Published in final edited form as:

*J Comput Phys*. 2015 August 15; 297: 182–193. doi:10.1016/j.jcp.2015.05.003.

# A semi-implicit augmented IIM for Navier–Stokes equations with open, traction, or free boundary conditions

**Zhilin Li**[a,*], **Li Xiao**[b], **Qin Cai**[b], **Hongkai Zhao**[c], and **Ray Luo**[d]

[a]Department of Mathematics and CRSC, North Carolina State University, Raleigh, NC 27695, USA

[b]Departments of Biomedical Engineering and Molecular Biology and Biochemistry, University of California, Irvine, CA 92697, USA

[c]Department of Mathematics, University of California, Irvine, CA 92697, USA

[d]Departments of Molecular Biology and Biochemistry, Chemical Engineering and Materials Science, and Biomedical Engineering, University of California, Irvine, CA 92697, USA

## 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, traction, and free 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 some problems of this type, 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. A fast Poisson solver can be used in our algorithm which gives us the potential for developing fast overall solvers in the future. 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 incompressible fluid and a compressible gas bubble is also presented.

### Keywords

Navier–Stokes equations; Finite difference approximation; Irregular domain; Open and traction boundary conditions; Local pressure boundary condition; Augmented immersed interface method (IIM)

## 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 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

[*]Corresponding author. zhilin@math.ncsu.edu (Z. Li).

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) [48–50],

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

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

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 Fig. 1 for an illustration. $\rho$ and $\mu$ 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 $\partial R$.

Along the interior boundary $\partial\Omega$, if the boundary condition is given by

$$\mu\partial_n\mathbf{u}-p\mathbf{n}=\mathbf{g}_2,\quad \text{on }\partial\Omega,\quad (3)$$

it is then called an open (or pseudo-traction) boundary condition. If the boundary condition is given by

$$\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\cdot\mathbf{n}-p\,\mathbf{n}=\mathbf{g}_2,\quad \text{on }\partial\Omega,\quad (4)$$

it is then called a traction boundary condition, see [35,38]. The traction boundary condition which reflects the force balance is more physical than the open boundary condition. In the case that $\Omega$ represents a gas bubble, we use the traction boundary condition in the normal and tangential directions, respectively,

$$\left\{\begin{array}{c}\mathbf{n}^T\cdot\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\cdot\mathbf{n}=p-p_{air}-\gamma\kappa,\\ \boldsymbol{\tau}^T\cdot\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^T)\cdot\mathbf{n}=0,\end{array}\right.\quad \text{on }\partial\Omega.\quad (5)$$

The boundary condition above is also called the free boundary condition. Note that, the boundary condition in the normal direction differs by a factor of 2 for the open and traction boundary conditions; while the boundary condition in the tangential direction is different. We refer the reader to [18,35] and the references therein for the well-posedness of the problem. The velocity is not necessarily zero along boundaries $\partial\Omega$ and $\partial R$, 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 of 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

$$\frac{d\mathbf{X}}{dt}=\mathbf{u}(\mathbf{X},t), \quad \mathbf{X}(t) \in \partial\Omega(t). \quad (6)$$

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 problems, flow problems on irregular domains, not many deal with open or traction boundary conditions, even fewer deal with arbitrary boundaries using finite difference methods. In [27], the authors proposed a technique on a moving orthogonal curvilinear coordinate system, which is constructed numerically and adjusted to fit the boundary shape at any time for an axisymmetric unsteady bubble deformation problem. The NSE solver is based on the backward Euler's method for streamfunction and the vorticity. In [9], the authors studied numerical simulation of free surface incompressible liquid flows surrounded by compressible gas, which is opposite to our situation (a bubble surrounded by a fluid). The volume of fluid (VOF) is used to track the moving interface between the bubble and the gas. The discretization is focused on the gas part. In [43,45], the authors proposed coupled level set methods and the volume of fluid (VOF) method for free surface problems. The implementation of VOF is non-trivial since one needs to consider quite a few geometric configurations and high order reconstructions of the boundary. Our method proposed in this paper provides an alternative method that a fast Poisson solver can be utilized. Other related research on deforming boundary problems in multiple dimensions and higher order treatment of jump conditions can be found in [7,17,19–21,40].

For small to medium Reynolds numbers, the projection method, for example [5,8,14], 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 or traction boundary conditions, see for example [18].

Another type of methods are called local pressure boundary conditions [25,26]. 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 [35]. 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 [25,26], the Neumann boundary condition is derived while in this paper, a Dirichlet boundary condition is used. Note that the original local pressure boundary method [25,26] 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 [24], 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. Note that, the AIIM framework is based on efficient solvers for elliptic interface problems. Thus some recent work on efficient solvers for elliptic interface problems [3,4,12] can be substituted for the IIM solver in this paper.

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 discretization 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.

