Numerical Methods for Solvent Stokes Flow and Solute-Solvent Interfacial Dynamics of Charged Molecules

Hui Sun [*] Shenggao Zhou [†] Li-Tien Cheng [‡] Bo Li [§]

July 30, 2018

Abstract

We develop a computational model for the solvation of charged molecules in an aqueous solvent. Such solvent is modeled as an incompressible fluid, and its dynamics is described by the Stokes equations. All the surface tension force, van der Waals dispersive force, electrostatic force, hydrostatic pressure, and viscous force are balanced on the solute-solvent interface, giving rise to the traction boundary conditions on such an interface. We use the level-set method for the solute-solvent interface motion. To describe the hydrophobic interaction that is characterized by the volume change of the solute region, we design special numerical boundary conditions for the Stokes equations. We then reformulate and discretize the Stokes equations with a second-order finite difference scheme using a MAC grid. At virtual nodes near the solute-solvent interface, we use interpolation to discretize the boundary conditions and formulate the difference equations at such nodes. The resulting linear system is reduced by the Schur complement and then solved by the least-squares method with various techniques of acceleration. Numerical tests show that our method has a second-order convergence in the maximum norm for both the velocity and pressure up to the boundary. We apply our methods to some model molecular systems. With the comparison of our approach with a well established, static solvation theory and method, our dynamic model and numerical techniques accurately reproduce the solvation free energies, and capture the dry and wet molecular states. Our work provides a possibility of describing solvent fluctuations through the solvent fluid flow.

[*]Department of Mathematics and Statistics, California State University – Long Beach, CA 90840. Email: hui.sun@csulb.edu.

[†]Department of Mathematics and Mathematical Center for Interdisciplinary Research, Soochow University, Suzhou, Jiangsu 215000, China. Email: sgzhou@suda.edu.cn.

[‡]Department of Mathematics, University of California, San Diego, CA 92093. Email: lcheng@math.ucsd.edu.

[§]Department of Mathematics and Quantitative Biology Graduate Program, University of California, San Diego, CA 92093. Email: bli@math.ucsd.edu.

Keywords. Charged molecules, implicit-solvent models, Stokes flow, numerical boundary conditions, virtual nodes, interfacial dynamics, level-set method, dry and wet states.

1 Introduction

Aqueous solvent plays a critical role in the structure and dynamics of charged molecules, particularly biological molecules, such as proteins, DNA and RNA, and biological membranes [7, 10, 37]. Such a role arises from the solvent molecular network and structural characteristics as well as the solvent macroscopic fluid mechanical properties. In fact, 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, 24, 28, 34, 49, 55, 56, 58, 60]. Explicit-water molecular dynamics simulations have suggested the validity of the Rayleigh–Plesset equation in modeling nanoscale bubbles in water [20, 31, 42, 47]. Such equation is derived from the Navier–Stokes equations of fluid mechanics under the spherical symmetry [47]. Continuum hydrodynamic models have also been used to describe the phase separation in multi-component lipid bilayers [11, 12].

In this work, we develop a fluid mechanics model and computational methods for the solvent dynamics in the solvation of charged molecules, aiming particularly at a good understanding on effect of the solvent viscosity [58] to the molecular conformational changes and the dry-wet transition [7, 10]. We construct our model based on the Stokes equations for the solvent flow, as well as the recently developed variational implicit-solvent model (VISM), a static theory for stable equilibrium conformations and solvation free energies of charged molecules [21, 22] (cf. also [18, 61, 67] and [9]), which we now briefly review.

Let us consider one or a few charged molecules consisting of N atoms with fixed atomic positions $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and partial charges $Q_1, \ldots, Q_N$, respectively. We denote by $\Omega_{\rm m}$ (m stands for molecule) the solute molecular region that contains all the solute atoms, and by $\Omega_{\rm w} = \mathbb{R}^3 \backslash \Omega_{\rm m}$ (w stands for water) the solvent region. These two regions are separated by the solute-solvent interface, or dielectric boundary, Γ; cf. Figure 1.1. In VISM, one determines the stable equilibrium conformations and solvation free energies of an underlying charged molecular system by minimizing a solvation free-energy functional of all possible dielectric boundaries Γ. The functional is [18, 21, 22, 61, 67]

$$G[\Gamma] = p_0 \mathrm{Vol}(\Omega_{\rm m}) + \gamma_0 \int_\Gamma (1 - 2\tau H)\, dS + \rho_{\rm w} \int_{\Omega_{\rm w}} U_{\rm vdW} \mathrm{d}V + G_{\rm ele}[\Gamma]. \tag{1.1}$$

The first term in (1.1), in which $\mathrm{Vol}\,(\Omega_{\rm m})$ is the volume of $\Omega_{\rm m}$ and p_0 is the difference between the solvent and solute pressures on the boundary Γ, describes the reversible work needed to create the solute region $\Omega_{\rm m}$. The second term in (1.1) is the surface energy, where the surface energy density (surface tension) not only includes the constant surface tension γ_0 for a flat interface but also the curvature correction with H the mean curvature (i.e., the average of the two principal curvatures) and τ the correction constant, often called the Tolman length [59]. The third term in (1.1), in which $\rho_{\rm w}$ is the constant bulk density of solvent molecules, describes the solute-solvent short-range excluded volume effect and the

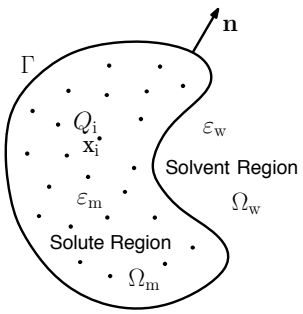

Figure 1.1: Schematic of molecular solvation with an implicit solvent. The dielectric boundary Γ separates the solute region $\Omega_{\rm m}$ and the solvent region $\Omega_{\rm w}$ that have the dielectric coefficients $\varepsilon_{\rm m}$ and $\varepsilon_{\rm w}$, respectively. All the solute atoms are located at $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and they carry the partial charges $Q_1, \ldots, Q_N$, respectively.

long-range dispersive interactions. The function $U_{\rm vdW}$ is the solute-solvent van der Waals (vdW) interaction potential. We take, as in molecular dynamics simulations,

$$U_{\rm vdW}(\mathbf{x}) = \sum_{i=1}^{N} U_{\rm LJ}^{(i)}\left(|\mathbf{x} - \mathbf{x}_i|\right),$$

where each $U_{\rm LJ}^{(i)}$ is a 12-6 Lennard-Jones (LJ) potential

$$U_{\rm LJ}^{(i)}(r) = 4\varepsilon_i \left[\left(\frac{\sigma_i}{r}\right)^{12} - \left(\frac{\sigma_i}{r}\right)^{6}\right]$$

with ε_i and σ_i the energy and length parameters. The last term $G_{\rm ele}[\Gamma]$ in (1.1) is the electrostatic part of the free energy, where Γ is the dielectric boundary. It is often determined by the Poisson–Boltzmann theory [2,13,19,38,53,65,67] or the Coulomb-field approximation [8,14,61].

The minimization of the VISM free-energy functional can be achieved computationally by the level-set method [45,46,51], in which the "normal velocity" is the normal component of the boundary force, defined to be the negative first variation, $-\delta_\Gamma G[\Gamma] : \Gamma \to \mathbb{R}$, of the functional $G[\Gamma]$ with respect to the variation of boundary Γ. We have [15,61,67]

$$\delta_\Gamma G[\Gamma] = p_0 + 2\gamma_0(H - \tau K) - \rho_{\rm w} U_{\rm vdW} + \delta_\Gamma G_{\rm ele}[\Gamma], \tag{1.2}$$

where K is the Gaussian curvature. If the electrostatic free energy $G_{\rm ele}[\Gamma]$ is given in the Poisson–Boltzmann formulation, then the variation $\delta_\Gamma G_{\rm ele}[\Gamma]$ can be obtained by shape derivative calculations [39,67]. Here, we shall use the Coulomb-field approximation of the electrostatic energy, assuming a lower ionic concentration. In this case, the electrostatic energy is given by

$$G_{\rm ele}[\Gamma] = \frac{1}{32\pi^2\varepsilon_0}\left(\frac{1}{\varepsilon_{\rm m}} - \frac{1}{\varepsilon_{\rm w}}\right)\int_{\Omega_{\rm w}} \left|\sum_{i=1}^{N} \frac{Q(\mathbf{x} - \mathbf{x}_i)}{|\mathbf{x} - \mathbf{x}_i|^3}\right|^2 d\mathbf{x}, \tag{1.3}$$

and its variation is given by

$$\delta_\Gamma G_{\rm ele}[\Gamma](\mathbf{x}) = \frac{1}{32\pi^2\varepsilon_0}\left(\frac{1}{\varepsilon_{\rm m}} - \frac{1}{\varepsilon_{\rm w}}\right)\left|\sum_{i=1}^{N}\frac{Q(\mathbf{x}-\mathbf{x}_i)}{|\mathbf{x}-\mathbf{x}_i|^3}\right|^2 \qquad \text{if } \mathbf{x}\in\Gamma, \tag{1.4}$$

where ε_0 is the vacuum permittivity, and $\varepsilon_{\rm m}$ and $\varepsilon_{\rm w}$ are the dielectric coefficients (i.e., relative permittivities) of the solute and solvent, respectively; cf. Figure 1.1. The rationale behind our choice of the Coulomb-field approximation is two-fold. First, it is an efficient model as no equations need to be solved. Second, such an approximation includes the effect of dielectric inhomogeneity, and captures the dry and wet states in molecular hydration, as demonstrated in our earlier work [61].

Extensive work on the level-set VISM has shown that this approach can capture multiple hydration states and subtle electrostatic effect, and provide reasonably good free-energy estimates [15, 17, 18, 26, 27, 48, 52, 61, 66–69].

In contrast to our VISM, we call our solvent fluid mechanics model the dynamic implicit-solvent model (DISM). In this model, all the solute region $\Omega_{\rm m}$, solvent region $\Omega_{\rm w}$, and the dielectric boundary Γ depend on time t. This dynamical model consists of the following main elements:

(1) The motion of the dielectric boundary $\Gamma(t)$ is defined by $\dot{\mathbf{x}} = \mathbf{u}(\mathbf{x})$ for any point $\mathbf{x} = \mathbf{x}(t) \in \Gamma(t)$, where a dot denotes the time derivative and $\mathbf{u} = (u, v, w)$ is the velocity field of solvent fluid. This means that the normal velocity of the moving boundary $\Gamma(t)$ is given by

$$V_n = \mathbf{u}\cdot\mathbf{n} \qquad \text{on } \Gamma(t), \tag{1.5}$$

where $\mathbf{n}$ is the unit normal at the boundary $\Gamma(t)$ pointing from the solute region $\Omega_{\rm m}(t)$ to the solvent region $\Omega_{\rm w}(t)$; cf. Figure 1.1;

(2) The solvent fluid is assumed to be incompressible, and is described by the Stokes equations

$$\nabla\cdot\mathbf{u} = 0 \quad \text{and} \quad \mu_{\rm w}\Delta\mathbf{u} - \nabla p_{\rm w} + \mathbf{g} = \mathbf{0} \quad \text{in } \Omega_{\rm w}(t), \tag{1.6}$$

where $\mu_{\rm w}$ and $p_{\rm w}$ are the dynamic viscosity and pressure of the solvent fluid, and $\mathbf{g}$ is the source term representing the density of a body force experienced by the solvent fluid. The velocity and pressure satisfy the far-field conditions:

$$\lim_{|\mathbf{x}|\to\infty}\mathbf{u}(\mathbf{x},t) = \mathbf{0} \quad \text{and} \quad \lim_{|\mathbf{x}|\to\infty}p_{\rm w}(\mathbf{x},t) = p_\infty, \tag{1.7}$$

where p_∞ is the constant bulk solvent pressure;

(3) The pressure $p_{\rm m} = p_{\rm m}(t)$ of solute molecules is assumed to be spatially constant and described simply by the ideal-gas law

$$p_{\rm m}(t)\mathrm{Vol}(\Omega_{\rm m}(t)) = Nk_{\rm B}T, \tag{1.8}$$

where $\mathrm{Vol}(\Omega_{\rm m}(t))$ is the volume of solute region. If the solute molecular region has multiple subregions (i.e., connected components), then this equation holds true for each

of the subregions with N replaced by the number of solute atoms in the subregion; (We note that the error arising from the efficient ideal-gas approximation is insignificant, as the solute volume $\mathrm{Vol}\,(\Omega_{\mathrm{m}})$ in the solvation free energy (1.1) is usually small.)

