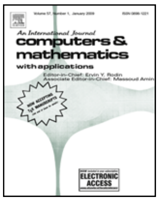
Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa



A lattice Boltzmann based implicit immersed boundary method for fluid–structure interaction

Jian Hao *, Luoding Zhu

Department of Mathematical Sciences and Center for Mathematical Biosciences, Indiana University - Purdue University, Indianapolis, IN 46202, USA

ARTICLE INFO

Article history:
Received 22 June 2009
Accepted 30 June 2009

Keywords:
Immersed boundary method
Implicit scheme
Lattice Boltzmann method
Fluid–structure interaction

ABSTRACT

The immersed boundary method is a practical and effective method for fluid–structure interaction problems. It has been applied to a variety of problems. Most of the time-stepping schemes used in the method are explicit, which suffer a drawback in terms of stability and restriction on the time step. We propose a lattice Boltzmann based implicit immersed boundary method where the immersed boundary force is computed at the unknown configuration of the structure at each time step. The fully nonlinear algebraic system resulting from discretizations is solved by an Inexact Newton–Krylov method in a Jacobian-free manner. The test problem of a flexible filament in a flowing viscous fluid is considered. Numerical results show that the proposed implicit immersed boundary method is much more stable with larger time steps and significantly outperforms the explicit version in terms of computational cost.

1. Introduction

The immersed boundary (IB) method is a very practical and effective method for solving problems of interactions between fluids and elastic structures. It was initially developed by Peskin to model and simulate blood flow in the human heart [1,2]. Since then it has been applied to a variety of problems, such as deformation of red blood cells [3], platelet aggregation in blood clotting [4], swimming of bacteria and sperm [5,6], waving motions of cilia [7], and insect flight [8]. For an extensive list of applications, see the references [9,10]. In an IB method, the fluid equations are discretized on a fixed Eulerian grid over the entire domain; the immersed boundary is discretized on a moving Lagrangian array of points that contribute to the force term in the fluid equations. The interaction of the fluid and the elastic structure is carried out through the Dirac delta function [9]. Thus, the IB method is very powerful in terms of computational efficiency, regardless of the geometry of the structure.

There exist other methods in the literature that are closely related to the IB method. The immersed interface method, a variant of the IB method, was developed by LeVeque and Li to address the first-order accuracy in the IB method for the problems with sharp interfaces [11,12]. The IB method also shares some features in common with the Fictitious Domain (FD) method for fluid–particle interactions in which the flow computation is done on a fixed space domain, and the rigid body motion for the particle is enforced through a Lagrange multiplier in the equations [13]. We refer the reader to [13–16] for the FD method and its applications.

Inspired by application needs, various versions of the IB method have been developed. In addition to the original versions [1,2,17–19], there exist the vortex-method version [20], the volume-conserved version [21,22], the adaptive mesh version [23], the (formally) second-order versions [24–26], the multigrid version [27,10], the penalty version [28], and the stochastic version [29,30]. Most of the time-stepping schemes in the IB methods in the literature are explicit where the

[*] Corresponding author. Tel.: +1 317 278 9616; fax: +1 317 274 3460.
E-mail addresses: jhao@math.iupui.edu (J. Hao), lzhu@math.iupui.edu (L. Zhu).

0898-1221/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.06.055

elastic force is calculated on the known configuration of a structure at each time step. An explicit IB method results in a severe restriction on the time step [9,31]. The dimensionless time step was of the order of 10^{-4} in the work [27,10,32,33]. The author in [34] reported that the dimensional time step for the 3D application was taken as small as the order of 10^{-8} (s).

To overcome the stiffness in the explicit IB method, much effort has been made in recent years to develop implicit or semi-implicit IB methods [11,35–42]. Wang presented an iterative matrix-free implicit immersed boundary/continuum method by using finite element formulations [42]. Mori and Peskin developed implicit second-order IB methods with boundary mass and compared their implicit methods with explicit ones [40]. Newren et al. compared several implicit solvers for their schemes in terms of computational costs [41]. Hou and Shi proposed an efficient semi-implicit scheme to remove the stiffness of the IB method [39]. The authors in [40,41] reported that implicit IB methods sometimes are actually more computationally expensive than the explicit methods. As pointed out by the authors in [39], so far the computational cost of using implicit IB methods is still too high to be effective for any practical computations. One possible reason may be solving the viscous incompressible Navier–Stokes (N–S) equations in the implicit IB methods is too costly.

As a first step to develop a practical implicit 3D IB method, we present a lattice Boltzmann (LB) based implicit immersed boundary method for fluid–structure-interaction problems. The LB method is an alternative approach for solving the viscous incompressible Navier–Stokes equations. It is second order accurate in both time and space, and is very computationally efficient, especially for 3D problems [43–45]. As expected the LB approach significantly reduces the computational cost in solving the N–S equations which renders the implicit IB method appropriate for practical computations. In our implicit IB method, the elastic force is based on the *unknown* configuration of the structure at next time step, instead of on the current known configuration. Consequently a highly nonlinear algebraic system needs to be solved after discretizations. We use a Jacobian-free Inexact Newton–Krylov method to solve the nonlinear system. The numerical results show that our implicit IB method is very stable with larger time steps, and outperforms the corresponding explicit version for the test problem in terms of computational cost.

This paper is organized as follows. In Section 2, we give mathematical formulation for the LB/IB method. In Section 3, we propose an implicit scheme and briefly describe the numerical solvers. In Section 4, we present the numerical results for the test problem. In Section 5, conclusions are drawn.

