

UC San Diego

UC San Diego Previously Published Works

Title

Numerical Treatment of Stokes Solvent Flow and Solute-Solvent Interfacial Dynamics for Nonpolar Molecules

Permalink

<https://escholarship.org/uc/item/37g5708v>

Journal

Journal of Scientific Computing, 67(2)

ISSN

0885-7474

Authors

Sun, Hui
Zhou, Shenggao
Moore, David K
[et al.](#)

Publication Date

2016-05-01

DOI

10.1007/s10915-015-0099-z

Peer reviewed



Published in final edited form as:

J Sci Comput. 2016 May ; 67(2): 705–723. doi:10.1007/s10915-015-0099-z.

Numerical Treatment of Stokes Solvent Flow and Solute-Solvent Interfacial Dynamics for Nonpolar Molecules

Hui Sun^{*}, Shenggao Zhou[†], David K. Moore[‡], Li-Tien Cheng[§], and Bo Li[¶]

Hui Sun: hus003@ucsd.edu; Shenggao Zhou: sgzhou@suda.edu.cn; David K. Moore: dkmoore@ucsd.edu; Li-Tien Cheng: lcheng@math.ucsd.edu; Bo Li: bli@math.ucsd.edu

^{*}Department of Mathematics, University of California, San Diego, CA 92093

[†]School of Mathematical Sciences and Mathematical Center for Interdiscipline Research, Soochow University, 1 Shizi Street, Suzhou, Jiangsu 215006, China

[‡]Department of Physics, University of California, San Diego, CA 92093

[§]Department of Mathematics, University of California, San Diego, CA 92093

[¶]Department of Mathematics and Quantitative Biology Graduate Program, University of California, San Diego, CA 92093

Abstract

We design and implement numerical methods for the incompressible Stokes solvent flow and solute-solvent interface motion for nonpolar molecules in aqueous solvent. The balance of viscous force, surface tension, and van der Waals type dispersive force leads to a traction boundary condition on the solute-solvent interface. To allow the change of solute volume, we design special numerical boundary conditions on the boundary of a computational domain through a consistency condition. We use a finite difference ghost fluid scheme to discretize the Stokes equation with such boundary conditions. The method is tested to have a second-order accuracy. We combine this ghost fluid method with the level-set method to simulate the motion of the solute-solvent interface that is governed by the solvent fluid velocity. Numerical examples show that our method can predict accurately the blow up time for a test example of curvature flow and reproduce the polymodal (e.g., dry and wet) states of hydration of some simple model molecular systems.

Keywords

Nonpolar molecules; solute-solvent interface; the Stokes equation; ghost fluid method; level-set method; interface motion; change of volume; traction boundary conditions

1 Introduction

Aqueous solvent plays a significant role in dynamical processes of biological molecules, such as conformational changes, molecular recognition, and molecular assembly, that control cellular functions of underlying biological systems [2, 12, 21, 22]. Implicit-solvent models are efficient descriptions of such dynamical processes. In such descriptions, the solvent is treated implicitly as a continuum and the effect of individual solvent molecules is coarse grained [3, 15, 23]. One of the successful dielectric boundary based implicit-solvent approaches is the variational implicit-solvent model (VISM) [7, 8]. In VISM, one minimizes

a macroscopic solvation free-energy functional of all possible dielectric boundaries, coupling both nonpolar and polar contributions, and the solute-solvent van der Waals (vdW) interaction. Implemented by a robust level-set numerical method [4, 5], VISM can predict polymodal states of hydration, such as wet and dry states, subtle electrostatic effects, and solvation free energies of an underlying bimolecular system [4, 9, 18, 25, 28, 29].

While dielectric boundary based implicit-solvent models, including VISM, have been successful in many cases, they treat the solvent as a structureless dielectric medium, neglecting other solvent effects, such as the solvent hydrodynamic effect. Recent experimental and theoretical studies have indicated that the solvent shear motion can induce protein conformational changes and the solvent viscosity can affect the kinetics of such changes [1, 10, 11, 16, 17, 19–21, 24].

In several recent works [13, 14, 26, 27], the authors have initiated the development of a fluid mechanics approach to treat the solvent fluid in molecular systems. The key features of such a new approach include: (1) the aqueous solvent (i.e., water or salted water) is treated as an incompressible fluid and its motion is by the Stokes or Navier-Stokes equation; (2) the solute pressure is simply described by the ideal-gas law; (3) the electrostatic interactions are modeled by the Poisson or Poisson–Boltzmann equation; and (4) all viscous force, electrostatic force, and vdW force are balanced on the solute-solvent interface that moves with solvent velocity. White [26] proposed to add the Landau–Lifshitz random stress tensor in the Stokes equation to model the solvent fluctuations. They also propose to describe the electrostatic interaction through the dielectric boundary force, without introducing ionic charge densities in the solvent. They further applied their model, termed dynamical implicit-solvent model (DISM), to a charged spherical molecule to derive a generalized Rayleigh–Plesset equation, a stochastic ordinary differential equation for the fluctuating radius. With the same deterministic model, Li et al. [13] study the linear stability of a cylindrical solute-solvent interface, and conclude that the viscosity can modify the critical wavelength of such stabilities. Luo et al. [14, 27] make a connection of solvent fluid mechanics model with statistical mechanics theory. They also develop numerical methods to solve the solvent fluid equations. In particular, they design boundary conditions on the boundary of a computational domain to allow solutes to change their volumes. They also implement an augmented immersed interface method for the Navier–Stokes flow with moving interface.

In this work, we develop numerical methods to solve the governing equations of the solvent fluid mechanics model. As the full model is quite complicated, we shall focus here on a two-dimensional setting for nonpolar molecular systems. Our main results are as follows:

1. We discretize the Stokes equation using a ghost fluid finite difference scheme. We show that our scheme is second-order accurate;
2. We refine the method proposed in [14, 27] to design artificial boundary conditions on the boundary of computational domain, allowing the change of solute volume;
3. We couple the level-set method with our Stokes solver to track the motion of solute-solvent interface. Our method captures both dry and wet states for some simple model systems.

We remark that our ghost fluid scheme is quite different from the augmented immersed interface method used in [14, 27]. We hope that our method can be better coupled with some other compact schemes, such as the compact coupling interface method [29] for electrostatic interactions.

The rest of the paper is organized as follows: In Section 2, we present the solvent fluid model. In particular, we describe the boundary conditions for the Stokes equation. In Section 3, we describe our numerical schemes. In Section 4, four test examples are provided to show the accuracy of our numerical schemes. In Section 5, we compute the interface around two nonpolar spherical solute atoms and the interface around two nonpolar plates. In Section 6, we draw conclusions and discuss our future work.

2 A Solvent Fluid Model

We denote by Ω the region of an underlying salvation system. It is divided into the solute region Ω_- and the solvent region Ω_+ , separated by the solute-solvent interface $\Gamma = \Omega_- \cap \Omega_+$; cf. Figure 2.1. We assume all Ω , Ω_- , and Ω_+ are open sets, and Ω_- is completely inside Ω , i.e., $\Omega_- \cap \Omega = \emptyset$. The interface Γ , the solute region Ω_- , and the solvent region Ω_+ can all depend on t . At a given time t , we denote by $\mathbf{u} = (u, v) : \Omega_+ \rightarrow \mathbb{R}^2$ and $p : \Omega_+ \rightarrow \mathbb{R}$ the velocity and pressure of the solvent fluid, respectively. We also denote by $p_- : \Omega_- \rightarrow \mathbb{R}$ the pressure inside the solute region Ω_- .

Our governing equations and boundary conditions are as follows [13, 26]:

- Interface motion:

$$\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x}(t)) \text{ for } \mathbf{x} = \mathbf{x}(t) \in \Gamma, \quad (2.1)$$

where a dot donates the time derivative.

- The Stokes equation for incompressible flow:

$$\begin{cases} \mu \Delta \mathbf{u} - \nabla p + \mathbf{G} = \mathbf{0} \text{ in } \Omega_+, \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega_+. \end{cases} \quad (2.2)$$

- The ideal-gas law:

$$p_- \text{Vol}(\Omega_-) = C_m. \quad (2.3)$$

- Traction interface conditions for the fluid velocity and pressure:

$$\begin{cases} \mu \mathbf{n} \cdot D(\mathbf{u}) \mathbf{n} - p_+ p_- = -\mathbf{f} \cdot \mathbf{n} \text{ for } \mathbf{x} \in \Gamma, \\ \mu \boldsymbol{\tau} \cdot D(\mathbf{u}) \mathbf{n} = -\mathbf{f} \cdot \boldsymbol{\tau} \text{ for } \mathbf{x} \in \Gamma. \end{cases} \quad (2.4)$$

- Boundary conditions on Ω for the velocity and pressure:

$$\mathbf{u}=\mathbf{u}_0 \text{ and } p=p_\infty \text{ on } \partial\Omega. \quad (2.5)$$

Here, μ denotes the viscosity of solvent fluid and $\mathbf{G}=(g^{(1)}, g^{(2)})$ is the density of a body force exerted in Ω_+ . In the ideal-gas law (2.3), C_m is a constant, proportional to the temperature and the number of solute atoms. If there are multiple components $\Omega_-^{(i)}(1 \leq i \leq I)$ of solutes then the ideal-gas law should be applied to each $\Omega_-^{(i)}$ and the constants $C_m^{(i)}$ for $\Omega_-^{(i)}$ can vary with i . As usual, we denote $D(\mathbf{u})=\nabla\mathbf{u}+\nabla\mathbf{u}^T$ to be the rate-of-strain tensor. The force \mathbf{f} in the traction boundary condition is given by

$$\mathbf{f}(\mathbf{x})=-\gamma H(\mathbf{x})\mathbf{n}+\mathbf{f}_{\text{vdW}}(\mathbf{x}) \text{ for } \mathbf{x} \in \Gamma,$$

where γ is the constant surface tension and H is the mean curvature. The surface vdW force \mathbf{f}_{vdW} is defined by [7, 8, 13, 29]

$$\mathbf{f}_{\text{vdW}}(\mathbf{x})=\left(\sum_{i=1}^I U_{LJ}^{(i)}(\mathbf{x})\right)\mathbf{n},$$

with each $U_{LJ}^{(i)}$ the Lennard-Jones potential for atom i . We have implicitly assumed that there are I solute atoms inside the solute region Ω_- . The boundary velocity \mathbf{u}_0 will be specified later. The boundary pressure p_∞ is a given function.