(4) At the solute-solvent interface $\Gamma(t)$, all the surface tension force, solute-solvent vdW interaction force, electrostatic force, hydrostatic pressure, and viscous force are balanced:

$$2\mu_{\mathrm{w}} D(\mathbf{u})\mathbf{n} + \delta_\Gamma G[\Gamma]\mathbf{n} = \mathbf{0} \qquad \text{on } \Gamma(t), \tag{1.9}$$

where $D(\mathbf{u}) = (\nabla\mathbf{u} + (\nabla\mathbf{u})^T)/2$ is the strain rate tensor and $\delta_\Gamma G[\Gamma]$ is given in (1.2) with $p_0 = p_{\mathrm{w}} - p_{\mathrm{m}}$ which is now spatially and temporally dependent.

The main objective of our current work is to develop robust numerical methods to solve the equations (1.5)–(1.9) in the three-dimensional setting. We simulate the solvation dynamics on a bounded region Ω that is sufficiently large to contain all the possible solute regions $\Omega_{\mathrm{m}}(t)$ for all time t. Let us still denote by $\Omega_{\mathrm{w}}(t)$ the solvent region, which is now defined as $\Omega_{\mathrm{w}}(t) = \Omega \setminus \Omega_{\mathrm{m}}(t)$. We shall design special numerical boundary conditions on the boundary $\partial\Omega$ (cf. Section 3) in replace of the far-field conditions (1.7). Our main contributions and results include the following:

(1) We use the level-set method to track the solute-solvent interfacial motion;

(2) To allow the change of volume of the solute molecular region, we design special numerical boundary conditions that allow the solvent fluid to freely flow through the boundary of the domain Ω with an arbitrary geometry. We call such boundary conditions free-flow boundary conditions;

(3) We reformulate the Stokes equations for solvent fluid using the pressure Poisson equation [25,30,32,33,54]. We solve the reformulated equations, together with our free-flow numerical boundary conditions and the interface conditions (1.9) on $\Gamma(t)$, using the MAC scheme [29] for regular grid points and a virtual node method [3, 44, 50] for the discretization at grids around the interface;

(4) We use the Schur complement to reduce our resulting linear system of algebraic equations, and then solve such a system using the least-squares method. We apply various kinds of techniques to speed up our three-dimensional computations. These include the generalized minimal residual method (GMRES), algebraic multigrid method (AMG), the additive Schwarz method (ASM), and a specially designed local relaxation preconditioning. We implement our numerical methods using the PETSc package [4–6] and the Hypre package [23];

(5) We verify that our method is second-order accurate and demonstrate that our model, particularly our free-flow numerical boundary conditions, and numerical methods are capable of capturing the viscous effect to the dynamics of dry-wet transitions in molecular solvation with complicated topological changes.

Several recent studies [40, 41, 57, 62, 63] have initiated the development of a dynamic solvent modeling approach and the related numerical methods for the solvent fluid flow. Luo *et al.* [62] propose to combine the fluid mechanics description of solvent with the molecular dynamics simulation of solute molecules. In [41, 62], Luo *et al.* develop a second-order augmented semi-implicit immersed-interface method for the solvent fluid motion with direc-

tional free-flow boundary conditions. The recent work by Luo *et al.* [63] reports results of their molecular dynamics fluid dynamics simulations of some nonpolar molecular systems. Due to the unrealistic parameters used, what they have simulated is essentially a relaxation process. Li *et al.* [40] study the linear stability of a cylindrical solute-solvent interface, and conclude that the viscosity can modify the critical wavelength of such stability. Sun *et al.* [57] develop a numerical method for the dynamic solvent model and captured the dry and wet states in a two-dimensional setting. Here, our dynamic implicit-solvent model (DISM) is developed systematically to include various kinds of effects, particularly the electrostatics which is crucial in biomolecular modeling. We construct general free-flow boundary conditions that can work for arbitrary geometries and allow the solvent fluid to flow in and out in all parts of the boundary of a computational domain. Such generality is important for studying biomolecular conformations which are often complex with different orientations. Our numerical methods are of high accuracy, even up to the solute-solvent interface. Such robust methods have allowed the track of the complex solute-solvent interface motion, and in particular, the capture of the dry and wet states and the transition from one to the other by the solvent fluid flow in three-dimensional space.

The rest of the paper is organized as follows: In Section 2, we describe the level-set method for moving the solute-solvent interface and the algorithm of our computation. In Section 3, we reformulate our system of equations and design the free-flow boundary conditions. In Section 4, we introduce nodes (grid points) of different types, and discretize the equations and boundary conditions. We then solve the resulting linear system, and study the convergence of our numerical methods. Some of the details are given in Appendix. In Section 5, we apply our model and numerical methods to a spherical bubble of nanometer size and a two-plate system. Finally, in Section 6, we draw conclusions and discuss future work.

2 The Level-Set Method and Algorithm

We apply the level-set method that we have developed for minimizing the VISM functional (1.1) to track the motion of solute-solvent interface $\Gamma(t)$ with the normal velocity V_n given by (1.5) [15,16,18,61,67]. Let $\phi = \phi(\mathbf{x}, t)$ denote a level-set function for the interface $\Gamma(t)$ at time t, i.e., $\Gamma(t) = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) = 0\}$, with the convention that $\Omega_{\rm m}(t) = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) < 0\}$ and $\Omega_{\rm w}(t) = \{\mathbf{x} \in \Omega : \phi(\mathbf{x}, t) > 0\}$. This level-set function is determined by the level-set equation

$$\partial_t \phi + V_n |\nabla \phi| = 0, \tag{2.1}$$

together with the boundary coundition $\partial_n \phi = 0$ on $\partial\Omega$, the boundary of Ω, and an initial level-set function. Here, the normal velocity V_n is defined in (1.5). It is extended constant along the normal direction to $\Gamma(t)$ to the entire computational region Ω with the fast marching method. We use the forward Euler method and the fifth-order WENO scheme to iteratively solve the level-set equation. We re-initialize the level-set function in each time step.

As usual in the level-set method [45, 46, 51], the mean curvature and Gaussian curvature

of $\Gamma(t)$ can be calculated by

$$H = \frac{1}{2}\nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) \quad \text{and} \quad K = \frac{\nabla\phi}{|\nabla\phi|} \cdot \text{adj}\left(\frac{\nabla^2\phi}{|\nabla\phi|}\right)\frac{\nabla\phi}{|\nabla\phi|} \qquad \text{on } \Gamma(t),$$

where adj$(\cdot)$ is the adjoint matrix. The surface and volume integrals of a function $\psi(\mathbf{x})$ can be expressed, respectively, as

$$\int_{\Gamma(t)} \psi(\mathbf{x})\, dS_{\mathbf{x}} = \int_{\Omega} \delta(\phi(\mathbf{x},t))\psi(\mathbf{x})\mathrm{d}\mathbf{x},$$
$$\int_{\Omega_{\mathrm{m}}(t)} \psi(\mathbf{x})\mathrm{d}V_{\mathbf{x}} = \int_{\Omega} \mathrm{Heav}(-\phi(t,\mathbf{x}))\psi(\mathbf{x})\mathrm{d}\mathbf{x},$$

where δ and Heav are the Dirac delta function (suitably regularized) and Heaviside function, respectively.

Our algorithm of simulation consists of the following main steps:

(1) Initialize the interface Γ (by defining an initial level-set function). Set $t = 0$;
(2) Calculate the unit normal, mean curvature, and Gaussian curvature on $\Gamma(t)$, using the level-set function;
(3) Calculate the volume of $\Omega_{\mathrm{m}}(t)$, and calculate the pressure $p_{\mathrm{m}}(t)$ by (1.8);
(4) Solve the fluid equations (1.6) together with the boundary conditions (1.9) on $\Gamma(t)$ and those on the boundary $\partial\Omega$ (defined in the next section);
(5) Calculate the normal velocity V_n by (1.5) on $\Gamma(t)$ and extend it to the entire region Ω;
(6) Solve the level-set equation (2.1) for one time step Δt;
(7) Set $t := t + \Delta t$ and go to Step (2).

3 Free-Flow Boundary Conditions and Reformulation of Stokes' Equations

In this section, we first discuss briefly why some of the commonly used boundary conditions for the velocity field will not allow the change of the volume of solute. Based on such discussion, we then introduce our free-flow numerical boundary conditions. Finally, we reformulate the system of fluid equations and the corresponding boundary conditions into an equivalent and coupled system of Poisson's equations and the boundary conditions. This section highlights one of our main contributions in this work.

We recall that we have denoted by $\mathbf{n}$ the unit exterior normal of the boundary $\partial\Omega$. Notice that $\partial\Omega_{\mathrm{w}}(t) = \Gamma(t) \cup \partial\Omega$ and that the normal $\mathbf{n}$ on $\Gamma(t)$ points from $\Omega_{\mathrm{m}}(t)$ to $\Omega_{\mathrm{w}}(t)$. By the Reynolds transport theorem [36], the divergence theorem, and the incompressibility of the solvent fluid, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{Vol}(\Omega_{\mathrm{m}}(t)) = \int_{\Gamma(t)} \mathbf{u}\cdot\mathbf{n}\mathrm{d}S = -\int_{\partial\Omega_{\mathrm{w}}(t)} \mathbf{u}\cdot\mathbf{n}\, dS + \int_{\partial\Omega} \mathbf{u}\cdot\mathbf{n}\, dS$$

$$= -\int_{\Omega_{\rm w}(t)} \nabla \cdot \mathbf{u}\, d\mathbf{x} + \int_{\partial\Omega} \mathbf{u}\cdot\mathbf{n}\, dS = \int_{\partial\Omega} \mathbf{u}\cdot\mathbf{n}\, {\rm d}S. \tag{3.1}$$

Consequently, if we use the impermeability condition ($\mathbf{u}\cdot\mathbf{n} = 0$), or periodic boundary conditions (for a rectangular parallelepiped region Ω), then the volume of the solute molecular region $\Omega_{\rm m}(t)$ will be a constant, independent of time t. If we fix the boundary value of $\mathbf{u}\cdot\mathbf{n}$, in particularly, if we use the Dirichlet boundary condition, then the rate of change of the volume of solute region will be fixed, which is not true in general in the hydrophobic interaction of a molecular system. Therefore, it is natural to leave the values $\mathbf{u}\cdot\mathbf{n}$ free but prescribe the tangential component $(I - \mathbf{n}\otimes\mathbf{n})\mathbf{u}$ on the boundary $\partial\Omega$. Specifically, we impose our numerical boundary conditions to be

$$(I - \mathbf{n}\otimes\mathbf{n})\mathbf{u} = \mathbf{u}_0 \qquad \text{and} \qquad p_{\rm w} = p_{\rm w0} \qquad \text{on } \partial\Omega, \tag{3.2}$$

where I is the identity matrix, and $\mathbf{u}_0 = (u_0, v_0, w_0)$ and $p_{\rm w0}$ are given functions defined on $\partial\Omega$ with $\mathbf{u}_0$ the tangential velocity on the boundary $\partial\Omega$. .

Notice that we are replacing $\Omega_{\rm w}(t) = \mathbb{R}^3 \setminus \Omega_{\rm m}(t)$ in (1.6) and the far-field conditions (1.7) by $\Omega_{\rm w}(t) = \Omega \setminus \Omega_{\rm m}(t)$ and the boundary conditions (3.2), respectively. Therefore, the solutions $\mathbf{u}$ and p for the two different systems may not be the same everywhere in Ω. However, we expect that, if the region Ω is large enough, the effect of these solutions to the motion of solute-solvent interface will not be significantly different. We shall verify this when we test our model and methods; cf. Section 5. Notice also that the first vector equation in (3.2) is equivalent to two scalar equations on the tangential components of the velocity field along the boundary $\partial\Omega$. These, together with the pressure boundary condition in (3.2), provide the correct number of equations in the boundary conditions for the Stokes equations for an incompressible fluid on a bounded region. In general, the pressure boundary condition is not needed for such equations on a bounded region. But here we are given the far-field pressure (1.7); and imposing the pressure boundary condition is practically necessary from the modeling purpose. Moreover, as we will see next, our systems of equations and boundary conditions will be reformulated into new ones where we have the pressure Poisson's equation for which the pressure boundary condition is needed to determine a unique solution.

