SANDIA REPORT

SAND2003-4113
Unlimited Release
Printed November 2003

Parallel Tempering Monte Carlo in LAMMPS

Mark Sears, Steve Plimpton, and Mark Rintoul

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-4113
Unlimited Release
Printed November 2003

Parallel Tempering Monte Carlo in LAMMPS

Mark Sears
Computational Materials & Molecular Biology

Steve Plimpton and Mark Rintoul
Computational Biology

Sandia National Laboratories
P.O. Box 5800, MS-0310
Albuquerque, NM 87185-0310

Abstract

We present here the details of the implementation of the parallel tempering Monte Carlo technique into a LAMMPS, a heavily used massively parallel molecular dynamics code at Sandia. This technique allows for many replicas of a system to be run at different simulation temperatures. At various points in the simulation, configurations can be swapped between different temperature environments and then continued. This allows for large regions of energy space to be sampled very quickly, and allows for minimum energy configurations to emerge in very complex systems, such as large biomolecular systems. By including this algorithm into an existing code, we immediately gain all of the previous work that had been put into LAMMPS, and allow this technique to quickly be available to the entire Sandia and international LAMMPS community. Finally, we present an example of this code applied to folding a small protein.

Contents

Figures:

SAND2003-4113
Unlimited Release
Printed November 2003

Parallel Tempering Monte Carlo in LAMMPS

Mark Sears
Computational Materials & Molecular Biology

Steve Plimpton and Mark Rintoul
Computational Biology

Sandia National Laboratories
P.O. Box 5800, MS-0310
Albuquerque, NM 87185-0310

Introduction

The purpose of this project is to add a Parallel Tempering Monte Carlo (PTMC) capability to the LAMMPS[1] atomistic molecular dynamics code. PTMC is a method that has been proposed and implemented in various forms over the last decade or so. The early ideas of simulated tempering[2], multicanonical ensembles[3] and expanded ensembles[4] were all schemes that generalized the possible moves that could be made to a single configuration Markov chain. The first mention of parallel tempering where a number of configurations are propagated simultaneously was an application to spin glasses[5], and this was rapidly followed by the use of this technique to look at biomolecules[6].

In the following we provide a brief description of PTMC and a discussion of the implementation in LAMMPS. The remainder of this report is devoted to analyzing the results from a problem based on a short (5 residue) peptide chain in aqueous solvent.

PTMC background

We begin with a discussion of the well-known Metropolis scheme for Monte Carlo sampling of a probability distribution. We imagine that S is a space of states and $P(s)$ is the probability of state $s \in S$.

We also imagine that given a state s we have some method for generating a new state s' "randomly" from s . The precise details of how the new state is generated are not important as long as the process satisfies a few simple conditions.

We construct a Markov chain of states, which represents the probability distribution $P(s)$ by the following simple rules. Assume that the state at some point is s_0 and we generate a proposed state s' . We accept or reject this state as follows:

1. If $P(s') \ge P(s_0)$ then accept s' as the next state in the chain.

2. Otherwise, accept s' with probability $P(s')/P(s_0)$.

Note that rejection of s' means that the next state in the chain remains s_0, and we don't need to know the normalizing factor P_0. We only need to be able to compute the ratio of probabilities of any two states. It is easy to show that if s_0 is chosen with probability $P(s_0)$ then s_1 will have probability $P(s_1)$ under the above rules. Let the generating method be described by a transition matrix $T_{ss'}$, which is the probability of generating the state s' from the state s. Clearly the probability of generating some state is unity, and the probability of generating the same state we assume to be zero. The probability of generating any particular state is greater than or equal to zero. We also need that the transition probability is symmetric, so that the probability of generating s' from s is the same as that of generating s from s'.

$T_{ss'}$ thus has the following properties:

$$\begin{aligned} &T_{ss'} = T_{s's} \\ &T_{ss'} \geq 0 \\ &T_{ss} = 0 \\ &\sum_{s'} T_{ss'} = 1 \end{aligned}$$

Now given a probability distribution at the beginning of a step of the above process we can compute the probability distribution after the step, all we need to know is the conditional probability for transition. There are three cases where the new state has greater or equal probability, the new state has less probability, and the new state is the same as the old state. The third case only happens when a new state has been proposed but is rejected. Assume that the probability of being in state s at the beginning of the step is $P(s)$, the target probability. We have that the probability of ending up in state s' is $Q(s')$ where

$$Q(s') = \sum_{s<s'} P(s)\frac{P(s')}{P(s)}T_{ss'} + \sum_{s>s'} P(s)T_{ss'} + P(s')\sum_{s>s'}(1-\frac{P(s')}{P(s)})T_{s's}$$

Using the above properties this becomes $Q(s') = P(s')$ and so the target probability distribution is preserved under the above process.

We go through this well known argument to show that the precise details of the method described by $T_{ss'}$ do not matter and we do not have to even know what the transition matrix is in detail. This argument also implies that we can build a Metropolis scheme from different moves (described by different transition matrices) and still preserve the Markov chain property, as long as each individual move has the above properties.

Given a long enough chain and the above modest assumptions about the method for generating new states we are guaranteed to be able to estimate any observable to any specified degree of accuracy, under the additional hypothesis that there exists a set of moves from the starting state to any other state. The accuracy we get in practice is typically proportional to $N^{1/2}$ where N is the chain length we generate. Accuracy is generally limited by the fact that the initial state is usually not representative of the distribution, so that a large number of equilibration steps must be taken at the beginning and then thrown away. Accuracy can also be severely limited by the difficulty of generating new states that are not immediately rejected: the acceptance ratio must be sufficiently high that we do generate a random set of states.

We can parallelize this scheme trivially by running multiple independent chains, however the initial equilibration overhead is the same for all the chains and so severely limits the effective speed-up. We can also attempt to parallelize the method of generating new states, although this is often much more difficult.

