CHAPTER 8

Challenges in Applying Monte Carlo Sampling to Biomolecular Systems

M. MEZEI

Mount Sinai School of Medicine, Department of Structural and Chemical Biology, One Gustave L. Levy Pl., New York, NY 10029, USA
Email: mihaly.mezei@mssm.edu

Dedicated to the memory of Professor Edwin S. Campbell

8.1 Introduction

The era of atomic-level simulations was ushered in by the 1952 paper of Metropolis and coworkers.[1] Molecular dynamics was introduced much later, by Rahman and Stillinger,[2] but Monte Carlo was the preferred choice of simulation technique for quite a while. Currently, however, many simulations of biomolecular systems use molecular dynamics.

On a purely philosophical level, however, the Monte Carlo approach should have the edge since the problem of accurately solving the equation of motion for a very large number of degrees of freedom ($O(10^6)$) appears to be a much more exacting task than the generation of a sample of conformation that 'just' follows a certain distribution (*i.e.* the Boltzmann distribution corresponding to the ensemble in which the simulation was done). Indeed, the Monte Carlo approach has been successfully applied in diverse areas.[3] This success, however, did not extend to the field of simulating macromolecular assemblies.

The aim of this chapter is – instead of reviewing the considerable progress made so far – to discuss the obstacles that prevent the wider use of the Monte

RSC Biomolecular Sciences No. 23
Innovations in Biomolecular Modeling and Simulations: Volume 1
Edited by Tamar Schlick

Published by the Royal Society of Chemistry, www.rsc.org

Carlo method for macromolecular simulations. Successful adoption of the Monte Carlo method for conformational sampling of macromolecular assemblies requires solution(s) to the following problems: (1) convince investigators that it is worth it; (2) devise move sets that generate large enough correlated changes that can be accepted with reasonable probability; (3) develop efficient treatment of non-pairwise additive potentials; (4) develop efficient treatment of long-range contributions to the system's energy; and (5) the efficient parallelization of the algorithm. In the remainder of this chapter these issues will be treated one by one. For many fundamental details see refs. 4–6 and for some recent applications see refs. 7 or 8. Richey provided a historical account of the development of the Markov-chain Monte Carlo method and the widening of the scope of its application.[34]

Note that there are two distinct approaches for the enhancement of conformational sampling: (a) enhancing the algorithm generating successive conformations during the simulation and (b) manipulating the treatment of the energy surface governing the simulation. Typical examples for the second approach are umbrella sampling[9] or replica exchange;[10] but they are all equally applicable to molecular dynamics and Monte Carlo and thus will not be discussed in this chapter – it is for the algorithm generating successive conformations where the choice between Monte Carlo and molecular dynamics arises.

8.2 Basic Ideas of Monte Carlo Sampling

The Metropolis method[1] obtains a Boltzmann-averaged sample of configurations by generating a Markov chain. Markov chains have the basic property that each member of the chain (series of configurations in the case of molecular systems) is obtained solely based on the member preceding it. For each probability distribution defining a new member (called transition probability) there is a corresponding distribution describing the members of the Markov chain defined by the transition probability. The most general relation between transition probabilities has been described by Hastings.[11]

The general pattern for selecting the next member of the Markov chain, usually called a Monte Carlo move, is by making a random change in the current configuration and accepting it with a probabilistic filter – the next member of the chain will be either the configuration just generated (if accepted) or the repeat of the previous configuration (if rejected). The classic case is the move of one atom by selecting $\Delta \boldsymbol{x}_i$ with uniform probability within a cube of fixed size around $\boldsymbol{x}_i$, and accepting the new configuration with probability $\min\{1,\exp(-\Delta E/kT)\}$ where ΔE is the energy difference between the two configurations. In other words, if the change lowers the energy, accept for sure, if not, then accept it with exponentially diminishing probability as the difference increases. The beauty of this method lies in the fact that it leads to Boltzmann-averaged distribution without having to calculate the partition function that is the normalizing factor in calculating the Boltzmann factor. Further, it requires only the *change* in the energy, leading to simplifications in the energy calculations in certain cases – this issue will be discussed in Sections 5 and 6 in more detail.

