Comparing Solvent Models for Molecular Dynamics of Protein

Sikandar Y. Mashayak and David E. Tanner

May 12, 2011

Summary

Discrete nature of water plays critical role in protein folding and thermodynamics. Explicit all-atom modeling of solvent severely limits the length and time scales of molecular dynamics simulations of protein. In this project, we evaluated accuracy and computational efficiency of four different solvent models, all-atom solvent models (SPC/E and TIP3P), Generalized Born Implicit Solvent, and Coarse- Grained solvent model. CG force-field, for solvent interactions, is determined using Force-matching with exclusion technique. Explicit and implicit solvent simulations are compared with vacuum to study the effects of solvent on protein structure.

1 Introduction

Molecular dynamics (MD) simulations are important tools for studying properties of biological macromolecules, such as protein. Advancements in computer architecture and availability of efficient programs have led to increase in the number of studies using molecular dynamics to simulate the properties of biological macromolecules[1, 2]. However, MD simulations still have limitations on the time and length scales that can be studied. Hence, it is challenging to simulate relevant biological phenomena, which occur at varied time (spans 20 orders of magnitude) and length scales (spans over six orders of magnitude) [3].

One of the approaches to decrease computational requirement is to develop computationally efficient force fields. Force field to model solvent is one of the critical aspects of MD simulations of protein. Water constitutes the environment in which proteins interact. Studies using MD simulations have been performed to investigate the role of water in protein folding and its properties[4, 5]. These studies have found that water-induced effects are significant in protein folding and thermodynamics. Hence, it is important to incorporate physically accurate and computationally efficient solvent models in protein MD simulations.

The objective of this project is to compare accuracy and computational efficiency of different water models in protein MD simulations. We did five different simulations of simple test protein, one with vacuum, other two with different explicit solvent models (SPC/E and TIP3P), fourth with Generalized Born Implicit Solvent. For the fifth case we used Coarse-Grained single site model for water in which interactions between water-water and water-protein have been coarse-grained using Force-Matching technique. Using each of these models, properties such as root mean square deviation (RMSD) and solvent accessible surface areas (SAS) of protein are computed and compared.

The remainder of this report is organized as follows. In section 2, description of our own test protein and different water models are discussed. Also, the Force-Matching technique to coarse-grain water interactions is described in brief. The methods and algorithms to perform MD simulations are described in section 3. In section 4, the results obtained using different solvent models are presented. Finally in section 5, conclusions are drawn and possible future work is described.

2 Models

2.1 Protein

To compare different solvent models, we have built our own test protein; see Fig. 1. Test protein is made from four alpha helices stacked two on two. Original tertiary and secondary structure was taken from biological protein. All 65 residues were converted to ALA (to make coarse graining procedure easier)using VMD. Using a simple protein build from alpha helices, makes verification of a solvent model easier; our protein only has 4 secondary structure domains to track, making comparison very straight forward.

2.2 Explicit Solvent

In the explicit solvent simulations of protein, water is represented by all-atom force field models. Due to its ubiquity and importance water is the most investigated liquid by molecular simulations[6]. Currently there are several water models being used in bio-molecular simulations such as SPC,SPC/E,TIP3P,TIP5P [7]. Each of these models are optimized to fit one or more physical properties of water, such as the radial distribution function, diffusivity, density anomaly etc. But none of these models can simultaneously reproduce all the properties of water.

In this study we selected TIP3P and SPC/E models for comparisons, as shown in Fig. 2. These simple, rigid and non-polarizable three site models remain the most commonly used water models in simulations of protein. Each site has a partial charge to account for Coulomb interactions, while oxygen atoms interact via a Lennard-Jones potential. Hence, the total interaction potential between two water molecules is computed by Eq. 1. Intramolecular interactions are neglected and geometry of the molecule is kept constant, as shown in

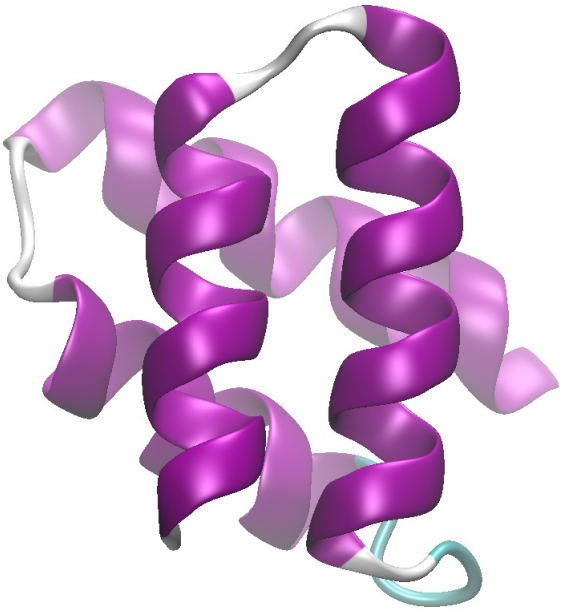

Figure 1: Structure of test protein. Test protein is made from four alpha helices stacked two on two. Original tertiary and secondary structure was taken from biological protein. All 65 residues were converted to ALA (to make coarse graining procedure easier)using VMD.

Fig. 3. Interactions and geometry parameters are listed in Table 2.2.

Figure 2: Protein in Explicit Solvent.

$$V_{ab} = 4\epsilon_{oo}\left(\frac{\sigma_{oo}^{12}}{r_{oo}^{12}} - \frac{\sigma_{oo}^{6}}{r_{oo}^{6}}\right) + \sum_{i\in a}\sum_{j\in b}\frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \tag{1}$$

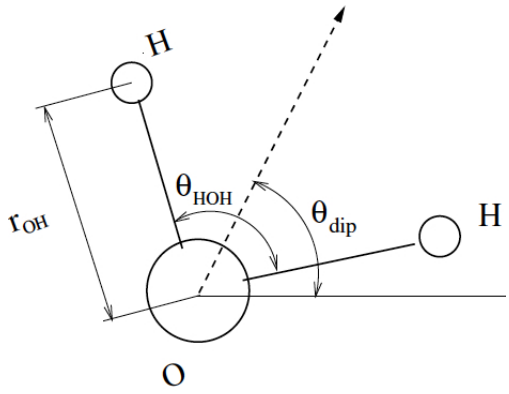

Figure 3: Three Sites Water Model

Table 1: Parameters of SPC/E and TIP3P Water Models.

