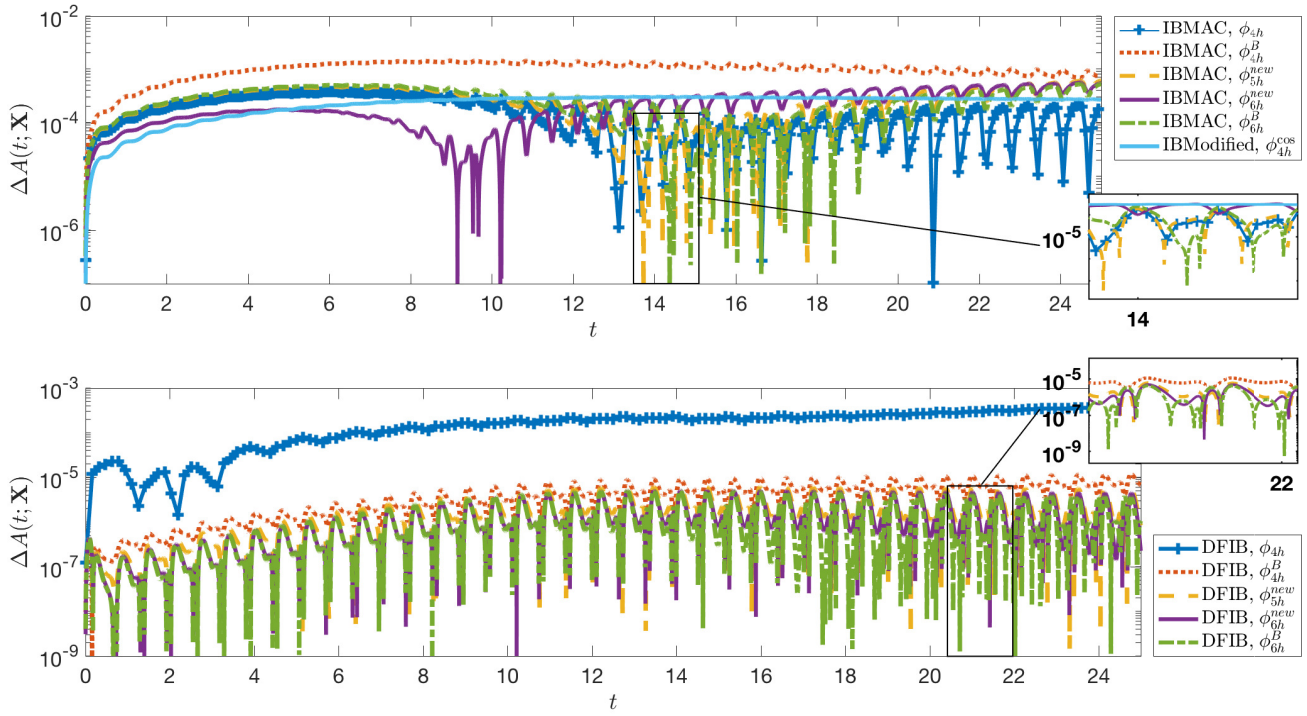


Fig. 8: Normalized area errors $\Delta A(t;\mathbf{X})$ of the 2D parametric membrane undergoing growing-amplitude oscillatory motion (corresponding to the motion shown in Fig. 6b) are plotted on the semi-log scale. Computations are performed using IBMAC and DFIB with the IB kernels: $\phi_{4h} \in \mathscr{C}^1$, $\phi^{\mathrm{B}}_{4h} \in \mathscr{C}^2$, $\phi^{\mathrm{new}}_{5h} \in \mathscr{C}^3$, $\phi^{\mathrm{new}}_{6h} \in \mathscr{C}^3$ and $\phi^{\mathrm{B}}_{6h} \in \mathscr{C}^4$, and IBModified with $\phi^{\mathrm{cos}}_{4h} \in \mathscr{C}^1$. The top panel shows area errors for IBMAC and IBModified, and the bottom panel shows area errors for DFIB.

Method	**IB Kernel**	$\Delta V(t=2;\mathbf{X})$ (growing)	ratio	$\Delta V(t=25;\mathbf{X})$ (steady)	ratio
IBCollocated	ϕ_{4h}	4.935e-03	-	1.842e-02	-
IBMAC	ϕ^{B}_{4h}	5.083e-04	9.7	6.858e-04	26.9
	ϕ_{4h}	1.894e-04	26.0	1.930e-04	92.4
	ϕ^{new}_{5h}	2.263e-04	21.8	5.976e-04	30.8
	ϕ^{new}_{6h}	1.057e-04	32.7	5.451e-04	33.8
	ϕ^{B}_{6h}	2.354e-04	21.1	5.248e-04	35.1
IBModified	ϕ^{cos}_{4h}	7.449e-05	66.3	2.673e-04	68.9
DFIB	ϕ_{4h}	6.227e-06	792.5	3.858e-04	47.7
	ϕ^{B}_{4h}	3.005e-07	1.6e+04	6.834e-06	2.7e+03
	ϕ^{new}_{5h}	4.169e-08	1.2e+05	8.368e-07	2.2e+04
	ϕ^{new}_{6h}	6.203e-08	7.9e+04	2.267e-07	8.1e+04
	ϕ^{B}_{6h}	6.963e-08	7.1e+04	1.073e-06	1.72+04

