A Comparison of Two Adaptive Versions of the Immersed Boundary Method

Boyce E. Griffith

Leon H. Charney Division of Cardiology, Department of Medicine, New York University School of Medicine, 550 First Avenue, New York, New York 10016

Abstract

The immersed boundary (IB) method is an approach to problems of fluid-structure interaction in which an elastic structure is immersed in a viscous incompressible fluid. In the continuous setting, the IB method couples an Eulerian description of the fluid to a Lagrangian description of the elastic structure via integral equations with Dirac delta function kernels, and in the discrete case, the delta function is approximated by a regularized version of the delta function. To improve the efficiency of the IB method, at least two different adaptive versions of the conventional IB methodology have been developed, one based on a staggered-grid discretization of the incompressible Navier-Stokes equations [1, 2] and the other based on a cell-centered discretization [3, 4], but we are aware of no direct comparisons between these two adaptive numerical methods. In this work, we present numerical experiments which allow us to compare directly the performance of adaptive cell-centered and staggered-grid IB methods. Our results indicate that the staggered-grid discretization generally yields reduced grid dependence, especially when the scheme is equipped with versions of the regularized delta function which satisfy a so-called even-odd condition. In our numerical experiments, we also find that the staggered-grid scheme is dramatically more accurate than the cell-centered discretization in terms of volume conservation.

Key words: immersed boundary method, incompressible Navier-Stokes equations, adaptive mesh refinement (AMR), projection method, cell centered, staggered grid, marker and cell (MAC), accuracy, grid dependence, volume conservation

Mathematical Reviews Index Classifications: 65M06, 65M50, 74F10, 76D05

Email address: boyce.griffith@nyumc.org (Boyce E. Griffith).
1 Introduction

The immersed boundary (IB) method [5] is a mathematical and computational approach to problems of fluid-structure interaction in which an elastic structure is immersed in a viscous incompressible fluid. Although this methodology was first introduced to describe the fluid dynamics of heart valves [6,7], it has subsequently been applied to a wide range of problems in biological fluid dynamics and, more generally, to a variety of problems in which an incompressible flow interacts with an immersed structure. An overview of many of the applications of the IB method is provided in Ref. [5].

The IB formulation of problems of fluid-structure interaction couples an Eulerian description of the fluid to a Lagrangian description of the elasticity of the immersed structure. In the continuous setting, these descriptions are coupled by two Lagrangian-Eulerian interaction equations which take the form of integral transforms with Dirac delta function kernels. When the continuous equations are discretized for computer simulation, the delta function is approximated by a smoothed version of the delta function. One major advantage of this approach is that it greatly simplifies the task of grid generation: the moving Lagrangian mesh is not required to conform to the Eulerian grid, and for a non-adaptive IB scheme, the Eulerian grid is likewise not required to conform to the moving Lagrangian mesh. For adaptive versions of the IB method, the Eulerian grid is usually refined locally in the vicinity of the immersed elastic structure, and in this case, the Eulerian discretization clearly must adapt to the evolving configuration of the immersed elastic structure. Nonetheless, moving body-fitted grids are not required either in the uniform-grid or in the adaptive case.

Although formally second-order accurate versions of the IB method have been constructed [8, 9], such schemes yield actual second-order convergence rates only for problems which possess sufficiently smooth solutions. For problems which possess jumps in the fluid pressure and the derivatives of the fluid velocity at fluid-structure interfaces, the IB method appears to be limited to first order accuracy, and high spatial resolution is typically required to resolve the flow features in the boundary layers in the vicinity of the immersed structures. To allow high spatial resolution to be deployed only where it is most needed (e.g., in the vicinity of the immersed boundaries and in regions of high vorticity which are shed from those boundaries), Roma et al. [1,2] and Griffith et al. [3,4] introduced two different adaptive versions of the IB method. Using either scheme, it has been demonstrated that the accuracy of the IB method is largely determined by the spatial resolution in the vicinity of the immersed structures, and that comparatively coarse spatial resolution may be used away from the immersed structures without appreciably impacting the accuracy of the computed solutions. Thus, adaptive versions of the IB method can greatly
improve the efficiency of the IB methodology without significantly reducing the quality of the numerical results.

There are several similarities between the adaptive IB method of Roma et al. [1, 2] and that of Griffith et al. [3, 4]. In particular, both schemes employ the structured adaptive mesh refinement (AMR) approach introduced by Berger and co-workers [10, 11], and both schemes employ a version of the projection method [12–14] to solve the incompressible Navier-Stokes equations which describe the dynamics of the fluid. A key difference between the two adaptive IB methods is that the scheme of Roma et al. employs a staggered-grid discretization of the incompressible Navier-Stokes equations, whereas the scheme of Griffith et al. employs a cell-centered Eulerian spatial discretization. Roma et al. therefore were able to employ an exact projection method, whereas Griffith et al. employed a so-called approximate projection method. (Using an adaptive staggered-grid discretization, it is straightforward to construct an exact projection method for the incompressible Navier-Stokes equations, i.e., a projection method which, in exact arithmetic, exactly enforces the discrete divergence-free constraint on the velocity and which, in floating-point arithmetic, enforces this constraint to a level determined by, e.g., the floating-point precision and the convergence tolerances of the linear solver. In the case of cell-centered pressures and velocities, by contrast, the construction of an exact projection method is complicated by the non-compact finite-difference stencil of the cell-centered discrete projection operator [15]. To avoid the difficulties presented by the exact cell-centered projection operator, a number of adaptive approximate projection methods have been developed [16–19]. Such methods enforce the discrete divergence-free constraint on the velocity only up to the truncation error of the numerical scheme, employing a finite-difference approximation to the continuous projection operator which is not idempotent, i.e., which is not a projection matrix.) The treatment of the nonlinear advection terms also differs between the two adaptive IB methods. In the staggered-grid scheme of Roma et al., centered differencing is used for the advection terms, whereas an explicit second-order Godunov method [18, 20, 21] is employed in the cell-centered scheme of Griffith et al., enabling its use in higher Reynolds number applications. Finally, the method of Roma et al. is implicit in time, whereas the scheme of Griffith et al. is only semi-implicit.

In the present study, we compare the numerical performance of two adaptive versions of the IB method which are similar (but not identical) to the methods of Roma et al. [1, 2] and Griffith et al. [3, 4]. We apply these adaptive schemes to a standard test problem for IB-type methods, namely, the simulation of the dynamics of a thin, closed elastic boundary immersed in a viscous incompressible fluid. In the cell-centered case, the adaptive IB scheme is the same as that described in Ref. [22], which is based on the scheme of Refs. [3] and [4]. In the staggered-grid case, we use an incompressible flow solver which is the extension of the method of Ref. [23] to the adaptive context. Unlike the scheme
of Roma et al., the adaptive staggered-grid incompressible flow solver of the present work uses the projection method as a preconditioner rather than as a solver. Additionally, our adaptive staggered-grid solver employs a second-order Godunov scheme for the nonlinear advection terms which is based on the piecewise parabolic method (PPM) [24].

