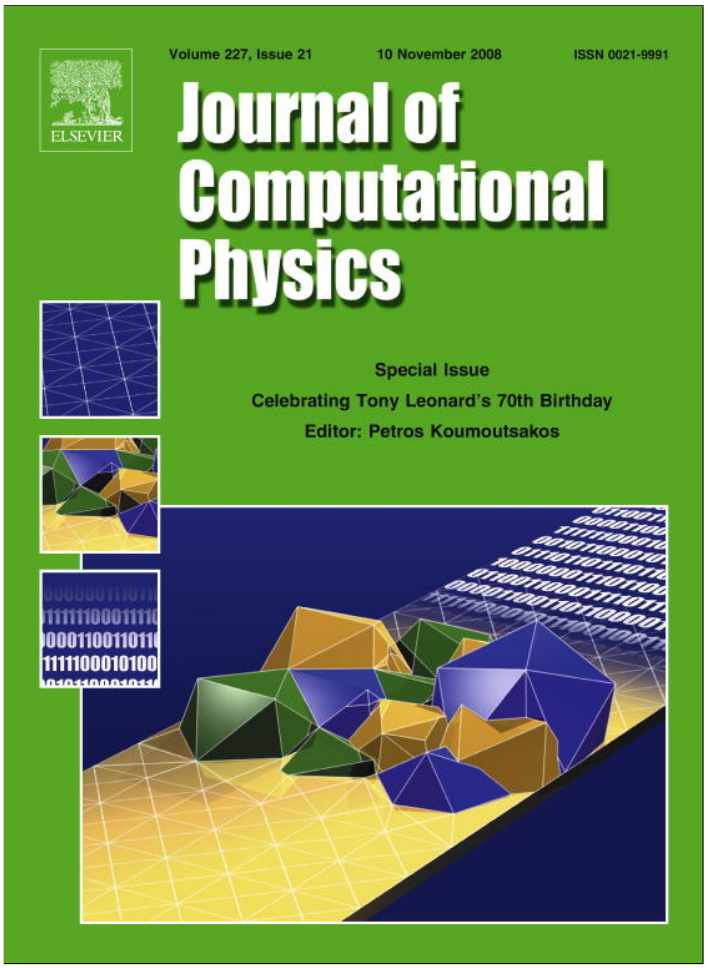
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 9138–9169
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Removing the stiffness of elastic force from the immersed boundary method for the 2D Stokes equations

Thomas Y. Hou [a,*], Zuoqiang Shi [a,b]

[a] Applied and Computational Mathematics, Caltech, Pasadena, CA 91125, United States
[b] Zhou Pei-Yuan Center for Applied Mathematics, Tsinghua University, Beijing 100084, China

Received 16 June 2007; received in revised form 21 December 2007; accepted 2 March 2008
Available online 10 March 2008

Abstract

The immersed boundary method has evolved into one of the most useful computational methods in studying fluid structure interaction. On the other hand, the immersed boundary method is also known to suffer from a severe timestep stability restriction when using an explicit time discretization. In this paper, we propose several efficient semi-implicit schemes to remove this stiffness from the immersed boundary method for the two-dimensional Stokes flow. First, we obtain a novel unconditionally stable semi-implicit discretization for the immersed boundary problem. Using this unconditionally stable discretization as a building block, we derive several efficient semi-implicit schemes for the immersed boundary problem by applying the small scale decomposition to this unconditionally stable discretization. Our stability analysis and extensive numerical experiments show that our semi-implicit schemes offer much better stability property than the explicit scheme. Unlike other implicit or semi-implicit schemes proposed in the literature, our semi-implicit schemes can be solved explicitly in the spectral space. Thus the computational cost of our semi-implicit schemes is comparable to that of an explicit scheme, but with a much better stability property.
© 2008 Elsevier Inc. All rights reserved.

Keywords: Incompressible flow; Stokes equations; Immersed boundary method; Removing the stiffness

1. Introduction

The immersed boundary method was originally introduced by Peskin in the 1970s to model the flow around heart valves. Now it has evolved into a general useful method in studying the motion of one or more massless, elastic surface immersed in an incompressible, viscous fluid, particularly in biofluid dynamics problems where complex geometries and immersed elastic membranes are present. The method has been successfully applied to a variety of problems including blood flow in the heart [25,16–18,26,19,20], vibrations of the cochlear basilar membrane [2,8], platelet aggregation during clotting [7,34], aquatic locomotion [5,6,11,35,3], flow with suspended particles [5,31], and inset flight [21,22], We refer to [27] for an extensive list of applications.

* Corresponding author.
E-mail addresses: hou@acm.caltech.edu (T.Y. Hou), shi@acm.caltech.edu (Z. Shi).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.03.002

Author's personal copy

The immersed boundary method employs a uniform Eulerian grid over the entire domain to describe the velocity field of the fluid and a Lagrangian description for the immersed elastic structure. The force generated by the elastic structure drives the fluid and the fluid moves the elastic structure. This interaction is expressed in terms of the spreading and interpolation operations by use of smoothed Delta functions. We remark that LeVeque and Li have introduced the Immersed Interface Method in [14] in which they avoid using the smoothed Delta functions. Instead they use the jump conditions to modify their finite difference discretization near the interface to obtain a more accurate discretization.

One of the main difficulties that the immersed boundary method encounters is that it suffers from a severe timestep restriction in order to keep the stability [27,32,30]. This has been the major limitation of the immersed boundary method. This restriction is typically much more severe than the one that would be imposed from using an explicit discretization for the convection term in the Navier–Stokes equation. The instability is known to arise from large boundary force and small viscosity [32]. Much effort has been made to remove this restriction. Some implicit and semi-implicit methods have been proposed in the literature [33,23,15]. Despite of these efforts, the timestep restriction has not been resolved satisfactorily. The computational cost of using an implicit or semi-implicit scheme is still too high to be effective in a practical computation. To date, almost all practical computations using the immersed boundary method have been performed using an explicit discretization.

In this paper, we develop several efficient semi-implicit schemes to compute the motion of an elastic interface immersed in a two-dimensional, incompressible Stokes flow. There are several important ingredients in deriving our semi-implicit schemes. The first one is to use the arclength and tangent angle formulation to describe the dynamics of the immersed interface [9]. We remark that Ceniceros and Roma have also used the arclength and tangent angle formulation to alleviate the stiffness of the viscous vortex sheet with surface tension in [4]. The second one is to obtain an unconditionally stable semi-implicit discretization of the immersed boundary problem. Throughout this paper, we use the term "stability" to mean that the energy norm of the solution can be bounded in terms of the energy norm of the initial data, which is a weaker result than proving that the difference between two solutions in the energy norm can be bounded in terms of the energy norm of their difference at time zero. The third ingredient is to perform small scale decomposition to the unconditionally stable discretization to obtain our efficient semi-implicit schemes. An important feature of our small scale decomposition is that the leading order term, which is to be discretized implicitly, can be expressed as a convolution operator. This property enables us to solve for the implicit solution explicitly using the Fourier transformation. Thus, the computational cost of our semi-implicit schemes is comparable to that of an explicit method. This offers a significant computational saving in using the immersed boundary method.

The small scale decomposition was first developed by Hou et al. [9,10]. They applied this method to remove the stiffness from interfacial flow with surface tension, which has proved to be very successful. Due to the coupling between the elastic boundary with the fluid, it is more difficult to remove the stiffness induced by the elastic force in the immersed boundary method. To remove the stiffness in the immersed boundary method, we need to decouple the stiffness induced by the elastic force from the fluid flow in such a way that the resulting semi-implicit discretization is still unconditionally stable. This is accomplished by using a semi-implicit discretization which preserves certain important solution structures which exist at the continuous level. Without obtaining this unconditionally stable semi-implicit discretization, a straightforward application of the small scale decomposition to the immersed boundary method would not provide an efficient semi-implicit scheme with the desirable stability property. Very recently, Newren et al. have obtained an unconditionally stable discretization for linear force in [24]. However, they did not perform small scale decomposition to their unconditionally stable discretization. As we will demonstrate in this paper, the unconditionally stable semi-implicit discretization without using the small scale decomposition is still very expensive and the gain over the explicit discretization is quite limited.

We develop several efficient semi-implicit schemes for both the steady Stokes flow and the unsteady Stokes flow respectively. In both cases, our semi-implicit schemes work very well. In the steady Stokes flow, we also develop a fourth order semi-implicit scheme by using the integral factor method. For the unsteady Stokes flow, we develop a second order semi-implicit method by combining our small scale decomposition with a well known second order temporal discretization [13,27]. To illustrate the stability properties of our semi-implicit schemes, we apply our methods to several prototype problems and test our schemes for a range of elastic coefficients and viscosity coefficients. Our numerical results confirm that the semi-implicit schemes remove the high

Author's personal copy

order stability constraint induced by the elastic force. In the case of unsteady Stokes equation, we also confirm the second order accuracy of our semi-implicit scheme.

This paper is organized as follows. First, we review the classical formulation of the immersed boundary method in Section 2. Then, we introduce the arclength and tangent angle formulation in Section 3. In Section 4, we describe the spatial discretization of the immersed boundary method. In Sections 5 and 6, we develop the numerical schemes for steady Stokes flow and unsteady Stokes flow respectively. The numerical results are presented in Section 7. Our numerical studies will focus on the stability restriction and computational cost of our methods. Some concluding remarks are given in Section 8.

2. Review of the immersed boundary method

For simplicity, we just consider a viscous incompressible fluid in a two-dimensional domain Ω, containing an immersed massless elastic boundary in the form of a closed simple curve Γ. The configuration of the boundary is given in a parametric form: $\mathbf{X}(\alpha,t), 0 \leqslant \alpha \leqslant L_b$, $\mathbf{X}(0,t) = \mathbf{X}(L_b,t)$, α tracks a material point of the boundary. We consider only the Stokes equations in this paper and would neglect the convection term. Then the governing equations are given as follows:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mu \triangle \mathbf{u} + \mathbf{f}(\mathbf{x},t), \tag{1}$$

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

$$\frac{\partial \mathbf{X}}{\partial t}(\alpha,t) = \mathbf{u}(\mathbf{X}(\alpha,t),t), \tag{3}$$

where $\mathbf{u}$ is the fluid velocity, p is the pressure, ρ and μ are constant fluid density and viscosity, $\mathbf{f}(\mathbf{x},t)$ is the force density, which is not zero only on the boundary and which is infinite there. The force density can be expressed as below

$$\mathbf{f}(\mathbf{x},t) = \int_0^{L_b} \mathbf{F}(\alpha,t)\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\alpha, \tag{4}$$

δ denotes the two-dimensional Dirac delta function and

$$\mathbf{F}(\alpha,t) = \frac{\partial}{\partial \alpha}(T\boldsymbol{\tau}), \tag{5}$$

$$T = T\left(\left|\frac{\partial \mathbf{X}}{\partial \alpha}\right|\right). \tag{6}$$

The choice of function T in this paper is computed by Hook's law

$$T = S_b\left(\left|\frac{\partial \mathbf{X}}{\partial \alpha}\right| - 1\right), \tag{7}$$

where S_b is the elastic coefficient of the boundary, and $\boldsymbol{\tau}$ is the unit tangent vector along the boundary, which is defined as

$$\boldsymbol{\tau} = \frac{\partial \mathbf{X}}{\partial s} \Big/ \left|\frac{\partial \mathbf{X}}{\partial s}\right|. \tag{8}$$

This choice of force density has been used widely in the literature in both computational and theoretical studies [12,29,33].

We can rewrite (3) in the following way:

$$\frac{\partial \mathbf{X}}{\partial t}(\alpha,t) = \int_\Omega \mathbf{u}(\mathbf{x},t)\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\mathbf{x}. \tag{9}$$

Next, we introduce the spreading and interpolation operations. The spreading and interpolation operators are defined as follows:

Author's personal copy

$$L(\mathbf{X})(g(\alpha))(\mathbf{x}) = \int_\Gamma g(\alpha)\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\alpha, \tag{10}$$

$$L^*(\mathbf{X})(u(\mathbf{x}))(\alpha) = \int_\Omega u(\mathbf{x})\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\mathbf{x}. \tag{11}$$

It is easy to show that L and L^* are adjoint operators:

$$\begin{aligned} <\mathbf{u}(\mathbf{x}), L(\mathbf{X})(g(\alpha))>_\Omega &= \int_\Omega \mathbf{u}(\mathbf{x})\left(\int_\Gamma g(\alpha)\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\alpha\right)\mathrm{d}\mathbf{x} \\ &= \int_\Omega \int_\Gamma \mathbf{u}(\mathbf{x})g(\alpha)\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\alpha\,\mathrm{d}\mathbf{x} = \int_\Gamma \int_\Omega \mathbf{u}(\mathbf{x})g(\alpha)\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\mathbf{x}\,\mathrm{d}\alpha \\ &= \int_\Gamma g(\alpha)\left(\int_\Omega \mathbf{u}(\mathbf{x})\delta(\mathbf{x} - \mathbf{X}(\alpha,t))\mathrm{d}\mathbf{x}\right)\mathrm{d}\alpha = < L^*(\mathbf{X})(u(\mathbf{x})),\quad g(\alpha)>_\Gamma, \end{aligned} \tag{12}$$

where the inner product are defined as follows:

$$\langle u, v\rangle_\Omega = \int_\Omega u(\mathbf{x})v(\mathbf{x})\mathrm{d}\mathbf{x}, \tag{13}$$

$$\langle f, g\rangle_\Gamma = \int_\Gamma f(\alpha)g(\alpha)\mathrm{d}\alpha. \tag{14}$$

Eqs. (1) and (2) are the familiar Stokes equations of viscous incompressible fluid. Eqs. (3) and (4) represent the interaction of the fluid and the elastic boundary. The elastic boundary applies the force to the fluid, the fluid carries the immersed boundary, and the force density is determined by the configuration of the boundary.

3. The arclength-tangent angle formulation

In studying the evolution of a curve, it is useful to represent the curve by its tangent angle θ and local arclength derivative s_α. Previously, Hou et al. [9] exploited this formulation and combined it with a so-called "small scale decomposition" reformulation to remove the stiffness induced by surface tension.

Consider the evolution of a simply closed curve Γ with known normal and tangent velocity fields, U, V. Assume the curve is represented by $\mathbf{X}(\alpha,t), \alpha \in [0, L_\mathrm{b}]$. We define the arclength derivative, s_α, and the tangent vector, θ, as follows

$$s_\alpha(\alpha,t) = |\mathbf{X}_\alpha(\alpha,t)|, \tag{15}$$

$$(x_\alpha(\alpha,t), y_\alpha(\alpha,t)) = s_\alpha(\alpha,t)(\cos\theta(\alpha,t), \sin\theta(\alpha,t)). \tag{16}$$

The closed curve Γ evolves according to

$$\frac{\partial \mathbf{X}}{\partial t} = \mathbf{u}(\mathbf{X},t) = U\mathbf{n} + V\boldsymbol{\tau}, \tag{17}$$

where $\boldsymbol{\tau}$ and $\mathbf{n}$ are the unit tangent and normal vectors of the curve respectively. According to the Frenet formula, we have $\frac{\partial \boldsymbol{\tau}}{\partial s} = k\mathbf{n}, \frac{\partial \mathbf{n}}{\partial s} = -k\boldsymbol{\tau}$, here s is the arclength variable. It is easy to see that s_α and θ satisfy the following evolution equations [9]:

$$(s_\alpha)_t = V_\alpha - \theta_\alpha U, \tag{18}$$

$$\theta_t = \frac{U_\alpha}{s_\alpha} + \frac{V\theta_\alpha}{s_\alpha}. \tag{19}$$

Given s_α and θ, the curve Γ can be reconstructed up to a translation by integrating (16). However, we also need a point on the boundary to provide the constant of integration.

Using the $s_\alpha - \theta$ formulation, we can reformulate the immersed boundary problem as follows:

$$\rho\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mu\triangle\mathbf{u} + L(\mathbf{X})(\mathbf{F}(s_\alpha,\theta)), \tag{20}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{21}$$

Author's personal copy

$$U = L^*(\mathbf{X})(\mathbf{u}(\mathbf{x})) \cdot \mathbf{n}, \tag{22}$$

$$V = L^*(\mathbf{X})(\mathbf{u}(\mathbf{x})) \cdot \boldsymbol{\tau}, \tag{23}$$

$$(s_\alpha)_t = V_\alpha - \theta_\alpha U, \tag{24}$$

$$\theta_t = \frac{U_\alpha}{s_\alpha} + \frac{V\theta_\alpha}{s_\alpha}, \tag{25}$$

where

$$\mathbf{F}(s_\alpha, \theta) = \frac{\partial}{\partial \alpha}(T\boldsymbol{\tau}) = S_b(s_{\alpha,\alpha}\boldsymbol{\tau} + (s_\alpha - 1)\theta_\alpha \mathbf{n}). \tag{26}$$

4. Spatial discretization

We use the spectral method to discretize both the Stokes equations and the immersed boundary equations in space since we work on periodic domains. We first discuss the discretization of the Stokes equations in a regular $N \times N$ Cartesian grid with a uniform meshsize h. Let $x_j = jh$ and $y_j = jh$. The discrete Fourier transform and inverse Fourier transform are defined as follows:

$$\mathcal{F}_{h,x}(\phi)(k,y) = \frac{1}{N}\sum_{j=0}^{N-1} \phi(x_j, y)\mathrm{e}^{-\mathrm{i}kx_j} = \widehat{\phi}(k,y), \quad -N/2+1 \leqslant k \leqslant N/2, \tag{27}$$

$$\mathcal{F}_{h,y}(\phi)(x,k) = \frac{1}{N}\sum_{j=0}^{N-1} \phi(x, y_j)\mathrm{e}^{-\mathrm{i}ky_j} = \widehat{\phi}(x,k), \quad -N/2+1 \leqslant k \leqslant N/2, \tag{28}$$

$$\mathcal{F}_{h,x}^{-1}(\widehat{\phi})(x_j,y) = \sum_{k=-N/2+1}^{N/2} \widehat{\phi}(k,y)\mathrm{e}^{\mathrm{i}kx_j} = \phi(x_j,y), \quad 0 \leqslant j \leqslant N-1, \tag{29}$$

$$\mathcal{F}_{h,y}^{-1}(\widehat{\phi})(x,y_j) = \sum_{k=-N/2+1}^{N/2} \widehat{\phi}(x,k)\mathrm{e}^{\mathrm{i}ky_j} = \phi(x,y_j), \quad 0 \leqslant j \leqslant N-1. \tag{30}$$

Now we introduce the discrete differential operator using the discrete Fourier transform defined above. For a function $\phi(x,y)$ defined in the fluid domain Ω, we approximate its spatial derivatives as follows:

$$(D_{h,x}\phi)(x,y) = \mathcal{F}_{h,x}^{-1}(\mathrm{i}k(\mathcal{F}_{h,x}\phi)(k,y)), \tag{31}$$

$$(D_{h,y}\phi)(x,y) = \mathcal{F}_{h,y}^{-1}(\mathrm{i}k(\mathcal{F}_{h,y}\phi)(x,k)). \tag{32}$$

Denote $\nabla_h = (D_{h,x}, D_{h,y})$. The differential operators are discretized in terms of $\mathbf{D}_h$:

$$\nabla p \to \nabla_h p, \tag{33}$$

$$\nabla \cdot \mathbf{u} \to \nabla_h \cdot \mathbf{u}, \tag{34}$$

$$\nabla^2 u \to \nabla_h \cdot \nabla_h u \equiv \nabla_h^2 u. \tag{35}$$

Next, we describe the discretization of the immersed boundary. We employ a Lagrangian grid with grid space $\Delta\alpha$. The number of grid points along the boundary is N_b. For a function $\psi(\alpha)$ defined on the interface Γ, we define the discrete Fourier transform and its inverse as follows:

$$\mathcal{F}_{\Delta\alpha}(\psi)(k) = \frac{1}{N_b}\sum_{j=0}^{N_b-1} \phi(\alpha_j)\mathrm{e}^{-\mathrm{i}k\alpha_j} = \widehat{\psi}(k), \quad \alpha_j = j\Delta\alpha, \tag{36}$$

$$\mathcal{F}_{\Delta\alpha}^{-1}(\widehat{\psi})(\alpha_j) = \sum_{k=-\frac{N_b}{2}+1}^{\frac{N_b}{2}} \widehat{\psi}(k)\mathrm{e}^{\mathrm{i}k\alpha_j} = \psi(\alpha_j). \tag{37}$$

Author's personal copy

When the interface is a closed curve, we can approximate the derivative operator along the interface by the spectral derivative:

$$(D_{\Delta\alpha}\psi)(\alpha) = \mathcal{F}^{-1}_{\Delta\alpha}(\mathrm{i}k(\mathcal{F}_{\Delta\alpha}\phi)(k)). \tag{38}$$

When the solution is not periodic, we can also use a finite difference method to discretize the derivative, we refer to [27] for more details.

