An Immersed Interface Method for the Incompressible Navier-Stokes Equations

Duc-Vinh Le*, Boo Cheong Khoo*†, Jaime Peraire*‡
*Singapore-MIT Alliance
†Department of Mechanical Engineering, National University of Singapore
‡Department of Aeronautics and Astronautics, Massachusetts Institute of Technology

***Abstract*— We present an immersed interface algorithm for the incompressible Navier Stokes equations. The interface is represented by cubic splines which are interpolated through a set of Lagrangian control points. The position of the control points is implicitly updated using the fluid velocity. The forces that the interface exerts on the fluid are computed from the constitutive relation of the interface and are applied to the fluid through jumps in the pressure and jumps in the derivatives of pressure and velocity. A projection method is used to time advance the Navier-Stokes equations on a uniform cartesian mesh. The Poisson-like equations required for the implicit solution of the diffusive and pressure terms are solved using a fast Fourier transform algorithm. The position of the interface is updated implicitly using a quasi-Newton method (BFGS) within each timestep. Several examples are presented to illustrate the flexibility of the presented approach.**

I. Introduction

We consider an incompressible fluid in a 2-dimensional domain Ω that contains a material interface Γ(t). The Navier-Stokes equations are written as,

$$\boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \mu\triangle\boldsymbol{u} + \boldsymbol{F} \tag{1}$$

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

with boundary and initial conditions,

$$\boldsymbol{u}|_{\partial\Omega} = \boldsymbol{u}_b \tag{3}$$

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0 \tag{4}$$

where $\boldsymbol{u} = (u_1, u_2)$, is the fluid velocity, p, is the pressure, and μ, is the viscosity of the fluid. Here, we assume that the density, $\rho \equiv 1$, and the viscosity, μ, are constant. The effect of a material interface $\Gamma(t)$ immersed in the fluid results in a singular force $\boldsymbol{F}$ which has the form

$$\boldsymbol{F}(\boldsymbol{x},t) = \int_{\Gamma(t)} \boldsymbol{f}(s,t)\delta(\boldsymbol{x} - \boldsymbol{X}(s,t))ds\ , \tag{5}$$

where, s is the arc-length, $\boldsymbol{X}(s,t)$ is the arc-length parameterization of $\Gamma(t)$, $\boldsymbol{x} = (x,y)$ is the spatial position, and $\boldsymbol{f}(s,t)$ is the force strength. Here, $\delta(\boldsymbol{x})$ is the two-dimensional Dirac function. Since we are dealing with a material interface, we have

$$\frac{\partial}{\partial t}\boldsymbol{X}(s,t) = \boldsymbol{u}\left(\boldsymbol{X},t\right) \tag{6}$$

The singular force $\boldsymbol{F}$ along $\Gamma(t)$ results in solutions which may be non-smooth across the interface, i.e. there may be jumps in pressure and in the derivatives of both pressure and velocity. These jumps in the solution and its derivatives can be related to the applied singular forces. The basic idea of the IIM is to discretize the Navier-Stokes equations on a uniform Cartesian grid and to account for the singular forces by explicitly incorporating the jumps into the difference equations. The main advantage of the IIM is that the solution of the Navier-Stokes equations on a uniform mesh can be obtained very efficiently with the use of fast solvers, and at the same time, complex geometrical changes can be handled in a rather seamless manner. The IIM is motivated by Peskin's original immersed boundary method (IB) [7]. Peskin's IB method has proven to be a very useful method for modelling fluid-structure interaction involving large geometry variations. The IB method uses a set of control points to represent the interface. The force densities are computed at these points and spread to the Cartesian grid points by a discrete representation of the delta function. The Navier-Stokes equations with the forcing terms are then solved for pressure and velocity at the Cartesian grid points. The velocity field is used to find the control points' velocity which is used to advance the position of the interface. Since the IB method uses the discrete delta function approach, it smears out sharp interfaces to a thickness of the order of the mesh width and it is only first-order accurate for problems with non-smooth solutions. In contrast, the IIM can avoid this smearing and maintains second-order accuracy by incorporating the known jumps at the interface into the finite difference scheme. The IIM was originally proposed by LeVeque and Li [3] for solving elliptic equations, and later extended to Stokes flows [4]. The method was developed further for the Navier-Stokes equations in [5] and [6]. In [5], the level set method is used to represent the interface. This has the advantage of simplifying the algorithm but does not appear to be adequate to represent certain types of interfaces such as elastic membranes. In [6], the interface is tracked explicitly in a Lagrangian manner, but the singular force $\boldsymbol{F}$ is split into components tangential and normal to the interface. The normal component is then incorporated into jump conditions for pressure across the interface. The tangential component is spread to the nearby Cartesian grid points using the discrete delta function, as in the IB method. Our implementation of the IIM is based on that presented in [5] and [6], but uses a set of control points to represent the interface and incorporates the entire singular force into jump conditions for pressure

and the derivatives of pressure and velocity. The rest of the paper is organized as follows. In section II, we present the relations that must be satisfied along the interface between the singular force, $\boldsymbol{F}$, and the jumps in the solution. In section III, we describe the generalized finite difference approximations which incorporate function jumps. In section IV, we describe the details of the numerical algorithm and in section V, we present some numerical examples.