2. Mathematical formulation by the LB/IB method

2.1. The LB/IB formulation

Consider an elastic structure in a viscous incompressible fluid flow. Choosing appropriate reference quantities for length, velocity and mass density, our lattice Boltzmann IB formulation for general fluid–structure interaction is formulated in dimensionless form as follows:

$$\frac{\partial g(\mathbf{x},\boldsymbol{\xi},t)}{\partial t}+\boldsymbol{\xi}\cdot\frac{\partial g(\mathbf{x},\boldsymbol{\xi},t)}{\partial \mathbf{x}}+\mathbf{f}(\mathbf{x},t)\cdot\frac{\partial g(\mathbf{x},\boldsymbol{\xi},t)}{\partial \boldsymbol{\xi}}=-\frac{1}{\tau}(g(\mathbf{x},\boldsymbol{\xi},t)-g^{(0)}(\mathbf{x},\boldsymbol{\xi},t)), \tag{1}$$

Eq. (1) is the Bhatnagar–Gross–Krook (BGK) equation [46] which is used to describe the motion of both the fluid and the immersed boundary. The function $g(\mathbf{x},\boldsymbol{\xi},t)$ is the single particle velocity distribution function, where $\mathbf{x}$ is the spatial coordinate, $\boldsymbol{\xi}$ is the particle velocity, and t is time. $g(\mathbf{x},\boldsymbol{\xi},t)\mathrm{d}\mathbf{x}\mathrm{d}\boldsymbol{\xi}$ represents the probability of finding a particle at time t located in $[\mathbf{x},\mathbf{x}+\mathrm{d}\mathbf{x}]$ moving with a velocity between $\boldsymbol{\xi}$ and $\boldsymbol{\xi}+\mathrm{d}\boldsymbol{\xi}$. The term $-\frac{1}{\tau}(g-g^{(0)})$ is the well-known BGK approximation to the complex collision operator in the Boltzmann equation, where the τ is the relaxation time (dimensionless). It is connected to the fluid kinematic viscosity ν (dimensionless) in the LBM. The $g^{(0)}$ is the Maxwellian distribution. The external force term $\mathbf{f}(\mathbf{x},t)=\mathbf{f}_{ib}(\mathbf{x},t)+\mathbf{f}_{ext}(\mathbf{x},t)$. The $\mathbf{f}_{ib}(\mathbf{x},t)$ is the force imparted by the immersed boundary to the fluid. The $\mathbf{f}_{ext}(\mathbf{x},t)$ is other external forces acting on the fluid, e.g. the gravity. The macroscopic variables such as fluid mass density (ρ) and momentum ($\rho\mathbf{u}$) can be computed from the velocity distribution function g via Eqs. (2) and (3).

$$\rho(\mathbf{x},t)=\int g(\mathbf{x},\boldsymbol{\xi},t)\mathrm{d}\boldsymbol{\xi}, \tag{2}$$

$$(\rho\mathbf{u})(\mathbf{x},t)=\int g(\mathbf{x},\boldsymbol{\xi},t)\boldsymbol{\xi}\mathrm{d}\boldsymbol{\xi}. \tag{3}$$

The Eulerian force density $\mathbf{f}_{ib}(\mathbf{x},t)$ defined on the fixed Eulerian lattice is calculated from the Lagrangian force density $\mathbf{F}(\alpha,t)$ defined on the Lagrangian grid by Eq. (4). See the Reference [9].

$$\mathbf{f}_{ib}(\mathbf{x},t)=\int \mathbf{F}(\alpha,t)\delta(\mathbf{x}-\mathbf{X}(\alpha,t))\mathrm{d}\alpha \tag{4}$$

where the function $\delta(\mathbf{x})$ is the Dirac δ-function. The Lagrangian force density $\mathbf{F}$ is computed as follows:

$$\mathbf{F}(\alpha,t)=-\frac{\partial\mathcal{E}}{\partial\mathbf{X}}=-\frac{\partial(\mathcal{E}_s+\mathcal{E}_b)}{\partial\mathbf{X}}. \tag{5}$$

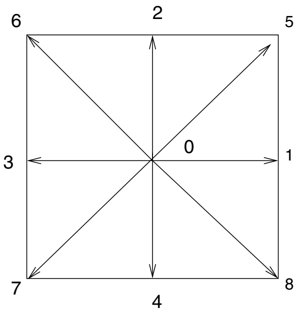


Fig. 1. D2Q9 lattice model.

In Eq. (5) the elastic potential energy density ($\mathcal{E}$) generated by stretching/compression ($\mathcal{E}_s$) and bending ($\mathcal{E}_b$) are defined by Eqs. (6) and (7), respectively.

$$\mathcal{E}_s = \frac{1}{2} K_s \int \left(\left| \frac{\partial \mathbf{X}}{\partial \alpha} \right| - 1 \right)^2 \mathrm{d}\alpha \tag{6}$$

$$\mathcal{E}_b = \frac{1}{2} K_b \int \left| \frac{\partial^2 \mathbf{X}(\alpha, t)}{\partial \alpha^2} \right|^2 \mathrm{d}\alpha. \tag{7}$$

In the above formulation for $\mathcal{E}_s$ and $\mathcal{E}_b$ we assume that the immersed boundary is an elastic fiber. For other immersed boundaries, similar integrals are used for computing the elastic potential energy. The energy density $\mathcal{E} = \mathcal{E}_s + \mathcal{E}_b$. The constant K_s is the stretching/compression coefficient and constant K_b is the bending rigidity. Both constants are related to the Young's modulus of the structure.

The motion of the immersed structure is described by a system of first-order ordinary differential equations, Eq. (8).

$$\frac{\partial \mathbf{X}}{\partial t}(\alpha, t) = \mathbf{U}(\alpha, t). \tag{8}$$

The $\mathbf{X}(\alpha, t)$ is the Eulerian coordinate of the immersed structure at time t whose Lagrangian coordinate is α. The immersed boundary velocity $\mathbf{U}(\alpha, t)$ is interpolated from the fluid velocity $\mathbf{u}(\mathbf{x}, t)$ by the same δ-function as used to convert the force from the boundary to the fluid. See Eq. (9).

$$\mathbf{U}(\alpha, t) = \int \mathbf{u}(\mathbf{x}, t)\delta(\mathbf{x} - \mathbf{X}(\alpha, t))\mathrm{d}\mathbf{x}. \tag{9}$$

Although the Eqs. (6) and (7) are formulated for an elastic fiber, the above lattice Boltzmann IB formulation is generic and applicable to any immersed boundaries. It remains the same for any discretization, explicit or implicit.

