

HHS Public Access

Author manuscript *J Comput Chem.* Author manuscript; available in PMC 2016 May 30.

Published in final edited form as:

J Comput Chem. 2015 May 30; 36(14): 1047–1059. doi:10.1002/jcc.23890.

LS-VISM: A Software Package for Analysis of Biomolecular Solvation

Shenggao Zhou^{1,*}, Li-Tien Cheng^{1,†}, Hui Sun¹, Jianwei Che², Joachim Dzubiella³, Bo Li^{1,‡}, and J. Andrew McCammon⁴

¹Department of Mathematics, University of California, San Diego, La Jolla, California 92093-0112, United States

²Parallel Computing Labs, 3525 Del Mar Heights Road, No. 288, San Diego, CA 92130

³Soft Matter and Functional Materials, Helmholtz-Center Berlin, 14109 Berlin, Germany, and Physics Department, Humboldt-University of Berlin, 12489 Berlin, Germany

⁴Department of Chemistry and Biochemistry, Department of Pharmacology, Howard Hughes Medical Institute, University of California, San Diego, La Jolla, California 92093-0365, United States

Abstract

We introduce a software package for the analysis of biomolecular solvation. The package collects computer codes that implement numerical methods for a variational implicit-solvent model (VISM). The input of the package includes the atomic data of biomolecules under consideration and the macroscopic parameters such as solute-solvent surface tension, bulk solvent density and ionic concentrations, and the dielectric coefficients. The output includes estimated solvation free energies and optimal macroscopic solute-solvent interfaces that are obtained by minimizing the VISM solvation free-energy functional among all possible solute-solvent interfaces enclosing the solute atoms. We review the VISM with various descriptions of electrostatics. We also review our numerical methods that consist mainly of the level-set method for relaxing the VISM free-energy functional and a compact coupling interface method for the dielectric Poisson-Boltzmann equation. Such numerical methods and algorithms constitute the central modules of the software package. We detail the structure of the package, format of input and output files, work flow of the codes, and the post-processing of output data. Our demo application to a host-guest system illustrates how to use the package to perform solvation analysis for biomolecules, including ligand-receptor binding systems. The package is simple and flexible with respect to minimum adjustable parameters and a wide range of applications. Future extensions of the package use can include the efficient identification of ligand binding pockets on protein surfaces.

^{*}To whom correspondence should be addressed: s4zhou@math.ucsd.edu. [†]To whom correspondence should be addressed: lcheng@math.ucsd.edu. [‡]To whom correspondence should be addressed: bli@math.ucsd.edu.

Keywords

biomolecular modeling; variational implicit-solvent model; solvation free energy; Poisson– Boltzmann theory; dielectric boundary force; level-set method; compact coupling interface method

I. INTRODUCTION

The stability and dynamics of biomolecules in aqueous solution are crucial to the biological function of the underlying biological systems. Computer studies of biomolecular interactions have been one of the main scientific tools to understand such stability and dynamics. One of the bottlenecks in such studies is to accurately and efficiently describe the effect of aqueous solvent, which is significant in biomolecular processes such as protein folding and molecular recognition.^{1,2} In this regard, explicit solvent molecular dynamics (MD) simulations and implicit-solvent continuum dielectric models have been the two major classes of theoretical approaches to biomolecular interactions.

MD simulations describe the motion of individual biomolecular atoms, solvent molecules, and ions by Newton's law of motion.^{3–7} Such an approach has been long and widely used for biomolecular modeling and many software packages, such as AMBER, CHARMM, GROMOS, and NAMD, have been developed. While overall successful, MD simulations are in general relatively slow, limited to not very large systems and not very long time simulations. One of the issues is that the long-range charge-charge interaction requires much of the computational work.

Implicit-solvent models,^{8–11} on the other hand, are much more efficient in the static and dynamic analysis of biomolecular systems. In a commonly used dielectric boundary implicit-solvent model, the solvation free energy is calculated as the sum of surface energy, defined to be the surface area of the dielectric boundary multiplied by the constant surface tension, and the electrostatic free energy calculated by the Poisson–Boltzmann (PB) theory^{12–22} or the generalized Born (GB) model.^{23,24} In most commonly used such PB or GB surface area models, the dielectric boundary is defined as the van der Waals surface (vdWS), solvent-excluded surface (SES), or solvent-accessible surface (SAS).^{25–29} Despite their successful applications to many cases, the general accuracy and applicability of these *a priori* defined surfaces with many system-dependent parameters are still questionable. Because the polar and nonpolar contributions are decoupled, these fixed-surface models can hardly capture the important details of hydrophobic interactions.^{30–35}

Recent years have seen the initial success of a new, variational implicit-solvent model (VISM)^{36,37}. VISM is a continuum-solvent dielectric boundary model, similar to other existing models, but is variational in nature. In VISM, one minimizes a solvation freeenergy functional of all possible solute-solvent interfaces, i.e., dielectric boundaries, to provide estimated solvation free energies and stable equilibrium dielectric boundaries. The VISM free energy consists of the solute-solvent interfacial energy, the solute-solvent van der Waals (vdW) interaction energy, and the electrostatic solvation free energy, all expressed in terms of the solute-solvent interface. It allows a curvature correction in the surface energy,

couples polar and nonpolar contributions self-consistently, and treats in a unified way biomolecular systems of any geometry and charge distributions. For several years, we have developed a robust level-set method for relaxing numerically the VISM free-energy functional together with a highly accurate compact coupled interface method (CCIM) for dielectric continuum electrostatics. Our extensive computational tests have shown that the level-set VISM can efficiently provide qualitatively accurate estimates of solvation free energies and the location of stable equilibrium dielectric boundaries. In particular, our continuum-solvent approach can describe the subtle charge-charge interaction, capture the hydrophobic evaporation, and predict the polymodal hydration states and the related hysteresis; see^{38–47}. Recently, we have applied our level-set VISM to the identification of protein-ligand binding sites on protein surfaces.⁴⁸ We notice that other related theories and models have been proposed in literature.^{49–52}

In this work, we introduce a software package, called LS-VISM, for the static analysis of biomolecular interactions in aqueous solution. It assembles computer codes of our numerical methods that implement the theory of VISM. The software package is designed mainly to estimate the solvation free energy of a biomolecule in water, as well as the corresponding components of the free energy, such as the solute-solvent interfacial energy, solute-solvent vdW interaction energy, and the electrostatic free energy. The package can be also used to describe the static properties of molecular recognition such as host-guest systems and protein-ligand binding. A minimal usage of the package is to solve the PB equation, linear or nonlinear, to obtain the electrostatic potential and free energy. The time of running our LS-VISM software package for a single static analysis of a biomolecular system varies from minutes to hours.

Typically the input data of our LS-VISM package include the following: (1) Solute atomic positions that can be often obtained from an existing database such as the protein data bank (PDB)⁵³; (2) The partial charges and solute-solvent Lennard-Jones (LJ) interaction parameters that can be obtained from the force-field in a MD simulation model; (3) The surface tension for the solute-solvent interface, the bulk solvent density and bulk ionic concentrations, and the dielectric coefficients of the solute and solvent; and (4) Initial guesses of solute-solvent interfaces and, if needed, the reaction coordinates for an underlying biomolecular system. The output from our package can include the solvation free energies, binding free energies for molecular recognition, the location of stable equilibrium dielectric boundaries, and the solvent potential of mean force.

Here we shall first review the basic theory of VISM. In particular, we present various descriptions of the electrostatics with different complexities. These include the Coulomb-field approximation (CFA)^{24,43,44,54,55} without solving partial differential equations, the dielectric Poisson's equation, and both linear and nonlinear PB equations that model the ionic effect. We then review our numerical methods that implement the VISM. These include a robust level-set method for relaxing the VISM free-energy functional, a highly accurate compact coupling interface method (CCIM) for solving dielectric Poisson's and PB equations, and various techniques of numerical integration and post processing.

The level-set method is used as follows: We first specify (by guess) an initial solute-solvent interface that may have a large free-energy value. We then evolve the surface by the level-set method in the direction of steepest decent of the VISM free energy. The normal velocity in the level-set method is exactly in magnitude the (normal component of) boundary force defined as the negative first variation of the VISM free-energy functional with respect to the location change of the dielectric boundary. Different initial surfaces can lead to different physically meaningful local minima of the VISM functional that correspond to different hydration states of the system.^{40,42,43,46} To accelerate the calculations, we implement a local level-set method where the level-set equation is solved in a narrow band centered at the solute-solvent surface. Several techniques of acceleration are also proposed. The initial surface is first relaxed with the efficient CFA and then relaxed with the more accurate PB or LPB description. Further, we design a two-grid treatment of the surface relaxation: the initial surface is first relaxed efficiently in a coarse grid using the CFA and is subsequently interpolated to a refined grid for further relaxation. This treatment is remarkably efficient for poorly chosen initial guesses.

