ERmod: Fast and Versatile Computation Software for Solvation Free Energy with Approximate Theory of Solutions

Shun Sakuraba,[a,b] and Nobuyuki Matubayasi*[c,d,e]

ERmod is a software package to efficiently and approximately compute the solvation free energy using the method of energy representation. Molecular simulation is to be conducted at two condensed-phase systems of the solution of interest and the reference solvent with test-particle insertion of the solute. The subprogram **ermod** in ERmod then provides a set of energy distribution functions from the simulation trajectories, and another subprogram **slvfe** determines the solvation free energy from the distribution functions through an approximate functional. This article describes the design and implementation of ERmod, and illustrates its performance in solvent water for two organic solutes and two protein solutes. Actually, the free-energy computation with ERmod is not restricted to the solvation in homogeneous medium such as fluid and polymer and can treat the binding into weakly ordered system with nano-inhomogeneity such as micelle and lipid membrane. ERmod is available on web at **http://sourceforge.net/projects/ermod**.

DOI: 10.1002/jcc.23651

Introduction

Solvation free energy is the change in the system free energy on transfer of a solute molecule from dilute gas to a solution system of interest.[1,2] Chemical processes in solution are often governed by the free energy of solvation. The solubility is a prime example. The solvent effect on the equilibrium and rate constants of chemical reaction is also determined by the solvation free energies of the reactive species involved, and a flexible solute molecule changes its structure to achieve more favorable sum of intramolecular energy and solvation free energy. In chemical applications, all-atom computation of the free energy is desired as the solvation reflects a variety of intermolecular interactions such as hydrogen bonding and hydrophobic effect at atomic resolution. The calculation of the solvation free energy with explicit solvent is thus a major target of theoretical-computational approach to solutions and fluids.[3,4]

The concept of solvation is not restricted to homogeneous fluid. In previous articles, we treated the binding of a molecule into micelle or lipid membrane as a solvation in an extended sense.[5,6] The basic idea is to view a solution system with host structure (micelle or membrane) as a mixed solvent. In the case of micelle, for example, the surfactant forming the micelle is seen as a solvent species together with water. The guest molecule to be bound is the only species regarded as the solute, and the binding of the guest into the host was formulated as a solvation in a mixed-solvent system which involves nanoscale or mesoscale inhomogeneity. The binding ability of micelle or membrane is then quantified by the free energy of solvation in inhomogeneous mixed solvent. In this case, the solvation free energy depends on the site of binding and can determine two, key properties of binding: binding strength (partition coefficient) and binding site. The free-energy profile over the site provides the partition coefficient in combination with the Kirkwood–Buff theory, and is the direct information on favorable and unfavorable binding sites. The reduction process (attachment of electron) in condensed phase was similarly treated as a solvation.[7,8] The electron to be attached is the solute in this case, and the reduction potential is rephrased as the free energy of solvation of the electron into a mixed solvent consisting of the oxidant and the environment surrounding it.

The free energy of solvation (in the extended sense) requires much computational demand with explicit solvent, however, when such standard techniques as free-energy perturbation and thermodynamic integration are used.[3,4] These methods are conducted in practice by introducing a number of intermediate states connecting the two states of the (mixed) solvent without solute and the target solution with the solute. It should be noted that the initial and final states are of physical interest, while the intermediate states are often not. An efficient alternative is then to carry out the simulations

[a] *S. Sakuraba*
Quantum Beam Science Directorate, Japan Atomic Energy Agency, Kizugawa, Kyoto 619-0215, Japan
[b] *S. Sakuraba*
Institute for Molecular Science, Okazaki, Aichi 444–8585, Japan
[c] *Nobuyuki Matubayasi*
Division of Chemical Engineering, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan.
E-mail: nobuyuki@cheng.es.osaka-u.ac.jp
[d] *Nobuyuki Matubayasi*
Elements Strategy Initiative for Catalysts and Batteries, Kyoto University, Katsura, Kyoto 615-8520, Japan
[e] *Nobuyuki Matubayasi*
Japan Science and Technology Agency (JST), CREST, Kawaguchi, Saitama 332-0012, Japan

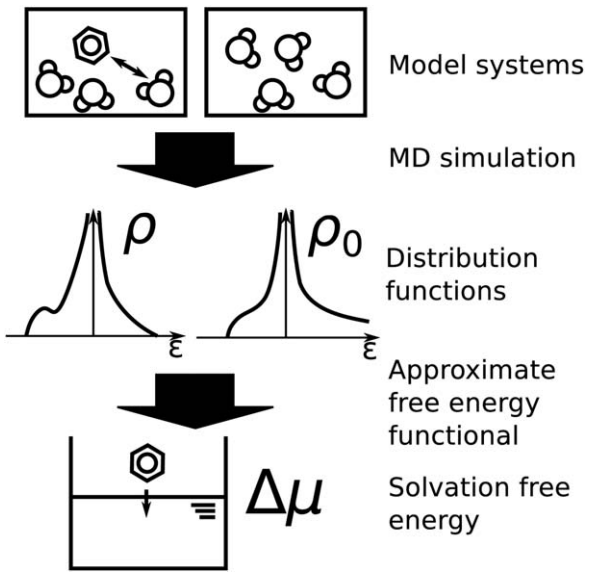

Figure 1. Overview of the calculation scheme in the energy-representation method. Molecular simulation is performed to generate a set of distribution functions which constitute the approximate functional for the solvation free energy.

only at the "end-point" states and use an approximate functional for the free energy.[9–21] Our strategy of free-energy computation belongs to this line. We approach the solvation free energy by combing the molecular simulation with a distribution-function theory of solutions.

Since the notion of solvation is not only extended to nano-organized systems such as micelle and lipid membrane but also strongly desired to handle such systems as supercritical fluid, ionic liquid, polymer, and QM/MM (quantum-mechanical/molecular-mechanical) system, the theory of solutions needs to be (re-)formulated to treat these frontline subjects. To meet this necessity, we developed the method of energy representation.[22–24] In this method, the solute–solvent distribution is expressed over the one-dimensional coordinate of solute–solvent pair interaction energy, and a set of energy distribution functions in the "end-point" states constitute an approximate functional for the solvation free energy. Among a variety of approximate free energy methods,[9–21] the method of energy representation is unique in compromising the accuracy, the efficiency, and the range of applicability. According to the benchmark tests for amino-acid analog solutes in solvent water, the error caused by the use of the approximate functional was observed not to be larger than the error due to the use of force field.[25–28] The method was applied to a variety of systems including polymer, supercritical fluid, and nonaqueous system,[29–31] and the solvation free energy of a protein with a few hundred residues can be feasibly evaluated.[32–37] The binding of a molecule into micelle and lipid membrane was treated in unified manner as solvation in inhomogeneous mixed solvent,[5,6] and the combination with QM/MM method was implemented.[38]

ERmod is an implementation of the Energy-Representation method and is a software MODule used currently with molecular dynamics (MD) simulation packages such as NAMD, Gromacs, and AMBER.[39–42] See Figure 1 for the software usage. MD simulations are carried out only at the "end-point" states. The trajectory outputs as well as the potential parameters and system setup are then handled within ERmod to construct the distribution functions, from which the solvation free energy is determined. The purpose of this article is to describe the software design of ERmod and several numerical aspects in implementation.

ERmod is an open-source, free software written with Fortran 90, C/C++, and Python. The program can run on majority of UNIX-like systems, and can be compiled with a wide variety of compilers. The program is also suitable to parallel computations using message passing interface (MPI) for internode communications and/or OpenMP for intranode communications. In addition to increasing the performance of the ERmod program sets, we focused on developing user-friendly supporting tools, to expand the applicability of the method and to eliminate common mistakes during the usage.

Theory

In the method of energy representation, the solvation free energy is computed through an approximate functional consisting of distribution functions in two systems.[23] One of the systems is called solution system, and contains the solute and solvent molecules at full coupling of their interaction. The other is called reference-solvent system. In this system, the configurations of the solute and solvent are separately prepared, and the distribution functions are calculated by placing the solute in the solvent system as a test particle. The solution and reference-solvent systems are the end-point states in the calculation of the solvation free energy with the free energy perturbation and thermodynamic integration methods. In the energy-representation method, the molecular simulation is conduced only in the solution and reference-solvent systems, leading to the reduction of computational load. In the subsection of "Approximate functional for the solvation free energy", we briefly review the method of energy representation and show the explicit form of the free-energy functional currently used.

In molecular simulation, the system is often treated with periodic boundary condition and the electrostatic interaction is commonly handled by the lattice-sum methods such as Ewald and particle mesh Ewald.[3,4,43–47] In these methods, a molecule or ion interacts with its own images and neutralizing background. This interaction is a finite-size effect, and is separately and rigorously treated in the ERmod program. In the subsection of "Finite-size correction for the electrostatic interaction in periodic boundary condition", we describe the finite-size correction scheme adopted in the program.

Approximate functional for the solvation free energy

The system of interest is a solution containing a single solute molecule. When the system contains more than one solute molecules, it suffices to regard one of the solute molecules as the "solute" and the others as part of the "solvent." The formulation is based on classical statistical mechanics, and is presented for general, mixed

solvent. No assumptions are made for whether the system of interest is homogeneous or inhomogeneous and for whether the molecules in the system are flexible or rigid. The method can be applied to both homogeneous and inhomogeneous systems and to both flexible and rigid molecules.

The solvation free energy $\Delta\mu$ is the free-energy change for turning on the intermolecular interaction of the solute with the solvent. In $\Delta\mu$, only the contribution from the potential energy is involved and the ideal (kinetic) contribution is excluded at the outset. Let ψ be the configuration of the solute molecule and $\mathbf{x}_{i,k}$ be the configuration of the ith molecule of the kth solvent species. The configuration is the collective denotation of the position and orientation and of the intramolecular degrees of freedom if the molecule is flexible. When the intermolecular interaction potential between the solute and the kth solvent species is v_k, the solvation free energy $\Delta\mu$ is expressed as

$$\exp(-\beta\Delta\mu)=\frac{\int d\psi d\mathbf{X}\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}v_k(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}{\int d\psi d\mathbf{X}\exp\left(-\beta\{\Psi(\psi)+U(\mathbf{X})\}\right)}, \quad (1)$$

where $\mathbf{X}$ represents the solvent configuration collectively, $U(\mathbf{X})$ is the total energy among the solvent, $\Psi(\psi)$ is the one-body energy of the solute, and β is the inverse of k_BT with the Boltzmann constant k_B and the temperature T. The canonical ensemble is adopted in eq. (1), and the following developments are similar in the other ensembles.

In eq. (1), the solute–solvent interaction is supposed to be pairwise additive (the pairwise additivity is supposed only for the solute–solvent interaction and is not for the solvent–solvent). This supposition is mostly valid in commonly used force fields, including CHARMM, AMBER, OPLS, and GROMOS.[48–55] A connection between the solvation free energy $\Delta\mu$ and distribution function is then provided by the Kirkwood charging formula. To formulate the Kirkwood charging formula, we introduce a set of intermediate states involving pairwise additive solute–solvent interaction. Let λ be the coupling parameter identifying the state and $u_{k,\lambda}(\psi,\mathbf{x}_k)$ be the interaction potential between the solute and the kth solvent species at the coupling parameter λ, where $\mathbf{x}_k$ refers to the configuration of the kth solvent species. When $\lambda=0$, the system is the reference-solvent system without the solute and $u_{k,0}(\psi,\mathbf{x}_k)=0$ (no solute–solvent interaction). When $\lambda=1$, the solute interacts with the solvent at full coupling and $u_{k,1}(\psi,\mathbf{x}_k)=v_k(\psi,\mathbf{x}_k)$, where v_k is the solute–solvent pair potential of interest. The form of $u_{k,\lambda}(\psi,\mathbf{x}_k)$ at $0<\lambda<1$ is arbitrary. The Kirkwood charging formula is an integration over the coupling parameter and is given by

$$\Delta\mu=\sum_k\int_0^1 d\lambda\int d\psi d\mathbf{x}_k\frac{\partial u_{k,\lambda}(\psi,\mathbf{x}_k)}{\partial\lambda}\rho_{k,\lambda}(\psi,\mathbf{x}_k), \quad (2)$$

where $\rho_{k,\lambda}(\psi,\mathbf{x}_k)$ is the solute–solvent pair distribution function in the presence of the solute–solvent interaction $u_{k,\lambda}$ and written as

$$\rho_{k,\lambda}(\psi,\mathbf{x}_k)=\frac{\int d\mathbf{X}\sum_i\delta(\mathbf{x}_{i,k}-\mathbf{x}_k)\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}u_{k,\lambda}(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}{\int d\psi d\mathbf{X}\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}u_{k,\lambda}(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}. \quad (3)$$

It should be noted that eq. (2) is exact but is not useful in numerical practice. This is because the solute–solvent distribution $\rho_{k,\lambda}(\psi,\mathbf{x}_k)$ is represented over a high-dimensional set of coordinates $(\psi,\mathbf{x}_k)$. A high-dimensional expression is avoided in the Kirkwood charging formula in the energy representation described next.