Due to the incompressibility of the solvent fluid, usual boundary conditions on Ω , such as no-slip or periodic boundary conditions, will lead to a constant volume of the solute region Ω_- . In order to simulate molecular conformational changes and multiple hydration states, which often result the volume change of solutes, we need to design numerical boundary conditions. To do so, we use the consistency condition

$$\int_{\partial\Omega}\mathbf{u} \cdot \mathbf{n}dl=\int_\Gamma\mathbf{u} \cdot \mathbf{n}dl. \quad (2.6)$$

This equates the fluid flux along Ω to that along Γ . It results from the incompressibility condition. The volume change of Ω_- is thus determined by the fluid flux along Ω . The calculation of such volume change depends on the way \mathbf{u}_0 is specified in (2.5). Here we design the boundary velocity \mathbf{u}_0 to achieve such volume change. Let $\Omega=(0, I_x) \times (0, I_y)$ for some $I_x>0$ and $I_y>0$, we specify the boundary velocity profile in one of the following two ways:

- A parabolic profile:

$$\begin{cases} u(0, y) = C(t)(l_y - y)y, & v(0, y) = 0, \\ u(l_x, y) = -C(t)(l_y - y)y, & v(l_x, y) = 0, \\ u(x, 0) = 0, & v(x, 0) = C(t)(l_x - x)x, \\ u(x, l_y) = 0, & v(x, l_y) = -C(t)(l_x - x)x. \end{cases} \quad (2.7)$$

- A circular profile:

$$\begin{cases} u(0, y) = C(t) \frac{2l_x}{l_x^2 + (2y - l_y)^2}, & v(0, y) = C(t) \frac{2l_y - 4y}{l_x^2 + (2y - l_y)^2}, \\ u(l_x, y) = -C(t) \frac{2l_x}{l_x^2 + (2y - l_y)^2}, & v(l_x, y) = C(t) \frac{2l_y - 4y}{l_x^2 + (2y - l_y)^2}, \\ u(x, 0) = C(t) \frac{4x - 2l_x}{(2x - l_x)^2 + l_y^2}, & v(x, 0) = C(t) \frac{2l_y}{(2x - l_x)^2 + l_y^2}, \\ u(x, l_y) = C(t) \frac{4x - 2l_x}{(2x - l_x)^2 + l_y^2}, & v(x, l_y) = -C(t) \frac{2l_y}{(2x - l_x)^2 + l_y^2}. \end{cases} \quad (2.8)$$

In these equations, the function $C(t)$ is to be determined by the consistency condition (2.6). The idea is that, under the assumption of the absence of external flow, the error of the boundary velocity prescribed by the profiles given in (2.7) and (2.8) would have relatively minor effect on the interface motion. This assumption is justified in Section 4 for a shrinking circle, whose analytical solution is well known.

Finally, to numerically track the motion of the solute-solvent interface Γ , we employ the level-set method. We denote by $\phi = \phi(\mathbf{x}, t)$ a level-set function of the interface $\Gamma = \Gamma(t)$ at time t , i.e., $\Gamma = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) = 0\}$. We also assume that $\Omega_- = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) < 0\}$ and $\Omega_+ = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) > 0\}$. The level-set equation is

$$\phi_t + u_n |\nabla \phi| = 0 \text{ in } \Omega, \quad (2.9)$$

where $u_n = \mathbf{u} \cdot \mathbf{n}$ is the normal component of the velocity field \mathbf{u} at the interface Γ , but is suitably extended to Ω .

In summary, we couple the Stokes equation (2.2) in Ω_+ with the traction interface conditions (2.4) and the ideal-gas law (2.3), the consistency condition (2.6), and the Dirichlet boundary condition (2.5) for p , and (2.7) or (2.8) for \mathbf{u} . The fluid velocity dictates the motion of Γ , which is tracked by the level-set equation (2.9).

3 Numerical Methods

In this section, we introduce our numerical methods. We divide our computational domain $\Omega = (0, l_x) \times (0, l_y)$ into $n_x \times n_y$ grid cells, with n_x and n_y two positive integers. We denote $h_x = l_x/n_x$ and $h_y = l_y/n_y$, and define $\mathbf{x}_{i,j} = ((i + 1/2)h_x, (j + 1/2)h_y)$, $\mathbf{x}_{i \pm 1/2, j} = ((i + 1/2 \pm 1/2)h_x, (j + 1/2)h_y)$, and $\mathbf{x}_{i, j \pm 1/2} = ((i + 1/2)h_x, (j + 1/2 \pm 1/2)h_y)$.

3.1 Discretization of the Stokes Equation

We use a marker-and-cell (MAC) grid for the unknown fluid components u , v , and p . We approximate p at the center $\mathbf{x}_{i,j}$ of each cell, u at the midpoints of vertical cell edges $\mathbf{x}_{i-1/2,j}$ and v at the midpoints of horizontal cell edges $\mathbf{x}_{i,j-1/2}$; cf. Figure 3.1. For convenience, we define the regular points, boundary points, and ghost points as follows:

1. The regular velocity points are those points on the edges of the cells and are located inside the fluid region Ω_+ .
2. The regular pressure points are those points on the center of the cells, of which at least one edge contains a regular velocity point.
3. The boundary velocity points are those points on Ω , or on an edge that intersects Ω .
4. The boundary pressure points are the center points $\mathbf{x}_{i,j}$ of those boundary cells, each of which has at least one edge entirely on Ω .
5. A ghost velocity point is a point located either inside Ω_- or on the interface Γ , and is a neighbor to a regular velocity point. Two velocity points are neighbors of each other if the edges they are on share a same vertex or if they are on the edges of the same cell.

With the assumption that Γ is far away from Ω , we can simply assume that any ghost point is not a boundary point, and any boundary point is a fluid point. We then denote

$$\begin{aligned}\mathcal{C}_u &= \{u\text{-points}\} = \{\text{regular points for } u\} \cup \{\text{boundary points for } u\} \cup \{\text{ghost points for } u\}, \\ \mathcal{C}_v &= \{v\text{-points}\} = \{\text{regular points for } v\} \cup \{\text{boundary points for } v\} \cup \{\text{ghost points for } v\}, \\ \mathcal{C}_p &= \{p\text{-points}\} = \{\text{regular points for } p\} \cup \{\text{boundary points for } p\}.\end{aligned}$$

For any non-boundary fluid point for velocity $\mathbf{x}_{i-1/2,j} \in \mathcal{C}_u$ or $\mathbf{x}_{i,j-1/2} \in \mathcal{C}_v$, or any non-boundary regular point for pressure $\mathbf{x}_{i,j} \in \mathcal{C}_p$, we apply the following second-order discretization scheme to the incompressible Stokes equation:

$$\begin{cases} \mu(D_+^x D_-^x u_{i-1/2,j} + D_+^y D_-^y u_{i-1/2,j}) + D_-^x p_{i,j} + g_{i-1/2,j}^{(1)} = 0, \\ \mu(D_+^x D_-^x v_{i,j-1/2} + D_+^y D_-^y v_{i,j-1/2}) + D_-^y p_{i,j} + g_{i,j-1/2}^{(2)} = 0, \\ D_+^x u_{i-1/2,j} + D_+^y v_{i,j-1/2} = 0, \end{cases} \quad (3.1)$$

Where

$$\begin{cases} D_+^x (\cdot)_{i,j} = \frac{(\cdot)_{i+1,j} - (\cdot)_{i,j}}{h_x}, & D_+^y (\cdot)_{i,j} = \frac{(\cdot)_{i,j+1} - (\cdot)_{i,j}}{h_y}, \\ D_-^x (\cdot)_{i,j} = \frac{(\cdot)_{i,j} - (\cdot)_{i-1,j}}{h_x}, & D_-^y (\cdot)_{i,j} = \frac{(\cdot)_{i,j} - (\cdot)_{i,j-1}}{h_y}. \end{cases}$$

Note that this scheme enforces the divergence free condition on each cell with center a non-boundary regular pressure point. For any boundary velocity point or pressure point, we apply the Dirichlet boundary condition (2.7) or (2.8), and (2.5), respectively.

The boundary velocity profile on $\Gamma(t)$ depends on a new variable $C(t)$, which is determined by discretizing (2.6). We approximate the left-hand side of (2.6) by numerical integrating along the lines $x = 0, I_x$ and $y = 0, I_y$:

$$\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} dl \approx \sum_{j=1}^{n_y} (u_{1/2,j} + u_{n_x+1/2,j}) h_x + \sum_{i=1}^{n_x} (v_{i,1/2} + v_{i,n_y+1/2}) h_y. \quad (3.2)$$

To approximate the right-hand side of (2.6), we consider the region ω bounded by Γ and the lines $x = h_x, I_x - h_x$, and $y = h_y, I_y - h_y$, and use the incompressibility condition to get

$$0 = \int_{\omega} \nabla \cdot \mathbf{u} dA = \int_{\Gamma} \mathbf{u} \cdot \mathbf{n} dl + \int_{\partial\omega \setminus \Gamma} \mathbf{u} \cdot \mathbf{n} dl.$$

Here the right-hand side of (2.6) can be approximated by

$$\int_{\Gamma} \mathbf{u} \cdot \mathbf{n} dl = - \int_{\partial\omega \setminus \Gamma} \mathbf{u} \cdot \mathbf{n} dl \approx \sum_{j=2}^{n_y-1} (u_{3/2,j} + u_{n_x-1/2,j}) h_x + \sum_{i=2}^{n_x-1} (v_{i,3/2} + v_{i,n_y-1/2}) h_y. \quad (3.3)$$

By substituting (2.7) or (2.8) into (3.3), and combining (2.6) and (3.2), we arrive at

$$C(t) = \frac{1}{\zeta} \left[\sum_{j=2}^{n_y-1} (u_{3/2,j} + u_{n_x-1/2,j}) h_x + \sum_{i=2}^{n_x-1} (v_{i,3/2} + v_{i,n_y-1/2}) h_y \right], \quad (3.4)$$