To develop efficient methods for solving the underlying Stokes equations for incompressible solvent fluid flow with a rather complicated geometry in three-dimensional space, we now reformulate our system of equations (1.6), (1.9), and (3.2). First, following [54], we convert these equations and boundary conditions into

$$\mu_{\rm w}\Delta\mathbf{u} - \nabla p_{\rm w} + \mathbf{g} = \mathbf{0} \qquad \text{in } \Omega_{\rm w}(t), \tag{3.3}$$

$$\Delta p_{\rm w} = \nabla\cdot\mathbf{g} \qquad \text{in } \Omega_{\rm w}(t), \tag{3.4}$$

$$2\mu_{\rm w} D(\mathbf{u})\mathbf{n} + \delta_\Gamma G[\Gamma]\mathbf{n} = \mathbf{0} \quad \text{and} \quad \nabla\cdot\mathbf{u} = 0 \qquad \text{on } \Gamma(t), \tag{3.5}$$

$$(I - \mathbf{n}\otimes\mathbf{n})\mathbf{u} = \mathbf{u}_0, \quad p_{\rm w} = p_{\rm w0}, \quad \text{and} \quad \nabla\cdot\mathbf{u} = 0 \qquad \text{on } \partial\Omega. \tag{3.6}$$

The Poisson's equation (3.4) is derived by applying the divergence operator to the Stokes equations in (1.6) and using the divergence-free condition in (1.6). The divergence-free equation on the interface $\Gamma(t)$ and boundary $\partial\Omega$ result from the same equation in $\Omega_{\rm w}(t)$ in

(1.6) with continuous extension to the boundary of $\Omega_{\rm w}(t)$. Conversely, (3.3) and (3.4) imply that $\Delta(\nabla\cdot\mathbf{u})=0$ in $\Omega_{\rm w}(t)$. The homogeneous Dirichlet boundary condition for $\nabla\cdot\mathbf{u}$ in (3.5) and (3.6) then imply that $\nabla\cdot\mathbf{u}=0$ throughout the computational domain $\Omega_{\rm w}(t)$.

Let $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ denote two orthogonal unit tangent vectors at the boundary of $\Omega_{\rm w}(t)$, which is the union of $\partial\Omega$ and $\Gamma(t)$. We assume the three vectors $\mathbf{n}$, $\boldsymbol{\tau}_1$, and $\boldsymbol{\tau}_2$ form a right-handed system, i.e., $\mathbf{n}\times\boldsymbol{\tau}_1\cdot\boldsymbol{\tau}_2=1$. We can now finally reformulate (3.3)–(3.6) into two systems of Poisson's equations for the velocity field and pressure, respectively, together with the corresponding boundary conditions on the boundary $\partial\Omega_{\rm w}(t)=\Gamma(t)\cup\partial\Omega$. The system for the velocity Poisson equation is

$$\mu_{\rm w}\Delta\mathbf{u}=\nabla p_{\rm w}-\mathbf{g}\quad\text{ in }\Omega_{\rm w}(t), \tag{3.7}$$

$$2\mu_{\rm w}\boldsymbol{\tau}_i\cdot D(\mathbf{u})\mathbf{n}=0\quad\text{for } i=1,2,\quad\text{and}\quad\nabla\cdot\mathbf{u}=0\quad\text{on }\Gamma(t), \tag{3.8}$$

$$\mathbf{u}\cdot\boldsymbol{\tau}_1=b_1,\quad\mathbf{u}\cdot\boldsymbol{\tau}_2=b_2,\quad\text{and}\quad\nabla\cdot\mathbf{u}=0\quad\text{on }\partial\Omega, \tag{3.9}$$

where $b_i=\mathbf{u}_0\cdot\boldsymbol{\tau}_i$ $(i=1,2)$. The system for the pressure Poisson's equation is

$$\Delta p_{\rm w}=\nabla\cdot\mathbf{g}\quad\text{ in }\Omega_{\rm w}(t), \tag{3.10}$$

$$p_{\rm w}=2\mu_{\rm w}\mathbf{n}\cdot D(\mathbf{u})\mathbf{n}+p_{\rm m}-2\gamma_0(H-\tau K)+\rho_{\rm w}U_{\rm vdW}-\delta_\Gamma G_{\rm ele}[\Gamma]\quad\text{on }\Gamma(t), \tag{3.11}$$

$$p_{\rm w}=p_{\rm w0}\quad\text{on }\partial\Omega, \tag{3.12}$$

where $p_{\rm m}=p_{\rm m}(t)$ is given in (1.8), and $-\delta_\Gamma G_{\rm ele}[\Gamma]$ is the dielectric boundary force. We shall use the Coulomb-field approximation of the electrostatic interaction energy, and therefore $\delta_\Gamma G_{\rm ele}[\Gamma]$ is given by (1.4). Note that (3.3) and (3.4) are exactly the same as (3.7) and (3.10), respectively, and that (3.6) is the same as (3.9) and (3.12) with $b_i=\mathbf{u}_0\cdot\boldsymbol{\tau}_i$ $(i=1,2)$. The divergence-free equation in (3.5) is the same as that in (3.8). Finally, since $p_0=p_{\rm w}-p_{\rm m}$, the force-balance equaiton in (3.5) is equivalent to the equations for the tangential components in (3.8) and that for the normal component in (3.11) (cf. (1.2)). Therefore, the system (3.3)–(3.6) is equivalent to the system (3.7)–(3.12).

4 Numerical Methods for Solvent Fluid Flow Equations

We fix time t and solve the two Poisson systems (3.7)–(3.12). For convenience, we write in this section $\Omega_{\rm w}$, $\Omega_{\rm m}$, and Γ, instead of $\Omega_{\rm w}(t)$, $\Omega_{\rm m}(t)$, and $\Gamma(t)$. We also write p instead of $p_{\rm w}$.

4.1 Different Types of Nodes and Discretization Scheme

We choose $\Omega=(-l_x,l_x)\times(-l_y,l_y)\times(-l_z,l_z)$, with l_x, l_y, and l_z positive numbers, and cover it with a finite difference grid with the number of grid points n_x, n_y, and n_z in the three directions. The grid sizes are $h_x=2l_x/n_x$, $h_y=2l_y/n_y$, and $h_z=2l_x/n_z$, respectively. We shall use the MAC grid for the fluid velocity and pressure. This means that the pressure values are stored at the centers of the grid cells, while the three velocity components are stored at the centers of the faces of the cubic grid cells.

Given a grid cell $(ih_x,(i+1)h_x)\times(jh_y,(j+1)h_y)\times(kh_z,(k+1)h_z)$, its center is $\mathbf{x}_{i,j,k}=((i+1/2)h_x,(j+1/2)h_y,(k+1/2)h_z)$, and its face centers are $\mathbf{x}_{i\pm1/2,j,k}$, $\mathbf{x}_{i,j\pm1/2,k}$, and $\mathbf{x}_{i,j,k\pm1/2}$, respectively. For $0\le i\le n_x-1$, $0\le j\le n_y-1$, and $0\le k\le n_z-1$, we call $\mathbf{x}_{i,j,k}$ a p-point, $\mathbf{x}_{i-1/2,j,k}$ a u-point, $\mathbf{x}_{i,j-1/2,k}$ a v-point, and $\mathbf{x}_{i,j,k-1/2}$ a w-point. (Recall that u, v, and w are the components of the velocity field.) Any u, v, w, or p point is called a fluid point. For convenience, we denote $\mathcal{P}_p$, $\mathcal{P}_u$, $\mathcal{P}_v$, $\mathcal{P}_w$, and $\mathcal{P}_f$, the set of p, u, v, w, and fluid points, respectively. We also denote

$$
\begin{aligned}
&d(\mathbf{x},\tilde{\mathbf{x}})=\sqrt{\frac{(x-\tilde{x})^2}{h_x^2}+\frac{(y-\tilde{y})^2}{h_y^2}+\frac{(z-\tilde{z})^2}{h_z^2}} \qquad \forall\ \mathbf{x}=(x,y,z),\ \tilde{\mathbf{x}}=(\tilde{x},\tilde{y},\tilde{z})\in\mathcal{P}_f,\\
&\mathcal{N}(\mathbf{x}_p)=\{\mathbf{x}\in\mathcal{P}_f : d(\mathbf{x}_p,\mathbf{x})\le 1/2\},\\
&\mathcal{N}(\mathbf{x}_\alpha):=\{\mathbf{x}\in\mathcal{P}_f : d(\mathbf{x}_\alpha,\mathbf{x})\le 1\} \qquad \forall\alpha\in\{u,v,w\}.
\end{aligned}
$$

We now define boundary nodes, virtual nodes, regular nodes, and void nodes. A p-point $\mathbf{x}_{i,j,k}\in\mathcal{P}_p$ is called

- a *boundary node* if $i\in\{0,n_x-1\}$ or $j\in\{0,n_y-1\}$ or $k\in\{0,n_k-1\}$;
- a *virtual node* if $\mathcal{N}(\mathbf{x}_{i,j,k})\cap\Omega_{\rm w}\neq\emptyset$ and $\mathcal{N}(\mathbf{x}_{i,j,k})\cap\Omega_{\rm m}\neq\emptyset$;
- a *regular node* if it is in $\Omega_{\rm w}$ and it is neither a boundary node nor a virtual node; and
- a *void node* if it is in $\Omega_{\rm m}$ and it is not a virtual node.

A u-point $\mathbf{x}_{i-1/2,j,k}\in\mathcal{P}_u$ is called

- a *boundary node* if $i\in\{0,n_x\}$ or $j\in\{0,n_y-1\}$ or $k\in\{0,n_k-1\}$;
- a *virtual node* if it is in $\Omega_{\rm m}\cup\Gamma$ and $\mathcal{N}(\mathbf{x}_{i-1/2,j,k})\cap\Omega_{\rm w}\neq\emptyset$;
- a *regular node* if it is in $\Omega_{\rm w}$ and $\mathbf{x}_{i-1/2,j,k}$ is not a boundary node; and
- a *void node* if it is in $\Omega_{\rm m}$ and it is not a virtual node.

The definition that a v or w-point is a boundary node, virtual node, regular node, or void node is similar.

We now discretize our underlying equations and boundary conditions to obtain finite difference equations for the approximations of the velocity and pressure at all the regular, boundary, and virtual nodes. We shall use the standard notation $D^x_\pm(\cdot)_{i,j,k}$, $D^y_\pm(\cdot)_{i,j,k}$, and $D^z_\pm(\cdot)_{i,j,k}$, where $(\cdot)_{\rm indices}=(\cdot)(\mathbf{x}_{\rm indices})$. For instance, $D^x_+(\cdot)_{i,j,k}=[(\cdot)_{i+1,j,k}-(\cdot)_{i,j,k}]/h_x$ and $D^x_-(\cdot)_{i,j,k}=[(\cdot)_{i,j,k}-(\cdot)_{i-1,j,k}]/h_x$. We apply the second-order finite difference scheme to discretize Poisson's equations (3.7) and (3.10) at regular nodes to get

$$\mu_{\rm w}\left(D^x_+D^x_-+D^y_+D^y_-+D^z_+D^z_-\right)u_{i-1/2,j,k}+D^x_-p_{i,j,k}+g^{(1)}_{i-1/2,j,k}=0, \tag{4.1}$$

$$\mu_{\rm w}\left(D^x_+D^x_-+D^y_+D^y_-+D^z_+D^z_-\right)v_{i,j-1/2,k}+D^y_-p_{i,j,k}+g^{(2)}_{i,j-1/2,k}=0, \tag{4.2}$$

$$\mu_{\rm w}\left(D^x_+D^x_-+D^y_+D^y_-+D^z_+D^z_-\right)w_{i,j,k-1/2}+D^z_-p_{i,j,k}+g^{(3)}_{i,j,k-1/2}=0, \tag{4.3}$$

$$\left(D^x_+D^x_-+D^y_+D^y_-+D^z_+D^z_-\right)p_{i,j,k}+(\nabla\cdot\mathbf{g})_{i,j,k}=0, \tag{4.4}$$

where $g^{(i)}$ $(i=1,2,3)$ are the components of $\mathbf{g}$.