	TIP3P	SPC/E
r_{OH}(Å)	0.9572	1.0
$HOH(^\circ)$	104.52	109.47
σ_{OO}(nm)	0.315058	0.316557
ϵ_{OO}(kJ/mol)	0.636386	0.650194
q_O(e)	-0.834	-0.8476
q_H(e)	+0.417	+0.4238

It is realistic to include water molecules explicitly in simulations of protein. But the presence of water molecules in the system increases the number of degrees of freedom by more than 1000. Although, discrete nature of water plays important role in protein thermodynamics, in most cases objective of protein simulations is to study properties of protein and not the detailed atomistic behavior of solvent molecules. In such cases, averaging over unimportant atomistic details of solvent greatly decreases the number of degrees of freedom in the system and improves the computational efficiency.

We used two approaches for averaging over unimportant solvent degrees of freedom. In one, we used Generalized Born implicit solvent model. While in second approach, we have implemented coarse-grained representation of solvent molecules, in which all atoms of water are grouped together in one site located at its center of mass. The descriptions of these models follow.

2.3 Coarse-Grained Solvent

Coarse-grained (CG) models are less structured representation of a molecule obtained by mapping two or more atoms onto a single interaction site. Signif-

icant speed ups are obtained due to lesser degrees of freedom, simpler, softer potentials and larger time steps.

In the literature, there are various different CG models proposed for bulk water [8, 9], as well as for water in protein neighborhood [10]. Implementation of CG models involves two major steps. The first is mapping a all-atom system onto a coarse-grained representation. The second is optimizing the effective interactions between coarse-grained sites. Similar to empirical potentials scheme, coarse-grained interactions are optimized to reproduce certain structural and thermodynamic properties of reference all-atom system. This second stage is more challenging and there are several systematic coarse-graining techniques developed to address it, such as iterative Boltzmann inversion, force-matching, inverse Monte Carlo etc. [11].

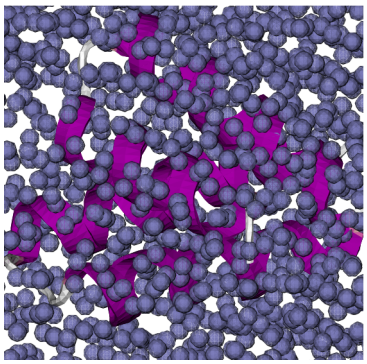

Figure 4: Protein in Coarse-Grained Solvent.

For this study, we use bulk protein-SPC/E water as our reference all-atom system for coarse-graining. Entire water molecule is mapped to one coarse-grained bead located at center of mass of molecule, while protein is represented by its original all-atom model as shown in Fig. 4. So we are required to derive effective interactions between water-water CG beads and between water CG bead and protein atoms.

Force field between these CG sites is calculated by Force-Matching (FM) with exclusions using VOTCA package[12]. Below we briefly describe what is FM technique and how exclusion approach can be used to derive CG force-field.

2.3.1 Force-Matching

The basic idea behind FM technique is to derive coarse-grained force field which reproduces the forces acting on coarse-grained sites as closely as possible. The force field is optimized by minimizing the difference between reference system forces and CG forces

$$\chi^2 = \sum_{i=1}^{L} \sum_{l=1}^{M} |\overrightarrow{F}_{il}^{AA} - \overrightarrow{F}_{il}^{CG}|^2 \tag{2}$$

where $\overrightarrow{F}_{il}^{AA}$ is the mapped reference force on CG site i and $\overrightarrow{F}_{il}^{CG}$ is the CG force. The sum is over L snapshots of reference trajectory and M coarse-grained beads.

Detail description of FM technique can be found in [9].

2.3.2 FM with Exclusions

For complicated molecules, deriving all CG interactions at once can be difficult. Different hybrid approaches to address this issue have been discussed in [12] .

In FM with *exclusions* approach only reference forces that need to be coarse-grained are recalculated from the reference all-atom trajectory. Then, force-matching is applied to these recalculated forces. Non-bonded and bonded forces which do not contribute to CG interactions under consideration are excluded.

Detail description of application of these technique to implement CG solvent model for our system is given in section 3.2.

2.4 Generalized Born Implicit Solvent

Generalized Born implicit solvent is a method for calculating solvation energies and forces on biomolecules for molecular dynamics. The method explained here was developed previously by [14] and was implemented in NAMD by David Tanner last year but completed early 2011. The following terms are required in the generalized Born energy and force calculation below:

r_{ij} - distance between atoms i and j; calculated from atom coordinates.

κ - debye screening length; calculated from ion concentration, $\kappa^{-1} = \sqrt{\frac{\epsilon_0 \epsilon_p kT}{2N_A e^2 I}}$; $\kappa^{-1} = 10\mathring{A}$ for 0.1 M monovalent salt.

ϵ_s - dielectric constant of solvent; default is $\epsilon_s = 80$.

ϵ_p - dielectric constant of protein; default is $\epsilon_p = 1$.

α_i - Born radius of atom i; defined below.

ρ_i - vdW radius of atom i; read from table.

ρ_0 - Born radius offset; $\rho_0 = 0.09\mathring{A}$.

$\rho_{i0} = \rho_i - \rho_0$

$\rho_{is} = \rho_{i0} S_{ij}$

S_{ij} - radius scaling factor; read from table.

H_{ij} - pairwise descreening; defined below.

k_e - coulomb's constant, $\frac{1}{4\pi\epsilon_0}$, 332.063711 kcal Å/ e^2.