We finally describe the structure and work flow of our software package, the format of input and output files, the visualization of equilibrium surfaces, and the post-processing analysis of solvation free energies. To illustrate the usage of the package, we present a demo application to a host-guest system. Various auxiliary files are assembled in the Appendix.

We organize the rest of our paper as follows: In Section II and III, we review the theory of VISM and our numerical methods that implement VISM. In Section IV, we detail the structure of our package, format of input and output files, work flow of the codes, and the post-processing of output data. In Section V, we present a demo application of the package to a host-guest system. Finally, in Section VI, we draw conclusions.

II. REVIEW OF THEORY

A. VISM Free-Energy Functional

We consider the solvation of biomolecules in an aqueous solvent. The geometry of such a solvation system consists of three parts: the solute region Ω_m , the solvent region Ω_w , and the solute-solvent interface Γ , cf. Figure 1. The solute molecules have *N* atoms that are located at positions $\mathbf{x}_1, \ldots, \mathbf{x}_N$ inside Ω_m , carrying partial charges Q_1, \ldots, Q_N , respectively. The solute-solvent interface is Γ treated as a dielectric boundary. This means that we assume by approximation that the dielectric coefficient ε_m in the solute region Ω_m and that ε_w in the solvent region are constants.

In VISM, we minimize the following solvation free-energy functional $G[\Gamma]$ of all possible solute-solvent interfaces Γ :

$$G(\Gamma) = \underbrace{\Delta P \operatorname{vol}(\Omega_{\mathrm{m}}) + \int_{\Gamma} \gamma \, dS}_{G_{\mathrm{geom}}[\Gamma]: \text{ geometrical part}} + \underbrace{\rho_{\mathrm{w}} \sum_{i=1}^{N} \int_{\Omega_{\mathrm{w}}} U_{i}(|\mathbf{x} - \mathbf{x}_{i}|) \, dV}_{G_{\mathrm{vdW}}[\Gamma]: \operatorname{vdW part}} + G_{\mathrm{elec}}[\Gamma], \quad (\mathrm{II.1})$$

Here, the first term $P \operatorname{vol} (\Omega_m)$ describes the energy of creating the solute cavity in the solvent medium, where P is the difference between the pressures of the solvent liquid and solute vapor, respectively. The second term, i.e., the surface integral term, is the surface energy, where γ is the solute-solvent interfacial surface tension. We assume that

$$\gamma = \gamma_0 (1 - 2\tau H)$$

where γ_0 is the surface tension for a planar liquid-vapor interface, τ is the curvature correction coefficient known as the Tolman length,^{30,56} and *H* is the local mean curvature (the average of the two principal curvatures) that is positive for a spherical solute. We call the sum of the first two terms in *G*[Γ] the geometrical part of the solvation free energy and denote it by *G*_{geom}[Γ]. The third term is the solute-solvent van der Waals (vdW) type interaction energy. The constant ρ_w is the bulk solvent density. For each *i*, the term *U_i*(|**x** -**x**_{*i*}|) is the vdW type interaction potential between the solute atom at **x**_{*i*} and a solvent molecule or ion at **x**. We employ the Lennard-Jones (LJ) potential

$$U_i(r) = 4\varepsilon_i \left[\left(\frac{\sigma_i}{r}\right)^{12} - \left(\frac{\sigma_i}{r}\right)^6 \right], \quad \text{(II.2)}$$

where the parameters ε_i of energy and σ_i of length can vary with solute atoms as in a conventional force-field. We call the third term van der Waals (vdW) part of the solvation free energy and denote it by $G_{vdW}[\Gamma]$.

The last term $G_{\text{elec}}[\Gamma]$ is the electrostatic part of the solvation free energy. In our package, we provide four different descriptions of $G_{\text{elec}}[\Gamma]$: the Coulomb-field approximation (CFA);^{24,43,44,54,55} Poisson's (P) equation; the linearized Poisson–Boltzmann (LPB) equation; and the Poisson–Boltzmann (PB) equation^{13,15–17,21,57}. They are given by

$$\left(\begin{array}{c} G_{\text{elec}}^{\text{CFA}}[\Gamma] = \frac{1}{32\pi^2 \varepsilon_0} \left(\frac{1}{\varepsilon_{\text{w}}} - \frac{1}{\varepsilon_{\text{m}}} \right) \int_{\Omega_{\text{w}}} \left| \sum_{i=1}^{N} \frac{Q_i(\mathbf{x} - \mathbf{x}_i)}{|\mathbf{x} - \mathbf{x}_i|^3} \right|^2 dV \right)$$

$$G_{\text{elec}}[\Gamma] = \begin{cases} G_{\text{elec}}^{\text{P}}[\Gamma] = \frac{1}{2} \sum_{i=1}^{N} Q_i \psi_{\text{reac}}(x_i) & P, \\ G_{\text{elec}}^{\text{LPB}}[\Gamma] = \frac{1}{2} \sum_{i=1}^{N} Q_i \psi_{\text{reac}}(x_i) & \text{LPB}, \end{cases}$$

$$\begin{cases} G_{\text{elec}}^{\text{PB}}[\Gamma] = \frac{1}{2} \sum_{i=1}^{N} Q_i \psi_{\text{reac}}(x_i) - \frac{1}{2} \int_{\Omega_w} \sum_{j=1}^{M} q_i c_j^{\infty} \psi e^{-\beta q_j \psi} dV - \beta^{-1} \int_{\Omega_w} \sum_{j=1}^{M} c_j^{\infty} (e^{-\beta q_j \psi} - 1) dV \quad \text{PB.} \end{cases}$$

Note that both the CFA and Poisson's equation treat the solvent without ions. So, when ions are present, the functions ψ_{reac} in $G_{\text{elec}}^{\text{P}}[\Gamma]$ and $G_{\text{elec}}^{\text{LPB}}[\Gamma]$ are different, as explained below. Note also that $G_{\text{elec}}^{\text{PB}}[\Gamma] = G_{\text{elec}}^{\text{LPB}}[\Gamma] + O(\psi^2)$, if all $|\beta q_j \psi| \ll 1$ and if the bulk charge neutrality $\sum_{j=1}^{M} q_j c_j^{\infty} = 0$ holds true.

In these formulations, the constant ε_0 is the vacuum permittivity, the function $\psi_{\text{reac}} = \psi - \psi_{\text{ref}}$ is the reaction field, $\psi = \psi(x)$ is the electrostatic potential for the solvated state, and $\psi_{\text{ref}} = \psi_{\text{ref}}(x)$ is the potential for the reference state

$$\psi_{\rm ref}(x) = \sum_{i=1}^{N} \frac{Q_i}{4\pi\varepsilon_0\varepsilon_{\rm m}|x-x_i|}$$

For the PB or LPB formulations, we assume that there are *M* ionic species in the solvent, and denote by c_j^{∞} and q_j the bulk concentration and charge, respectively, of the *j*th species. As usual, $\beta^{-1} = k_B T$, where k_B is the Boltzmann constant and *T* is the absolute temperature. In the PB formulation, the potential $\psi = \psi(x)$ solves the boundary-value problem of dielectric PB equation

$$\begin{cases} -\varepsilon_{0}\varepsilon_{\mathrm{m}}\nabla^{2}\psi = \sum_{i=1}^{N}Q_{i}\delta_{x_{i}} \quad \text{in }\Omega_{\mathrm{m}}, \\ -\varepsilon_{0}\varepsilon_{\mathrm{w}}\nabla^{2}\psi = \sum_{j=1}^{M}q_{j}c_{j}^{\infty}e^{-\beta q_{j}\psi} \quad \text{in }\Omega_{\mathrm{w}}, \\ \llbracket\psi\rrbracket = 0 \quad \text{on }\Gamma, \\ \llbracket\varepsilon\frac{\partial\psi}{\partial n}\rrbracket = 0 \quad \text{on }\Gamma, \\ \psi = \psi_{0} \quad \text{on }\partial\Omega. \end{cases}$$
(II.3)

