Efficient Implementation of Constant pH Molecular Dynamics on Modern Graphics Processors

Evan J. Arthur[a] and Charles L. Brooks III*[b]

The treatment of pH sensitive ionization states for titratable residues in proteins is often omitted from molecular dynamics (MD) simulations. While static charge models can answer many questions regarding protein conformational equilibrium and protein–ligand interactions, pH-sensitive phenomena such as acid-activated chaperones and amyloidogenic protein aggregation are inaccessible to such models. Constant pH molecular dynamics (CPHMD) coupled with the Generalized Born with a Simple sWitching function (GBSW) implicit solvent model provide an accurate framework for simulating pH sensitive processes in biological systems. Although this combination has demonstrated success in predicting pK_a values of protein structures, and in exploring dynamics of ionizable side-chains, its speed has been an impediment to routine application. The recent availability of low-cost graphics processing unit (GPU) chipsets with thousands of processing cores, together with the implementation of the accurate GBSW implicit solvent model on those chipsets (Arthur and Brooks, J. Comput. Chem. 2016, 37, 927), provide an opportunity to improve the speed of CPHMD and ionization modeling greatly. Here, we present a first implementation of GPU-enabled CPHMD within the CHARMM-OpenMM simulation package interface. Depending on the system size and nonbonded force cutoff parameters, we find speed increases of between one and three orders of magnitude. Additionally, the algorithm scales better with system size than the CPU-based algorithm, thus allowing for larger systems to be modeled in a cost effective manner. We anticipate that the improved performance of this methodology will open the door for broad-spread application of CPHMD in its modeling pH-mediated biological processes. © 2016 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.24435

Introduction

Proteins typically maintain their native structure and optimal functionality under a narrow range of pH.[1–3] Consequently, many biological systems tightly control local solvent pH to tune the effectiveness of enzymes, or to promote a useful protein conformation.[1,4,5] Mitochondrial ATP synthase utilizes a transmembrane proton gradient to power its rotary catalysis mechanism,[6–8] and the departure from a normal pH range is known to be a driving force in forming the amyloid fibrils associated with Alzheimer's disease.[9,10] Additional examples of pH driven processes include the proton-activated gate mechanism of the KcsA potassium channel,[11] and the catalytic pathway of dihydrofolate reductase.[12] Finally, a notable survey by Aguilar et al. showed that about 60% of the protein–ligand complexes indicated that at least one titratable residue of the protein assumed a different protonation state between bound and unbound states.[13] Although important to many biological processes, pH-dependence in biomacromolecule simulations remains a nonstandard tool that awaits both wider acceptance, and finer tuning of its models.

Typical molecular dynamics (MD) simulations fix all amino acid protonation states to those of isolated residues in a neutral pH environment. While this pH-insensitive approach is sufficient to fold some proteins and observe their conformational equilibria,[14] it arguably fails to capture phenomena dependent on local ionization effects of side-chains or perturbations to a residue's pK_a.[15,16] This failure is particularly problematic for histidine residues, in that they have two hydrogen atoms that titrate with near-neutral pH. This ionizability indicates that in biologically relevant pH environments histidine's protonation state and tautomeric configuration are often unclear.[17]

In recent decades a series of models of varying complexity and accuracy promise to bring accurate pH responsiveness to MD simulations. Protonation-state modeling of amino acids in MD simulations is based on setting up a pH-sensitive extended Hamiltonian that modifies the force field parameters and structure of a given molecule. This began by discretely titrating protons, and progressing a simulation using instantaneous switches between protonated and unprotonated states. Mertz and Pettitt used an open system Hamiltonian to model the titration of acetic acid,[18] and Sham et al. applied a linear response approximation through the protein-dipoles Langevin-dipoles model to calculate lysozyme residue pK_a values.[19] Additional work has been done where Monte Carlo (MC) sampling guides the protonation state of an otherwise classical MD simulation. Baptista et al. used explicitly represented

[a] E. J. Arthur
Department of Chemistry, University of Michigan, 930 N. University Ave, Ann Arbor, Michigan 48109
[b] C. L. Brooks III
Department of Chemistry and Biophysics Program, University of Michigan, 930 N. University Ave, Ann Arbor, Michigan 48109
E-mail: brookscl@umich.edu
Contract grant sponsor: NIH; Contract grant numbers: GM103695 and GM107233 (to C.L.B.)

solvent molecules with an implicit solvent Poisson–Boltzmann function to determine protonation states.[20,21] Meanwhile, Mongan et al. utilized generalized Born (GB) implicit solvation to add a solvation free energy component to the protonation function.[22] While all these discrete models can predict pK_a values for individual amino acids to within one pK unit, they are computationally expensive. Whether the expense stems from the need to relax numerical instabilities caused by instantaneous protonation/deprotonation events, or from the MC algorithms' ability to titrate only one hydrogen at a time, such methods may require an unreasonable amount of time to study large systems with many titratable groups.

One possible solution to these issues with discrete titration methods is to use continuous titration of H^+ atoms. Brooks and coworkers developed one such method called constant pH molecular dynamics (CPHMD), which uses λ-dynamics coupled to transitions between protonation states.[23,24] This method uses the Generalized Born implicit solvent model with a Simple sWitching function (GBSW) model,[25] or the related Generalized Born with Molecular Volume (GBMV) model,[24] to efficiently couple the protonation state to the solvation free energy of the molecule. Khandogin and Brooks then introduced proton tautomerism capabilities to this method, which allows multisite titrating residues, such as histidines, to be modeled more accurately.[26] Since the method is continuous, there are no instantaneous protonation/deprotonation events, and multiple residues can titrate simultaneously. Additionally, such continuous titration methods allow for the efficient coupling of protonation states among neighboring residues. The result is a pH simulation method that can calculate pK_a values of protein structures to within 0.75 pK units,[16] and can resolve the dominant folding pathway of the pH-sensitive HdeA homodimer.[15]

CPHMD's efficiency, however, is bound by the rate-limited component of the calculation: the GBSW solvent model. As such, when running on a single-core central processing unit (CPU), CPHMD achieves on the order of 1 nanosecond (ns) of simulation time per day when simulating a solute system of about 1000 atoms. Since typical uses of CPHMD, such as predicting pK_a shifts of protein residues, may require many nanoseconds of simulation time,[16] even smaller proteins, such as lysozymes, may require about a week to converge on useful results. Larger systems, such as asymmetric viral capsid subunits with tens of thousands of atoms, may require unreasonably long simulation times if captured in full atomic detail.[27,28] Fortunately, the GBSW solvent model has recently been refactored to function on new, parallel graphics processing unit (GPU) hardware, and is now between 1 and 2 orders of magnitude faster than its CPU counterpart.[16,29] By incorporating the CPHMD model into the GPU-GBSW algorithm, there holds the promise of speeding up pH simulations substantially.

This study represents an increment in the ongoing adaptation of efficient and useful algorithms onto parallel-processing GPUs. Such chipsets can contain thousands of processing cores, and are able to process C-like languages such as Open Computing Language and Compute Unified Device Architecture (CUDA). This combination of features has opened up a new frontier of parallel processing where expensive computer clusters can be replaced with single, affordable graphics cards. Simulation packages such as CHARMM,[30] AMBER,[31] OpenMM,[32] GROMACS,[33] and NAMD[34] all offer GPU-accelerated options for many types of studies, and most of those options receive speed increases of greater than an order of magnitude over their CPU counterparts.