Now we discuss the discretization of the spreading and interpolation operators. These two operators both involve the use of a discrete delta function. The discrete delta function we use is introduced by Peskin in [27]:

$$\delta_h(x,y) = \frac{1}{h^2}\phi\left(\frac{x}{h}\right)\phi\left(\frac{y}{h}\right), \tag{39}$$

and

$$\phi(r) = \begin{cases} \frac{1}{8}\left(3 - 2|r| + \sqrt{1+4|r|-4r^2}\right), & |r| \leqslant 1, \\ \frac{1}{8}\left(5 - 2|r| - \sqrt{-7+12|r|-4r^2}\right), & 1 \leqslant |r| \leqslant 2, \\ 0, & |r| > 2. \end{cases} \tag{40}$$

Using the above discrete delta function, we can discretize the spreading and interpolation operator as follows

$$L_h(\mathbf{X})(g(\alpha))(\mathbf{x}) = \sum_{\alpha\in\mathcal{G}_\Gamma} g(\alpha)\delta_h(\mathbf{x} - \mathbf{X}(\alpha,t))\Delta\alpha, \tag{41}$$

$$L_h^*(\mathbf{X})(u(\mathbf{x}))(\alpha) = \sum_{\mathbf{x}\in\mathcal{G}_\Omega} u(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}(\alpha,t))h^2. \tag{42}$$

The summation above is over grid points in Γ in (41) and over grid points in Ω in (42). Operator L_h and L_h^* are still adjoint using the following discrete inner product:

$$\langle f, g\rangle_{\Gamma_h} = \sum_{\alpha\in\mathcal{G}_\Gamma} f(\alpha)g(\alpha)\Delta\alpha, \tag{43}$$

$$\langle u, v\rangle_{\Omega_h} = \sum_{\mathbf{x}\in\mathcal{G}_\Omega} u(\mathbf{x})v(\mathbf{x})h^2. \tag{44}$$

Using the inner product defined above, we have:

$$\begin{aligned} < u(\mathbf{x}), L(\mathbf{X})(g(\alpha))>_{\Omega_h} &= \sum_{\mathbf{x}\in\mathcal{G}_\Omega} u(\mathbf{x})L(\mathbf{X})(g(\alpha))h^2 = \sum_{\mathbf{x}\in\mathcal{G}_\Omega} u(\mathbf{x})h^2 \sum_{\alpha\in\mathcal{G}_\Gamma} g(\alpha)\delta_h(\mathbf{x} - \mathbf{X}(\alpha,t))\Delta\alpha \\ &= \sum_{\mathbf{x}\in\mathcal{G}_\Gamma} g(\alpha)\Delta\alpha \sum_{\alpha\in\mathcal{G}_\Omega} u(\mathbf{x})\delta_h(\mathbf{x} - \mathbf{X}(\alpha,t))h^2 =< L_h^*(\mathbf{X})(u(\mathbf{x})), g(\alpha)>_{\Gamma_h}. \end{aligned} \tag{45}$$

As we will see later, this discrete self-adjoint property is crucial in obtaining our unconditional stable semi-discrete scheme for the immersed boundary problem.

5. Steady Stokes flow

5.1. Formulation

For simplicity, we study the steady Stokes flow first. The governing equations for the steady Stokes flow are given as follows:

$$0 = -\nabla p + \mu\triangle\mathbf{u} + L(\mathbf{X})(\mathbf{F}(s_\alpha,\theta)), \tag{46}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{47}$$

$$U = \mathbf{u}(\mathbf{X}(\alpha,t),t)\cdot\mathbf{n}, \tag{48}$$

$$V = \mathbf{u}(\mathbf{X}(\alpha,t),t)\cdot\boldsymbol{\tau}, \tag{49}$$

$$(s_\alpha)_t = V_\alpha - \theta_\alpha U, \tag{50}$$

$$\theta_t = \frac{U_\alpha}{s_\alpha} + \frac{V\theta_\alpha}{s_\alpha}. \tag{51}$$

Author's personal copy

In this simple case, we can use a boundary integral method for the two dimension Stokes flow (page 60 of [28]) to solve Eqs. (46) and (47) to get the velocity on the boundary:

$$u(\mathbf{X}(\alpha,t)) = \frac{1}{4\pi\mu}\int_\Gamma \left(-(\ln r)F_1(\alpha',t) + \frac{r_1^2}{r^2}F_1(\alpha',t) + \frac{r_1 r_2}{r^2}F_2(\alpha',t)\right)\mathrm{d}\alpha', \tag{52}$$

$$v(\mathbf{X}(\alpha,t)) = \frac{1}{4\pi\mu}\int_\Gamma \left(-(\ln r)F_2(\alpha',t) + \frac{r_2^2}{r^2}F_2(\alpha',t) + \frac{r_1 r_2}{r^2}F_1(\alpha',t)\right)\mathrm{d}\alpha', \tag{53}$$

where $r = |\mathbf{r}|$ and

$$\mathbf{r} = (r_1, r_2) = \mathbf{X}(\alpha,t) - \mathbf{X}(\alpha',t), \quad \mathbf{F} = (F_1, F_2), \quad \mathbf{u} = (u, v). \tag{54}$$

5.2. Small scale decomposition

As we can see from (52) and (53), the velocity field can be expressed as a singular integral with a kernel $\ln(r)$. However, the singular velocity integral is nonlinear and non-local. It is difficult to solve for the implicit solution if we treat the velocity integral fully implicitly. The main idea of the small scale decomposition technique introduced in [9] is to decompose the singular velocity integral into the sum of a linear singular operator which is a convolution operator independent of time t and the configuration of the curve, and a remainder operator which is regular. Since the remaining operator, which is nonlinear and non-local, is regular, the simplified convolution integral operator captures accurately the high frequency spectral property of the original velocity integral. Thus, if we treat only the leading order convolution operator implicitly, but keep the regular remainder operator explicitly, we can effectively remove the stiffness of the original velocity field which comes mainly from the high frequency modes of the solution. In this subsection, we will show how to perform such small scale decomposition for the immersed boundary method applied to the Stokes flow.

Observe that in the integral representation of the velocity field, (52) and (53), the only singular part of the kernel is $\ln(r)$. The other part of the kernel is smooth. Thus to the leading order contribution of the velocity field can be expressed as follows:

$$\mathbf{u}(\mathbf{X}(\alpha,t)) \sim \frac{1}{4\pi\mu}\int_\Gamma -(\ln r)\mathbf{F}(\alpha',t)\mathrm{d}\alpha', \tag{55}$$

$$V = \mathbf{u}(\mathbf{X}(\alpha,t),t)\cdot\boldsymbol{\tau}(\alpha) \tag{56}$$

$$\sim \frac{1}{4\pi\mu}\int_\Gamma -(\ln r)\mathbf{F}(\alpha',t)\cdot\boldsymbol{\tau}(\alpha)\mathrm{d}\alpha'$$

$$= \frac{S_\mathrm{b}}{4\pi\mu}\int_\Gamma -(\ln r)(s_{\alpha,\alpha'}\boldsymbol{\tau}(\alpha') + (s_\alpha - 1)\theta_{\alpha'}\mathbf{n}(\alpha'))\cdot\boldsymbol{\tau}(\alpha)\mathrm{d}\alpha'. \tag{57}$$

Next, we perform a Taylor expansion for r, $\boldsymbol{\tau}(\alpha\prime)\cdot\boldsymbol{\tau}(\alpha)$ and $\mathbf{n}(\alpha\prime)\cdot\boldsymbol{\tau}(\alpha)$ as a function of $\alpha\prime$ around α. By keeping only the leading order term, we have

$$r \sim s_\alpha(\alpha)|\alpha - \alpha'|, \tag{58}$$

$$\boldsymbol{\tau}(\alpha')\cdot\boldsymbol{\tau}(\alpha) \sim 1, \tag{59}$$

$$\mathbf{n}(\alpha')\cdot\boldsymbol{\tau}(\alpha) \sim 0. \tag{60}$$

Substituting the above Taylor expansions to (55), we get

$$V \sim \frac{S_\mathrm{b}}{4\pi\mu}\int \ln(s_\alpha(\alpha)|\alpha - \alpha'|)s_{\alpha,\alpha'}\,\mathrm{d}\alpha'. \tag{61}$$

Integrating by part, we obtain

$$V \sim \frac{S_\mathrm{b}}{4\pi\mu}\int \frac{s_{\alpha'}}{\alpha' - \alpha}\mathrm{d}\alpha' = -\frac{S_\mathrm{b}}{4\mu}\mathcal{H}[s_\alpha], \tag{62}$$

Author's personal copy

where $\mathcal{H}$ is the Hilbert transform

$$\mathcal{H}[f](\alpha) = \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{f(\alpha')}{\alpha - \alpha'}\mathrm{d}\alpha'. \tag{63}$$

Using the same method, we can get the leading order contributions of U and U_α as follows:

$$U \sim \frac{S_b}{4\pi\mu}\int -\ln|\alpha - \alpha'|(s_{\alpha'} - 1)\theta_{\alpha'}\mathrm{d}\alpha' \tag{64}$$

$$U_\alpha \sim \frac{S_b}{4\pi\mu}\int \frac{(s_{\alpha'} - 1)\theta_{\alpha'}}{\alpha' - \alpha}\mathrm{d}\alpha' = -\frac{S_b}{4\mu}\mathcal{H}[(s_\alpha - 1)\theta_\alpha]. \tag{65}$$

Note that if f is a smooth function, then the commutator $[\mathcal{H}, f]u \equiv \mathcal{H}(fu) - f\mathcal{H}(u)$ is a smoothing operator for u. Thus we can factor a smooth function from the Hilbert transform without changing its leading order spectral property. Suppose that s_α is smooth, then we obtain to the leading order that

$$U_\alpha \sim -\frac{S_b}{4\mu}(s_\alpha - 1)\mathcal{H}[\theta_\alpha]. \tag{66}$$

Applying the same analysis to the Eqs. (50) and (51) gives

$$(s_\alpha)_t = -\frac{S_b}{4\mu}\mathcal{H}[D_{\Delta\alpha}s_\alpha] + \left(D_{\Delta\alpha}V - D_{\Delta\alpha}\theta U + \frac{S_b}{4\mu}\mathcal{H}[D_{\Delta\alpha}s_\alpha]\right), \tag{67}$$

$$\theta_t = -\frac{S_b}{4\mu}\left(1 - \frac{1}{s_\alpha}\right)\mathcal{H}[D_{\Delta\alpha}\theta] + \left(\frac{D_{\Delta\alpha}U}{s_\alpha} + \frac{VD_{\Delta\alpha}\theta}{s_\alpha} + \frac{S_b}{4\mu}\left(1 - \frac{1}{s_\alpha}\right)\mathcal{H}[D_{\Delta\alpha}\theta]\right). \tag{68}$$

Note that the leading order operator is linear. This suggests a natural semi-implicit discretization of the immersed boundary problem.

Since we are dealing with a closed immersed boundary, it is natural to work in the Fourier space. Furthermore, the Hilbert operator has a very simple kernel under the Fourier transformation. Notice that θ is not a periodic function of α. Its value increases 2π every time α increases L_b. Nevertheless, if we let

$$\theta(\alpha, t) = \frac{2\pi}{L_b}\alpha + \phi(\alpha, t), \quad \alpha \in [0, L_b], \tag{69}$$

then ϕ is periodic. It is more convenient to work with ϕ than θ. Substituting (69) into (68) and taking the Fourier transform on both sides of (67) and (68), we obtain

$$\hat{s}_{\alpha,t} = -\frac{S_b}{4\mu}|k|\hat{s}_\alpha + \left[\mathcal{F}(D_{\Delta\alpha}V - D_{\Delta\alpha}\theta U) + \frac{S_b}{4\mu}|k|\hat{s}_\alpha\right], \tag{70}$$

$$\hat{\phi}_t = -\frac{S_b}{4\mu}\gamma|k|\hat{\phi} + \left[\mathcal{F}\left(\frac{D_{\Delta\alpha}U}{s_\alpha} + \frac{VD_{\Delta\alpha}\theta}{s_\alpha}\right) + \frac{S_b}{4\mu}\gamma|k|\hat{\phi}\right], \tag{71}$$

where $\gamma = \max_\alpha(1 - \frac{1}{s_\alpha})$. We have also used the fact that $\widehat{\mathcal{H}}_k = -\mathrm{isgn}(k)$ with $\mathrm{sgn}(k)$ being the signature function. The first term on the right hand side captures the leading order high frequency contribution of the terms from the right hand side. An important property of this leading order term is that it is linear in $\hat{s}_\alpha$ and $\hat{\theta}$ and has constant coefficient in space. This provides a straightforward application of the implicit time discretization.

Since our small scale decomposition is exact near the equilibrium, we can use this result to get the stability constraint of the explicit scheme by using a frozen coefficient analysis. The stability constraint is given by

$$\Delta t < C\frac{\mu}{S_b}\frac{h}{\gamma}. \tag{72}$$

As we can see, the time step needs to be very small if S_b is large and μ is small. For example, if $S_b = 100$, and $\mu = 10^{-2}$, then the stability would require that $\Delta t \leqslant C10^{-4}h$.

Author's personal copy

5.3. Semi-implicit schemes

Based on the small scale decomposition presented in the previous subsection, we propose two types of semi-implicit schemes in this section. The first implicit time discretization uses the backward Euler method to discretize the leading order term while keeping the lower order term explicit. This gives rise to the following semi-implicit scheme:

$$\frac{\hat{s}_\alpha^{n+1}-\hat{s}_\alpha^n}{\Delta t} = -\frac{S_b}{4\mu}|k|\hat{s}_\alpha^{n+1} + \left[\mathcal{F}(D_{\Delta\alpha}V^n - D_{\Delta\alpha}\theta^n U^n) + \frac{S_b}{4\mu}|k|\hat{s}_\alpha^n\right], \tag{73}$$

$$\frac{\hat{\phi}^{n+1}-\hat{\phi}^n}{\Delta t} = -\frac{S_b}{4\mu}\gamma|k|\hat{\phi}^{n+1} + \left[\mathcal{F}\left(\frac{D_{\Delta\alpha}U^n}{s_\alpha^{n+1}} + \frac{V^n D_{\Delta\alpha}\theta^n}{s_\alpha^{n+1}}\right) + \frac{S_b}{4\mu}\gamma|k|\hat{\phi}^n\right]. \tag{74}$$

We call the above discretization the semi-implicit method. Near equilibrium, the stability constraint of this numerical method is $\Delta t < C(S_b, \mu)$, independent of the meshsize h. Since the small scale decomposition only captures the leading order contribution from the high frequency components, this method cannot eliminate the effect of S_b and μ completely. The coefficients S_b and μ can still affect the time stability through the low frequency components of the solution, which comes from the second term of the right hand side. In order to obtain a semi-implicit discretization with better stability property, we can incorporate the low frequency contribution from the second term in our implicit discretization. This scheme can be found in the Appendix A. We call it the semi-implicit scheme of second kind.

The accuracy of the semi-implicit schemes presented above is just first order. In order to get a high order time discretization, we can use the integral factor method. The integral factor method factors out the leading order linear term prior to time discretization. They usually provide stable and high order time integration methods for stiff problems. To use the integral factor method, we rewrite (70) and (71) as

$$\frac{\partial}{\partial t}(e^{\eta t}\hat{s}_\alpha) = \exp\left(\frac{S_b}{4\mu}|k|t\right)P(\hat{s}_\alpha, \hat{\phi}), \tag{75}$$

$$\frac{\partial}{\partial t}(e^{\xi t}\hat{\phi}) = \exp\left(\frac{S_b}{4\mu}\gamma|k|t\right)Q(\hat{s}_\alpha, \hat{\phi}), \tag{76}$$

where

$$\eta = \frac{S_b}{4\mu}|k|, \quad \xi = \frac{S_b}{4\mu}\gamma|k|, \tag{77}$$

$$P(\hat{s}_\alpha, \hat{\phi}) = \mathcal{F}(V_\alpha - \theta_\alpha U) + \frac{S_b}{4\mu}|k|\hat{s}_\alpha, \tag{78}$$

$$Q(\hat{s}_\alpha, \hat{\phi}) = \mathcal{F}\left(\frac{U_\alpha}{s_\alpha} + \frac{V\theta_\alpha}{s_\alpha}\right) + \frac{S_b}{4\mu}\gamma|k|\hat{\phi}. \tag{79}$$

Now it is straightforward to discretize this system to high order. In particular, we can apply the classical fourth order Runge–Kutta method to discretize the above system to obtain a fourth order semi-implicit scheme.