The design of a successful Monte Carlo sampling algorithm consists of the selection of the trial moves (move set) and the distribution these moves are sampled from. First requirement is that the move set should provide for the sampling of all the degrees of freedom in the system. This obviously includes the change in the atomic coordinates. For thermodynamic ensembles other than the canonical, further moves can include the change in the volume (in the isobaric-isothermal ensemble) or even the number of particles (in the grand-canonical ensemble), requiring the creation or annihilation (attempts) of part of the system.

The guiding principle in selecting the type of moves and the distribution(s) they are sampled from is to simultaneously maximize the change in the configuration and minimize the increase in the energy since the computational effort in calculating ΔE is (in most cases) independent of the magnitude of the change. In the basic example above, this involves optimizing the edge of the cube within which the random change is made. The selection probability can also be different from the uniform distribution – for example, moves can be biased in the direction of force acting on the atom moved[12] – with a concomitant modification in the acceptance probability – the so-called force-biased sampling. This idea can be applied to the volume changes[13] and the insertion/deletion steps[14] in the isothermal-isobaric and in the grand-canonical ensembles, respectively. The important thing in the use of such biased sampling is that the probability distribution of the bias is well defined since it is needed in the modified acceptance expression.

While molecules can, in principle, be considered 'just' a collection of atoms and sampled accordingly, the resulting simulation would be very inefficient since the intramolecular energy changes very steeply with the change in bondlength. Sampling of molecules involves the sampling of orientation and conformation as well. This is usually achieved by some random rotation of the molecule and by some random changes in the intramolecular coordinates, usually torsion or bond angles. This is an area rich in possibilities whose discussion is outside the scope of this chapter. Some examples will be discussed in Section 8.3.

8.3 Why 'Bother' with Monte Carlo?

The spectacular success of molecular dynamics raises the question of the worthiness of the effort in trying to apply the Monte Carlo method for such problems. However, success begets success, and the effort expended in improving both the theoretical and computational aspects of molecular dynamics has far exceeded the effort spent on developing Monte Carlo. Reversing this disparity in the efforts expended thus can lead to development of efficient Monte Carlo applications. One example of an issue that has never been thoroughly examined is the fact that while the most general form of generating a Metropolis move has been described by Hastings;[11] most applications use a more limited form and no systematic study has been performed to determine the optimal choice.

Besides the philosophical argument cited in the Introduction it has also been frequently remarked that the very fact that is the source of molecular dynamics'

success – the strict reliance on the time evolution of the system – is also an intrinsic limitation since each step the simulation can make is very small. Monte Carlo, not having this tie to time, is free, at least in principle, both to make much larger steps and to take shortcuts in the configurational space. The problem is that when one takes shortcuts, it is easy to get lost. It is also important to note that with the Monte Carlo approach only indirect kinetic information (*via* analysis of activation barriers) can be obtained.

The superiority of current molecular dynamic applications over Monte Carlo is, however, not absolute. In a comparison of Monte Carlo and molecular dynamics on liquid hexane[15] Monte Carlo outperformed molecular dynamics. Also, simulation of aqueous systems in the grand-canonical ensemble[14] is an order of magnitude more efficient with Monte Carlo than with molecular dynamics.[16] However, these successes have not (yet) carried to simulations of biomacromolecules.

8.4 Correlated Moves

Another significant difference between molecular dynamics and Monte Carlo is that while devising the best numerical method for the calculation of the next conformation is a strictly scientific enterprise, there is an art to the design of Monte Carlo moves – followed, of course, by the scientific analysis and implementation. As discussed in Section 8.2, a successful Monte Carlo move makes a large change in the conformation that only changes the energy by an amount that is commensurable to kT so that the acceptance probability is high enough. This generally requires the selection a 'soft' degree of freedom (*e.g.* changing torsion angles instead of translating atoms). An additional limitation on the change is that the ratio of the probability of selecting this change and its reverse be known (or be computable).