Table 4: Normalized area errors $\Delta A(t;\mathbf{X})$ of the 2D parametric membrane undergoing growing-amplitude oscillatory motion (Fig. 6a) are reported for $t = 2$ and 25. The ratios are computed using the area error for IBCollocated with ϕ_{4h} as the benchmark.

where nbor(k) denotes the set of indices of triangles that share $\mathbf{X}_k$ as a vertex[2]. Each component of the term $\partial|\mathbf{T}_l|/\partial\mathbf{X}_k$ in Eq. (5.15) can be computed analytically [13],

$$\left(\frac{\partial|\mathbf{T}_l|}{\partial\mathbf{X}_k}\right)_\alpha = \frac{\partial}{\partial\mathbf{X}_{k,\alpha}}\left(\frac{1}{2}\left|(\mathbf{X}_k-\mathbf{X}'_k)\times(\mathbf{X}'_k-\mathbf{X}''_k)\right|\right)$$
$$= \frac{1}{2}\left((\mathbf{X}'_k-\mathbf{X}''_k)\times\hat{\mathbf{n}}_l\right)_\alpha, \quad \alpha=1,2,3, \tag{5.16}$$

where $\mathbf{X}_k,\ \mathbf{X}'_k,\ \mathbf{X}''_k$ denote the three vertices of the triangle $\mathbf{T}_l$ ordered in the counterclockwise direction and $\hat{\mathbf{n}}$ is the unit outward normal vector of $\mathbf{T}_l$.

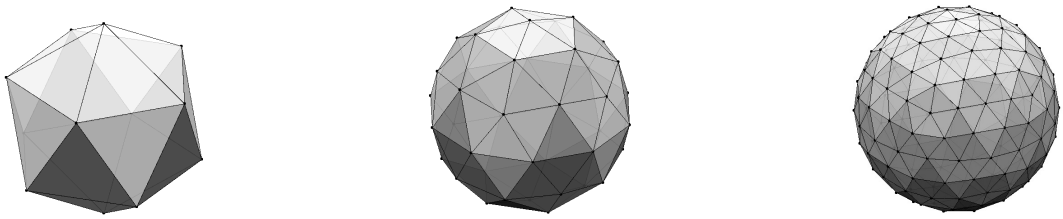
The computation is performed in the periodic box $\Omega=[0,1]^3$ with Eulerian meshwidth $h=\frac{1}{128}$ using DFIB with ϕ^{new}_{6h}. For the quasi-static test, the initial fluid velocity is set to be zero, and for the dynamic test, we set $\boldsymbol{u}(\boldsymbol{x},0)=(0,\ \sin(4\pi x),\ 0)$. In the computational results shown in Fig. 10, the spherical membrane is discretized by triangulation (as shown in Fig. 9) with 5 successive levels of refinement from the regular icosahedron (Fig. 9a), which results in a triangular mesh with $M=10242$ vertices and $P=20480$ facets. The radius of the spherical membrane is set to be $R\approx 0.1$ which corresponds to $h_s\approx\frac{h}{2}$. The remaining parameters in the computation are $\rho=1,\ \mu=0.05,\ \gamma=1$ and the time-step size $\Delta t=\frac{h}{4}$. In Fig. 10 we show snapshots of the 3D elastic membrane at $t=0,\ \frac{1}{32},\ \frac{1}{4}$ and $\frac{1}{2}$ for the dynamic case. The elastic interface is instantaneously deformed by the fluid flow in the y-direction, and due to surface tension, the membrane eventually relaxes back to the spherical equilibrium configuration. Colored markers that move passively with the divergence-free interpolated fluid velocity are added for visualizing the fluid flow in the vicinity of the interface.