$\{\delta, \beta, \gamma\} = \{0.8, 0, 2.91\}$ or $\{1.0, 0.8, 4.85\}$

2.4.1 Calculating Energy

The total electrostatic energy for a system of charges in a dielectric is the sum of Coulomb and generalized Born (GB) energies;

$$E_{\mathrm{T}}^{\mathrm{Elec}} = E_{\mathrm{T}}^{\mathrm{Coul}} - E_{\mathrm{T}}^{\mathrm{GB}} \ . \tag{3}$$

The total generalized Born energy for the system of charges is the sum over pairwise energies and self energies according to

$$E_{\mathrm{T}}^{\mathrm{GB}} = \sum_i \sum_{j>i} E_{ij}^{\mathrm{GB}} + \sum_i E_{ii}^{\mathrm{GB}} \tag{4}$$

where the pair and self energies are defined by

$$E_{ij}^{\mathrm{GB}} = -k_e D_{ij} \frac{q_i q_j}{f_{ij}^{\mathrm{GB}}} \ . \tag{5}$$

The pairwise dielectric term is given by

$$D_{ij} = \left(\frac{1}{\epsilon_p} - \frac{\exp\left(-\kappa f_{ij}^{\mathrm{GB}}\right)}{\epsilon_s} \right) \ , \tag{6}$$

and the famous generalized Born equation is given by

$$f_{ij}^{\mathrm{GB}} = \sqrt{r_{ij}^2 + \alpha_i \alpha_j \exp\left(\frac{-r_{ij}^2}{4\alpha_i \alpha_j} \right)} \ . \tag{7}$$

[15]. The atomic Born radius, α, is the measure of electrostatic screening experienced by an atom and is evaluated by

$$\alpha_k = \left[\frac{1}{\rho_{k0}} - \frac{1}{\rho_k} \tanh\left(\delta\psi_k - \beta\psi_k^2 + \gamma\psi_k^3 \right) \right]^{-1} \tag{8}$$

according to [14], and ψ is the sum of overlapping neighboring spheres and is calculated by

$$\psi_k = \rho_{k0} \sum_l H_{kl} \ . \tag{9}$$

The distance dependent overlap of two spheres, H_{ij}, is given by a piecewise functions whose different regimes are given by [16, 14]

$$\text{Regimes} = \begin{cases} 0 & r_{ij} > r_c + \rho_{js}, i \text{ and } j \text{ don't overlap} \\ \text{I} & r_{ij} > r_c - \rho_{js}, i \text{ and } j \text{ just barely overlap} \\ \text{II} & r_{ij} > 4\rho_{js}, \text{arbitrary smoothing distance} \\ \text{III} & r_{ij} > \rho_{i0} + \rho_{js}, i \text{ and } j \text{ heavy overlap} \\ \text{IV} & r_{ij} > \left|\rho_{i0} - \rho_{js}\right|, i \text{ and } j \text{ almost burried} \\ \text{V} & \rho_{i0} < \rho_{js}, i \text{ inside } j \\ \text{VI} & \text{otherwise}, j \text{ inside } i \end{cases} \tag{10}$$

and whose values are

$$H_{ij} = \begin{cases} 0 & 0 \\ \text{I} & \frac{1}{8r_{ij}}\left[1+\frac{2r_{ij}}{r_{ij}-\rho_{js}}+\frac{1}{r_c^2}\left(r_{ij}^2-4r_c r_{ij}-\rho_{js}^2\right)+2\ln\frac{r_{ij}-\rho_{js}}{r_c}\right] \\ \text{II} & \frac{\rho_{js}^2}{r_{ij}^2}\frac{\rho_{js}}{r_{ij}^2}\left[a+\frac{\rho_{js}^2}{r_{ij}^2}\left(b+\frac{\rho_{js}^2}{r_{ij}^2}\left(c+\frac{\rho_{js}^2}{r_{ij}^2}\left(d+\frac{\rho_{js}^2}{r_{ij}^2}e\right)\right)\right)\right] \\ \text{III} & \frac{1}{2}\left[\frac{\rho_{js}}{r_{ij}^2-\rho_{js}^2}+\frac{1}{2r_{ij}}\ln\frac{r_{ij}-\rho_{js}}{r_{ij}+\rho_{js}}\right] \\ \text{IV} & \frac{1}{4}\left[\frac{1}{\rho_{i0}}\left(2-\frac{1}{2r_{ij}\rho_{i0}}\left(r_{ij}^2+\rho_{i0}^2-\rho_{js}^2\right)\right)-\frac{1}{r_{ij}+\rho_{js}}+\frac{1}{r_{ij}}\ln\frac{\rho_{i0}}{r_{ij}+\rho_{js}}\right] \\ \text{V} & \frac{1}{2}\left[\frac{\rho_{js}}{r_{ij}^2-\rho_{js}^2}+\frac{2}{\rho_{i0}}+\frac{1}{2r_{ij}}\ln\frac{\rho_{js}-r_{ij}}{r_{ij}+\rho_{js}}\right] \\ \text{VI} & 0 \end{cases} \tag{11}$$

Ref [16].

2.4.2 Calculating Force

The net GB force on an atom is given by

$$\begin{aligned} \vec{F}_i^{GB} &= -\sum_j\left[\frac{dE_T^{GB}}{dr_{ij}}\right]\hat{r}_{ij}\ , \quad \vec{r}_{ij}=\vec{r}_i-\vec{r}_j \\ &= -\sum_j\left[\frac{\partial E_{ij}^{GB}}{\partial r_{ij}}+\sum_k\frac{\partial E_T^{GB}}{\partial\alpha_k}\frac{d\alpha_k}{dr_{ij}}\right]\hat{r}_{ij} \\ &= -\sum_j\left[\frac{\partial E_{ij}^{GB}}{\partial r_{ij}}+\frac{\partial E_T^{GB}}{\partial\alpha_i}\frac{d\alpha_i}{dr_{ij}}+\frac{\partial E_T^{GB}}{\partial\alpha_j}\frac{d\alpha_j}{dr_{ij}}\right]\hat{r}_{ij} \end{aligned} \tag{12}$$

