

An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity¹

Ming-Chih Lai² and Charles S. Peskin

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

Received February 16, 1999; revised February 1, 2000

A formally second-order accurate immersed boundary method is presented and tested in this paper. We apply this new scheme to simulate the flow past a circular cylinder and study the effect of numerical viscosity on the accuracy of the computation by comparing the numerical results with those of a first-order method. The numerical evidence shows that the new scheme has less numerical viscosity and is therefore a better choice for the simulation of high Reynolds number flows with immersed boundaries. © 2000 Academic Press

1. INTRODUCTION

Problems of biological fluid mechanics often involve the interaction of a viscous incompressible fluid with an elastic membrane. One can consider this membrane as a boundary immersed in the fluid. The immersed boundary method was designed to solve this kind of problem. The main idea is to use a regular Eulerian computational grid for the fluid mechanics together with a Lagrangian representation of the immersed boundary. The immersed boundary exerts a singular force on the fluid and at the same time moves at the local fluid velocity. The interaction between the fluid and immersed boundary can be modeled by a well chosen discretized approximation to the Dirac delta function, which is called a discrete delta function. This approach has been applied successfully to problems of blood flow pattern in the heart [20–24], wave propagation in the cochlea [3, 12], flow in collapsible tubes [27], aquatic animal locomotion [7–9], platelet aggregation during blood clotting [8, 11], the flow of suspensions [10, 30], flow and transport in a renal arteriole [1], and the cell and tissue deformation under shear flow [4, 6, 29].

¹ Supported by National Science Foundation under research Grant DMS/FD 92-20719.

² Corresponding author. Present address: Department of Mathematics, Chung Cheng University, Minghsiung, Chiayi 621, Taiwan. E-mail: mclai@math.ccu.edu.tw.

One common question about this method is how the numerical scheme performs at different Reynolds numbers. In other words, it is important to check whether the numerics interfere with the physics, especially for high Reynolds number flow. For example, the immersed boundary method calls a Navier–Stokes solver as a subroutine, so if the upwind scheme is used to approximate the nonlinear advection terms in the solver, then the numerical viscosity produced by the computation may not be negligible at high Reynolds number.

In this paper, we perform numerical simulations of flow past a circular cylinder using the immersed boundary method. We use this benchmark problem as a test of two different schemes. One of them is a new scheme, in which most (but not all) aspects of the method are second-order accurate. Our interests are not only to demonstrate the validity of the methodology but also to study the effect of numerical viscosity on the accuracy of high Reynolds number flow computations. It is important to note that our aim is to improve the immersed boundary methodology, which is typically applied to problems in which the boundary moves a lot. We are not trying to compete with other methods such as virtual boundary method for fixed boundary problems [13, 14, 28]. The reason why we choose a fixed boundary problem (flow past a circular cylinder) is that this is a problem for which the Reynolds number dependence of various important quantities is known, both from experiments and from reliable numerical simulations.

In Section 2, we review the mathematical formulation and the numerical scheme of the immersed boundary method. In Section 3, we introduce a new, formally second-order accurate scheme which will be used in a later section. In Section 4, we simulate the flow past a circular cylinder using those two different schemes and compare the results with experimental works. Conclusions are discussed in Section 5.

2. REVIEW OF THE IMMERSSED BOUNDARY METHOD

For simplicity, we consider the model problem of a viscous incompressible fluid in a two-dimensional square domain Ω containing an immersed massless boundary in the form of a simple closed curve Γ [25], the configuration of which will be given in parametric form, $\mathbf{X}(s, t)$, $0 \leq s \leq L_b$, $\mathbf{X}(0, t) = \mathbf{X}(L_b, t)$, where s tracks a material point of the immersed boundary. The equations of motion of the system are

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

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

$$\mathbf{f}(\mathbf{x}, t) = \int_0^{L_b} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds \quad (3)$$

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

$$\mathbf{F}(s, t) = \mathbf{S}(\mathbf{X}(\cdot, t), t). \quad (5)$$

Here $\mathbf{x} = (x, y)$, $\mathbf{u}(\mathbf{x}, t) = (u_1(\mathbf{x}, t), u_2(\mathbf{x}, t))$ is the fluid velocity and $p(\mathbf{x}, t)$ is the fluid pressure. The coefficients ρ and μ are the constant fluid density and viscosity. The force density (with respect to $d\mathbf{x} = dx dy$) acting on the fluid is $\mathbf{f}(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))$, while the boundary force density (with respect to ds) is $\mathbf{F}(s, t) = (F_1(s, t), F_2(s, t))$.

Equations (1)–(2) are the familiar Navier–Stokes equations of a viscous incompressible fluid. Equations (3)–(4) represent the interaction between the immersed boundary and the fluid; the Dirac delta function in these equations is two-dimensional; thus $\delta(\mathbf{x}) = \delta(x)\delta(y)$. In Eq. (3), the force density is applied to the fluid by the immersed boundary, while in Eq. (4), the immersed boundary is carried along with the fluid. Equation (5) states that the boundary force on the particular segment at time t is determined by the boundary configuration at time t , where the function \mathbf{S} satisfies a generalized Hooke’s law if the boundary is elastic. The explicit time dependence in this relationship allows for the possibility of an active boundary, like a muscle, the elasticity of which is time dependent. In our application, however, \mathbf{S} will represent a stiff passive force which tends to keep the boundary very close to a circular configuration.