In the method of energy representation, the value of the solute–solvent pair interaction energy is adopted as the coordinate for the distribution of the solvent molecule around the solute. When the energy coordinate for the kth solvent species is denoted as ϵ_k, the corresponding instantaneous distribution $\hat{\rho}_k$ is defined as

$$\hat{\rho}_k(\epsilon_k)=\sum_i\delta\left(v_k(\psi,\mathbf{x}_{i,k})-\epsilon_k\right), \quad (4)$$

where the sum is taken over the kth solvent species. Equation (4) is the histogram of the v_k value in an instantaneous (snapshot) configuration. Although the potential function is typically expressed as the sum of Lennard–Jones (LJ) and Coulombic terms in site–site form, no specific form needs to be assumed for the function v_k. It should be further noted within eq. (4) that v_k serves only to introduce the coordinate ϵ_k and does not identify the system for which the statistical ensemble is generated. It is possible to adopt a solute–solvent interaction other than v_k for ensemble generation and still to construct the distribution through eq. (4) using the function v_k.

To formulate the Kirkwood charging formula in the energy representation, we restrict the set of solute–solvent pair interaction potentials $u_{k,\lambda}(\psi,\mathbf{x}_k)$ to those which are constant over equienergy surface of the solute–solvent pair potential $v_k(\psi,\mathbf{x}_k)$ of interest. In this case, when the value of $v_k(\psi,\mathbf{x}_k)$ is denoted as ϵ_k, the pair potential $u_{k,\lambda}$ can be written as $u_{k,\lambda}(\epsilon_k)$. At the end points ($\lambda=0$ and 1), $u_{k,0}(\epsilon_k)=0$ and $u_{k,1}(\epsilon_k)=\epsilon_k$. It is then possible to show that the Kirkwood charging formula is given by

$$\Delta\mu=\sum_k\int_0^1 d\lambda\int d\epsilon_k\frac{\partial u_{k,\lambda}(\epsilon_k)}{\partial\lambda}\rho_{k,\lambda}(\epsilon_k), \quad (5)$$

where $\rho_{k,\lambda}(\epsilon_k)$ is the ensemble average of eq. (4) in the presence of the solute–solvent interaction $u_{k,\lambda}$ and is written as

$$\rho_{k,\lambda}(\epsilon_k)=\frac{\int d\psi d\mathbf{X}\hat{\rho}_k(\epsilon_k)\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}u_{k,\lambda}(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}{\int d\psi d\mathbf{X}\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}u_{k,\lambda}(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}. \quad (6)$$

Equation (5) is exact and consists only of functions represented over the one-dimensional coordinate ϵ_k. $\rho_{k,\lambda}(\epsilon_k)$ describes the solute–solvent pair correlation over the energy coordinate ϵ_k, and the solvent-mediated (indirect) part $\omega_{k,\lambda}(\epsilon_k)$

of the solute–solvent potential of mean force in the energy representation is introduced as

$$\rho_{k,\lambda}(\epsilon_k)=\rho_{k,0}(\epsilon_k)\exp\left(-\beta\left(u_{k,\lambda}(\epsilon_k)+\omega_{k,\lambda}(\epsilon_k)\right)\right). \quad (7)$$

With eq. (7) and the end-point properties of $u_{k,0}(\epsilon_k)=0$ and $u_{k,1}(\epsilon_k)=\epsilon_k$, eq. (5) reduces to

$$\begin{aligned}\Delta\mu&=\sum_k\int d\epsilon_k\epsilon_k\rho_k(\epsilon_k)-\sum_k\int_0^1 d\lambda\int d\epsilon_k\frac{\partial\rho_{k,\lambda}(\epsilon_k)}{\partial\lambda}u_{k,\lambda}(\epsilon_k)\\&=\sum_k\left\{\int d\epsilon_k\epsilon_k\rho_k(\epsilon_k)-k_BT\int d\epsilon_k\left[\left(\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)\right)\right.\right.\\&\left.\left.-\rho_k(\epsilon_k)\log\left(\frac{\rho_k(\epsilon_k)}{\rho_{k,0}(\epsilon_k)}\right)-\beta\int_0^1 d\lambda\frac{\partial\rho_{k,\lambda}(\epsilon_k)}{\partial\lambda}\omega_{k,\lambda}(\epsilon_k)\right]\right\} \quad (8)\end{aligned}$$

where $\rho_{k,1}(\epsilon_k)$ of eq. (6) with $\lambda=1$ is written as $\rho_k(\epsilon_k)$ for notational brevity. When $u_{k,\lambda}$ is taken so that $\rho_{k,\lambda}$ varies linearly against λ, eq. (8) further simplifies into

$$\begin{aligned}\Delta\mu=\sum_k\left\{\int d\epsilon_k\epsilon_k\rho_k(\epsilon_k)-k_BT\int d\epsilon_k\left[\left(\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)\right)\right.\right.\\\left.\left.-\rho_k(\epsilon_k)\log\left(\frac{\rho_k(\epsilon_k)}{\rho_{k,0}(\epsilon_k)}\right)-\beta\left(\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)\right)\left(\int_0^1 d\lambda\omega_{k,\lambda}(\epsilon_k)\right)\right]\right\} \quad (9)\end{aligned}$$

Equations (8) and (9) are exact. In the currently used version of the energy-representation method, an approximation is introduced to the integral of $\omega_{k,\lambda}(\epsilon_k)$ over λ within eq. (9).

In the approximate treatment of the solvation free energy $\Delta\mu$ in the energy-representation method, $\Delta\mu$ is evaluated from distribution functions obtained in two systems of solution and reference solvent. The solution system refers to the numerator of eq. (1). It corresponds to $\lambda=1$ of the coupling parameter, and the solute interacts with the solvent at full coupling under the solute–solvent interaction potential of interest. In the following, the ensemble average of a quantity Q in the solution system is denoted as

$$\langle Q\rangle=\frac{\int d\psi d\mathbf{X}Q\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}v_k(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}{\int d\psi d\mathbf{X}\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}v_k(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}. \quad (10)$$

Equation (10) is simply an average generated with the standard molecular simulation. Conversely, the reference-solvent system is introduced with respect to the denominator of eq. (1), and refers to $\lambda=0$ of the coupling parameter. The ensemble average in it is then written in the following as

$$\langle Q\rangle_0=\frac{\int d\psi d\mathbf{X}Q\exp\left(-\beta\{\Psi(\psi)+U(\mathbf{X})\}\right)}{\int d\psi d\mathbf{X}\exp\left(-\beta\{\Psi(\psi)+U(\mathbf{X})\}\right)}. \quad (11)$$

In eq. (11), no solute–solvent interaction is operative when the ensemble is prepared. The solvent configuration is generated from molecular simulation without the solute. The solute is placed as a test particle without disturbing the solvent configuration, according to the probability distribution of the configuration ψ constructed by the one-body energy $\Psi(\psi)$.

The approximate $\Delta\mu$ functional in the current version of energy-representation method is given by a set of definitions and equations listed as[6,23]

$$\Delta\mu=\sum_k\Delta\mu_k, \quad (12)$$

$$\begin{aligned}\Delta\mu_k=\int d\epsilon_k\epsilon_k\rho_k(\epsilon_k)-k_BT\int d\epsilon_k\left[\left(\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)\right)\right.\\\left.-\rho_k(\epsilon_k)\log\left(\frac{\rho_k(\epsilon_k)}{\rho_{k,0}(\epsilon_k)}\right)-\{\alpha_k(\epsilon_k)F_k(\epsilon_k)\right.\\\left.+(1-\alpha_k(\epsilon_k))F_{k,0}(\epsilon_k)\}\left(\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)\right)\right] \quad (13)\end{aligned}$$

$$\rho_k(\epsilon_k)=\langle\hat{\rho}_k(\epsilon_k)\rangle, \quad (14)$$

$$\rho_{k,0}(\epsilon_k)=\langle\hat{\rho}_k(\epsilon_k)\rangle_0, \quad (15)$$

$$\chi_{kl,0}(\epsilon_k,\eta_l)=\langle\hat{\rho}_k(\epsilon_k)\hat{\rho}_l(\eta_l)\rangle_0-\langle\hat{\rho}_k(\epsilon_k)\rangle_0\langle\hat{\rho}_l(\eta_l)\rangle_0, \quad (16)$$

$$\omega_k(\epsilon_k)=-k_BT\log\left(\frac{\rho_k(\epsilon_k)}{\rho_{k,0}(\epsilon_k)}\right)-\epsilon_k, \quad (17)$$

$$\begin{aligned}\sigma_{k,0}(\epsilon_k)=-k_BT\sum_l\int d\eta_l\left(\frac{\delta_{kl}\delta(\epsilon_k-\eta_l)}{\rho_{k,0}(\epsilon_k)}-\left(\chi_{kl,0}\right)^{-1}(\epsilon_k,\eta_l)\right)\\\times\left(\rho_l(\eta_l)-\rho_{l,0}(\eta_l)\right), \quad (18)\end{aligned}$$

$$F_k(\epsilon_k)=\begin{cases}\beta\omega_k(\epsilon_k)+1+\dfrac{\beta\omega_k(\epsilon_k)}{\exp\left(-\beta\omega_k(\epsilon_k)\right)-1} & \text{when } \omega_k(\epsilon_k)\le 0\\ \dfrac{1}{2}\beta\omega_k(\epsilon_k) & \text{when } \omega_k(\epsilon_k)\ge 0,\end{cases} \quad (19)$$

$$F_{k,0}(\epsilon_k)=\begin{cases}-\log\left(1-\beta\sigma_{k,0}(\epsilon_k)\right)+1+\dfrac{\log\left(1-\beta\sigma_{k,0}(\epsilon_k)\right)}{\beta\sigma_{k,0}(\epsilon_k)} & \text{when } \sigma_{k,0}(\epsilon_k)\le 0\\ \dfrac{1}{2}\beta\sigma_{k,0}(\epsilon_k) & \text{when } \sigma_{k,0}(\epsilon_k)\ge 0,\end{cases} \quad (20)$$

$$\alpha_k(\epsilon_k)=\begin{cases}1 & \text{when } \rho_k(\epsilon_k)\geq\rho_{k,0}(\epsilon_k)\\ 1-\left(\dfrac{\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)}{\rho_k(\epsilon_k)+\rho_{k,0}(\epsilon_k)}\right)^2 & \text{when } \rho_k(\epsilon_k)\leq\rho_{k,0}(\epsilon_k).\end{cases} \quad (21)$$

Among eqs. (12)–(21), eq. (13) is the approximate expression, and eqs. (14)–(21) define the functions in the left-hand sides (or the corresponding vectors and matrices in numerical implementation). Within eq. (13), furthermore, only the $\{\alpha_k(\epsilon_k)F_k(\epsilon_k)+(1-\alpha_k(\epsilon_k))F_{k,0}(\epsilon_k)\}$ term is approximate and replaces the (exact) integral of $\omega_{k,\lambda}(\epsilon_k)$ over λ in eq. (9). It is of interest to note that the solvation free energy $\Delta\mu$ is given by a sum of constituent free energy $\Delta\mu_k$. Although $\Delta\mu_k$ is in general not observable, it can be conveniently regarded as the contribution from the kth solvent species for the purpose of interpretation, analysis, and prediction.

Equation (14) is the average distribution (histogram) of the solute–solvent pair energy value ϵ_k in the solution system. Equation (15) is similarly the average distribution in the reference solvent and corresponds to the density of states for solute–solvent pair interaction. Equation (16) is a correlation matrix and describes the solvent–solvent correlation in the reference solvent. When the solute molecule is placed as a test particle in the reference-solvent system, it often overlaps with solvent molecules. The overlapping configurations contribute to $\rho_{k,0}$ and $\chi_{kl,0}$ at large energy coordinate and account (approximately) for the excluded-volume effect in the solvation free energy. Actually, when ϵ_k is prohibitively larger than the thermal energy (k_BT), its value does not have physical meaning *per se* except to specify whether the solute–solvent pair configuration is in the excluded-volume domain. In the free-energy calculation through eqs. (12)–(21), the ϵ_k value in the excluded-volume domain acts only as an index for the distribution functions $\rho_{k,0}$ and $\chi_{kl,0}$ and is not used in the calculation of the solvation free energy $\Delta\mu$. This point is described in detail in Appendix A.

