Journal of Computational Physics 231 (2012) 6159–6172

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

Partially implicit motion of a sharp interface in Navier–Stokes flow

J. Thomas Beale

Department of Mathematics, Duke University, Box 90320, Durham, NC 27708, USA

ARTICLE INFO

Article history:
Received 5 December 2011
Received in revised form 11 April 2012
Accepted 13 May 2012
Available online 26 May 2012

Keywords:
Fluid–structure interaction
Immersed boundary
Immersed interface
Navier–Stokes flow
Implicit methods
Small scale decomposition

ABSTRACT

We develop a numerical method for the coupled motion of Navier–Stokes flow with an elastic interface of zero thickness which exerts tension and bending forces on the fluid. The interface motion is made partially implicit by approximating a backward Euler step in the high wavenumbers as in the small scale decomposition method of Hou, Lowengrub and Shelley. This modified step is combined with the method of Beale and Layton [J.T. Beale, A.T. Layton, A velocity decomposition approach for moving interfaces in viscous fluids, J. Comput. Phys. 228 (2009) 3358–67]; the fluid velocity is found by computing the Stokes velocity and a more regular remainder. The resulting scheme is second order in space and first order in time; it can be made second order in time by extrapolation. The discontinuities in the pressure and velocity gradient are preserved. The partially implicit method allows much larger time steps than an explicit method with negligible added effort. The formulas in the Fourier transform for the implicit approximation in high wavenumbers are similar to those derived in Hou and Shi [T.Y. Hou, Z. Shi, An efficient semi-implicit immersed boundary method for the Navier–Stokes equations, J. Comput. Phys. 227 (2008) 9138–69] in a different context.

1. Introduction

We are concerned with models for the coupled motion of a viscous, incompressible fluid and an immersed material interface which imposes an elastic force on the fluid in response to its deformation. We assume the fluid is described by the Navier–Stokes equations (NSE) and the interface is a closed curve with zero thickness. There has been extensive development of numerical methods in which the fluid pressure and velocity are computed on a rectangular grid and the interface is represented separately by Lagrangian variables and moved with the fluid velocity. The most widely used method is the immersed boundary method (IBM) introduced by Peskin [26,8,23,24,27,33] in which the interfacial force is conveyed to the fluid by a carefully designed discrete delta function on the grid points. In another class of methods, including the immersed interface method (IIM) the interfacial force is incorporated by imposing jump conditions directly on the fluid variables [19,17,18,20,31,36].

It has long been recognized that the time step in such methods can be severely limited if the motion of the interface is explicit. Considerable effort has led to the design of implicit and semi-implicit methods to alleviate the difficulty [5,16,23–25,33]. Another approach, however, is to identify the source of greatest stiffness arising from small scales or large wavenumbers in the interface motion and to modify the velocity to approximate an implicit step in the high wavenumbers, thereby avoiding the iterative solves needed for a fully implicit method. This was the approach introduced in [9] for certain interface models. It has been successfully used for elastic interfaces in Stokes flow [10,14,30,32,34,15]. This approach was also applied in [11] for NSE flow using the IBM. In the present work we develop a partially implicit time-stepping procedure for interfaces

E-mail address: beale@math.duke.edu
URL: http://www.math.duke.edu/faculty/beale

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.05.018

in NSE flow appropriate for sharp interface methods such as the IIM. As in [3] we decompose the Navier–Stokes velocity as the sum of the Stokes velocity and a remainder which is less singular at the interface. The resulting method is only slightly more involved at each time step than an explicit method; the time step can be much larger than for an explicit method but less than that of an implicit method. Such a method should be useful when the flow is not far from the Stokes regime but the full NSE evolution is desired.

We assume the fluid flow is two-dimensional and periodic on a rectangular domain Ω, and the interface is a closed curve Γ inside Ω. We write the interface as a function $\mathbf{X} = \mathbf{X}(\alpha, t)$ of a material coordinate α, $-A \leqslant \alpha \leqslant A$. We denote the unit tangent and normal vectors by $\mathbf{t} = (\partial\mathbf{X}/\partial\alpha)/|\partial\mathbf{X}/\partial\alpha|$ and $\mathbf{n}$, and the current arclength by s, so that $s_\alpha = \partial s/\partial\alpha = |\mathbf{X}_\alpha|$; our notation is slightly different from that used with the IBM. We suppose the fluid has the same constant density on both sides of Γ. The Navier–Stokes equations in nondimensionalized form are

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \nu\nabla^2\mathbf{u} + \mathbf{F} + \mathbf{G} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{2}$$

where $\mathbf{u}$ is the fluid velocity, p is the pressure, ν is the kinematic viscosity. Here $\mathbf{G}$ is a background force and $\mathbf{F}$ is the interfacial force, supported on Γ; that is, $\mathbf{F} = \mathbf{f}\delta_\Gamma$, where the density $\mathbf{f}$ is a function on Γ, so that for a test function $w(\mathbf{x})$ on Ω

$$\int_\Omega \mathbf{F}(\mathbf{x})w(\mathbf{x})d\mathbf{x} = \int_\Gamma \mathbf{f}(\mathbf{x}(\mathbf{s}), t)w(\mathbf{x}(\mathbf{s}))ds = \int_{-A}^{A} \mathbf{f}(\mathbf{X}(\alpha, t), t)w(\mathbf{X}(\alpha, t))s_\alpha d\alpha \tag{3}$$

We consider forces due to stretching and bending, $\mathbf{f} = \mathbf{f}^{(s)} + \mathbf{f}^{(b)}$, with $\mathbf{f}^{(s)}$ a function only of s_α. As in [26] we suppose $\mathbf{f}^{(s)}$ is determined by an energy density $\boldsymbol{\epsilon}(s_\alpha)$, with the form $\mathbf{f}^{(s)} = (\partial/\partial s)(\epsilon'(s_\alpha)\mathbf{t})$. The most familiar case is $\boldsymbol{\epsilon}(\sigma) = (\gamma_0/2)(\sigma-1)^2$, so that

$$\mathbf{f}^{(s)}(\alpha, t) = \gamma_0\partial_s((s_\alpha - 1)\mathbf{t}), \tag{4}$$

a linear response to the stretching of the material from its natural length. We consider a bending force of the form

$$\mathbf{f}^{(b)} = -c_b\partial_s^4\mathbf{X} = -c_b\partial_s^3\mathbf{t}. \tag{5}$$

It will be important that $\mathbf{f}$ always has the form $\mathbf{f} = \partial_s\Phi$. The interfacial force can be expressed in jump conditions for the pressure and velocity gradient (see e.g. [18,20,27]),

$$[p] = \mathbf{f}\cdot\mathbf{n}, \quad \left[\frac{\partial p}{\partial n}\right] = \frac{\partial}{\partial s}(\mathbf{f}\cdot\mathbf{t}) \tag{6}$$

$$[\mathbf{u}] = 0, \quad \nu\left[\frac{\partial\mathbf{u}}{\partial\mathbf{n}}\right] = -(\mathbf{f}\cdot\mathbf{t})\mathbf{t}. \tag{7}$$

The equations of motion are completed by setting the velocity of the Lagrangian markers on Γ to the fluid velocity

$$\frac{d}{dt}\mathbf{X}(\alpha, t) = \mathbf{u}(\mathbf{X}(\alpha, t), t) \tag{8}$$

As in [3] we compare the NSE velocity and pressure with those of the Stokes equations

$$\nabla p_s = \nu\nabla^2\mathbf{u}_s + \mathbf{F} + \mathbf{G}, \quad \nabla\cdot\mathbf{u}_s = 0. \tag{9}$$

In the velocity decomposition method of [3], we write the NSE velocity $\mathbf{u}$ as the sum of the Stokes velocity and a remainder $\mathbf{u}_r$, and similarly for the pressure,

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_r, \quad p = p_s + p_r. \tag{10}$$

The jump conditions for $\mathbf{u}_s, p_s$ are the same as in (6), (7) essentially because the velocity, and therefore its material derivative, are continuous across Γ ([20]). Thus the remainder variables $\mathbf{u}_r, p_r$ are more regular than $\mathbf{u}, p$. Subtracting (9) from (1), we get equations for $\mathbf{u}_r, p_r$ resembling NSE,

$$\frac{\partial\mathbf{u}_r}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}_r + \nabla p_r = \nu\nabla^2\mathbf{u}_r + \mathbf{F}_b \tag{11}$$

$$\nabla\cdot\mathbf{u}_r = 0, \tag{12}$$

where $\mathbf{F}_b$ is minus the material derivative of the Stokes velocity,

$$\mathbf{F}_b = -\frac{\partial\mathbf{u}_s}{\partial t} - \mathbf{u}\cdot\nabla\mathbf{u}_s \tag{13}$$

and is also continuous at Γ. This formulation has the advantage that the two problems can be solved separately, with a choice of methods for each. In [3] we solve the Stokes problem with the IIM [18] and the remainder problem using the semi-Lagrangian method [7,35], which preserves the material derivatives.

The method of [3] was second order accurate in space and time with explicit time steps for the interface. In the present paper we develop a similar method, second order in space and first order in time, with partially implicit time steps for the interface. Second order accuracy in time can be achieved by extrapolation. In Section 2 we outline the complete method and explain the treatment of the fluid variables. In Section 3 we begin with a first order accurate time discretization with a backward Euler step in place of (8). Since the new interface is unknown, we derive an approximation in which a primary contribution is the update of the interfacial force resulting from the change in the interface. In Section 4 we write the force update by using a singular integral representation and approximating the most singular contribution. This contribution is computed in the discrete Fourier transform on the interface. This procedure is motivated by the small scale decomposition method of [9]. A similar procedure was used in [11], for an elastic interface in Navier–Stokes flow with force of the form (4), but the context and derivation here are different. In [11] the flow was computed using the IBM and the interface was represented by tangent angle and arclength, as in [9].