Since the boundary force $\mathbf{F}(s, t)$ is integrated with a two-dimensional Dirac delta function over a one-dimensional curve Γ , the force density $\mathbf{f}(\mathbf{x}, t)$ is a one-dimensional singular Dirac delta function. Mathematically, $\mathbf{f}(\mathbf{x}, t)$ can be viewed as a distribution function whose action on any test function $\mathbf{w}(\mathbf{x}, t)$ is defined by

$$\begin{aligned} \langle \mathbf{f}, \mathbf{w} \rangle &= \int_{\Omega} \mathbf{f}(\mathbf{x}, t) \cdot \mathbf{w}(\mathbf{x}, t) \, d\mathbf{x} \\ &= \int_{\Omega} \int_0^{L_b} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) \, ds \cdot \mathbf{w}(\mathbf{x}, t) \, d\mathbf{x} \\ &= \int_0^{L_b} \mathbf{F}(s, t) \cdot \int_{\Omega} \mathbf{w}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) \, d\mathbf{x} \, ds \\ &= \int_0^{L_b} \mathbf{F}(s, t) \cdot \mathbf{w}(\mathbf{X}(s, t), t) \, ds. \end{aligned}$$

In particular, if we choose $\mathbf{w}(\mathbf{x}, t)$ to be the velocity $\mathbf{u}(\mathbf{x}, t)$, then the above identity implies that the total work done by the immersed boundary is equal to the total work done on the fluid.

Numerical scheme. The immersed boundary method is a mixed *Eulerian–Lagrangian* finite difference method for computing the flow interacting with an immersed boundary. So we need two distinct discretized grids: the regular *lattice points* cover the whole fluid domain and the *boundary points* discretize the immersed boundary. For simplicity, let the fluid domain $\Omega = [0, L] \times [0, L]$ and the fluid variables be defined on a fixed $N \times N$ *Eulerian* grid labeled as $\mathbf{x} = (x_i, y_j) = (ih, jh)$ for $i, j = 0, 1, \dots, N - 1$, where $h = \Delta x = \Delta y = \frac{L}{N}$ is the mesh width. On the other hand, we use another set of M *Lagrangian* points $\mathbf{X} = (X_k, Y_k)$ for $k = 0, 1, \dots, M - 1$ to discretize the immersed boundary with the initial boundary mesh width $\Delta s = L_b/M$. The boundary force is defined on these Lagrangian points. It is very important to note that the lattice points are fixed but the boundary points are moving, and those two sets of points usually do not coincide with each other.

Let D^+ , D^- , D^0 denote the forward, backward, and centered spatial difference operators defined on the fluid variables. As in Eqs. (1)–(5), we shall use lower-case variables to represent values defined on the lattice points, and upper-case variables to represent values on the boundary points. The superscript on a variable represents the time step index. Thus, $\mathbf{u}^n(\mathbf{x})$ and $\mathbf{X}^n(s)$ are the approximations of $\mathbf{u}(\mathbf{x}, n\Delta t)$ and $\mathbf{X}(s, n\Delta t)$, respectively. At the beginning of the time step, the $(\mathbf{u}^n(\mathbf{x}), \mathbf{X}^n(s))$ are given, so we need to march to $(\mathbf{u}^{n+1}(\mathbf{x}), \mathbf{X}^{n+1}(s))$.

The simplest version of the immersed boundary method is an *explicit* scheme which the boundary force is computed in the beginning of the time step. It can be done by the following steps:

(1) Compute the boundary force $\mathbf{F}^n(s)$ from the boundary configurations $\mathbf{X}^n(s)$ and apply the force to the fluid,

$$\mathbf{F}^n(s) = \mathbf{S}^n(\mathbf{X}^n), \quad (6)$$

$$\mathbf{f}^n(\mathbf{x}) = \sum_s \mathbf{F}^n(s) \delta_h(\mathbf{x} - \mathbf{X}^n(s)) \Delta s, \quad (7)$$

where the discrete delta function

$$\delta_h(\mathbf{x}) = d_h(x) d_h(y), \quad (8)$$

and

$$d_h(r) = \begin{cases} \frac{1}{8h} (3 - 2|r|/h + \sqrt{1 + 4|r|/h - 4(|r|/h)^2}), & |r| \leq h, \\ \frac{1}{8h} (5 - 2|r|/h - \sqrt{-7 + 12|r|/h - 4(|r|/h)^2}), & h \leq |r| \leq 2h, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

(2) Solve the Navier–Stokes equations with the force term $\mathbf{f}^n(\mathbf{x})$ to update the velocity field $\mathbf{u}^{n+1}(\mathbf{x})$,

$$\rho \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \sum_{i=1}^2 u_i^n D_i^\pm \mathbf{u}^n \right) = -\mathbf{D}^0 p^{n+1} + \mu \sum_{i=1}^2 D_i^+ D_i^- \mathbf{u}^{n+1} + \mathbf{f}^n, \quad (10)$$

$$\mathbf{D}^0 \cdot \mathbf{u}^{n+1} = 0, \quad (11)$$

where the upwind difference operator is defined as

$$u_i^n D_i^\pm = \begin{cases} u_i^n D_i^-, & u_i^n > 0, \\ u_i^n D_i^+, & u_i^n < 0, \end{cases} \quad (12)$$

$i = 1, 2$, and where the gradient difference operator is defined as

$$\mathbf{D}^0 = (D_1^0, D_2^0). \quad (13)$$

(3) Interpolate the new velocity from the lattice into the boundary points, and move the boundary points to new positions $\mathbf{X}^{n+1}(s)$,

$$\mathbf{U}^{n+1}(s) = \sum_{\mathbf{x}} \mathbf{u}^{n+1}(\mathbf{x}) \delta_h(\mathbf{x} - \mathbf{X}^n(s)) h^2, \quad (14)$$

$$\mathbf{X}^{n+1}(s) = \mathbf{X}^n(s) + \Delta t \mathbf{U}^{n+1}(s), \quad (15)$$

where δ_h is the discrete delta function defined above.

The above scheme is first-order accurate in time and space. In particular, in step (2), the upwind difference method is used for the spatial discretization of the nonlinear advection terms, and in step (3), the forward Euler method is used for the time integration of the boundary equation.

