Journal of Computational Physics **171,** 822–842 (2001)
doi:10.1006/jcph.2001.6813, available online at http://www.idealibrary.com on IDEAL®

The Immersed Interface Method for the Navier–Stokes Equations with Singular Forces

Zhilin Li and Ming-Chih Lai

Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, North Carolina 27695-8205; and Department of Mathematics, Chung Cheng University, Minghsiung, Chiayi 621, Taiwan
E-mail: zhilin@math.nscu.edu; mclai@math.ccu.edu.tw

Received September 12, 2000; revised April 5, 2001

Peskin's Immersed Boundary Method has been widely used for simulating many fluid mechanics and biology problems. One of the essential components of the method is the usage of certain discrete delta functions to deal with singular forces along one or several interfaces in the fluid domain. However, the Immersed Boundary Method is known to be first-order accurate and usually smears out the solutions. In this paper, we propose an immersed interface method for the incompressible Navier–Stokes equations with singular forces along one or several interfaces in the solution domain. The new method is based on a second-order projection method with modifications only at grid points near or on the interface. From the derivation of the new method, we expect fully second-order accuracy for the velocity and nearly second-order accuracy for the pressure in the maximum norm including those grid points near or on the interface. This has been confirmed in our numerical experiments. Furthermore, the computed solutions are sharp across the interface. Nontrivial numerical results are provided and compared with the Immersed Boundary Method. Meanwhile, a new version of the Immersed Boundary Method using the level set representation of the interface is also proposed in this paper. © 2001 Academic Press

Key Words: Navier–Stokes equations; interface; discontinuous and nonsmooth solution; immersed interface method; immersed boundary method; projection method; level set method.

1. INTRODUCTION

In this paper, we consider the incompressible Navier–Stokes equations in a bounded domain $\Omega \subset \mathcal{R}^2$ that contains one or several interfaces $\Gamma(t)$:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \mu\Delta\mathbf{u} + \mathbf{G} + \mathbf{F}, \quad \mathbf{x} \in \Omega, \tag{1.1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{1.2}$$

$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_b, \quad \text{BC}, \tag{1.3}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \quad \text{IC}. \tag{1.4}$$

Here we write **F** and **G** separately to distinguish different irregularities. The singular force **F** has support only on the immersed interface $\Gamma(t)$ and has the form

$$\mathbf{F}(\mathbf{x}, t) = \int_{\Gamma(t)} \mathbf{f}(s, t)\delta_2(\mathbf{x} - \mathbf{X}(s, t))\, ds, \tag{1.5}$$

where $\mathbf{X}(s, t)$ is the arc-length parameterization (s is the arc-length parameter) of the interface, and $\mathbf{f}\,(s, t)$ is the force strength. The above integral is over the entire interface, and δ_2 is the two-dimensional Dirac function, $\delta_2(\mathbf{x}) = \delta(x)\delta(y)$ with $\mathbf{x} = (x, y)$. The term **G** may have a finite jump across the interface, but is bounded and piecewise continuous in the entire domain. Throughout this paper, we simply assume that the density $\rho \equiv 1$ and the viscosity μ is a constant. Figure 1 is an illustration of the geometry and the local coordinates for the problems discussed in this paper.

Although we assume that the density and the viscosity are two constants, the model (1.1)–(1.5) has many applications, for example, cardiac blood flow [32–34], platelet aggregation and coagulation [9, 10], swimming microorganisms [8], biofilm formation [6], suspension flows [38], and others; see also [35] for a brief review.

In the 1970–1980s, Peskin [32, 33] developed the Immersed Boundary Method (IB method) which used the model (1.1)–(1.5) to simulate the blood flow through heart valves. The valve leaflet (the interface) exerts some force into the blood (the fluid) and moves along with the fluid simultaneously. Peskin's IB method has become a standard numerical method for interface problems that involve singular forces along one or several interfaces. The method has been applied successfully to many fluid and biological problems.

However, it is also known that generally the IB method is only first-order accurate for nonsmooth but continuous quantities. Different variations of IB method have been proposed to improve the accuracy. For example, Lai and Peskin [18] proposed a new *formally* second-order IB method with reduced numerical viscosity and applied the new method to simulate

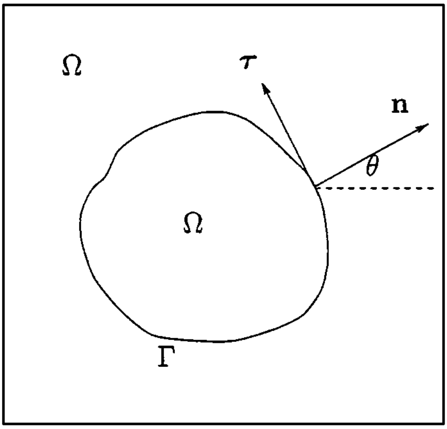

FIG. 1. A diagram of the geometry for the interface problems discussed in this paper. We use **n** and τ to denote the unit normal and tangential directions of the interface, respectively; and θ is the angle between the normal direction and the x-axis.

the flow around a circular cylinder. By formal second-order accuracy, the authors meant that the scheme would be second-order accurate if the delta function is replaced by a fixed smooth function, independent of the mesh. The numerical result has a good agreement with the experimental data; however, the *fully* second-order accuracy in the maximum norm, which is the goal of this paper, has not been achieved there. More recently, Cortez and Minion [5] proposed another high-order scheme using the smoothed blob projection to compute the force. Their method shows better accuracy in L_1 and L_2 norms and preserves the volume better than the original IB method. Nevertheless, both approaches described in [5] and [18] use the discrete delta function which usually smears out the solution across the interface.

In contrast to the discrete delta function approach, the immersed interface method (IIM) [20–23, 25], which incorporates the jump conditions into the finite difference scheme, was developed to improve the accuracy of the IB method and to obtain sharp interface solutions. The IIM method has been shown to be second-order accurate in the maximum norm for elliptic problems [26] and has been applied to Stokes flow [21] and other moving interface/free boundary problems [14, 27, 28]. In this paper, we extend the IIM method to the full Navier–Stokes equations (1.1)–(1.5) and compare the new method with different versions of the IB method.

The Ghost Fluid Method (GFM), which is also a sharp interface method that uses the jump conditions in the solution and the flux, has been developed by Fedkiw, Liu, Kang and their group for elliptic interface problems [29]. The GFM is simpler than IIM and the resulting linear system of equations for the self-adjoint elliptic equations is symmetric positive definite. The trade-off is that the solution is first-order accurate in the maximum norm. The method has been applied to multiphase incompressible flow [16] and two phase incompressible flames simulations [30].

Another sharp interface method in which the jump conditions in the *coefficients* was taken into account in constructing the algorithm was developed in [11] and [12] for the incompressible Navier–Stokes equations.

The method developed in this paper is for *sharp interface models*. Besides the differences in the methodology, our method distinguishes from [16, 29, 30] in accuracy, and distinguishes from [11, 12] in dealing with different problems. There are some other models and methods for interface problems, notably, the *phase field model* and the finite volume method, for example, [36]. Each model has advantages and disadvantages, and may have specific applications. It is easy to compute surface tension and curvature with sharp interface models. It is also an advantage to use the level set method with sharp interface models.

The rest of the paper is organized as follows. In Section 2, we introduce the jump conditions for (1.1)–(1.5). In Section 3, we present the IIM scheme for (1.1)–(1.5) which modifies a projection method by adding some correction terms. In Section 4, numerical results for nontrivial examples including one with a moving interface are presented and analyzed. A new version of the IB method with the level set formulation is also proposed there. Some conclusions will be given in Section 5.

2. JUMP CONDITIONS ACROSS THE INTERFACE

Because of the singular forces, the velocity of the solution to (1.1)–(1.5) is typically nonsmooth, and the pressure may be discontinuous, across the interface. The jump conditions of the velocity and the pressure are quantitatively described in the following theorem.

THEOREM 1. *Let (X, Y) be a point on the interface. Let Γ be a smooth closed interface within the domain. Let the unit outward normal direction be $\mathbf{n} = (\cos\theta, \sin\theta)$, where θ is the angle between the outward normal direction and the x-axis. Then we have the jump conditions*

$$[\mathbf{u}] = \mathbf{0}, \quad [\mu \mathbf{u}_{\mathbf{n}}] = -\hat{f}_2 \tau, \tag{2.6}$$

$$[p] = \hat{f}_1, \quad [p_{\mathbf{n}}] = \frac{\partial \hat{f}_2}{\partial s} + [\mathbf{G}] \cdot \mathbf{n}, \tag{2.7}$$

where $\tau = (-\sin\theta, \cos\theta)$ is the unit tangent direction, and $\hat{f}_1$ and $\hat{f}_2$ are the force strengths in the normal and tangential directions, respectively

$$\begin{aligned} \hat{f}_1(s,t) &= f_1(s,t)\cos\theta + f_2(s,t)\sin\theta, \\ \hat{f}_2(s,t) &= -f_1(s,t)\sin\theta + f_2(s,t)\cos\theta. \end{aligned} \tag{2.8}$$

The jump $[\cdot]$ is defined as the difference of the limiting value from outside of the interface and that from the inside, and s is the arc-length parameter of the interface.

