Accelerating the Generalized Born with Molecular Volume and Solvent Accessible Surface Area Implicit Solvent Model Using Graphics Processing Units

Xiping Gong,[a] Mara Chiricotto,[a] Xiaorong Liu,[a] Erik Nordquist,[a] Michael Feig,[b] Charles L. Brooks III,[c] and Jianhan Chen *[a,d]

The generalized Born with molecular volume and solvent accessible surface area (GBMV2/SA) implicit solvent model provides an accurate description of molecular volume and has the potential to accurately describe the conformational equilibria of structured and disordered proteins. However, its broader application has been limited by the computational cost and poor scaling in parallel computing. Here, we report an efficient implementation of both the electrostatic and nonpolar components of GBMV2/SA on graphics processing unit (GPU) within the CHARMM/OpenMM module. The GPU-GBMV2/SA is numerically equivalent to the original CPU-GBMV2/SA. The GPU acceleration offers ~60- to 70-fold speedup on a single NVIDIA TITAN X (Pascal) graphics card for molecular dynamic simulations of both folded and unstructured proteins of various sizes. The current implementation can be further optimized to achieve even greater acceleration with minimal reduction on the numerical accuracy. The successful development of GPU-GBMV2/SA greatly facilitates its application to biomolecular simulations and paves the way for further development of the implicit solvent methodology. © 2019 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.26133

Introduction

It is crucial to provide an accurate description of the solvent environment during biomolecular simulations, where the solvent plays a vital role in governing the conformational fluctuations and transitions.[1–3] Conventionally, explicit solvent models provide a relatively detailed and accurate description on interactions between the solvent molecules and solutes and are regarded as standard approaches to explore the influence of solvent on the solute molecule.[4] However, it dramatically increases the computational cost of a simulation, and the solvent friction further adds to the difficulty of sampling the solute conformations. Implicit solvent is a viable alternative that captures the effective influence of solvent on the solute by direct estimation of the solvation free energy as a function of the solute coordinates.[5] Implicit treatment of solvent substantially reduces the system size, thus allowing significant reduction of computational cost and faster sampling of solute conformations.[6–10]

There are many approaches for estimating the solvation free energy in implicit solvent treatment, including the Poisson–Boltzmann (PB) and generalized Born (GB) models. Both PB and GB are based on continuum electrostatics treatment of solvent environment.[11–15] Compared with the PB model, the GB approximation allows the analytical evaluation of molecular forces and is more suitable for fast molecular dynamics (MD) simulations. The most important task in GB models is to evaluate the effective Born radius of each atom, which is dependent on all solute coordinates. GB models can be numerically equivalent to the underlying PB calculations, given accurate effective Born radii.[5,14] Numerous approaches have been developed for efficient calculations of effective Born radii[16–26,28,29], including the fast analytical continuum treatment of solvation,[16] the GB Surface Area (GBSA) from Onufriev, Bashford, and Case (OBC),[17] analytical generalized Born plus nonpolar 2,[18] and numerical integration-based ones such as the GB with simple smoothing function (GBSW)[19–21] and GB with molecular volume (GBMV2)[23, 24] models. The GBMV2 model, in particular, contains an analytical approximation of the Lee–Richards molecular volume and avoids unphysical solvent-inaccessible high dielectric protein interior regions.[23,24,28,29] It can reproduce the first solvent peak in the potentials of mean force of interactions between polar chemical groups.[21] A comparison of several implicit solvent models has also suggested that the GBMV2 model provides the best agreement with the experimental data, such as hydration free energies of small molecules.[30,31] Recently, it was demonstrated that an optimized GBMV2 model

[a] X. Gong, M. Chiricotto, X. Liu, E. Nordquist, J. Chen
Department of Chemistry, University of Massachusetts, Amherst, Amherst, Massachusetts, 01003
E-mail: jianhanc@umass.edu

[b] M. Feig
Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, Michigan, 48824

[c] C. L. Brooks
Department of Chemistry, University of Michigan, Ann Arbor, Michigan, 48109

[d] J. Chen
Department of Biochemistry and Molecular Biology, University of Massachusetts, Amherst, Amherst, Massachusetts, 01003
E-mail: jianhanc@umass.edu

Contract Grant sponsor: Division of Molecular and Cellular Biosciences; Contract Grant number: MCB 1817332; Contract Grant sponsor: National Institute of General Medical Sciences; Contract Grant numbers: R35 GM126948, R35 GM130587

could provide a reliable description of both folded and unfolded protein conformations.[28] In particular, it shows minimal over-compaction bias in simulation of disordered proteins frequently associated with many implicit and explicit solvent protein force fields.[28,32–35] A key limitation to broader application of GBMV2, however, is that it is ~16 times slower than vacuum calculations and scales poorly to parallel multicore executions.[29]

One powerful technique to improve the efficiency is the use of graphics processing units (GPUs) that can have thousands of parallel processing cores. GPU-accelerated algorithms available in many MD engines, such as CHARMM,[36] AMBER,[37,38] GROMACS,[39] NAMD,[40,41] and OpenMM,[42] have offered up to two orders of magnitude speedup over traditional CPU-based codes. Some efforts have also been made on the GPU acceleration of GB implicit solvent models. The GB/OBC model in Amber has been implemented and achieved routine microsecond MD simulations.[43] The GBSW model has also been implemented in a CHARMM/OpenMM module that displays around 100-fold improvement on the efficiency while maintaining similar numerical accuracy.[44] Notably, these early implementations only include the electrostatic solvation energy and thus might not be directly deployed for biomolecular simulations without the contribution of nonpolar solvation energy. Recently, an efficient pair-wise approximation of the solvent accessible SA (SASA) has been added into the GBSA/OBC GPU model, albeit with limited accuracy.[45] The correlation between atomic SASAs calculated by the GPU model and exact numerical results varies significantly from 0.54 to 0.91 for a number of test proteins.

Here, we report the implementation of an efficient GPU-accelerated GBMV2/SA algorithm in a CHARMM/OpenMM module. The implementation takes advantage of the similarities between GBMV2 and GBSW algorithms and builds on several existing kernels of the GPU-GBSW module. The numerical scheme for computing Born radii also allows for implementation of an efficient algorithm for calculating atomic SAs. Together, the current implementation provides a complete realization of the GBMV2/SA model on GPUs, making it appropriate for general MD simulations of biomolecules. In the below sections of this article, the detailed methodologies of GPU-GBMV2/SA algorithm are discussed in "Methods" section, including the treatment of electrostatic and nonpolar solvation contributions, the lookup table algorithm for efficient volume integration, and the scheme of GPU implementations. Key points of the original GBMV2 model are highlighted. In the "Results and Discussion" section, the accuracy and efficiency of GPU-GBMV2/SA are benchmarked against the CPU-GBMV2/SA implementation, and the remaining computational bottlenecks are also discussed. Finally, the conclusions and an outlook toward future work are given in "Conclusions" section.