We compare the cell-centered and staggered-grid adaptive IB schemes in terms of their abilities to approximate translation- and rotation-invariant numerical solutions to problems which, in the continuous setting, are translationally and nearly rotationally invariant. For such problems, the degree to which the numerical solutions depend on the position or orientation of the immersed structure with respect to the background Eulerian grid provides a measure of the accuracy of the numerical methods. We also compare the volume-conservation properties of the two schemes. All of these tests are performed for several different choices for the regularized delta function. Although the staggered-grid discretization generally yields the most accurate results, our numerical experiments demonstrate that the two adaptive schemes perform comparably in terms of translation and rotation invariance at intermediate spatial resolutions. At very coarse and very fine spatial resolutions, however, the staggered-grid solver is found to be significantly less sensitive to perturbations than the cell-centered solver, i.e., the staggered-grid scheme exhibits significantly reduced grid dependence. The staggered-grid scheme also yields dramatically improved volume conservation in comparison to the cell-centered solver in all cases considered in the present study. An interesting finding, which mirrors earlier empirical results for Stokes flow IB methods [25, 26], is that the highest accuracy is obtained for versions of the regularized delta function which satisfy a so-called even-odd condition [5]. We also find for the present tests that increasing the number of discrete moment conditions satisfied by the regularized delta function does not necessarily yield increased accuracy.

2 Continuous formulation

In the IB formulation of problems of fluid-structure interaction, the momentum, velocity, and incompressibility of the coupled fluid-structure system are described in Eulerian form, and the elasticity of the immersed structure is described in Lagrangian form. Let \( \mathbf{x} = (x_1, x_2) \in U \) denote Cartesian (physical) coordinates, where \( U \subset \mathbb{R}^2 \) is the physical domain, let \( s \in \Omega \) denote Lagrangian (material) coordinates attached to the immersed elastic boundary, where \( \Omega \subset \mathbb{R} \) is the Lagrangian coordinate domain, and let \( \mathbf{X}(s, t) \in U \) denote the physical position of material point \( s \) at time \( t \), with \( \mathbf{X}(. , t) \) denoting the configuration of the entire structure at time \( t \). We shall take the density \( \rho \) and dynamic viscosity \( \mu \) of the fluid to be constant, and we shall assume that the immersed elastic boundary is massless, although it is possible to formu-
late versions of the IB method which do not require such assumptions. The equations of motion for the coupled fluid-structure system are

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

\[ \nabla \cdot \mathbf{u}(x, t) = 0, \]

\[ \mathbf{f}(x, t) = \int_{\Omega} \mathbf{F}(s, t) \delta(x - \mathbf{X}(s, t)) \, ds \]

\[ \frac{\partial \mathbf{X}}{\partial t}(s, t) = \int_{U} \mathbf{u}(x, t) \delta(x - \mathbf{X}(s, t)) \, dx, \]

\[ \mathbf{F}(s, t) = \mathcal{F}[\mathbf{X}(\cdot, t)](s), \]

where \( \mathbf{u}(x, t) = (u(x, t), v(x, t)) \) is the Eulerian fluid velocity field, \( p(x, t) \) is the Eulerian fluid pressure, \( \mathbf{f}(x, t) = (f_1(x, t), f_2(x, t)) \) is the Eulerian elastic force density (i.e., the elastic force density with respect to the physical coordinate system, so that \( \mathbf{f}(x, t) \, dx \) has units of force), \( \mathbf{F}(s, t) = (F_1(s, t), F_2(s, t)) \) is the Lagrangian elastic force density (i.e., the elastic force density with respect to the material coordinate system, so that \( \mathbf{F}(s, t) \, ds \) has units of force), \( \delta(x) = \delta(x_1) \delta(x_2) \) is the two-dimensional Dirac delta function, and \( \mathcal{F} : \mathbf{X} \mapsto \mathbf{F} \) maps the configuration of the Lagrangian structure to the corresponding Lagrangian elastic force density. In the present work, we take

\[ \mathbf{F}(s, t) = \mathcal{F}[\mathbf{X}(\cdot, t)](s) = \frac{\partial^2 \mathbf{X}}{\partial s^2}(s, t), \]

although it is important to emphasize that the IB formulation does not require \( \mathcal{F} \) to take the particular form given by Eq. (6).

In this IB formulation, the Lagrangian description of the immersed structure and the Eulerian description of the fluid are coupled via two Lagrangian-Eulerian interaction equations, Eqs. (3) and (4). In the present setting, in which the immersed structure is a thin elastic boundary, the Eulerian force density \( \mathbf{f}(x, t) \, dx \) is singular whereas the Lagrangian elastic force density \( \mathbf{F}(s, t) \) is not. Nonetheless, \( \mathbf{f}(x, t) \) and \( \mathbf{F}(s, t) \) are equivalent densities. In particular, it is possible to show that

\[ \int_{V} \mathbf{f}(x, t) \, dx = \int_{\mathbf{X}^{-1}(V, t)} \mathbf{F}(s, t) \, ds, \]

where \( V \subset U \) is arbitrary and where

\[ \mathbf{X}^{-1}(V, t) = \{ s \in \Omega : \mathbf{X}(s, t) \in V \}. \]

By contrast, because the fluid velocity is continuous as a result of the presence of viscosity in the system, the defining property of \( \delta(x) \) implies that Eq. (4) is equivalent to \( \frac{\partial \mathbf{X}}{\partial t}(s, t) = \mathbf{u}(\mathbf{X}(s, t), t) \). This may be interpreted as the no-slip and no-penetration conditions of a viscous incompressible fluid, but note
A. B.

Fig. 1. The two basic Eulerian spatial discretizations considered in the present study. 
A. The cell-centered discretization, in which the pressure and the components of the velocity are all described at the centers of the Cartesian grid cells. B. The staggered-grid or MAC (marker-and-cell) discretization, in which the pressure is described at the centers of the Cartesian grid cells and the edge-normal components of the velocity are described at the centers of the edges of the Cartesian grid cells.

that the no-slip and no-penetration conditions do not appear as constraints on the fluid motion. Instead, they are used to determine the motion of the immersed elastic structure. For a more complete discussion of the continuous formulation of the IB method, see Ref. [5].

3 Spatial discretizations

3.1 Eulerian discretizations

In the present work, the physical domain $U$ is taken to be $U = [-8, +8]^2$ with periodic boundaries. (Although earlier implementations of the IB method required the use of periodic boundary conditions for the fluid domain $U$, neither the IB method nor our present implementation of the IB method require the use of periodic boundary conditions [22]. In the present context, periodic boundary conditions are chosen to approximate an infinite domain.) We use a block-structured adaptively-refined Cartesian grid to discretize $U$, but we outline primarily the uniform-grid discretizations. In the present study, we consider two different adaptive discretizations of the incompressible Navier-Stokes equations. One discretization is the cell-centered discretization described in Ref. [22]. This adaptive scheme is a version of the adaptive cell-centered projection method of Ref. [4] which, in turn, is based on the inviscid adaptive approximate projection methods of Minion [16] and Martin and Colella [18] and the inviscid hybrid approximate projection method of Almgren et al. [27]. The other discretization which we employ is an adaptive version of the staggered-
grid (i.e., marker-and-cell or MAC [28]) scheme described in Ref. [23]. A complete description of this adaptive staggered-grid discretization will be presented elsewhere.