Here $\llbracket u \rrbracket = u|_{\Omega_{\mathbf{u}}} - u|_{\Omega_{\mathbf{m}}}$ denotes the jump across the interface Γ of a function u from $\Omega_{\mathbf{m}}$ to $\Omega_{\mathbf{w}}$, and $\varepsilon = \varepsilon_{\mathbf{m}}$ in $\Omega_{\mathbf{m}}$ and $\varepsilon = \varepsilon_{\mathbf{w}}$ in $\Omega_{\mathbf{w}}$. We take the boundary value ψ_0 to be the Yukawa potential

$$\psi_0(x) = \sum_{i=1}^{N} \frac{Q_i e^{-\kappa |x-x_i|}}{4\pi \varepsilon_0 \varepsilon_{\mathbf{w}} |x-x_i|},$$

where $\kappa = \sqrt{\sum_{j=1}^{M} c_j^{\infty} q_j^2 / \varepsilon_0 \varepsilon_w k_B T}$ is the inverse Debye length. In $G_{\text{elec}}^{\text{PB}}[\Gamma]$ and $G_{\text{elec}}^{\text{LPB}}[\Gamma]$, the potentials ψ that define $\psi_{\text{reac}} = \psi - \psi_{\text{ref}}$ solve the corresponding boundary-value problem of Poisson's equation and the linearized PB equation, respectively. They are exactly the same as (II.3), except that the second equation in (II.3) is replaced by $\nabla^2 \psi = 0$ in Ω_w and $-\nabla r^2 \psi + \kappa^2 \psi = 0$ in Ω_w , respectively.

B. Effective Boundary Force

The negative first variation of the VISM solvation free-energy functional $G = G[\Gamma]$ with respect to the location change of the dielectric boundary Γ defines the (normal component of) effective boundary force on Γ . It is a function defined on the boundary Γ , and is given by ^{17,55,58–61}

$$F_{n}(\mathbf{x}) = -\delta_{\Gamma}G[\Gamma] = -\Delta P - 2\gamma_{0}[H(\mathbf{x}) - \tau K(\mathbf{x})] + \rho_{w} \sum_{i=1}^{N} U_{i}(|\mathbf{x} - \mathbf{x}_{i}|) - \delta_{\Gamma}G_{\text{elec}}[\Gamma](\mathbf{x}) \quad \text{for } \mathbf{x} \in \Gamma, \quad \overset{(\text{III})}{4\gamma_{0}} = -\delta_{\Gamma}G[\Gamma](\mathbf{x}) - \delta_{\Gamma}G_{\text{elec}}[\Gamma](\mathbf{x}) - \delta_{\Gamma}G_{\text{elec}}[\Gamma](\mathbf{x}) - \delta_{\Gamma}G_{\text{elec}}[\Gamma](\mathbf{x}) = -\delta_{\Gamma}G[\Gamma](\mathbf{x}) - \delta_{\Gamma}G_{\text{elec}}[\Gamma](\mathbf{x}) - \delta_{\Gamma}G_$$

where $K(\mathbf{x})$ is the Gaussian curvature (i.e. the product of the two principal curvatures) at a point **x** on Γ , and

 $-\delta_{\Gamma}G_{\text{elec}}[\Gamma]$

(0

$$\begin{pmatrix} 0 & \text{Nonpolar System,} \\ -\delta G_{\text{elec}}^{\text{CFA}}[\Gamma] = \frac{1}{32\pi^2 \varepsilon_0} \left(\frac{1}{\varepsilon_{\text{w}}} - \frac{1}{\varepsilon_{\text{m}}} \right) \left| \sum_{i=1}^{N} \frac{Q_i(\mathbf{x} - \mathbf{x}_i)}{|\mathbf{x} - \mathbf{x}_i|^3} \right|^2$$
 CFA,

$$= \begin{cases} -\delta_{\Gamma} G_{\text{elec}}^{\text{P}}[\Gamma] = \frac{\varepsilon_{0}}{2} \left(\frac{1}{\varepsilon_{\text{w}}} - \frac{1}{\varepsilon_{\text{m}}}\right) \left(\varepsilon \frac{\partial \psi}{\partial n}\right)^{2} + \frac{\varepsilon_{0}}{2} (\varepsilon_{\text{m}} - \varepsilon_{\text{w}}) |(I - \mathbf{n} \otimes \mathbf{n}) \nabla \psi|^{2} \\ -\delta_{\Pi} G_{\text{LPB}}^{\text{LPB}}[\Gamma] - \frac{\varepsilon_{0}}{2} \left(\frac{1}{\varepsilon_{\text{w}}} - \frac{1}{\varepsilon_{\text{m}}}\right) \left(\varepsilon \frac{\partial \psi}{\partial n}\right)^{2} + \frac{\varepsilon_{0}}{2} (\varepsilon_{\text{m}} - \varepsilon_{\text{w}}) |(I - \mathbf{n} \otimes \mathbf{n}) \nabla \psi|^{2} \end{cases}$$

LPB,

Ρ,

(II.

5)

$$= \delta_{\Gamma} G_{\text{elec}} \left[\Gamma \right] = \frac{\varepsilon_0}{2} \left(\frac{\varepsilon_w}{\varepsilon_w} - \frac{\varepsilon_m}{\varepsilon_m} \right) \left(\varepsilon \frac{\partial \psi}{\partial n} \right)^2 + \frac{\varepsilon_0}{2} (\varepsilon_m - \varepsilon_w) |(I - \mathbf{n} \otimes \mathbf{n}) \nabla \psi|^2 - \beta^{-1} \sum_{i=1}^M c_j^{\infty} (e^{-\beta q_j \psi} - 1)$$
 PB

formulations are different when ions are present. The effective boundary force will be used in our numerical computation.

III. REVIEW OF NUMERICAL METHODS

A. Level-set method

We employ the level-set method to minimize the free-energy functional (II.1). We begin with an initial surface that encloses all the solute atoms. The free energy of such an initial surface can be very large. We then move the surface in the direction of steepest descent of the free energy until a steady state is reached. To use the level-set method, we first represent a surface Γ as the zero level set of a level-set function $\varphi(\mathbf{x})$: $\Gamma = \{\mathbf{x} : \varphi(\mathbf{x}) = 0\}$. With this level-set function $\varphi = \varphi(\mathbf{x})$, the unit normal $\mathbf{n}(\mathbf{x})$, the mean curvature $H(\mathbf{x})$, and the Gaussian curvature $K(\mathbf{x})$ at a point \mathbf{x} on the surface can be readily expressed as $\mathbf{n} = \nabla \varphi / \nabla \varphi |, H = (1/2)$ $\nabla \cdot \mathbf{n}$, and $K = \mathbf{n} \cdot \operatorname{adj} (\nabla^2 \varphi) \mathbf{n}$, respectively. Here $\nabla^2 \varphi$ is the Hessian matrix of the function φ with entries being the second order partial derivatives $\partial_{ij}^2 \phi$ of the level-set function φ , and adj ($\nabla^2 \varphi$) is the adjoint matrix of the Hessian $\nabla^2 \varphi$. With t denoting time, the motion of a moving surface $\Gamma = \Gamma(t)$ is then tracked by the evolution of the level-set function $\varphi = \varphi(\mathbf{x}, t)$ whose zero level-set is $\Gamma(t)$ at each t. Such evolution is governed by the level-set equation

$$\frac{\partial \phi}{\partial t} + F_n |\nabla \phi| = 0,$$
 (III.1)

where the function $F_n = F_n(\mathbf{x})$ is the effective boundary force (II.4). Note that the boundary force (II.4) is defined on the interface Γ . To solve the Eq. (III.1), we need to extend this force away from the surface to the whole computational box. To accelerate the relaxation, we implement a local level-set method in which we extend the boundary force (II.4) to a

narrow band surrounding the surface. We then solve the level-set equation in the narrow band with the zero Neumann boundary condition.

Our level-set method is an optimization method of the steepest descent type. Different initial surfaces can relax to different local minimizers, due to the nonconvexity of the free-energy functional. Such local minimizers correspond to polymodal hydration states. In order to capture different local minimizers, we usually use two types of initial surfaces: a tight wrap that is a union of the surfaces of vdW spheres centered at solute atoms with reduced radii, and a loose wrap that is a large surface loosely enclosing all the solute atoms. In our package, for instance, we choose the loose initial as a box that contains all the solute atoms.