The proof can be found in [19, 21] with minor modifications.

2.1. *Additional Interface Relations*

In this subsection, we derive some additional interface relations by differentiating the known jump conditions in Theorem 1 along the interface, and by making use of the momentum equation (1.1). These interface relations are summarized in Theorem 2 and will be used in our new numerical scheme. For interface problems, most physical quantities along/across the interface are associated with the normal and tangential directions. Thus, we introduce a local coordinate system which lies along those directions, and derive more interface relations in the local coordinate system.

THEOREM 2. *Under the same notations and assumptions as in Theorem 1, we define the local coordinates at (X, Y), a point on the interface, as*

$$\begin{aligned} \xi &= (x - X)\cos\theta + (y - Y)\sin\theta, \\ \eta &= -(x - X)\sin\theta + (y - Y)\cos\theta. \end{aligned} \tag{2.9}$$

Then the interface Γ can be represented by $\xi = \chi(\eta)$ in the neighborhood of $(\xi, \eta) = (0, 0)$, which satisfies $\chi(0) = 0$, $\chi'(0) = 0$, and $\chi''(0) = \kappa$, the curvature of the interface at $(0, 0)$. The following interface relations are true at (X, Y):

$$\begin{gathered} [p] = \hat{f}_1, \quad [p_\xi] = \frac{\partial \hat{f}_2}{\partial \eta}, \\ [\mathbf{u}] = \mathbf{0}, \quad [\mu \mathbf{u}_\xi] = -\hat{f}_2 \tau, \quad [\mathbf{u}_\eta] = \mathbf{0}, \\ [\mu \mathbf{u}_{\eta\eta}] = \kappa \hat{f}_2 \tau, \quad [\mu \mathbf{u}_{\xi\eta}] = -\frac{\partial \hat{f}_2}{\partial \eta}\tau - \kappa \hat{f}_2 \mathbf{n}, \\ [\mu \mathbf{u}_{\xi\xi}] = -[\mu \mathbf{u}_{\eta\eta}] + [p_\xi]\,\mathbf{n} + [p_\eta]\,\tau + [\mathbf{u}_\xi]\,\mathbf{u} \cdot \mathbf{n} - [\mathbf{G}]. \end{gathered} \tag{2.10}$$

The proof of the theorem is quite technical, long, and tedious. However, it is straightforward if we know the key ideas. Therefore, we will provide only the outline of the proof.

Sketch of the proof. The first four equalities are directly copied from the jump conditions in Theorem 1. Let the interface Γ be $\xi = \chi(\eta)$ in the neighborhood of $(\xi, \eta) = (0, 0)$. Then we know that $\chi(0) = 0$, $\chi'(0) = 0$, and $\chi''(0) = \kappa$. Differentiating $[\mathbf{u}] = \mathbf{0}$ along the interface and using $\chi'(0) = 0$, we get the fifth equality. By differentiating $[\mathbf{u}] = \mathbf{0}$ along the interface twice and using $\chi'(0) = 0$, and $\chi''(0) = \kappa$, we get the sixth equality. By differentiating the fourth equality along the interface and using $\frac{\partial \tau}{\partial \eta} = \kappa \mathbf{n}$, we get the seventh equality.[1] Finally, by expressing the momentum equation (1.1) in the local coordinate (2.9), we get the last equality.

2.2. *Projection of the Jump Relations on x- and y-Direction*

In order to be numerically useful, those interface relations in (2.10) in the local coordinate (2.9) are transformed into the jump relations in the Cartesian coordinates. Since μ is continuous, we have

$$\begin{aligned}
[u_x] &= [u_\xi]\cos\theta - [u_\eta]\sin\theta,\\
[u_y] &= [u_\xi]\sin\theta + [u_\eta]\cos\theta,\\
[u_{xx}] &= [u_{\xi\xi}]\cos^2\theta - 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\sin^2\theta,\\
[u_{yy}] &= [u_{\xi\xi}]\sin^2\theta + 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\cos^2\theta.
\end{aligned} \tag{2.11}$$

Using these identities and the interface relations in (2.10), we can easily write down $[\mathbf{u}]$, $[\mathbf{u}_x]$, $[\mathbf{u}_y]$, $[\mathbf{u}_{xx}]$, $[\mathbf{u}_{yy}]$, $[p]$, $[p_x]$, and $[p_y]$ at any point on the interface in terms of the force strength and its derivatives, and the geometric information of the interface.

3. THE NUMERICAL ALGORITHM

Our numerical method is based on the projection method for solving the incompressible Navier–Stokes equations with careful treatment at grid points that are near or on the interface. There are several versions of the projection method for solving the incompressible Navier–Stokes equations; see for example, [1, 15, 17] and many others. The projection method that we used in this paper is the one described in [2] which is based on the pressure increment formulation of [1, 15] with an additional accurate pressure correction (in [2]; the method is denoted by Pm II). Our key modification to the projection method (Pm II) is to add some correction terms at grid points near or on the interface. Those correction terms are determined from the interface relations that we stated in the previous sections.

For simplicity, we assume that the domain Ω is a rectangle $[a, b] \times [c, d]$, and the spatial spacing is $h = (b - a)/M = (d - c)/N$, where M and N are the number of grid points in the x and y directions, respectively. Here, a standard uniform Cartesian grid is used. Our method from time t^n to t^{n+1} can be written as

$$\begin{gathered}
\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u}\cdot\nabla_h\mathbf{u})^{n+\frac{1}{2}} = \nabla_h p^{n-\frac{1}{2}} + \frac{\mu}{2}(\Delta_h\mathbf{u}^* + \Delta_h\mathbf{u}^n) + \mathbf{G}^{n+\frac{1}{2}} + \mathbf{C}_1^n,\\
\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_b^{n+1},
\end{gathered} \tag{3.12}$$

[1] In our notation, a circle has negative curvature.

where $(\mathbf{u}\cdot\nabla_h\mathbf{u})^{n+\frac{1}{2}}$ is approximated using

$$(\mathbf{u}\cdot\nabla_h\mathbf{u})^{n+\frac{1}{2}} = \frac{3}{2}(\mathbf{u}^n\cdot\nabla_h)\mathbf{u}^n - \frac{1}{2}(\mathbf{u}^{n-1}\cdot\nabla_h)\mathbf{u}^{n-1} + \mathbf{C}_2^n. \tag{3.13}$$

The projection step is the following:

$$\Delta_h\phi^{n+1} = \frac{\nabla_h\cdot\mathbf{u}^*}{\Delta t} + C_3^n, \quad \left.\frac{\partial\phi^{n+1}}{\partial\mathbf{n}}\right|_{\partial\Omega} = 0, \tag{3.14}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\nabla_h\phi^{n+1} + \mathbf{C}_4^n. \tag{3.15}$$

$$\nabla_h p^{n+\frac{1}{2}} = \nabla_h p^{n-\frac{1}{2}} + \nabla_h\phi^{n+1} + \mathbf{C}_5^n. \tag{3.16}$$

In the expressions above, ∇_h and Δ_h are the standard central difference operators regardless of the interface, thus,

$$\Delta_h u_{ij} = \left(\frac{u_{i+1,j}-u_{i-1,j}}{2h}, \frac{u_{i,j+1}-u_{i,j-1}}{2h}\right),$$

$$\Delta_h u_{ij} = \frac{u_{i-1,j}+u_{i,j-1}+u_{i+1,j}+u_{i,j+1}-4u_{ij}}{h^2}.$$

It is clear that the modifications that we made to the projection method are those correction terms C_k^n at grid points near or on the interface.

3.1. *Determining the Correction Terms*

At a regular grid point (x_i, y_j), meaning that all the grid points in the standard centered five-point stencil are from the same side of the interface, we use the standard central difference scheme to discretize Eqs. (1.1)–(1.4) without any correction; that is $C_k^n \equiv 0$. At an irregular grid point (x_i, y_j), we need to determine the correction terms so that the finite difference scheme can be as accurate as possible. We assume that the interface cuts a grid line between two grid points no more than once. This is guaranteed if the interface is expressed in terms of a level set function.