Below we list the necessary derivatives requisite for calculating these forces.

$$\frac{\partial E_{ij}^{\mathrm{GB}}}{\partial r_{ij}} = -k_e\left[\frac{q_iq_j}{f_{ij}^{\mathrm{GB}}}\frac{\partial D_{ij}}{\partial r_{ij}}-\frac{q_iq_jD_{ij}}{\left(f_{ij}^{\mathrm{GB}}\right)^2}\frac{\partial f_{ij}^{\mathrm{GB}}}{\partial r_{ij}}\right] \tag{13}$$

$$\frac{\partial D_{ij}}{\partial r_{ij}} = \frac{\kappa}{\epsilon_s}\exp\left(-\kappa f_{ij}^{\mathrm{GB}}\right)\frac{\partial f_{ij}^{\mathrm{GB}}}{\partial r_{ij}} \tag{14}$$

$$\frac{\partial f_{ij}^{\mathrm{GB}}}{\partial r_{ij}} = \frac{r_{ij}}{f_{ij}^{\mathrm{GB}}}\left[1-\frac{1}{4}\exp\left(\frac{-r_{ij}^2}{4\alpha_i\alpha_j}\right)\right] \tag{15}$$

$$\frac{\partial E_T^{GB}}{\partial\alpha_k} = \sum_i\sum_{j>i}\left[\frac{\partial E_{ik}^{GB}}{\partial\alpha_k}+\frac{\partial E_{kj}^{GB}}{\partial\alpha_k}\right]+\sum_i\frac{\partial E_{ii}^{GB}}{\partial\alpha_k} \tag{16}$$

$$\frac{\partial E_{ij}}{\partial\alpha_i} = -\frac{1}{\alpha_i}\frac{k_eq_iq_j}{2f_{ij}^2}\left(\frac{\kappa}{\epsilon_s}\exp\left(-\kappa f_{ij}\right)-\frac{D_{ij}}{f_{ij}}\right)\left(\alpha_i\alpha_j+\frac{r_{ij}^2}{4}\right)\exp\left(\frac{-r_{ij}^2}{4\alpha_i\alpha_j}\right) \tag{17}$$

$$\frac{\partial E_{ij}}{\partial \alpha_j} = -\frac{1}{\alpha_j}\frac{k_e q_i q_j}{2f_{ij}^2}\left(\frac{\kappa}{\epsilon_s}\exp\left(-\kappa f_{ij}\right) - \frac{D_{ij}}{f_{ij}}\right)\left(\alpha_i\alpha_j + \frac{r_{ij}^2}{4}\right)\exp\left(\frac{-r_{ij}^2}{4\alpha_i\alpha_j}\right) \tag{18}$$

$$\frac{d\alpha_i}{dr_{ij}} = \frac{\alpha_i^2\rho_{i0}}{\rho_i}\left(1-\tanh^2\left(\delta\psi_i - \beta\psi_i^2 + \gamma\psi_i^3\right)\right)\left(\delta - 2\beta\psi_i + 3\gamma\psi_i^2\right)\frac{\partial H_{ij}}{\partial r_{ij}} \tag{19}$$

$$\frac{d\alpha_j}{dr_{ij}} = \frac{\alpha_j^2\rho_{j0}}{\rho_j}\left(1-\tanh^2\left(\delta\psi_j - \beta\psi_j^2 + \gamma\psi_j^3\right)\right)\left(\delta - 2\beta\psi_j + 3\gamma\psi_j^2\right)\frac{\partial H_{ji}}{\partial r_{ij}} \tag{20}$$

$$\frac{\partial H_{ij}}{\partial r_{ij}} = \begin{cases} 0 & 0 \\ \text{I} & \left[-\frac{(r_c+\rho_{js}-r_{ij})(r_c-\rho_{js}+r_{ij})\left(\rho_{js}^2+r_{ij}^2\right)}{8r_c^2r_{ij}^2\left(\rho_{js}-r_{ij}\right)^2} - \frac{1}{4r_{ij}^2}\ln\frac{r_{ij}-\rho_{js}}{r_c}\right] \\ \text{II} & \left[-4a\frac{\rho_{js}^3}{r_{ij}^5} - 6b\frac{\rho_{js}^5}{r_{ij}^7} - 8c\frac{\rho_{js}^7}{r_{ij}^9} - 10d\frac{\rho_{js}^9}{r_{ij}^{11}} - 12e\frac{\rho_{js}^{11}}{r_{ij}^{13}}\right] \\ \text{III} & \frac{1}{2}\left[-\frac{\rho_{js}\left(r_{ij}^2+\rho_{js}^2\right)}{r_{ij}\left(r_{ij}^2-\rho_{js}^2\right)^2} - \frac{1}{2r_{ij}^2}\ln\frac{r_{ij}-\rho_{js}}{r_{ij}+\rho_{js}}\right] \\ \text{IV} & \frac{1}{4}\left[-\frac{1}{2\rho_{i0}^2} + \frac{r_{ij}^2\left(\rho_{i0}^2-\rho_{js}^2\right)-2r_{ij}\rho_{js}^3+\rho_{js}^2\left(\rho_{i0}^2-\rho_{js}^2\right)}{2r_{ij}^2\rho_{i0}^2\left(r_{ij}+\rho_{js}\right)^2} - \frac{1}{r_{ij}^2}\ln\frac{\rho_{i0}}{r_{ij}+\rho_{js}}\right] \\ \text{V} & \frac{1}{2}\left[-\frac{\rho_{js}\left(r_{ij}^2+\rho_{js}^2\right)}{r_{ij}\left(r_{ij}^2-\rho_{js}^2\right)^2} - \frac{1}{2r_{ij}^2}\ln\frac{\rho_{js}-r_{ij}}{r_{ij}+\rho_{js}}\right] \\ \text{VI} & 0 \end{cases} \tag{21}$$

2.5 Vacuum

Molecular Dynamics simulation which use neither explicit nor implicit solvent are considered in vacuum. Vacuum simulation are very inexpensive but also very unrealistic. One of the biggest problems is that the electrostatics are not screened at all so electrostatic forces are too high. For this reason proteins in vacuum tend to to become very rigid like a clenched fist and do not behave as they would in solvent. We include vacuum simulations here only for comparison.