To discretize the level-set equation (III.1), we rewrite it as 38-40,42,43,46

$$\frac{\partial \phi}{\partial t} = -F_n |\nabla \phi| = A + B |\nabla \phi|,$$

where

$$A(\mathbf{x}) = 2\gamma_0 [H(\mathbf{x}) - \tau K(\mathbf{x})] |\nabla \phi(\mathbf{x})|,$$

$$B(\mathbf{x}) = \Delta P - \rho_w U(\mathbf{x}) + \delta_\Gamma G_{\text{elec}}(\mathbf{x}).$$

The term $A = A(\varphi)$ is first linearized with respect to φ at a previous time step and the parameter τ is then adjusted to enforce parabolicity of the linearized equation. The spatial derivatives in *A* are discretized by the central differencing. We discretize the hyperbolic term $B | \nabla \varphi |$ by an upwind scheme. In our package, we use a fifth-order WENO (weighted essential-no-oscillation) scheme⁶². The discretization of the electrostatic force $\delta_{\Gamma}G_{\text{elec}}(\mathbf{x})$ is detailed in the next section.

To discretize the time derivative in the level-set equation (III.1), we use the forward Euler method

$$\frac{\phi^{(k+1)}(\mathbf{x}) - \phi^{(k)}(\mathbf{x})}{\Delta t} = -F_n^{(k)}(\mathbf{x}) |\nabla \phi^{(k)}(\mathbf{x})|, \quad \text{(III.2)}$$

where $\varphi^{(k)}(\mathbf{x})$ and $F_n^{(k)}(\mathbf{x})$ denote the numerical approximation of $\varphi(\mathbf{x}, t_k)$ and $F_n(\mathbf{x}, t_k)$, respectively, at time $t_k = k$ t (k = 1, 2, ...) with time step size t. In our package, we use the local level-set method⁶³ and update the level-set equation φ by (III.2) in a narrow band surrounding the surface, together with a Neumann boundary condition. To satisfy the Courant–Friedrichs–Lewy (CFL) condition⁶⁴, we choose

$$\Delta t = \frac{0.5h}{\max_{\mathbf{x}} [\operatorname{Trace}\left(C(\phi(\mathbf{x}))\right)/h + |B_1(\mathbf{x})|]}, \quad \text{(III.3)}$$

$$B_1(\mathbf{x}) = \Delta P + \rho_{\mathbf{w}} U(\mathbf{x}) + \delta_{\Gamma} G_{\text{elec}}(\mathbf{x}).$$

The maximum in (III.3) is taken over all the grid points in the narrow band.^{42,43}

After the level-set function evolves for several steps, we reinitialize the updated level-set function φ by solving

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

where φ_0 is the level-set function before reinitialization, sign (φ_0) represents the sign of φ_0 , and time *t* is different from the one in the original level-set equation. We start from the initial value φ_0 at *t* = 0 and solve the equation with a few iterations that often lead to a steady state.

B. Numerical Method for Electrostatics

We briefly review the numerical methods for the nonlinear PB equation. The Poisson equation and the linearized PB equation can be solved in the same way. The CFA does not require solving any equation; cf. (II.5). To avoid singularity arising from the point charges, we solve the following equations for the reaction field ψ_{reac} that can be derived from (II.3) and the corresponding equation for ψ_{ref}^{20} :

$$\begin{cases} -\nabla^{2}\psi_{\text{reac}} = 0 \quad \text{in } \Omega_{\text{m}}, \\ -\varepsilon_{0}\varepsilon_{\text{w}}\nabla^{2}\psi_{\text{reac}} = \sum_{j=1}^{M}q_{j}c_{j}^{\infty}e^{-\beta q_{j}(\psi_{\text{reac}} + \psi_{\text{ref}})} \quad \text{in } \Omega_{\text{w}}, \\ \llbracket\psi_{\text{reac}}\rrbracket = 0 \quad \text{on } \Gamma, \\ \llbracket\varepsilon\frac{\partial\psi_{\text{reac}}}{\partial n}\rrbracket = (\varepsilon_{\text{m}} - \varepsilon_{\text{w}})\frac{\partial\psi_{\text{ref}}}{\partial n} \quad \text{on } \Gamma, \\ \psi_{\text{reac}} = \psi_{0} - \psi_{\text{ref}} \quad \text{on } \partial\Omega. \end{cases}$$
(III.4)

This nonlinear equation is solved by Newton's iteration

$$\begin{array}{l} & -\nabla^2 \psi_{\text{reac}}^{(l+1)} = 0 \quad \text{in} \, \Omega_{\text{m}}, \\ & -\varepsilon_0 \varepsilon_{\text{w}} \nabla^2 \psi_{\text{reac}}^{(l+1)} + \left(\sum_{j=1}^M \beta q_j^2 c_j^\infty e^{-\beta q_j(\psi_{\text{reac}}^{(l)} + \psi_{\text{ref}})} \right) \, \psi_{\text{reac}}^{(l+1)} = \sum_{j=1}^M q_j c_j^\infty (1 + \beta q_j \psi_{\text{reac}}^{(l)}) \, e^{-\beta q_j(\psi_{\text{reac}}^{(l)} + \psi_{\text{ref}})} & \text{in} \, \Omega_{\text{w}}, \\ & \llbracket \psi_{\text{reac}}^{(l)} \rrbracket = 0 \quad \text{on} \, \Gamma, \\ & \llbracket \varepsilon \frac{\partial \psi_{\text{reac}}^{(i)}}{\partial n} \rrbracket = (\varepsilon_{\text{m}} - \varepsilon_{\text{w}}) \frac{\partial \psi_{\text{ref}}^{(l)}}{\partial n} & \text{on} \, \Gamma, \\ & \psi_{\text{reac}}^{(l)} = \psi_0 - \psi_{\text{ref}} & \text{on} \, \partial\Omega, \end{array}$$

where l is the iteration step. Each iteration amounts to solving an elliptic interface problem

$$\begin{cases} -\nabla^2 u = 0 \quad \text{in } \Omega_{\rm m}, \\ -\varepsilon_0 \varepsilon_{\rm w} \nabla^2 u + bu = s \quad \text{in } \Omega_{\rm w}, \\ \llbracket u \rrbracket = f \quad \text{on } \Gamma, \\ \llbracket \varepsilon \frac{\partial u}{\partial n} \rrbracket = g \quad \text{on } \Gamma, \\ u = h \quad \text{on } \partial \Omega, \end{cases}$$
(III.5)

where b, s, f, g, and h are given functions.

To numerically solve (III.5), we develop a compact coupling interface method (CCIM).^{46,65–67} This method is based on the coupling interface method (CIM)⁶⁵ and a compact scheme for irregular geometry.⁶⁸ We cover the whole computational box by a uniform finite-difference grid. We call a grid point a regular point if its six neighboring grid points have the same dielectric coefficient; otherwise, we call it an irregular point. We use the standard 7-point central differencing scheme to discretize the laplacian on regular points.

An irregular points needs to be treated carefully. In a one-dimension setting, the numerical solution on both sides of such a point are approximated by linear polynomials in the first-order CIM (CIM1) scheme and quadratic polynomials in the second-order CIM (CIM2) scheme. Coupled equations for first-order derivatives (CIM1) and second-order derivatives (CIM2) are derived from two jump conditions on the interface. In multi-dimensions, the derivatives are approximated dimension-by-dimension and the jump data are decomposed into normal and tangential directions to derive coupled equations with information coming from each coordinate. For CIM2, the cross derivatives are approximated by one-sided interpolations. The unknowns of the coupled equations are first-order derivatives in CIM1 and are second-order principal derivatives in CIM2.⁶⁵ The coupled equations are solved locally to obtain the discretization scheme. In the implementation, the CIM2 scheme is preferred if there are enough adjacent grid points that have the same dielectric coefficient as the discretization point. Adaptively using the CIM1 scheme and CIM2 scheme, depending on the local geometry of the interface, leads to a hybrid CIM scheme.