First, we establish the following lemma, which is the basis for determining the correction terms.

LEMMA 1. *Let $u(x)$ be a piecewise twice differentiable function. Assume that $u(x)$ and its derivatives have finite jumps $[u]$, $[u_x]$, and $[u_{xx}]$, at $x^* = x + \alpha h$, $-1 \le \alpha \le 1$, then the following relations hold,*

$$\frac{u(x+h)-u(x-h)}{2h} = \begin{cases} u'(x) + \dfrac{C(x,\alpha)}{2h} + O(h^2), & \textit{if } 0 \le \alpha \le 1, \\[2ex] u'(x) - \dfrac{C(x,\alpha)}{2h} + O(h^2), & \textit{if } -1 \le \alpha < 0, \end{cases} \tag{3.17}$$

$$\frac{u(x+h)-2u(x)+u(x-h)}{h^2} = u''(x) + \frac{C(x,\alpha)}{h^2} + O(h), \tag{3.18}$$

where

$$C(x,\alpha) = [u] + [u_x](1-|\alpha|)h + [u_{xx}]\frac{(1-|\alpha|)^2h^2}{2}, \tag{3.19}$$

and the jumps are defined as

$$[u] = \begin{cases} \lim_{x\to x^*+} u(x) - \lim_{x\to x^*-} u(x), & \textit{if } 0 \le \alpha \le 1, \\ \lim_{x\to x^*-} u(x) - \lim_{x\to x^*+} u(x), & \textit{if } -1 \le \alpha < 0. \end{cases} \tag{3.20}$$

$$[u_x] = \begin{cases} \lim_{x\to x^*+} u_x(x) - \lim_{x\to x^*-} u_x(x), & \textit{if } 0 \le \alpha \le 1, \\ \lim_{x\to x^*-} u_x(x) - \lim_{x\to x^*+} u_x(x), & \textit{if } -1 \le \alpha < 0. \end{cases} \tag{3.21}$$

$$[u_{xx}] = \begin{cases} \lim_{x\to x^*+} u_{xx}(x) - \lim_{x\to x^*-} u_{xx}(x), & \textit{if } 0 \le \alpha \le 1, \\ \lim_{x\to x^*-} u_{xx}(x) - \lim_{x\to x^*+} u_{xx}(x), & \textit{if } -1 \le \alpha < 0. \end{cases} \tag{3.22}$$

Proof. Without loss of generality, we assume that $0 < \alpha \le 1$. Therefore, $x - h$ and x are on the same side while $x + h$ is on the other side. We use the Taylor expansion twice for $u(x + h)$ at x^*, then at x, as follows,

$$\begin{aligned} u(x+h) = u(x^* + (1-\alpha)h) &= u(x^*+) + u_x(x^*+)(1-\alpha)h \\ &\quad + u_{xx}(x^*+)\frac{(1-\alpha)^2h^2}{2} + O(h^3) \\ &= u(x^*-) + u_x(x^*-)(1-\alpha)h + u_{xx}(x^*-)\frac{(1-\alpha)^2h^2}{2} + C(x,\alpha) + O(h^3) \\ &= u(x) + u_x(x)(x^*-x) + u_{xx}(x)\frac{(x^*-x)^2}{2} + u_x(x)(1-\alpha)h \\ &\quad + u_{xx}(x)(x^*-x)(1-\alpha)h + u_{xx}(x)\frac{(1-\alpha)^2h^2}{2} + C(x,\alpha) + O(h^3). \end{aligned}$$

The Taylor expansion for $u(x - h)$ is

$$u(x-h) = u(x) - hu_x(x) + u_{xx}(x)\frac{h^2}{2} + O(h^3).$$

After substituting these two expansions into the left-hand sides of (3.17) and (3.18) and with some manipulations, we get the desired equalities.

Remark 1. The lemma above is the basis of our numerical algorithm for determining the correction terms.

- The correction terms given in the lemma can be found in the literature. For example, (3.18) is a special case in the expressions (2.22) and (2.24) in [22]. It was also derived independently in [29] for the ghost fluid method. Equation (3.17) is a special case in the expressions (26) and (39) in [24]. Nevertheless, in the lemma we have “uniform” formulae for both first- and second-order derivatives; see below for the explanation.
- Lemma 1 tells us that the correction term is simply the product of the finite difference coefficient, corresponding to the grid point from different sides of the interface in reference to the master grid point, and $C(x, \alpha)$ which corresponds to the singular source term. For example, if $-1 < \alpha < 0$, in Lemma 1, which means $x - h$ and x are on different sides of the interface, then the correction terms for u_x is the product of the coefficient $-1/(2h)$ and $C(x, \alpha)$. Therefore, it is very easy to modify the finite difference scheme to maintain a certain order of accuracy.

- If there are two interfaces between $(x-h, x+h)$, say, $x_1^* = x + \alpha_1 h$ and $x_2^* = x + \alpha_2 h$, with $\alpha_1 < 0$ and $\alpha_2 \geq 0$, then we just need to add another correction term, for example,

$$\frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = u''(x) + \frac{C(x,\alpha_1)}{h^2} + \frac{C(x,\alpha_2)}{h^2} + O(h),$$
$$\frac{u(x+h) - u(x-h)}{2h} = u'(x) + \frac{C(x,\alpha_1)}{2h} - \frac{C(x,\alpha_2)}{2h} + O(h^2).$$

3.2. *Correction Terms to the Projection Method*

In our numerical scheme (3.12)–(3.16), there are several correction terms to be determined. For the sake of simplicity, we will only explain how to evaluate some of the correction terms of the x-component of $\mathbf{C}_1^n$. The other correction terms can be treated in the same way so we omit the details here. Assume that x^*, $x_i \leq x^* \leq x_{i+1}$, is an intersection of the interface and the x-axis. The contribution to $\mathbf{C}_1^n$ in the x-direction from this interface point includes the following:

- the correction term for $u^n D_{x,h} u^n$ from the nonlinear term $(\mathbf{u}^n \cdot \nabla_h)\mathbf{u}^n$:

$$-\frac{3}{4h}\left(\left[u_x^n\right](x_{i+1} - x^*) + \left[u_{xx}^n\right]\frac{(x_{i+1} - x^*)^2}{2}\right)u^n(x^*, y_j);$$

- the correction term for $u^{n-1} D_{x,h} u^{n-1}$ from the nonlinear term $(\mathbf{u}^{n-1} \cdot \nabla_h)\mathbf{u}^{n-1}$:

$$\frac{1}{4h}\left(\left[u_x^{n-1}\right](x_{i+1} - x^*) + \left[u_{xx}^{n-1}\right]\frac{(x_{i+1} - x^*)^2}{2}\right)u^{n-1}(x^*, y_j);$$

- the correction term for $D_{x,h} p^{n-\frac{1}{2}}$ from $\nabla_h p^{n-\frac{1}{2}}$:

$$-\frac{1}{2h}\left(\left[p^{n-\frac{1}{2}}\right] + \left[p_x^{n-\frac{1}{2}}\right](x_{i+1} - x^*)\right);$$

- the correction term for $\mu(u_{i-1,j}^* - 2u_{i,j}^* + u_{i+1,j}^*)/(2h^2)$ from $\mu\Delta_h u^*/2$:

$$-\frac{\mu}{2h^2}\left(\left[u_x^{n+1}\right](x_{i+1} - x^*) + \left[u_{xx}^{n+1}\right]\frac{(x_{i+1} - x^*)^2}{2}\right);$$

- the correction term for $\mu(u_{i-1,j}^n - 2u_{i,j}^n + u_{i+1,j}^n)/(2h^2)$ from $\mu\Delta_h u^n/2$:

$$-\frac{\mu}{2h^2}\left(\left[u_x^n\right](x_{i+1} - x^*) + \left[u_{xx}^n\right]\frac{(x_{i+1} - x^*)^2}{2}\right).$$

We use the *bilinear interpolation* to approximate the values of u and v at (x^*, y_j) from the values of u and v at the neighboring four grid points. Such approximation is at least first-order accurate since the velocity is continuous. Note that we have used the jump conditions at t^{n+1} to approximate the jump conditions at t^* corresponding to the same treatment for the velocity boundary condition in the projection method.

Similarly, if the interface cuts through between x_{i-1} and x_i, we need to add the corresponding correction terms as well. In this way, we take care of the correction terms in the x-direction for $\mathbf{C}_1^n$.

