Proceedings of the National Institute for Mathematical Sciences
Volume 1, Number 2, September 2006, Pages 100–103

A SIMPLE IMMERSED INTERFACE METHOD FOR THE STOKES FLOW WITH SINGULAR FORCES

MING-CHIH LAI, HSIAO-CHIEH TSENG, AND ZHILIN LI

Abstract. In this paper, we introduce a simple version of immersed interface method (IIM) for the Stokes flow with singular forces along an interface. The numerical method is based on applying the Taylor's expansions along the normal direction and incorporating the solution and its normal derivative jumps into the finite difference approximations. The numerical results show that the scheme is second-order accurate.

1. Introduction

The fluid problems with interfaces have many applications in science and engineering. For example, Peskin's formulation for the blood flow in the heart valve leaflet (interface) involves exerting singular forces along the leaflet into the blood (fluid)[20]. Another example is the Hele-Shaw flow which can be formed by pumping a more viscous fluid into a less viscous surrounding fluid in two parallel thin plates. The shape of the interface is known to exhibit a fingering phenomenon. The governing equations for the above problems involve solving elliptic partial differential equations with possible discontinuous coefficients or with possible jump conditions for the unknown and its derivative across the interface. Several classes of practical finite difference methods have been developed in the past three decades. Those methods provide different treatments near the interface while keep the standard difference discretization away from the interface. We summarize those different techniques as follows.

The immersed boundary (IB) method of Peskin [21] is the first Cartesian method to provide a simple solution to the fluid problems interacting with complicated interfaces. The method is to treat the interface as a singular force (source) generator along which the force can be transmitted into the fluid smoothly. More precisely, the method uses the Lagrangian markers to track the interface and affects the fluid in Eulerian grid via a smooth version of discrete delta function. Since the equations are solved on Cartesian grid, many fast fluid solvers can be applied easily. For instance, the original IB method exploits Fast Fourier Transform (FFT) in solving the Stokes equations in rectangular domain. The method is easy to implement but

Received by the editors August 30, 2006.
2000 *Mathematics Subject Classification.* Primary 65N06, 65N50.
Key words and phrases. Immersed interface method; Stokes flow; Singular forces.
The authors are supported in part by MOE-ATU program and National Science Council of Taiwan under research grant NSC-94-2115-M-009-004.
The third author is partially supported by USA grants: NSF-NIH #0201094, NSF DMS-0412654, ARO 43751-MA, and AFSOR FA9550-06-1-0241.

it is only first-order accurate. Besides, the solution is smoothing out across the interface. Although there are different variants of formally second-order schemes recently [8, 6], the overall accuracy can only be increased for the sufficiently smooth problems. Despite this first-order accuracy issue, the IB method nowadays is getting more attention on applying to the flow interacting with rigid boundaries which has important applications in computational fluid dynamics. The reason for that is quite simple since it does not need to handle grid generations for unsteady problems which can save the computational effort significantly.

Unlike the IB method to have numerical smearing near the interface, the immersed interface method was invented to capture the solution or its derivative jumps sharply. The idea is to incorporate those jump conditions into the finite difference discretization by modifying the difference approximations near the interface. Mayo [17] used the idea to solve the Poisson equation in an irregular domain. Mayo named those irregular grid points where the discrete five-point Laplacian across the interface. LeVeque and Li later applied to an elliptic interface problem with discontinuous coefficients and singular sources [12]. Since the problem is more difficult to solve than the Poisson problem, they have to introduce the local coordinates and derive the interface relations on that coordinates. Their method appears to be the first finite difference method that is second-order accurate for the problem. Dumett and Keener has extended the method to the anisotropic case of elliptic interface problems [5]. There are other applications to the fluid problems [11, 15, 14, 24], just to name a few. Readers who are interested in IIM can refer to Li's recent review article [9] and the new published book [10].

There are several methods sharing the similar spirit as IIM in literature. For example, the boundary condition capturing method proposed by Liu, Fedkiw, and Kang [16] can be implemented to solve the elliptic interface problems dimension by dimension. The method captures the solution and its normal derivative jumps sharply while smoothing the tangential derivative. The method is easy to implement; however it is only first-order accurate. There are other variants of IIM such as the the explicit jump [25] and the decomposed immersed interface method [3]. Beale and Lai [1] applied the boundary integral method to compute the solution for the Laplace problem near the interface and used it to form the correction terms. This method has been further extended to the Stokes problem with singular forces [2].

In this paper, we extend the method used in [23] for the Poisson problem to the Stokes flow with singular forces along an interface. The numerical method is based on applying the Taylor's expansions along the normal direction and incorporating the solution and its normal derivative jumps into the finite difference approximations. The numerical results show that the scheme is indeed second-order accurate.

The paper is organized as follows. In Section 2, we review the immersed interface method for the Poisson problems with interface given in [23]. Then the numerical scheme for the Stokes flow with singular forces is discussed and the numerical results are shown in Section 3.