Both of these approaches fail to deal with the fact that the acceptance ratio is often strongly suppressed in systems with large barriers. MC steps which move across the barrier have extremely low probability and are never taken in practice, thus the scheme fails to sample the distribution correctly. PTMC is an approach that can help improve the acceptance ratio with the added benefit of being trivially parallel.

In the ordinary Metropolis scheme we evolve a single chain at a single temperature. The chain samples the probability distribution $P(s,T)$ where we now call out the temperature parameter explicitly. The Boltzman distribution $P(s,T) = P_0 e^{-E(s)/T}$ defines the probability of a state s at temperature T where $P_0 = 1/Z$ and Z is the partition function. Because the Metropolis scheme only uses ratios of probabilities of course the partition function factor does not need to be computed. If we replicate the system K times with temperatures $T_1 \ldots T_K$, then we can imagine a joint probability distribution $\mathbf{P}(s_1 \ldots s_K, T_1 \ldots T_K)$. Of course our replicas are independent, and therefore the joint distribution is just the product of the individual ones:

$$\mathbf{P}(s_1 \ldots s_K, T_1 \ldots T_k) = \prod_{k=1\ldots K} P(s_k, T_k)$$

Now we define the PTMC move that allows the simultaneous generation of all the chains. For most steps we use ordinary Metropolis steps to evolve each chain at its desired temperature. At intervals we attempt to swap chains. If one chooses a pair of chains, say 1 and 2, then before the swap the joint probability is proportional to

$$e^{-E_1/T_1} e^{-E_2/T_2}$$

and after the swap the probability is proportional to

$$e^{-E_1/T_2} e^{-E_2/T_1}$$

The ratio of probabilities is therefore proportional to

$$e^f$$

where f is the quantity

$$f = (1/T_1 - 1/T_2)(E_1 - E_2)$$

Then if $f \geq 0$ we accept the chain swap unconditionally. If $f < 0$ we accept the swap with probability e^f. Again it is easy to show that this move will generate states in the correct joint probability distribution.

The following points are more or less trivial:

- In practice we do not swap the chains themselves. Instead, we swap the temperatures associated with the chains. This implies a parallel implementation that is extremely trivial.
- If the two temperatures are close, then the acceptance probability will be high.
- Just because we have a parallel scheme does not mean that there is any benefit to it.

Suppose we are only interested in the behavior of the system at one of the temperatures, say T_1. Then the results of all of the other chains are useless. The basic counter argument is that PTMC allows chains to run at a number of different temperatures, and in practice running at high temperatures allows the system to jump over barriers much more easily. Therefore a PTMC simulation with K chains can in principle produce results much more quickly and with a better sampling than the single chain simulation run. The test problem and analysis described below attempts to justify this statement.

Another scheme is available with PTMC that we have not attempted to exploit. If we run a large number of simulations at nearby temperatures then any individual state could be considered to contribute to several temperatures. This scheme is called umbrella sampling and it is a means of making more effective use of the computational effort. However, this is only of benefit in the case that we are interested in more than just the lowest temperature.

There is no known way to decide ahead what are the correct set of temperatures and how many chains to run in a simulation. The following heuristics are available:

- The range of temperatures is defined at the minimum by the target temperature and at the maximum by a temperature that is high enough so that the system rapidly equilibrates.
- The best set of temperatures is such that the acceptance for exchange of chains is approximately the same between all pairs of chains. This will generally imply the use of a geometric ratio between the temperatures.
- If there are too many chains then there may be useless extra work. If there are too few chains then equilibration may take too long.

This leaves a lot of room for trial and error, and we suspect that some experimentation will be needed for individual problems to make the best use of this facility.

Design of PTMC capability in LAMMPS

We chose to work with LAMMPS because it is an existing robust and parallel molecular dynamics simulation code widely used here at Sandia. We note that LAMMPS is already parallel but that many simulations are too small to make effective use of very large numbers of processors. Implementation of PTMC on top of LAMMPS could therefore significantly extend the parallel range of LAMMPS. For example, if a system normally would make efficient use of 32 processors and we run 32 PTMC chains in parallel then we can now use 1024 processors very effectively.

LAMMPS is not a Monte Carlo code. However, molecular dynamics is a very good state generator and this is the mode in which the LAMMPS core is used. In a typical case we want to run K chains on P processors each. The approach we implemented uses a total of $KP+1$ processors. One processor is assigned the task of supervising all the others. The remaining KP processors are split into K sets of P processors each, and each set gets one chain (task) to run. We used the MPI communicator facility to build the processor groups and to isolate communications within each task from the communications that the master process uses to control tasks. This allowed minimal modifications to LAMMPS. The PTMC code was written as an umbrella Fortran-90 code that runs LAMMPS as a small set of subroutines.

The master process reads an input file that describes the number of tasks, list of temperatures, and starting configuration files for each task. The master process sends messages to the slave tasks, which wait in a control loop for work to do. The slave tasks run the LAMMPS simulation for a specified number of time steps and then return to the control loop. A message is sent from the slave task to the master task with the current potential energy of the task simulation.

When all of the slave tasks have entered the control loop the master process has all of the information it needs to perform the PTMC interchange step between any two slaves. The PTMC move described above is attempted with some probability (.8) between adjacent temperatures only. The master code keeps track of which temperature is assigned to which chain. All of the interchanges are tried and then all of the tasks begin running again with their new assigned temperatures.

In the end, almost no modifications to the original LAMMPS code were needed. The PTMC capability resides almost entirely in a few short driver routines that replace the LAMMPS main subroutine and call the remainder of the LAMMPS subroutines.