In Section 5 we present numerical results. We verify convergence, test the validity of the partially implicit approximation, and estimate time steps with the force (4). We illustrate the application to a nonlinear force, as in [24,5]. We also present examples with a bending force which can be compared with calculations of inextensible vesicles [13,12,34]. Brief conclusions are given in Section 6.

2. Summary of the method

We outline the method with first-order time discretization but with the equations exact in the spatial variables. The modification to second order in time is discussed later. At time n we assume that the quantities $\mathbf{u}_s^n, \mathbf{u}_r^n, \mathbf{X}^n, \mathbf{u}_{sB}^n$, and $\mathbf{f}^n$ are known. Here $\mathbf{u}_s^n, \mathbf{u}_r^n$ are the Stokes and regular velocity fields at time n, $\mathbf{X}^n$ is the current interface position, as a function of the Lagrangian coordinate, $\mathbf{u}_{sB}^n$ is the restriction of $\mathbf{u}_s^n$ to the interface, and $\mathbf{f}^n$ is the force density on the interface Γ^n determined by $\mathbf{X}^n$. With time step $\Delta t = \tau$, we advance to time $n+1$ in three steps:

(1) Update X to X^{n+1} using an interface velocity $\mathbf{u}_B^*$, discretizing (8) as

$$\frac{\mathbf{X}^{n+1} - \mathbf{X}^n}{\tau} = \mathbf{u}_B^* \tag{14}$$

As derived in the next two sections, $\mathbf{u}_B^*$ is obtained from $\mathbf{u}_{sB}^n$ and $\mathbf{u}_r^n$ by a modification in the high wave numbers to approximate the velocity at time $n+1$. Compute the force $\mathbf{f}^{n+1}$ determined by $\mathbf{X}^{n+1}$.

(2) Solve for the new Stokes velocity $\mathbf{u}_s^{n+1}$,

$$\nabla p_s^{n+1} = \nu \nabla^2 \mathbf{u}_s^{n+1} + \mathbf{F}^{n+1}, \quad \nabla \cdot \mathbf{u}_s^{n+1} = 0 \tag{15}$$

where $\mathbf{F}^{n+1} = \mathbf{f}^{n+1} \delta_{\Gamma^{n+1}}$.

(3) Solve for the new regular velocity $\mathbf{u}_r^{n+1}$ discretizing (11), (12) with the viscosity implicit and the material derivatives formed with velocity values at time n along backward characteristics,

$$\frac{\mathbf{u}_r^{n+1} - \mathbb{T}\mathbf{u}_r^n}{\tau} + \nabla p_r^{n+1} = \nu \nabla^2 \mathbf{u}_r^{n+1} - \frac{\mathbf{u}_s^{n+1} - \mathbb{T}\mathbf{u}_s^n}{\tau}, \quad \nabla \cdot \mathbf{u}_r^{n+1} = 0 \tag{16}$$

Here $\mathbb{T}\mathbf{u}_r^n(\mathbf{x}) = \mathbf{u}_r^n(\tilde{\mathbf{x}})$, where $\tilde{\mathbf{x}}$ is the location reached at time n by traveling backwards in time with velocity $\mathbf{u}$, starting at time $n+1$ at $\mathbf{x}$; this earlier location $\tilde{\mathbf{x}}$ is often called the "departure point".

For the solution of the Stokes problem in (2), it is natural to use the IIM, as applied to Stokes flow in [18], and this was done in [3]. In the present work, as in [15], we use the boundary integral representation of the free space Stokes pressure and velocity (e.g. see [29]). We first compute the values at grid points near the interface, written as nearly singular integrals. We compute the integrals using the method of [2] as applied to Stokes flow in [6]. We regularize the singularity, calculate a value with a standard quadrature, and then add an analytical correction for the regularization. With these values at nearby grid points, we form a discrete Laplacian and invert to obtain the pressure and velocity at all grid points; this procedure was suggested in [22] and used in [2]. The periodic boundary conditions are incorporated in the discrete Poisson problem. This method for solving the Stokes equations in the fluid domain was described in detail in [15].

In solving (3) we use the semi-Lagrangian method [7,35], as was done in [3], but the first order version (16) is simpler than the BDF2 version used before. Starting with a grid point $\mathbf{x}$ for time $n+1$, we approximate the departure point $\tilde{\mathbf{x}}$ by

$$\mathbf{x}^* = \mathbf{x} - (\tau/2)\mathbf{u}^n(\mathbf{x}), \quad \tilde{\mathbf{x}} = \mathbf{x}_0 - \tau \mathbf{u}^n(\mathbf{x}^*) \tag{17}$$

We have to interpolate $\mathbf{u}^n$ in the second equation, and we have to interpolate $\mathbf{u}_s^n$ and $\mathbf{u}_r^n$ to $\tilde{\mathbf{x}}$ in (16). Since $\mathbf{u}_s$ has discontinuous gradient at Γ, we extrapolate grid values across Γ to find $\mathbf{u}_s^n(\tilde{\mathbf{x}})$. We then solve for $\mathbf{u}_r^{n+1}$ using the projection method. We rewrite (16) as

$$\mathbf{u}_r^{n+1} - \tau \nu \nabla^2 \mathbf{u}_r^{n+1} = -\tau \nabla p_r^{n+1} + \mathbb{T}\mathbf{u}_r^n - \mathbf{u}_s^{n+1} + \mathbb{T}\mathbf{u}_s^n \tag{18}$$

and apply the projection P onto divergence free vector fields,

$$P = I - \nabla(\nabla^2)^{-1}\nabla\cdot \tag{19}$$

resulting in the equation

$$\mathbf{u}_r^{n+1} - \tau\nu\nabla^2\mathbf{u}_r^{n+1} = P\big(\mathbb{T}\mathbf{u}_r^n - \mathbf{u}_s^{n+1} + \mathbb{T}\mathbf{u}_s^n\big). \tag{20}$$

Finally, with

$$R = (I - \tau\nu\nabla^2)^{-1} \tag{21}$$

we have

$$\mathbf{u}_r^{n+1} = RP\big(\mathbb{T}\mathbf{u}_r^n - \mathbf{u}_s^{n+1} + \mathbb{T}\mathbf{u}_s^n\big). \tag{22}$$

It is important here that R and P commute with each other and with ∇^2 with periodic boundary conditions. We replace R and P with grid operators based on the usual second order differences for ∇ and ∇^2 and perform the operations with the Fourier transform. The discrete projection is approximate rather than exact. In [3] the pressure was treated slightly differently.

We now describe the modification for second order accuracy in time. For the regular velocity $\mathbf{u}_r$ we use the backward difference formula (BDF2) discretization of (11), (12) as in [3], Section 2.2, rather than (16). Because we do not know a second order version of the derivation in Sections 3 and 4 for partially implicit update of the interface position $\mathbf{X}$, we use extrapolation with respect to time to eliminate $O(\tau)$ errors that were introduced. To describe the second order method, let $S_{\mathbf{X}}(\mathbf{X}^n, \mathbf{u}^n, \tau)$ be the update of $\mathbf{X}$ as in (14), with time step τ, and let $S_{\mathbf{u}}(\mathbf{X}^{n+1}, \mathbf{u}^n, \mathbf{u}^{n-1}, \tau)$ be the update of $\mathbf{u}$ resulting from (15) and the BDF2 solution of (11), (12) replacing (16). Now, supposing $\mathbf{X}$ and $\mathbf{u}$ are known up to time n, we set

$$\begin{aligned}
&\mathbf{X}^* = S_{\mathbf{X}}(\mathbf{X}^{n-1}, \mathbf{u}^{n-1}, \tau); \quad \mathbf{u}^* = S_{\mathbf{u}}(\mathbf{X}^*, \mathbf{u}^{n-1}, \mathbf{u}^{n-2}, \tau);\\
&\mathbf{X}^{n+1} = 2S_{\mathbf{X}}(\mathbf{X}^*, \mathbf{u}^*, \tau) - S_{\mathbf{X}}(\mathbf{X}^{n-1}, \mathbf{u}^{n-1}, 2\tau);\\
&\mathbf{u}^{n+1} = S_{\mathbf{u}}(\mathbf{X}^{n+1}, \mathbf{u}^n, \mathbf{u}^{n-1}, \tau)
\end{aligned} \tag{23}$$

Thus two velocity evaluations are needed for each time step.

The overall operation count for either version is $O(N^2 \log N)$ per time step, where N is the number of grid points in each direction and we assume that $O(N)$ points are used on the interface. The largest cost is for two-dimensional FFT's used in the inversion for the Stokes problem and for RP in (22). The direct computation of integrals for Stokes flow uses $O(N^2)$; in principle this could be replaced by a fast method. The partially implicit update of the interface used here can be done with a 1D FFT, having negligible cost.