We now discretize the boundary conditions (3.9) and (3.12) on $\partial\Omega$ to define the finite difference equations at boundary nodes. Consider the part of the boundary $\partial\Omega$ with $x=l_x$

corresponding to $i = n_x$. On this part, we have $\mathbf{n} = (1, 0, 0)$, $\boldsymbol{\tau}_1 = (0, 1, 0)$, and $\boldsymbol{\tau}_2 = (0, 0, 1)$. Hence, the boundary conditions (3.9) on this part are $v = b_1$, $w = b_2$, and $\partial_x u = -\partial_y b_1 - \partial_z b_2$. These can be discretized as

$$\begin{aligned}
&v_{n_x-1,j-1/2,k} = b_1(\mathbf{x}_{n_x-1,j-1/2,k}), \qquad j = 1, \ldots, n_y - 1,\ k = 0, \ldots, n_z - 1,\\
&w_{n_x-1,j,k-1/2} = b_2(\mathbf{x}_{n_x-1,j,k-1/2}), \qquad j = 0, \ldots, n_y - 1,\ k = 1, \ldots, n_z - 1,\\
&D^x_- u_{n_x-1/2,j,k} = \frac{b_1(\mathbf{x}_{n_x-1,j-1/2,k}) - b_1(\mathbf{x}_{n_x-1,j+1/2,k})}{h_y} + \frac{b_2(\mathbf{x}_{n_x-1,j,k-1/2}) - b_2(\mathbf{x}_{n_x-1,j,k+1/2})}{h_z},\\
&\qquad\qquad j = 0, \ldots, n_y - 1,\ k = 0, \ldots, n_z - 1.
\end{aligned}$$

The boundary conditions for the pressure (3.12) on this part is simply discretized as

$$p_{n_x-1,j,k} = p_{\mathrm{w}0}(\mathbf{x}_{n_x-1,j,k}), \qquad j = 0, \ldots, n_y - 1,\ k = 0, \ldots, n_z - 1.$$

The boundary conditions for the velocity field and pressure on the other parts of the boundary $\partial\Omega$ can be discretized similarly.

To discretize the boundary conditions (3.8) and (3.11), and define the finite difference equations for all the virtual nodes, we sweep all the virtual nodes. For each of such a node, we use the boundary conditions on the part of the boundary Γ that is near this node. We then express the values and partial derivatives of the components of velocity field and pressure in the boundary conditions in terms of linear combinations of the velocity and pressure values at grid points, both regular and virtual, creating the finite difference equations at these grid points. To achieve high accuracy, we use high-order interpolation at these regular and virtual grid points. We shall detail our discretization of the traction boundary condition in (3.8) in Appendix A1. The divergence-free condition in (3.8) and the pressure boundary condition in (3.11) are similar, and will not be discussed.

4.2 Solving the Linear System

We denote by N_u, N_v, N_w, and N_p the number of both boundary nodes and regular nodes for u, v, w, and p, respectively, and by M_u, M_v, M_w, and M_p the number of virtual nodes for u, v, w, and p, respectively. Clearly, $N_s = O(n_x n_y n_z)$ and $M_s = O((n_x n_y n_z)^{2/3})$ for $s \in \{u, v, w, p\}$. We also denote $N = N_u + N_v + N_w + N_p$ and $M = M_u + M_v + M_w + M_p$. The discretization described in Section 4.1 leads to the following linear system of equations:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}. \tag{4.5}$$

Here, A is the $N \times N$ matrix

$$A = \begin{bmatrix} A_u & O & O & A_{px} \\ O & A_v & O & A_{py} \\ O & O & A_w & A_{pz} \\ O & O & O & A_p \end{bmatrix},$$

where O represents a zero matrix. The matrices A_u, A_v, A_w, and A_p correspond to the finite difference equations resulting from the discretization of the Laplacian of u, v, w, and p in (4.1)–(4.4), respectively, at regular points, together with those equations from the discretization of the boundary conditions on $\partial\Omega$. Each matrix A_s ($s \in \{u, v, w, p\}$) is an $N_s \times N_s$ non-singular matrix. The matrices A_{px}, A_{py}, and A_{pz} correspond to the discretization of derivatives of p in (4.1)–(4.3). The matrix A is nonsingular. The matrix B is an $N \times M$ matrix. It stores the coefficients of the unknown velocity and pressures values at the virtual nodes in the finite difference equations in (4.1)–(4.4) for all the regular and boundary nodes. Note that B is a rather sparse matrix. The matrices C and D are $9M \times N$ and $9M \times M$ matrices, respectively. They correspond to the $9M$ equations from the discretization at all the virtual nodes where we use the boundary conditions. The matrix C stores the coefficients of the unknowns at regular and boundary nodes, while the matrix D stores the coefficients of the unknowns at the virtual nodes. The prefactor 9 is determined by the fact that we choose 9 base points on the interface Γ for each of the M virtual nodes. The vector $\mathbf{y} = (\mathbf{u}_{br}, \mathbf{v}_{br}, \mathbf{w}_{br}, \mathbf{p}_{br})^T$ (T denotes the transpose) is the N-component column vector for the unknowns u, v, w, and p at all the N boundary and regular nodes. The vector $\mathbf{z} = (\mathbf{u}_v, \mathbf{v}_v, \mathbf{w}_v, \mathbf{p}_v)^T$ is the M-component column vector for the unknowns u, v, w, and p at all the M virtual nodes. The vector $\mathbf{c}$ has N components, corresponding to all the g-terms in the N equations in (4.1)–(4.4). Finally, the vector $\mathbf{d}$ is a $9M$-component column vector, corresponding to the right-hand sides of the finite difference equations at all the virtual nodes using the boundary conditions on Γ.

The over-determined system of equations (4.5) is equivalent to the systems

$$S\mathbf{z} = \mathbf{d} - CA^{-1}\mathbf{c}, \tag{4.6}$$

$$A\mathbf{y} = \mathbf{c} - B\mathbf{z}, \tag{4.7}$$

where $S = D - CA^{-1}B$ (the Schur complement). The system (4.7) can be solved with the block backward substitution, together with an algebraic multigrid method (AMG) for each block. The particular AMG algorithm we apply here is the BoomerAMG [64] from Hypre package [23]. The system (4.6) is still over-determined, and we solve for its least-squares solution. This means that we solve the normal equation

$$S^T S\mathbf{z} = S^T(\mathbf{d} - CA^{-1}\mathbf{c}). \tag{4.8}$$

Since the matrix $S^T S$ is dense and not necessarily diagonal dominant and since A is an $N \times N$ matrix, quite large for our three-dimensional problem, we solve the system (4.8) by an iterative method with preconditioning. We shall obtain an approximation $\tilde{S}$ for S, by finding an efficient approximation to $A^{-1}B$, as detailed in Appendix A2. We then apply the GMRES to (4.8). In the GMRES iteration, we use an additive Schwarz method (ASM) for preconditioning $\tilde{S}^T\tilde{S}$ and the AMG method to calculate $A^{-1}\mathbf{c}$. We employ a distributed-memory parallel implementation of our method; and all of our implementation is done with the PETSc package [4–6].

4.3 Convergence Test

Here, we test the convergence of our numerical algorithm on solving the fluid flow equations with a spherical interface. Our applications to some simple molecular systems described in the next section also demonstrate the convergence of our method; cf. in particular, Table 2 in Section 5.1 and Figure 5.1 in Section 5.2.

Let us set $\Omega = (-0.5, 0.5)^3$, $\Omega_{\rm m} = \{(x, y, z) \in \Omega : \sqrt{x^2 + y^2 + z^2} < 0.2\}$, $\Omega_{\rm w} = \Omega \setminus \overline{\Omega_{\rm m}}$, $\Gamma = \partial\Omega_{\rm m}$, and $\mu_{\rm w} = 1$. We consider the system of equations (3.7)–(3.12), with $\Omega_{\rm w}$, $\Omega_{\rm m}$, and Γ replacing $\Omega_{\rm w}(t)$, $\Omega_{\rm m}(t)$, and $\Gamma(t)$, respectively, and with $p = p_{\rm w}$ and the equation $p_{\rm w} = 2\mu_{\rm w}\mathbf{n} \cdot D(\mathbf{u})\mathbf{n} + \mathbf{f}$ replacing (3.11). The function $\mathbf{g}$, b_1, b_2, $\mathbf{f}$, and $p_{\rm w0}$ in this system are determined by the following exact solution:

$$
\begin{aligned}
u(x, y, z) &= -\sin x \sin y \sin z, \\
v(x, y, z) &= -\cos x \cos y \cos z, \\
w(x, y, z) &= -\cos x \sin y (\cos z + \sin z), \\
p_{\rm w}(x, y, z) &= -\sin x \sin y \sin z.
\end{aligned}
$$

We set $n_x = n_y = n_z = n$, and the tolerance of absolute error 1e-9. Figure 4.1 shows the log-log plots of the solution errors in the L^1-norm, L^2-norm, and L^∞-norm, respectively, against n, the number of grid point in each coordinate direction. We observe that the errors decay as n increases in the order $O(n^{-2.5})$ for the velocity components and $O(n^{-2})$ for the pressure for all the norms used. The spikes in these error plots are generic in resolving irregular domain with cartesian grids. We see that our method has the second-order convergence up to the boundary.

5 Applications

We provide three examples of application of our model, which consists of (1.5), (3.7)–(3.12), and (1.8) with $\mathbf{g} = \mathbf{0}$, and our numerical methods. In the first example, we study the solvation of a single-ion and that of a two-plate system. We use our DISM (dynamical implicit-solvent model) to simulate the dynamics of each of these systems to obtain steady states of solute-solvent interfaces, Then, we compute the VISM (variational implicit-solvent model) solvation free energies with such interfaces. We compare our results with the VISM energies. In the second example, we simulate the dynamic of shrinking of a bubble, charged or uncharged, in water, comparing our result with those obtained by solving a generalized Rayleigh–Plesset equation. In the last example, we simulate the two-plate system to demonstrate that our method can capture the dry-wet transition due to the hydrophobic interactions.

Unless otherwise stated, we use the parameters specified in Table 1. Note that we take the value of viscosity $\mu_{\rm w}$ from the reference [20]. We shall also set the tangential components of the velocity field on the boundary of Ω to be $b_1 = 0$ and $b_2 = 0$; cf. (3.9). Other parameters, such as the Tolman length τ, Lennard-Jones parameters ε_i and σ_i, and the partial charges Q_i, will be specified later. We shall use $k_{\rm B}T$ for energy and Å for length. We shall also use a uniform grid with the number of grid points being $n = n_x = n_y = n_z$.

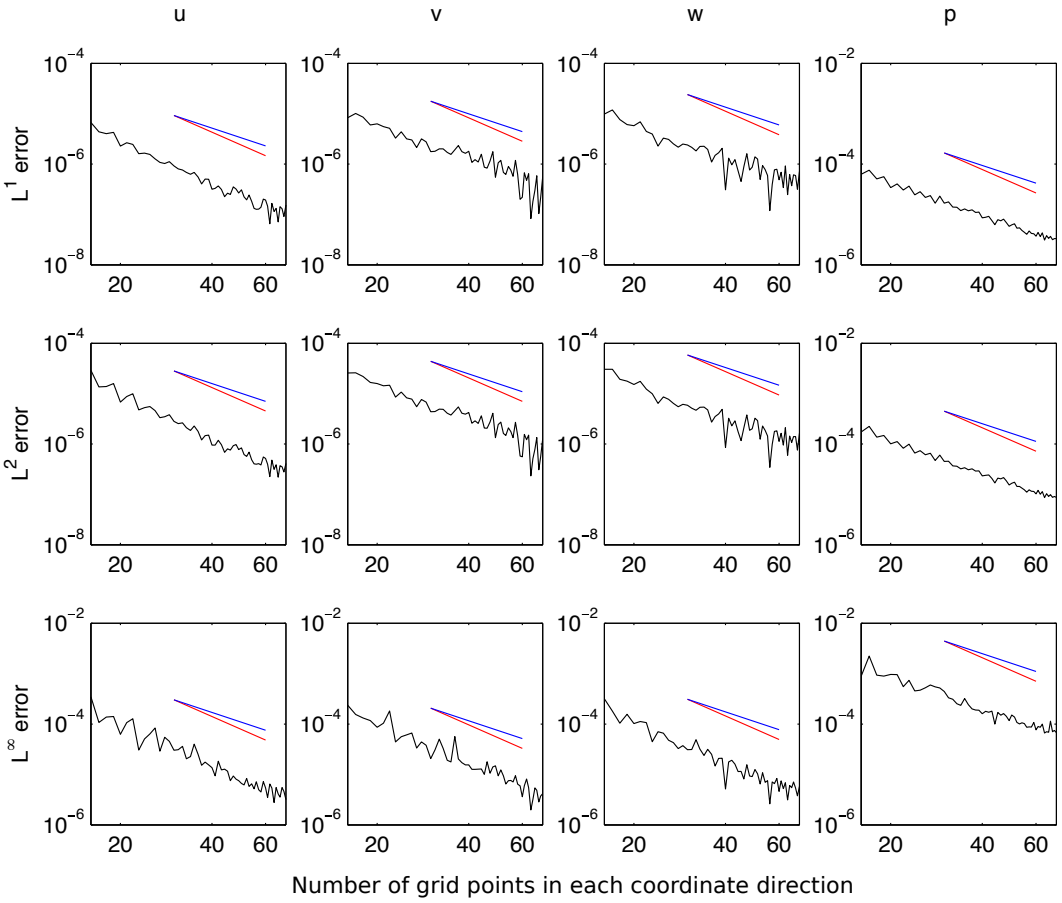

Figure 4.1: Log-log plots of the errors of the velocity components and the pressure, marked by u, v, w, and p, respectively, in the L^1-norm, L^2-norm, and L^∞-norm versus the number of grid points along each coordinate direction. The blue and red lines with slopes -2 and -2.5, respectively, are plotted as references.