Test problem and analysis

The main test problem that we chose to run is a simulation of a short (5 residue) peptide chain (met-enkephalin) in water solvent. This system was chosen because published results[7] were available for comparison and because a simple peptide such as this is a truncated example of the kinds of systems that we would like to be able to simulate. Fig.1 is a representation of the met-enkephalin molecule. The solvent waters in the simulation are not shown.

The initial configurations were generated by Marcus Martin using the Towhee code[8]. This is a configurational bias Monte Carlo code that can generate a plausible configuration for a molecule together with all of the necessary force-field parameters, including a surrounding set of solvent (water) molecules. We don't discuss this process any further except to make the comment that it is extremely non-trivial to generate a complete description of a molecular system for LAMMPS. The basic configuration consists of the met-enkephalin molecule (the 5 amino acid chain together with modifications for the amino and carboxyl end groups) plus a surrounding set of water molecules in the TIP-3P[9] model. The basic force-field parameters are those of the CHARMM-28 set[10]. This is the major difference between our computation and that of reference [7], where the AMBER[9,11] force field was used. The force field includes the usual electrostatic and non-bonded (Lennard-Jones) forces, together with bond, bond-angle, bond-torsion, and improper forces. We chose to run an all-atom simulation including hydrogens, although the hydrogen vibrations were partially simulated with the SHAKE algorithm. The amino terminus was treated as protonated and the carboxyl terminus was unprotonated.

The major run parameters are the number of tasks, the set of temperatures, the time step for evolving the MD part of the simulation, the interval of time to run between parallel tempering swaps, and the total length of time for the simulation. We chose these to be similar to those in the above reference. In addition, there are some parameters for the thermostat in the MD simulation. These were set to relax the temperature slowly to the target temperature for that sample. After initialization, a sample run of length 2×10^6 steps was made using ordinary MD (without tempering). This data is compared with the data for the parallel tempering runs that were of length 10^5 steps for 16 configurations.

Acceptance ratios for replica exchange were measured during the tempering run, and we found that values of .25 were typical. No adjacent pairs were exceptional, indicating that the above temperatures are a good choice to balance out the distribution of configurations.

Fig. 2 shows how one of the sample configurations visits various temperatures during the course of the simulation. All of the sample configurations appeared to have similar random appearance. We did not find or think of a good statistical test to decide whether the trajectories represent a correct sampling, other than to see if all the configurations visit all of the temperatures with equal probability.

We also made Ramachandran plots of the density of states in ψ,ϕ angle space for each of the five residues. We compared these plots with the results from the above paper. Only the grossest features of the plots were similar, however. We do not yet know if this is due to a subtle error in our input data or force field model, or is indicative of a fundamental difference in the force fields or represents some error in the above paper. Figure 6 shows a Ramachandran plot for a random sample of 150 proteins in the PDB. Figure 7 shows the Ramachandran plot for the configurations at 275K compared with Fig. 8, which is the Ramachandran plot for the pure MD run. Figures 9, 10, 11, 12, and 13 are the plots for the individual residues (TYR, GLY, GLY, PHE, and MET).

In order to analyze the results we attempted a cluster analysis based on principal components analysis (PCA). The coordinates used were the 10 $\phi-\psi$ backbone angles. Figure 14 shows the results.

An additional cluster analysis was performed based on a scatter diagram of the end-end radius R_{e^2} versus the radius of gyration R_{g^2}. Only the C_α backbone atoms were analyzed. We note that for an ordinary polymer chain this distribution should lie around a straight line and we see that this behavior is obtained.

Figure 3 shows this plot for the lowest temperature configurations and Fig. 4 combines the data for all the configurations. We can compare these plots with Fig. 5, which shows the configurations obtained in the long straight MD run. We do not see much difference in these plots, which indicates that the PT and MD runs are visiting much the same set of configurations.