2. Poisson Equation with Jump Discontinuities

In this section, we consider the following Poisson problem on a computational domain $\Omega = [a, b] \times [c, d]$ with an imbedding interface $\Gamma = \{\mathbf{X}(s) = (X(s), Y(s)), 0 \leq s < L\}$, where s is the arc length parameter and $\mathbf{X}(0) = \mathbf{X}(L)$. The interface Γ

separates the domain Ω into two regions,; namely inside (Ω^-) and outside (Ω^+) the interface. Across Γ, the solution and its normal derivatives exhibit jump discontinuities. The problem can be written as

$$\Delta u(\mathbf{x}) = f(\mathbf{x}), \text{ in } \Omega, \tag{2.1}$$

$$[u](s) = g(s), \qquad \left[\frac{\partial u}{\partial \mathbf{n}}\right](s) = h(s) \text{ on } \Gamma, \tag{2.2}$$

$$u(\mathbf{x}) = u_b(\mathbf{x}) \text{ on } \partial\Omega, \tag{2.3}$$

where the jumps $[u](s) = u^+ - u^-$ and $\left[\frac{\partial u}{\partial \mathbf{n}}\right](s) = \frac{\partial u^+}{\partial \mathbf{n}} - \frac{\partial u^-}{\partial \mathbf{n}}$ are defined as the difference of the limiting values from two different sides of the interface. Here, we assume the interface parametrization $\mathbf{X}(s)$ and the jump discontinuity functions $g(s)$ and $h(s)$ are fairly smooth (say $C^2(\Gamma)$) on the interface.

2.1. **An IIM to incorporate the jumps in the normal direction.** In this subsection, we review the scheme developed in [23] to solve Eqs. (2.1)-(2.3). Before we proceed, we first lay out a uniform Cartesian grid in Ω with mesh width $h = \Delta x = \Delta y$. The value $u_{i,j}$ denotes the discretized solution located at the grid point $\mathbf{x}_{i,j}$ with the coordinates $x_i = a + i\Delta x$, $y_j = c + j\Delta y$. We then classify the grid points as the regular or irregular points. For the regular grid point, we mean that at which the standard five-point Laplacian will not cross the interface. On the other hand, suppose the five-point Laplacian involves using the grid points inside and outside the interface, we call such point as the irregular point. Since the solution is smooth either inside or outside the interface, the discrete Laplacian at regular points do not need to modify. However, it has to be modified at those irregular points since the solution is not smooth across the interface and the modification must depend on the jumps $[u]$, and $\left[\frac{\partial u}{\partial \mathbf{n}}\right]$. We explain how to modify the discrete Laplacian in the following.

Let $\mathbf{x}_{i,j}$ be the irregular point as shown in Fig. 1. The five-point Laplacian at $\mathbf{x}_{i,j}$ can be written as

$$\begin{aligned}
\Delta_h u &= \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} \\
&= \frac{u^+_{i-1,j} - 2u^-_{i,j} + u^-_{i+1,j}}{h^2} + \frac{u^+_{i,j+1} - 2u^-_{i,j} + u^-_{i,j-1}}{h^2} \\
&= \frac{u^-_{i-1,j} - 2u^-_{i,j} + u^-_{i+1,j}}{h^2} + \frac{u^-_{i,j+1} - 2u^-_{i,j} + u^-_{i,j-1}}{h^2} \\
&\quad + \frac{u^+_{i-1,j} - u^-_{i-1,j}}{h^2} + \frac{u^+_{i,j+1} - u^-_{i,j+1}}{h^2} \\
&= (u^-_{xx})_{i,j} + (u^-_{yy})_{i,j} + \mathcal{O}(h^2) + \frac{u^c_{i-1,j}}{h^2} + \frac{u^c_{i,j+1}}{h^2} \\
&= f^-_{i,j} + \frac{1}{h^2}\left(u^c_{i-1,j} + u^c_{i,j+1}\right) + \mathcal{O}(h^2).
\end{aligned}$$

To derive the the correction term $u^c_{i-1,j}$, we need to find the projection point of $\mathbf{x}_{i-1,j}$ at the interface (say $\mathbf{X}_* = \mathbf{X}(s_*)$), and then apply the Taylor's expansions

along the normal direction at $\mathbf{X}_*$. That is,

$$\begin{aligned} u^c_{i-1,j} &= u^+_{i-1,j} - u^-_{i-1,j} \\ &= \left(u^+_* + d\frac{\partial u^+_*}{\partial \mathbf{n}} + \frac{d^2}{2}\frac{\partial^2 u^+_*}{\partial \mathbf{n}^2} + \mathcal{O}(h^3) \right) - \left(u^-_* + d\frac{\partial u^-_*}{\partial \mathbf{n}} + \frac{d^2}{2}\frac{\partial^2 u^-_*}{\partial \mathbf{n}^2} + \mathcal{O}(h^3) \right) \\ &= [u]_{\mathbf{X}_*} + d\left[\frac{\partial u}{\partial \mathbf{n}}\right]_{\mathbf{X}_*} + \frac{d^2}{2}\left[\frac{\partial^2 u}{\partial \mathbf{n}^2}\right]_{\mathbf{X}_*} + \mathcal{O}(h^3). \end{aligned}$$

The value d is the signed distance between the grid point $\mathbf{x}_{i-1,j}$ and the projection $\mathbf{X}_*$. Notice that, if the grid point $\mathbf{x}_{i-1,j}$ is inside the interface, then d must be negative.