The major question in the design is the selection of the set of atoms to be moved. First, it is clear that if the change in energy when moving two sets of atoms is the sum of energy changes for moving the two sets independently (and, concomitantly, the computational efforts are also additive or nearly so) then moving them separately is more advantageous, since collision caused by move in one of the sets would cause the rejection of the combined move while it would still accept the move of the rest of the set if the moves are performed separately. This would suggest that generally it is advantageous to select as small a set of atoms to move as possible.

Biomolecular systems, however, are generally dense, interconnected, and heterogeneous. This makes the design of efficient moves rather difficult since the additivity of the energy changes assumed in the argument above is unlikely to hold. Instead, the chance for finding larger moves with smaller energy change lies in finding correlated moves of selected atoms that end up avoiding clashes while making significant change in the conformation. Changing torsion angles on a side chain or performing a local backbone moves[17] are examples of such correlated moves using a small set of atoms.

The opposite to moving a single atom or a small subset is the moving all atoms in each step. This was shown to be feasible[18] if the change in the conformation is along one of the eigenvectors of the systems Hessian. Unfortunately, the complexity of the calculation of the eigenvectors is $O(n^3)$, making this technique difficult to extend to large systems. Furthermore, its efficiency is lost when applied to systems in explicit waters.[19] It remains to be seen if using judiciously chosen Hessian blocks of limited size that could reduce the complexity to $O(n^2)$, (as suggested for the simulation of lipid bilayers)[6] can be found to be implemented efficiently.

To illustrate the frustration that can result in attempts of 'clever' Monte Carlo moves, Figure 8.1 shows the average displacement and orientational correlation of lipid molecules in two simulations. In one, simulations were performed using a move set that includes whole molecule translations, rotations and torsion angle changes on a bilayer of DMPC molecules[20] using the program MMC.[21] In the other, selected rotations of one lipid around the bilayer normal were accompanied by a similar rotation of the lipid nearest to the first lipid, but in the opposite direction. It was expected that such correlated rotations would act like two cogwheels and reduce the clashes between lipids, resulting in higher acceptance rates and thereby improved sampling. However, no significant change was observed either in the diffusion rate or in the decay of orientational correlations, indicating that different approaches are needed for accelerated the sampling.

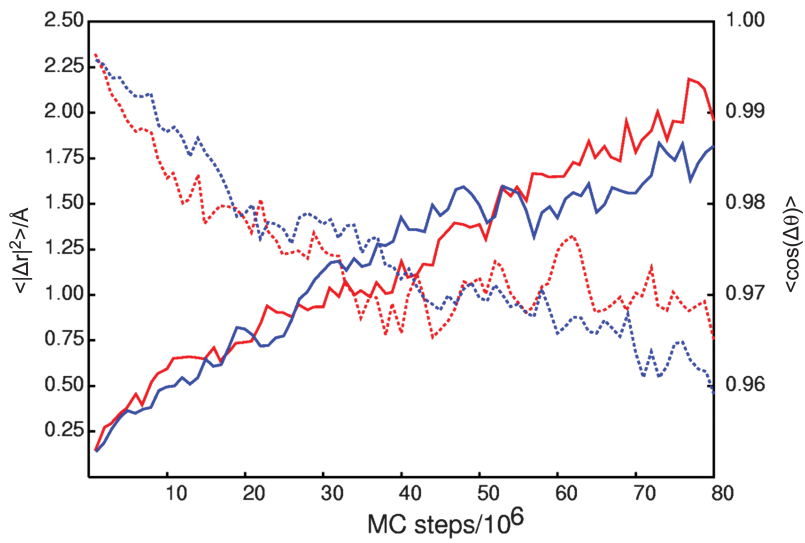

Figure 8.1 Comparison of lipid sampling with correlated two-lipid rotation (red) with one lipid rotation (blue). Full line: COM displacement square; dotted line: decay of the orientational correlation around the membrane normal.

8.5 Cooperative Potentials

Most of the successful Monte Carlo techniques change a small fraction of the atoms at each single step. This means that the efficiency of the method degrades when the energy update involves atom pairs that were not changed – which is just the case for most Monte Carlo moves.