3. FORMALLY SECOND-ORDER METHOD

In this section, we present a new *formally* (but not actually, see below) second-order scheme of the immersed boundary method. The new scheme incorporates the old one, which is used, however, only in a preliminary way at each time step, to advance the solutions from time level n to time level $n + \frac{1}{2}$.

Preliminary step,

$$\mathbf{F}^n(s) = \mathbf{S}^n(\mathbf{X}^n), \quad (16)$$

$$\mathbf{f}^n(\mathbf{x}) = \sum_s \mathbf{F}^n(s) \delta_h(\mathbf{x} - \mathbf{X}^n(s)) \Delta s, \quad (17)$$

$$\rho \left(\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\Delta t/2} + \sum_{i=1}^2 u_i^n D_i^\pm \mathbf{u}^n \right) = -\mathbf{D}^0 \tilde{p}^{n+1/2} + \mu \sum_{i=1}^2 D_i^+ D_i^- \mathbf{u}^{n+1/2} + \mathbf{f}^n, \quad (18)$$

$$\mathbf{D}^0 \cdot \mathbf{u}^{n+1/2} = 0, \quad (19)$$

$$\frac{\mathbf{X}^{n+1/2}(s) - \mathbf{X}^n(s)}{\Delta t/2} = \sum_{\mathbf{x}} \mathbf{u}^{n+1/2}(\mathbf{x}) \delta_h(\mathbf{x} - \mathbf{X}^n(s)) h^2. \quad (20)$$

Once $\mathbf{u}^{n+1/2}$ and $\mathbf{X}^{n+1/2}$ are known, we use them to take a full step from time level n to time level $n + 1$, as follows:

Main step,

$$\mathbf{F}^{n+1/2}(s) = \mathbf{S}^{n+1/2}(\mathbf{X}^{n+1/2}(s)), \quad (21)$$

$$\mathbf{f}^{n+1/2}(\mathbf{x}) = \sum_s \mathbf{F}^{n+1/2}(s) \delta_h(\mathbf{x} - \mathbf{X}^{n+1/2}(s)) \Delta s, \quad (22)$$

$$\rho \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \frac{1}{2} \sum_{i=1}^2 (u_i^{n+1/2} D_i^0 \mathbf{u}^{n+1/2} + D_i^0 (u_i^{n+1/2} \mathbf{u}^{n+1/2})) \right) \quad (23)$$

$$= -\mathbf{D}^0 p^{n+1/2} + \frac{1}{2} \mu \sum_{i=1}^2 D_i^+ D_i^- (\mathbf{u}^n + \mathbf{u}^{n+1}) + \mathbf{f}^{n+1/2},$$

$$\mathbf{D}^0 \cdot \mathbf{u}^{n+1} = 0, \quad (24)$$

$$\frac{\mathbf{X}^{n+1}(s) - \mathbf{X}^n(s)}{\Delta t} = \sum_{\mathbf{x}} \frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2} \delta_h(\mathbf{x} - \mathbf{X}^{n+1/2}(s)) h^2. \quad (25)$$

One can see that the Navier–Stokes solver for the fluid equations in the main step of the new scheme is a time-centered or Crank–Nicolson scheme, in which a skew-symmetric finite difference discretization is used for the nonlinear convection term. The reason of the choice for skew-symmetric differencing will be discussed later. The trapezoidal quadrature rule is applied to the moving boundary equation.

We say that an immersed boundary method has *formal* second-order accuracy if it would be second-order accuracy when applied to a problem in which the delta functions are replaced by fixed smooth functions, independent of the mesh. That is, if instead of δ_h we had a fixed smooth function, independent of h , and if that same smooth function appeared instead of the Dirac δ function in our original problem formulation, then the above scheme would be second-order accurate, in space and time. (This is true despite the first-order

accuracy of the preliminary step, from time level n to time level $n + 1/2$.) We are not, however, in that fortunate hypothetical situation, and it has been noted by several authors [2, 15, 17] that in the actual case of delta function forces applied along a boundary, the use of the discrete delta function in the boundary-fluid interaction prevents the immersed boundary method from being more than first-order accurate. This is because the velocity, although continuous, does not have a continuous derivative across the boundary, and the discrete delta function provides too much smoothing. We have not overcome this difficulty here, which is why we say that the method is only *formally* second-order accurate. The formal second-order accuracy is useful, however, because it reduces numerical viscosity.

The new scheme, like the old one, is *explicit* in the computation of the boundary forces. These must be calculated twice per time step in the new scheme, once from the boundary configuration \mathbf{X}^n and again from the boundary configuration $\mathbf{X}^{n+1/2}$.

The motivation for the skew-symmetric differencing of the convection terms is as follows. Let us consider (as a model problem) the equation of a scalar function ϕ transported by an incompressible fluid with velocity \mathbf{u}

$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0, \quad (26)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (27)$$

If the boundary conditions are periodic, then $\|\phi\|^2$ (the integral of the square of ϕ) is conserved. That is,

$$\frac{d}{dt} \|\phi\|^2 \equiv \frac{d}{dt} \int_{\Omega} \phi^2 d\mathbf{x} = 0. \quad (28)$$

The above equality can be easily derived by using the equations with periodic boundary conditions and integration by parts.

Equation (28) is satisfied when ϕ is a function of continuous variables of time and space. It is desirable for the discrete solution to satisfy the analogous conservation property. This is because the weak instability caused by the nonlinear convection term can be avoided if the discrete convection form satisfies this conservation property and the usual Courant–Friedrichs–Lewy number condition for linearized stability [26]. Now let us focus only on the spatial discretization and consider the semi-discrete evolution equation as

$$\phi_t = L\phi, \quad (29)$$

where L is some linear spatial operator. If the spatial difference operator L in the above equation is skew-symmetric, then the conservation property (28) holds for the discrete solution [5].