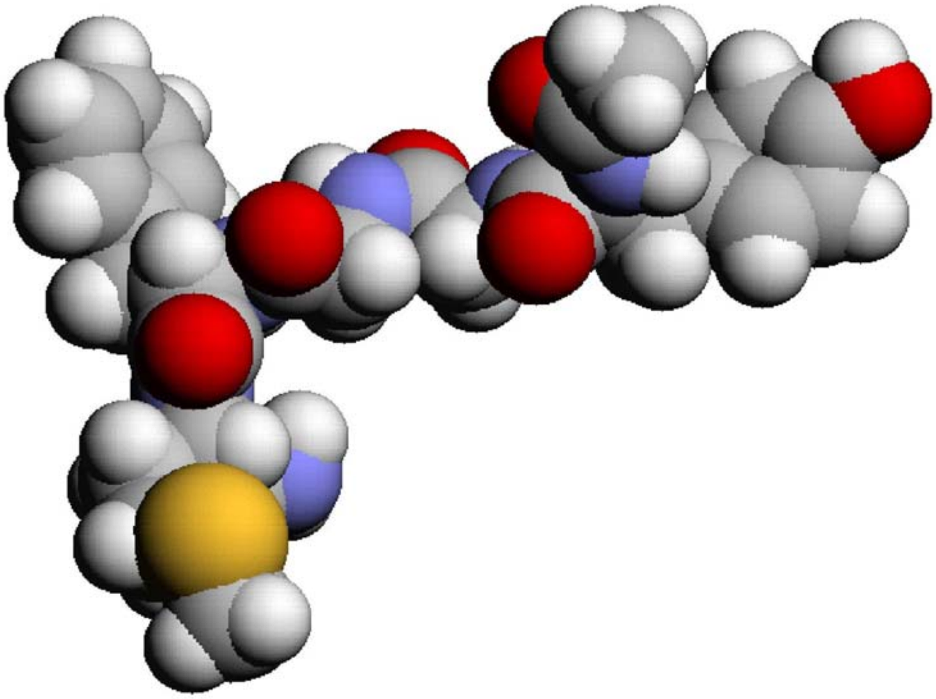

Figure 1: RASMOL plot of met-enkephalin

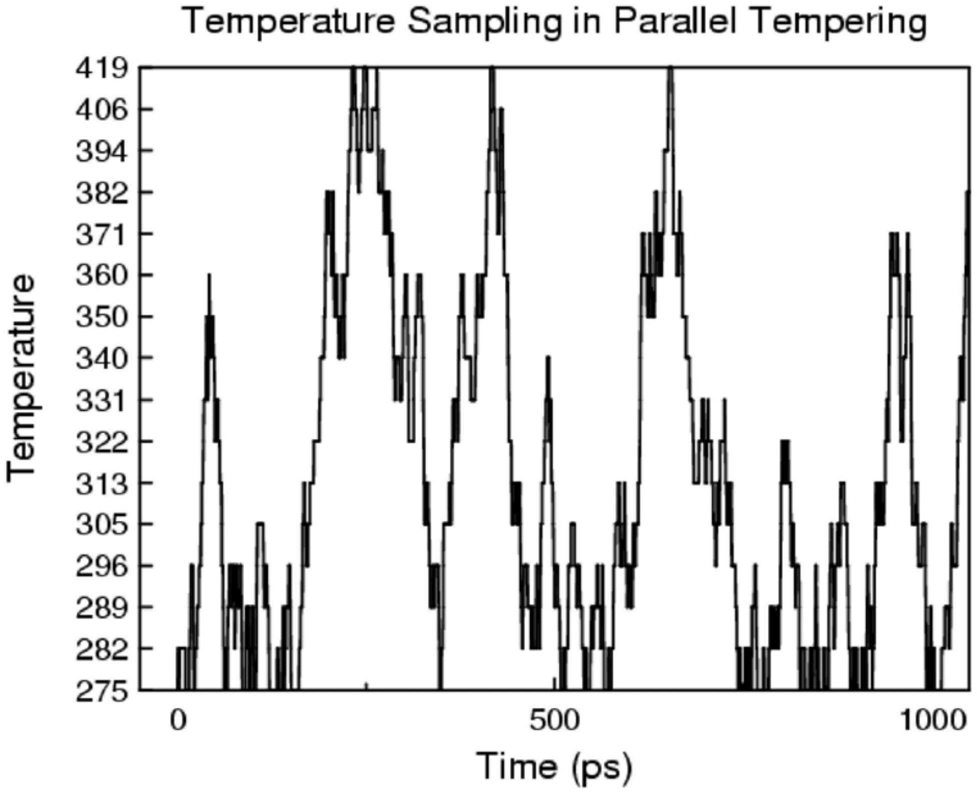

Figure 2: Temperature of a replica during simulation to 1000 ps

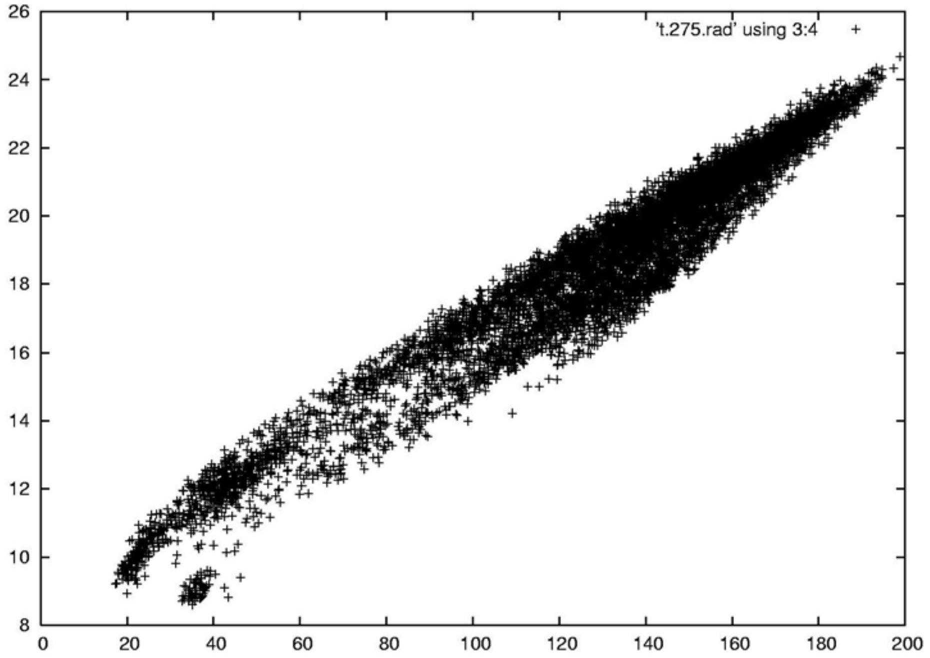

Figure 3: Distribution of radius of gyration R_g^2 versus end-end distance R_e^2 for configurations at 275K