3. An implicit immersed boundary method

We consider the 2D case. In order to compare the stability and the computational cost, we first introduce the explicit version of the LB/IB method. Also it will make the presentation of the implicit LB/IB method more convenient.

3.1. Explicit discretization

The above nonlinear system of differential-integral equations (Eqs. (1)–(9)) is discretized on a uniform fixed Eulerian square lattice for the fluid with the uniform mesh width h (the number of grid nodes is N_x and N_y in x and y directions, respectively), plus a collection of moving Lagrangian discrete points for the immersed boundary with mesh width $\Delta\alpha \simeq \frac{1}{2}h$. The D2Q9 model is used to discretize the BGK equation Eq. (1) (see Fig. 1). In the D2Q9 model particles impinging and exiting at each lattice node can move along 8 different directions. They are also allowed to stay together with rest state at the node. Thus, the particle velocity space $\boldsymbol{\xi}$ is discretized by a finite set of 9 velocities. The discrete velocity can be written as $\boldsymbol{\xi}_0 = (0, 0)$, $\boldsymbol{\xi}_1 = (1, 0)$, $\boldsymbol{\xi}_2 = (0, 1)$, $\boldsymbol{\xi}_3 = (-1, 0)$, $\boldsymbol{\xi}_4 = (0, -1)$, $\boldsymbol{\xi}_5 = (1, 1)$, $\boldsymbol{\xi}_6 = (-1, 1)$, $\boldsymbol{\xi}_7 = (-1, -1)$, and $\boldsymbol{\xi}_8 = (1, -1)$.

Let $g_j(\mathbf{x}, t)$ be the distribution function along $\boldsymbol{\xi}_j$. A second-order space and time discretization in a Lagrangian coordinate system is applied to derive the lattice Boltzmann equation (LBE) that advances $g_j(\mathbf{x}, t)$ forward by one step

$$g_j(\mathbf{x} + \boldsymbol{\xi}_j, t + 1) = g_j(\mathbf{x}, t) - \frac{1}{\tau}(g_j(\mathbf{x}, t) - g_j^0(\mathbf{x}, t)) + \left(1 - \frac{1}{2\tau}\right) w_j \left(\frac{\boldsymbol{\xi}_j - \mathbf{u}}{{c_s}^2} + \frac{\boldsymbol{\xi}_j \cdot \mathbf{u}}{{c_s}^4} \boldsymbol{\xi}_j \right) \cdot \mathbf{f}. \tag{10}$$

Here $c_s = c/\sqrt{3}$ is speed of sound of the model, and c is the lattice speed which is as follows for the D2Q9 model: $c = 0$ for $j = 0$; 1 for $j = 1, \dots, 4$; $\sqrt{2}$ for $j = 5, \dots, 8$.

In this model, the relaxation time τ is related to the dimensionless fluid viscosity ν by the equation $\nu = \frac{2\tau-1}{6}$. We treat the external forces in Eq. (10) in a similar way to that proposed by Guo et al. in [47]. It can be proved that this treatment is more accurate than the one used in the IB-LBM [48,49]. Here the fluid velocity $\mathbf{u}$, forces $\mathbf{f}_{ib}$ and $\mathbf{f}_{ext}$ are evaluated at time t.

The macroscopic variables such as density $\rho(\mathbf{x}, t)$ and momentum $\rho\mathbf{u}(\mathbf{x}, t)$ are related to the $g_j(\mathbf{x}, t)$ at each node by

$$\rho(\mathbf{x}, t) = \sum_j g_j(\mathbf{x}, t), \tag{11}$$

$$(\rho\mathbf{u})(\mathbf{x}, t) = \sum_j \boldsymbol{\xi}_j g_j(\mathbf{x}, t) + \frac{\mathbf{f}(\mathbf{x}, t)}{2}. \tag{12}$$

For an isothermal fluid, the equilibrium distribution function g_j^0 (which is a function of ρ and $\mathbf{u}$) in the D2Q9 model is given by

$$g_j^0(\mathbf{x}, t) = \rho(\mathbf{x}, t) w_j \left(1 + 3\boldsymbol{\xi}_j \cdot \mathbf{u}(\mathbf{x}, t) + \frac{9}{2}(\boldsymbol{\xi}_j \cdot \mathbf{u}(\mathbf{x}, t))^2 - \frac{3}{2}\mathbf{u}(\mathbf{x}, t) \cdot \mathbf{u}(\mathbf{x}, t)\right) \tag{13}$$

where w_j is the weight, which takes the values: $w_j = 4/9$ for $j = 0$; $1/9$ for $j = 1, \dots, 4; 1/36$ for $j = 5, \dots, 8$.

The bounce-back scheme is used to model the no-slip boundary condition for a *fixed* rigid wall. Notice that no special treatment for the freely-moving immersed flexible boundary is needed on the LBM part. It is handled by the IB method through the immersed boundary force. The LBE “feels” the existence of the immersed flexible boundary through the force.