II. Jump conditions across the interface

We have already mentioned that when singular forces are applied on a material interface, the solutions of the Navier-Stokes equations may be non-smooth. Let $\boldsymbol{n}$ and $\boldsymbol{\tau}$ be the unit outward normal and tangential vectors to the interface, respectively. The normal, $f_1 = \boldsymbol{f}(s,t)\cdot\boldsymbol{n}$, and tangential, $f_2 = \boldsymbol{f}(s,t)\cdot\boldsymbol{\tau}$, components of the force density, can be related to the jump conditions for pressure and velocity as follows (see [4]–[6] for details),

$$[\boldsymbol{u}] = \boldsymbol{0}, \qquad [\mu \boldsymbol{u}_\xi] = -f_2\boldsymbol{\tau}, \qquad [\boldsymbol{u}_\eta] = \boldsymbol{0} \tag{7}$$

$$[p] = f_1, \qquad [p_\xi] = \frac{\partial f_2}{\partial s}, \qquad [p_\eta] = \frac{\partial f_1}{\partial s}\ . \tag{8}$$

The jump, $[\,\cdot\,]$, denotes the difference between the value of its argument outside and inside the interface, and (ξ,η) are the coordinates associated with the directions of $\boldsymbol{n}$ and $\boldsymbol{\tau}$, respectively. In order to construct the appropriate finite difference stencils we will also require the jumps in the second derivatives of velocity and pressure. These, can be obtained by differentiating the above expressions to obtain,

$$\begin{aligned} [\mu \boldsymbol{u}_{\eta\eta}] &= \kappa f_2 \boldsymbol{\tau}, \qquad [\mu \boldsymbol{u}_{\xi\eta}] = -\frac{\partial f_2}{\partial \eta}\boldsymbol{\tau} - \kappa f_2 \boldsymbol{n}, \\ [\mu \boldsymbol{u}_{\xi\xi}] &= -[\mu \boldsymbol{u}_{\eta\eta}] + [p_\xi]\boldsymbol{n} + [p_\eta]\boldsymbol{\tau} + [\boldsymbol{u}_\xi]\boldsymbol{u}\cdot\boldsymbol{n}\ , \end{aligned} \tag{9}$$

$$\begin{aligned} [p_{\eta\eta}] &= \frac{\partial^2 f_1}{\partial \eta^2} - \kappa[p_\xi], \quad [p_{\xi\eta}] = \frac{\partial^2 f_2}{\partial \eta^2} + \kappa[p_\eta], \\ [p_{\xi\xi}] &= -[\nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u})] - [p_{\eta\eta}]\ . \end{aligned} \tag{10}$$

Here, κ is the signed valued of the curvature of the interface (i.e. we assume that $\boldsymbol{n}\times\boldsymbol{\tau} = \boldsymbol{k} \equiv$ constant, so that $\boldsymbol{n}$ can point either towards, or outwards, the center of curvature). From expressions (7)–(10) the values of the jumps of the first and second derivatives of velocity and pressure with respect to the (x,y) coordinates are easily obtained by a simple coordinate transformation. For instance, we write,

$$\begin{aligned} [\boldsymbol{u}_x] &= [\boldsymbol{u}_\xi]n_1 + [\boldsymbol{u}_\eta]\tau_1 \\ [\boldsymbol{u}_{yy}] &= [\boldsymbol{u}_{\xi\xi}]n_2^2 + 2[\boldsymbol{u}_{\xi\eta}]n_2\tau_2 + [\boldsymbol{u}_{\xi\eta}]\tau_2^2\ , \end{aligned}$$

where $\boldsymbol{n} = (n_1, n_2)$ and $\boldsymbol{\tau} = (\tau_1, \tau_2)$ are the Cartesian components of the normal and tangential vectors to the interface at the point considered.

III. Generalized finite difference formulas

From Taylor series expansions, it is possible to show that if the interface cuts a grid line between two grid points at $x = \alpha$, $x_i \le \alpha < x_{i+1}$, then, the following approximations hold for a piecewise twice differentiable function $v(x)$:

$$v_x(x_i) = \frac{v_{i+1} - v_{i-1}}{2h} - \frac{1}{2h}\sum_{m=0}^{2}\frac{(h^+)^m}{m!}[v^{(m)}] + O(h^2) \tag{11}$$

$$v_x(x_{i+1}) = \frac{v_{i+2} - v_i}{2h} - \frac{1}{2h}\sum_{m=0}^{2}\frac{(h^-)^m}{m!}[v^{(m)}] + O(h^2) \tag{12}$$

$$\begin{aligned} v_{xx}(x_i) &= \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} - \frac{1}{h^2}\sum_{m=0}^{2}\frac{(h^+)^m}{m!}[v^{(m)}] \\ &\quad + O(h) \end{aligned} \tag{13}$$