Cooperativity is generally introduced either by the use of three-body, four-body, *etc*., terms[22] or by the introduction of polarization – either by induced dipoles (or, possibly, higher order multipoles)[23,24] or by induced changes in the partial charges or molecular geometry,[25] essentially mimicking induced multipoles. Since the multibody terms are generally short ranged, their use with Monte Carlo sampling will not change the computational complexity of the calculations. However, cooperative potentials using some form of polarizability will essentially raise the complexity of the energy calculation when the attempted move involves a small number of atoms only.

While Monte Carlo calculations have been performed with polarizable potentials on relatively small systems despite the significant additional computational cost, this is clearly an unsatisfactory solution since for larger systems the additional expense will become prohibitive. Possible solutions to this problem is either (a) limit the update of the polarization state to every n-th step assuming that the change in the polarization state in a single step is (can be considered) negligible or (b) run the simulation with the pairwise additive part of the potential and correct the Boltzmann probability of the conformations selected for averaging with the cooperative part during post-processing. However, the first solution clearly introduces an error while the second solution can exacerbate round-off errors if the cooperative terms fluctuate more than a few kT (due to the exponential in the Boltzmann factor). Other approximations to the calculation of the polarization contribution have been developed but tested for liquid water only.[26] So far no efficient and procedure with sufficient numerical precision has been developed for the use of cooperative potentials with local Monte Carlo moves with proteins or nucleic acids where the presence of fully charged atoms would make approximate solutions significantly less accurate than for water. Thus it may well be that use of cooperative potentials will be restricted to Monte Carlo moves where the whole (or, at least, a large part of the) system is changed at each step.

8.6 Long-range Energy Contributions

The standard technique for dealing with the long-range contributions to the electrostatic energy is the use of Ewald sums. This technique, however, raises the same problem for Monte Carlo methods where a small part of the system is changed in each step as the calculation of the energy with cooperative potentials, due to the need for the summation in the reciprocal space.

The root of the difficulty of treating long-range electrostatics is the slow *and* conditional convergence of the dipolar lattice sum in three dimensions. Without using an explicit treatment of the full extent of the long-range contributions, for

acceptable accuracy, rather long cutoffs distances are needed that, incurs a large additional computational burden. There are, however, two possible options that may be amenable to efficient Monte Carlo implementation. First, R. Sperb[27] published a formalism that uses fast-converging series, without resorting to the reciprocal space. This can thus be implemented efficiently since these series would involve only the changed part of the system. Second, E. Campbell[28,29] has developed a formalism for calculating the Ewald sums (for multipoles of any order, not just for dipoles) from two components: one, called crystal constants, that are functions only of the periodic system (and include the contributions from the direct and reciprocal space) and an other component whose terms depend on the actual configuration. Change in a conformation thus affects the part not involved in lattice sums and thus could also be amenable to efficient Monte Carlo implementation.

8.7 Parallelization

Any algorithm that is designed for simulating large molecular assemblies has to be amenable to massive parallelization. Monte Carlo methods are usually considered amenable to 'embarrassingly parallel' treatment where the same calculation is repeated on different processors with different random-number seeds and the result is averaged at the end. The problem is more difficult if fine-grained parallelization is required. There are two obstacles to efficient implementation of massive fine-grained parallelization.

The tasks involved in a Monte Carlo move include some additional calculations beyond that of the change in the energy of interaction between the atoms changed in the trial move and the rest – the part that is amenable in general to massive parallelization. Other tasks include the generation of the trial change in the configuration and (in most cases) the change in the interaction energy between the moved atoms. While these usually take only a small fraction of the computational effort, if massive fine grained parallelization is planned then this fraction will increase proportionally to the number of processors, and ultimately limiting the parallel efficiency. Since these smaller tasks are rarely amenable to massive parallelization the only possibility is to perform them simultaneously on different parts of the system; but that is only possible if the system is large enough and the moves are local enough that they can be generated in such a way that all changes are independent of each other. For example, torsion angle changes on different side chains can be generated on different protein side chains and the energy change involved can be calculated in parallel and then tested for acceptance sequentially using the massively parallelized energy calculation. Such a decomposition has been discussed by Heffelfinger and Lewwitt.[30] The idea of 'pre-fetching' (*i.e.* calculate multiple likelihoods ahead of time and only use the ones that are needed) were introduced and tested by Brockwell.[33]