ω_k of eq. (17) is an energy-represented version of the solvent-mediated part of solute–solvent potential of mean force. It vanishes when the solvent–solvent correlation is absent. $\sigma_{k,0}$ of eq. (18) is also the solvent-mediated part of the response function of the solute–solvent distribution to the solute–solvent interaction in the reference solvent. With the definition of $\omega_{k,\lambda}(\epsilon_k)$ in eq. (7), $\omega_k(\epsilon_k)=\omega_{k,1}(\epsilon_k)$ and

$$\sigma_{k,0}(\epsilon_k)=\frac{\partial\omega_{k,\lambda}(\epsilon_k)}{\partial\lambda}\Big|_{\lambda=1} \quad (22)$$

holds when $\rho_{k,\lambda}$ varies linearly against λ. The matrix inversion in eq. (18) cannot always be performed in straightforward manner, though, and the procedure for determining $\sigma_{k,0}$ through eq. (18) is described in Appendix B. Equations (19) and (20) introduce a combined Percus–Yevick-type and hypernetted-chain-type approximation. They are the expressions in the solution and reference-solvent systems, respectively, and are mixed with a weighting function of eq. (21). With the form of the weighting function in eq. (21), the value of $\omega_k(\epsilon_k)$ needs to be determined only at ϵ_k with $\rho_k(\epsilon_k)\neq 0$ and $\sigma_{k,0}(\epsilon_k)$ is necessary only at ϵ_k with $\rho_{k,0}(\epsilon_k)\neq 0$.

The first term of eq. (13) is equal to the average sum of the interaction energy between the solute and the kth solvent species. When the distribution functions ρ_k, $\rho_{k,0}$, and $\chi_{kl,0}$ are constructed through eqs. (14)–(16), the energy coordinate ϵ_k needs to be discretized over finite meshes in practice. An experience is then that the numerical error due to the introduction of meshes is the largest in the first term of eq. (13). In the ERmod program, the first term of eq. (13) is separately calculated simply as the average sum of the solute–solvent interaction energy in the solution system, and the mesh error is present only in the other terms.

Equation (13) involves a term with form of $\rho_k\log\left(\rho_k/\rho_{k,0}\right)$. This term is problematic when ϵ_k with $\rho_k\neq 0$ and $\rho_{k,0}=0$ is present. Such a domain may appear in the low-energy tail of ϵ_k in which the density of states for the solute–solvent interaction is very small. In Appendix C, we present a method of interpolation/extrapolation for treating the problematic ϵ_k domain.

The computation of the pair interaction energy between the solute and solvent plays a key role in the method of energy representation. We show the expression for the pair energy in Appendix D, and describe its efficient scheme of computation in Appendix E.

Finite-size correction for the electrostatic interaction in periodic boundary condition

In molecular simulation, the system size is always finite and the periodic boundary condition is commonly used. A correction against the finiteness and/or periodicity of the system is often necessary for quantitative analysis of thermodynamic quantities. When the intermolecular interaction consists of the LJ and electrostatic terms, the correction scheme is formulated independently for each term. The LJ term is corrected against the use of finite cutoff length. The standard scheme is to integrate the LJ potential function beyond the cutoff length, with the supposition that the solvent local structure is no more affected by the presence of solute and that the solute–solvent radial distribution function is unity.[3] Although this scheme is well used in homogeneous system, the treatment of inhomogeneous systems such as lipid membrane and interface needs still to be developed. A general scheme of finite-size correction is not established yet for the LJ interaction, and the ERmod program corrects the computed free energy of solvation with respect only to the electrostatic interaction.

In this subsection, we assume that the molecular simulation is conducted in periodic boundary condition and that the electrostatic interaction is handled by lattice-sum method. In the lattice-sum methods such as (full) Ewald and smooth particle-mesh Ewald (SPME), each charge in the system interacts with its own images and neutralizing background. This interaction may be called self energy since it appears as a one-body function in the potential. It was then reported for the electrostatic part of solvation free energy that the system-size dependence in periodic boundary system can be removed effectively when

the self energy is incorporated into the lattice-sum potential function.[45–47] The ERmod program adopts the correction scheme with the self energy when the lattice-sum method is used in periodic boundary condition.

In the solution system, all of the solute and solvent molecules stay in periodic boundary. The self energies of the solute and solvent are then parts of the one-body energy $\Psi(\psi)$ of the solute and the solvent–solvent energy $U(\mathbf{X})$ in eq. (10), respectively, where ψ is the solute configuration and $\mathbf{X}$ denotes the solvent configuration collectively. In the reference-solvent system, a typical setup is that only the solvent is subject to the lattice-sum treatment and that the solute is simulated at isolation without use of a lattice-sum method by adopting the bare form $1/r$ of the electrostatic interaction. In this case, the self-energy term is absent for the solute energy even when the simulation of isolated solute is conducted in periodic boundary for the sake of numerical convenience (for example, to prevent the coordinate from becoming indefinitely large). In other words, $U(\mathbf{X})$ contains the self-energy part and $\Psi(\psi)$ does not in eq. (11). Correspondingly, the solvent–solvent energy is the same but the one-body energy of the solute is different between the solution and reference-solvent systems at given $(\psi, \mathbf{X})$ (and the system volume in the isothermal-isobaric ensemble).

When the typical setup described above is adopted, the ERmod program corrects the ensemble averages in the reference-solvent system by taking into account the difference in the one-body energy of the solute between the solution and reference-solvent systems. This difference v_{FSC} is the self energy of the solute and its explicit expression is given in Appendix F. Equation (1) can then be rewritten as

$$\Delta\mu = -k_B T \log \langle \exp(-\beta v_{\mathrm{FSC}}) \rangle_0 - k_B T \log \frac{\left\langle \exp\left(-\beta \sum_{i,k} v_k(\psi, \mathbf{x}_{i,k})\right) \exp(-\beta v_{\mathrm{FSC}}) \right\rangle_0}{\langle \exp(-\beta v_{\mathrm{FSC}}) \rangle_0}, \quad (23)$$

where $\langle \cdots \rangle_0$ is the ensemble average in the reference solvent given by eq. (11). The ERmod program computes the first term of eq. (23) directly without approximation. The second term is obtained through eqs. (12)–(21) by modifying the ensemble average $\langle Q \rangle_0$ in the reference solvent from eq. (11) to

$$\frac{\langle Q \exp(-\beta v_{\mathrm{FSC}}) \rangle_0}{\langle \exp(-\beta v_{\mathrm{FSC}}) \rangle_0}, \quad (24)$$

which corresponds to a weighted statistical average. Accordingly, the distribution function $\rho_{k,0}$ and the correlation matrix $\chi_{kl,0}$ are constructed through eq. (24) in ERmod.

When v_{FSC} is much smaller in magnitude than $k_B T$, the first term of eq. (23) is negligible and the modification of the ensemble average from eq. (11) to (24) has no appreciable effect. This is the case when the solute is neutral and its size is much smaller than the system size. When the solute is ionic, the first term of eq. (23) can be significant. Even in the ionic case, though, the difference between eqs. (11) and (24) affects the $\Delta\mu$ value only within the margin of error when the ion is much smaller in size than the total system.

ERmod Software Design

General computational flow

ERmod is currently designed to be a postprocessor of molecular simulation. As user has his own preferred simulation platform, we tried not to restrict the choice of molecular simulation program. ERmod reads an output from molecular simulation represented as a trajectory file, calculates the solute–solvent interaction energies, and provides an approximate value of solvation free energy.

Before running the ERmod program, two simulations are to be conducted for the solution system of interest and for the reference-solvent system. The solution system contains the solute molecule at full coupling of the solute–solvent interaction, and the ensemble average is taken with eq. (10). The reference solvent is the system without the solute. The solute is placed into the reference-solvent system as a test particle, and eq. (11) [or eq. (24)] is the expression for the ensemble average. When the solute molecule is flexible, its simulation at isolation (in vacuum) is also to be carried out before the insertion to prepare a set of intramolecular configurations.

Figure 2 shows the workflow of ERmod. It is separated into two parts. ERmod's subprogram, **ermod**, reads trajectories from molecular simulation and calculates the statistical average of the histogram of the solute–solvent pair energy. The **ermod** program outputs either ρ_k of eq. (14) or $\rho_{k,0}$ and $\chi_{kl,0}$ of eqs (15) and (16). ERmod's subprogram, **slvfe**, computes the solvation free energy from ρ_k, $\rho_{k,0}$, and $\chi_{kl,0}$ through eqs. (12), (13), and (17)–(21). The topology information to be fed as input files into **ermod** or **slvfe** can be generated manually by user or be translated via scripts from topology and/or log files of the molecular simulation. In ERmod, scripts of automatic translation are prepared for commonly used MD simulation packages such as NAMD, GROMACS, and AMBER.[39–42] To read a variety of trajectory formats from the simulation, furthermore, the VMD molfile plugin is supported for trajectory I/O.

As seen from the theoretical developments in the subsection of "Approximate functional for the solvation free energy", the computation of the pair interaction energy between the solute and solvent is a key part of the **ermod** program. In Appendix E, we describe the efficient scheme of pair-energy computation implemented in **ermod**.

Parallelization strategy

The ERmod package consists of two subprograms of **ermod** and **slvfe**. As noted in the previous subsection, **slvfe** determines the solvation free energy from the outputs of **ermod** through eqs. (12), (13), and (17)–(21). The computation time for **slvfe** is negligible compared to **ermod**, and the parallelization is elaborated only for **ermod**.

ermod outputs the averaged distribution functions of solute–solvent pair interaction energy. The snapshots of trajectory can be processed in any order, and we exploited this inherent parallelism in **ermod**. **ermod** is parallelized with MPI.

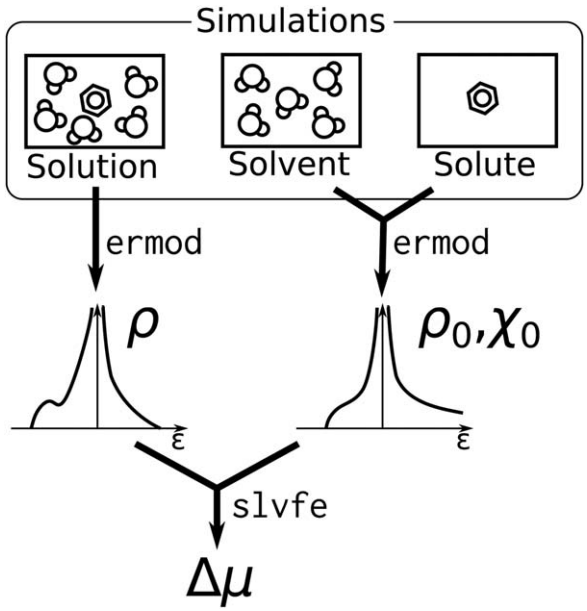

Figure 2. Schematic computational flow of ERmod. Molecular simulation is conducted for two condensed-phase systems of the solution of interest and the reference solvent, as well as for the solute molecule at isolation if it is flexible. The subprogram **ermod** provides ρ_k of eq. (14) from the trajectory of the solution system and $\rho_{k,0}$ and $\chi_{kl,0}$ of eqs. (15) and (16) from the trajectories of the reference-solvent and single-solute systems through test-particle insertion. The subprogram **slvfe** outputs the target property of solvation free energy through eqs. (12), (13), and (17)–(21).

Each MPI process handles a distinct snapshot of trajectory independently. The communication is not necessary in the treatment of a single snapshot, and arises only at the trajectory file I/O and the accumulation of the resulting distribution functions. This parallelization strategy simplifies the structure of the program with performance benefit; for example, the parallel FFT operation can be avoided, which is costly in distributed-memory parallel machines. **ermod** keeps high parallel efficiency even with massively parallel environment owing to its "embarrassingly parallel" architecture.

The intranode parallelization is also supported in **ermod** with OpenMP. The hybrid parallelization (MPI and OpenMP) is useful to compute the solvation free energy of large solute such as protein, due to lowered memory requirements compared to flat MPI (parallelization with MPI only).

Illustration of Performance

System setups

To illustrate the practical aspect of the energy-representation method, four solutes were examined in solvent water: toluene and ethanol as small molecule and bovine pancreatic trypsin inhibitor (BPTI) and T4 lysozyme (T4L) as protein. The number of solvent (water) molecules was 1000 when the solute is toluene or ethanol, 15,000 for the BPTI solute, and 40,000 for the T4L solute. As noted in the subsection of "General computational flow", the simulation was conducted for two condensed-phase systems for each solute. One is the solution system containing a single solute molecule and the solvent water, and the other is the pure-solvent system consisting only of water. The number of solvent molecules, the force field, the statistical ensemble are identical between the two systems; the only difference is the presence or absence of the solute.