It is easy to check that $\mathbf{u} \cdot \mathbf{D}^0$ is *not* skew-symmetric in general, not even if we impose the condition $\mathbf{D}^0 \cdot \mathbf{u} = 0$. Using the incompressibility constraint, we can, however, reformulate Eq. (26) as

$$\phi_t + \frac{1}{2}(\mathbf{u} \cdot \nabla \phi + \nabla \cdot (\mathbf{u}\phi)) = 0. \quad (30)$$

Now we can discretize space in the above equation by centered differences and obtain

$$\phi_t + \frac{1}{2}(\mathbf{u} \cdot \mathbf{D}^0 \phi + \mathbf{D}^0 \cdot (\mathbf{u}\phi)) = 0, \quad (31)$$

in which the spatial difference operator applied to ϕ is indeed skew-symmetric, so that $\|\phi\|^2$ is conserved for arbitrary \mathbf{u} . To show this, use summation by parts and the periodic boundary conditions.

4. SIMULATIONS OF A TIME PERIODIC VORTEX STREET BEHIND A CIRCULAR CYLINDER

The flow around a circular cylinder immersed in a fluid stream is studied as a typical model problem for separated flows and boundary layer theory. It has been the subject of many theoretical, experimental, and computational works. Depending on the Reynolds number, different kind of flow behaviors can be characterized. At a low Reynolds number, the flow is viscosity dominated and is called a creeping flow. At somewhat higher Reynolds number (up to $Re = 45$), two symmetrical standing vortices are formed and attached behind the cylinder. When the Reynolds number gets higher, these vortices can stretch farther and farther downstream from the cylinder and eventually become distorted and break apart to develop an alternating vortex shedding called the Kármán vortex street. For Reynolds numbers up to 200, this flow is purely laminar and the vortex street is stable and time periodic. Readers who are interested in more detail about this flow can refer to [18].

4.1. Virtual boundary formulation. In [13, 14], Goldstein *et al.* used an approach called the *virtual boundary method* to simulate the turbulent flow over a modeled riblet-covered surface and other problems. Their method shares the same spirit with the immersed boundary method but solves a rigid boundary problem rather than an elastic boundary problem.

The main idea of the virtual boundary method is to treat the body surface as a *virtually* existent boundary embedded in the fluid which applies force on the fluid so that the fluid will be at rest on the surface (no-slip condition). Let us denote the boundary Γ (body surface) by $\{\mathbf{X}^e(s); 0 \leq s \leq L_b\}$. The force $\mathbf{F}(s, t)$ on the boundary is determined by the requirement that the fluid velocity $\mathbf{u}(\mathbf{x}, t)$ should satisfy the no-slip condition on the boundary. Mathematically, we need to solve the two-dimensional Navier–Stokes equations with some boundary velocity constraints as

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p = \mu \Delta \mathbf{u} + \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}^e(s)) ds, \quad (32)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (33)$$

$$0 = \mathbf{u}(\mathbf{X}^e(s), t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}^e(s)) d\mathbf{x}, \quad (34)$$

$$\mathbf{u}(\mathbf{x}, t) \rightarrow \mathbf{u}_{\infty} \quad \text{as } |\mathbf{x}| \rightarrow \infty. \quad (35)$$

The “count” of equations and unknowns in this formulation seems reasonable, since we have introduced two unknown components of $\mathbf{F}(s, t)$ defined on the boundary Γ into the the Navier–Stokes equations in (32) and, at the same time, we have imposed two additional constraints on the boundary in (34). Since the body force is not known *a priori*, it must be calculated in some feedback way in which the velocity on the boundary is used to determine the desired force. In the virtual boundary formulation, the force is expressed as

$$\mathbf{F}(s, t) = \alpha \int_0^t \mathbf{U}(s, t') dt' + \beta \mathbf{U}(s, t), \quad (36)$$

where \mathbf{U} is the fluid velocity at these surface points. When α and β are chosen negative and large enough in magnitude, then \mathbf{U} will stay close to zero. To avoid interpolating the velocity field from grid points to the boundary points, Goldstein *et al.* let the boundary points coincide with grid points. However, in order to generate a smooth surface rather than a step-like surface, the boundary force is multiplied by a narrow Gaussian distribution so that the nearby grid points can receive a part of force influences. Although this local smoothing will blur the location of the surface within one grid site, the method can produce promising results if sufficient spatial resolution is used.

More recently, Saiki and Biringen [28] applied the virtual boundary method to simulate the flow past a cylinder. They used an area-weighted average function to interpolate the fluid velocity to the boundary points and extrapolate the boundary force back to the grid points. This fluid-boundary interaction process is different from Goldstein *et al.* and is more like the immersed boundary method. It turns out that the area-weighted average function is nothing but the discrete hat function in [2]. In [28], very good agreement is found between their calculations and previous computational and experimental results for steady and time-dependent flow at low to moderate Reynolds numbers.

In the following section, we perform similar simulations as in [28]. It is important to note that we choose the same problem as a test of two different immersed boundary methods. However, there are two differences between Saiki *et al.* and our present calculations. First, all simulations are done using the spectral method in [28] but a finite difference method is used for our calculations. Second, in our immersed boundary computations, the boundary points are moving slightly but in theirs the boundary points are actually fixed in space.