For fluid structure interaction with a gas bubble, our model uses different governing equations for fluid and gas separately. Note that, another model is to treat the problem using a two-phase model in which the governing equations are the same, see for example [2,6,10,11,13,15,16,36,37,39,41,42,44,46,47]. There are advantages and limitations for each model. Our method distinguishes from others in several aspects. Our method is a Cartesian based finite difference method. The discretization of the jump conditions is second order accurate. A fast Poisson solver can be used in our algorithm which gives us the potential for developing fast overall solvers in the future.

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 $\Delta t$. 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.

$$\frac{3\mathbf{u}^{k+1}-4\mathbf{u}^{k}+\mathbf{u}^{k-1}}{2\Delta t}+(\mathbf{u}\cdot\nabla\mathbf{u})^{k+1}+\nabla\tilde{p}^{k+1}=\mu\Delta\mathbf{u}^{k+1}+\mathbf{G}^{k+1}, \quad (7)$$

$$\left(-\tilde{p}^{k+1}+\mathbf{n}^{T}\cdot\mu\left(\nabla\mathbf{u}^{\mathbf{k+1}}+\nabla\left(\mathbf{u}^{k+1}\right)^{T}\right)\cdot\mathbf{n}+\gamma\kappa+p_{air}\right)\Big|_{\partial\Omega}=0, \quad (8)$$

$$\boldsymbol{\tau}^{T}\cdot\mu\left(\nabla\mathbf{u}^{k+1}+\nabla\left(\mathbf{u}^{k+1}\right)^{T}\right)\cdot\mathbf{n}\Big|_{\partial\Omega}=0, \quad (9)$$

$$\left[\mathbf{u}^{k+1}\right]_{\partial\Omega}=0, \qquad \left[\frac{\partial\mathbf{u}^{k+1}}{\partial n}\right]_{\partial\Omega}=\mathbf{q}^{k+1}, \quad (10)$$

where

$$\tilde{p}^{k+1}=2p^{k}-p^{k-1}, \quad (11)$$

$$(\mathbf{u}\cdot\nabla\mathbf{u})^{k+1}=2\,(\mathbf{u}\cdot\nabla\mathbf{u})^{k}-(\mathbf{u}\cdot\nabla\mathbf{u})^{k-1} \quad (12)$$

$$\mathbf{G}^{k+1}=0, \quad \text{if}\,(x,y)\in\Omega, \quad (13)$$

$\kappa$ is the curvature of the free boundary $\partial\Omega$, and $\gamma$ 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 at least for fixed boundaries. The convergence of the stability of the scheme is given in [18]. Certainly it is viable to use a second or higher order Runge–Kutta discretization in time, nevertheless, the proof of the stability of the scheme is still open problem for open or traction boundary conditions. If the boundary moves across a grid line, that is, $(x_i, y_j)$ is in inside and outside of the domain at different time levels, a simple correction term can be added as explained in [23]. 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

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

The pressure $p^{\widetilde{k+1}}$ is extrapolated from the pressure at previous two steps and so is the non-linear 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 traction condition (5) with an approximated $p^{k+1}$. We refer the readers to [24,28,30,31] 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 or traction boundary conditions.

Since we are only interested in the quantities outside of Ω (the gas bubbles), $p^{\widetilde{k+1}}$ 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 $p^{\widetilde{k+1}}$ 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 [30]. We will see that the coefficient matrix for the augmented variable $\mathbf{q}^{k+1}$ is well-conditioned with almost $O(1)$ condition number.

**<u>Step 2:</u>** Solve the pressure from the equation obtained after applying the gradient operator to the momentum equation,

$$\Delta p^{k+1} = \begin{cases} -\nabla \cdot (\mathbf{u} \cdot \nabla \mathbf{u})^{k+1} + \nabla \cdot \mathbf{G}^{k+1} & \text{if } (x, y) \in R \backslash \Omega, \\ 0 & \text{if } (x, y) \in \Omega, \end{cases} \quad (14)$$

$$p^{k+1}|_{\partial\Omega} = \left( \mathbf{n}^T \cdot \mu \left( \nabla \mathbf{u}^{\mathbf{k+1}} + \nabla \left( \mathbf{u}^{k+1} \right)^T \right) \cdot \mathbf{n} + \gamma\kappa + p_{air} \right) \Big|_{\partial\Omega}, \quad (15)$$

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 [29], see also [22,30,34] 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 [30] 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 $(x_i^*, y_j), x_i^*{=}x_i{+}\alpha h_x$, $-1 \le a \le 1$, then the second order partial derivative in $x$ can be approximated by

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j){=}\frac{u_{i-1,j}{-}2u_{i-1,j}{+}u_{i+1,j}}{h_x^2}{-}\frac{C(x_i, \alpha)}{h_x^2}{+}O\left(h\right), \quad (16)$$

where

$$\frac{C(x_i, \alpha)}{h_x^2}{=}[u]{+}[u_x]\left(1{-}|\alpha|\right)h{+}[u_{xx}]\frac{(1{-}|\alpha|)^2 h_x^2}{2}, \quad (17)$$

see Lemma 1 in [32]. 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 $\theta$ is the angle between the outward normal direction and the $x$-axis, see Fig. 1. We define the local coordinates at $(X, Y)$ as

$$\begin{gathered}\xi{=}(x{-}X)\cos\theta{+}(y{-}Y)\sin\theta,\\ \eta{=}{-}(x{-}X)\sin\theta{+}(y{-}Y)\cos\theta.\end{gathered} \quad (18)$$