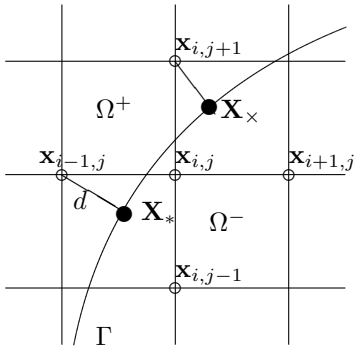

FIGURE 1. The five-point Laplacian at the irregular point $\mathbf{x}_{i,j}$.

The terms $[u]_{\mathbf{X}_*}$ and $\left[\frac{\partial u}{\partial \mathbf{n}}\right]_{\mathbf{X}_*}$ are given from Eq. (2.2); however, the term $\left[\frac{\partial^2 u}{\partial \mathbf{n}^2}\right]_{\mathbf{X}_*}$ is unknown. To find the second normal derivative term, one can rewrite the Laplace operator in Eq. (2.1) on the interface

$$\frac{\partial^2 u}{\partial \mathbf{n}^2}(\mathbf{X}(s)) + \kappa(\mathbf{X}(s))\frac{\partial u}{\partial \mathbf{n}}(\mathbf{X}(s)) + \frac{\partial^2 u}{\partial s^2}(\mathbf{X}(s)) = f(\mathbf{X}(s)),$$

where the value $\kappa(\mathbf{X}(s))$ is the local curvature of the interface. From the above equation, we can easily compute the second normal derivative jump at $\mathbf{X}_*$. That is,

$$\left[\frac{\partial^2 u}{\partial \mathbf{n}^2}\right]_{\mathbf{X}_*} = [f]_{\mathbf{X}_*} - \kappa_{\mathbf{X}_*}\left[\frac{\partial u}{\partial \mathbf{n}}\right]_{\mathbf{X}_*} - \frac{\partial^2}{\partial s^2}[u]_{\mathbf{X}_*}.$$

Thus, the final correction term can be obtained as

$$\begin{aligned} u^c_{i-1,j} &= [u]_{\mathbf{X}_*} + d\left[\frac{\partial u}{\partial \mathbf{n}}\right]_{\mathbf{X}_*} + \frac{1}{2}d^2\left[\frac{\partial^2 u}{\partial \mathbf{n}^2}\right]_{\mathbf{X}_*} \\ &= [u]_{\mathbf{X}_*} + d\left[\frac{\partial u}{\partial \mathbf{n}}\right]_{\mathbf{X}_*} + \frac{1}{2}d^2\left([f]_{\mathbf{X}_*} - \kappa_{\mathbf{X}_*}\left[\frac{\partial u}{\partial \mathbf{n}}\right]_{\mathbf{X}_*} - \frac{\partial^2}{\partial s^2}[u]_{\mathbf{X}_*} \right). \end{aligned}$$

The correction term $u^c_{i,j+1}$ can be computed in a similar way.

It is worth mentioning that the correction term $u^c_{i-1,j}$ must be computed up to third-order accurate since it appears in the discrete Laplacian term by dividing h^2. Thus, the local truncation errors at those irregular points are $\mathcal{O}(h)$. Despite this, the overall accuracy is still second-order since the errors at regular points are $\mathcal{O}(h^2)$. If the jump of the second normal derivative is not taken into account, then the discrete Laplacian at those irregular points has $\mathcal{O}(1)$ errors which degrades the overall accuracy to first-order.

2.2. **Finding the orthogonal projection of a grid point on the interface.** There are two different techniques (based on how the interface is represented) of finding the orthogonal projection of a grid point on the interface. The first technique is to use the arc-length parametrization for the interface. Let the grid point be denoted by $\mathbf{x}_{i,j}$, the projection point of $\mathbf{x}_{i,j}$ on the interface $\mathbf{X}(s) = (X(s), Y(s))$ can be found by making the inner product of tangential and normal vectors at the projection to be zero. That is,

$$\big(x_i - X(s)\big)\, X'(s) + \big(y_j - Y(s)\big)\, Y'(s) = 0.$$

The above nonlinear equation can be solved easily by a simple root finding algorithm such as the bisection method if the interface representation is explicitly known and given. Thus, the signed distance can be computed directly by

$$d = \pm\sqrt{(x_i - X(s_*))^2 + (y_j - Y(s_*))^2}, \quad (x_i, y_j) \in \Omega^{\pm},$$

where $(X(s_*), Y(s_*))$ is the projection point. Furthermore, the local curvature at the projection point can also be computed by the formula $\kappa = X'Y'' - Y'X''$.

Suppose now the interface is not explicitly given by an analytical curve but rather be represented by a few number of Lagrangian markers. We can construct the interface by those markers using the cubic spline interpolation so an approximated curve function can be obtained. A cubic spline interpolation package is developed for a closed interface in 2D. The package can be easily called to compute the arc-length, the first and second derivatives of the cubic spline function defined at those Lagrangian markers, see in [9].