The volume enclosed by the triangular surface mesh is approximated by the total volume of tetrahedra formed by each facet and one common reference point (e.g. the origin) using the scalar triple product. To study volume conservation of the DFIB method in 3D, we compare the normalized volume error defined by

$$\Delta V(t;\mathbf{X}) := \frac{|\operatorname{Vol}(t;\mathbf{X})-\operatorname{Vol}(0;\mathbf{X})|}{\operatorname{Vol}(0;\mathbf{X})} \tag{5.17}$$

using IBMAC and DFIB with $h_s=h,\frac{h}{2},\frac{h}{4}$, which correspond to triangular meshes with 4,5,6 levels of refinement from the regular icosahedron respectively. For the quasi-static case (Fig. 11a), volume errors for DFIB are at least 2 orders of magnitude smaller than those of IBMAC. Further, volume errors for DFIB keep decreasing as the Lagrangian mesh is refined from $h_s=h$ to $\frac{h}{4}$. For the dynamic case (Fig. 11b), both methods suffer a significant amount of volume loss arising from the rapid deformation at the beginning of simulation. The volume error for DFIB with $h_s=h$ is similar to those of IBMAC in magnitude and they all grow consistently in time. However, for $h_s=\frac{h}{4},\frac{h}{2}$, the volume errors for DFIB remain steady in time once the membrane comes to rest. It appears that the behavior of volume error changes in nature from $h_s=h$ to $\frac{h}{2}$, which coincides with the conventional recommendation that the best choice of Lagrangian mesh spacing in the IB method is $h_s=\frac{h}{2}$ in practice. Finally, we remark that the improvement in volume conservation does not seem to be as substantial as the improvement in area conservation in 2D. We suspect

[2]Here $\Delta\mathbf{s}$ is the Lagrangian area associated with each node and $\mathbf{F}_k$ is the Lagrangian force density with respect to Lagrangian area, but note that we do not need $\mathbf{F}_k$ and $\Delta\mathbf{s}$ separately; only their product is used in the numerical scheme.



(a) (b) (c)

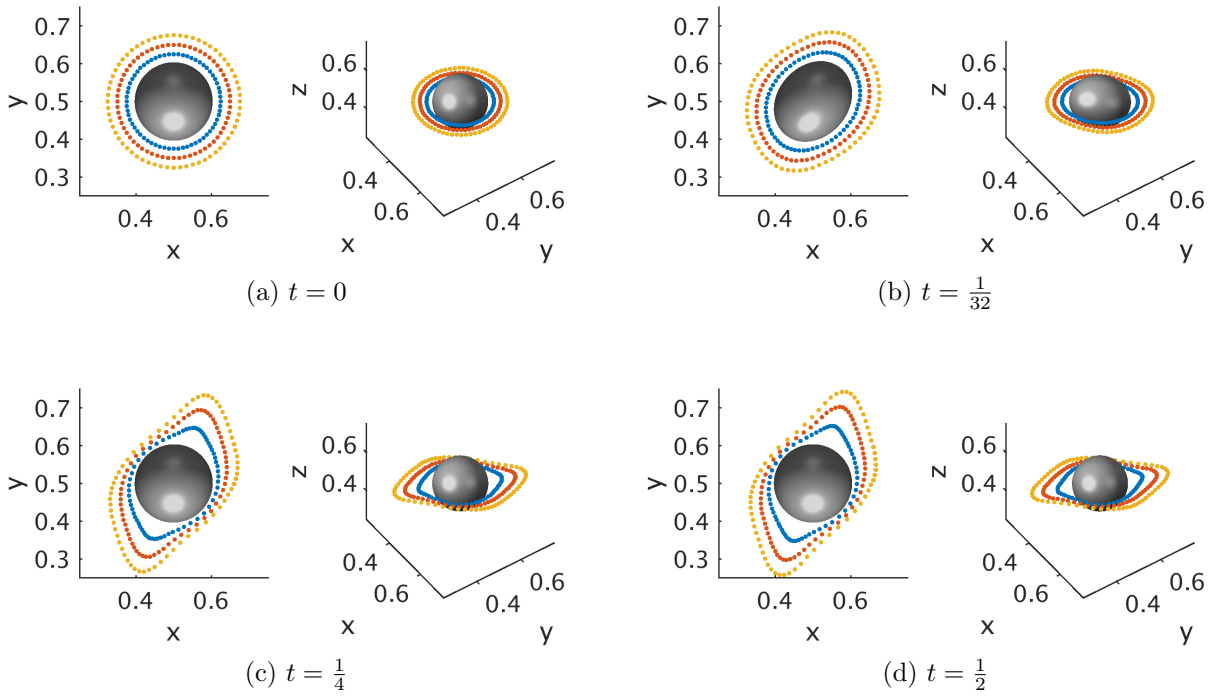
Fig. 9: Triangulation of a spherical surface mesh via refinement of a regular icosahedron. (a) Regular icosahedron (b) Refined mesh after one level of refinement (c) Refined mesh after two levels of refinement.

that this may be attributed to the larger approximation error in computing the volume using the tetrahedral approximation (after the triangular mesh is deformed), whereas we have re-parametrized the computed interface using cubic splines when approximating the area in 2D. Nevertheless, the reduction in volume error from the Lagrangian mesh-refinement experiments indeed confirms that the DFIB method can generally achieve better volume conservation if the immersed boundary is sufficiently resolved ($h_s \leq \frac{h}{2}$).