4.2. Immersed boundary computation. In order to simulate the flow around a rigid boundary using the immersed boundary method, we should allow the boundary to move a little bit rather than be fixed. As long as the immersed boundary $\mathbf{X}(s, t)$ stays close to the body surface $\mathbf{X}^e(s)$, we can rewrite Eqs. (32)–(35) as

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) + \nabla p = \mu \Delta \mathbf{u} + \int_{\Gamma} \mathbf{F}(s, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds, \quad (37)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (38)$$

$$\frac{\partial \mathbf{X}(s, t)}{\partial t} = \mathbf{u}(\mathbf{X}(s, t), t) = \int_{\Omega} \mathbf{u}(\mathbf{x}, t) \delta(\mathbf{x} - \mathbf{X}(s, t)) d\mathbf{x}, \quad (39)$$

$$\mathbf{u}(\mathbf{x}, t) \rightarrow \mathbf{u}_{\infty} \quad \text{as } |\mathbf{x}| \rightarrow \infty. \quad (40)$$

Now we need to choose an appropriate forcing term $\mathbf{F}(s, t)$ in Eq. (37) to make sure that the boundary points will stay close to the body surface. One straightforward choice is

$$\mathbf{F}(s, t) = \kappa (\mathbf{X}^e(s) - \mathbf{X}(s, t)), \quad (41)$$

where κ is a positive constant such that $\kappa \gg 1$. The direct interpretation of Eq. (41) is that we connect the boundary points \mathbf{X} to fixed equilibrium points \mathbf{X}^e with a very stiff spring whose stiffness constant is κ . So if the boundary points fall away from the desired location, the force on the spring will pull these boundary points back. Thus, as time goes on, we can expect that the boundary points will always be close to their desired configurations. It is important to note that the force in Eq. (41) is very similar to the one used for the virtual boundary method in Eq. (36). Actually, the force term chosen in (41) is a particular case of (36) with $\beta = 0$.

In order to mimic the real situation of the flow around a circular cylinder, we need to choose a relative large computational fluid domain (compared to the size of the cylinder) and modify the inflow velocity field of this domain (similar to imposing the far field velocity). Despite the misbehavior of the flow near the computational boundary due to the periodicity of the method, we can still capture the fluid behavior such as the vortex shedding behind the cylinder.

Let us explain more carefully how to simulate this situation. As mentioned before, we consider the surface of the cylinder as a rigid boundary immersed in a fluid. In the simulation, this boundary will move slightly, but κ will be chosen so large that the motion will not be noticeable. Thus, we need to solve Eqs. (37)–(39) by the immersed boundary method. To start the motion, we set the initial velocity to be the far field velocity \mathbf{u}_∞ everywhere in the fluid domain. Once the nonzero velocity is applied to the immersed boundary points, a force field is created by the movement of these points. To maintain the far field boundary condition, we proceed as follows: After each time step, we simply modify the velocity in a thin vertical strip Ω_B running along the left (inflow) side of the computational domain to be \mathbf{u}_∞ . This modification can be thought of as the application of a force to the fluid, the force being confined to the strip Ω_B and being chosen by a feedback mechanism to be just sufficient to make the velocity equal to the prescribed far field velocity within that strip. Throughout the paper, we assume that the far field velocity points in the x direction, $\mathbf{u}_\infty = (u_\infty, 0)$.

The Reynolds number in this flow is defined as

$$Re = \frac{\rho u_\infty D}{\mu}, \quad (42)$$

where D is the diameter of the cylinder. We can also define the dimensionless time scale as

$$T = \frac{u_\infty t}{(1/2)D}. \quad (43)$$

The principal result of our computation is the velocity field. For output purposes, though, we can easily compute the drag and lift coefficients, and the Strouhal number.

Drag coefficient. The drag force on a body submerged in a stream arises from two sources, the shear stress and the pressure distribution along the body. The dimensionless drag coefficient is defined by

$$C_D = \frac{F_D}{(1/2)\rho u_\infty^2 D}, \quad (44)$$

where F_D is the drag force. In the present computation, we have the opportunity to evaluate the drag force F_D in three different ways:

(1) We can determine the drag force simply by looking at the x component of the force applied by the boundary to the fluid. This of course, is equal to the negative of the drag, by Newton's third law of motion. Thus,

$$F_D = - \int_{\Omega} f_1 d\mathbf{x} = - \int_0^{L_b} F_1 ds, \quad (45)$$

where f_1 and F_1 are the x -components of the force densities \mathbf{f} and \mathbf{F} , respectively.

(2) As mentioned before, after each time step, we modify the velocity in Ω_B to equal to \mathbf{u}_∞ . So the change of the x -component of the momentum can be calculated by

$$\Delta m = \int_{\Omega_B} \rho(u_\infty - u_1) d\mathbf{x}. \quad (46)$$

Since the rate of momentum change is simply the force, the drag force can also be computed by

$$F_D = \frac{\Delta m}{\Delta t}, \quad (47)$$

where Δt is the time step size in the computation.

(3) The integral form of the x -component of the momentum equations on any fluid domain Ω_0 can be described by

$$\begin{aligned} & \frac{\partial}{\partial t} \int_{\Omega_0} \rho u_1 d\mathbf{x} + \int_{\partial\Omega_0} \rho u_1 \mathbf{u} \cdot \mathbf{n} ds \\ &= - \int_{\Omega_0} p n_1 d\mathbf{x} + \int_{\partial\Omega_0} \mu \left(\frac{\partial u_1}{\partial x_j} + \frac{\partial u_j}{\partial x_1} \right) n_j ds + \int_{\Omega_0} f_1 d\mathbf{x}. \end{aligned} \quad (48)$$

When the flow becomes steady, the above equation reduces to

$$\int_{\partial\Omega_0} \rho u_1 \mathbf{u} \cdot \mathbf{n} ds = - \int_{\Omega_0} p n_1 d\mathbf{x} + \int_{\partial\Omega_0} \mu \left(\frac{\partial u_1}{\partial x_j} + \frac{\partial u_j}{\partial x_1} \right) n_j ds + \int_{\Omega_0} f_1 d\mathbf{x}. \quad (49)$$

So the drag force can be calculated by

$$F_D = - \int_{\Omega_0} f_1 d\mathbf{x} = - \int_{\partial\Omega_0} \rho u_1 \mathbf{u} \cdot \mathbf{n} ds - \int_{\partial\Omega_0} p n_1 ds + \int_{\partial\Omega_0} \mu \left(\frac{\partial u_1}{\partial x_j} + \frac{\partial u_j}{\partial x_1} \right) n_j ds. \quad (50)$$

Therefore, we simply pick any square domain Ω_0 enclosing the cylinder and compute the line integrals of the above equation to obtain the drag force.