Another technique is to introduce a level set function in the whole computational domain so that the interface is represented by the zero level set [19]. More precisely, we choose a function $\phi(\mathbf{x})$ such that $\phi(\mathbf{x}) = 0$ represents the interface, and it is positive (negative) when $\mathbf{x}$ is outside (inside) the interface. In general, the level set function is chosen as the signed distance function; for example, if the interface is a 2D unit circle, then the level set function can be chosen as $\phi(x, y) = \sqrt{x^2 + y^2} - 1$. In practical computation, the interface is explicitly defined by a grid function $\phi_{i,j} = \phi(\mathbf{x}_{i,j})$. The unit normal vector $\mathbf{n}$ can be simply computed by the formula $\frac{\nabla_h \phi}{|\nabla_h \phi|}$ and the curvature is $\nabla_h \cdot \mathbf{n}$, where ∇_h is the centered difference approximation to the gradient operator. Since the curvature and normal vector are both defined on the grid points, we can use the standard bi-linear interpolation to obtain those quantities on the interface without any difficulty. In addition, it is quite easy to distinguish those irregular and regular grid points based on the sign of the level set function.

In the level set formulation, finding the orthogonal projection $\mathbf{X}_*$ of an irregular point $\mathbf{x}_{i,j}$ on the interface can be performed as follows. Suppose we write the

projection as

$$\mathbf{X}_* = \mathbf{x}_{i,j} + d\,\mathbf{p}_{i,j}, \qquad \mathbf{p}_{i,j} = \frac{\nabla_h \phi_{i,j}}{|\nabla_h \phi_{i,j}|}.$$

Then $d = \mathcal{O}(h)$ is the distance parameter that we have to compute. Since $\mathbf{X}_*$ is on the interface, we should have $\phi(\mathbf{X}_*) = 0$. We can compute the distance d up to third-order accuracy by using Taylor's expansions and solving the quadratic equation

$$\phi(\mathbf{x}_{i,j}) + d\,|\nabla_h \phi_{i,j}| + \frac{d^2}{2}(\mathbf{p}_{i,j}^T\, He(\phi_{i,j})\, \mathbf{p}_{i,j}) = 0,$$

where $He(\phi_{i,j})$ is the Hessian matrix of ϕ.

2.3. **Numerical Examples.** We consider three different exact solutions (shown in Table 1) for Poisson equation in $\Omega = [0,1] \times [0,1]$ with jump discontinuities on a circle interface with the radius 0.5. In our tests, we choose $N \times N$ grid points in Ω. As in [23], the interface is just a circle so the projection point can be found straightforwardly. Thus, the results shown below are almost the same as in [23]. However, in our present code, we have also implemented the uniformly chosen markers to construct the interface and thus from which to find the necessary projections and the curvatures. Our numerical scheme and interface construction can be applied to any smooth curve in principle.

Table 2 shows the maximum errors comparison with the numerical results obtained in [12]. One can see that our results are comparable with the ones obtained in [12]. For Example 3, the discrete Laplacian approximates the Laplacian operator exactly so that the error is roughly equal to the machine precision. For Example 1 and 2, we can see that the present scheme is indeed second-order convergent.

	Example 1	Example 2	Example 3
u_-	1	$\exp(x)\cos(y)$	$x^2 - y^2$
u_+	$1 + \ln(2\sqrt{x^2+y^2})$	0	0
f_-	0	0	0
f_+	0	0	0

TABLE 1. Three test examples for Eqns. (2.1).-(2.3)

	N	[12]	Present	ratio
Example 1	20	2.3908E-3	3.1141E-03	-
	40	8.3461E-4	8.9776E-04	3.4688
	80	2.4451E-4	2.4755E-04	3.6267
	160	6.6856E-5	6.8990E-05	3.5881
	320	1.5672E-5	1.7056E-05	4.0449
Example 2	20	4.37883E-4	2.3451E-04	-
	40	1.07887E-4	6.0435E-05	3.8804
	80	2.77752E-5	1.5221E-05	3.9705
	160	7.49907E-6	3.9066E-06	3.8962
	320	1.74001E-7	9.7236E-07	4.0176
Example 3	20	-	2.9143E-16	-
	40	-	3.4694E-16	-
	80	-	6.1062E-16	-
	160	-	9.7145E-16	-
	320	-	7.6345E-15	-

TABLE 2. Numerical results for Poisson problem with an interface.

3. Stokes Flow with Singular Forces

The steady Stokes equations are of the form

$$-\nabla p + \mu\,\Delta \mathbf{u} + \mathbf{f} = 0 \qquad \text{in } \Omega, \tag{3.1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{3.2}$$

$$\mathbf{u} = \mathbf{u}_b \qquad \text{on } \partial\Omega, \tag{3.3}$$

where $\mathbf{u} = \big(u(\mathbf{x}), v(\mathbf{x})\big)$ is the fluid velocity, $p = p(\mathbf{x})$ is the fluid pressure, and the constant μ is the fluid viscosity. The force density $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$ is applied by the immersed boundary with the arc-length parametrization $\Gamma = \{\mathbf{X}(s) = (X(s), Y(s)), 0 \le s \le L\}$ into the fluid. More precisely, the force term $\mathbf{f}$ has the following form