The duration of the time step is set to 1 in the LBM. Let n be the time step index: $g^n = g(\mathbf{x}, \boldsymbol{\xi}, n)$, $\mathbf{X}^n(\alpha) = \mathbf{X}(\alpha, n)$, $\mathbf{u}^n = \mathbf{u}(\mathbf{x}, n)$, $p^n = p(\mathbf{x}, n)$, $\rho^n = \rho(\mathbf{x}, n)$. Let each fiber be represented by a discrete collection of points whose Lagrangian coordinate is α. Let $\alpha = m\Delta\alpha$, where m is an integer. The “half-integer” points are given by $\alpha = (m + 1/2)\Delta\alpha$. For any function $\phi(\alpha)$, let

$$(D_\alpha\phi)(\alpha) = \frac{\phi(\alpha + \frac{\Delta\alpha}{2}) - \phi(\alpha - \frac{\Delta\alpha}{2})}{\Delta\alpha}. \tag{14}$$

Then the stretching energy and the corresponding force are discretized as the following:

$$\mathcal{E}_s = \frac{1}{2}K_s \sum_m (|D_\alpha \mathbf{x}| - 1)^2 \Delta\alpha = \frac{1}{2}K_s \sum_{m=1}^{n_f-1} \left(\frac{|\mathbf{X}_{m+1} - \mathbf{X}_m|}{\Delta\alpha} - 1\right)^2 \Delta\alpha \tag{15}$$

$$(\mathbf{F}_s)_l = \frac{K_s}{(\Delta\alpha)^2} \sum_{m=1}^{n_f-1} (|\mathbf{X}_{m+1} - \mathbf{X}_m| - \Delta\alpha) \frac{\mathbf{X}_{m+1} - \mathbf{X}_m}{|\mathbf{X}_{m+1} - \mathbf{X}_m|}(\delta_{ml} - \delta_{m+1,l}). \tag{16}$$

Here $(\mathbf{F}_s)_l$, $l = 1, 2$, is the component of $\mathbf{F}_s$ along the l direction. The bending energy and the corresponding force are discretized as follows:

$$\mathcal{E}_b = \frac{1}{2}K_b \sum_m |D_\alpha D_\alpha \mathbf{X}|^2 \Delta\alpha = \frac{1}{2}K_b \sum_{m=2}^{n_f-1} \left[\frac{|\mathbf{X}_{m+1} + \mathbf{X}_{m-1} - 2\mathbf{X}_m|^2}{(\Delta\alpha)^4}\right] \Delta\alpha \tag{17}$$

$$(\mathbf{F}_b)_l = \frac{K_b}{(\Delta\alpha)^4} \sum_{m=2}^{n_f-1} (\mathbf{X}_{m+1} + \mathbf{X}_{m-1} - 2\mathbf{X}_m)(2\delta_{ml} - \delta_{m+1,l} - \delta_{m-1,l}). \tag{18}$$

Here $(\mathbf{F}_b)_l$, $l = 1, 2$, is the component of $\mathbf{F}_b$ along the l direction; n_f is the total number of grid points of the fiber, the δ_{kl} is the Kronecker symbol.

Note that the total Lagrangian force density $\mathbf{F}(\alpha, t) = \mathbf{F}_s(\alpha, t) + \mathbf{F}_b(\alpha, t)$. The two integral relations can be discretized as follows:

$$\mathbf{f}_{ib}^n(\mathbf{x}) = \sum_\alpha \mathbf{F}^n(\alpha)\delta_h(\mathbf{x} - \mathbf{X}^n(\alpha))\Delta\alpha \tag{19}$$

$$\mathbf{U}^{n+1}(\alpha) = \sum_{\mathbf{x}} \mathbf{u}^{n+1}(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}^n(\alpha))h^2. \tag{20}$$

Here the notation $\sum_\alpha$ means the sum over all the discrete collection of points (sum over all discrete points of a fiber in the form $\alpha = m\Delta\alpha$, where m is an integer). The notation $\sum_{\mathbf{x}}$ means the sum over all the discrete points of the form

$\mathbf{x} = (ih, jh)$, where i, and j are integers, h is the mesh width. The δ_h is a smoothed approximation of the Dirac δ function. In the IB method, the δ_h is written in the following form:

$$\delta_h(\mathbf{x}) = h^{-2}\psi\left(\frac{x}{h}\right)\psi\left(\frac{y}{h}\right) \tag{21}$$

where h is the mesh spacing, $\mathbf{x} = (x, y)$, and ψ is chosen as:

$$\psi(r) = \begin{cases} \frac{1}{4}\left(1 + \cos\left(\frac{\pi r}{2}\right)\right), & \text{if } |r| \le 2 \\ 0, & \text{otherwise.} \end{cases}$$

Note that the support of δ_h is a square with width $4h$ at each discrete fiber point. With $\mathbf{U}^{n+1}(\alpha)$ known from Eq. (20), the structure motion equations are discretized as follows:

$$\mathbf{X}^{n+1}(\alpha) = \mathbf{X}^{n}(\alpha) + \mathbf{U}^{n+1}(\alpha). \tag{22}$$

An *explicit version* of LB/IB scheme is summarized as follows.

Given the values of all variables at time step n, then:

(1) compute elastic force $\mathbf{F}^{n+1}$ from $\mathbf{X}^n$ by Eqs. (16) and (18),
(2) spread the Lagrangian force density $\mathbf{F}^{n+1}$ onto fluid lattice by Eq. (19),
(3) compute particle collision, i.e. calculate $\frac{1}{\tau}(g - g^0)$ in Eq. (10) via equilibrium velocity distribution from Eq. (13),
(4) update velocity distribution function via streaming and external forcing by Eq. (10),
(5) compute new fluid velocity $\mathbf{u}^{n+1}$ from Eqs. (11) and (12),
(6) interpolate velocity of the immersed boundary $\mathbf{U}^{n+1}$ from the velocity $\mathbf{u}^{n+1}$ of the ambient fluid by Eq. (20),
(7) finally, update the configuration of the immersed boundary (i.e. computing $\mathbf{X}^{n+1}(\alpha)$) via Eq. (22).

