

Penalty immersed boundary method for an elastic boundary with mass

Cite as: Phys. Fluids **19**, 053103 (2007); <https://doi.org/10.1063/1.2734674>

Submitted: 21 December 2006 • Accepted: 27 March 2007 • Published Online: 16 May 2007

Yongsam Kim and Charles S. Peskin



View Online



Export Citation

ARTICLES YOU MAY BE INTERESTED IN

[A penalty immersed boundary method for a rigid body in fluid](#)

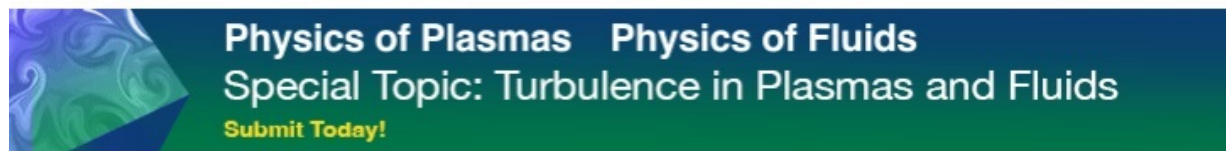
Physics of Fluids **28**, 033603 (2016); <https://doi.org/10.1063/1.4944565>

[Fluid-structure interaction simulation based on immersed boundary-lattice Boltzmann flux solver and absolute nodal coordinate formula](#)

Physics of Fluids **32**, 047109 (2020); <https://doi.org/10.1063/1.5144752>

[Dynamics of an inverted flexible plate in a uniform flow](#)

Physics of Fluids **27**, 073601 (2015); <https://doi.org/10.1063/1.4923281>



Penalty immersed boundary method for an elastic boundary with mass

Yongsam Kim^{a)} and Charles S. Peskin^{b)}

*Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street,
New York, New York 10012*

(Received 21 December 2006; accepted 27 March 2007; published online 16 May 2007)

The immersed boundary (IB) method has been widely applied to problems involving a moving elastic boundary that is immersed in fluid and interacting with it. Most of the previous applications of the IB method have involved a massless elastic boundary and used efficient Fourier transform methods for the numerical solutions. Extending the method to cover the case of a massive boundary has required spreading the boundary mass out onto the fluid grid and then solving the Navier-Stokes equations with a variable mass density. The variable mass density of this previous approach makes Fourier transform methods inapplicable, and requires a multigrid solver. Here we propose a new and simple way to give mass to the elastic boundary and show that the method can be applied to many problems for which the boundary mass is important. The method does not spread mass to the fluid grid, retains the use of Fourier transform methodology, and is easy to implement in the context of an existing IB method code for the massless case. Two verifications of the method are given. One is a numerical convergence study that shows that our numerical scheme is second-order accurate for a particular test problem. The other is direct comparison with experimental data of vortex-induced vibrations of a massive cylinder, which shows that the results obtained by the present method are quite comparable to the experimental data. © 2007 American Institute of Physics.

[DOI: [10.1063/1.2734674](https://doi.org/10.1063/1.2734674)]

I. INTRODUCTION

The purpose of this paper is to propose a simple way to extend the immersed boundary (IB) method to a situation in which an elastic moving boundary has mass that cannot be neglected and to show several examples of the application of this new methodology. In the examples chosen, the boundary mass plays a crucial dynamical role. This is in contrast to most previous applications of the IB method, in which the immersed boundary or structure was either thin and massless, or thick and neutrally buoyant in the ambient fluid.

What these two cases have in common is that the mass of the boundary or structure is equal to the mass of the fluid that it displaces. (In the case of a thin massless boundary, this is trivially true: the boundary has no mass and displaces no volume.) The present paper is concerned with the opposite case, in which the immersed boundary or structure has a *greater* mass than that of the fluid it replaces. For a thin boundary that displaces no volume, this simply means that the boundary has mass. For a thick immersed structure, it means that the density of the structure is greater than that of the ambient fluid.

The IB method was developed to study flow patterns around heart valves, and is a generally useful method for problems in which elastic materials interact with a viscous incompressible fluid. In the IB formulation, the fluid equations are written in Eulerian form, and the fluid variables (velocity and pressure) are stored on a fixed Cartesian computational lattice. The equations that describe the immersed boundary are written in Lagrangian form, and the boundary

variables (position and Lagrangian force density) are stored on a discretization of a moving curvilinear coordinate system attached to the immersed boundary. The Cartesian and curvilinear computational meshes do not coincide, and data are transferred from one to the other with the help of a regularized Dirac delta function. This whole approach has been applied successfully to problems of blood flow in the heart,¹⁻⁵ wave propagation in the cochlea,^{6,7} platelet aggregation during blood clotting,⁸ and several other problems.⁹⁻¹³

In these applications, however, the elastic boundary either has no mass in the case of a thin boundary or is neutrally buoyant in the case of a thick “boundary” like the muscular heart wall. We use the word “massless” to cover both cases, the meaning being that the presence of the immersed boundary or structure does not *add* any mass beyond what would be there anyway if the boundary or structure were replaced by fluid. The massless assumption appears in one version of the mathematical derivation of the IB method^{3,4} (see, however, Ref. 14) and can be applied when the mass of the moving boundary is too small to have a significant effect on the overall motions of the fluid and the elastic material. With the massless assumption, however, we cannot approach many other problems for which the boundary mass is important. Consider, for example, the simulation of a flapping filament in a flowing soap film as studied by Zhu and Peskin,^{15,16} who have shown that a massless filament does not flap at all, and that filament mass is essential for the sustained flapping that is seen not only in simulations with mass but also in laboratory experiments.¹⁷ Subsequent experiments (Zhang, personal communication) have shown as predicted that an extremely fine filament fails to flap under the same conditions in which flapping of a more massive

^{a)}Electronic mail: yk11@cims.nyu.edu

^{b)}Electronic mail: peskin@cims.nyu.edu

filament had been seen. Many other examples, in which the massless assumption is not reasonable, arise in aerodynamics. Since the density of air is so small, it is usually the case that elastic boundaries immersed in air have mass that cannot be neglected. Not only the inertial effect of the boundary mass but also the effect of gravity on the boundary mass are important. We study such examples in this paper.

In Ref. 15, the method used to handle boundary mass was to spread that mass out onto the fluid grid, in much the same manner as boundary force is conventionally applied to the fluid grid in the IB computations. When this is done, a variable density $\rho(\mathbf{x}, t)$ appears in the Navier-Stokes equations, and this complicates the numerical solver of those equations. Specifically, it makes Fourier methods inapplicable and requires that an iterative method such as MULTI-GRID be used instead.

Our current approach, which was introduced in the Ph.D. thesis of the first author,¹⁸ and which has already been applied to the aerodynamics of the vocal folds in Ref. 19, stays much closer to the IB method for the massless case. In particular, it does not spread mass to the fluid and so it requires the same constant-density Navier-Stokes solver as is used in the massless case. The key idea is to split the immersed boundary conceptually into two Lagrangian components. One component is massless and interacts with the fluid exactly as in a traditional IB computation; i.e., it moves at the local fluid velocity and applies force locally to the fluid. The other component carries mass. It is connected to the massless component by a system of stiff springs. It has no elasticity of its own, and does *not* interact with the fluid directly. Instead, the massive component of the immersed boundary moves as though in a vacuum, according to Newton's law of motion with the forces being those generated by the system of stiff springs that connects the massive component of the boundary to the massless component. If the effects of gravity are to be modeled, then the gravitational force is also applied directly to the massive component of the immersed boundary. Because the springs that connect the massive and massless components of the boundary are stiff, the two components stay close together, and the system comprised of these two components acts like an immersed boundary with mass. This may not be obvious, but will be demonstrated by a convergence study (Sec. IV) in which the spring stiffness increases as other computational parameters such as meshwidth and timestep are refined.

Even though the two boundaries (which we call massive and massless boundaries) are supposed to coincide, we allow them to separate by an amount that depends inversely on the stiffness of the spring connecting them. The stiffness parameter of this spring is analogous to a penalty parameter in a constrained optimization problem, which is why we call this new mass-handling idea the penalty immersed boundary (pIB) method. We shall see later that, in the limit of infinite spring stiffness, the two boundaries do indeed coincide, and moreover that the force applied by the massive boundary to its massless counterpart converges to a finite limit, which in fact is equal to the D'Alembert force associated with the mass of the immersed boundary.

In order to provide further verification that the pIB

method correctly gives mass to the immersed boundary, we simulate the vortex-induced vibration of an elastically mounted circular cylinder and compare the simulation results to experimental data (Sec. V). Since vortex-induced vibration has been extensively explored by many researchers experimentally and computationally, it is a good benchmark problem for the validation of our numerical scheme. We shall see that the results obtained by the pIB method are quite comparable to the existing data both qualitatively and quantitatively. Finally, we introduce two three-dimensional (3-D) problems as applications of the pIB method: windsock and flag (Sec. VI).

II. EQUATIONS OF MOTION

We begin by stating the mathematical formulation of the equations of the motion for a system comprised of a three-dimensional viscous incompressible fluid containing an immersed, elastic boundary with mass:¹⁴

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

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

$$\rho(\mathbf{x}, t) = \rho_0 + \int M(r, s) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] dr ds, \quad (3)$$

$$\mathbf{f}(\mathbf{x}, t) = \int \mathbf{F}(r, s, t) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] dr ds, \quad (4)$$

$$\frac{\partial \mathbf{X}}{\partial t}(r, s, t) = \mathbf{u}[\mathbf{X}(r, s, t), t] = \int \mathbf{u}(\mathbf{x}, t) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] d\mathbf{x}, \quad (5)$$

$$\mathbf{F}(r, s, t) = -\frac{\partial E}{\partial \mathbf{X}}[\mathbf{X}(r, s, t), t]. \quad (6)$$

Equations (1) and (2) are the familiar Navier-Stokes equations for a viscous incompressible fluid subject to an applied body force per unit volume $\mathbf{f}(\mathbf{x}, t)$. The density $\rho(\mathbf{x}, t)$ is the nonuniform mass density of the whole system (fluid + boundary) and μ is the constant fluid viscosity. The unknown functions in the fluid equations are the fluid velocity $\mathbf{u}(\mathbf{x}, t)$; the fluid pressure $p(\mathbf{x}, t)$; and the force per unit volume applied by the immersed boundary to the fluid $\mathbf{f}(\mathbf{x}, t)$, where $\mathbf{x} = (x, y, z)$ are fixed Cartesian coordinates, and t is the time.

Equation (6) is the immersed boundary equation that is written in Lagrangian form. The parameters r, s label a fixed material point. The unknown $\mathbf{X}(r, s, t)$ completely describes the motion of the immersed boundary, and also its spatial configuration at any given time. Note that $\mathbf{X}(r, s, t)$ represents a two-dimensional (2-D) surface in three-dimensional space, which is the case in all the 3-D applications in this paper. The function $\mathbf{F} = \mathbf{F}(r, s, t)$ is the force density applied by the immersed boundary to the fluid, in the sense that $\mathbf{F}(r, s, t) dr ds$ is the force applied to the fluid by the patch of immersed boundary $dr ds$. The elastic contribution to this

force density is given by the variational derivative $-\partial E/\partial \mathbf{X}$ of the elastic energy functional $E[\mathbf{X}(\cdot, \cdot, t)]$. This variational derivative is implicitly defined by

$$dE(t) = \int \frac{\partial E}{\partial \mathbf{X}}(r, s, t) \cdot d\mathbf{X}(r, s, t) dr ds, \quad (7)$$

where $d\mathbf{X}$ is a perturbation of the boundary configuration and dE is the resulting perturbation in the elastic energy of the boundary (to first order).

In this paper, the energy functional $E[\mathbf{X}(\cdot, \cdot, t)]$ is chosen a simple way. In a 2-D space, we consider an immersed boundary as a one-dimensional (1-D) rod and use two elasticities: one that resists stretching and compression, and another that resists bending:

$$E[\mathbf{X}(\cdot)] = \frac{1}{2} c_s \int \left(\left| \frac{\partial \mathbf{X}}{\partial s} \right| - 1 \right)^2 ds + \frac{1}{2} c_b \int \left| \frac{\partial^2 \mathbf{X}}{\partial s^2} \right|^2 ds, \quad (8)$$

where c_s and c_b are constants. Regarding a 2-D boundary immersed in a 3-D space as a shell or membrane, one could use quite general strain-stress relations.^{20,11} In all our 3-D simulations, however, we use a very simple law to generate the elastic force. The 2-D immersed boundaries are modeled as collections of 1-D rods. Suppose that $\mathbf{X}(r, s)$ is an immersed boundary. For a fixed r or s , $\mathbf{X}(r, s)$ is a curve that is parametrized as a function of s or r , respectively. Now each such curve can be assigned the same strain-stress relation as is used for the 1-D immersed boundary in our 2-D simulation, and all the forces generated from all these rods are summed up to make an elastic force $\mathbf{F}(r, s)$.