We remark that although the fourth order semi-implicit scheme based on the integral factor approach is much more accurate than the first order semi-implicit discretization, the stability of the fourth order method is weaker than the first semi-implicit scheme based on the backward Euler discretization. The fact that the higher order discretization gives a weaker stability property is a phenomenon which has been observed for almost all time integration methods. It is not a restriction of our semi-implicit schemes for the immersed boundary problem.

The semi-implicit schemes we describe above only update the θ and s_α variables. We also need to reconstruct the boundary at t^{n+1} from θ^{n+1} and s_α^{n+1}. For this purpose, we need to update a reference point of

Author's personal copy

the boundary. This will be done by using an explicit time integration method. The simplest one is the forward Euler method:

$$x^{n+1}(0) = x^n(0) + \Delta t(V^n \cos(\theta^n(0)) - U^n \sin(\theta^n(0))), \tag{80}$$

$$y^{n+1}(0) = y^n(0) + \Delta t(V^n \sin(\theta^n(0)) + U^n \cos(\theta^n(0))), \tag{81}$$

where U and V are evaluated at the reference point. A higher order integration method can be also used. In the explicit update of the reference point, we can use the values of U and V obtained using the semi-implicit discretization from the previous time steps to extrapolate the values of U and V in the intermediate time steps in our explicit update of the reference point. Once we have updated the reference point, we can obtain the configuration of the boundary (x, y) from (s_α, θ) by integrating (16)

$$x^{n+1}(\alpha) = x^{n+1}(0) + \int_0^\alpha s_\alpha^{n+1}(\alpha') \cos(\theta^{n+1}(\alpha'))\mathrm{d}\alpha', \tag{82}$$

$$y^{n+1}(\alpha) = y^{n+1}(0) + \int_0^\alpha s_\alpha^{n+1}(\alpha') \sin(\theta^{n+1}(\alpha'))\mathrm{d}\alpha'. \tag{83}$$

We can use more than one reference point, then average them to get the last configuration. This can improve the stability constraint significantly. Actually, in our computation, we use two reference points $X(0), X(N_b/2)$, then take the average to determine the position of the interface at next time step. Since we update only two reference points, the extra cost in updating the reference point is small compared with the overall computational cost.

6. Unsteady Stokes flow

6.1. Formulation

In this section, we will extend the semi-implicit discretization developed for the steady Stokes flow to the unsteady Stokes flow. The governing equations of the immersed boundary method for the unsteady Stokes flow are as follows:

$$\rho \frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \mu \triangle \mathbf{u} + L(\mathbf{X})(\mathbf{F}(s_\alpha, \theta)), \tag{84}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{85}$$

$$U = \mathbf{u}(\mathbf{X}(\alpha, t), t) \cdot \mathbf{n}, \tag{86}$$

$$V = \mathbf{u}(\mathbf{X}(\alpha, t), t) \cdot \boldsymbol{\tau}, \tag{87}$$

$$s_{\alpha t} = V_\alpha - \theta_\alpha U, \tag{88}$$

$$\theta_t = \frac{U_\alpha}{s_\alpha} + \frac{V\theta_\alpha}{s_\alpha}. \tag{89}$$

It is much more difficult to solve the fluid velocity $\mathbf{u}$ analytically from (84) and (85). As for the steady Stokes flow, we will first derive an unconditionally stable time discretization which will be given in next section and then apply the small scale decomposition to the unconditionally stable time discretization to obtain our efficient semi-implicit schemes.

6.2. An unconditionally stable semi-implicit discretization

In this section, we will describe our unconditionally stable semi-implicit discretization of the immersed boundary method for the incompressible unsteady Stokes equations and prove its unconditional stability in the sense of total energy is non-increasing.

The unconditionally stable semi-implicit discretization is consisted of two steps. In the first step, we update $s_\alpha, \mathbf{u}$ from t^n to t^{n+1}, then we get θ^{n+1} in the second step.

Step 1: Update of $\mathbf{u}^{n+1}$ and s_α^{n+1}.

Author's personal copy

$$\rho\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}=-\nabla_h p^{n+1}+\mu\nabla_h^2\mathbf{u}^{n+1}+L_{h,n}(\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{90}$$

$$\nabla_h^2 p^{n+1}=\nabla_h\cdot L_{h,n}(\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{91}$$

$$V^{n+1}=L_{h,n}^*(\mathbf{u}^{n+1})\cdot\boldsymbol{\tau}^n, \tag{92}$$

$$U^{n+1}=L_{h,n}^*(\mathbf{u}^{n+1})\cdot\mathbf{n}^n, \tag{93}$$

$$\frac{s_\alpha^{n+1}-s_\alpha^n}{\Delta t}=D_{\Delta\alpha}V^{n+1}-D_{\Delta\alpha}\theta^n U^{n+1}, \tag{94}$$

where $\boldsymbol{\tau}^n=(\cos(\theta^n),\sin(\theta^n))$, $\mathbf{n}^n=(-\sin(\theta^n),\cos(\theta^n))$, $L_{h,n}=L_h(\mathbf{X}^n), L_{h,n}^*=L_h^*(\mathbf{X}^n)$, ∇_h and $D_{\Delta\alpha}$ are discrete derivative operators for the Eulerian grid and the Lagrangian grid respectively, and

$$\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)=S_\mathrm{b}(D_{\Delta\alpha}s_\alpha^{n+1}\boldsymbol{\tau}^n+(s_\alpha^{n+1}-1)D_{\Delta\alpha}\theta^n\mathbf{n}^n). \tag{95}$$

Step 2: Update of θ^{n+1}. After we have obtained $\mathbf{u}^{n+1}$, p^{n+1} and s_α^{n+1}, we update θ at t^{n+1} using the following semi-implicit scheme:

$$\rho\frac{\bar{\mathbf{u}}^{n+1}-\mathbf{u}^n}{\Delta t}=-\nabla_h\bar{p}^{n+1}+\mu\nabla_h^2\bar{\mathbf{u}}^{n+1}+L_{h,n}(\mathbf{F}(s_\alpha^{n+1},\theta^{n+1};\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{96}$$

$$\nabla_h^2\bar{p}^{n+1}=\nabla_h\cdot L_{h,n}(\mathbf{F}(s_\alpha^{n+1},\theta^{n+1};\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{97}$$

$$\overline{V}^{n+1}=L_{h,n}^*(\bar{\mathbf{u}}^{n+1})\cdot\boldsymbol{\tau}^n, \tag{98}$$

$$\overline{U}^{n+1}=L_{h,n}^*(\bar{\mathbf{u}}^{n+1})\cdot\mathbf{n}^n, \tag{99}$$

$$\frac{\theta^{n+1}-\theta^n}{\Delta t}=\frac{1}{s_\alpha^{n+1}}(D_{\Delta\alpha}\overline{U}^{n+1}+D_{\Delta\alpha}\theta^n\overline{V}^{n+1}). \tag{100}$$

where

$$\mathbf{F}(s_\alpha^{n+1},\theta^{n+1};\boldsymbol{\tau}^n,\mathbf{n}^n)=S_\mathrm{b}(D_{\Delta\alpha}s_\alpha^{n+1}\boldsymbol{\tau}^n+(s_\alpha^{n+1}-1)D_{\Delta\alpha}\theta^{n+1}\mathbf{n}^n). \tag{101}$$

It is important to note that the above discretization is not fully implicit. In fact, both the spreading and interpolation operators are evaluated at the interface $\mathbf{X}^n$ from the previous time step. Moreover, when solve the s_α^{n+1} and $\mathbf{u}^{n+1}$, in (90)–(94), we use θ^n instead of θ^{n+1} to evaluate the force density. This makes our semi-implicit discretization linear with respect to the implicit solution variables, $\mathbf{u}^{n+1}$, θ^{n+1}, and s_α^{n+1}. The above semi-implicit discretization essentially decouples the stiffness induced by the elastic force from the fluid equations. This enables us to remove the stiffness of the immersed boundary method effectively by applying the small scale decomposition and arclength/tangent angle formulation as was done in [9].

In the following, we will prove that this semi-implicit discretization is unconditionally stable in the energy norm.

By using a discrete summation by parts, we can show that

$$\langle f,D_{\Delta\alpha}g\rangle_{\Gamma_h}=-\langle D_{\Delta\alpha}f,g\rangle_{\Gamma_h},\quad \langle\mathbf{u},\nabla_h g\rangle_{\Omega_h}=-\langle\nabla_h\cdot\mathbf{u},g\rangle_{\Omega_h}. \tag{102}$$

First, we define the total energy of the physical system. The total energy includes the kinetic energy K and the potential energy P, which are defined below:

$$K=\frac{1}{2}\rho\langle\mathbf{u},\mathbf{u}\rangle_{\Omega_h}=\frac{\rho}{2}\sum_{i,j=1}^{N}\mathbf{u}_{ij}\cdot\mathbf{u}_{ij}h^2, \tag{103}$$

$$P=\frac{1}{2}S_\mathrm{b}\langle s_\alpha-1,s_\alpha-1\rangle_{\Gamma_h}=\frac{S_\mathrm{b}}{2}\sum_{j=1}^{N_\mathrm{b}}(s_{\alpha,j}-1)^2\Delta\alpha. \tag{104}$$

The total energy is then defined as

$$E=K+P. \tag{105}$$

Below we will prove the unconditional stability of our semi-implicit discretization. To simplify the presentation, we still denote the discrete spectral derivative $D_{\Delta\alpha}g$ of a function g as g_α.

Author's personal copy

Taking the discrete inner product defined by (44) of (90) with $\mathbf{u}^{n+1}+\mathbf{u}^n$ and using (102), we obtain

$$\begin{aligned}
2(K^{n+1}-K^n) &= \rho\langle \mathbf{u}^{n+1}+\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} = \rho\langle -\mathbf{u}^{n+1}+\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} + 2\rho\langle \mathbf{u}^{n+1}, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} \\
&= -\rho\langle \mathbf{u}^{n+1}-\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} + 2\Delta t(\langle \mathbf{u}^{n+1}, -\nabla_h p + \mu\nabla_h^2\mathbf{u}^{n+1} + L_{h,n}(\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n))\rangle_{\Omega_h}) \\
&= -\rho\langle \mathbf{u}^{n+1}-\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} - 2\Delta t\langle \mathbf{u}^{n+1}, \nabla_h p\rangle_{\Omega_h} + 2\Delta t\langle \mathbf{u}^{n+1}, \mu\nabla_h^2\mathbf{u}^{n+1}\rangle_{\Omega_h} \\
&\quad + 2\Delta t\langle \mathbf{u}^{n+1}, L_{h,n}(\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n))\rangle_{\Omega_h} \\
&= -\rho\langle \mathbf{u}^{n+1}-\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} - 2\Delta t\langle \nabla_h\cdot\mathbf{u}^{n+1}, p\rangle_{\Omega_h} - 2\mu\Delta t\langle \nabla_h\mathbf{u}^{n+1}, \nabla_h\mathbf{u}^{n+1}\rangle_{\Omega_h} \\
&\quad + 2\Delta t\langle L_{h,n}^*(\mathbf{u}^{n+1}), \mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\rangle_{\Gamma_h}.
\end{aligned} \tag{106}$$

The second term on the right hand side of (106) is zero because the discrete velocity field is divergence free, i.e. $\nabla_h\cdot\mathbf{u}^{n+1}=0$. The fourth term can be rewritten as

$$\begin{aligned}
\langle L_{h,n}^*(\mathbf{u}^{n+1}), \mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\rangle_{\Gamma_h} &= \langle V^{n+1}\boldsymbol{\tau}^n + U^{n+1}\mathbf{n}^n, S_b(s_{\alpha,\alpha}^{n+1}\boldsymbol{\tau}^n + (s_\alpha^{n+1}-1)\theta_\alpha^n\mathbf{n}^n)\rangle_{\Gamma_h} \\
&= S_b(\langle V^{n+1}, s_{\alpha,\alpha}^{n+1}\rangle_{\Gamma_h} + \langle U^{n+1}, (s_\alpha^{n+1}-1)\theta_\alpha^n\rangle_{\Gamma_h}).
\end{aligned} \tag{107}$$

Combining (106) and (107), we can get

$$\begin{aligned}
2(K^{n+1}-K^n) &= -\rho\langle \mathbf{u}^{n+1}-\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} - 2\mu\Delta t\langle \nabla_h\mathbf{u}^{n+1}, \nabla_h\mathbf{u}^{n+1}\rangle_{\Omega_h} + 2S_b\Delta t(\langle V^{n+1}, s_{\alpha,\alpha}^{n+1}\rangle_{\Gamma_h} \\
&\quad + \langle U^{n+1}, (s_\alpha^{n+1}-1)\theta_\alpha^n\rangle_{\Gamma_h}).
\end{aligned} \tag{108}$$

Similarly, by taking the discrete inner product defined by (43) of (93) with $s_\alpha^{n+1}+s_\alpha^n-2$ and using (102), we get

$$\begin{aligned}
2(P^{n+1}-P^n) &= S_b\langle s_\alpha^{n+1}+s_\alpha^n-2, s_\alpha^{n+1}-s_\alpha^n\rangle_{\Gamma_h} \\
&= S_b\langle -s_\alpha^{n+1}+s_\alpha^n, s_\alpha^{n+1}-s_\alpha^n\rangle_{\Gamma_h} + 2S_b\langle s_\alpha^{n+1}-1, s_\alpha^{n+1}-s_\alpha^n\rangle_{\Gamma_h} \\
&= -S_b\langle s_\alpha^{n+1}-s_\alpha^n, s_\alpha^{n+1}-s_\alpha^n\rangle_{\Gamma_h} + 2S_b\Delta t\langle s_\alpha^{n+1}-1, V_\alpha^{n+1}-\theta_\alpha^n U^{n+1}\rangle_{\Gamma_h} \\
&= -S_b\langle s_\alpha^{n+1}-s_\alpha^n, s_\alpha^{n+1}-s_\alpha^n\rangle_{\Gamma_h} + 2S_b\Delta t(-\langle s_{\alpha,\alpha}^{n+1}, V^{n+1}\rangle_{\Gamma_h} - \langle s_\alpha^{n+1}-1, \theta_\alpha^n U^{n+1}\rangle_{\Gamma_h}).
\end{aligned} \tag{109}$$

Adding (108) to (109), we have

$$\begin{aligned}
E^{n+1}-E^n &= -\frac{1}{2}\rho\langle \mathbf{u}^{n+1}-\mathbf{u}^n, \mathbf{u}^{n+1}-\mathbf{u}^n\rangle_{\Omega_h} - \mu\Delta t\langle \nabla_h\mathbf{u}^{n+1}, \nabla_h\mathbf{u}^{n+1}\rangle_{\Omega_h} - \frac{1}{2}S_b\langle s_\alpha^{n+1}-s_\alpha^n, s_\alpha^{n+1}-s_\alpha^n\rangle_{\Gamma_h} \\
&\leqslant 0.
\end{aligned} \tag{110}$$

This proves that our semi-implicit discretization is unconditionally stable in the sense that the total energy is non-increasing.

Remark 1. In our proof presented above, we have used two important properties of our semi-implicit discretization. The first property is that the discrete spreading and interpolation operators are adjoint. The second property is that the velocity field satisfies the discrete divergence free condition. It is clear from the above proof that as long as these two properties are satisfied by our spatial discretization, the corresponding semi-implicit discretization introduced in the previous subsection is unconditionally stable.

Remark 2. We remark that the proof above is similar in spirit to that of a semi-linear discretization obtained by Newren et al. in [24]. There is some minor difference between the unconditionally stable semi-implicit discretization obtained by Newren et al and our unconditionally stable semi-implicit discretization. In the problem considered by Newren et al., the force is a linear function of the interface. On the other hand, in the problem we consider, the force is a nonlinear function of the interface (the rest length of the boundary is not zero). By using the $s_\alpha-\theta$ formulation, the force is a linear function of s_α. By treating θ explicitly, we obtain a semi-implicit discretization that is linear with respect to s_α. Due to the decoupling between s_α and θ, we need to solve two $N_b\times N_b$ linear systems instead of one $2N_b\times 2N_b$ linear system in the semi-implicit discretization obtained by Newren et al.

Author's personal copy

Remark 3. For the steady Stokes flow, we can also prove the following semi-implicit discretization is unconditionally stable:

Step 1:

$$0 = -\nabla_h p^{n+1} + \mu \nabla_h^2 \mathbf{u}^{n+1} + L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{111}$$

$$\nabla_h^2 p^{n+1} = \nabla_h \cdot L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n))), \tag{112}$$

$$V^{n+1} = L_{h,n}^*(\mathbf{u}^{n+1}(\mathbf{x})) \cdot \boldsymbol{\tau}^n, \tag{113}$$

$$U^{n+1} = L_{h,n}^*(\mathbf{u}^{n+1}(\mathbf{x})) \cdot \mathbf{n}^n, \tag{114}$$

$$\frac{s_\alpha^{n+1} - s_\alpha^n}{\Delta t} = D_{\Delta\alpha} V^{n+1} - D_{\Delta\alpha}\theta^n U^{n+1}, \tag{115}$$

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \frac{1}{s_\alpha^{n+1}}\left(D_{\Delta\alpha} U^{n+1} + D_{\Delta\alpha}\theta^{n+1} V^{n+1}\right). \tag{116}$$

Step 2:

$$0 = -\nabla_h \bar{p}^{n+1} + \mu \nabla_h^2 \bar{\mathbf{u}}^{n+1} + L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^{n+1}; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{117}$$

$$\nabla_h^2 \bar{p}^{n+1} = \nabla_h \cdot L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^{n+1}; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{118}$$

$$\overline{V}^{n+1} = L_{h,n}^*(\bar{\mathbf{u}}^{n+1}) \cdot \boldsymbol{\tau}^n, \tag{119}$$

$$\overline{U}^{n+1} = L_{h,n}^*(\bar{\mathbf{u}}^{n+1}) \cdot \mathbf{n}^n, \tag{120}$$

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \frac{1}{s_\alpha^{n+1}}(D_{\Delta\alpha}\overline{U}^{n+1} + D_{\Delta\alpha}\theta^n \overline{V}^{n+1}). \tag{121}$$

In this case the total energy is just the potential energy

$$E = \frac{S_b}{2}\langle s_\alpha - 1, s_\alpha - 1\rangle_{\Gamma_h} = \frac{S_b}{2}\sum_{j=1}^{N_b}(s_{\alpha,j} - 1)^2 \Delta\alpha. \tag{122}$$

As in the case of the unsteady Stokes flow, as long as the velocity field satisfies the discrete divergence free condition and the discrete spreading and interpolation operators are adjoint, we can prove that above semi-implicit discretization is unconditionally stable in the sense of total energy is non-increasing.