Let \( h = \Delta x_1 = \Delta x_2 \) be the Cartesian grid spacing, and let \((i,j)\) label individual Cartesian grid cells for \(i,j = 0,1,\ldots,N-1\). In both the cell-centered and the staggered-grid schemes, the pressure is approximated at the centers of the Cartesian grid cells. We denote by \( x_{i,j} \) the physical position of the center of Cartesian grid cell \((i,j)\), and we use the notation \( p^n_{i,j} \) to denote our approximation to the pressure at position \( x_{i,j} \) at time \( t^n = n \Delta t \). Likewise, we denote by \( x_{i-\frac{1}{2},j} \) and \( x_{i,j-\frac{1}{2}} \) the positions of the centers of the \(x\) - and \(y\) -edges (i.e., the edges \( x = \text{constant} \) and \( y = \text{constant} \)) of the Cartesian grid cells, and we use similar notation to denote quantities defined at the centers of such edges, so that the staggered-grid velocities on the left and bottom sides of cell \((i,j)\) are \( u^n_{i-\frac{1}{2},j} \) and \( v^n_{i,j-\frac{1}{2}} \), respectively. In the cell-centered discretization, the components of the velocities are defined at the centers of the Cartesian grid cells, i.e., the velocities are \( u^n_{i,j} \) and \( v^n_{i,j} \). See Fig. 1. In the uniform-grid context, we employ standard compact second-order accurate finite-difference approximations to the divergence, gradient, and Laplace operators. In the adaptively-refined context, we employ generalizations of these standard finite difference approximations which are at least first-order accurate in the vicinity of interfaces in grid resolution and which are second-order accurate away from such coarse-fine interfaces in the locally-refined grid.

3.2 Lagrangian discretization

In the present work, we take the immersed elastic structure to be a thin closed elastic boundary. The Lagrangian material coordinates are \( s \in \Omega = [0,1] \) with periodic boundary conditions at \( s = 0 \) and \( s = 1 \). The standard initial configuration of the immersed structure is

\[
X(s,t)|_{t=0} = (0.2 \cos(2 \pi s), 0.3 \sin(2 \pi s)),
\]

but note that in most of our numerical experiments, we perturb this standard initial configuration to test the grid dependence of the two adaptive IB methods; see Sec. 6. We employ a node-centered discretization of \( \Omega \), with \( \Delta s = \frac{1}{M} \) for an integer value of \( M > 0 \), and with \( s_l = l \Delta s \), \( l = 0, 1, \ldots, M - 1 \). The value of \( M = \frac{1}{\Delta s} \) is chosen to ensure that the physical positions of the nodes of the Lagrangian mesh are sufficiently close to avoid significant leaks in the immersed elastic boundary. We denote the physical position of node \( l \) at time \( t^n = n \Delta t \) by \( X^n_l \), and we compute a finite-difference approximation to \( F(s,t) \)
at Lagrangian mesh node $s_l$ via

$$F^n_l = \frac{X^n_{l+1} - 2X^n_l + X^n_{l-1}}{\Delta s^2}, \quad l = 0, \ldots, M - 1,$$

where the periodicity of the Lagrangian mesh implies that $X^n_{-1} = X^n_{M-1}$ and $X^n_M = X^n_0$. Note that this discretization can be viewed as a system of linear springs with zero resting lengths.

### 3.3 Lagrangian-Eulerian interaction

#### 3.3.1 Regularized versions of the Dirac delta function

To discretize the equations of Lagrangian-Eulerian interaction, the IB method employs a smoothed or regularized version of the Dirac delta function, $\delta_h(x)$. The conventional approach taken by the IB method, which we follow in the present work, is to define $\delta_h(x)$ to be the tensor product of one-dimensional regularized delta functions,

$$\delta_h(x) = \delta_h(x_1) \delta_h(x_2),$$

with

$$\delta_h(x) = \frac{1}{h} \phi \left( \frac{x}{h} \right).$$

Various choices are available for the function $\phi(r)$. In the present work, we consider three such choices.

A regularized delta function frequently used with the IB method is the four-point delta function, which is defined in terms of the function

$$\phi^{IB4}(r) = \begin{cases} 
\frac{1}{8} \left( 3 - 2 |r| + \sqrt{1 + 4 |r| - 4r^2} \right), & 0 \leq |r| < 1, \\
\frac{1}{8} \left( 5 - 2 |r| - \sqrt{-7 + 12 |r| - 4r^2} \right), & 1 \leq |r| < 2, \\
0, & 2 \leq |r|.
\end{cases}$$

The regularized delta function constructed in terms of $\phi^{IB4}(r)$ has a support of four meshwidths in each coordinate direction (e.g., a support of 16 grid cells in two spatial dimensions) and satisfies two discrete moment conditions along with a quadratic condition [5]. An important property of the four-point delta function is that it satisfies the even-odd condition, so that

$$\sum_{j \in \mathbb{Z}} \phi^{IB4}(r - j) = \sum_{j \in \mathbb{Z}} \phi^{IB4}(r - j) = \frac{1}{2} \quad \forall r \in \mathbb{R}.$$
Note that Eq. (14) implies that \( \phi^{IB4}(r) \) satisfies the zeroth-order discrete moment condition,
\[
\sum_{j \in \mathbb{Z}} \phi^{IB4}(r - j) = 1 \quad \forall r \in \mathbb{R}.
\tag{15}
\]

The even-odd condition (also referred to as the balanced condition [25]) was introduced to reduce grid-scale oscillations related to the checkerboard instability associated with node-centered and cell-centered finite-difference methods for the incompressible Navier-Stokes equations [5].

Requiring \( \phi(r) \) to satisfy all of the conditions satisfied by the four-point delta function along with two additional discrete moment conditions yields the six-point delta function,
\[
\phi^{IB6}(r) = \begin{cases} 
\frac{61}{112} - \frac{11}{42} |r| - \frac{11}{56} |r|^2 + \frac{1}{12} |r|^3 & 0 \leq |r| < 1, \\
+ \frac{\sqrt{3}}{336} (243 + 1584 |r| - 748 |r|^2 - 1560 |r|^3) & 1 \leq |r| < 2, \\
+ 500 |r|^4 + 336 |r|^5 - 112 |r|^6 & 2 \leq |r| < 3, \\
0 & 3 \leq |r|.
\end{cases}
\tag{16}
\]

Note that, like \( \phi^{IB4}(r) \), the six-point delta function satisfies the even-odd condition. Unlike \( \phi^{IB4}(r) \), which satisfies \( \phi^{IB4}(r) \geq 0 \) for all \( r \in \mathbb{R} \), \( \phi^{IB6}(r) \) possesses negative tails, i.e., \( \phi^{IB6}(r) < 0 \) for some \( |r| < 3 \).

We may also employ delta functions which do not satisfy the even-odd condition. One such delta function, which was introduced in the context of the adaptive staggered-grid IB method of Roma et al. [1,2], is the three-point delta function,
\[
\phi^{IB3}(r) = \begin{cases} 
\frac{1}{3} \left( 1 + \sqrt{1 - 3 |r|^2} \right) & 0 \leq |r| < \frac{1}{2}, \\
\frac{1}{6} \left( 5 - 3 |r| - \sqrt{1 - 3(1 - |r|)^2} \right) & \frac{1}{2} \leq |r| < \frac{3}{2}, \\
0 & \frac{3}{2} \leq |r|.
\end{cases}
\tag{17}
\]

Like \( \phi^{IB4}(r) \), \( \phi^{IB3}(r) \geq 0 \) for all \( r \in \mathbb{R} \). Moreover, \( \phi^{IB3}(r) \) and \( \phi^{IB4}(r) \) satisfy the same conditions, except that the three-point function satisfies the zeroth-order moment condition but not the even-odd condition. Because it does not satisfy the even-odd condition, \( \phi^{IB3}(r) \) is unsuitable for cell-centered discretizations which suffer from a well-known checkerboard instability; however, we may use \( \phi^{IB3}(r) \) with the stable staggered-grid discretization. Indeed, we find that the three-point delta function appears to yield a stable numerical method when used in conjunction with our adaptive staggered-grid discretization. Our em-
Empirical results indicate, however, that the three-point delta function is generally at least modestly, and in some cases substantially, less accurate than the four-point delta function.