Equations (3)–(5), which we call interaction equations, involve the three-dimensional Dirac delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)\delta(z)$, which expresses the local character of the interaction. The constant ρ_0 is the fluid density, and $M(r, s)$ is the mass density of the immersed boundary in the sense that $M(r, s) dr ds$ is the mass of the patch $dr ds$. In (3) and (4), the delta function $\delta(\mathbf{x}) = \delta(x)\delta(y)\delta(z)$ is still three-dimensional, but there are only two integrals $dr ds$. As a result, $\rho(\mathbf{x}, t)$ and $\mathbf{f}(\mathbf{x}, t)$ are each singular, like a one-dimensional delta function with the singularity supported on the immersed boundary. Note that the integrals of $\rho(\mathbf{x}, t)$ and $\mathbf{f}(\mathbf{x}, t)$ over any finite volume are finite. In the case of $\rho(\mathbf{x}, t)$, for example, we may integrate over an arbitrary region Ω to find

$$\begin{aligned} \int_{\Omega} \rho(\mathbf{x}, t) d\mathbf{x} &= \rho_0 \text{Vol}(\Omega) \\ &+ \int \int M(r, s) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] dr ds d\mathbf{x} \\ &= \rho_0 \text{Vol}(\Omega) + \int_{\mathbf{X}(r, s) \in \Omega} M(r, s) dr ds, \end{aligned} \quad (9)$$

which represents the sum of the mass of the immersed material and the fluid that are contained within a fixed space region Ω . Similarly, the integral of $\mathbf{f}(\mathbf{x}, t)$ over such a volume is the total force applied to the fluid by the part of the immersed material contained within that volume.

Equation (5) is the equation of motion of the immersed elastic boundary. It is a consequence of the no-slip condition that holds in a viscous fluid. This is rewritten in terms of the Dirac delta function in the second form of Eq. (5). We do so in order to expose a certain symmetry with Eq. (4), in which the force generated by the immersed boundary is re-expressed as a body force acting on the fluid. This symmetry is important in the construction of our numerical scheme. Note, however, that the integral in Eq. (5) is a triple integral ($dx dy dz$), unlike the integral in Eqs. (3) and (4). Thus, $\partial \mathbf{X}/\partial t$ is finite, unlike ρ and $\mathbf{f}$.

Note that Zhu and Peskin^{15,16} use this system of equations (1)–(6) and solve it directly. This introduces a nonuniform mass density $\rho(\mathbf{x}, t)$ [see Eq. (3)]. We avoid this in the present paper by the use of the formulation described below.

First, we substitute the density defined in Eq. (3) into Eq. (1), and separate the left-hand side of (1) into two terms: one involving the constant density ρ_0 and the other containing the singular part of $\rho(\mathbf{x}, t)$ that comes from the immersed boundary. Then,

$$\rho_0 \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f} - \mathbf{f}_D, \quad (10)$$

$$\mathbf{f}_D(\mathbf{x}, t) = \int \frac{D\mathbf{u}(\mathbf{x}, t)}{Dt} M(r, s) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] dr ds, \quad (11)$$

where D/Dt is the material derivative. Using the relation

$$\frac{\partial^2 \mathbf{X}}{\partial t^2}(r, s, t) = \frac{D\mathbf{u}}{Dt}[\mathbf{X}(r, s, t), t], \quad (12)$$

we can interpret $\mathbf{f}_D(\mathbf{x}, t)$ in Eq. (11) as an Eulerian body force obtained by using the Dirac delta function to transform the Lagrangian expression

$$\mathbf{F}_D(r, s, t) = M(r, s) \frac{\partial^2 \mathbf{X}}{\partial t^2}(r, s, t), \quad (13)$$

which is known as the D'Alembert force. In a numerical procedure, one might try to evaluate $\mathbf{F}_D(r, s, t)$ by applying an explicit time difference to $\partial \mathbf{X}/\partial t$ as evaluated by Eq. (5) at the current timestep and at the previous timestep. $\mathbf{F}_D$ could then be applied to the fluid grid like any other immersed boundary force by using the Dirac delta function. This method was proposed by Wang and is compared to other approaches in Ref. 21. The conclusion of that study was that the direct application of the D'Alembert force in an explicit scheme is only numerically stable when the boundary mass is very small. (Use of the D'Alembert force in an *implicit* context has recently been proposed by Mori.) Here we use a different way of approximating $\mathbf{F}_D$, as described in the following.

We modify the equations discussed so far to make what we call the penalty immersed boundary (pIB) method.¹⁸ To do this, we introduce two boundaries immersed in a 3-D fluid; one is a massive boundary $\mathbf{Y}(r, s, t)$ having all the mass of the elastic immersed boundary, and the other is a massless boundary $\mathbf{X}(r, s, t)$. The massive boundary $\mathbf{Y}(r, s, t)$ has mass density $M(r, s)$ but does not interact with the surrounding fluid directly. On the other hand, the massless boundary

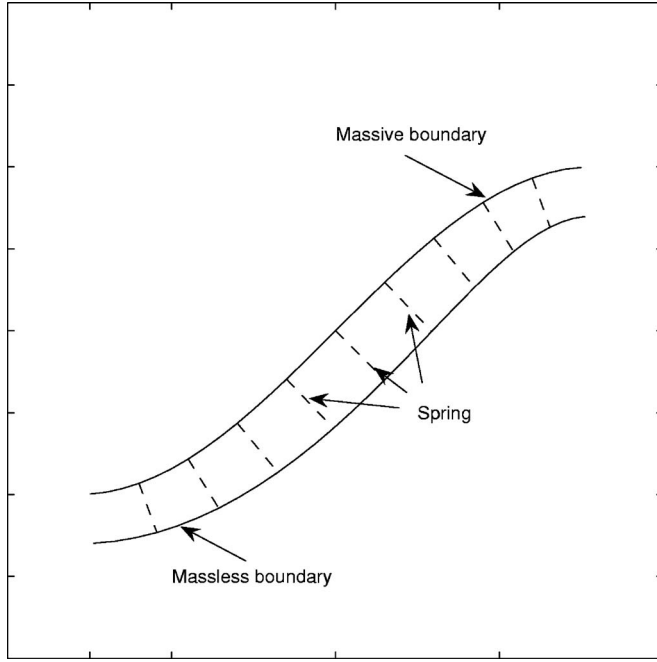


FIG. 1. Massive and massless boundaries are linked together by a collection of a very stiff springs with rest length zero.

$\mathbf{X}(r, s, t)$ interacts with the fluid: it moves with the local fluid velocity and exerts an elastic force locally on the fluid. Both the massless and massive boundaries are supposed to represent the same material surface. If a pair of corresponding boundary points moves apart, a restoring force appears in order to make them move close together (see Fig. 1). The restoring force $\mathbf{F}_K$ acting on the massless boundary is

$$\mathbf{F}_K(r, s, t) = K[\mathbf{Y}(r, s, t) - \mathbf{X}(r, s, t)], \quad (14)$$

where K is a constant chosen so that $K \gg \tilde{F}/L$, where $\tilde{F}$ is a characteristic force density and L is a characteristic length of the problem in question.

Now the massive boundary $\mathbf{Y}(r, s, t)$ moves according to the following equation:

$$M(r, s) \frac{\partial^2 \mathbf{Y}(r, s, t)}{\partial t^2} = -\mathbf{F}_K(r, s, t). \quad (15)$$

Note that the only force acting on the massive boundary is the reaction force $-\mathbf{F}_K$ (we shall add to this a gravitational force later), and that the massive boundary does not interact with the fluid directly. On the other end of the spring, the massless boundary $\mathbf{X}(r, s, t)$ applies to the nearby fluid both its elastic force $\mathbf{F}_E = -\partial E / \partial \mathbf{X}$ and the spring force $\mathbf{F}_K$.

Consider the case in which K in (14) goes to infinity. The massive boundary $\mathbf{Y}(r, s, t)$, then, approaches the massless boundary $\mathbf{X}(r, s, t)$, in which case, using $\mathbf{X}(r, s, t)$ only, we can see that $\mathbf{F}_K(r, s, t)$ is the same as $-\mathbf{F}_D(r, s, t)$ of Eq. (13). Note that, in practice, K cannot be infinite, but we can keep $\mathbf{X}(r, s, t)$ and $\mathbf{Y}(r, s, t)$ as close as we like by choosing K sufficiently large. For two different quantitative demonstrations of this fact, see Secs. IV and V.

With a large constant K (not infinity), we can use Eqs. (14) and (15) instead of Eq. (13) (D'Alembert force) and derive the equations of motion for the PIB method as follows:

$$\rho_0 \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \mu \nabla^2 \mathbf{u} + \mathbf{f}, \quad (16)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (17)$$

$$\mathbf{f}(\mathbf{x}, t) = \int \mathbf{F}(r, s, t) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] dr ds, \quad (18)$$

$$\frac{\partial \mathbf{X}}{\partial t}(r, s, t) = \mathbf{u}[\mathbf{X}(r, s, t), t] = \int \mathbf{u}(\mathbf{x}, t) \delta[\mathbf{x} - \mathbf{X}(r, s, t)] d\mathbf{x}, \quad (19)$$

$$\mathbf{F} = \mathbf{F}_E + \mathbf{F}_K, \quad (20)$$

$$\mathbf{F}_E = -\frac{\partial E}{\partial \mathbf{X}}, \quad (21)$$

$$\mathbf{F}_K(r, s, t) = K[\mathbf{Y}(r, s, t) - \mathbf{X}(r, s, t)], \quad (22)$$

$$M(r, s) \frac{\partial^2 \mathbf{Y}}{\partial t^2} = -\mathbf{F}_K(r, s, t) - M(r, s) g \mathbf{e}_3, \quad (23)$$

where g is the gravitational acceleration and $\mathbf{e}_3$ is a unit vector in the vertical direction (against gravity). We have here added a gravitational term on the right-hand side of Eq. (23) in order to allow for the influence of gravity on the massive immersed boundary. In Eq. (16), the symbol p now denotes (pressure + $\rho g z$). Thus, $-\nabla p = -\nabla(\text{pressure}) - \rho g \mathbf{e}_3$, and the gravitational force acting directly on the fluid is automatically included in Eq. (16) in this manner.

Note that there is no need in the above formulation to evaluate explicitly the force of the fluid on the immersed boundary. This would be complicated to do, since it would require evaluation of the fluid stress tensor at a moving interface of complicated geometry. What we evaluate instead is the force of the immersed boundary on the fluid (which is equal and opposite to that of the fluid on the immersed boundary, by Newton's law of equal and opposite forces). We evaluate the force of the boundary on the fluid as the sum of the elastic force and the penalty force [see Eqs. (20)–(22)]. This sidesteps the difficulty of explicitly evaluating the fluid stress tensor on the two sides of the boundary.

III. NUMERICAL IMPLEMENTATION OF THE PIB METHOD

We now describe a formally second-order IB method to solve the equations of motion.^{2,22} The word “formally” is used as a reminder that this scheme is only second-order accurate for problems with sufficiently smooth solutions. Even though our solutions are not smooth (the velocity has jumps in derivative across the immersed boundary), the use of the formally second-order method results in improved accuracy (see Ref. 22).

The specific formally second-order method that we use is the one described in Ref. 2 generalized here to take into account the massive boundary that is linked to the immersed elastic boundary by stiff springs. In this method, each timestep proceeds in two substeps, which are called the preliminary and final substeps. In the preliminary substep, we get data at time level $n + \frac{1}{2}$ from data at time level n by a first-order accurate method. Then the final substep starts again at time level n and proceeds to time level $n + 1$ by a second-order accurate method. This Runge-Kutta framework allows the second-order accuracy of the final substep to be the overall accuracy of the scheme.²³

We use a superscript to denote the time level. Thus, $\mathbf{X}^n(r, s)$ is shorthand for $\mathbf{X}(r, s, n\Delta t)$, where Δt is the duration of the timestep, and similarly for all other variables. Our goal is to compute updated $\mathbf{u}^{n+1}$, $\mathbf{V}^{n+1}$, where $\mathbf{V} = d\mathbf{Y}/dt$, $\mathbf{X}^{n+1}$, and $\mathbf{Y}^{n+1}$ from given data $\mathbf{u}^n$, $\mathbf{V}^n$, $\mathbf{X}^n$, and $\mathbf{Y}^n$.

Before describing how this is done, we have to say a few words about the spatial discretization. There are two such discretizations: one for the fluid and the other for the elastic boundary.

The grid on which the fluid variables are defined is a fixed uniform cubic lattice of meshwidth $h = \Delta x_1 = \Delta x_2 = \Delta x_3$. Now we introduce the central difference operator D_i , defined for $i = 1, 2, 3$, as follows:

$$(D_i \phi)(\mathbf{x}) = \frac{\phi(\mathbf{x} + h\mathbf{e}^i) - \phi(\mathbf{x} - h\mathbf{e}^i)}{2h}, \quad (24)$$

where $\mathbf{e}^i$ is the unit vector in the i th coordinate direction. As the notation suggests, the difference operator in i th direction D_i corresponds to the i th component of the differential operator ∇ . Thus, $\mathbf{D}_p$ will be the discrete gradient of p , and $\mathbf{D} \cdot \mathbf{u}$ will be the discrete divergence of $\mathbf{u}$.