Due to OpenMM's effectiveness in harnessing the capabilities of GPUs with a wide variety of hardware, a CHARMM-OpenMM interface was developed to combine the strengths of both simulation packages.[30,32] CHARMM's robust algorithms can be used to design and parameterize a simulation, and OpenMM's efficient GPU-based algorithms can be used to propagate dynamics.[30,32] Now with the recent incorporation of the GBSW solvent model into the CHARMM-OpenMM interface, many of CHARMM's algorithms parameterized for use with GBSW, such as CPHMD, can be adapted for parallel processing on GPUs as well. In this study, we take advantage of the recent incorporation of GBSW onto GPUs, and discuss the adaptation of CPHMD onto this new parallel architecture. First we explain the underlying theory behind λ-dynamics: how a λ coordinate is used to represent the titration state of a residue, and how that coordinate is propagated. Then, we delve into how it was originally implemented for CHARMM, and examine fitting CPHMD into the GBSW algorithm. Here, we discuss the algorithmic improvements, and show how many force contributions on λ are calculated alongside the free energy of solvation. Finally, we present benchmarks achieved by the new algorithm, and comment on future directions for pH simulations.

Methods

The λ coordinates and their underlying energy function for single-site titration

For clarity in following discussions, we present the underlying theory of CPHMD. We start by setting up the framework for a single residue with one titrating hydrogen. The rudimentary picture of titration events is an equilibrium association/disassociation reaction of a model compound $A_{(aq)}$ in aqueous solution from a titrating proton.

$$AH_{(aq)} \leftrightarrow A^-_{(aq)} + H^+_{(aq)} \tag{1}$$

Here, the protonation free energy is defined by

$$\Delta G^{exp}(\text{model}) = -k_B T \ln 10\left(\text{p}K_a^{exp} - \text{pH}\right) \tag{2}$$

where k_B is Boltzmann's constant, and T is the temperature. We can approximate the above equations through classical simulations by interpreting the protonation interaction as a change in free energies:

$$\begin{aligned}&\Delta G^{exp}(\text{protein}) - \Delta G^{exp}(\text{model})\\ &= \Delta G^{classical}(\text{protein}) - \Delta G^{classical}(\text{model})\end{aligned} \tag{3}$$

This relationship then leads to an estimate of experimental free energy of protonation for a single titrating site:

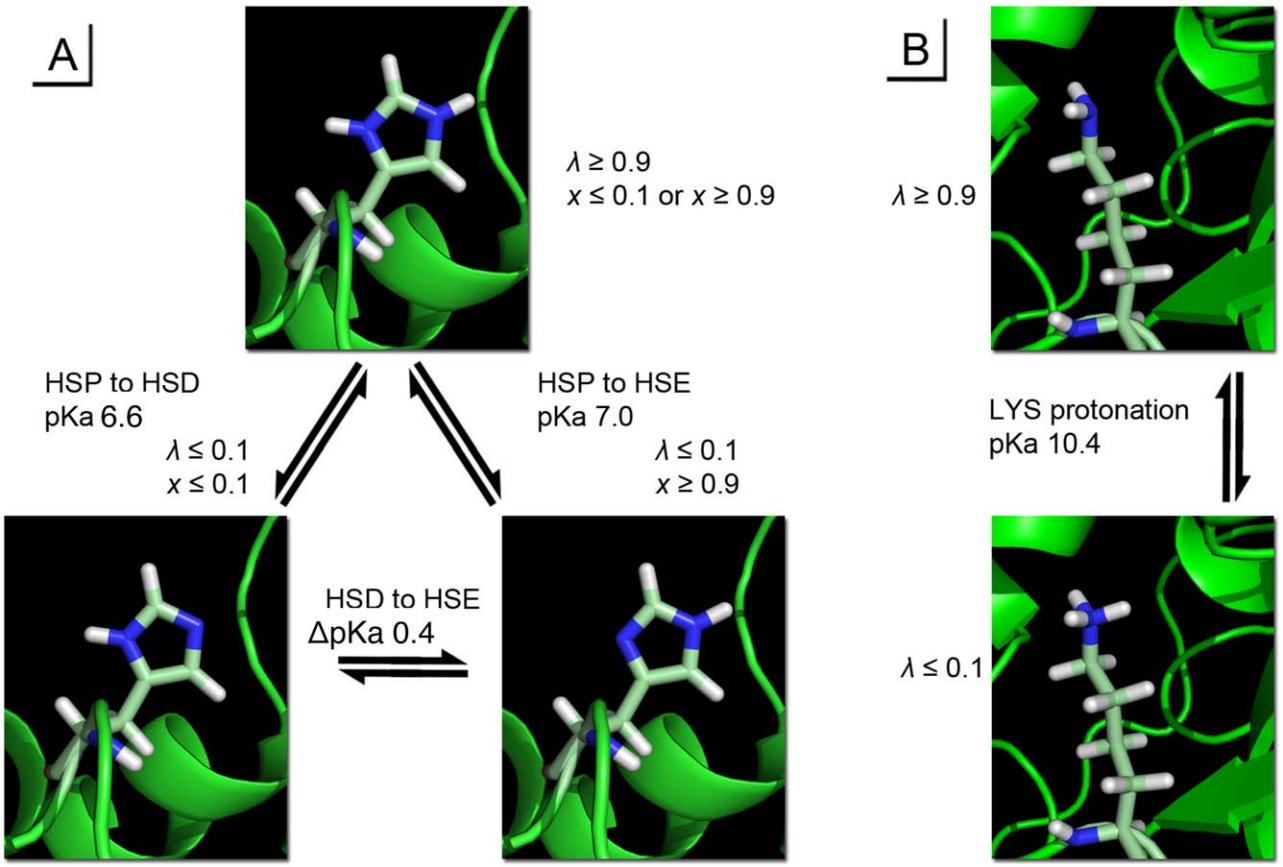

Figure 1. Shown are Cartoons of the protonated and unprotonated states of A) histidine and B) lysine. Also noted are the reference pK_a values of each transition when occurring in an isolated residue, as well as the λ values at each state. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$\Delta G^{\text{exp}}(\text{protein}) = \Delta G^{\text{classical}}(\text{protein}) - \Delta G^{\text{classical}}(\text{model}) + \Delta G^{\text{exp}}(\text{model}) \quad (4)$$

From this perspective, we infer that titratable groups have an intrinsic free energy of protonation that is perturbed by the protein environment mainly through nonbonded interactions. We model this perturbation by extending the system's Hamiltonian with a nongeometric dimension of λ. As mentioned in the introduction, the CPHMD model uses a series of λ coordinates where each λ value tracks the progress of protonation-deprotonation events at a single titration site. For a particular residue i, these coordinates are generated from

$$\lambda_i = \sin^2(\theta_i) \quad (5)$$

where i is the residue being titrated. In this form the θ variable is bound to all real numbers, and λ is bound to the continuous range $0 \leq \lambda_i \leq 1$. The sine-squared function then favors λ values near the boundary protonated (1) and unprotonated (0) states. Because λ is only physically relevant as it nears these boundary states, we impose cutoffs on interpreting λ. In CPHMD an unprotonated state is $\lambda_i \leq 0.1$, a protonated state is $\lambda_i \geq 0.9$, and a mixed state is $0.1 < \lambda_i < 0.9$. Figure 1 illustrates the protonation states and their corresponding λ values. Potentials and their derivative forces on λ are then interpreted as potentials and forces on θ.