The biggest obstacle to massive parallelization on distributed memory systems is the need of communication at the Metropolis decision stage, since the

acceptance decision depends on the energy of the *whole* system. While shared memory systems can initiate communication with little or no latency, this is not the case for distributed memory system – the ones that are likely to be used for massively parallel applications, given their much lower price. Similar problems would arise for GPU implementations, since communication with current GPUs have relatively large latencies.[31]

The effect of latency on the parallel efficiency of the fine-grained parallelization using MPI was tested on two systems of TIP3P[32] water molecules under periodic boundary conditions using the minimum-image convention. Calculations were performed on two systems containing 3000 and 30 000 molecules, respectively, on computers employing either distributed memory (marked as DM) or shared memory architecture (marked as SM). The distributed memory runs were on our G5 cluster using gigabit Ethernet connections and the shared memory runs were on an 8-CPU SGI Origin system using R12K processors and the simulations were run with the program MMC.[21] Figure 8.2 shows the results for up to 8 processors, plotting the efficiency factor $E = T(N_{cpu})*N_{cpu}/T(N_{cpu}=1)$ as a function of the number of processors, N_{cpu} (note the logarithmic scale on the horizontal axis). Here $T(N_{cpu})$ and $T(N_{cpu}=1)$ are the run times for simulations using N_{cpu} and 1 processors, respectively. Ideal parallel efficiency would result in $E = 1$.

The results on the parallelization of the Monte Carlo steps also show nearly ideal efficiency on the shared memory system indicating that the parallelization is distributing the workload efficiently among the processors. However, due to

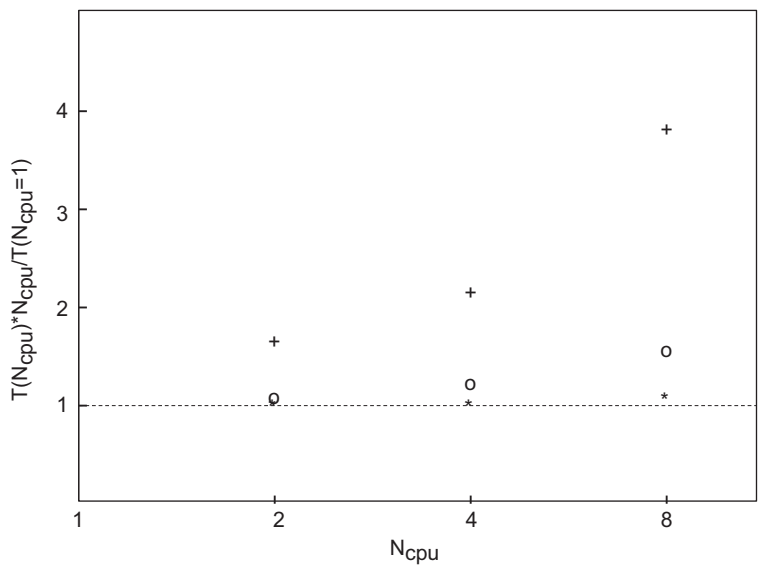

Figure 8.2 Parallel efficiencies of water simulations. +: 3000 waters, distributed memory; ○: 30 000 waters, distributed memory; *: 3000 waters, shared memory; broken line: ideal speedup.

the latency on the communication on the distributed memory system the efficiency is disappointingly low. As expected, there is an improvement with the increased system size.

8.8 Conclusion

This chapter discussed the open problems that need to be resolved for the Monte Carlo algorithms to be used for simulating macromolecular systems. One of the problems facing an efficient Monte Carlo implementation is partly the result of the extensive freedom one has in the choice of sampling algorithms since definitive comparison of such choices require large-scale simulations on a variety of systems. Other problems arise when the energy change is needed to be calculated for cooperative potentials, for the use of the well-tested Ewald summation to calculate the long-range electrostatics contributions and for massive parallelization on distributed memory computer systems. Possible solutions were suggested for the calculation of long-range electrostatics and for some aspects of parallelization. It is hoped that further developments, both in algorithm design and test as well as in hardware/software development will solve most of these problems.

