Algorithms **2009**, *2*, 215-226; doi:10.3390/a2010215

OPEN ACCESS
algorithms
ISSN 1999-4893
www.mdpi.com/journal/algorithms

Article

Automatic Determination of Stepsize Parameters in Monte Carlo Simulation Tested on a Bromodomain-Binding Octapeptide

Jason R. Banfelder [1], Joshua A. Speidel [1] and Mihaly Mezei [2,⋆]

[1] 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, USA; E-mails: jrb2004@med.cornell.edu; speidel@sainc.com
[2] Department of Structural and Chemical Biology, Mount Sinai School of Medicine, NYU, New York, NY 10029, USA

⋆ Author to whom correspondence should be addressed; E-mail: Mihaly.Mezei@mssm.edu.

Received: 30 October 2008; in revised form: 7 January 2009 / Accepted: 24 January 2009 / Published: 10 February 2009

Abstract: The proportional integral controller, commonly used in engineering applications for process control, has been implemented for the tuning of the stepsizes in Metropolis Monte Carlo simulations. Similarly to the recent application for tuning the chemical potential parameter in grand-canonical ensemble simulation, the process-control approach was found to work well for the problem of selecting the stepsize for each torsion angle that results in a targeted acceptance rate during the simulation of an octapeptide in aqueous environment.

Keywords: Monte Carlo, acceptance rate, stepsize, process control.

1. Introduction

The Metropolis Monte Carlo method [1] is the first computational procedure that was shown capable of sampling an ensemble of configurations suitable for statistical-mechanical treatment. While current simulation work in macromolecular systems generally employs the molecular dynamics approach with significant success, there are applications where Monte Carlo was found to be more efficient [2]. Also, the ease of restricting the sampling to a limited number of degrees of freedom makes Monte Carlo an attractive choice [3]. However, it is important to recognize that the Metropolis method actually encompasses a large family of methods [4] and thus there is still room for significant improvements and for 'niche' applications [5].

One of the factors determining the computational efficiency of Monte Carlo (MC) simulations is the proper selection of the range of changes from which trial moves will be sampled, commonly represented by the so-called stepsize parameters, e.g., the edge of the cube around a molecule within which the trial position must fall or the width of a torsion angle interval within which the trial angle should fall. Generally, the larger these parameters the larger conformational change is made in each step, but the smaller the acceptance probability will be. However, selection of stepsizes resulting in a targeted acceptance rate is a nontrivial process for system having several degrees of freedom of different types.

The present paper proposes an automated method that employs the proportional-integral controller (PIC) technique, well-known in engineering application. It has recently been shown to work extremely well in controlling the density during grand-canonical ensemble simulations [6].

In general, it is difficult to establish the optimal acceptance rate since the sampling efficiency is only known with any certainty after several long simulations. For systems with smooth free energy landscape, the accumulated experience in the field leads to target acceptance rates around 30% instead of the assumption of the early practitioners that 50% is the best [7, 8]. For systems with a rugged free energy landscape, however, the optimum shifts toward smaller acceptance rates since larger stepsizes may cross easier barriers separating separate free-energy basins [9].

The present paper leaves the question of establishing the optimal acceptance rates for further study. Instead, it provides an automated solution of the problem of determining the stepsize parameters that produce a targeted acceptance rate, e.g., 30%. The availability of such technique, however, could help in establishing the optimum acceptance rate in a different study.

It should also be emphasized that while optimizing the stepsize selection is likely to produce a significant improvement in the ease and efficiency of simulating biochemical systems with the Monte Carlo approach, other developments, such as the generalized ensemble approach (e.g., Ref. [10]) or the development of more complex moves may prove more important. However, such developments can also employ the stepsize tuning technique described in the present paper.

2. Background

The selection of stepsize parameters in MC simulations is generally performed by trial and error, based on shorter runs. Given that the simulation efficiency is not sensitive to small changes in these parameters, such a procedure is adequate for systems with a few types of degrees of freedom sampled, e.g., for neat liquids or solutions involving small solutes. For systems containing a large number of degrees of freedom whose stepsize parameters have to be tuned separately (e.g., a polypeptide with a large number of different types of torsion angles), the trial and error approach breaks down since it requires much longer trial runs and it is practically impossible to consider the effect of the change in the stepsize parameter of one type of degrees of freedom on the acceptance rate of a different type. As a result, such simulations are generally run with poorly tuned stepsize parameters.