The potential energy that governs protonation states contains five λ-dependent components. We start with the pH dependence of the deprotonation free energy as follows from ΔG^{exp} in eq. (4). This potential connects λ to the pK_a of a residue in its isolated, reference state:

$$U^{\text{pH}}(\lambda_i) = \lambda_i(\text{p}K_a(i) - pH)(k_B T \ \ln\ 10) \quad (6)$$

Here, p$K_a(i)$ is the pK_a of titrating group i. Next we have the potential of mean force (PMF) along the λ coordinate from ΔG^{model}. This term corresponds to the negative of free energy needed to deprotonate a model residue:

$$U^{\text{model}}(\lambda_i) = A_i(\lambda_i - B_i)^2 \quad (7)$$

Equation (7) is a quadratic fit to the thermodynamic work potential of deprotonating a model compound, and it splits the protonation state into two low-energy wells that represent the protonated and unprotonated states. Then a barrier potential is added that disfavors mixed states of λ:

$$U^{\text{barrier}}(\lambda_i) = 4\beta_i(\lambda_i - 1/2)^2 \quad (8)$$

The barrier scaling parameter β_i is an empirical coefficient designed to tune the propensity for a λ value to remain in either protonated or unprotonated states, while facilitating

transitions between them. In the current iteration of CPHMD, β_i assumes a value of either 2.5 or 1.75 kcal/mol, depending on the residue. Finally, we arrive at the two charge-dependent potentials: the Coulombic and GB. The classical Coulombic potential is

$$U^{\text{elec}}(\lambda_i)=\sum_{a,\,i}\sum_{b}K^{\text{elec}}\frac{q_{a,\,i}(\lambda_i)\;\;q_b}{r_{ab}} \tag{9}$$

Here, K^{elec} is Coulomb's constant, $q_a$and q_b are the partial charges of atoms a and b respectively, and r_{ab} is the distance between those atoms. Note that this potential for residue i includes the interactions between all atoms a in residue i to all other atoms in the system. Meanwhile, $q_a(\lambda_i)$ is a λ-dependent charge of atom a, which follows the form

$$q_{a,\,i}(\lambda_i)=\lambda_i\,q_{a,\,i}^{\text{unprot}}\;+\;(1-\lambda_i)q_{a,\,i}^{\text{prot}} \tag{10}$$

where charges on titrating atom a can be in protonated ($q_{a,\,i}^{\text{prot}}$) and unprotonated ($q_{a,\,i}^{\text{unprot}}$) states. We note that in an effective charge model of pH, titrating residues are allowed to interact. As such, any atom b from a titrating residue j interacting with residue i has its own $q_{b,\,j}^{\text{prot}}$ and $q_{b,\,j}^{\text{unprot}}$. Thus, the partial charge $q_b$follows one of two possibilities:

$$q_b=\begin{cases}q_b & \text{non}-\text{titrating}\\ \lambda_j\,q_{b,\,j}^{\text{unprot}}\;+\;\left(1-\lambda_j\right)q_{a,\,j}^{\text{prot}} & \text{titrating}\end{cases} \tag{11}$$

That is if atom b lies in a nontitrating residue, that atom's partial charge is simply the standard partial charge from that residue's force field. If atom b lies in a titrating residue j and its charge is affected by the protonation state of j, then its partial charge is derived from the same λ-dependent relationship from eq. (10). Since atoms near a titrating site can have their partial charges affected by titration states, many more than the titrating hydrogen atoms can possess a λ-dependent charge state. We also note that at times $j=i$ if we observe the Coulombic interaction between two atoms on the same titrating residue. The final λ-dependent potential is that from the GB solvent model as expressed in the Still equation:[35]

$$U^{GB}(\lambda_i)=\sum_{a,\,i}\sum_{b}\tau\frac{q_{a,\,i}(\lambda_i)\;\;q_b}{f_{ab}^{GB}} \tag{12}$$

where

$$f_{ab}^{GB}=\left[r_{ab}^2+R_a^{\text{Born}}R_b^{\text{Born}}\exp\left(-r_{ab}^2/\left(4R_a^{\text{Born}}R_b^{\text{Born}}\right)\right)\right]^{1/2} \tag{13}$$

Here, $q_a(\lambda_i)$ and q_b follow the same form as in eqs. (10) and (11), respectively; r_{ab} is the distance between atoms a and b; τ is the factor that scales the Born energy by the difference in dielectric values at the dielectric boundary and by any contributing salt effects;[36] and the values R_a^{Born} and R_b^{Born} represent the Born radii of atoms a and b, respectively. The Born radii are the effective distance between an atom and the solute–solvent dielectric boundary, and they are calculated through volumetric integration following the GBSW implicit solvent model.[25]

If we pull together the complete potential for a titrating residue i from eqs. (6) through (13), then we arrive at the form

$$\begin{aligned}U_i^{\text{total}}(\lambda_i)=&U_i^{\text{pH}}(\lambda_i)\;+\;U_i^{\text{model}}(\lambda_i)\;+\;U_i^{\text{barrier}}(\lambda_i)\\&+\;U_i^{\text{elec}}(\lambda_i)\;+\;U_i^{\text{GB}}(\lambda_i)+U_i^{\text{VDW}}\;+\;U_i^{\text{internal}}\end{aligned} \tag{14}$$

The so-called "internal energy" term (U^{internal}) corresponds to the bond, angle, and torsional energy terms of a classical energy force field. In this model, the titration state is dynamically independent of this potential. Although several models of CPHMD include a λ-dependent van der Waals term (U^{VDW}),[26,37,38] during this study it was found that at most it contributes a minimal amount to a given residue's force on λ, while it nearly doubles the calculation time of CPHMD. This term is negligible compared to the force on λ from other effects, and omitting it from the calculation showed no effect on the accuracy of CPHMD. Thus, in the interest of speeding up the original algorithm, the λ-dependent potential U^{VDW} was ignored in this implementation of CPHMD.

Although we now have the proper setup for addressing residues with a single titration site, such as in lysine, we need to address how CPHMD handles tautomerization in residues such as in aspartic acid and histidine.