3.2. An implicit scheme

The general idea of our implicit scheme is as follows: Suppose all variables are known at a discrete time step indexed by n, and we want to compute the values of variables at next time step $n+1$. In an explicit method, the IB force $\mathbf{f}_{ib}$ is computed from the known configuration of the immersed structure at previous time step n. In an implicit method, $\mathbf{f}_{ib}$ is computed from the *unknown* configuration at next time step $n + 1$. This results in a large-scale highly nonlinear system of algebraic equations as follows. Let $\mathcal{F}$ denote the operator acting on the configuration $\mathbf{X}(\alpha, t + 1)$ to produce the Lagrangian elastic force density, $\mathcal{S}$ denote the spreading operator of Lagrangian force density to fluid lattice, $\mathcal{L}$ denote the operator to advance the velocity distribution function from n to $n+1$, $\mathcal{U}$ denote the operator to recover fluid velocity from distribution function, $\mathcal{I}$ denote the operator to interpolate the immersed boundary velocity from the fluid, then the nonlinear algebraic equation system (after g^{n+1} is eliminated) for advancing the solutions from n to $n + 1$ is given by (note the time step size Δt is 1)

$$\mathcal{I}\mathcal{U}\mathcal{L}\mathcal{S}\mathcal{F}\mathbf{X}^{n+1} = \frac{\mathbf{X}^{n+1} - \mathbf{X}^{n}}{\Delta t} \tag{23}$$

Note that the particle velocity distribution functions are eliminated and only the unknown $\mathbf{X}^{n+1}$ is present in the above system of nonlinear algebraic equations. Still the above nonlinear system is very complex, and the Jacobian of the system is not available. A Jacobian-free Newton–Krylov (JFNK) method is applied to solve the nonlinear system (23). The Krylov method we use here is the GMRES (Generalized Minimal Residual Method). It is used to solve the linear system at each Newton's step for updating the correction to the solution of the nonlinear system. Also note that the implicit IB scheme we propose here is similar to the one in [40]. The major differences are: (1) the LB method is used for solving the N–S equations in our work while a conventional approach was used in [40]; (2) the JFNK is used to solve the discrete nonlinear algebraic systems while the DFFT (Discrete Fast Fourier Transform) was used to solve the linearized discrete system.

3.3. Implementation of the implicit IB method

The above formulation (23) of the implicit IB method via the lattice Boltzmann approach is generic: it remains the same for both two and three dimensions (in space). The only difference is that the dimensions of the unknown vector $\mathbf{X}$ (i.e. the configuration of the immersed structure) depending on the specific structure of the immersed object is usually smaller in 2D than in 3D (in space). To test and validate our implicit IB method, the implementation in FORTRAN has been done in the case of two dimensions (spatial) first.

Consider a nonlinear algebraic system:

$$F(u) = 0, \qquad F : R^N \to R^N \tag{24}$$

which is an abstraction of the nonlinear system resulted from our implicit IB method, i.e. Eq. (23). A well tested and widely used existing software package SUNDIAL, specifically the subroutine KINSOL [50] is used to solve the nonlinear system (24). KINSOL was written in C, but provides the FORTRAN 77 interface FKINSOL which was used in our applications.

One feature of the nonlinear system solver KINSOL is that it employs the Inexact Newton's method developed by Brown and Saad in 1990 [51] which results in the following iteration:

Inexact Newton iteration

1. Given an initial guess u_0
2. For $n = 0, 1, 2, \ldots$ until convergence do:
   (a) Solve $J(u_n)\delta_n = -F(u_n)$
   (b) Set $u_{n+1} = u_n + \delta_n$
   (c) Test for convergence.

Here J is the Jacobian of the nonlinear system. The second feature is that the linear iterative method during each Newton's step is the Krylov method GMRES [52]. The linear solver is by default applied in a matrix-free manner, with (Jacobian) matrix–vector products obtained by a finite difference quotient of the form:

$$J(u)v \approx \frac{F(u + \sigma v) - F(u)}{\sigma}, \tag{25}$$

where u is the current solution, and σ is a scalar, appropriately chosen to minimize numerical error, v is any vector to be multiplied to $J(u)$.

In our numerical simulations, the stopping tolerance for Newton's method is either $\|F\|_{L_2} \leq 10^{-20}$ or $\|\delta_n\|_{L_2} \leq 10^{-5}$. Here F is the nonlinear function in Eq. (24), δ_n is the correction to Newton's step. The maximum Krylov subspace dimension is 10. The number of Newton's iterations is around 3–4 and the number of linear iterations is around 3–9. The solver appears to be quite efficient without a preconditioner in our test/trial problem. Perhaps this is because it is used for each lattice Boltzmann time step whose physical time step is relatively small compared to the dimensionless time step size 1.0 in the LB units. The largest dimensionless physical time step (in contrast with the lattice Boltzmann time step 1) of the implicit IB method is 0.0178, for the problem with Reynolds number $Re = 122$ on lattice size 64 × 128. If even greater physical time step is desired (when the accuracy is not a concern), more iterations (both Newton's and the GMRES') are needed. An approximate diagonal matrix as right preconditioner did not expedite the convergence much. A proper preconditioner may be needed to speed up the convergence in this case. Given the fact that the dimensionless time step size was on the order of 10^{-4} in several versions of the IB method used by the author while the implicit IB method by the lattice Boltzmann approach is able to take $\Delta t = 1.78 \times 10^{-2}$. This is a big step forward.

After $\mathbf{X}^{n+1}$ is known, the velocity $\mathbf{u}^{n+1}$ can be computed by

$$\mathbf{u}^{n+1}(\mathbf{x}, t) = \mathcal{U}\mathcal{L}\mathcal{S}\mathcal{F}\mathbf{X}^{n+1}.$$

Thus the solution is advanced forward by one step from n to $n + 1$.