A procedure for automated tuning of stepsize parameters has already been introduced [11]. In that work the simulation was partitioned into blocks where acceptance rate statistics are gathered on the basis of which the stepsize parameters are modified to move toward the target acceptance rate. That work also emphasized the important point that the ensemble generated during tuning will deviate from the Boltzmann distribution. Thus, unless strict adherence to the Boltzmann distribution is not important (e.g., for simulated annealing), tuning should *precede* data gathering, even though continuing the tuning may improve the sampling efficiency as the stepsize parameters adapt to new conformations of the system. This tuning algorithm has been implemented into the CHARMM program for simulation of macromolecules [12] and tested in detail on the alanine dipeptide in vacuum [8].

Our recent experience with the automatic tuning of the chemical potential parameter called B in grand-canonical ensemble simulations to a target density gave us the motivation to develop the tuning technique presented here. For the tuning of B, three algorithms were tested: (1) change B based on the average densities over two successive blocks; (2) change B based on the fluctuation of the number of molecules N calculated in successive blocks; and (3) change B continually as dictated by the PIC technique. While the first two techniques even failed to converge on larger systems the PIC technique always worked and outperformed the other two by several orders of magnitude. The reason for this is that in order for the block-average based methods to work the block averages should be converged enough (otherwise the change made moved B in the wrong direction), but that unduly lengthens the calculation. This result thus suggests that the tuning based on simulation blocks is significantly less efficient than tuning by the PIC technique.

3. Methods

The process control technique identifies a process variable whose value has to be kept around its target and a manipulated variable MV whose value the controller changes *continually* to stabilize the process variable. In our case the process variable is the acceptance rate a and the manipulated variable is the stepsize parameter Δ. The canonical proportional-integral derivative control equation (PID) is written as follows [13]:

$$MV(t) = K_C\epsilon(t) + \frac{K_C}{\tau_I}\int_0^t \epsilon(t)dt + K_C\tau_D\frac{d\epsilon}{dt} + c_s \tag{1}$$

where ϵ is the current deviation of the process variable from its target, t is the time, and the controller parameters K_C, τ_I, τ_D, c_s, are the proportional gain, integral time, derivative time, and controller bias, respectively. 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 [14].

Furthermore, we implemented the differential form of the control equation:

$$\frac{d\Delta}{dt} = K_C\frac{d\epsilon}{dt} + \frac{K_C}{\tau_I}\epsilon \tag{2}$$

This transformation solves the so-called ‘integral windup’ problem, whereby an historic period of large deviation dominates the integral term [14]. It has the added benefit of eliminating the controller bias as an additional tuning parameter.

In general, the controller tuning parameters K_C and τ_I are chosen to make the controller most sensitive to changes in the process variable without introducing undue fluctuations in it as a result of controller-induced changes. The general method of arriving at such controller tuning parameters K_C and τ_I is the open-loop protocol of Ziegler and Nichols [15]. Note, however, that the controller behavior is much less sensitive to the precise values of K_C and τ_I when compared to the sensitivity of the acceptance rate to the stepsize.

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

$$\Delta_{i+1} = \Delta_i + K_C(\epsilon_i - \epsilon_{i-1}) + \frac{K_C}{\tau_I}\epsilon_i \tag{3}$$

The calculation of ϵ_i, however, requires some additional thought. Since any single step is either accepted or rejected, the acceptance rate at step i has to represent an average over a certain stretch of the simulation. To avoid having to either limit the controller action to simulation blocks or to having to store the acceptance information for a large number of steps, we chose to define the (average) acceptance rate a_i at step i as a weighted average of the previous acceptance rates, with successively smaller weight given to more distant contributions. This can be achieved using

$$a_i = w_\mathrm{a} * a_{i-1} + (1 - w_\mathrm{a}) * M_i^{\mathrm{acc/rej}} = (1 - w_\mathrm{a}) \sum_{j=1}^{i} M_j^{\mathrm{acc/rej}} * w_\mathrm{a}^{i-j} + M_0^{\mathrm{acc/rej}} * w_\mathrm{a}^{i} \tag{4}$$

