A semi-implicit augmented IIM for Navier-Stokes equations with open and traction boundary conditions

Zhilin Li* Qin Cai† Hongkai Zhao‡ Ray Luo§

March 29, 2013

Abstract

In this paper, a new Navier-Stokes solver based on a finite difference approximation is proposed to solve incompressible flows on irregular domains with open and traction boundary conditions, which can be applied to simulations of fluid structure interaction, implicit solvent model for biomolecular applications and other free boundary or interface problems. For this type of problem, the projection method and the augmented immersed interface method (IIM) do not work well or does not work at all. The proposed new Navier-Stokes solver is based on the local pressure boundary method, and a semi-implicit augmented IIM so that a fast Poisson solver can be used. The time discretization is based on a second order multi-step method. Numerical tests with exact solutions are presented to validate the accuracy of the method. Application to fluid structure interaction between an imcompressible fluid and a compressible gas bubble is also presented.

keywords: Navier-Stokes equations, finite difference approximation, irregular domain, open and traction boundary condition, local pressure boundary condition, augmented immersed interface method (IIM).

AMS Subject Classification 2000 65M06, 65M85, 76M20.

1 Introduction

One of our original motivations of this work is to develop new bio-molecular solvation methods for atomistic simulations. Efficient atomistic simulation of large and complex bio-molecular systems is

*Center for Research in Scientific Computation (CRSC) and Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA

†Departments of Biomedical Engineering and Molecular Biology and Biochemistry, University of California, Irvine, CA 92697

‡Department of Mathematics, University of California, Irvine, CA 92697

§Departments of Molecular Biology and Biochemistry and Biomedical Engineering, University of California, Irvine, CA 92697

still one of the remaining challenges in computational molecular biology. One approach is to model the solvation energetics in a mean-field manner by treating the solvent molecules collectively as a continuum. To further improve the quality of the continuum solvent models, we proposed to further model the solvent collective motion as that of an incompressible fluid governed by the Navier-Stokes equations (NSE),

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \mu\Delta\mathbf{u} + \mathbf{G}, \quad \mathbf{x}\in R\setminus\Omega, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \quad \mathbf{x}\in R\setminus\Omega, \qquad \mathbf{u}(\mathbf{x},t)\Big|_{\partial R} = u_b(\mathbf{x},t), \qquad \mathbf{u}(\mathbf{x},0) = u_0(\mathbf{x}), \tag{2}$$

where $\mathbf{G}(x,y,t)$ is a body force that can include van der Waals and electrostatic forces (due to the existence of biomolecules), R is the computation domain (often rectangular), $\mathbf{u} = (u,v)$ is the fluid velocity, p is the pressure, and $\Omega \subset R$ is an inclusion, such as a biomolecule or a gas bubble to begin with, see Figure 1 for an illustration. ρ and μ are the fluid density and viscosity, respectively. Other boundary conditions such as Neumann or mixed boundary condition can also be prescribed along the outer boundary ∂R. Along the interior boundary $\partial\Omega$, there is typically a physical free boundary condition. In the case that Ω represents a gas bubble, we have

$$\begin{cases} \mathbf{n}^T\cdot\mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\cdot\mathbf{n} = p - p_{air} - \gamma\kappa, \\ \boldsymbol{\tau}^T\cdot\mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)\cdot\mathbf{n} = 0, \end{cases} \quad \text{on } \partial\Omega. \tag{3}$$

Such a boundary condition is also called a traction, or open boundary condition if the right hand sides are all zero [17, 18]. We refer the reader to [17, 5] and the references therein for the well-posedness of the problem. The velocity is not necessarily zero along boundaries $\partial\Omega$ and ∂R, but it is consistent with the incompressibility condition for the fluid, i.e., $\int_{\partial\Omega}\mathbf{u}\cdot\mathbf{n}\,ds + \int_{\partial R}\mathbf{u}\cdot\mathbf{n}\,ds = 0$. In Section 3, we give an example about how to enforce this consistency condition.

We will consider both the fixed domain and the free boundary problem. For the free boundary problem, the evolution of equation for the boundary problem $\partial\Omega$ is

$$\frac{d\mathbf{X}}{dt} = \mathbf{u}(\mathbf{X},t), \qquad \mathbf{X}(t)\in\partial\Omega(t). \tag{4}$$

Numerically we use the level set method to capture the free boundary $\partial\Omega$.

While there are many papers in the literature about numerical methods for free boundary problem, flow problems on irregular domain, few deal with open and traction boundary conditions. For small to medium Reynolds numbers, the projection method, for example, [2, 3, 4] is one of the most popular methods to solve the NSE because its stability and accuracy. The NSE can be solved by solving several Helmholtz/Poisson equations. Nevertheless, the projection method is based on Helmholtz-Hodge decomposition. The projection method may not work well for open and traction boundary conditions, see for example, [5]. In fact, There are some similarities between the problem studied in this paper and the drop spreading problem discussed in [10] in which we have a free boundary condition (traction boundary condition) on the drop surface but a Dirichlet BC at two points. The Dirichlet boundary condition does make a difference for numerical simulation. If we