Method

In GB models, the total solvation free energy is generally divided into electrostatic and nonpolar contributions,

$$\Delta G_{\mathrm{solv}} = \Delta G_{\mathrm{elec}} + \Delta G_{\mathrm{np}}, \tag{1}$$

where the nonpolar component involves the free energy cost of creating the solute cavity in the solvent and turning on the nonpolar solute–solvent interaction, and the electrostatic component corresponds to the free energy cost of the subsequent step of charging up the solute.[9] The nonpolar contribution is often estimated directly from SASA, even though it has been shown that this approximation limited its ability to capture conformational dependence of the nonpolar free energy.[9,46,47]

Electrostatic solvation free energy and forces

The GB approximation developed by Still and coworkers[48] allows the electrostatic energy to be written as a pairwise summation,

$$\begin{gathered}\Delta G^{\mathrm{elec}} = -\frac{1}{2}\sum_{i,j}\tau_{ij}\frac{q_i q_j}{f_{ij}^{\mathrm{GB}}},\\ f_{ij}^{\mathrm{GB}} = \sqrt{\mathbf{R}_{ij}^2 + R_i^{\mathrm{GB}} R_j^{\mathrm{GB}} \exp\left(-\mathbf{R}_{ij}^2 / K_s R_i^{\mathrm{GB}} R_j^{\mathrm{GB}}\right)}.\end{gathered} \tag{2}$$

where q_i and R_i^{GB} are the atomic charge and effective Born radius of the atom i, respectively, and K_s is an empirical constant that is set to 8 in the GBMV2 model.[23,24] $\tau_{ij} = \left(1/\varepsilon_{\mathrm{solute}} - \exp\left(-\kappa f_{ij}^{\mathrm{GB}}\right)/\varepsilon_{\mathrm{solvent}}\right)$, which provides an effective description of the salt screening effects with κ being a Debye–Hückel screening parameter.[49] The effective Born radius is defined as the radius of an equivalent spherical cavity that yields the same atomic self-polarization free energy. It is thus a function of the positions of all solute atoms. The pairwise GB expression allows analytical evaluation of atomic forces and is thus particularly suitable for MD simulations.

The GB forces with respect to the atomic positions include two terms,

$$F_a^{\mathrm{elec}} = -\frac{\partial \Delta G^{\mathrm{elec}}}{\partial \mathbf{R}_a} = -\left(\sum_{ij}\frac{\partial \Delta G^{\mathrm{elec}}}{\partial \mathbf{R}_{ij}}\frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{R}_a} + \sum_i \frac{\partial \Delta G^{\mathrm{elec}}}{\partial R_i^{\mathrm{GB}}}\frac{\partial R_i^{\mathrm{GB}}}{\partial \mathbf{R}_a}\right), \tag{3}$$

where

$$\begin{aligned}\sum_{ij}\frac{\partial \Delta G^{\mathrm{elec}}}{\partial \mathbf{R}_{ij}}\frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{R}_a} = -\sum_i\left(\tau_{ia}\text{-}\frac{\kappa \exp\left(\text{-}\kappa f_{ia}^{\mathrm{GB}}\right) f_{ia}^{\mathrm{GB}}}{\varepsilon_{\mathrm{solvent}}}\right)\\ \frac{q_i q_a\left[1-\exp\left(-R_{ia}^2/K_s R_{ia}^{\mathrm{GB}} R_{ia}^{\mathrm{GB}}\right)/K_s\right]}{\left(f_{ia}^{\mathrm{GB}}\right)^3}(\mathbf{R}_i - \mathbf{R}_a),\end{aligned} \tag{4}$$

and

$$\begin{aligned}&\sum_i \frac{\partial \Delta G^{\mathrm{elec}}}{\partial R_i^{\mathrm{GB}}}\frac{\partial R_i^{\mathrm{GB}}}{\partial \mathbf{R}_a}\\ &= \frac{1}{2}\sum_i\left[\sum_j\left(\tau_{ij}\text{-}\frac{\kappa \exp\left(\text{-}\kappa f_{ij}^{\mathrm{GB}}\right) f_{ij}^{\mathrm{GB}}}{\varepsilon_{\mathrm{solvent}}}\right)\right.\\ &\left.\frac{q_i q_j \exp\left(-\mathbf{R}_{ij}^2/K_s R_i^{\mathrm{GB}} R_j^{\mathrm{GB}}\right)}{\left(f_{ij}^{\mathrm{GB}}\right)^3}\left(R_j^{\mathrm{GB}} + \frac{\mathbf{R}_{ij}^2}{K_s R_i^{\mathrm{GB}}}\right)\right]\frac{\partial R_i^{\mathrm{GB}}}{\partial \mathbf{R}_a}.\end{aligned} \tag{5}$$

It can be seen from eqs. (2)–(5) that the GB energy and forces depend on the effective Born radii, R_i^{GB}, and their derivatives with respect to atomic positions, $\partial R_i^{\mathrm{GB}}/\partial \mathbf{R}_a$.

Born radii and their derivatives

Computing the effective Born radius of each solute atom is the key step for calculating the GB electrostatic solvation free energy. In GBMV2 model, the calculation of a given Born radius considers the contributions from the Coulomb field approximation and an empirical high-order correction term:

$$R_i^{\mathrm{GB}} = \frac{P_1}{G_i} + P_2, \tag{6}$$

and

$$G_i = \left(1 - \frac{1}{\sqrt{2}}\right)\frac{1}{4\pi}\int \frac{V(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_i|^4}d\mathbf{r} + \left(\frac{1}{4\pi}\int \frac{V(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_i|^7}d\mathbf{r}\right)^{1/4}, \tag{7}$$

where P_1 and P_2 are empirical fitting coefficients, $\mathbf{R}_i$ are atomic coordinates, and $V(\mathbf{r})$ is the molecular volume function.[23,24] Optimal values of P_1 and P_2 are obtained by linear regression fitting of atomic GB radii of model proteins to the reference values obtained from high-resolution PB calculations.[19–21,28] Detailed expressions for derivatives of Born radii are given in Appendix S1.

Numerical integration

The important component of computing Born radius is to evaluate the three-dimensional integrals as shown in eq. (7). In the GBMV2 and GBSW models, the integrals are evaluated using numerical quadrature, where they are split up into radial and angular components.[19,23,24] The radial integral is approximated by Riemann–Stieltjes summation with the standard set of radial grid points, while the angular integral is calculated by the Lebedev quadrature.

$$\begin{aligned}\frac{1}{4\pi}\int \frac{V(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_i|^k}d\mathbf{r} &= \frac{1}{4\pi}\int \frac{V(\mathbf{r}+\mathbf{R}_i)}{|\mathbf{r}|^k}d\mathbf{r} \\ &\approx \frac{1}{(3-k)\left(R_i^{\mathrm{eff}}\right)^{k-3}} + \sum_{m\in \mathrm{rad}}\ \sum_{n \in \mathrm{ang}} w_{mn}^k V(\mathbf{r}_{mn}+\mathbf{R}_i),\end{aligned} \tag{8}$$

where w_{mn}^k is the weight of each grid point r_{mn} and R_i^{eff} is an effective integration starting point less than the van der Waals (vdW) radius of each atom, in order to avoid the singularity of integrals. It is noted that the precise definition of the (solute) molecular volume in eq. (8) is a key quantity in determining the Born radii. The vdW-like volume employed in GBSW is simple and efficient to evaluate, and it provides stable forces.[19] However, it generates small and unphysical solvent-inaccessible high dielectric regions inside the solute, leading to an overestimation of solvation free energy and a systematic overstabilization of nonspecific compact conformations.[20,21] This critical shortcoming is effectively solved by adopting an approximate Lee–Richards molecular volume in GBMV2.