$$\begin{aligned} v_{xx}(x_{i+1}) &= \frac{v_{i+2} - 2v_{i+1} + v_i}{h^2} + \frac{1}{h^2}\sum_{m=0}^{2}\frac{(h^-)^m}{m!}[v^{(m)}] \\ &\quad + O(h) \end{aligned} \tag{14}$$

where $v^{(m)}$, denotes the m-th derivative of v, $v_i = v(x_i)$, $h^+ = x_{i+1} - \alpha$, $h^- = x_i - \alpha$, and h, is the mesh width in the x direction. The jumps in v and its derivatives are defined as

$$[v^{(m)}]_\alpha = \lim_{x\to\alpha^+} v^{(m)}(x) - \lim_{x\to\alpha^-} v^{(m)} \tag{15}$$

in short, $[\,\cdot\,] = [\,\cdot\,]_\alpha$, and $v^{(0)} = v$.
Note that the sign convention for the jump used in the above expressions is different from that in the previous section. This will need to be taken into account when these expressions are used to discretize the Navier-Stokes equations.

The above expressions can be generalized [8] for the case in which we have two interface crossings between x_{i-1} and x_{i+1}. In such cases we need to consider two possibilities:

a) when $x_{i-1} \le \alpha_1 < x_i \le \alpha_2 < x_{i+1}$ we have

$$\begin{aligned} v_x(x_i) &= \frac{v_{i+1} - v_{i-1}}{2h} - \frac{1}{2h}\sum_{m=0}^{2}\frac{(h_1^-)^m[v^{(m)}]_{\alpha_1}}{m!} \\ &\quad - \frac{1}{2h}\sum_{m=0}^{2}\frac{(h_2^+)^m[v^{(m)}]_{\alpha_2}}{m!} + O(h^2) \end{aligned} \tag{16}$$

$$\begin{aligned} v_{xx}(x_i) &= \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} + \frac{1}{h^2}\sum_{m=0}^{2}\frac{(h_1^-)^m[v^{(m)}]_{\alpha_1}}{m!} \\ &\quad - \frac{1}{h^2}\sum_{m=0}^{2}\frac{(h_2^+)^m[v^{(m)}]_{\alpha_2}}{m!} + O(h) \end{aligned} \tag{17}$$

where $h_1^+ = x_i - \alpha_1$, $h_1^- = x_{i-1} - \alpha_1$, $h_2^+ = x_{i+1} - \alpha_2$, and $h_2^- = x_i - \alpha_2$.

b) when $x_i \le \alpha_1 < \alpha_2 < x_{i+1}$, $h_1^+ = x_{i+1} - \alpha_1$, $h_1^- = x_i - \alpha_1$, $h_2^+ = x_{i+1} - \alpha_2$, and $h_2^- = x_i - \alpha_2$, we have

$$v_x(x_i) = \frac{v_{i+1} - v_{i-1}}{2h} - \frac{1}{2h}\sum_{m=0}^{2} \frac{(h_1^+)^m [v^{(m)}]_{\alpha_1}}{m!} - \frac{1}{2h}\sum_{m=0}^{2} \frac{(h_2^+)^m [v^{(m)}]_{\alpha_2}}{m!} + O(h^2) \quad (18)$$

$$v_x(x_{i+1}) = \frac{v_{i+2} - v_i}{2h} - \frac{1}{2h}\sum_{m=0}^{2} \frac{(h_1^-)^m [v^{(m)}]_{\alpha_1}}{m!} - \frac{1}{2h}\sum_{m=0}^{2} \frac{(h_2^-)^m [v^{(m)}]_{\alpha_2}}{m!} + O(h^2) \quad (19)$$

$$v_{xx}(x_i) = \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2} - \frac{1}{h^2}\sum_{m=0}^{2} \frac{(h_1^+)^m [v^{(m)}]_{\alpha_1}}{m!} - \frac{1}{h^2}\sum_{m=0}^{2} \frac{(h_2^+)^m [v^{(m)}]_{\alpha_2}}{m!} + O(h) \quad (20)$$

$$v_{xx}(x_{i+1}) = \frac{v_{i+2} - 2v_{i+1} + v_i}{h^2} + \frac{1}{h^2}\sum_{m=0}^{2} \frac{(h_1^-)^m [v^{(m)}]_{\alpha_1}}{m!} + \frac{1}{h^2}\sum_{m=0}^{2} \frac{(h_2^-)^m [v^{(m)}]_{\alpha_2}}{m!} + O(h) \quad (21)$$