Where

$$\zeta = \begin{cases} \frac{1}{6} [(2n_x^3 + n_x) h_x^3 + (2n_y^3 + n_y) h_y^3] & \text{for parabolic profile (2.7),} \\ \sum_{i=1}^{n_x} \frac{4l_y h_x}{l_y^2 + l_x^2 (2i-1-n_x)^2} + \sum_{j=1}^{n_y} \frac{4l_x h_y}{l_x^2 + l_y^2 (2j-1-n_y)^2} & \text{for circular profile (2.8).} \end{cases} \quad (3.5)$$

For any ghost velocity point $\mathbf{x}_{i+1/2, j} \in \mathcal{C}_u$ or $\mathbf{x}_{i, j+1/2} \in \mathcal{C}_v$, we first project it onto Γ and find the projection point $\mathbf{y}_{i+1/2, j}$ or $\mathbf{y}_{i, j+1/2}$, respectively. Then we use the nearby fluid velocity points, ghost velocity points, and pressure points to form a locally third-order approximation to the derivative components u_x, u_y, v_x, v_y , and pressure component p at $\mathbf{y}_{i+1/2, j}$ and $\mathbf{y}_{i, j+1/2}$, respectively. Thus, we obtain a third-order approximation to the traction interface conditions

at the projection points. This third-order approximation scheme on the interface covers various cases of the local geometry of interface. In some other cases, where the geometry of Γ does not allow enough points for a third order scheme, we switch to a second order or even first order scheme, which causes the solution to be reverted to a lower order approximation. We include a detailed discussion and formulation of the discretization scheme for ghost velocity points in the appendix.

By discretizing the system as illustrated above, we obtain a linear system, with an array of unknown composed of u , v at the fluid points, p at the pressure points, and $C(t)$. The discretization matrix \mathcal{A} is asymmetric and sparse. It takes the following form

$$\mathcal{A} = \begin{bmatrix} A_u & O_u & A_{p_x} & \mathbf{b}_u^t \\ O_v & A_v & A_{p_y} & \mathbf{b}_v^t \\ A_{u_x} & A_{v_y} & O_p & \mathbf{0}^t \\ \mathbf{c}_u & \mathbf{c}_v & \mathbf{0} & 1 \end{bmatrix}, \quad (3.6)$$

where the last row and last column correspond to (3.4) and (3.5), respectively. Rows of A_u , A_v correspond mainly to the five-point stencils for Laplacian. All entries of O_u , O_v , and O_p are zero, except a few that correspond to ghost points. All A_{p_x} , A_{p_y} , A_{u_x} , and A_{v_y} result mainly from the discretization of p_x , p_y , u_x , and v_y , respectively. In each of these submatrices, there are rows corresponding to the ghost points, where the discretization of the (2.4) couples the u , v , and p points and introduces many nonzero off-diagonal entries. This linear problem is similar in structure to a saddle point problem. It may suggest a solution method involving the Schur complement reduction. However, the highly coupled interface conditions together with the geometry dependent discretization make the conditional number of \mathcal{A} very high. Furthermore, submatrices such as A_u , A_v are far away from diagonal dominant. At a few ghost points, the diagonal entries have their magnitude smaller than off-diagonals. As a consequence, it is not efficient to apply any Krylov subspace solver to even a subproblem. Leaving the development of numerical algorithms to future work, as a first step, we use UMFPACK, an implementation of a direct multifrontal sparse LU factorization method, to solve this system [6]. The time cost for UMFPACK to solve a sparse linear system of size $\sim 30,000$ -by- $30,000$ is $\mathcal{O}(0.1)$ seconds.

3.2 Solving the Level-Set Equation

We now consider the discretization of the level-set equation (2.9). For the time derivative, we use the explicit forward Euler scheme

$$\frac{\phi^{(k+1)}(\mathbf{x}) - \phi^{(k)}(\mathbf{x})}{\Delta t} + u_n^{(k)}(\mathbf{x}) |\nabla \phi^{(k)}(\mathbf{x})| = 0, \quad (3.7)$$

where $\phi^{(k)}(\mathbf{x})$ and $u_n^{(k)}(\mathbf{x})$ are the approximations of $\phi(\mathbf{x}, t_k)$ and $u_n(\mathbf{x}, t_k)$, respectively, at time $t_k = k \Delta t$ ($k = 1, 2, \dots$) and Δt is the time step satisfying the CFL condition

$$\Delta t = \frac{0.25h}{\max_{\mathbf{x}} |u_n(\mathbf{x}, t_k)|}. \quad (3.8)$$

If $u_n(\mathbf{x}, t) = 0$ for all \mathbf{x} on Γ at a certain time t , then Γ no longer moves, and a steady state interface is reached.

To approximate the normal velocity $u_n(\mathbf{x})$, we use the level-set function, to interpolate the points on the interface, calculate the normal fluid velocity on these points, and use a fast marching algorithm to extend the values in the normal direction to all spatial grid points. Note that we here use the same grid as that for the discretization of the pressure p . For spatial derivatives, we use a fifth-order WENO method to approximate $|\nabla \phi^{(k)}(\mathbf{x})|$. We use the homogeneous Neumann boundary conditions at the outer boundary of the computational box.

To keep the level-set function as a signed distance function, we reinitialize ϕ by solving the equation

$$\phi_t + \text{sign}(\phi_0)(|\nabla \phi| - 1) = 0$$

for a few steps with the initial value $\phi = \phi_0$ at $t = 0$. Here ϕ_0 is the level-set function before reinitialization, and the time t is different from that in the original level-set equation.

4 Numerical Tests

4.1 Flow Outside a Circular Region

In this example, we test the convergence of our Stokes solver on flow outside a circular region. We set $\Omega = (0, 1) \times (0, 1)$ and $\Omega_- = \{(x, y) \in \Omega : \phi(x, y) < 0\}$, where the level-set function is given by

$$\phi(x, y) = \sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.13. \quad (4.1)$$

Note that Ω_- is a circular region. Note also that the region Ω_- and the interface Γ are fixed all the time. So there is no moving interface in this test. Furthermore, we set $\mu = 1$ and fix $p_- = 0$. We select the boundary velocity \mathbf{u}_0 , the boundary pressure p_∞ , the fluid body force \mathbf{G} , and the interface force \mathbf{f} to yield the following velocity and pressure fields for an incompressible flow:

$$\begin{cases} u(x, y) = \sin(2\pi x)\cos(2\pi y), \\ v(x, y) = -\cos(2\pi x)\sin(2\pi y), \\ p(x, y) = -\cos 2\pi x \cos(2\pi y). \end{cases} \quad (4.2)$$

We then solve the Stokes equation (2.2) on $\Omega_+ = \{(x, y) \in \Omega : \varphi(x, y) > 0\}$ with such boundary conditions. The numerical solutions of u , v , and p to this test example are plotted in Figure 4.1 on an $N \times N$ spatial grid with $N = 400$. We analyze the error between the numerical solutions and the analytical solutions (4.2), by generating in Figure 4.2 six log-log plots of the L^2 -norm and L^∞ -norm of the error for u , v , and p versus N , the number of grid points in both x and y directions. The log-log plots show that the error for u , v , and p all decay in an order of $O(N^{-2})$ in both L^2 -norm and L^∞ -norm, with spikes intermittently. These spikes arise due to the insufficient grid resolution for the curved interfacial geometry resulting unpredicted sudden increase of the conditional number of discretization matrix. In average, the conditional number increases in the order of $O(N^3)$. We believe that such spikes can be reduced by discretizing the interface condition using a least-squares approach.

4.2 Flow Outside a Clover Shaped Region

This example is the same as that in Subsection 4.1, except that $\Omega_- = (1/2, 1/2) + \Omega_c$, where Ω_c is a three-fold clover region defined in polar coordinates as

$$\Omega_c = \left\{ (r, \theta) : r < \frac{1}{4} + \frac{1}{16} \cos(3\theta) \right\}. \quad (4.3)$$

Our numerical solutions of u , v , and p are plotted in Figure 4.3, with the spatial discretization parameter $N = 400$ in both x and y directions. An analysis of the error between the numerical and analytical solutions shows similar behavior than the previous example in terms of L^2 -norm and L^∞ -norm of the error on the three fluid components u , v , and p , as shown in Figure 4.4. We get an average second-order convergence in u , v , and p .

4.3 Flow around a Disk with Designed Numerical Boundary Conditions

We now test on our numerical boundary conditions. Define

$$\phi = \sqrt{(x - 0.5)^2 + (y - 0.5)^2} - 0.2. \quad (4.4)$$

Define also the circular region $\Omega_- = \{(x, y) \in \mathbb{R}^2 : \varphi(x, y) < 0\}$. One can verify that

$$p(\mathbf{x}) = 0 \text{ and } \mathbf{u}(\mathbf{x}) = - \frac{0.02(x - 0.5, y - 0.5)}{(x - 0.5)^2 + (y - 0.5)^2} \quad (4.5)$$

solve the Stokes equation (2.2) with $\mu = 1$, $\mathbf{G} = \mathbf{0}$, together with the interface condition $p_- = 0$, and $\mathbf{f} = -\mathbf{n}$, and also that its flux along Γ is

$$-\int_{\Gamma} \mathbf{u} \cdot \mathbf{n} dl = 0.04\pi. \quad (4.6)$$

We solve the Stokes equation numerically for \mathbf{u} and p on $\Omega = (0, 1) \times (0, 1)$ with $p_{\infty} = 0$, and the boundary velocity profile given by (2.7) or (2.8). In Figure 4.5, we plot the solution with profile (2.7) and $n_x = n_y = N = 101$. The tendency of shrinking of the volume of Ω_- can be observed from the inward velocity field.

As we increase N , the flux $\zeta C(t)$ approaches a constant. In Figure 4.6, we plot the flux $\zeta C(t)$ as a function of N . We see that for both (2.7) and (2.8), $\zeta C(t)$ converges to a certain value. For (2.8), $\zeta C(t)$ approaches the analytical value 0.04π , because the circular velocity profile is exact in this case. For (2.7), $\zeta C(t)$ approaches to a value which is slightly less than, but reasonably close to 0.04π . This shows that even though the assumptions of boundary velocity profiles are artificial, they can be expected to work relatively accurately. To test the convergence of $\zeta C(t)$ with respect to increasing N , we take the value of $\zeta C(t)$ at $N = 600$ as a reference value for both (2.7) and (2.8). We then compute the absolute values of the differences between the fluxes at the other values of N and the reference values. This process gives us the convergence plot in Figure 4.6, from which we observe a convergence rate of roughly first order for (2.7) and second order for (2.8).