6. Conclusions

In this paper, we introduce an IB method with divergence-free velocity interpolation and force spreading. Our IB method makes use of staggered-grid discretization to define an edge-centered discrete vector potential. By interpolating the discrete vector potential in the conventional IB fashion, we obtain a continuum vector potential whose curl directly yields a continuum Lagrangian velocity field that is exactly divergence-free by default. The corresponding force-spreading operator is constructed to be the adjoint of velocity interpolation so that energy is preserved in the interaction between the fluid and the immersed boundary. Both the new interpolation and spreading schemes require solutions of discrete vector Poisson equations which can be efficiently solved by a variety of algorithms. The transfer of information from the Eulerian grid to the Lagrangian mesh (and vice versa) is performed using $\nabla\delta_h$ on the edge-centered staggered grid $\mathbb{E}$. We have found that volume conservation of DFIB improves with the smoothness of the IB kernel used to construct δ_h, and we have numerically tested that IB kernels that are at least $\mathscr{C}^2$ are good candidate kernels that can be used to construct the regularized delta function in the DFIB method.

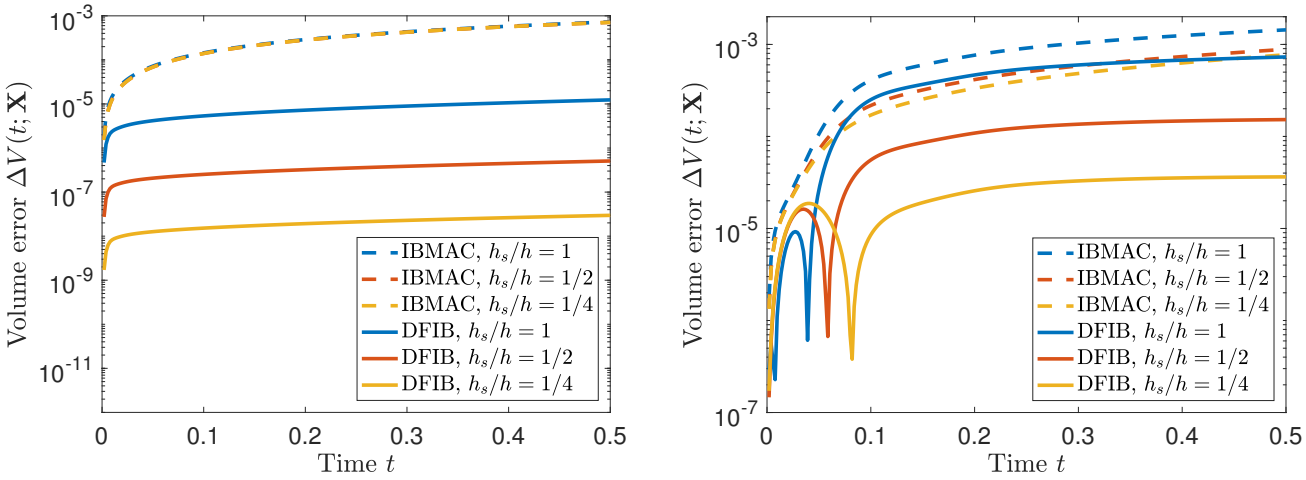
We have incorporated the divergence-free interpolation and spreading operators in a second-order time-stepping scheme, and applied it to several benchmark problems in two and three spatial dimensions . First, we have tested that our method achieves second-order convergence in both the fluid velocity and the Lagrangian deformation map for the 2D surface tension problem, which is admittedly a special case, since its continuum solution has a continuous normal derivative of the tangential velocity across the immersed boundary. The highlight of the DFIB is its capability of substantially reducing volume error in the immersed structure as it moves and deforms in the process of fluid-structure interaction. Through numerical simulations of quasi-static and dynamic membranes, we have confirmed that the DFIB method improves volume conserva-



(a) $t = 0$ (b) $t = \frac{1}{32}$

(c) $t = \frac{1}{4}$ (d) $t = \frac{1}{2}$

Fig. 10: Deformation of a 3D elastic membrane immersed in a viscous fluid with initial velocity $\boldsymbol{u}(\boldsymbol{x}, t) = (0,\ \sin(4\pi x),\ 0)$ at $t = 0, \frac{1}{32}, \frac{1}{4}$ and $\frac{1}{2}$. The computation is performed using DFIB with ϕ_{6h}^{new} in the periodic box $\Omega = [0,1]^3$ with Eulerian meshwidth $h = \frac{1}{128}$. The elastic membrane, initially in spherical configuration with radius $R \approx 0.1$, is discretized by a triangular surface mesh with $M = 10242$ vertices and $P = 20480$ facets so that $h_s = \frac{h}{2}$ in the initial configuration. Colored markers that move passively with the divergence-free interpolated fluid velocity are added for visualizing the fluid flow in the vicinity of the membrane interface.



