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

Zhilin Li
Center for Research in Scientific Computation & Department of Mathematics
North Carolina State University, Raleigh, NC 27695-8205, USA
zhilin@math.nscu.edu

Ming-Chih Lai
Department of Mathematics
Chung Cheng University, Minghsiung, Chiayi 621, Taiwan
mclai@math.ccu.edu.tw

keywords: Navier-Stokes equations, interface, discontinuous and non-smooth solution, immersed interface method, immersed boundary method, projection method, level set method.

AMS Classification; 65M06, 65M12, 76T05

Abstract

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. Non-trivial numerical results including one with moving interface 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.

1 Introduction

In this paper, we consider the incompressible Navier-Stokes equations in a bounded domain Ω 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 $\mathbf{F}$ and $\mathbf{G}$ separately to distinguish different irregularities. The singular force $\mathbf{F}$ has support only on the immersed interface $\Gamma(t)$ and is written by

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

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, $\delta_2(\mathbf{x}) = \delta_1(x)\delta_1(y)$ with $\mathbf{x} = (x,y)$. The term $\mathbf{G}$ 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 continuous. Fig. 1 is an illustration of the geometry and the local coordinates for the problems discussed in this paper.

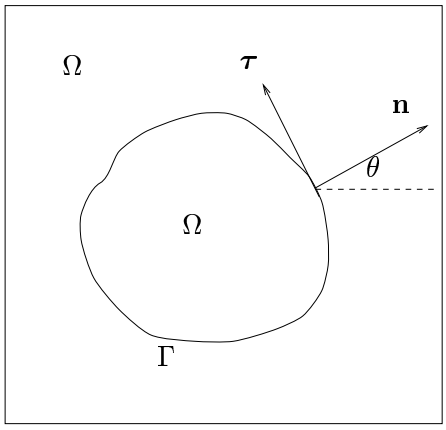


Figure 1: A diagram of the geometry for the interface problems discussed in this paper. We use $\mathbf{n}$ and τ to denote the unit normal and tangent directions of the interface, respectively, and θ is the angle between the normal direction and the x-axis.

Although we assume that the density and the velocity are continuous, the model (1.1)-(1.5) has many applications, see for example, cardiac blood flow [23, 24, 25], platelet aggregation

and coagulation [7, 8], swimming microorganisms [6], biofilm formation [4], suspension flows [29], and others, see also [26] for a brief review.

In 1970-80's, Peskin [23, 24] developed the Immersed Boundary Method (IBM) 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 at the same time it moves along with the fluid. Peskin's IBM 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 IBM method is only first order accurate for non-smooth but continuous quantities such as a velocity field. Recently Lai and Peskin [12] proposed a new *formally* second order IBM method with reduced numerical viscosity and applied the new method to the flow around a circular cylinder. The numerical result has a good agreement with the experimental data. Nevertheless, the *fully* second order accuracy has not been achieved in this new scheme. The large errors usually appear near the interface due to the singular force, see Fig. 2 (b) and Fig. 2 (c) in Section 4, for example.

The immersed interface method (IIM) [14, 16, 17, 18] is intended to improve the accuracy of the IBM 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 [19] and has been applied to the Stokes flow [15] and other moving interface/free boundary problems [10, 21, 20]. 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 IBM method.

The method developed in this paper is for the *sharp interface models*. There are some other models and methods for interface problems. Noticeably, the *phase field model* and the finite volume method, for example, [27]. Each model has advantages and disadvantages, and may has 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 derive 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 non-trivial examples including one with moving interface are presented and analyzed. A new version of the IBM method with the level set formulation is also proposed there. Some conclusions will be given in Section 5.

2 Jump Conditions Across the Interface

Due to the singular forces, the velocity of the solution to (1.1)-(1.5) is typically non-smooth, 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 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 following jump conditions:*

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

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

where $\boldsymbol{\tau} = (-\sin\theta, \cos\theta)$ 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 [13, 15] 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 using 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, for example, the flux, are described in terms of 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 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).

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

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 the outline of the proof. The readers who wish to get the full and detailed proof can contact the authors.