The spatial local truncation error of our scheme is $O(h^2)$ at regular grid points, but $O(h)$ at irregular grid points. However, since the number of irregular grid points usually is one dimensional lower than the total number of grid points, we expect the global accuracy is still second order for the velocity, and nearly second order for the pressure. A proof of second-order convergence for an elliptic interface problem can be found in [26]; see also the numerical experiments in Section 4.

It is worth pointing out that the modifications that we made to the projection method only affect the right-hand sides of the original projection method. Therefore, our method does not change the stability nature of the original projection method, which is stable if the CFL condition is satisfied.

Once we have computed those correction terms, we need to solve two Helmholtz equations to get $\mathbf{u}^*$ in (3.12) and one Poisson equation to get ϕ^{n+1} in (3.14). The discrete systems for the Helmholtz and Poisson equation are symmetric and positive definite, and are solved using the fast solver *HWSCRT* from Fishpack.

3.3. *Further Correction Near the Boundary and the Interface*

Using the projection method with the correction terms added at irregular grid points, we have observed that the velocity is second-order accurate but the pressure is only first order. The largest error in the pressure appears near the boundary or the interface. As analyzed in [2, 7, 37, 39], this is due to the inherent numerical boundary layer in the projection method. In [2], see also [17], a correction scheme is proposed which is

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n+1} - \frac{\mu}{2}(\nabla_h \cdot \mathbf{u}^* + C_6^n), \tag{3.23}$$

where ϕ^{n+1}, in our notation, is the solution of the Poisson equation of (3.14)–(3.15) from t^n to t^{n+1}, C_6^n again is a correction term only needed at irregular grid points near or on the interface. This correction does dramatically improve the accuracy for the pressure (see Section 4).

In our implementation, we use the centered difference scheme to compute the correction term $\nabla_h \cdot \mathbf{u}^*$ in interior grid points. Then we use an extrapolation scheme to extend the quantity to the boundary of the rectangle, for example, at the boundary $x = a$,

$$\nabla_h \cdot \mathbf{u}^*(0, j) = 3\nabla_h \cdot \mathbf{u}^*(1, j) - 3\nabla_h \cdot \mathbf{u}^*(2, j) + \nabla_h \cdot \mathbf{u}^*(3, j). \tag{3.24}$$

Near the interface, even though the velocity is second-order accurate, the term $\nabla_h \cdot \mathbf{u}^*/\Delta t$ may have only zeroth order accuracy since the error may not be smooth near the interface, and $\Delta t \sim h$. This will reduce the accuracy of pressure to first order. Our remedy to this problem is to replace $\nabla_h \cdot \mathbf{u}^*/\Delta t$ at an irregular grid point by the same quantity at the nearest regular grid point. This again improves the accuracy of the pressure. The same technique was also used independently in [4] where a second-order accurate symmetric sharp interface method for the Poisson equation with Dirichlet boundary conditions was proposed.

We will call the numerical method described in this section as the projection-IIM method.

4. NUMERICAL EXAMPLES

In this section, we present some numerical experiments for solving the velocity and presure in (1.1)–(1.4) at some fixed time t. The velocity, and the presure up to a constant, are uniquely determined by the initial conditions $\mathbf{u}(\mathbf{x}, 0)$, $p(\mathbf{x}, 0)$, and $\Gamma(s, 0)$; the initial interface; the boundary condition $\mathbf{u}(\mathbf{x}, t)|_{\partial\Omega}$; and the force strength $(\hat{f}_1, \hat{f}_2)$ in the normal and tangential directions along the interface.

All the calculations in this section were perfomed at North Carolina State University using Sun workstations. Most simulations are done within minutes to a couple of hours depending on the number of grid points. Unless specified differently, we use a level set function to represent the interface; see (4.40).

4.1. *Example 1: A Noninterface Problem*

We start our numerical tests by checking the accuracy of the projection method for a regular problem, and use the result as the basis for comparing different algorithms for the interface examples. The following exact solution is taken from [40]:

$$\begin{aligned} u(x, y, t) &= -\sin^2(\pi x)\sin(2\pi y)\cos t, \\ v(x, y, t) &= \sin^2(\pi y)\sin(2\pi x)\cos t. \end{aligned} \tag{4.25}$$

Define a function ψ by

$$\psi = \left[(x - x^2)\sin(\pi x)(y - y^2)\sin(\pi y) - \frac{16}{\pi^6}\right]\cos t. \tag{4.26}$$

The pressure then is

$$p(x, y, t) = -\frac{\partial\psi}{\partial t} + \nu\Delta\psi. \tag{4.27}$$

The force term $\mathbf{G}$ is determined from the exact solution. The domain is the unit square $\Omega = [0, 1] \times [0, 1]$. In this example, we have $\frac{\partial p}{\partial \mathbf{n}} \neq 0$ on the boundary. Thus, the numerical boundary layer might deteriorate the accuracy of the pressure.

Table I shows the grid refinement analysis at $t = 10$. Throughout this paper, the error computed in the maximum norm is up to the boundary, meaning that the error at the boundary is also taken into account. The order of accuracy is defined as

$$\text{order} = \frac{\log[E(N)/E(2N)]}{\log 2}. \tag{4.28}$$

The pressure has been adjusted by some constant that minimizes the error in the maximum norm. We can see the velocity is second-order accurate, but the pressure is first order using the original projection method. However, if we add the correction term in (3.23), then the pressure is very close to second-order. The last column in Table I is the errors of the pressure with the correction term in (3.23). While it is not fully second-order accurate, it is better than the original projection method without the correction. Away from the boundary, the pressure is second-order accurate.

TABLE I
The Grid Refinement Analysis of the Projection Method with μ = 1, Δt = h, for Example 1 at t = 10

	Projection method without (3.23)				With (3.23)	
N	$\|E_1(\mathbf{u})\|_\infty$	Order	$\|E_1(p)\|_\infty$	Order	$\|E_2(p)\|_\infty$	Order
32	2.7021×10^{-4}		0.0608		3.2871×10^{-2}	
64	8.3429×10^{-5}	1.6955	0.0269	1.1785	9.2432×10^{-3}	1.8303
128	2.2471×10^{-5}	1.8925	0.0109	1.2975	3.2825×10^{-3}	1.4936
256	5.7841×10^{-6}	1.9580	0.0045	1.2665	1.1341×10^{-4}	1.5333

Note. The second column $\|E_1(\mathbf{u})\|_\infty$, and the fourth column $\|E_1(p)\|_\infty$ are the errors of the velocity and pressure using the projection method without correction in (3.23), respectively. While the velocity is nearly second order accurate, the pressure is roughly first order. The sixth column $\|E_2(p)\|_\infty$ is the error for the pressure using the projection method with correction in (3.23). While it is not entirely second order, the accuracy of the pressure has been improved. The velocity (not listed) remains second-order accurate with smaller errors.

In practice, it is difficult to get fully second-order accurate results for pressure for some problems using projection methods with a single Cartesian grid. It may be due to the difficulty in dealing with the numerical boundary layer and the four corners of the rectangular domain. We did notice that, however, the full second-order accuracy was achieved for pressure in [2] for the channel flow and a problem with homogeneous boundary conditions. We think it may be possible to achieve full second-order accuracy for pressure if a suitable extrapolation scheme is used to evaluate the correction term at the boundary, particularly, at the four corner points. Usually, it is more accurate to use the MAC grid than to use a single Cartesian grid. So we believe that it is possible to achieve full second-order accuracy using the MAC grid. The trade-off is the increased complexity in the implementation.

4.2. *Example 2: An Interface Problem with a Constant Jump in the Pressure*

In this example, we consider a fixed interface,

$$\frac{(x-0.5)^2}{a^2} + \frac{(y-0.5)^2}{b^2} = 1, \tag{4.29}$$

in the domain $[0, 1] \times [0, 1]$. The normal and tangential force strength are $\hat{f}_1 = 10$ and $\hat{f}_2 = 0$, respectively. We also set $G_1 = 0$ and $G_2 = 0$. The solution now is given by

$$u = 0, \quad v = 0, \quad p = \begin{cases} C, & \text{if } \frac{(x-0.5)^2}{a^2} + \frac{(y-0.5)^2}{b^2} - 1 > 0, \\ -10 + C, & \text{if } \frac{(x-0.5)^2}{a^2} + \frac{(y-0.5)^2}{b^2} - 1 \le 0, \end{cases} \tag{4.30}$$

where C is an arbitrary constant. Here, we compare our method with Peskin's IB method using the discrete cosine delta function. Depending on how the interface is represented, there are two different versions of the IB method.