In CCIM, we further employ the jump data on the interface to achieve high-order aproximations on irregular points with complex geometry. Here, the first and second-order derivatives are treated as unknowns of the local coupled equations.⁶⁸ For instance, ψ_x , ψ_y , ψ_z , ψ_{xx} , ψ_{yy} , and ψ_{zz} are six unknowns at an irregular grid point in the coupled equations in three dimensions. With first-order derivatives available (as unknowns), the cross derivatives can be discretized with the same order of accuracy as the classical CIM2 but with much fewer grid points. The compactness makes the high-order discretization possible at many irregular points where the classical CIM2 fails. The resulting stencil is a seven-point stencil plus the grid points involved in calculating cross derivatives. To derive six coupled equations for those six unknowns, we need the following jump data along the tangential direction on the interface:

$$abla f=0, \quad
abla^2 f=0, \quad
abla g=-(arepsilon_{
m w}-arepsilon_{
m m}) \left(
abla^2 \psi_{
m ref} \mathbf{n}+
abla \mathbf{n}
abla \psi_{
m ref}
ight),$$

where the unit vector **n** and the matrix ∇ **n** are computed with the level-set function φ . Note that, if possible, we always prefer to use adjacent grid points that have the same dielectric coefficient as the discretization point under consideration. We store the grid points and their coefficients in the discretization stencil for the first derivatives so that we do not need to recompute them when we calculate the dielectric boundary force. Once the numerical solutions on the grid points are obtained, the first derivatives on irregular points can be calculated efficiently by the stored information.

We solve the resulting linear system by the biconjugate gradient stabilized (BiCGStab) method using incomplete lower-upper factorization as a preconditioner. Our numerical tests show that the preconditioned BiCGStab method is more efficient than the algebraic multi-grid method for the scale of the problem under consideration. With obtained solutions on the computational grid, we approximate ψ and ψ on the interface using a high order interpolation scheme.⁴⁶ The dielectric boundary force can then be calculated according to (II.5).

C. Energy evaluation

To evaluate the surface integrals and volume integrals in the expression of solvation free energy, we employ the following two formulas

$$\begin{split} &\int_{\Omega_{\Gamma}} g dS {=} \int_{\mathbb{R}^3} g |\nabla \phi| \delta(\phi) dV, \\ &\int_{\Omega_{\mathrm{m}}} f dV {=} \int_{\mathbb{R}^3} f(1{-}\mathrm{Heav}(\phi)) \, dV, \end{split}$$

respectively. Here Heav(·) and δ (·) are the Heaviside function and Dirac δ -function, respectively. We use the numerical scheme developed in the work⁶⁹ to approximate the δ -function. For the integration of the LJ potential over an infinite region, we use a stable, efficient numerical integration technique developed in the work.⁴² To compute the electrostatic solvation energy corresponding to the charged solute atoms, i.e.,

 $(1/2)\sum_{i=1}^{N}Q_i\psi_{\text{reac}}(x_i)$, we first use the trilinear interpolation to approximate the reaction potential at atom positions and then sum the products.

We remark that, for the LJ parameters adopted from a commonly used water model for charged molecular systems, we often employ a parallel shift of the VISM surface toward the solute region by a fitting parameter ξ (in Å) when we calculate the electrostatic part of the solvation free energy. Our previous studies show that a value of ξ close to 1 Å works quantitatively well. Such a shift accounts for the asymmetric effect of the structure of water molecules, which is ignored in implicit water models.

D. Algorithm

Step 0. Input all the parameters P, γ_0 , τ , ρ_w , ε_0 , ε_m , ε_w , and atomic parameters \mathbf{x}_i , ε_i ; σ_i , and Q_i for all i = 1, ..., N. Choose a computational box according to the atom coordinates and discretize the box uniformly with the prescribed computational grid size. Prescribe the electrostatic description and the initial surface. Precompute on the computational grid points

$$\rho_{\mathbf{w}} \sum_{i=1}^{N} U_i(|\mathbf{x}-\mathbf{x}_i|) \quad \text{and} \quad \frac{1}{32\pi^2 \varepsilon_0} \left(\frac{1}{\varepsilon_{\mathbf{w}}} - \frac{1}{\varepsilon_{\mathbf{m}}}\right) \left| \sum_{i=1}^{N} \frac{Q_i(\mathbf{x}-\mathbf{x}_i)}{|\mathbf{x}-\mathbf{x}_i|^3} \right|^2.$$

Generate an initial surface and initialize the corresponding level-set function. Let k = 1 and start the relaxation.

Step 1. Locate a narrow band that is centered at with a width of 12 to 16 grid points. Compute the gradient φ , the Hessian $\nabla^2 \varphi$, and the curvatures *H* and *K* using central differencing schemes at each grid point in the band.

Step 2. Compute the free energy (II.1).

Step 3. Compute and extend the boundary force F_n by (II.4).

Step 4. Determine *t* according to (III.3) and update the level-set function using the Euler scheme (III.2).

Step 5. Reinitialize the level-set function φ .

Step 6. Check the stopping criterion. Stop if the zero level-set converges or the number of iterations exceeds the prescribed maximum running steps. Otherwise, locate the surface Γ by the current level-set function, set $k \leftarrow k + 1$, and go back to Step 1.

IV. SOFTWARE

In this section, we describe the package structure, the flow chart of the package, the format of input and output files, and the post-processing of output data. All of the codes are written in C++.

A. Structure Files

The package includes the following files and directories:

- 1. vism.out: An executable file for the LS-VISM package.
- **2.** visualization: This folder contains a Matlab subroutine, plotsurf.m, to visualize the final surface.
- 3. examples: Some examples illustrate the application of the package to some test cases. In each example, there are two input files: readin and atominfo (cf. Subsection IV C). Also, there is a Matlab script visual.m, a function used for calling the subroutines in the visualization folder (cf. Appendix D). If an initial surface is provided (initialstatus=-1), then there is an input file for the level-set function of the initial surface. The name of the file is provided by the input variable initialsurf in the readin file.

B. Flow chart

Figures 2 and 3 display the work flow of the LS-VISM package. After the LS-VISM gets started, it first goes to the process of Initialization. In this process, the code first reads in the parameters in the readin file (cf. Table IV.1) and the atominfo file. Based on the coordinates

of the solute atoms, the geometric center of the biomolecule is moved to the origin and the computational box is determined accordingly. For the case of a loose initial (initialstatus=0), we first use a coarse grid (e.g., $30 \times 30 \times 30$) to relax the VISM free-energy functional with CFA algorithm. We then generate an initial guess by trilinear interpolation for the fine-grid relaxation. For the case of tight initial (initialstatus=1), the initial surface is a union of spheres generated by each atom with reduced vdW radii. For the last case (initialstatus=-1), the code resumes iterations from a previously calculated surface that is determined by the variable initialsurf. After generating the initial surface, the code starts LS relaxation, which is shown in Figure 3. The algorithm "AlgNtrl" treats nonpolar systems. The algorithms "AlgCFA", "AlgP", "AlgLPB", and "AlgPB" calculate the electrostatics using the CFA, Poisson's equation, linearized PB equation, and fully nonlinear PB equation, respectively. The algorithm "AlgCFA" is also used in the first part of the level-set relaxation for charged systems.

C. Input and output file format

The current version of our package has two input files: readin and atominfo, and three output files: Energies.dat, surf.dx, and surf.dat.

The file readin contains the parameters used in the package. These parameters are summarized in Table IV.1. The file atominfo contains solute atomic coordinates, partial charges, and force-field parameters. They are listed in the order: Atom index, x coordinate, y coordinate, z coordinate, LJ- σ_i , LJ- ε_i , charge value Q_i . A sample atominfo file is given in Appendix A.

The first output file is Energies.dat. It lists the values of total solvation energy and that of surface energy, solute-solvent vdW interaction energy, and electrostatic solvation energy, for all steps of the level-set relaxation. The other two files contain basic computational box, grid information, and an array representing the 3D level-set function of the final surface. One of them is the file surf.dat that can be processed by Matlab subroutines written for visualization. The other one is the file surf.dx that can be imported to VMD for visualization. See the hydration surface plots in Subsection IV D and Section V.

D. Visualization

Our package provides two methods to visualize the final equilibrium hydration surface. One is implemented in a Matlab function, plotsurf, in which the zero level-set surface is first interpolated using the level-set function and then the surface is colored according to the value of mean curvature *H*. Different colors indicate the local concavity and convexity of the hydration surface. See typical results in the top row of Figures reff:CB7surf and 5. The other is using the VMD software.⁷⁰ The volumetric 3D level-set function stored in surf.dx is used to construct the VISM molecular surface using the isosurface representation with isovalue 0. See typical results in the bottom row of Figures 4 and 5.