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.

$$\begin{gathered}[\mathbf{u}]{=}\mathbf{0}, \quad [\mathbf{u}_\xi]{=}\mathbf{q}, \quad [\mathbf{u}_\eta]{=}\mathbf{0},\\ [\mathbf{u}_{\eta\eta}]{=}{-}\kappa\mathbf{q}, \quad [\mathbf{u}_{\xi\eta}]{=}{-}\frac{\partial \mathbf{q}}{\partial \eta},\\ [\mathbf{u}_{\xi\xi}]{=}{-}[\mathbf{u}_{\eta\eta}]{+}[p_\xi]\,\mathbf{n}{+}[p_\eta]\,\boldsymbol{\tau}{+}[\mathbf{u}_\xi]\mathbf{u}\cdot\mathbf{n}{-}[\mathbf{G}].\end{gathered} \quad (19)$$

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:

$$\begin{gathered}[u_x]{=}[u_\xi]\cos\theta{-}[u_\eta]\sin\theta,\\ [u_y]{=}[u_\xi]\sin\theta{+}[u_\eta]\cos\theta,\\ [u_{xx}]{=}[u_{\xi\xi}]\cos^2\theta{-}2[u_{\xi\eta}]\cos\theta\sin\theta{+}[u_{\eta\eta}]\sin^2\theta,\\ [u_{yy}]{=}[u_{\xi\xi}]\sin^2\theta{+}2[u_{\xi\eta}]\cos\theta\sin\theta{+}[u_{\eta\eta}]\cos^2\theta.\end{gathered} \quad (20)$$

With these jump condition, we can approximate $u_x$, $u_y$, $u_{xx}$, $u_{xy}$ according to (16)–(17) 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 $\partial R$ for the Poisson equation for the pressure. Often it is an approximate normal derivative condition. However, with a prescribed velocity, the incompressibility condition, and the NSE equations, we can use the techniques described in [26,33] 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

$$(\mathbf{u}\cdot\nabla)\mathbf{u}\cdot\mathbf{n}=uu_x+vu_y=-uv_y+vu_y, \quad (21)$$

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

$$u_{xx}(a,y_j)\approx 2\frac{u_{1,j}-u_{0,j}-v_y(a,y_j)h}{h^2}. \quad (22)$$

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

$$p_x(a,y_j)=\mu u_{xx}(a,y_j)+G_1-u_t(a,y_j)-(\mathbf{u}\cdot\nabla)\mathbf{u}(a,y_j), \quad (23)$$

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

$$u(x,y,t)=w(t)\left(\frac{y}{\sqrt{x^2+y^2}}-2y\right), \quad (24)$$

$$v(x,y,t)=w(t)\left(\frac{x}{\sqrt{x^2+y^2}}+2x\right), \quad (25)$$

$$p(x,y,t)=\left((x^2+y^2)^2-\frac{1}{4}\right)^2. \quad (26)$$

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 **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

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

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,

$$order_u = \frac{\log\left(\|E_u\|_{\infty,2N} / \|E_u\|_{\infty,N}\right)}{\log 2}; \quad order_p = \frac{\log\left(\|E_p\|_{\infty,2N} / \|E_p\|_{\infty,N}\right)}{\log 2}. \quad (27)$$

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

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.

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 open boundary condition example

The proposed method in this paper is flexible in dealing with different boundary conditions. All we need to do is to modify the augmented equations accordingly. In Table 3, we show a grid refinement analysis against the exact the same solution for the ‘open boundary condition’ at a final time $T = 1.5$ with $w(t) = \sin t$. The right hand side $\mathbf{g}_2$ is determined from the true solution. Second order accuracy is confirmed for the problem with fixed boundary.

### 3.2. 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 $\lambda$ is a constant. Again we first test our code for a steady state case with $\gamma = 1$ in which we have an analytic solution,

$$u(x,y,t)=\frac{x}{x^2+y^2}, \quad (28)$$

$$v(x,y,t)=\frac{y}{x^2+y^2}, \quad (29)$$

$$p(x,y,t)=-\frac{1}{2(x^2+y^2)}, \quad (30)$$

$$\mathbf{G}=\mathbf{0}, \quad (31)$$

outside the unit circle $r = 1$ but within the domain [−2, 2] × [−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 4, we show the grid refinement analysis at $T = 1.5$. We can see second order convergence for all the variables.

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 $\phi(\mathbf{x}, t)$ 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

$$\varphi_t+\nabla\varphi\cdot\mathbf{u}=0 \quad (32)$$

along with a re-initialization process.