We shall also make use of the central difference Laplacian L :

$$(L\phi)(\mathbf{x}) = \sum_{i=1}^3 \frac{\phi(\mathbf{x} + h\mathbf{e}^i) + \phi(\mathbf{x} - h\mathbf{e}^i) - 2\phi(\mathbf{x})}{h^2}. \quad (25)$$

The fluid mesh and the elastic boundary mesh defined below are connected by a smoothed approximation to the Dirac delta function. It is denoted δ_h and is of the following form:

$$\delta_h(\mathbf{x}) = h^{-3} \phi\left(\frac{x_1}{h}\right) \phi\left(\frac{x_2}{h}\right) \phi\left(\frac{x_3}{h}\right), \quad (26)$$

where $\mathbf{x} = (x_1, x_2, x_3)$, and the function ϕ is given by

$$\phi(r) = \begin{cases} \frac{3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}}{8}, & \text{if } |r| < 1, \\ \frac{5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}}{8}, & \text{if } 1 \leq |r| < 2, \\ 0, & \text{if } 2 \leq |r|. \end{cases} \quad (27)$$

The motivation and derivation of this particular choice are discussed in Refs. 3 and 4.

We are now ready to describe a typical timestep of the numerical scheme. The preliminary substep, which goes from time level n to $n + \frac{1}{2}$, proceeds as follows:

First, update the position of the massless boundary $\mathbf{X}^{n+1/2}(r, s)$:

$$\frac{\mathbf{X}^{n+1/2} - \mathbf{X}^n}{\Delta t/2} = \sum_{\mathbf{x}} \mathbf{u}^n(\mathbf{x}) \delta_h[\mathbf{x} - \mathbf{X}^n(r, s)] h^3. \quad (28)$$

Here and throughout the paper, $\Sigma_{\mathbf{x}}$ denotes the sum over the cubic lattice in physical space on which the fluid variables are defined. Similarly, $\Sigma_{r,s}$ will denote the sum over rectangular lattice in (r, s) space on which the two boundary positions $\mathbf{X}$, $\mathbf{Y}$, the force density $\mathbf{F}$, and the mass density M are defined.

The key to generalizing the formally second-order method to the massive case is to handle the massive boundary in a manner that closely parallels the numerical treatment of the immersed boundary itself. Thus, we update $\mathbf{Y}^{n+1/2}$ in a manner similar to $\mathbf{X}^{n+1/2}$:

$$\frac{\mathbf{Y}^{n+1/2} - \mathbf{Y}^n}{\Delta t/2} = \mathbf{V}^n, \quad (29)$$

where $\mathbf{V}^n$ is the velocity vector of the massive boundary (the known value at time $n\Delta t$, like $\mathbf{u}^n$).

Next, calculate the force density $\mathbf{F}^{n+1/2}$, which is the sum of two parts; one is an elastic force and the other is from the spring linked between massless and massive boundaries:

$$\mathbf{F}_K^{n+1/2} = K(\mathbf{Y}^{n+1/2} - \mathbf{X}^{n+1/2}), \quad (30)$$

$$\mathbf{F}^{n+1/2} = -\frac{\partial E}{\partial \mathbf{X}}(\mathbf{X}^{n+1/2}) + \mathbf{F}_K^{n+1/2}. \quad (31)$$

These are discretizations of Eqs. (20)–(22), respectively.

Now we have to change this elastic force density defined on Lagrangian grid points into the force at Eulerian spatial grid points to be applied in the Navier-Stokes equations. This is done by a discretization of Eq. (18),

$$\mathbf{f}^{n+1/2}(\mathbf{x}) = \sum_{r,s} \mathbf{F}^{n+1/2}(r, s) \delta_h[\mathbf{x} - \mathbf{X}^{n+1/2}(r, s)] \Delta s \Delta r. \quad (32)$$

With $\mathbf{f}^{n+1/2}$ in hand, we can turn to solving the Navier-Stokes equations:

$$\begin{aligned} \rho \left[\frac{u_i^{n+1/2} - u_i^n}{\Delta t/2} + \frac{1}{2} [\mathbf{u} \cdot \mathbf{D} u_i + \mathbf{D} \cdot (\mathbf{u} u_i)]^n \right] + D_i \tilde{p}^{n+1/2} \\ = \mu L u_i^{n+1/2} + f_i^{n+1/2}, \end{aligned} \quad (33)$$

for $i = 1, 2, 3$, and

$$\mathbf{D} \cdot \mathbf{u}^{n+1/2} = 0. \quad (34)$$

The notation $\tilde{p}^{n+1/2}$ is used to distinguish this pressure from the one that is computed when solving Eqs. (38) and (39), below. Note that in Eqs. (33) and (34) there are four unknowns per grid point, which are $u_i^{n+1/2}$, $i = 1, 2, 3$, and $\tilde{p}^{n+1/2}$, and that these unknowns enter into these equations linearly. The four unknowns at different grid points are coupled by the difference equations. Because these equations have constant coefficients, they may be uncoupled by the use of the

discrete Fourier transform [implemented by the fast Fourier transform (FFT) algorithm] into separate systems of four equations in four unknowns, one such 4×4 system at each grid point in Fourier space. If the coefficients were not constant, the discrete Fourier transform could still be applied, but it would not succeed in uncoupling the system in the manner described above.

Again we have to calculate the velocity $\mathbf{V}^{n+1/2}$ of the massive boundary in the same fashion:

$$M \left(\frac{\mathbf{V}^{n+1/2} - \mathbf{V}^n}{\Delta t/2} \right) = -\mathbf{F}_K^{n+1/2} - M g \mathbf{e}_3. \quad (35)$$

This is nothing more than Euler's method for Eq. (23) and completes the preliminary substep.

The final substep is the use of $\mathbf{u}^{n+1/2}$, $\mathbf{V}^{n+1/2}$, $\mathbf{X}^{n+1/2}$, and $\mathbf{Y}^{n+1/2}$ obtained in the preliminary substep to compute the corresponding quantities at time level $n+1$ by the second-order-accurate midpoint rule.

First, using the fluid velocity $\mathbf{u}^{n+1/2}$ and massive boundary velocity $\mathbf{V}^{n+1/2}$, we can find the massless boundary configuration $\mathbf{X}^{n+1}$ and massive boundary position $\mathbf{Y}^{n+1}$:

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\Delta t} = \sum_{\mathbf{x}} \mathbf{u}^{n+1/2}(\mathbf{x}) \delta_h[\mathbf{x} - \mathbf{X}^{n+1/2}(r, s)] h^3, \quad (36)$$

$$\frac{\mathbf{Y}^{n+1} - \mathbf{Y}^n}{\Delta t} = \mathbf{V}^{n+1/2}. \quad (37)$$

The last thing that we have to do is to update the fluid velocity data:

$$\begin{aligned} \rho \left[\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{2} [\mathbf{u} \cdot \mathbf{D} u_i + \mathbf{D} \cdot (\mathbf{u} u_i)]^{n+1/2} \right] + D_i p^{n+1/2} \\ = \frac{1}{2} \mu L (u_i^{n+1} + u_i^n) + f_i^{n+1/2}, \end{aligned} \quad (38)$$

for $i=1, 2, 3$, and

$$\mathbf{D} \cdot \mathbf{u}^{n+1} = 0. \quad (39)$$

Just as in the case of Eqs. (33) and (34), Eqs. (38) and (39) are a constant-coefficient linear system in the unknowns u_i^{n+1} and $p^{n+1/2}$. This linear system is solved by fast Fourier transform. For the velocity of the massive boundary, we have

$$M \left(\frac{\mathbf{V}^{n+1} - \mathbf{V}^n}{\Delta t} \right) = -\mathbf{F}_K^{n+1/2} - M g \mathbf{e}_3. \quad (40)$$

Since we have now computed $\mathbf{u}^{n+1}$, $\mathbf{V}^{n+1}$, $\mathbf{X}^{n+1}$, and $\mathbf{Y}^{n+1}$, the timestep is complete.

To complete the description of the numerical pIB method, we need to explain the boundary conditions imposed along the edges of our computational domain. In all cases, we use periodic boundary conditions that are very convenient for the solution of the linear systems of Eqs. (33) and (34) and Eqs. (38) and (39) by fast Fourier transform, but we also break the periodicity in various ways depending on the application. Note that the ways in which we break the periodicity, explained in more detail below, still allows for the use of FFT, since we do so not by changing boundary conditions per se but instead by applying forces.

In all of the applications considered here, we need to impose an “inflow velocity.” This is done on two (or more generally an even number of) adjacent parallel grid planes (or grid lines in the 2-D case). It is very important to use two adjacent planes rather than just one, in order to avoid the spatial oscillations that would otherwise propagate throughout the domain via the central-difference structure of our numerical scheme. Although we typically choose either the first two or last two grid planes in some coordinate direction, this is of no fundamental significance, since the underlying domain is periodic and all grid points are created equal. One consequence of the underlying periodicity that should be noted is that when we specify the inflow velocity we are also specifying the outflow velocity. This needs to be done far enough downstream to avoid undue influence on the flow in the neighborhood of the immersed boundaries that are of interest. Specifying the inflow (and outflow) velocity in the manner described below effectively breaks the periodicity in the sense that it prevents vortices that are shed downstream from reappearing upstream.

The method used to drive the “inflow velocity” in this paper is to apply an external force per unit volume equal to

$$\mathbf{f}_0(\mathbf{x}, t) = \begin{cases} \alpha_0 [\mathbf{u}_0(t) - \mathbf{u}(\mathbf{x}, t)], & \mathbf{x} \in \Omega_0, \\ 0, & \text{otherwise,} \end{cases} \quad (41)$$

where Ω_0 is the region in which we want to control the velocity, $\mathbf{u}_0(t)$ is the desired velocity in that region, and α_0 is a constant. When α_0 is large, the grid velocity is driven rapidly towards $\mathbf{u}_0(t)$ within Ω_0 . In particular, the approach of $\mathbf{u}(\mathbf{x}, t)$ to $\mathbf{u}_0(t)$ within Ω_0 will occur approximately exponentially with a time constant equal to ρ/α_0 .

Another type of boundary condition one might want to impose is to put no-slip walls on faces of the domain. The method that we shall describe for this purpose can indeed be used to put no-slip walls of any shape anywhere within the domain; planar walls along the faces are just a special case.

We create fixed no-slip boundaries by laying out an array of “target points” to mark their desired positions. To avoid leaks, the target points should be spaced about half a mesh-width apart (or closer). These target points are fixed in position and do not interact with fluid. Each of them is connected, however, by a stiff spring to a massless immersed boundary point that moves at the local fluid velocity and applies the force generated by the stiff spring locally to the fluid. This provides a feedback mechanism for computing the boundary force needed to enforce the no-slip condition.

Note that the “target point” method we have just described is essentially a special case of the pIB method of this paper. In the pIB method, massless immersed boundary points that move at the local fluid velocity and apply force locally to the fluid are linked by stiff springs to mass points that do not interact with the fluid directly but move according to Newton's law in a vacuum. In the target point method, the situation is exactly the same except that the mass points no longer move, so they are renamed target points. The accuracy of the target point method for modeling rigid walls should therefore be similar to that of the pIB method for modeling immersed boundaries with mass; indeed, we would

expect better accuracy in the fixed boundary case because of the simplification that the positions of the target points are known. Empirical evidence of the convergence of the pIB method is presented in Secs. IV and V.

The various methods that we have discussed above for breaking the periodicity of the domain are special cases which hint at the following general remark: In the context of an immersed boundary method, it is not much of a restriction to use periodic boundary conditions for the domain occupied by the fluid. This is because one can immerse within that domain whatever dynamic or static geometry the application demands.

Finally, we remark that there is no fundamental requirement of using a periodic domain in conjunction with the pIB method. Any uniform grid fluid solver can be used, with whatever boundary conditions that solver can accommodate. We have emphasized the periodic case because of the efficiency that follows from the use of the FFT algorithm for solving the linear systems that arise in our numerical scheme.

IV. NUMERICAL VERIFICATION OF THE PIB METHOD

In this section, we validate the pIB method in two ways:

- (1) By a convergence study that shows that the pIB method is second-order accurate when applied to a problem involving an immersed elastic shell that has a greater mass density than the surrounding fluid.
- (2) By verifying directly from the results of that convergence study that the distance between a massless immersed boundary point and its massive twin converges to zero, and moreover that the force density applied to the immersed boundary by the penalty springs converges to a finite limit, as the penalty parameter approaches infinity.

The kind of test problem that we use here was introduced in Ref. 24. It involves an immersed, finite-thickness elastic shell comprised of circumferential fibers only. The surprising result of Ref. 24 was that finite thickness provides enough regularity for a formally second-order-accurate IB method to achieve *actual* second-order accuracy, despite the possibly sudden change in material properties that occurs at the shell-fluid interfaces. In Ref. 24, two cases were considered. In one of these, the stiffness of the shell tapered smoothly to zero at the shell-fluid interfaces, but in the other case the stiffness was constant, thus providing the greater numerical challenge of resolving an abrupt change in material properties at the borders of the shell. In both cases considered in Ref. 24, though, the shells were neutrally buoyant, and therefore the mass density of the whole system was uniform.