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. Differentiating $[\mathbf{u}] = \mathbf{0}$ along the interface twice and using $\chi'(0) = 0$, and $\chi''(0) = \kappa$ we get the sixth equality. Differentiating the fourth equality along the interface and using $\frac{\partial \boldsymbol{\tau}}{\partial \eta} = \kappa \mathbf{n}$, we get the seventh equality[1]. Finally, 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

We use the projection method developed by Bell, Colella, and Glaz [1] to solve the full Navier-Stokes equations. We will call their method as BCG method in this paper. The key modification to the BCG projection method is to add some correction terms at grid points near 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. A standard uniform grid is used but the scheme discussed in this section can be generated to the MAC grid without substantial difficulty. Our method

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

from time t^n to t^{n+1} can be written as:

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

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:

$$\Delta_h \left(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}} \right) = \frac{\nabla_h \cdot \mathbf{u}^*}{\Delta t} + C_3^n, \tag{3.14}$$

$$\left. \frac{\partial \left(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}} \right)}{\partial \mathbf{n}} \right|_{\partial\Omega} = 0, \tag{3.15}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \left(p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}} \right) + \mathbf{C}_4^n. \tag{3.16}$$

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

$$\begin{aligned} \nabla_h u_{ij} &= \left(\frac{u_{i+1,j} - u_{i-1,j}}{2h}, \frac{u_{i,j+1} - u_{i,j-1}}{2h} \right) \\ \Delta_h u_{ij} &= \frac{u_{i-1,j} + u_{i,j-1} + u_{i+1,j} + u_{i,j+1} - 4u_{ij}}{h^2}. \end{aligned}$$

It is clear that the modifications that we made to the BCG method are those correction terms C_k^ns at grid points near the interface.

3.1 Determine 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 the equations (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 remains accurate. 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 theorem which is the basis for determining the correction terms.

Theorem 3 *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:*

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

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

where

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

and the jumps are defined as follows

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

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

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

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,

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

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

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

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

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

- *The correction terms given in the theorem are slightly different from the original immersed interface method [14, 16, 18]. We use Taylor expansion twice here so that we can approximate the derivatatives at* the master grid point *instead of a point on the interface as we used to do in the original immersed interface method. This new approach has smaller error constant and enables us to get "uniform" formulae for both first and second order derivatives, see below for the explanation.*

- *Theorem 3 tells us that the correction term is simply the product of the finite difference coefficient, corresponding to the grid point from different side of the interface in reference to the master grid point, and $C(x,\alpha)$. For example, if $-1<\alpha<0$, in Theorem 3, which means $x-h$ and x are on different side of the interface, then the correction terms for u_x is the product of the coefficinet $-1/(2h)$ and $C(x,\alpha)$. Therefore, it is very easy to modify the finite difference scheme to maintain 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,*

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

3.2 Correction terms to the BCG method

In our numerical scheme (3.12)-(3.16), there are several correction terms to be determined. For the sake of simplicity, we will just explain how to evaluate some of 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 detail 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 non-linear term $(\mathbf{u}^n\cdot\nabla_h)\,\mathbf{u}^n$:

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

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

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

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

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

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

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

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

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

We use the *bilinear interpolation* from the values of u and v at the neighboring four grid points to approximate the values of u and v at (x^*, y_j). 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 BCG 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. That will 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 for elliptic interface problem of second order convergence can be found in [19], see also the numerical experiments in Section 4.

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

3.3 Further correction near the boundary and the interface

Using the BCG method with the correction terms as described before, we have observed that the velocity is second order accurate but the pressure is only first order. The largest error of the pressure appears near the boundary and the interface. As analyzed in [2, 5, 28, 30], this is due to the inherent numerical boundary layer in the projection method. In [2], see also [11], a correction scheme is proposed which is

$$p^{n+\frac{1}{2}} = p^{n-\frac{1}{2}} + \phi^{n+1} - \Delta t\,\frac{\mu}{2}\left(\nabla_h \cdot \mathbf{u}^* + C_5\right), \tag{3.23}$$

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_5 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.

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.

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

4 Numerical Examples

All the calculations in this section were perfomed at North Carolina State University using Sun workstations. Most simulations are done within minutes or a couples of hours depending on the number of grid points. Generally, we use a level set function to represent the interface, see (4.39).

Example 1: A non-interface problem

We start our numerical tests by checking the accuracy of the BCG projection method for a regualr problem, and use the result as the basis for comparing different algorithms for the interface examples. The following exact solution is taken from [31].

$$\begin{aligned} u(x,y,t) &= -\sin^2(\pi x)\,\sin(2\pi y)\,\cos t, \\ v(x,y,t) &= \sin^2(\pi y)\,\sin(2\pi x)\,\cos t. \end{aligned} \tag{4.24}$$

Define a function ψ by

$$\psi = \left[(x-x^2)\sin(\pi x)(y-y^2)\sin(\pi y) - \frac{16}{\pi^6}\right]\cos t. \tag{4.25}$$