Proton tautomerism

Similarly to how one λ variable is used to track the progress of titration states of a residue, Khandogin and Brooks incorporated tautomeric behavior into CPHMD by providing residues with a second λ variable, called x, to track the progress of tautomeric states.[26] This arrangement is illustrated in Figure 1a with histidine. Just as in λ dynamics for titration states, transitions between tautomeric states are linearly interpolated using the x variable. What results are potentials that become bivariate in λ and x, and each tautomeric residue has four charge states: tautomer A in protonated and unprotonated states, and tautomer B in protonated and unprotonated states. What we shall see later is that residues can have equivalent states in this setup. Histidine's protonated state, for example, is a residue saturated with protons. As such tautomers A and B of the protonated state are equivalent. We now review the influence of including two λ parameters for a tautomeric titrating residue. The pH dependent potential becomes

$$\begin{aligned}U^{\text{pH}}(\lambda_i,x_i)=&\lambda_i[x_i\left(\text{p}K_{\text{a}}^{\text{A}}(i)-\text{pH}\right)\\&+(1-x_i)\left(\text{p}K_{\text{a}}^{\text{B}}(i)-\text{pH}\right)](k_BT\;\ln\;10)\end{aligned} \tag{15}$$

where the pK_{a} values of tautomers A and B are pK_{a}^{A} and pK_{a}^{B} respectively. While these pK_{a} values for aspartic acid and glutamic acid are equivalent and only serve as a sampling expedient,[26] in residues with asymmetric titrating sites such as histidine they are not. The PMF for protonation becomes a bivariate polynomial from eq. (7), which then expands into the general form

$$U^{\text{model}}(\lambda_i, x_i) = a_0\lambda_i^2 x_i^2 + a_1\lambda_i^2 x_i + a_2\lambda_i x_i^2 + a_3\lambda_i x_i + a_4\lambda_i^2 + a_5 x_i^2 + a_6\lambda_i + a_7 x_i + a_8 \quad (16)$$

The barrier potential is simply a summation of terms that disfavor the mixed states of both λ and x, and follows the form

$$U^{\text{barrier}}(\lambda_i, x_i) = 4\beta_i^{\lambda}(\lambda_i - 1/2)^2 + 4\beta_i^{x}(x_i - 1/2)^2 \quad (17)$$

Note that there are two barrier scaling parameters β_i^{λ} and β_i^{x} for λ and x. Although different biases for tautomeric and protonation transitions are possible in this equation, in the discussed CPHMD model they are identical for all titrating residues.

The charge-dependent potentials in eqs. (9) and (12) are only modified in that charges for atoms can now be dependent on the new x coordinate. The Coulombic and GB potentials then follow the forms

$$U^{\text{elec}}(\lambda_i, x_i) = \sum_{a,\,i}\sum_{b} K^{\text{elec}}\frac{q_{a,\,i}(\lambda_i, x_i)\ q_b}{r_{ab}} \quad (18)$$

and

$$U^{\text{GB}}(\lambda_i, x_i) = \sum_{a,\,i}\sum_{b} \tau\frac{q_{a,\,i}(\lambda_i, x_i)\ q_b}{f_{ab}^{\text{GB}}} \quad (19)$$

respectively. The bivariate charge $q_{a,\,i}(\lambda_i, x_i)$ then follows the form

$$q_{a,\,i}(\lambda_i, x_i) = \lambda_i\left[x_i\, q_{a,\,i}^{A,\,\text{unprot}} + (1-x_i)q_{a,\,i}^{B,\,\text{unprot}}\right] + (1-\lambda_i)\left[x_i\, q_{a,\,i}^{A,\,\text{prot}} + (1-x_i)q_{a,\,i}^{B,\,\text{prot}}\right] \quad (20)$$

Where charges on titrating atom a are derived from the protonated and unprotonated variants of both A and B tautomers, $q_{a,\,i}^{A,\,\text{prot}}$, $q_{a,\,i}^{A,\,\text{unprot}}$, $q_{a,\,i}^{B,\,\text{prot}}$, and $q_{a,\,i}^{B,\,\text{unprot}}$. Similarly, the charge on atom b emerges as

$$q_b = \begin{cases} q_b & \text{non-titrating} \\ \lambda_j\left[x_j\, q_{b,\,j}^{A,\,\text{unprot}} + (1-x_j)q_{b,\,j}^{B,\,\text{unprot}}\right] & \text{titrating} \\ \quad + (1-\lambda_j)\left[x_j\, q_{b,\,j}^{A,\,\text{prot}} + (1-x_j)q_{b,\,j}^{B,\,\text{prot}}\right] & \end{cases} \quad (21)$$

We now arrive at a general-purpose setup for evaluating the underlying potential for continuous transitions among various charge states of a particular residue. Deriving the forces with respect to λ and x, while important, serves little purpose for illuminating the topics explored in the remainder of this study. With the framework above, we now can discuss the construction of the original algorithm, and the changes made to refactor it for efficient parallel processing on GPUs.

Refactoring CPHMD

The original CPHMD model was built with mathematical precision and function portability in mind. It is a stand-alone module that can be applied to both implicit and explicit solvent systems, and except for atom coordinate and Born radii updates, it receives no input from other functions during a simulation. During the course of a timestep, each titrating coordinate λ_i is scanned to identify the residue type (such as whether the residue has one or two titrating hydrogens), and then an appropriate functional is applied to calculate its pH [eqs. (6) and (15)], model [eqs. (7) and (16)], and barrier [eqs. (8) and (17)] potentials. Next, neighboring atom-atom interactions are scanned for whether one or both atoms reside in titrating groups. If a titrating atom–atom pair is found, then contributions to the electrostatic [eqs. (9) and (18)] and GB [eqs. (12) and (19)] potentials are integrated. Neighboring atom–atom pairs are then scanned again to calculate the VDW potential (ignored in this new iteration of CPHMD). Finally, the force on θ is calculated, and λ via θ is advanced a timestep using Langevin dynamics.[39] In this setup, there are several opportunities presented to us for improving the algorithm both in the efficiency of its execution in parallel, and by weaving portions of the calculation into existent functions elsewhere in the simulation.

We first note that the majority of clock cycles used for calculating λ dynamics are spent on neighboring atom–atom interactions when accumulating the electrostatic and GB potentials. While the calculations required for each atom pair are computationally cheap, the large number of interatomic interactions in a protein containing thousands of atoms can make this multitude of cheap calculations altogether expensive. As show in Figure 2a, about 12% of a 2000-atom simulation is spent only on this calculation.

Both CPHMD and the GBSW solvent model require calculating the Still equation [eqs. (12) and (13)] to address part of the neighboring atom potential, so a significant speed improvement can be made by placing all of CPHMD's atom–atom processes inside the neighboring atom process of the GBSW solvent model. This way, as GBSW produces the solute molecule's electrostatic solvation free energy and its derivative force on atoms, CPHMD processes neighboring atom potentials on λ simultaneously. Thus, the large number of redundant atom–atom distance calculations can be reduced significantly during a simulation. This setup gains additional speedup through GBSW by using OpenMM's efficient parallel possessing of neighboring-atom interactions. As shown in Figure 2, by combining the CPHMD and GBSW algorithms we see that pH modeling with CPHMD accounts for a much smaller fraction of the overall simulation time.

Due to the nature of parallel processing, bottlenecks are often created from the longest portions of nonparallel code. While a single-core process can be sped up dramatically by creating a case-by-case set of calculations, navigating through the additional overhead to make the situation-specific decision can slow parallel processes down. Regarding the equations described earlier, a titrating residue with one tautomer requires fewer calculations than a titrating residue with two. As we place each residue's force calculations in parallel processes, however, the speed of the code is improved by regarding all titrating residues as possessing two tautomeric states.