3. Approximating the backward Euler step

We begin with the problem discrete in time, exact in space, and implicit in $\mathbf{X}^{n+1}$. We will approximate the new velocity $\mathbf{u}^{n+1}$ on Γ^{n+1} to obtain $\mathbf{u}_{\mathrm{B}}^*$ mentioned earlier. The implicit equations are

$$\mathbf{X}^{n+1} = \mathbf{X}^n + \tau\mathbf{u}^{n+1}(\mathbf{X}^{n+1}), \tag{24}$$

$$\mathbf{u}^{n+1} - \mathbb{T}\mathbf{u}^n + \tau\nabla p^{n+1} = \tau\nu\nabla^2\mathbf{u}^{n+1} + \tau\mathbf{F}^{n+1}, \quad \nabla\cdot\mathbf{u}^{n+1} = 0 \tag{25}$$

where $\mathbf{F}^{n+1}$ is the force determined by $\mathbf{X}^{n+1}$. We suppose $\mathbf{G} = 0$ for simplicity. Our task is to replace the right side of (24) with a quantity which can be computed at time n. It will be important that for each material location α on the interface, and any function $w(\mathbf{x})$,

$$(\mathbb{T}w)(\mathbf{X}^{n+1}(\alpha)) = w(\mathbf{X}^n(\alpha)) \tag{26}$$

As before we can write $\mathbf{u}^n = \mathbf{u}_s^n + \mathbf{u}_r^n$, each with divergence zero. As in (18)–(22) we apply the projection P and then the Poisson solver R to (25) to obtain

$$\mathbf{u}^{n+1} = \tau RP\mathbf{F}^{n+1} + RP\mathbb{T}\mathbf{u}^n \tag{27}$$

Also for the Stokes part we will use the projection of (9) at time n,

$$0 = \tau\nu\nabla^2\mathbf{u}_s^n + \tau P\mathbf{F}^n \tag{28}$$

Note that $R(\tau\nu\nabla^2) = R - I$ and therefore

$$R(\tau\nu\nabla^2\mathbf{u}_s^n) = R\mathbf{u}_s^n - \mathbf{u}_s^n \tag{29}$$

and thus from (28)

$$\tau RP\mathbf{F}^n = -R(\tau\nu\nabla^2\mathbf{u}_s^n) = \mathbf{u}_s^n - R\mathbf{u}_s^n \tag{30}$$

Now we subtract and add $\tau\mathbb{T}(RP\mathbf{F}^n)$ to Eq. (27) for $\mathbf{u}^{n+1}$, using $\mathbb{T}$ applied to Eq. (30):

$$\mathbf{u}^{n+1} = \tau RP\mathbf{F}^{n+1} - \tau\mathbb{T}(RP\mathbf{F}^n) + \mathbb{T}\mathbf{u}_s^n - \mathbb{T}(R\mathbf{u}_s^n) + RP\mathbb{T}\mathbf{u}_s^n + RP\mathbb{T}\mathbf{u}_r^n \tag{31}$$

We have substituted $\mathbf{u}^n = \mathbf{u}_r^n + \mathbf{u}_s^n$ in the last term in (27) so that in (31) the last two terms with $\mathbf{u}_s^n$ appear to be a commutator. Now add and subtract $\mathbb{T}(R\mathbf{u}_r^n)$,

$$\mathbf{u}^{n+1} = \tau RP\mathbf{F}^{n+1} - \tau\mathbb{T}(RP\mathbf{F}^n) + \mathbb{T}\mathbf{u}_s^n + \mathbb{T}(R\mathbf{u}_r^n) - \mathbb{T}(R\mathbf{u}_s^n) + RP\mathbb{T}\mathbf{u}_s^n + RP\mathbb{T}\mathbf{u}_r^n - \mathbb{T}(R\mathbf{u}_r^n) \tag{32}$$

and combine the terms with $\mathbf{u}_s^n$ and $\mathbf{u}_r^n$ at the end:

$$\mathbf{u}^{n+1} = \tau RP\mathbf{F}^{n+1} - \tau\mathbb{T}(RP\mathbf{F}^n) + \mathbb{T}\mathbf{u}_s^n + \mathbb{T}(R\mathbf{u}_r^n) + RP\mathbb{T}\mathbf{u}^n - \mathbb{T}(R\mathbf{u}^n) \tag{33}$$

We need to use this at $\mathbf{X}^{n+1}$, the location of the new interface. Using (26) we simplify most of the terms in (33) to obtain

$$\mathbf{u}^{n+1}(\mathbf{X}^{n+1}) = \tau RP\mathbf{F}^{n+1}(\mathbf{X}^{n+1}) - \tau RP\mathbf{F}^n(\mathbf{X}^n) + \mathbf{u}_s^n(\mathbf{X}^n) + R\mathbf{u}_r^n(\mathbf{X}^n) + [RP\mathbb{T}\mathbf{u}^n - \mathbb{T}(R\mathbf{u}^n)](\mathbf{X}^{n+1}) \tag{34}$$

So far we have an exact equation. We will make two approximations for use in (24). Since $R\mathbf{u}^n = RP\mathbf{u}^n$, the last term in (34) is a commutator of $\mathbb{T}$ and RP applied to $\mathbf{u}^n$. For a velocity with bounded gradient, it is no larger than $O(\tau)$ since $(\mathbb{T} - I)\mathbf{u}^n = O(\tau)$. As a contribution to the truncation error in (24), we can neglect this $O(\tau)$ term in the velocity in (24) while deriving a first-order method. (Of course we should expect the error to grow as $1/\nu$ and the force coefficients become large.) The velocity expression is now reduced to

$$\mathbf{u}^{n+1}(\mathbf{X}^{n+1}) \approx \tau(RP\mathbf{F}^{n+1})(\mathbf{X}^{n+1}) - \tau(RP\mathbf{F}^n)(\mathbf{X}^n) + \tilde{\mathbf{u}}^n(\mathbf{X}^n) \tag{35}$$

where we have set

$$\tilde{\mathbf{u}}^n = \mathbf{u}_s^n + R\mathbf{u}_r^n \tag{36}$$

The terms $RP\mathbf{F}^{n+1}$ and $RP\mathbf{F}^n$ can be written as integrals on the interface whose kernel is the fundamental solution K of the "modified Stokes operator" RP; see (40)–(43). Each term is actually $O(\tau^{-1/2})$ on the interface. (This can be seen from Eq. (53) for the transform of K, with $k = O(1)$, or more directly from the fact that the K must have the form $\tau^{-1}G(\mathbf{x}/\sqrt{\tau})$.) Now since $\mathbf{F}^{n+1} - \mathbf{F}^n = O(\tau)$, the term appearing in the velocity is $O(\tau \cdot \tau \cdot \tau^{-1/2}) = O(\tau^{3/2})$. Thus it is negligible in the truncation error in (24). This suggests that we can replace it by our guess of its most important part without losing first order accuracy. We do this in order to gain stability. In the next section we will derive an approximation

$$\tau RP\mathbf{F}^{n+1}(\mathbf{X}^{n+1}) - \tau RP\mathbf{F}^n(\mathbf{X}^n) \approx \mathcal{A}(\mathbf{X}^{n+1} - \mathbf{X}^n) \tag{37}$$

where $\mathcal{A}$ is a linear operator. Then, combining (24) with (35)–(37) we get

$$\mathbf{X}^{n+1} - \mathbf{X}^n = \tau\mathcal{A}(\mathbf{X}^{n+1} - \mathbf{X}^n) + \tau\tilde{\mathbf{u}}^n(\mathbf{X}^n) \tag{38}$$

and solving for $\mathbf{X}^{n+1}$,

$$\mathbf{X}^{n+1} - \mathbf{X}^n = \tau(I - \tau\mathcal{A})^{-1}\tilde{\mathbf{u}}^n(\mathbf{X}^n) \equiv \tau\mathbf{u}_B^* \tag{39}$$

with the modified velocity written earlier in (14).

We use (39) in this work, but we comment briefly on the possibility of computing the neglected terms. For the commutator at the end of (34), we could calculate $RP\mathbb{T}\mathbf{u}^n$ at grid points by following a point backward in time, as in (17), to obtain $\mathbb{T}\mathbf{u}^n$, and then apply RP in the Fourier transform as described after (22). Similarly we can find $\mathbb{T}(R\mathbf{u}^n)$. Having values at the grid points, we need to interpolate the difference to $\mathbf{X}^{n+1}$; we could extrapolate the location from earlier time steps, although it is not clear how this would affect the stability. Rather than using the approximation (37), we could replace the exact quantity on the left by the integral expression in (43) and write $\mathbf{f}^{n+1}$ in terms of $\mathbf{X}^{n+1}$, leading to a nonlinear integral equation for $\mathbf{X}^{n+1}$, and resulting in a fully implicit method. It could be simplified by lagging the update of the interface location in the first integral, as has been done with implicit methods for the immersed boundary method, giving a more refined approximation than the one used here. For recent work on implicit methods for the IBM applied to Navier–Stokes flow, see [4,5,11,24,25].

4. Approximating the stiff part of the force

To derive an expression of the form (37) we think of

$$\dot{\mathbf{v}} \equiv \tau RP\mathbf{F}^{n+1}(\mathbf{X}^{n+1}) - \tau RP\mathbf{F}^n(\mathbf{X}^n) \tag{40}$$

as a change or variation in the velocity due to the variation in interface position, $\dot{\mathbf{X}} = \mathbf{X}^{n+1} - \mathbf{X}^n$. We approximate the changes with variational derivatives and derive simple formulas in the Fourier transform on the interface by identifying the most singular part, as in [9,11] and other work. We will assume for now that the force exerted by the interface Γ has the form

$$\mathbf{F} = \mathbf{f}\delta_\Gamma, \quad \mathbf{f} = \partial_s\Phi, \quad \Phi = \epsilon'(s_\alpha)\mathbf{t} \tag{41}$$

The operator RP is given by a convolution with the fundamental solution. Although the problem is in a periodic box, we can replace the operators with those for free space, since the difference at the interface is smooth, and we are interested in

approximating high wave numbers. The free space fundamental solution consists of functions $K_{ij}, i,j = 1,2$, where $K_{ij}(\mathbf{x})$ is the ith component of $RP(\delta \mathbf{e}_j)$, δ is the usual delta function in $\mathbb{R}^2$, and P and R are defined in (19),(21).

$$RP = (I - \tau v \Delta)^{-1}(I - \nabla(\nabla^2)^{-1}\nabla\cdot) \tag{42}$$

The convolution of this fundamental solution with $\mathbf{F}$ results in an integral over Γ, so that $\dot{v}_i$ as a function of α, $i = 1,2$, is

$$\dot{v}_i = \tau \int_{\Gamma^{n+1}} \sum_j K_{ij}(\mathbf{X}^{n+1}(\alpha) - \mathbf{y}) \mathbf{f}_j^{n+1}(\mathbf{y}) ds(\mathbf{y}) - \tau \int_{\Gamma^n} \sum_j K_{ij}(\mathbf{X}^n(\alpha) - \mathbf{y}) \mathbf{f}_j^n(\mathbf{y}) ds(\mathbf{y}) \tag{43}$$

We can rewrite each integral in Lagrangian coordinates, with $\mathbf{y} = \mathbf{X}(\alpha')$ at time $n+1$ or n. For simplicity, we will replace $\mathbf{X}^{n+1}$ inside K in the first term by $\mathbf{X}^n$, for both α and α', and similarly for s'_α in the change of variables. (That is, in the first term we use the old interface but the new force; see [25] for comments about this replacement.) Thus, with $\dot{\Phi} = \Phi^{n+1} - \Phi^n$ we get

$$\dot{\mathbf{v}}(\alpha) \approx \tau \int K(\mathbf{X}^n(\alpha) - \mathbf{X}^n(\alpha')) \partial_s \dot{\Phi}(\alpha') s^n_{\alpha'} d\alpha' \tag{44}$$

or, after canceling two factors of $s_{\alpha'}$,

$$\dot{\mathbf{v}}(\alpha) \approx \tau \int K(\mathbf{X}(\alpha) - \mathbf{X}(\alpha')) \partial_{\alpha'} \dot{\Phi}(\alpha') d\alpha' \tag{45}$$

We have omitted superscript n's and indices i,j.