Analytical approximation of the molecular volume

The molecular volume (MV) is defined as the solute volume that is formed by rolling a water probe on the solute.[29] Two methods have been previously implemented in the CPU version of GBMV2.[23,24] One is to use arbitrarily precise numerical grids for a highly accurate calculation of Born radii; but this method is computationally expensive, does not provide an analytical gradient, and thus is not suitable for efficient MD simulations. The other method introduces an efficient analytical approximation to the MV with comparable precision of calculating Born radii, which is also suitable for GPU acceleration. The molecular volume is given by a Fermi–Dirac switching function from a preprocessed "raw" molecular volume, $S(r)$,

$$V(\mathbf{r}_{mn}+\mathbf{R}_i) = \frac{1}{1+\exp[\beta(S(\mathbf{r}_{mn}+\mathbf{R}_i)-\lambda)]}, \tag{9}$$

where β and λ are the parameters that represent the width and midpoint of the switching function, respectively.

The expression of $S(\mathbf{r})$ in the GBMV2 model involved two terms,

$$\begin{aligned}&S(\mathbf{r}) = S_{\mathrm{vdW}}(\mathbf{r}) + S_{\mathrm{MV2}}(\mathbf{r}),\quad S_{\mathrm{vdW}}(\mathbf{r}) = 2\sum_j F_{\mathrm{vdW}}(\mathbf{r}),\\ &S_{\mathrm{MV2}}(\mathbf{r}) = S_0\left[\sum_j F_{\mathrm{MV2}}(|\mathbf{r}-\mathbf{R}_j|)\right]\frac{\sum_j |\mathbf{r}-\mathbf{R}_j|^2 F_{\mathrm{MV2}}^2(|\mathbf{r}-\mathbf{R}_j|)}{\left|\sum_j(\mathbf{r}-\mathbf{R}_j)F_{\mathrm{MV2}}(|\mathbf{r}-\mathbf{R}_j|)\right|^2},\end{aligned} \tag{10}$$

where S_{vdW} $(\mathbf{r})$ is the vdW volume contribution and $S_{\mathrm{MV2}}(\mathbf{r})$ includes a vector-based scaling term to account for the discrepancy between vdW and MV volumes. There are two significant points: One is that the atomic volume function, F_{MV2} $(\mathbf{r})$, has a longer tail compared to the F_{vdW} $(\mathbf{r})$, in order to probe more overlap regions between atoms. The other is that the representation of MV. For the vdW volume, because S_{vdw} $(\mathbf{r})$ is a monotonic function with the number of atoms, the summation can be immediately terminated when its value exceeds a certain cutoff. $S_{\mathrm{MV2}}(\mathbf{r})$, however, contains vector-based scaling approximation (VSA) term that helps to distinguish the "gap" (between atoms) and "open" (otherwise) regions, which is required to consider all atoms in proximity. As such, GBMV2 is considerably more expensive that GBSW, especially for small systems.

Additional details of the GBMV2 algorithms can be found in the equations in Appendix S1 and in the original paper.[24] Importantly, from eq. (10) it can be seen that the next step is to calculate the $S(\mathbf{r})$ at each numerical integration grid point (eq. 8), which can be accelerated by a lookup table algorithm.

SASA nonpolar solvation free energy and forces

The nonpolar energy can be decomposed into a short-range repulsive energy and long-range solute–solvent dispersion energy, and is, in the first-order approximation, proportional to SASA.[9,24] Thus, the nonpolar energy in the GBMV2 model is estimated as

$$\Delta G_{\text{np}} = \sum_i \gamma_i A_i, \tag{11}$$

where the γ_i and A_i are the effective surface tension coefficient and SASA of each atom, respectively. The surface coefficient is often assumed to be same for all atom types, reducing eq. (11) to $\Delta G_{\text{np}} = \gamma \sum_i A_i$. This linear approximation has been shown to provide an adequate description of nonpolar solvation energy for many biomolecular applications.[9,24]

The atomic SASA can be expressed as

$$A_i = \int_{|\mathbf{r}-\mathbf{R}_i| = R_i^{\text{vdW}} + R_{\text{w}}} f\left(\bar{V}_i(\mathbf{r})\right) d\mathbf{r}, \tag{12}$$

where the excluded volume $\bar{V}_i(\mathbf{r}) = \sum_{j \neq i} V_j(\mathbf{r})$ involves volumes for all atoms except for atom i, and the smooth function f represents the exposed rate at $\mathbf{r}$ point, which should be one if the excluded volume is zero, and it should be zero if the sum of excluded volume is one. In the GBMV2/SA model, an analytic expression of the vdW volume is used,

$$\bar{V}_i(\mathbf{r}) = \sum_{j \neq i} 2f\left(u_j\right), \tag{13}$$

the u_j and exposed function f are written as

$$f(u) = \begin{bmatrix} 1 & u \leq 0 \\ 1 - 10u^3 + 15u^4 - 6u^5 & 1 > u > 0 \\ 0 & u \geq 1 \end{bmatrix}, \tag{14}$$

and

$$u_j = \frac{\left|\mathbf{r}-\mathbf{R}_j\right|^2 - \left(R_j^{\text{vdW}} + t_-^{\text{vdW}}\right)^2}{\left(R_j^{\text{vdW}} + t_+^{\text{vdW}}\right)^2 - \left(R_j^{\text{vdW}} + t_-^{\text{vdW}}\right)^2}, \tag{15}$$

where R_j^{vdW} is the vdW radius of j atom and R_{w} is the radius of solvent molecule, for the water molecule, which is 1.4 Å. The switching widths, t_+^{vdW} and t_-^{vdW}, have been optimized to 1.2 and 1.5 Å, respectively, for $R_{\text{w}} = 1.4$ Å.

The general integral of eq. (12) cannot be solved analytically. However, a straightforward numerical expression is given as follows:

$$A_i \approx 4\pi\left(R_i^{\text{vdW}} + R_{\text{w}}\right)^2 \sum_m w_m f\left(\bar{V}_i(\mathbf{r}_m + \mathbf{R}_i)\right), \tag{16}$$

where the excluded molecular volume at each grid point is determined quickly by the lookup table algorithm described below. Detailed derivations of the nonpolar energy and forces term can be found in Appendix S1.