6.3. Small scale decomposition

In order to apply the small scale decomposition to our unconditionally stable time discretization, we would like to solve for the velocity field at time t^{n+1} from the space-continuous version of (90) and (91) using an integral representation:

$$\begin{aligned}\mathbf{u}^{n+1}(\mathbf{x}) &= \left(1 - \frac{\mu\Delta t}{\rho}\nabla^2\right)^{-1}\left(\mathbf{u}^n + \frac{\Delta t}{\rho}(1 - \nabla(\nabla^2)^{-1}\nabla\cdot)L_n(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n))\right)\\ &= \left(1 - \frac{\mu\Delta t}{\rho}\nabla^2\right)^{-1}\left(\mathbf{u}^n + \frac{\Delta t}{\rho}L_n(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n))\right) - \frac{\Delta t}{\rho}\left(1 - \frac{\mu\Delta t}{\rho}\nabla^2\right)^{-1}(\nabla^2)^{-1}(\nabla\nabla\\ &\quad \cdot L_n(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n))).\end{aligned}$$

To solve for the velocity field at t^{n+1}, we need to use the following free space fundamental solutions in two space dimensions which are defined as follows:

$$\left(1 - \frac{\mu\Delta t}{\rho}\nabla^2\right)E_1 = \delta(\mathbf{x} - \mathbf{x}'), \tag{123}$$

$$\nabla^2\left(1 - \frac{\mu\Delta t}{\rho}\nabla^2\right)E_2 = \delta(\mathbf{x} - \mathbf{x}'). \tag{124}$$

Author's personal copy

These two fundamental solutions can be expressed in terms of the modified Bessel function of the second kind [1]:

$$E_1 = \frac{\lambda^2}{2\pi} K_0(\lambda|\mathbf{x}-\mathbf{x}'|), \tag{125}$$

$$E_2 = \frac{1}{2\pi}(K_0(\lambda|\mathbf{x}-\mathbf{x}'| + \ln(|\mathbf{x}-\mathbf{x}'|)), \tag{126}$$

where $\lambda^2 = \frac{\rho}{\mu\Delta t}$ and K_0 is a modified Bessel function of the second kind. By integrating by part, we can further express the velocity $\mathbf{u}^{n+1}$ as

$$\begin{aligned}\mathbf{u}^{n+1}(\mathbf{x}) &= \frac{1}{2\pi}\int_\Omega \lambda^2 K_0(\lambda|\mathbf{x}-\mathbf{x}'|)\mathbf{u}^n(\mathbf{x}')\mathrm{d}\mathbf{x}' + \frac{1}{2\pi}\frac{\Delta t}{\rho}\int_{\Gamma^n}\lambda^2 K_0(\lambda|\mathbf{x}-\mathbf{X}^n(\alpha')|)\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\mathrm{d}\alpha' \\ &\quad -\frac{1}{2\pi}\frac{\Delta t}{\rho}\int_\Omega \nabla_{\mathbf{x}}\nabla_{\mathbf{x}}(K_0(\lambda|\mathbf{x}-\mathbf{x}'|)+\ln(|\mathbf{x}-\mathbf{x}'|))\cdot L^n(\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n))\mathrm{d}\mathbf{x}' \\ &= \frac{1}{2\pi}\int_\Omega \lambda^2 K_0(\lambda|\mathbf{x}-\mathbf{x}'|)\mathbf{u}^n(\mathbf{x}')\mathrm{d}\mathbf{x}' + \frac{1}{2\pi}\frac{\Delta t}{\rho}\int_{\Gamma^n}\lambda^2 K_0(\lambda|\mathbf{x}-\mathbf{X}^n(\alpha')|)\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\mathrm{d}\alpha' \\ &\quad -\frac{1}{2\pi}\frac{\Delta t}{\rho}\int_{\Gamma^n} G(\mathbf{x}-\mathbf{X}^n(\alpha'))\cdot\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\mathrm{d}\alpha',\end{aligned} \tag{127}$$

where G is defined as follows:

$$G_{ij}(\mathbf{r}) = \frac{\delta_{ij}}{|\mathbf{r}|^2} - \frac{2r_i r_j}{|\mathbf{r}|^4} + \frac{1}{2}\lambda^2(K_0(\lambda|\mathbf{r}|)+K_2(\lambda|\mathbf{r}|))\frac{r_i r_j}{|\mathbf{r}|^2} - \lambda K_1(\lambda|\mathbf{r}|)\left(\frac{\delta_{ij}}{|\mathbf{r}|}-\frac{r_i r_j}{|\mathbf{r}|^3}\right) \tag{128}$$

and K_0, K_1, K_2 are all modified Bessel functions of the second kind [1].

In this subsection, we will perform a small scale decomposition to the velocity field based on the integral representation (127). Recall that in our semi-implicit discretization, the velocity field at t^{n+1} is evaluated on the boundary $\mathbf{X}^n$ at t^n. Thus we should perform our small scale decomposition for $\mathbf{u}^{n+1}(\mathbf{X}^n)$. To this end, we first write down the integral expression of $\mathbf{u}^{n+1}(\mathbf{X}^n)$ as follows:

$$\begin{aligned}\mathbf{u}^{n+1}(\mathbf{X}^n(\alpha)) &= \frac{1}{2\pi}\int_\Omega \lambda^2 K_0(\lambda|\mathbf{X}^n(\alpha)-\mathbf{x}'|)\mathbf{u}^n(\mathbf{x}')\mathrm{d}\mathbf{x}' + \frac{1}{2\pi}\frac{\Delta t}{\rho}\int_{\Gamma^n}\lambda^2 K_0(\lambda|\mathbf{X}^n(\alpha)-\mathbf{X}^n(\alpha')|)\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\mathrm{d}\alpha' \\ &\quad -\frac{1}{2\pi}\frac{\Delta t}{\rho}\int_{\Gamma^n} G(\mathbf{X}^n(\alpha)-\mathbf{X}^n(\alpha'))\cdot\mathbf{F}(s_\alpha^{n+1},\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)\mathrm{d}\alpha'.\end{aligned} \tag{129}$$

To perform the small scale decomposition to the above velocity integral, we would like to decompose the singular velocity kernel as the sum of a linear singular operator of convolution type and a remainder operator which is regular. Using the Taylor expansion for α' around α, we get the following decomposition:

$$\begin{aligned}V^{n+1}(\alpha) &= \mathbf{u}^{n+1}(\mathbf{X}^n(\alpha))\cdot\boldsymbol{\tau}^n(\alpha) \sim \frac{S_b\Delta t}{2\pi\rho}\int_{\Gamma^n}\lambda^2 K_0(\lambda s_\alpha^n|\alpha-\alpha'|)s_{\alpha,\alpha'}^{n+1}\mathrm{d}\alpha' \\ &\quad -\frac{S_b\Delta t}{2\pi\rho}\int_{\Gamma^n}\left(\frac{1}{2}\lambda^2(K_0(\lambda s_\alpha^n|\alpha-\alpha'|)+K_2(\lambda s_\alpha^n|\alpha-\alpha'|)) - \frac{1}{(s_\alpha^n)^2(\alpha-\alpha')^2}\right)s_{\alpha,\alpha'}^{n+1}\mathrm{d}\alpha',\end{aligned} \tag{130}$$

where $s_\alpha^n \mid$ inside $K_0(\lambda s_\alpha^n|\alpha-\alpha'|)$ is evaluated at α. Notice that [1]

$$\frac{\mathrm{d}^2}{\mathrm{d}\alpha'^2}\left(\frac{1}{(s_\alpha^n(\alpha))^2}K_0(\lambda s_\alpha^n|\alpha-\alpha'|)\right) = \frac{1}{2}\lambda^2\big(K_0(\lambda s_\alpha^n|\alpha-\alpha'|)+K_2(\lambda s_\alpha^n|\alpha-\alpha'|)\big). \tag{131}$$

Author's personal copy

Integrating the right hand side of (130) by parts twice, we get

$$V^{n+1}(\alpha) \sim \frac{1}{2\pi\rho} S_b \Delta t \int_{\Gamma^n} \lambda^2 K_0(\lambda s_\alpha^n |\alpha - \alpha'|) s_{\alpha,\alpha'}^{n+1} d\alpha' - \frac{S_b \Delta t}{2\pi\rho (s_\alpha^n)^2} \int_{\Gamma^n} (K_0(\lambda s_\alpha^n |\alpha - \alpha'|) - \ln(\alpha - \alpha')) s_{\alpha,\alpha'\alpha'\alpha'}^{n+1} d\alpha'. \tag{132}$$

Similarly, we can obtain the leading order contribution of $\overline{U}^{n+1}$ as follows:

$$\overline{U}^{n+1}(\alpha) = \bar{\mathbf{u}}^{n+1}(\mathbf{X}^n(\alpha)) \cdot \mathbf{n}^n(\alpha) \sim \frac{S_b \Delta t}{2\pi\rho (s_\alpha^n)^2} \int_{\Gamma^n} (K_0(\lambda s_\alpha^n |\alpha - \alpha'|) - \ln(\alpha - \alpha'))((s_\alpha^{n+1} - 1)\theta_{\alpha'}^{n+1})_{\alpha'\alpha'} d\alpha'. \tag{133}$$

Using this decomposition, we obtain the following scheme:

$$\frac{s_\alpha^{n+1} - s_\alpha^n}{\Delta t} = T(s_\alpha^{n+1}) + (D_{\Delta\alpha} V^{*,n+1} - D_{\Delta\alpha}\theta^n U^{*,n+1} - T(s_\alpha^n)), \tag{134}$$

$$\rho \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla_h p^{n+1} + \mu \nabla_h^2 \mathbf{u}^{n+1} + L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{135}$$

$$\nabla_h^2 p^{n+1} = \nabla_h \cdot L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{136}$$

$$V^{n+1} = L_{h,n}^*(\mathbf{u}^{n+1}) \cdot \boldsymbol{\tau}^n, \tag{137}$$

$$U^{n+1} = L_{h,n}^*(\mathbf{u}^{n+1}) \cdot \mathbf{n}^n, \tag{138}$$

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \frac{S(\theta^{n+1})}{s_\alpha^{n+1}} + \left(\frac{1}{s_\alpha^{n+1}}(D_{\Delta\alpha} U^{n+1} + D_{\Delta\alpha}\theta^n V^{n+1}) - \frac{S(\theta^n)}{s_\alpha^{n+1}}\right), \tag{139}$$

where

$$T(s_\alpha^{n+1}) = \left(\frac{1}{2\pi\rho} S_b \Delta t \int_{\Gamma^n} \lambda^2 K_0(\lambda s_\alpha^n |\alpha - \alpha'|) s_{\alpha,\alpha'}^{n+1} d\alpha'\right)_\alpha - \left(\frac{S_b \Delta t}{2\pi\rho (s_\alpha^n)^2} \int_{\Gamma^n} (K_0(\lambda s_\alpha^n |\alpha - \alpha'|) - \ln(\alpha - \alpha')) s_{\alpha,\alpha'\alpha'\alpha'}^{n+1} d\alpha'\right)_\alpha,$$

$$S(\theta^{n+1}) = \left(\frac{S_b \Delta t}{2\pi\rho (s_\alpha^n)^2} \int_{\Gamma^n} (K_0(\lambda s_\alpha^n |\alpha - \alpha'|) - \ln(\alpha - \alpha'))((s_\alpha^{n+1} - 1)\theta_{\alpha'})_{\alpha'\alpha'} d\alpha'\right)_\alpha$$

and $\mathbf{u}^{*,n+1}$ is the velocity at t^{n+1} which is calculated explicitly

$$\rho \frac{\mathbf{u}^{*,n+1} - \mathbf{u}^n}{\Delta t} = -\nabla_h p^{*,n+1} + \mu \nabla_h^2 \mathbf{u}^{*,n+1} + L_{h,n}(\mathbf{F}(s_\alpha^n, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{140}$$

$$\nabla_h^2 p^{*,n+1} = \nabla_h \cdot L_{h,n}(\mathbf{F}(s_\alpha^n, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{141}$$

$$V^{*,n+1} = L_{h,n}^*(\mathbf{u}^{*,n+1}) \cdot \boldsymbol{\tau}^n, \tag{142}$$

$$U^{*,n+1} = L_{h,n}^*(\mathbf{u}^{*,n+1}) \cdot \mathbf{n}^n. \tag{143}$$

The derivation of the above semi-implicit scheme is given in Appendix B.

However, the expressions of T and S are still too complicated and need to be further simplified. The leading order linear operator, which contains $K_0(\lambda s_\alpha^n(\alpha)|\alpha - \alpha'|)$, is not a convolution operator. Thus, it does not have a simple kernel under the Fourier transform as the Hilbert operator in the case of the steady Stokes flow. To further simplify the kernel, we approximate $s_\alpha^n(\alpha)$ by $\min_\alpha s_\alpha^n(\alpha)$. With this approximation, the corresponding leading order operator is a convolution operator and can be diagonalized under the Fourier transform. Denote $\beta = \lambda \min_\alpha s_\alpha^n(\alpha)$. In Appendix C, we will show that

$$\mathcal{F}\left(\frac{1}{\pi}\int_{-\infty}^{+\infty} K_0(\beta|\alpha - \alpha'|) f(\alpha') d\alpha'\right) = \frac{\widehat{f}(k)}{\sqrt{\beta^2 + k^2}}. \tag{144}$$

Using (144) and replacing $s_\alpha^n(\alpha)$ by $\min_\alpha s_\alpha^n(\alpha)$, we can simplify the leading order term $T(s_\alpha^{n+1})$ and $S(\theta^{n+1})$ under the Fourier transform:

Author's personal copy

$$\widehat{T}(s_\alpha^{n+1}) \sim -\frac{S_b \Delta t}{2\rho(\min_\alpha s_\alpha^n)^2}\left(\frac{(\lambda \min_\alpha s_\alpha^n)^2 k^2 + k^4}{\sqrt{(\lambda \min_\alpha s_\alpha^n)^2 + k^2}} - |k|^3\right)\hat{s}_\alpha^{n+1}, \tag{145}$$

$$\widehat{S}(\theta^{n+1}) \sim -\frac{S_b \Delta t \max_\alpha(s_\alpha^{n+1} - 1)}{2\rho(\min_\alpha s_\alpha^n)^2}\left(|k|^3 - \frac{k^4}{\sqrt{(\lambda \min_\alpha s_\alpha^n)^2 + k^2}}\right)\hat{\theta}^{n+1}. \tag{146}$$

When $\mu \gg 1$, we have $\lambda = \frac{1}{\sqrt{\mu \Delta t}} \ll 1$. By Taylor expanding (145) and (146) with respect to λ and keeping only the first order term, we obtain the leading order term as follows:

$$\widehat{T}(s_\alpha^{n+1}) \sim -\frac{S_b}{4\mu}|k|\hat{s}_\alpha^{n+1}, \tag{147}$$

$$\widehat{S}(\theta^{n+1}) \sim -\frac{S_b}{4\mu}\max_\alpha(s_\alpha^{n+1} - 1)|k|\hat{\theta}^{n+1}, \tag{148}$$

which is the same as the steady Stoke flow. This is also consistent with one's physical intuition. When the viscosity is very large, the flow changes very slowly. The inertial term can be neglected.

When $\mu \ll 1$, then $\lambda = \frac{1}{\sqrt{\mu \Delta t}} \gg 1$, the asymptotic expansion is

$$\widehat{T}(s_\alpha^{n+1}) \sim -\frac{S_b \sqrt{\Delta t}}{2(\min_\alpha s_\alpha^n)\sqrt{\rho\mu}}k^2 \hat{s}_\alpha^{n+1}, \tag{149}$$

$$\widehat{S}(\theta^{n+1}) \sim -\frac{S_b \Delta t \max_\alpha(s_\alpha^{n+1} - 1)}{2\rho(\min_\alpha s_\alpha^n)^2}k^3 \hat{\theta}^{n+1}. \tag{150}$$

From the asymptotic expansion above, we can see that our small scale decomposition is also consistent with the linearized stability analysis which Stockie and Wetton got in [30]. Using the leading order term above, we can get the leading order term of the eigenvalue same with the result in [30].

We can also obtain the corresponding stability constraint for the explicit scheme near the equilibrium:

$$\Delta t < C(S_b, \mu)h^\beta, \tag{151}$$

where $1 \leqslant \beta \leqslant 3/2$. The value of β depends on μ. If $\mu \ll 1$, then we have $\beta \approx 3/2$. On the other hand, if $\mu \gg 1$, we have $\beta \approx 1$.