3 Methods

3.1 Explicit Solvent

3.1.1 SPC/E

NPT ensemble simulations for all-atom Alanine protein in SPC/E water were performed by GROMACS 4.5.1 [13] using CHARMM force field. Protein was solvated in 2575 water molecules inside a cubic box with periodic boundary conditions. The temperature of system is maintained at $300K$ by Nose-Hoover thermostat with time constant of $0.1ps$. Pressure was kept at $1bar$ using Berendsen barostat. Electrostatics interactions were calculated using particle mesh Ewald (PME) to account for long range effects. Initial configuration for production run was found by performing energy minimization. Time step of $2fs$ was used

for total of $10ns$ run and trajectory information (coordinates, velocities and forces) was stored at every $2ps$ for post-processing.

3.2 CG Solvent

We used protein in SPC/E water, described in section 3.1.1, as our reference system for coarse-graining solvent degrees of freedom. Each water molecule is represented by a single coarse-grained bead and Alanine protein is kept in its full atomistic detail as in reference system. For this mapping scheme, we defined interactions of protein and water CG beads by implementing force field between center of mass of each monomer and CG water bead. Therefore, forces on protein due to solvent molecules only depend upon distances between COM of monomers and CG water beads.

Here, we represent COM of water, (NH3-CHCH3-CO)-, -(NH-CHCH3-CO)-, and -(NH-CHCH3-COO) as SOL, VE1, VA, and VE2 respectively, as shown in Fig. 5. The force field between sites SOL-SOL, SOL-VE1, SOL-VA and SOL-VE2 are obtained by force-matching, with exclusion approach, using VOTCA tool.

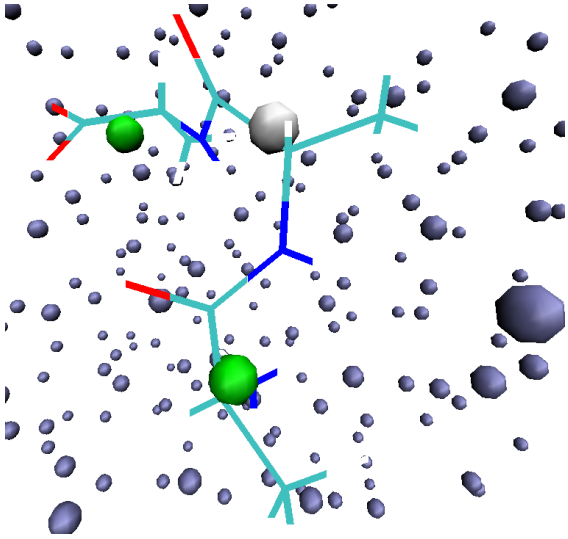

Figure 5: Protein in Coarse-Grained Solvent.

For SOL-SOL coarse graining, reference forces are obtained by recalculating non-bonded interactions between only solvent molecules for the reference trajectory. Whereas, for SOL-VE1, SOL-VA, and SOL-VE2, reference forces are obtained by excluding protein-protein bonded, protein-protein non-bonded, and solvent-solvent interactions, i.e. only non-bonded interactions between protein and solvent are recalculated from reference trajectory.

With PME option in GROMACS, there is no straight-forward way to exclude non-bonded electrostatic interactions between excluded groups. Hence, to obtain all-atom reference forces, we used trajectory sampled by PME option, but recalculated forces for this trajectory using Cut-Off of $1.2nm$ for electrostatics. This procedure may cause inconsistency between reference forces and reference trajectory, which depends upon range of electrostatics interactions. To study the effects of this inconsistency on computation of CG force field, we performed a test simulation of SPC/E water-protein system with Cut-Off of $1.5nm$ for electrostatics and used it for computing second set of CG force field. To evaluate these force fields, RDFs for SOL-SOL, VE1-SOL, VA-SOL, and VE2-SOL, predicted by all-atom simulations and CG simulations, are computed.

The CG force field obtained is as shown in Fig. 6. It can be observed that CG interactions between SOL-SOL, and SOL-VA do not differ much for Cut-Off and PME reference system. However, there is significant difference between CG potentials for SOL-VE1, and SOL-VE2 calculated from Cut-off and PME reference system. These differences can be attributed to inconsistency between reference trajectory of PME and recalculated forces obtained by Cut-Off option.

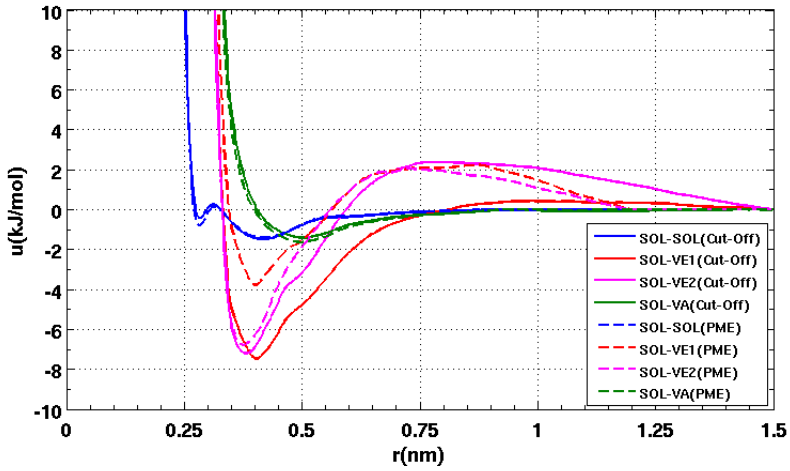

Figure 6: Coarse-Grained Solvent Interaction Potentials.