The force filed used was CHARMM27 for the solutes and TIP3P for water,[48,49,56,57]1 and the LJ energy and length parameters between unlike atoms were combined with the Lorentz–Berthelot rule.[3,58] The N- and C-termini of BPTI and T4L were ionic and were represented as NH_3^+ and COO^-, respectively. MD was conducted in a cubic unit cell using GROMACS 4.5.6,[40,42] and the resulting trajectory was stored in the xtc format with the default scale factor (precision) of 1000 and fed into the **ermod** program. The periodic boundary condition was adopted with the minimum image convention. The electrostatic interaction was handled by the SPME method with a real-space cutoff of 13.5 Å, a spline order of 4 (cubic spline interpolation), and a relative tolerance of 10^{-5} (inverse decay length of 0.23 $Å^{-1}$). The reciprocal-space mesh size for each of the *x*, *y*, and *z* directions was 32 for the toluene and ethanol systems, 72 for the BPTI system, and 96 for the T4L system. The Lennard–Jones (LJ) interaction was truncated by applying the switching function with the switching range of 10–12 Å.[59] The equation of motion was integrated with the leap-frog algorithm at a time step of 2 fs. The LINCS algorithm was used to fix the lengths of the bonds involving the hydrogen atom, and the water molecules were kept rigid with SETTLE.[60,61] The simulation was carried out in the isothermal-isobaric (NPT) ensemble when the solute is toluene or ethanol, and was done in the canonical (NVT) ensemble when the solute is BPTI or T4L; the edge length of the cubic unit cell was 76.9 and 107.1 Å for the BPTI and T4L systems, respectively. The temperature was set at 300 K using the stochastic dynamics with an inverse friction constant of 1.0 ps, and the pressure was maintained at 1 atm with the Parrinello–Rahman method at a time constant of 1.0 ps when the NPT ensemble is used.[62–64]

In the subsection of "Convergence in the free-energy computation", we examine the convergence behavior of the solvation free energy computed by MD and ERmod. In the subsection of "Running time", we illustrate the running times of the ERmod program in comparison to MD. As default, the ERmod program is compiled at double precision, and was so in this work. GROMACS is often used at double precision, as done in the subsection of "Convergence in the free-energy computation", while it is compiled at single precision with the default setting. In the subsection of "Running time" we compare the running times of GROMACS MD both at single and double precisions. Still, the formats are the same for the output trajectories from the single- and double-precision MD runs, and the running time of ERmod does not reflect the precision of MD.

Convergence in the free-energy computation

When the solute is small and is toluene or ethanol, the solution MD was performed over 1 ns. The snapshot (instantaneous

[1]The original TIP3P model is used, and the hydrogen atom does not carry the Lennard–Jones term.

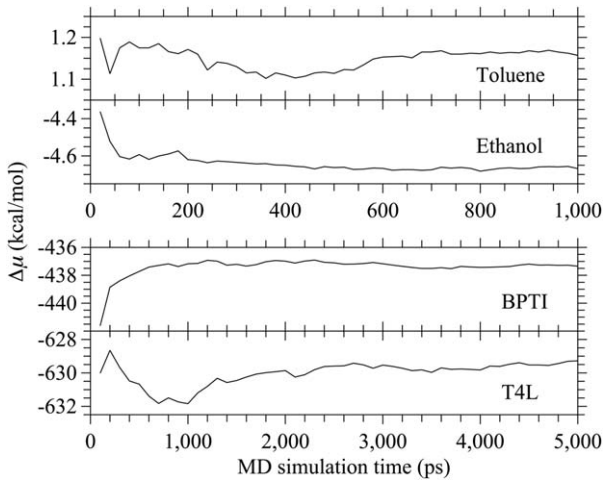

Figure 3. Convergence of the solvation free energy $\Delta\mu$ as function of the simulation time of the solution system for toluene, ethanol, BPTI, and T4L from top to bottom. The snapshot (instantaneous configuration) is sampled at every 50 fs for toluene and ethanol and at every 100 fs for BPTI and T4L. The MD length of the reference solvent (pure water) is fixed at 100 ps for the toluene and ethanol solutes with the sampling interval of 100 fs and the number of insertions of 1000 per solvent configuration. When the solute is BPTI or T4L, the MD length of the reference solvent is kept at 5 ns with the sampling interval of 100 fs and the number of insertions of 200 per solvent configuration. The long-range correction to the LJ interaction beyond the cutoff is not applied to the $\Delta\mu$ values shown, while the self-energy correction for the electrostatic interaction is applied with the scheme described in the subsection of "Finite-size correction for the electrostatic interaction in periodic boundary condition" and Appendix F.

configuration) was sampled every 50 fs to construct the energy distribution function through eqs. (4) and (14). The pure solvent (pure water) was simulated over 100 ps and was sampled every 100 fs. To prepare a set of intramolecular configurations of the solute for the test-particle insertion, the single-molecule simulation for an isolated solute was also done in vacuum over 100 ns and the configuration was saved every 100 fs. The solute was inserted into the pure-solvent system at random position with random orientation through eq. (11). The test-particle insertion was carried out 1000 times per pure-solvent configuration sampled, to generate a set of solute–solvent configurations for constructing the distribution functions of eqs. (15) and (16).

When the solute is protein and is BPTI or T4L, the MD length was 5 ns for both the solution and the pure solvent (pure water). The snapshot was sampled every 100 fs for the solution system. The pure-solvent system was also sampled every 100 fs, and the protein solute was inserted 200 times at each solvent configuration sampled. The structure of the protein solute was frozen to the one obtained after the minimization in vacuum from the crystal structure in Protein Data Bank (PDB); the PDB codes are 6PTI and 2LZM for BPTI and T4L, respectively. The minimization was conducted through 1000 steps of the steepest descent method followed by 5000 steps of the conjugate gradient method. Only during the steepest descent minimization, a restraining force of 1000 kJ/mol/nm^2 was applied to the heavy (nonhydrogen) atoms with reference to their PDB coordinates to keep the structure intact. With the frozen structure, both the position and orientation were further fixed in the solution MD. The test-particle insertion used the same frozen structure and was carried out at random position with fixed orientation; the insertion orientation was kept fixed to the one in the corresponding solution MD. This setup was adopted since the self energy in the Ewald (SPME) method varies with the orientation when the solute size is not much smaller than the size of the MD unit cell. The orientation dependence of the self energy is a finite-size effect, and is canceled between the solution and reference-solvent systems in the present setup. The insertion position was set to be random, conversely, since the system is homogeneous and the self energy does not vary with the position.[2]

We then examine the convergence of the solvation free energy $\Delta\mu$ as function of the simulation times of the solution and reference-solvent (pure water) systems. Figure 3 shows the convergence against the MD length of the solution system when the MD length of the reference solvent is fixed at 100 ps for the toluene and ethanol solutes and at 5 ns for the protein solutes. It is seen for the small solutes that the $\Delta\mu$ convergence within 0.1 kcal/mol is reached in 100 ps of the solution MD. When the solute is protein, the 1 kcal/mol convergence is achieved in 1 ns for the smaller BPTI and in 2 ns for the larger T4L.

The $\Delta\mu$ convergence is shown in Figure 4 against the number of test-particle insertions in the reference-solvent system with the MD length of the solution kept at 1 ns for toluene and ethanol and at 5 ns for the proteins. When the solute is small and is toluene or ethanol, 10^5 insertions corresponds to a 10-ps MD of the pure-water solvent since the solvent system is sampled every 100 fs and the number of insertions per solvent configuration is 1000. When the solute is BPTI or T4L, a 500-ps MD of pure water gives rise to 10^6 insertions since the number of insertions is 200 per solvent configuration. According to Figure 4, 10^5 insertions or a 10-ps MD of pure water is enough for the $\Delta\mu$ convergence within 0.1 kcal/mol when the solute is small. The 1 kcal/mol convergence is achieved, on the other hand, within 2×10^6 insertions (corresponding to 1-ns MD of pure water) for BPTI and within 4×10^6 insertions (corresponding to 2-ns MD) for T4L.

Running time

We compare the running times of molecular simulation and ERmod's subprogram **ermod**; the running time is negligible for another subprogram **slvfe** and the comparison is done only for **ermod**. For this set of comparisons, the solution MD was performed over 100 ps for the toluene and ethanol solutes and over 2 ns for the protein solutes, and the reference-solvent (pure water) MD was conducted over 10 ps for the small solutes and over 2 ns for the protein solutes. The single-molecule simulation was also done for 10 ns when the solute

[2]When the (original) Ewald method is used, the self energy is exactly invariant to the translation. In the SPME method, the self energy depends on the position to negligible extent.

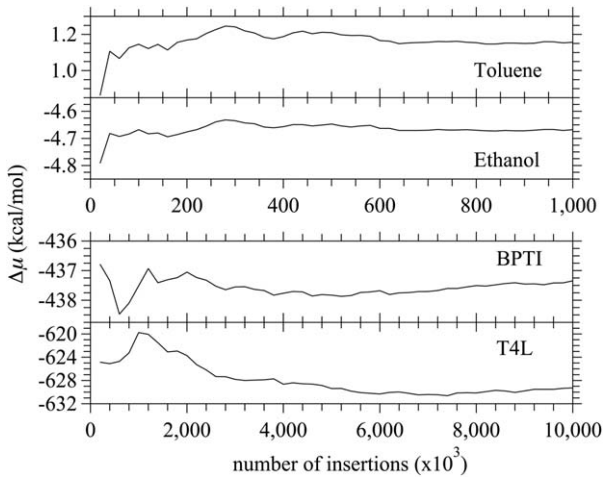

Figure 4. Convergence of the solvation free energy $\Delta\mu$ as function of the number of test-particle insertions in the reference-solvent system for toluene, ethanol, BPTI, and T4L from top to bottom. The snapshot (instantaneous configuration) is sampled every 100 fs, and the number of insertions per solvent configuration is 1000 for ethanol and toluene and is 200 for BPTI and T4L. When the solute is small and is ethanol or toluene, 10^5 insertions corresponds to a 10-ps MD of pure water. When the solute is BPTI or T4L, 10^6 insertions corresponds to a 500-ps MD of pure water. The MD length of the solution system is fixed at 1 ns for the small solutes with the sampling interval of 50 fs, and it is 5 ns for the protein solutes with the sampling interval of 100 fs. The long-range correction to the LJ interaction beyond the cutoff is not applied to the $\Delta\mu$ values shown, while the self-energy correction for the electrostatic interaction is applied with the scheme described in the subsection of "Finite-size correction for the electrostatic interaction in periodic boundary condition" and Appendix F.

is toluene or ethanol. The sampling intervals and the number of insertions per solvent configuration are identical to those described in the previous subsection. When the solute is toluene or ethanol, the running times were measured on a single core within Intel Xeon X5690 CPU. When the solute is BPTI or T4L, the parallel computations were carried out on a computation node equipped with two Intel Xeon X5690 CPUs (12 cores in total). Table 1 lists the running times thus measured. The times for the **ermod** calculations correspond to a few tens of percents of the MD running times. With ERmod, the solvation free energy can be obtained with a modest increase of the computation time from the conventional simulations only of the solution and reference-solvent systems of interest.

As illustrated in Table 1, the **ermod** calculation for the solution system requires negligible computational load compared to the MD run. The running time of **ermod** is dominated by the reference-solvent part, and is desirable to be shortened by parallel computing. We then examined the speedup of **ermod** with the number of computation nodes for the reference-solvent calculation of T4L at the conditions identical to those for Table 1. This examination was done using a computer cluster composed of two CPUs of Intel Xeon E5-2690 (16 cores in total) with 10 Gigabit–Ethernet interconnect installed on each node. The speedup is shown in Figure 5 against the number of nodes. Due to the "embarrassingly parallel" nature of the program, **ermod** scales well on general-purpose clusters.

Concluding Remarks

We described the design and implementation of the software ERmod. As demonstrated in the section of "Illustration of Performance", the current implementation of ERmod allows the computation of the solvation free energy only with the cost of the molecular simulations with and without the solute in the solvent. The ERmod package is distributed on web at **http://sourceforge.net/projects/ermod** under the GNU General Public License. The source code, manual, and sample data are available there.

According to eqs. (12)–(21), a numerical difficulty arises when ρ_k of eq. (14) in the solution system is non-zero and $\rho_{k,0}$ of eq. (15) in the reference-solvent vanishes in some domain of ϵ_k (there is no problem, conversely, when ρ_k is zero). This is encountered when the solute or solvent changes its (intramolecular) structure significantly through solute–solvent interaction. A cure is proposed in Appendix C. When the variation of the solute structure is large, the difficulty can also be circumvented by modifying the sampling scheme with knowledge of computational results in solution.[24] With this scheme, the functional for the solvation free energy is modified in response.

A challenge is present when the (intramolecular) structure changes drastically for the solvent. When the protein and water are viewed as a mixed solvent, the induced fit of ligand corresponds to such category. To treat the ligand-binding

Table 1. Running time (wallclock time) of MD and **ermod** in second, shown to three significant figures when the running time is more than 100 s and to two significant figures when the time is less.

			MD (single precision)[a]			MD (double precision)[a]			**ermod**	
Solute	N_{solute} [b]	$N_{solvent}$ [c]	Solution	Pure solvent	Isolated solute	Solution	Pure solvent	Isolated solute	Solution	Reference solvent
Toluene[d]	15	3000	1320	131	75	2640	259	83	14	299
ethanol[d]	9	3000	1320	131	47	2650	259	51	13	281
BPTI[e]	875	45,000	34,200	32,400	–[f]	70,800	69,800	–[f]	258	29,600
T4L[e]	2643	120,000	89,500	84,000	–[f]	190,000	179,000	–[f]	725	89,100

[a] The equilibration time is not included in the running time of MD. [b] The number of atoms contained in the (single) solute molecule. [c] The total number of solvent atoms in the whole system. [d] The running times were measured on a single core within Intel Xeon X5690 CPU when the solute is toluene or ethanol. [e] The running times were measured on a computation node equipped with two Intel Xeon X5690 CPUs (12 cores in total) when the solute is BPTI or T4L. [f] The running time is absent for the isolated-solute MD, as the protein is treated as rigid and a single-molecule MD of isolated solute is not necessary in that case.