The change $\dot{\Phi}$ in Φ is determined by $\dot{\mathbf{X}} = \mathbf{X}^{n+1} - \mathbf{X}^n$. We will need to write $\dot{\mathbf{X}}$ in tangential and normal components, denoted $\dot{X}_1, \dot{X}_2$

$$\dot{\mathbf{X}} = \dot{X}_1 \mathbf{t} + \dot{X}_2 \mathbf{n} \tag{46}$$

as functions of α, where $\mathbf{t}, \mathbf{n}$ are the unit tangent and normal vectors at $\mathbf{X}^n(\alpha)$. Varying $\Phi(\alpha) = \epsilon'(s_\alpha)\mathbf{t}$, we have

$$\dot{\Phi} \approx \epsilon''(s_\alpha)\dot{s}_\alpha \mathbf{t} + \epsilon'(s_\alpha)\dot{\mathbf{t}}.$$

From $s_\alpha^2 = \mathbf{X}_\alpha \cdot \mathbf{X}_\alpha$, we get $\dot{s}_\alpha s_\alpha \approx \dot{\mathbf{X}}_\alpha \cdot \mathbf{X}_\alpha$, so that $\dot{s}_\alpha \approx \dot{\mathbf{X}}_\alpha \cdot \mathbf{t} = \partial_\alpha(\dot{\mathbf{X}} \cdot \mathbf{t}) - \dot{\mathbf{X}} \cdot \partial_\alpha \mathbf{t}$ or

$$\dot{s}_\alpha \approx \partial_\alpha \dot{X}_1 - s_\alpha \kappa \dot{X}_2 \tag{47}$$

where κ is the curvature, defined by $\mathbf{t}_s = \kappa \mathbf{n}$. The variation $\dot{\mathbf{t}}$ must be normal, so that $\dot{\mathbf{t}} \approx (s_\alpha^{-1}\dot{\mathbf{X}}_\alpha \cdot \mathbf{n})\mathbf{n}$ and

$$\dot{\mathbf{t}} \approx \left(s_\alpha^{-1}\partial_\alpha \dot{X}_2 + \kappa \dot{X}_1\right)\mathbf{n} \tag{48}$$

(Further details are given e.g. in [1], p. 1284.) We neglect the second terms in both (47), (48) as less important since they do not have α-derivatives on $\dot{\mathbf{X}}$. Thus we obtain

$$\dot{\Phi}(\alpha) \approx c_1(\alpha)(\partial_\alpha \dot{X}_1)\mathbf{t} + c_2(\alpha)(\partial_\alpha \dot{X}_2)\mathbf{n}, \quad c_1(\alpha) = \epsilon''(s_\alpha), \quad c_2(\alpha) = s_\alpha^{-1}\epsilon'(s_\alpha) \tag{49}$$

To calculate (45) with α fixed, we can temporarily assume, because of the rotational invariance of the problem, that the x_1-axis is tangent to Γ at $\mathbf{X}(\alpha)$ in the $x = (x_1, x_2)$ plane. For our approximation we need only be concerned with the singular part of the integrand. Since the singular behavior is for α' near α, we will replace $\mathbf{X}(\alpha) - \mathbf{X}(\alpha')$ in (45) by $(s_\alpha(\alpha - \alpha'), 0)$, where $s_\alpha = s_\alpha(\alpha) = |\partial_\alpha \mathbf{X}(\alpha)|$; the difference is much smoother than either term. We also replace K with a localized version $K^{(0)}$, multiplying K with $\psi(\alpha - \alpha')$, where $\psi(\beta) = 1$ for β small and $\psi = 0$ outside a small interval about $\beta = 0$; the difference is again smooth. Thus we have

$$\dot{\mathbf{v}} \approx \tau \int K^{(0)}(s_\alpha(\alpha - \alpha')) \partial_{\alpha'} \dot{\Phi}(\mathbf{X}(\alpha')) d\alpha' \tag{50}$$

We use the Fourier transform of K, first in two dimensions and then one. We write the transform and its inverse for $\varphi : \mathbb{R}^2 \to \mathbb{R}$, as

$$\tilde{\varphi}(\xi) = (2\pi)^{-2} \int_{\mathbb{R}^2} \varphi(x) e^{-ix\xi} dx, \quad \varphi(x) = \int_{\mathbb{R}^2} \tilde{\varphi}(\xi) e^{ix\xi} d\xi \tag{51}$$

Thus in $\mathbb{R}^2$ we have $\tilde{\delta} = (2\pi)^{-2}$ and $[(I - \tau v \Delta)^{-1}\delta]\tilde{}\,(\xi) = (2\pi)^{-2}(1 + \tau v \xi^2)^{-1}$. We set $\xi = (k, \ell)$ and $\xi^2 = k^2 + \ell^2$, so that in the transform P_{11} multiplies by $1 - k^2/\xi^2 = \ell^2/\xi^2$ etc., and thus

$$\tilde{K}_{11} = \frac{\ell^2}{4\pi^2 \xi^2 (1 + \tau v \xi^2)}, \quad \tilde{K}_{22} = \frac{k^2}{4\pi^2 \xi^2 (1 + \tau v \xi^2)}, \quad \tilde{K}_{12} = -\frac{k\ell}{4\pi^2 \xi^2 (1 + \tau v \xi^2)} \tag{52}$$

and $\tilde{K}_{21} = \tilde{K}_{12}$. Next we form the one-dimensional transform $\hat{K}_{ij}$ of K_{ij} as a function of x_1 on $x_2 = 0$, with a similar definition; it results from integrating $\tilde{K}_{ij}$ in ℓ. With $\lambda^2 = (\tau v)^{-1}$, we get $\hat{K}_{12} = \hat{K}_{21} = 0$ and

$$\hat{K}_{11}(k) = \frac{1}{4\pi}\frac{\lambda^2}{\sqrt{k^2+\lambda^2}+|k|}, \quad \hat{K}_{22}(k) = \frac{1}{4\pi}\frac{\lambda^2|k|}{\sqrt{k^2+\lambda^2}(\sqrt{k^2+\lambda^2}+|k|)} \tag{53}$$

These could also be derived from the formulas for K_{ij} in terms of Bessel functions, as was done in [11].

We are now ready to approximate the integral (50). Suppose $-A \leqslant \alpha \leqslant A$; since $\dot{\Phi}$ is periodic in α we can write

$$\dot{\Phi}(\alpha') = \sum_k \dot{\Phi}_k e^{i\pi k\alpha'/A}, \quad \dot{\Phi}_k = \frac{1}{2A}\int_{-A}^{A} \dot{\Phi}(\alpha') e^{-i\pi k\alpha'/A} d\alpha' \tag{54}$$

and similarly for $\mathbf{\dot{X}}$. Assuming $\dot{\Phi}$ is smooth, so that the series converges rapidly enough, we substitute into (50) obtaining

$$\dot{\mathbf{v}}(\alpha) = \tau\sum_k \frac{i\pi k}{A}\dot{\Phi}_k \int_{-A}^{A} K^{(0)}(s_\alpha(\alpha-\alpha'))e^{i\pi k\alpha'/A} d\alpha' \tag{55}$$

We set $x_1 = s_\alpha(\alpha-\alpha')$ and change variables in the integral. We extend the integral to all x_1, since $K^{(0)}$ is local, and finally we replace $K^{(0)}$ by K; since the difference is smooth it contributes terms rapidly decreasing in k. We now rewrite the integral in terms of $\hat{K}$:

$$\dot{\mathbf{v}}(\alpha) = \tau\sum_k \frac{i\pi k}{As_\alpha}\dot{\Phi}_k e^{i\pi k\alpha/A}\int_{-\infty}^{\infty} K(x_1,0)e^{-i(\pi k/As_\alpha)x_1} dx_1 \tag{56}$$

$$\dot{v}_\ell(\alpha) = 2\pi\tau\sum_k \frac{i\pi k}{As_\alpha}\dot{\Phi}_{k,\ell}e^{i\pi k\alpha/A}\hat{K}_{\ell\ell}\left(\frac{\pi k}{As_\alpha}\right), \quad \ell = 1,2 \tag{57}$$

We set $\eta = \pi k/(As_\alpha\lambda)$ so that

$$\hat{K}_{11}\left(\frac{\pi k}{As_\alpha}\right) = \frac{\lambda}{4\pi}\frac{1}{\sqrt{\eta^2+1}+|\eta|}, \quad \hat{K}_{22}\left(\frac{\pi k}{As_\alpha}\right) = \frac{\lambda}{4\pi}\frac{|\eta|}{\sqrt{\eta^2+1}(\sqrt{\eta^2+1}+|\eta|)} \tag{58}$$

and thus, since $\lambda^2\tau = 1/\nu$,

$$\dot{v}_\ell(\alpha) = \frac{i}{2\nu}\sum_k e^{i\pi k\alpha/A}\mathrm{sgn}(\eta)g_\ell(\eta)\dot{\Phi}_{k,\ell}, \quad \ell = 1,2 \tag{59}$$

where we define

$$g_1(\eta) = \frac{|\eta|}{\sqrt{\eta^2+1}+|\eta|}, \quad g_2(\eta) = \frac{\eta^2}{\sqrt{\eta^2+1}(\sqrt{\eta^2+1}+|\eta|)}. \tag{60}$$