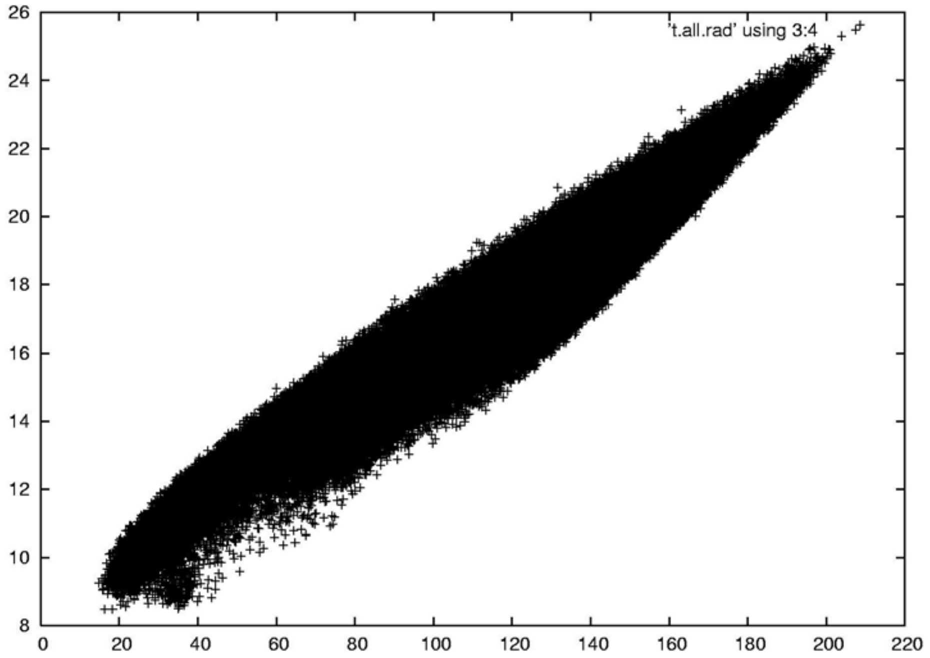

Figure 4: Same as Figure 3, but for configurations at all temperatures

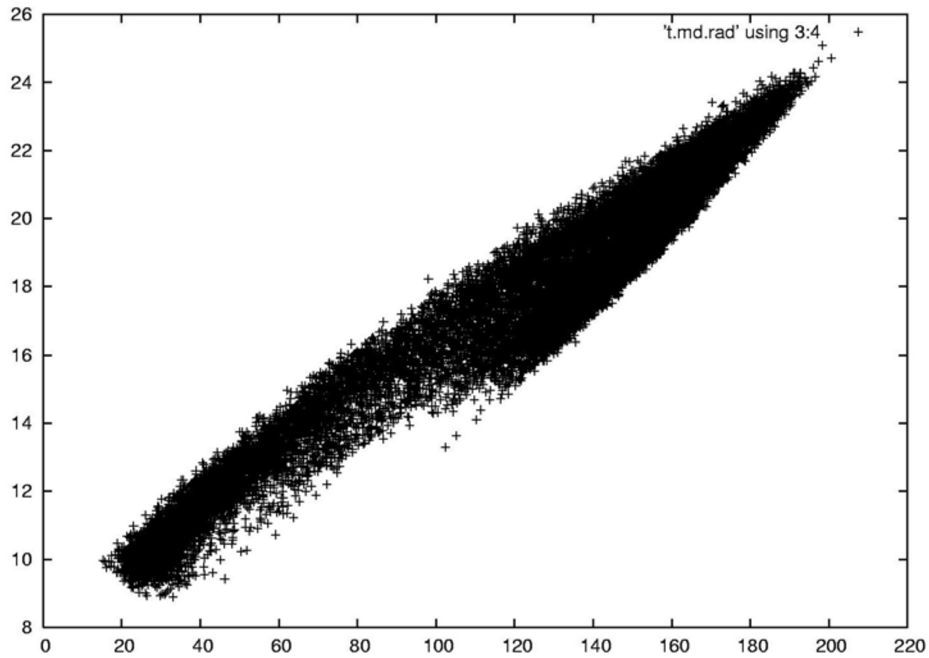

Figure 5: Same as Fig. 3 and Fig. 4, but using configurations from MD simulation

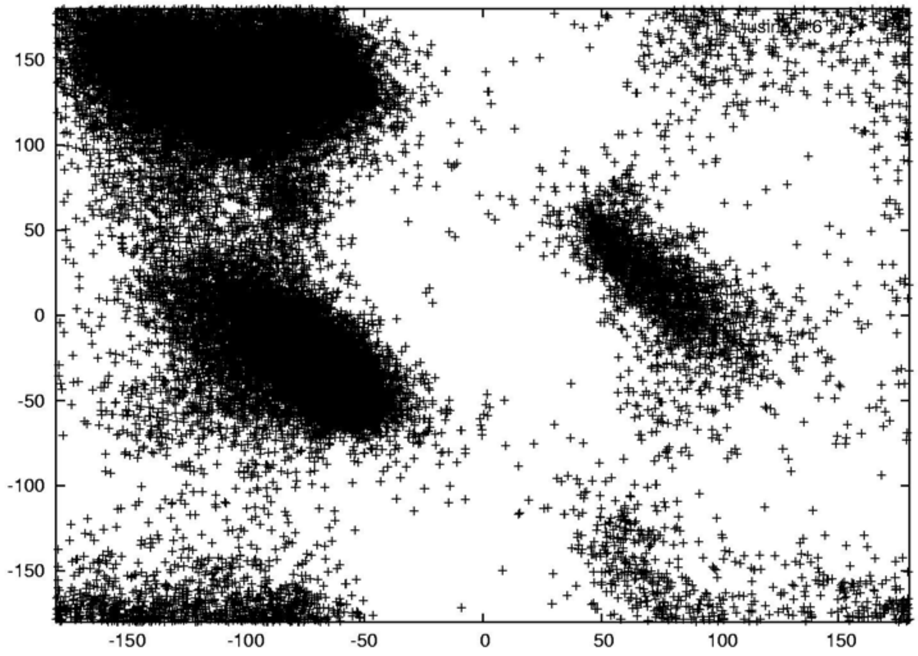

Figure 6: Ramachandran plot for 150 samples from the PDB