Lift coefficient. When the body starts shedding a vortex, a lift force on the body is generated by the fluid. The dimensionless lift coefficient is defined by

$$C_L = \frac{F_L}{(1/2)\rho u_\infty^2 D}, \quad (51)$$

where F_L is the lift force. As in the drag force calculation, the simplest way to measure the lift force in the computation is the direct calculation of the y -component of the force density. That is,

$$F_L = - \int_{\Omega} f_2 d\mathbf{x} = - \int_0^{L_b} F_2 ds, \quad (52)$$

where f_2 and F_2 are the y -components of the force densities \mathbf{f} and \mathbf{F} , respectively.

Strouhal number. When the steady flow becomes unstable and the body starts shedding vortices, the frequency with which the vortices are shed from the body can be made dimensionless by the formula

$$St = \frac{f_q D}{u_\infty}, \quad (53)$$

where f_q is the vortex shedding frequency. The new parameter St is called the Strouhal number. In our computation, it is easy to measure the dimensionless time period T_p between vortices shedding. Thus, using the fact that $f_q = 1/T_p$ and the definition of dimensionless time scale in Eq. (43), St is measured by

$$St = \frac{2}{T_p}. \quad (54)$$

Computation details. We choose a large computational domain $\Omega = [0, 8] \times [0, 8]$ and a cylinder with diameter $D = 0.30$ whose center is located at $(1.85, 4.0)$; thus, the cylinder is very small compared to Ω (the size of the cylinder diameter versus the size of the domain is about 1 : 27). The fluid density is $\rho = 1.0$ and the far field velocity is $u_\infty = 1.0$. The fluid viscosity is varied to achieve the desired Reynolds number in any particular computation; see Eq. (42). In [19], the authors use the preconditioned multigrid method to simulate this flow as a test problem for their schemes and also collect very detailed experimental and numerical results for comparison. As in [19], three different Reynolds numbers $Re = 100, 150, 200$ are considered and our numerical results are compared with the experimental results given there.

The computation starts with a random perturbation of the initial velocity in order to speed up the transition to alternate vortex shedding and thereby save computing time. This initial perturbation affects only the onset time of the vortex shedding.

Table I provides the results for these two schemes with different parameters such as mesh width h , time step Δt , and the stiffness constant κ . Scheme 1 is the first-order method, and Scheme 2 is the present, formal second-order method. The maximum displacement is measured by the maximum norm of all boundary points deviated from their initial positions divided by the radius of the cylinder. This is a measure of the relative motion of the cylinder surface. The stiffness κ is chosen so that the maximum displacement of any boundary point is within 5%. Once the stiffness constant κ is chosen, the time step is determined by the stability constraint $\Delta t \approx C\sqrt{h/\kappa}$ which is derived in [16]. The drag and lift coefficients are time-averaged since the flow is unsteady. All calculations are up to time $T = 240$.

From Table I, we can see the Strouhal number is getting close to the experimental measured number as the mesh is refined. In the paper of Saiki and Biringen [28], the computed

TABLE I
The Values of the Different Quantities at $Re = 100$

Method	h	κ	Δt	C_D	C_L	St	Max disp.
Scheme 1	1/64	4.8×10^4	1.8×10^{-3}	1.5406	0.2829	0.133	3.06%
Scheme 2	1/64	4.8×10^4	1.8×10^{-3}	1.5167	0.2904	0.155	4.54%
Scheme 1	1/128	9.6×10^4	0.9×10^{-3}	1.4630	0.3290	0.144	1.51%
Scheme 2	1/128	9.6×10^4	0.9×10^{-3}	1.4473	0.3299	0.165	2.57%

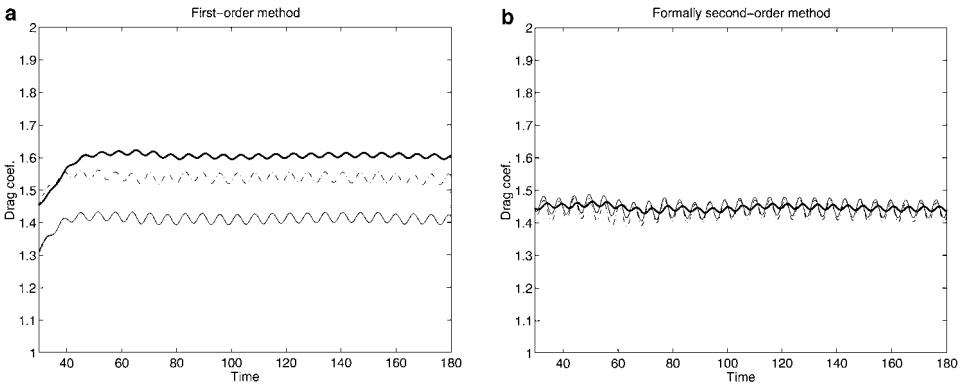


FIG. 1. Three different measures ((1) — ; (2) — ; (3) ---) of drag coefficients at $Re = 150$ for (a) first-order method and (b) formally second-order method.

Strouhal numbers obtained from different researchers' computations ranged from 0.16–0.18 at $Re = 100$. The Strouhal numbers computed by the formally second-order method have a good agreement with experimental measured data. As for the drag coefficients, our present results are higher than the experimental and some previous computed data [28]. However, in our numerical experiments, if we choose the computational domain larger, the drag coefficient becomes smaller. This is not surprising, since the boundary influence becomes weaker when the domain is larger. Furthermore, as the mesh is refined, the drag coefficient becomes smaller as well. Note that our immersed boundary approach has the advantage such that the drag and lift coefficients can be measured more easily than other methods.