4.2.1. The Lagrangian particle approach. In this traditional approach, the interface is represented by a set of Lagrangian points $\mathbf{X}_k$, $k = 1, 2, \ldots, N_b$, where N_b is the total number of the particle points and is usually chosen in such a way that

$$\max \Delta s_k = \max \|\mathbf{X}_{k+1} - \mathbf{X}_k\| \sim h.$$

The forces defined on those particle points are then distributed to the nearby grid points (x, y) by the formula

$$\sum_{k=1}^{N_b} \mathbf{f}(s_k)\delta_w(x - X_k)\delta_w(y - Y_k)\Delta s_k,$$

where δ_w is the one-dimensional discrete delta function

$$\delta_w(x) = \begin{cases} \frac{1}{4w}(1 + \cos(\pi x/2w)), & \text{if } |x| < 2w, \\ 0, & \text{if } |x| \geq 2w. \end{cases} \tag{4.31}$$

Although there are other discrete delta functions in the literature, the computational results are not much different for two- and three-dimensional problems. The most common choices of w is h, the spatial mesh size. As we can see, when w gets larger, the cost to spread the singular force to the grid points increases significantly. Our numerical tests for this and other examples show that the accuracy of the computed velocity remains pretty much the same for different choices of w. However, the pressure, if it is discontinuous, smears out more widely as w increases. Therefore, the best choice of w is h. The sixth column in Table II is the grid refinement analysis for the velocity which is clearly first-order accurate.

4.2.2. The level set representation. If the interface is represented by the zero level set of a two-dimensional function $\varphi(x, y)$ as in (4.40), it is not straightforward to use the IB method directly. However, if the tangential component of the singular force is zero, then we can write

$$\int_{\Gamma(t)} \mathbf{f}(s)\delta_2(\mathbf{x} - \mathbf{X}(s))\, ds = \mathbf{f} \cdot \mathbf{n}\delta(\varphi)\nabla\varphi; \tag{4.32}$$

see [3] and others. It is easy to apply the discrete delta function in the above form.

TABLE II
The Grid Refinement Analysis and Comparison of the Three Projection-IB Methods for Example 2 with μ = 0.1, a = 0.35, b = 0.25, and Δt = h

	Level Set IB width $w = h$		Level Set IB width $w = \sqrt{h}$		Lagrangian IB width $w = h$	
N	$\lVert E(\mathbf{u})\rVert_\infty$	Order	$\lVert E(\mathbf{u})\rVert_\infty$	Order	$\lVert E(\mathbf{u})\rVert_\infty$	Order
32	1.2434×10^{-1}		4.9254×10^{-3}		5.1204×10^{-2}	
64	6.3619×10^{-2}	0.96673	2.9594×10^{-3}	0.7349	2.5839×10^{-2}	0.9867
128	4.0730×10^{-2}	0.64339	1.1062×10^{-3}	1.4197	1.2968×10^{-2}	0.9929
256	4.3059×10^{-2}	0.09459	4.1558×10^{-4}	1.4124	6.5055×10^{-3}	0.9952

Note. The initial data is taken from the solution in Example 1. The second and fourth columns are the error of the computed velocity using the level set function with width w being h and $\sqrt{h}$, respectively. The sixth column is the error of the computed velocity using the original IB method with w being h. Generally, the velocity is first-order accurate. The pressure which is not listed in the table has zeroth order accuracy because it is smeared out.

Unfortunately, the choice of the width of the discrete delta function is crucial for the IB method. If $w = h$, the IB method barely converges; see the second column of Table II. T. Hou [13] may be the first one to note this and suggest that the best choice of the width is $w = \sqrt{h}$. This has also been confirmed in our numerical tests; see the fourth column of Table II. However, $w = \sqrt{h}$ is much larger than h if h is small. The IB method smears out the pressure and shows zero-order convergence for the pressure. Since the width has to be taken as $w = \sqrt{h}$, we cannot guarantee the right pressure even at those grid points that are away from the interface.

A remedy to the problem is to use the projections of irregular grid points of a particular side, say $\varphi(x, y) \leq 0$, on the interface as the control points. Then the Peskin's original IB method can be applied with those control points. In [14, 28], we described in detail how to find the projection of an irregular grid point on the interface. Using this new approach, we still can use a thin layer $(w = h)$ of the interface in the IB method, so that the solution will only be smeared out in the thin layer while we can keep the advantages of the level set method.

4.2.3. The grid refinement analysis and comparison. Table II lists the comparison of three different versions of the IB method using the particle approach and the level set function. We have also tested the example with different but modest μ, different major axis a and minor axis b, and different initial conditions. All the results are qualitatively similar. In Table III, we show the grid refinement analysis of the projection-IIM method developed in this paper. Without the modification of the pressure equation (3.23), the velocity in the second column is second-order accurate, while the pressure in the fourth column is first-order accurate with the largest errors occurring near the boundary or the interface. With the correction term in (3.23) added, we see second-order accuracy for both the velocity and the pressure. Since the pressure is piecewise constant in this case, the accuracy for the pressure is much better than second order. However, this is not true for general problems.

In a further numerical experiment, we revisit Example 1 but add a singular force term $\hat{f}_1 = 10$ to the Navier–Stokes equations in (1.1). The exact solution for the velocity is the same

TABLE III
The Grid Refinement Analysis of Projection-IIM Method for Example 2. The Parameters are μ = 0.1, a = 0.35, b = 0.25, and Δt = h

	Projection IIM without (3.23)				Projection IIM with (3.23)	
N	$\Vert E(\mathbf{u})\Vert_\infty$	Order	$\Vert E(p)\Vert_\infty$	Order	$\Vert E(p)\Vert_\infty$	Order
32	1.6412×10^{-4}		1.5785×10^{-2}		9.9058×10^{-3}	
64	3.9857×10^{-5}	2.0418	7.9281×10^{-3}	0.9935	1.1533×10^{-3}	3.1026
128	9.8987×10^{-6}	2.0095	3.9676×10^{-3}	0.9987	1.5946×10^{-4}	2.8544
256	1.2367×10^{-6}	2.0008	1.9842×10^{-3}	0.9997	2.4832×10^{-5}	2.6830

Note. The first and third columns are the errors of the computed velocity and the pressure without the correction term in (3.23), respectively. We can clearly see that the velocity is second-order accurate while the pressure is first-order because of the boundary layer. The sixth column is the error of the computed pressure using the same method but with the correction term for the pressure in (3.23). The accuracy actually is better than second-order because of the fact that the pressure is piecewise constant. The computed velocity (not shown in this table) with the correction is also more accurate meaning a smaller error constant.

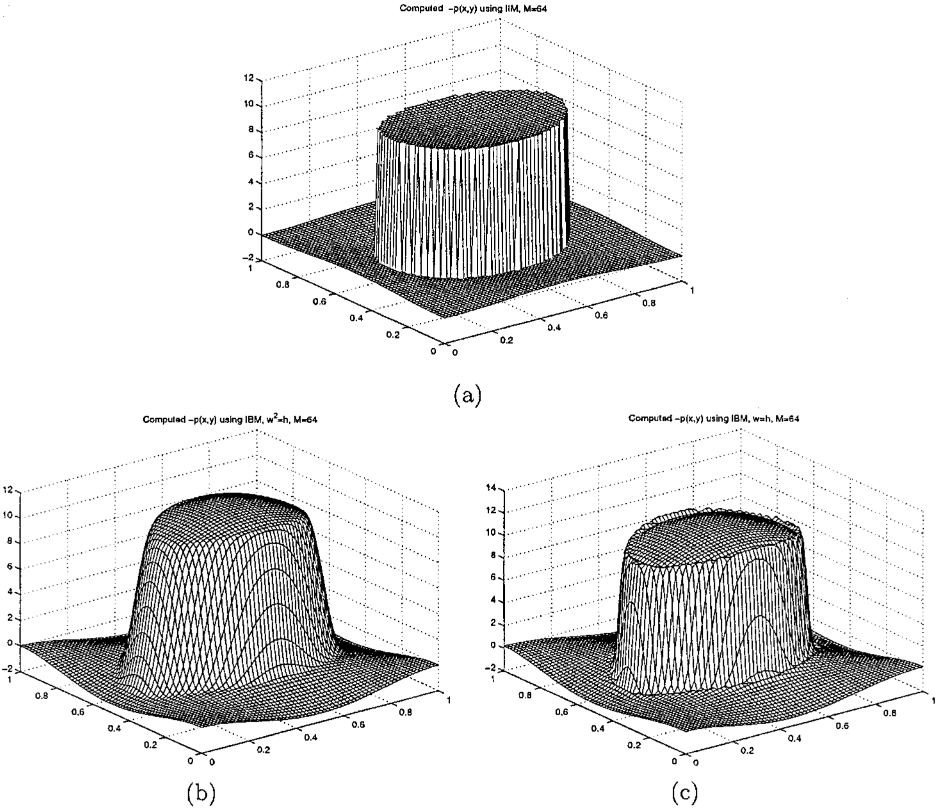

FIG. 2. Computed pressure for Example 1 with an added force term $\hat{f}_1 = 10$ using a 64 by 64 grid. (a) The result computed using the projection-IIM method which catches the jump in the pressure. (b) The result computed using the IB method with the interface represented by a level set function $(w = \sqrt{h})$. The width of the discrete delta function is $w = \sqrt{h}$. (c) The result computed using the original IB method $(w = h)$. The error for the pressure is $O(1)$ for both (b) and (c).