The pressure then is

$$p(x,y,z) = -\frac{\partial \psi}{\partial t} + \nu\Delta\psi. \tag{4.26}$$

The force term is determined from the exact solution. The domain is the unit square $\Omega = [0,\ 1] \times [0,\ 1]$. In this example, we have $\frac{\partial p}{\partial \mathbf{n}} \neq \mu\Delta\mathbf{u}\cdot\mathbf{n} = 0$ on the boundary. Thus, the numerical boundary layer might deteriorate the accuracy of the pressure. Table 1 shows the grid refinement analysis at $t = 10$. The pressure has been adjusted by some constant that minimizes the error in the maximum norm. We can see the velocity is nearly second order accurate, but the pressure is first order using the original BCG method. However, if we add an additional term as suggested in (3.23), then the pressure is very close to second order. The

last column in Table 1 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 BCG method without the correction. Away from the boundary, the pressure is second order accurate.

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

Table 1: Grid refinement analysis of the BCG method with $\mu = 1$, $\Delta t = h$, for Example 1 at $t = 10$. 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 BCG method without correction in (3.23) for the pressure, 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 BCG 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 error constants.

4.1 Example 2: An interface problem with a constant jump in the pressure

In this example, we consider a fixed interface,

$$\frac{(x-0.5)^2}{a^2} + \frac{(y-0.5)^2}{b^2} = 1, \tag{4.27}$$

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

$$u = 0, \quad v = 0, \quad p = \begin{cases} C, & \text{if } \dfrac{(x-0.5)^2}{a^2} + \dfrac{(y-0.5)^2}{b^2} - 1 > 0, \\ -10 + C, & \text{if } \dfrac{(x-0.5)^2}{a^2} + \dfrac{(y-0.5)^2}{b^2} - 1 \le 0, \end{cases} \tag{4.28}$$

where C is an arbitrary constant. In this example, $\frac{\partial p}{\partial \mathbf{n}} = \mu \Delta \mathbf{u} \cdot \mathbf{n} = 0$ on the boundary. Thus, there is no numerical boundary layer when the BCG projection method is applied. We compare our method with Peskin's IBM method using the discrete cosine delta function. Depending on how the interface is represented, there are two different versions of the IBM method:

- The particle approach. In this traditional approach, the interface is represented by a set of Lagrangian points $\mathbf{X}_k$, $k = 1, 2, \cdots, N_b$, 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

$$\delta_w(x) = \begin{cases} \dfrac{1}{4w}\left(1 + \cos\left(\pi x/2w\right)\right), & \text{if } |x| < 2w, \\ 0, & \text{if } |x| \geq 2w. \end{cases} \tag{4.29}$$

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 choice 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 choice 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 2 is the grid refinement analysis for the velocity which is clearly first order accurate.

- The level set representation. If the interface is represented by the zero level set of a two dimensional functions $\varphi(x, y)$ as in (4.39), it is not straightforward to use the IBM method directly. However, if the tangential component of the singular force is zero, then we can write

$$\int_{\Gamma(t)} \mathbf{f}(s)\, \delta_2(\mathbf{x} - \mathbf{X}(s))ds = \mathbf{f} \cdot \mathbf{n}\, \delta(\varphi)\, \nabla\varphi, \tag{4.30}$$

see [3] etc. It is easy to apply the discrete delta function in the above form. Unfortunately, the choice of the width of the discrete delta function is crucial for the IBM method. If $w = h$, the IBM method barely converges, see the second column of Table 2. T. Hou [9] may be the first one to find out such a problem 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 2. However, $w = \sqrt{h}$ is quite large if h is small. The IBM smears out the pressure which means zero order convergence for the pressure. Since the width has to be taken as $w = \sqrt{h}$, we can not guarantee the right pressure even at those grid points that are away from the interface.

A solution to the problem is to use the projections of irregular grid points of a particular side, say $\varphi \leq 0$ in (4.39), on the interface as the control points. Then the original Peskin's

IBM method can be applied with those control points. In [10, 21], we described in detail how to find the projections 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 IBM 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.

N	$\|E^L_{IBM,h}(\mathbf{u})\|_\infty$	order	$\|E^L_{IBM,\sqrt{h}}(\mathbf{u})\|_\infty$	order	$\|E^P_{IBM,h}(\mathbf{u})\|_\infty$	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

Table 2: Grid refinement analysis of the BCG-IBM methods for Example 2 with $\mu = 0.1$, $a = 0.35$, and $b = 0.25$, $\Delta t = h$. 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 IBM method with w being h. Generally speaking, the velocity is first order accurate. The pressure which is not listed in the table has zeroth order accuracy since it is smeared out.