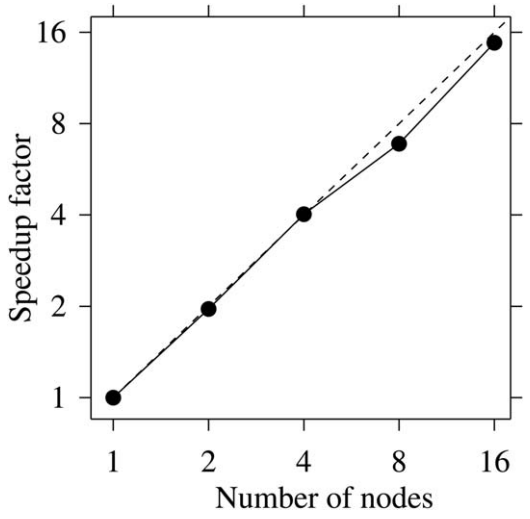

Figure 5. Speedup of the **ermod** computation time as function of the number of nodes used for the reference-solvent calculation of T4L. Each node has 16 cores, and the number of cores used for the **ermod** computation is 16 times the number of nodes shown. Except for the number of cores, the conditions of the calculations are identical to those used for preparing Table 1. The ordinate represents the speedup of running time compared to the case with the single-node processing. The dashed line represents the ideal case of perfect scaling, and the solid line is drawn for eye guide. Both the abscissa and ordinate are shown logarithmically.

into protein with a large conformational change, it may be useful to view the protein further as a collection of "segments"; the protein molecule itself is to be treated as a mixed solvent system as did for polymer system.[31] Actually, the concept of energy coordinate allows a flexible identification for what segment of molecule and/or group of atoms is treated as a single "solvent" species. The introduction of energy coordinate is a kind of coarse-graining procedure for reducing the information content of solute–solvent configuration, and it can be of interest to explore energy-based descriptions of dynamical process and constructions of coarse-grained model.

Acknowledgments

This work is supported by the Grants-in-Aid for Scientific Research (Nos. 21300111, 23651202, and 26240045) from the Japan Society for the Promotion of Science, by the Grant-in-Aid for Scientific Research on Innovative Areas (No. 20118002) and the Elements Strategy Initiative for Catalysts & Batteries from the Ministry of Education, Culture, Sports, Science, and Technology, and by the Nanoscience Program, the Computational Materials Science Initiative, the Theoretical and Computational Chemistry Initiative, the HPCI Strategic Program, and the Strategic Programs for Innovative Research of the Next-Generation Supercomputing Project. Development and performance evaluation of the software was conducted partly using PRIMERGY at Research Center for Computational Science in National Institute of Natural Sciences as well as computational resources of the HPCI systems provided by T2K at University of Tsukuba, TSUBAME2.0 and TSUBAME2.5 at Tokyo Institute of Technology, and Cray XE6 at Kyoto University through the HPCI System Research Project (Project IDs:hp120093, hp130022, hp140156, and hp140214). We are also grateful to Dr. Y. Karino of RIKEN Quantitative Biology Center for valuable comments.

Appendix A: On the Energy Values in the Excluded-Volume Domain

In the test-particle insertion of the solute into the reference solvent, the pair energy histogram of eq. (4) is often non-zero at ϵ_k much larger than k_BT. A prohibitively large ϵ_k corresponds to an overlapping configuration of the solute–solvent pair, which determines the excluded-volume contribution to the solvation free energy $\Delta\mu$. In the free-energy calculation through eqs. (12)–(21), the ϵ_k value itself acts only as an index for the distribution functions $\rho_{k,0}$ and $\chi_{kl,0}$ in the reference solvent when ϵ_k is highly unfavorable and the corresponding ρ_k in the solution is (essentially) zero. The ϵ_k value in the excluded-volume domain is not used in the $\Delta\mu$ calculation, except in the construction of $\rho_{k,0}$ and $\chi_{kl,0}$. The contribution to $\Delta\mu$ from the large ϵ_k domain is then not affected by the change in the ϵ_k value as far as ϵ_k remains prohibitively large.

To show this property, we consider a family of solute–solvent interactions $u_k(\epsilon_k)$ which are constant over equienergy surface of the solute–solvent pair potential v_k of interest. It is then possible to show through a line parallel to the blip-function method[58,65] that

$$\frac{\delta(\Delta\mu)}{\delta\exp\left(-\beta u_k(\epsilon_k)\right)} = -k_BT\exp\left(\beta u_k(\epsilon_k)\right)\rho_k(\epsilon_k;u_k) \qquad \text{(A1)}$$

holds, where $\rho_k(\epsilon_k;u_k)$ is the ensemble average of eq. (4) in the presence of the solute–solvent interaction $u_k(\epsilon_k)$. Here we restrict our attention only to the ϵ_k domain corresponding to the excluded volume and the set of $u_k(\epsilon_k)$ which are prohibitively large in that ϵ_k domain. The Boltzmann factor $\exp\left(-\beta u_k(\epsilon_k)\right)$ is exponentially small in this domain. For a change of $u_k(\epsilon_k)$ within the set, the corresponding variation of $\exp\left(-\beta u_k(\epsilon_k)\right)$ is also exponentially small. It then follows that such variations of $u_k(\epsilon_k)$ do not change the $\Delta\mu$ value. This is because the right-hand side of eq. (A1) remains finite even when $u_k(\epsilon_k)$ is large, as is generally proven in Refs. [58, 65]. Thus, the choice of the $u_k(\epsilon_k)$ form has no effect on the $\Delta\mu$ calculation in the excluded-volume domain.

Appendix B: Inversion of the Correlation Matrix

In this Appendix, we describe the numerical procedure to invert the correlation matrix $\chi_{kl,0}(\epsilon_k,\eta_l)$ and obtain $\sigma_{k,0}(\epsilon_k)$ through eq. (18). As shown in Appendix B of Ref. [24], $\chi_{kl,0}$ is positive definite and is straightforwardly inverted when the number of solvent molecules interacting with the solute varies with the system configuration. This holds when the grand canonical ensemble is used or when the solute–solvent interaction is restricted in finite domain and is not operative over the whole system. A care is needed when the number of interacting

solute–solvent pairs is invariable over the ensemble of configurations. When all the solvent molecules in the system are counted to be within the interaction range and a statistical ensemble with constant number is adopted, $\chi_{kl,0}$ is positive semidefinite and has null eigenvalue. The treatment of such $\chi_{kl,0}$ is summarized below for general, mixed-solvent system.

For the following discussion, it is convenient to rewrite eq. (18) as

$$\sum_l \int d\eta_l \chi_{kl,0}(\epsilon_k,\eta_l)\bar{u}_{l,0}(\eta_l) = -k_B T\left(\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)\right) \quad \text{(B1)}$$

$$\sigma_{k,0}(\epsilon_k) = -\bar{u}_{l,0}(\epsilon_k) - k_B T \frac{\rho_k(\epsilon_k)-\rho_{k,0}(\epsilon_k)}{\rho_{k,0}(\epsilon_k)}, \quad \text{(B2)}$$

where $\bar{u}_{k,0}$ is defined as the solution to the linear equation of eq. (B1). As we focus on numerical issue, we adopt the discretized forms of equations, as done in actual implementation. Let $\rho_k(i_k)$ and $\rho_{k,0}(i_k)$ be the average numbers of kth solvent species in the i_kth energy interval in the solution and reference-solvent systems, respectively. They are obtained by integrating the distribution functions $\rho_k(\epsilon_k)$ and $\rho_{k,0}(\epsilon_k)$ over the i_kth interval of the energy coordinate ϵ_k, respectively. Similarly, we let $\chi_{kl,0}(i_k,j_l)$ denote the integral of the correlation matrix $\chi_{kl,0}(\epsilon_k,\eta_l)$ over the i_kth interval of ϵ_k and the j_lth interval of η_l. Equation (B1) is then expressed in discretized form as

$$\sum_l \sum_{j_l} \chi_{kl,0}(i_k,j_l)\bar{u}_{l,0}(j_l) = -k_B T\left(\rho_k(i_k)-\rho_{k,0}(i_k)\right), \quad \text{(B3)}$$

where $\bar{u}_{k,0}(i_k)$ is a solution to eq. (B3) and is the discretized form of $\bar{u}_{k,0}(\epsilon_k)$.

Let z_p be the pth eigenvalue of the symmetric matrix $\chi_{kl,0}(i_k, j_l)$ and $g_p(i_k)$ be the corresponding eigenvector, which satisfy

$$\sum_l \sum_{j_l} \chi_{kl,0}(i_k,j_l) g_p(j_l) = z_p g_p(i_k) \quad \text{(B4)}$$

According to the arguments in Appendix B of Ref. [24], all the z_p are non-negative and the null eigenvalue is M-fold degenerate, where M is the number of (mixed) solvent species. Equation (B3) is then solved as

$$\bar{u}_{k,0}(i_k) = C_k - k_B T \sum_p{}' g_p(i_k) \frac{1}{z_p} \sum_{l,j_l} g_p(j_l)\left(\rho_l(j_l)-\rho_{l,0}(j_l)\right), \quad \text{(B5)}$$

where the set of eigenvectors is taken to be orthonormal and the terms corresponding to the null eigenvalue are omitted in the sum over p (denoted as $\sum_p'$). C_k is a constant independent of the discretized coordinate, and a procedure to fix C_k was proposed in Appendix B of Ref. [24] to assure that the solvation free energy is an intensive property. For each k, C_k is determined by setting $\bar{u}_{k,0}(I_k)$ to be zero at the interval I_k which contains $\epsilon_k = 0$. The matrix form of the operation in eq. (B5) given by

$$\sum_p{}' g_p(i_k) \frac{1}{z_p} g_p(j_l) \quad \text{(B6)}$$

is called Moore–Penrose pseudoinverse of $\chi_{kl,0}(i_k,j_l)$.

When the numerical error is of order κ for the eigenvalues and eigenvectors of $\chi_{kl,0}(i_k,j_l)$, each term in the sum of eq. (B5) involves an error of order κ. Due to the numerical error, in particular, all the eigenvalues may be apparently non-zero and $\chi_{kl,0}(i_k,j_l)$ may be invertible at the level of numerical manipulation. This procedure cannot be justified, however, since the correct limit is not achieved at $\kappa \to 0$. As described in Appendix B of Ref. [24], the terms corresponding to the null eigenvalue at $\kappa \to 0$ need to be omitted in the sum of eq. (B5). With a sufficient number of configurations for the reference-solvent ensemble, a numerically obtained $\chi_{kl,0}(i_k,j_l)$ has M eigenvalues which are close to zero; the other eigenvalues are positive and much larger. The numerical procedure is then simple in practice, and is only to omit the terms with the smallest M eigenvalues.

Appendix C: Treatment of $\rho_k \log\left(\rho_k/\rho_{k,0}\right)$ with Vanishing $\rho_{k,0}$

The free-energy functional of eqs. (12)–(21) involves a term with form of $\rho_k(\epsilon_k)\log\left(\rho_k(\epsilon_k)/\rho_{k,0}(\epsilon_k)\right)$, where ρ_k and $\rho_{k,0}$ are the distribution functions in the solution and reference-solvent systems, respectively. A numerical problem then arises when an ϵ_k domain with $\rho_k(\epsilon_k) \neq 0$ and $\rho_{k,0}(\epsilon_k) = 0$ exists. In this Appendix, we present a method of interpolation/extrapolation for treating such a problematic ϵ_k domain. The solvent-mediated part $\omega_k(\epsilon_k)$ of the solute–solvent potential of mean force in the energy representation is defined by eq. (17). We determine $\omega_k(\epsilon_k)$ at an ϵ_k with $\rho_k(\epsilon_k) \neq 0$ and $\rho_{k,0}(\epsilon_k) = 0$ through linear interpolation/extrapolation of the $\omega_k(\eta_k)$ in η_k domain with $\rho_k(\eta_k) \neq 0$ and $\rho_{k,0}(\eta_k) \neq 0$. Once $\omega_k(\epsilon_k)$ is fixed, $\log\left(\rho_k(\epsilon_k)/\rho_{k,0}(\epsilon_k)\right)$ is provided by eq. (17). This scheme of interpolation/extrapolation was introduced in Appendix A of Ref. [31] and is described below. Since the scheme is common to all the solvent species identified by the subscript k, for notational simplicity we do not write the subscript in the following description of the interpolation/extrapolation scheme.