6.4. The numerical scheme

Based on the small scale decomposition we developed in the last subsection, we can now describe our semi-implicit numerical scheme. Combining the time discretization (90)–(100) with the decomposition (130)–(133) and using the approximation (145) and (146), we obtain the following semi-implicit numerical scheme:

Step 1: Update of $\mathbf{u}^{n+1}$ and s_α^{n+1}.

$$\frac{s_\alpha^{n+1} - s_\alpha^n}{\Delta t} = T(s_\alpha^{n+1}) + (D_{\Delta\alpha}V^{*,n+1} - D_{\Delta\alpha}\theta^n U^{*,n+1} - T(s_\alpha^n)), \tag{152}$$

$$\rho\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla_h p^{n+1} + \mu\nabla_h^2 \mathbf{u}^{n+1} + L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{153}$$

$$\nabla_h^2 p^{n+1} = \nabla_h \cdot L_{h,n}(\mathbf{F}(s_\alpha^{n+1}, \theta^n; \boldsymbol{\tau}^n, \mathbf{n}^n)), \tag{154}$$

where

$$\widehat{T}(s_\alpha^{n+1}) = -\frac{S_b \Delta t}{2\rho(\min_\alpha s_\alpha^n)^2}\left(\frac{\left(\lambda \min_\alpha s_\alpha^n\right)^2 k^2 + k^4}{\sqrt{(\lambda \min_\alpha s_\alpha^n)^2 + k^2}} - |k|^3\right)\hat{s}_\alpha^{n+1} \tag{155}$$

Author's personal copy

and $\mathbf{u}^{*,n+1}$ is the intermediate velocity at t^{n+1} which is calculated by solving the unsteady Stokes equations implicitly while evaluating the elastic force explicitly:

$$\rho\frac{\mathbf{u}^{*,n+1}-\mathbf{u}^n}{\Delta t}=-\nabla_h p^{*,n+1}+\mu\nabla_h^2\mathbf{u}^{*,n+1}+L_{h,n}(\mathbf{F}(s_\alpha^n,\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{156}$$

$$\nabla_h^2 p^{*,n+1}=\nabla_h\cdot L_{h,n}(\mathbf{F}(s_\alpha^n,\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{157}$$

$$V^{*,n+1}=L_{h,n}^*(\mathbf{u}^{*,n+1})\cdot\boldsymbol{\tau}^n, \tag{158}$$

$$U^{*,n+1}=L_{h,n}^*(\mathbf{u}^{*,n+1})\cdot\mathbf{n}^n. \tag{159}$$

Step 2:Update of θ^{n+1}. Once we have updated $\mathbf{u}$, p, and s_α at t^{n+1}, we update θ^{n+1} using the following semi-implicit scheme:

$$\frac{\theta^{n+1}-\theta^n}{\Delta t}=\frac{S(\theta^{n+1})}{\min\limits_\alpha s_\alpha^{n+1}}+\left(\frac{1}{s_\alpha^{n+1}}(D_{\Delta\alpha}U^{n+1}+D_{\Delta\alpha}\theta^n V^{n+1})-\frac{S(\theta^n)}{\min_\alpha s_\alpha^{n+1}}\right), \tag{160}$$

where

$$V^{n+1}=L_{h,n}^*(\mathbf{u}^{n+1})\cdot\boldsymbol{\tau}^n, \tag{161}$$

$$U^{n+1}=L_{h,n}^*(\mathbf{u}^{n+1})\cdot\mathbf{n}^n, \tag{162}$$

$$\widehat{S}(\theta^{n+1})=-\frac{S_b\Delta t\max\limits_\alpha(s_\alpha^n-1)}{2\rho(\min\limits_\alpha s_\alpha^n)^2}\left(|k|^3-\frac{k^4}{\sqrt{(\lambda\min\limits_\alpha s_\alpha^n)^2+k^2}}\right)\hat{\theta}^{n+1}. \tag{163}$$

This is our semi-implicit scheme for the unsteady Stokes flow. The spectral discretization in space has the advantage of being high order accurate and the leading order operator has a simple kernel under the Fourier transform. As it is, the time discretization is only first order. Based on the first order semi-implicit scheme that we develop in this subsection, we will develop a second order semi-implicit scheme in the next subsection.

A near equilibrium stability analysis shows that the stability constraint of this semi-implicit scheme is of the form $\Delta t < C(S_b,\mu)$, which is independent of the wave number, but still dependent on S_b and μ. This is due to the fact that the Small Scale Decomposition does not capture the low frequency components of the solution accurately. The low frequency components of the solution can affect the stability of the time discretization in two ways. The first one is through the small scale decomposition, which only captures the leading order contribution of the solution at high wave numbers. The second one comes from the second term of the right hand side of the dynamic equations for s_α and θ. As in the case of the steady Stokes flow, we can include the leading order contribution from the second term in our leading order term and treat them implicitly. This treatment would significantly improve the stability property especially when the elastic coefficient is large or the viscosity is small. This improved stability is at the expense of solving a linear system for the implicit solution at each time step. We call this semi-implicit discretization as the semi-implicit method of the second kind. More discussions on the semi-implicit method of the second kind can be found in Appendix A.

Remark 4. The leading order term we derive above is calculated analytically using the space-continuous formulation with an unsmoothed Dirac delta function. As Stockie and Wetton pointed out in [32], this analysis over-predicts the stiffness of the Immersed Boundary method in a practical computation. If we use the leading order approximation directly, the semi-implicit scheme with the leading order terms derived above tends to over-dissipate the solution. To alleviate this effect in the practical implementation, we rescale the leading order term by a coefficient which is calculated at the first time step in the following way:

$$C_V=\frac{\max_\alpha V_\alpha^{1,*}}{\max_\alpha T(s_\alpha^0)},$$

$$C_U=\frac{\max_\alpha U^1}{\max_\alpha S_U(\theta^0)},$$

where $S_U(\theta^0)$ is the leading order term of $\overline{U}^1$, which can be computed from $S(\theta^0)$ via the Fourier transform. The leading order term we use in a practical computation is actually $C_V T(s_\alpha^{n+1})$ and $C_U S(\theta^{n+1})$.

Author's personal copy

6.5. A second order semi-implicit scheme

Based on the first order semi-implicit scheme we have developed in the previous subsection, we will derive the corresponding second order semi-implicit scheme in this subsection.

First, we need to use a robust implicit second order temporal discretization. To simplify the presentation, we will only describe the semi-discrete algorithm. The space discretization is done in the same way as before. The second order temporal discretization we use consists of two steps. In the first step, we take a fractional time step from t^n to $t^{n+\frac{1}{2}}$. It is same with the first order semi-implicit discretization (90)–(100), except the time-step is $\frac{\Delta t}{2}$.

In the second step, we integrate the unsteady Stokes equations from t^n to t^{n+1} based on the midpoint and the trapezoidal rules:

Step 1: Update of $\mathbf{u}^{n+1}$ and s_α^{n+1}.

$$\rho\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} = -\nabla\bar{p} + \mu\nabla^2\bar{\mathbf{u}} + L_{n+\frac{1}{2}}\left(\mathbf{F}\left(\bar{s}_\alpha, \theta^{n+\frac{1}{2}}; \boldsymbol{\tau}^{n+\frac{1}{2}}, \mathbf{n}^{n+\frac{1}{2}}\right)\right), \tag{164}$$

$$\nabla^2\bar{p} = \nabla\cdot L_{n+\frac{1}{2}}\left(\mathbf{F}\left(\bar{s}_\alpha, \theta^{n+\frac{1}{2}}; \boldsymbol{\tau}^{n+\frac{1}{2}}, \mathbf{n}^{n+\frac{1}{2}}\right)\right), \tag{165}$$

$$\overline{V} = L^*_{n+\frac{1}{2}}(\bar{\mathbf{u}})\cdot\boldsymbol{\tau}^{n+\frac{1}{2}}, \tag{166}$$

$$\overline{U} = L^*_{n+\frac{1}{2}}(\bar{\mathbf{u}})\cdot\mathbf{n}^{n+\frac{1}{2}}, \tag{167}$$

$$\frac{s_\alpha^{n+1}-s_\alpha^n}{\Delta t/2} = \overline{V}_\alpha - \theta_\alpha^{n+\frac{1}{2}}\overline{U}, \tag{168}$$

where $\boldsymbol{\tau}^{n+\frac{1}{2}} = (\cos(\theta^{n+\frac{1}{2}}), \sin(\theta^{n+\frac{1}{2}}))$, $\mathbf{n}^{n+\frac{1}{2}} = (-\sin(\theta^{n+\frac{1}{2}}), \cos(\theta^{n+\frac{1}{2}}))$, $L_{n+\frac{1}{2}} = L(\mathbf{X}^n + \frac{1}{2}), L^*_{n+\frac{1}{2}} = L^*(\mathbf{X}^n + \frac{1}{2})$ and

$$\mathbf{F}(\bar{s}_\alpha, \theta^{n+\frac{1}{2}}; \boldsymbol{\tau}^{n+\frac{1}{2}}, \mathbf{n}^{n+\frac{1}{2}}) = S_b(D_{\Delta\alpha}\bar{s}_\alpha\boldsymbol{\tau}^{n+\frac{1}{2}} + (\bar{s}_\alpha - 1)D_{\Delta\alpha}\theta^{n+\frac{1}{2}}\mathbf{n}^{n+\frac{1}{2}}). \tag{169}$$

Step 2: Update of θ^{n+1}. After we have obtained $\mathbf{u}^{n+1}$ and s_α^{n+1}, we update θ at t^{n+1} using the following semi-implicit scheme:

$$\rho\frac{\widetilde{\mathbf{u}}^{n+1}-\mathbf{u}^n}{\Delta t} = -\nabla\tilde{p}^{n+1} + \mu\nabla^2\left(\frac{\widetilde{\mathbf{u}}^{n+1}+\mathbf{u}^{n+1}}{2}\right) + L_{n+\frac{1}{2}}(\mathbf{F}(\bar{s}_\alpha, \bar{\theta}; \boldsymbol{\tau}^{n+\frac{1}{2}}, \mathbf{n}^{n+\frac{1}{2}})), \tag{170}$$

$$\nabla^2\tilde{p}^{n+1} = \nabla\cdot L_{n+\frac{1}{2}}(\mathbf{F}(\bar{s}_\alpha, \bar{\theta}; \boldsymbol{\tau}^{n+\frac{1}{2}}, \mathbf{n}^{n+\frac{1}{2}})), \tag{171}$$

$$\widetilde{V} = L^*_{n+\frac{1}{2}}\left(\frac{\widetilde{\mathbf{u}}^{n+1}+\mathbf{u}^{n+1}}{2}\right)\cdot\boldsymbol{\tau}^{n+\frac{1}{2}}, \tag{172}$$

$$\widetilde{U} = L^*_{n+\frac{1}{2}}\left(\frac{\widetilde{\mathbf{u}}^{n+1}+\mathbf{u}^{n+1}}{2}\right)\cdot\mathbf{n}^{n+\frac{1}{2}}, \tag{173}$$

$$\frac{\theta^{n+1}-\theta^n}{\Delta t} = \frac{1}{s_\alpha^{n+\frac{1}{2}}}(\widetilde{U}_\alpha + \theta_\alpha^{n+\frac{1}{2}}\widetilde{V}), \tag{174}$$

where

$$\mathbf{F}(\bar{s}_\alpha, \bar{\theta}; \boldsymbol{\tau}^{n+\frac{1}{2}}, \mathbf{n}^{n+\frac{1}{2}}) = S_b(\bar{s}_{\alpha,\alpha}\boldsymbol{\tau}^{n+\frac{1}{2}} + (\bar{s}_\alpha - 1)\bar{\theta}_\alpha\mathbf{n}^{n+\frac{1}{2}}) \tag{175}$$

and

$$\bar{\mathbf{u}} = \frac{\mathbf{u}^{n+1}+\mathbf{u}^n}{2}, \quad \bar{s}_\alpha = \frac{s_\alpha^{n+1}+s_\alpha^n}{2}, \quad \bar{\theta} = \frac{\theta^{n+1}+\theta^n}{2}. \tag{176}$$

Here, $L_{n+\frac{1}{2}}$ and $L^*_{n+\frac{1}{2}}$ are the spreading and the interpolation operators evaluated at $\mathbf{X}^{n+\frac{1}{2}}$. Using the same method of analysis, we can prove that the above second order semi-implicit discretization is unconditionally stable in the sense that the total energy is non-increasing.

The first step in our second order method is identical to the first order method except that the time step is $\frac{\Delta t}{2}$ instead of Δt. Thus we can use the first order semi-implicit scheme introduced in last subsection to compute it directly.

Author's personal copy

In the second step, we can also apply the Small Scale Decomposition with some modifications. After applying the Small Scale Decomposition to the second step of the two-step method, the second step of the semi-implicit scheme has the form

Step 1: Update $\mathbf{u}^{n+1}$ and s_α^{n+1}.

$$\frac{s_\alpha^{n+1}-s_\alpha^n}{\Delta t}=T\left(\frac{s_\alpha^{n+1}+s_\alpha^n}{2}\right)+\left(\overline{V}_\alpha^*-\theta_\alpha^{n+\frac{1}{2}}\overline{U}^*-T\left(s_\alpha^{n+\frac{1}{2}}\right)\right), \tag{177}$$

$$\rho\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}=-\nabla\bar{p}+\mu\nabla^2\bar{\mathbf{u}}+L_{n+\frac{1}{2}}\left(\mathbf{F}\left(\bar{s}_\alpha,\theta^{n+\frac{1}{2}};\boldsymbol{\tau}^{n+\frac{1}{2}},\mathbf{n}^{n+\frac{1}{2}}\right)\right), \tag{178}$$

$$\nabla^2\bar{p}=\nabla\cdot L_{n+\frac{1}{2}}\left(\mathbf{F}\left(\bar{s}_\alpha,\theta^{n+\frac{1}{2}};\boldsymbol{\tau}^{n+\frac{1}{2}},\mathbf{n}^{n+\frac{1}{2}}\right)\right), \tag{179}$$

The leading order terms, T is given by

$$\widehat{T}(\bar{s}_\alpha)=-\frac{S_b\Delta t}{4\rho\left(\min\limits_\alpha s_\alpha^{n+\frac{1}{2}}\right)^2}\left(\frac{\left(\bar{\lambda}\min\limits_\alpha s_\alpha^{n+\frac{1}{2}}\right)^2k^2+k^4}{\sqrt{\left(\bar{\lambda}\min\limits_\alpha s_\alpha^{n+\frac{1}{2}}\right)^2+k^2}}-|k|^3\right)\widehat{\bar{s}}_\alpha, \tag{180}$$

where $\bar{\lambda}^2=\frac{2\rho}{\mu\Delta t}$, and $\mathbf{u}^{*,n+1}$ is the intermediate velocity at t^{n+1} which is obtained by solving the unsteady Stokes equations implicitly but with the forcing evaluated explicitly:

$$\rho\frac{\mathbf{u}^{*,n+1}-\mathbf{u}^n}{\Delta t}=-\nabla\bar{p}^*+\mu\nabla^2\bar{\mathbf{u}}^*+L_{n+\frac{1}{2}}\left(\mathbf{F}\left(s_\alpha^{n+\frac{1}{2}},\theta^{n+\frac{1}{2}};\boldsymbol{\tau}^{n+\frac{1}{2}},\mathbf{n}^{n+\frac{1}{2}}\right)\right), \tag{181}$$

$$\nabla^2\bar{p}^*=\nabla\cdot L_{n+\frac{1}{2}}\left(\mathbf{F}\left(s_\alpha^{n+\frac{1}{2}},\theta^{n+\frac{1}{2}};\boldsymbol{\tau}^{n+\frac{1}{2}},\mathbf{n}^{n+\frac{1}{2}}\right)\right), \tag{182}$$

$$\overline{V}^*=L^*_{n+\frac{1}{2}}(\bar{\mathbf{u}}^*)\cdot\boldsymbol{\tau}^{n+\frac{1}{2}}, \tag{183}$$

$$\overline{U}^*=L^*_{n+\frac{1}{2}}(\bar{\mathbf{u}}^*)\cdot\mathbf{n}^{n+\frac{1}{2}} \tag{184}$$

and

$$\bar{\mathbf{u}}^*=\frac{\mathbf{u}^{*,n+1}+\mathbf{u}^n}{2}. \tag{185}$$

Step 2:After the $\mathbf{u}^{n+1}$ and s_α^{n+1} are calculated in step 1, we update θ at t^{n+1} using the following semi-implicit scheme:

$$\frac{\theta^{n+1}-\theta^n}{\Delta t}=S\left(\frac{\theta^{n+1}+\theta^n}{2}\right)+\frac{1}{s_\alpha^{n+\frac{1}{2}}}\left(\overline{U}_\alpha+\bar{\theta}_\alpha\overline{V}\right)-S\left(\theta^{n+\frac{1}{2}}\right), \tag{186}$$

where

$$\overline{V}=L^*_{n+\frac{1}{2}}\left(\frac{\mathbf{u}^{n+1}+\mathbf{u}^n}{2}\right)\cdot\boldsymbol{\tau}^{n+\frac{1}{2}}, \tag{187}$$

$$\overline{U}=L^*_{n+\frac{1}{2}}\left(\frac{\mathbf{u}^{n+1}+\mathbf{u}^n}{2}\right)\cdot\mathbf{n}^{n+\frac{1}{2}}. \tag{188}$$

and the leading order term is

$$\widehat{S}(\bar{\theta})=-\frac{S_b\Delta t\max\limits_\alpha\left(s_\alpha^{n+\frac{1}{2}}-1\right)}{4\rho\left(\min\limits_\alpha s_\alpha^{n+\frac{1}{2}}\right)^3}\left(|k|^3-\frac{k^4}{\sqrt{\left(\bar{\lambda}\min\limits_\alpha s_\alpha^{n+\frac{1}{2}}\right)^2+k^2}}\right)\widehat{\bar{\theta}}. \tag{189}$$

This completely defines our second order semi-implicit scheme.

Author's personal copy

7. Numerical results

In this section, we will perform a number of numerical experiments to test the stability of our semi-implicit schemes for both the steady and unsteady Stokes equations. We also compare the performance of our semi-implicit schemes with the explicit scheme and the fully implicit scheme. Our numerical results indicate convincingly that our semi-implicit schemes has a much better stability property that that of the explicit scheme. Moreover, the computational cost of our semi-implicit schemes is comparable to that of an explicit scheme. Thus our semi-implicit schemes offer significant computational saving over the explicit scheme, especially when the number of grid points is large.

7.1. Model problem

The test problem we use is one typically seen in the literature, in which the immersed boundary is a closed loop initially in the shape of an ellipse. We choose an ellipse initially aligned in the coordinate directions with horizontal semi-axis $a = 0.32$ and vertical semi-axis $b = 0.24$. The boundary can be parameterized as follows:

$$\begin{cases} x(\alpha, 0) = 0.5 + 0.32\cos\alpha, \\ y(\alpha, 0) = 0.5 + 0.24\sin\alpha. \end{cases} \tag{190}$$

The fluid is initially at rest in a periodic domain $\Omega = [0, 1] \times [0, 1]$. We use a periodic boundary condition for the fluid flow. For the initial condition defined above, the rest state of the boundary is a circle with radius $r = 0.2$. For the unsteady Stokes flow, the immersed boundary with the above initial condition evolves as damped oscillations around a circular equilibrium state. The area is conserved during the time evolution since the flow is incompressible. For the steady Stokes flow, the boundary converges to the circular state without oscillations.

We use a uniform $N \times N$ grid to discretize the fluid domain, Ω. We choose $N_b = 2N$ number of grid points to discretize the immersed boundary so that there are approximately two immersed boundary points per mesh width. We use the spectral method to discretize the spatial derivatives both in the fluid domain and along the immersed boundary. The leading order singular integral is also discretized by the spectral method.