Table 2 lists the comparison of two different versions of the IBM method using the particle approach and the level set function. We have also tested the example with different but modest μ, the major axis a and minor axis b, and different initial conditions. All the results are qualitatively similar.

In Table 3, we show the grid refinement analysis of the BCG-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 and 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 a 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 the singular force term $\hat{f}_1 = 10$ to the Navier-Stokes equations in (1.1). The exact solution for the velocity is the same as Example 1, but the pressure is the sum of the pressure in Example 1 and the piecewise constant in (4.28). The analysis and observations are similar to the discussions above. Fig. 2 (a) shows the pressure computed using the BCG-IIM method. The computed pressure has the right jump up to second order. Fig. 2 (b) and Fig. 2 (c) are the pressure computed using the IBM method with the level set representation of the interface ($w = \sqrt{h}$) and the original IBM method with particle representation of the interface ($w = h$), respectively. In Fig. 2 (b), we

N	$\|E_{IIM}(\mathbf{u})\|_\infty$	order	$\|E_{IIM2}(p)\|_\infty$	order	$\|E_{IIM}(p)\|_\infty$	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

Table 3: The grid refinement analysis of BCG-IIM method for Example 2. The parameters are $\mu = 0.1$, $a = 0.35$ and $b = 0.25$, and $\Delta t = h$. The second and fourth 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 due to the boundary layer on the boundary and on the interface. 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 due to 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.

see that the pressure is smeared out by the width $w = \sqrt{h}$ while in Fig. 2 (c), we see larger errors but thinner layer ($w = h$) near the interface.

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

Example 3: An interface problem with non-constant jump in the pressure

In this example, the interface and the domain are the same as in Example 2. However, the jump in the pressure is *not a constant* anymore. This is a typical case for two phase flow where the jump in the pressure is usually proportional to the surface tension and the curvature of the interface. Again, we assume the tangential surface force is zero so that the velocity is smooth and $[p_{\mathbf{n}}] = 0$. The exact pressure is then determined from the following Poisson equation

$$\Delta p = 0, \quad \left.\frac{\partial p}{\partial \mathbf{n}}\right|_{\partial\Omega} = 0, \tag{4.31}$$

with the jump condition

$$[p]|_\Gamma = \hat{f}_1 = x + y, \quad (x, y) \in \Gamma, \qquad \left.\left[\frac{\partial p}{\partial \mathbf{n}}\right]\right|_\Gamma = 0. \tag{4.32}$$

The piecewise but bounded force term $\mathbf{G}$ is

$$\mathbf{G}(x, y) = \nabla p(x, y), \quad \text{for any} \quad (x, y) \in \Omega - \Gamma. \tag{4.33}$$