Figure 1 shows the time evolution of the drag coefficients measured by three different methods for $Re = 150$. We can see that the three measures of computed drag coefficients are almost the same for the formal second-order method but different for the first-order method. This suggests that our new scheme is more accurate than the first-order method, despite the fact that the Dirac delta approximation will still result in some loss of accuracy. The time evolution of the lift coefficients is shown in Fig. 2.

It is interesting to note in these figures that the oscillation frequency of the drag is twice that of the lift (vortex shedding frequency). This is because the vortex sheds from upper and

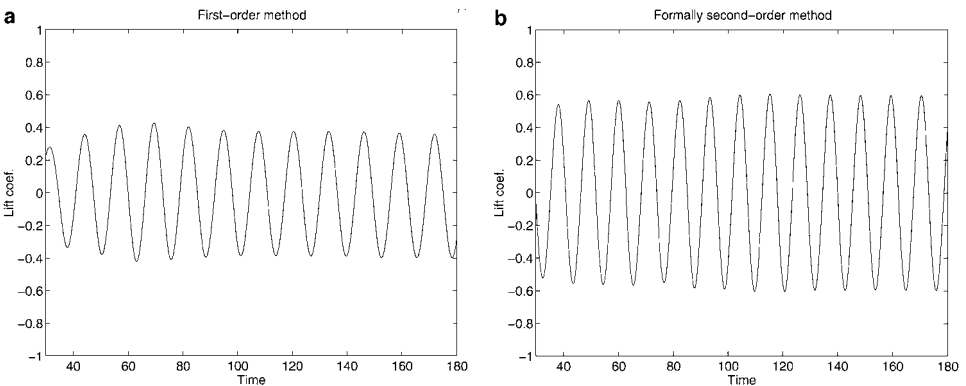


FIG. 2. The lift coefficients at $Re = 150$ for (a) first-order method and (b) formally second-order method.

TABLE II
The Comparison of the Numerical and Experimental Strouhal Number

Re	Scheme 1	Scheme 2	Williamson (Exp.)	Roshko (Exp.)
100	0.144	0.165	0.166	0.164
150	0.156	0.184	0.183	0.182
200	0.163	0.190	0.197	0.190

lower surfaces alternately (thus, the lift changes sign alternately), but the upper and lower vortex shedding makes almost the same contribution to the drag.

Table II shows the comparison of the computed Strouhal number with experimental data by Williamson and Roshko, as reported in [19]. We can see that the numerical results of the formally second-order method are in excellent agreement with the experimental data. In the numerical results of the first-order method, the Strouhal number (dimensionless frequency of vortex shedding) is about 20% too low. We attribute this to the numerical viscosity of the first-order method, which seems to be much reduced in the formally second-order computation.

The instantaneous vorticity contours of vortex shedding computed at the final time by those methods are plotted in Fig. 3. We can see the Kármán vortex street of the flow around a circular cylinder in these vorticity contour lines. The higher Strouhal number of the formally second-order computation is evident in the closer spacing between the vortices in the lower

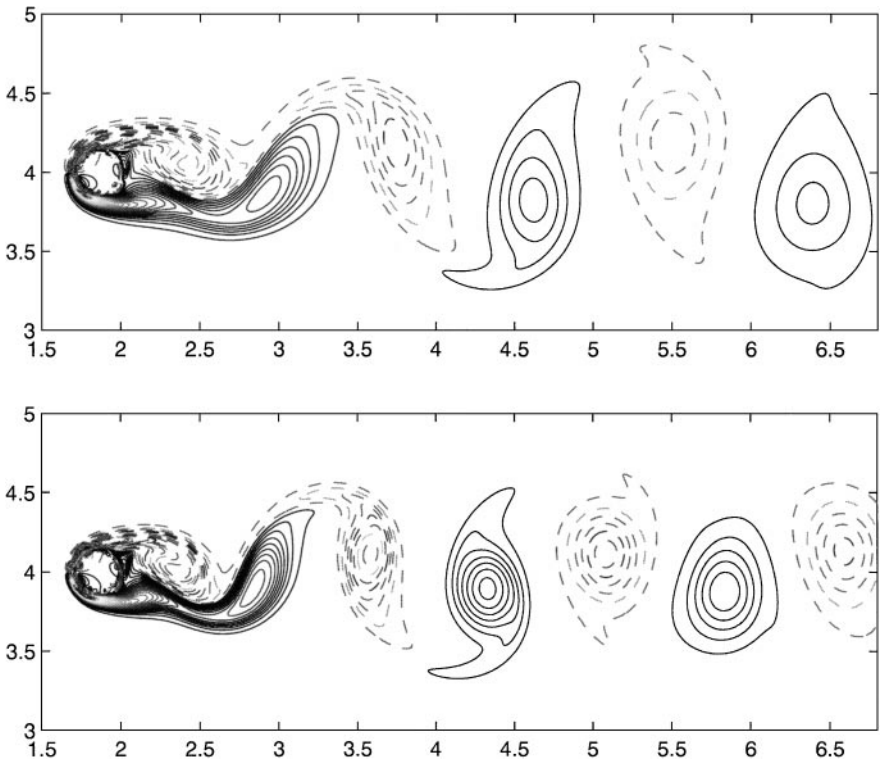


FIG. 3. Vorticity contours of the flow around a cylinder at $Re = 150$. First-order method (top); formally second-order method (bottom).

part of the figure. Notice, too, that the vortices themselves are considerably more diffuse in the first-order results (top) than in the formally second-order results (bottom). It seems likely that this is a consequence of the higher numerical viscosity of the first-order method.