From (49) we have $\dot{\Phi}_\ell(\alpha') = c_\ell(s_\alpha)\partial_\alpha\dot{X}_\ell(\alpha')$. Here s_α is evaluated at α', but we replace it with the value at α in keeping with the local approximation. The corresponding relation in Fourier coefficients is $\dot{\Phi}_{k,\ell} = c_\ell(s_\alpha)(i\pi k/A)\dot{X}_{k,\ell}$. Combining with the above, we get

$$\dot{v}_\ell(\alpha) = -\frac{1}{2\nu}c_\ell(s_\alpha)\sum_k e^{i\pi k\alpha/A}\frac{\pi|k|}{A}g_\ell(\eta)\dot{X}_{k,\ell}, \quad \ell = 1,2 \tag{61}$$

So far we have considered α arbitrary but fixed with horizontal tangent at $\mathbf{X}^n(\alpha)$. We now consider α arbitrary and write $\mathbf{\dot{X}} = \dot{X}_1\mathbf{t} + \dot{X}_2\mathbf{n}$ and similarly for $\dot{\mathbf{v}}$. The expressions (61) give $\dot{\mathbf{v}}$ in the form $\dot{\mathbf{v}} = \mathcal{A}\mathbf{\dot{X}}$, as stated in (37), and we are ready to solve for $\mathbf{\dot{X}}$ as in (38),(39). To calculate the solution, we use the discrete Fourier transform, replacing the exact Fourier series above. With $-A \leqslant \alpha \leqslant A$, we discretize with $\alpha = jh_B$, $-N_B/2+1 \leqslant j \leqslant N_B/2$. The discrete transform of $\dot{X}_\ell(jh_B)$, for $\ell = 1,2$, is $\dot{X}_{k,\ell}$, with $-N_B/2+1 \leqslant k \leqslant N_B/2$,

$$\dot{X}_\ell(jh_B) = \sum_k \dot{X}_{k,\ell}e^{i\pi kjh_B/A}, \quad \dot{X}_{k,\ell} = \frac{1}{N_B}\sum_j \dot{X}_\ell(jh_B)e^{-i\pi kjh_B/A} \tag{62}$$

and similarly for $\dot{\mathbf{v}}, \tilde{\mathbf{u}}$. To discuss the approximate solution of (38) further we will consider specific cases separately.

Linear tension force. For the familiar case $\Phi = \gamma_0(s_\alpha - 1)$, we have $\epsilon(\sigma) = \gamma_0(\sigma-1)^2/2$, and we see from (49) that $c_1 = \gamma_0$ and $c_2 = \gamma_0(1 - s_\alpha^{-1})$ in (61). The coefficient of $\dot{X}_{k,\ell}$ in (61) depends on α as well as k. To simplify the equation to be solved, we can replace $s_\alpha(\alpha)$ in the definition of η by $\min s_\alpha$, so that now

$$\eta = c_0 k, \quad c_0 = \frac{\pi}{A(\min s_\alpha)\lambda} \tag{63}$$

The functions g_1, g_2 are increasing in η, so that replacing s_α by its minimum has the effect of magnifying the coefficient of $\dot{X}_{k,\ell}$. Similarly in c_2 we replace s_α by the maximum. In the resulting approximate version of Eq. (38) we can solve directly for $\dot{X}_{k,\ell}$ in terms of the Fourier coefficients of $\tilde{\mathbf{u}}$. For $-N_B/2+1 \leqslant k \leqslant N_B/2$ we set

$$m_1(k) = \left[1 + \frac{\tau\gamma_0}{2\nu}\left(\frac{\pi}{A}|k|\right)g_1(c_0k)\right]^{-1} \tag{64}$$

$$m_2(k) = \left[1 + \frac{\tau\gamma_0}{2\nu}\left(1 - \frac{1}{(\max s_\alpha)}\right)\left(\frac{\pi}{A}|k|\right)g_2(c_0k)\right]^{-1} \tag{65}$$

Then combining (61) as modified with (38) and inverse transforming, we get

$$\dot{X}_\ell(jh) = \tau \sum_{k=-N_B/2+1}^{N_B/2} m_\ell(k)\tilde{u}_{k,\ell}e^{i\pi kjh_B/A} \tag{66}$$

and finally $\mathbf{X}^{n+1} - \mathbf{X}^n = \dot{X}_1\mathbf{t} + \dot{X}_2\mathbf{n}$. The replacement of s_α by constants gives us the simplest version of the approximate solution of (38). Such a procedure was used in [11], leading to formulas similar to (64), (65), and this version is used in our examples in Section 5.

General tension force. For a general force as in (41) we can proceed as above with

$$m_\ell(k) = \left[1 + \frac{\tau}{2\nu}c_\ell^{(0)}\left(\frac{\pi}{A}|k|\right)g_\ell(c_0k)\right]^{-1} \tag{67}$$

where $c_\ell^{(0)}$ is an upper bound for $c_\ell(\alpha)$. For example, if $\epsilon(\sigma) = \gamma_0\sigma^2/2 + \gamma_1\sigma^3/3$, so that $\epsilon'(\sigma) = \gamma_0\sigma + \gamma_1\sigma^2$ then

$$c_1^{(0)} = \gamma_0 + 2\gamma_1(\max s_\alpha), \quad c_2^{(0)} = \gamma_0 + \gamma_1(\max s_\alpha). \tag{68}$$

Bending force. With a bending force we have, in place of (41), $\mathbf{f} = -c_b\partial_s^4\mathbf{X} \equiv \partial_s\Phi$, with $\Phi = -c_b\partial_s^3\mathbf{X}$ or $\Phi = -c_b\partial_s^2\mathbf{t} = -c_bs_\alpha^{-1}(\partial_\alpha(s_\alpha^{-1}\partial_\alpha\mathbf{t}))$. In the variation of Φ, the most important term (having the highest derivative) is $-c_bs_\alpha^{-2}\partial_\alpha^2\dot{\mathbf{t}}$. With $\dot{\mathbf{t}} \approx (s_\alpha^{-1}\partial_\alpha\dot{X}_2)\mathbf{n}$ as before, we get $\dot{\Phi} \approx -c_bs_\alpha^{-3}\partial_\alpha^3(\dot{X}_2)\mathbf{n}$ in place of (49). From this we find $\dot{\Phi}_{k,1} = 0, \dot{\Phi}_{k,2} = -c_bs_\alpha^{-3}(i\pi k)/A)^3$. In place of (61) we get $\dot{v}_1 = 0$ and

$$\dot{v}_2 = -\frac{c_b}{2\nu}s_\alpha^{-3}\sum_k e^{i\pi k\alpha/A}\left(\frac{\pi|k|}{A}\right)^3 g_2(\eta)\dot{X}_{k,2}. \tag{69}$$

If we have only the bending force, then $\dot{\mathbf{X}}$ is given by (66) with $m_1 = 1$ and

$$m_2(k) = \left[1 + \frac{\tau c_b}{2\nu}(\min s_\alpha)^{-3}\left(\frac{\pi}{A}|k|\right)^3 g_2(c_0k)\right]^{-1} \tag{70}$$

If the force is a sum of bending and elastic terms, e.g. $\mathbf{f} = \gamma_0\partial_s((s_\alpha - 1)\mathbf{t}) - c_b\partial_s^4\mathbf{X}$, then m_1 is given by (64), and m_2 has terms as in (65) and (70) added inside the brackets.

5. Numerical results

We suppose the Eqs. (1)–(8) have been nondimensionalized with length scale L, time scale T and velocity $U = L/T$, so that if $\mathbf{x}', t', \mathbf{u}'$ are dimensional variables, then $\mathbf{x} = \mathbf{x}'/L, t = t'/T, \mathbf{u} = \mathbf{u}'/U$ etc. The nondimensionalized viscosity coefficient is $\nu = \nu'/LU = 1/\mathrm{Re}$, with Re the usual Reynolds number. If the tension force has the form (4), the nondimensional tension coefficient γ_0 is $\gamma_0'T^2/(\rho'L^3)$, ρ' being the density, and for a bending force as in (5) the bending coefficient is $c_b = c_b'T^2/(\rho'L^5)$. Thus, for example, with the tension force (4), two problems reduce to the same nondimensional equations if $L^{(1)} = L^{(2)}$, $T^{(1)} = T^{(2)}/\lambda, U^{(1)} = \lambda U^{(2)}$, $\nu'^{(1)} = \lambda\nu'^{(2)}$, and $\gamma_0'^{(1)} = \lambda^2\gamma_0'^{(2)}$ for some constant λ. We can rescale a given problem so that γ_0 is 1, changing the other parameters. In our experiments with the force (4) we will consider the problem nondimensionalized with L fixed and $\gamma_0 = 1$, and vary the viscosity ν.

We use a square computational region $0 < x_1, x_2 < 1$ and periodic boundary conditions. We discretize $\mathbf{x} = (x_1, x_2)$ on a square grid with spacing $h = 1/N$ and use N_B markers on the interface, equally spaced in the Lagrangian coordinate α. We always choose $N_B = 2N$.

For our accuracy tests we use the familiar test problem in which the initial interface is an ellipse stretched from its natural circular configuration, as in [5,10,11,17,18,24,25,33]. Usually the initial velocity is set to zero and the interface markers are equally spaced in the parameter θ as in (71) below. If this is done, there is a nonzero initial tangential force due to the variation in $\partial s/\partial\alpha$, and the initial velocity zero violates the jump condition (7). Thus the initial state is inconsistent with the governing equations. The problem is meaningful, but an extra singularity is introduced. For this reason, we prefer to use the Stokes velocity as the initial state in our accuracy tests. (Another way to avoid this extra singularity would be to use initial velocity zero but space the markers equally in arclength, so that $\partial s/\partial\alpha$ is constant on the interface, leading to zero tangential force. This was apparently done in [11].)