5.1 Solvation Free Energy of One Charged Particle and a Two-Plate System

For the system of a single charged particle, such as an ion, with a point charge Q located at the origin, the spherical symmetry simplifies the VISM free-energy functional (1.1) into a function of the radius R of the spherical solute region:

$$G[R] = 4\pi\gamma_0 R^2 + 16\pi\rho_{\mathrm{w}}\varepsilon\left(\frac{\sigma^{12}}{9R^9} - \frac{\sigma^6}{3R^3}\right) + \frac{Q^2}{8\pi\varepsilon_0 R}\left(\frac{1}{\varepsilon_{\mathrm{w}}} - \frac{1}{\varepsilon_{\mathrm{m}}}\right). \tag{5.1}$$

Here, we have set the pressure difference $p_0 = 0$ as the volume contribution is very small compared with other parts in the free energy. We also set the Tolman length $\tau = 0$ for the convenience of computation and comparison between the results of VISM and DISM. Note that $\varepsilon = \varepsilon_1$ and $\sigma = \sigma_1$ are the Lennard-Jones parameters, and $Q = Q_1$ is the point charge,

Table 1: Parameters.

Parameter	Symbol	Value	Unit
temperature	T	298	K
surface tension	γ_0	0.18	$k_{\rm B}T/\text{Å}^2$
solvent viscosity	$\mu_{\rm w}$	0.1285	$k_{\rm B}T\,\text{ps}/\text{Å}^3$
solvent density	$\rho_{\rm w}$	0.033	Å^{-3}
solute dielectric constant	$\varepsilon_{\rm m}$	1	
solvent dielectric constant	$\varepsilon_{\rm w}$	80	
vacuum permittivity	ε_0	1.4372×10^{-4}	$\text{e}^2/(k_{\rm B}T\,\text{Å})$

for the single particle. We use $\varepsilon = 0.361\ k_{\rm B}T$ and $\sigma = 4.577$ Å. The one variable-function (5.1) can be minimized numerically with high accuracy.

We have also solved the system of equations in the DISM. They include (1.5), (3.7)–(3.12), and (1.8), with $\Omega = (-8, 8)^3$. As in the VISM calculation, we set $p_0 = 0$ and $\tau = 0$. We also set $p_{\rm m} = 0$ on $\Gamma(t)$ to replace (1.8) as the contribution of the pressure for a small system (a few particles) is small. Once we obtain a steady-state interface position, we can use the VISM functional (1.1) to calculate the VISM energy.

We consider two cases: $Q = 0.5$ e and $Q = 2$ e. We summarize our VISM and DISM results in Table 2, where the surface energy, LJ energy, and CFA energy corresponding to the second term (with $\tau = 0$), the third term, and the last term in (1.1), respectively. The electrostatic energy $G_{\rm ele}[\Gamma]$ is given in (1.3). The radius in Table 2 means the equilibrium radius of the spherical solutes-solvent interface. The units for the charge, energy, and length are e (elementary charge), $k_{\rm B}T$, and Å, respectively. Clearly, the two models agree with each other very well for these two cases.

Table 2: Energy comparison for a one-particle system.

Charge Q	Model	Surf. Energy	LJ Energy	CFA Energy	Total Energy	Radius
0.5	DISM	37.329	-8.601	-16.884	11.843	4.062
0.5	VISM	37.310	-8.447	-16.888	11.976	4.061
2	DISM	29.734	14.040	-302.783	-259.009	3.626
2	VISM	29.853	13.467	-301.808	-258.488	3.633

We now consider a system of two parallel plates [15, 61, 67]. More precisely, we place 36 atoms on the plane $z = 6$ and another 36 atoms on the plane $z = -6$. (The unit for distance or length is Å.) The positions of these atoms are given by $(2.1945\,i, 2.1945\,j, 6\,k)$ with $i, j \in \{-5, -3, -1, 1, 3, 5\}$ and $k \in \{-1, 1\}$. Again, we set $p_0 = 0$ and $\tau = 0$. The Lennard-Jones parameters associated with all these solute particles are the same: $\sigma = 3$ Å and $\varepsilon = 0.5\ k_{\rm B}T$. Our VISM and DISM computations are done on $\Omega = (-20, 20)^3$. We

consider two cases: (1) uncharged particles, i.e., the partial charge is set to be 0 e for each of the particles; and (2) partially charged with $Q = 0.1$ e assigned at each of the 72 particles. Our results are summarized in Table 3. Again, we see that the dynamic model and our numerical methods can reproduce very well the VISM sovlation free energy for this system.

Table 3: Energy comparison for the two-plate system.

Charge Q	Model	Surf. Energy	LJ Energy	CFA Energy	Total Eenergy
0	DISM	689.410	-342.586	0	346.824
0	VISM	703.623	-349.372	0	354.251
0.1	DISM	690.816	-339.031	-883.269	-531.484
0.1	VISM	701.843	-348.505	-880.263	-526.925

5.2 Dynamics of a Nano-Bubble Shrinking and Collapsing

In this section, we apply our DISM and numerical methods to study the dynamics of shrinking and collapsing of a nano-bubble, assumed to be spherical and centered at the origin. We compare the numerical solutions of our DISM equations with those of the classical or generalized Rayleigh–Plesset equations that have been tested and validated against molecular dynamics simulations [20, 43, 47]. The specific, generalized Rayleigh–Plesset equation that we compare with is (cf. Eq. (5) in [20] without the viscosity correction)

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \frac{r}{4\mu_{\mathrm{w}}} f(r),$$

where $r = r(t)$ is the radius of the spherical bubble at time t and

$$f(r) = p_{\mathrm{m}} - p_{\mathrm{w}} - 2\gamma_0 \left(\frac{1}{r} - \frac{\tau}{r^2}\right) + 4\rho_{\mathrm{w}}\varepsilon \left(\frac{\sigma^{12}}{r^{12}} - \frac{\sigma^6}{r^6}\right) - \frac{1}{32\pi^2\varepsilon_0}\left(\frac{1}{\varepsilon_{\mathrm{m}}} - \frac{1}{\varepsilon_{\mathrm{w}}}\right)\frac{Q^2}{r^4}.$$

Note that $f(r)$ is exactly the VISM boundary force given in (1.2) and (1.4) for a spherical dielectric boundary with only one solute particle.

We shall consider three cases.

Case 1. The collapse of an air bubble. We set $Q = 0$, the Lennard-Jone parameters $\varepsilon = 0$ and $\sigma = 0$, and the Tolman length $\tau = 0$, We use the values of solvent viscosity μ_{w}, solvent bulk density ρ_{w}, and surface tension γ_0 as listed in Table 1. The pressure value on the boundary of computational box is set to be $p_{\mathrm{w}0} = 1$ bar and that on the spherical interface to be $p_{\mathrm{m}} = 0$ to have the approximation of the pressure difference around 1 bar [20].

Case 2. The shrinking of an uncharged particle. We set the partial charge $Q = 0$ e, the Lennard-Jones parameters $\varepsilon = 0.361\ k_{\mathrm{B}}T$ and $\sigma = 4.577$ Å, and the Tolman length $\tau = 0.9$ Å. We use the values of solvent viscosity μ_{w}, solvent bulk density ρ_{w}, and surface

tension γ_0 as listed in Table 1. The pressure value on the boundary of computational box is set to be $p_{w0} = 1$ bar.

Case 3. The shrinking of a charged particle. We set the partial charge $Q = 1$ e, the Lennard-Jones parameters $\varepsilon = 0.361\ k_B T$ and $\sigma = 4.577$ Å, and the Tolman length $\tau = 0.9$ Å. We use the values of solvent viscosity μ_w, solvent bulk density ρ_w, and surface tension γ_0 as listed in Table 1. The pressure value on the boundary of computational box is set to be $p_{w0} = 1$ bar.

In our DISM computations for all these cases, we set $\Omega = (-15, 15)^3$, the number of grid points in each coordinate direction $n = 64$, and the initial radius of the bubble $r_0 = 7.5$ Å. The initial boundary $\Gamma(0)$ is the zero level set of the function $\phi(\mathbf{x}) = |\mathbf{x}| - r_0$. Note that in our DISM computations, we obtain the radius $r = r(t)$ of the bubble by computing the surface area of the bubble using our level-set function and then calculate $r(t) = \sqrt{\text{Area}\,(t)/(4\pi)}$, where Area (t) is the surface area of the spherical bubble at time t.

In Figure 5.1, we plot the radius $r = r(t)$ of the spherical particle vs. the time t obtained by our DISM computations and by solving the generalized Rayleigh–Plesset equation for each of the three cases. We observe that a very good agreement is reached. Since the Rayleigh–Plesset equations have been well validated, our results demonstrate that our computational model and methods can reproduce well the shrinking and collapsing dynamics of a spherical (uncharged or charged) particle. In particular, our numerical boundary conditions do allow the change of volume of a region embedded in an incompressible fluid. We note that our DISM result deviates from that of the generalized Rayleigh–Plesset equation when the radius gets very small. This is due to the numerical resolution of our DISM computations. This deviation is more significant when we used a coarser grid with $n = 32$.

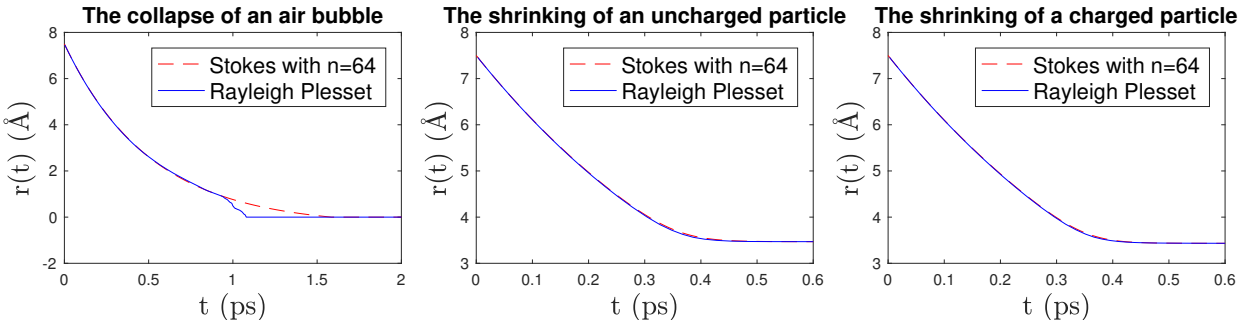

Figure 5.1: The comparison between the DISM computational results and the solution to the generalized Rayleigh–Plesset equation for the three cases of the nano-bubble shrinking and collapsing as described in the text.