Expressions involving three or more interface crossings could also be derived but we have not found it necessary for the examples considered. Finally, we will also require centered and backwards approximations for $v(x_{i+1/2})$. Thus we have, a) when $x_i \le \alpha < x_{i+1/2}$, with $h^- = x_i - \alpha$

$$v(x_{i+1/2}) = \frac{1}{2}(v_i + v_{i+1}) + \frac{1}{2}\sum_{m=0}^{1} (h^-)^m [v^{(m)}] + O(h^2) \quad (22)$$

b) when $x_{i+1/2} \le \alpha < x_{i+1}$, with $h^+ = x_{i+1} - \alpha$

$$v(x_{i+1/2}) = \frac{1}{2}(v_i + v_{i+1}) - \frac{1}{2}\sum_{m=0}^{1} (h^+)^m [v^{(m)}] + O(h^2) \quad (23)$$

and,
a) when $x_{i-1} \le \alpha < x_i$ with $h^- = x_i - \alpha$,

$$v(x_{i+1/2}) = \frac{3}{2}v_i - \frac{1}{2}v_{i-1} - \frac{1}{2}\sum_{m=0}^{1} (h^-)^m [v^{(m)}] + O(h^2) \quad (24)$$

b) when $x_i \le \alpha < x_{i+1/2}$ with $\bar{h}^+ = x_{i+\frac{1}{2}} - \alpha$,

$$v(x_{i+1/2}) = \frac{3}{2}v_i - \frac{1}{2}v_{i-1} + \sum_{m=0}^{1} (\bar{h}^+)^m [v^{(m)}] + O(h^2)\ . \quad (25)$$

IV. The numerical algorithm

A. Projection method

We employ a non-staggered pressure-increment projection algorithm for the discretization of the Navier-Stokes equations. Several variants of projection methods have been proposed. We choose the particular form presented in [1] which is second order accurate for both velocities and pressure. Given the velocity at time level n, $\boldsymbol{u}^n$, and the pressure at time level $n-1/2$, $p^{n-1/2}$, we compute the velocity, $\boldsymbol{u}^{n+1}$, and pressure, $p^{n+1/2}$, in three steps:

Step 1: Compute an intermediate velocity field $\boldsymbol{u}^*$ by solving

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\triangle t} = -\overline{(\boldsymbol{u}\cdot\nabla\boldsymbol{u})^{n+1/2}} - \overline{\nabla p^{n+1/2}} + \mu\overline{\nabla^2 \boldsymbol{u}^{n+1/2}} + \gamma_1 [\boldsymbol{u}_t]_{\alpha_t} \quad (26)$$

$$\boldsymbol{u}^*|_{\partial\Omega} = \boldsymbol{u}_b^{n+1}$$

where the advective term is extrapolated using the formula,

$$\overline{(\boldsymbol{u}\cdot\nabla\boldsymbol{u})^{n+1/2}} = \frac{3}{2}(\boldsymbol{u}\cdot\nabla_h\boldsymbol{u})^n - \frac{1}{2}(\boldsymbol{u}\cdot\nabla_h\boldsymbol{u})^{n-1} + \gamma_2[\boldsymbol{u}\cdot\nabla\boldsymbol{u}]_{\alpha_t} + \boldsymbol{C}_1 \quad (27)$$

the pressure term is approximated simply as,

$$\overline{\nabla p^{n+1/2}} = \nabla_h p^{n-1/2} + \gamma_3 [\nabla p]_{\alpha_t} + C_2 \quad (28)$$

and the diffusion term is given implicitly as,

$$\overline{\nabla^2 \boldsymbol{u}^{n+1/2}} = \frac{1}{2}(\nabla_h^2 \boldsymbol{u}^* + \nabla_h^2 \boldsymbol{u}^n) + \gamma_4 [\nabla^2 \boldsymbol{u}]_{\alpha_t} + \boldsymbol{C}_3 \quad (29)$$

Step 2: Compute a pressure update ϕ^{n+1} by solving the Poisson equation

$$\nabla_h^2 \phi^{n+1} = \frac{\nabla_h \cdot \boldsymbol{u}^*}{\triangle t} + C_4, \qquad \boldsymbol{n}\cdot\nabla\phi^{n+1}|_{\partial\Omega} = 0 \quad (30)$$

Step 3: Update pressure and velocity field according to,

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \Delta t \nabla_h \phi^{n+1} + \boldsymbol{C}_5 \quad (31)$$

$$p^{n+1/2} = \overline{p^{n+1/2}} + \phi^{n+1} - \frac{\mu}{2}(\nabla_h \cdot \boldsymbol{u}^*) + C_6 \quad (32)$$

Here, $\overline{p^{n+1/2}} = p^{n-1/2} + \gamma_3 [p]_{\alpha_t}$, and $[\,\cdot\,]_{\alpha_t}$ denotes a jump in time which is only non zero when the interface crosses the grid point over the time interval considered. The coefficients $\gamma_i, i = 1, \ldots, 4$, corresponding to the first order corrections in time, are determined from expressions (11), (12) and (22)–(25). At the interface, $[\,\cdot\,]_{\alpha_t} = \pm[\,\cdot\,]_\alpha$, where $[\,\cdot\,]_\alpha$ denotes spatial jump and the sign depends on the motion of the interface. Also, by differentiating $[\boldsymbol{u}] = \boldsymbol{0}$, we obtain $[\boldsymbol{u}_t] = -[\boldsymbol{u}\cdot\nabla\boldsymbol{u}]$. Note that in equation (26) we have used the fact that the velocity is continuous and therefore $[\boldsymbol{u}]_{\alpha_t} = \boldsymbol{0}$. The operators ∇_h and ∇_h^2 are the standard three point central difference operators and $\boldsymbol{C}_i, i = 1, \ldots, 6$, are the spatial correction terms which are non-zero at the points which are near the interface. In step 2 we need an approximation to $\nabla\cdot\boldsymbol{u}^*$ at the boundary points of the rectangular domain. Following [5] we extrapolate the

value of $\nabla \cdot \boldsymbol{u}^*$ at the boundaries. For example, at the left vertical boundary we use

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

In our projection method, we need to solve two Helmholtz equations for $\boldsymbol{u}^*$ in (26) and one Poisson equation for ϕ^{n+1} in (30). Since the correction terms $\boldsymbol{C}_1$ and C_4 only affect the right-hand side of the discrete systems for the Helmholtz and Poisson equations, we can use the fast Fourier transform solver (e.g. [2]).