4. Numerical results

4.1. Test problem

We consider a flexible (massless) filament with its upper tip fixed interacting with a flowing viscous fluid in a rectangular domain. The fluid flows from top to bottom. The filament is initially placed in a fixed angle (see Fig. 2), and is carried away by the flowing-by fluid. Because the upper end is fixed, the filament oscillates sideways for a period of time before it settles down to an equilibrium state (just as an elevated free pendulum reaches a static state after oscillation for some time, but the filament is flexible in our case). The parameters used here are given below. The fluid domain is $[0, 0.09] \times [0, 0.18]$ (m^2); the length of the filament is 0.0165 (m); the inflow velocity is −2 (m/s). The kinematic viscosity ν (dimensionless) is 2.7×10^{-4}. The constant $\hat{K}_b$ for bending is 0.01; $\hat{K}_s$ for stretching is 5.0; $\hat{K}_{st}$ for the virtual fixed end is 117.335. The Reynolds number Re is 122.

4.2. Validation of the implicit IB method

(a) Qualitative. Fig. 2 plots the positions of the filament immersed in the fluid for several instants, where the dashed curve indicates the filament approaching the equilibrium at $t = 22$. The filament approximately pointed to the flow direction, as pointed out in the paper [10]. Fig. 3 shows the vorticity contours for different Reynolds numbers: $Re = 122$ on the left panel and $Re = 611$ on the right panel at time $t = 22$. The lattice is 128 × 256. Notice that more intensive vortex shedding is seen for the $Re = 611$ case than the $Re = 122$ case, and the boundary layer on the two side walls are substantially thinner in the higher Reynolds number case (because the thickness of the boundary layer is proportional to $\frac{1}{\sqrt{Re}}$). All these results are physically correct and in agreement with those in the literature [10,27].

(b) Quantitative. We use three different lattices 128 × 256, 256 × 512, and 512 × 1024 for the convergence test. The physical time step size is halved when the spatial grid width is halved. All the dimensionless parameters are kept the same on the three lattices. The corresponding relaxation time τ (dimensionless) is 0.530, 0.560, and 0.620, respectively. The filament mesh width $\Delta\alpha$ (dimensionless) is approximately 0.5. So there are 48, 95, and 189 total Lagrangian discrete points for

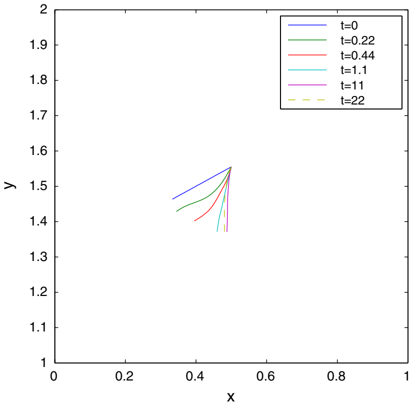


Fig. 2. Filament position versus time on a 128 × 256 lattice. Positions of the filament immersed in the fluid at several instants; the dashed curve indicates the filament approaching the equilibrium at $t = 22$.

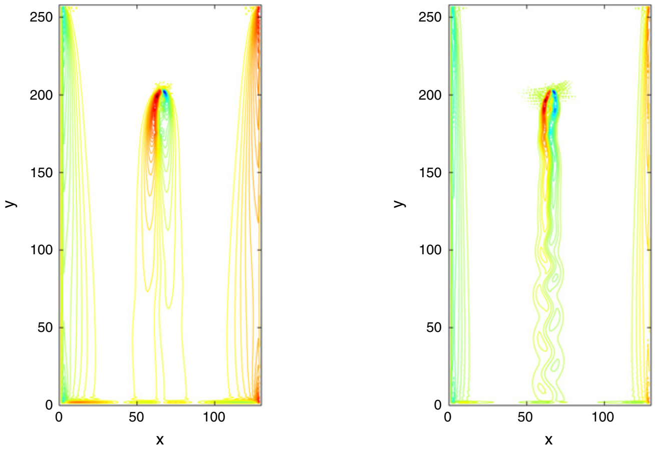


Fig. 3. Vorticity contours for a filament immersed in the fluid for $Re = 122$ (left) and $Re = 611$ (right) at $t = 22$. The lattice size is 128 × 256 and the dimensionless physical time step $\Delta t = 0.0022$.

the filament on the three lattices, respectively. The time step Δt for the lattice 128 × 256 is 0.0022 (dimensionless). We should point out that the (nearly) maximum time step Δt for the above lattice to maintain the stability can be as large as 0.0089 (dimensionless). The free end of the filament is a special and important point for the simulation. Fig. 4 shows the x-component V_x and y-component V_y of the velocity (dimensionless) of the free end of the filament versus time for the three lattices. The figure shows that the velocity of the free end is convergent in both space and time. Both velocity components V_x and V_y change quickly during the early stage of the interaction; the change of the velocity is damped after $t = 1.1$; both velocity components become almost zero after $t = 4.4$. There are some minor oscillations when t is between 11 and 14.3. Fig. 5 shows two velocity profiles at $t = 22$ for different lattices at $y = 1.0$ and $y = 1.75$, when the filament is at equilibrium. The velocity profile $V_y(x, 1.0)$ includes a fluid–filament-interaction region, while the profile $V_y(x, 1.75)$ represents the velocity field essentially without such an interaction (far away from the filament vertically). In both locations the V_y is convergent in space.

We computed the ratio $\frac{\|u_{4h}-u_{2h}\|_2}{\|u_{2h}-u_h\|_2}$ to estimate the order of convergence for the implicit IB method, where h refers to the maximum mesh size. For the lattice 512 × 1024, the mesh size $h = 1/512$. The calculated estimates of the order of convergence based on different mesh steps are given in Table 1. The estimated order of convergence for $h = 1/512$ is 0.96, while the estimated order for $h = 1/256$ is 1.52. It indicates that our implicit numerical method converges and it is first order due to the fluid–filament interaction, although the lattice Boltzmann method itself is second order accurate in both space and time for solving the Navier–Stokes equations. The first order of convergence is expected for our test problem because of the singularity inherited in the IB formulation for a 1D immersed boundary in a 2D flow.