Here, we consider the corresponding two problems for elastic shells that are not neutrally buoyant, so that the mass density is greater in the region occupied by the shell than in the region occupied by the fluid. In one case, we taper both the stiffness and mass density [in the manner described below; see Eq. (46)] to “blur” the boundary between the shell and the surrounding fluid, but in the other (more difficult)

case, we let the stiffness and mass density of the shell be constant, so that there is a sharp discontinuity in both of these material properties as one passes from shell to fluid.

A. The test problem

We consider the unit square of $[01] \times [01]$ filled with an incompressible fluid in which an initially elliptic flexible elastic shell is immersed. (Throughout this section we use dimensionless variables.) The periodic domain is described using a fixed $N \times N$ Cartesian grid with uniform meshwidths $h = \Delta x = \Delta y$, where $h = 1/N$. Similarly, for the shell, we take the curvilinear coordinate space $[01] \times [01]$ and employ a fixed $N_r \times N_s$ computational lattice in this space, where $N_r = 3N/8$ and $N_s = 75N/16$. Let $\Delta r = 1/N_r$ and $\Delta s = 1/N_s$. The nodes of the curvilinear mesh are the points

$$(r, s) = (r_0, s_0) + (m\Delta r, n\Delta s) = \left(\frac{\Delta r}{2}, \frac{\Delta s}{2} \right) + (m\Delta r, n\Delta s), \quad (42)$$

where $m \in 0, 1, \dots, N_r - 1$ and $n \in 0, 1, \dots, N_s - 1$. Here, the shift by $\Delta r/2$ avoids having fibers at the exact edges of the shell, while the shift by $\Delta s/2$ is for notational consistency only.

The initial shape of the elastic shell is given by the mapping

$$\mathbf{X}(r, s, 0) = \left(\frac{1}{2}, \frac{1}{2} \right) + \left\{ \left[\alpha + \gamma \left(r - \frac{1}{2} \right) \right] \cos(2\pi s), \right. \\ \left. \left[\beta + \gamma \left(r - \frac{1}{2} \right) \right] \sin(2\pi s) \right\}, \quad (43)$$

with $\alpha = 0.2$, $\beta = 0.25$, and $\gamma = 0.0625$. The initial configuration of the entire structure is a “thick” elliptical shell, and $\mathbf{X}(r, s, t)$ is the position at time t of the same material point whose coordinates at $t = 0$ are given by $\mathbf{X}(r, s, 0)$ (see Fig. 2). The material curves $r = \text{constant}$ will be called “fibers,” and will have the standard form of the equation of an ellipse with s as a parameter. Note that s is not equal to arc length. The arc length associated with ds is given by $\sqrt{|\partial \mathbf{X} / \partial s|^2} ds$. Our shell is anisotropic and can only generate stress in the (circumferential) fiber direction. This is not a fundamental limitation. To model a general incompressible elastic material in the context of the IB method, one needs to discretize the elastic energy function and then evaluate the elastic force that needs to be applied to the fluid at each immersed boundary points by taking minus the gradient of the discretized elastic energy (see Ref. 14). Note that the shell considered here is made of *incompressible* elastic material, since the incompressibility constraint holds everywhere. The value of γ determines the thickness of the shell and is chosen so that the thickness of the initial shell is approximately four Cartesian mesh widths on the coarsest grid considered.

The force generated by the elastic fibers that comprise the shell is taken to be of the form

$$\mathbf{F}(r, s, t) = A(r) \frac{\partial^2 \mathbf{X}}{\partial s^2}(r, s, t), \quad (44)$$

and the mass density is chosen as

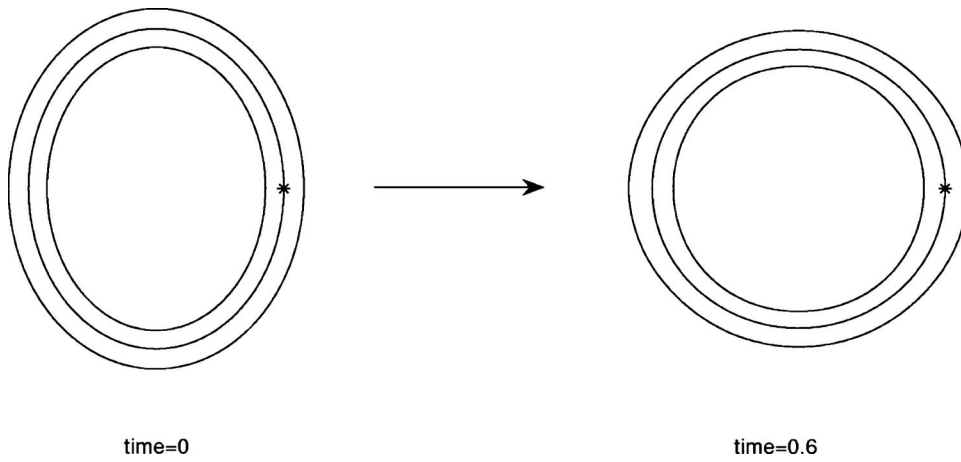


FIG. 2. Motion of a massive elastic shell. Three selected fibers with $r = \Delta r/2$, $1/2$, and $1 - \Delta r/2$ are shown at $t=0$ and $t=0.6$. The computational shell is comprised of many such fibers immersed in viscous incompressible fluid.

$$M(r, s) = 0.5A(r), \quad (45)$$

where the function $A(r)$ that appears in both of the above formulae may be used to provide the tapered material properties mentioned above by setting

$$A(r) = 1 + \sin(2\pi r - \pi/2) \quad (46)$$

(recall that $0 \leq r \leq 1$), or alternatively where constant material properties may be chosen instead by setting

$$A(r) = 1. \quad (47)$$

The case of constant material properties is, of course, of greater physical interest, but the tapered case is included to see how the method behaves when the stated problem involves only smooth transitions of material properties. Both cases will be considered below. It seems reasonable to make the penalty stiffness proportional to the mass density, so we replace Eq. (22) by

$$\mathbf{F}_K(r, s, t) = KA(r)[\mathbf{Y}(r, s, t) - \mathbf{X}(r, s, t)] \quad (48)$$

with the same $A(r)$ that may be tapered, or not, as above, and where $K = 10 \times N^2$, where the computational mesh of the Eulerian (fluid) variables is $N \times N$. The reason for this choice of K will be discussed below. Note in particular that the penalty parameter increases rapidly as the computational mesh is refined.

Now we choose the mesh sizes of the domain $N = 64, 128, 256, 512$, and 1024 . Recall that $h = 1/N$ and that Δr and Δs are also proportional to $1/N$. For each N , we set $\Delta t = 0.04/N$. That is, when we refine all the meshwidths by a factor of 2, the timestep Δt is also reduced by the same factor. The numerical constant 0.04 is, of course, problem dependent, and has been chosen as large as possible without causing numerical instability in this particular problem. Although this is sufficient for stability within the range of parameters tested, the explicit computation of the boundary force may impose a more severe stability restriction once the parameters have been sufficiently refined (see Ref. 25). Note that the penalty parameter K increases as the meshwidth and timestep are refined. The above manner of increasing it has the property that $K\Delta t^2$ is constant. The fact that this relationship between K and Δt preserves the numerical stability of

the scheme despite the increase of K was proved by Lai²⁶ with a simple 1-D model and was observed in our 2-D simulations.

The fluid parameters of our convergence study are a density of $\rho = 1.0$ and a viscosity of either $\mu = 0.005$ or $\mu = 0.0005$. In the first case the Reynolds number of the resulting flow is approximately 100, and in the second case it is approximately 1000.

B. Result of convergence test

In this section we restrict consideration to problems in two space dimensions, since that makes it feasible to refine the computational grid sufficiently to study convergence. Since we do not know the exact solution of the problem, the estimation of a convergence ratio requires three numerical solutions for three consecutive values of N . Letting ψ be a grid function defined on the Cartesian grid, the discrete L^2 norm $\|\psi\|_2$ is given by $\|\psi\|_2 = (\sum_{i,j} \psi_{i,j}^2 h^2)^{1/2}$. For a vector-valued mesh function $\mathbf{u} = (u_1, u_2)$, we have $\|\mathbf{u}\|_2 = (\|u_1\|_2^2 + \|u_2\|_2^2)^{1/2}$.

For a computed quantity $q_N(t)$ depending on the mesh number N at time t , the convergence rate in L^2 norm is defined as the following:

$$r[q_N(t)] = \log_2 \left(\frac{\|q_N(t) - P_N^{2N} q_{2N}(t)\|_2}{\|q_{2N}(t) - P_{2N}^{4N} q_{4N}(t)\|_2} \right), \quad (49)$$

where P_N^{2N} is the restriction (a projection operator) of the function defined on the $2N \times 2N$ grid to the set of points that this grid has in common with the $N \times N$ grid.

Figure 3 shows the convergence rates $r[q_N(t)]$ as functions of time in three different cases with two different fluid viscosities. The quantity $q_N(t)$ is the velocity vector $\mathbf{u}$ computed on the $N \times N$ grid at time t . The first case has no mass of the elastic fibers (first row), and the second and third ones have the massive fibers. While the second one uses constants M and K without tapering (second row), i.e., with $A(r) = 1$ in the notation introduced above, the third one uses the tapered M and K (third row), i.e., with $A(r) = 1 + \sin(2\pi r - \pi/2)$. The left and right columns are for the cases with the two different fluid viscosities; i.e., $\mu = 0.005$ and $\mu = 0.0005$, respectively. In each panel, we have two curves corresponding to the cases

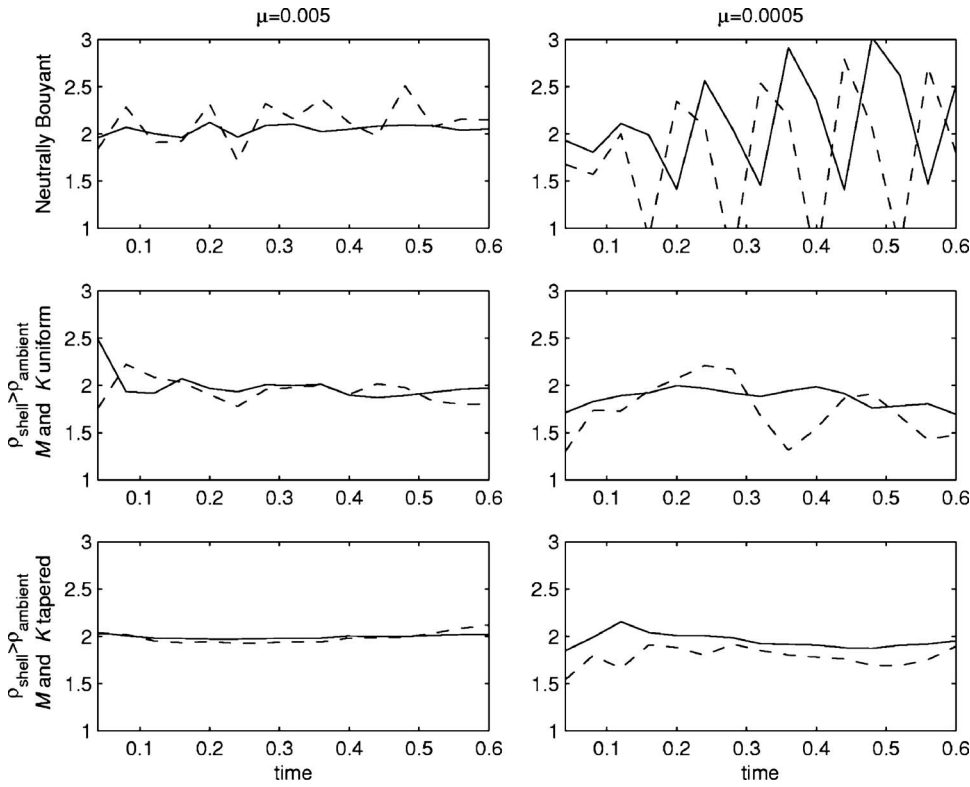


FIG. 3. Convergence rate of the computed velocity field $\mathbf{u}(\mathbf{x}, t)$ of a fluid containing an immersed elastic shell. The convergence rate (defined in the text) is plotted as a function of time for two different viscosities (columns) in three different cases (rows). In all panels, the dashed line is the convergence rate obtained using the grids $N = 128, 256, 512$, and the solid line is the rate obtained with $N = 256, 512, 1024$. The dimensionless time interval over which the convergence rate is plotted is sufficient for at least one period of the (damped) oscillation of the immersed elastic shell. At the higher viscosity (left column), the convergence rate is near 2 (second-order accuracy) in all cases. At the lower viscosity, the better of the two resolutions (solid line) is needed to get second-order accuracy.

$N=128$ and $N=256$. (Recall, however, that the convergence rate for $N=128$ is obtained from the solutions computed on $N=128, 256, 512$, and similarly that the convergence rate for $N=256$ is obtained from the solutions computed on $N=256, 512, 1024$.)