5. CONCLUSION

In this paper, we have proposed a new, formally second-order accurate scheme for the immersed boundary method, and we have tested this methodology by applying it to the problem of flow past a circular cylinder. This test problem was chosen because of the availability of experimental data over a substantial range of Reynolds number, with significant Reynolds number dependent effects that one would like a numerical method to replicate. The previous, first-order accurate scheme of the immersed boundary method has also been tested for comparison.

The most significant differences between the old and the new scheme appear in computations where the flow is unsteady despite the steady boundary conditions, due to alternate vortex shedding from the upper and lower surfaces of the cylinder. Here the Strouhal number (dimensionless frequency of vortex shedding) is about 20% too low in the case of the first-order scheme, but agrees very well with experiment in the case of the formally second-order scheme. In vorticity plots, the shed vortices look considerably more diffuse in the first-order results than in the formally second-order results. Another difference between these computations is that various measures of drag disagree with each other by as much as 20% in the first-order results but are in excellent mutual agreement in the formally second-order results.

Despite its lack of true second-order accuracy, the formally second-order scheme introduced in this paper produces results that are closer to the available experimental data than the previous first-order scheme. This appears to be primarily because the new scheme has less numerical viscosity than the old one. This is a considerable benefit; it helps us to be able to simulate higher Reynolds number flow in immersed boundary problems.

ACKNOWLEDGMENTS

This work is a part of the first author's Ph.D. thesis at Courant Institute of Mathematical Sciences, New York University. It was supported by National Science Foundation under research Grant DMS/FD 92-20719. The computation was performed at the Applied Mathematics Laboratory, New York University, and also at the Pittsburgh Supercomputing Center and at the San Diego Supercomputer Center under an allocation of resources MCA93S004P from the MetaCenter and NRAC Allocation Committees, respectively. The authors are indebted to Olof Widlund for suggesting (many years ago!) the use of skew-symmetric differencing of the advection terms.

REFERENCES

1. K. 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).
2. R. P. Beyer and R. J. Leveque, Analysis of a one-dimensional model for the immersed boundary method, *SIAM J. Numer. Anal.* **29**, 332 (1992).
3. R. P. Beyer, A computational model of the cochlea using the immersed boundary method, *J. Comput. Phys.* **98**, 145 (1992).
4. D. C. Bottino, Modeling viscoelastic networks and cell deformation in the context of the immersed boundary method, *J. Comput. Phys.* **147**, 86 (1998).

5. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York/Berlin, 1988), p. 114.
6. C. D. Eggleton and A. S. Popel, Large deformation of red blood cell ghosts in a simple shear flow, *Phys. Fluids* **10**, 1834 (1998).
7. L. J. Fauci, Interaction of oscillating filaments—A computational study, *J. Comput. Phys.* **86**, 294 (1990).
8. L. J. Fauci and A. L. Fogelson, Truncated Newton methods and the modeling of complex immersed elastic structures, *Comm. Pure Appl. Math.* **46**, 787 (1993).
9. L. J. Fauci and C. S. Peskin, A computational model of aquatic animal locomotion, *J. Comput. Phys.* **77**, 85 (1988).
10. A. L. Fogelson and C. S. Peskin, A fast numerical method for solving three-dimensional Stokes equations in the presence of suspended particles, *J. Comput. Phys.* **79**, 50 (1988).
11. A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, *J. Comput. Phys.* **56**, 111 (1984).
12. E. Givberg, *Modeling Elastic Shells Immersed in Fluid*, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, September 1997 (unpublished).
13. D. Goldstein, R. Handler, and L. Sirovich, Modeling a no-slip flow with an external force field, *J. Comput. Phys.* **105**, 354 (1993).
14. D. Goldstein, R. Handler, and L. Sirovich, Direct numerical simulation of turbulent flow over a modelled riblet covered surface, *J. Fluid Mech.* **302**, 333 (1995).
15. M. C. Lai, *Simulations of the Flow Past an Array of Circular Cylinders as a Test of the Immersed Boundary Method*, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, September 1998 (unpublished).
16. M. C. Lai, Numerical stability analysis of a one-dimensional model for the immersed boundary method, submitted for publication.
17. R. J. Leveque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, 1019 (1994).
18. H. J. Lugt, *Vortex Flow in Nature and Technology* (Wiley, New York, 1983).
19. C. Liu, X. Zheng, and C. H. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* **139**, 35 (1998).
20. D. M. McQueen and C. S. Peskin, Shared-memory parallel vector implementation of the immersed boundary method for the computation of blood flow in the beating mammalian heart, *J. Supercomput.* **11**, 213 (1997).
21. C. S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* **10**, 252 (1972).
22. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
23. C. S. Peskin and D. M. McQueen, A general method for the computer simulation of biological systems interacting with fluids, *Sympos. Soc. Exp. Biol.* **49**, 265 (1995).
24. C. 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*, edited by H. G. Othmer, F. R. Adler, M. A. Lewis, and J. C. Dallon (Prentice-Hall, Englewood Cliffs, NJ, 1996), p. 309.
25. C. S. Peskin and B. F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, *J. Comput. Phys.* **105**, 33 (1993).
26. S. A. Piasek and G. P. Williams, Conservation properties of convection difference schemes, *J. Comput. Phys.* **6**, 392 (1970).
27. M. E. Rosar, *A Three-Dimensional Computer Model for Fluid Flow through a Collapsible Tube*, Ph.D. thesis, Courant Institute of Mathematical Sciences, New York University, 1994 (unpublished).
28. E. M. Saiki and S. Biringen, Numerical simulation of a cylinder in uniform flow: Application of a virtual boundary method, *J. Comput. Phys.* **123**, 450 (1996).
29. J. M. Stockie and S. I. Green, Simulating the motion of flexible pulp fibres using the immersed boundary method, *J. Comput. Phys.* **147**, 147 (1998).
30. D. Sulsky and J. U. Brackbill, A numerical method for suspension flow, *J. Comput. Phys.* **96**, 339 (1991).