CG solvent system is then simulated using GROMACS 4.5.1 and CG force-field is specified using tabulated potentials. It was observed that when using NPT ensemble system box started expanding drastically. This is because, CG force-field could not reproduce the pressure of reference system. Hence, to keep the density of solvent same as reference system, CG solvent system is simulated using NVT ensemble with same 2575 solvent molecules and box size (equilibrated) of reference system. Except the removal of barostat, all other simulation parameters are same as reference system. Techniques for CG force-field pressure-correction exist [8], but we have not implemented those here due

to time constraint.

3.3 TIP3P,GBIS,Vacuum

NAMD simulation tool was used for protein in vacuum, TIP3P and GBIS solvent models. TIP3P was simulated in NPT ensemble, while vacuum and GBIS were simulated in NVT. Temperatuer was kept at 300 K by a Langevin thermostat. In NPT, pressure was kept at 1 atm by a Langevin-piston barostat. In TIP3P system, long-range electrostatics were caculated using PME with PBC.

3.4 Properties

For comparing performance of the different solvent models we computed root mean square deviation (RMSD) and solvent accessible surface areas (SASA) of protein for each system. Default RMSD computation routines available in VMD and GROMACS are used to compute RMSD. Whereas, to compute SASA we have implemented the Linear Combinations of Pairwise Overlaps (LCPO) algorithm described below.

3.4.1 SASA :: Linear Combinations of Pairwise Overlaps

One measure of protein behavior is the solvent accessible surface area (SASA) of a protein. The SASA is governed by the interactions (or lack of) of hydrophobic and hydrophilic amino acids with water. Also, solvent imposes a surface tension near the protein-solvent interface which affects protein structure and dynamics. For this reason we expect a good solvent model to faithfully reproduce the SASA and we use it as metric of comparison between the different solvent models.

The Linear Combinations of Pairwise Overlaps (LCPO) is a method by [17] for approximating the SASA. This algorithm has been implemented by David Tanner in a Tcl (Tool Command Language) script which can be executed in VMD. LCPO calculates the SASA of each atom by estimating the overlap between the atom and neighboring atoms; the more a protein atom is overlapped by other protein atoms, the less the atom is exposed to solvent. LCPO defines the SASA of an atom with four terms:

$$A_i = P_1 S_i + P_2 \sum_{j \in N(i)} A_{ij} + P_3 \sum_{\substack{j,k \in N(i) \\ k \in N(j) \\ k \neq j}} A_{jk} + P_4 \sum_{j \in N(i)} A_{ij} \sum_{\substack{k \in N(i) \\ k \in N(j) \\ k \neq j}} A_{jk} \tag{22}$$

where the overlap between spheres i and j is

$$A_{ij} = \pi R_i \left[2R_i - r_{ij} - \frac{1}{r_{ij}} \left(R_i^2 - R_j^2 \right) \right] \tag{23}$$

The parameters P1, P2, P3 and P4 are already parameterized for different atom types. The first term involves the surface area of the atom before overlap,

$$S_i = 4\pi R_i^2 \tag{24}$$

where R is the atom radius (vdW radius plus probe radius of 1.4 Å).

The second term calculates the total overlaps of all neighboring ($j \in N(i)$ means any atom j for which $r_{ij} < R_i + R_j$) atoms with atom i. This term will oversubtracted surface area in as much as two neighboring atoms both overlap the same portion of atom i. The third term corrects this.

The third term is the sum of overlaps of i's neighbors with each other. The more i's neighbors overlap each other, the more they over subtracted surface area in the second term.

The fourth term is a further correction for multiple overlaps. Each overlap of j with i is weighted by how much j is overlapped with all mutual neighbors k.

This very fast approximation is generally within 2% agreement with numerical surface calculators but can be as high as 10%.

4 Results

Equilibrium statistics of 5 ns (Table 2) for various simulations allow to compare contrasting effects of solvent models on the test protein.

Figure 7 shows that both explicit solvents maintain a RMSD of ≈0.2 nm, this represents an accurate preservation of protein structure. GBIS has a worse RMSD than explicit solvents (0.227 vs 0.44 nm). CG(Cut-Off) solvent model predicts RMSD close to that of reference SPC/E(Cut-Off) system (0.47 vs 0.35 nm). However, RMSD predicted by CG(PME) is way-off compared to SPC/E(PME) (0.638 vs 0.245 nm). This can be contributed to inconsistencies of reference forces, used for coarse-graining, recalcualted using cut-off from reference trajectory, which was obtained by PME. Although GBIS and CG(Cut-Off) predict RMSD different than explicit solvent models, it is not as bad as vacuum (0.587 nm), signifying that GBIS and CG(Cut-Off) models represent water better than vacuum.

In Fig. 8 trace plot of SASA predicted by different solvent models is shown. It can be observed that simulations where solvent effects were considered, explicitly or implicitly, predict SASA approximately the same (30.0±2.0 nm^2). Whereas, vacuum has a much smaller surface area (27.90 nm^2) signifying that vacuum does cause the protein to clench tight, like a fist making the protein smaller than it should be. From this it can be said that solvent plays an important role in determining protein structure.