as in Example 1, but the pressure is the sum of the pressure in Example 1 and the piecewise constant in (4.30). The analysis and observations are similar to the discussions above. Figure 2a shows the pressure computed using the projection-IIM method. The computed pressure has the right jump up to second order. Figures 2b and 2c show the pressure computed using the IB method with the level set representation of the interface $(w = \sqrt{h})$ and the original IB method with particle representation of the interface $(w = h)$, respectively. In Fig. 2b, we see that the pressure is smeared out by the width $w = \sqrt{h}$ while in Fig. 2c, we see larger errors but a thinner layer $(w = h)$ near the interface.

For this test problem, our projection-IIM method is faster compared with the IB method. The reason, we believe, is the triple loop of the IB method to spread the forces to nearby grid points.

4.3. *Example 3: An Interface Problem with Nonconstant Jump in the Pressure*

In this example, we want to check how well our method can handle a nonconstant jump in the pressure and a finite jump in **G**. The interface Γ and the domain Ω are the same as in Example 2.

The source term $\mathbf{G}$ is the gradient of a function $q(x, y)$, which is the solution of the following

$$\Delta q = 0, \quad (x, y) \in \Omega, \qquad \left.\frac{\partial q}{\partial \mathbf{n}}\right|_{\partial\Omega} = 0, \tag{4.33}$$

$$[q]|_\Gamma = x + y, \qquad \left.\left[\frac{\partial q}{\partial \mathbf{n}}\right]\right|_\Gamma = 0. \quad (x, y) \in \Gamma. \tag{4.34}$$

The function $q(x, y)$ and $\mathbf{G} = \nabla q$ are computed using the IIM method [25] with a 512 by 512 fine grid. Note that from $[\frac{\partial q}{\partial \mathbf{n}}] = 0$ and $[q] = x + y$, it is easy to derive

$$[G_1] = [q_x] = \sin^2\theta - \sin\theta\cos\theta, \quad [G_2] = [q_y] = \cos^2\theta - \sin\theta\cos\theta,$$

where $\mathbf{n} = (\cos\theta,\ \sin\theta)$ is the unit normal direction of Γ pointing outward.

The full Navier–Stokes equations of Example 3 are (1.1)–(1.5) with the following parameters

$$\mathbf{u}|_{\partial\Omega} = \mathbf{0}, \quad \mu = 1, \quad \mathbf{G} = \nabla q(x, y),$$
$$\hat{f}_2 = 0, \quad \hat{f}_1 = x + y, \quad (x, y) \in \Gamma,$$

with initial conditions for $\mathbf{u}$ and p.

For this example, we do not know the exact solutions. However, the steady state solution for this example is

$$\lim_{t\to\infty} \mathbf{u}(\mathbf{x}, t) = \mathbf{0}, \quad \lim_{t\to\infty} p(\mathbf{x}, t) = q(\mathbf{x}).$$

In Table IV, we show the grid refinement analysis for the *steady state solution* obtained using the projection-IIM method at $t = 10$. The initial velocity is taken from (4.25) in Example 1, while the pressure is

$$p_0(x, y) = \cos\pi x\ \cos\pi y + q(x, y).$$

In this way, we guarantee the initial pressure has the right jump, $[p] = \hat{f}_1$. We see that the projection-IIM method gives a second-order accurate solution for both the velocity and the pressure. The results are qualitatively the same if different initial conditions are used, as

TABLE IV
The Grid Refinement Analysis of Projection-IIM Applied to Example 3 with μ = 1, Δt = h, at t = 10

N	$\|E_{IIM}(\mathbf{u})\|_\infty$	Order	$\|E_{IIM}(p)\|_\infty$	Order
32	2.4930×10^{-5}		5.1614×10^{-4}	
64	3.5197×10^{-6}	2.8244	8.4365×10^{-5}	2.6131
128	4.6660×10^{-7}	2.9152	1.6308×10^{-5}	2.3710
256	3.5073×10^{-8}	2.9332	5.7023×10^{-6}	1.5160

Note. The initial velocity is taken from (4.25) in Example 1. The initial pressure is taken as $\cos\pi x\ \cos\pi y + q(x, y)$.

TABLE V
The Grid Refinement Analysis of Projection-IIM Applied to Example 4 with μ = 0.02, Δt = h, at t = 10; Second-Order Convergence Is Achieved

N	$\|E_{IIM}(\mathbf{u})\|_\infty$	Order	$\|E_{IIM}(p)\|_\infty$	Order
32	2.4215×10^{-3}		1.1513×10^{-2}	
64	5.3547×10^{-4}	2.1771	3.2255×10^{-3}	1.8357
128	1.4970×10^{-4}	1.8388	9.1307×10^{-4}	1.8357
256	3.6173×10^{-5}	2.0491	1.9727×10^{-4}	2.2106

long as the jump condition in p is initially satisfied. Note that $\cos \pi x \ \cos \pi y$ satisfies the homogeneous Neumann boundary condition $\nabla p \cdot \mathbf{n} = 0$ on the boundary. If we take the initial pressure from (4.27) in Example 1, then the computed pressure is only first-order accurate since $[p] \neq \hat{f}_1$ initially while the velocity is still second-order accurate.

4.4. *Example 4: Circular Flow with a Line Force*

Now we consider an example with nonsmooth velocity. The exact solution is

$$u(x, y, t) = \begin{cases} h(t)\left(\frac{y}{r} - 2y\right) & \text{if } r > \frac{1}{2}, \\ 0 & \quad r \leq \frac{1}{2}, \end{cases} \tag{4.35}$$

$$v(x, y, t) = \begin{cases} h(t)\left(-\frac{x}{r} + 2x\right) & \text{if } r > \frac{1}{2}, \\ 0 & \quad r \leq \frac{1}{2}, \end{cases} \tag{4.36}$$

$$p(x, y, t) = 0, \tag{4.37}$$

where $r = \sqrt{x^2 + y^2}$. The interface is the circle $r = \frac{1}{2}$ and the solution domain is $[-1, 1] \times [-1, 1]$. It is easy to verify that the velocity satisfies the incompressibility constraint, and it is continuous but has a finite jump in the normal direction

$$[u_n] = [u_r] = -2h(t) \sin \theta, \quad [v_n] = [v_r] = 2h(t) \cos \theta,$$

across the interface. Thus, the normal and tangential force strength are

$$\hat{f}_1 = 0, \quad \hat{f}_2 = -2h(t).$$

Note that, $\mathbf{u}_t \cdot \mathbf{n}$ is not zero on the boundary if $\frac{dh}{dt} \neq 0$, which may affect the performance of the projection method.[2] Outside of the interface $r = 0.5$, the nonzero and bounded source term $\mathbf{G}$ is derived directly from the exact solution using Maple. There is also a finite jump in $\mathbf{G}$.

With $h(t) = 1$, our method converges to the steady state solution. The grid refinement analysis reveals similar results as in Example 3. In our time-dependent test, we take $h(t) = 1 - e^{-t}$. We present the numerical results at $t = 10$ in Table V and plot the x-component

[2] It seems that the projection method used here requires $\mathbf{u}_t \cdot \mathbf{n}$ to be zero along the boundary. Numerical tests showed that if $\mathbf{u}_t \cdot \mathbf{n}$ is not zero, then the velocity still converges but not with second-order accuracy; the pressure may not converge at all for the projection method if $\mathbf{u}_t$ is large on the boundary. However, some other projection methods may work even if the condition $\mathbf{u}_t \cdot \mathbf{n} = 0$ is violated.