References

1. N. A. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, *J. Chem. Phys.*, 1953, **21**, 1087.
2. A. Rahman and F.H. Stillinger, *J. Chem. Phys.*, 1971, **55**, 3336.
3. P. Diaconis, *Bull. Amer. Math. Soc.*, 2009, **46**, 179.
4. W. W. Wood, in *Physics of Simple Liquids*, ed. H. N. V. Temperley, J. S. Rowlinson and G. S. Rushbrooke, 1968, North Holland, Amsterdam, pp. 115–229.
5. M. O. Allen and D. J. Tildesley, *Computer Simulation of Liquids*, 1987, Oxford University Press.
6. M. Mezei, in *Third International Workshop for Methods for Macromolecular Modeling Conference Proceedings*, ed. H. H. Gan and T. Schlick, 2002, Springer, New York, pp. 177–198.
7. J. P. Ulmschneider and W. L. Jorgensen, *J. Am. Chem. Soc.*, 2004, **126**, 1849.
8. M. Cui, M. Mezei and R. Osman, *Protein Eng. Des. Sel.*, 2008, **21**, 729.
9. G. M. Torrie and J. P. Valleau, *J. Comp. Phys.*, 1977, **23**, 187.
10. D. D. Franz, D. L. Freeman and J. D. Doll, *J. Chem. Phys.*, 1979, **93**, 2769.
11. W. K. Hastings, *Biometrika*, 1970, **57**, 97.
12. M. Rao, C. S. Pangali and B. Berne, *Molec. Phys.*, 1979, **71**, 3366.
13. P. Jedlovszky and M. Mezei, *Mol. Phys.*, 1999, **96**, 293–296.
14. M. Mezei, *Molec. Phys.*, 1987, **61**, 565; Erratum, 1989, **67**, 1207.
15. W. L. Jorgensen and J. Tirado-Rives, *J. Phys. Chem.*, 1996, **100**, 14508.
16. M. Mezei, *J. Chem. Phys.*, 2000, **112**, 1059.

17. L. R. Dodd, T. D. Boone and D. N. Theodoru, *Molec. Phys.*, 1993, **78**, 961.
18. T. Noguti and N. Go, *Biopolymers*, 1985, **24**, 527.
19. N. Go, *Personal Communication*, 1999.
20. P. Jedlovszky and M. Mezei, *J. Chem. Phys.*, 1999, **111**, 10770.
21. M. Mezei, URL: http://inka.mssm.edu/~mezei/mmc
22. J. Detrich, G. Corongiu and E. Clementi, *Int. J. Quant. Chem: Quant. Chem Symp.*, 1984, **18**, 701.
23. F. H. Stillinger and C.W. David, *J. Chem. Phys.*, 1978, **69**, 1473.
24. E. S. Campbell and M. Mezei, *J. Chem. Phys.*, 1977, **67**, 2338.
25. H. J. C. Berendsen and G. A. van der Velde, *CECAM Workshop Report*, 1972, 63.
26. M. W. Mahoney and W. L. Jorgensen, *J. Chem. Phys.*, 2001, **114**, 9337.
27. R. Sperb, *Molecular Simulation*, 1999, **22**, 199.
28. E. S. Campbell, *J. Phys. Chem. Solids*, 1965, **26**, 1395.
29. E. S. Campbell and M. Mezei, *J. Comp. Phys.*, 1976, **21**, 114.
30. G. S. Heffelfinger and M. E. Lewitt, *J. Comp. Chem.*, 1996, **2**, 250.
31. J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, *Proceedings of the IEEE*, 2008, **96**, 879.
32. W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey and M. L. Klein, *J. Chem. Phys.*, 1983, **79**, 926.
33. A. E. Brockwell, *J. Comp. Graph. Stat.*, 2006, **15**, 246.
34. M. Richey, *The Am. Math Monthly*, 2010, **117**, 383.