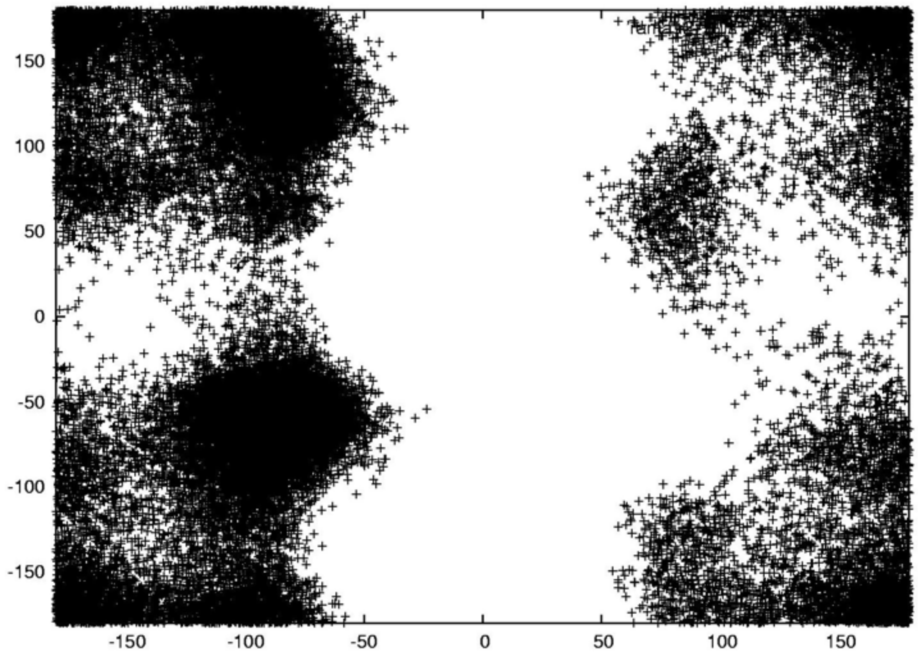

Figure 7: Ramachandran plot for configurations at 275K (all residues)

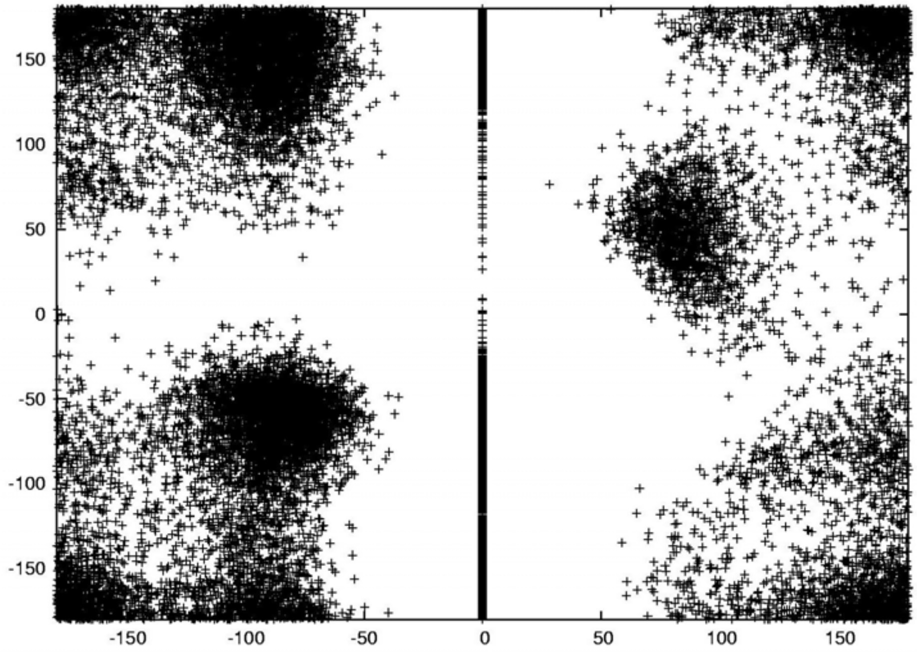

Figure 8: Ramachandran plot for MD run (all residues)

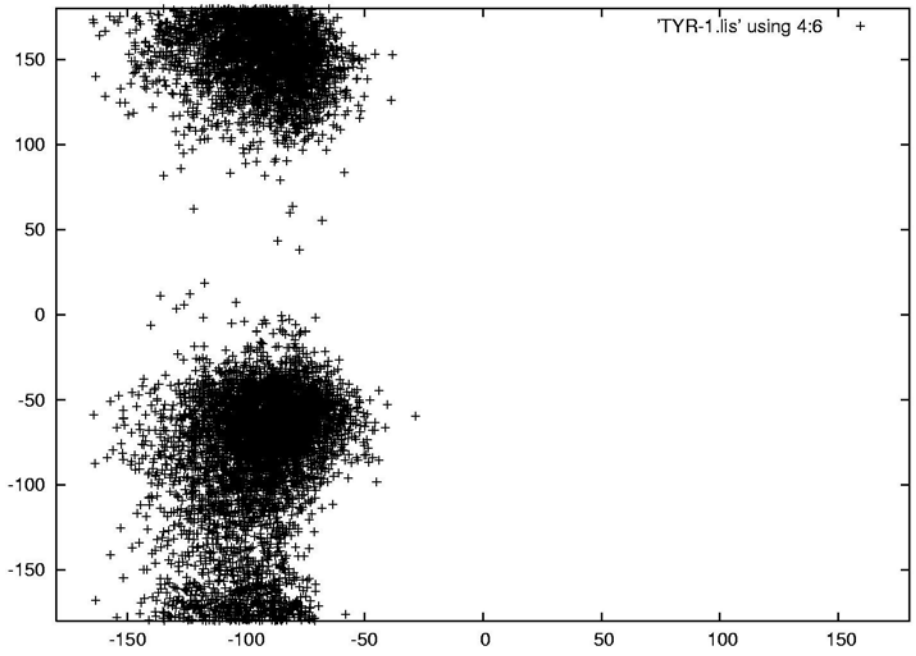

Figure 9: Ramachandran plot TYR-1 residue.