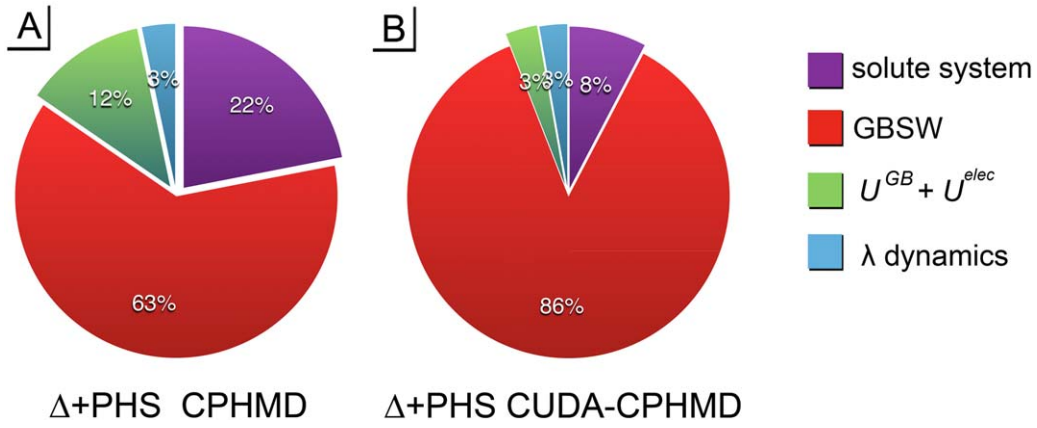

Figure 2. The approximate distributions of CPU time spent on running simulation components of Δ+PHS staphylococcal nuclease molecule. This protein contains 2132 atoms and 37 titrating residues. A) run using the original algorithm on a single processing core in CHARMM. B) run using the newly refactored CUDA-CPHMD algorithm. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In this new implementation of CPHMD, single-titration residues, such as lysine, are given extraneous x coordinates. Lysine then uses the barrier potential from eq. (16), where the x-coupled coefficients a_0, a_1, a_2, a_3, and a_5 are set to a value of 0.0. Without the overhead for residue identification, the longest calculation required, that is calculating the force on θ for a residue with two tautomeric states, is shortened. What results is a speed improvement when calculating all components of the total potential on λ coordinates. As shown in Figure 2b, using the parallel CUDA-CPHMD algorithm for a small system impacts the processing time by approximately 6%, as opposed to 15% for the original algorithm.

Benchmarking CUDA-CPHMD

We finally reach an efficient setup where using the CPHMD model results in little slowdown of the overall simulation time. We chose several systems to benchmark the new algorithm, and explore the speed benefits it offers. We chose the naja atra snake cardiotoxin (PDB: 1CVO),[40] the Δ+PHS hyperstable variant of staphylococcal nuclease (PDB: 3BDC),[41] and the asymmetric subunit of the bacteriophage HK97 head capsid (PDB: 2FT1).[42] This trio provided a range of system sizes and residue configurations. To add additional statistics, the seven proteins of the HK97 capsid were assembled into six additional subsystems, all of which appear in Figure 3 to show for a range of system sizes the speed dependence on system size. All simulations were using the CHARMM22 force field[43,44] using the Langevin integrator[45] with a timestep of 2 femtoseconds. These were NT (constant particle number and temperature) simulations at 298K in unbounded volumes using the CUDA-GBSW solvent model, and CUDA-CPHMD to model titration states and advance λ coordinates. Atomic radii for the GBSW solvent model were provided through work by Chen et al.[46] The hardware specifications of the computer used appear in Table SI1 of the Supporting Information. We found speed improvements of between 1 and 3 orders of magnitude in the CUDA-CPHMD algorithm over its CPU counterpart.

As we combine the improved efficiency and parallel execution of both GBSW and CPHMD (shown in Figs. 3a–3d), substantial speed gains are found in this new version of pH modeling over its predecessor. For smaller 1000-atom systems, we see a speed improvement of over 20-fold when comparing a 12-threaded CPHMD simulation to the new CUDA-CPHMD, and an improvement of over 150-fold when compared to the single-core algorithm (shown in Fig. 3a). For larger 29,000 atom systems, we see a speed improvement of over 1000-fold (shown in Fig. 3c). Since the neighboring-atom component does not scale linearly with system size, larger systems experience a greater calculation time penalty than smaller ones. Fortunately, simple changes, such as using nonbonded cutoffs, can mitigate such problems. For instance, a nonbonded cutoff of 14 Å sped up the large viral capsid simulation to 6.7 ns/day (a 270% speed increase versus the no cutoff case).

Accuracy of the new CUDA-CPHMD algorithm

Single-Residue Systems. Speed gains in implementing CPHMD are an important goal both for increasing the algorithm's applicability to a wider range of system sizes, and for its ability to converge on useful results more rapidly. Its accuracy, however, must not be compromised as we reconfigure the execution of the algorithm. In Figure 3e, we show that there is little difference between the original CPHMD and CUDA-CPHMD algorithms when calculating the force on λ. We maintain an average unsigned error (AUE) of less than 0.00017 kcal/mol in this force. We also note that 99.9% of the AUE between the two CPHMD methods is from the slight differences in Born radii calculated from the original and CUDA implementations of GBSW. Thus, we conclude that CUDA-CPHMD accurately reproduces the original algorithm's force on λ.

While CUDA-CPHMD may be able to produce the force on λ coordinates, we ran additional tests to see whether or not residue protonation states are also reproduced. Due to each residue's pH-dependent biasing potential, a single residue alone in solution presumably should find an optimal protonation state depending on the environmental pH. At pH environments below a residue's pK_a the residue should favor a protonated state $(\lambda_i \leq 0.1)$, and conversely a residue exposed to a pH above its pK_a should favor an unprotonated state