Lookup table algorithm

The numerical volume integrations in GBMV2 (and GBSW) require quick access of all atoms within a certain distance that could contribute to the volume function. This is enabled by constructing a lookup table.[19,23,24] Specifically, the lookup table contains a spatially uniform cubic grid enclosing all solute atoms. At each grid point, all the atoms that are less than a certain distance, R_{max}, are stored in a lookup table array,

$$|\mathbf{r}-\mathbf{R}_i| \leq R_{\text{max}} = \max\left[R_i^{\text{vdW}}\right] + 2.1 + \frac{\sqrt{3}}{2}c + R_{\text{buffer}}, \tag{17}$$

where c is the width of the grid cell, the value 2.1 Å is the length of the tail of the atomic function F_{MV2} ($\mathbf{r}$), and R_{buffer} is an adjustable length that determines how far any atom can move before rebuilding the lookup table. The default value of R_{buffer} is zero, meaning the lookup table will be updated at each simulation step. By using the lookup procedure, the cost of computing the molecular volumes is reduced to linear scaling with the number of the grid points. It is noted that the number of neighbor atoms at each grid point is much larger in GBMV2 than GBSW due to the longer tail of atomic function, which contributes to a twofold to threefold computational cost increase.

Parallelization and CUDA implementation

The existing GBSW kernels were adapted for the implementation of GPU-GBMV2/SA. As a plugin of CHARMM/OpenMM program, the overall design of the GPU-GBSW model is considered as a stand-alone solvent model in the OpenMM library.[44] It contains eight kernels, four of which are used to implement the lookup table, and the other four are used to calculate the electrostatic solvation energies and forces of hydrogen and nonhydrogen atoms. Kernels to support the lookup table were directly modified to support a larger value of R_{max} and the greater table depth required for GBMV2. In GBMV2, hydrogens have nonzero input radii and do not need to be treated

Table 1. Layout of key kernels for GPU GBMV2/SA.

Kernels	Description
calcBornR	To calculate the Born radius of each atom and save the temporary variables for the rapid calculations of the electrostatic forces. Each block is assigned to one atom, and 256 threads are used to loop over all the grid points. The equations can be found in the electrostatic energies part of Appendix S1.
computeGBMVForce	To calculate the GB electrostatic energies and the derivatives with respect to atomic coordinates.
reduceGBMVForce	To calculate the electrostatic forces. Each block is assigned to one atom, and 256 threads are used to loop all the grid points. The equations can be found in the electrostatic force part of Appendix S1.
calcSASA	To calculate the nonpolar energies and forces. Each block is assigned to one atom, and 256 threads are used to loop all the grid points. The equations can be found in eqs. (12)–(16) and nonpolar force part of Appendix S1.

Kernels for lookup table are similar to those used in GPU GBSW[44] and thus not listed.

separately. As such, the GBMV2 electrostatic term only requires three kernels (see Table 1). A new kernel, calcSASA, was developed to calculate atomic SASA and forces. The GPU algorithm for computing SASA terms is similar, where the number of blocks is equal to the number of atoms and threads loop over all quadrature integration grip points. Note that the calcSASA kernel is an independent kernel that can be used for both GPU-GBMV2 and GPU-GBSW models. Pseudocodes illustrating GPU algorithms for computing the electrostatic solvation energies and forces are provided in Appendix S1. Similar algorithms are implemented for the calculations of nonpolar solvation energies and forces, and the pseudocodes thus not provided. A difference between the current implementation and previous GBSW CUDA plugin is that 256 threads are used to loop over all quadrature points per block, which provides optimal computational efficiency in our tests. Another difference is that the Born radii gradients are not saved, because the number of contributing atoms is much larger due to the longer tail of atomic volume function (2.6 Å in GBMV2 vs. 0.3 Å in GBSW). Instead, compact intermediate arrays are saved in the global memory to reduce the computational complexity and cost of electrostatic solvation forces (see the pseudocode in Appendix S1, "CUDA algorithm for computing the electrostatic solvation forces", arrays V, S, X_1, X_2, X_3, and X_4).

Computational details

The correctness and accuracy of GPU-GBMV2/SA were mainly assessed by its ability to reproduce atomic energies and forces of the original CPU-GBMV2/SA implementation in CHARMM as well as PB-derived atomic self-solvation free energies. The model systems include the set of 22 small proteins previously used for the numerical parametrization of the original GBSW and GBMV models.[21,24] The accuracy of GPU-GBMV2/SA was also validated by examining the interaction energy profiles between selected side chains, in comparison with explicit solvent results from previous works.[20,21] The numerical stability of the GPU-GBMV2/SA model was assessed by examining the energy conservation properties under different configurations. Furthermore, a small helical model peptide, $(AAQAA)_3$, was used to examine the stability of GPU-GBMV2/SA in long-time MD simulations and its ability to recapitulate the peptide conformational equilibrium. For this purpose, two distinct initial structures, an ideal helix, and a fully extended conformation were used to initiate independent control and folding simulations, allowing a rigorous diagnosis of convergence. A time step of 2 fs was used. The previously optimized GBMV2/SA protein force field[28,29] was used, and the results were directly compared with those from CPU simulations.

The efficiency of the GPU versus CPU versions of GBMV2/SA was benchmarked using five folded proteins ranging from 856 to 77,304 atoms as well as an intrinsically disordered protein, the N-terminal transactivation domain (TAD) of p53 (926 atoms). The initial structures of folded proteins were downloaded from the Protein Data Bank (PDB) and then energy minimized followed by 5000 steps of NVT equilibration. The initial structure of p53-TAD was taken from a previous study.[32] Default GBMV2/SA parameters were used in all calculations, except for three keywords, beta = −12, P3 = 0.65, P6 = 8, which correspond to β, S_0, and K_s, in eqs. (2) and (9), respectively. The atomic input radii are from the previously optimized GBMV2 force field.[28] The cutoff distance for nonbonded interactions was set at 20 Å, and a time step of 2 fs was used. All GPU simulations were done on an NVIDIA TITAN X (Pascal) graphics card, and CPU calculations were carried out on an Intel Xeon E5-2620 v4 2.10GHz CPU. Performance of key GPU kernels was analyzed using the nvvp and nvprof tools. The result reports are provided in Figure S1, which includes threads per block, registers per thread, theoretical versus achieved occupancy, and so on.