For our first tests we choose the initial curve as the ellipse parametrized by θ,

$$x_1 = .5 + (1/3)\cos\theta, \quad x_2 = .5 + (1/4)\sin\theta \quad -\pi \leqslant \theta \leqslant \pi \tag{71}$$

We choose the unstretched configuration to be the circle of radius $1/5$ and the material coordinate to be $\alpha = \theta/5, |\alpha| \leqslant A = \pi/5$. This choice is comparable to that in [24,5]. We place the initial interface markers equally spaced in

θ and assume the initial pressure and velocity are those determined by the Stokes equations. We assume a tension force of the form (4) with $\gamma_0 = 1$.

If $\nu > .1$, the ellipse relaxes toward the circle of the same area. For $\nu < .1$, it undergoes a damped oscillation before approaching the circle. We will say that a "half-cycle" is the first time that the x_1-intercept has a local minimum and the x_2-intercept a local maximum. For $\nu = .1, .05, .01$, this time is $.9, .45, .3$. Fig. 1 shows the x_1 – and x_2-intercepts as functions of time with $\nu = .01$. For comparison, Fig. 2 gives the intercepts with the initial velocity zero rather than Stokes. Fig. 2 is similar to Fig. 8.5 in [17].

Stability. We compare three versions of the method: the code with first order partially implicit time step, as described in Section 2; the second order partially implicit version as in (23); and a code with second order explicit time steps of Adams–Bashforth type. All three are designed to be second order in space. For brevity we refer to the three codes as "first order", "second order" and "explicit". We first estimated experimentally the largest stable time step for the test problem described above with various choices of ν. We performed 50 time steps and judged stability by the absence of unphysical oscillations in the tangent to the interface or in the velocity at the interface. (With too large a time step our method of interpolation near the interface often fails before these oscillations appear.)

The results are shown in Table 1. Values of $\Delta t/h$ are shown. The maximum velocity that occurs in the solution is the maximum initial Stokes velocity, about $.04/\nu$. Thus the CFL condition predicts $\Delta t/h = 25\nu$. The maximum $\Delta t/h$ is considerably larger than the CFL prediction, for the first or second order implicit code, for $\nu \geqslant .01$. For $\nu = .005$ it is worse for lower N, but almost matches the CFL step in the highest resolution. It appears that for smaller ν, higher spatial resolution is needed to run with the CFL time step. To test this hypothesis, we tried the second order implicit code with $\nu = .001,\ N = 600,\ N_B = 1200$; it ran successfully for 50 time steps with $\Delta t/h = .025$, the CFL velocity.

Validity of the partially implicit formula. To test the validity of the approximations made in Sections 3, 4 for the interface motion, we compared solutions calculated by the first and second order partially implicit codes with a solution from the explicit code with small steps. For each test we ran the implicit code twice, reducing Δt in the second run by a factor of 2 and doubling the number of time steps. The spatial resolution was fixed at $N = 400,\ N_B = 800$. We chose $\nu = .1,\ \gamma_0 = 1$ and final time .4. For the explicit code $\Delta t/h = .1$, requiring 1600 time steps. We computed relative errors in velocity and interface location, treating the explicit solution as exact, even though it has $O(\Delta t^2)$ error. The relative error in velocity is $\|\mathbf{u}_{\Delta t} - \mathbf{u}_{\text{expl}}\|/\|\mathbf{u}_{\text{expl}}\|$, or the same with Δt replaced by $\Delta t/2$, where the norm is the discrete L^2 norm on square grid points. The definition for the interface position is similar, with L^2 norms with respect to the material parameter on the curve. We found empirical rates of convergence, also based on L^2 norms. Thus for the velocity field the rate is

$$p = \log_2 \|\mathbf{u}_{\Delta t} - \mathbf{u}_{\text{expl}}\|/\|\mathbf{u}_{\Delta t/2} - \mathbf{u}_{\text{expl}}\| \tag{72}$$

and similarly for the interface. Results are shown in Table 2. For each pair of runs we display the larger of the two values of $\Delta t/h$, the relative errors with the larger time step, and the empirical rates of convergence. The errors and rates based on L^∞ or maximum norm errors are similar. E.g., for the second order code with $\Delta t/h = .5$ the relative errors in maximum norm for u_1, u_2, and $\mathbf{X}$ are 1.5e−3, 1.7e−3, and 4.5e−5. The results for the first order code clearly show the expected $O(\Delta t)$ error. For the second order code, the order in Δt increases with refinement well past first order. It does not approach 2, probably

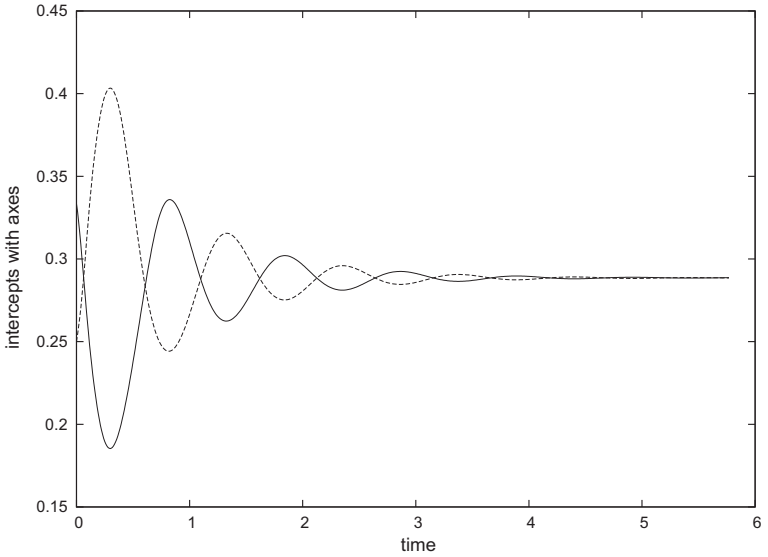

Fig. 1. x_1-intercept (solid) and x_2-intercept (dashed) for the ellipse initialized with Stokes velocity.

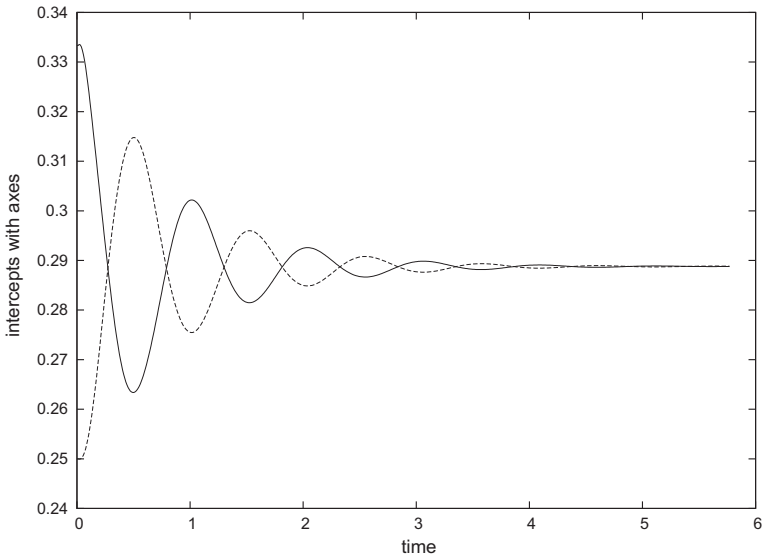

Fig. 2. x_1-intercept (solid) and x_2-intercept (dashed) for the ellipse initialized with zero velocity.

Table 1
Largest observed stable values of $\Delta t/h$, tension force.

v	Explicit			First order			Second order		
	N =100	200	400	N =100	200	400	N =100	200	400
1	3	3	3	300	300	400	180	180	250
.1	.3	.3	.3	30	40	50	20	25	35
.01	.02	.04	.04	2.5	5	5	.5	1	2.5
.005	.003	.01	.015	.008	.07	.15	.005	.02	.12

Table 2
Empirical rates of convergence to the explicit code as $\Delta t \to 0$.

$\Delta t/h$	First order				Second order			
	Velocity		Interface		Velocity		Interface	
	rel err	Rate	rel err	Rate	rel err	Rate	rel err	Rate
2	1.91e−2	1.16	1.20e−2	1.15	5.58e−3	1.18	2.31e−3	1.13
1	8.56e−3	1.11	5.43e−3	1.12	2.46e−3	1.35	1.06e−3	1.34
.5	3.96e−3	1.07	2.50e−3	1.09	9.63e−4	1.43	4.17e−5	1.43
.25	1.89e−3	1.04	1.17e−3	1.07	3.58e−4	0.57	1.55e−5	−1.12

because the two codes being compared each have $O(\Delta t^2)$ errors. The rate deteriorates at the finest level, as should be expected. The rate of convergence of the second order code is tested directly in Table 4 below.

Convergence. Next we test the order of spatial accuracy in the first order partially implicit code; it is designed to be second order in space. With time step Δt and number of time steps fixed, we perform runs with $N = 100, 200, 400$. We then calculate an empirical order of accuracy, as in [24], except that we do not interpolate; we compare values only on the coarsest grid for both velocity field and interface position. Thus the order for the velocity field is

$$p = \log_2 \|\mathbf{u}_h - \mathbf{u}_{h/2}\| / \|\mathbf{u}_{h/2} - \mathbf{u}_{h/4}\|, \tag{73}$$

where the norm is on the coarsest grid, and similarly for the interface. In Table 3 the smallest value of $\Delta t/h$, the number of time steps, and the final time are displayed. (E.g. in the first line for $N = 100, 200, 400$, $\Delta t/h = 10, 20, 40$, resp.) Empirical convergence rates are given for the velocity field and for the interface position, at the final time, in discrete L^2 and L^∞ norms. In the longer runs with $v = .1$ the relative errors in interface position are 10^{-5} to $4 \cdot 10^{-5}$, even though the order of accuracy deteriorates. In similar tests with N fixed and Δt reduced we observe first order convergence in time.