From the left column ($\mu=0.005$), we can see that the actual convergence rates are roughly equal to 2, which implies that the scheme is second-order accurate. This accuracy holds in both the neutrally buoyant and non-neutrally buoyant cases, regardless of whether the mass density and elasticity are uniformly distributed across the shell or tapered to zero at the borders of the shell. The right column ($\mu=0.0005$) shows the expected result that lower viscosity requires finer grids in order to see the second-order accuracy of the IB or pIB method. The convergence rates obtained with $N=128, 256, 512$ (dashed lines) do not quite show second-order accuracy, but the convergence rates obtained with $N=256, 512, 1024$ (solid lines) do indeed show second-order accuracy, the only anomaly being in the neutrally buoyant case (top row) where the convergence rate seems to oscillate about 2 as a function of time instead of being steadily equal to 2 as in the more massive cases or in the more viscous cases. Overall, the results obtained here are similar to those of Ref. 24, the difference being that the pIB method allows us to obtain such results even when the immersed elastic shell has a greater density than the surrounding fluid.

We now consider results that show directly that the penalty springs of the pIB method accomplish what they are designed to do. The top panel of Fig. 4 shows the maximum distance, plotted as a function of time, between any massless immersed boundary point and its massive twin. Recall that these points are linked by a spring-like force with stiffness

that increases as $O(N^2)$. The maximum distance is plotted on a logarithmic scale for each of the cases $N=128, 256, 512$. Clearly, the discrepancy between the location of an immersed boundary point and that of its massive twin is converging to zero as $N \rightarrow \infty$. The bottom panel of Fig. 4 shows the even more striking result that the force generated by the penalty springs is converging to a finite limit as $N \rightarrow \infty$, despite the blowup in the penalty spring stiffness.

Overall, Fig. 4 provides empirical evidence that the pIB method works as advertised. One certainly might worry about the high-frequency oscillation introduced into the system by the large value of the penalty stiffness K : In a non-linear system, such a spurious high-frequency oscillation might interact with the frequency modes of interest in unpredictable ways. However, the results in Fig. 4 indicate that this spurious oscillation is not only of high frequency; it is also of low amplitude—in fact, of an amplitude that approaches zero as the numerical parameters are refined. The combination of these results with the convergence rate data shown in Fig. 3 strongly suggests that the pIB method is indeed convergent.

To illustrate the influence of mass on the behavior of an immersed elastic boundary, the top panel of Fig. 5 compares the neutrally buoyant and non-neutrally buoyant cases by plotting horizontal and vertical radii of the middle fiber ($r=1/2$) in each case as functions of time. In each case, the two radii oscillate in a damped manner, with one radius having its maximum at about the same time as the other radius is at a minimum. As one would expect, the oscillation has a lower frequency and is sustained longer in the more massive case.

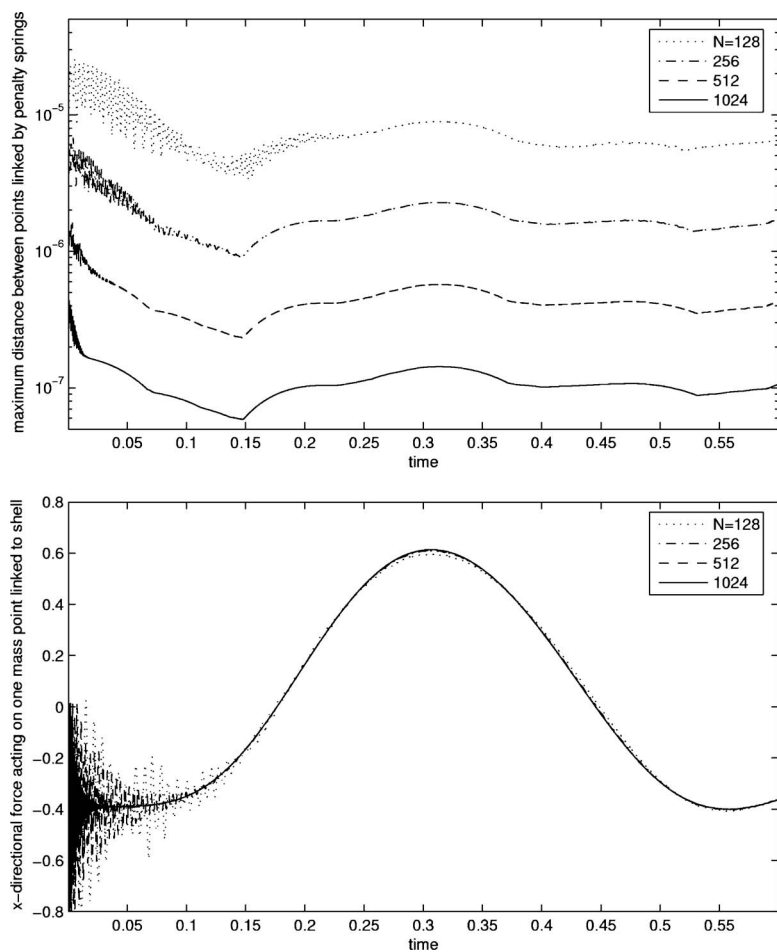


FIG. 4. Effects of the increase in the penalty parameter as the meshwidth is refined. Recall that the penalty stiffness is proportional to N^2 , where the Eulerian (fluid) grid is $N \times N$. The top panel shows the maximum distance between any of the massless immersed boundary points and their massive twins, plotted as a function of dimensionless time, in each of the cases $N = 128, 256, 512, 1024$. Note the logarithmic vertical scale. The bottom panel plots the x component of the force on the massive twin of the point marked “*” in Fig. 2, also for $N = 128, 256, 512, 1024$. Note the convergence to zero of the maximum distance, and the convergence to a finite limit of the penalty force, as N increases.

The bottom panel in Fig. 5 shows that the volume (area) contained within the middle fiber stays almost at a constant value as time goes on. While some volume loss is typical of IB computations with a thin boundary, the same computation with a thick shell preserves the volume regardless of the

mass of the shell. For approaches that lead to a good conservation of volume even in the case of a thin boundary, see Refs. 5, 27, and 28.

Although we have presented evidence here that the formally second-order accurate scheme of the pIB method is

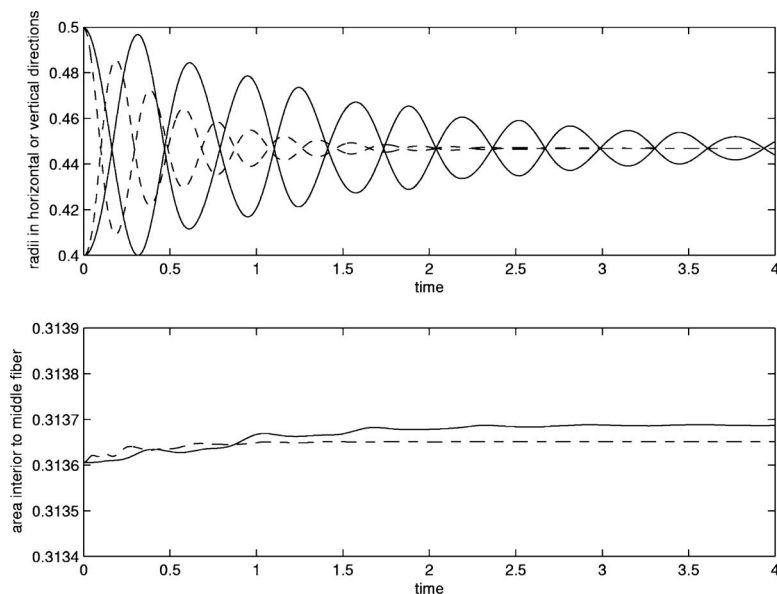


FIG. 5. The influence of mass on the behavior of an immersed elastic shell. Top panel shows the horizontal and vertical radii as functions of dimensionless time for a neutrally buoyant shell (dashed lines) and for a shell with a greater mass density than that of the surrounding fluid (solid lines). In the more massive case, the oscillations are slower and are sustained longer. The bottom panel checks for volume conservation by plotting the area enclosed by the middle fiber of the 2-D elastic shell, which should be constant in time, since the shell and fluid are both incompressible. Note the expanded vertical scale of the plot. The relative volume error over several cycles of oscillation is about 10^{-4} .

actually second-order-accurate in the case of an elastic shell of nonzero thickness, it is important to emphasize that only first-order accuracy is observed when the pIB method is applied to a problem involving a thin immersed boundary. (This behavior is shared with the IB method generally; it is not a distinctive feature of the pIB method.) It is natural, then, to ask what is the benefit of applying a formally second-order accurate IB or pIB scheme to a problem with a thin boundary, since only first-order accuracy will be realized in practice. We believe that the principal benefit is the reduction in numerical viscosity associated with the skew-symmetric differencing of the nonlinear terms in comparison to a first order upwind scheme. This benefit has in fact been demonstrated in Ref. 22 (see also Ref. 2). In particular, it was shown in Ref. 22 that the frequency of vortex shedding in flow past a cylinder is accurately predicted by the formally second-order immersed boundary method; i.e., that the results of such computations are in good agreement with experiment over a range of Reynolds numbers where the frequency of vortex shedding shows significant dependence on the Reynolds number. It is not clear, though, whether the skew-symmetric differencing of the nonlinear terms in any sense calls for the second-order-accurate time discretization that is also part of the formal second-order accuracy of our scheme. In some applications, it seems possible to use the skew-symmetric differencing of the nonlinear terms together with first-order-accurate time discretization, but in other applications this combination appears to be unstable.

V. VERIFICATION OF THE PIB METHOD BY COMPARISON WITH EXPERIMENTAL DATA: VORTEX-INDUCED VIBRATIONS OF A CIRCULAR CYLINDER

The vortex-induced vibration of an elastically constrained circular cylinder is a benchmark problem in fluid-structure interaction, see the excellent survey in Ref. 29, and the references therein. The situation most commonly considered²⁹⁻³¹ is the one in which the axis of the cylinder is perpendicular to the free stream, and in which the only allowed motions of the cylinder are in the direction that is perpendicular both to the axis of the cylinder and to the direction of the free stream. These are known as transverse motions, and that is the case that we consider here.

In order to introduce a massive cylinder, we modify pIB method explained in Sec. II. We assume that the mass of the cylinder is concentrated at its center (point mass). This is permissible since only translations of the cylinder are considered, not rotations. Then the massless boundary $\mathbf{X}(s,t)$ is a circle (boundary of the cylinder) and the massive “boundary” $\mathbf{Y}(t)$ is a point (center of the cylinder). When we previously assumed that the two boundaries were supposed to be the same, we used $\mathbf{F}_K(r,s,t) = K[\mathbf{Y}(r,s,t) - \mathbf{X}(r,s,t)]$ [Eq. (22)], which acted to keep the distance between two boundaries as close as possible. Using a generalization of this idea, we replace Eq. (22) by the following:

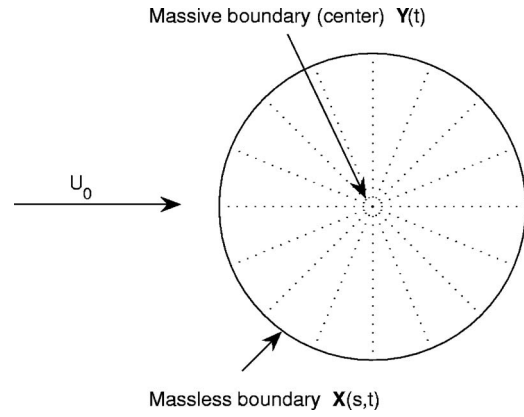


FIG. 6. Massive and massless boundaries are linked together by a collection of a very stiff springs. In this case, the massive “boundary” is simply the single point $\mathbf{Y}(t)$ at the center of the circle. In addition to the penalty springs (dotted radial lines), there is also a spring-dashpot combination (not shown) that is attached to the mass point $\mathbf{Y}(t)$, and acts as the elastic mechanical load on the cylinder [see Eq. (51)].

$$\mathbf{F}_K(s,t) = -K\{\mathbf{X}(s,t) - \mathbf{Y}(t) - \mathbf{Z}(s)\}, \quad (50)$$

where K is the penalty constant and $\mathbf{Z}(s) = [R \cos(s), R \sin(s)]$, where R is the radius of the cylinder, see Fig. 6. Note that this restoring force not only tends to maintain the circular configuration of the boundary $\mathbf{X}(s,t)$ at a distance R about the center $\mathbf{Y}(t)$, but also tends to prevent the rotational motion of the cylinder. On the other hand, this restoring force does not in any way restrict the translational motion of the cylinder as a whole.