Our scheme of interpolation/extrapolation is carried out by minimizing

$$\sum_\eta W(\eta;\epsilon)\left(A(\epsilon)\eta + B(\epsilon) - \omega(\eta)\right)^2 \quad \text{(C1)}$$

with respect to the coefficients $A(\epsilon)$ and $B(\epsilon)$ according to a weighting function $W(\eta;\epsilon)$ defined as

$$W(\eta;\varepsilon) \propto \exp\left(-\beta|\eta-\epsilon|\right) \frac{\rho(\eta)\rho_0(\eta)}{\rho(\eta)+\rho_0(\eta)}. \quad \text{(C2)}$$

The $\omega(\epsilon)$ at the ϵ with $\rho(\epsilon) \neq 0$ and $\rho_0(\epsilon) = 0$ is then given by

$$\omega(\epsilon) = A(\epsilon)\epsilon + B(\epsilon). \quad \text{(C3)}$$

The weighting function of eq. (C2) is motivated by Ref. [66]. It is non-zero only at η with $\rho(\eta) \neq 0$ and $\rho_0(\eta) \neq 0$ and emphasizes the η domain in which both of $\rho(\eta)$ and $\rho_0(\eta)$ are large.

The exponential factor is incorporated to measure the "distance" of sampled η from the coordinate ε toward which the interpolation/extrapolation is done. Note that the coefficients $A(\epsilon)$ and $B(\epsilon)$ depend on ϵ since the weighting function has a factor with ϵ.

The interpolation/extrapolation scheme given above introduces another approximation in addition to the use of the free-energy functional. The validity of the approximate interpolation/extrapolation may then be numerically assessed through comparison to a free-energy calculation that resorts to an intermediate state. The intermediate state is defined as a subset of the configurations in the solution system in which all the values ϵ_k of solute–solvent pair energy corresponds to non-zero $\rho_{k,0}(\epsilon_k)$ obtained numerically in the reference solvent. To write the free-energy expression with the intermediate state, we use a characteristic function $\Theta(\psi,\mathbf{X})$, where ψ is the configuration of the solute molecule and $\mathbf{X}$ denotes the solvent configuration collectively. The function is defined as $\Theta(\psi,\mathbf{X})=1$ when $\rho_{k,0}(\epsilon_k)\neq 0$ for all the solute–solvent pair energies ϵ_k and as $\Theta(\psi,\mathbf{X})=0$ otherwise. When $\Delta\mu_1$ and $\Delta\mu_2$ denote the free-energy changes from the reference solvent to the intermediate state and from the intermediate state to the solution system of interest, respectively, the solvation free energy $\Delta\mu$ is decomposed through eq. (1) as

$$\Delta\mu=\Delta\mu_1+\Delta\mu_2, \tag{C4}$$

$$\exp(-\beta\Delta\mu_1)=\frac{\int d\psi d\mathbf{X}\Theta(\psi,\mathbf{X})\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}v_k(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}{\int d\psi d\mathbf{X}\Theta(\psi,\mathbf{X})\exp(-\beta\{\Psi(\psi)+U(\mathbf{X})\})}, \tag{C5}$$

$$\exp(-\beta\Delta\mu_2)=\frac{\int d\psi d\mathbf{X}\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}v_k(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}{\int d\psi d\mathbf{X}\Theta(\psi,\mathbf{X})\exp\left(-\beta\left\{\Psi(\psi)+\sum_{i,k}v_k(\psi,\mathbf{x}_{i,k})+U(\mathbf{X})\right\}\right)}, \tag{C6}$$

where $\mathbf{x}_{i,k}$ is the configuration of the ith molecule of the kth solvent species, $\Psi(\psi)$ is the one-body energy of the solute, $U(\mathbf{X})$ is the total energy among the solvent, and v_k is the interaction potential of interest between the solute and the kth solvent species. The intermediate state corresponds to the numerator in the right-hand side of eq. (C5) and the denominator in the right-hand side of eq. (C6). The denominator in the right-hand side of eq. (C5) is actually the reference-solvent system since $\Theta(\psi,\mathbf{X})=1$ for each $(\psi,\mathbf{X})$ in the reference solvent by definition. As seen from eq. (C6), the intermediate state is a subset of the configurations in the solution system conditioned by $\Theta(\psi,\mathbf{X})$.

$\Delta\mu_1$ can be computed without resorting to the approximate scheme of linear interpolation/extrapolation presented above. When the distribution functions are obtained through the conditional averages in the solution system with $\Theta(\psi,\mathbf{X})$, the method of energy representation can be used only with the approximation described in the subsection of "Approximate functional for the solvation free energy". There is no need of using the approximate interpolation/extrapolation due to the definition of the ensemble of the intermediate state. $\Delta\mu_2$ can be evaluated from

$$\Delta\mu_2=k_BT\log P \tag{C7}$$

where P is the probability that the solution system is at configuration with $\Theta(\psi,\mathbf{X})=1$. This is an exact expression, and no approximation is involved. Equations (C4)–(C7) can thus be used to assess the performance of the linear interpolation/extrapolation.

In the above scheme with the intermediate state, the **ermod** run for the reference solvent needs to be done before the solution run to obtain the distribution function $\rho_{k,0}(\epsilon_k)$ numerically. Still, $\Delta\mu_1$ and $\Delta\mu_2$ can be determined from a single run of the solution system. Indeed, $\Delta\mu_1$ is computed using the distribution functions conditioned by $\Theta(\psi,\mathbf{X})$ in the solution system, and $\Delta\mu_2$ is given by the probability counted during the run of the solution system. In Ref. [31], the performance of the linear interpolation/extrapolation was assessed with eqs. (C4)–(C7). It was seen for the water absorptions into polymers that the error introduced by the approximate interpolation/extrapolation is negligible within the precision of 0.1 kcal/mol.

Appendix D: Expression for Pair Energy

As seen in the subsection of "Approximate functional for the solvation free energy", the computation of the pair interaction energy between the solute and solvent is essential in the method of energy representation since the functional for the solvation free energy is constructed from distribution functions of the pair interaction energy. In this Appendix, we show the expression for the pair interaction energy when the interaction is a sum of the LJ and Coulombic terms and the electrostatic term is handled in the periodic boundary condition by the SPME method with the surrounding medium of infinite dielectric constant.[44] The pair-energy expression in the (full) Ewald method was given in Appendix B of Ref. [6]. The SPME method is an approximation to the Ewald method, and is widely used for large-scale simulations, in particular.

It is straightforward to construct the LJ part of the solute–solvent pair energy. The LJ interaction $u_{LJ,ij}$ between a pair of interaction sites (atoms) i and j is given by

$$u_{LJ,ij}=4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12}-\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right], \tag{D1}$$

where ϵ_{ij} and σ_{ij} are the LJ parameters of energy and length between the atoms i and j, respectively, and r_{ij} is the radial distance between the two sites evaluated usually in the minimum image convention when the periodic boundary condition is adopted. The LJ part in the solute–solvent pair energy can then be obtained simply by summing eq. (D1) over the sites in the solute–solvent pair of interest.

The derivation is more involved for the SPME pair energy. To obtain the pair-energy expression, we start with the electrostatic energy of the total system in the SPME method written as[44]

$$U_{\mathrm{SPME}} = U_{\mathrm{real}} + U_{\mathrm{recp}} + U_{\mathrm{corr}}, \tag{D2}$$

where U_{real} is the real-space energy, U_{recp} is the reciprocal-space energy, and U_{corr} is the correction term including the interaction of the particle with its own periodic images and neutralizing background. If the box size of the system is large enough compared to the cutoff length, the real-space energy is approximated as

$$U_{\mathrm{real}} = \sum_{i<j\cap(i,j)\notin\mathcal{M}} q_i q_j \frac{1-\mathrm{erf}\left(\kappa r_{ij}\right)}{r_{ij}}, \tag{D3}$$

where q_i is the electric charge on the ith atom, and the summation is done over the pairs not within the list of exclusion pairs $\mathcal{M}$. The list $\mathcal{M}$ usually contains pairs of neighboring bonded atoms (1–2 interaction) and of atoms bonded to the same atom (1–3 interaction). κ is the coefficient to determine the rate of decay of the real-space part of the electrostatic interaction, and $\mathrm{erf}(\cdot)$ is the Gauss error function defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp\left(-y^2\right) dy. \tag{D4}$$

The error function quickly approaches to unity in the limit of $x \to \infty$, and each term within eq. (D3) decreases fast with the distance r_{ij}.

The total reciprocal-space energy U_{recp} is given by

$$U_{\mathrm{recp}} = \frac{1}{2} \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} D(m_1,m_2,m_3) \mathcal{F}[Q](m_1,m_2,m_3) \times \mathcal{F}^*[Q](m_1,m_2,m_3), \tag{D5}$$

where (m_1,m_2,m_3) represents the integer coordinate in $K_1 \times K_2 \times K_3$ grids satisfying $0 \le m_i < K_i$, and (K_1,K_2,K_3) specifies the grid sizes. $\mathcal{F}[\cdot]$ is the 3-dimensional discrete Fourier transform through[3]

$$\mathcal{F}[f](m_1,m_2,m_3) = \sum_{k_1=0}^{K_1-1} \sum_{k_2=0}^{K_2-1} \sum_{k_3=0}^{K_3-1} f(k_1,k_2,k_3) \times \exp\left[-2\pi\imath\left(\frac{m_1k_1}{K_1} + \frac{m_2k_2}{K_2} + \frac{m_3k_3}{K_3}\right)\right], \tag{D6}$$

with $\imath = \sqrt{-1}$, and $\mathcal{F}^*$ denotes the complex conjugate. In the implementation of SPME, the discrete Fourier transform is performed by fast Fourier transform (FFT). With FFT, a 3-dimensional discrete Fourier transform of size $K_1 \times K_2 \times K_3$ can be performed in $O(K_1K_2K_3 \log K_1K_2K_3)$ time. $D(m_1,m_2,m_3)$ is a function introduced as

$$D(m_1,m_2,m_3) = \begin{cases} 0 & (m_1,m_2,m_3) = 0 \\ \dfrac{1}{\pi V \gamma(m_1,m_2,m_3)} \dfrac{\exp\left(-\pi^2\mathbf{v}^2/\kappa^2\right)}{\mathbf{v}^2} & \text{otherwise}, \end{cases} \tag{D7}$$

where $\gamma(m_1,m_2,m_3)$ is a normalizing factor which will be shown below, $\mathbf{v}$ is the 3-dimensional vector expressed as $\mathbf{v} = m_1'\mathbf{a}_1^* + m_2'\mathbf{a}_2^* + m_3'\mathbf{a}_3^*$, and $m_i' = m_i$ for $0 \le m_i \le K_i/2$ and $m_i' = m_i - K_i$ otherwise. The periodic cell is spanned by a set of lattice vectors $\mathbf{a}_i$ ($i = 1, 2, 3$), V is the volume of the unit cell equal to $\mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3)$, and $\mathbf{a}_i^*$ is the conjugate of the lattice vectors $\mathbf{a}_i$ satisfying $\mathbf{a}_i^* \cdot \mathbf{a}_j = \delta_{ij}$, where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. The grid charge distribution $Q(k_1,k_2,k_3)$ is given by

$$Q(k_1,k_2,k_3) = \sum_i Q_i(k_1,k_2,k_3), \tag{D8}$$

$$Q_i(k_1,k_2,k_3) = q_i \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} \sum_{l_3=-\infty}^{\infty} \prod_{d=1}^{3} M_n(u_{id}K_d - k_d + l_dK_d) \tag{D9}$$

Q_i can be considered as the atom-wise grid charge distribution with q_i being the charge on the ith atom. M_n is the nth order Cardinal B-spline function, defined recursively as

$$M_n(x) = \frac{x}{n-1} M_{n-1}(x) + \frac{n-x}{n-1} M_{n-1}(x-1) \tag{D10}$$

and $M_2(x) = 1 - |x-1|$ for $0 \le x \le 2$ and $M_2(x) = 0$ otherwise. Note that $M_n(x) = 0$ if $x \le 0$ or $x \ge n$. The order of spline interpolation n is typically chosen between 4 and 10. The normalized coordinate $\mathbf{u}_i = (u_{i1}, u_{i2}, u_{i3})$ is introduced as $\mathbf{u}_i = x_i\mathbf{a}_1^* + y_i\mathbf{a}_2^* + z_i\mathbf{a}_3^*$, where (x_i, y_i, z_i) is the coordinate of the ith atom. In eq. (D9), Q_i can be computed in $O(n^3)$ time and $O(n)$ memory space. The normalizing factor γ in eq. (D7) is represented with M_n as

$$\gamma(m_1,m_2,m_3) = \prod_{i=1}^{3} \left| \sum_{j=0}^{n-2} M_n(j+1) \exp\left(\frac{-2\pi\imath j m_i}{K_i}\right) \right|^2. \tag{D11}$$