Results and Discussion

Electrostatic solvation energies and forces

Proper GPU implementation of the GBMV2 is first assessed by its ability to reproduce the atomic electrostatic self-solvation energies and forces. As summarized in Figure 1, atomic self-solvation energies and forces of all 22 small proteins are essentially identical between the GPU and original CPU implementations. The numerical differences between CPU and GPU results (see inserts) are extremely small, completely negligible compared to the absolute GB electrostatic energies and

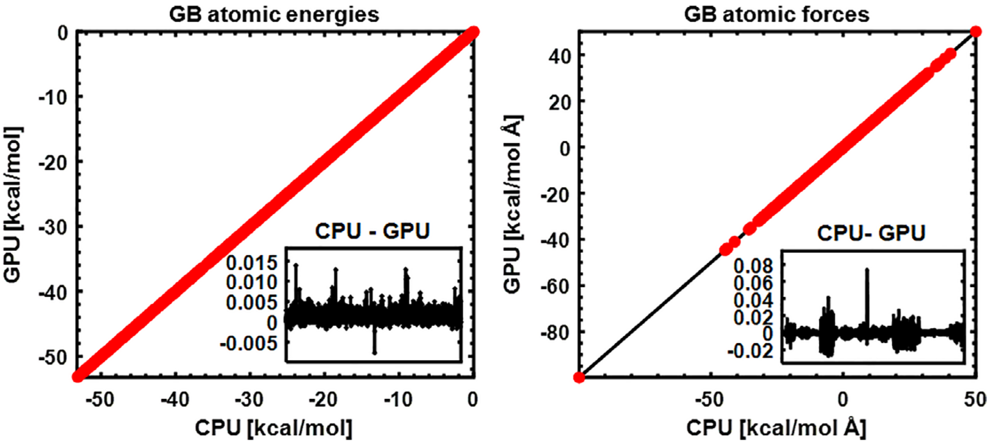

Figure 1. Accuracy of GPU-GBMV2/SA atomic electrostatic self-solvation energies (left) and forces (right), compared with those of CPU-GBMV2. The diagonal line ($y = x$) is shown for reference. All atoms of 22 small proteins are included in this comparison. The inserted panels show the difference between CPU and GPU results (in the same unit, kcal/mol or kcal/mol Å for each of all atoms from the protein test set). [Color figure can be viewed at wileyonlinelibrary.com]

forces. This demonstrates that the electrostatic solvation term of GPU-GBMV2/SA has been implemented correctly in the CUDA platform.

We also validated that atomic self-solvation energies provided by GPU-GBMV2 are consistent with PB-derived results, which is a key indicator of the quality of a GB implicit solvent model. Given the numerical equivalence of GPU- and CPU-GBMV2 models, GPU-GBMV2 should achieve a similar correlation with PB. Indeed, as summarized in Figure 2, the correlation coefficient between effective Born radii derived from PB and GPU-GBMV2 is 0.9985, consistent with the results of CPU-GBMV2.[24] We note that the superb ability of GBMV2 to reproduce PB is attributed to both the higher-order correction to the Coulomb field approximation (eq. 7) and effective approximation of MV (eq. 10).[24]

Nonpolar solvation energy and forces

Nonpolar solvation energy plays important roles in driving the conformations of proteins, although it makes smaller contributions to the total solvation energies compared to the GB term. Figure 3 shows that the nonpolar energies and forces of GPU-GBMV2/SA are also numerically equivalent to those calculated by the original CPU-GBMV2/SA, indicating that both SASA energies and forces have been implemented in the present CUDA platform correctly. As such, it can be expected that the errors of nonpolar energies are on the order of 1–2% comparing with the exact SASA analytic model for proteins.[24] The successful implementation of the SASA term in the CUDA platform provides a complete GPU-GBMV2/SA implicit solvent model that can now be readily deployed for biomolecular simulations. In addition, it also paves the way for the future development of better nonpolar solvation models, such as by including the dispersion contribution.[9]

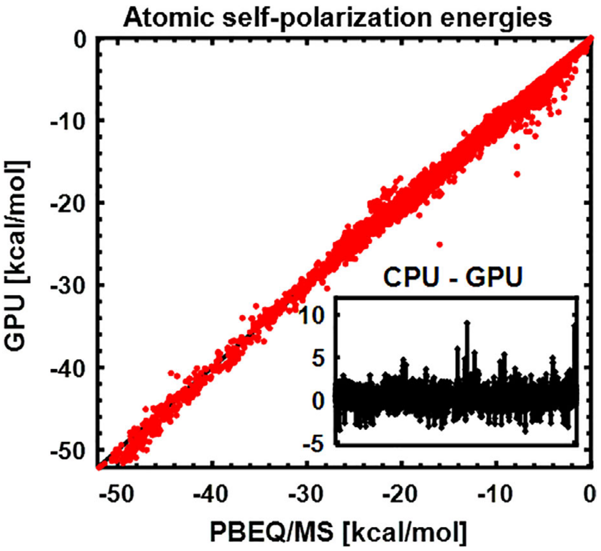

Figure 2. Atomic electrostatic self-solvation energies derived from GPU-GBMV2 versus PB. All atoms from 22 small proteins are included. The insert shows the difference for each atom. [Color figure can be viewed at wileyonlinelibrary.com]

Energy conservation and numerical stability

After establishing the correctness of the GPU implementation, we evaluated the numerical stability of GBMV2/SA by examining the energy conservation properties in microcanonical simulations with three different surface tension parameters (γ). As summarized in Figure 4, the energies from CPU and GPU calculations display similar trends for all three cases, suggesting that the GPU version has similar numerical stability compared to the CPU version. The energy drifts over 300 ps are significant, but in line with a previous analysis of the numerical stability of GBMV2 on CPU.[25] The energy fluctuations in GPU calculations (after removing the linear drift) are slightly higher than those in CPU runs, likely due to the use of mixed single/double precisions. Comparison of the energy conservation properties from simulations with different γ show that SASA as implemented is numerically highly stable. We note that GBMV2 is numerically less stable compared to GBSW because of the sharp molecular surface definition as well as the VSA term. Nonetheless, peptide simulations suggest that GBMV2 can be reliable even with a 2-fs time step with a proper thermostat in canonical simulations, showing no sign of numerical instabilities or any significant artifacts in the resulting trajectories.[28]

Sidechain interaction and peptide folding simulations

Before applying GPU-GBMV2/SA to protein simulations, we first validated its ability to accurately describe interactions between various backbone and side chain chemical groups. The balance of these interactions governs the ability of a force field to properly capture the protein conformational equilibria. Figure 5 compares the free energy profiles of two representative sidechains pairs. It demonstrates that GPU-GBMV2/SA exactly reproduce CPU-GBMV2/SA as expected, and the implicit solvent results also closely match the profiles derived from free energy calculations in TIP3P explicit solvent.[20]

The peptide $(AAQAA)_3$ has been widely used as a model flexible peptide for force field evaluation and calibration.[20,21,28] Figure 6 shows the time evolution of helicity of $(AAQAA)_3$ during two independent control and folding simulations at 270 K in GPU-GBMV2/SA. It can be observed that several reversible conformational transitions between the (partial) helices and unfolded structures were sampled in both simulations within 200 ns, indicating that the implicit treatment of solvent using the GBMV2/SA model greatly facilitates protein conformational sampling without the friction from explicit solvent molecules. The resulting average residue helicity profiles are well converged; the root-mean-square deviation (RMSD) value between results from control and folding GPU runs are only 0.021. These results are comparable to results derived from previous replica exchange simulations on a CPU platform.[28]

Computational efficiency

Figure 7 summarizes the performance of GPU-GBMV2/SA in comparison to the CPU version for six folded and unfolded proteins of various sizes and topologies. It shows that the GPU