V. APPLICATION DEMO: A HOST-GUEST SYSTEM

To illustrate how to use the LS-VISM package to conduct biomolecular solvation analysis, we present an example of a host-guest system: a Bicyclo[2.2.2]octane (B2) guest binding to

a Cucurbit[7]uril (CB[7]) host. This host-guest system has recently attracted extensive attention due to its ultrahigh binding affinity.^{71–77} It has been well tested in our recent works with the electrostatics described by the CFA and the PB theory.^{45,46} Here we show the typical procedures of computing binding affinities for two binding partners with our LS-VISM package.

We first consider hydration behavior of the CB[7]-B2. We run the software package for the three cases with loose initials and tight initials. Once the input files, readin and atominfo, are ready, it is simple to get the LS-VISM package running with the command line

./vism.out

Descriptions are printed in the terminal to display the running progress of the package. When the calculations are done, we can find in Energies.dat the history of the total solvation free energies and corresponding contributions from each component in the VISM freeenergy functional. See Table V.1 for the results on the host-guest system. Note that we employ a boundary shift of the equilibrium surface by 1 Å towards the solute region when we calculate the electrostatic part of the solvation energy. See our previous studies^{44–46} for more details. For visualization, we use a Matlab function visual.m (cf. Appendix D) to call the function plotsurf to plot the zero level-set surface by running the following command in Matlab:

figure; visual(`surf.dat');

The surface can also be displayed by using VMD with the isosurface representation. The surface is set to be transparent to reveal the relative location of the hydration surface to the atoms in biomolecules. See Figures 4 and 5 for the hydration surface of CB[7], B2, and the bound state of the host-guest system, respectively.

From Table V.1 and the hydration surfaces, we observe that the host molecule presents two hydration states: a wet state and a dry state. In the dry state, water molecules cannot penetrate into the toroidal center of the host molecule; in the wet state, the host cavity is completely hydrated. Such polymodal hydration states have been previously observed in concave hydrophobic pockets due to capillary evaporations. We can see from the table that the wet state, resulting from a tight initial, has lower solvation energy, indicating that the wet state is a more favorable state for the CB[7] molecule. For B2 and the bound state of the host-guest system, both loose and tight initials lead to the same equilibrium hydration surface.

The binding affinity is given by⁷⁸

$$\Delta G_{\rm B} = \Delta G_{\rm geom}[\Gamma] + \Delta G_{\rm vdW}[\Gamma] + \Delta G_{\rm elec}[\Gamma] + \Delta G_{\rm vdW}^{R} + \Delta G_{\rm elec}^{R} + \Delta G_{\rm TS} + \Delta G_{\rm Val}, \quad (V.1)$$

where " " means the energy difference between a bound state and an unbounded state. In an unbound state, the host and guest are treated separately. The first three terms come from the VISM free-energy functional (II.1), the fourth and fifth terms represent the contributions from the vdW and Coulombic interaction energies between two binding partners in a reference state, and the last two terms are energy differences upon binding due to the entropy and valence terms (bond-stretch, angle-bend, dihedral, etc.).

From Table V.1, we can compute the first three terms in (V.1). For the geometric part, both calculations with loose initial and tight initial predict favorable contributions, -59.9 kJ/mol and -92.9 kJ/mol, respectively, due to the reduction of water-accessible area upon binding. The vdW interaction between water and solutes shows an unfavorable energy differences: 34.2 kJ/mol for the dry state and 57.7 kJ/mol for the wet, since the favorable water-solute interaction is suppressed upon binding, especially for the wet state. The electrostatic part of the solvation disfavors binding, with an energy penalty of 10.4 kJ/mol for the dry case and 35.7 kJ/mol for the wet case. The fourth and fifth terms in (V.1) which are independent of hydration states, can be readily computed: -95.8 kJ/mol for vdW interactions and -30.2 kJ/mol for Coulombic interactions. Note that the two binding partners are treated as rigid bodies in our LS-VISM package. With the current version of VISM, we are unable to estimate the remaining terms. To complete our prediction of binding affinity, we take the values from the work⁷⁸: 72.8 kJ/mol and 5.0 kJ/mol for entropy penalty and valence energy differences, respectively. Adding these components together, we predict a favorable binding affinity: -63.6 kJ/mol for the dry case and -47.8 kJ/mol for the wet case. We see that our calculations agree very well with the results of -50.3 kJ/mol predicted by M2 algorithm^{75,79} and 56.0 kJ/mol of the experimental data.78

VI. CONCLUSIONS

We have introduced a software package for the static analysis of the solvation of biomolecules and molecular binding in water. The input data for the package include mainly the atomic coordinates and partial charges of solute atoms, the LJ parameters in a force-field, the solute-solvent surface tension, the bulk water density and bulk ionic concentrations, and some initial guesses of solute-solvent interfaces. The output includes the location of solute-solvent interface, an approximate value of the solvation free energy, and an estimate of the binding free energy for molecular binding. In addition, all the components of the solvation free energy—the solute-solvent vdW interaction energy, the solute-solvent interfacial energy, and the electrostatic energy—can be obtained. We have detailed the structure of our software, the format of input and output files, and the post-processing of output data. We have also presented a demo to show how to use the package.

Our software package collects computer codes that implement a variational implicit-solvent model (VISM). The VISM couples the geometrical, electrostatic, and solute-solvent van der Waals (vdW) interactions, all expressed through the solute-solent interface (i.e., the dielectric boundary) into a continuum approximation of the solvation free energy of an underlying biomolecular system. The Coulomb-field approximation, dielectric Poisson's equation, and the dielectric Poisson–Boltzmann equation (linear or nonlinear) are all implemented in the package to describe the electrostatic interactions. Our computational

methods include mainly the level-set method for the gradient geometrical flow that relaxes the VISM solvation free-energy functional of solute-solvent interfaces, a robust and highly accurate compact coupling interface method (CCIM) for the continuum dielectric electrostatics, and various integration techniques for the evaluation of solvation free energies.

Several aspects of our current VISM software need to be improved. Let us first discuss some issues in the theoretical improvement of VISM. As we reviewed, the final calculation of the electrostatic solvation free energy of an underlying biomolecular system uses a shifted dielectric boundary.^{44,46} This shift is designed to resolve the issue of charge asymmetry in the continuum description of electrostatics,^{80–82} which results from using the force-field parameters in many of the current water models. But it is inconsistent with the variational principle of free-energy minimization. One remedy then is to re-parameterize all the solutewater LJ parameters. Another is to consider solute-solvent interfacial conditions for the Poisson or PB treatment.⁸³ A second theoretical issue is the estimate of entropy. This is completely missing in our current theory. Calculations of derivatives with respect to the VISM parameters, such as the temperature of the surface tension, bulk solvent density, and the dielectric coefficients, may not provide accurate estimates of the hydration entropy. Conformations need to be sampled. Such sampling can be possibly done through the VISM calculations of the potential of mean force in the space of relative molecular positions and orientations. The last issue for improvement is the description of solute-solvent interfacial fluctuations that contribute crucially to the free energy. We are currently developing a hybrid method that combines our level-set method with a surface Monte Carlo method to treat such fluctuations.

Let us now discuss several issues related directly to our computational package. First, to use our package in its current version, users need to prepare the input file atominfo. The force-field parameters in such a file can be obtained from molecular dynamics (MD) simulation software, such as CHARMM, AMBER, and GROMOS. A program that converts such force-field parameters in the MD software to our input file will be very useful. This will be considered in our next improvement of the package. Second, we still have the efficiency issue. Our level-set calculation usually takes minutes to hours. The computational time depends on numerical resolutions, system size, models of electrostatics to use, and many other factors. The expensive part of our computation is the numerical solution process for solving the level-set equation and the PB equation. It is desirable then to develop a fast linear system solver, such as an algebraic multigrid solver, for these equations. We are also considering the use of GPU computing to improve the efficiency of our package. Finally, we are working on the development of a website to maintain the package can possibly include the efficient identification of ligand binding pockets on protein surfaces.⁴⁸

Acknowledgments

This work was supported by the NSF through grant DMS-1319731 (L.-T.C. and B.L.), and the NIH through the grant R01GM096188 (L.-T.C., J.C., B.L., and J.A.M.). Work in the McCammon group is supported in part by NSF, NIH, HHMI, and NBCR. J.D. acknowledges support by the Deutsche Forschungsgemeinschaft (DFG). The authors

thank Dr. Michael Gilson, Dr. Zuojun Guo, Dr. Zhongming Wang, and Dr. Yanxiang Zhao for many helpful discussions.