**3.2.1. 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 taken 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

$$\delta p = p_{gas} - T_{surf} - p_0 = \frac{\pi}{(\pi R^2(t))^{1.4}} - \frac{1}{R(t)} - p_0. \quad (33)$$

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 radius $R(t)$ versus time when $\mu = 2$ or $Re = 1$ with the mesh size $m = n = 64$ for the domain [−2, 2] × [−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.

If we increase the Reynolds number by reducing $\mu$ 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 $7.000 \times 10^{-3}$ for the contracting case, and $1.900 \times 10^{-3}$ for the expanding case. Similar behaviors are also observed in [43].

If we increase the Reynolds number further by reducing $\mu$ 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. Note that, our simulation results agree with Fig. 10 in [27] qualitatively.

Now we start with an initial bubble with more complicated geometry

$$r = 1 + 0.3\sin(7\theta), \qquad 0 \le \theta < 2\pi. \quad (34)$$

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.

As a final example, we tested our code for a time dependent problem in which there is a source in the air bubble according to

$$p_{gas} = \frac{\lambda}{(V + 0.5\sin t)^{1.4}}. \quad (35)$$

The initial bubble shape is

$$r = 1.4 + 0.05\sin(5\theta), \qquad 0 \le \theta < 2\pi. \quad (36)$$

In Fig. 6, we show the average of the radius $\bar{R}(t)$ versus the time. Again, the surface tension will bring the bubble to a circular shape. The bubble shrinks quickly and overshoots. Then the bubble oscillates around the mean equilibrium $R = 0.7753 \cdots$ as expected.

### 3.3. A 3D simulation

Our method has been implemented in three dimensions. The main algorithms is similar but the implementation requires some substantial effort, and the computation cost is much more demanding. We show one example of an air bubble rising in a fluid due to the gravity. The air bubble is set as 1 cm initially. The parameters are chosen as follows: the surface tension $\sigma$ = 38 N/m, the viscosity $\mu$ = 0.90 Pa s, and the gravity $g$ = 9.8 m/s$^2$. The box size is 4.8 cm × 4.8 cm × 4.8 cm. Some snapshots at different time are shown in Fig. 7. The bubble moves 2.2 cm in 0.15 s. Below are a sketch of the theoretical derivation for the simplified situation. Even so, the analysis roughly agrees with the experimental result.

The buoyancy force on the bubble in the relatively high viscous flow is $F_b = 4\pi\rho g\, R^3/3$, where $R$ is the radius of the initial bubble. The viscosity force is $F_\mu = 6\pi\mu R\rho v$. Assume that the mass of the bubble is negligible, then it is accelerated immediately. The acceleration of the bubble is equivalent to a same bubble with water density but in the opposite direction. The motion of the bubble follows Newton's second law with the force $F_b - F_\mu = 4\pi\rho g a R^3/3$, where $a$ is the acceleration rate. Since we know $F_b$ and $F_\mu$, we get the acceleration rate $a = g - 9\mu v/(2\rho R^2)$. Thus we can write down the motion equation of the bubble as

$$\ddot{x} = g - 9\mu v/(2\rho R^2)\,\dot{x} = g - \eta\,\dot{x}, \quad (37)$$

where $\eta = 9\mu v/(2\rho R^2) = 40.5$ s$^{-1}$ in our simulation. This damped Newtonian equation has the analytical solution,

$$x(t) = C_1 + C_2 e^{-\eta t} + \frac{g}{\eta} t \quad (38)$$

where $C_1$ and $C_2$ are two constants. Given $x(0) = 0$, $\dot{x}(0) = 0$, we then have

$$x(t) = -\frac{g}{\eta^2}\left(1 - e^{-\eta t}\right) + \frac{g}{\eta} t. \quad (39)$$

The expression above gives $x$(0.15 s) = 3 cm. It is roughly agree with the experiment (2.2 cm). The difference is likely due to the ignored boundary condition and treating the air bubble as a sphere with a fixed volume, etc., in the analysis which is not exactly true in the simulation.

## 4. Conclusions

In this paper, we have developed an efficient and stable method for Navier–Stokes equations on irregular domain with open, traction, or free 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.

## Acknowledgments

We would like to thank the referee for helpful comments and suggestions. The first author was partially supported by the US AFSOR grant FA9550-09-1-0520, the US NSF grant DMS-0911434, and the NIH grant 5R01GM96195-2, the CNSF grants 10971102, 11371199, 11471166, and BK20141443. Z. Li is also supported by a Lecture Professorship fund by Nanjing Normal University. H. Zhao is partially supported by NSF grant DMS-1115698. L. Xiao, Q. Cai, and R. Luo are partially supported by NIH grants GM093040 and GM079383.