$$\mathbf{f}(\mathbf{x}) = \int_\Gamma \mathbf{F}(s)\delta(\mathbf{x} - \mathbf{X}(s))\, ds, \tag{3.4}$$

where δ is the 2D Dirac delta function. The boundary force $\mathbf{F}$ is further represented by

$$\mathbf{F} = (F_1, F_2) = (\mathbf{F}\cdot\mathbf{n})\mathbf{n} + (\mathbf{F}\cdot\tau)\tau = F_{\mathbf{n}}\,\mathbf{n} + F_\tau\,\tau, \tag{3.5}$$

where $\mathbf{n}$, τ are unit normal and tangent vectors on the interface and $F_{\mathbf{n}}$, F_τ are the associated forces, respectively.

Since there exists a delta function singularity along the interface, one can expect that the velocity and the pressure will not be smooth across the interface. In fact, the boundary forces $\mathbf{F}$ plays an important role to the jump conditions of $\mathbf{u}$ and p, and their derivatives. In IIM, we usually need to reformulate the problems (3.1)-(3.4) to the homogeneous Stokes problem with known jump conditions along the interface. We outline the jump conditions in the following [7, 13, 22].

Theorem 3.1. *Let* $\mathbf{u}$ *and* p *be the solution of the Stokes equations (3.1)-(3.2) in* Ω*. Then the jumps across the interface* Γ *are*

$$[\mathbf{u}] = 0, \qquad \mu \left[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] = -F_\tau \, \tau, \qquad [p] = F_{\mathbf{n}}, \qquad \left[\frac{\partial p}{\partial \mathbf{n}}\right] = \frac{\partial F_\tau}{\partial s}.$$

3.1. **Numerical Scheme.** In present work, we use the staggered grid with uniform mesh width $h = \Delta x = \Delta y$ for the fluid velocity and the pressure as in Figure 2. By applying the divergence operator to the homogeneous Stokes equations, we have

$$\Delta p = 0 \qquad \text{in } \Omega, \tag{3.6}$$

$$[p] = F_{\mathbf{n}}, \quad \left[\frac{\partial p}{\partial \mathbf{n}}\right] = \frac{\partial F_\tau}{\partial s} \qquad \text{on } \Gamma. \tag{3.7}$$

This is the exact type of the Poisson equation that we have discussed in the previous section so the scheme there can be applied directly. One remaining question is how to choose the pressure boundary condition on the computational boundary $\partial\Omega$. In this work, we simply use the Dirichlet boundary condition since our test solutions are given (see below). In practice, one may use the periodic boundary or zero Neumann boundary conditions instead.

Once we solve the pressure, the velocity field can be obtained by solving another two Poisson-type equations with the interface as

$$\Delta \mathbf{u} = \frac{\nabla p}{\mu}, \qquad \text{in } \Omega, \tag{3.8}$$

$$[\mathbf{u}] = 0, \quad \left[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] = -\frac{1}{\mu} F_\tau \, \tau, \qquad \text{on } \Gamma \tag{3.9}$$

$$\mathbf{u} = \mathbf{u}_b \qquad \text{on } \partial\Omega. \tag{3.10}$$

So the numerical scheme in previous section can be also applied directly to the Stokes problem. However, in order to compute the pressure gradient in the right hand side of Eq. (3.8) more accurately, one has to take its jump across the interface into account. To derive the jumps $[p_x]$ and $[p_y]$, we first recall that $\tau = (\tau_1, \tau_2)$, the unit tangent vector, which equals to $(-n_2, n_1)$ at each point on Γ. Since

$$\begin{aligned} \left[\frac{\partial p}{\partial \mathbf{n}}\right] &= [p_x] \; n_1 + [p_y] \; n_2 = \frac{\partial F_\tau}{\partial s}, \\ \frac{\partial \, [p]}{\partial s} &= [p_x] \; \tau_1 + [p_y] \; \tau_2 = - \, [p_x] \; n_2 + [p_y] \; n_1 = \frac{\partial F_{\mathbf{n}}}{\partial s}, \end{aligned}$$

we have

$$[\nabla p] = ([p_x] \, , [p_y]) = \left(n_1 \, \frac{\partial F_\tau}{\partial s} - n_2 \, \frac{\partial F_{\mathbf{n}}}{\partial s} \; , \quad n_2 \, \frac{\partial F_\tau}{\partial s} + n_1 \, \frac{\partial F_{\mathbf{n}}}{\partial s} \right).$$