4.4 Moving Interface Driven by Solvent Fluid Flow with Curvature Force

We now test our level-set method coupled with our Stokes solver on a moving circular interface. Let us define

$$p(\mathbf{x})=0 \text{ and } \mathbf{u}(\mathbf{x}) = -\frac{r(t)(2x-1, 2y-1)}{(2x-1)^2 + (2y-1)^2}, \quad (4.7)$$

where $r(t) = 0.2 - t/2$ with $0 \leq t < 0.4$. It is easy to verify that p and \mathbf{u} solve the Stokes equation (2.2) with $\mu = 1$ and $\mathbf{G} = \mathbf{0}$ in the region $\mathbb{R}^2 / \overline{\Omega_-(t)}$, where

$$\Omega_-(t) = \{(x, y) \in \mathbb{R}^2 : (x - 0.5)^2 + (y - 0.5)^2 < r(t)^2\}.$$

Moreover, p and \mathbf{u} satisfy the interface condition (2.4) on $\Gamma(t) = \partial\Omega_-(t)$ with $p_- = 0$ and $\mathbf{f} = -\mathbf{n}/r(t)$ (i.e., the curvature flow). The time needed for the circle $\Gamma(t)$ to shrink to the center $(0.5, 0.5)$ is $t = 0.4$.

We now set $\Omega = (0, 1) \times (0, 1)$, and the initial interface to be

$$\Gamma(0) = \{(x, y) \in \Omega : (x - 0.5)^2 + (y - 0.5)^2 = 0.2^2\}.$$

We use our level-set method and ghost fluid Stokes solver to solve numerically (2.1) and (2.2) with $\mathbf{G} = \mathbf{0}$, (2.4) with $p_- = 0$ and $\mathbf{f} = -H\mathbf{n}$, where H is the curvature, and the boundary conditions $p_\infty = 0$ and boundary profile (2.7) and (2.8). Our numerical solution for $\Gamma(t)$ as plotted in Figure 4.7 exhibits circular shrinking with constant speed. The center is slightly shifted due to small numerical error. The numerical critical time t_c for both boundary profiles (2.7) and (2.8) agree well with the analytic value.

5 Applications

In this section, we present two examples of application of our method to two model nonpolar molecular systems. One is a two-particle system and the other is a two plate system.

5.1 A Two-Particle System

We consider $\Omega = (0, l_x) \times (0, l_y)$ with l_x and l_y two positive numbers. We assume two particles located at $\mathbf{x}_1 = (l_x/2 - c, l_y/2)$ and $\mathbf{x}_2 = (l_x/2 + c, l_y/2)$, respectively, where $c > 0$ is an adjustable parameter. We define the initial regions Ω_- and Ω_+ using this level-set function

$$\phi(x, y) = \begin{cases} \sqrt{(x - l_x/2 + c)^2 + (y - l_y/2)^2} & \text{if } x \leq l_x/2 - c, \\ |y - l_y/2| & \text{if } l_x/2 - c \leq x \leq l_x/2 + c, \\ \sqrt{(x - l_x/2 - c)^2 + (y - l_y/2)^2} & \text{if } x \geq l_x/2 + c. \end{cases} \quad (5.1)$$

We set the body force $\mathbf{G} = \mathbf{0}$. We also combine the curvature force and Lenard-Jones force into the surface force

$$\mathbf{f} = \left(-\gamma \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) + 4\varepsilon \sum_{i=1}^2 \left(\frac{\sigma^{12}}{|\mathbf{x} - \mathbf{x}_i|^{12}} - \frac{\sigma^6}{|\mathbf{x} - \mathbf{x}_i|^6} \right) \right) \mathbf{n}, \quad (5.2)$$

where γ , ε , and σ are all constants, and take the following values for this example: $\gamma = 1$, $\varepsilon = 25$, $\sigma = 0.2$. Furthermore, we take $C_m = 0.001$ in (2.3) and (2.4).

We then solve the Stokes equation coupled with level-set equation (2.9), with $l_y = 1$ and various choices of l_x and c : (1) $l_x = 1$, $c = 0.05$; (2) $l_x = 1$, $c = 0.2$; (3) $l_x = 2$, $c = 0.25$. The final steady state for Γ is plotted in Figure 5.1. We see that the interface breaks into two parts as the distance between the two particles is large enough; cf. case (3).

5.2 A Two-Plate System

In this example, we set $\Omega = (0, 1) \times (0, 1)$ and two plates composed of six atoms each. These atoms are located at coordinates $(0.5 - c, 0.25 + 0.1k)$ and $(0.5 + c, 0.25 + 0.1k)$ with $k \in \{0, 1, \dots, 5\}$ and c an adjustable constant. With certain parameter choices, one observe dry/wet polymodal states depending on the initial solute region Ω_- . We choose $c = 0.15$, $\gamma = 5$, $\varepsilon = 25$, $\sigma = 0.1$, $C_m = 0.001$ in (5.2). Moreover, for a loose initial condition of Γ , we define

$$\Omega_- = \{(x, y) \in \Omega : 0.15 \leq x \leq 0.85, 0.2 \leq y \leq 0.8\}.$$

For a tight initial condition on Γ , we define

$$\Omega_- = \{(x, y) \in \Omega : 0.25 \leq x \leq 0.45, 0.2 \leq y \leq 0.8\} \cup \{(x, y) \in \Omega : 0.55 \leq x \leq 0.75, 0.2 \leq y \leq 0.8\}.$$

We solve the Stokes equation (2.2) coupled with level-set equation (2.9). The interface Γ goes to a dry final state from the loose initial condition, and a wet final state from the tight initial condition; cf. Figure 5.2. The evolution of Γ in between the two states captures the dynamics of the transitioning process, which cannot be obtained by an energy variational approach such as VISM. In our future work, we would like to develop a 3D fluid solver, and compare our numerical results with the results obtained from molecular dynamics simulations.

6 Conclusions

In this paper, we model the aqueous solvent by the incompressible Stokes flow and treat the solute with the ideal-gas law. The solute-solvent interface moves with the solvent fluid flow. All the viscous, pressure, surface tension, and the solute-solvent vdW forces are balanced on the interface, leading to a traction boundary condition. To allow the change of solute volume, we design special numerical boundary conditions on the boundary of our computation domain. We design a second-order ghost fluid method for solving the Stokes equation. We also couple the level-set equation for the moving solute-solvent interface. Our methods accurately predict the blowup time of a shrinking bubble under surface tension. Moreover, our methods capture the dry and wet polymodal interfaces for the two-plate system.

Some existing issues of this model include: (1) The boundary conditions assume a velocity profile, which may not be realistic; (2) The numerical error is sensitive to the location of the interface; (3) We solve the linear system using a direct LU factorization, which can be problematic in extension to three dimensions.

As a first step in the future, we need to propose more realistic boundary conditions. We also plan to decrease the sensitivity of error dependence on the interface by considering a least square problem. Moreover, an extension of our two dimensional fluid solver to three dimensions is necessary. After we develop such a fluid solver, we can add noise to the solvent and observe the fluctuations of the interface. Through a combination of such a three dimensional fluid solver with fluctuations and a robust Poisson–Boltzmann solver, one can better observe and describe the dynamics of a solvent–solute system.

Acknowledgments

This work was supported by the US National Science Foundation (NSF) through grant DMS-1319731 and the US National Institutes of Health (NIH) through grant R01GM096188. Work in McCammon's group is supported in part

by NSF, NIH, HHMI, and NBCR. The authors thank Dr. Robert Krasny, Dr. Ray Luo, and Mr. Li Xiao for helpful discussions.

Appendix

In the appendix, we provide details of the ghost fluid discretization on the interface. First of all, with the notations $\mathbf{n} = (n_1, n_2)$ and $\boldsymbol{\tau} = (-n_1, n_2)$, the traction boundary conditions (2.4) read

$$\begin{cases} 2n_1^2 u_x + 2n_1 n_2 (u_y + v_x) + 2n_2^2 v_y - p = f_{\perp}, \\ -2n_1 n_2 u_x + (n_1^2 - n_2^2)(u_y + v_x) + 2n_1 n_2 v_y = f_{\parallel}, \end{cases}$$

where $f_{\perp} = \mathbf{f} \cdot \mathbf{n}$ and $f_{\parallel} = \mathbf{f} \cdot \boldsymbol{\tau}$. Some straightforward algebraic calculations together with the incompressibility condition (2.2) lead to

$$2u_x - (n_1^2 - n_2^2)p = (n_1^2 - n_2^2)f_{\perp} - 2n_1 n_2 f_{\parallel}, \quad (\text{A.1})$$

$$2(n_2^2 - n_1^2)v_y + 2n_1 n_2 (u_y + v_x) - p = f_{\perp}. \quad (\text{A.2})$$

For any ghost velocity point \mathbf{x} , we find a point $\mathbf{x}^* \in \Gamma$, such that $|\mathbf{x} - \mathbf{x}^*| = \text{dist}(\mathbf{x}, \Gamma)$. We call \mathbf{x}^* a projection point of \mathbf{x} onto Γ . We then discretize equation (A.1) at each projection point $\mathbf{x}_{i-1/2, j}^*$ corresponding to each ghost velocity point $\mathbf{x}_{i-1/2, j}$ of u . Similarly, we discretize equation (A.2) at each projection point $\mathbf{x}_{i, j-1/2}^*$ corresponding to each ghost velocity point $\mathbf{x}_{i, j-1/2}$ of v .