Finally, the correction term reads

$$U_{\mathrm{corr}} = -\frac{\kappa}{\sqrt{\pi}} \sum_i q_i^2 - \sum_{i<j\cap(i,j)\in\mathcal{M}} q_i q_j \frac{\mathrm{erf}\left(\kappa r_{ij}\right)}{r_{ij}} - \frac{\pi}{2\kappa^2 V} \left(\sum_i q_i\right)^2. \tag{D12}$$

The first term removes the excess self-interaction of a charge. The second term is the correction for the pairs within the list of exclusion pairs $\mathcal{M}$; note that while the sum in eq. (D3) is taken over the (i,j) pairs not within $\mathcal{M}$, the sum in the second term of eq. (D12) is taken only over the (i,j) pairs contained in $\mathcal{M}$. The third term comes from the requirement that the Ewald

[3]It should be noted that the sign within the exponential of eq. (D6) is opposite to that of Ref. [44]. In this paper, we adopt the convention which is more prevalent among computational chemistry community.

electric potential (to be exact, the pair electrostatic energy introduced next) is averaged over the unit cell to zero.[45–47]

The interaction energy in SPME can now be cast into pairwise additive form. The pairwise electrostatic interaction energy $u_{\mathrm{pair},ij}$ between the ith and jth atoms $(i \neq j)$ is expressed as

$$u_{\mathrm{pair},ij} = u_{\mathrm{real},ij} + u_{\mathrm{recp},ij} + u_{\mathrm{corr},ij}, \tag{D13}$$

and the self-interaction energy $u_{\mathrm{self},i}$ of the ith atom is given by

$$u_{\mathrm{self},i} = \frac{1}{2} u_{\mathrm{recp},ii} + \frac{1}{2} u_{\mathrm{corr},ii}. \tag{D14}$$

The terms in the pair interaction of eq. (D13) and the self interaction of eq. (D14) are written as

$$u_{\mathrm{real},ij} = \delta_{(i,j)\notin\mathcal{M}} q_i q_j \frac{1-\mathrm{erf}\left(\kappa r_{ij}\right)}{r_{ij}}, \tag{D15}$$

$$u_{\mathrm{recp},ij} = \sum_{m_1=0}^{K_1-1} \sum_{m_2=0}^{K_2-1} \sum_{m_3=0}^{K_3-1} D(m_1,m_2,m_3) F[Q_i](m_1,m_2,m_3) \mathcal{F}^*[Q_j](m_1,m_2,m_3), \tag{D16}$$

$$u_{\mathrm{corr},ij} = -\delta_{(i,j)\in\mathcal{M}} q_i q_j \frac{\mathrm{erf}\left(\kappa r_{ij}\right)}{r_{ij}} - \frac{\pi q_i q_j}{\kappa^2 V}, \tag{D17}$$

where $\delta_{(i,j)\notin\mathcal{M}}$ is equal to 1 for (i,j) pair not within the exclusion list $\mathcal{M}$ and is 0 otherwise, and $\delta_{(i,j)\in\mathcal{M}}$ is equal to 1 for (i,j) within $\mathcal{M}$ and is 0 otherwise. $u_{\mathrm{corr},ii}$ of eq. (D14) is obtained by taking the limit of $r_{ii} \to 0$ in eq. (D17) and is given by

$$u_{\mathrm{corr},ii} = -q_i^2 \lim_{r_{ii}\to 0} \frac{\mathrm{erf}\left(\kappa r_{ii}\right)}{r_{ii}} - \frac{\pi q_i^2}{\kappa^2 V} = \left(-\frac{2\kappa}{\sqrt{\pi}} - \frac{\pi}{\kappa^2 V}\right) q_i^2. \tag{D18}$$

The SPME part in the solute–solvent pair energy can be obtained by summing eqs. (D15)–(D17) over the sites in the solute–solvent pair of interest. If eq. (D16) is implemented as is, however, the reciprocal-space part demands too much computational cost since FFT needs to be repeated over all the atoms of interest. Instead of translating eq. (D16) directly into the program, we use the linearity of Fourier transform to accelerate the calculation. The details are described in Appendix E.

Appendix E: Efficient Calculation of Solute–Solvent Pair Energy

When the solvation free energy is to be obtained through the functional in the subsection of "Approximate functional for the solvation free energy", the pair interaction energy of the solute molecule needs to be computed against each solvent molecule. An efficient algorithm of pair-energy calculation is thus a key part of the **ermod** program, and is described in this Appendix. It should be stressed, though, that the energy-representation method uses only the value of solute–solvent pair interaction and does not resort to the functional form of the interaction potential. Due to this property, the method can be used for any form of pair interaction potential, including one with spatially diffuse charge distribution.[7,8,38] Still, the molecular simulation is mostly done using the van der Waals interaction with LJ potential of 12-6 type and the electrostatic interaction with a set of point charges. In **ermod**, an efficient algorithm is implemented for calculating those LJ and electrostatic interactions.

ermod can handle both periodic and nonperiodic systems, and contains two efficient algorithms for pair-energy calculation. One is for the interaction with finite cutoff. This is applied to the LJ interaction, the bare Coulombic interaction, and the real-space part of the SPME potential. The LJ interaction can be treated also with switching function.[59,67] The other of the algorithms is for the reciprocal-space part of the SPME potential in periodic system.

According to eqs. (D1) and (D13), the pair-interaction energy between the solute molecule and the sth solvent molecule is given in the SPME method by

$$\sum_{i\in\text{solute}} \sum_{j\in s\text{th solvent}} \left(u_{\mathrm{LJ},ij} + u_{\mathrm{pair},ij}\right) = \sum_{i\in\text{solute}} \sum_{j\in s\text{th solvent}} \left(u_{\mathrm{LJ},ij} + u_{\mathrm{real},ij} + u_{\mathrm{corr},ij}\right) + \sum_{i\in\text{solute}} \sum_{j\in s\text{th solvent}} u_{\mathrm{recp},ij}, \tag{E1}$$

where i runs over the atoms (interaction sites) within the solute, j is for the atoms belonging to the sth solvent molecule, and $u_{\mathrm{LJ},ij}$, $u_{\mathrm{real},ij}$, $u_{\mathrm{corr},ij}$, and $u_{\mathrm{recp},ij}$ in the right-hand side are given by eqs. (D1), (D15), (D17), and (D16), respectively. The treatment of the first three is described in the subsection of "Interactions with cutoff" within the present Appendix, and a computation scheme of the reciprocal-space part of the SPME interaction is developed in the subsection of "SPME reciprocal-space energy". When the bare form is adopted for the Coulombic interaction with finite cutoff, only the method in the former subsection is utilized.

Interactions with cutoff

The LJ interaction and the real-space part of the SPME interaction are typically short-ranged. Their contributions to the solute–solvent pair energy are determined through the simple summation with eq. (E1) of the pairwise values of $u_{\mathrm{LJ},ij}$, $u_{\mathrm{real},ij}$, and $u_{\mathrm{corr},ij}$ on the atom-atom basis. The time complexity of this procedure is $O(N_{\mathrm{solute}} N_{\mathrm{solvent}})$, though, where N_{solute} is the number of atoms contained in the (single) solute molecule and N_{solvent} is the total number of solvent atoms in the whole system. The computational cost may then be expensive when the solute and/or solvent is large (for example, protein).

To reduce the computational cost, we implemented a cell-link list based method[68,69] in **ermod**. In this method, the system is decomposed into a set of cells and each atom (conceptually) belongs to one of the cells. The minimum distances between the pairs of cells are then computed, and the cell pairs within the

cutoff distance are marked as possibly interacting pairs of cells. Only the atom pairs belonging to possibly interacting cell pairs are examined for whether they are within the interaction cutoff length or not.

It should be noted that in many state-of-art molecular simulation software implementations, the list of atoms close to each other are maintained as neighbor list.[3,4,70] The neighbor list is updated every some steps of MD calculation, to reduce the number of atom pairs to be treated for energy and force evaluation. The neighbor-list method is not adopted in **ermod**, however, since the time interval between snapshots of **ermod**'s input trajectory can be set arbitrarily; two "adjacent" snapshots may not be close in time and the atoms in them may be displaced significantly.

SPME reciprocal-space energy

The reciprocal-space contribution to the SPME energy is given by eq. (D16) for a pair of atoms, and is expressed as

$$u_{\mathrm{recp},s}=\sum_{i\in\mathrm{solute}}\sum_{j\in s\mathrm{th\ solvent}}u_{\mathrm{recp},ij} \tag{E2}$$

between the solute molecule and the sth solvent molecule. If the sum is straightforwardly taken in eq. (E2), however, the Fourier transform needs to be conducted for each atom through eq. (D16). This requires a large cost of computation and is avoided in **ermod**.[4] Instead, the sum of eq. (E2) is rearranged to allow faster calculation by substituting eq. (D16) into eq. (E2) and using the linearity of Fourier transform as

$$u_{\mathrm{recp},s}=\sum_{m_1=0}^{K_1-1}\sum_{m_2=0}^{K_2-1}\sum_{m_3=0}^{K_3-1}D(m_1,m_2,m_3)\mathcal{F}\left[\sum_{i\in\mathrm{solute}}Q_i\right](m_1,m_2,m_3)\times\mathcal{F}^*\left[\sum_{j\in s\mathrm{th\ solvent}}Q_j\right](m_1,m_2,m_3) \tag{E3}$$

$$=\sum_{m_1=0}^{K_1-1}\sum_{m_2=0}^{K_2-1}\sum_{m_3=0}^{K_3-1}\left\{D(m_1,m_2,m_3)\mathcal{F}\left[\sum_{i\in\mathrm{solute}}Q_i\right](m_1,m_2,m_3)\times\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\sum_{k_3=0}^{K_3-1}\sum_{j\in s\mathrm{th\ solvent}}Q_j(k_1,k_2,k_3)\exp\left(2\pi\imath\left(\frac{m_1k_1}{K_1}+\frac{m_2k_2}{K_2}+\frac{m_3k_3}{K_3}\right)\right)\right\} \tag{E4}$$

$$=\sum_{k_1=0}^{K_1-1}\sum_{k_2=0}^{K_2-1}\sum_{k_3=0}^{K_3-1}\left[\sum_{j\in s\mathrm{th\ solvent}}Q_j(k_1,k_2,k_3)\right]G_{\mathrm{solute}}(k_1,k_2,k_3) \tag{E5}$$

where

$$G_{\mathrm{solute}}(k_1,k_2,k_3)=\mathcal{F}^{-1}\left[D\cdot\mathcal{F}\left[\sum_{i\in\mathrm{solute}}Q_i\right]\right](k_1,k_2,k_3). \tag{E6}$$

From eq. (E3) to (E4), we used $Q_i^*=Q_i$. The inverse 3-dimensional, discrete, unnormalized Fourier transform $\mathcal{F}^{-1}$ of a function f is introduced as

$$\mathcal{F}^{-1}[f](k_1,k_2,k_3)=\sum_{m_1=0}^{K_1-1}\sum_{m_2=0}^{K_2-1}\sum_{m_3=0}^{K_3-1}f(m_1,m_2,m_3)\times\exp\left[2\pi\imath\left(\frac{m_1k_1}{K_1}+\frac{m_2k_2}{K_2}+\frac{m_3k_3}{K_3}\right)\right]. \tag{E7}$$

In **ermod**, the reciprocal-space part of the solute–solvent pair energy is obtained through eq. (E5). For each snapshot of solute–solvent configuration, G_{solute} of eq. (E6) is calculated by combining a real-to-complex 3-dimensional FFT and a complex-to-real 3-dimensional inverse FFT in $O(K_1K_2K_3\log K_1K_2K_3)$ time and $O(K_1K_2K_3)$ memory space. The Fourier transform is thus performed only for the solute molecule and not for the solvent molecules; FFT needs to be carried out only twice per solute–solvent configuration. As noted with respect to eq. (D9), the construction of Q_i (for both cases of solute and solvent) needs $O(n^3)$ time and $O(n)$ memory space. The total reciprocal-space pair energy can thus be calculated in $O(n^3N_{\mathrm{solute}}+n^3M_{\mathrm{solvent}}+K_1K_2K_3\log K_1K_2K_3)$ time and $O(n+K_1K_2K_3)$ space, where N_{solute} and M_{solvent} are the numbers of atoms (interaction sites) in the single solute and solvent molecules, respectively.

As described with eq. (11), the solute is inserted into the reference solvent as a test particle. The test-particle insertion is typically carried out many times (1000 as the default in the ERmod program) for each solvent configuration sampled. Still, it is sufficient for a single solvent configuration used for insertion that

$$\sum_{j\in s\mathrm{th\ solvent}}Q_j(k_1,k_2,k_3)$$

within eq. (E5) be computed only once for each solvent molecule. This is because the solvent coordinates and the box size remain unchanged during a set of test-particle insertions into a single solvent configuration. When the number of insertions is large, the calculation time is then saved for the reference-solvent system with the cost of small memory consumption.