B. Correction terms for spatial derivatives

In this section, we will show how to evaluate the correction terms $\boldsymbol{C}_i, i = 1, \ldots, 6$ in expressions (27)-(32). Let $C\{v\}$ denote the correction term for a quantity v, for example, from (11) we can write

$$C\{v_x(x_i)\} = -\frac{1}{2h}\left([v] + h^+[v_x] + \frac{(h^+)^2}{2}[v_{xx}]\right) \qquad (33)$$

Then the correction terms $\boldsymbol{C}_1$-C_6 in (26)-(32) are evaluated as follows:

$$\boldsymbol{C}_1 = \frac{3}{2}C\{(\boldsymbol{u}\cdot\nabla\boldsymbol{u})^n\} - \frac{1}{2}C\{(\boldsymbol{u}\cdot\nabla\boldsymbol{u})^{n-1}\} \qquad (34)$$

$$C_2 = C\{\nabla p^{n-1/2}\} \qquad (35)$$

$$\boldsymbol{C}_3 = \frac{1}{2}\left(C\{\nabla_h^2\boldsymbol{u}^*\} + C\{\nabla_h^2\boldsymbol{u}^n\}\right) \qquad (36)$$

$$C_4 = \frac{C\{\nabla\cdot\boldsymbol{u}^*\}}{\triangle t} + C\{\nabla^2 p^{n-\frac{1}{2}}\} - C\{\nabla^2 p^{n+\frac{1}{2}}\} + \gamma_3[\nabla^2 p]_{\alpha_t} \qquad (37)$$

$$\boldsymbol{C}_5 = -\triangle t\left(C\{\nabla p^{n+\frac{1}{2}}\} - C\{\nabla p^{n-\frac{1}{2}}\}\right) + \Delta t\gamma_3[\nabla p]_{\alpha_t} \qquad (38)$$

$$C_6 = -\frac{\mu}{2}C\{\nabla\cdot\boldsymbol{u}^*\} \qquad (39)$$

All the correction terms are included at least to first order accuracy. As explained in [3], the overall second order accuracy of the scheme is maintained provided only the singular points are treated with a first order scheme. This can be intuitively understood by realizing that when the mesh is refined, the area of the domain represented by these points is reduced.

C. Implicit scheme for moving interface

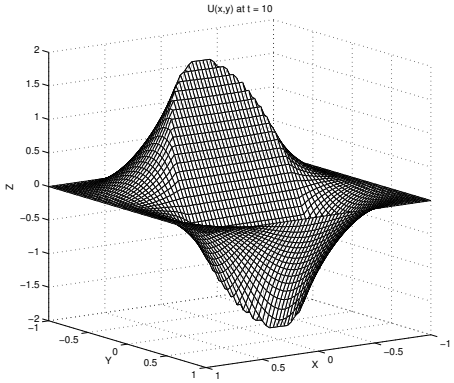
In [4], it was found that when the position of the interface is advanced using an explicit method, the size of the timestep required for stability is very small. For this reason they recommend the use of an implicit scheme. Following [4], we update the position of the control points according to,

$$\boldsymbol{X}^{n+1} = \boldsymbol{X}^n + \frac{1}{2}\Delta t\left(\boldsymbol{u}\left(\boldsymbol{X}^n\right) + \boldsymbol{u}\left(\boldsymbol{X}^{n+1}\right)\right) \qquad (40)$$

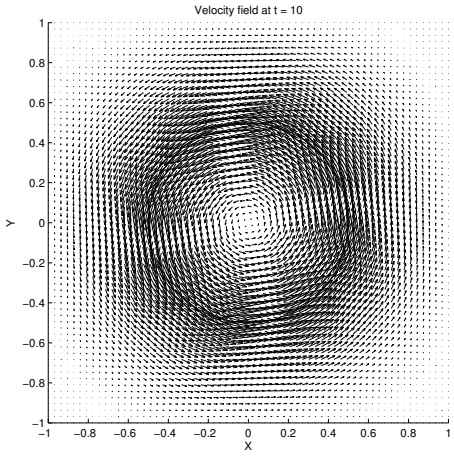
This equation is implicit and couples the motion of the interface with the solution at all grid points. Therefore, at each time step we need to solve a non-linear system of equations for the position of the control points of the form $g\left(\boldsymbol{X}^{n+1}\right) = 0$, where

$$g\left(\boldsymbol{X}\right) = \boldsymbol{X} - \boldsymbol{X}^n - \frac{1}{2}\triangle t\left(\boldsymbol{u}\left(\boldsymbol{X}^n\right) + \boldsymbol{u}\left(\boldsymbol{X}\right)\right) \qquad (41)$$

Since the computation of the exact linearization of $g(\boldsymbol{X})$ would be very expensive, we have implemented a quasi-Newton method in which the jacobian matrix is updated using a BFGS method. In practice, this approach requires only a few iterations as the solution at the previous timestep provides a very good initial guess for the iteration.