5.3 Viscosity Effect to the Conformational Change of a Charged Two-Plate System

In this section, we study the conformational change of a charged two-plate system and the effect of viscosity to such changes. We consider the same two-plate system as in Section 5.1

with the same Lennard-Jone parameters $\varepsilon = 0.5\ k_{\rm B}T$ and $\sigma = 3$ Å for all the particles. As before, we also set $\tau = 0$. We assume each of the 36 particles on one of the plates carries a partial charge of $Q = 0.2$ e and each of those on the other plate carries a partial charge of $Q = -0.2$ e. We use $p_{\rm m}(t) = 1\, k_{\rm B}T/{\rm Vol}(\Omega_{\rm m}(t))$ to replace (1.8). This approximation avoids the change of particle numbers when the solute-solvent interface undergoes a topological change. We simulate our DISM in the region $\Omega = (-20, 20)^3$ with the number of grids $n = 81$ in each of the three coordinate directions. We select three values of the solvent viscosity $\mu_{\rm w}$: 0.12, 0.25, and 0.5 $k_{\rm B}T \cdot {\rm ps/\AA}^3$. For each of these values, we run our code. In each of the simulations, we choose the initial solute-solvent interface to be a box defined by the zero level set of the level-set function $\phi(0, \mathbf{x}) = \max(x - 10, y - 10, z - 10)$. We then run our code with no charges to get a steady-state solute-solvent interface, which corresponds to a wet state as shown in the first snapshot in Figure 5.2. This interface is then used as an initial solute-solvent interface in our simulation with the partial charge $Q = 0.2$ e for each of the 36 particles in one plate and $Q = -0.2$ e for each of the 36 particles in the other plate. In each of our three simulations with different solvent viscosity values, we capture the dynamical evolution of the solute-solvent interface—the dry-wet transition—as shown in Figure 5.2.

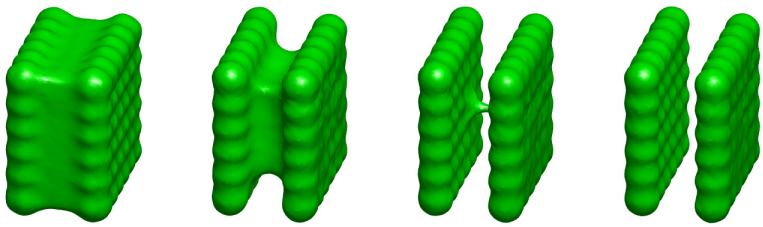

Figure 5.2: Snapshots of conformations of the two-plate system from our DISM simulations.

To understand the viscosity effect to the conformational change, we plot in Figure 5.3 the surface area of the solute-solvent interface $\Gamma(\tilde{t})$ vs. the rescaled simulation time $\tilde{t} = t \times 0.25\, k_{\rm B}T \cdot {\rm ps/\AA}^3/\mu_{\rm w}$. We observe that the three curves collapse into one, indicating that the viscosity slows down the conformational charge and the effect of viscosity is simply rescaling the time of the dynamics in a reciprocal manner. Note that the surface area first increases, corresponding to the drying process that shrinks the hydrophobic pocket; and then decreases slightly to a steady state, corresponding to the dry-wet process breaking the one component solute region into tow. This entire process occurs in a scale at the order of 20 ps, and the topological change takes place roughly at time $t = 27$ ps.

6 Conclusions

We have developed a computational model and numerical methods for the dynamics of solute-solvent interface in the solvation of charged molecules in an aqueous solvent. We treat the

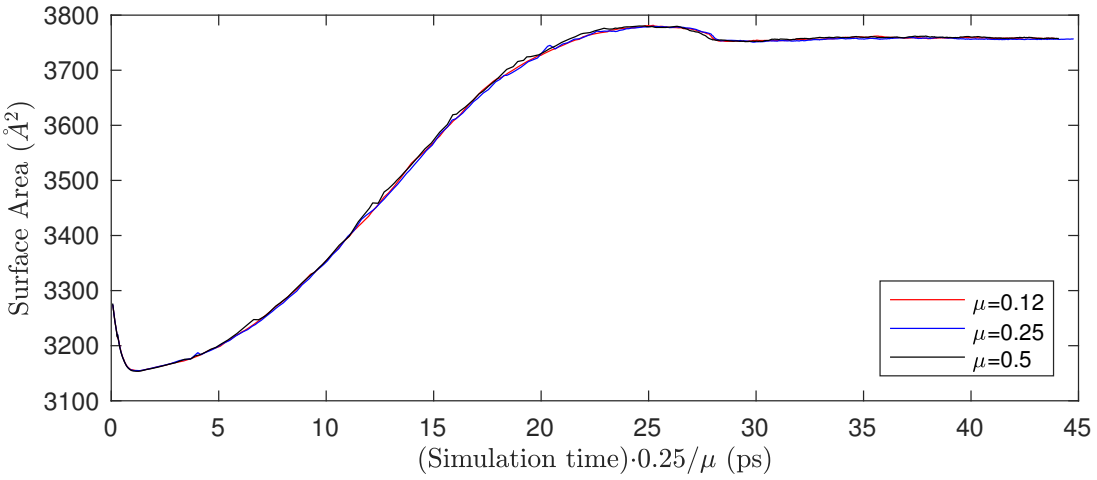

Figure 5.3: The area of the solute-solvent interface vs. the rescaled simulation time for three different values of solvent viscosity $\mu = \mu_{\rm w}$.

solvent as incompressible fluid and describe its flow using Stokes' equations. The interface motion is determined by the solvent fluid motion. All the forces are balanced on the solute-solvent interface, giving rise to the traction boundary conditions on such an interface. These forces consist of the viscous force as well as the surface tension force, van der Waals dispersive force, and the electrostatic force. The later three parts arise from the solvation free energy in a static theory of solvation of charged molecules. One of our main contributions is the construction of the free-flow numerical boundary conditions that allow the change of volume of a solute region embedded in the region of incompressible solvent fluid. We reformulate our underlying fluid flow equations into Poisson's equations for the velocity and pressure, together with boundary conditions. We discretize the traction boundary conditions using a third-order scheme at several points on the solute-solvent interface nearby a given virtual grid point near such an interface. We speed up our computations using various kinds of techniques in solving systems of linear equations.

Our computational tests have shown that our numerical methods are of second-order accuracy up to the boundary. Such accuracy on the boundary results from our discretization of the traction boundary conditions using many different points. Our numerical tests also show that the divergence free condition (i.e., the fluid incompressibility) is retained with a spatial first-order accuracy. With our new free-flow numerical boundary conditions, we have captured the change of volume of solute region in several applications. Moreover, our DISM captures the interfacial geometry very well, as indicated by our comparison of solvation free energies for some molecular systems and our DISM computations of the dynamics of shrinking and collapsing of a bubble.

We now discuss several issues. First, our free-flow boundary conditions allow the solvent fluid to flow into or out of the region near the solute-solvent interface, and therefore increase (or decrease) the total solvent density in the region. This raises the question if our model

can predict well the kinetic energy well of an underlying molecular system. A possible difficulty in calculating the kinetic energy in our DISM approach and then comparing it with the molecular dynamics simulations lies in the fact that such simulations often keep the total number of solvent molecules. Nevertheless, further studies along this line will be much interested.

Second, we observe that the dry-wet transition shown in Figure 5.2 is in the opposite direction of the wet-dry transition, or dewetting transition, as captured with a stochastic level-set VISM approach [69] (cf. Figure 3 there). From the energy landscape theory, it is clear that the wet-try transition can occur more often than the other way around; cf. again [69] and the references therein. However, it is well appreciated that the solvent fluctuation are critical to molecular conformational changes and molecular regcognition in an aqueous environment [7, 10, 37]. Such fluctuations are reflected in the continuum fluid flow equations. Therefore, it is likely that the solvent fluid flow helps the the dry-wet transition as indicated by our computations. Certainly, more studies are needed to understand such two-way transitions.

Third, our computations show an interesting scaling law on the change of interfacial areas vs. the fluid viscosity. This may or may not be explained just by a simple change of variable. Further detailed studies are needed to confirm such a scaling.

Fourth, we have not studied the well-posedness of our underlying system of equations with various kinds of boundary conditions, including the free-flow boundary condition. As we have pointed out, the number of unknowns functions and the number of boundary conditions do match. But, any further understanding of such mathematical issue will be helpful to improve our modeling and numerical computation.

Finally, with our model and numerical methods, it is possible to include the solvent fluctuations that are crucial for molecular systems. One possible way to describe such fluctuations is to study the Navier–Stokes–Landau–Lifshitz equations of fluid equations with fluctuating stress [35]. This will be our future work.

Acknowledgment. H. Sun was supported in part by an AMS–Simons Foundation Travel Grant, Simons Foundation Collaborative Grant with grant number 522790, and an XSEDE startup allocation through the US National Science Foundation grant DMS-170031. S. Zhou was supported in part by the National Natural Science Foundation of China (NSFC) through grant NSFC 11601361 and NSFC 21773165, and Natural Science Foundation of Jiangsu Province, China, through grant BK20160302. L.-T. Cheng and B. Li were supported in part by the US National Science Foundation through grant DMS-1620487. The authors thank Dr. Joachim Dzubiella for helpful discussions.

Appendix

A1. Discretization of the Traction Boundary Condition on the Solute-Solvent Interface

Here we detail our high-accurate method for discretizing the traction boundary condition in (3.8). Fix a virtual node $\mathbf{x}$. We assume that this is a u-point. The case that it is a v or w-point is similar. Let $\mathbf{x}^* \in \Gamma$ be the orthogonal projection of $\mathbf{x}$ on Γ, determined by the condition that the vector connecting these two points is normal to Γ at $\mathbf{x}^*$. To find such a projection, we approximate the normal at $\mathbf{x}^*$ by $\hat{\mathbf{n}} = \nabla\phi(\mathbf{x})/|\nabla\phi(\mathbf{x})|$, using the level-set function ϕ. This is a good approximation, as our level-set function is close to the signed distance function due to the reinitialization in each step. We define $q(\alpha) = \phi(\mathbf{x} - \alpha\hat{\mathbf{n}})$ and solve for α^* such that $q(\alpha^*) = 0$ to determine $\mathbf{x}^* = \mathbf{x} - \alpha^*\hat{\mathbf{n}}$. In the code, we interpolate $q(\alpha)$ at $\alpha = 0$, $\phi(\mathbf{x})$, and $2\phi(\mathbf{x})$, and find the zero α^* of the resulting quadratic interpolation polynomial. Once we find $\mathbf{x}^*$, we calculate $\mathbf{n} = \nabla\phi(\mathbf{x}^*)/|\nabla\phi(\mathbf{x}^*)|$, $\boldsymbol{\tau}_1 = \mathbf{e} \times \mathbf{n}$, and $\boldsymbol{\tau}_2 = \mathbf{n} \times \boldsymbol{\tau}_1$, where $\mathbf{e}$ is a unit vector along a coordinate direction with $|\mathbf{e} \cdot \mathbf{n}| < \sqrt{2}/2$. We then find 8 more points on the boundary Γ approximately. They are $\mathbf{x}^* \pm (h/2)\boldsymbol{\tau}_1$, $\mathbf{x}^* \pm (h/2)\boldsymbol{\tau}_2$, $\mathbf{x}^* \pm (h/2)(\boldsymbol{\tau}_1 + \boldsymbol{\tau}_2)$, and $\mathbf{x}^* \pm (h/2)(\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2)$, where $h = \max\{h_x, h_y, h_z\}$. We call each of these 8 points and the point $\mathbf{x}^*$ a base point corresponding to $\mathbf{x}$.

We now identify some grid points around the virtual node $\mathbf{x}$ and discretize the derivatives of velocity components at the base points using the function values at these grid points. We shall call such grid points stencil points. We aim to achieve a second-order convergence scheme for the velocity and pressure up to Γ. Therefore, we construct a third-order discretization scheme for the gradients of the velocity and pressure. This requires the use of 20 stencil points for each of the velocity components and 10 stencil points for the pressure. We consider various cases in constructing these stencil points. By the definition of a virtual node, u-point, and $\mathcal{N}(\mathbf{x})$, one of the following two conditions should be satisfied:

(1) $\{\mathbf{x} \pm (h_x, 0, 0), \mathbf{x} \pm (0, h_y, 0), \mathbf{x} \pm (0, 0, h_z)\} \cap \Omega_{\mathrm{w}} \neq \emptyset$;

(2) $\{\mathbf{x} + (l_1 h_x, l_2 h_y, 0), \mathbf{x} + (l_1 h_x, 0, l_2 h_z), \mathbf{x} + (0, l_1 h_y, l_2 h_z) : l_1, l_2 \in \{-1/2, 1/2\}\} \cap \Omega_{\mathrm{w}} \neq \emptyset$.