7.2. Steady Stokes flow

First, in order to reduce the number of parameters in our test problem, we write the equations in terms of the following dimensionless variables to get the non-dimensional model [33],

$$t' = \frac{t}{t_0}, \quad \mathbf{x}' = \frac{\mathbf{x}}{L}, \quad \mathbf{u}' = \frac{\mathbf{u}t_0}{L}, \quad p' = \frac{pt_0}{\mu}, \quad \mathbf{f}' = \frac{\mathbf{f}Lt_0}{\mu},$$

where L is the size of computational domain, t_0 is characteristic time.

Using these new variables, we have

$$0 = -\nabla p' + \triangle \mathbf{u}' + \mathbf{f}'(\mathbf{x}', t'), \tag{191}$$

$$0 = \nabla \cdot \mathbf{u}'. \tag{192}$$

For the equations of the elastic boundary, the dimensionless variables are

$$\mathbf{X}' = \frac{\mathbf{X}}{L}, \quad s'_\alpha = \frac{s_\alpha}{L}, \quad \theta' = \theta, \quad \alpha' = \frac{\alpha}{L}, \quad T' = \frac{T}{S_b}, \quad \mathbf{F}' = \frac{\mathbf{F}L}{S_b}, \quad \boldsymbol{\tau}' = \boldsymbol{\tau}, \quad \mathbf{n}' = \mathbf{n}.$$

Then the equations describe the interaction of the boundary and the fluid become

$$U' = \mathbf{u}'(\mathbf{X}'(\alpha', t'), t') \cdot \mathbf{n}', \tag{193}$$

$$V' = \mathbf{u}'(\mathbf{X}'(\alpha', t)', t') \cdot \tau', \tag{194}$$

$$s'_{\alpha,t'} = V'_{\alpha'} - \theta'_{\alpha'} U', \tag{195}$$

$$\theta'_{t'} = \frac{1}{s'_\alpha}(U'_{\alpha'} + V'\theta'_{\alpha'}), \tag{196}$$

Author's personal copy

where

$$\mathbf{f}'(\mathbf{x}',t') = \frac{S_b t_0}{\mu L}\int_0^{L_b/L} \mathbf{F}'(\alpha',t)'\delta(\mathbf{x}'-\mathbf{X}'(\alpha',t'))\mathrm{d}\alpha', \tag{197}$$

$$\mathbf{u}'(\mathbf{X}'(\alpha',t'),t') = \int_\Omega \mathbf{u}'(\mathbf{x}',t')\delta(\mathbf{x}'-\mathbf{X}'(\alpha',t'))\mathrm{d}\mathbf{x}'. \tag{198}$$

There are two non-dimensional parameters in this problem:

$$\frac{S_b t_0}{\mu L}, \quad \frac{L_b}{L}.$$

If we let $t_0 = \frac{\mu L}{S_b}$, then the only parameter in this dimensionless model is L_b/L which is fixed in our test problem. So we can always fix $S_b = \mu = 1$ in our numerical study.

The stability analysis in the steady Stokes flow suggests us to use the total energy as a criterion to test the stability of different numerical methods. For the steady Stokes equations, the total energy is equal to the potential energy. In Fig. 1, we show that the energy for four different numerical methods: the explicit scheme, the semi-implicit scheme of first kind, the 4th order semi-implicit scheme using the integral factor method and the unconditionally stable semi-implicit scheme. In this figure and the subsequent figures, we use the the legend "semi-implicit" to denote the semi-implicit scheme of first kind, and the legend "integral factor" to denote the semi-implicit scheme based on the integral factor method. We use two different time steps, 0.1 and 1, respectively. When $\Delta t = 0.1$, all the four methods are stable. They give almost identical results. When $\Delta t = 1$, the explicit scheme becomes unstable, but all the semi-implicit schemes are stable. In fact, all the semi-implicit schemes remain stable with much larger time steps. In Fig. 2, we plot the energy of the system for semi-implicit schemes of first kind and the semi-implicit scheme based on the integral factor method with $\Delta t = 10$. Fig. 3 shows the configuration obtained by the two semi-implicit schemes at the final time with $\Delta t = 10$. They both remain as a circle, but lose some area compared with the original state.

Next, we compare the performance of our semi-implicit schemes with the explicit and fully implicit schemes. The fully implicit scheme we use here was originally proposed by Tu and Peskin in [33]. In order to make a fair comparison, we run the implicit schemes (semi-implicit and fully implicit) with a time step small enough to make sure that the computational results have a reasonable accuracy. We take $\Delta t = 4$ for the fully implicit and the semi-implicit schemes. With this time step, the area loss is less than 5%. For the explicit scheme, we take $\Delta t = 1/4, 1/8, 1/16, 1/32$ which corresponds to $N = 64, 128, 256, 512$ respectively. These time steps are the largest possible to keep the stability of the explicit scheme. The time we compute is $T = 20$. The result

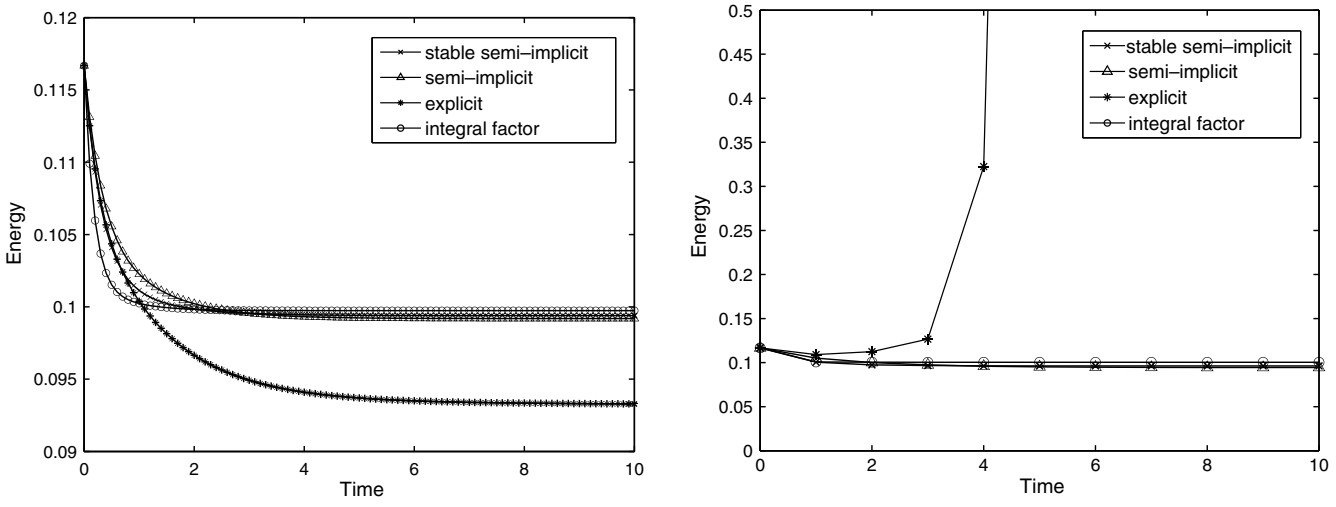


Fig. 1. Energy of the system for four different schemes. $N = 128$, $S_b = 1$, $\mu = 1$. Left one: $\Delta t = 0.1$; Right: $\Delta t = 1$. Here the legend "stable semi-implicit" stands for the unconditionally stable semi-implicit scheme, "semi-implicit" for the semi-implicit scheme of first kind, and "integral factor" for the semi-implicit scheme based on the integral factor method.

Author's personal copy

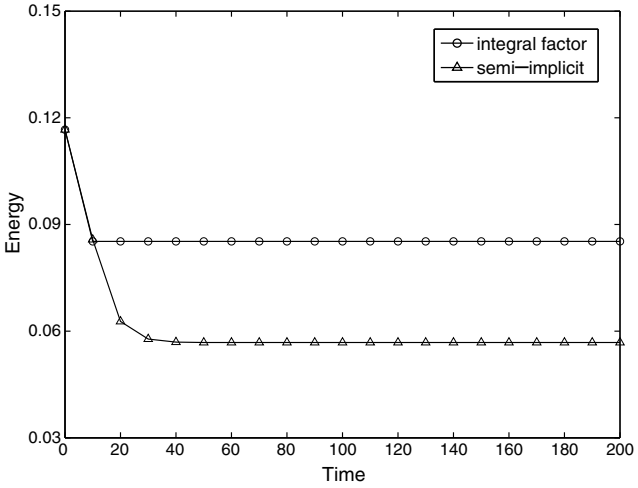


Fig. 2. Energy for the semi-implicit scheme of first kind (labeled as "semi-implicit") and the semi-implicit scheme based on the integral factor method (labeled as "integral factor"). $\Delta t = 10$, $N = 128$, $S_b = 1$, $\mu = 1$.

is shown in Table 1. From this comparison, we can see that the performance of our semi-implicit schemes is much better than the explicit, the fully implicit scheme, and the unconditionally stable semi-implicit scheme in all cases. As we can see, the larger the number of the spatial grid points is, the more computational saving we would get using our semi-implicit schemes. Even for a modest grid size, our semi-implicit schemes still give a significant computational saving compared with the explicit or the fully implicit scheme. It is interesting to note that although the computational cost of the unconditionally stable semi-implicit scheme (labeled as s,s,i) is faster than the fully implicit method, the computational cost of the unconditionally stable semi-implicit method is still more expensive than the explicit scheme. This makes the unconditionally stable semi-implicit scheme not very practical.

7.3. Unsteady Stokes flow

We can also get the non-dimensional model for unsteady stokes flow. Similar as the steady stokes case, we define the following dimensionless variables:

$$t' = \frac{t}{t_0}, \quad \mathbf{x}' = \frac{\mathbf{x}}{L}, \quad \mathbf{u}' = \frac{\mathbf{u}t_0}{L}, \quad p' = \frac{pt_0}{\mu}, \quad \mathbf{f}' = \frac{\mathbf{f}Lt_0}{\mu},$$

where L is the size of computational domain, t_0 is characteristic time. Using these new variables, we have

$$\frac{\partial \mathbf{u}'}{\partial t'} = \frac{\mu t_0}{\rho L^2}(-\nabla p' + \triangle \mathbf{u}' + \mathbf{f}'(\mathbf{x}', t')), \tag{199}$$

$$0 = \nabla \cdot \mathbf{u}'. \tag{200}$$

For the equations of the elastic boundary, the dimensionless variables are

$$\mathbf{X}' = \frac{\mathbf{X}}{L}, \quad s'_\alpha = \frac{s_\alpha}{L}, \quad \theta' = \theta, \quad \alpha' = \frac{\alpha}{L}, \quad T' = \frac{T}{S_b}, \quad \mathbf{F}' = \frac{\mathbf{F}L}{S_b}, \quad \boldsymbol{\tau}' = \boldsymbol{\tau}, \quad \mathbf{n}' = \mathbf{n}.$$

Then the equations describe the interaction of the boundary and the fluid become

$$U' = \mathbf{u}'(\mathbf{X}'(\alpha', t'), t') \cdot \mathbf{n}', \tag{201}$$

$$V' = \mathbf{u}'(\mathbf{X}'(\alpha', t'), t') \cdot \tau', \tag{202}$$

$$s'_{\alpha,t'} = V'_{\alpha'} - \theta'_{\alpha'} U', \tag{203}$$

$$\theta'_{t'} = \frac{1}{s'_\alpha}(U'_{\alpha'} + V'\theta'_{\alpha'}), \tag{204}$$

Author's personal copy

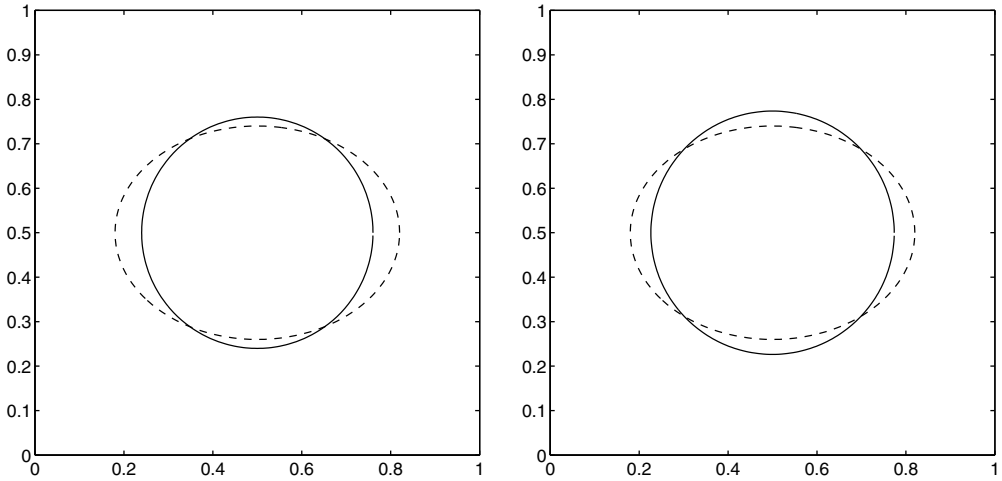


Fig. 3. Dashed line: the initial boundary configuration; Solid line: the boundary configuration after 20 time steps with $\Delta t = 10$, $N = 128$, $S_b = 1$, $\mu = 1$. Left: semi-implicit ; Right: integral factor method.

Table 1
Execution time for each computation in seconds

N	exp	s,i,1	s,i,2	s,s,i	f,i
64	1	0.4	2	7	9
128	5	0.7	3	30	39
256	30	1.3	7	139	206
512	344	4.2	19	611	1200

The legends are defined as follows: "exp" stands for the explicit scheme, "s,i,1" the semi-implicit method of first kind, "s,i,2" the semi-implicit scheme of the second kind, "s,s,i" the unconditionally stable semi-implicit method, and "f,i" the fully implicit scheme. N is the number of grid points along each dimension.

where

$$\mathbf{f}'(\mathbf{x}', t') = \frac{S_b t_0}{\mu L} \int_0^{L_b/L} \mathbf{F}'(\alpha', t')\delta(\mathbf{x}' - \mathbf{X}'(\alpha', t'))\mathrm{d}\alpha', \tag{205}$$

$$\mathbf{u}'(\mathbf{X}'(\alpha', t'), t') = \int_\Omega \mathbf{u}'(\mathbf{x}', t')\delta(\mathbf{x}' - \mathbf{X}'(\alpha', t'))\mathrm{d}\mathbf{x}'. \tag{206}$$

From the non-dimensional analysis, we can see that there are three non-dimensional parameters in this problem:

$$\frac{S_b t_0}{\mu L}, \quad \frac{\mu t_0}{\rho L^2}, \quad \frac{L_b}{L}.$$

If we let $t_0 = \frac{\mu L}{S_b}$, then the parameters left in this dimensionless model is $\frac{\mu t_0}{\rho L^2} = \frac{\mu^2}{\rho L S_b}$ and $\frac{L_b}{L}$. L_b and L are fixed in our test problem only depends on the initial condition. So $\frac{\mu^2}{\rho L S_b}$ is the only parameter in our test model. For this reason, we always fix the elastic coefficient S_b to 1, but vary μ.

In our computations, we use the following parameter values:

$$\rho = 1, \quad S_b = 1, \quad \mu = 0.1, 0.01, 0.005.$$

We vary the number of the spatial grid points along each dimension in the following fashion:

$$N = 64, 128, 256, 512.$$

The criterion that we use to check whether one scheme is stable or not is that the total energy of the system is non-increasing and the boundary configuration lies within the computational domain.

Author's personal copy

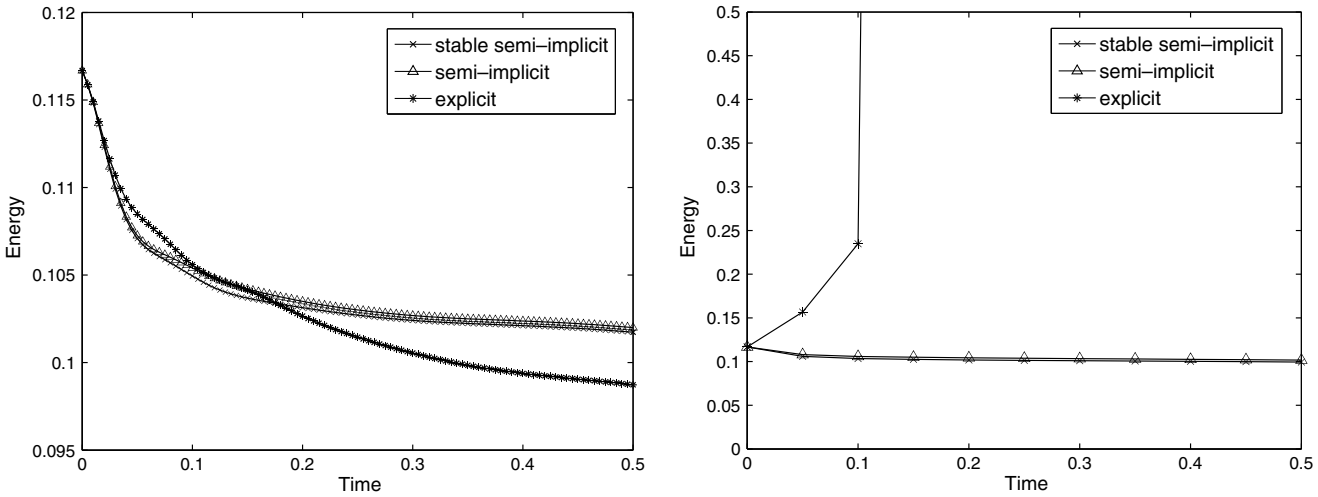


Fig. 4. Total energy of the unsteady Stokes system for different schemes with two different timesteps. $N = 128$, $S_b = 1$, $\mu = 0.01$. Left: $\Delta t = 0.005$; Right: $\Delta t = 0.05$. The legend "semi-implicit" stands for the solution obtained by the semi-implicit scheme of first kind, "stable semi-implicit" the unconditionally stable semi-implicit method.

Next, we perform some numerical experiments to test the stability of our semi-implicit schemes for the unsteady Stokes flow. Fig. 4 shows that the energy obtained by the explicit scheme and the semi-implicit scheme of the first kind. We take two different timesteps, 0.005 and 0.05. With $\Delta t = 0.005$, the explicit and semi-implicit schemes are all stable, and they give nearly identical results. With $\Delta t = 0.05$, the explicit scheme becomes unstable, but the semi-implicit schemes remain stable. Even if we increase the timestep to $\Delta t = 1$, the semi-implicit methods are still stable, as we can see from Fig. 5. For the semi-implicit scheme of the first kind, we have used the small scale decomposition and further simplification of the singular integral kernel. Therefore, the total energy in our semi-implicit scheme is not guaranteed to decrease monotonically in time. Nonetheless, we observe that the total energy still decreases in time as is the case for the unconditionally stable semi-implicit scheme. In Fig. 6, we also plot the boundary configuration at the final time step, which is an approximate circle.