We have done numerous simulations with larger time steps for the test problem to check the stability. Our proposed implicit IB method is much more stable than the corresponding explicit version. For the test problem with $Re = 122$ and lattice size 256 × 512, the (nearly) maximum time step Δt for the explicit method is around 1.39×10^{-4} (dimensionless), while the (nearly) maximum time step Δt for the proposed implicit method is around 4.44×10^{-3} (dimensionless). The time step of the implicit method for this case is 30 times larger than the one of the explicit version. To compare computational

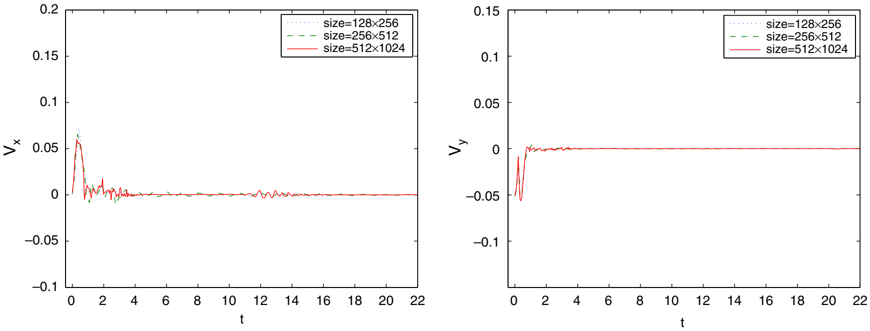


Fig. 4. Convergence test (a). The x-component V_x (left) and y-component V_y (right) of the velocity for the free end of the filament changed over time for different lattices.

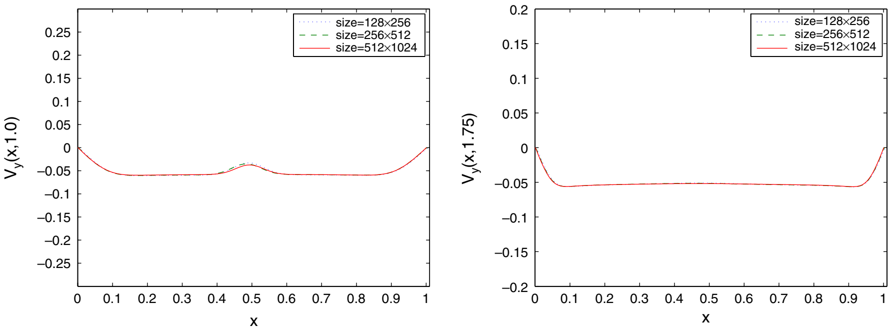


Fig. 5. Convergence test (b). The cross section profiles of the velocity field at $t = 22$ for different lattices at $y = 1.0$ (left) and $y = 1.75$ (right). The filament was approximately at equilibrium.

Table 1
Estimation of the order of convergence.

Mesh step	Ratio value	Estimated order
$h = 1/512$	1.95	0.96
$h = 1/256$	2.86	1.52

cost, we run the above two simulations up to $t = 22$. The computational cost for the implicit method is only 16.5% of the cost for the explicit method. Numerical simulations indicate that our implicit method is much more stable and significantly outperforms the corresponding explicit version.

5. Conclusions

We have proposed a 2D implicit immersed boundary method via the lattice Boltzmann approach. Validation of the method has been performed on a fluid–filament-interaction problem. The numerical results for the test problem are physically correct and in agreement with those in the literature. The results show that the proposed implicit IB method is much more stable with larger time steps and significantly outperforms the explicit version. It appears that the implicit IB method can be used in practical computations. We plan to apply the proposed implicit IB method to more complicated application problems in the near future.

Acknowledgments

The authors thank Ray Chin for the helpful discussion. This work was supported by National Science Foundation Grant number: DMS-0713718, and was also supported by the Center for Mathematical Biosciences at Indiana University - Purdue University Indianapolis.