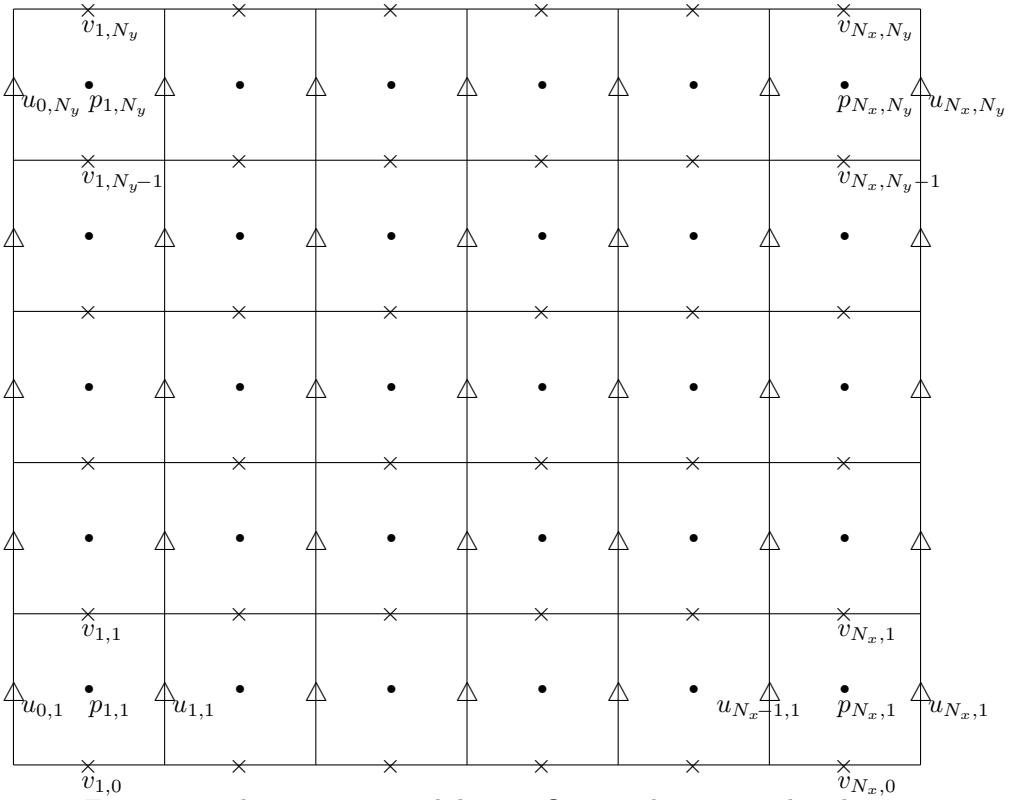

FIGURE 2. The computational domain Ω using the staggered grid with mesh width h. The pressure locates at the center of the cell, the velocity component u locates at left-right faces, and v locates at top-bottom faces.

We summarize the complete solution procedures as follows.

(1) Solve the pressure Poisson equation

$$\Delta_h p = C^{(p)}, \qquad \text{in} \quad \Omega$$

with modified right-hand side functions at those irregular points which the modifications are based on the jump conditions for the pressure. More precisely, the correction terms at those irregular points can be computed as

$$\begin{aligned} C^{(p)} &= \frac{1}{h^2}\left([p] + \alpha\left[\frac{\partial p}{\partial \mathbf{n}}\right] + \frac{1}{2}\alpha^2\left([\Delta p] - \kappa\left[\frac{\partial p}{\partial \mathbf{n}}\right] - \frac{\partial^2 [p]}{\partial s^2}\right)\right) \\ &= \frac{1}{h^2}\left(F_{\mathbf{n}} + \alpha\frac{\partial F_\tau}{\partial s} + \frac{1}{2}\alpha^2\left(-\kappa\frac{\partial F_\tau}{\partial s} - \frac{\partial^2 F_{\mathbf{n}}}{\partial s^2}\right)\right). \end{aligned}$$

(2) Compute p_x, p_y on the u- and v-grid by centered difference $(p_{i+1,j} - p_{i,j})/h$, $(p_{i,j+1} - p_{i,j})/h$, respectively. It needs some correction when the p-grid points straddle across the interface. The key point is to correct the pressure at the location differing from the location of the velocity. Let us explain the procedure more clearly by taking the case where the locations of $p_{i,j}$ and $u_{i,j}$ fall in the same side of the interface (say Ω^-) while the location

of $p_{i+1,j}$ falls in the other side of the interface (Ω^+). In this case we need to add the correction term $C^{(p)}_{i+1,j}$ of $p_{i+1,j}$ such that the approximation of p_x can be computed by $((p_{i+1,j} + C^{(p)}_{i+1,j}) - p_{i,j})/h$. The other cases can be handled similarly.

(3) Solve the Poisson equation

$$\begin{aligned} \mu\,\Delta_h \mathbf{u} &= \nabla_h p + \mathbf{C}^{(\mathbf{u})} \qquad \text{in} \quad \Omega, \\ \mathbf{u} &= \mathbf{u}_b \qquad \text{on } \partial\Omega, \end{aligned}$$

where the corrections are made only at irregular points.

$$\begin{aligned} \mathbf{C}^{(\mathbf{u})} &= \frac{\mu}{h^2}\left([\mathbf{u}] + \alpha \left[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] + \frac{1}{2}\,\alpha^2 \left([\Delta \mathbf{u}] - \kappa \left[\frac{\partial \mathbf{u}}{\partial \mathbf{n}}\right] - \frac{\partial^2 [\mathbf{u}]}{\partial s^2}\right)\right) \\ &= \frac{1}{h^2}\left(-\alpha F_\tau \tau + \frac{1}{2}\,\alpha^2\,\left([\nabla p] + \kappa\, F_\tau \tau\right)\right). \end{aligned}$$

3.2. **Numerical results.** In this subsection, we consider three different test examples given in [4]. The computational domain is chosen to be $\Omega = [-2, 2] \times [-2, 2]$ and the interface is just a circle with radius one. Thus, the interface can be simply represented by $\Gamma = \{\mathbf{X}(\theta) = (\cos(\theta), \sin(\theta)), 0 \le \theta < 2\pi\}$ so that the parameter θ is exactly the arc-length parameter. Table 3-5 list the maximum errors of the velocity (u and v) and the pressure (p) for those three examples. The numerical results demonstrate that our scheme is indeed second-order accurate.