Appendix F: Self Energy of the Solute

As described in the subsection of "Finite-size correction for the electrostatic interaction in periodic boundary condition", the self energy of the solute is the difference in the one-body energy of the solute between the solution and reference-solvent systems. This difference is denoted as v_{FSC} in that section and is given by

[4]When the sum over the sites within the molecule [i and j in eq. (D16)] is taken first through eq. (D8), the number of Fourier transforms is equal to the number of molecules in the system. This can be still expensive.

$$v_{\mathrm{FSC}} = \sum_{i\in\mathrm{solute}} u_{\mathrm{self},i} + \sum_{i,j\in\mathrm{solute}\cap i<j} u_{\mathrm{pair},ij} - \sum_{i,j\in\mathrm{solute}\cap i<j\cap(i,j)\notin\mathcal{M}} \frac{q_i q_j}{r_{ij}}, \quad \text{(F1)}$$

where the sums are taken over the atoms (interaction sites) within the solute molecule, $u_{\mathrm{self},i}$ and $u_{\mathrm{pair},ij}$ are given by eqs. (D14) and (D13), respectively, and $\mathcal{M}$ is the list of exclusion pairs. The first and second terms of eq. (F1) are of lattice-sum form and are the electrostatic energy within the solute which stays in the solution system. The third term refers to the solute at isolation and corresponds to the electrostatic interaction in the bare form. With eqs. (D15), (D17), and (D18), v_{FSC} reduces to

$$v_{\mathrm{FSC}} = \frac{1}{2}\sum_{i,j\in\mathrm{solute}} \left(u_{\mathrm{recp},ij} - q_i q_j \frac{\mathrm{erf}\left(\kappa r_{ij}\right)}{r_{ij}} \right) - \frac{\pi}{2\kappa^2 V}\left(\sum_i q_i\right)^2, \quad \text{(F2)}$$

where (i,j) and (j,i) are both in the sum in the first term and $\mathrm{erf}\left(\kappa r_{ij}\right)/r_{ij}$ is understood as $2\kappa/\sqrt{\pi}$ when $i=j$ and $r_{ij}=0$ by virtue of eq. (D4). The exclusion list is absent in eq. (F2). Indeed, the self energy is the interaction of a molecule with its own images and neutralizing background and can be treated without taking the exclusion list into explicit account.

Keywords: solvation free energy · energy representation · solution theory · molecular simulation · distribution function

How to cite this article: S. Sakuraba, N. Matubayasi, *J. Comput. Chem.* **2014**, DOI: 10.1002/jcc.23651

[1] A. Ben-Naim, *Solvation Thermodynamics*; Plenum Press: New York, **1987**.
[2] F. Hirata, ed. *Molecular Theory of Solvation*; Kluwer Academic Publishers: Dordrecht, The Netherlands, **2003**.
[3] M. P. Allen, D. J. Tildesley, *Computer Simulation of Liquids*; Oxford Science Publications: New York, **1987**.
[4] D. Frenkel, B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*; Academic Press, London, **1996**.
[5] N. Matubayasi, K. K. Liang, M. Nakahara, *J. Chem. Phys.* **2006**, *124*, 154908.
[6] N. Matubayasi, W. Shinoda, M. Nakahara, *J. Chem. Phys.* **2008**, *128*, 195107.
[7] H. Takahashi, H. Ohno, R. Kishi, M. Nakano, N. Matubayasi, *Chem. Phys. Lett.* **2008**, *456*, 176.
[8] H. Takahashi, H. Ohno, R. Kishi, M. Nakano, N. Matubayasi, *J. Chem. Phys.* **2008**, *129*, 205103.
[9] R. M. Levy, M. Belhadj, D. B. Kitchen, *J. Chem. Phys.* **1991**, *95*, 3627.
[10] V. Luzhkov, A. Warshel, *J. Comput. Chem.* **1992**, *13*, 199.
[11] J. Åqvist, C. Medina, J. E. Samuelsson, *Protein Eng.* **1994**, *7*, 385.
[12] H. A. Carlson, W. L. Jorgensen, *J. Phys. Chem.* **1995**, *99*, 10667.
[13] S. M. Kast, *Phys. Chem. Chem. Phys.* **2001**, *3*, 5087.
[14] M. V. Vener, I. V. Leontyev, Y. A. Dyakov, M. V. Basilevsky, M. D. Newton, *J. Phys. Chem. B* **2002**, *106*, 13078.
[15] I. Fdez. Galván, M. L. Sanchez, M. E. Martin, F. J. Olivares del Valle, M. A. Aguilar, *J. Chem. Phys.* **2003**, *118*, 255.
[16] H. Freedman, T. N. Truong, *J. Chem. Phys.* **2004**, *121*, 2187.
[17] M. Higashi, S. Hayashi, S. Kato, *J. Chem. Phys.* **2007**, *126*, 144503.
[18] G. N. Chuev, M. V. Fedorov, J. Crain, *Chem. Phys. Lett.* **2007**, *448*, 198.
[19] T. Yamamoto, *J. Chem. Phys.* **2008**, *129*, 244104.
[20] A. I. Frolov, E. L. Ratkova, D. S. Palmer, M. V. Fedorov, *J. Phys. Chem. B* **2011**, *115*, 6011.
[21] B. Lin, B. M. Pettitt, *J. Comput. Chem.* **2011**, *32*, 878.
[22] N. Matubayasi, M. Nakahara, *J. Chem. Phys.* **2000**, *113*, 6070.
[23] (a) N. Matubayasi, M. Nakahara, *J. Chem. Phys.* **2002**, *117*, 3605; (b) N. Matubayasi, M. Nakahara, J. *Chem. Phys.* **2003**, *118*, 2446 (erratum).
[24] N. Matubayasi, M. Nakahara, *J. Chem. Phys.* **2003**, *119*, 9686.
[25] M. R. Shirts, J. W. Pitera, W. C. Swope, V. S. Pande, *J. Chem. Phys.* **2003**, *119*, 5740.
[26] M. R. Shirts, V. S. Pande, *J. Chem. Phys.* **2005**, *122*, 134508.
[27] Y. Karino, M. V. Fedorov, N. Matubayasi, *Chem. Phys. Lett.* **2010**, *496*, 351.
[28] Y. Karino, N. Matubayasi, *Phys. Chem. Chem. Phys.* **2013**, *15*, 4377.
[29] N. Matubayasi, M. Nakahara, *J. Chem. Phys.* **2005**, *122*, 074509.
[30] M. Takeuchi, N. Matubayasi, Y. Kameda, B. Minofar, S. Ishiguro, Y. Umebayashi, *J. Phys. Chem. B* **2012**, *116*, 6476.
[31] T. Kawakami, I. Shigemoto, N. Matubayasi, *J. Chem. Phys.* **2012**, *137*, 234903.
[32] H. Saito, N. Matubayasi, K. Nishikawa, H. Nagao, *Chem. Phys. Lett.* **2010**, *497*, 218.
[33] Y. Karino, N. Matubayasi, *J. Chem. Phys.* **2011**, *134*, 041105.
[34] K. Takemura, H. Guo, S. Sakuraba, N. Matubayasi, A. Kitao, *J. Chem. Phys.* **2012**, *137*, 215105.
[35] T. Mizukami, H. Saito, S. Kawamoto, T. Miyakawa, M. Iwayama, M. Takasu, H. Nagao, *Int. J. Quantum Chem.* **2012**, *112*, 344.
[36] K. Takemura, R. R. Burri, T. Ishikawa, T. Ishikura, S. Sakuraba, N. Matubayasi, K. Kuwata, A. Kitao, *Chem. Phys. Lett.* **2013**, *559*, 94.
[37] H. Saito, M. Iwayama, T. Mizukami, J. Kang, M. Tateno, H. Nagao, *Chem. Phys. Lett.* **2013**, *556*, 297.
[38] H. Takahashi, N. Matubayasi, M. Nakahara, T. Nitta, *J. Chem. Phys.* **2004**, *121*, 3989.
[39] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kalé, K. Schulten, *J. Comput Chem.* **2005**, *26*, 1781.
[40] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435.
[41] D. A. Case, T. A. Darden, T. E. Cheatham, III, C. L. Simmerling, J. Wang, R. E. Duke, R. Luo, R. C. Walker, K. M. M. W. Zhang, B. Roberts, S. Hayik, A. Roitberg, G. Seabra, J. Swails, A. W. Goetz, I. Kolossváry, K. F. Wong, F. Paesani, J. Vanicek, R. M. Wolf, J. Liu, X. Wu, S. R. Brozell, T. Steinbrecher, H. Gohlke, Q. Cai, X. Ye, J. Wang, M.-J. Hsieh, G. Cui, D. R. Roe, D. H. Mathews, M. G. Seetin, R. Salomon-Ferrer, C. Sagui, V. Babin, T. Luchko, S. Gusarov, A. Kovalenko, P. A. Kollman, *AMBER 12*; University of California: San Francisco, **2012**.
[42] S. Pronk, S. Páll, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, E. Lindahl, *Bioinformatics* **2013**, *29*, 845.
[43] T. Darden, D. York, L. Pedersen, *J. Chem. Phys.* **1993**, *98*, 10089.
[44] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, L. Pedersen, *J. Chem. Phys.* **1995**, *103*, 8577.
[45] F. Figueirido, G. S. Del Buono, R. M. Levy, *J. Chem. Phys.* **1995**, *103*, 6133.
[46] G. Hummer, L. R. Pratt, A. E. García, *J. Phys. Chem.* **1996**, *100*, 1206.
[47] F. Figueirido, G. S. Del Buono, R. M. Levy, *J. Phys. Chem. B* **1997**, *101*, 5622.
[48] A. D. MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Gao, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiórkiewicz-Kuczera, D. Yin, M. Karplus, *J. Phys. Chem. B* **1998**, *102*, 3586.
[49] J. B. Klauda, R. M. Venable, J. A. Freites, J. W. O'Connor, D. J. Tobias, C. Mondragon-Ramirez, I. Vorobyov, A. D. MacKerell, Jr., R. W. Pastor, *J. Phys. Chem. B* **2010**, *114*, 7830.
[50] J. W. Ponder, D. A. Case, *Adv. Protein Chem.* **2003**, *66*, 27.
[51] V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, C. Simmerling, *Proteins: Struct. Funct. Bioinf.* **2006**, *65*, 712.

[52] W. L. Jorgensen, D. S. Maxwell, J. Tirado-Rives, *J. Am. Chem. Soc.* **1996**, *118*, 11225.
[53] G. A. Kaminski, R. A. Friesner, J. Tirado-Rives, W. L. Jorgensen, *J. Phys. Chem. B* **2001**, *105*, 6474.
[54] W. F. van Gunsteren, S. R. Billeter, A. A. Eising, P. H. Hünenberger, P. Krüger, A. E. Mark, W. R. P. Scott, I. G. Tironi, *Biomolecular Simulation: The GROMOS96 Manual and User Guide*; Hochschulverlag AG an der ETH Zürich: Zürich, Switzerland, **1996**.
[55] L. D. Schuler, X. Daura, W. F. van Gunsteren, *J. Comput. Chem.* **2001**, *22*, 1205.
[56] P. Bjelkmar, P. Larsson, M. A. Cuendet, B. Hess, E. Lindahl, *J. Chem. Theory Comput.* **2010**, *6*, 459.
[57] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, M. L. Klein, *J. Chem. Phys.* **1983**, *79*, 926.
[58] J.-P. Hansen, I. R. McDonald, *Theory of Simple Liquids*, 3rd ed.; Academic Press: London, **2006**.
[59] D. van der Spoel, P. J. van Maaren, *J. Chem. Theory Comput*. **2006**, *2*, 1.
[60] S. Miyamoto, P. A. Kollman, *J. Comput. Chem.* **1992**, *13*, 952.
[61] B. Hess, H. Bekker, H. J. C. Berendsen, J. G. E. M. Fraaije, *J. Comput. Chem.* **1997**, *18*, 1463.
[62] M. Parrinello, A. Rahman, *J. Appl. Phys.* **1981**, *52*, 7182.
[63] S. Nosé, M. L. Klein, *Mol. Phys.* **1983**, *50*, 1055.
[64] W. F. van Gunsteren, H. J. C. Berendsen, *Mol. Simul.* **1988**, *1*, 173.
[65] H. C. Andersen, J. D. Weeks, D. Chandler, *Phys. Rev. A* **1971**, *4*, 1597.
[66] S. Sakuraba, N. Matubayasi, *J. Chem. Phys.* **2011**, *135*, 114108.
[67] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus, *J. Comput. Chem.* **1983**, *4*, 187.
[68] B. Quentrec, C. Brot, *J. Comput. Phys.* **1973**, *13*, 430.
[69] R. W. Hockney, J. W. Eastwood, *Computer Simulation Using Particles*; McGraw-Hill: New York, **1981**.
[70] L. Verlet, *Phys. Rev.* **1967**, *159*, 98.

Received: 12 March 2014
Revised: 10 May 2014
Accepted: 16 May 2014
Published online on 00 Month 2014