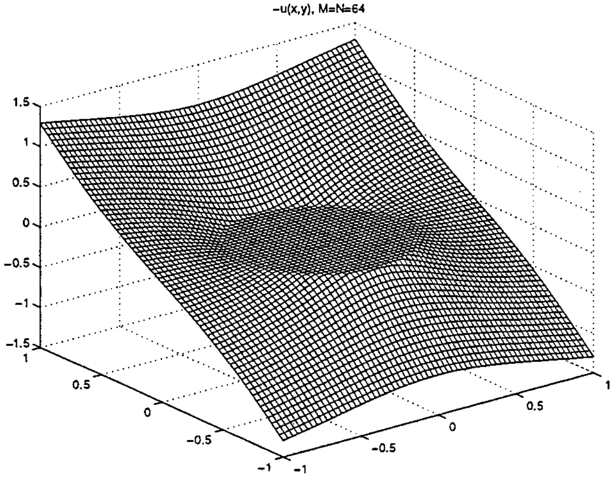

FIG. 3. Plot of the x-component of the velocity $-u(x, y)$ of Example 4 at $t = 10$. The results are computed with a 64×64 grid and $\mu = 0.02$. The time step size is $\Delta t = h$.

of the velocity in Fig. 3. One can easily see that the velocity is second-order accurate, and the pressure is nearly second-order accurate in Table V.

We also tested a slightly different example by setting

$$\mathbf{G} = \mathbf{0}, \quad \hat{f}_1 = 0, \quad \hat{f}_2 = 10 \quad \mathbf{u}|_{\partial\Omega} = \mathbf{0}.$$

The domain and the interface are the same. The initial velocity and the pressure are taken to be zero on the rectangular domain. Since the force is only along the tangential direction, the pressure is continuous, but the normal derivative of the velocity has a nonconstant jump across the interface. While the exact solution is difficult to find, the motion of the steady state is nothing but a simple rotation along the interface. In Fig. 4a, we plot the x-component

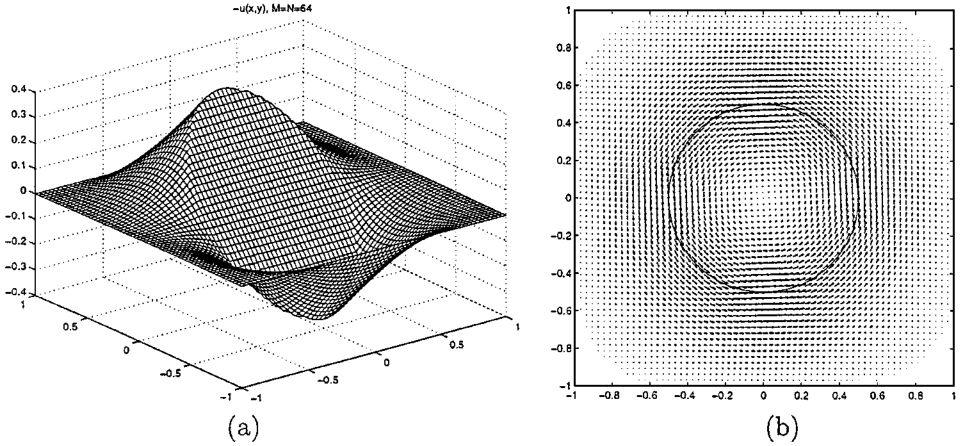

FIG. 4. The computed steady state velocity for the modified example in Example 4 with a circular interface $r = 0.5$ with a 64×64 grid. Other parameters are $\mathbf{G} = \mathbf{0}$, $\hat{f}_1 = 0$, $\hat{f}_2 = 10$, and $\mu = 0.02$. The time step size is $\Delta t = h$. (a) Plot of $-u(x, y)$ at $t = 10$ which is not smooth. (b) The plot of the velocity field at $t = 10$, the flow approaches a steady rotation along the interface.

of the velocity $-u(x, y)$ at $t = 10$ computed using a 64 by 64 grid. We can clearly see the nonconstant jump in the normal derivative. In Fig. 4b, we plot the velocity field at $t = 10$. The flow approaches a steady rotation along the circular interface.

4.5. *Example 5: A Moving Interface Problem*

Now we consider a moving interface problem which has uniform density and viscosity. The initial interface is given by

$$r(\theta) = r_0 + \epsilon \sin(k\theta), \quad 0 \le \theta \le 2\pi. \tag{4.38}$$

The initial velocity and the pressure are all set to be zero. The only driven force of the fluid motion is the surface tension which is proportional to the curvature κ, that is,

$$\hat{f}_1(\Gamma, t) = \epsilon\kappa, \quad \hat{f}_2(\Gamma, t) = 0. \tag{4.39}$$

The velocity is smooth but the pressure has a nonconstant jump

$$[p]|_{\Gamma(t)} = \epsilon\kappa, \quad [p_n]\,|_{\Gamma(t)} = 0.$$

In our test, we take $r_0 = 0.5$, $k = 5$, $\epsilon = 0.05$, and $\mu = 0.1$.

We use the level set method [31] to update the interface. We refer the readers to [14, 27, 28] for the details about how to combine the immersed interface method with the level set method. In the level set method, the interface is the zero level set of a two-dimensional function

$$\varphi(x, y, t) \begin{cases} <0, & \text{if } (x, y) \text{ is in the inside of the interface,} \\ =0, & \text{if } (x, y) \text{ is on the interface,} \\ >0, & \text{if } (x, y) \text{ is in the outside of interface.} \end{cases} \tag{4.40}$$

At each time step, we apply the projection-IIM method described in Section 3 to compute the velocity. Then the velocity is used to solve the Hamilton–Jacobian equation

$$\varphi_t + \vec{u} \cdot \dot{\nabla}\, \varphi = 0 \tag{4.41}$$

to obtain the new level set function at $t = t^{n+1}$. Since we use the level set method to update the interface, the time step is chosen as

$$\Delta t = \min\left\{ \frac{h}{2\|\mathbf{u}\|_\infty}, h \right\} \tag{4.42}$$

to maintain the stability for the level set method and the accuracy for the projection-IIM method.

In Fig. 5, we plotted the computed interface at different times with a 160 by 160 grid. Using the immersed interface method, we know the intersections of the interface and the grid lines. The set of intersections, which is on the interface at a particular time, is

An animation of the evolution of the interface in this example can be found at http://www4.ncsu.edu/˜zhilin/research.html.

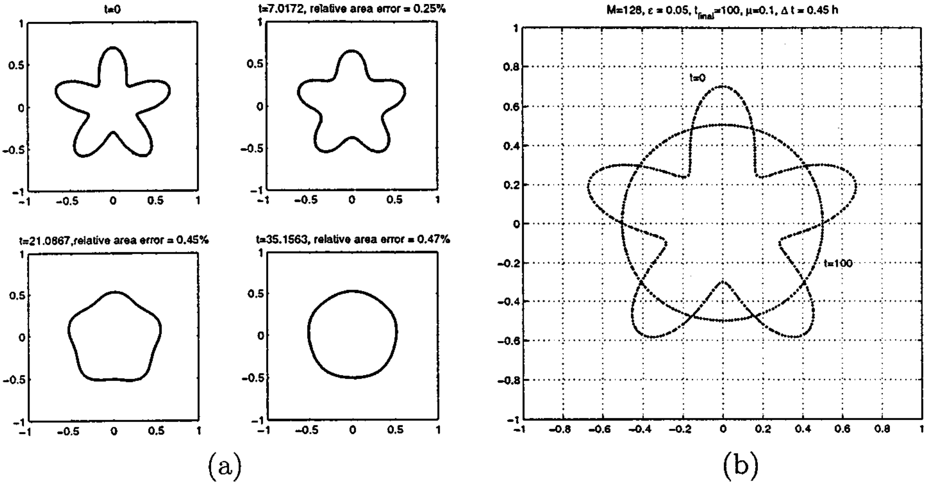

FIG. 5. The numerical evolution of the moving interface for Example 5 using the projection-IIM method coupled with the level set formulation. The parameters are: $\mu = 0.1, \epsilon = 0.05$, $M = N = 160$, and $\Delta t = \min\{\frac{h}{2\|\mathbf{u}\|_\infty}, h\}$. Because of the surface tension, the interface is relaxing to a circle. (a) The computed interface at $t = 0,\ t = 7.0172,\ t = 21.0867$, and $t = 35.1563$. (b) The initial interface $t = 0$ and the computed interface at $t = 100$.

one dimensional. Therefore we can afford to store the interface at many different time levels. In Fig. 5a, we plotted the interface at $t = 0, t = 7.0172, t = 21.0867$, and $t = 35.1563$. The relative error in the area is 0.25% at $t = 7.0172$, 0.45% at $t = 21.0867$, and 0.47% at $t = 35.1563$. In Fig. 5b, we plotted the interface at $t = 0$, and $t = 100$, which is almost in the equilibrium state, a circle. The relative area change (loss) at $t = 100$ is about 1.61%. Note that, in the equilibrium state, the velocity approaches zero everywhere and the pressure approaches two different constants inside and outside the interface.