(a)



(b)

Fig. 1. Velocity field at time t = 10 with a 64×64 grid, $\mu = 0.02$, $f_1 = 0$, $f_2 = 10\mu$. 1(a) Plot of the x component of velocity field at time t = 10. 1(b) Plot of velocity field at time t = 10.

V. Numerical results

In order to illustrate the capabilities of the method described, we present some numerical results for two test problems which involve immersed boundaries.

A. Circular Flow

In this example, we consider a fixed interface problem with non-smooth velocity [5]. The interface is a circle, $r = \frac{1}{2}$, at the center of the computational domain $[-1,\ 1] \times [-1,\ 1]$. The normal and tangential stress are $f_1 = 0$ and $f_2 = 10\mu$, respectively, and the viscosity is $\mu = 0.02$. The initial velocity and pressure are taken to be zero on the square domain. Since $f_1 = 0$ and f_2 is constant, the pressure and its derivatives are continuous across the interface. On the other hand, the normal derivative of the tangential component of the velocity is discontinuous. This can be observed from figures 1(a) and 2. Figure 1(b) shows the steady state solution which corresponds to a rigid body motion inside the interface.

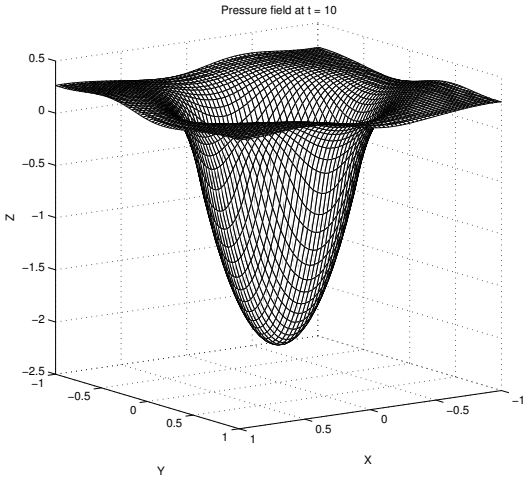


Fig. 2. Pressure distribution at steady state. Note that the pressure and its derivatives are continuous.

B. Membrane with surface tension

In this example, we consider a deformable interface problem with surface tension [4]. The initial interface is an ellipse with major and minor axes a = 0.75, b = 0.5, respectively. The force density is now given by

$$\boldsymbol{f}(s,t) = \gamma \frac{\partial^2}{\partial s^2} \boldsymbol{X}(s,t)\ , \tag{42}$$

which can be seen to be, at all points, normal to the interface. The initial velocity and pressure are set to zero. The computational domain is $[-3,\ 3]\times[-3,\ 3]$, and we set $\rho = 1$ and $\mu = 0.1$ throughout the domain. In our test, we take $\gamma = 10$ and $\boldsymbol{u}|_{\partial\Omega} = \boldsymbol{0}$. In figure 5, we plot the location of the interface at different times with a 160 by 160 grid and 50 control points. The interface oscillates as it settles down to the equilibrium state. The evolution of the major and minor axes is shown in figure 3. Figure 4 shows the evolution of the major and minor axes of the ellipse when $\mu = 1$ and $\gamma = 1$. In this case, the Reynolds number is smaller and the interface relaxes gradually to the equilibrium state without oscillations. In the equilibrium state, the interface is a circle, the velocity is zero everywhere and the pressure has two different constant values inside and outside the interface, as show in figure 6.

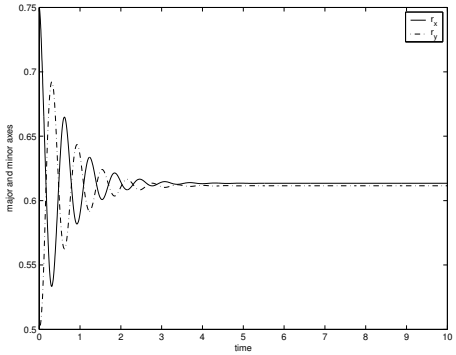


Fig. 3. The time evolution of the ellipse axes r_x and r_y with $\mu = 0.1$ and $\gamma = 10$. The interface oscillates as it converges to the equilibrium state

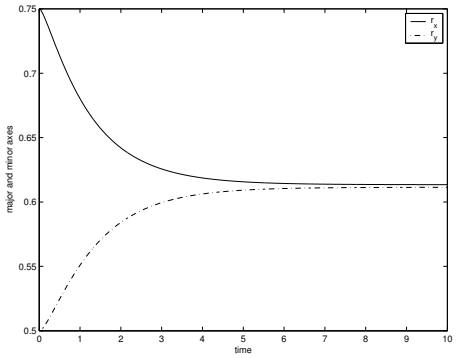


Fig. 4. The time evolution of the ellipse axes r_x and r_y with $\mu = 1$ and $\gamma = 1$. The interface relaxes gradually to the equilibrium state without oscilaltion.