### 3.3.2 Cell-centered discretization

The equations of Lagrangian-Eulerian interaction, Eqs. (3) and (4), are approximated in the cell-centered case by

\[
f_{i,j} = \sum_l F_l \delta_h(x_{i,j} - X_l) \Delta s \tag{18}
\]

and

\[
\frac{d}{dt} X_l = \sum_{i,j} u_{i,j} \delta_h(x_{i,j} - X_l) \Delta x_1 \Delta x_2. \tag{19}
\]

To simplify the exposition of the time-stepping methods in Sec. 4, we introduce the force-spreading operator \( S[X] \), which is defined via

\[
f_{i,j} = (S[X] F)_{i,j} = \sum_l F_l \delta_h(x_{i,j} - X_l) \Delta s. \tag{20}
\]

Because we use the same regularized delta function both to spread forces from the Lagrangian mesh to the Cartesian grid and also to interpolate velocities from the Cartesian grid onto the Lagrangian mesh, the interpolation operator is the adjoint of the force-spreading operator, i.e.,

\[
\frac{d}{dt} X_l = (S^*[X] u)_l = \sum_{i,j} u_{i,j} \delta_h(x_{i,j} - X_l) \Delta x_1 \Delta x_2. \tag{21}
\]

We use the shorthand \( f = S[X] F \) and \( \frac{dX}{dt} = S^*[X] u \).

### 3.3.3 Staggered-grid discretization

In the staggered-grid case, we perform force spreading and velocity interpolation component-wise, so that, with \( f = (f_1, f_2) \) and with \( F = (F_1, F_2) \),

\[
(f_1)_{i, \frac{1}{2}, j} = \sum_l (F_1)_l \delta_h(x_{i, \frac{1}{2}, j} - X_l) \Delta s, \quad \text{and} \tag{22}
\]

\[
(f_2)_{i, j, \frac{1}{2}} = \sum_l (F_2)_l \delta_h(x_{i, j, \frac{1}{2}} - X_l) \Delta s. \tag{23}
\]
and, with $u = (u, v)$ and with $X = (X_1, X_2)$,

$$
\frac{d}{dt} (X_1)_t = \sum_{i,j} u_{i-\frac{1}{2},j} \delta_h (x_{i-\frac{1}{2},j} - X_t) \Delta x_1 \Delta x_2, \quad \text{and} \quad (24)
$$

$$
\frac{d}{dt} (X_2)_t = \sum_{i,j} v_{i,j-\frac{1}{2}} \delta_h (x_{i,j-\frac{1}{2}} - X_t) \Delta x_1 \Delta x_2. \quad (25)
$$

Note that twice as many delta function weights must be computed in the staggered-grid case than are required by the cell-centered discretization. (In three spatial dimensions, three times as many delta function weights would need to be computed in the staggered-grid case than in the cell-centered case.)

As in the cell-centered case, staggered-grid force spreading and velocity interpolation are adjoint linear operators, and we denote force spreading by $f = S[X] F$ and velocity interpolation by $\frac{dX}{dt} = S^*[X] u$. It is important to keep in mind that although we employ the same notation to denote force spreading and velocity interpolation in the cell-centered and staggered-grid cases, the actual force-spreading and velocity-interpolation operators used in these two cases are in fact different.

4 Temporal discretizations

4.1 Cell-centered time discretization

In our cell-centered scheme, we follow an approach introduced in the inviscid case by Almgren et al. [27] by employing an approximate projection algorithm which is the hybridization of pressure-increment and pressure-update projection algorithms. In particular, the updated velocity is obtained via a second-order pressure-increment projection method [14, 29], and the updated pressure is obtained via a second-order pressure-update projection method [29, 30]. When exact projections are employed, the approximations to $u$ and $p$ obtained by the two different projection algorithms would be the same; however, this is not the case when approximate projections are used in place of exact projections [27].

The time discretization is as follows: Given $X^n, u^n,$ and $p^{n-\frac{1}{2}}$, we first compute $X^{(n+1)}$, an initial prediction of the positions of the nodes of the Lagrangian mesh at time $t^{n+1}$, via

$$
X^{(n+1)} = X^n + \Delta t S^*[X^n] u^n. \quad (26)
$$

We then compute the Lagrangian force densities $F^n$ and $F^{(n+1)}$ which correspond to the current and predicted structure configurations, $X^n$ and $X^{(n+1)}$. 
and obtain a timestep-centered approximation to the Eulerian elastic force density via
\[
f^{(n+\frac{1}{2})} = \frac{1}{2} \left( S[X^n] F^n + S[X^{(n+1)}] F^{(n+1)} \right).
\]
(27)

Next, we obtain the updated value of the fluid velocity in two steps. First, we solve
\[
\rho \left( \frac{u^* - u^n}{\Delta t} + N^{(n+\frac{1}{2})} \right) = -\nabla_h p^{n-\frac{1}{2}} + \frac{\mu}{2} \nabla_h^2 (u^* + u^n) + f^{(n+\frac{1}{2})}
\]
(28)
for the intermediate velocity field \(u^*\), where \(N^{(n+\frac{1}{2})}\) is obtained by an explicit second-order Godunov procedure [4, 9, 18, 21] which uses the xsPPM7 version [31] of the piecewise parabolic method (PPM) [24], and where, in the uniform-grid context, \(\nabla_h \cdot \), \(\nabla_h \), and \(\nabla_h^2\) are standard compact finite-difference approximations to the divergence, gradient, and Laplace operators. (Because \(\nabla_h^2\) is the compact 5-point Laplacian, note that for the cell-centered discretization, \(\nabla_h^2 \neq \nabla_h \cdot \nabla_h\).) Second, we approximately project \(u^*\) by solving
\[
\nabla_h^2 \varphi = \frac{\rho}{\Delta t} \nabla_h \cdot u^*
\]
(29)
for \(\varphi\) and by evaluating
\[
u^{n+1} = u^* - \frac{\Delta t}{\rho} \nabla_h \varphi.
\]
(30)

Note that defining \(u^{n+1}\) in this manner is equivalent to defining \(u^{n+1}\) via
\[
u^{n+1} = P_h u^* = \left( I - \nabla_h \left( \nabla_h^2 \right)^{-1} \nabla_h \cdot \right) u^*,
\]
(31)
in which the operator \(P_h = \left( I - \nabla_h \left( \nabla_h^2 \right)^{-1} \nabla_h \cdot \right)\) is a finite difference approximation to the continuous projection operator \(\bar{P} = \left( I - \nabla \left( \nabla^2 \right)^{-1} \nabla \cdot \right)\). The approximate projection operator \(P_h\) is not itself a projection matrix \((P_h^2 \neq P_h)\), however, because \(\nabla_h^2 \neq \nabla_h \cdot \nabla_h\). Consequently, in general, \(\nabla_h \cdot u^{n+1} \neq 0\). Nonetheless, \(\nabla_h \cdot u^{n+1} \rightarrow 0\) as \(h \rightarrow 0\), and, for smooth solutions, \(\nabla_h \cdot u^{n+1} = \mathcal{O}(h^2)\).

We could, in principle, use \(\varphi\) defined by Eq. (29) along with \(p^{n-\frac{1}{2}}\) to compute an approximation to the pressure at time \(t^{n+\frac{1}{2}}\). We do not do so because we have found that such a scheme suffers from grid-scale oscillations which are severely exacerbated by local mesh refinement [3, 9]. Instead, we obtain the updated value of the pressure via a \textit{pressure-free} projection method. First, we solve
\[
\rho \left( \frac{\bar{u}^* - u^n}{\Delta t} + N^{(n+\frac{1}{2})} \right) = \frac{\mu}{2} \nabla_h^2 (\bar{u}^* + u^n) + f^{(n+\frac{1}{2})}
\]
(32)
for $\tilde{u}^*$. Next, we solve
\[ \nabla_h^2 \tilde{\varphi} = \frac{\rho}{\Delta t} \nabla_h \cdot \tilde{u}^* \] (33)
for $\tilde{\varphi}$. Finally, we obtain our approximation to the updated pressure via
\[ p^{n+\frac{1}{2}} = \left( I - \frac{\Delta t \mu}{2 \rho \nabla_h^2} \right) \tilde{\varphi}. \] (34)
(See, e.g., Ref. [29] for the derivation of this pressure-update equation.) Note that, generally, $u^* \neq \tilde{u}^*$ and $\varphi^* \neq \tilde{\varphi}^*$. Although the value of $p^{n+\frac{1}{2}}$ does not affect the value of $u^{n+1}$, it does affect the values of $u^{n+2}, u^{n+3}, \ldots$.

Last, with $u^{n+1}$ and $p^{n+\frac{1}{2}}$ in hand, we compute a final approximation to the updated positions of the nodes of the Lagrangian mesh by evaluating
\[ X^{n+1} = X^n + \frac{\Delta t}{2} \left( S^n[X^n] u^n + S^n[X^{n+1}] u^{n+1} \right). \] (35)
Note that this is an explicit equation for $X^{n+1}$.

### 4.2 Staggered-grid time discretization

The time discretization employed for the adaptive staggered-grid spatial discretization is an extension of the uniform-grid method for the incompressible Navier-Stokes equations described in Ref. [23]. This scheme employs truncated fixed-point iteration to discretize the coupled system of equations, treating the linear terms in the incompressible Navier-Stokes terms implicitly, and treating all other terms explicitly. Let $X^{n+1,k}, u^{n+1,k},$ and $p^{n+\frac{1}{2},k}$ denote the approximations to the values of $X$ and $u$ at time $t^{n+1}$ and to the value of $p$ at time $t^{n+\frac{1}{2}}$ obtained after $k$ steps of fixed-point iteration. We obtain $X^{n+1,k+1}, u^{n+1,k+1},$ and $p^{n+\frac{1}{2},k+1}$ by solving the linear system of equations
\[ \rho \left( \frac{u^{n+1,k+1} - u^n}{\Delta t} + N^{n+\frac{1}{2},k} \right) = -\nabla_h p^{n+\frac{1}{2},k+1} + \mu \nabla_h^2 u^{n+\frac{1}{2},k+1} + f^{n+\frac{1}{2},k}, \] (36)
\[ \nabla_h \cdot u^{n+1,k+1} = 0, \] (37)
\[ f^{n+\frac{1}{2},k} = S[X^{n+\frac{1}{2},k}] F^{n+\frac{1}{2},k}, \] (38)
\[ \frac{X^{n+1,k+1} - X^n}{\Delta t} = S^*[X^{n+\frac{1}{2},k}] u^{n+\frac{1}{2},k+1} \] (39)
\[ F^{n+\frac{1}{2},k} \approx F[X^{n+\frac{1}{2},k}], \] (40)
where
\[
X^{n+1/2,k} = \frac{1}{2} \left( X^{n+1,k} + X^n \right), \tag{41} \\
X^{n+1,0} = X^n, \tag{42} \\
u^{n+1/2,k} = \frac{1}{2} \left( u^{n+1,k} + u^n \right), \tag{43} \\
u^{n+1,0} = u^n, \tag{44} \\
u^n|_{n=0} = 0, \tag{45} \\
p^{n+1/2,0} = p^{n-1/2}, \tag{46} \\
p^{n+1/2}|_{n=0} = 0, \tag{47}
\]

and where \(N^{n+1/2,k} \approx \left[ (u^{n+1/2,k} \cdot \nabla)u^{n+1/2,k} \right]\) is an explicit PPM approximation to the nonlinear advection term described in Ref. [23]. As in the cell-centered case, we employ standard compact second-order accurate approximations to the divergence, gradient, and Laplace operators [23]. Note that solving Eqs. 36–40 for \(X^{n+1,k+1}\), \(u^{n+1,k+1}\), and \(p^{n+1/2,k+1}\) requires only the solution of a Crank-Nicolson-type discretization of the time-dependent incompressible Stokes equations. As in Ref. [23], we solve this system of equations via the flexible GMRES (FGMRES) algorithm [32], using \(u^{n+1,k}\) and \(p^{n+1/2,k}\) as initial approximations to \(u^{n+1,k+1}\) and \(p^{n+1/2,k+1}\), and using a pressure-free projection method with multigrid subdomain solvers as a preconditioner [23].

A minimum of two cycles of fixed-point iteration must be performed at each timestep to obtain a second-order accurate timestepping scheme, and in the computations of the present work, we perform precisely two cycles per timestep, so that \(X^{n+1} = X^{n+1,2}\), \(u^{n+1} = u^{n+1,2}\), and \(p^{n+1/2} = p^{n+1/2,2}\). Numerical experimentation indicates that doing so results in a stability restriction on the timestep size which is related to the explicit treatment of the advection term, limiting the maximum CFL number (see Eq. (48)) to one-half. The maximum CFL number of the fluid solver can be increased to one if we were to perform an additional cycle of fixed-point iteration per timestep [23]; however, in practice, we are generally not able to compute at such high CFL numbers because our explicit treatment of the elasticity generated by the immersed structure results in an additional stability restriction on \(\Delta t\). Because performing an additional cycle of fixed point iteration does not appear to improve either the accuracy or the stability of the present IB method, we simply perform two cycles of fixed-point iteration per timestep.
Fig. 2. Schematic diagram of a two-level locally-refined Cartesian grid with a Lagrangian immersed boundary embedded in the finest level. In an actual IB computation, the Lagrangian mesh spacing must be finer than indicated by the schematic to avoid significant leaks.

5 Adaptive mesh refinement

We solve the incompressible Navier-Stokes equations on a block-structured adaptively-refined hierarchical Cartesian grid. The locally-refined grid is comprised of a sequence of nested levels which are labeled from coarsest to finest by $\ell = 0, 1, \ldots, \ell_{\text{max}}$. Each level is comprised of one or more rectangular grid patches, and the grid patches on a particular grid level $\ell$ all share the same Cartesian grid spacing $h_\ell$. To simplify the implementation, we require the patches on a particular level $\ell$ to be non-overlapping. The Cartesian grid spacing on a particular level $\ell$ is related to that on level $\ell + 1$ by an integer refinement ratio $r$, so that $h_\ell = r h_{\ell + 1}$. A schematic diagram of a two-level locally-refined grid is shown in Fig. 2. In the computations of the present work, we set $r = 4$ and $\ell_{\text{max}} = 4$.

The locally-refined grid is constructed in a recursive fashion: First, level 0 is constructed to cover the entire physical domain $U$. Next, having constructed levels $0, 1, \ldots, \ell < \ell_{\text{max}}$, level $\ell + 1$ is generated by (1) tagging cells on level $\ell$ for refinement, (2) covering the tagged level $\ell$ grid cells by rectangular boxes generated by a box-generation algorithm (e.g., the Berger-Rigoutsos point-clustering algorithm [33]), and (3) refining the generated boxes by the integer refinement ratio $r$ to form the level $\ell + 1$ grid patches. (The grid-generation algorithm used in our implementation may modify the generated boxes to enforce the proper-nesting condition [11].) This grid-generation procedure ensures that the physical region covered by the union of the level $\ell + 1$ grid patches is strictly contained within the region covered by the union of the level $\ell$ grid patches.