## References

1. Adams, J.; Swarztrauber, P.; Sweet, R. Fishpack: efficient Fortran subprograms for the solution of separable elliptic partial differential equations. http://www.netlib.org/fishpack/
2. Agarwal A, Tai CF, Chung JN. Unsteady development of a deformable bubble rising in a quiescent liquid. Comput Methods Appl Mech Eng. 2010; 199(17–20):1080–1090.
3. Almgren A, Bell J, Collella P, Marthaler T. A Cartesian grid projection method for the incompressible Euler equations in complex geometries. SIAM J Sci Comput. 1997; 18:1289–1309.
4. Bedrossian J, Von Brecht J, Zhu S, Sifakis E, Teran J. A second order virtual node method for Poisson interface problems on irregular domains. J Comput Phys. 2010; 221:6405–6426.
5. Bell JB, Colella P, Glaz HM. A second-order projection method for the incompressible Navier–Stokes equations. J Comput Phys. 1989; 85:257–283.
6. Billaud M, Gallice G, Nkonga B. A simple stabilized finite element method for solving two phase compressible–incompressible interface flows. Comput Methods Appl Mech Eng. 2011; 200:1272–1290.
7. Bouffanais R, Deville MO. Mesh update techniques for free-surface flow solvers using spectral element method. J Sci Comput. 2006; 27(1–3):137–149.
8. Brown DL, Cortez R, Minion ML. Accurate projection methods for the incompressible Navier–Stokes equations. J Comput Phys. 2001; 168:464.
9. Caboussat A, Picasso M, Rappaz J. Numerical simulation of free surface incompressible liquid flows surrounded by compressible gas. J Comput Phys. 2005; 203:626–649.
10. Caiden R, Fedkiw R, Anderson C. A numerical method for two phase flow consisting of separate compressible and incompressible regions. J Comput Phys. 2001; 166:1–27.
11. Can E, Prosperetti A. A level set method for vapor bubble dynamics. J Comput Phys. 2012; 231(4): 1533–1552.
12. Coco A, Russo G. Second order multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, I: one dimensional problems. Numer Math, Theory Methods Appl. 2012; 5:19–42.
13. Compere G, Marchandise E, Remacle JF. Transient adaptivity applied to two-phase incompressible flows. J Comput Phys. 2008; 227(3):1923–1942.
14. Liu WEJ. Projection method: I: convergence and numerical boundary-layers. SIAM J Numer Anal. 1995; 32:1017–1057.
15. Francois M, Marianne SJ, Dendy ED, Kothe DB, Sicilian JM, Williams NW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. J Comput Phys. 2006; 213(1):141–173.
16. Gerlach D, Tomar G, Biswas G, Durst F. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. Int J Heat Mass Transf. 2006; 49(3):740–754.
17. Gibou F, Fedkiw R. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains with applications to the Stefan problem. J Comput Phys. 2005; 202(2):577–601.
18. Guermond JL, Minev P, Shen J. Error analysis of pressure-correction schemes for the time-dependent Stokes equations with open boundary conditions. SIAM J Numer Anal. 2005; 43:239–258.
19. Helenbrook BT. A two-fluid spectral-element method. Comput Methods Appl Mech Eng. 2001; 191(3):273–294.
20. Ho LW, Patera AT. A Legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows. Comput Methods Appl Mech Eng. 1990; 80(1):355–366.

21. Ho LW, Patera AT. Variational formulation of three-dimensional viscous free-surface flows: natural imposition of surface tension boundary conditions. Int J Numer Methods Fluids. 1991; 13(6):691–698.

22. Hunter J, Li Z, Zhao H. Autophobic spreading of drops. J Comput Phys. 2002; 183:335–366.

23. Li Z. Immersed interface method for moving interface problems. Numer Algorithms. 1997; 14:269–293.

24. Ito K, Li Z, Lai MC. An augmented method for the Navier–Stokes equations on irregular domains. J Comput Phys. 2009; 228:2616–2628.

25. Johnston H, Liu J. Finite difference schemes for incompressible flow based on local pressure boundary conditions. J Comput Phys. 2002; 180:120–154.

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

27. Kang IS, Leal LG. Numerical solution of axisymmetric, unsteady free-boundary problems at finite Reynolds number. Phys Fluids. 1987; 30:1929–1940.

28. Li Z, Lai MC, He G, Zhao H. An augmented method for free boundary problems with moving contact lines. Comput Fluids. 2010; 39:1033–1040.

29. Li, Z. 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

30. Li Z, Ito K. The Immersed Interface Method – Numerical Solutions of PDEs Involving Interfaces and Irregular Domains. SIAM Frontier Series in Applied Mathematics. 2006:FR33.

31. Li Z, Ito K, Lai MC. An augmented approach for Stokes equations with a discontinuous viscosity and singular forces. Comput Fluids. 2007; 36:622–635.

32. Li Z, Lai MC. The immersed interface method for the Navier–Stokes equations with singular forces. J Comput Phys. 2001; 171:822–842.

33. Li Z, Wan X, Ito K, Lubkin S. An augmented pressure boundary condition for a Stokes flow with a non-slip boundary condition. Commun Comput Phys. 2006; 1:874–885.