To obtain a second-order convergence scheme for u , v , and p up to Γ , we design a third-order discretization of u_x , u_y , v_x , v_y , and p in the fluid region. This requires 10 stencil points for u , v , and 6 stencil points for p . We denote by $\mathcal{S}(u, \mathbf{x}^*, r)$, $\mathcal{S}(v, \mathbf{x}^*, r)$, and $\mathcal{S}(p, \mathbf{x}^*, r)$ respectively, the sets of u , v and p stencil points for discretizing ∇_u , ∇_v and p with an order r at the projection point $\mathbf{x}_{i, j-1/2}^*$. Then

$$|\mathcal{S}(u, \mathbf{x}^*, r)| = \frac{(r+1)(r+2)}{2}, \quad |\mathcal{S}(v, \mathbf{x}^*, r)| = \frac{(r+1)(r+2)}{2}, \quad \text{and} \quad |\mathcal{S}(p, \mathbf{x}^*, r)| = \frac{r(r+1)}{2}.$$

In choosing these stencil points, we follow three criteria: (1) Each of these stencil points needs to be either a ghost point or a fluid point; (2) The stencil points need to include \mathbf{x} , that is $\mathbf{x} \in \mathcal{S}(u, \mathbf{x}^*, r) \cup \mathcal{S}(v, \mathbf{x}^*, r) \cup \mathcal{S}(p, \mathbf{x}^*, r)$; (3) All $\mathcal{S}(u, \mathbf{x}^*, r)$, $\mathcal{S}(v, \mathbf{x}^*, r)$, and $\mathcal{S}(p, \mathbf{x}^*, r)$ satisfy

$$\left\{ \begin{array}{l} \max\{i:\mathbf{x}_{i,j} \in S(u, \mathbf{x}^*, r)\} - \min\{i:\mathbf{x}_{i,j} \in S(u, \mathbf{x}^*, r)\}=r, \\ \max\{j:\mathbf{x}_{i,j} \in S(u, \mathbf{x}^*, r)\} - \min\{j:\mathbf{x}_{i,j} \in S(u, \mathbf{x}^*, r)\}=r, \\ \max\{i:\mathbf{x}_{i,j} \in S(v, \mathbf{x}^*, r)\} - \min\{i:\mathbf{x}_{i,j} \in S(v, \mathbf{x}^*, r)\}=r, \\ \max\{j:\mathbf{x}_{i,j} \in S(v, \mathbf{x}^*, r)\} - \min\{j:\mathbf{x}_{i,j} \in S(v, \mathbf{x}^*, r)\}=r, \\ \max\{i:\mathbf{x}_{i,j} \in S(p, \mathbf{x}^*, r)\} - \min\{i:\mathbf{x}_{i,j} \in S(p, \mathbf{x}^*, r)\}=r-1, \\ \max\{j:\mathbf{x}_{i,j} \in S(p, \mathbf{x}^*, r)\} - \min\{j:\mathbf{x}_{i,j} \in S(p, \mathbf{x}^*, r)\}=r-1. \end{array} \right. \quad (\text{A.3})$$

The criterion (3) is important for the invertibility of matrix A in equation (A.4), as shall be discussed later.

We now describe the process of constructing $S(u, \mathbf{x}^*, r)$, $S(v, \mathbf{x}^*, r)$, and $S(p, \mathbf{x}^*, r)$, and the corresponding schemes. In this process, we use the following notations

$$(\lambda_1 h_x, \lambda_2 h_y) = \mathbf{x}^* - \mathbf{x}, (s_1, s_2) = (\text{sign}(\lambda_1), \text{sign}(\lambda_2)).$$

It is easy to see that $|\lambda_1| < 1$ and $|\lambda_2| < 1$. Since a ghost point is a neighbor to a fluid point, at least one of its neighbor needs to be in Ω_+ . We name a neighbor of \mathbf{x} a check point, if that neighbor point is in Ω_+ . There are totally six different cases for the combination of ghost point and check point: (a) $\mathbf{x} = \mathbf{x}_{i-1/2,j}$ and $\varphi(\mathbf{x}_{i-1/2+j,s_1}) > 0$; (b) $\mathbf{x} = \mathbf{x}_{i-1/2,j}$ and $\varphi(\mathbf{x}_{i-1/2+j,s_2}) > 0$; (c) $\mathbf{x} = \mathbf{x}_{i-1/2,j}$ and $\varphi(\mathbf{x}_{i-1/2+j,s_1/2, j+s_2/2}) > 0$; (d) $\mathbf{x} = \mathbf{x}_{i,j-1/2}$ and $\varphi(\mathbf{x}_{i+j,s_1, j-1/2}) > 0$; (e) $\mathbf{x} = \mathbf{x}_{i,j-1/2}$ and $\varphi(\mathbf{x}_{i, j-1/2+s_2}) > 0$; (f) $\mathbf{x} = \mathbf{x}_{i,j-1/2}$ and $\varphi(\mathbf{x}_{i+j,s_1/2, j-1/2+s_2/2}) > 0$. For each of these cases, we obtain the corresponding $S(u, \mathbf{x}^*, r)$, $S(v, \mathbf{x}^*, r)$, and $S(p, \mathbf{x}^*, r)$ with $r \in \{1, 2, 3\}$. In Figure A.1, we schematically plot all six cases and the corresponding $S(u, \mathbf{x}^*, r)$, $S(v, \mathbf{x}^*, r)$, and $S(p, \mathbf{x}^*, r)$ for $r = 3$. Notice that for cases (a)–(c), equation (A.1) is discretized, and $S(v, \mathbf{x}^*, r)$ is not needed.

We now describe the steps of constructing a third-order discretization scheme for case (a), whereas all other cases just follow tediously. We consider a u ghost point $\mathbf{x}_{i-1/2,j}$ with its projection point \mathbf{x}^* . Let us denote

$$\begin{aligned} \mathbf{X} &= (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9)^T \\ &= \left(\mathbf{x}_{i-\frac{1}{2},j}, \mathbf{x}_{i-\frac{1}{2}+s_1,j}, \mathbf{x}_{i-\frac{1}{2}-s_1,j+s_2}, \mathbf{x}_{i-\frac{1}{2},j+s_2}, \mathbf{x}_{i-\frac{1}{2}+s_1,j+s_2}, \mathbf{x}_{i-\frac{1}{2}+2s_1,j+s_2}, \mathbf{x}_{i-\frac{1}{2},j+2s_2}, \mathbf{x}_{i-\frac{1}{2}+s_1,j+2s_2}, \mathbf{x}_{i-\frac{1}{2}+2s_1,j+2s_2}, \mathbf{x}_{i-\frac{1}{2},j+2s_2} \right) \end{aligned}$$

and $\mathbf{F} = (u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, u_{xxx}, u_{xyy}, u_{xxy}, u_{yyy})^T$, where $(\cdot)^T$ denotes transpose. Then a third-order Taylor expansion leads to a linear system $\mathbf{X} = A\mathbf{F}(\mathbf{x}^*)$, where

$$A = \begin{bmatrix} 1 & h_{x,1} & h_{y,1} & h_{x,1}^2/2 & h_{x,1}h_{y,1} & h_{y,1}^2/2 & h_{x,1}^3/6 & h_{x,1}^2h_{y,1}/2 & h_{x,1}h_{y,1}^2/2 & h_{y,1}^3/6 \\ 1 & h_{x,2} & h_{y,2} & h_{x,2}^2/2 & h_{x,2}h_{y,2} & h_{y,2}^2/2 & h_{x,2}^3/6 & h_{x,2}^2h_{y,2}/2 & h_{x,2}h_{y,2}^2/2 & h_{y,2}^3/6 \\ 1 & h_{x,3} & h_{y,3} & h_{x,3}^2/2 & h_{x,3}h_{y,3} & h_{y,3}^2/2 & h_{x,3}^3/6 & h_{x,3}^2h_{y,3}/2 & h_{x,3}h_{y,3}^2/2 & h_{y,3}^3/6 \\ 1 & h_{x,4} & h_{y,4} & h_{x,4}^2/2 & h_{x,4}h_{y,4} & h_{y,4}^2/2 & h_{x,4}^3/6 & h_{x,4}^2h_{y,4}/2 & h_{x,4}h_{y,4}^2/2 & h_{y,4}^3/6 \\ 1 & h_{x,5} & h_{y,5} & h_{x,5}^2/2 & h_{x,5}h_{y,5} & h_{y,5}^2/2 & h_{x,5}^3/6 & h_{x,5}^2h_{y,5}/2 & h_{x,5}h_{y,5}^2/2 & h_{y,5}^3/6 \\ 1 & h_{x,6} & h_{y,6} & h_{x,6}^2/2 & h_{x,6}h_{y,6} & h_{y,6}^2/2 & h_{x,6}^3/6 & h_{x,6}^2h_{y,6}/2 & h_{x,6}h_{y,6}^2/2 & h_{y,6}^3/6 \\ 1 & h_{x,7} & h_{y,7} & h_{x,7}^2/2 & h_{x,7}h_{y,7} & h_{y,7}^2/2 & h_{x,7}^3/6 & h_{x,7}^2h_{y,7}/2 & h_{x,7}h_{y,7}^2/2 & h_{y,7}^3/6 \\ 1 & h_{x,8} & h_{y,8} & h_{x,8}^2/2 & h_{x,8}h_{y,8} & h_{y,8}^2/2 & h_{x,8}^3/6 & h_{x,8}^2h_{y,8}/2 & h_{x,8}h_{y,8}^2/2 & h_{y,8}^3/6 \\ 1 & h_{x,9} & h_{y,9} & h_{x,9}^2/2 & h_{x,9}h_{y,9} & h_{y,9}^2/2 & h_{x,9}^3/6 & h_{x,9}^2h_{y,9}/2 & h_{x,9}h_{y,9}^2/2 & h_{y,9}^3/6 \end{bmatrix}$$

(A.4)

with $(h_{x,k}, h_{y,k}) := \mathbf{x}_k - \mathbf{x}^*$. A sufficient condition to have A invertible is (A.3). Inverting A symbolically, we get $\mathbf{F}(\mathbf{x}^*) = A^{-1}\mathbf{X}$, and the second entry of $\mathbf{F}(\mathbf{x}^*)$ corresponds to a third-order discretization of $u_x(\mathbf{x}^*)$:

$$u_x(\mathbf{x}^*) = -\frac{(\lambda_2 - s_2)(\lambda_2 - 2s_2)}{2s_1h_x}u_0 + \frac{(\lambda_2 - s_2)(\lambda_2 - 2s_2)}{2s_1h_x}u_1 - \frac{3\lambda_1^2 - 6\lambda_1s_1 + 2}{6s_1h_x}u_2 - \frac{s_1(\lambda_1 + \lambda_1\lambda_2s_2) + 3\lambda_2s_2/2 - \lambda_2^2 - 3\lambda_1^2/2}{s_1h_x}u_3 - \frac{s_1(\lambda_1}{2s_1h_x}$$

where the notation $\mathbf{u}(\mathbf{X}) = (u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9)_T$ is used. The third entry of $\mathbf{F}(\mathbf{x}^*)^t$ gives a third-order discretization of $u_y(\mathbf{x}^*)$. However, it is not necessary in discretizing equation (A.1).