To assess temporal convergence we must note the unusual dependence of the error in semi-Lagrangian methods on Δt. With second order accuracy in space and time, the expected error has the order $h^2 + (\Delta t)^2 + h^p/\Delta t$, where p is the order of

Table 3
Empirical convergence rates for the first order implicit scheme as $h \to 0, \Delta t$ fixed.

ν	Smallest $\Delta t/h$	Time steps	Final time	Velocity		Interface	
				L^2	L^∞	L^2	L^∞
.1	10	20	2	1.85	1.81	2.28	2.24
	10	100	10	1.85	1.81	1.44	1.45
	5	20	1	1.86	1.81	2.08	2.13
	5	100	5	1.85	1.82	1.44	1.45
.05	5	20	1	1.85	1.80	2.04	2.05
	5	100	5	1.85	1.81	1.32	1.34
	2.5	20	.5	1.90	1.88	1.96	1.97
	2.5	50	1.25	1.86	1.83	1.86	2.02
	2.5	100	2.5	1.85	1.82	1.58	1.61
.01	1	25	.25	1.85	1.63	1.37	1.49
	.5	25	.125	1.98	1.82	1.90	1.90
	.5	100	.5	1.96	1.86	1.97	1.88

Table 4
Empirical convergence rates for the second order implicit scheme, $\Delta t/h$ fixed.

ν	$\Delta t/h$	Time steps	Final time	Velocity		Interface	
				L^2	L^∞	L^2	L^∞
1	50	10	5	1.34	1.41	1.35	1.34
	5	10	.5	2.02	2.10	2.29	2.21
	5	100	5	2.98	2.55	4.81	4.59
.1	10	10	1	2.27	2.24	1.82	1.80
	5	10	.5	2.58	2.50	2.26	2.23
	5	20	1	2.10	2.04	1.30	1.29
.05	5	10	.5	2.23	2.36	1.63	1.65
	5	20	1	2.94	2.91	2.27	2.24
	2	10	.2	1.46	.85	2.40	2.41
	2	22	.44	3.90	3.15	3.24	3.33
	1	10	.1	2.23	2.24	1.88	1.81
	1	45	.45	.80	1.18	.71	1.01
.01	.25	10	.025	1.80	2.25	2.21	2.22
	.25	50	.125	1.90	1.84	1.72	1.62
	.25	100	.25	1.68	2.00	2.31	2.48

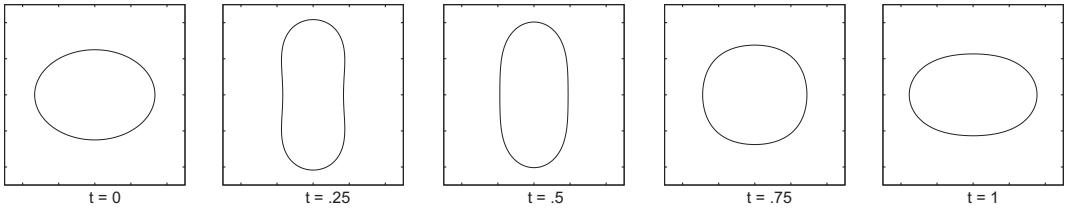

Fig. 3. The interface with $\gamma_0 = 1, \gamma_1 = 1$ at times t = 0, .25, .5, .75, 1.

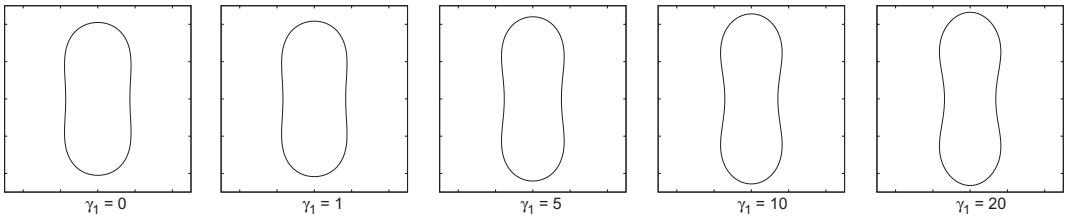

Fig. 4. The interface at time t = .25 with $\gamma_0 = 1$ and $\gamma_1 = 0, 1, 5, 10, 20$.

Table 5
Largest observed stable values of $\Delta t/h$, bending force.

c_b	ν	Explicit			First order			Second order		
		100	200	400	100	200	400	100	200	400
.01	1	1.2e−2	3e−3	1e−3	>50	>50	>50	>50	>50	>50
.1	1	1.5e−3	4e−4	1.2e−4	.5	1.0	1.6	.15	.25	.5
1	1	2e−4	5e−5	8e−6	.02	.03	.04	.007	.009	.012
10	1	1e−5	2e−6	6e−7	1e−4	1.5e−5	3e−6	3.5e−5	8e−6	2e−6
10	10	6e−6	2e−6	6e−7	.03	.05	.05	.015	.025	.05

Table 6
Angle of inclination and frequency of a vesicle in shear flow.

Minor radius	Reduced area	Angle/π				Frequency/χ			
		$\chi = 1$	10	50	100	$\chi = 1$	10	50	100
1/4	.55	.11	.09	.09	.09	.19	.22	.21	.21
1/3	.66	.12	.11	.11	.11	.23	.27	.26	.26
2/5	.75	.14	.12	.12	.12	.26	.30	.30	.29
1/2	.84	.15	.14	.14	.13	.31	.31	.34	.33
2/3	.94	.18	.15	.16	.15	.37	.38	.38	.36

accuracy of the interpolation (see [7,35]). Thus the error can actually get worse if Δt is decreased with h fixed. Indeed, we see this effect with our second order code. Instead we test accuracy by reducing Δt and h with $\Delta t/h$ constant. In Table 4 empirical orders of accuracy are found for the second order implicit code. We take $N = 100, 200, 400$, with $\Delta t/h$ and the final time fixed. We compute the order of accuracy according to (71) as before. The number of steps with $N = 100$ and the final time are displayed. Rates are given for the velocity field and for the interface position, at the final time, in discrete L^2 and L^∞ norms. In some cases the time is chosen to correspond to the half-cycle mentioned above. We see second order convergence with a small number of time steps, but with more steps the rate is less predictable.

Nonlinear tension force. We performed calculations with the same test problem as above, changing the force density to $\mathbf{f} = \partial_s((\gamma_0 s_\alpha + \gamma_1 s_\alpha^2)\mathbf{t})$, similar to examples used in [24,5]. We used a partially implicit approximation as in (66)–(68). We take α to be the angle θ above and choose $\nu = .01$. In contrast to the earlier case, the interface buckles inward at the left and right and later becomes convex again. The buckling is more pronounced if either γ_0 or γ_1 is increased. In Fig. 3 we show the interface position at times $0, .25, .5, .75, 1$ with $\gamma_0 = \gamma_1 = 1$. At time 1 the intercepts have reached a local extreme. Afterwards the interface oscillates and approaches equilibrium in a way similar to Fig. 1. In Fig. 4 we show the interface at time .25 for several choices of γ_1, with $\gamma_0 = 1$. The initial velocity is the Stokes velocity and thus depends on γ_1. The partially implicit time step appeared to have a stabilizing effect but less so than for the linear force.

Bending force. We compute examples of interface motion in Navier–Stokes flow with a bending force (5) on the interface combined with a tension force of the form (4), that is

$$\mathbf{f} = \gamma_0 \partial_s((s_\alpha - 1)\mathbf{t}) - c_b \partial_s^3 \mathbf{t}, \quad \mathbf{t} = \partial_s \mathbf{X} \tag{74}$$

where γ_0 and c_b are constants.

We begin by estimating stable time steps, as we did for the tension force in Table 1. We use the same initial curve (71), but we choose the initial markers to be equally spaced in arclength, and we take the initial arclength to be the material coordinate. The initial velocity is the Stokes velocity. We set $\nu = 1,\ \gamma_0 = 1$, and vary c_b. In this case the initial maximum Stokes velocity is about $5.9 c_b$. Results are reported in Table 5. For the explicit scheme, we see $\Delta t = O(h^3/c_b)$, which we would expect from Eq. (69) as well as the case of Stokes flow. For the implicit schemes, for each $c_b \leqslant 1$ the time step is $O(h)$, but for $c_b = 10$, it reverts to $O(h^3)$. This seems to occur for large c_b and small ν. We tried increasing ν to 10 with $c_b = 10$ (last line of the table) and found that the relation $\Delta t = O(h)$ was restored.

There have been careful numerical studies of inextensible vesicles in Stokes flow; the inextensibility means that the interface cannot stretch or contract, so that $s_\alpha = \partial s/\partial\alpha \equiv 1$. This constraint is imposed by the choice of a variable, unknown tension force [28,13,30,34]. Methods for inextensible vesicles in Navier–Stokes flow have also been introduced [12,21]. In [12] results were compared with those for Stokes flow in [13]. Here we do not impose inextensibility but instead choose large constant γ_0 in (74), with the expectation that the tension term will keep s_α near 1 and thus approximate the inextensible constraint. (This analogy was suggested by M.-C. Lai.) We find behavior similar to that reported for the inextensible case.

We use a choice of physical parameters as in [12], Section 4.1, similar to that in [13], in order to compare with the earlier work. We compute nondimensionalized solutions in a square $(-\pi, \pi) \times (-\pi, \pi)$ with periodic boundary conditions. We choose time scale $T = .1$ sec as in [12]. Our typical initial state is an ellipse with semimajor axis 1 in nondimensional units, corresponding to about $L = 20\ \mu$m. With scales T and L, the parameters in [12] lead to nondimensional bending coefficient

$c_b = 30$ and viscosity $\nu = 250$. We choose $\gamma_0 = 10^5$ to 10^6 to keep s_α near 1. In the regime described, the flow is dominated by Stokes flow. We impose the shear velocity $\chi(\sin y, 0)$, where χ is the dimensionless shear rate in [12,13].