VI. Conclusion

We have presented a formally second order accurate immersed interface method for the solution of the incompressible Navier-Stokes equations with moving interfaces. The implementation has been tested with two simple examples involving rotational flow and a moving interface with surface tension. The method gives a very good resolution of the flow near the discontinuity. The work presented here represents a modest extension of that presented in [4] and [5], but currently work is underway to further extend the current approach. In particular, we plan to extend the current method to handle rigid boundaries and to deal with fluids of different density and viscosity. Future work will extend this approach to three dimensions.

References

[1] D. L. Brown, R. Cortez, and M. L. Minion, Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comput. Phys.* **168**, 464 (2001).

[2] J. Adams, P. Swarztrauber and R. Sweet, FISHPACK: Efficient FORTRAN Subprograms for the Solution of Separable Eliptic Partial Differential Equations, National Center for Athmospheric Research, 1999, http://www.scd.ucar.edu/css/software/fishpack/.

[3] 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).

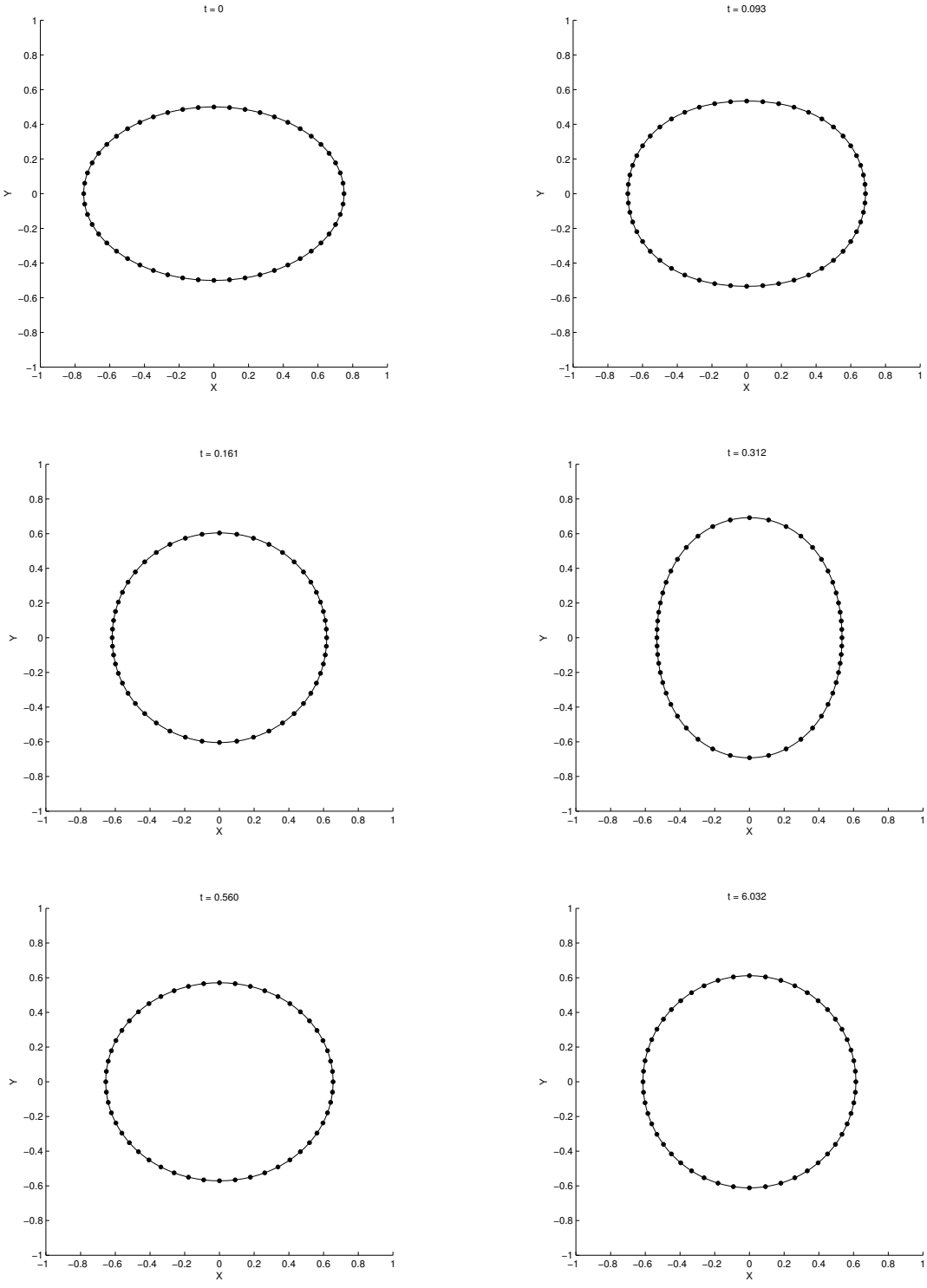


Fig. 5. Location of the interface at different times for $\mu = 0.1$ and $\gamma = 10$.

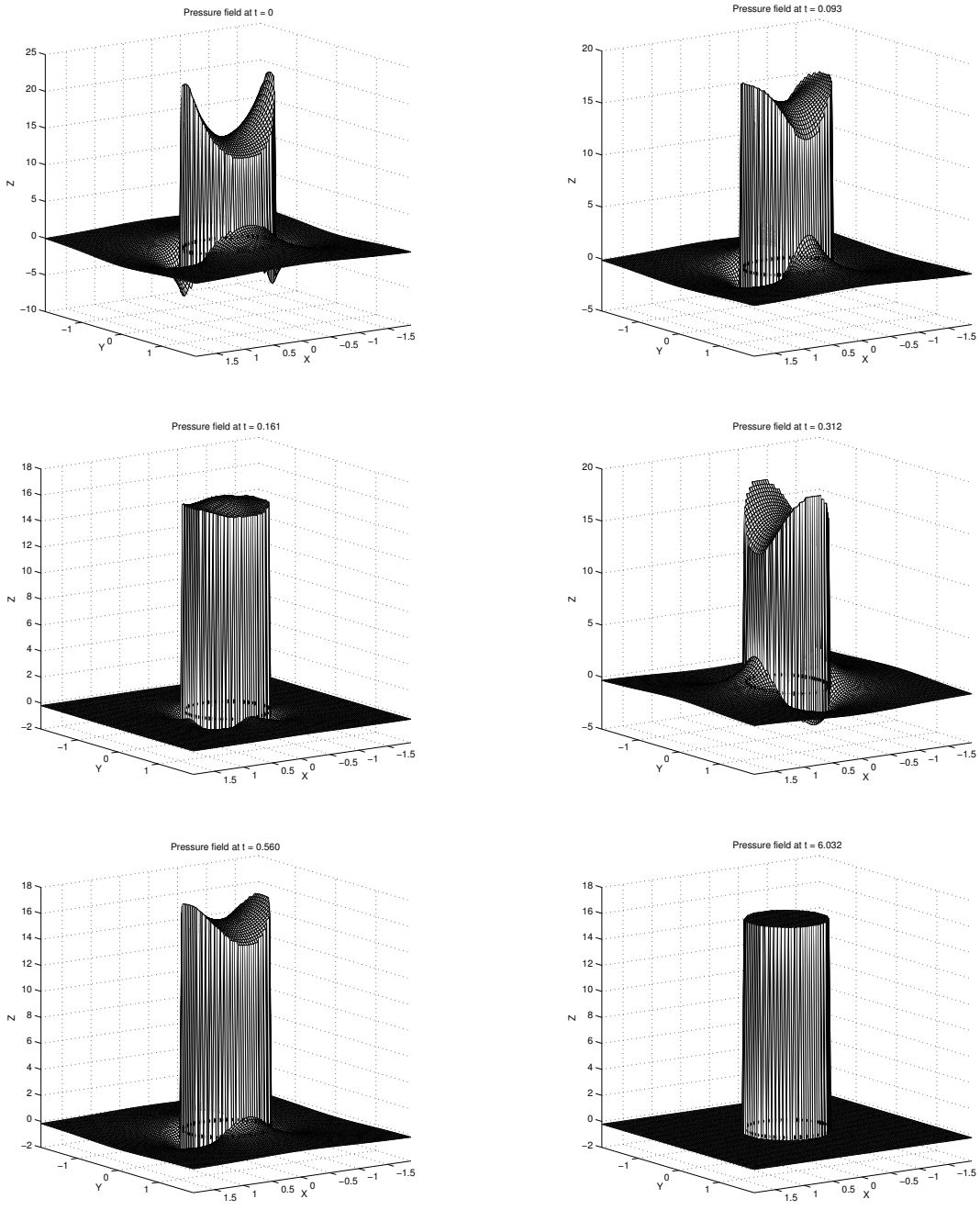


Fig. 6. Pressure distribution at different times for $\mu = 0.1$ and $\gamma = 10$.

[4] 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).

[5] Z. Li and M. C. Lai, The immersed interface method for the Navier-Stokes equations with singular forces, *J. Comput. Phys.* **171**, 822 (2001).

[6] L. Lee, An immersed interface method for the incompressible Navier-Stokes equations, Ph.D. thesis (University of Washington, 2002).

[7] C. S. Peskin, The immersed boundary method, *Acta Numerica* (2002), pp. 479-517.

[8] A. Wiegmann and K. P. Bube, The explicit-jump immersed interface method: Finite difference methods for PDEs with piecewise smooth solutions, *SIAM J. Numer. Anal.* **37**, No. 3, 827 (1997).

[9] Z. Li, Immersed interface methods for moving interface problems, *Numer. Algorithms* **14**, 269 (1997).