(a) Quasi-static test (b) Dynamic test

Fig. 11: Normalized volume error $\Delta V(t; \mathbf{X})$ of a 3D elastic membrane using IBMAC and DFIB with $h_s = h, \frac{h}{2}, \frac{h}{4}$, where $h = \frac{1}{128}$. For (a) the quasi-static test, $\Delta V(t, \mathbf{X})$ of DFIB decreases with mesh refinement, while there is no improvement in volume error for IBMAC. For (b) the dynamic test, the volume error in DFIB remains (almost) steady in time for $h_s = \frac{h}{2}, \frac{h}{4}$ as the membrane rests, whereas we see no substantial improvement in volume conservation with mesh refinement for IBMAC, and the volume loss keeps increasing in time. For this set of computations, the $\mathscr{C}^3$ 6-point kernel ϕ_{6h}^{new} is used.

tion by several orders of magnitude compared to IBMAC and IBModified. Furthermore, owing to the divergence-free nature of its velocity interpolation, the DFIB method reduces volume error with Lagrangian mesh refinement while keeping the Eulerian grid fixed. A similar refinement study would not yield improved volume conservation when using the conventional IB method.

Unlike other improved IB methods that either make use of non-standard finite-difference operators (IBModified [19]) that complicate the implementation of the fluid solver, or rely on analytically-computed correction terms (IIM [20, 21] or Blob-Projection method [31]) which may not be readily accessible in many applications, the DFIB method is generally applicable, and it is straightforward to implement in both 2D and 3D from an existing IB code that is based on the staggered-grid discretization. Moreover, the additional costs of performing the new interpolation and spreading do not increase the overall complexity of computation and are modest compared to the existing IB methods.

We point out two limitations of our present work. First, the current version of DFIB method is based on the assumption of periodic boundary conditions. Extending the method to include physical boundary conditions at the boundaries of the computational domain is one possible direction of future work. Second, we note that the pressure gradient generated by the Lagrangian forces is part of the resulting Eulerian force density in the DFIB method because force spreading is also constructed to be discretely divergence-free. We do not yet see an obvious way to extract the pressure from the Eulerian force density in case it is needed for output purposes. However, this may also be an important advantage of our method from the standpoint of accuracy, since it means that jumps in pressure across the interface do not require any explicit representation.

Acknowledgement

Y. Bao and A. Donev were supported in part by the National Science Foundation under award DMS-1418706, and by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0008271. B.E. Griffith acknowledges research support from the National Science Foundation (NSF awards ACI 1450327, DMS 1410873, and CBET 1511427) and the National Institutes of Health (NIH award HL117063).

Appendix A. Vector identities of discrete differential operators

Suppose $\varphi(\mathbf{x})$ is a scalar grid function defined on $\mathbb{C}$, and $\mathbf{u}(\mathbf{x})$ and $\mathbf{a}(\mathbf{x})$ are vector grid functions defined on $\mathbb{F}$ and $\mathbb{E}$ respectively. The following discrete vector identities are valid on the periodic staggered grid just as in the continuum case,

$$\mathbf{D}^h \times \mathbf{G}^h \varphi = 0, \tag{A.1}$$

$$\mathbf{D}^h \cdot (\mathbf{D}^h \times \mathbf{u}) = 0, \tag{A.2}$$

$$\mathbf{D}^h \times (\mathbf{D}^h \times \mathbf{u}) = \mathbf{G}^h (\mathbf{D}^h \cdot \mathbf{u}) - \mathbf{L}^h \mathbf{u}, \tag{A.3}$$

$$\sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x}) \cdot (\mathbf{G}^h \varphi)(\mathbf{x}) h^3 = -\sum_{\mathbf{x}\in\mathbb{C}} (\mathbf{D}^h \cdot \mathbf{u})(\mathbf{x})\, \varphi(\mathbf{x}) h^3, \tag{A.4}$$

$$\sum_{\mathbf{x}\in\mathbb{E}} \mathbf{a}(\mathbf{x}) \cdot (\mathbf{D}^h \times \mathbf{u})(\mathbf{x}) h^3 = \sum_{\mathbf{x}\in\mathbb{F}} (\mathbf{D}^h \times \mathbf{a})(\mathbf{x}) \cdot \mathbf{u}(\mathbf{x}) h^3. \tag{A.5}$$

Eqs. (A.1) and (A.3) are merely discrete analogues of well-known vector identities involving gradient, divergence and curl. These identities can be proved in the same manner as their

continuous counterparts. Eqs. (A.4) and (A.5) can be verified via "summation by parts". Note that Eqs. (A.2) and (A.3) also hold if we replace $\mathbf{u}$ (which lives on $\mathbb{F}$) by $\mathbf{a}$ (which lives on $\mathbb{E}$).