We remark that the semi-implicit scheme is not unconditionally stable, although its stability is much better than the explicit scheme. This is due to the fact that we have used the small scale decomposition and further approximation of the leading order singular integral operator to simplify the computation of the implicit solution. As we mentioned before, the small scale decomposition captures only the high frequency contribution to

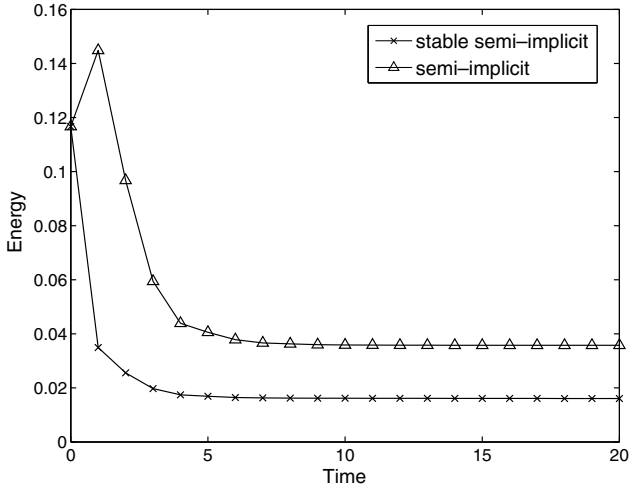


Fig. 5. Total energy of the system for two semi-implicit schemes. Here "stable semi-implicit" stands for the unconditionally stable semi-implicit scheme, and "semi-implicit" the semi-implicit scheme of first kind, $\Delta t = 1$, $N = 128$, $S_b = 1$, $\mu = 0.01$.

Author's personal copy

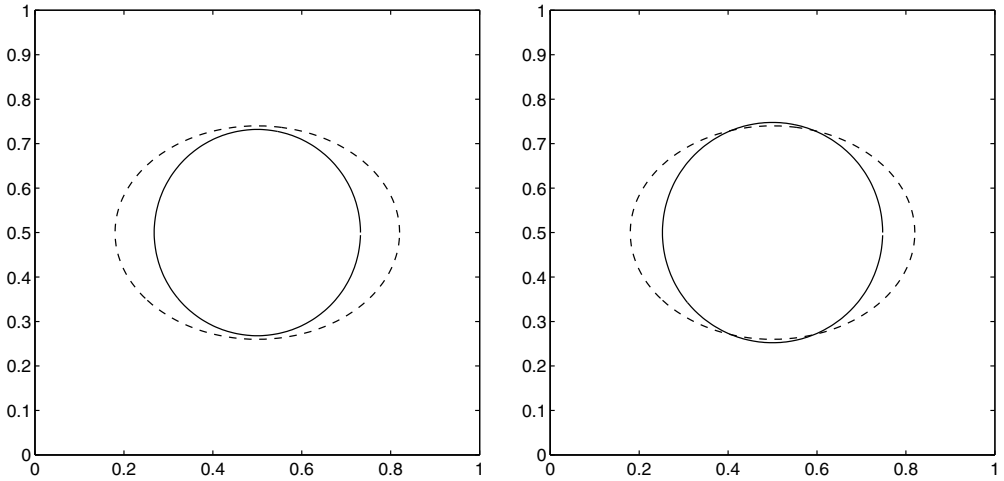


Fig. 6. Dashed line: the initial boundary configuration; Solid line: the boundary configuration after 20 time steps with $\Delta t = 1$, $N = 128$, $S_b = 1$, $\mu = 0.01$. Left: the unconditionally stable semi-implicit scheme; Right: the semi-implicit scheme of the first kind.

the stiffness, but it does not remove the stiffness of the system induced by the low frequency components of the solution. Thus there is still some mild timestep stability constraint for the semi-implicit scheme. The time step has a mild dependence on the elastic coefficient S_b and the viscous coefficient μ. On the other hand, our numerical study shows that the time step for the semi-implicit scheme is independent on the meshsize.

We also compare the performance of our semi-implicit schemes with the explicit scheme. We do not compare the performance of our semi-implicit schemes with the fully implicit scheme here because the fully implicit scheme is quite expensive and is not competitive with the explicit scheme. In order to keep the area loss is no more than 5%, we take $\Delta t = \frac{1}{4}$ for all of the semi-implicit schemes. For the explicit scheme, we take $\Delta t = 1/64, 1/128, 1/256, 1/512$ which correspond to the spatial mesh sizes $N = 64, 128, 256, 512$ respectively, when $\mu = 0.05$. When $\mu = 0.01$ and 0.005, the time step is set to be $\Delta t = 1/128, 1/256, 1/512, 1/1024$ and $t = 1/256, 1/512, 1/1024, 1/2048$. These time steps are the largest ones we can take to keep the stability. We compute the solution up to $T = 2$. The results are documented in Table 2. We can clearly see that the semi-implicit scheme of the first kind gives a significant improvement over the explicit scheme. The cost for the semi-implicit scheme of the second kind is higher than that for the semi-implicit scheme of the first kind. This is because we need to solve for a linear system at each time step for the semi-implicit scheme of the second kind. The cost increases as the number of the spatial grid points increases. The semi-implicit scheme of the second kind and the unconditionally stable semi-implicit scheme both need to solve a linear system at each time step. Their complexity are same, both are $O(N_b^2)$. But for the unconditionally stable semi-implicit scheme, the scaling constant in front of N_b^2 is much larger than the semi-implicit scheme of the second kind. The reason is that the cost of computing the coefficient matrix of the linear system for the unconditionally stable semi-implicit scheme is much higher. As we can see from Table 2, the unconditionally stable semi-implicit scheme (labeled as s,s,i) is still quite expensive compared with our semi-implicit schemes that use the Small Scale

Table 2
Execution times for each computation in seconds

N	$\mu = 0.05$				$\mu = 0.01$				$\mu = 0.005$			
	exp	s,i,1	s,i,2	s,s,i	exp	s,i,1	s,i,2	s,s,i	exp	s,i,1	s,i,2	s,s,i
64	1.8	0.5	4	11	3.3	0.5	4	12	6.6	0.5	4	12
128	9	1	10	48	18	0.9	10	47	35	0.9	10	48
256	58	2.4	25	229	116	2.4	25	228	236	2.4	25	226
512	738	12	99	980	1461	12	98	982	2910	12	98	977

The legends are defined as follows: "exp" stands for the explicit scheme, "s,i,1" the semi-implicit scheme of the first kind, "s,i,2" the semi-implicit scheme of the second kind, "s,s,i" the unconditionally stable semi-implicit scheme.

Author's personal copy

Decomposition. For $\mu = 0.05$, the unconditionally stable semi-implicit scheme is even more expensive compared with the explicit scheme. Although the unconditionally stable semi-implicit scheme is slightly faster than the explicit scheme for smaller μ, the semi-implicit scheme (labeled as s,i,1) which uses SSD to further simply the singular integral kernel gives a much more efficient algorithm. It gives a factor of 242 times speed-up over the explicit scheme in the case of $\mu = 0.005$ and $N = 512$.

7.4. Second order semi-implicit scheme for the unsteady Stokes flow

In this subsection, we perform numerical experiments to test the convergence rate and the stability property of our second order semi-implicit scheme. To check the convergence rate in time, we set $N = 256$ and vary the time step in powers of 2 from $\frac{1}{16}$ to $\frac{1}{128}$. When $\mu = 0.005$, the solution becomes more singular. In order to fully resolve the spatial solution, we increase the spatial resolution to $N = 512$. Following [23], we compute the time discretization error at time t as follows:

$$e_T(u;\Delta t) = \|u(T;\Delta t) - u(T;\Delta t/2)\|_{l^2}. \tag{207}$$

For a vector field $\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))$ defined on the Cartesian grid with $x_i = ih$, $y_j = jh$, the discrete l^2 norm is defined as follows

$$\|\mathbf{u}\|_{l^2} = \left(\sum_{i,j} \big(u_1^2(x_i,y_j) + u_2^2(x_i,y_j)\big)h^2\right)^{\frac{1}{2}}. \tag{208}$$

Similarly, the discrete l^2 norm for a vector field $\mathbf{w}(\alpha) = (w_1(\alpha), w_2(\alpha))$ defined on the interface Γ is defined below:

$$\|\mathbf{w}\|_{l^2} = \left(\sum_{i} \big(w_1^2(\alpha_i) + w_2^2(\alpha_i)\big)\Delta\alpha\right)^{\frac{1}{2}}. \tag{209}$$

We compute the solution up to $T = 1$ and evaluate the convergence rate based on the numerical solution at $T = 1$ with different temporal resolutions. The results are shown in Fig. 7 and Table 3. As we can see, the convergence rate is approximately second order.

Next we check the stability of the second order semi-implicit scheme. Fig. 8 shows that the total energy for the second order explicit and second order semi-implicit schemes with different timesteps 0.002 and 0.02. We

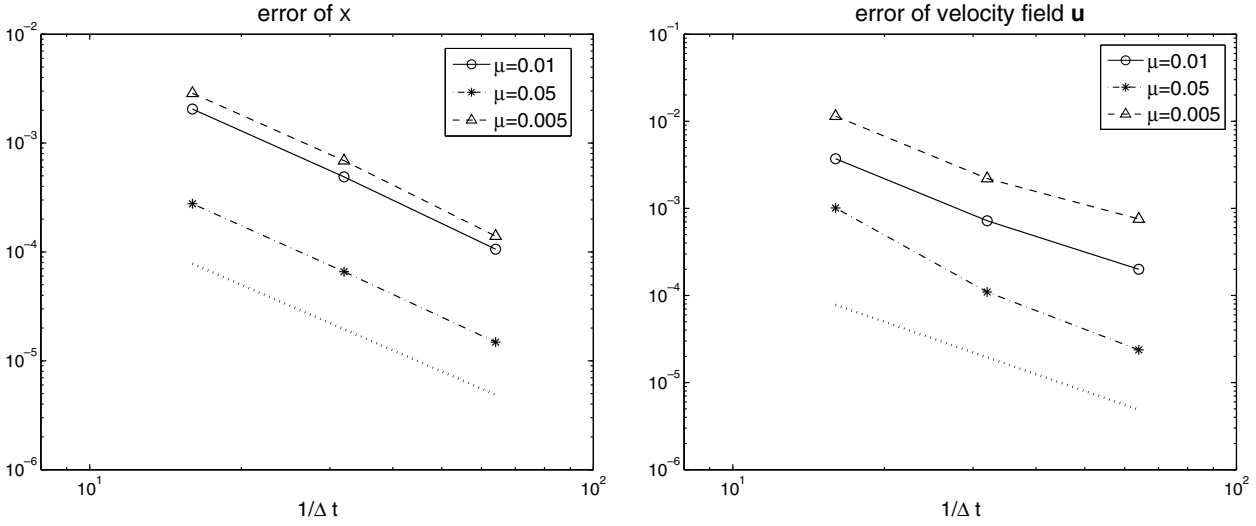


Fig. 7. Plot of l^2 errors in time at time $T = 1$ for the second order semi-implicit scheme of the first kind. We choose $S_b = 1$ and $N = 256$ in all computations except in the case of $\mu = 0.005$ where N is increased to 512. The line at the bottom of each graph is a reference line which corresponds to the second order convergence rate.

Author's personal copy

Table 3
Convergence rates for **X** and **u** fitted from the data shown in Fig. 7

μ	Convergence rate of **X**	Convergence rate of **u**
0.05	2.11	2.70
0.01	2.13	2.10
0.005*	2.17	1.96

The case * is computed using a refined mesh 512×512.

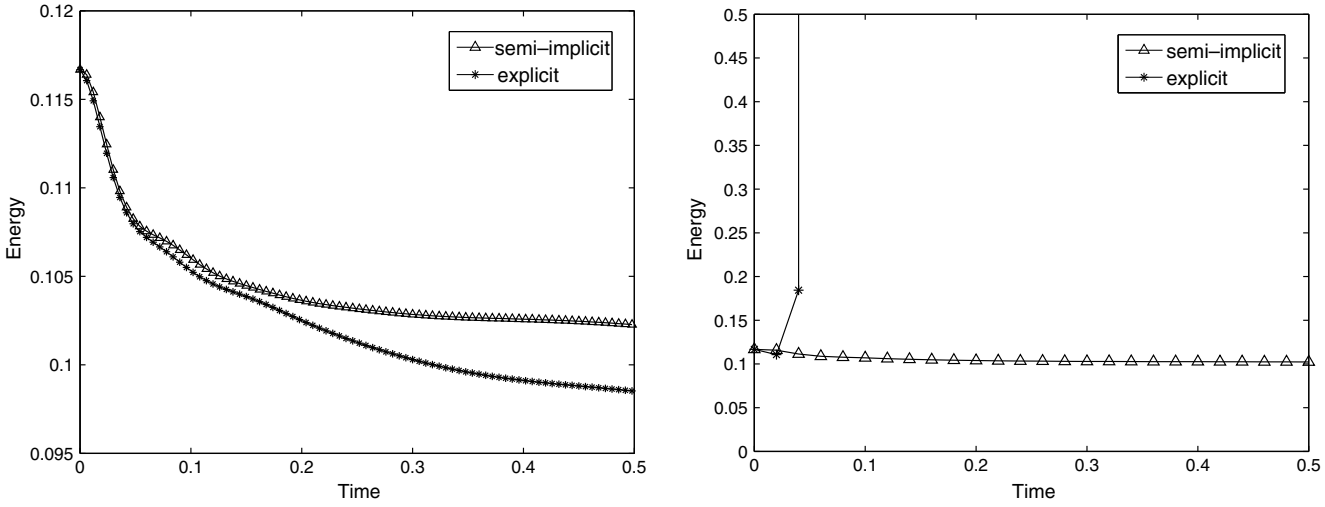


Fig. 8. Total energy of the unsteady Stokes system for the second order semi-implicit scheme and the second order explicit scheme. $N = 128$, $S_b = 1$, $\mu = 0.01$. Left: $\Delta t = 0.002$; Right: $\Delta t = 0.02$.

choose the same second order explicit scheme that was used in [13]. With $\Delta t = 0.002$, both the explicit and semi-implicit schemes are stable and they give nearly identical results. With $\Delta t = 0.02$, the explicit scheme becomes unstable, but the semi-implicit scheme is still stable. The stability restriction of the semi-implicit scheme is far less severe than the corresponding explicit scheme. Our numerical study also shows that the time step for the semi-implicit scheme is independent on the meshsize while the explicit scheme requires finer time step as the spatial mesh is refined.

Finally, we compare the performance of our second order semi-implicit scheme with that of the second order explicit scheme. As before, in order to keep the accuracy with 5%, we take $\Delta t = \frac{1}{4}$ for our semi-implicit schemes. For the explicit scheme, we take $\Delta t = 1/128, 1/256, 1/512, 1/1024$ which correspond to the spatial mesh sizes $N = 64, 128, 256, 512$ respectively, when $\mu = 0.05$. When $\mu = 0.01$ and 0.005, the time step is set to be $\Delta t = 1/128, 1/256, 1/1024, 1/2048$. These time steps are the largest ones that we can take to keep the stability of the explicit scheme. We compute the solution up to $T = 2$. The result is shown in Table 4. Again, we observe the same qualitative behavior as the first order schemes we reported earlier.

Table 4
Execution time for each computation in seconds.

N	$\mu = 0.05$		$\mu = 0.01$		$\mu = 0.005$	
	Explicit	Semi-implicit	Explicit	Semi-implicit	Explicit	Semi-implicit
64	7	0.8	7	0.8	7	0.8
128	37	1.6	38	1.6	38	1.6
256	249	4.4	504	4.6	506	4.5
512	3088	24	6182	25	6200	25

Here "explicit" stands for the second order explicit scheme and "semi-implicit" the second order semi-implicit scheme.

Author's personal copy

8. Concluding remarks

In this paper, we have developed several efficient semi-implicit immersed boundary methods for solving the immersed boundary problem for the steady and unsteady Stokes equations. The immersed boundary method has emerged as one of the most useful numerical methods in computing fluid structure interaction, and has found numerous applications. But it also suffers from the severe time step stability limitation due to the stiffness of the elastic force. Guided by our stability analysis, we have developed several efficient semi-implicit schemes which remove the stiffness of the immersed boundary method. We have demonstrated both analytically and computationally that our semi-implicit schemes have much better stability property than the explicit scheme. More importantly, unlike most existing implicit or semi-implicit schemes, our semi-implicit schemes can be implemented very efficiently. In fact, our semi-implicit scheme of the first kind has a computational cost that is essentially the same as that of an explicit scheme in each time step, but with a much better stability property. The saving in the computational cost is quite substantial. We have demonstrated this improved stability for a range of parameters and numerical resolutions. Our computational results show that the larger the spatial resolution is, the bigger the computational saving our semi-implicit schemes can offer. Thus the semi-implicit schemes we develop in this paper provide an effective alternative discretization to the explicit method.

One of the essential steps in developing our semi-implicit schemes is to obtain an unconditionally stable semi-implicit discretization of the immersed boundary problem. This provides us with a building block to construct our efficient semi-implicit schemes. There are two important observations in constructing the unconditionally stable semi-implicit discretization. The first one is that we need to preserve certain important solution structures at the discrete level. Specifically, we need to ensure that the discrete velocity field is divergence free, and the discrete spreading and interpolation operators are adjoint. Another essential step is to decouple the stiffness of the elastic force from the fluid flow in some appropriate way. This is difficult to achieve if we use the Cartesian coordinate. But it becomes easier if we use the arclength and tangent angle formulation to describe the dynamics of the immersed interface as was done in [9]. On the other hand, as we demonstrated in this paper, the unconditionally stable semi-implicit scheme is still very expensive to implement, and the saving over the expicit scheme is rather limited.

Based on this unconditionally stable semi-implicit discretization, we have developed several efficient schemes for both the steady and the unsteady Stokes flows. By applying the small scale decomposition to the unconditionally stable semi-implicit time discretization and further simplifying the leading order singular kernel, we obtain our semi-implicit scheme. The advantage of this semi-implicit scheme is that the leading order term can be expressed as a convolution operator, which can be evaluated explicitly using the Fourier transformation. This allows us to solve for the implicit solution explicitly in the spectral space, which offers substantial computational saving over the explicit scheme.

It is a natural step to extend the the semi-implicit schemes developed for the unsteady Stokes equations to the Navier–Stokes equations. The discretization of the Navier–Stokes equations shares many similar properties as the unsteady Stokes equations if we treat the convection term explicitly. We have performed a number of numerical experiments to test the stability and the robustness of our semi-implicit immersed boundary methods for the Navier–Stokes equations. The results are qualitatively similar to those for the unsteady Stokes equations which we have presented in this paper. These results will be reported in a subsequent paper.

Acknowledgements

We would like to thank Profs. Charles Peskin and Hector Ceniceros for a number of stimulating discussions on the Immersed Boundary method. The research was in part supported by DOE under the DOE Grant DE-FG02-06ER25727 and by NSF under the NSF FRG Grant DMS-0353838, ITR Grants ACI-0204932 and DMS-0713670.

Appendix A. The semi-implicit scheme of the second kind

In this appendix, we will derive the semi-implicit method of the second kind in more detail. As we mentioned before, the small scale decomposition only captures the leading order contribution from the high fre-

Author's personal copy

quency components, which can not remove the stiffness induced by S_b and μ completely. The coefficients S_b and μ can still affect the time stability through the low frequency components of the solution, which comes from the lower order term of the right hand side. In order to obtain a semi-implicit discretization with better stability property, we can incorporate the low frequency contribution from the second term in our implicit discretization. For the steady Stokes flow, this gives rise to the following decomposition:

$$\frac{\hat{s}_\alpha^{n+1}-\hat{s}_\alpha^n}{\Delta t} = -\frac{S_b}{4\mu}|k|\hat{s}_\alpha^{n+1} + \mathcal{F}\left(\theta_\alpha^n \int \ln|\alpha-\alpha'|(s_\alpha^{n+1}-1)\theta_\alpha^n \,d\alpha'\right) + \left[\mathcal{F}\left(V_\alpha^n - \theta_\alpha^n U^n - \theta_\alpha^n \int \ln|\alpha-\alpha'|(s_\alpha^n-1)\theta_\alpha^n d\alpha'\right) + \frac{S_b}{4\mu}|k|\hat{s}_\alpha^n\right], \tag{210}$$

$$\frac{\hat{\phi}^{n+1}-\hat{\phi}^n}{\Delta t} = -\frac{S_b}{4\mu}\gamma|k|\hat{\phi}^{n+1} + \mathcal{F}\left(\frac{1}{s_\alpha^{n+1}}V^n\theta_\alpha^{n+1}\right) + \left[\mathcal{F}\left(\frac{U_\alpha^n}{s_\alpha^{n+1}}\right) + \frac{S_b}{4\mu}\gamma|k|\hat{\phi}^n\right], \tag{211}$$

where $\gamma = \max_\alpha(1-\frac{1}{s_\alpha})$. By replacing the continuous derivative by the discrete derivative, and discretizing the continuous integral by the trapezoidal rule, we obtain our second semi-implicit scheme. We call this semi-implicit scheme the semi-implicit scheme of the second kind. Near equilibrium, we can show that the semi-implicit scheme of the second kind is unconditionally stable. Moreover, the stability property is independent of the meshsize, elastic coefficient S_b and viscosity coefficient μ. Our numerical study also confirms this. The trade-off is that we need to solve a linear system at each time step to obtain the implicit solution at t^{n+1}.

Similarly, in the case of unsteady Stokes flow, we can also include the second term of the right hand side in the leading order term to derive a scheme with better stability property. In this case, the leading order term becomes:

$$\widehat{T}(s_\alpha^{n+1}) = -\frac{S_b\Delta t}{2(\min_\alpha s_\alpha^n)^2}\left(\frac{\left(\lambda\min_\alpha s_\alpha^n\right)^2k^2+k^4}{\sqrt{(\lambda\min_\alpha s_\alpha^n)^2+k^2}} - |k|^3\right)\hat{s}_\alpha^{n+1} - \frac{S_b\Delta t}{2\pi}\mathcal{F}\left(\frac{\theta_\alpha^n}{(s_\alpha^n)^2}\int (K_0(\lambda s_\alpha^n|\alpha-\alpha'|) - \ln(\alpha-\alpha'))((s_\alpha^{n+1}-1)\theta_{\alpha'}^n)_{\alpha'\alpha'}\,d\alpha'\right), \tag{212}$$

$$\widehat{S}(\theta^{n+1}) = -\frac{S_b\Delta t\max_\alpha(s_\alpha^n-1)}{2(\min_\alpha s_\alpha^n)^2}\left(|k|^3 - \frac{k^4}{\sqrt{(\lambda\min_\alpha s_\alpha^n)^2+k^2}}\right)\hat{\theta}^{n+1} + \mathcal{F}\left(\frac{1}{s_\alpha^{n+1}}V^{n+1}\theta_\alpha^{n+1}\right), \tag{213}$$

where the derivative will be discretized by the spectral method and the integration will be discretized by the trapezoidal rule. We call the above scheme the semi-implicit scheme of the second kind for the unsteady Stokes flow. Near equilibrium stability analysis suggests that the semi-implicit scheme of the second kind is unconditionally stable.

Appendix B. Derivation of the semi-implicit scheme (140)–(143)

In this appendix, we will derive the semi-implicit scheme (140)–(143). We define the operator $\mathcal{G}(s_\alpha;\mathbf{u}^n,\theta^n,\mathbf{X}^n): s_\alpha \to \mathbf{u}$ by the following equations:

$$\frac{\mathbf{u}-\mathbf{u}^n}{\Delta t} = -\nabla_h p + \mu\nabla_h^2\mathbf{u} + L_{h,n}(\mathbf{F}(s_\alpha,\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{214}$$

$$\nabla_h^2 p = \nabla_h\cdot L_{h,n}(\mathbf{F}(s_\alpha,\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)). \tag{215}$$

Given s_α, we obtain $\mathbf{u}$ by solving above equations. From the definition of operator $\mathcal{G}$, we have

$$\mathbf{u}^{n+1} = \mathcal{G}(s_\alpha^{n+1};\mathbf{u}^n,\theta^n,\mathbf{X}^n),$$

$$\mathbf{u}^{*,n+1} = \mathcal{G}(s_\alpha^n;\mathbf{u}^n,\theta^n,\mathbf{X}^n),$$

where $\mathbf{u}^{*,n+1}$ is calculated from Eqs. (143)-(141).

Author's personal copy

Then the equation of s_α can be rewritten as

$$\begin{aligned}\frac{s_\alpha^{n+1}-s_\alpha^n}{\Delta t} &= D_{\Delta\alpha}V^{n+1} - D_{\Delta\alpha}\theta^n U^{n+1}\\ &= D_{\Delta\alpha}(L^*_{h,n}(\mathbf{u}^{n+1})\cdot\boldsymbol{\tau}^n) - D_{\Delta\alpha}\theta^n(L^*_{h,n}(\mathbf{u}^{n+1}\cdot\mathbf{n}^n))\\ &= D_{\Delta\alpha}(L^*_{h,n}(\mathcal{G}(s_\alpha^{n+1};\mathbf{u}^n,\theta^n,\mathbf{X}^n))\cdot\boldsymbol{\tau}^n) - D_{\Delta\alpha}\theta^n(L^*_{h,n}(\mathcal{G}(s_\alpha^{n+1};\mathbf{u}^n,\theta^n,\mathbf{X}^n))\cdot\mathbf{n}^n)\\ &\equiv \mathcal{R}(s_\alpha^{n+1};\mathbf{u}^n,\theta^n,\mathbf{X}^n).\end{aligned} \tag{216}$$

Using the small scale decomposition, we get

$$\mathcal{R}(s_\alpha^{n+1};\mathbf{u}^n,\theta^n,\mathbf{X}^n)\sim T(s_\alpha^{n+1}). \tag{217}$$

By treating the leading order term implicitly, we obtain our semi-implicit method as follows:

$$\begin{aligned}\frac{s_\alpha^{n+1}-s_\alpha^n}{\Delta t} &= T(s_\alpha^{n+1}) + (\mathcal{R}(s_\alpha^n;\mathbf{u}^n,\theta^n,\mathbf{X}^n) - T(s_\alpha^n))\\ &= T(s_\alpha^{n+1}) + D_{\Delta\alpha}(L^*_{h,n}(\mathcal{G}(s_\alpha^n;\mathbf{u}^n,\theta^n,\mathbf{X}^n))\cdot\boldsymbol{\tau}^n) - D_{\Delta\alpha}\theta^n(L^*_{h,n}(\mathcal{G}(s_\alpha^n;\mathbf{u}^n,\theta^n,\mathbf{X}^n))\cdot\mathbf{n}^n) - T(s_\alpha^n)\\ &= T(s_\alpha^{n+1}) + D_{\Delta\alpha}(L^*_{h,n}(\mathbf{u}^{*,n+1})\cdot\boldsymbol{\tau}^n) - D_{\Delta\alpha}\theta^n(L^*_{h,n}(\mathbf{u}^{*,n+1})\cdot\mathbf{n}^n) - T(s_\alpha^n)\\ &= T(s_\alpha^{n+1}) + (D_{\Delta\alpha}V^{*,n+1} - D_{\Delta\alpha}\theta^n U^{*,n+1} - T(s_\alpha^n)).\end{aligned} \tag{218}$$

This is exactly our semi-implicit scheme (134).

In the steady Stokes case, we can define the operator $\mathcal{G}(s_\alpha;\mathbf{u}^n,\theta^n,\mathbf{X}^n): s_\alpha\to\mathbf{u}$ similarly:

$$0 = -\nabla_h p + \mu\nabla_h^2\mathbf{u} + L_{h,n}(\mathbf{F}(s_\alpha,\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)), \tag{219}$$

$$\nabla_h^2 p = \nabla_h\cdot L_{h,n}(\mathbf{F}(s_\alpha,\theta^n;\boldsymbol{\tau}^n,\mathbf{n}^n)). \tag{220}$$

In this case, we have

$$\mathcal{G}(s_\alpha^n;\mathbf{u}^n,\theta^n,\mathbf{X}^n) = \mathbf{u}^n.$$

Thus, the semi-implicit scheme becomes

$$\frac{s_\alpha^{n+1}-s_\alpha^n}{\Delta t} = T(s_\alpha^{n+1}) + (D_{\Delta\alpha}V^n - D_{\Delta\alpha}\theta^n U^n - T(s_\alpha^n)). \tag{221}$$

Appendix C. The derivation of the Fourier transform of K_0

In this appendix, we derive the the Fourier transform of K_0, which is given in (144). By the definition of the Fourier transform, we have

$$\begin{aligned}&\mathcal{F}\left(\frac{1}{\pi}\int_{-\infty}^{+\infty}K_0(\beta|\alpha-\alpha'|)f(\alpha')\,\mathrm{d}\alpha'\right)\\ &= \frac{1}{\pi}\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty}K_0(\beta|\alpha-\alpha'|)f(\alpha')\,\mathrm{d}\alpha'\right)\mathrm{e}^{-\mathrm{i}k\alpha}\,\mathrm{d}\alpha\\ &= \frac{1}{\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}K_0(\beta|\alpha-\alpha'|)f(\alpha')\mathrm{e}^{-\mathrm{i}k\alpha}\,\mathrm{d}\alpha'\,\mathrm{d}\alpha\\ &= \frac{1}{\pi}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}K_0(\beta|\alpha-\alpha'|)f(\alpha')\mathrm{e}^{-\mathrm{i}k(\alpha-\alpha')}\mathrm{e}^{-\mathrm{i}k\alpha'}\,\mathrm{d}\alpha'\,\mathrm{d}(\alpha-\alpha')\\ &= \frac{1}{\pi}\int_{-\infty}^{+\infty}\left(\int_{-\infty}^{+\infty}K_0(\beta|\alpha-\alpha'|)\mathrm{e}^{-\mathrm{i}k(\alpha-\alpha')}\,\mathrm{d}(\alpha-\alpha')\right)f(\alpha')\mathrm{e}^{-\mathrm{i}k\alpha'}\,\mathrm{d}\alpha'\\ &= \int_{-\infty}^{+\infty}\frac{f(\alpha')}{\sqrt{\beta^2+k^2}}\mathrm{e}^{-\mathrm{i}k\alpha'}\,\mathrm{d}\alpha' = \frac{\widehat{f}(k)}{\sqrt{\beta^2+k^2}},\end{aligned} \tag{222}$$

Author's personal copy

where $\mathcal{F}(f(\alpha))(k) = \int_{-\infty}^{+\infty} f(\alpha)e^{ik\alpha}d\alpha$ is the Fourier transform. In the calculation above, we have used the expression of the Bessel function ([1, p. 376])

$$K_0(x) = \int_0^{+\infty} \frac{\cos(tx)}{\sqrt{1+t^2}}\,dt \tag{223}$$

and the identity below:

$$\begin{aligned}
\int_{-\infty}^{+\infty} K_0(\beta|x|)e^{-ikx}\,dx &= \int_{-\infty}^{+\infty}\int_0^{+\infty} \frac{\cos(\beta tx)}{\sqrt{1+t^2}}e^{-ikx}\,dt\,dx \\
&= \frac{1}{2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{\cos(\beta tx)}{\sqrt{1+t^2}}e^{-ikx}\,dt\,dx \\
&= \frac{1}{2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{e^{i\beta tx}}{\sqrt{1+t^2}}e^{-ikx}\,dt\,dx \\
&= \frac{1}{2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \frac{e^{i(\beta t-k)x}}{\sqrt{1+t^2}}\,dt\,dx \\
&= \pi\int_{-\infty}^{+\infty} \frac{\delta(\beta t-k)}{\sqrt{1+t^2}}dt = \frac{\pi}{\sqrt{\beta^2+k^2}}.
\end{aligned} \tag{224}$$

This proves (144).

References

[1] M. Abramowitz, I.A. Stegun, Modified Bessel Functions I and K, Dover, New York, 1972 (9th printing).
[2] R.P. Beyer, A computational model of the cochlea using the immersed boundary method, J. Comput. Phys. 98 (1992) 145–162.
[3] R. Cortez, N. Cowen, R. Dillon, L.J. Fauci, Simulation of swimming organisms: Coupling internal mechanics with external fluid dynamics, Comput. Sci. Eng. 6 (2004) 38–45.
[4] H.D. Ceniceros, A.M. Roma, Study of the long-time dynamics of viscous vortex sheet with a fully adaptive nonstiff method, Phys. Fluid 16 (2004) 4285–4318.
[5] L. Fauci, C.S. Peskin, A computational model of aquatic animal locomotion, J. Comput. Phys. 77 (1988) 85–108.
[6] L.J. Fauci, Interaction of oscillating filaments – A computational study, J. Comput. Phys. 86 (1990) 294–313.
[7] A.L. Fogelson, A mathematical model and numerical methods for studying platelet adhesion and aggregation during blood clotting, J. Comput. Phys. 56 (1984) 111–134.
[8] E. Givelberg, Modeling Elastic Shells Immersed in Fluid, Ph.D. Thesis, 1997, Courant Institute of Mathematical Sciences, New York University.
[9] T.Y. Hou, J. Lowengrub, M. Shelley, Removing the stiffness from interfacial flows with surface tension, J. Comput. Phys. 114 (1994) 312–338.
[10] T.Y. Hou, J. Lowengrub, M. Shelley, The long-time motion of vortex sheets with surface tension, Phys. Fluid A 9 (1997) 1933–1954.
[11] M. Hopkins, L.J. Fauci, A computational model of the collective fluid dynamics of motile microorganisms, J. Fluid Mech. 455 (2002) 149–174.
[12] E.N. Jung, C.S. Peskin, Two-dimensional simulations of valveless pumping using the immersed boundary method, SIAM J. Sci. Comput. 23 (2001) 19–45.
[13] 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.
[14] 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.
[15] A.A. Mayo, C.S. Peskin, An implicit numerical method for fluid dynamics problems with immersed elastic boundaries, Contemp. Math. 141 (1993) 261–277.
[16] D.M. McQueen, C.S. Peskin, E.L. Yellin, Fluid dynamics of the mitral valve: Physiological aspects of a mathematical model, Am. J. Physiol. 242 (1982) H1095–H1110.
[17] D.M. McQueen, C.S. Peskin, Computer assisted design of pivoting-disc prosthetic mitral valves, J. Thorac. Cardiovasc. Surg. 86 (1983) 126–135.
[18] D.M. McQueen, C.S. Peskin, Computer assisted design of butterfly bileaflet valves for mitral position, Scand. J. Thorac. Cardiovasc. Surg. 19 (1985) 139–148.
[19] D.M. McQueen, C.S. Peskin, A three-dimensional computational method for blood flow in the heart: (II) Contractile fibers, J. Comput. Phys. 82 (1989) 289–297.

Author's personal copy

[20] D.M. McQueen, C.S. Peskin, Heart simulation by an immersed boundary method with formal second order accuracy and reduced viscosity, in: Mechanics for a New Millennium Proceedings of the International Conference on Theoretical and Applied mechanics (ICTAM), Kluwer Academic Publishers, 2000.
[21] L.A. Miller, C.S. Peskin, When vortices stick: and aerodynamic transition in tiny insects, J. Exp. Biol. 207 (2004) 3073–3088.
[22] L.A. Miller, C.S. Peskin, A computational fluid dynamics of 'clap and fling' in small insects, J. Exp. Biol. 208 (2005) 195–212.
[23] Y. Mori, C.S. Peskin, Implicit second-order immersed boundary methods with boundary mass, Comput. Method Appl. Mech. Engrg. (2007), doi:10.1016/j.cma.2007.05.028.
[24] 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.
[25] C.S. Peskin, Numerical study of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.
[26] C.S. Peskin, D.M. McQueen, A three-dimensional computational method for blood flow in the heart: (I) Immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (1989) 372–405.
[27] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[28] C. Pozrikidis, Boundary Integral and Singularity Methods for Linearized Viscous Flow, Cambridge University Press, 1992.
[29] 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.
[30] J.M. Stockie, B.R. Wetton, Stability analysis for the immersed fiber problem, SIAM J. Appl. Math. 55 (1995) 1577–1591.
[31] J.M. Stockie, S.I. Green, Simulating the motion of flexible pulp fibers using the immersed boundary method, J. Comput. Phys. 147 (1998) 147–165.
[32] 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.
[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. Stat. Comput. 13 (1992) 1361–1376.
[34] N.T. Wang, A.L. Fogelson, Computational methods or continuum models of platelet aggregation, J. Comput. Phys. 151 (1999) 649–675.
[35] L. Zhu, C.S. Peskin, Simulation of a flexible flapping filament in a flowing soap film by the immersed boundary method, J. Comput. Phys. 179 (2002) 452–468.