The first condition corresponds to the case where one consecutive stencil point along one of the axes is a regular node. It has 6 possibilities, which are all symmetric. Figure A.1 (a) shows the 20 stencil points corresponding to the case that $\mathbf{x} + (h_x, 0, 0) \in \Omega_{\mathrm{w}}$. Note that we have circled some stencil points to check if they are all regular points. We call these points the check points. Note also that the stencil points in Figure A.1 (a) are representative, for all other cases in condition (1) can be derived by symmetry with respect to the coordinate axes. The second condition corresponds to the case where a neighboring node which corresponds to another velocity component is a regular point. It has 8 possibilities, and again they are all symmetric. Figure A.1 (b) and (c) show the stencil points and check points for the case that $\mathbf{x} + (h_x/2, h_y/2, 0) \in \Omega_{\mathrm{w}}$. All other cases can be considered by symmetry.

Let us label all these 20 stencil points as $\mathbf{x}_k$ ($k = 0, \ldots, 19$) with $\mathbf{x}_0 = \mathbf{x}$. Fix a base point $\mathbf{x}_{\mathrm{b}}$ corresponding to the virtual node $\mathbf{x}$. If $u = u(x, y, z)$ is smooth, then we can Taylor expand $u(\mathbf{x}_k)$ at $\mathbf{x}_{\mathrm{b}}$ up to the third-order derivatives. Let us denote by u_{b}, $u_{\mathrm{b},x}$, $u_{\mathrm{b},y}$, $u_{\mathrm{b},z}$, $u_{\mathrm{b},xy}$, etc.

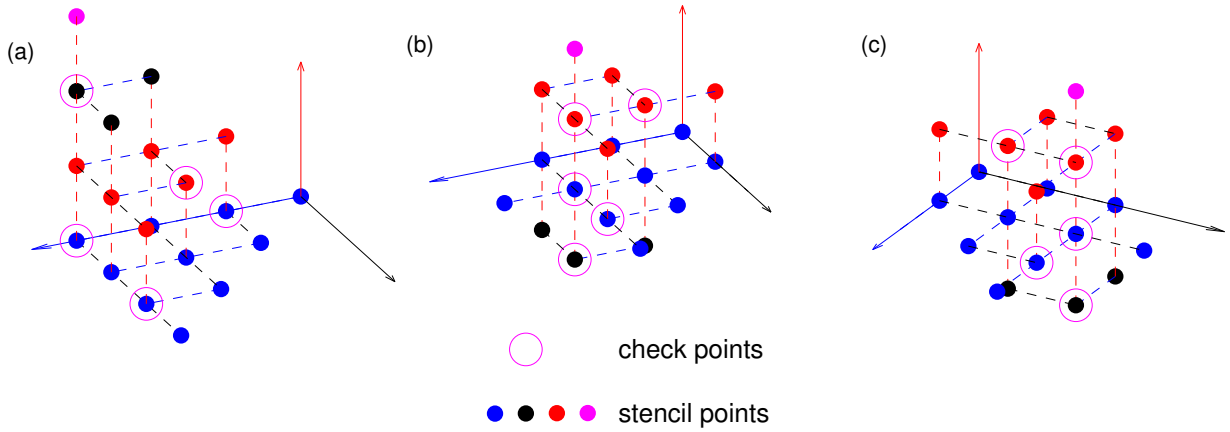

Figure A.1: Stencil points and check points for a virtual node and u-point $\mathbf{x}$ located at the origin, corresponding to various cases of the 20 stencil points for a third-order scheme. In (a), $\mathbf{x}+(h_x,0,0)$ is a regular point. In (b) and (c), $\mathbf{x}+(h_x/2,h_y/2,0)$ is a regular point.

approximate values of $u(\mathbf{x}_\mathrm{b})$, $u_x(\mathbf{x}_\mathrm{b})$, $u_y(\mathbf{x}_\mathrm{b})$, $u_z(\mathbf{x}_\mathrm{b})$, $u_{xy}(\mathbf{x}_\mathrm{b})$, etc. Let us also denote by u_k the approximate values (the finite difference values) of $u(\mathbf{x}_k)$ $(k=0,\ldots,19)$. Then, based on the Taylor expansions for $k=0,\ldots,19$, we relate the function values and derivatives at base points with the finite difference values at the stencil points by the following system of linear equations

$$\begin{bmatrix} 1 & \mathbf{h}_0 & \mathbf{h}_0^2 & \mathbf{h}_0^3 \\ 1 & \mathbf{h}_1 & \mathbf{h}_1^2 & \mathbf{h}_1^3 \\ 1 & \mathbf{h}_2 & \mathbf{h}_2^2 & \mathbf{h}_2^3 \\ 1 & \mathbf{h}_3 & \mathbf{h}_3^2 & \mathbf{h}_3^3 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \mathbf{h}_{19} & \mathbf{h}_{19}^2 & \mathbf{h}_{19}^3 \end{bmatrix} \begin{bmatrix} u_\mathrm{b} \\ u_{\mathrm{b},x} \\ u_{\mathrm{b},y} \\ u_{\mathrm{b},z} \\ \vdots \\ u_{\mathrm{b},xyz} \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{19} \end{bmatrix},$$

where

$$\mathbf{h}_k = \mathbf{x}_k - \mathbf{x}_\mathrm{b} := (h_{x,k}, h_{y,k}, h_{z,k}),$$

$$\mathbf{h}_k^2 = \left(\frac{h_{x,k}^2}{2}, \frac{h_{y,k}^2}{2}, \frac{h_{z,k}^2}{2}, h_{x,k}h_{y,k}, h_{x,k}h_{z,k}, h_{y,k}h_{z,k} \right),$$

$$\mathbf{h}_k^3 = \left(\frac{h_{x,k}^3}{6}, \frac{h_{y,k}^3}{6}, \frac{h_{z,k}^3}{6}, \frac{h_{x,k}^2 h_{y,k}}{2}, \frac{h_{x,k}^2 h_{z,k}}{2}, \frac{h_{y,k}^2 h_{x,k}}{2}, \frac{h_{y,k}^2 h_{z,k}}{2}, \frac{h_{z,k}^2 h_{x,k}}{2}, \frac{h_{z,k}^2 h_{y,k}}{2}, h_{x,k}h_{y,k}h_{z,k} \right).$$

Let us denote $\mathbf{h} = \mathbf{x}_\mathrm{b} - \mathbf{x} = (\lambda_1 h_x, \lambda_2 h_y, \lambda_3 h_z)$ and $(s_1,s_2,s_3) = (\mathrm{sign}(\lambda_1), \mathrm{sign}(\lambda_2), \mathrm{sign}(\lambda_3))$. Then each $\mathbf{h}_k = \mathbf{x}_k - \mathbf{x} - \mathbf{h}$, and hence all $\mathbf{h}_k$, $\mathbf{h}_k^2$, and $\mathbf{h}_k^3$ $(k=0,\ldots,19)$ can be represented in terms of λ_i and s_i $(i=1,2,3)$. With Matlab, we invert symbolically the 20×20 coefficient matrix, obtaining the general formulas for u_b, $u_{\mathrm{b},x}$, etc. for any u-point virtual node and any of its corresponding base points.

Finally, in the traction boundary condition for each of the 9 base points, we replace the derivatives of velocity components and pressure values by the linear combination of the

function values at 20 stencil points, leading to the finite difference equations on the given virtual node.

We note that, due to the complex geometry of the boundary Γ, there may not be enough stencil points available for a third-order discretization. In such a case, we construct a second-order discretization scheme for the gradients of velocity and pressure, requiring 10 stencil points for the velocity and 4 for the pressure. If there are still not enough grid points, we then construct a first-order discretization scheme, with only 4 stencil points for the velocity and 1 for the pressure. By our definition of a virtual node, a first-order discretization scheme can always be constructed.

A2. A Local Relaxation Scheme

Here we describe our method of local relaxation in finding an approximation of $A^{-1}B$, and further the approximation $\tilde{S}$ of S. We first observe that the discretization of the traction boundary conditions involves only local stencils composed of a virtual node and several nearby regular nodes. That means that the nonzero elements in each row of C are clustered near the virtual nodes locally. Moreover, we note that each column vector $\mathbf{b}$ of B has exactly one nonzero element at an index corresponding to a virtual node. Hence, for each such virtual node $\mathbf{x}_{\text{index}}$, we fix a grid box $[-a, a]^3$ centered at that virtual node, and then we solve $A\boldsymbol{\zeta} = \mathbf{b}$ locally by setting the components of $\boldsymbol{\zeta}$ corresponding to the boundary of the box to be zero, and perform Gauss–Seidel sweeps in eight directions. We denote by $\zeta(\mathbf{x}_{\text{index}})$ and $b(\mathbf{x}_{\text{index}})$ the components of $\boldsymbol{\zeta}$ and $\mathbf{b}$, respectively. We summarize our local relaxation sweep algorithm in the following pseudo-code:

for $(i := \pm a$ to $\mp a) \wedge (j := \pm a$ to $\mp a) \wedge (k := \pm a$ to $\mp a)$
  newIndex = index + (i, j, k);
  if ($\mathbf{x}_{\text{newIndex}}$ is a void node) $\vee$ $(i = \pm a) \vee (j = \pm a) \vee (k = \pm a)$
    $\zeta(\mathbf{x}_{\text{newIndex}}) = 0$;
  else if $\mathbf{x}_{\text{newIndex}}$ is a virtual node
    $\zeta(\mathbf{x}_{\text{newIndex}}) = b(\mathbf{x}_{\text{newIndex}})$;
  else if $\mathbf{x}_{\text{newIndex}}$ is a regular node
    $\zeta(\mathbf{x}_{\text{newIndex}}) = \sum_{i''=\pm1, j''=\pm1, k''=\pm1} \zeta(\mathbf{x}_{\text{newIndex}+(i'',j'',k'')})/6 + b(\mathbf{x}_{\text{newIndex}})$;
  end
end

We apply the local sweep algorithm a few times until convergence is reached, leading to an approximation of the corresponding rows in $A^{-1}\mathbf{b}$, where $\mathbf{b}$ has nonzero value at a u, v, or w virtual node. However, for $\mathbf{b}$ corresponding to a p virtual node, the above algorithm needs to be complemented by a propagation algorithm, which passes the influence of p to velocity components through ∇p:

for $(i := i' - \pm a$ to $i' + \mp a) \wedge (j := j' - \pm a$ to $j' + \mp a) \wedge (k := k' - \pm a$ to $k' + \mp a)$
  if $\mathbf{x}_{i-1/2,j,k}$ is an regular node

$b(\mathbf{x}_{i-1/2,j,k}) \leftarrow b(\mathbf{x}_{i-1/2,j,k}) - D^x_- p_{i,j,k}$;
else if $\mathbf{x}_{i,j-1/2,k}$ is an regular node
$b(\mathbf{x}_{i,j-1/2,k}) \leftarrow b(\mathbf{x}_{i,j-1/2,k}) - D^y_- p_{i,j,k}$;
else if $\mathbf{x}_{i,j,k-1/2}$ is an regular node
$b(\mathbf{x}_{i,j,k-1/2}) \leftarrow b(\mathbf{x}_{i,j,k-1/2}) - D^z_- p_{i,j,k}$;
end
end

After applying the propagation algorithm for p, we apply the sweep algorithm for u, v, w again until convergence. The sweep algorithm is applied in the box $[-a, a]^3$ centered at the virtual node. Typically, we choose a to be between 5 and 10. We then take the values of the vector $\boldsymbol{\zeta}$ from a smaller box $[-a', a']^3$ centered at the virtual node, as the only nonzero components of $\boldsymbol{\zeta}$. Typically, we choose a' to be 4. By approximating all the columns of $A^{-1}B$ in this way, we obtain an approximation $\tilde{S}$ of $S = D - CA^{-1}B$.

References

[1] A. Alexander-Katz, M. F. Schneider, S. W. Schneider, A. Wixforth, and R. R. Netz. Shear-flow-induced unfolding of polymeric globules. *Phys. Rev. Lett.*, 97:138101, 2006.

[2] D. Andelman. Electrostatic properties of membranes: The Poisson–Boltzmann theory. In R. Lipowsky and E. Sackmann, editors, *Handbook of Biological Physics*, volume 1, pages 603–642. Elsevier, 1995.

[3] D. C. Assêncio and J. M. Teran. A second order virtual node algorithm for Stokes flow problems with interfacial forces. *J. Comput. Phys.*, 250:77–105, 2013.

[4] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015.

[5] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc Web page, 2015.