For the equation of motion of the massive center point of the cylinder, we need to sum all the forces from the massless boundary on the massive center point $\mathbf{Y}(t)$, but we also need to include the forces of the mechanical restraints (spring and dashpot) that hold the cylinder on place in the laboratory experiment. Thus, we are led to the equation

$$M \frac{d^2 \mathbf{Y}(t)}{dt^2} + B \frac{d \mathbf{Y}(t)}{dt} + S \mathbf{Y} = - \int \mathbf{F}_K(s,t) ds, \quad (51)$$

where the integration is done over the massless boundary. Although M and B in Eq. (51) are scalars, S is a 2×2 diagonal matrix with diagonal entries (S_x, S_y) . The physical parameters were chosen as follows (similar to those in Ref. 31): the cylinder mass per unit length $M = 10$ g/cm, the damping coefficient $B = 0.0045$ (dyn s/cm)/cm, and the spring constant $(S_x, S_y) = (10^6, 50)$ (dyne/cm)/cm. Note in particular that the physical experiment of³¹ includes a damping force (“dashpot”) modeled by the term $B d\mathbf{Y}/dt$ in our equation of motion. Whereas the moderate value of S_y allows transverse motion, the large value of S_x prevents motion in the direction of the flow. Note that the denominator “cm” in these units appears because we consider a 2-D cross section perpendicular to the axis of a 3-D circular cylinder. The cylinder diameter $D = 1.0$ cm. Important fluid parameters include the fluid density $\rho = 1$ g/cm³, dynamic viscosity $\mu = 0.01$ g/(cm s). The constant free stream velocity (inflow velocity) U_0 varies from 1.3 to 2.75 cm/s, which induces the Reynolds number (Re) to vary from 130 to 275. The mass ratio $M/\rho\pi(D/2)^2$ is 12.73 and the damping ratio $B/2\sqrt{S_y}[M+\rho\pi(D/2)^2]$ is

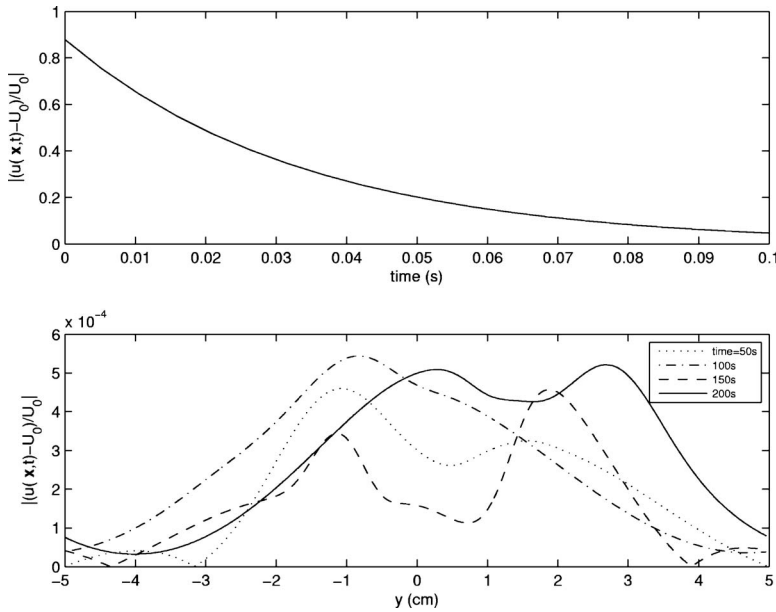


FIG. 7. The relative error of the x component of the calculated velocity $\mathbf{u}(\mathbf{x},t)$ in comparison to the desired velocity U_0 ; i.e., $|[u(\mathbf{x},t) - U_0]/U_0|$. The top panel shows the error at early times at a point of which the distance from the left upstream end is D . The bottom panel shows the error at selected later times on the line that is perpendicular to the free stream with the distance of D from the left upstream end of the domain. We can see that, except for the earliest times, the two velocities match well; i.e., our method of driving the inflow velocity very quickly forces the velocity in the upstream region to approach the desired inflow. In addition, the control of the inflow velocity is accurate to within 0.06% at all the times of the simulations after the initial transient time.

0.000 97. These two values combine to form another important dimensionless parameter, which is the product of mass ratio and damping ratio. This product is 0.0123 in our simulation. These dimensionless parameters, which are thought to characterize the system, are almost the same in our simulations as in Refs. 30 and 31.

Flow past the cylinder is from left to right and is controlled in the manner described in Sec. III. The top panel of Fig. 7 shows the relative error of the x component of the calculated velocity $u(\mathbf{x},t)$ in comparison to that of the desired velocity U_0 [see Eq. (41)]. We can see that, except for very early times, the two velocities match well; i.e., our method of driving the inflow velocity very quickly forces the velocity in the chosen region to approach the desired inflow. The bottom panel of Fig. 7 shows the relative error of the calculated velocity $u(\mathbf{x},t)$ on a line at various later times.

The line is perpendicular to the inflow direction with the distance of one cylinder diameter from the left upstream end. We can observe that the control of the inflow velocity is accurate to within 0.06% in all the times of the simulations after the initial transient time. Thus, we can consider the upstream flow to be uniform, which is desirable for this simulation.

Another computational issue is how large to make the penalty coefficient K , which appears in Eqs. (22) and (50). Our strategy is first to choose an allowed deviation from the nominal radius of the cylinder. We then adjust K until it is large enough that this allowed deviation is not exceeded. When adjusting K , we reduce Δt as needed to avoid numerical instability.

Although this process does involve trial and error, the tests for success are extremely simple, and it is easy to zero

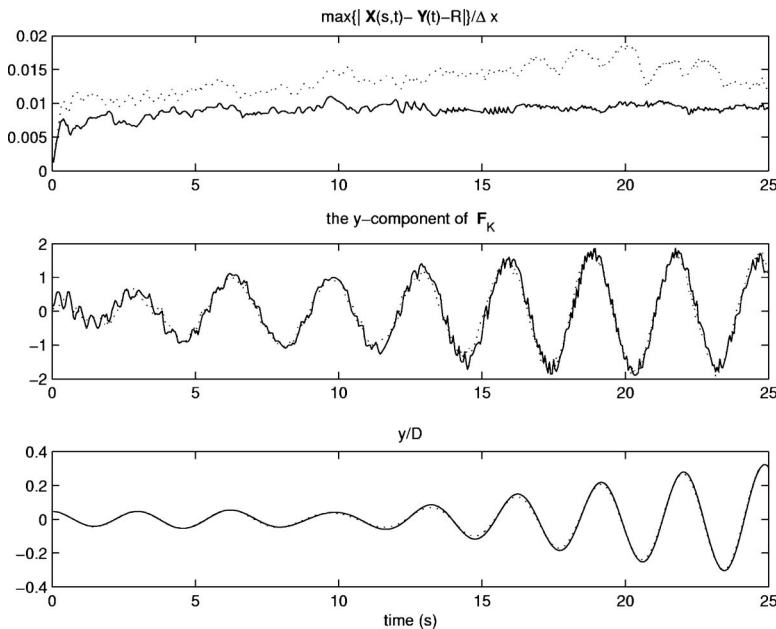


FIG. 8. The maximum deviation of the cylinder radius from the rest length R relative to the meshwidth Δx (top), the y component of the force $\mathbf{F}_K$ in Eq. (50) (middle), and the ratio of cylinder displacement $y(t)$ to the diameter of the cylinder D . Two different spring constants are used: $K=2 \times 10^6$ (dotted line) and 4×10^6 (solid line). Even though the deviations look quite different (top), the two forces generated by the two different penalty springs are very close (middle), and the transverse motions of the cylinder (bottom) in the two cases are almost the same.

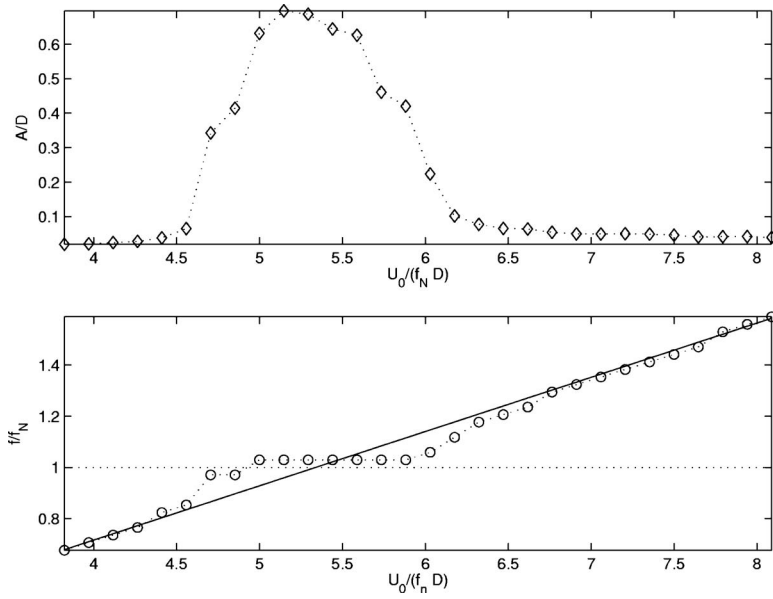


FIG. 9. The ratio of the amplitude A of the cylinder's oscillation to the diameter D of the cylinder (top), and the ratio of the frequency f of the cylinder's oscillation to the natural frequency $f_N=0.34$ Hz of the system (bottom). The x axis in both figures is the velocity ratio $U_0/(f_N D)$. Note the dramatically larger amplitude of vibration (top) that occurs when the velocity ratio is between 5 and 6, and the lock-in phenomenon (bottom) exhibited by the frequency of vibration (and vortex shedding) within this same range. The solid line in the bottom plot shows the frequency of vortex shedding that would occur if the cylinder were fixed in place. For comparison, this Strouhal frequency is plotted in units of the natural frequency of the elastically constrained cylinder.

in on good choices for K and Δt , as follows: The test for whether K is large enough is simply to monitor the maximum deviation of the radius of the cylinder from its nominal value ($D/2$). Since this deviation will be inversely proportional to K (for large K) it is easy to see from the results of a trial run how K needs to be adjusted for the next trial. Numerical instability resulting from too large a choice of Δt is unmistakable: distances between neighboring immersed boundary points rapidly exceed the size of the domain. In addition, in adjusting Δt when K changes, one can use the rule that (for a given spatial discretization) stability is determined by $K(\Delta t)^2$. This has the favorable consequence that to make K four times larger (for example), one only needs to make Δt twice as small.

We choose the allowed deviation of the radius to be 0.02 times the meshwidth. Figure 8 shows the following three different values as functions of time: the maximum deviation of the cylinder radius from the rest length R relative to the meshwidth Δx (top), the y -component of the force $\mathbf{F}_K$ in Eq. (50) (middle), and the ratio of amplitude $y(t)$ to the diameter

of the cylinder D . We use two different penalty spring constants: $K=2 \times 10^6$ (dotted line) and 4×10^6 (solid line). We can see from the top panel that the maximum deviation of the cylinder radii are controlled within 2.0×10^{-2} meshwidth with both spring constants. Even though these two graphs of maximum deviation are quite different, the two forces generated by the two different penalty springs are very close (middle), and the transverse motions of the cylinder in the two cases are almost the same. Note that this observation was also true in Sec. IV B (see Fig. 4).

It is known that an elastically mounted cylinder experiences large amplitude vibration only when the frequency of vortex shedding nearly coincides with the natural frequency of (damped) vibrations of the system when there is no inflow. Figure 9 shows that the simulation result obtained by the pIB method clearly contains the “lock-in” phenomenon observed in the actual experiment. The top panel plots the ratio of amplitude A of the cylinder's transverse vibration to the diameter D of the cylinder, and the bottom panel shows the ratio of the frequency f of the cylinder oscillation to the

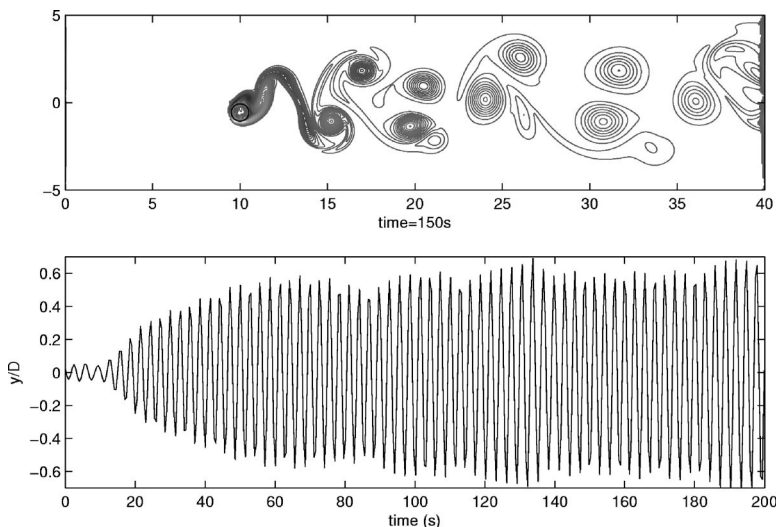


FIG. 10. Vortex-induced vibrations of a cylinder elastically mounted in a flow. The applied horizontal flow velocity is 1.8 cm/s. The upper figure shows position of the cylinder and vorticity contours in the whole computational domain at $t=150$ s. The lower figure is a plot of the y coordinate of the center of the cylinder as a function of time, in units of the cylinder diameter D . Note the gradual buildup of the amplitude of the oscillation to a nearly periodic steady state.