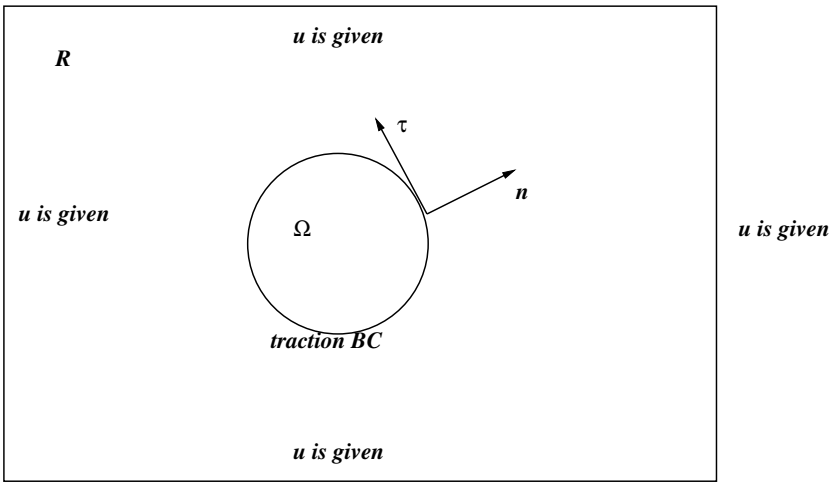

Figure 1: A diagram of the set-up of the problem. The open traction boundary condition is defined along the boundary $\partial\Omega$; Dirichlet boundary conditions are defined along the boundary ∂R.

take the Dirichlet BC away with a closed bubble boundary and the entire free boundary condition, the solution eventually blows up. This is one of our motivations to develop different but more stable approach.

Another type of methods are called local pressure boundary conditions [8, 9]. There are similarities between local pressure boundary conditions and the local vorticity boundary conditions. The simplicity of the local pressure boundary condition approach and its easy application to more general flow settings make the resulting scheme an attractive alternative to the projection type methods for solving incompressible Navier-Stokes equations in the velocity-pressure formulation. This approach is also implemented using a finite element formulation [17]. The key idea in the local pressure boundary condition approach is to solve the pressure from the momentum equation. A Neumann or Dirichlet boundary condition and the incompressibility condition are used to evaluate the pressure boundary condition for the pressure, which is the key to the stability. In the original local pressure boundary method [8, 9], the Neumann boundary condition is derived while in this paper, a Dirichlet boundary condition is used. Note that the original local pressure boundary method [8, 9] does not directly apply to traction boundary conditions for which derivatives of the velocity and the pressure are coupled together. A direct interpolation of the velocity to get the Laplacian from the previous step is likely unstable. Note that most of the methods using finite difference discretization are based on rectangular domains. The same treatments are often quite sensitive to curved boundaries.

Another challenging is how to solve NSE on irregular domain efficiently. In [7], the authors proposed the augmented immersed interface method (AIIM) for NSE on irregular domain. The most important advantage of AIIM is that we can treat the irregular domain problem on a rectangular domain so that fast solvers for Poisson/Helmholtz equations or Navier-Stokes equations can be applied after we introduce a co-dimension one quantity along the irregular boundary. The question

here is how to combine the AIIM with the new local pressure boundary condition approach. In this paper, we use the AIIM for solving the velocity, then use our existing fast Poisson solver (also based on AIIM) for irregular domains to solve the pressure.

In our previous work on AIIM for Navier-Stokes equations, we were using the traditional projection method that is based on Crank-Nicholson (trapezoidal) type of scheme for the prediction step. The augmented variable is also split in two time steps. One of obvious advantage of this approach is that the method can be second order both in space and time with only two step quantities and it is relatively simple to implement. However, it is known that the Crank-Nicholson type discretizetion is marginal stable which may not be ideal for non-linear problems or curved boundaries. Thus it is more stable if we can use a fully implicit discretization for the diffusion term. This is why we called our discretization as a semi-implicit method since other terms in the Navier-Stokes equations are still discretized explicitly.

The rest of papers are organized as follows. In the next section, we outline the main steps of our algorithm and explain the new ideas of our method and the rationality. In Section 3, we validate our method using examples that have exact solutions. Then we present some numerical simulations of fluid structure interaction of a fluid with a gas bubble. We conclude in the last section.

2 The AIIM using local pressure boundary condition

As mentioned before, we assume that the domain R is a rectangle $[a,\ b] \times [c,\ d]$ with a gas bubble inclusion Ω. The spatial spacing is chosen as $h_x = (b-a)/M$, $h_y = (d-c)/N$, where M and N are the number of grid lines in the x and y directions, respectively. Let the time step size be Δt. We use a standard uniform Cartesian grid for simplicity. Here we emphasize the spatial discretization. The time discretization is based on a multi-step method.

From one time step t^k to the next time level t^{k+1}, our proposed new algorithm has the following steps.