Similarly, one can derive a third-order discretization scheme for $p(\mathbf{x}^*)$:

$$p(\mathbf{x}^*) = \left(\frac{3}{8} - \lambda_1s_1 + \frac{\lambda_1^2}{2}\right)p_0 - \frac{(2\lambda_1 - 3s_1)(2\lambda_1s_2 - 2\lambda_2s_1 + 3s_1s_2)}{4s_2}p_1 + \frac{((2\lambda_1s_2 - 2\lambda_2s_1 + s_1s_2)(2\lambda_1s_2 - 2\lambda_2s_1 + 3s_1s_2))}{8}p_2 - \frac{(\lambda_2 - s_2)(2\lambda_1 - 3s_1)}{2s_1s_2}p_3 +$$

$$\text{Where } p_0 = p(\mathbf{x}_{i-\frac{s_1+1}{2}, j+s_2}), p_1 = p(\mathbf{x}_{i-\frac{s_1-1}{2}, j+s_2}), p_2 = p(\mathbf{x}_{i+\frac{3s_1-1}{2}, j+s_2}), \\ p_3 = p(\mathbf{x}_{i+\frac{s_1-1}{2}, j+2s_2}), p_4 = p(\mathbf{x}_{i+\frac{3s_1-1}{2}, j+2s_2}), p_5 = p(\mathbf{x}_{i+\frac{3s_1-1}{2}, j+3s_2}).$$

We use this symbolic inverse matrix method to find a third-order scheme for discretizing equation (A.1) around \mathbf{x}^* . To find a second-order or first-order scheme, we form a smaller matrix A by applying a second or first-order Taylor expansion to $u(\mathbf{x}_k)$, $v(\mathbf{y}_k)$, and $p(\mathbf{z}_k)$ at \mathbf{x}^* , with $\mathbf{x}_k \in \mathcal{S}(u, \mathbf{x}^*, r)$, $\mathbf{y}_k \in \mathcal{S}(v, \mathbf{x}^*, r)$, and $\mathbf{z}_k \in \mathcal{S}(p, \mathbf{x}^*, r)$, respectively. Using similar computations, we are able to construct the first, second, and third-order schemes for all six cases (a)–(f), but we do not need to record the ten-page formulas here.

References

1. Alexander-Katz A, Schneider MF, Schneider SW, Wixforth A, Netz RR. Shear–flow–induced unfolding of polymeric globules. *Phys Rev Lett*. 2006; 97:138101. [PubMed: 17026077]
2. Baron R, McCammon JA. Molecular recognition and ligand association. *Annu Rev Phys Chem*. 2013; 64:151–175. [PubMed: 23473376]
3. Chen J, Brooks CL III, Khandogin J. Recent advances in implicit solvent based methods for biomolecular simulations. *Curr Opin Struct Biol*. 2008; 18:140–148. [PubMed: 18304802]
4. Cheng L, Dzubiella J, McCammon JA, Li B. Application of the level–set method to the implicit solvation of nonpolar molecules. *J Chem Phys*. 2007; 127:084503. [PubMed: 17764265]
5. Cheng L, Li B, Wang Z. Level–set minimization of potential controlled hadwiger valuations for molecular solvation. *J Comput Phys*. 2010; 229:8497–8510. [PubMed: 22323839]
6. Davis TA. Algorithm 832: UMFPACK v4.3—an unsymmetric–pattern multifrontal method. *ACM Trans Math Softw*. 2004; 30:196–199.
7. Dzubiella J, Swanson J, McCammon J. Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models. *Phys Rev Lett*. 2006; 96:087802. [PubMed: 16606226]
8. Dzubiella J, Swanson J, McCammon J. Coupling nonpolar and polar solvation free energies in implicit solvent models. *J Chem Phys*. 2006; 124:084905. [PubMed: 16512740]
9. Guo Z, Li B, Dzubiella J, Cheng LT, McCammon JA, Che J. Heterogeneous hydration of p53/MDM2 complex. *J Chem Theory Comput*. 2014; 10:1302–1313. [PubMed: 24803860]
10. Hagen SJ. Solvent viscosity and friction in protein folding dynamics. *Curr Protein Pept Sci*. 2010; 11:385–395. [PubMed: 20426733]
11. Klimov DK, Thirumalai D. Viscosity dependence of folding rates of protein. *Phys Rev Lett*. 1997; 79:317–320.
12. Levy Y, Onuchic JN. Water mediation in protein folding and molecular recognition. *Annu Rev Biophys Biomol Struct*. 2006; 35:389–415. [PubMed: 16689642]
13. Li B, Sun H, Zhou S. Stability of a cylindrical solute–solvent interface: Effect of geometry, electrostatics, and hydrodynamics. *SIAM J Appl Math*. 2015; 75:907–928. [PubMed: 26877555]
14. Li Z, Cai Q, Zhao H, Luo R. A semi–implicit augmented IIM for Navier–Stokes equations with open and traction boundary conditions. *J Comput Phys*. 2015; 297:182–193. [PubMed: 27087702]
15. Roux B, Simonson T. Implicit solvent models. *Biophys Chem*. 1999; 78:1–20. [PubMed: 17030302]
16. Schneider SW, Nuschele S, Wixforth A, Gorzelanny C, Alexander-Katz A, Netz RR, Schneider MF. Shear–induced unfolding triggers adhesion of von Willebrand factor fibers. *Proc Natl Acad Sci USA*. 2007; 104:7899–7903. [PubMed: 17470810]
17. Sekhar A, Latham MP, Vallurupalli P, Kay LE. Viscosity-dependent kinetics of protein conformational exchange: Microviscosity effects and the need for a small viscogen. *J Phys Chem B*. 2014; 118:4546–4551. [PubMed: 24707961]
18. Setny P, Wang Z, Cheng LT, Li B, McCammon JA, Dzubiella J. Dewetting–controlled binding of ligands to hydrophobic pockets. *Phys Rev Lett*. 2009; 103:187801. [PubMed: 19905832]
19. Siedlecki CA, Lestini BJ, Kottke-Marchant KK, Eppell SJ, Wilson DL, Marchant RE. Shear–dependent changes in the three–dimensional structure of human von Willebrand factor. *Blood*. 1996; 88:2939–2950. [PubMed: 8874190]
20. Singh I, Themistou E, Porcar L, Neelamegham S. Fluid shear induces conformation change in human blood protein von Willebrand factor in solution. *Biophys J*. 2009; 96:2313–2320. [PubMed: 19289057]
21. Szymczak P, Cieplak M. Hydrodynamic effects in proteins. *J Phys: Condens Matter*. 2011; 23:033102. [PubMed: 21406855]
22. Tanford, C. *The Hydrophobic Effect: Formation of Micelles and Biological Membranes*. John Wiley; New York: 1973.
23. Tomasi J, Persico M. Molecular interactions in solution: An overview of methods based on continuous distributions of the solvent. *Chem Rev*. 1994; 94:2027–2094.

24. Vergauwe RMA, Uji-i H, De Ceunynck K, Vermant J, Vanhoorelbeke K, Hofkens J. Shear-stress-induced conformational changes of von Willebrand factor in water-glycerol mixture observed with single molecule microscopy. *J Phys Chem B*. 2014; 118:5660–5669. [PubMed: 24754487]
25. Wang Z, Che J, Cheng L, Dzubiella J, Li B, McCammon JA. Level-set variational implicit-solvent modeling of biomolecules with the coulomb-field approximation. *J Chem Theory Comput*. 2012; 8:386–397. [PubMed: 22346739]
26. White, M. PhD thesis. University of California; San Diego: 2015. Mathematical Theory and Numerical Methods for Biomolecular Modeling.
27. Xiao L, Cai Q, Li Z, Zhao H, Luo R. A multi-scale method for dynamics simulation in continuum solvents I: Finite-difference algorithm for Navier–Stokes equation. *Chem Phys Lett*. 2014; 616:67–74. [PubMed: 25404761]
28. Zhou S, Cheng L, Sun H, Che J, Dzubiella J, Li B, McCammon JA. LS-VISM: A software package for analysis of biomolecular solvation. *J Comput Chem*. 2015; 36:1047–1059. [PubMed: 25766844]
29. Zhou S, Cheng LT, Dzubiella J, Li B, McCammon JA. Variational implicit solvation with Poisson–Boltzmann theory. *J Chem Theory Comput*. 2014; 10:1454–1467. [PubMed: 24803864]

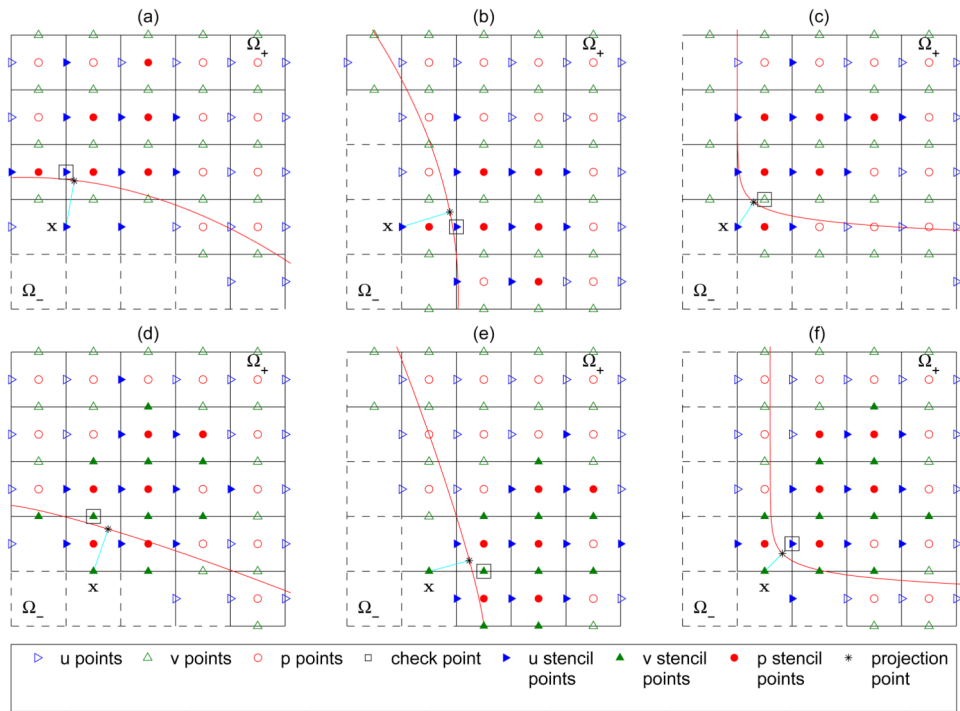


Figure A.1.
Different cases of combination of a ghost point and a check point.