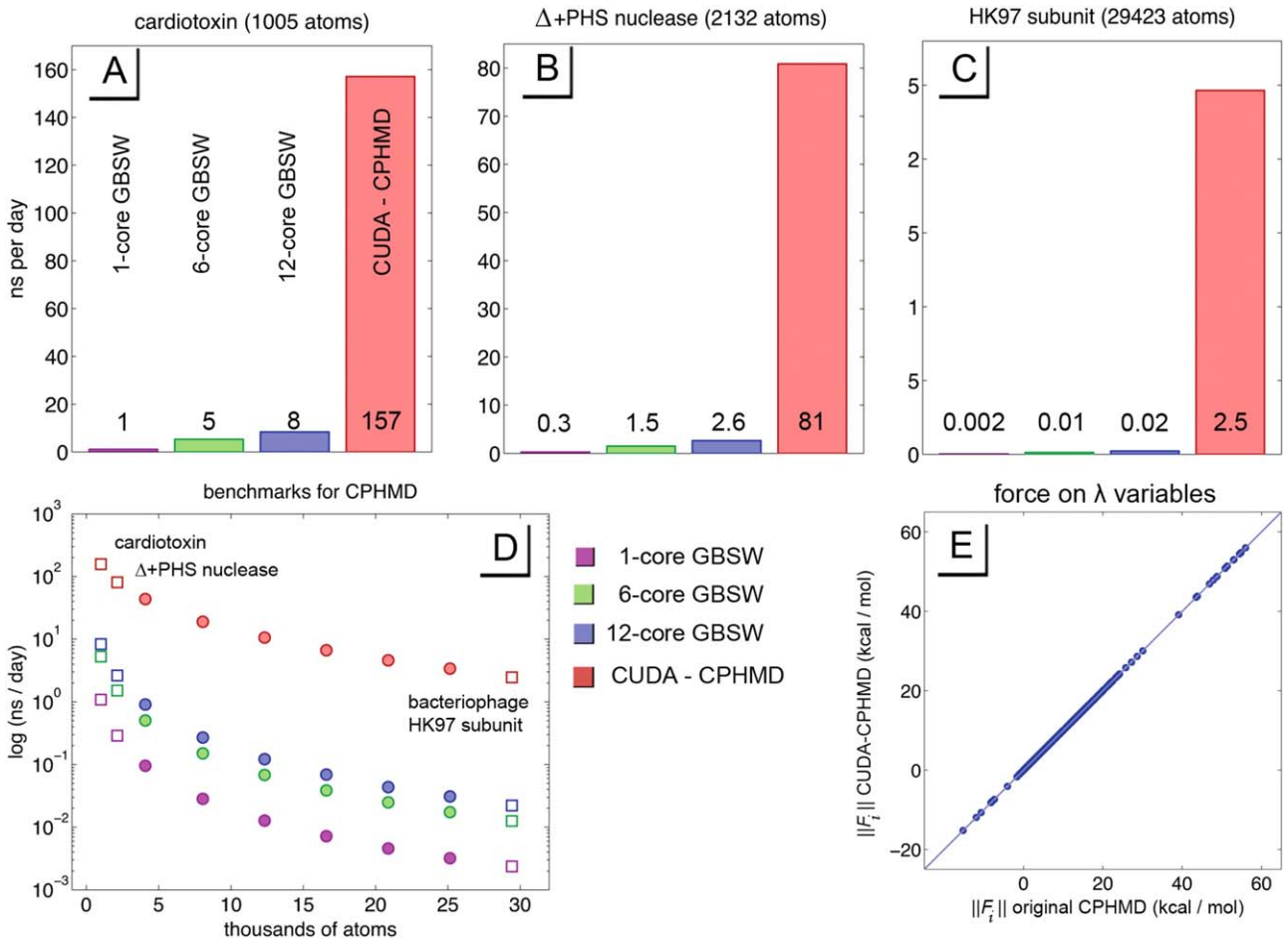

Figure 3. The benchmarks for the new CUDA-CPHMD algorithm. The individual systems tested were A) the naja atra snake cardiotoxin (PDB: 1CVO); B) the Δ+PHS hyperstable variant of staphylococcal nuclease (PDB: 3BDC); and C) the asymmetric subunit of the bacteriophage HK97 head capsule (PDB: 2FT1). As shown, the new algorithm is substantially faster than the original CPU algorithm by up to three orders of magnitude. In D) the same benchmarks from earlier are shown (squares) alongside subsystems from the seven proteins of the bacteriophage subunit (circles). Notice that the CUDA algorithm scales more linearly with system size than its CPU-based counterpart. E) compares the force on λ as calculated on all 595 λ coordinates from both CPHMD algorithms. There is less than a 0.00017 (kcal/mol) average unsigned error (AUE) between the two algorithms. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$(\lambda_i \geq 0.9)$. By calculating the fraction of protonated to unprotonated states of residues at various pH values and fitting the results to the Henderson–Hasselbalch equation of states, we expect the point of inflection to reproduce the pK_a of that residue.

We ran simulations of aspartic acid, glutamic acid, histidine, and lysine to calculate their protonation states, as shown in Figure 4. These residues were simulated using the same setup from the benchmarking section as NT simulations in an unbound volume, and CUDA-CPHMD was used both to model titration states and advance λ coordinates. The backbone atom ends were capped with the ACE and CT2 patches in CHARMM. Each dot in Figure 4 represents the average residue titration state from 200 ps of simulation time, and the residues ran at an average speed of 690 ns/day.

We find that without optimizing the simulations for speed, accuracy, or convergence of protonation states, that the pK_a values could be captured to within 0.5 pK units. Interestingly, all states reported a small, systematic overestimation of the pK_a, and the exact source of this discrepancy remains unclear. The CUDA-GBSW solvent model slightly overestimates solvation energy by an average of approximately 0.0003 kcal/mol. However, this overestimation of energy should bias deprotonation events to occur slightly more often, and thus lower the calculated pK_a. What is clear from these data, though, is that like its predecessor, the CUDA-CPHMD algorithm models the pH dependence of titration well for single residue systems. Next we explore multiresidue titration and the influence of protein conformation on pK_a values.

Multiple-Residue Systems. The end purpose for CPHMD is to enable the study of complex pH-coupled phenomena of biological systems, such as pH-dependent protein conformation and cooperative titration effects among neighboring residues. As such, we test the accuracy of the CUDA-CPHMD algorithm by its ability to recapitulate residue pK_a values from both experiments and previous replica exchange studies, as shown in Figure 5. We study nine model protein systems here:

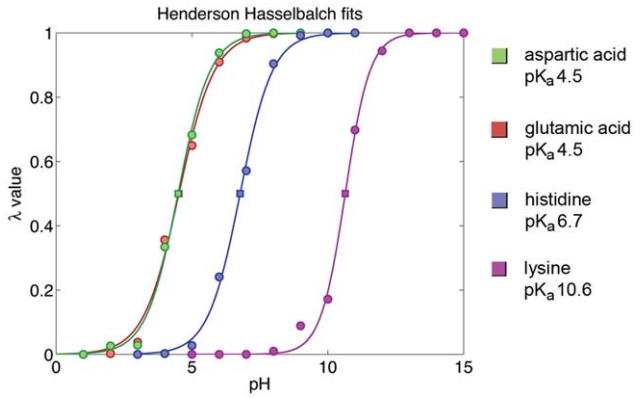

Figure 4. The pK_a calculations for four single residues: aspartic acid, glutamic acid, histidine, and lysine. The protonation state (dots) was calculated from the fraction of λ values in pure unprotonated and protonated states. The point of inflection (boxes) of Henderson-Hasselbalch equation fits (lines) indicates the calculated pK_a values. Even without optimizing for efficiency, convergence of data, or simulation parameters, we find the calculated pK_a values match those from the force field to within 0.5 pK units. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

barnase[2,47,48] (PDB code 1A2P); the serine protease inhibitor CI-2 from barley seeds[47,49] (PDB code 2CI2); the hyperstable variant of staphylococcal nuclease, Δ+PHS[3,16] (PDB code 3BDC); hen egg white lysozyme[47,50,51] (PDB code 1LSA); the N-terminal domain of ribosomal protein L9[47,52] (PDB code 1CQU); turkey ovomucoid[47,53,54] (PDB code 1OMU); ribonuclease A[47,55] (PDB code 7RSA); ribonuclease H from *Escherichia coli*[47,56,57] (PDB code 2RN2); and *Bacillus circulans* xylanase[47,58,59] (PDB code 1BCX). Each protein was simulated in 11 pH windows from the pH −1 to the pH 9. Within each window, the proteins were simulated for 80 ps in 10 independent trajectories, which resulted in a total of 4.4 ns of simulation time per structure. All titrating residues were allowed to change protonation state using the new CUDA-CPHMD algorithm (Figs. 5a, 5d, and 5e) and the original CPHMD algorithm (Figs. 5b and 5d); and salt concentrations were added using concentrations that corresponded to the experiments.[16,47] The simulations were run using the CHARMM22 force field[43,44] with the Langevin integrator[45] with an integration timestep of 2 fs. These were NT (constant particle number and temperature) simulations each in an unbounded volume at a temperature of 298 K using a Langevin heat bath. Atomic radii were optimized through work by Chen et al.[46] Similarly to the single-residue simulations, pK_a values were calculated by fitting the Henderson–Hasselbalch equation of states to the average protonation state λ of each titrating residue. Again,