34. Li Z, Zhao H, Gao H. A numerical study of electro-migration voiding by evolving level set functions on a fixed Cartesian grid. J Comput Phys. 1999; 152:281–304.

35. Liu J. Open and traction boundary conditions for the incompressible Navier–Stokes equations. J Comput Phys. 2009; 228:7250–7267.

36. Liovic P, Francois M, Rudman M, Manasseh R. Efficient simulation of surface tension-dominated flows through enhanced interface geometry interrogation. J Comput Phys. 2010; 229(19):7520–7544.

37. van der Pijl SP, Segal A, Vuik C, Wesseling P. A mass conserving level-set method for modelling of multi-phase flows. Int J Numer Methods Fluids. 2005; 47(4):339–361.

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

39. Raessi M, Mostaghimi J, Bussmann M. A volume-of-fluid interfacial flow solver with advected normals. Comput Fluids. 2010; 39(8):1401–1410.

40. Ronquist EM, Patera AT. A Legendre spectral element method for the Stefan problem. Int J Numer Methods Eng. 1987; 24(12):2273–2299.

41. Samiei E, Shams M, Ebrahimi R. A novel numerical scheme for the investigation of surface tension effects on growth and collapse stages of cavitation bubbles. Eur J Mech B, Fluids. 2011; 30(1):41–50.

42. Sussman M. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. J Comput Phys. 2003; 187(1):110–136.

43. Sussman M, Puckett EG. A coupled level set and volume of fluid method for computing 3D and axisymmetric incompressible two-phase flows. J Comput Phys. 2000; 162:301–337.

44. Sussman M. A parallelized, adaptive algorithm for multiphase flows in general geometries. Comput Struct. 2005; 83:435–444.

45. Sussman M, Smith KM, Hussaini MY, Ohta M, Zhi-Wei R. A sharp interface method for incompressible two-phase flows. J Comput Phys. 2007; 221:469–505.

46. Tai CF, Chung JN. A direct numerical simulation of axisymmetric liquid–gas two-phase laminar flows in a pipe. Int J Numer Methods Fluids. 2011; 66(1):97–122.

47. Zhang ZY, Zhang HS. Surface tension effects on the behavior of a cavity growing, collapsing, and rebounding near a rigid wall. Phys Rev E. 2004; 70(5):056310.

48. Xiao L, Cai Q, Li Z, Zhao H, Luo R. A multi-scale method for dynamics simulation in continuum solvent models I: finite-difference algorithm for Navier–Stokes equation. Chem Phys Lett. 2014; 616–617:67–74.

49. Xiao L, Wang C, Luo R. Recent progress in adapting Poisson–Boltzmann methods to molecular simulations. J Theor Comput Chem. 2014; 13:1430001.

50. Botello-Smith WM, Cai Q, Luo R. Biological applications of classical electrostatics methods. J Theor Comput Chem. 2014; 13:1440008.

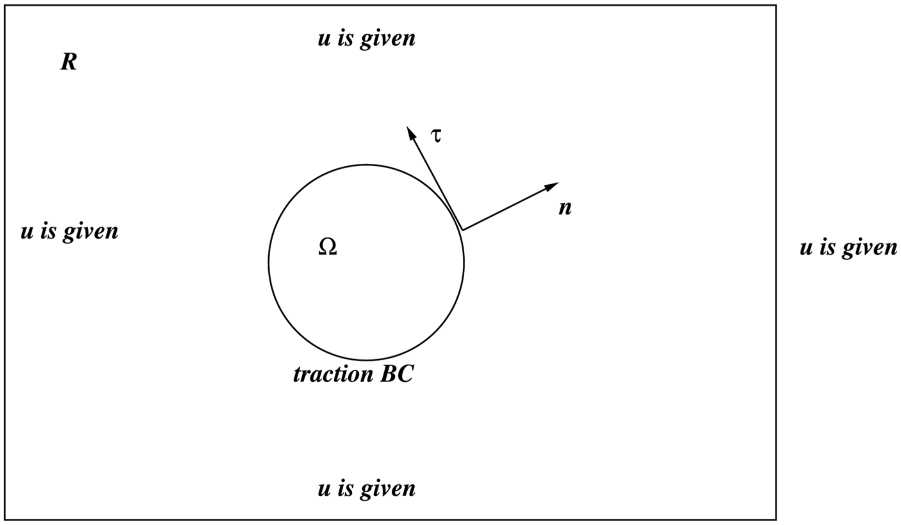

**Fig. 1.**

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

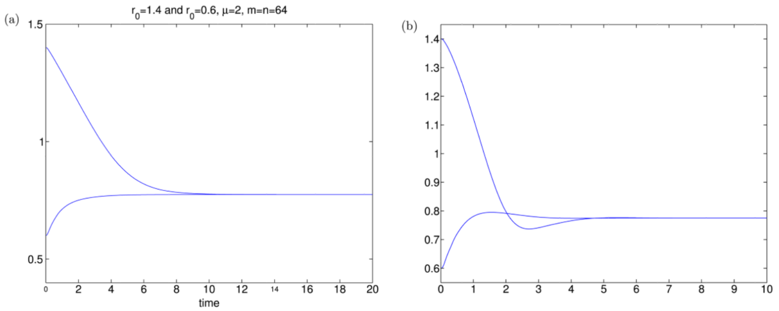

**Fig. 2.**

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.