References

[1] C.S. Peskin, Flow patterns around heart valves: A digital computer method for solving the equations of motion, PhD thesis, Albert Einstein Coll. Med., 1972.
[2] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[3] C.D. Eggleton, A.S. Popel, Large deformation of red blood cell ghosts in a simple shear flow, Phys. Fluids 10 (1998) 1834–1845.
[4] A.L. Fogelson, Continuum models of platelet aggregation: Formulation and mechanical properties, SIAM J. Appl. Math. 52 (1992) 1089–1110.
[5] R. Dillon, L. Fauci, D. Gaver, A microscale model of bacterial swimming, chemotaxis and substrate transport, J. Theor. Biol. 177 (1995) 325–340.
[6] L.J. Fauci, A. McDonald, Sperm motility in the presence of boundaries, B. Math. Biol. 57 (1994) 679–699.
[7] R. Dillon, L. Fauci, An integrative model of internal axoneme mechanics and external fluid dynamics in ciliary beating, J. Theor. Biol. 207 (2000) 415–430.
[8] L.A. Miller, C.S. Peskin, When vortices stick: An aerodynamic transition in tiny insect flight, J. Exp. Biol. 207 (2004) 3073–3088.
[9] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[10] L. Zhu, C.S. Peskin, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, J. Comput. Phys. 179 (2002) 452–468.
[11] R.J. LeVeque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.
[12] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.
[13] R. Glowinski, T.-W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow, J. Comput. Phys. 169 (2001) 363–427.
[14] T.-W. Pan, R. Glowinski, Direct simulation of the motion of neutrally buoyant circular cylinders in plane Poiseuille flow, J. Comput. Phys. 181 (2002) 260–279.
[15] J. Hao, T.-W. Pan, R. Glowinski, D.D. Joseph, A fictitious domain/distributed Lagrange multiplier method for the particulate flow of Oldroyd-B fluids: A positive definiteness preserving approach, J. Non-Newtonian Fluid Mech. 156 (2009) 95–111.
[16] J. Hao, T.-W. Pan, S. Canic, R. Glowinski, D. Rosenstrauch, A fluid–cell interaction and adhesion algorithm for tissue-coating of cardiovascular implants, Multiscale Model. Simul. 7 (2009) 1669–1694.
[17] C.S. Peskin, D.M. McQueen, Computational biofluid dynamics, Contemp. Math. 141 (1993) 161–186.
[18] C.S. Peskin, D.M. McQueen, A general method for the computer simulation of biological systems interacting with fluids, Sympos. Soc. Exp. Biol. 49 (1995) 265–276.
[19] C.S. Peskin, D.M. McQueen, Fluid dynamics of the heart and its valves, in: H.G. Othmer, F.R. Adler, M.A. Lewis, J.C. Dallon (Eds.), Case Studies in Mathematical Modeling: Ecology, Physiology, and Cell Biology, Prentice-Hall, NJ, 1996, pp. 309–337.
[20] M.F. McCracken, C.S. Peskin, A vortex method for blood flow through heart valves, J. Comput. Phys. 35 (1980) 183–205.
[21] C.S. Peskin, B.F. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1993) 33–46.
[22] M.E. Rosar, C.S. Peskin, Fluid flow in collapsible elastic tubes: A three-dimensional numerical model, New York J. Math. 7 (2001) 281–302.
[23] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (1999) 509–534.
[24] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.
[25] D.M. McQueen, C.S. Peskin, L. Zhu, The immersed boundary method for incompressible fluid–structure interaction, in: Proceedings of the First M.I.T. Conference on Computational Fluid and Solid Mechanics, Boston, USA, 2001, pp. 26–30.
[26] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficient smooth problems, J. Comput. Phys. 208 (2005) 75–105.
[27] L. Zhu, Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method, PhD thesis, Courant Institute, New York University, 2001.
[28] Y. Kim, C.S. Peskin, Penalty immersed boundary method for an elastic boundary with mass, Phys. Fluids 19 (2007) 053103.
[29] P.J. Atzberger, P.R. Kramer, C.S. Peskin, A stochastic immersed boundary method for fluid–structure dynamics at microscopic length scales, J. Comput. Phys. 224 (2007) 1255–1292.
[30] P.J. Atzberger, P.R. Kramer, Error analysis of a stochastic immersed boundary method incorporating thermal fluctuations, Math. Comput. Simul. 79 (2008) 379–408.
[31] J.M. Stockie, B.R. Wetton, Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes, J. Comput. Phys. 154 (1999) 41–64.
[32] L. Zhu, C.S. Peskin, Interaction of two flexible filaments in a flowing soap film, Phys. Fluids 15 (2003) 1954–1960.
[33] L. Zhu, Scaling laws for drag of a compliant body in an incompressible viscous flow, J. Fluid Mech. 607 (2008) 387–400.
[34] E. Givelberg, Modeling elastic shells immersed in fluid, Comm. Pure Appl. Math. 57 (2004) 283–309.
[35] C. Tu, C.S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods, SIAM J. Sci. Stat. Comput. 13 (1992) 1361–1376.
[36] A.A. Mayo, C.S. Peskin, An implicit numerical method for fluid dynamics problems with immersed elastic boundaries, Contemp. Math. 141 (1993) 261–277.
[37] L.J. Fauci, A.L. Fogelson, Truncated Newton methods and the modeling of complex elastic structures, Comm. Pure Appl. Math. 46 (1993) 787–818.
[38] K. Taira, T. Colonius, The immersed boundary method: A projection approach, J. Comput. Phys. 225 (2007) 2118–2137.
[39] T.Y. Hou, Z. Shi, An efficient semi-implicit immersed boundary method for the Navier–Stokes equations, J. Comput. Phys. 227 (2008) 8968–8991.
[40] Y. Mori, C.S. Peskin, Implicit second-order immersed boundary methods with boundary mass, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2049–2067.
[41] E.P. Newren, A.L. Fogelson, R.D. Guy, R.M. Kirby, A comparison of implicit solvers for the Immersed Boundary equations, Comput. Methods Appl. Mech. Engrg. 197 (2008) 2290–2304.
[42] X.S. Wang, An iterative matrix-free method in implicit immersed boundary/continuum methods, Comput. Struct. 85 (2007) 739–748.
[43] D.A. Wolf-Gladrow, Lattice Gas Cellular Automata and Lattice Boltzmann Models: An Introduction, Springer, Berlin, 2000.
[44] S.Y. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Ann. Rev. Fluid Mech. 30 (1998) 329–364.
[45] L. Zhu, D. Tretheway, L. Petrold, C. Meinhart, Simulation of fluid slip at 3D hydrophobic microchannel walls by the lattice Boltzmann method, J. Comput. Phys. 202 (2005) 181–195.
[46] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases, I: Small amplitude process in charged and neutral one-component system, Phys. Rev. 94 (1954) 511–525.
[47] Z. Guo, C. Zheng, B. Shi, Discrete lattice effects on the forcing term in the lattice Boltzmann method, Phys. Rev. E 65 (2002) 046308.
[48] Z.G. Feng, E.E. Michaelides, The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems, J. Comput. Phys. 195 (2004) 602–628.
[49] Z.G. Feng, E.E. Michaelides, Proteus: A direct forcing method in the simulations of particulate flows, J. Comput. Phys. 202 (2005) 20–51.
[50] A.C. Hindmarsh, P.N. Brown, K.E. Grant, S.L. Lee, R. Serban, D.E. Shumaker, C.S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Trans. Math. Software 31 (2005) 363–396.
[51] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Stat. Comput. 11 (1990) 450–481.
[52] P.N. Brown, A.C. Hindmarsh, Reduced storage matrix methods in stiff ODE systems, J. Appl. Math. Comput. 31 (1989) 49–91.