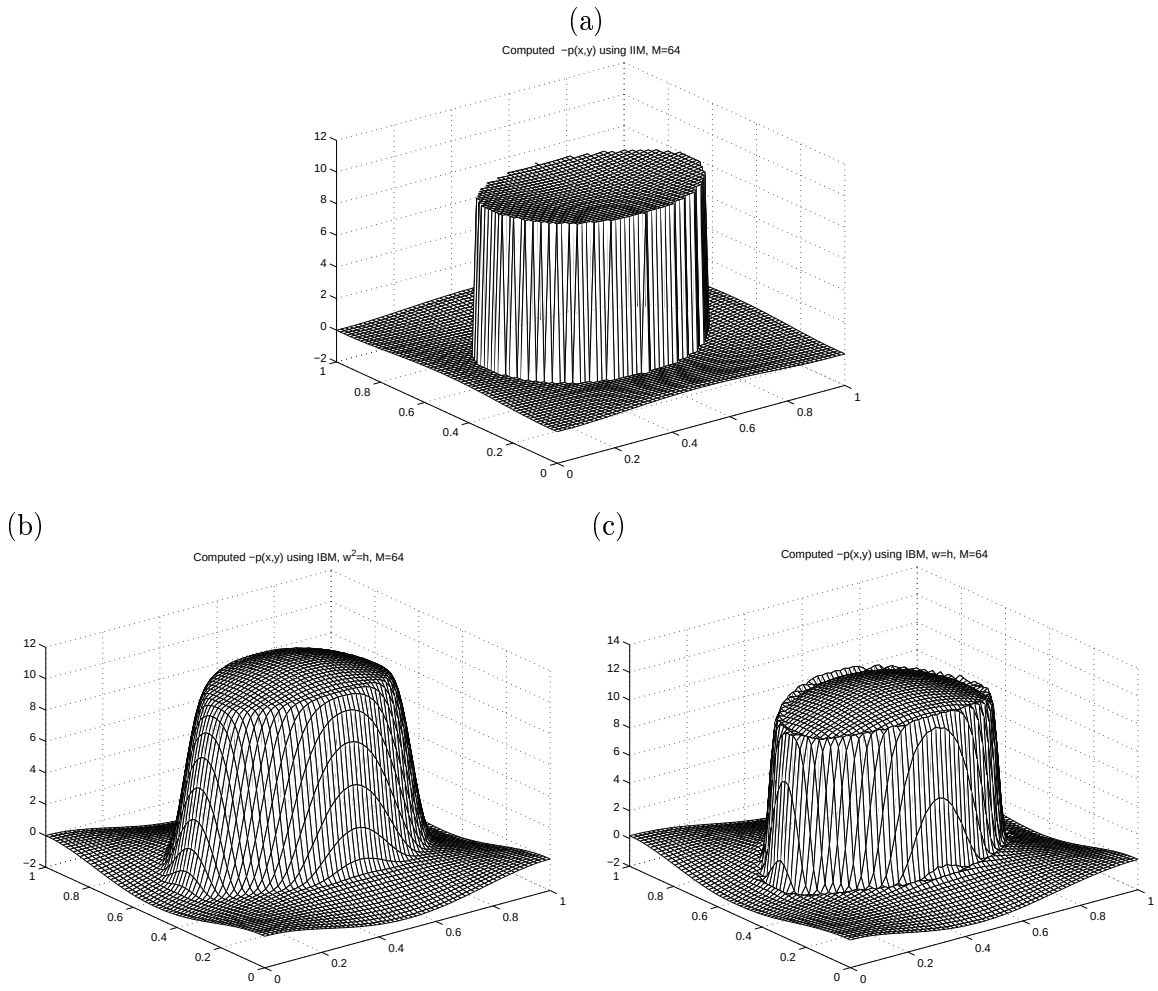


Figure 2: Computed pressure for Example 1 with an added force term $\hat{f}_1 = 10$ using a 64 by 64 grid. (a) The result computed using the BCG-IIM method which catches the jump in the pressure. (b) The result computed using the IBM method with the interface represented by a level set function ($w =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

Note that, $\mathbf{G}$ is bounded but discontinuous across Γ. It is easy to show that the jump in $\mathbf{G}$ is

$$[G_1] = \sin^2\theta - \sin\theta\cos\theta, \qquad [G_2] = \cos^2\theta - \sin\theta\cos\theta,$$

where again, $\mathbf{n} = (\cos\theta, \sin\theta)$ is the unit outward normal direction at a point on the interface. We can also show that

$$\frac{\partial \hat{f}_1}{\partial \eta} = \cos\theta - \sin\theta, \qquad \frac{\partial^2 \hat{f}_1}{\partial \eta^2} = \kappa(\cos\theta + \sin\theta).$$

These jumps are used to determine the correction terms for the BCG-IIM method.

The steady state solution for this example is again $\mathbf{u} = \mathbf{0}$ while the pressure is the solution of the Poisson equations with jump condition given in (4.32). In Table 4, we show the grid refinement analysis for BCG-IIM method without the correction term for the pressure in (3.23) at $t = 10$. The initial velocity is taken from (4.24) in Example 1, while the pressure is

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

where $p_e(x, y)$ is the exact solution of the Poisson equation. In this way, we can guarantee the initial pressure has the right jump condition. We see that the BCG-IIM method gives second order accurate solution for both the velocity and the pressure. The disappearance of the numerical boundary layer seems due to the fact that the term $\cos\pi x \cos\pi y$ satisfies the homogeneous Neumann boundary condition and it is consistent with $\mu\Delta\mathbf{u}\cdot\mathbf{n} = \nabla p\cdot\mathbf{n} = 0$ on the boundary. If we take the initial pressure from (4.25) in Example 1, then the computed pressure again is only first order accurate near the boundary while the velocity is still second order accurate.

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

Table 4: Grid refinement analysis of BCG-IIM applied to Example 3 with $\mu = 1$, $\Delta t = h$, at $t = 10$. The initial velocity is taken from (4.24) in Example 1. The initial pressure is taken as $\cos\pi x \cos\pi y + p_e(x, y)$, where $p_e(x, y)$ is the computed solution of the Poisson equation (4.31) with the jump conditions (4.32).