Example 1. Normal forces on a circle. Let the boundary force

$$\mathbf{F}(\theta) = 2\,\sin(3\,\theta)\mathbf{X}(\theta),$$

so that the forces are normal to the boundary. The exact solution is given by

$$p(r,\theta) = \begin{cases} -r^3 \sin(3\theta), & r < 1 \\ r^{-3} \sin(3\theta), & r > 1 \end{cases}$$

$$u(r,\theta) = \begin{cases} \frac{3}{8} r^2 \sin(2\theta) + \frac{1}{16} r^4 \sin(4\theta) - \frac{1}{4} r^4 \sin(2\theta), & r < 1 \\ \frac{1}{8} r^{-2} \sin(2\theta) - \frac{3}{16} r^{-4} \sin(4\theta) + \frac{1}{4} r^{-2} \sin(4\theta), & r \ge 1 \end{cases}$$

$$v(r,\theta) = \begin{cases} \frac{3}{8} r^2 \cos(2\theta) - \frac{1}{16} r^4 \cos(4\theta) - \frac{1}{4} r^4 \cos(2\theta), & r < 1 \\ \frac{1}{8} r^{-2} \cos(2\theta) + \frac{3}{16} r^{-4} \cos(4\theta) - \frac{1}{4} r^{-2} \cos(4\theta), & r \ge 1. \end{cases}$$

N	$\|u_{\text{err}}\|_\infty$	Ratio	$\|v_{\text{err}}\|_\infty$	Ratio	$\|p_{\text{err}}\|_\infty$	Ratio
16	6.31625996E-02	-	5.71511520E-02	-	1.80064548E-01	-
32	1.15154249E-02	5.4850	1.04312328E-02	5.4788	6.70713180E-02	2.6846
64	2.46761660E-03	4.6666	3.44416038E-03	3.0286	2.41626231E-02	2.7758
128	2.31427096E-04	10.6626	6.66326129E-04	5.1688	3.79474345E-03	6.3673

TABLE 3. A grid refinement analysis of u, v, p for **Example 1.**

Example 2. Tangential forces on a circle. In this example, we consider the boundary force is only along the tangential direction as

$$\mathbf{F}(\theta) = 2\,\sin(3\theta)\mathbf{X}'(\theta).$$

The exact solution is given by

$$p(r,\theta) = \begin{cases} -r^3\cos(3\theta), & r<1 \\ -r^{-3}\cos(3\theta), & r>1 \end{cases}$$

$$u(r,\theta) = \begin{cases} \frac{1}{8}r^2\cos(2\theta) + \frac{1}{16}r^4\cos(4\theta) - \frac{1}{4}r^4\cos(2\theta), & r<1 \\ -\frac{1}{8}r^{-2}\cos(2\theta) + \frac{5}{16}r^{-4}\cos(4\theta) - \frac{1}{4}r^{-2}\cos(4\theta), & r\geq 1 \end{cases}$$

$$v(r,\theta) = \begin{cases} -\frac{1}{8}r^2\sin(2\theta) + \frac{1}{16}r^4\sin(4\theta) + \frac{1}{4}r^4\sin(2\theta), & r<1 \\ \frac{1}{8}r^{-2}\sin(2\theta) + \frac{5}{16}r^{-4}\sin(4\theta) - \frac{1}{4}r^{-2}\sin(4\theta), & r\geq 1. \end{cases}$$

N	$\|u_{\text{err}}\|_\infty$	Ratio	$\|v_{\text{err}}\|_\infty$	Ratio	$\|p_{\text{err}}\|_\infty$	Ratio
16	5.56812172E-02	-	1.84446032E-02	-	1.92715009E-01	-
32	9.06017465E-03	6.1457	5.85585435E-03	3.1497	2.08500578E-02	9.2429
64	3.34797871E-03	2.7061	1.43026464E-03	4.0942	5.88284456E-03	3.5442
128	6.04375427E-04	5.5395	3.19944420E-04	4.4703	1.32162086E-03	4.4512

TABLE 4. A grid refinement analysis of u, v, p for **Example 2.**

Example 3. Combining the normal and tangential forces on a circle. In the last example, we combine the forces in previous two examples so that the force has both normal and tangential components, that is,

$$\mathbf{F}(\theta) = 2\,\sin(3\theta)\mathbf{X}(\theta) + 2\,\sin(3\theta)\mathbf{X}'(\theta).$$

The exact solution is just a summation of those two solutions in Example 1 and 2.