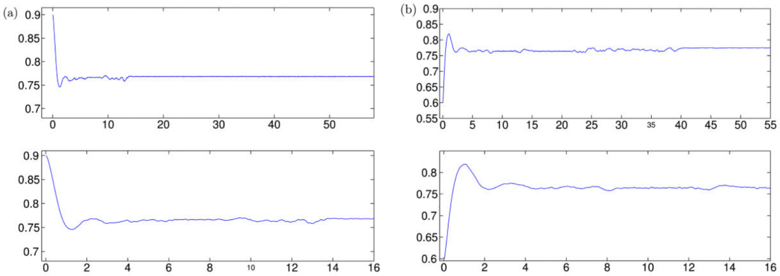

**Fig. 3.**

The radius versus time when $\mu = 0.2$. The inertial of the gas bubble causes some oscillations around the equilibrium but eventually stabilized. 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.

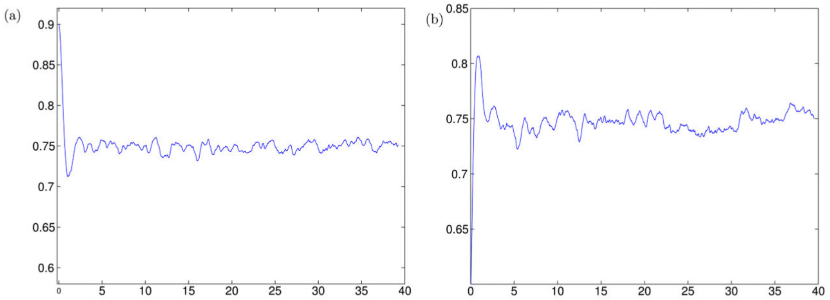

**Fig. 4.**

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, $R(0) = 0.6$.

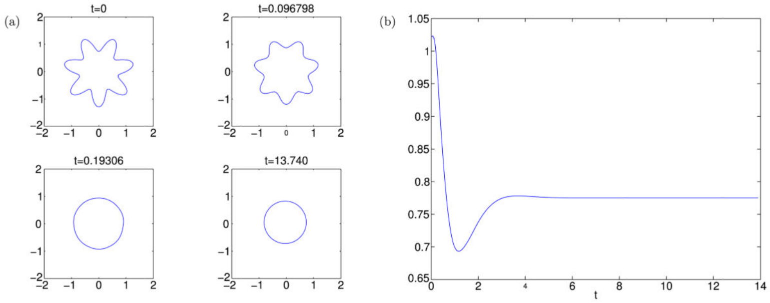

**Fig. 5.**

(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 $\mu = 0.5$.