Appendix B. Existence of discrete vector potential

Lemma 1. *Suppose* $\mathbf{D}^h \cdot \mathbf{u} = 0$ *and* $\mathbf{D}^h \times \mathbf{u} = 0$ *for* $\mathbf{x} \in \mathbb{F}$*, then* $\mathbf{u}(\mathbf{x})$ *is a constant function on* $\mathbb{F}$*.*

Proof. To prove this statement, we use Eqs. (A.3) to (A.5),

$$\begin{aligned}\sum_{\mathbf{x}\in\mathbb{E}}(\mathbf{D}^h \times \mathbf{u})(\mathbf{x}) \cdot (\mathbf{D}^h \times \mathbf{u})(\mathbf{x})h^3 &= \sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x}) \cdot (\mathbf{D}^h \times (\mathbf{D}^h \times \mathbf{u}))h^3 \\ &= \sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x}) \cdot \mathbf{G}^h(\mathbf{D}^h \cdot \mathbf{u})h^3 - \sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x}) \cdot (\mathbf{L}^h\mathbf{u})h^3 \\ &= -\sum_{\mathbf{x}\in\mathbb{C}}(\mathbf{D}^h \cdot \mathbf{u})^2h^3 + \sum_{\substack{\mathbf{x}\in\mathbb{E},i\neq j\\ \mathbf{x}\in\mathbb{C},i=j}} \left(D_j^h u_i\right)^2 h^3.\end{aligned}$$

Thus,

$$\sum_{\substack{\mathbf{x}\in\mathbb{E},i\neq j\\ \mathbf{x}\in\mathbb{C},i=j}} \left(D_j^h u_i\right)^2 h^3 = \sum_{\mathbf{x}\in\mathbb{E}} \left|(\mathbf{D}^h \times \mathbf{u})\right|^2 h^3 + \sum_{\mathbf{x}\in\mathbb{E}}(\mathbf{D}^h \cdot \mathbf{u})^2h^3. \tag{B.1}$$

Since $\mathbf{D}^h \cdot \mathbf{u} = 0$ and $\mathbf{D}^h \times \mathbf{u} = 0$ by hypothesis, the left-hand side of Eq. (B.1) is also zero. But this implies u_i is constant for $i = 1, 2, 3$.

Lemma 2. *If* ψ *is a scalar grid function that lives on one of the staggered grids, such that*

$$\sum_{\mathbf{x}} \psi(\mathbf{x})h^3 = 0, \tag{B.2}$$

then there exists a grid function φ *such that*

$$L^h\varphi = \psi. \tag{B.3}$$

Proof. This lemma states the solvability of the discrete Poisson problem Eq. (B.3). Since $L^h = \mathbf{D}^h \cdot \mathbf{G}^h$ is symmetric with respect to the inner product on the periodic grid

$$(\varphi, \psi) = \sum_{\mathbf{x}} \varphi(\mathbf{x})\psi(\mathbf{x})h^3, \tag{B.4}$$

what we have to show is that any ψ satisfying Eq. (B.2) is orthogonal to any φ_0 in the null space of $\mathbf{L}^h$. But the null space of $\mathbf{L}^h$ with periodic boundary conditions contains only the constant function, and hence $(\psi, \varphi_0) = 0$ because of Eq. (B.2) as required.

Now we are ready to state the theorem that guarantees the existence of a discrete vector potential $\mathbf{a}(\mathbf{x})$ for $\mathbf{x} \in \mathbb{E}$ given a discretely divergence-free velocity field $\mathbf{u}(\mathbf{x})$ for $\mathbf{x} \in \mathbb{F}$.

Theorem 3. *Suppose* $\mathbf{u}(\mathbf{x})$ *is a periodic grid function for* $\mathbf{x} \in \mathbb{F}$*, and* $\mathbf{u}(\mathbf{x})$ *satisfies*

$$\sum_{\mathbf{x}\in\mathbb{F}} \mathbf{u}(\mathbf{x}) h^3 = 0 \quad \textit{and} \quad \mathbf{D}^h \cdot \mathbf{u} = 0, \tag{B.5}$$

then there exists a grid function $\mathbf{a}(\mathbf{x})$ *for* $\mathbf{x} \in \mathbb{E}$ *such that*

$$\mathbf{u} = \mathbf{D}^h \times \mathbf{a}. \tag{B.6}$$

Proof. We choose $\mathbf{a}(\mathbf{x})$ to be any solution of

$$-\mathbf{L}^h \mathbf{a} = \mathbf{D}^h \times \mathbf{u}. \tag{B.7}$$

Such an $\mathbf{a}(\mathbf{x})$ exists by Lemma 2, because

$$\begin{aligned}\sum_{\mathbf{x}\in\mathbb{E}} \left(\mathbf{D}^h \times \mathbf{u}\right)_i (\mathbf{x}) &= \epsilon_{ijk} \sum_{\mathbf{x}\in\mathbb{E}} 1 \cdot D_j u_k \, h^3 \\ &= -\epsilon_{ijk} \sum_{\mathbf{x}\in\mathbb{F}} (D_j 1) u_k \, h^3 \\ &= 0.\end{aligned}$$