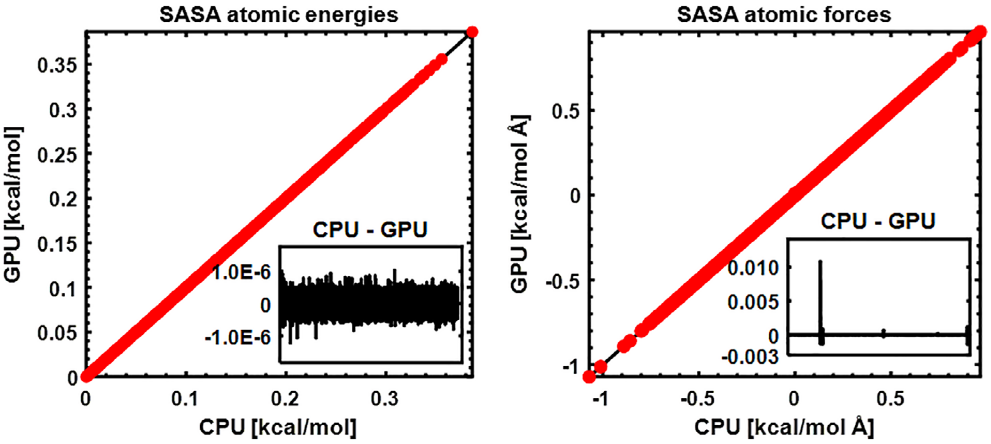

Figure 3. The accuracy of GPU and CPU-GBMV2/SA in calculating atomic SASA energies (left) and forces (right). The surface tension coefficient is 5 cal/mol Å^2. All atoms from 22 small proteins are included. The inserted panels show the difference between CPU and GPU results (in the same unit, kcal/mol or kcal/mol/Å for each of all atoms from the protein test set). [Color figure can be viewed at wileyonlinelibrary.com]

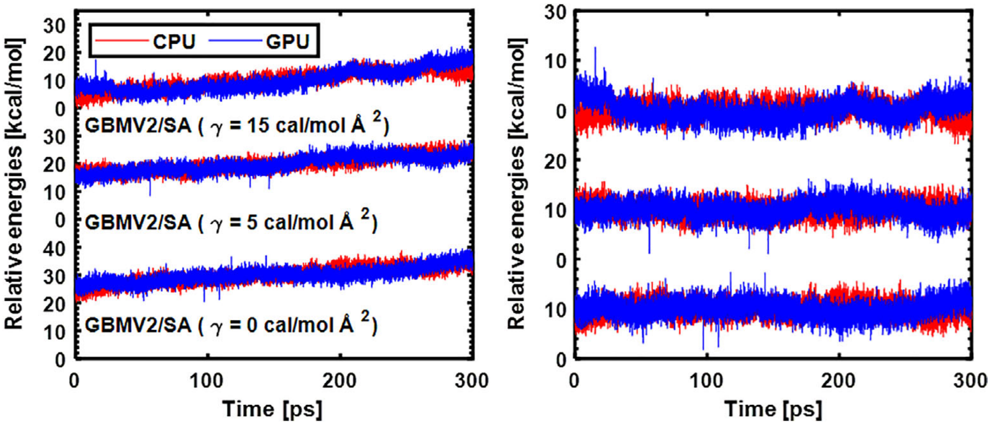

Figure 4. Energy conservation of MD simulations for a small protein (PDB: 1BDC) in CPU- and GPU-GBMV2/SA. Energies versus simulation time before (left) and after (right) removing the linear drift. The time step was set to 1 fs. The relative CPU/GPU energy drift rates are 0.0072/0.0085, 0.0048/0.0068 and 0.0071/0.0110 (unit: %/ps) for three cases (γ = 0, 5, and 15 cal/ mol Å^2), respectively. The standard fluctuations of CPU/GPU energies (after removing the linear drift) are 1.5434/1.5942, 1.4566/1.5963, and 1.5934/2.0047 kcal/mol), for three cases, respectively. Only the last 100-ps trajectories were included in the energy drift analysis. [Color figure can be viewed at wileyonlinelibrary.com]

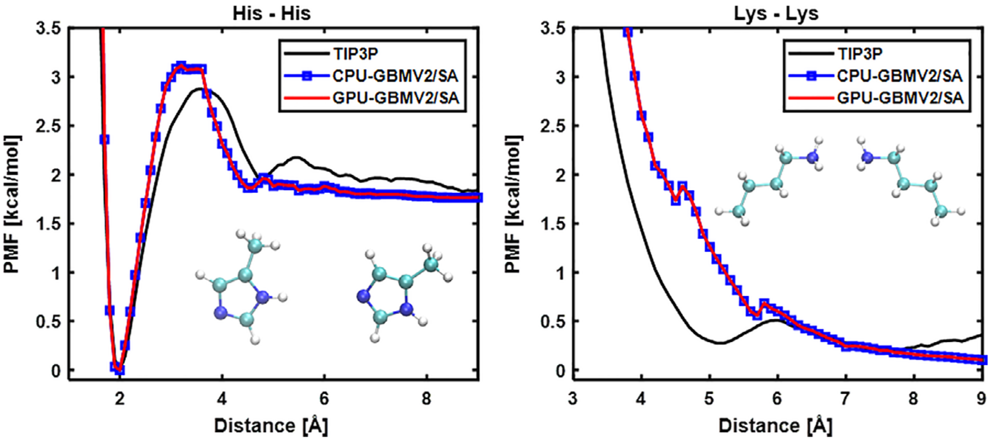

Figure 5. Free energy profiles of interactions for two sidechain pairs, (left) His–His and (right) Lys–Lys, in TIP3P, CPU- and GPU-GBMV2/SA solvent. γ = 5 cal/mol Å^2 was used. [Color figure can be viewed at wileyonlinelibrary.com]

version offers ~ 60 to 70-fold speed up, with the larger systems exhibiting slightly superior efficiency. We note that a faster version of CPU-GBMV2/SA has been previously developed,[25] which extensively utilizes pre-calculated data arrays to speed up the evaluation of Born radii and derivatives. Our current testing shows that the fast CPU version is ~50% more efficient than the standard one. We also note that the current multicore parallel implementation of GBMV2/SA scales poorly beyond over 8 cores, with the speedup maxing out ~6x using 16 Intel Xeon E5-2620 v4 2.10GHz CPU cores (see Table S1). We have also profiled the timing distribution of each kernel in GPU-GBMV2/SA. The four kernels associated with the lookup table account for