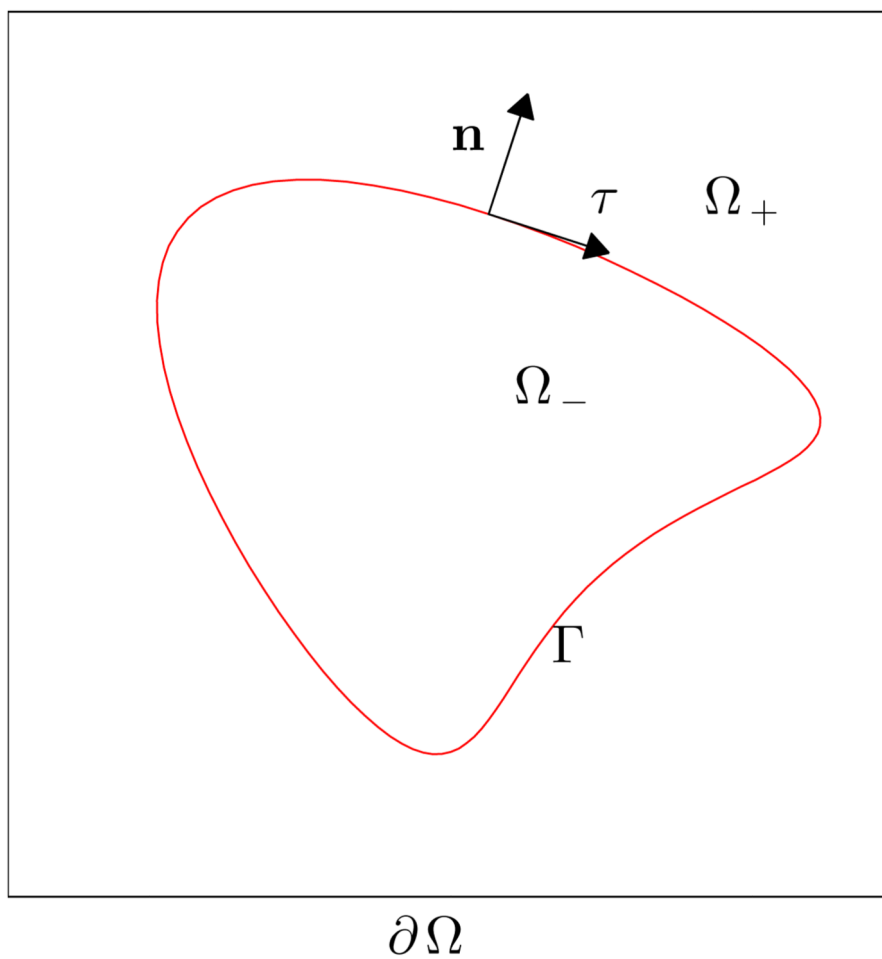


Figure 2.1. The geometry of a salvation system. The solute-solvent interface Γ separates the solute region Ω_- from the solvent region Ω_+ . The unit normal and unit tangent vectors at Γ are denoted by \mathbf{n} and $\boldsymbol{\tau}$ respectively.

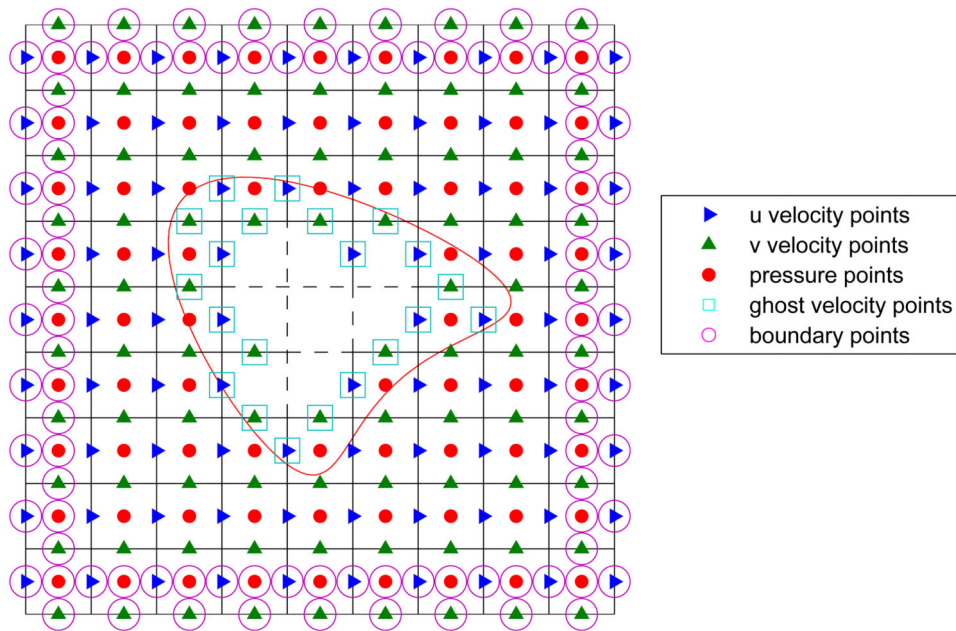


Figure 3.1.
A schematic MAC grid.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

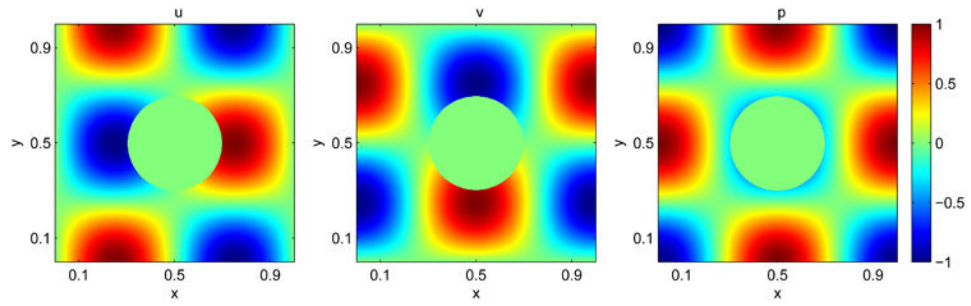


Figure 4.1. Numerical solution to the Stokes equation (2.2). From left to right: u , v component of fluid velocity and the fluid pressure p .

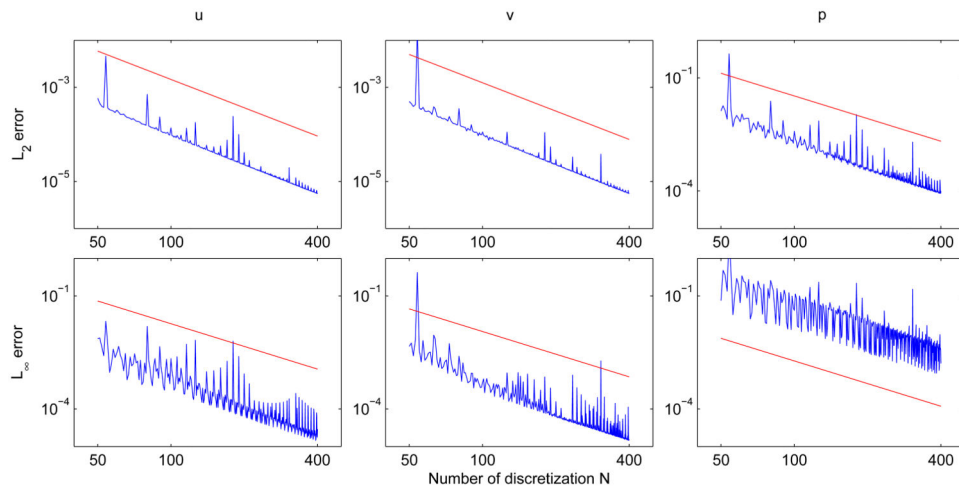


Figure 4.2.

Log-log plots of errors vs. the number of discretization (grid points). Top row from left to right: The L^2 -norm of the error for u , v and p , with the red straight line having slope -2 . Bottom row from left to right: The L^∞ -norm of the error for u , v and p , with the red straight line having slope -2 . The presence of spikes is due to the insufficient grid resolution that leads to sudden increase of the conditional number.

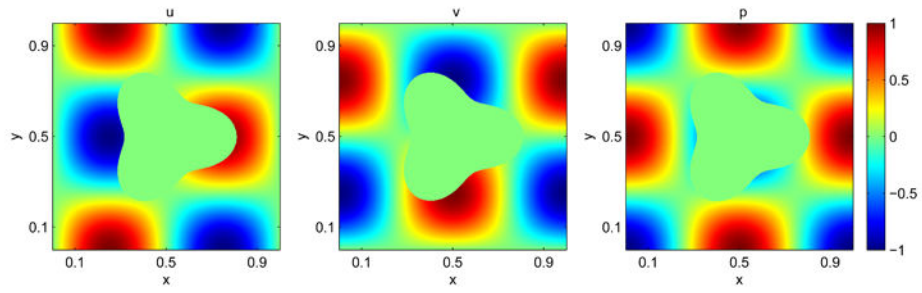


Figure 4.3. Numerical solution to the Stokes equation (2.2) described in Subsection 4.2. From left to right: u , v component of fluid velocity, and the fluid pressure p .