Example 4: Circular flow with a line force

Now we consider an example with non-smooth velocity. The exact solution is the following:

$$u(x,y,t) = \begin{cases} h(t)\left(\frac{y}{r} - 2y\right) & \text{if } r > \frac{1}{2}, \\ 0 & r \le \frac{1}{2}, \end{cases} \tag{4.34}$$

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

$$p(x,y,t) = 0, \tag{4.36}$$

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 finite jump in the normal direction

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

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

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

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 non-zero and bounded source term $\mathbf{G}$ is derived directly from the exact solution using the Maple. Therefore 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. If we take $h(t) = 1 - e^{-t}$, then for small t, the velocity using our method still converges, but not nearly second order accurate, but not the pressure for the reason explained in the footnote. As time t increases, $\frac{dh}{dt}$ will approach to zero, and we can expect the method converges for all the quantities. We present the numerical results at $t = 10$ in Table 5 and plot the x-component of the velocity in Fig. 3.

A modified example that we tested is to set

$$\mathbf{G} = \mathbf{0}, \qquad \hat{f}_1 = 0, \qquad \hat{f}_2 = 10.$$

The domain and the interface are the same. The velocity is 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 non-constant jump across the interface. While the exact solution

[2]In the projection method, Hodge decomposition 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 BCG method if u_t is large on the boundary.

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

Table 5: Grid refinement analysis of BCG-IIM applied to Example 4 with $\mu = 0.02$, $\Delta t = h$, at $t = 10$. Second order convergence is achieved.

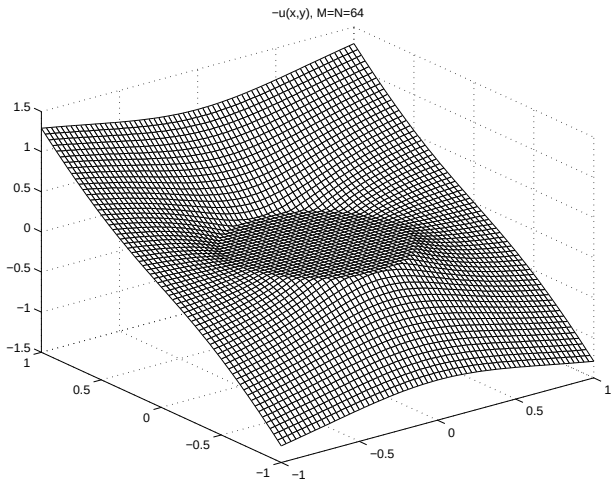


Figure 3: Computed velocity of Example 4 with a 64×64 grid and $\mu = 0.02$. The time step size again is $\Delta t = h$. Plot of $-u(x, y)$ at $t = 10$ which is not smooth.

is difficult to find, the motion of the steady state is a simple rotation along the interface. In Fig. 4 (a), we plotted the x- component of the velocity $-u(x, y)$ at $t = 10$ computed using a 64 by 64 grid. We can clearly see the non-constant jump in the normal derivative. In Fig. 4 (b), we plotted the velocity field at $t = 10$. The flow approaches to a steady state rotation along the circular interface. The initial velocity and the pressure are all taken as zero.

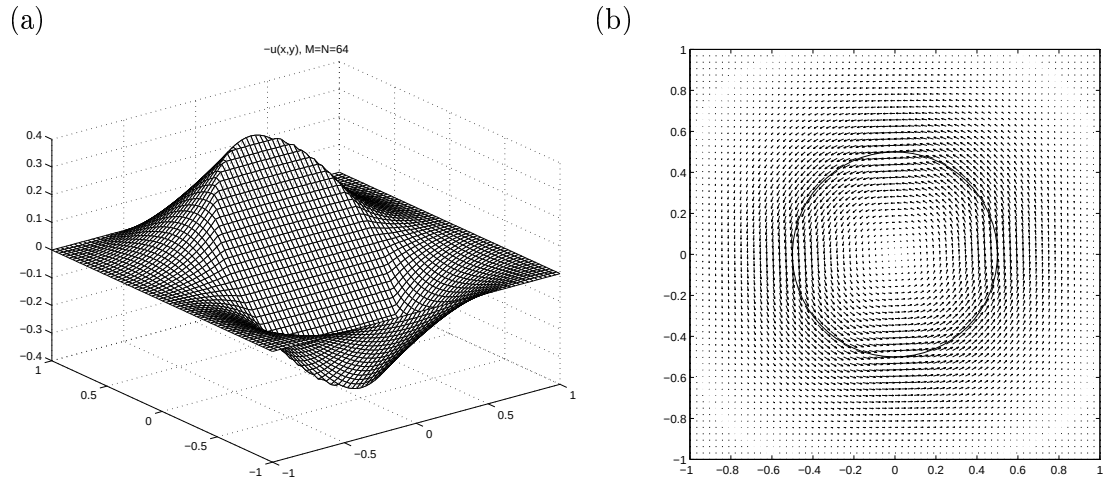


Figure 4: The computed steady state velocity for the modified example in Example 4 with a circular interface $r = 0.5$ with a 64×64 grid. Other parameters are $\mathbf{G} = \mathbf{0}$, $\hat{f}_1 = 0$, $\hat{f}_2 = 10$, and $\mu = 0.02$. The time step size again is $\Delta t = h$. (a) Plot of $-u(x,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