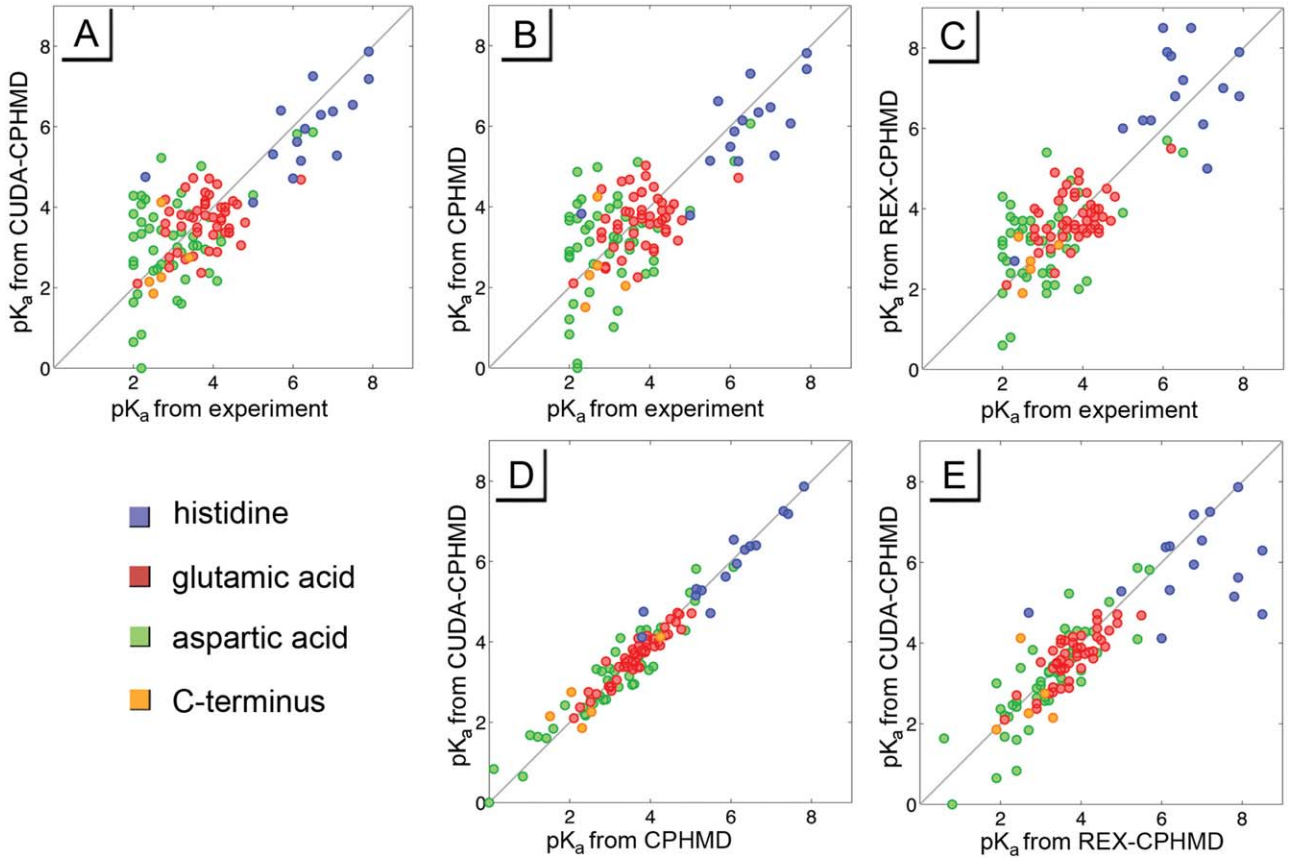

Figure 5. The pK_a calculations for all histidine (blue), glutamic acid (red), aspartic acid (green), and titrating C-terminus (orange) residues in all nine of the test proteins. Each dot corresponds to a pK_a value resulting from fitting the Henderson-Hasselbalch equation to the fraction of λ values in pure unprotonated states. We present comparisons between pK_a values from A) CUDA-CPHMD and experiment; B) CPHMD and experiment; C) CPHMD with replica exchange (REX-CPHMD) and experiment; D) CUDA-CPHMD and CPHMD; and E) CUDA-CPHMD and REX-CPHMD. Even without optimizing the simulations to accommodate various titration equilibria for each protein, the CUDA-CPHMD algorithm successfully recapitulates experimental pK_a values to within 0.79 pK units of AUE. The experimental and REX-CPHMD results are from Ref. [47] and papers cited therein. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the point of inflection of the fit corresponds to the pK_a value of that residue. We report these values in Figure 5.

The AUE for all residues using CUDA-CPHMD was 0.79 pK units, which compares favorably to the AUE of 0.97 pK units using the null approximation (all pK_a values correspond to their reference values). This was the same 0.79 pK units of AUE that the original algorithm achieved, which further supports CUDA-CPHMD's accurately representing its CPU counterpart. Interestingly, while the average accuracy of CUDA-CPHMD and CPHMD were less than the 0.75 pK units of AUE achieved using the replica exchange methods from earlier studies, the nonreplica-exchange pK_a calculations had a smaller standard deviation of error and fewer outlying predictions.[16,47] Additional accuracy should be possible by coupling CUDA-CPHMD with the enhanced sampling of replica exchange in temperature or pH.[16,47] This result holds great promise in establishing dynamic titration as a common feature of protein simulations.

Conclusions

In this study, we present a significantly faster version of the CPHMD algorithm adapted for parallel processing in the CHARMM-OpenMM interface. While algorithmically the new CUDA-CPHMD algorithm represents little change over its predecessor, the speed improvements are so great that previously unreasonable simulations are now straightforward to perform. For instance, what may have been a year-long simulation of the HK97 head capsule can now be performed in about 160 min. With this newfound speed is an opportunity to fine-tune the CPHMD titration model for a variety of protein systems, and to explore the impact of pH environments on side-chain dynamics both at the microsecond timescale and with all-atom detail.

Similarly to GBSW, the CPHMD model carries with it over a decade of research and parameterization.[26,47,52] One model of particular interest is pH replica exchange (REX),[60] which has been shown to predict pK_a values of protein structures within single nanoseconds of simulation time.[60,61] Coupled with the improved speed of CPHMD, adapting REX will enable a useful and rapid method for characterizing the chemical environment of protein interiors.

Keywords: implicit solvation · solvation · solvent model · compute unified device architecture · parallelization

How to cite this article: E. J. Arthur, C. L. Brooks III. *J. Comput. Chem.* **2016**, *37*, 2171–2180. DOI: 10.1002/jcc.24435