By applying $\mathbf{D}^h\cdot$ to Eq. (B.7) and using the property that $\mathbf{L}^h$ and $\mathbf{D}^h\cdot$ commute, we also have

$$-\mathbf{L}^h(\mathbf{D}^h \cdot \mathbf{a}) = \mathbf{D}^h \cdot (\mathbf{D}^h \times \mathbf{u}) = 0. \tag{B.8}$$

Because the null space of $\mathbf{L}^h$ contains only the constant function, it follows that

$$\mathbf{G}^h(\mathbf{D}^h \cdot \mathbf{a}) = 0. \tag{B.9}$$

If we use Eq. (B.9) and Eq. (A.3) for $\mathbf{a}$, we can rewrite Eq. (B.7) as

$$\mathbf{D}^h \times (\mathbf{D}^h \times \mathbf{a}) = \mathbf{D}^h \times \mathbf{u}, \tag{B.10}$$

or

$$\mathbf{D}^h \times (\mathbf{D}^h \times \mathbf{a} - \mathbf{u}) = 0. \tag{B.11}$$

But we also know from Eq. (A.2) and the requirement that $\mathbf{D}^h \cdot \mathbf{u} = 0$ that

$$\mathbf{D}^h \cdot (\mathbf{D}^h \times \mathbf{a} - \mathbf{u}) = 0. \tag{B.12}$$

From Eqs. (B.11) and (B.12) and Lemma 1, it follows that

$$\mathbf{D}^h \times \mathbf{a} - \mathbf{u} = \text{constant}. \tag{B.13}$$

The constant must be zero, however, since $\mathbf{D}^h \times \mathbf{a}$ has zero sum by "summation by parts", and $\mathbf{u}$ has zero sum by assumption. This completes the proof of the existence of a vector potential satisfying $\mathbf{u} = \mathbf{D}^h \times \mathbf{a}$.

References

[1] C. S. Peskin, The immersed boundary method, Acta Numerica 11 (January 2002) (2003) 479–517.

[2] C. S. Peskin, Flow patterns around heart valves: A numerical method, Journal of Computational Physics 10 (2) (1972) 252–271.

[3] C. S. Peskin, Numerical analysis of blood flow in the heart, Journal of Computational Physics 25 (3) (1977) 220–252.

[4] D. McQueen, C. Peskin, Shared-Memory Parallel Vector Implementation of the Immersed Boundary Method for the Computation of Blood Flow in the Beating Mammalian Heart, The Journal of Supercomputing 11 (3) (1997) 213–236.

[5] B. Griffith, R. Hornung, D. McQueen, C. Peskin, An adaptive, formally second order accurate version of the immersed boundary method, J. Comput. Phys. 223 (1) (2007) 10–49, software available at `https://github.com/ibamr/ibamr`.

[6] B. Griffith, X. Luo, D. McQueen, C. Peskin, Simulating the fluid dynamics of natural and prosthetic heart valves using the immersed boundary method, International Journal of Applied Mechanics 1 (01) (2009) 137–177.

[7] B. E. Griffith, Immersed boundary model of aortic heart valve dynamics with physiological driving and loading conditions, Int J Numer Meth Biomed Eng 28 (2012) 317–345.

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

[9] E. Lushi, C. S. Peskin, Modeling and simulation of active suspensions containing large numbers of interacting micro-swimmers, Computers & Structures 122 (2013) 239–248.

[10] T. G. Fai, B. E. Griffith, Y. Mori, C. S. Peskin, Immersed Boundary Method for Variable Viscosity and Variable Density Problems Using Fast Constant-Coefficient Linear Solvers I: Numerical Method and Results, SIAM Journal on Scientific Computing 35 (5) (2013) B1132–B1161.

[11] T. G. Fai, Fluid Mechanics of the Red Blood Cell and its Cytoskeleton by an Immersed Boundary Method with Nonuniform Viscosity and Density (2014).

[12] Y. Kim, M.-C. Lai, C. S. Peskin, Numerical simulations of two-dimensional foam by the immersed boundary method, Journal of Computational Physics 229 (13) (2010) 5194–5207.