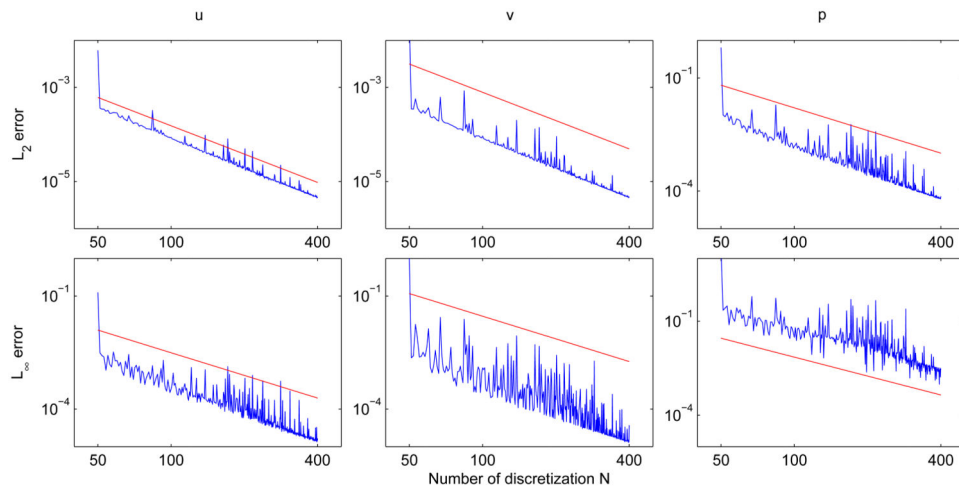


Figure 4.4.

Top row from left to right: The L^2 -norm of the error for u , v and p , with the red straight line being of slope -2 . Bottom row from left to right: The L^∞ -norm of the error for u , v and p , with the red straight line being of slope -2 .

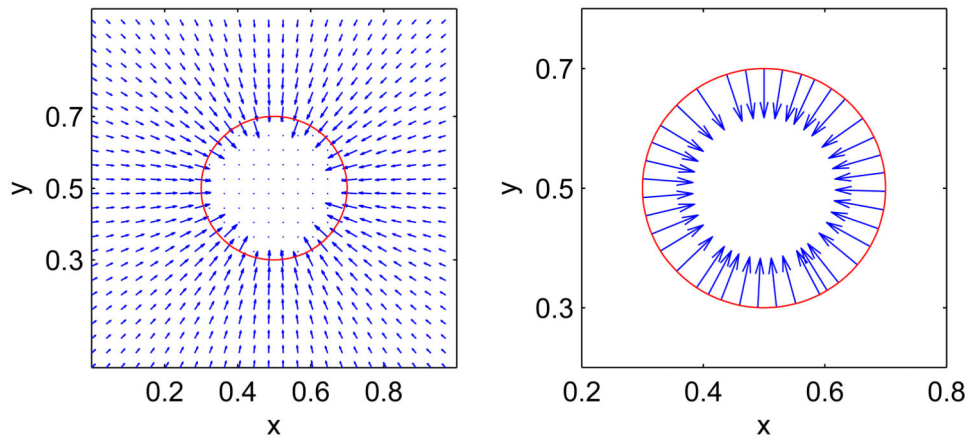


Figure 4.5. Numerical solution to the Stokes equation (2.2) and the corresponding flux. Left: The quiver velocity field \mathbf{u} in the whole domain Ω . Right: A zoom-in velocity field \mathbf{u} along Γ .

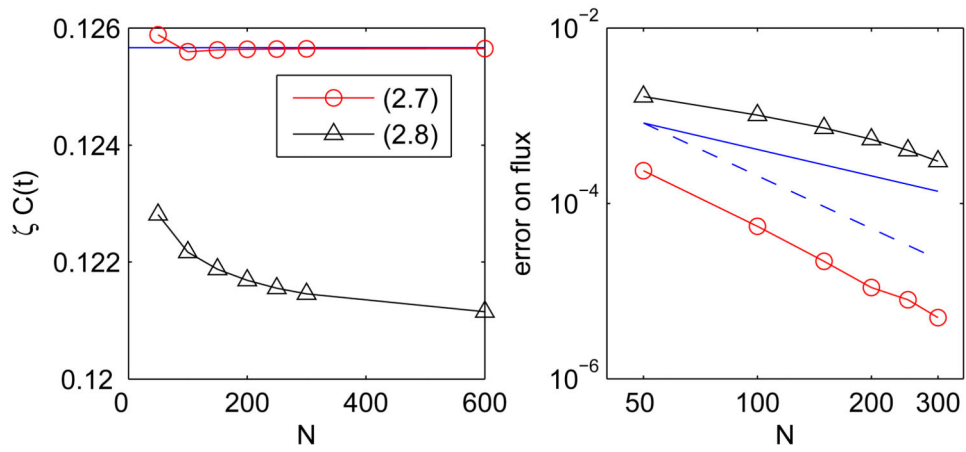


Figure 4.6.

Left: The total fluid flux $\zeta C(t)$ versus N . The blue horizontal line is 0.04π . The analytic value of $\zeta C(t)$: the black and red lines and dots correspond to the solutions with profile (2.7) and profile (2.8), respectively. Right: A log-log plot of the error on the flux versus N . As a comparison, the solid blue line has slope -1 and the dashed blue line has slope -2 .

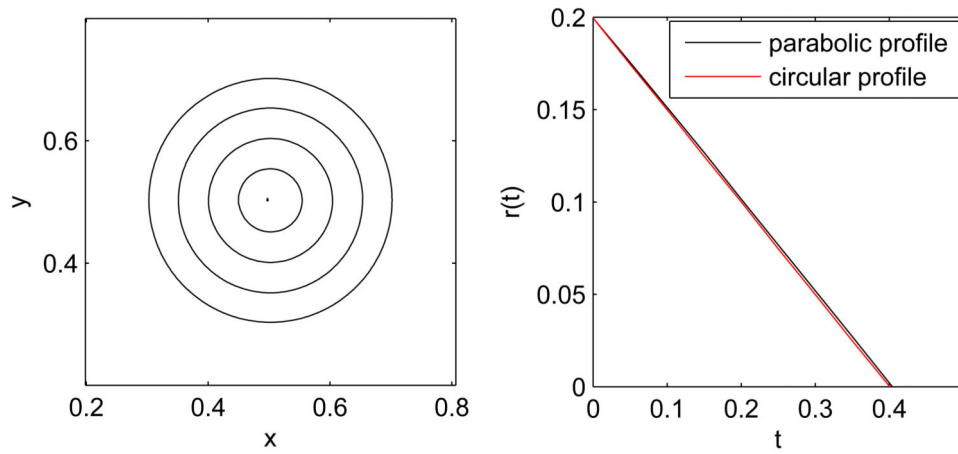


Figure 4.7.

Left: The solution to the curvature flow problem with parabolic profile (2.7), at time $t = 0, 0.1, 0.2, 0.3, 0.4$, from outside to inside, respectively. Right: The numerical radius $r(t)$ versus t with parabolic profile (2.7) (black curve) and with circular profile (2.8) (red line). The critical time is $t_c = 0.4042$ for the black line, and $t_c = 0.4004$ for the red line, respectively.

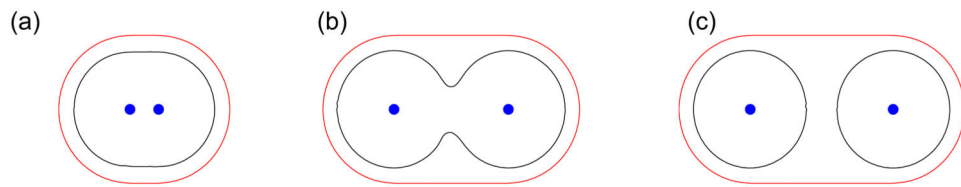


Figure 5.1.

The initial interface (red curve) and the numerically computed steady-state interface (black curve). The two blue dots are the positions of two particles. Left: $\Omega = (0, 1) \times (0, 1)$, $\mathbf{x}_1 = (0.45, 0.5)$, and $\mathbf{x}_2 = (0.55, 0.5)$. Middle: $\Omega = (0, 1) \times (0, 1)$, $\mathbf{x}_1 = (0.3, 0.5)$, and $\mathbf{x}_2 = (0.7, 0.5)$. Right: $\Omega = (0, 2) \times (0, 1)$, $\mathbf{x}_1 = (0.75, 0.5)$, and $\mathbf{x}_2 = (1.25, 0.5)$.

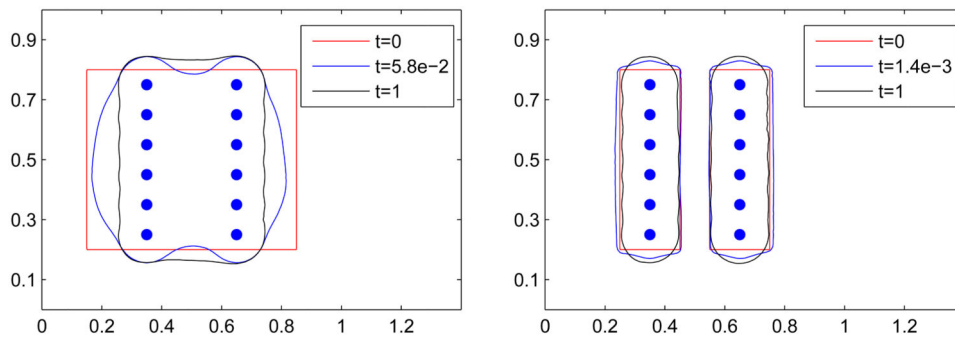


Figure 5.2. The numerical solutions of the interface Γ : an initial (red curve), an intermediate (blue curve), and a final, steady state (black curve) interfaces. The blue dots are the atom positions. Both plots are with $\gamma=5$ and $\sigma=0.1$. Left: The dry final state with loose initial state. Right: The wet final state with tight initial state.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript