JCTC Journal of Chemical Theory and Computation

Automatic Control of Solvent Density in Grand Canonical Ensemble Monte Carlo Simulations

Joshua A. Speidel,[†] Jason R. Banfelder,[†,‡] and Mihaly Mezei*[,§]

Department of Physiology and Biophysics and HRH Prince Alwaleed Bin Talal Bin Abdulaziz Alsaud Institute for Computational Biomedicine, Weill Cornell Medical College, New York, New York 10021, and Department of Molecular Physiology and Biophysics, Box 1218, Mount Sinai School of Medicine, NYU, One Gustave L. Levy Place, New York, New York 10029

Received February 1, 2006

Abstract: We present automated methods for determining the value of Adams' *B* parameter corresponding to a target solvent density in grand canonical ensemble Monte Carlo simulations. The method found to work best employs a proportional-integral control equation commonly used in industrial process control applications. We show here that simulations employing this method rapidly converge to the desired target density. We further show that this method is robust over a wide range of system sizes. This advance reduces the overall CPU time and user effort in determining the equilibrium excess chemical potential in these systems.

Introduction

Simulations in the grand canonical ensemble have two unique and attractive features: they can be used to identify the chemical potential without additional costly free-energy simulations, and the combination of insertions and deletions results in large molecular displacements during the simulation that are not possible in any of the closed ensembles at condensed phase densities. These large displacements are particularly useful for solvating isolated pockets, as exist in most proteins. This simulation method can also be used to solvate lipid membranes, allowing penetration of water molecules deep into the interior of the lipid far more rapidly than with molecular dynamics.[1]

Systems solvated this way can serve as an initial configuration of a molecular dynamics run.[2] Alternatively, the simulation can be extended, and solvation sites can be deduced from it using, e.g., the generic site approach.[3] It has also been demonstrated that potential of mean force calculations benefit from the GCE framework: the changes in the distance between bulky solutes that these simulations require can be enabled through the removal or insertion of intervening solvents, rather than waiting for them to diffuse out/into the region between the solutes.[3]

A limitation of this simulation method is that insertions and deletions are currently feasible only for small, neutral molecules such as water. This is because the probability of accepting a random insertion of a bulky or charged molecule in a solvated system is very low. The insertion of water molecules has only become practical with the introduction of cavity biased sampling. A second limitation of this method is that when simulating the solvation of a system, the chemical potential yielding the target density is initially unknown. Therefore, prior to simulating at a target density, a tuning phase is necessary to identify the chemical potential parameter that yields the target density. In practice, this required several runs with the chemical potential parameter adjusted each time based on the results of previous run(s). Such manual interventions not only consume human time but also in general lengthen the overall time of the simulation. This article presents three procedures to perform this tuning without user intervention and compares their performances for systems of varying sizes and compositions.

* Corresponding author phone: (212)241-2186; fax: (212)860-3369; e-mail: Mihaly.Mezei@mssm.edu.

[†] Department of Physiology and Biophysics, Weill Cornell Medical College.

[‡] HRH Prince Alwaleed Bin Talal Bin Abdulaziz Alsaud Institute for Computational Biomedicine, Weill Cornell Medical College.

[§] Department of Molecular Physiology and Biophysics, Mount Sinai School of Medicine, NYU.

10.1021/ct0600363 CCC: $33.50
Published on Web 07/01/2006

Background

Monte Carlo simulations in the Grand Canonical (T,V,μ) Ensemble (GCE) are conveniently performed with the introduction of the parameter B which is related to the excess chemical potential μ' as

$$\mu' = kTB - kT\ln\langle N\rangle \tag{1}$$

where $\langle N\rangle$ is the average number of particles,[4] k is the Boltzmann constant, and T is the absolute temperature. Note that B depends not only on the excess chemical potential but also on the system size and composition as well. In addition to the conventional translations and rotations, simulations in the GCE require periodic insertions and deletions of molecules. The Cavity Biased variant (CB/GCE)[5,6] attempts insertion of a molecule only if a cavity of an appropriate radius is found and accepts the insertion with probability

$$P_i = \min\{1, P_N^{\mathrm{cav}} \exp[B + (E(r_{N+1}) - E(r_N))/kT]/(N+1)\} \tag{2}$$

where $E(r_N)$ is the potential energy of the system of N particles at configuration r_N, and P_N^{cav} is the probability of finding a cavity of a specific size. To maintain microscopic reversibility, the probability of a deletion of a particle is given by

$$P_d = \min\{1, N \exp[-B + (E(r_N) - E(r_{N-1}))/kT]/P_{N-1}^{\mathrm{cav}}\} \tag{3}$$

The robustness of the CB/GCE technique was demonstrated by its robustness in modeling solvent molecules in crystal hydrates and protein active sites.[3,5,7,8] In these simulations, the value of B directly affects the probability of a successful insertion or deletion. An automated method to identify B must correctly change B when the calculated density differs from the target density. At a given temperature, there is a natural fluctuation of the density that will occur for any given chemical potential. Therefore, we are primarily concerned with identifying the correct B parameter such that the mean density equals the target density over an equilibrated portion of the simulation.

Methods

Iterative Tuning Based on Fluctuations. Fluctuation (F) in the number of molecules is related to the mean number of molecules and the B parameter through eq 4:[9]

$$\Delta\langle N\rangle/\Delta B \approx F_{N,B} = \langle N^2\rangle - \langle N\rangle^2 \tag{4}$$

B is tuned in an iterative process that assumes that the relation of eq 4 is constant to a reasonable extent over a finite range of $\langle N\rangle$'s. Initially, $\Delta\langle N\rangle/\Delta B$ is either calculated or estimated experimentally from the isothermal compressibility of the pure liquid at the target density[4] ($F_{N,B}^{\mathrm{initial}}$). Each iteration, i, first simulates the CB/GCE system for X^e_{MC} steps to allow equilibration with the newly chosen B value, followed by X^s_{MC} steps to gather statistics for $\langle N_i\rangle$. The sensitivity coefficient $S_{N,B}^i$ is calculated as a linear combination of the pure liquid value and the value calculated from $\langle N^2\rangle - \langle N\rangle^2$ at the end of iteration i, using the whole run:

$$S_{N,B}^i = (F_{N,B}^{\mathrm{initial}} + i{*}F_{N,B}^i)/(i+1) \tag{5}$$

The change in B is determined by eq 6:

$$\Delta B = (N_{\mathrm{target}} - \langle N_i\rangle)/(S_{N,B}^i) \tag{6}$$

The maximum $|\Delta B|$ value is limited by default to 1.0 to further dampen oscillations in B. The current implementation gathers statistics for the fluctuation cumulatively over the whole simulation. The use of the fluctuation in $\langle N\rangle$ calculated separately in each iteration is precluded by the slow convergence of fluctuations, i.e., the simulation would not converge in X^s_{MC}, steps unless it was very long.

Iterative Tuning with Empirical Estimates of the Sensitivity Coefficient. In this method the sensitivity coefficient is a scaling factor that incorporates the effect of a change in B on the mean number of particles (eq 7).

$$S_{N.B}^i = \sum_{j=0}^{i} (\langle N_j\rangle - \langle N_{j-1}\rangle)/(B_j - B_{j-1}) \tag{7}$$

The change in B is then determined according to eq 8:

$$\Delta B = (N_{\mathrm{target}} - \langle N_i\rangle)/S_{N,B}^i \tag{8}$$

As with the fluctuation method, the $|\Delta B|$ is limited to the default value of 1.0. Also, there is a filter whereby if in the previous iteration the change in $\langle N\rangle$ is of opposite sign of the change in B, in the next iteration B is unchanged. Such an occurrence clearly indicates inadequate equilibration and/or statistics because by definition as B increases, so should $\langle N\rangle$. This filter implicitly increases X^e_{MC} in these instances allowing the system more time to adjust to the new value of B. This has the benefit of simplifying the choice of X^e_{MC} and X^s_{MC} for iteration lengths that are providing adequate statistics for $S_{N,B}^i$ but allow for frequent enough changes to reach convergence as fast as possible.

Tuning Using Process Control Principles. The canonical proportional-integral-derivative control equation (PID) commonly used in engineering control applications is eq 9[10]

$$\mathrm{MV}(t) = K_C\epsilon(t) + \frac{K_C}{\tau_I}\int_0^t \epsilon(t)\mathrm{d}t + K_c\tau_D\frac{\mathrm{d}\epsilon}{\mathrm{d}t} + c_s \tag{9}$$

where MV is the manipulated variable, K_C is the proportional gain, τ_I is the integral time, τ_D is the derivative time, c_s is the controller bias, and ϵ is the current deviation of the process variable from its target. It has been shown that the dynamic stability of a controlled system can be sensitive to the selection of the derivative time. Therefore, the derivative term is frequently omitted unless empirically shown to be necessary.[11] When applying this equation to our simulation systems, the value we seek to control is the density of bulk water. In eq 9, ϵ is the deviation away from the desired bulk density. To achieve the target density, the manipulated variable in our system is B, whose effect is described in the Introduction.

We further implement the differential form of the control equation:

$$\frac{dB}{dt} = K_C \frac{d\epsilon}{dt} + \frac{K_C}{\tau_I}\epsilon \tag{10}$$

This transformation solves the so-called 'integral windup' problem, whereby an historic period of large deviation dominates the integral term.[11] It has the added benefit of eliminating the controller bias as another tuning parameter.

The process control equation has an implicit time dependence that is not present in Monte Carlo simulations. To emphasize this, we rewrite the control equation once more in finite difference form in terms of the simulation step number, i:

$$B_{i+1} = B_i + K_c(\epsilon_i - \epsilon_{i-1}) + \frac{K_C}{\tau_I}\epsilon_i \tag{11}$$

Unlike the first two methods, this technique changes the B parameter at every insertion/deletion attempt.

Calculations

All three methods were implemented into the Monte Carlo program MMC.[12]

Grand Canonical Ensemble Simulation Systems. For each of the three tuning methods, four simulations of differing size and components were monitored for their ability to identify the value of Adams' B parameter that yields the target density.

All our systems targeted equilibrium with bulk water at 310 K, 0.997 g/mL. The first three simulations were small, medium, and large pure TIP3P water simulations. With the cubic edge length of the boxes equal to 16.48, 34.99, and 75.26 Å, respectively, the simulations targeted 142, 1420, and 14 200 waters, respectively. To mimic systems with a solute, one water was treated as the solute, and the density was monitored in the region outside a cubic cell centered at the center-of-mass of this water. The position of this water molecule was kept fixed throughout the simulation. The edge of the cube, enclosing the waters perturbed by the solute, was set to 6 Å.

The fourth simulation included a trypsin protein with a bound benzamidine.[13] The protein simulation volume had an edge length of 75.26 Å. The center of mass of the protein was placed at the center of the simulation unit cell. The density was monitored in the region outside a rectangular volume centered at the origin with minimum x, y, z and maximum x, y, z coordinates of −26, −24, −25, 24, 26, and 24, respectively. At the target density, approximately 10 124 water molecules fill this monitored volume.

All simulations used scaled force biased sampling[14] where the level of biasing was scaled down in the vicinity of the solute. The water potential used was TIP3P,[15] and the solute−solvent interactions were described by the CHARMM force field.[16] The solvent−solvent potential was cut off at 12 Å. Fifty water molecules were randomly placed in the simulation cell at the start of the simulation. The solute was kept fixed in all simulations. At each MC step a randomly selected solvent was translated with a maximum step size of 0.275 Å and rotated with a maximum angle of 60°. Initially the B parameter was set to 1.0.

The small water simulations were run for 5M steps. The medium and large water simulations were run for 25M and 50M steps, respectively. The large protein simulation was run for 50M steps.

Sensitivity Coefficient Based Tuning Simulations. As the system size increases, generally more time is needed to average the properties of the system and to reach equilibrium after a change in the control parameter. For these simulations, the small water simulation used 10K steps of equilibration followed by 10K steps to gather statistics. The medium water simulation (1420 water molecule target) used 50K steps of equilibration followed by 50K steps to gather statistics. The large water and protein simulations used 100K steps for equilibrium and 100K steps to gather statistics. The density and B parameter were reported at 10 000 equally spaced steps in each simulation, i.e., every 200 steps for the small water simulation.

Fluctuation Based Tuning Simulations. These simulations are the same as those of the sensitivity coefficient based simulations, except that the initial configuration has a number of waters approximately equal to the target number of waters. This was done because the accumulated fluctuation generated by starting with 50 water molecules was thought to prevent the B parameter from correctly responding to changes in $\langle N \rangle$. In an attempt to minimize that influence, we therefore performed an initial short CB/GCE simulation with B set to 10.0 until the desired number of waters was reached. The system that resulted from that simulation was used as the starting structure of the fluctuation based tuning simulations.

Proportional Integral Control Based Tuning Simulations. Simulations were run for the same lengths as the fluctuation and sensitivity coefficient based simulations. The B parameter in these simulations is changed at each step of the simulation. The two controller tuning parameters (K_C and τ_I) were determined to be −112 and 13,000 respectively, by following the open loop protocol of Ziegler and Nichols for the large protein−ligand system.[17] This method of tuning the control parameters is based on the response of the system to a step change in the manipulated variable (B, in our case). However, instead of considering the steady-state result (i.e., the equilibrated value of the density) in response to the step change, it considers the delay of the response of the controlled variable (the so-called 'dead-time' in the vernacular of control engineering) and the initial rate of the response. This method has the advantage that it can be applied even to nonself-regulating systems (i.e., those systems that have an unbound response to the step change) because it only depends on the dynamic character of the *initial* response to the perturbation.

Once the dead-time and initial response rate for a process in known, the Ziegler-Nichols method provides rules for determining the PID controller constants that appear in eq 9. Evaluation of the dead-time and initial response rate is somewhat subjective as it involves an assessment of how long the system took to 'appreciably respond' after the perturbation was applied. However, as long as the dead-time/initial response rate pairs are self-consistent, the Ziegler-Nichols tuning method will yield acceptable (and usually indistinguishable) empirical performance.

Table 1. Testing Convergence

	A. convergence to target density[a]			B. convergence in B[b]		
	fluctuation	sensitivity coefficient	PIC	fluctuation	sensitivity coefficient	PIC
small water	+	+	+	/	×	-4.46 ± 0.35
medium water	−	×	+	×	×	-2.42 ± 0.14
large water	−	+	+	×	-0.23 ± 1.07	-0.18 ± 0.08
protein water	−	×	+	×	×	-0.22 ± 0.15

[a] An '×' indicates that the density was not normal for either the last 50 or 25% of the simulation. A '−' indicates that the density was normal but was not converged for the last 50 or 25% of the simulation. A '+' indicates that the density converged to the target density by the criteria described in the text. [b] A '/' indicates that B was not normally distributed in one of the block average segments analyzed. An '×' indicates that the simulation did not converge to the target density. A '−' indicates that the simulation was converged to the target density but not converged to a single B value. Simulations that converged to the target and a single B have the value of $B \pm$ the standard deviation in the table.

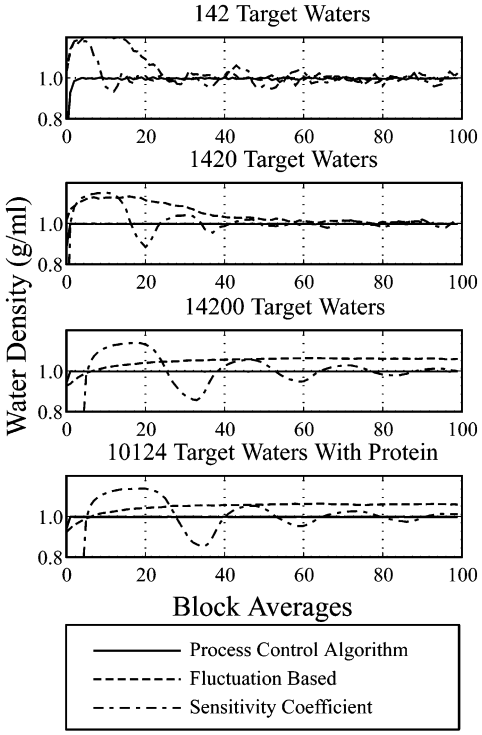

Figure 1. Density block average. The block average of the density is plotted for each simulation. The solid line is the PIC method, the dashed line is the fluctuation based method, and the dash−dot line is the sensitivity coefficient method. Each simulation is divided into 100 equally sized blocks. The block averages for the small, medium, and large water simulation were 50 000 steps, 250 000, and 500 000 steps per block average. The large protein and water simulation used 500 000 steps per block average.

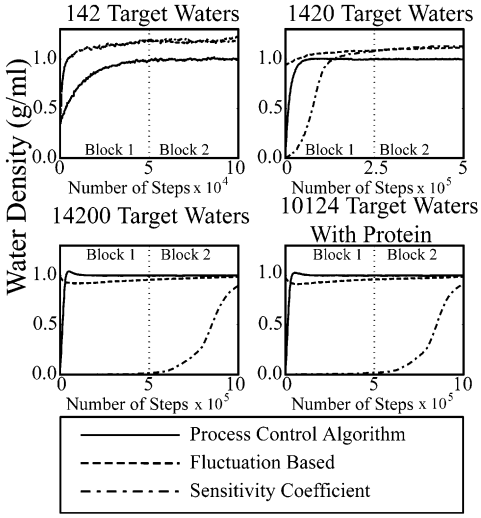

Figure 2. Early system density. The instantaneous density during the first 2% of each simulation is shown. The labels 'Block 1' and 'Block 2' indicate the portion of the plot that would be averaged to produce the block average densities used in Figure 1.

In engineering control scenarios, proportional-integral controller (PIC) performance is typically robust over a wide range of operating conditions.[11] By analogy, the parameters determined here ($K_c = -112$ and $\tau_I = 13\,000$) are expected to be effective in a variety of simulation systems without modification. Although the range of systems for which a given set of parameters is effective is difficult to predict, we note that they perform quite well on all systems tested here (which vary over 2 orders of magnitude in size). Additionally, these parameters produce the targeted density within 1M steps when applied in other simulation systems (1SDO, 80 Å by 80 Å by 80 Å system;[18] 1TY6, 100 Å by 80 Å by 70 Å[19]) with equal effect. In systems that are many orders of magnitude larger (or otherwise different in their dynamic character) a retuning of the parameters may give enhanced performance. However, systems of such size are outside of the purview of most GCE simulation methodologies at the present time.

Grand Canonical Ensemble Simulations at Fixed *B*. The last configuration of the PIC protein-water tuning simulation was used as the starting point of these simulations. The density initially was 0.997 g/mL in the region monitored in the previous simulations. All parameters detailed for the protein-water simulation above were used, with two exceptions, the tuning keyword was off, and the B parameter was fixed throughout the simulations. Simulations were allowed to evolve for 20M steps, and the density was reported every 2000 steps.

Results and Discussion

Achieving the Target Density. The effectiveness of the three tuning methods in approaching and maintaining the targeted density of water is shown in Figure 1.

The densities of the steps corresponding to the first two block averages are shown in Figure 2. The process control

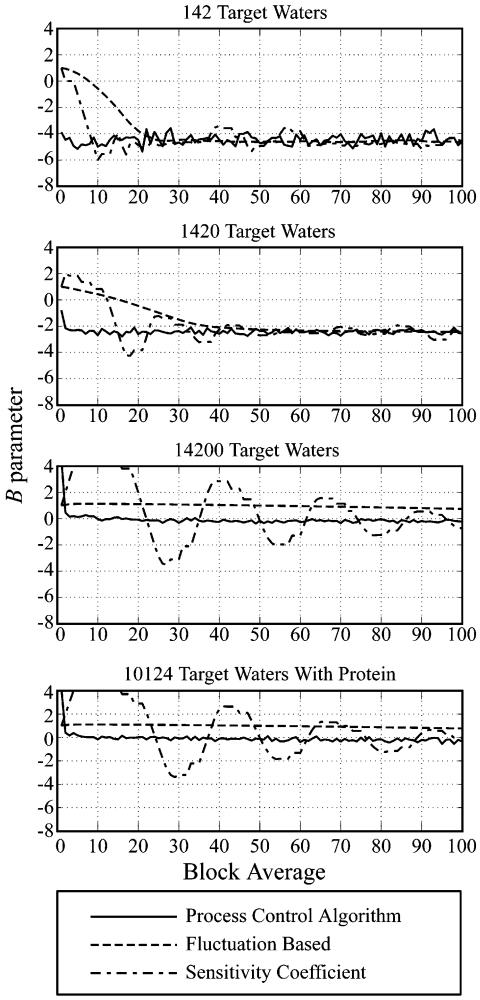

Figure 3. *B* parameter block average. The block average of the *B* parameter is plotted for each simulation. The solid line is the PIC method, the fluctuation based method is indicated by the dashed line, and the dash−dot line is the sensitivity coefficient method.

algorithm approaches the target density within these two blocks and maintains it throughout the simulation. In the small water simulation, the sensitivity coefficient algorithm and the fluctuation based algorithm have the exact same densities for the first 20K steps of the simulation due to the 10K of equilibration and 10K of statistical gathering time that they require. After that, they independently modify the *B* parameter and their densities diverge. Also, in the medium water, large water, and large water−protein simulations, the fluctuation based algorithm starts with a density near the target as noted in the computational methods section.

Generally, the standard deviation of the density in the PIC simulations decreases with increased targeted *N*. This is due to the discrete method of adding and subtracting water molecules. For example, in a simulation targeting 142 water molecules, a single water molecule addition increases the density by 0.007 g/mL, while for simulations with target *N* equal to 1420, a single water molecule addition increases the density by 0.002 g/mL.

Testing for Convergence. In equilibrated systems, the distribution of the density is expected to be close to normal. First, we apply the Lilliefors test for normality of the density distribution in the last 50 and 25% of the simulations. If the distribution is normal, we then apply the Student's *t*-test at 95% confidence with a mean of 0.997 and unknown variance. A simulation that passes all of these tests is considered converged to the target density. All methods converge to the target density in the small water simulation. Table 1A indicates that only the PIC method converges to the target density in all simulation systems.

Inspection of Figure 1 shows that the sensitivity coefficient method appears to be converging to the correct density in many of the simulations but has not achieved convergence by our criteria in the medium water and protein/water simulations. Figure 1 also shows that the fluctuation method uses the most steps to approach the target density in the small and medium simulations. In the large simulation with and without protein, the fluctuation method does not even reach the target density. This is possibly due to an overestimation of the fluctuation. As the target system gets larger, changes in $\langle N \rangle$ produce an increasing error of this estimate.

In all four simulation conditions, the PIC algorithm identifies and maintains the target density the soonest. This may be due to the method with which the PIC method adjusts the *B* parameter at each step with no inherent lag time for averaging or equilibration. The PIC algorithm in all four simulations reliably identifies the target density and maintains that state. Figure 2 shows the evolution of *B* for all simulations.

Identifying B. The ultimate goal of the protocol is to identify the correct *B* parameter. The convergence of the simulation to the target density is a necessary, but not sufficient, condition for correct determination of this parameter. We consider *B* converged when the simulation has passed two criteria for block averages 51−75 and 76−100. The first criteria is a Lilliefors test for normality in these sections individually. While normality is not an absolute requirement for convergence in *B*, this test determines whether the paired Student's *t*-test between the two segments of the simulations will be reliable. A 5% significance level is used in the Student's *t*-test. Cases where the target density is not converged are not considered. Table 1B shows that only the PIC method produces a converged *B* value for all simulation systems.

In all cases, the PIC simulations converged to the target density within 500K steps, while the *B* parameter required more steps to reach equilibrium. In all PIC simulations, the *B* parameter appears steady by the midpoint of the simulations. The target *N* of the simulation has some effect on this convergence, with larger target numbers requiring more steps to converge to a value in *B*. Further tests for convergence used the mean value of *B* from the block averages of the last 50% of the PIC simulation for the protein-water simulation system where the mean density was $0.9970 \pm 8.6 \times 10^{-5}$.

Testing the Identified B. To determine if we had identified the correct value of *B* in the large protein simulation, and that the fluctuations of *B* around its mean did not introduce a bias, we ran a series of simulations each with constant *B*. The starting system from these simulations used the final system from the PIC simulations and varied

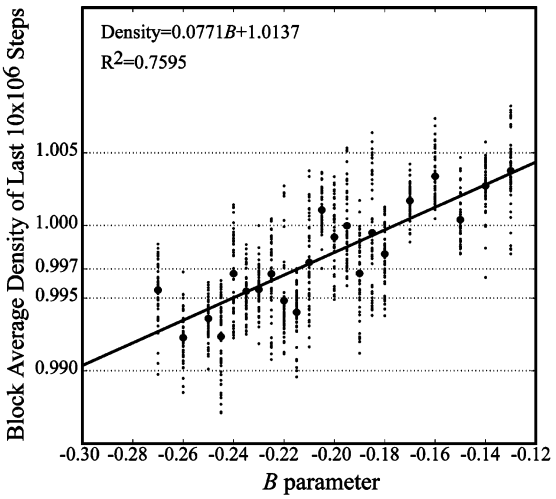

Figure 4. Fixed *B* parameter simulation results. The small black points show the block averages plotted for the last 50% (1×10^7 steps) of the simulations. Each block is the average of 200 000 steps. The larger black dots mark the mean of those block averages. The line shows the linear regression through those means.

B around the mean of the last 50% of the simulation. Each simulation ran for 20M steps, and data analysis was performed on 100 equally sized block averages. In Figure 4, these block averages are plotted against *B* for the last 10M steps (50 block averages). The larger black dot marks the mean of those block averages. The linear regression through those mean values yields a line producing an idealized value of *B* for this system. The value of *B* from the PIC simulations deviated from this ideal by 0.0046 and was well within the standard deviation of *B* ($\pm$0.1470) for the last 50% of the PIC simulation.

The R^2 correlation of the line though the means of the fixed *B* simulations indicates that there is a fair amount of uncertainty in the density at a constant *B*. This effect can be minimized through longer simulation times, increasing the statistical sampling of the ensemble of densities and *B* parameters.

Conclusions

We show that it is possible to tune the *B* parameter of the CB/GCE simulations efficiently and without user intervention to a good approximation of the actual value. Of the three methods tested, the PIC method performed best by far, and it is the recommended method. Also, the parameters of the PIC method established in this work are robust enough to be applicable to many diverse system sizes.

Tuning the B parameter without user intervention streamlines the simulation process and allows completion of a project in significantly shorter time. The time saving comes both in terms of CPU time used and elapsed time because tuning can be completed in fewer simulation steps than with manual tuning, and there is no need to interrupt the runs to manually adjust the *B* parameter.

Acknowledgment. This work was supported in part by NIH Grant P01 GM66531.

References

(1) Mezei, M. Efficient Monte Carlo sampling for long molecular chains using local moves, tested on a solvated lipid bilayer. *J. Chem. Phys.* **2003**, *118* (8), 3874−3879.

(2) Ebersole, B. J.; Visiers, I.; Weinstein, H.; Sealfon, S. C. Molecular basis of partial agonism: orientation of indoleamine ligands in the binding pocket of the human serotonin 5-HT2A receptor determines relative efficacy. *Mol. Pharmacol.* **2003**, *63* (1), 36−43.

(3) Marrone, T. J.; Resat, H.; Hodge, C. N.; Chang, C. H.; McCammon, J. A. Solvation studies of DMP323 and A76928 bound to HIV protease: analysis of water sites using grand canonical Monte Carlo simulations. *Protein Sci.* **1998**, *7* (3), 573−9.

(4) Adams, D. J. Grand canonical ensemble Monte Carlo for a Lennard-Jones fluid. *Mol. Phys.* **1975**, *29*, 307−311.

(5) Resat, H.; Mezei, M. Grand canonical ensemble Monte Carlo simulation of the dCpG/proflavine crystal hydrate. *Biophys. J.* **1996**, *71* (3), 1179−90.

(6) Mezei, M., Grand-canonical ensemble Monte Carlo study of dense liquid Lennard-Jones, soft spheres and water. *Mol. Phys.* **1987**, *61* (3), 565−582.

(7) Resat, H.; Mezei, M. Grand canonical Monte Carlo simulation of water positions in crystal hydrates. *J. Am. Chem. Soc.* **1994**, *116*, 7451−7452.

(8) Woo, H. J.; Dinner, A. R.; Roux, B. Grand canonical Monte Carlo simulations of water in protein environments. *J. Chem. Phys.* **2004**, *121* (13), 6392−400.

(9) McQuarrie, D. *Statistical Thermodynamics*; University Science Books: Mill Valley, CA 94941, 1973.

(10) Stephanopolous, G. *Chemical Process Control, an introduction to Theory and practice*; Prentice Hall: Englewood Cliff, NJ, 1984.

(11) St. Clair, D. *Controller Tuning and Control Loop Performance*, 2nd ed.; Straight Line Control: 1993.

(12) MMC is available at the URL http://inka.mssm.edu/~mezei/mmc (last accessed: 05/16/2006).

(13) Chamorro, J. A.; Cuesta-Seijo, J. A.; Garca-Granda, S. Pancreatic bovine trypsin native and inhibited with benzamidine from synchotron data. To be published **2006**; PDB ID: 1S0R.

(14) Mezei, M. Distance-scaled force biased Monte Carlo simulation for solutions containing a strongly interacting solute. *Mol. Simul.* **1991**, *5*, 405−408.

(15) Jorgensen, W.; Chandrashekar, J.; Madura, J.; Impey, R.; Klein, M. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* **1983**, *79* (2), 926−935.

(16) Brooks, B.; Bruccoleri, R.; Olafson, B.; States, D.; Swaminathan, W.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.* **1983**, *4* (2), 187−217.

(17) Zeigler, J. G.; Nichols, N. B. Optimum settings for automatic controllers. *Trans. ASME* **1942**, 759.

(18) Niv, M. Y., personal communication.

(19) Filizola, M., personal communication.

CT0600363