In the computations presented herein, two cell-tagging criteria are employed to generate the locally-refined grid. The first criterion ensures that the im-
mersed structure is embedded in the finest level of the locally-refined grid by tagging cell \((i, j)\) and its eight neighbors whenever there is a Lagrangian mesh node \(l\) such that \(X_l\) is contained in the physical region covered by Cartesian grid cell \((i, j)\). This construction ensures that there are sufficiently many level \(\ell_{\text{max}}\) grid cells surrounding each Lagrangian mesh node to avoid complicating the discretization of the equations of Lagrangian-Eulerian interaction. In particular, generating the locally-refined grid in this manner allows us to use the standard uniform-grid discretization of the Lagrangian-Eulerian interaction equations without modification in the adaptive IB schemes. Cells are also tagged for refinement wherever the local magnitude of the vorticity exceeds 
\[0.25 \| \nabla \times \mathbf{u} \|_{\infty},\]
although for the present application, regions of high vorticity remain confined along the immersed boundary. Consequently, essentially no Cartesian grid cells are tagged for refinement which would not have been tagged by the Lagrangian mesh node-based tagging criterion alone.

The locally-refined grid is regenerated at a regular interval which is chosen to ensure that the support of \(\delta_h(x - X_l)\) is contained within the union of the level \(\ell_{\text{max}}\) grid patches for each Lagrangian mesh node \(l\). In other words, the locally-refined grid is regenerated often enough to ensure that each node of the Lagrangian mesh spreads force to and interpolates velocity from only the finest level of the hierarchical Cartesian grid. To simplify the regridding process, we assume that the timestep size \(\Delta t\) is chosen to satisfy a CFL condition of the form
\[\| \mathbf{u} \|_{\infty} \Delta t \leq C h_{\ell}, \quad 0 \leq \ell \leq \ell_{\text{max}},\]
where \(C\) is the so-called CFL number. This condition implies that the Lagrangian mesh nodes may move at most \(C\) Cartesian grid meshwidths in any coordinate direction per timestep. Thus, it is sufficient to regrid the patch hierarchy once per \(\lfloor 1/C \rfloor\) timesteps. In the present study, we choose \(\Delta t\) to satisfy Eq. (48) with \(C = \frac{1}{6}\), and we regenerate the locally-refined grid every sixth timestep. (For our cell-centered IB scheme to remain stable, it is necessary that \(\Delta t\) be chosen to satisfy Eq. (48) with \(C < 1\), and for our staggered-grid scheme, it is necessary that \(\Delta t\) satisfy Eq. (48) with \(C < \frac{1}{2}\). These conditions are not sufficient to guarantee stability of the adaptive IB schemes, however, because our time-explicit treatment of the Lagrangian structure results in an additional stability restriction of the form \(\Delta t = \mathcal{O}\left(h_{\ell_{\text{max}}}^2\right)\), which is more severe than either of these CFL-type restrictions.)

Each time that a new locally-refined Cartesian grid is generated, Eulerian quantities must be transferred from the old grid hierarchy to the new one. In newly-refined regions of the physical domain, the cell-centered velocity is prolonged from the old grid to the new one via slope-limited conservative linear interpolation [4], the staggered-grid velocity is prolonged via a specialized interpolation scheme which preserves the discrete divergence and curl of the MAC velocity field [34], and the pressure is prolonged by simple (non-
Fig. 3. Three views of a five-level locally-refined grid at the initial time $t = 0$ for the case $h_{\ell_{\text{max}}} = 1/256$. The Cartesian grid cells are indicated by thin black lines, and the Lagrangian mesh appears as a curved thick black line embedded in the finest level of the locally-refined Cartesian grid. Notice that the Lagrangian mesh is well separated from the nearest coarse-fine interfaces in the locally-refined Cartesian grid, allowing us to use the standard uniform-grid force-spreading and velocity-interpolation operators without modification in the adaptive schemes.

Fig. 4. The pressure field and the locally-refined Cartesian grid patches plotted at equally-spaced time intervals during the first oscillation of the immersed elastic boundary. This computation was performed using the adaptive staggered-grid discretization with $\delta_{h}^{\text{IB4}}(x)$ and with $h_{\ell_{\text{max}}} = 1/512$. Note that only the portion of the physical domain which is in the vicinity of the immersed boundary is shown in the panels of the present figure; see also Fig. 3.

conservative) linear interpolation. In newly-coarsened regions of the domain, the values of the velocity and pressure are set to be the (conservative) averages of the underlying fine-grid values from the old grid hierarchy. In preliminary computational experiments, we found no advantage to re-projecting the cell-centered velocity field following regridding, and consequently we do not do so in the experiments reported herein. Because we interpolate the staggered-grid velocity in a second-order accurate, divergence- and curl-preserving manner, such regrid projections are not required in the staggered-grid context.

6 Numerical results

All of our present numerical results are obtained by simulating the dynamics of a thin elastic interface immersed in a viscous incompressible fluid; see Figs. 3 and 4. The fluid domain $U$ is taken to have periodic boundaries, and the large size of $U$ practically eliminates interactions between periodic copies of the im-
mersed structure. Consequently, we believe that periodic boundary conditions are a good approximation to free-space conditions for this problem. In the continuous setting, this problem is translationally invariant, and because of the size of the physical domain, it is also very nearly rotationally invariant. In the discrete setting, however, the IB method only approximates translation and rotation invariance for translationally- and rotationally-invariant problems [5]. In the continuous setting, independent of translations or rotations of the initial configuration of the immersed elastic interface, the area of the region enclosed by that interface remains constant as a consequence of the incompressibility of the fluid and the boundary-advection equation, Eq. (4), which is equivalent to the no-slip and no-penetration conditions. In the discrete setting, the IB method does not yield exact volume conservation [35].

To compare the grid-dependence properties of the cell-centered and staggered-grid adaptive IB schemes, we perform several series of simulations in which we perturb the initial configuration of the immersed interface either by a randomly-chosen linear displacement \((\delta x, \delta y)\), or by a randomly-chosen rotation \(\delta \theta\) about the origin. Let \(T\) index a particular trial, corresponding to a particular choice of \((\delta x, \delta y)\) or \(\delta \theta\), and let \(X^{n,T}\) denote the configuration of the elastic structure at time \(t^n\) for trial \(T\). The initial configuration of the elastic structure for trial \(T\) is given by

\[
X^0_T = \mathcal{P}_T X^0,
\]

(49)

where \(\mathcal{P}_T\) is an operator which is used to perturb the standard initial configuration \(X^0\) of the elastic interface described in Sec. 3.2. In the present work, \(\mathcal{P}_T\) is either a translation operator or a rotation operator. We do not consider the effects of combining translations and rotations in the present study.

For each trial \(T\), we simulate the perturbed system using the perturbed initial conditions \(X^{0,T}\), and for each corresponding pair of trials \((T_1, T_2)\), we apply the inverse of the appropriate perturbation operator to each Lagrangian structure and compute the maximum of the Euclidean norm of the difference between the positions of corresponding Lagrangian mesh nodes. In particular, we compute

\[
D^n_{T_1,T_2} = \max_{0 \leq l \leq M-1} \|\mathcal{P}_{T_1}^{-1} X^{n,T_1}_l - \mathcal{P}_{T_2}^{-1} X^{n,T_2}_l\|,
\]

(50)

where \(\mathcal{P}_T^{-1} X^{n,T}_l\) is the position of Lagrangian mesh node \(l\) at time \(t^n\) in the original, unperturbed configuration. (Note that when we compute \(D^n_{T_1,T_2}\), the values of \(X^{n,T_1}\) and \(X^{n,T_2}\) are always numerical solutions obtained using the same Eulerian discretization method and the same regularized delta function.) If the numerical scheme is exactly translationally or rotationally invariant, then \(D \equiv 0\); however, generally \(D > 0\), and the value of \(D\) provides a measure of the extent to which the scheme is yielding grid-dependent numerical results.

For each trial \(T\), we also compute the percent area change \(A^n_T\), which we
define so that, if the initial area is preserved exactly, then \( A \equiv 0 \), and so that \( A > 0 \) corresponds to a loss in volume. In practice, \( A \neq 0 \) because the continuous interpolated Lagrangian velocity field is generally not continuously divergence free, even if the Eulerian velocity field satisfies exactly the discrete divergence-free condition [35]. In an IB computation, typically \( A > 0 \).

In all of our simulations, we take \( \rho = 1 \) and \( \mu = 0.0025 \), yielding a peak Reynolds number of approximately 400. We employ a five-level locally-refined Cartesian grid with a refinement ratio of four between levels. We let the size of the coarsest level of the locally-refined grid range from \( 4 \times 4 \) to \( 32 \times 32 \), yielding Cartesian grid spacings of \( h_{\ell_{\text{max}}} = \frac{1}{64} \) to \( \frac{1}{512} \) on the finest level of the hierarchical grid. (Note that the corresponding uniformly-fine computations would require Cartesian grid sizes ranging from \( 1024 \times 1024 \) to \( 8192 \times 8192 \).) The timestep size is taken to be \( \Delta t = 0.08 h_{\ell_{\text{max}}} \), which allows the numerical scheme to remain stable throughout the computation for all of the grid spacings considered, and which also allows us to regrid the patch hierarchy once every six timesteps without having any Lagrangian mesh nodes escape from the finest level of the locally-refined Cartesian grid. The number of Lagrangian nodes ranges from 206 for the coarsest set of simulations to 1641 for the finest set of simulations. These values are chosen so that neighboring nodes of the Lagrangian mesh are physically separated by a distance of approximately \( 0.5 h_{\ell_{\text{max}}} \).

Computations were performed using the IBAMR and IBTK codes [36], using Subversion revision 814 of IBAMR and Subversion revision 502 of IBTK. Much of the functionality of IBAMR/IBTK is provided by the SAMRAI [37–39], PETSc [40–42], and hypre [43,44] libraries.

### 6.1 Translation invariance

In our first set of numerical experiments, we randomly perturb the initial position of the structure by a linear displacement \((\delta x, \delta y)\). For each Eulerian discretization method, choice of \( \delta h(x) \), and grid spacing, we perform 100 trials, and in each trial, \( \delta x \) and \( \delta y \) are independently chosen random variables. Because we are using periodic boundary conditions, the IB method is insensitive to linear shifts which are multiples of the Cartesian grid spacing, and it suffices to consider \( \delta x \) and \( \delta y \) which are uniformly distributed in \([-\frac{1}{2} h_{\ell_{\text{max}}}, +\frac{1}{2} h_{\ell_{\text{max}}}]\), where \( h_{\ell_{\text{max}}} \) is the grid spacing on the finest level of the locally-refined grid. (In principle, shifts in the position of the immersed structure can also affect the positioning of finer levels of the Cartesian grid with respect to coarser levels; however, in our tests, the AMR grid-generation algorithm has been configured to minimize such effects. See also the experiments of Sec. 6.2, in which we consider the effects of large rotations of the immersed structure with respect to
Fig. 5. The mean values over all pairs of trials of $\mathcal{D}$ for the randomly-shifted problem of Sec. 6.1 for the two Eulerian discretizations and the various choices of $\delta_h(x)$. Note that the vertical scale differs in each panel. At intermediate resolutions, the staggered-grid and cell-centered schemes yield comparable results, but at very coarse and at very fine resolutions, the staggered-grid scheme yields substantially reduced grid dependence. Note also that the staggered-grid scheme with the four-point delta function yields superior results at the coarsest resolution, whereas the staggered-grid scheme with the six-point delta function yields superior results at finer grid resolutions.

The mean values of $\mathcal{D}$ for both Eulerian discretization methods and the various choices of regularized delta functions are plotted as functions of time in Fig. 5, and the maximum values of $\mathcal{D}$ are plotted in Fig. 6. These re-
Fig. 6. Similar to Fig. 5, but here plotting the maximum values over all pairs of trials of $D$. Again, the four-point delta function yields superior results at the coarsest resolution, whereas the six-point delta function yields superior results at finer grid resolutions.

Results provide a measure of the extent to which the versions of the IB method are sensitive to grid-scale linear perturbations. Note that smaller values of $D$ indicate lower dependence of the numerical results on the position of the immersed structure with respect to the background computational grid. It is clear from the figures that the staggered-grid discretization offers reduced grid dependence compared to the cell-centered discretization when either the four-point or the six-point delta function is used, with the four-point delta function being the most accurate at the coarsest spatial resolution and the six-point delta function being the most accurate at all other resolutions. The accuracy of the cell-centered discretization with the four-point delta function is comparable to the accuracy of the staggered-grid scheme only at the two intermediate spatial resolutions, $h_{\text{max}} = \frac{1}{128}$ and $\frac{1}{256}$. The cell-centered scheme with the six-point delta function yields the least-accurate results in all cases,
Fig. 7. The mean values over all trials of $\mathcal{A}$ for the randomly-shifted problem of Sec. 6.1 for the two Eulerian discretizations and the various choices of $\delta_h(x)$. Note the tremendous difference in volume-conservation properties between the cell–centered and staggered-grid Eulerian discretizations. The four-point delta function yields superior volume conservation at all grid spacings considered. See also Fig. 8.

sometimes by a wide margin. This is a surprising result because the six-point delta function satisfies more discrete moment conditions than the four-point function and might therefore be expected to yield higher accuracy. It is also interesting to note that the three-point delta function performs more poorly than the four-point delta function in most cases, especially at the two coarser spatial resolutions, even though both functions satisfy the same number of discrete moment conditions. In particular, the primary difference between the two functions is that the four-point function satisfies the even-odd condition, whereas the three-point function does not.

The mean values of $\mathcal{A}$ are plotted as functions of time in Figs. 7 and 8. (There is very little variation in $\mathcal{A}$ between trials, and the maximum values of $\mathcal{A}$ are
Fig. 8. Similar to Fig. 7, but here only showing the results for the staggered-grid discretization. As in the cell-centered case, the four-point delta function yields superior volume conservation at all grid spacings considered. Nearly identical to the mean values.) The improvement in accuracy offered by the staggered-grid scheme is dramatic, and in all cases, the best volume conservation is obtained by the staggered-grid spatial discretization with the four-point delta function. The six-point delta function yields significantly less accurate results than the four-point delta function in both the cell-centered and the staggered-grid case, and, at least in these computations, the three-point function appears to be less accurate than the six-point function; however, see Fig. 11. It is interesting to note in Fig. 8 that for the staggered-grid discretization, the volume loss yielded by the three- or six-point delta functions at the finest spatial resolution considered (i.e., $h_{\ell_{\text{max}}} = \frac{1}{512}$) is approximately the same as that yielded by the four-point delta function at the coarsest spatial resolution considered (i.e., $h_{\ell_{\text{max}}} = \frac{1}{64}$), although in all cases, the loss in volume is substantially less than 1% over the simulated time interval.
Fig. 9. Similar to Fig. 5, but here for the randomly-rotated problem of Sec. 6.2. Unlike the case of random grid-scale shifts, however, note that the staggered-grid scheme equipped with the four-point delta function yields the highest accuracy in all cases.

6.2 (Approximate) rotation invariance

In our second set of experiments, we randomly perturb the initial position of the structure by rotating it about the origin by a rotational displacement $\delta \theta$. For each Eulerian discretization method, choice of $\delta_h(x)$, and grid spacing, we perform 100 trials, and in each trial, $\delta \theta$ is a random variable uniformly distributed in $[-\frac{1}{4}\pi, +\frac{1}{4}\pi]$. (In this case, it does not suffice to consider only small rotations with respect to the Cartesian grid because different numerical results, not related by a simple rotation, are obtained for all $0 < \delta \theta \leq \frac{1}{4}\pi$. We consider the larger interval $-\frac{1}{4}\pi \leq \delta \theta \leq \frac{1}{4}\pi$ to account for any differences in the locally-refined grids which result from the different rotated structure configurations.) As in the translation-invariance tests of Sec. 6.1, we simulate
Fig. 10. Similar to Fig. 9, but here for the randomly-rotated problem of Sec. 6.2. As in Fig. 9, the staggered-grid scheme equipped with the four-point delta function yields the highest accuracy in all cases.

The mean values of $\mathcal{D}$ for both spatial discretizations and the various choices of regularized delta functions are plotted as functions of time in Fig. 9, and the maximum values of $\mathcal{D}$ are plotted in Fig. 10. In this case, $\mathcal{D}$ provides a measure of the sensitivity of the numerical scheme to the orientation of the immersed boundary with respect to the background computational grid. Like the case of random linear perturbations, the staggered-grid scheme yields the most accurate results for the present set of experiments, but unlike the randomly-shifted experiments, the staggered-grid scheme with the four-point delta function is generally more accurate than the staggered-grid scheme with
the three- and six-point delta functions. Moreover, in the present experiments, the three-point delta function is at least approximately as accurate as the six-point delta function, and at the coarsest grid resolution, the three-point delta function is markedly more accurate than the six-point delta function.

The volume conservation results for this set of experiments are essentially the same as those of the experiments of Sec. 6.1 shown in Figs. 7 and 8. Therefore, these data are not shown. As before, the cell-centered discretization yields large volume losses, whereas the staggered-grid scheme yields superior volume conservation, and the staggered-grid discretization equipped with the four-point delta function yields the highest accuracy.

6.3 Volume conservation in pseudo-steady state conditions

Our final set of tests examines the volume-conservation properties of the staggered-grid discretization over longer time intervals which are sufficient for the immersed elastic structure to reach near steady-state conditions. (For the present test case, the IB methods of this study do not reach a true steady state configuration.) In particular, we initialize the immersed structure in its standard configuration, and we simulate the dynamics of the coupled fluid-structure system up to time \( t = 24 \), a duration which is 20 times that of the simulations of Secs. 6.1 and 6.2. Simulation results are shown in Fig. 11. It is clear from the figure that the four-point delta function yields superior volume conservation and, moreover, that the pseudo-steady state rate of volume loss is significantly lower for the four-point delta function when compared to the six-point delta function. It is interesting to note that the volume loss yielded by the six-point delta function on the finest grid, for which \( h_{\ell_{\max}} = \frac{1}{512} \), is only slightly lower than that yielded by the four-point delta function on the coarsest grid, for which \( h_{\ell_{\max}} = \frac{1}{64} \). It is also interesting to note that although the six-point delta function appears to yield better volume conservation for shorter-duration simulations, the volume conservation yielded by the three-point delta function is superior to that yielded by the six-point delta function over longer time intervals (compare Figs. 8 and 11). Note that for this test, the maximum area loss is approximately 5% at \( h_{\ell_{\max}} = \frac{1}{64} \) and approximately 1% at \( h_{\ell_{\max}} = \frac{1}{512} \) when we use the four-point delta function.

7 Conclusions

In this study, we have compared the accuracy of two different adaptive versions of the IB method in terms of their grid-dependence and volume-conservation properties. For the simple test problem considered, an adaptive staggered-grid
Fig. 11. Volume conservation in pseudo-steady state conditions. The percent area loss is plotted as a function of time for several long-running simulations at different grid spacings and with different choices of the regularized delta function.

A scheme similar to that of Roma et al. [1, 2] is found to yield results which are generally superior to those of an adaptive cell-centered scheme similar to that of Griffith et al. [3, 4]. In particular, so long as the appropriate version of the regularized delta function $\delta_h(x)$ is used, the adaptive staggered-grid scheme yields reduced grid dependence and dramatically improved volume conservation. Interestingly, the staggered-grid discretization produces improved volume conservation without requiring non-standard, non-compact finite difference approximations to the divergence and gradient operators like those constructed by Peskin and Printz [35]. A potential line of future research is to try to determine additional (or alternative) constraints to impose on $\delta_h(x)$ to yield further improvements in volume conservation in the staggered-grid context. We believe that the test problem of the present study, in which the only stress-free configuration for the elastic structure corresponds to a configuration in which the structure has collapsed to a single point, represents a
worst-case scenario for the IB method in terms of volume conservation. Most “real” elastic materials will possess a non-degenerate stress-free configuration, and for such materials, we expect that both IB schemes considered in the present work will yield better volume conservation than the results of the present study might indicate.

An important finding of this study is that the three-point delta function, which was first used in the context of the adaptive staggered-grid scheme of Roma et al. [1, 2], yields relatively poor results compared to the four- and six-point delta functions conventionally used with node- and cell-centered IB methods. The primary difference between the four-point and three-point delta functions is that the four-point delta function satisfies the even-odd condition, whereas the three-point delta function does not. Although it may seem like it is not important for $\delta_h(x)$ to satisfy the even-odd condition for a discretization which does not suffer from the checkerboard instability, our numerical results indicate that satisfying the even-odd condition has benefits beyond increased stability. To our knowledge, this is the first study to suggest that it is important to use delta functions which satisfy the even-odd condition in the context of a staggered-grid version of the IB method. Our results reinforce those of Bringley and Peskin [25], who found in the context of a Stokes flow IB method that regularized delta functions which satisfy the even-odd condition (which is referred to as the balanced condition in Ref. [25]) generally yield higher accuracy than those which do not satisfy the even-odd condition. Similar results were reported by Mori [26], who also observed that regularized delta functions which satisfy the even-odd condition yield higher convergence rates than regularized delta functions which do not satisfy the even-odd condition, again, in the context of a Stokes flow IB method.

Although we have previously documented cases in which the six-point delta function yields superior accuracy when compared to the four-point delta function [4, 9], the empirical results of the present study lead us to conclude that a good choice for a general-purpose IB code is the staggered-grid IB method used in conjunction with the four-point delta function. We also believe that this work underscores the importance of using an implementation of the IB method which facilitates experimentation with alternative discretizations, including various choices for the regularized delta functions, for the many cases in which it is not clear a priori which discretization choices will yield the most accurate results.

Acknowledgments

The author wishes to acknowledge discussions of this work with Charles Peskin and David McQueen and wishes to thank David McQueen for reading
an early draft of this manuscript. Computations were performed at New York University using computer facilities funded in large part by a generous donation by St. Jude Medical, Inc. This work was funded in part by a grant from the American Heart Association.

References


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


[43] hypre: High performance preconditioners, 

preconditioners, in: P. M. A. Sloot, C. J. K. Tan, J. J. Dongarra, A. G. 
Hoekstra (Eds.), Computational Science - ICCS 2002 Part III, Vol. 2331 of 
Lecture Notes in Computer Science, Springer-Verlag, 2002, pp. 632–641, also 