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

[1] J. E. Nielsen, J. A. McCammon, *Protein Sci.* **2003**, *12*, 1894.
[2] D. Sali, M. Bycroft, A. R. Fersht, *Nature* **1988**, *335*, 740.
[3] B. Cannon, D. Isom, A. Robinson, J. Seedorff, B. Garcia-Moreno, *Biophys. J.* **2007**, 403A.
[4] G. Rabbani, E. Ahmad, N. Zaidi, S. Fatima, R. Khan, *Cell Biochem. Biophys.* **2012**, *62*, 487.
[5] G. R. Wagner, R. M. Payne, *J. Biol. Chem.* **2013**, *288*, 29036.
[6] L. A. Baker, I. N. Watt, M. J. Runswick, J. E. Walker, J. L. Rubinstein, *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 11675.
[7] B. D. Cain, R. D. Simoni, *J. Biol. Chem.* **1986**, *261*, 10043.
[8] V. K. Rastogi, M. E. Girvin, *Nature* **1999**, *402*, 263.
[9] A. B. Clippingdale, J. D. Wade, C. J. Barrow, *J. Pept. Sci.* **2001**, *7*, 227.
[10] C. M. Dobson, *Nature* **2003**, *426*, 884.
[11] L. G. Cuello, D. M. Cortes, V. Jogini, A. Somporpisut, E. Perozo, *FEBS Lett.* **2010**, *584*, 1126.
[12] E. E. Howell, J. E. Villafranca, M. S. Warren, S. J. Oatley, J. Kraut, *Science* **1986**, *231*, 1123.
[13] B. Aguilar, R. Anandakrishnan, J. Z. Ruscio, A. V. Onufriev, *Biophys. J.* **2010**, *98*, 872.
[14] K. Lindorff-Larsen, S. Piana, R. O. Dror, D. E. Shaw, *Science* **2011**, *334*, 517.
[15] L. S. Ahlstrom, S. M. Law, A. Dickson, C. L. Brooks, III, *J. Mol. Biol.* **2015**, *427*, 1670.
[16] E. J. Arthur, J. D. Yesselman, C. L. Brooks, III, *Proteins* **2011**, *79*, 3276.
[17] D. Bashford, D. A. Case, C. Dalvit, L. Tennant, P. E. Wright, *Biochemistry* **1993**, *32*, 8045.
[18] J. E. Mertz, B. M. Pettitt, *Int. J. Supercomput. Appl. High Performance Comput.* **1994**, *8*, 47.
[19] Y. Y. Sham, Z. T. Chu, A. Warshel, *J. Phys. Chem. B* **1997**, *101*, 4458.
[20] A. M. Baptista, P. J. Martel, S. B. Petersen, *Proteins* **1997**, *27*, 523.
[21] A. M. Baptista, V. H. Teixeira, C. M. Soares, *J. Chem. Phys.* **2002**, *117*, 4184.
[22] J. T. Mongan, D. A. Case, J. A. McCammon, *Abstr. Pap. Am. Chem. Soc.* **2005**, *229*, U768.
[23] X. J. Kong, C. L. Brooks, III, *J. Chem. Phys.* **1996**, *105*, 2414.
[24] M. S. Lee, F. R. Salsbury, C. L. Brooks, III, *J. Chem. Phys.* **2002**, *116*, 10606.
[25] W. Im, M. S. Lee, C. L. Brooks, III, *J. Comput. Chem.* **2003**, *24*, 1691.
[26] J. Khandogin, C. L. Brooks, *III. Biophys. J.* **2005**, *89*, 141.
[27] X. Zeng, S. Mukhopadhyay, C. L. Brooks, *III. Proc. Natl. Acad. Sci.* **2015**, *112*, 2034.
[28] E. R. May, K. Arora, C. L. I. Brooks, *J. Am. Chem. Soc.* **2014**, *136*, 3097.
[29] E. J. Arthur, C. L. Brooks, III, *J. Comput. Chem.* **2016**, *37*, 927.
[30] B. R. Brooks, C. L. Brooks, III, 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.
[31] 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.
[32] 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.
[33] B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl, *J. Chem. Theory Comput.* **2008**, *4*, 435.
[34] 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.
[35] W. C. Still, A. Tempczyk, R. C. Hawley, T. Hendrickson, *J. Am Chem. Soc.* **1990**, *112*, 6127.
[36] J. Srinivasan, M. W. Trevathan, P. Beroza, D. A. Case, *Theor. Chem. Acc.* **1999**, *101*, 426.
[37] S. Donnini, F. Tegeler, G. Groenhof, H. Grubmüller, *J. Chem. Theory Comput.* **2011**, *7*, 1962.
[38] M. S. Lee, F. R. Salsbury, C. L. Brooks, III, *Proteins* **2004**, *56*, 738.
[39] G. E. Uhlenbeck, L. S. Ornstein, *Phys. Rev.* **1930**, *36*,
[40] A. K. Singhal, K. Y. Chien, W. G. Wu, G. S. Rule, *Biochemistry* **1993**, *32*, 8036.
[41] C. A. Castañeda, C. A. Fitch, A. Majumdar, V. Khangulov, J. L. Schlessman, B. E. García-Moreno, *Proteins* **2009**, *77*, 570.
[42] L. Gan, J. A. Speir, J. F. Conway, G. Lander, N. Cheng, B. A. Firek, R. W. Hendrix, R. L. Duda, L. Liljas, J. E. Johnson, *Structure* **2006**, *14*, 1655.
[43] A. D. MacKerell, D. Bashford, M. Bellott, R. L. Dunbrack, J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, 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, 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.
[44] A. D. MacKerell, M. Feig, C. L. Brooks, III, *J. Comput. Chem.* **2004**, *25*, 1400.
[45] L. Verlet, *Phys. Rev.* **1967**, *159*, 98.
[46] J. H. Chen, W. P. Im, C. L. Brooks, III, *J. Am. Chem. Soc.* **2006**, *128*, 3728.
[47] J. Khandogin, C. L. Brooks, III, *Biochemistry* **2006**, *45*, 9363.
[48] M. Oliveberg, V. L. Arcus, A. R. Fersht, *Biochemistry* **1995**, *34*, 9424.
[49] Y. J. Tan, M. Oliveberg, B. Davis, A. R. Fersht, *J. Mol. Biol.* **1995**, *254*, 980.
[50] T. Takahashi, H. Nakamura, A. Wada, *Biopolymers* **1992**, *32*, 897.
[51] K. Bartik, C. Redfield, C. M. Dobson, *Biophys. J.* **1994**, *66*, 1180.
[52] B. Kuhlman, D. L. Luisi, P. Young, D. P. Raleigh, *Biochemistry* **1999**, *38*, 4896.
[53] W. Schaller, A. D. Robertson, *Biochemistry* **1995**, *34*, 4714.
[54] W. R. Forsyth, M. K. Gilson, J. Antosiewicz, O. R. Jaren, A. D. Robertson, *Biochemistry* **1998**, *37*, 8643.
[55] W. R. Baker, A. Kintanar, *Arch. Biochem. Biophys.* **1996**, *327*, 189.
[56] Y. Oda, M. Yoshida, S. Kanaya, *J. Biol. Chem.* **1993**, *268*, 88.
[57] Y. Oda, T. Yamazaki, K. Nagayama, S. Kanaya, Y. Kuroda, H. Nakamura, *Biochemistry* **1994**, *33*, 5275.
[58] L. A. Plesniak, G. P. Connelly, L. P. McIntosh, W. W. Wakarchuk, *Protein Sci.* **1996**, *5*, 2319.
[59] M. D. Joshi, A. Hedberg, L. P. McIntosh, *Protein Sci.* **1997**, *6*, 2667.
[60] D. Sabri Dashti, Y. Meng, A. E. Roitberg, *J. Phys. Chem. B* **2012**, *116*, 8805.
[61] H Nymeyer, S. Gnanakaran, A. E. Garcìa, In *Methods in Enzymology*; B. Ludwig, L. J. Michael, Eds.; Academic Press, **2004**, pp. 119–149.

Received: 29 April 2016
Revised: 2 June 2016
Accepted: 3 June 2016
Published online on 12 July 2016