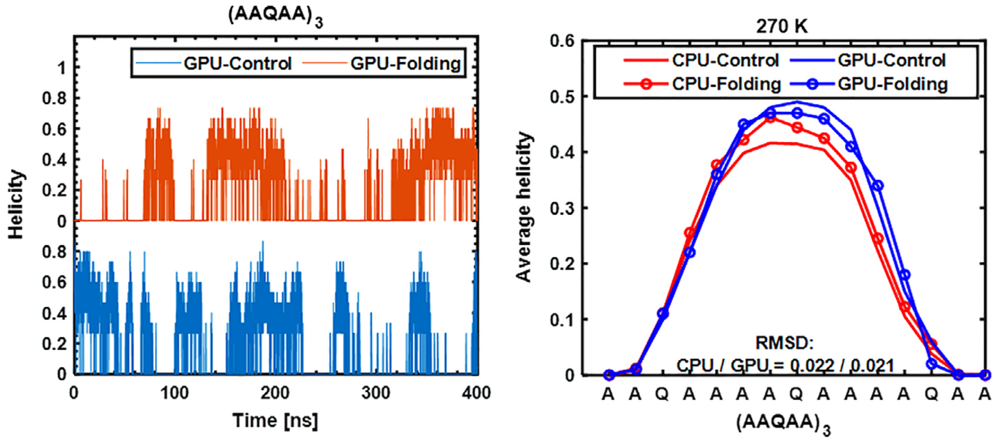

Figure 6. Left: Helicity of (AAQAA)$_3$ during folding and control GPU-GBMV2/SA simulations at 270 K. Right: Average residue helicity profiles calculated from GPU simulations in comparison with previous results derived from CPU simulations.[29] The RMSD values shown are the root mean square differences between profiles derived from control and folding simulations. [Color figure can be viewed at wileyonlinelibrary.com]

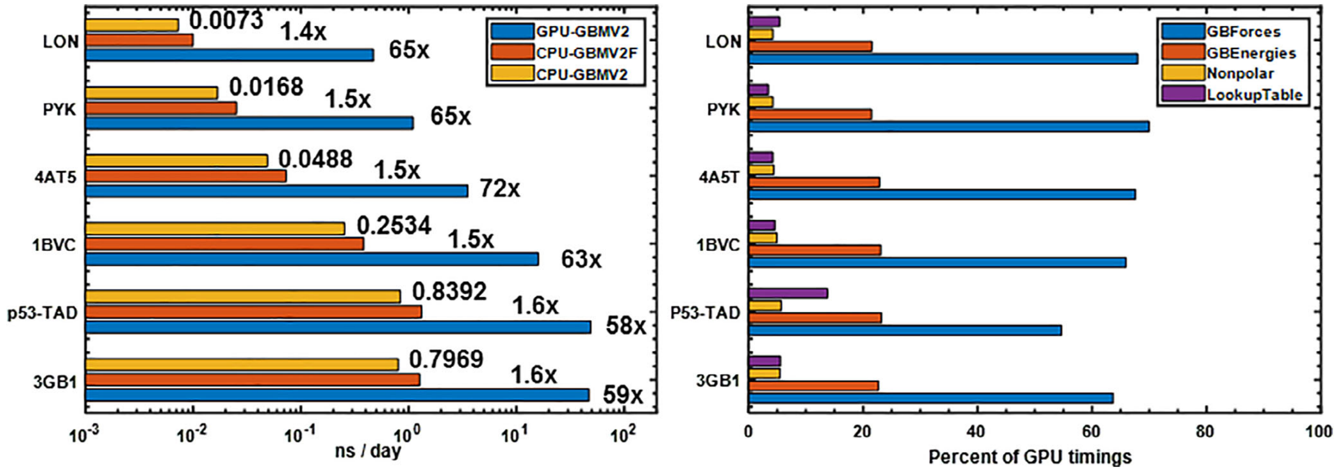

Figure 7. (Left) Timings of CPU- and GPU-GBMV2/SA simulations. The numbers next to the CPU-GBMV2/SA bars are the production time in ns/day, and the ratios next to the fast CPU-GBMV2/SA and GPU-GBMV2/SA are folds of speedup compared to CPU-GBMV2/SA. The production rates of GPU simulations are (in ns/day): 47.00 (3GB1), 48.96 (p53-TAD), 15.93 (1BVC), 3.52 (4AT5), 1.10 (PYK) and 0.47 (LON). (Right) Percentages of time spent in various parts of GPU-GBMV2/SA calculation, including constructing and updating the lookup table ("Lookup Table"), nonpolar energies and forces ("Nonpolar") and electrostatic energies and forces calculations ("GBEnergies" and "GBForces"). The GPU and CPU calculations were done on one NVIDIA TITAN X (Pascal) and one core of Intel Xeon E5-2620 v4 2.10 GHz CPU, respectively. [Color figure can be viewed at wileyonlinelibrary.com]

only ~5% of the time, although it is memory intensive. The calculations of electrostatic and nonpolar terms take up around 85 and 7% of the total time, respectively. Thus, the bottleneck of the GBMV2/SA algorithm is clearly the calculation of Born radii and their derivatives. The reason is that the calculation of the Born radius for each atom involves a complicated expression based on around 800 numerical quadrature points and 100 neighbor atoms for each grid point; the derivatives of Born radii involve even more extensive operations (see detailed expressions in Appendix S1). Consequently, the GB force calculations are about three-fold slower than the GB energies calculations.

Conclusions

A GPU-accelerated GBMV2/SA model has been implemented within the CHARMM/OpenMM interface, including both the GB electrostatic and SASA nonpolar solvation terms. The GB term has been implemented based on the existing CUDA kernels of the GPU-GBSW model.[41] Together with a SASA nonpolar term, it provides a complete and accurate GBMV2/SA implicit solvent model that is suitable for protein simulations. Results show that the GPU-GBMV2/SA solvation energies and forces are essentially the same as those in the original CPU-GBMV2/SA model with negligible errors, giving rise to similar energy conservation properties. Benchmarks based on a set of folded and unfolded proteins show that the current implementation of GPU-GBMV2/SA offers about 60- to 70-fold speedup on a single NVIDIA TITAN X graphics card compared to a single core of an Intel Xeon E5-2620 v4 2.10GHz CPU. While the speedup is somewhat modest compared to those achieved by GBSW or GBSA/OBC in Amber, it is still quite substantial and will enable the application of GBMV2 for MD of larger systems and for longer timescales for both folded and unfolded proteins.

We note that there is still room for further improvement of the computational efficiency of GPU-GBMV2/SA. For example, a key bottleneck is the large lookup table required for evaluating the volume integrals due to longer tails required for analytical approximation of MV. The numbers of atoms within the proximity of each grid point can be as high as ~100. It is likely that the list can be truncated without significant reduction to the numerical accuracy. One can also optimize the usage of computational memory of lookup table array, for example, by using the flexible allocation or avoiding the allocation by looping neighbor grid boxes. Development of the GPU-GBMV2/SA algorithm will also allow one to perform extensive folding simulations of model proteins and peptides, in order to critically evaluate the ability of the simple SASA nonpolar model for describing the conformation equilibria.[9] This will pave the way for further development of better treatments of the nonpolar solvation that can more accurately capture the conformational dependence of solvation free energies.

Acknowledgments

This work was supported by National Science Foundation (MCB 1817332 to JC) and National Institutes of Health (R35 GM130587 to CLB and R35GM126948 to MF).