In the study of inextensible vesicles in 3D shear flow in [13] it was noted that, with shear flow imposed, the vesicle approaches a state with a characteristic angle of inclination. The velocity on the interface becomes nearly tangential, so that particles on the interface follow a "tanktreading" motion, rotating with a frequency ω. The angle and ω/χ were found to depend significantly on the reduced volume but much less on the shear rate χ. Subsequent studies have found quantitatively similar results for 2D Stokes flow [34] and 2D Navier–Stokes flow [12]. In 2D we use the reduced area $A/(\pi R_0^2)$ where $R_0 = L_0/(2\pi)$, A is the area of the vesicle, and L_0 is its perimeter. In our experiments we chose the initial state as an ellipse with semimajor axis 1 and semiminor axis b, with $b = 1/4, 1/3, 2/5, 1/2, 2/3$; the reduced area is determined by b. We tested shear rates $\chi = 1, 10, 50, 100$. After the interface velocity became close to tangential, we measured the angle and the frequency $\omega = 2\pi/T_0$, where the period of rotation T_0 was found as $T_0 = \int (v^{\tan})^{-1} ds$. Values of the angle of inclination and ω/χ are shown in Table 6. The results are generally consistent with those in [13,34,12].

6. Conclusions

We have developed a numerical method for a moving elastic interface in Navier–Stokes flow which is partially implicit in the sense that the time step (30) for the interface location uses a modification of the current velocity in the high wavenumbers to predict the new interface. The approximation is derived analytically in Section 4 using a procedure like that in [9,11]. The partially implicit interface motion is combined with the velocity decomposition method of [3]. The numerical tests in Section 5 demonstrate that the new method is practical and accurate for a variety of problems with tension and bending forces at the interface. Stability was maintained with time steps much larger than for the corresponding explicit method, with negligible extra effort per step. The validity of the partially implicit approximation was demonstrated and convergence was verified. The discontinuities in pressure and velocity gradient are preserved in this method, as in the IIM, whereas they are regularized in the IBM. This distinction between the two methods makes their performance difficult to compare, but both are representative of widely used approaches for dealing with subgrid effects in difference methods for continuum problems.

The derivation in Section 4 amounts to a linear stability analysis of the direct contribution to the interface velocity from the force. Formula (61) predicts that explicit interface motion should have time step constraint $\Delta t = O(h\nu/\gamma_0)$ for linear tension force, whereas (69) gives $\Delta t = O(h^3\nu/c_b)$ for the bending force. These are obtained in the limit $k \to O(1/h)$ and are the same as for Stokes flow. However, the factor $g_\ell(\eta)$ introduces a second length scale, since $\eta = O(\sqrt{\nu\tau}k)$, distinct from Stokes flow. The partially implicit method controls the term that appears most important, and the time step is greatly improved, but it depends on the physical parameters, and the method is far from being unconditionally stable.

The time step in this implementation appears to be determined by the CFL condition, $\Delta t \approx h/V$, where V is the magnitude of velocity. This may be due to the manner of interpolation of the Stokes velocity, needed for the semi-Lagrangian method, taking into account the jump in the velocity gradient. In principle the stability could be improved, especially since the velocity decomposition offers flexibility in the separate choices of methods for solving the Stokes problem and the remainder problem. With the IBM, convection can be treated implicitly to overcome the CFL limitation [24], and presumably a similar method could be used here for the remainder problem.

The primary method developed here is second order in space and first order in time. It can be made second order in time by extrapolation. With the first order method it is easy to use adaptive time steps based on the CFL condition. The second order method gains accuracy but has somewhat smaller time steps, though still much larger than those of the explicit method.

The derivation of the partially implicit time step applies to Navier–Stokes flow with a variety of forces, including the bending force (5), which especially leads to stiffness because of the high order derivative. The analysis is general enough to apply to other models. The combination of bending and tension forces can mimic the case of an inextensible membrane, but our approach here does not apply directly to the inextensible case. The approximation derived here could in principle be used with other sharp interface methods not using the velocity decomposition. A promising alternative would be to use an approximation such as the present one as a preconditioner in an implicit method. Such a connection was made in the case of Stokes flow in [30,34].

Acknowledgements

We thank Anita T. Layton for cooperation with this work and for numerous helpful comments. We thank Ming-Chih Lai for suggesting the analogy between a large tension coefficient and the inextensibility condition at a Workshop on Fluid–Structure Interaction Problems sponsored by the National Center for Theoretical Sciences, Taiwan. This work was supported in part by the National Science Foundation under Grant DMS-0806482.

References

[1] J.T. Beale, T.Y. Hou, J.S. Lowengrub, Growth rates for the linearized motion of fluid interfaces away from equilibrium, Commun. Pure Appl. Math. 46 (1993) 1269–1301.
[2] J.T. Beale, M.-C. Lai, A method for computing nearly singular integrals, SIAM J. Numer. Anal. 38 (2001) 1902–1925.

[3] J.T. Beale, A.T. Layton, A velocity decomposition approach for moving interfaces in viscous fluids, J. Comput. Phys. 228 (2009) 3358–3367.
[4] H.D. Ceniceros, J.E. Fisher, A fast, robust, and non-stiff immersed boundary method, J. Comput. Phys. 230 (2011) 5133–5153.
[5] H.D. Ceniceros, J.E. Fisher, A.M. Roma, Efficient solutions to robust, semi-implicit discretizations of the immersed boundary method, J. Comput. Phys. 228 (2009) 7137–7158.
[6] R. Cortez, The method of regularized Stokeslets, SIAM J. Sci. Comput. 23 (2001) 1204–1225.
[7] D. Durran, Numerical Methods for Fluid Dynamics with Application to Geophysics, Springer, New York, 2010.
[8] B. Griffith, 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 (2005) 75–105.
[9] T.Y. Hou, J.S. Lowengrub, M.J. Shelley, Removing the stiffness from interfacial flows with surface tension, J. Comput. Phys. 114 (1994) 312–338.
[10] T.Y. Hou, Z. Shi, Removing the stiffness of elastic force from the immersed boundary method for the 2D Stokes equations, J. Comput. Phys. 227 (2008) 9138–9169.
[11] T.Y. Hou, Z. Shi, An efficient semi-implicit immersed boundary method for the Navier–Stokes equations, J. Comput. Phys. 227 (2008) 8968–8991.
[12] Y. Kim, M.-C. Lai, Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method, J. Comput. Phys. 229 (2010) 4840–4853.
[13] M. Kraus, W. Wintz, U. Seifert, R. Lipowsky, Fluid vesicles in shear flow, Phys. Rev. Lett. 77 (1996) 3685–3688.
[14] M.C.A. Kropinski, An efficient numerical method for studying interfacial motion in two-dimensional creeping flows, J. Comput. Phys. 171 (2001) 479–508.
[15] A.T. Layton, J.T. Beale, A partially implicit hybrid method for computing interface motion in Stokes flow, Discrete Contin. Dyn. Syst. Ser. B 17 (2012) 1139–1153.
[16] D.V. Le, J. White, J. Peraire, K.M. Lim, B.C. Khoo, An implicit immersed boundary method for three-dimensional fluid membrane interactions, J. Comput. Phys. 228 (2009) 8427–8445.
[17] L. Lee, R. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.
[18] 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.
[19] Z. Li, K. Ito, The Immersed Interface Method, SIAM, Philadelphia, 2006.
[20] Z. Li, M.-C. Lai, The immersed interface method for the Navier–Stokes equations with singular forces, J. Comput. Phys. 171 (2001) 822–842.
[21] Z. Li, M.-C. Lai, New finite difference methods based on IIM for inextensible interfaces in incompressible flows, East Asian J. Appl. Math. 1 (2011) 155–171.
[22] A. Mayo, Fast high order accurate solution of Laplace's equation on irregular regions, SIAM J. Sci. Statist. Comput. 6 (1985) 144–157.
[23] A. Mayo, C. Peskin, An implicit numerical method for fluid dynamics problems with immersed elastic boundaries, Contemp. Math. 141 (1993) 261–277.
[24] Y. Mori, C.S. Peskin, Implicit second order immersed boundary methods with boundary mass, Comput. Methods Appl. Mech. Eng. 197 (2008) 2049–2067.
[25] E.P. Newren, A.L. Fogelson, R.D. Guy, R.M. Kirby, Unconditionally stable discretizations of the immersed boundary equations, J. Comput. Phys. 222 (2007) 702–719.
[26] C. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[27] C. Peskin, B. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries, J. Comput. Phys. 105 (1993) 33–46.
[28] C. Pozrikidis, The axisymmetric deformation of a red blood cell in uniaxial straining Stokes flow, J. Fluid Mech. 216 (1990) 231–254.
[29] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, Cambridge, 1992.
[30] J.S. Sohn, Y.-H. Tseng, S. Li, A. Voigt, J.S. Lowengrub, Dynamics of multicomponent vesicles in a viscous fluid, J. Comput. Phys. 229 (2010) 119–144.
[31] Z. Tan, D.V. Le, Z. Li, K.M. Lim, B.C. Khoo, An immersed interface method for solving incompressible viscous flows with piecewise constant vorticity across a moving elastic membrane, J. Comput. Phys. 227 (2008) 9955–9983.
[32] A.-K. Tornberg, M.J. Shelley, Simulating the dynamics and interactions of flexible fibers in Stokes flows, J. Comput. Phys. 196 (2004) 8–40.
[33] 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. Statist. Comput. 13 (1992) 1361–1376.
[34] S.K. Veerapaneni, D. Gueyffier, D. Zorin, G. Biros, A boundary integral method for simulating the dynamics of inextensible vesicles suspended in a viscous fluid in 2D, J. Comput. Phys. 228 (2009) 2334–2353.
[35] D. Xiu, G. Karniadakis, A semi-Lagrangian high-order method for Navier–Stokes equations, J. Comput. Phys. 172 (2001) 658–684.
[36] S. Xu, Z.J. Wang, An immersed interface method for simulating the interaction of a fluid with moving boundaries, J. Comput. Phys. 216 (2006) 454–493.