[13] Y. Kim, M.-C. Lai, C. S. Peskin, Y. Seol, Numerical simulations of three-dimensional foam by the immersed boundary method, Journal of Computational Physics 269 (2014) 1–21.

[14] B. Kallemov, A. P. S. Bhalla, B. E. Griffith, A. Donev, An immersed boundary method for rigid bodies, Communications in Applied Mathematics and Computational Science 11 (1) (2016) 79–141, software available at `https://github.com/stochasticHydroTools/RigidBodyIB`.

[15] F. B. Usabiaga, B. Kallemov, B. Delmotte, A. P. S. Bhalla, B. E. Griffith, A. Donev, Hydrodynamics of Suspensions of Passive and Active Rigid Particles: A Rigid Multiblob Approach, submitted to CAMCoS, ArXiv:1602.02170. Software available at `https://github.com/stochasticHydroTools/RotationalDiffusion` (2016).

[16] Y. Bao, J. Kaye, C. S. Peskin, A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance, Journal of Computational Physics 316 (2016) 139–144.

[17] M.-C. Lai, C. S. Peskin, An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity, Journal of Computational Physics 160 (2) (2000) 705–719.

[18] B. E. Griffith, C. S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems, Journal of Computational Physics 208 (1) (2005) 75–105.

[19] C. S. Peskin, B. F. Printz, Improved Volume Conservation in the Computation of Flows with Immersed Elastic Boundaries, Journal of Computational Physics 105 (1) (1993) 33–46.

[20] Z. Li, M.-C. Lai, The Immersed Interface Method for the NavierStokes Equations with Singular Forces, Journal of Computational Physics 171 (2) (2001) 822–842.

[21] L. Lee, R. J. LeVeque, An Immersed Interface Method for Incompressible Navier–Stokes Equations, SIAM Journal on Scientific Computing 25 (3) (2003) 832–856.

[22] D. B. Stein, R. D. Guy, B. Thomases, Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains`arXiv:1609.03851`.

[23] 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, Journal of Computational Physics 304 (2016) 252–274.

[24] B. E. Griffith, On the Volume Conservation of the Immersed Boundary Method, Communications in Computational Physics 12 (2) (2012) 401–432.

[25] A. Dutt, V. Rokhlin, Fast Fourier Transforms for Nonequispaced Data, SIAM Journal on Scientific Computing 14 (6) (1993) 1368–1393.

[26] L. Greengard, J. Lee, Accelerating the Nonuniform Fast Fourier Transform, SIAM Review 46 (3) (2004) 443–454.

[27] W. Strychalski, R. D. Guy, Intracellular Pressure Dynamics in Blebbing Cells, Biophysical Journal 110 (5) (2016) 1168–1179.

[28] D. Devendran, C. S. Peskin, An immersed boundary energy-based method for incompressible viscoelasticity, Journal of Computational Physics 231 (14) (2012) 4613–4642.

[29] A. M. Roma, C. S. Peskin, M. J. Berger, An Adaptive Version of the Immersed Boundary Method, Journal of Computational Physics 153 (1999) 509–534.

[30] B. Griffith, An accurate and efficient method for the incompressible Navier-Stokes equations using the projection method as a preconditioner, J. Comp. Phys. 228 (20) (2009) 7565–7595.

[31] R. Cortez, M. Minion, The Blob Projection Method for Immersed Boundary Problems, Journal of Computational Physics 161 (2) (2000) 428–453.

[32] F. B. Usabiaga, J. B. Bell, R. Delgado-Buscalioni, A. Donev, T. G. Fai, B. E. Griffith, C. S. Peskin, Staggered Schemes for Fluctuating Hydrodynamics, SIAM J. Multiscale Modeling and Simulation 10 (4) (2012) 1369–1408.

[33] M. Unser, Splines: a perfect fit for signal and image processing, IEEE Signal Processing Magazine 16 (6) (1999) 22–38.

[34] M. Unser, A. Aldroubi, M. Eden, On the asymptotic convergence of B-spline wavelets to Gabor functions, IEEE Transactions on Information Theory 38 (2) (1992) 864–872.

[35] M.-c. Lai, Z. Li, A remark on jump conditions for the three-dimensional Navier-Stokes equations involving an immersed moving membrane, Applied Mathematics Letters 14 (2) (2001) 149–154.

[36] R. Cortez, C. S. Peskin, J. M. Stockie, D. Varela, Parametric Resonance in Immersed Elastic Boundaries, SIAM Journal on Applied Mathematics 65 (2) (2004) 494–520.

[37] W. Ko, J. M. Stockie, Correction to "Parametric Resonance in Immersed Elastic Boundaries" `arXiv:1207.4744`.

[38] W. Ko, J. M. Stockie, Parametric Resonance in Spherical Immersed Elastic Shells, SIAM Journal on Applied Mathematics 76 (1) (2016) 58–86.