Keywords: CHARMM · generalized Born · OpenMM · protein conformation · solvation free energy

How to cite this article: X. Gong, M. Chiricotto, X. Liu, E. Nordquist, M. Feig, C. L. Brooks III, J. Chen. *J. Comput. Chem* **2020**, *41*, 830–838. DOI: 10.1002/jcc.26133

Additional Supporting Information may be found in the online version of this article.

[1] S. M. Vaiana, M. Manno, A. Emanuele, M. B. Palma-Vittorelli, M. U. Palma, *J Biol Phys* **2001**, *27*, 133.
[2] J. W. Ponder, D. A. Case, *Adv. Protein Chem.* **2003**, *66*, 27.
[3] A. D. Mackerell, *J. Comput. Chem.* **2004**, *25*, 1584.
[4] J. Wagoner, N. A. Baker, *J. Comput. Chem.* **2004**, *25*, 1623.
[5] A. Onufriev, D. A. Case, D. Bashford, *J. Comput. Chem.* **2002**, *23*, 1297.
[6] C. J. Cramer, D. G. Truhlar, *Chem. Rev.* **1999**, *99*, 2161.
[7] B. Roux, T. Simonson, *Biophys. Chem.* **1999**, *78*, 1.
[8] M. Feig, C. L. Brooks, *Curr. Opin. Struct. Biol.* **2004**, *14*, 217.
[9] J. Chen, C. L. Brooks, *Phys. Chem. Chem. Phys.* **2008**, *10*, 471.
[10] J. H. Chen, C. L. Brooks, J. Khandogin, *Curr. Opin. Struc. Biol.* **2008**, *18*, 140.
[11] N. A. Baker, *Curr. Opin. Struct. Biol.* **2005**, *15*, 137.
[12] M. K. Gilson, M. E. Davis, B. A. Luty, J. A. Mccammon, *J. Phys. Chem.* **1993**, *97*, 3591.
[13] A. Nicholls, B. Honig, *J. Comput. Chem.* **1991**, *12*, 435.
[14] D. Bashford, D. A. Case, *Annu. Rev. Phys. Chem.* **2000**, *51*, 129.
[15] V. Tsui, D. A. Case, *Biopolymers* **2001**, *56*, 275.
[16] U. Haberthur, A. Caflisch, *J. Comput. Chem.* **2008**, *29*, 701.
[17] A. Onufriev, D. Bashford, D. A. Case, *Proteins* **2004**, *55*, 383.
[18] E. Gallicchio, K. Paris, R. M. Levy, *J. Chem. Theory Comput.* **2009**, *5*, 2544.
[19] W. P. Im, M. S. Lee, C. L. Brooks, *J. Comput. Chem.* **2003**, *24*, 1691.
[20] J. H. Chen, W. P. Im, C. L. Brooks, *J. Am. Chem. Soc.* **2006**, *128*, 3728.
[21] J. H. Chen, *J. Chem. Theory Comput.* **2010**, *6*, 2790.
[22] A. Ghosh, C. S. Rapp, R. A. Friesner, *J. Phys. Chem. B* **1998**, *102*, 10983.
[23] M. S. Lee, F. R. Salsbury, C. L. Brooks, *J. Chem. Phys.* **2002**, *116*, 10606.
[24] M. S. Lee, M. Feig, F. R. Salsbury, C. L. Brooks, *J. Comput. Chem.* **2003**, *24*, 1348.
[25] J. Chocholousova, M. Feig, *J. Comput. Chem.* **2006**, *27*, 719.
[26] J. Mongan, C. Simmerling, J. A. McCammon, D. A. Case, A. Onufriev, *J. Chem. Theory Comput.* **2007**, *3*, 156.
[27] H. Nguyen, D. R. Roe, C. Simmerling, *J. Chem. Theory Comput.* **2013**, *9*, 2020.
[28] K. H. Lee, J. H. Chen, *J. Comput. Chem.* **2017**, *38*, 1332.
[29] F. M. Richards, *Annu. Rev. Biophys. Biol.* **1977**, *6*, 151.
[30] A. Juneja, M. Ito, L. Nilsson, *J. Chem. Theory Comput.* **2013**, *9*, 834.
[31] J. L. Knight, C. L. Brooks, *J. Comput. Chem.* **2011**, *32*, 2909.
[32] X. R. Liu, J. H. Chen, *J. Chem. Theory Comput.* **2019**, *15*, 4708.
[33] R. B. Best, W. W. Zheng, J. Mittal, *J. Chem. Theory Comput.* **2014**, *10*, 5113.
[34] S. Piana, J. L. Klepeis, D. E. Shaw, *Curr. Opin. Struc. Biol.* **2014**, *24*, 98.
[35] S. Piana, A. G. Donchev, P. Robustelli, D. E. Shaw, *J. Phys. Chem. B* **2015**, *119*, 5113.
[36] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, M. Karplus, *J. Comput. Chem.* **2009**, *30*, 1545.
[37] D. A. Case, T. E. Cheatham, T. Darden, H. Gohlke, R. Luo, K. M. Merz, A. Onufriev, C. Simmerling, B. Wang, R. J. Woods, *J. Comput. Chem.* **2005**, *26*, 1668.
[38] R. Salomon-Ferrer, A. W. Gotz, D. Poole, S. Le Grand, R. C. Walker, *J. Chem. Theory Comput.* **2013**, *9*, 3878.
[39] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435.
[40] J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, K. Schulten, *J. Comput. Chem.* **2005**, *26*, 1781.
[41] D. E. Tanner, J. C. Phillips, K. Schulten, *J. Chem. Theory Comput.* **2012**, *8*, 2521.
[42] P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A. Beauchamp, T. J. Lane, L. P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, V. S. Pande, *J. Chem. Theory Comput.* **2013**, *9*, 461.
[43] A. W. Gotz, M. J. Williamson, D. Xu, D. Poole, S. Le Grand, R. C. Walker, *J. Chem. Theory Comput.* **2012**, *8*, 1542.
[44] E. J. Arthur, C. L. Brooks, *J. Comput. Chem.* **2016**, *37*, 927.
[45] H. Huang, C. Simmerling, *J. Chem. Theory Comput.* **2018**, *14*, 5797.
[46] R. M. Levy, L. Y. Zhang, E. Gallicchio, A. K. Felts, *J. Am. Chem. Soc.* **2003**, *125*, 9523.
[47] J. H. Chen, C. L. Brooks, *J. Am. Chem. Soc.* **2007**, *129*, 2444.
[48] W. C. Still, A. Tempczyk, R. C. Hawley, T. Hendrickson, *J. Am. Chem. Soc.* **1990**, *112*, 6127.
[49] J. Srinivasan, M. W. Trevathan, P. Beroza, D. A. Case, *Theor. Chem. Acc.* **1999**, *101*, 426.

Received: 23 September 2019
Revised: 21 November 2019
Accepted: 7 December 2019
Published online on 24 December 2019