N	$\|u_{\text{err}}\|_\infty$	Ratio	$\|v_{\text{err}}\|_\infty$	Ratio	$\|p_{\text{err}}\|_\infty$	Ratio
16	1.17188492E-01	-	6.20299174E-02	-	2.02998893E-01	-
32	1.22218170E-02	9.5884	1.25574945E-02	4.9396	8.55532629E-02	2.3727
64	3.73073530E-03	3.2759	3.67928831E-03	3.4130	2.61792711E-02	3.2679
128	5.23096557E-04	7.1320	7.33869339E-04	5.0135	4.55700360E-03	5.7448

TABLE 5. A grid refinement analysis of u, v, p for **Example 3.**

REFERENCES

1. J. T. Beale and M.-C. Lai, *A method for computing nearly singular integrals*, SIAM Journal on Numerical Analysis, Vol. 38, pp. 1902–1925, (2001).
2. J. T. Beale and A. T. Layton, *On The Accuracy of Finite Difference Methods for Elliptic Problems with Interfaces*, preprint, (2005).
3. P. A. Berthelsen, *A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions*, Journal of Computational Physics, Vol. 197, pp. 364–386, (2004).
4. R. Cortez, *The Method of Regularized Stokeslets* SIAM Journal of Scientific Computing, Vol. 23, No. 4, pp. 1204–1225, (2001).
5. , M. Dumett and J. P. Keener, *A Numerical Method for Solving Anisotropic Elliptic Boundary Value Problems in 3D*, SIAM Journal on Scientific Computing, Vol. 25, pp. 348–367, (2003).
6. B. E. Griffith and C. S. Peskin, *On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems*, Journal of Computational Physics, Vol. 208, pp. 75–105, (2005).

7. M.-C. Lai, *Simulations of the flow past an array of circular cylinders as a test of the immersed boundary method*, Ph.D thesis, Mathematics, New York Universoity, (1998).
8. M.-C.Lai and C. S. Peskin, *An immersed boundary method with formal second-order accuracy and reduced numerical viscosity* Journal of Computational Physics, Vol. 160, pp. 705–719, (2000).
9. Z. Li, *An overview of the immersed interface method and its applications*, Taiwanese J. Mathematics, Vol. 7, No. 1, pp. 1–49, (2003).
10. Z. Li and K. Ito, *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM Frontiers in Applied Mathematics 33, (2006).
11. Z. Li and M.-C. Lai, *The Immersed Interface Method for the Navier-Stokes Equations with Singular Forces* Journal of Computational Physics, Vol. 171, pp. 822–842, (2001).
12. R. LeVeque and Z. Li, *The immersed interface method for elliptic equations with discontinuous c oefficients and singular sources*, SIAM Journal on Numerical Analysis, Vol. 31, pp. 1019–1044, (1994).
13. R. LeVeque and Z. Li, *Immersed interface method for Stokes flow with elastic boundaries or surface tension*, SIAM Journal on Scientific Computing, Vol. 18, pp. 709–735, (1997).
14. D. V. Le, B. C. Khoo and J. Peraire, *An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries*, Journal of Computational Physics, in press (2006).
15. M. N. Linnick and H. F. Fasel, *A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains*, Journal of Computational Physics, Vol. 204, pp. 157–192, (2005).
16. X.-D. Liu, R. P. Fedkiw and M. Kang, *A boundary condition capturing method for Poisson's equation on irregular domains*, Journal of Computational Physics, Vol. 160, pp. 151–178, (2000).
17. A. Mayo, *The fast solution of Poisson's and the biharmonic equations on irregular regions*, SIAM Journal on Numerical Analysis, Vol. 21, pp. 285–299, (1984).
18. A. McKenny, L. Greengard and A. Mayo, *A Fast Poisson Solver for Complex Geometries*, Journal of Computational Physics, Vol. 118, pp. 348–355, (1995).
19. S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, Vol. 79, pp. 12–49, (1988).
20. C. S. Peskin, *Numerical analysis of blood flow in the heart*, Journal of Computational Physics, Vol. 25, pp. 220–252, (1972).
21. C.S. Peskin, *The immersed boundary method*, Acta numerica, pp. 1-39, (2002).
22. C. S. Peskin and B. F. Printz, *Improved volume conservation in the computation of flows with immersed elastic boundaries*, Journal of Computational Physics, Vol. 105, pp. 33–46, (1993).
23. D. Russell and Z. J. Wang, *A cartesian grid method for modeling multiple moving objects in 2D incompressible viscous flow*, Journal of Computational Physics, Vol. 191, pp.177–205, (2003).
24. S. Xu and Z. J. Wang, textitAn immersed interface method for simulating the interaction of a fluid with moving boundaries, Journal of Computational Physics, in press, (2006).
25. A. Wiegmann and K. P. Bube, *The explicit-jump immersed interface method: Finite difference method for PDEs with picecwise smooth solutions*, SIAM Journal on Numerical Analysis, Vol. 37, pp. 827–862, (2000).

Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, TAIWAN.

E-mail address: mclai@math.nctu.edu.tw

Center for Research in Scientific Computation and Department of Mathematics, North Carolina State University, Raleigh, NC27695, USA