References

- 1. Levy Y, Onuchic J. Ann Rev Biophys Biomol Struct. 2006; 35:389. [PubMed: 16689642]
- 2. Baron R, Mccammon JA. Annu Rev Phys Chem. 2013; 64:151. [PubMed: 23473376]
- McCammon, JA.; Harvey, SC. Dynamics of proteins and nucleic acids. Cambridge University Press; 1987.
- 4. Leach, A. Molecular Modeling: Principles and Applications. Prentice Hall, Inc; 2001.
- Frenkel, D.; Smit, B. Understanding Molecular Simulation: From Algorithms to Applications. Academic Press; 1996.
- 6. Allen, MP.; Tildesley, DJ. Computer Simulation of Liquids. Clarendon Press; 1987.
- 7. Rapaport, DC. The art of molecular dynamics simulations. 2. Cambridge Univ. Press; 2004.
- 8. Tomasi J, Persico M. Chem Rev. 1994; 94:2027.
- 9. Roux B, Simonson T. Biophys Chem. 1999; 78:1. [PubMed: 17030302]
- 10. Feig M, Brooks CL III. Curr Opin Struct Biol. 2004; 14:217. [PubMed: 15093837]
- 11. Baker NA. Curr Opin Struct Biol. 2005; 15:137. [PubMed: 15837170]
- 12. Fixman F. J Chem Phys. 1979; 70:4995.
- 13. Davis ME, McCammon JA. Chem Rev. 1990; 90:509.
- 14. Sharp KA, Honig B. Annu Rev Biophys Chem. 1990; 19:301.
- 15. Sharp KA, Honig B. J Phys Chem. 1990; 94:7684.
- Baker NA, Sept D, Joseph S, Holst MJ, McCammon JA. Proc Natl Acad Sci USA. 2001; 98:10037. [PubMed: 11517324]
- 17. Che J, Dzubiella J, Li B, McCammon JA. J Phys Chem B. 2008; 112:3058. [PubMed: 18275182]
- 18. Gouy M. J Phys Théor Appl. 1910; 9:457.
- 19. Chapman DL. Philos Mag. 1913; 25:475.
- 20. Li B. SIAM J Math Anal. 2009; 40:2536.
- 21. Fogolari F, Briggs JM. Chem Phys Lett. 1997; 281:135.
- 22. Grochowski P, Trylska J. Biopolymers. 2008; 89:93. [PubMed: 17969016]
- 23. Born M. Z Phys. 1920; 1:45.
- 24. Bashford D, Case DA. Ann Rev Phys Chem. 2000; 51:129. [PubMed: 11031278]
- 25. Lee B, Richards FM. J Mol Biol. 1971; 55:379. [PubMed: 5551392]
- 26. Richards FM. Annu Rev Biophys Bioeng. 1977; 6:151. [PubMed: 326146]
- 27. Connolly ML. J Appl Cryst. 1983; 16:548.
- 28. Richmond TJ. J Mol Biol. 1984; 178:63. [PubMed: 6548264]
- 29. Connolly ML. J Mol Graphics. 1992; 11:139.
- 30. Chen J, Brooks CL III. J Am Chem Soc. 2007; 129:2444. [PubMed: 17288425]
- 31. Berne BJ, Weeks JD, Zhou R. Annu Rev Phys Chem. 2009; 60:85. [PubMed: 18928403]
- 32. Chandler D. Nature. 2005; 437:640. [PubMed: 16193038]
- 33. Zhou R, Huang X, Margulis CJ, Berne BJ. Science. 2004; 305:1605. [PubMed: 15361621]
- 34. Wang L, Friesner RA, Berne BJ. Faraday Disc. 2010; 146:247.
- 35. Hua L, Zangi R, Berne BJ. J Phys Chem C. 2009; 113:5244.
- Dzubiella J, Swanson JMJ, McCammon JA. Phys Rev Lett. 2006; 96:087802. [PubMed: 16606226]
- Dzubiella J, Swanson JMJ, McCammon JA. J Chem Phys. 2006; 124:084905. [PubMed: 16512740]
- Cheng LT, Dzubiella J, McCammon JA, Li B. J Chem Phys. 2007; 127:084503. [PubMed: 17764265]

- 39. Cheng LT, Xie Y, Dzubiella J, McCammon JA, Che J, Li B. J Chem Theory Comput. 2009; 5:257. [PubMed: 20150952]
- 40. Cheng LT, Wang Z, Setny P, Dzubiella J, Li B, McCammon JA. J Chem Phys. 2009; 131:144102. [PubMed: 19831428]
- Setny P, Wang Z, Cheng LT, Li B, McCammon JA, Dzubiella J. Phys Rev Lett. 2009; 103:187801. [PubMed: 19905832]
- 42. Cheng LT, Li B, Wang Z. J Comput Phys. 2010; 229:8497. [PubMed: 22323839]
- Wang Z, Che J, Cheng LT, Dzubiella J, Li B, McCammon JA. J Chem Theory Comput. 2012; 8:386. [PubMed: 22346739]
- 44. Guo Z, Li B, Dzubiella J, Cheng LT, McCammon JA, Che J. J Chem Theory Comput. 2013; 9:1778. [PubMed: 23505345]
- 45. Zhou S, Rogers KE, de Oliveira CF, Baron R, Cheng LT, Dzubiella J, Li B, McCammon JA. J Chem Theory Comput. 2013; 9:4195. [PubMed: 24039554]
- 46. Zhou S, Cheng LT, Dzubiella J, Li B, McCammon JA. J Chem Theory Comput. 2014; 10:1454. [PubMed: 24803864]
- Guo Z, Li B, Dzubiella J, Cheng LT, McCammon JA, Che J. J Chem Theory Comput. 2014; 10:1302. [PubMed: 24803860]
- 48. Guo Z, Li B, Cheng L-T, Zhou S, McCammon JA, Che J. J Chem Theory Comput. 2014 submitted.
- 49. Wagoner JA, Baker NA. Proc Natl Acad Sci, USA. 2006; 103:8331. [PubMed: 16709675]
- 50. Chen J, CLB, Khandogin J. Curr Opin Struct Biol. 2008; 18:140. [PubMed: 18304802]
- 51. Bates PW, Chen Z, Sun YH, Wei GW, Zhao S. J Math Biol. 2009; 59:193. [PubMed: 18941751]
- 52. Chen Z, Baker NA, Wei GW. J Comput Phys. 2010; 229:8231. [PubMed: 20938489]
- 53. Berman H, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, Shindyalov I, Bourne P. Nucleic Acids Res. 2000; 28
- 54. Borgis D, Levy N, Marchi M. 2003; 119:3516.
- 55. Cheng HB, Cheng LT, Li B. Nonlinearity. 2011; 24:3215. [PubMed: 23853423]
- 56. Tolman RC. J Chem Phys. 1949; 17:333.
- 57. Zhou HX. J Chem Phys. 1994; 100:3152.
- 58. Li B, Cheng X-L, Zhang Z-F. SIAM J Applied Math. 2011; 71:2093. [PubMed: 24058212]
- 59. Cai Q, Ye X, Wang J, Luo R. Chem Phys Lett. 2011; 514:368. [PubMed: 22125339]
- 60. Cai Q, Ye X, Luo R. Phys Chem Chem Phys. 2012; 14:15917. [PubMed: 23093365]
- 61. Xiao L, Cai Q, Ye X, Wang J, Luo R. J Chem Phys. 2013; 139:094106. [PubMed: 24028101]
- 62. Jiang G, Shu C. J Comput Phys. 1996; 126:202.
- 63. Peng D, Merriman B, Osher S, Zhao H, Kang M. J Comput Phys. 1999; 155:410.
- 64. Courant R, Friedrichs K, Lewy H. Math Annalen. 1928; 100:32.
- 65. Chern IL, Shu YC. J Comput Phys. 2007; 225:2138.
- 66. Shu YC, Chern IL, Chang CC. J Comput Phys. 2014; 275:642.
- 67. Cheng LT, Li B, White M, Zhou S. SIAM J Appl Math. 2013; 73:594. [PubMed: 23885130]
- 68. Mayo A. SIAM J Numer Anal. 1984; 21:285.
- 69. Smereka P. J Comput Phys. 2006; 211:77.
- 70. Humphrey W, Dalke A, Schulten K. J Molec Graphics. 1996; 14:33.
- Liu S, Ruspic C, Mukhopadhyay P, Chakrabarti S, Zavalij PY, Isaacs S. J Am Chem Soc. 2005; 127:15959. [PubMed: 16277540]
- Rekharsky MV, Mori T, Yang C, Ko YH, Selvapalam N, Kim H, Sobransingh D, Kaifer AE, Liu S, Isaacs L, et al. Proc Natl Acad Sci USA. 2007; 104:20737. [PubMed: 18093926]
- 73. Moghaddam S, Inoue Y, Gilson MK. J Am Chem Soc. 2009; 131:4012. [PubMed: 19133781]
- 74. El-Barghouthi MI, Assaf KI, Rawashdeh AMM. J Chem Theory Comput. 2010; 6:984.
- 75. Chen W, Chang CE, Gilson MK. Biophys J. 2004; 87:3035. [PubMed: 15339804]
- 76. Muddana HS, Gilson MK. J Chem Theory Comput. 2012; 8:2023. [PubMed: 22737045]