In Table VI, we perform the grid refinement analysis at time $t = 15$. Since the exact solution is not available, we calculate the error between the computed solution and the one obtained by the finest grid, 512 by 512. We denote such error as $\tilde{E}_N$, for example, $\tilde{E}_N(p) = \|p_N - p_{512}\|_\infty$. For a second-order method, the error ratio $\tilde{E}_N/\tilde{E}_{2N}$ is close to 4 if N is much smaller than 512, the finest number of the grid lines. The ratio approaches 5 as N is getting closer to the finest number of the grid lines; see [22] for the methodology. From Table VI, we can see that our method behaves like a second-order method and agrees with the results of other examples in this section.

TABLE VI
The Grid Refinement Analysis of Projection-IIM Applied to Example 5 at *t* = 15

N	$\|\tilde{E}_N(\mathbf{u})\|_\infty$	$\cdot\|\tilde{E}_{N/2}(\mathbf{u})\|_\infty/\|\tilde{E}_N(\mathbf{u})\|_\infty$	$\|\tilde{E}_N(p)\|_\infty$	$\|\tilde{E}_{N/2}(p)\|_\infty/\|\tilde{E}_N(p)\|_\infty$
32	1.8080×10^{-3}		0.2427	
64	4.5062×10^{-4}	4.0123	0.0717	3.3849
128	1.1920×10^{-4}	3.7804	0.0181	3.9613
256	2.4400×10^{-5}	4.8852	4.2500×10^{-3}	4.2588

Note. The error is computed against the result obtained using the finest grid 512 by 512.

5. CONCLUSIONS

We have developed an immersed interface method coupled with the projection method for the full incompressible Navier–Stokes equations that involve a singular force along one or several interfaces. We derived some useful interface relations that are used to determine the correction terms to the original projection method. The new projection-IIM method has the same stability nature as the original projection method since only the right-hand sides in the projection method are modified. The numerical results show second-order convergence for the projection-IIM method. The computed solutions remain sharp across the interface since the jump conditions are enforced.

We also propose a new approach of the Immersed Boundary Method when the interface is represented by a level set function. Instead of using the discrete delta function directly for the level set function with larger width $\sqrt{h}$, we use the projections of irregular grid points on the interface as the control points, and then use the original Immersed Boundary Method with the width being $w = h$.

Research is under way for the Navier–Stokes equations that involves not only a singular force along one or several interfaces but also discontinuities in the density and the viscosity. The success of the new method depends on whether we can derive the jump conditions for the velocity and the pressure.

ACKNOWLEDGMENTS

The first author is partially supported by the Army Research Office under Grant 39676-MA, and the NSF under Grant DMS0073403. The second author was supported by National Science Council of Taiwan under Research Grants NSC-89-2119-M-194-001 and NSC-89-2115-M-194-027. The authors thank Dr. C. Wang for the original projection code, and are also grateful to Dr. R. LeVeque and Dr. M. Minion for reading and commenting on the original manuscript. Finally, the authors thank the referees for their useful comments, corrections, and suggestions.

REFERENCES

1. J. B. Bell, P. Colella, and H. M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85**, 257 (1989).
2. D. L. Brown, R. Cortez, and M. L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **168**, 464 (2001).
3. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, A level set formulation of Eulerian interface capturing method for incompressible fluid flows, *J. Comput. Phys.* **124**, 449–464, 1996.
4. L.-T. Cheng, R. Fedkiw, F. Gibou, and M. Kang, A symmetric method for implicit time discretization of the Stefan problem, UCLA CAM Report 00-37 (2000).
5. R. Cortez and M. L. Minion, The blob projection method for immersed boundary problems, *J. Comput. Phys.* **161**, 428 (2000).
6. R. H. Dillon, L. J. Fauci, and A. L. Fogelson, Modeling biofilm processes using the immersed boundary method, *J. Comput. Phys.* **129**, 57 (1996).
7. W. E. and J. Liu, Projection method: I. convergence and numerical boundary-layers, *SIAM J. Numer. Anal.* **32**, 1017 (1995).
8. L. J. Fauci, Interaction of oscillating filaments—A computational study, *J. Comput. Phys.* **86**, 294 (1990).
9. A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, *J. Comput. Phys.* **56**, 111 (1984).
10. A. L. Fogelson, Continuum models for platelet aggregation: Formulation and mechanical properties, *SIAM J. Numer. Anal.* **52**, 1089 (1992).
11. B. T. Helebrook and C. K. Law, The role of Landau-Darrieus instability in large scale flows, *Combust. Flame* **117**, 155 (1999).

12. B. T. Helebrook, L. Martinelli, and C. K. Law, A numerical method for solving incompressible flow problems with a surface of discontinuity, *J. Comput. Phys.* **148**, 366 (1999).
13. T. Hou, Private communications (2000).
14. T. Hou, Z. Li, S. Osher, and H. Zhao, A hybrid method for moving interface problems with application to the Hele-Shaw flow, *J. Comput. Phys.* **134**, 236 (1997).
15. J. Van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Comput.* **7**, 870 (1986).
16. M. Kang, R. Fedkiw, and X. Liu, A boundary condition capturing method for multiphase incompressible flow, *SIAM J. Sci. Comput*, in press, Also, UCLA CAM Report 99-27 (1999).
17. J. Kim and P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, *J. Comput. Phys.* **59**, 308 (1985).
18. M. Lai and C. S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, *J. Comput. Phys.* **160**, 705 (2000).
19. M.-C. Lai and Z. Li, A remark on jump conditions for the three-dimensional Navier–Stokes equations involving an immersed moving membrane, *Applied Math. Lett.* **14**, 149 (2001).
20. R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, 1019 (1994).
21. R. J. LeVeque and Z. Li, Immersed interface method for Stokes flow with elastic boundaries or surface tension, *SIAM J. Sci. Comput.* **18**, 709 (1997).
22. Z. Li, *The Immersed Interface Method—A Numerical Approach for Partial Differential Equations with Interfaces*, Ph.D. thesis (University of Washington, 1994).
23. Z. Li, A note on immersed interface methods for three dimensional elliptic equations, *Computers Math. Appl.* **31**, 9 (1996).
24. Z. Li, Immersed interface method for moving interface problems, *Numer. Algorithms* **14**, 269 (1997).
25. Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM J. Numer. Anal.* **35**, 230 (1998).
26. Z. Li and K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, *SIAM J. Sci. Comput.* in press.
27. Z. Li and B. Soni, Fast and accurate numerical approaches for Stefan problems and crystal growth, *Numer. Heat Trans. B: Fundamentals* **35**, 461 (1999).
28. Z. Li, H. Zhao, and H. Gao, A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid, *J. Comput. Phys.* **152**, 281 (1999).
29. X. Liu, R. Fedkiw, and M. Kang, A boundary condition capturing method for Poisson's equation on irregular domain, *J. comput. Phys.* **160**, 151 (2000).
30. D. Nguyen, R. Fedkiw, and M. Kang, *A Boundary Condition Capturing Method for Incompressible Flame Discontinuities*, UCLA CAM Report 00-19 (2000).
31. S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
32. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
33. C. S. Peskin, Lectures on mathematical aspects of physiology, *Lect. Appl. Math.* **19**, 69 (1981).
34. C. S. Peskin and D. M. McQueen, Modeling prosthetic heart valves for numerical analysis of blood flow in the heart, *J. Comput. Phys.* **37**, 113 (1980).
35. C. S. Peskin and D. M. McQueen, A general method for the computer simulation of biological systems interacting with fluids, *Symposia of the Society for Experimental Biology* **49**, 265 (1995).
36. E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* **130**, 269 (1997).
37. J. Shen, On error estimates of the projection methods for the Navier–Stokes equations: second-order scheme, *Math. Comp.* **65**, 1039 (1996).
38. D. Sulsky and J. U. Brackbill, A numerical method for suspension flow, *J. Comput. Phys.* **96**, 339 (1991).
39. R. Teman, Remark on the pressure boundary condition for the projection method, *Theor. Comput. Fluid Dyn.* **3**, 181 (1991).
40. C. Wang and J. Liu, Analysis of finite difference schemes for unsteady Navier–Stokes equations in vorticity formulation, preprint (2000).