Performance of CG solvent models can also be judged by comparing local structure, such as solvent RDF about protein monomers. Fig. 9 shows RDFs for SOL-SOL, VA-SOL, VE1-SOL, and VE2-SOL, predicted by CG models and reference all-atom models. It can be observed that CG models do well in case of SOL-SOL and VA-SOL local structure. But, local structures for VE1-SOL, and VE2-SOL are significantly different than that of reference system. One could attribute these errors to inability of Force-Matching technique to capture average electrostatics forces between VE1-SOL and VE2-SOL, since VE1 and VE2 sits have net charges of +1.0 e and -1.0 e respectively. Whereas, VA and

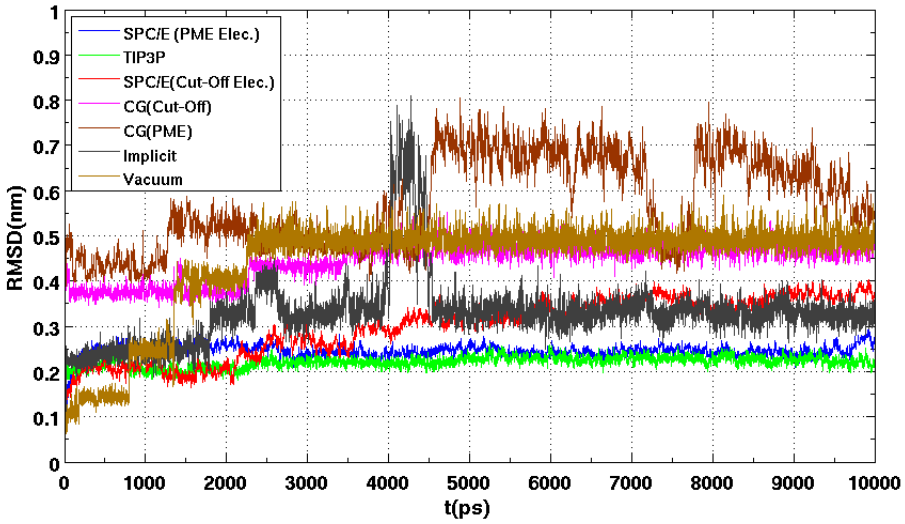

Figure 7: Root Mean Squared Deviations (RMSD) for Different Solvent Models

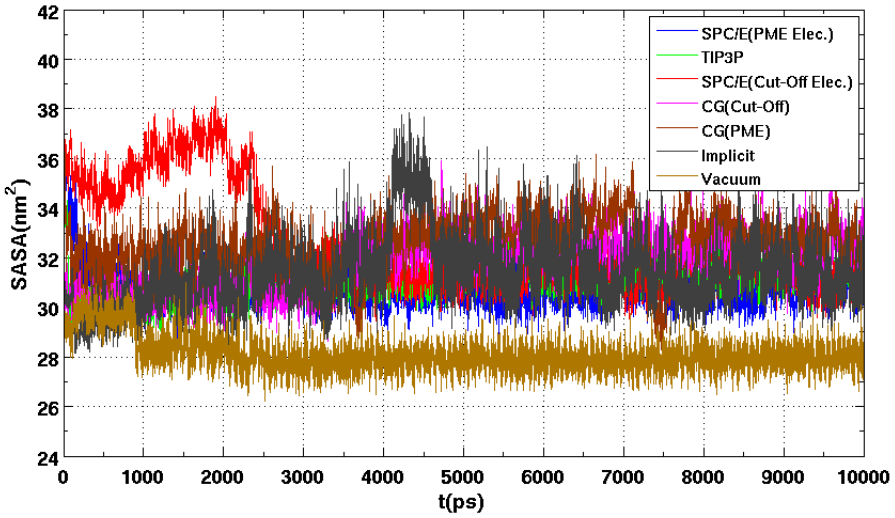

Figure 8: Solvent Accessible Surface Areas (SASA) for Different Solvent Models

SOL sites are neutral. Further investigations are needed to understand these results. Also, other coarse-graining strategies can be tried to improve the results

Table 2: Simulation analysis. For the last 5 ns of each simulation, the mean RMSD and SASA with their respective errors in the mean, variance and auto-correlation time.

Solvent	RMSD (nm)				SASA (nm^2)			
	mean	error	σ	τ	mean	error	σ	τ
SPC/E(PME)	0.245	0.0015	0.011	47	30.64	0.053	0.602	19.5
TIP3P	0.227	0.001	0.001	64	31.29	0.019	0.434	9.4
SPC/E(Cut-Off)	0.35	0.0078	0.022	31.8	31.43	0.12	0.68	72.6
CG(Cut-Off)	0.47	0.0007	0.018	4.23	32.53	0.051	0.816	9.93
CG(PME)	0.638	0.017	0.074	133.1	33.03	0.20	1.06	85.6
GBIS	0.44	0.002	0.02	62	31.47	0.14	1.09	80
Vacuum	0.587	0.0004	0.02	14	27.90	0.01	0.489	2.1

of CG solvent models.