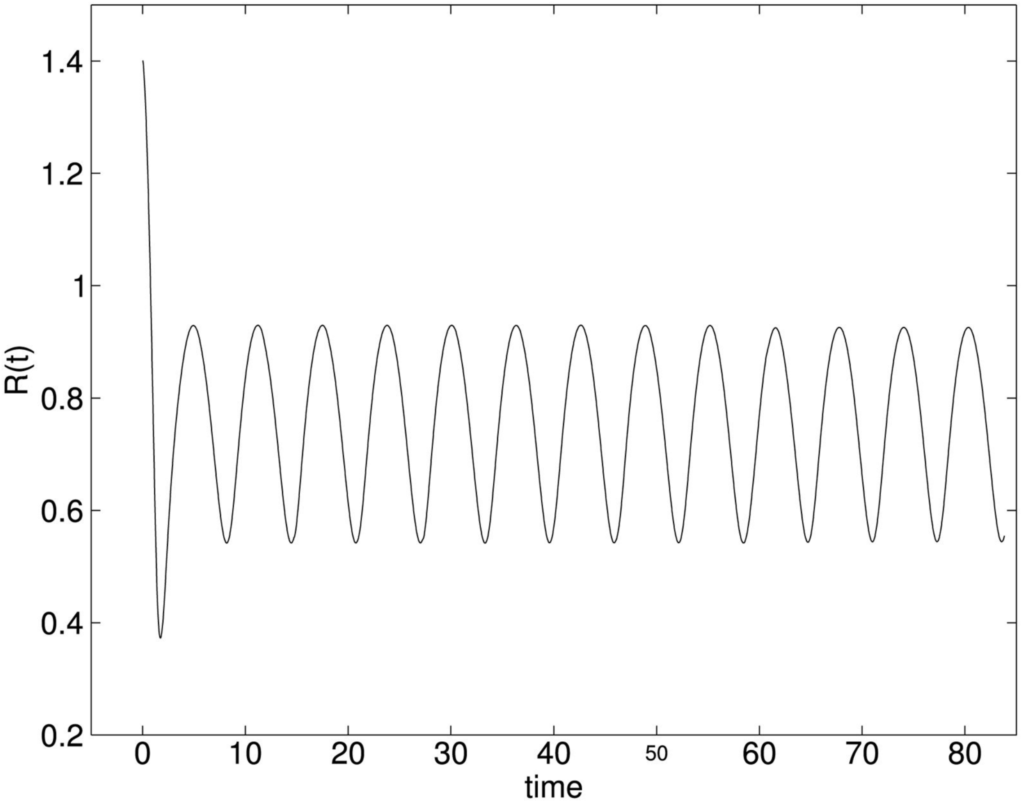

**Fig. 6.**

The average of the radius when the air bubble has a source.

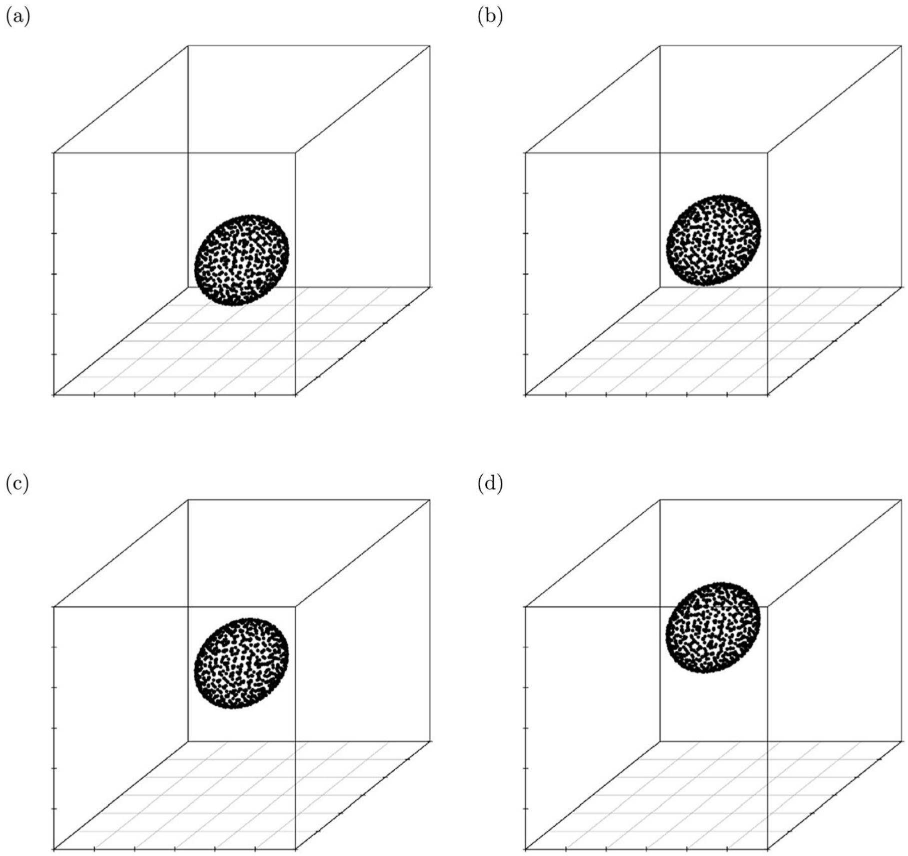

**Fig. 7.**

A 3D air bubble motion in a fluid with gravity $g = 9.8$ m/s$^2$ at different time. (a) $t = 0$; (b) $t = 0.05$ s; (c) $t = 0.1$ s; (d) $t = 0.15$ s.

**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.

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

### Table 2

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

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

## Table 3

A grid refinement analysis against the exact solution for ‘Open Boundary Condition’ at a final time $T = 1.5$ with $w(t) = \sin t$.

| $N$ | $\|\|E_u\|\|_\infty$ | $order_u$ | $\|\|E_p\|\|_\infty$ | $order_p$ |
|---|---|---|---|---|
| 16 | $3.0692 \times 10^{-2}$ | | $4.9878 \times 10^{-3}$ | |
| 32 | $7.2849 \times 10^{-3}$ | 2.0749 | $1.4994 \times 10^{-3}$ | 1.7340 |
| 64 | $1.9398 \times 10^{-3}$ | 1.9090 | $4.1105 \times 10^{-4}$ | 1.8670 |
| 128 | $4.6722 \times 10^{-4}$ | 2.0537 | $9.8679 \times 10^{-5}$ | 2.0585 |
| 256 | $1.1839 \times 10^{-4}$ | 1.9806 | $2.3234 \times 10^{-5}$ | 2.0865 |
| 512 | $2.9746 \times 10^{-5}$ | 1.9927 | $5.4167 \times 10^{-6}$ | 2.1008 |

## Table 4

A grid refinement analysis against the exact solution for the fluid and air bubble interaction example at a final time $T = 1.5$.

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