Example 5: A moving interface problem

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

$$r(\theta) = r_0 + \epsilon \sin(k\theta), \quad 0 \le \theta \le 2\pi. \tag{4.37}$$

The only force now is the surface tension which is proportional to the curvature κ, that is,

$$\hat{f}_1(\Gamma, t) = \epsilon\kappa, \quad \hat{f}_2(\Gamma, t) = 0. \tag{4.38}$$

The velocity is smooth but the pressure has a non-constant 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 $\mu = 0.02$. The initial velocity and the pressure are all set to be zero.

We use the level set method [22] to update the interface. We refer the readers to [10, 21, 20] 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

$$\varphi(x,y,t)\begin{cases} <0, & \text{if } (x,y) \text{ is in the inside of the interface,} \\ =0, & \text{if } (x,y) \text{ is on the interface,} \\ >0, & \text{if } (x,y) \text{ is in the outside of interface.} \end{cases} \tag{4.39}$$

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

$$\varphi_t + \vec{u}\cdot\nabla\varphi = 0 \tag{4.40}$$

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

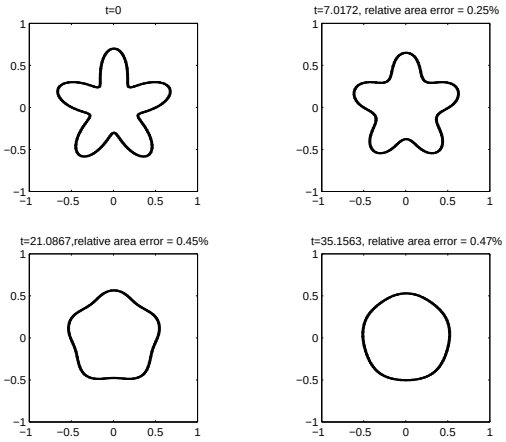
$$\Delta t = \min\left\{ \frac{h}{2\|\mathbf{u}\|_\infty}, h \right\} \tag{4.41}$$

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

In Fig. 5, we plotted the computed interface at different time with a 160 by 160 grid[3]. The other parameters are $\mu = 0.1$, $\epsilon = 0.05$. Using the immersed interface method, we know the intersections of the interface and the grid lines. The set of the intersections which are on the interface at a particular time, is one dimensional. Therefore we can afford to store the interface at many different time levels. In Fig. 5 (a), we plotted the interface at $t = 0$, $t = 7.0172$, $t = 21.0867$, and $t = 35.1563$. The relative error in the area are 0.25% at $t = 7.0172$, 0.45% at $t = 21.0867$, and 0.47% at $t = 35.1563$. In Fig. 5 (b), we just plotted the interface at $t = 0$ and $t = 100$, which is almost in the equilibrium state, a circle. The maximum norm of the velocity at $t = 100$ is $\|\mathbf{u}\|_\infty = 3.8056 \times 10^{-4}$ indicating second order convergence for the velocity. The pressure approaches to two different constant inside and outside. The relative area change (loss) at $t = 100$ is about 1.61%.

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

(a)



(b)