Step 1: Solve the velocity with a fixed free boundary using the AIIM.

$$\frac{3\mathbf{u}^{k+1} - 4\mathbf{u}^k + \mathbf{u}^{k-1}}{2\Delta t} + (\mathbf{u} \cdot \nabla\, \mathbf{u})^{k+1} + \nabla \tilde{p}^{k+1} = \mu \Delta\, \mathbf{u}^{k+1} + \mathbf{G}^{k+1}, \tag{5}$$

$$\left(-\tilde{p}^{k+1} + \mathbf{n}^T \cdot \mu \left(\nabla \mathbf{u}^{\mathbf{k+1}} + \nabla \left(\mathbf{u}^{k+1}\right)^T\right) \cdot \mathbf{n}\ + \gamma\kappa + p_{air}\right)\Big|_{\partial\Omega} = 0, \tag{6}$$

$$\boldsymbol{\tau}^T \cdot \mu \left(\nabla \mathbf{u}^{k+1} + \nabla \left(\mathbf{u}^{k+1}\right)^T\right) \cdot \mathbf{n}\Big|_{\partial\Omega} = 0, \tag{7}$$

$$\left[\mathbf{u}^{k+1}\right]_{\partial\Omega} = 0, \qquad \left[\frac{\partial \mathbf{u}^{k+1}}{\partial n}\right]_{\partial\Omega} = \mathbf{q}^{k+1}, \tag{8}$$

where

$$\tilde{p}^{k+1} = 2p^k - p^{k-1}, \tag{9}$$

$$(\mathbf{u}\cdot\nabla\,\mathbf{u})^{k+1} = 2\,(\mathbf{u}\cdot\nabla\,\mathbf{u})^{k} - (\mathbf{u}\cdot\nabla\,\mathbf{u})^{k-1} \tag{10}$$

$$\mathbf{G}^{k+1} = 0, \qquad \text{if } (x,y)\in\Omega, \tag{11}$$

κ is the curvature of the free boundary $\partial\Omega$, and γ is the coefficient of the surface tension.

Note that this step is to approximate the velocity at time step $k+1$. It is obvious that the discretization of the momentum equations is second order accurate both in space and time, see also [5]. The momentum equation is extended to the entire rectangular domain so that a fast Helmholtz solver, say from [1] can be applied. The equivalent Helmholtz can be written as

$$\Delta\,\mathbf{u}^{k+1} - \frac{3}{2\mu\Delta t}\,\mathbf{u}^{k+1} = \begin{cases} \dfrac{1}{\mu}\left((\mathbf{u}\cdot\nabla\,\mathbf{u})^{k+1} + \nabla\tilde{p}^{k+1} + \dfrac{-4\mathbf{u}^k + \mathbf{u}^{k-1}}{2\Delta t} - \mathbf{G}^{k+1}\right) & \text{if } (x,y)\in R\setminus\Omega, \\ \mathbf{0} & \text{if } (x,y)\in\Omega. \end{cases}$$

The pressure $\tilde{p}^{k+1}$ is extrapolated from the pressure at previous two steps and so is the nonlinear term $(\mathbf{u}\cdot\nabla\,\mathbf{u})^{k+1}$. The augmented variable $\mathbf{q}^{k+1}$ which is the jump in the normal derivative of the velocity is chosen such that the velocity $\mathbf{u}^{k+1}$ satisfy the open traction condition (3) with an approximated p^{k+1}. We refer the readers to [12, 13, 7] for the AIIM for Navier-Stokes equations. Note that, here we use the backward Euler's method instead of the Crank-Nicholson type discretization for the stability consideration as we explained in the Introduction section. This seems to be important for the stability of the algorithm for open and traction boundary conditions.

Since we are only interested in the quantities outside of Ω (the gas bubbles), $\tilde{p}^{k+1}$ and $(\mathbf{u}\cdot\nabla\,\mathbf{u})^{k+1}$ can be moved to the right hand side as a source term for the Helmholtz equation of $\mathbf{u}^{k+1}$. The quantities of $\tilde{p}^{k+1}$ and $(\mathbf{u}\cdot\nabla\,\mathbf{u})^{k+1}$ are approximated use standard second order central finite difference schemes if they are two grid distance away from the boundary. At grid points near or on the boundary, they can be approximated using a first order approximation without affecting second order accuracy, see for example [12]. We will see that the coefficient matrix for the augmented variable $\mathbf{q}^{k+1}$ is well-conditioned with almost $O(1)$ condition number.

Step 2: Solve the pressure from the equation obtained after applying the gradient operator to the momentum equation,

$$\Delta p^{k+1} = \begin{cases} -\nabla\cdot(\mathbf{u}\cdot\nabla\,\mathbf{u})^{k+1} + \nabla\cdot\mathbf{G}^{k+1} & \text{if } (x,y)\in R\setminus\Omega, \\ 0 & \text{if } (x,y)\in\Omega, \end{cases} \tag{12}$$

$$p^{k+1}\big|_{\partial\Omega} = \left(\mathbf{n}^T\cdot\mu\left(\nabla\mathbf{u}^{\mathbf{k+1}} + \nabla\left(\mathbf{u}^{k+1}\right)^T\right)\cdot\mathbf{n} + \gamma\kappa + p_{air}\right)\Big|_{\partial\Omega}, \tag{13}$$

where the boundary condition is satisfied from the pressure outside of the air bubble. This is a Poisson equation defined on an exterior irregular domain. We simply call the IIM packages for Poisson equations on irregular domain [11], see also [6, 12, 16] for more technical details.

2.1 Some discretization details

In the AIIM, the key step is to obtain the matrix-vector multiplication for the linear system of equations for the augmented variable. It contains two major steps. One step is to solve the Navier-Stokes equation given the augmented variable; The second step is to evaluate the residual of the boundary condition, we refer the readers for Chapter 6 of the book [12] for the detail.

Given a value of the augmented variable, here is the jump of normal derivative of the velocity across the air bubble boundary. We briefly explain how to discretize the Helmholtz and Poisson equations. Consider a grid point (x_i, y_j), if the boundary does not cut the centered five-point finite difference stencil, then we use the standard second order central finite difference scheme to discretize the Helmholtz and Poisson equations. Those grid points are call regular. Otherwise a grid point (x_i, y_j) is called irregular at which the boundary $\partial\Omega$ cuts through the centered five-point stencil. At an irregular grid point (x_i, y_j), assume that the boundary $\partial\Omega$ cuts the grid line at (x_i^*, y_j), $x_i^* = x_i + \alpha h_x$, $-1 \le \alpha \le 1$, then the second order partial derivative in x can be approximated by

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u_{i-1,j} - 2u_{i-1,j} + u_{i+1,j}}{h_x^2} - \frac{C(x_i, \alpha)}{h_x^2} + O(h), \tag{14}$$

where

$$\frac{C(x_i, \alpha)}{h_x^2} = [u] + [u_x]\,(1 - |\alpha|)\,h + [u_{xx}]\frac{(1 - |\alpha|)^2\,h_x^2}{2}, \tag{15}$$

see Lemma 1 in [14]. From $[\frac{\partial \mathbf{u}}{\partial n}] = \mathbf{q}$, we can get $[\frac{\partial \mathbf{u}}{\partial x}]$, $[\frac{\partial^2 \mathbf{u}}{\partial x^2}]$ as described below. For simplicity of notation, we have dropped the index $k+1$.

Let (X, Y) be a point on the interface $\partial\Omega$ which is a smooth closed 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, see Figure 1. we define the local coordinates at (X, Y) as

$$\begin{aligned} \xi &= (x - X)\cos\theta + (y - Y)\sin\theta, \\ \eta &= -(x - X)\sin\theta + (y - Y)\cos\theta. \end{aligned} \tag{16}$$

Then $\partial\Omega$ 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 $\partial\Omega$ at $(0, 0)$. The following interface relations at (X, Y) can be derived from the NSE and the interface conditions.

$$\begin{aligned} &[\mathbf{u}] = \mathbf{0}, \qquad [\mathbf{u}_\xi] = \mathbf{q}, \qquad [\mathbf{u}_\eta] = \mathbf{0}, \\ &[\mathbf{u}_{\eta\eta}] = -\kappa\,\mathbf{q}, \qquad [\mathbf{u}_{\xi\eta}] = -\frac{\partial \mathbf{q}}{\partial \eta}, \\ &[\mathbf{u}_{\xi\xi}] = -[\mathbf{u}_{\eta\eta}] + [p_\xi]\,\mathbf{n} + [p_\eta]\,\boldsymbol{\tau} + [\mathbf{u}_\xi]\mathbf{u}\cdot\mathbf{n} - [\mathbf{G}]. \end{aligned} \tag{17}$$

Once we have the jump relations in the local coordinate system, then it is easy to transform them

back in the original coordinate system and get the following relations:

$$\begin{aligned}
[u_x] &= [u_\xi]\cos\theta - [u_\eta]\sin\theta, \\
[u_y] &= [u_\xi]\sin\theta + [u_\eta]\cos\theta, \\
[u_{xx}] &= [u_{\xi\xi}]\cos^2\theta - 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\sin^2\theta, \\
[u_{yy}] &= [u_{\xi\xi}]\sin^2\theta + 2[u_{\xi\eta}]\cos\theta\sin\theta + [u_{\eta\eta}]\cos^2\theta.
\end{aligned} \tag{18}$$

With these jump condition, we can approximate u_x, u_y, u_{xx}, u_{xy} according to (14)-(15) to get the finite difference equations for the Helmholtz and Poisson equations.

2.2 Pressure boundary condition along outer boundary

We need a boundary condition along outer boundary ∂R for the Poisson equation for the pressure. Often it is an approximates normal derivative condition. However, with a prescribed velocity, the incompressibility condition, and the NSE equations, we can use the techniques described in [9, 15] to get more accurate normal derivative boundary condition for the pressure. We use the side $x = a$ to explain the process. For simplicity and preciseness, we will ignore the time index. Since we know the velocity $\mathbf{u} = (u, v)$ along $x = a$ which is a function of y, we also know u_y and v_y. From the incompressibility condition $u_x + v_y = 0$, we also know u_x which is $-v_y$ along $x = a$. Thus along $x = a$, we have

$$(\mathbf{u}\cdot\nabla)\mathbf{u}\cdot\mathbf{n} = uu_x + vu_y = -uv_y + vu_y, \tag{19}$$

which is known quantity. The key part is how to approximate the Laplacian of $\mathbf{u}\cdot\mathbf{n} = u_{xx} + u_{yy}$. Since we know u along $x = a$, we just need to approximate u_{xx}. Note that

$$u(a+h, y) = u(a, y) + u_x(a, y)h + \frac{h^2}{2}u_{xx}(a, y) + O(h^2),$$

from which we get an approximation for u_{xx} in terms of the values of u at the grid point

$$u_{xx}(a, y_j) \approx 2\,\frac{u_{1,j} - u_{0,j} - v_y(a, y_j)h}{h^2}. \tag{20}$$

The normal derivative of the pressure along $x = a$ can be approximated by

$$p_x(a, y_j) = \mu u_{xx}(a, y_j) + G_1 - u_t(a, y_j) - (\mathbf{u}\cdot\nabla)\mathbf{u}(a, y_j), \tag{21}$$

where G_1 is the x- component of the external force $\mathbf{G}$.

3 Numerical examples

As a first numerical test for our proposed method, we consider an example in a stationary irregular domain in which the exact solution is known analytically. We use it as an accuracy check. The

analytic solution is

$$u(x,y,t) = w(t)\left(\frac{y}{\sqrt{x^2+y^2}} - 2y\right), \tag{22}$$

$$v(x,y,t) = w(t)\left(\frac{x}{\sqrt{x^2+y^2}} + 2x\right), \tag{23}$$

$$p(x,y,t) = \left((x^2+y^2)^2 - \frac{1}{4}\right)^2. \tag{24}$$

The domain of the interest is the domain bounded by $x^2+y^2 \geq 1/4$ and $-1 \leq x, y \leq 1$. The source term $\mathbf{G}$ is derived directly from the exact solution. The Dirichlet boundary condition is prescribed also using the exact solution along the rectangular boundary.

In Table 1, we show the grid refinement analysis to check the order of the accuracy of our method. In the test, we take $T = 5$, $w(t) = 1 - e^{-t}$. Since we are interested in the computed solutions in the domain Ω, we set

$$\|E_u\|_\infty = \max_{r_{ij}\geq 1/2}\{|U_{ij}^k - u(x_i,y_j,T)| + \max_{r_{ij}\geq 1/2}\{|V_{ij}^k - v(x_i,y_j,T)|\},$$

$$\|E_p\|_\infty = \max_{r_{ij}\geq 1/2}\{|P_{ij}^k - p(x_i,y_j,T)|\},$$

to be the error in the velocity and the pressure at time T. The number *order* is the approximated order of accuracy from the two consecutive errors,

$$order_u = \frac{\log\left(\|E_u\|_{\infty,2N}/\|E_u\|_{\infty,N}\right)}{\log 2}; \quad order_p = \frac{\log\left(\|E_p\|_{\infty,2N}/\|E_p\|_{\infty,N}\right)}{\log 2}. \tag{25}$$

Second order accuracy is clearly seen for both the velocity and the pressure.

N	$\|E_u\|_\infty$	$order_u$	$\|E_p\|_\infty$	$order_p$
16	$3.2956\,10^{-2}$		$3.0911\,10^{-1}$	
32	$5.9132\,10^{-3}$	2.4786	$8.1862\,10^{-2}$	1.9168
64	$1.1330\,10^{-3}$	2.3838	$2.0834\,10^{-2}$	1.9743
128	$2.6351\,10^{-4}$	2.1042	$5.2709\,10^{-3}$	1.9828
256	$7.5585\,10^{-5}$	1.8017	$1.3087\,10^{-3}$	2.0099
512	$1.8711\,10^{-5}$	2.0142	$3.2783\,10^{-4}$	1.9971

Table 1: A grid refinement analysis against the exact solution at a final time $T = 5$ with $w(t) = 1 - e^{-t}$, where $\|E_\mathbf{u}\|_\infty$ is the sum of the maximal error in the velocity component u and v, *order* is the approximated convergence order computed from the two consecutive errors.

It is interesting to note that the convergence seems to be independent of the initial data for long time computations. If we start with $\mathbf{u}^0 = \mathbf{u}(\mathbf{x}, 0)/2.5$, $\mathbf{u}^{-1} = \mathbf{u}(\mathbf{x}, -\Delta t)/2.5$, $p^0 = 0$, $p^{-1} = 0$, we still get almost the exact same results if the final time is large enough.

In Table 2, we show the grid refinement result for $w(t) = \sin t$. In this test, there is no steady state solution since the source term depends on time t all the time. We still see clean second order accuracy in the infinity norm.

N	$\|E_u\|_\infty$	$order_u$	$\|E_p\|_\infty$	$order_p$
16	$3.2727\,10^{-1}$		$3.0214\,10^{-1}$	
32	$1.8534\,10^{-2}$	4.1422	$7.2537\,10^{-2}$	2.0584
64	$5.2028\,10^{-3}$	1.8328	$1.8357\,10^{-2}$	1.9824
128	$1.1859\,10^{-3}$	2.1333	$4.5648\,10^{-3}$	2.0077
256	$2.9045\,10^{-4}$	2.0296	$1.1103\,10^{-3}$	2.0396
512	$6.2627\,10^{-5}$	2.2134	$2.7829\,10^{-4}$	1.9962

Table 2: A grid refinement analysis against the exact solution at a final time $T = 5$ with $w(t) = \sin t$.

Surprisingly, we also got almost the same result if we do not use the exact initial data but with $\mathbf{u}^0 = \mathbf{u}(\mathbf{x}, 0)/2.5$, $\mathbf{u}^{-1} = \mathbf{u}(\mathbf{x}, -\Delta t)/2.5$, $p^0 = 0$, $p^{-1} = 0$.

3.1 An example for fluid and air bubble interaction

Here we use a simple model to simulate fluid and air bubble interaction. The fluid is incompressible and is modeled by the NSE. The air bubble is compressible and is governed by ideal gas law $p_{gas}(t) = \lambda/V^\gamma(t)$, where $p_{gas}(t)$ is the pressure in the gas bubble, $V(t)$ is the volume of the bubble, and λ is a constant. Again we first test our code for a steady state case with $\gamma = 1$ in which we have an analytic solution,

$$u(x, y, t) = \frac{x}{x^2 + y^2}, \tag{26}$$

$$v(x, y, t) = \frac{y}{x^2 + y^2}, \tag{27}$$

$$p(x, y, t) = -\frac{1}{2(x^2 + y^2)}, \tag{28}$$

$$\mathbf{G} = \mathbf{0}, \tag{29}$$

outside the unit circle $r = 1$ but within the domain $[-2,\ 2] \times [-2,\ 2]$. The Dirichlet boundary condition is prescribed using the exact solution along the rectangular boundary. Note that $\mathbf{n} \cdot \mathbf{u} \neq 0$ at the boundary $r = 1$ in this example. In Table 3, we show the grid refinement analysis at $T = 1.5$. We can see second order convergence for all the variables.

N	$\|E_u\|_\infty$	$order_u$	$\|E_p\|_\infty$	$order_p$
16	$4.844\,10^{-2}$		$5.0781\,10^{-3}$	
32	$1.5468\,10^{-2}$	1.6470	$1.6917\,10^{-3}$	1.5859
64	$3.8509\,10^{-3}$	2.0060	$5.3791\,10^{-4}$	1.6530
128	$1.1232\,10^{-3}$	1.7775	$1.3885\,10^{-4}$	1.9538
256	$3.0374\,10^{-4}$	1.8868	$3.6570\,10^{-5}$	1.9248
512	$7.8641\,10^{-5}$	1.9495	$9.0872\,10^{-6}$	2.0088

Table 3: A grid refinement analysis against the exact solution at a final time $T = 1.5$.

Now we show the test results for the dynamic case. To be more realistic, we now use $\gamma = 1.4$. We start with a circular boundary and set $\sigma = 1, \lambda = \pi$. The initial velocity and pressure are all set to zero. The initial boundary was set as the zero level set of a Lipschitz continuous function $\varphi(\mathbf{x}, t)$ at $t = 0$, often the signed distance function (or approximated) to the boundary $\partial\Omega$. The level set is updated by the level set equation

$$\varphi_t + \nabla\varphi \cdot \mathbf{u} = 0 \tag{30}$$

along with a re-initialization process.

Enforce the consistent boundary condition

To ensure the consistent condition $\int_{\partial R} + \int_{\partial\Omega} \mathbf{u} \cdot \mathrm{n}\, ds = 0$. We adjust the outer Dirichlet boundary condition at each time step. In many applications, the outer boundary often is a truncated one. Various approaches have been developed to approximate the boundary condition. Here we also propose a one for our problem. With an approximate Dirichlet boundary condition, we can solve the system to get the velocity. Then we compute the flux along the traction boundary $\int_{\partial\Omega} \mathbf{u} \cdot \mathrm{n}\, ds$. It is reasonable to assume that problem is symmetric and fluid can freely get in and out only along the normal direction. Thus we can distribute the flux along four sides of the outer boundary. Assume that the rectangular domain is $[a,\ b] \times [c\ d]$. We take the side $x = a$ as example. We would set $v = 0$, $u = C(d - y)(y - c)$. The constant is take as $C = \frac{1}{4}\int_{\partial\Omega} \mathbf{u} \cdot \mathrm{n}\, ds / \int_c^d (d - y)(y - c)dy$. In this way, the velocity is continuous up to all the boundaries.

Due to the symmetry, if we start with a circular gas bubble, the bubble should remain circular. The pressure inside the gas bubble is $p_{gas}(t) = \lambda/V^{1.4}(t)$, where $V(t)$ is the volume of the gas bubble at time t. The driving force is due to the difference in the pressure

$$\delta p = p_{gas} - T_{surf} - p_0 = \frac{\pi}{(\pi R^2(t))^{1.4}} - \frac{1}{R(t)} - p_0. \tag{31}$$

If $p_0 = 0$, the equilibrium R_e is the solution of $\pi/(\pi R^2)^{1.4} = 1/R$, which is $R_e = \pi^{-2/9} = 0.7753\cdots$. If we start with $R(0) > R_e$, then the gas bubble will shrink otherwise it will expand. In Fig. 2, we plot the the radius $R(t)$ versus time when $\mu = 2$ or $Re = 1$ with the mesh size $m = n = 64$

for the domain $[-2,\ 2] \times [-2,\ 2]$. In this case, the boundary of the gas bubble converges to its equilibrium monotonically. The error of the equilibrium radius is of $O(10^{-4})$ indicating second order convergence.

(a) (b)

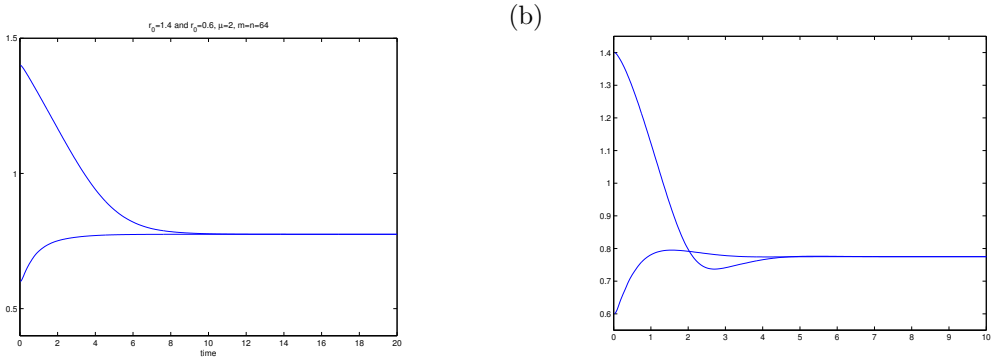

Figure 2: The radius versus time. The initial circle is $R = 1.4$ and $R = 0.6$ respectively. (a) $\mu = 2$, the radius converges to its equilibrium monotonically. (b), $\mu = 0.5$, the radius first converges to its equilibrium, then overshoots, then stabilizes around the equilibrium.

(a) (b)

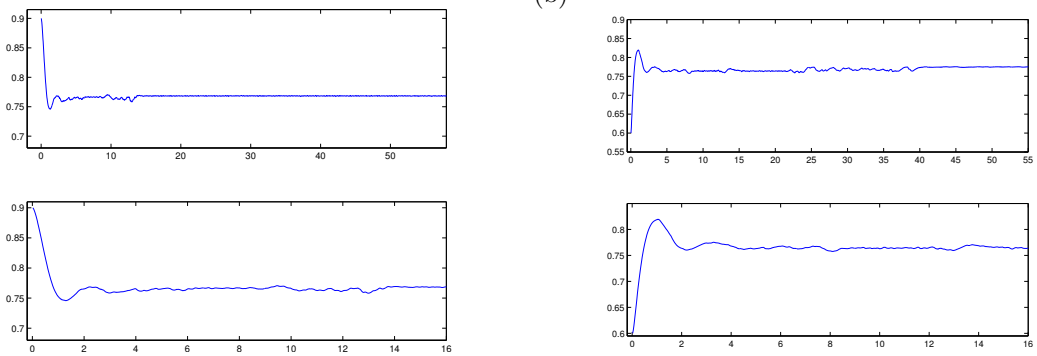

Figure 3: The radius versus time when $\mu = 0.2$. The inertial of the gas bubble causes some oscillations around the equilibrium but eventually stabalized. The The left plot, $R(0) = 0.9$; The right plot, $R(0) = 0.6$. In each case, the bottom plot is a zoom-in plot.

If we increase the Reynolds number by reducing μ to $\mu = 0.2$, then the effect of the inertial term in the Navier-Stokes becomes more apparent, we can see larger overshoot and more oscillatory behavior of the radius around the equilibrium, see Fig. 3. The error in the radius when it approaches equilibrium is about 7.000×10^{-3} for the contracting case, and 1.900×10^{-3} for the expanding case. Similar behaviors are also observed in [19].

If we increase the Reynolds number further by reducing μ to $\mu = 0.15$, we see the bubble oscillates around the equilibrium, see Fig. 4. Note that in the figure, the axis in time and the radius have

very different scales. Over a short period of time, the change in the radius is rather smooth as in the bottom plots of Fig. 2 if the same scale is used.

(a) (b)

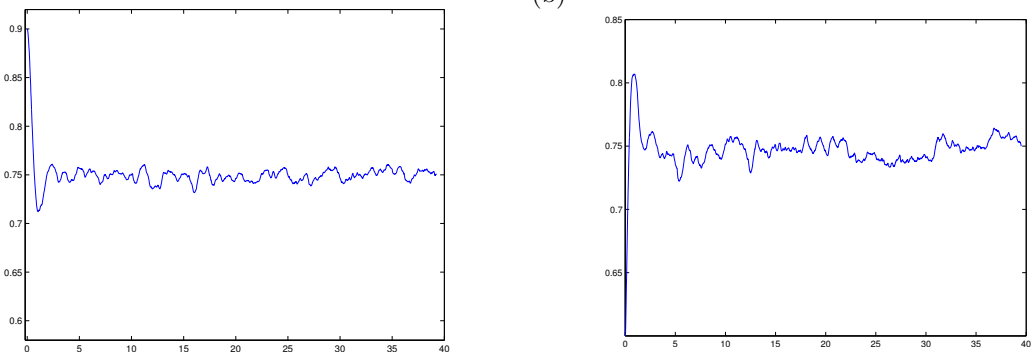

Figure 4: The radius versus time when $\mu = 0.15$. The inertial of the gas bubble causes oscillations around the equilibrium. The left plot, $R(0) = 0.9$; The right plot, $R(0) = 0.6$.

Finally we start with an initial bubble with more complicated geometry

$$r = 1 + 0.3\sin(7\theta), \qquad 0 \le \theta < 2\pi. \tag{32}$$

In Fig. 5, we present a few snap-shots of the bubble interface at several time instances with $\mu = 0.5$. In this case, the surface tension force is more dominant and the bubble relaxes to the circular shape quickly before it converges to its equilibrium circular shape. It over-shoots a couple of times around the equilibrium before it stabilizes.

(a) (b)

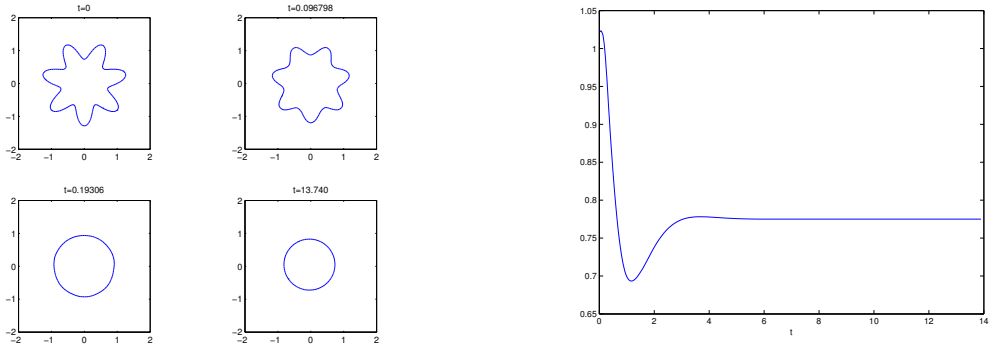

Figure 5: (a), Snap-shots of the bubble at $t = 0$, $t = 0.096798$, $t = 0.19306$, and $t = 13.740$, respectively. (b), The averaged radius versus time when $\mu = 0.5$.

4 Conclusions and acknowledgments

In this paper, we have developed an efficient and stable second order method for Navier-Stokes equations on irregular domain with open and traction boundary conditions. The method is based on the AIIM for the velocity prediction, and a separate Poisson solver for the pressure on the irregular domain with an approximate Dirichlet boundary condition. Numerical examples have shown the robustness and stability of the proposed method.

The author was partially supported by the US ARO grants 550694-MA, the AFSOR grant FA9550-09-1-0520, the US NSF grant DMS-0911434, and the NIH grant 096195-01. Zhao is partially supported by NSF grant DMS-1115698.

References

[1] J. Adams, P. Swarztrauber, and R. Sweet. Fishpack: Efficient Fortran subprograms for the solution of separable elliptic partial differential equations. http://www.netlib.org/fishpack/.

[2] 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.

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

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

[5] J. L. Guermond, P. Minev, and J. Shen. Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions. *SIAM J. Numer. Anal.*, 43:239–258, 2005.

[6] J. Hunter, Z. Li, and H. Zhao. Autophobic spreading of drops. *J. Comput. Phys.*, 183:335–366, 2002.

[7] K. Ito, Z. Li, and M-C. Lai. An augmented method for the Navier-Stokes equations on irregular domains. *J. Comput. Phys.*, 228:2616–2628, 2009.

[8] H. Johnston and J. Liu. Finite difference schemes for incompressible flow based on local pressure boundary conditions. *J. Comput. Phys.*, 180:120–154, 2002.

[9] H. Johnston and J. Liu. Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term. *J. Comput. Phys.*, 188:221–259, 2004.

[10] Z. Li, , M-C. Lai, G. He, and H. Zhao. An augmented method for free boundary problems with moving contact lines. *Computers and Fluids*, 39:1033–1040, 2010.

[11] Z. Li. IIMPACK: A collection of fortran codes for interface problems. Anonymous ftp at ftp.ncsu.edu under the directory: /pub/math/zhilin/Package and http://www4.ncsu.edu/~zhilin/IIM, last updated: 2008.

[12] Z. Li and K. Ito. *The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*. SIAM Frontier Series in Applied mathematics, FR33, 2006.

[13] Z. Li, K. Ito, and M-C. Lai. An augmented approach for Stokes equations with a discontinuous viscosity and singular forces. *Computers and Fluids*, 36:622–635, 2007.

[14] Z. Li and M-C. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *J. Comput. Phys.*, 171:822–842, 2001.

[15] Z. Li, X. Wan, K. Ito, and S. Lubkin An augmented pressure boundary condition for a Stokes flow with a non-slip boundary condition. *Communications in Computational Physics*, 1:874–885, 2006.

[16] 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.

[17] J. Liu. Open and traction boundary conditions for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 228:7250–7267, 2009.

[18] A. Poux, S. Glockner, and M. Azaiz. Improvements on open and traction boundary conditions for Navier-Stokes time-splitting methods. *J. Comput. Phys.*, 230:4011–4027, 2011.

[19] M. Sussman and E.G. Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, 162:301-337, 2000.