[6] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[7] R. Baron and J. A. Mccammon. Molecular recognition and ligand association. *Annu. Rev. Phys. Chem.*, 64:151–175, 2013.

[8] D. Bashford and D. A. Case. Generalized Born models of macromolecular solvation effects. *Ann. Rev. Phys. Chem.*, 51:129–152, 2000.

[9] P. W. Bates, Z. Chen, Y. H. Sun, G. W. Wei, and S. Zhao. Geometric and potential driving formation and evolution of biomolecular surfaces. *J. Math. Biol.*, 59:193–231, 2009.

[10] M.-C. Bellissent-Funel, A. Hassanali, M. Havenith, R. Henchman, P. Pohl, F. Sterpone, D. van der Spoel, Y. Xu, and A. E. Garcia. Water determines the structure and dynamics of proteins. *Chem. Rev.*, 116:7673–7697, 2016.

[11] B. A. Camley and F. L. H. Brown. Dynamic simulations of multicomponent lipid membranes over long length and time scales. *Phys. Rev. Lett.*, 105:148102, 2010.

[12] B. A. Camley and F. L. H. Brown. Fluctuating hydrodynamics of multicomponent membranes with embedded proteins. *J. Chem. Phys.*, 141:075103, 2014.

[13] J. Che, J. Dzubiella, B. Li, and J. A. McCammon. Electrostatic free energy and its variations in implicit solvent models. *J. Phys. Chem. B*, 112:3058–3069, 2008.

[14] H. B. Cheng, L.-T. Cheng, and B. Li. Yukawa-field approximation of electrostatic free energy and dielectric boundary force. *Nonlinearity*, 24:3215–3236, 2011.

[15] L.-T. Cheng, J. Dzubiella, J. A. McCammon, and B. Li. Application of the level-set method to the implicit solvation of nonpolar molecules. *J. Chem. Phys.*, 127:084503, 2007.

[16] L.-T. Cheng, B. Li, and Z. Wang. Level-set minimization of potential controlled Hadwiger valuations for molecular solvation. *J. Comput. Phys.*, 229:8497–8510, 2010.

[17] L.-T. Cheng, Z. Wang, P. Setny, J. Dzubiella, B. Li, and J. A. McCammon. Interfaces and hydrophobic interactions in receptor-ligand systems: A level-set variational implicit solvent approach. *J. Chem. Phys.*, 131:144102, 2009.

[18] L.-T. Cheng, Y. Xie, J. Dzubiella, J. A. McCammon, J. Che, and B. Li. Coupling the level-set method with molecular mechanics for variational implicit solvation of nonpolar molecules. *J. Chem. Theory Comput.*, 5:257–266, 2009.

[19] M. E. Davis and J. A. McCammon. Electrostatics in biomolecular structure and dynamics. *Chem. Rev.*, 90:509–521, 1990.

[20] J. Dzubiella. Interface dynamics of microscopic cavities in water. *J. Chem. Phys.*, 126:194504, 2007.

[21] J. Dzubiella, J. Swanson, and J. McCammon. Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models. *Phys. Rev. Lett.*, 96:087802, 2006.

[22] J. Dzubiella, J. Swanson, and J. McCammon. Coupling nonpolar and polar solvation free energies in implicit solvent models. *J. Chem. Phys.*, 124:084905, 2006.

[23] R. D. Falgout and U. M. Yang. Hypre: A library of high performance preconditioners. In P. M. A. Sloot, C. J. K. Tan, J. J. Dongarra, and A. G. Hoekstra, editors, *ICCS '02 Proceedings of the International Conference on Computational Science-Part III*, pages 632–641. Springer–Verlag, 2002.

[24] I. J. Finkelstein, A. M. Massari, and M. D. Fayer. Viscosity-dependent protein dynamics. *Biophys. J.*, 92:3652–3662, 2007.

[25] P. M. Gresho and R. L. Sani. On pressure boundary conditions for the incompressible Navier–Stokes equation. *Int. J. Numer. Mathods Fluids*, 7:1111–1145, 1987.

[26] Z. Guo, B. Li, J. Dzubiella, L.-T. Cheng, J. A. McCammon, and J. Che. Evaluation of hydration free energy by level-set variational implicit-solvent model with Coulomb-field approximation. *J. Chem. Theory Comput.*, 9:1778–1787, 2013.

[27] Z. Guo, B. Li, J. Dzubiella, L.-T. Cheng, J. A. McCammon, and J. Che. Heterogeneous hydration of p53/mdm2 complex. *J. Chem. Theory Comput.*, 10:1302–1313, 2014.

[28] S. J. Hagen. Solvent viscosity and friction in protein folding dynamics. *Curr. Protein Pept. Sci.*, 11:385–395, 2010.

[29] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8:2182–2189, 1965.

[30] W. D. Henshaw. A fourth-order accurate method for the incompressible Navier–Stokes equations on overlapping grids. *J. Comput. Phys.*, 113:13–25, 1994.

[31] R. Holyst, M. Litniewski, and P. Garstecki. Large-scale molecular dynamics verification of the Rayleigh–Plesset approximation for collapse of nanobubbles. *Phys. Rev. E*, 82:066309, 2010.

[32] H. Johnston and J.-G. Liu. A finite difference method for incompressible flow based on local pressure boundary conditions. *J. Comput. Phys.*, 180:120–154, 2002.

[33] H. Johnston and J.-G. Liu. Accurate, stable and efficient Navier–Stokes solvers based on explicit treatment of the pressure term. *J. Comput. Phys.*, 199:221–259, 2004.

[34] D. K. Klimov and D. Thirumalai. Viscosity dependence of folding rates of protein. *Phys. Rev. Lett.*, 79:317–320, 1997.

[35] L. D. Landau and E. M. Lifshitz. *Fluid Mechanics*, volume 6 of *Course of Theoretical Physics*. Pergamon, 1959.

[36] L. G. Leal. *Advanced transport phenomena: Fluid mechanics and convective transport processes*. Cambridge University Press, 2007.

[37] Y. Levy and J. Onuchic. Water mediation in protein folding and molecular recognition. *Ann. Rev. Biophys. Biomol. Struct.*, 35:389–415, 2006.

[38] B. Li. Minimization of electrostatic free energy and the Poisson–Boltzmann equation for molecular solvation with implicit solvent. *SIAM J. Math. Anal.*, 40:2536–2566, 2009.

[39] B. Li., X.-L. Cheng, and Z.-F. Zhang. Dielectric boundary force in molecular solvation with the Poisson–Boltzmann free energy: A shape derivative approach. *SIAM J. Applied Math.*, 71:2093–2111, 2011.

[40] B. Li, H. Sun, and S. Zhou. Stability of a cylindrical solute-solvent interface: Effect of geometry, electrostatics, and hydrodynamics. *SIAM J. Appl. Math.*, 75:907–928, 2015.

[41] Z. Li, Q. Cai, H. Zhao, and R. Luo. A semi-implicit augmented IIM for Navier–Stokes equations with open and traction boundary conditions. *J. Comput. Phys.*, 297:182–193, 2015.

[42] F. Lugli and Z. Francesco. Molecular dynamics of nanobubbles' collapse in ionic solutions. *ChemPhysChem*, 8:47–49, 2007.

[43] F. Lugli, S. Höfinger, and F. Zerbetto. The collapse of nanobubbles in water. *J. Am. Chem. Soc.*, 127:8020–8021, 2005.

[44] N. Molino, Z. Bao, and R. Fedkiw. A virtual node algorithm for changing mesh topology during simulation. *ACM Trans. Graph.*, 23:385–392, 2004.

[45] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer, New York, 2002.

[46] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[47] M. S. Plesset and A. Prosperetti. Bubble dynamics and cavitation. *Annu. Rev. Fluid Mech.*, 9:145–185, 1977.

[48] C. G. Ricci, B. Li, L.-T. Cheng, J. Dzubiella, and J. A. McCammon. 'martinizing' the variational implicit solvent method (VISM): Solvation free energy for coarse-grained proteins. *J. Phys. Chem. B*, 121:6538–6548, 2017.

[49] S. W. Schneider, S. Nuschele, A. Wixforth, C. Gorzelanny, A. Alexander-Katz, R. R. Netz, and M. F. Schneider. Shear-induced unfolding triggers adhesion of von Willebrand factor fibers. *Proc. Natl Acad. Sci. USA*, 104:7899–7903, 2007.

[50] C. Schroeder, A. Stomakhin, R. Howes, and J. M. Teran. A second order virtual node algorithm for Navier–Stokes flow problems with interfacial forces and discontinuous material properties. *J. Comput. Phys.*, 265:221–245, 2014.

[51] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 2nd edition, 1999.

[52] P. Setny, Z. Wang, L.-T. Cheng, B. Li, J. A. McCammon, and J. Dzubiella. Dewetting-controlled binding of ligands to hydrophobic pockets. *Phys. Rev. Lett.*, 103:187801, 2009.

[53] K. A. Sharp and B. Honig. Electrostatic interactions in macromolecules: Theory and applications. *Annu. Rev. Biophys. Chem.*, 19:301–332, 1990.

[54] D. Shirokoff and R. R. Rosales. An efficient method for the incompressible Navier–Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary. *J. Comput. Phys.*, 230:8619–8646, 2011.

[55] C. A. Siedlecki, B. J. Lestini, K. K. Kottke-Marchant, S. J. Eppell, D. L. Wilson, and R. E. Marchant. Shear-dependent changes in the three-dimensional structure of human von Willebrand factor. *Blood*, 88:2939–2950, 1996.

[56] I. Singh, E. Themistou, L. Porcar, and S. Neelamegham. Fluid shear induces conformation change in human blood protein von Willebrand factor in solution. *Biophys. J.*, 96:2313–2320, 2009.

[57] H. Sun, S. Zhou, D. K. Moore, L.-T. Cheng, and B. Li. Numerical treatment of Stokes solvent flow and solute-solvent interfacial dynamics for nonpolar molecules. *J. Sci. Comput.*, 67(2):705–723, 2016.

[58] P. Szymczak and M. Cieplak. Hydrodynamic effects in proteins. *J. Phys.: Condens. Matter*, 23:033102, 2011.

[59] R. C. Tolman. The effect of droplet size on surface tension. *J. Chem. Phys.*, 17:333–337, 1949.

[60] R. M. A. Vergauwe, H. Uji-i, K. De Ceunynck, J. Vermant, K. Vanhoorelbeke, and J. Hofkens. Shear-stress-induced conformational changes of von Willebrand factor in water-glycerol mixture observed with single molecule microscopy. *J. Phys. Chem. B*, 118:5660–5669, 2014.

[61] Z. Wang, J. Che, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. Level-set variational implicit solvation with the Coulomb-field approximation. *J. Chem. Theory Comput.*, 8:386–397, 2012.

[62] L. Xiao, Q. Cai, Z. Li, H. Zhao, and R. Luo. A multi-scale method for dynamics simulation in continuum solvents I: Finite-difference algorithm for Navier–Stokes equation. *Chem. Phys. Lett.*, 616:67–74, 2014.

[63] L. Xiao and R. Luo. Exploring a multi-scale method for molecular simulation in continuum solvent model: Explicit simulation of continuum solvent as an incompressible fluid. *J. Chem. Phys.*, 147:214112, 2017.

[64] U. M. Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41:155–177, 2002.

[65] H. X. Zhou. Macromolecular electrostatic energy within the nonlinear Poisson–Boltzmann equation. *J. Chem. Phys.*, 100:3152–3162, 1994.

[66] S. Zhou, L. Cheng, H. Sun, J. Che, J. Dzubiella, B. Li, and J. A. McCammon. Ls-vism: A software package for analysis of biomolecular solvation. *J. Comput. Chem.*, 36:1047–1059, 2015.

[67] S. Zhou, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. Variational implicit solvation with Poisson–Boltzmann theory. *J. Chem. Theory Comput.*, 10(4):1454–1467, 2014.

[68] S. Zhou, K. E. Rogers, C. F. de Oliveira, R. Baron, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. Variational implicit-solvent modeling of host-guest binding: A case study on cucurbit[7]uril. *J. Chem. Theory Comput.*, 9:4195–4204, 2013.

[69] S. Zhou, H. Sun, L.-T. Cheng, J. Dzubiella, B. Li, and J. A. McCammon. Stochastic level-set variational implicit-solvent approach to solute-solvent interfacial fluctuations. *J. Chem. Phys.*, 145:054114, 2016.