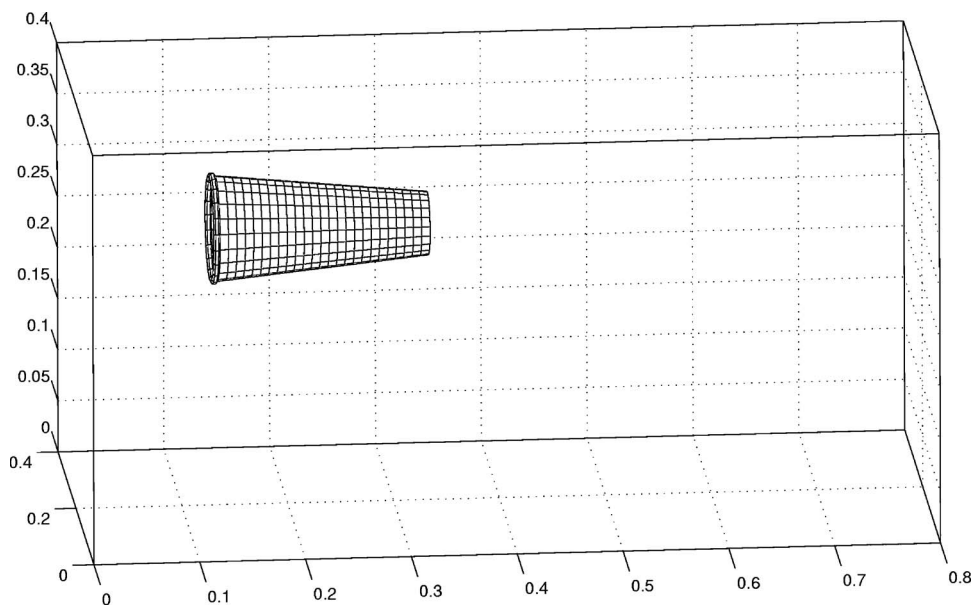


FIG. 11. Initial configuration of windsock and computational domain which is a box $0.8 \times 0.4 \times 0.4 \text{ m}^3$. The left end of the windsock is attached to a ring which is fixed in the domain. Wind is driven from left to right by prescribing inflow velocity at the left face of the domain.

natural frequency f_N in still water. We measured the natural frequency f_N numerically by the pIB method, and it is 0.34 Hz, which is slightly lower than the natural frequency in vacuum (0.356 Hz). The x axis in both figures is the “reduced velocity” or “velocity ratio” $U_0/(f_N D)$.

In Fig. 9, we can observe the lock-in phenomenon, in which the cylinder experiences a large amplitude vibration in the middle range of the flow speed (top) and that, in this region, the frequency of the cylinder oscillation remains constant and very close to the natural frequency (see the data points “-o-” in the bottom figure). Outside of the lock-in region, the amplitude of the cylinder’s oscillation is small (top), and the frequency of the oscillation is close to the Strouhal frequency (solid line, bottom) of a fixed cylinder. The Strouhal frequency is defined as $0.212(1 - 21.2/\text{Re})U_0/D$.³² This agrees with the well known results that (1) in the lock-in region, the frequency of vortex formation is close to the natural frequency, and (2) in non-lock-in region, it is close to the Strouhal frequency.^{29,30}

We have used here almost the same physical parameters as those in Refs. 30 and 31, have and obtained almost the same result as those in Refs. 30 and 31. This agreement is not only qualitative but also quantitative. The lock-in region corresponds to the range from 5 to 6 of the reduced velocity, which is the case both in this simulation and in the real experiment. The maximum amplitude of the oscillation is around 0.7, which is also very close to the experimental data (see the results in Refs. 29–31). In particular, Fig. 9 is similar to Fig. 2 in Ref. 30, Fig. 3 in Ref. 31, and Fig. 5 in Ref. 29. A vorticity plot and a graph of the oscillations of the cylinder are shown in Fig. 10.

VI. APPLICATIONS: WINDSOCK AND FLAG IN 3-D WIND

A windsock is a device used at airports to make the speed and direction of the wind visible to pilots. The magnitude of the wind and the weight of the windsock oppose each other to determine the extent to which the windsock stands

out horizontally or hangs down vertically. Thus, the mass of windsock is essential for the windsock simulations.

For the simulation, we make the windsock with the shape of a part of a slightly tapered cone, of which the circle at the upstream end is only a little larger than that at the downstream end. The upstream end of the windsock is attached to a fixed ring (see Fig. 11). The fixed ring is created by the method explained in Sec. III, generating a fixed no-slip boundary that can be of any shape. We then make a wind blow in one direction from left to right.

Table I contains the parameters used here (and also in the simulations of a flag considered below). The size of the windsock, which is represented by its length times the circumference of the fixed circle to which it is attached, and the surface mass density of windsock material are chosen arbitrarily. The other parameters are all realistic except viscosity. Air has viscosity $2 \times 10^{-5} \text{ kg/m s}$, which is 150–250 times smaller than the viscosities of the simulations. The reason for this modification is that we want to reduce Reynolds number (Re) in the simulations. We believe that, with computational resolution affordable so far, we can properly compute up to a Reynolds number of several hundreds. With the modified viscosity, the windsock has maximum $\text{Re}=396$ based on its

TABLE I. Parameters of the 3-D simulations.

Parameters	Windsock	Flag
Fluid density, kg/m^3	1.2	1.2
Viscosity, kg/m s	3.0×10^{-3}	5.0×10^{-3}
Inflow velocity, m/s	0.0–5.0	0.0–4.6
Density (mass), kg/m^2	0.08	0.09
Dimensions, m	$0.2 \times 0.08\pi$	0.333×0.25
Gravitational acceleration, m/s^2	9.8	9.8
Reynolds number	Up to 400	Up to 370
Domain (box), $\text{m} \times \text{m} \times \text{m}$	$0.8 \times 0.4 \times 0.4$	$1.28 \times 0.64 \times 0.64$

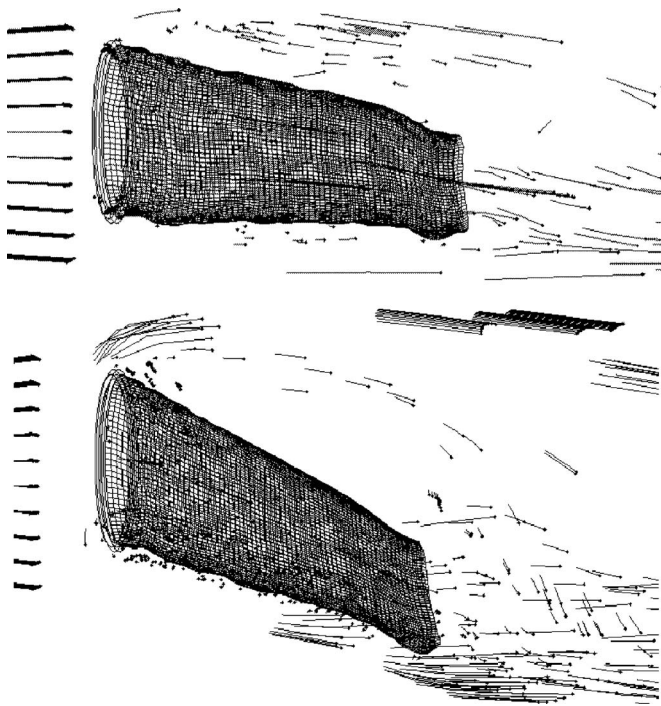


FIG. 12. Sideview of windsocks with different wind speeds: 5.0 m/s (top) and 2.5 m/s (bottom). The larger the wind speed is, the less tilt angle windsock has. The lines around the windsocks show the motion of some fluid markers. Each marker has left a trail showing its recent trajectory.

length, and the flag considered below has maximum $Re = 370$ based on its width. Note that, in all applications considered in this paper, we modify the viscosities usually existing in real experiments to make the Reynolds number between 100 and 400. Of course, a change on the Reynolds number alters the physics of the simulation, so it would be desirable to compute at the actual Reynolds number of the target application. In this connection, it should be noted that the restriction on the Reynolds number is not a fundamental one but is simply a question of the resolution of the computational mesh that can, in practice, be used. Future extensions of the methodology described here to higher Reynolds num-

ber will most likely involve adaptive mesh refinement and/or turbulence modeling. Research on both of these approaches in the context of immersed boundary methods is already underway.^{33,34}

At higher Reynolds numbers, one might worry that the low-amplitude high-frequency oscillations associated with the penalty springs would interact with the physical dynamics of the problem in spurious ways. In principle, this should not be a problem because the amplitude of these oscillations can be made as small as we like by sufficiently increasing the stiffness of the penalty springs. In practice, though, it may be helpful to damp these oscillations by replacing the penalty springs by spring-dashpot combinations (see Ref. 35, and references therein).

Figure 12 shows perspective views of the windsock with two different magnitudes of the wind velocity: 5.0 m/s on the top and 2.5 m/s on the bottom of the figures. A few fluid markers are also shown. Each marker has left a trail showing its recent trajectory (streak line). The windsock in the upper part of the figure has enough wind to make it stand out almost straight in the direction of the wind. The windsock in the lower part of the figure has half the wind speed of the upper one and slopes downwards at about 30° below the horizontal. Thus, we see that each windsock finds a different angle (with respect to the wind direction) depending on the wind speed.

The next 3-D application is a flapping flag in the wind. This application provides a 3-D generalization of the 2-D filament simulation done by Zhu and Peskin.¹⁵ In the flag problem, however, the wind flow is from the side and the gravity force acts downward while both in the 2-D filament case are in the downward direction. Thus the importance of the mass of the flag turns out to combine those of the filament and the windsock. Because of gravity, the flag will sag, and, because of the flag's inertia, we can also expect to see it flap. For related work on modeling cloth in the context of computer graphics (see the excellent survey in Ref. 36, and the references therein).

The parameters are shown in Table I and the initial configuration of the flag is drawn in Fig. 13. The initial configu-

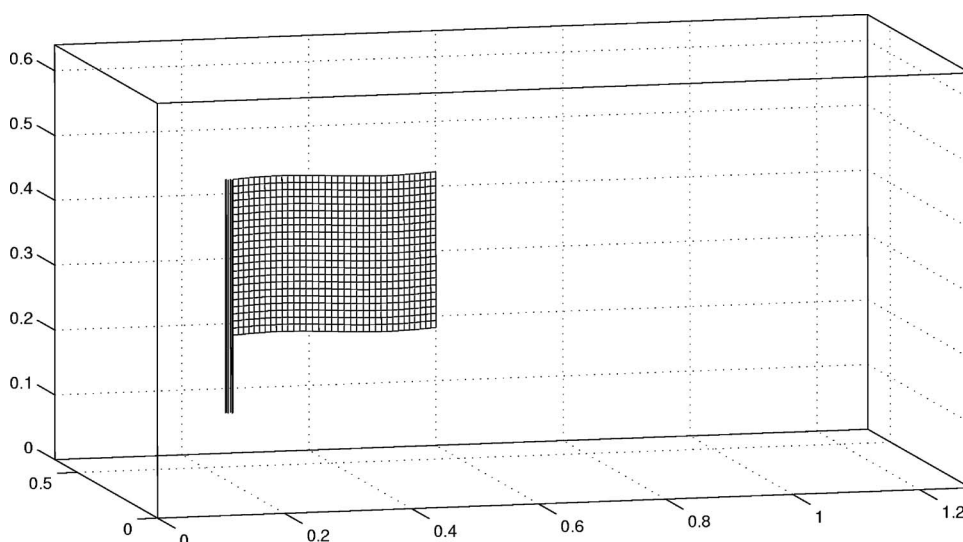


FIG. 13. Initial configuration of a flag and the computational domain which is a box $1.28 \times 0.64 \times 0.64 \text{ m}^3$. The left end part of the flag is attached with the flagpole which is fixed in the domain. The flow of the wind is from left to right.

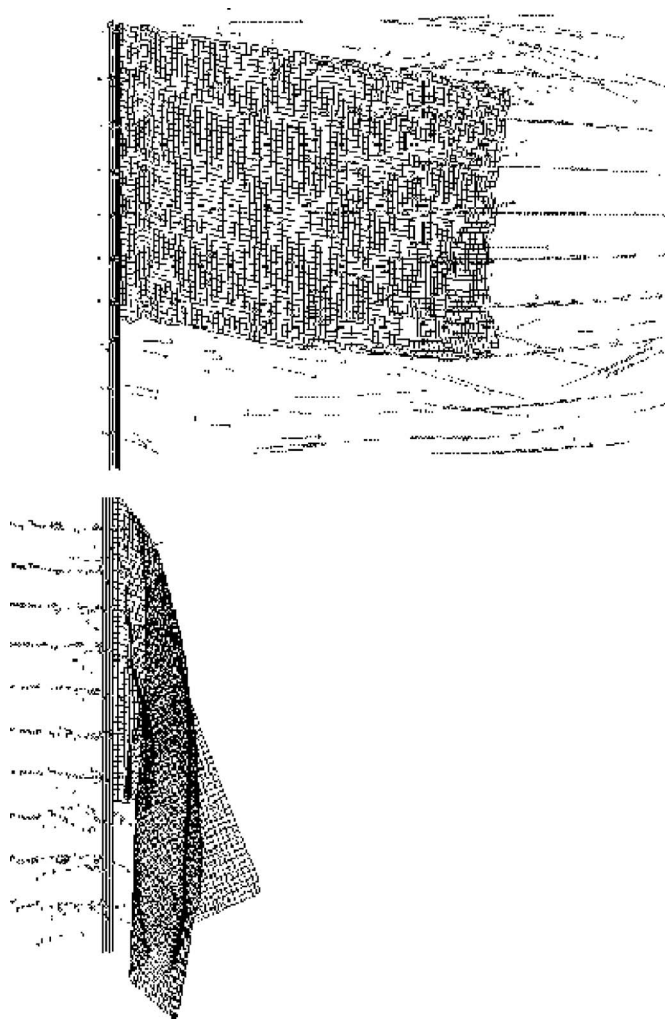


FIG. 14. A large enough windspeed (4.6 m/s, top) helps the flag overcome gravity and sustain a flapping motion. We can see the flag still sagging slightly. In the case of no wind (bottom), the flag falls down and stops moving. Note the folds which are a consequence of the (small) bending rigidity of the flag (time=1.2 s).