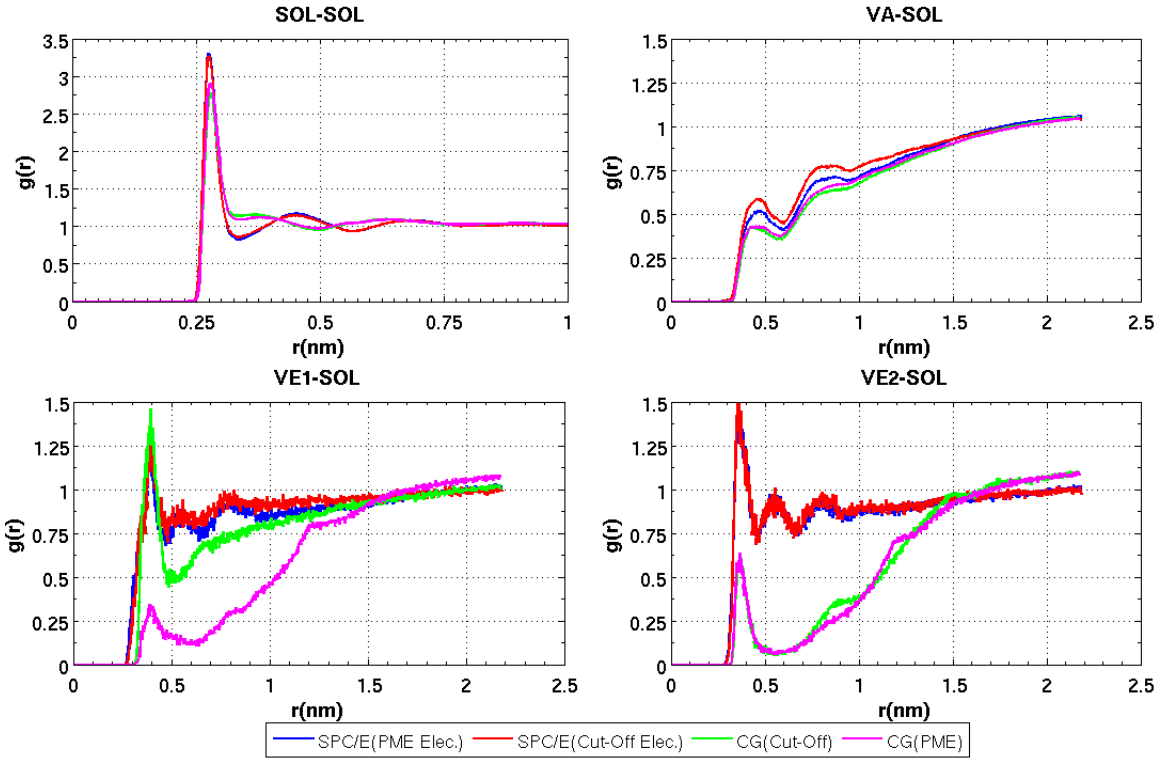

Figure 9: Radial Distributions of solvent about protein.

5 Conclusions

In this project work we simulated simple test protein with different solvent models, taking into consideration discrete nature of water molecules explicitly (SPC/E, TIP3P) and implicitly (GBIS, CG). We compared Root Mean Square Deviations (RMSD) and Solvent Accessible Surface Area (SASA) predicted by these different models. All-atom SPC/E and TIP3P solvent models represent an accurate preservation of protein structure. It was observed that GBIS and CG models predict RMSD different than those of explicit solvent models, but they do represent effects of solvent, which are missing in just vacuum simulations. Effects of solvent on protein structure are further demonstrated by SASA. Vacuum causes protein to occupy smaller surface compared to protein in solvent. Both implicit models are able to predict SASA approximately same as explicit models.

It was observed that long-range electrostatics effects are missing in CG force fields computed using FM *with exlusions* method. And there is scope for further improvement in CG models. It would be a good exercise to do coarse-graining using different techniques, such as Iterative Boltzmann Inversion, Inverse Monte Carlo, etc. Also, one can further develope these solvent models for real protein system and check the robustness of these techniques.

References

[1] Karplus M, McCammon J. A. Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, 9(9):646-52, 2002.

[2] Scheraga H A, Khalili M, Liwo A. Protein-folding dynamics: overview of molecular simulation techniques. *Bioinformatics*, 58:57-83, 2007.

[3] Sharma S., Ding F., Nie H., Watson D., Unnithan A., Lopp J., Pozefsky D., Dokholyan N. V. iFold: a platform for interactive folding simulations of proteins. *Annual Review of Physical Chemistry*, 22(21):2693-4, 2006.

[4] Rhee Y. M., Sorin E. J., Jayachandran G., Lindahl E., Pande V. S. Simulations of the role of water in the protein-folding mechanism. *Proceedings of the National Academy of Sciences of the United States of America*, 101(17):6456-61, 2004.

[5] Lucent D., Vishal V., Pande V. S. Protein folding under confinement: a role for solvent. *Proceedings of the National Academy of Sciences of the United States of America*, 104(25):10430-4, 2007.

[6] Guillot B. A Reappraisal of what we have learnt during three decades of computer simulations on water. *Journal of Molecular Liquids*, 101(1-3):219-260, 2002.

[7] Jorgensen W. L., Tirado-Rives J. Potential energy functions for atomic-level simulations of water and organic and biomolecular systems. *Proceedings of the National Academy of Sciences of the United States of America*, 102(19):6665-70, 2005.

[8] Wang H., Junghans C., Kremer K. Comparative atomistic and coarse-grained study of water: What do we lose by coarse-graining? *The European Physical Journal. E, Soft Matter*, 28(2):221-9, 2009.

[9] Izvekov S., Voth G A. Multiscale coarse graining of liquid-state systems. *The Journal of Chemical Physics*, 123:134105, 2005.

[10] Villa A., van der Vegt N. F. A., Peter C. Self-assembling dipeptides: including solvent degrees of freedom in a coarse-grained model. *Physical Chemistry Chemical Physics*, 11:2068-2076, 2009.

[11] Ruhle V., Junghans C., Lukyanov A., Kremer K., Andrienko D. Versatile Object-Oriented Toolkit for Coarse-Graining Applications. *Journal of Chemical Theory and Computation*, 5(12):3211-3223, 2009.

[12] Ruhle V., Junghans C. Hybrid approaches to coarse-graining using the VOTCA package: liquid hexane. *Macromolecular Theory and Simulations*, 20, 2011.

[13] D. van der Spoel, E. Lindahl, B. Hess, A. R. van Buuren, E. Apol, P. J. Meulenhoff, D. P. Tieleman, A. L. T. M. Sijbers, K. A. Feenstra, R. van Drunen and H. J. C. Berendsen. Gromacs User Manual version 4.5.4. *www.gromacs.org*, 2010.

[14] Alexey Onufriev, Donald Bashford, and David A. Case. Exploring protein native states and large-scale conformational changes with a modified generalized born model. *PROTEINS: Structure, Function, and Genetics*, 55:383–394, 2004.

[15] W. Clark Still, Anna Tempczyk, Ronald C. Hawley, and Thomas Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *Journal of the American Chemical Society*, 112:6127–6129, 1990.

[16] Gregory D. Hawkins, Christopher J. Cramer, and Donald G. Truhlar. Parameterized models of aqueous free energies of solvation based on pairwise descreening of solute atomic charges from a dielectric medium. *Journal of Physical Chemistry*, 100:19824–19839, 1996.

[17] Jorg Weiser, Peter S. Shenkin, and W. Clark Still. Approximate atomic surfaces from linear combinations of pairwise overlaps (lcpo). *Journal of Computational Chemistry*, 20:217–230, 1998.