where $M_i^{\mathrm{acc/rej}}$ is unity if the move was accepted at step i and zero if not. The second equation in Eq. (4) converts the recursion formula into an explicit one. Each application of the recursion introduces higher powers of w_a, resulting in the sum over the number of steps involving different powers of w_a. In order to include contributions from a long enough stretch of earlier moves, w_a has to be close to one. As a result, this formula would give disproportionate weight to the first contribution unless it is zero (since everything else is multiplied by $(1 - w_\mathrm{a})$) and thus the deviation from the target acceptance rate would be underestimated for a significant portion of the run. For this reason, the recursion is always started from zero acceptance rate, even though using the target acceptance rate is usually considered the better choice.

For simulations requiring strictly Boltzmann-weighted ensembles the tuning phase will be followed by simulation with fixed stepsizes. Since the stepsizes fluctuate continually during tuning phase — both due to the controller action and, more importantly, to the changes in the conformation — the current value of each stepsize will have random deviation from the value that would, on the average, result in acceptance rate closest to the target. Therefore, instead of using the value at the time the tuning is stopped, it is better to choose a value that is averaged over the last stretch of the tuning run. We tested both using simple averages over a stretch that already reached the targeted acceptance rates and using a weighted average, analogous to Eq. (4). Using w_Δ as the weight the weighted average of the stepsizes Δ_j over i moves, $\langle\Delta_i\rangle$ is

$$\langle\Delta_i\rangle = \frac{1 - w_\Delta}{1 - w_\Delta^{i+1}} \sum_{j=0}^{i} \Delta_j w_\Delta^{i-j} \tag{5}$$

The normalization factor before the summation (based on the summation formula of the geometric series) is introduced to correct for the fact that the summation omits the term without a $(1-w_\Delta)$ factor. Denoting the weighted sum of Δ's with S_i, we have the following recursion to calculate it:

$$\sum_{j=0}^{i} \Delta_j w_\Delta^{i-j} = S_i = w_\Delta * S_{i-1} + \Delta_i\,. \tag{6}$$

Note that the weight factor $w_{\rm a}$ used for the stepsize averaging can be different from the w_Δ used for the acceptance rate calculation. Note again, that the closer w_Δ is to one, the longer stretch of the simulation will contribute significantly to the calculated stepsize average.

The effect of the weighting schemes described by Eqs. (4-6) can be quantified if we write w as $w = 1 - 1/n$. Since $(1 - 1/n)^n \simeq e^{-1}$, $w^N \simeq e^{-N/n} = 10^{-N/(n \ln 10)}$.

4. Calculations

The stepsize tuning was implemented into the MC program MMC [16] for the translation and rotation of non-solvent molecules and for the torsion angle sampling, both for regular torsion moves and for local torsion moves. Regular torsion changes a single torsion angle. In local torsion moves [17–19] a backbone angle change (called the 'driver torsion') is combined with complementary changes in the six successive torsion angles to keep the remaining part of the backbone unchanged. The choice of the stepsize only affects the range of the driver torsion — the rest of the torsion angles are determined by the geometrical constraints imposed by the fixed bond lengths and bond angles. Note, that there is no guarantee to find a solution after a driver torsion change (in that case, of course, the move attempt is rejected). The larger the driver torsion change, the higher is the probability that no solution will be found.

Tuning of the torsion stepsizes poses by far the most serious problem since the acceptance of a given torsion change depends strongly on the environment and relative topology of each torsion. Thus the test of the tuning method introduced will focus on tuning torsion angle stepsizes.

For each degree of freedom a separate controller was implemented. However, we used the same controller tuning parameters K_C and τ_I (see Eq. 3) and same averaging weight $w_{\rm a}$ (see Eq. 4) for all simple torsions, and a different set for all local torsions.

The system chosen for the test was the octapeptide (RHK(Ac-K)LMFK), the segment of the protein p53 that binds to PCAF Bromodomain. It was simulated with a ca 6 Å layer of water, stabilized with the Primary Hydration Shell technique [20], the MC variant of the original version developed for MD simulations [21]. The water potential used was TIP3P [22] and both the solute-solute and the solute-solvent interactions were described by the CHARMM-27 force field [23]. The ϕ and ψ backbone torsion angles were sampled also with local moves [17–19] enhanced with the reverse proximity criterion [24]. The peptide bond ω was kept fixed and the iterative algorithm described in Ref.[24] was used to find the proximal solution to the chain closure problem.

This system has a total of 52 torsion angles and 294 water molecules. Eight of the 52 angles were sampled by local moves (resulting in changing six additional backbone angles). Since every 20th step involved the attempted change of a torsion angle every 1040 steps attempted to move each torsion angle once.

5. Results

As discussed above, the general method of arriving at the values of K_C and τ_I is the open-loop protocol of Ziegler and Nichols [15]. From our earlier experience [6], however, we had already reasonable values. Accordingly, the final values of K_C and τ_I and the weight factor $w_{\rm a}$ of Eq. (4) were selected by runs using combinations of $K_C = 1$ and $K_C = 10$, $\tau_I = 2000$ and $\tau_I = 20,000$, $w_{\rm a} = 0.99$, $w_{\rm a} = 0.999$, and $w_{\rm a} = 0.9999$ and monitoring the fluctuation of the acceptance rates and of the tuned stepsizes. In

Table 1. Mean fluctuation of the tuned parameters during tuning.

N_{MC}	K_C	τ_I	w_{a}	$\langle\langle\ \text{SD}_{\text{a}}^{\text{T}}\rangle\rangle$	$\langle\langle\ \text{SD}_{\Delta}^{\text{T}}\rangle\rangle$	$\langle\langle\ \text{SD}_{\text{a}}^{\text{L}}\rangle\rangle$	$\langle\langle\ \text{SD}_{\Delta}^{\text{L}}\rangle\rangle$
25M	1	2000	0.99	4.3	3.8	4.6	1.6
25M	1	2000	0.999	2.7	3.3	2.2	1.0
25M	1	2000	0.9999	4.7	5.7	1.5	0.7
50M	1	2000	0.99	4.6	3.8	3.5	1.7
50M	1	2000	0.999	2.8	3.3	1.9	1.1
50M	1	2000	0.9999	2.4	6.2	2.1	1.0
25M	10	2000	0.99	2.7	8.0	4.2	3.8
25M	10	2000	0.999	1.4	2.9	1.3	1.5
25M	10	2000	0.9999	2.2	4.2	0.3	1.1
50M	10	2000	0.99	3.1	7.0	3.3	4.0
50M	10	2000	0.999	2.1	3.4	1.2	1.6
50M	10	2000	0.9999	2.6	3.2	1.1	1.0
25M	1	20000	0.99	9.0	2.6	5.4	2.1
25M	1	20000	0.999	8.6	1.9	3.5	1.6
25M	1	20000	0.9999	8.7	2.1	4.5	0.9
50M	1	20000	0.99	8.1	3.2	4.9	2.5
50M	1	20000	0.999	8.1	2.7	2.7	1.7
50M	1	20000	0.9999	7.6	2.5	3.5	1.5
25M	10	20000	0.99	5.8	13.3	4.5	6.2
25M	10	20000	0.999	3.8	9.1	1.2	3.6
25M	10	20000	0.9999	2.9	4.7	0.9	1.6
50M	10	20000	0.99	4.8	13.2	6.2	5.5
50M	10	20000	0.999	2.9	9.3	2.1	3.5
50M	10	20000	0.9999	3.0	4.7	1.0	1.5

Legend: (a) N_{MC} is the number of Monte Carlo steps. One million (1M) steps correspond to approximately 1000 move attempts of each torsion angle. (b) K_C and τ_I refer to the parameters in the PID equations (Eq. (1-3)). (c) w_{a} is the acceptance rate weight factor in Eq. (4). (d) The subscript a refers to acceptance rates in %. (e) The subscript Δ refers to torsion stepsize in degrees: the change in a torsion angle is in the range $[-\Delta/2,\Delta/2]$. (f) The standard deviations SD are calculated separately for each torsion angle sampled, from the start of each run. (g) The superscripts L and T indicate averages over torsion with local (concerted) moves and simple torsions, respectively. (h) The SD's are averaged over the respective torsion angle set and over the three separate runs ($\langle\langle\ \rangle\rangle$)

engineering control scenarios, PIC performance is typically robust over a wide range of operating conditions [14]. By analogy, the parameters determined here are expected to be effective in a variety of simulation systems without modification.

Three runs were performed with each parameter combination whose initial conformations were extracted from preliminary runs, at least $5 * 10^8$ MC steps apart. The fluctuations were averaged separately for torsions that were sampled with local moves and those sampled as simple torsions. Table 1 shows the parameter combinations used, and the calculated standard deviations, of the acceptance rates and stepsizes after 25M (1M=10^6) and 50M MC steps. Standard deviations were calculated from 5M long block averages. Based on the calculated fluctuations we selected $K_C = 10$, $\tau_I = 2000$ for the controller parameters. As for the weight to be used for the calculation of the 'current' acceptance rate, $w_\mathrm{a} = 0.999$ appeared to work best for simple torsions and $w_\mathrm{a} = 0.9999$ for the local torsional moves.

The evolution of the acceptance rates and step sizes in one of the three runs during tuning is given in Tables 2 and 3, resp., for the ϕ, ψ and χ_1 angles of the two middle residues (LYS$_3$ and Ac-LYS$_4$). The backbone torsion angles (ϕ and ψ) were sampled with local moves while the side chain torsion angles (χ_1) were sampled as simple torsion. Note that the convergence characteristics of the side chains are expected to be similar to those of the terminal residue(s). The results show that the targeted acceptance rate (30%) is reached reasonably quickly and stays steady.

Table 2. Evolution of the acceptance rates for selected torsion angles during tuning.

	ϕ_3	ϕ_3	ψ_3	ψ_3	χ_3	χ_3	ϕ_4	ϕ_4	ψ_4	ψ_4	χ_4	χ_4
N_MC	$\overline{a}$	a^t	$\overline{a}$	a^t	$\overline{a}$	a^t	$\overline{a}$	a^t	$\overline{a}$	a^t	$\overline{a}$	a^t
1M	18	11	23	14	21	13	19	11	19	12	20	12
2M	20	18	22	18	22	19	20	18	20	18	23	20
3M	22	22	23	23	23	23	22	22	22	22	24	24
4M	24	25	25	27	24	27	23	25	23	25	24	26
5M	24	26	25	28	25	27	24	27	24	26	25	27
10M	25	28	27	29	27	30	26	28	26	30	27	30
20M	27	30	29	31	29	30	28	29	28	29	29	30
30M	28	30	29	30	29	30	28	28	29	30	29	30
40M	28	30	29	30	29	29	28	29	29	30	29	31
50M	29	29	29	30	29	30	29	30	29	30	29	29

Legend: (a) N_MC is the number of Monte Carlo steps$/10^6$. (b) ϕ, ψ, and χ are the torsion angles over the N-C$_\alpha$, C$_\alpha$-C, and C$_\alpha$-C$_\beta$ bonds, respectively; (c) the subscripts 3 and 4 refer to the residue numbers. (d) a^t: tuned acceptance rate (see Eq. (4)), in %. (e) $\overline{a}$: cumulative average of the acceptance rates.

For the choice w_Δ of Eq. (5), tuning runs of length 25M, 50M, and 75M were followed by 50M long runs with fixed stepsizes. The stepsizes were obtained using Eq.(5-6) with $w_\Delta = 0.999$, 0.9999, and 0.99999, after tuning runs of 25M, 50M and 75M long and also by averaging the tuned stepsizes over the last 10M or 25M of the run. The resulting acceptance rate ranges over the three independent runs and the average fluctuation of the acceptance rates in each run are shown in Table 4. For simple torsions, adequately tuned stepsizes were obtained in the 25M long run, using $w_\Delta = 0.999$. However, the backbone torsions require both longer tuning runs and longer stretch to average. Using Eq. (6), at least 50M long tuning run is necessary with $w_\Delta = 0.99999$. Similarly, using simple averaging, averaging over the shorter length used (10M) produced by far the largest acceptance rate spread (after the 50M tuning runs). The tuned stepsizes obtained with simple averaging were somewhat less reliable than the set obtained with Eq. (6).

Table 3. Evolution of the stepsize parameters for selected torsion angles during tuning.

	ϕ_3	ϕ_3	ψ_3	ψ_3	χ_3	χ_3	ϕ_4	ϕ_4	ψ_4	ψ_4	χ_4	χ_4
N_{MC}	Δ^{t}	$\overline{\Delta}$	Δ^{t}	$\overline{\Delta}$	Δ^{t}	$\overline{\Delta}$	Δ^{t}	$\overline{\Delta}$	Δ^{t}	$\overline{\Delta}$	Δ^{t}	$\overline{\Delta}$
1M	2.4	6.0	21	20	27	24	7.0	11	5.2	8.1	22	24
2M	0.3	3.6	8.1	17	22	25	5.6	8.9	0.1	5.4	31	25
3M	0.3	2.7	6.3	13	22	23	2.9	7.0	1.4	3.8	29	26
4M	1.8	2.3	20	13	27	22	1.5	5.9	1.4	3.0	26	26
5M	0.0	2.0	16	13	22	22	1.0	5.1	1.7	2.6	20	26
10M	0.8	1.2	1.1	11	18	20	0.7	3.0	8.2	2.5	25	26
20M	1.0	1.3	11	8.0	21	19	0.7	2.7	5.2	4.4	16	24
30M	2.4	1.3	6.9	7.6	19	18	0.9	1.9	7.6	4.2	25	26
40M	4.2	1.8	15	8.4	14	18	0.5	1.7	2.8	4.2	23	24
50M	0.1	1.7	5.6	7.8	16	18	0.8	1.6	.2	6.2	16	24

Legend: (a) N_{MC} is the number of Monte Carlo steps$/10^6$. (b) ϕ, ψ, and χ are the torsion angles over the N-C$_\alpha$, C$_\alpha$-C, and C$_\alpha$-C$_\beta$ bonds, respectively; (c) the subscripts 3 and 4 refer to the residue numbers. (d) Δ^{t}: tuned step size (see Eq. (6)), in degrees. (e) $\overline{\Delta}$: cumulative average of the step size.

In engineering applications the controller tuning parameters K_C and τ_I are selected to minimize the fluctations in the process variable that are *due to the changes in the manipulated variable* [15]. However, when the evolution of the system results in a change in the relation between the manipulated variable and the process variable then a well controlled system should respond by introducing changes in the manipulated variable. In the previous PIC application [6] the process variable — the solvent density — had an essentially isotropic conformational space and thus the fluctuations in the manipulated variable — the chemical potential parameter B — were small. In the current application, however, there is a significant conformation dependence of the relation between the acceptance rate and the stepsize. This explains the relatively large acceptance rate fluctuation and the lack of success to find K_C and τ_I values that eliminate the fluctuation. Note, however, that the fluctuations in the acceptance rates after tuning are actually larger than the fluctuations during tuning, demonstrating that the tuning process is able to respond to the changing relation between stepsizes and acceptance rates. The magnitude of these fluctuations also shows the importance of selecting the right method for the averaging to determine the fixed stepsize to be used after the tuning.

Table 4. Acceptance rates and their fluctuations using tuned (fixed) stepsizes determined with different averaging strategies.

N_{MC}	$N^{\mathrm{t}}_{\mathrm{MC}}$	w_Δ	$N^{\mathrm{a}}_{\mathrm{MC}}$	$\langle a^{\mathrm{T}}\rangle_-$	$\langle a^{\mathrm{T}}\rangle_+$	$\langle\langle SD^{\mathrm{T}}_{\mathrm{a}}\rangle\rangle$	$\langle a^{\mathrm{L}}\rangle_-$	$\langle a^{\mathrm{L}}\rangle_+$	$\langle\langle SD^{\mathrm{L}}_{\mathrm{a}}\rangle\rangle$
25M	25M	0.999		29	31	4.4	31	44	14.5
50M	25M	0.999		29	32	4.9	34	42	17.6
25M	25M	0.9999		28	29	5.3	25	36	11.8
50M	25M	0.9999		30	30	4.7	30	42	12.3
25M	50M	0.999		29	30	5.1	29	36	20.5
50M	50M	0.999		29	30	4.5	29	36	16.4
25M	50M	0.9999		30	30	4.3	33	41	11.0
50M	50M	0.9999		29	31	5.4	27	35	13.3
25M	50M	0.99999		29	31	5.0	27	29	10.0
50M	50M	0.99999		29	29	4.9	29	29	7.7
25M	50M		10M	30	31	4.6	25	35	11.2
50M	50M		10M	30	30	4.9	28	32	11.9
25M	50M		25M	29	30	5.3	25	31	10.4
50M	50M		25M	29	29	5.2	24	32	11.2
25M	75M	0.99999		29	31	4.4	28	29	12.0
50M	75M	0.99999		30	31	4.5	29	29	12.4
25M	75M		10M	30	31	3.7	29	30	8.4
50M	75M		10M	30	31	4.2	29	32	7.5
25M	75M		25M	28	30	5.2	25	30	8.5
50M	75M		25M	30	31	4.9	28	32	7.5

Legend: (a) N_{MC} is the number of Monte Carlo steps$/10^6$. (b) $N^{\mathrm{t}}_{\mathrm{MC}}$ is the number of tuning steps. (c) w_Δ is the stepsize averaging weight factor in Eq. (6). $N^{\mathrm{t}}_{\mathrm{MC}}$ MC steps of the tuning run. (d)$N^{\mathrm{a}}_{\mathrm{MC}}$ is the length of the run the stepsizes were simply averaged over (when Eq. (6) was not used). (e) $\langle\,\rangle_-$ and $\langle\,\rangle_+$ refer to the smallest and largest value from the three independent runs. (f) $\langle\langle\,\rangle\rangle$ represent averaging the averages over three independent runs.

6. Discussion

We presented an efficient and robust way to automatically tune stepsize parameters in a Monte Carlo simulation. While for systems involving a few degrees of freedom trial and error approach is usually adequate, for more complex systems it soon becomes an exercise in frustration as the acceptance rates converge much slower and the change of one stepsize may affect more than one acceptance rate.

Automatic tuning results in several distinct benefits: First, by eliminating the ad-hoc trials the overall computer time use is reduced. Second, by minimizing the need for human interventions, the overall timeframe of the project is reduced. One result of the relentless increase in processor speeds is that the human time spent with the setup of the project becomes the dominant part of the effort, thus savings at this step will have large impact. Last, but not least, automating the tuning ensures the uniform quality of the tuned parameters. Since the efficiency of sampling is the result of a complex interplay among the various parameters, lower-quality tuning of just a few simulation parameters can have a significant impact on the overall sampling efficiency.

An alternative algorithm for tuning stepsizes [11], based on block averages, has been tested on the alanine dipeptide [8]. Extensive testing resulted in the choice of blocks that included 200 consecutive move attempt of a torsion angle. In our system that would correspond to $2 * 10^5$ (0.2M) MC steps. However, the acceptance rates for systems with as many torsional degrees of freedom as we have are far form converged during runs of this length. Therefore, the update of the stepsize could easily make a change in the wrong direction, necessitating many more iterations until convergence (such occurrences have indeed been observed in our earlier work tuning the chemical potential parameter). In contrast, the method introduced in this paper robustly approaches the target acceptance rate in a few million MC steps.

There are two important questions related to the stepsize tuning that the present paper did not address. First of all, it is important to point out that the conformational distribution sampled during the tuning phase of a simulation contains an unspecified bias with respect to the Boltzmann distribution [11]. This is true for any algorithm that adjusts the stepsize during sampling. This is not a problem for simulations that are performed as part of simulated annealing. For simulations that aim to obtain Boltzmann averages, however, the tuning step has to be performed separately from the production run.

The other question is the selection of optimal target acceptance rate. While in general the sampling efficiency is not too sensitive to the target acceptance rate — this happens to be the case for the test presented here — there are examples where the optimized setup resulted in different optimized acceptance rates for different sampling algorithms [24]. The availability of a robust automatic stepsize tuning algorithm, however, is expected to help establishing the optimization protocol for the selection of target acceptance rate.

References and Notes

1. Metropolis, N.; Rosenbluth, A.W.; Rosenbluth, M.N.; Teller A.H.; Teller, E.J. Equations of State calculations by fast computing machines. *J. Chem. Phys.* **1953**, *21,* 1987-1092.
2. Mezei, M. Comment on “Molecular dynamic simulations in the grand canonical ensemble: Formulation of a bias potential for umbrella sampling” Comparison of the sampling efficiency in the grand canonical ensemble using molecular dynamics and cavity-biased Monte Carlo. *J. Chem. Phys.*

2000. *112,* 1059-1060.

3. Cui, M.; Mezei,M.; Osman, R. Prediction of protein loop structures using a local move Monte Carlo approach and a grid-based force field. *Protein Eng. Des. Sel.* **2008**. doi: 10.1093/protein/gzn056
4. Hastings, W.K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **1970**, *57,* 97-109.
5. Mezei, M. On the potential of Monte Carlo methods for simulating macromolecular assemblies. in *Third International Workshop for Methods for Macromolecular Modeling Conference Proceedings*; Gan H.H., Schlick, T., Eds.; Springer, New York, 2002; pp 177-196.
6. Speidel, J; Banfelder, J.A.; Mezei, M. Automatic Control of Solvent Density in Grand Canonical Ensemble Monte Carlo Simulations. *J. Chem. Theory and Comp.* **2006**, *2,* 1429-1434.
7. Kincaid, R.H.; Scheraga, H.A. Acceleration of convergence in Monte Carlo simulations of aqueous solutions using the Metropolis algorithm. Hydrophobic hydration of methane, *J. Comp. Chem.* **1982**, *3,* 525-547.
8. Hu, J.; Ma, A.; Dinner, A.R. Monte Carlo simulations of Biomolecules: the MC module in CHARMM *J. Comp. Chem.* **2005**, *27,* 203-216.
9. B.A. Berg and H.-X. Zhou, Rugged Metropolis sampling with simultaneous updating of two dynamical variables. *Phys. Rev. E* **2005** *72,* 016712.
10. A. Mitsutake, Y. Sugita and Y. Okamoto, Generalized-Ensemble Algorithms for Molecular Simulations of Biopolymers *Biopolymers (Peptide Science)* **2001**, *60,* 96.
11. Bouzida, D.; Kumar, S.; Swendsen, R.H. Efficient Monte Carlo methods for the computer simulation of biological molecules. *Phys. Rev. A.* **1992**, *45,* 8894-8901.
12. Brooks, B.; Bruccoleri, R.; Olafson, B.; States, D.; Swaminathan, S.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.* **1983**, *4,* 187-217.
13. Stephanopolous, G. *Chemical Process Control, an Introduction to Theory and practice.* Prentice-Hall: Englewood Cliff, N.J.,1984.
14. St. Clair, D. *Controller Tuning and Control Loop Performance.* 2nd ed.; Straight Line Control: 1993.
15. Zeigler J.; Nichols, N.B. Optimum Settings for Automatic Controllers. *Trans. ASME* **1942**, 759.
16. Mezei, M. MMC:Monte Carlo simulation of molecular assemblies, URL: http://inka.mssm.edu/∼mezei/mmc.
17. Dodd, L.R.; Boone, T.D.; Theodorou, D.N. A concerted rotation algorithm for atomistic Monte Carlo simulation of polymer melts and glasses. *Mol. Phys.* **1993**, *78,* 961.
18. Hoffmann D.; Knapp, E.W. Polypeptide folding with off-lattice Monte Carlo dynamics: the method. *Eur. Biophys. J.* **1996**, *24,* 387-404.
19. Deem, M.W.; Bader, J.S. A Configurational Bias Monte Carlo Method for Linear and Cyclic Peptides. *Molec. Phys.* **1996**, *97,* 1245-1260.
20. Kentsis, A.; Mezei, M.; Osman, R. MC-PHS: A Monte Carlo implementation of the primary hydration shell for protein folding and design. *Biophys. J.* **2003**, *84,* 805-915.
21. Beglov, D; Roux, B. Finite representation of an infinite bulk system: solvent boundary potential for computer simulations. *J. Chem. Phys.* **1994**, *100,* 9050-9063.
22. Jorgensen, W.; Chandrashekar, J.; Madura, J.; Impey, R.; Klein, M. Comparison of simple potential

functions for simulating liquid water. *J. Chem. Phys.* **1983**, *79,* 926-935.

23. Brooks, B.; Bruccoleri, R.; Olafson, B.; States, D.; Swaminathan, S.; Karplus, M. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comp. Chem.* **1983**, *4,* 187-217.
24. Mezei, M. Efficient Monte Carlo sampling for long molecular chains using local moves, tested on a solvated lipid bilayer. *J. Chem. Phys.* **2003**, *118,* 3874-3879.