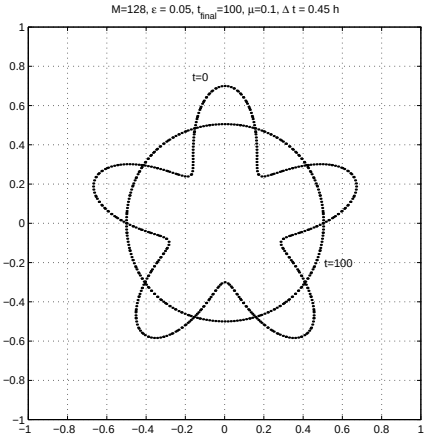


Figure 5: The numerical evolution of the moving interface for Example 5 using the BCG-IIM method coupled with the level set formulation. The parameters are: $\mu = 0.1$, $\epsilon = 0.05$, $M = N = 160$, and $\Delta t = \min\left\{ \frac{h}{2\|\mathbf{u}\|_\infty}, h \right\}$. Due to the surface tension, the interface is relaxing to a circle. The figures are plots of the intersections of the interface and the x- and y- axis so that we do not need to store two dimensional level set functions at different time. (a) The computed interface at $t = 0$, $t = 7.0172$, $t = 21.0867$, and $t = 35.1563$. The relative error in the area are 0.25% at $t = 7.0172$, 0.45% at $t = 21.0867$, and 0.47% at $t = 35.1563$. (b) The initial interface $t = 0$ and the computed interface at $t = 100$. The relative area change (loss) at $t = 100$ is about 1.61%.

5 Conclusions

We have developed an immersed interface method coupled with the BCG 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 BCG method. The new BCG-IIM method has the same stability nature as the original projection method since only right hand sides in the BCG method are modified. The numerical results show second order convergence for the BCG-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 width being $2h$.

Research is under way for the Navier-Stokes equations that involve a singular force along one or several interfaces and involve *discontinuous* 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.

6 Acknowledgment

The first author is partially supported by the Army Research Office under grant number 39676-MA, and the NSF under grant number DMS0073403. The second author was partially supported by National Science Council of Republic of China Grant NSC-89-2119-M-194-001. The authors would like to thank Dr. C. Wang for the original BCG projection code. We are also grateful to Dr. R. LeVeque for reading and commenting on the original manuscript.

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.*, 85:257–283, 1989.

[2] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. Preprint, 2000.

[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] R. H. Dillon, L. J. Fauci, and A. L. Fogelson. Modeling biofilm processes using the immersed boundary method. *J. Comput. Phys.*, 129:57–73, 1996.

[5] W. E. and J. Liu. Projection method: I. convergence and numerical boundary-layers. *SIAM J. Numer. Anal.*, 32:1017–1057, 1995.

[6] L. J. Fauci. Interaction of oscillating filaments – A computational study. *J. Comput. Phys.*, 86:294–313, 1990.

[7] A. L. Fogelson. A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting. *J. Comput. Phys.*, 56:111–134, 1984.

[8] A. L. Fogelson. Continuum models for platelet aggregation: Formulation and mechanical properties. *SIAM J. Numer. Anal.*, 52:1089, 1992.

[9] T. Hou. Private communications, 2000.

[10] 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.*, 134:236–252, 1997.

[11] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.*, 59:308–323, 1985.

[12] M. Lai and C. S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *J. Comput. Phys.*, 160:705–719, 2000.

[13] 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. Letters*, in press, 2000.

[14] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, 31:1019–1044, 1994.

[15] R.J. LeVeque and Z. Li. Immersed interface method for Stokes flow with elastic boundaries or surface tension. *SIAM J. Sci. Comput.*, 18:709–735, 1997.

[16] Z. Li. *The Immersed Interface Method — A Numerical Approach for Partial Differential Equations with Interfaces*. PhD thesis, University of Washington, 1994.

[17] Z. Li. A note on immersed interface methods for three dimensional elliptic equations. *Computers Math. Appl.*, 31:9–17, 1996.

[18] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.

[19] Z. Li and K. Ito. Maximum principle preserving schemes for interface problems with discontinuous coefficients. *NCSU CRSC-TR00-04*, 2000.

[20] Z. Li and B. Soni. Fast and accurate numerical approaches for Stefan problems and crystal growth. *Numerical Heat Transfer, B: Fundamentals*, 35:461–484, 1999.

[21] 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.*, 152:281–304, 1999.

[22] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[23] C. S. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25:220–252, 1977.

[24] C. S. Peskin. Lectures on mathematical aspects of physiology. *Lectures in Appl. Math.*, 19:69–107, 1981.

[25] C. S. Peskin and D. M. McQueen. Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 37:113–132, 1980.

[26] 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.

[27] 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.*, 130:269–282, 1997.

[28] J. Shen. On error estimates of the projection methods for the Navier-Stokes equations: second-order scheme. *Math. Comp.*, 65:1039–1065, 1996.

[29] D. Sulsky and J. U. Brackbill. A numerical method for suspension flow. *J. Comput. Phys.*, 96:339–368, 1991.

[30] R. Teman. Remark on the pressure boundary condition for the projection method. *Theoret. Comput. Fluid Dynamics*, 3:181–184, 1991.

[31] C. Wang and J. Liu. Analysis of finite difference schemes for unsteady Navier-Stokes equations in vorticity formulation. Preprint, 2000.