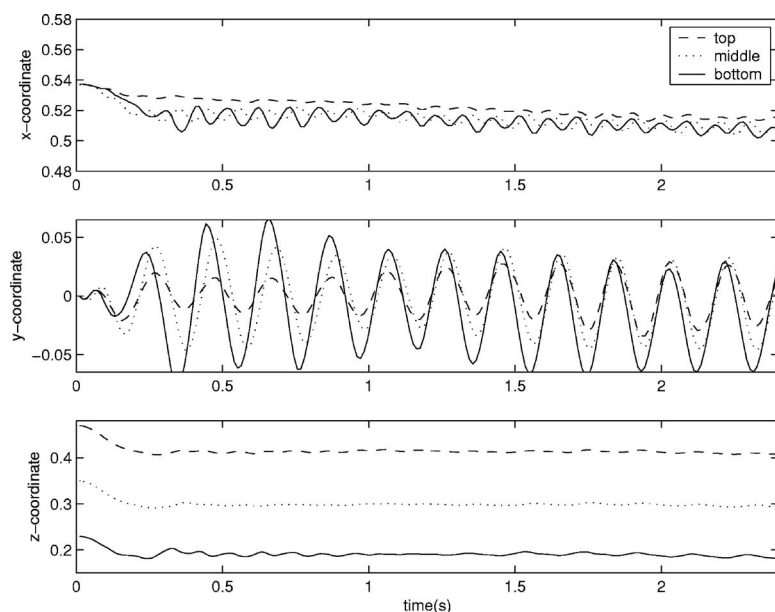


FIG. 15. Cartesian coordinates of the positions of three different material points on the downstream edge of the flag. Three points are chosen at the top, middle, and bottom of the downstream edge. The x axis is in the direction of the wind, the y axis is perpendicular to the flag and the wind direction, and the z axis is vertical. We can see from the middle panel that the flag is flapping, and, from the bottom panel, that the flag sags a little at the beginning of the computation and then stops sagging. Note (middle panel) that the flapping motions of top and bottom of the flag are out of phase; i.e., the bottom part runs ahead of the top part. In addition, the amplitude of the flapping motion is much greater at the bottom of the free edge than at the top.

ration of the flag is that of a sine curve from the top view. A vertical flagpole in the form of a cylinder is provided. It is fixed in place in the same way as the ring of points that anchors the windsock. The left edge of the flag is attached directly to the flagpole, and wind blows from left to right in these simulations.

Figure 14 shows the motions of the flag in two different situations. As can be seen in the lower frame in Fig. 14, when there is no wind, the flag falls down and stops moving. However, if the wind is strong enough, the flag overcomes gravity and straightens up in the wind direction. The case in the upper frame has the wind speed 4.6 m/s, which induces the Reynolds number 370 based on the data given in Table I, and, with this sufficiently large wind speed, the flag flaps, as shown below.

In order to observe more clearly the flapping and sagging motion of the flag in the top of Fig. 14, we can plot the trajectories of several points on the downstream edge of the flag as functions of time. In Fig. 15, choosing three points (top, middle, and bottom) of the downstream end of the flag, we plot the three coordinates of their motion in each case. The top frame shows the x coordinate of the motion, which represents the movement in the direction of wind. Since the y axis is perpendicular to both the flag and the wind direction, the middle frame clarifies the flapping motion of the flag. The bottom and middle points of the flag go through a relatively large oscillation, but the top undergoes only a small oscillation. One more interesting observation is that the three points are moving out of phase; i.e., the bottom part is moving ahead of the top part. These two phenomena (out of phase and different amplitudes) can be easily seen when we watch the motion of a real flag. The bottom panel shows the z coordinate (height) of the flag. The flag settles down little bit at the beginning of the computation and then stops sagging because of the drag force of the wind.

VII. SUMMARY AND CONCLUSIONS

We have introduced a new version of the IB method that can handle a massive elastic boundary, and we have shown that the new method can be well applied to many problems in which mass of the immersed boundary is important. This method has the virtue of simplicity: one can easily implement it in the context of an existing IB method code for the massless case. The new method contains a penalty parameter, which is the stiffness coefficient of the springs that connect the massless immersed boundary points to their massive twins. In principle, this stiffness should be infinite, and one might think that it would have to be so large as to require an excessively small timestep for numerical stability, but in practice we were pleased to find that this is *not* the case.

Since the purpose of this paper has been to introduce and illustrate the new method, we have not yet pursued the applications begun here in as much detail as they deserve. The flapping flag, in particular, seems to have many aspects that are worthy of further study. For example, we can see from the flag simulation that whether a flag is flapping or not depends on several parameters: mass density of the flag itself, speed of the incident wind, and even the amplitude of the initial perturbation of the flag from a planar configuration. A detailed exploration of the behavior of the flag as a function of these parameters is needed.

Another subject for future research is a comparison of the three different methods that have now been proposed for the simulation of massive immersed boundaries within the framework of the IB method. These are the mass-spreading method,^{15,16} the direct use of the D'Alembert force,²¹ and the pIB method of the present paper. A preliminary comparison is presented in Ref. 21, but further work is needed to sort out the advantages and disadvantages of the different approaches.

ACKNOWLEDGMENTS

We especially thank D. M. McQueen for kindly sharing experience and knowledge of various aspects of the IB method. We have made extensive use of his 3-D visualization software in carrying out this work. We also thank L. Zhu for helpful discussion about the flapping filament problem and the mass-spreading version of the IB method. This research was supported by the National Science Foundation under KDI research Grant No. DMS-9980069.

¹C. S. Peskin, "Numerical analysis of blood flow in the heart," *J. Comput. Phys.* **25**, 220 (1977).

²C. S. Peskin and D. M. McQueen, "Heart simulation by an immersed boundary method with formal second-order accuracy and reduced numerical viscosity," in *Mechanics for a New Millennium: Proceedings of the International Conference on Theoretical and Applied Mechanics (ICTAM)*, edited by H. Aref and J. W. Phillips (Kluwer Academic, Norwell, MA, 2001).

³C. S. Peskin and D. M. McQueen, "Three dimensional computational method for flow in the heart: Immersed elastic fibers in a viscous incompressible fluid," *J. Comput. Phys.* **81**, 372, 1989.

⁴C. S. Peskin and D. M. McQueen, "Fluid dynamics of the heart and its valves," in *Case studies in Mathematical Modeling: Ecology, Physiology, and Cell Biology* (Prentice Hall, Englewood Cliffs, NJ, 1996), pp. 309–337.

⁵C. S. Peskin and B. F. Printz, "Improved volume conservation in the computation of flows with immersed elastic boundaries," *J. Comput. Phys.* **105**, 33 (1993).

⁶R. P. Beyer, "A computational model of the cochlea using the immersed boundary method," *J. Comput. Phys.* **98**, 145 (1992).

⁷A. L. Fogelson and C. S. Peskin, "A fast numerical method for solving the three-dimensional Stokes equations in the presence of suspended particles," *J. Comput. Phys.* **79**, 50 (1988).

⁸A. L. Fogelson, "A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting," *J. Comput. Phys.* **56**, 111 (1984).

⁹K. M. Arthurs, L. C. Moore, C. S. Peskin, E. B. Pitman, and H. E. Layton, "Modeling arteriolar flow and mass transport using the immersed boundary method," *J. Comput. Phys.* **147**, 402 (1998).

¹⁰L. J. Fauci and C. S. Peskin, "A computational model of aquatic animal locomotion," *J. Comput. Phys.* **77**, 85 (1988).

¹¹E. Givberg, "Modeling elastic shells immersed in fluid," Ph.D. thesis, Department of Mathematics, New York University, 1997.

¹²E. Jung and C. S. Peskin, "Two-dimensional simulations of valveless pumping using the immersed boundary method," *SIAM J. Sci. Comput. (USA)* **23**, 19 (2001).

¹³Y. Kim and C. S. Peskin, "2-D parachute simulation by the Immersed Boundary Method," *SIAM J. Sci. Comput. (USA)* **28**, 2294 (2006).

¹⁴C. S. Peskin, "The immersed boundary method," *Acta Numerica* **11**, 479, 2002.

¹⁵L. Zhu and C. S. Peskin, "Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method," *J. Comput. Phys.* **179**, 452 (2002).

¹⁶L. Zhu, "Simulation of a flapping flexible filament in a flowing soap film by the Immersed Boundary method," Ph.D. thesis, Department of Mathematics, New York University, 2001.

¹⁷J. Zhang, S. Childress, A. Libchaber, and M. Shelley, "Flexible filaments in a flowing soap film as a model for one-dimensional flags in a two-dimensional wind," *Nature (London)* **408**, 835, 2000.

¹⁸Y. Kim, "The penalty immersed boundary method and its applications to aerodynamics," Ph.D. thesis, Department of Mathematics, New York University, 2003.

¹⁹C. Duncan, G. Zhai, and R. Scherer, "Modeling coupled aerodynamics and vocal fold dynamics using immersed boundary methods," *J. Acoust. Soc. Am.* **120**, 2859 (2006).

²⁰S. S. Antman, *Nonlinear Problems of Elasticity* (Springer-Verlag, New York, 1995).

²¹Y. Kim, L. Zhu, X. Wang, and C. S. Peskin, "On various techniques for computer simulation of boundaries with mass," in *Computational Fluid and Solid Mechanics: Proceedings Second M.I.T. Conference on Computational Fluid and Solid Mechanics*, edited by K. J. Bathe (Elsevier, Orlando, 2003), pp. 1746–1750.

²²M. C. Lai and C. S. Peskin, "An immersed boundary method with formal second-order accuracy and reduced numerical viscosity," *J. Comput. Phys.* **160**, 705 (2000).

²³A. Iserles, *A First Course in the Numerical Analysis of Differential Equations* (Cambridge University Press, Cambridge, 1996).

²⁴B. E. Griffith and C. S. Peskin, "On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems," *J. Comput. Phys.* **208**, 75, 2005.

²⁵C. Tu and C. S. Peskin, "Stability and instability in the computation of flows with moving immersed boundaries: a comparison of three methods," *SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput.*, **13**, 1361 (1992).

²⁶M. C. Lai, "Simulations of the flow past an array of circular cylinders as a test of the immersed boundary method," Ph.D. thesis, Department of Mathematics, New York University, 1998.

²⁷R. Cortez and M. Minion, "The blob projection method for immersed boundary problems," *J. Comput. Phys.* **161**, 428 (2000).

²⁸L. Lee and R. J. Leveque, "An immersed interface method for incompressible Navier-Stokes equations," *SIAM J. Sci. Comput. (USA)* **25**, 832 (2003).

²⁹C. H. K. Williamson and R. Govardhan, "Vortex-induced vibrations," *Annu. Rev. Fluid Mech.* **36**, 413 2004.

³⁰P. Anagnostopoulos and P. W. Bearman, "Response characteristics of a vortex-excited cylinder at low Reynolds numbers," *J. Fluids Struct.* **6**, 39 (1992).

³¹H. Blackburn, R. Govardhan, and C. H. K. Williamson, "A complementary numerical and physical investigation of vortex-induced vibration," *J. Fluids Struct.* **15**, 481 (2000).

- ³²A. Roshko, "On the development of turbulent wake from vortex street," NACA Rep., 1954, p. 1191.
- ³³A. Cristallo and R. Verzicco, "Combined immersed boundary/large-eddy-simulations of incompressible three dimensional complex flows," *Flow, Turbul. Combust.* **77**, 3 (2006).
- ³⁴B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin, "An adaptive, formally second order accurate version of the immersed boundary method," *J. Comput. Phys.* **223**, 1 (2007).
- ³⁵R. Mittal and G. Iaccarino, "Immersed boundary methods," *Annu. Rev. Fluid Mech.* **37**, 239 (2005).
- ³⁶H. N. Ng and R. L. Grimsdale, "Computer graphics techniques for modeling cloth," *IEEE Comput. Graphics Appl.* **16**, 28 (1996).