- 77. Nguyen CN, Young TK, Gilson MK. J Chem Phys. 2012; 137:044101. [PubMed: 22852591]
- Moghaddam S, Yang C, Rekharsky M, Ko YH, Kim K, Inoue Y, Gilson MK. J Am Chem Soc. 2011; 133:3570. [PubMed: 21341773]
- 79. Chang CE, Gilson MK. J Am Chem Soc. 2004; 126:13156. [PubMed: 15469315]
- 80. Latimer WM, Pitzer KS, Slansky CM. J Chem Phys. 1939; 7:108.
- Mobley DL, Barber AE II, Fennell CJ, Dill KA. J Phys Chem B. 2008; 112:2405. [PubMed: 18251538]
- Mukhopadhyay A, Fenley AT, Tolokh IS, Onufriev AV. J Phys Chem B. 2012; 116:9776. [PubMed: 22762271]
- 83. Bardhan JP, Knepley MG. J Chem Phys. 2014; 141:131103. [PubMed: 25296776]

Appendix A. A Sample atominfo file

#This file contains the input parameters in the order: #ATOM index x y z coordinates LJ_sigma LJ_epsilon Partial Charges 1 3.107 1.17 -0.08 1.51453 0.49857 -0.527991 2 -1.964 0.877 2.917 1.51453 0.49857 -0.527991 ... 155 4.983 0.685 2.743 1.17599 0.09140 0.090441 156 5.143 3.355 3.663 0.677083 0.20691 0.394426 END

Appendix B. A Sample readin file

```
#Set the parameters in this file
#Geometric Biomolecule center will be moved to the origin
#Length unit: Angstrom, Energy unit: kJ/mol, Concentration unit: M
#Box size will be determined by atomic coordinates
#Charged status: 0: Nonpolar system
#Charged status: 1: Coulomb Field approximation
#Charged status: 2: Poisson's equation, without ionic solutions
#Charged status: 3: Linearized Poisson--Boltzmann equation
#Charged status: 4: Nonlinear Poisson--Boltzmann equation
#Initial surface: 0: Loose initial surface
#Initial surface: 1: Tight initial surface
#Initial surface: -1: Resume from a given surface
estat 2 #Nonpolar:0, CFA:1, Poisson:2, LPB:3, NPB:4
runsteps 1000 #Maximum running steps, default 10000
watLJsig 1.5768 #LJ parameter of water: Sigma (Angstrom)
watLJeps 0.2601 #LJ parameter of water: Epsilon (kJ/mol)
solatom atominfo #Atomic coordinates, LJ, and partial charges
gridsize 100 100 100 #input grid size nx, ny, nz
pdiff 0.0 #Pressure difference (kJ/mol/Angstrom^3)
gamma 0.1315 #Surface tension (kJ/mol/Angstrom<sup>2</sup>)
tolman 1.0 #Tolman length (Angstrom)
```

solvden 0.0331 #Water number density (1/Angstrom^3)
epssolvent 78.0 #Dielectric constant for the solvent region
epssolute 1.0 #Dielectric constant for the solute region
nionsp 2 #Number of ionic species
cbulk 0.1 0.1 #Bulk concentrations for different ionic species
valence 1.0 -1.0 #Valences for different ionic species
shiftdist 1.0 #Shift distance(Angstrom)
initialstatus 1 #Initial surface: Loose: 0; Tight: 1; Resume: -1
initialsurf HGsurf.dat #Name of the file for initial surface, if
initialstatus=-1
outputsurf HGsurf #Name of the file for final equilibrium surface
end

Appendix C. Units

TABLE C.1

Units in the LS-VISM package.

Variables	Units
Length	Angstrom (Å)
Energy	kJ/mol
Temperature	Kelvin (K)
Permittivity	\mathcal{E}_0
Charge	elementary charge (e)
Concentration	Molarity (M)

Appendix D. A Matlab function: visual.m

function visual(filename)

%Syntax: visual('surf.dat')

%Visualization of zero level-set of the function provided in "surf.dat"

% It calls the subroutines in the visualization folder

addpath('../../visualization/');

fid= fopen(filename);

plotsurf(fid);

fclose(fid);

end



FIG. 1.

A schematic diagram of a solvation system with an implicit solvent. A solute-solvent interface Γ separates the solute region Ω_m from the solvent region Ω_w . Here **n** is the unit normal along the interface Γ pointing from the solute to solvent region. The relative permittivities of the solute region and solvent region are denoted by ε_m and ε_w , respectively. The solute atoms are located at **x**₁, ..., **x**_N and carry partial charges $Q_1, ..., Q_N$, respectively.











FIG. 4.

Stable equilibrium surfaces of CB[7] obtained by LS-VISM. Top row: visualization with the Matlab function plotsurf.m. The color represents the mean curvature: yellow for convex, green for flat, and blue for concave. Bottom row: visualization with VMD.



FIG. 5.

Stable equilibrium surfaces of B2 (first column) and CB[7]-B2 (second column) obtained by LS-VISM. Top row: visualization with the Matlab function plotsurf.m. The color represents the mean curvature: yellow for convex, green for flat, and blue for concave. Bottom row: visualization with VMD.

TABLE IV.1

Descriptions of input parameters in the file readin.

Parameter	Description
estat	It is set to be 0 for nonpolar systems, 1 for CFA formulation, 2 for PB without ionic solution (Poisson), 3 for linearized PB, 4 for nonlinear PB.
runsteps	VISM relaxation steps. Default value is 1000.
waterLJsig	The σ value of LJ potential. Unit: Å.
waterLJeps	The ε value of LJ potential. Unit: kJ/mol.
solatom	The file that contains the information of solute atoms, e.g., coordinates, partial charges, etc. Default value is "atominfo". See Subsection IV C.
gridsize	The computational grid size. Default value is 80 80 80.
pdiff	The pressure difference between the solute region and solvent region. Unit: kJ/mol/Å3.
gamma	The solute-solvent interfacial tension. Unit: kJ/mol/Å ² .
tolman	The Tolman length τ . Unit: Å.
solvden	The solvent number density $\rho_{\rm w}$. Unit: Å ⁻³ .
epssolvent	The relative permittivity ε_w of the solvent region.
epssolute	The relative permittivity $\varepsilon_{\rm m}$ of the solute region.
nionsp	The number of ionic species.
cbulk	The bulk concentrations for different ionic species: $c_1^{\infty}, \cdots, c_M^{\infty}$. Unit: M.
valence	The valences for different ionic species: z_1, \dots, z_M
shiftdist	The dielectric boundary shift distance ξ for the calculation of the final electrostatic solvation energy. Unit: Å.
initialstatus	The type of initial surface. Value 0 represents a loose initial surface, Value 1 represents a tight initial surface, and Value –1 means the calculation resumes from a given surface. See Subsection IV B.
initialsurf	Name of the file of initial surface, if initialstatus=-1.
outputsurf	Name of the file of final output surface.
end	This indicates the end of the file readin.

TABLE V.1

Individual contributions to the host-guest solvation free energy (in kJ/mol) predicted by the level-set VISM calculations with different initial surfaces.

System	Initial	$G_{\text{geom}}[]$	$G_{vdW}[$]	$G_{\rm elec}[$]	Total Solvation Free Energy
CB[7](Dry)	Loose	209.2	-211.5	-471.4	-473.7
CB[7](Wet)	Tight	241.9	-235.0	-496.4	-489.5
B2	Loose and Tight	58.9	-45.3	-47.5	-33.9
CB[7]-B2	Loose and Tight	208.2	-222.6	-508.6	-523.0