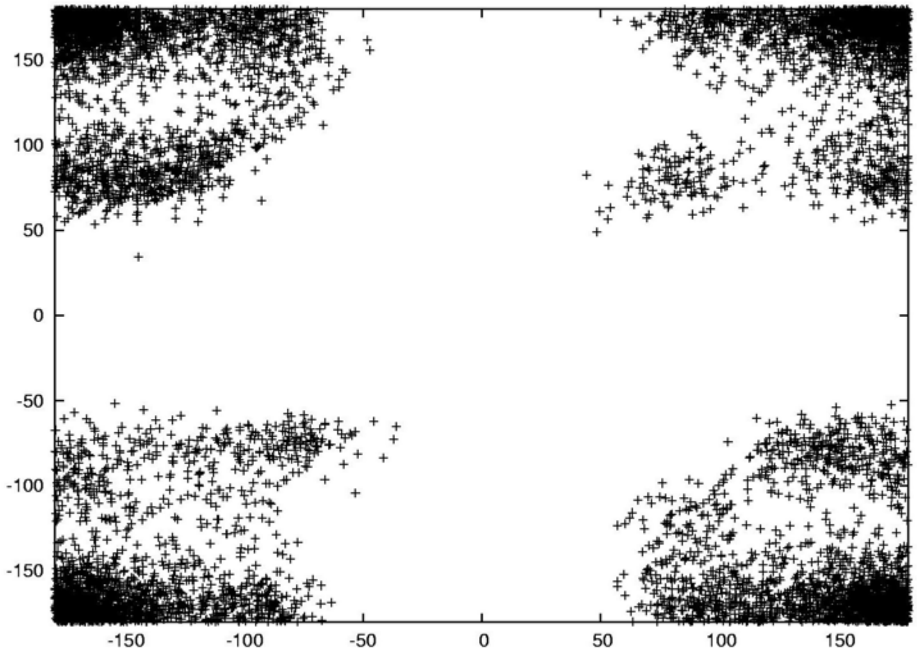

Figure 10: Ramachandran plot GLY-1 residue

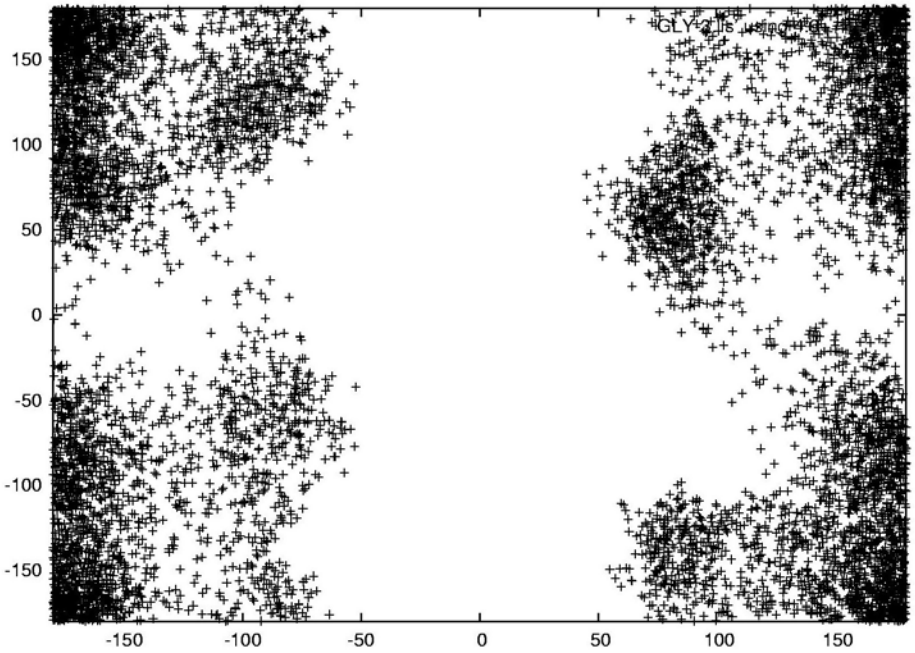

Figure 11: Ramachandran plot GLY-3 residue

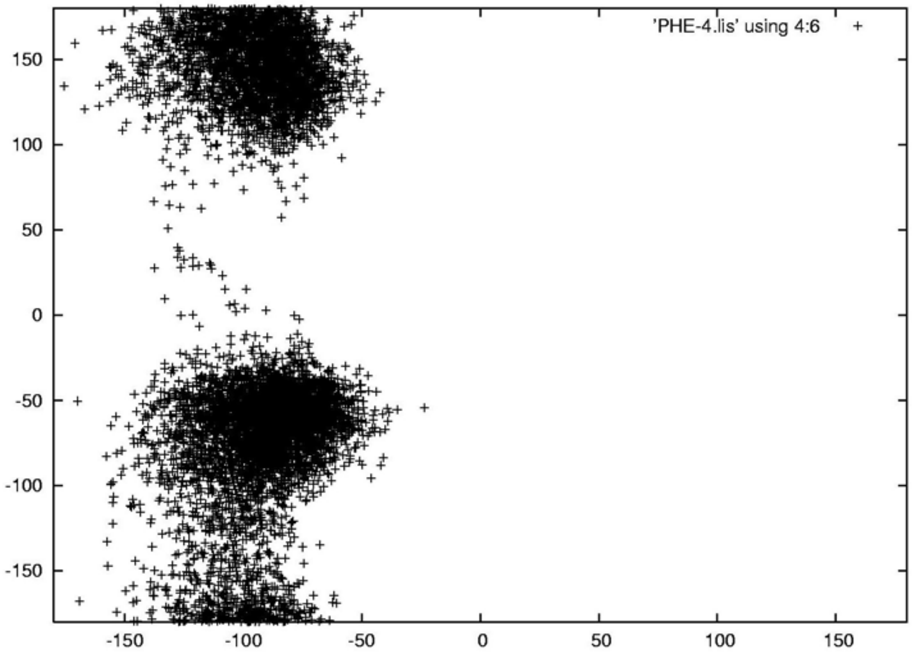

Figure 12: Ramachandran plot PHE-4 residue

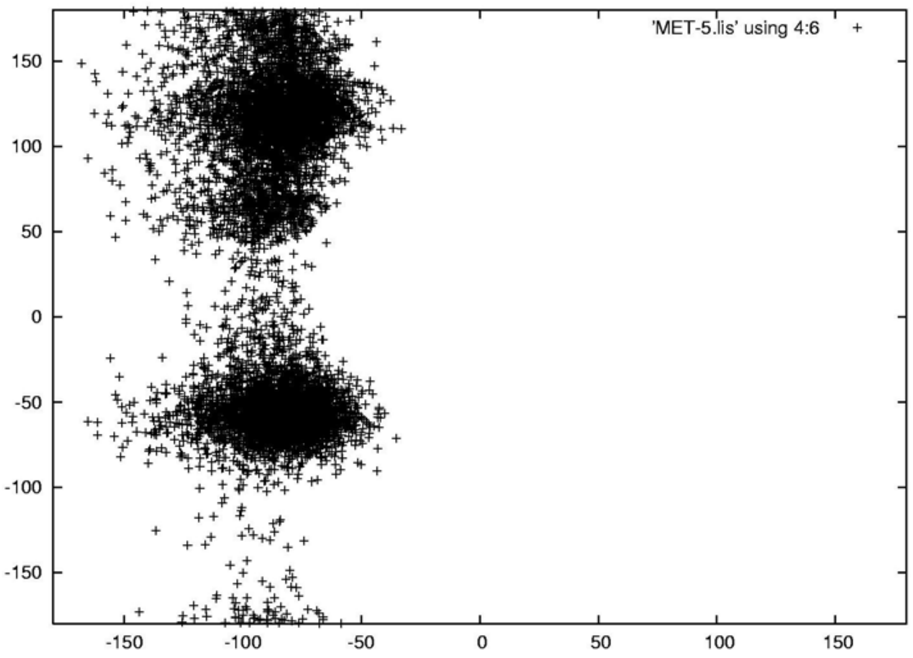

Figure 13: Ramachandran plot MET-5 residue

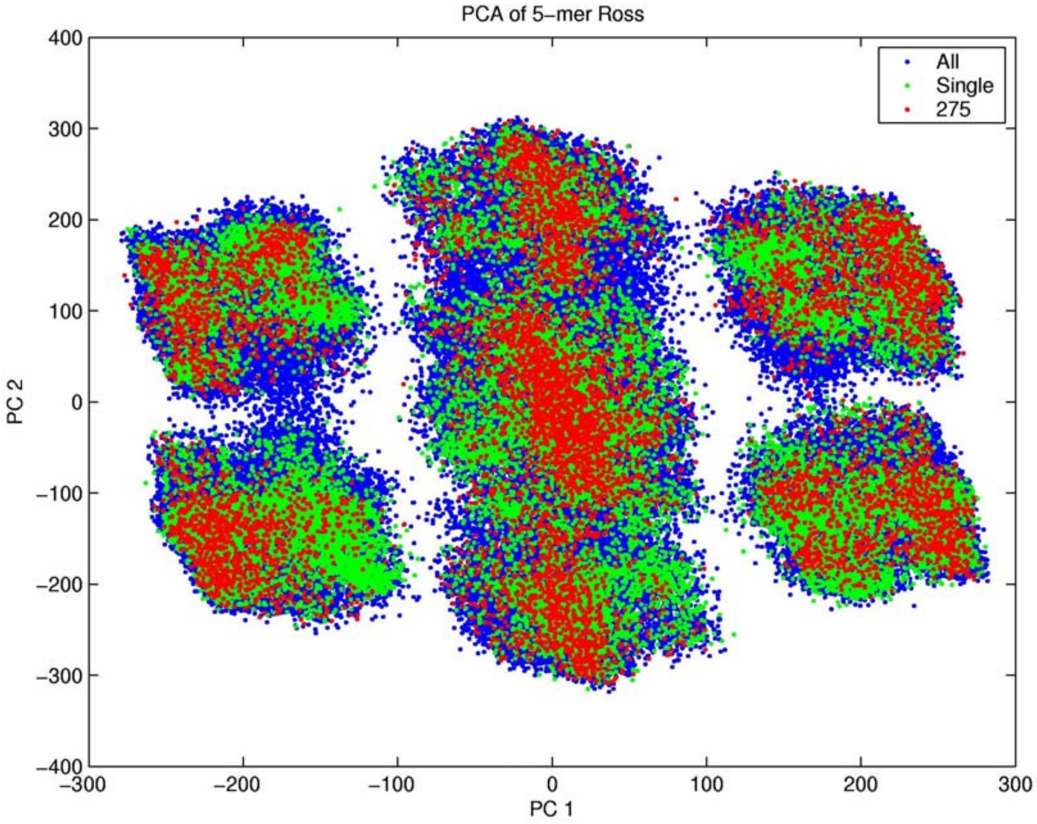

Figure 14: Principal component analysis results

Issues

During the course of this work we identified a number of issues.

First of all, the problem of generating the input configurations and analyzing the output for LAMMPS becomes much larger for PTMC runs. Debugging the test problem turned out to be a far larger problem than the actual code development for the parallel tempering capability.

There are some apparent difficulties in force fields for molecular systems. LAMMPS has primarily been used for simulations of model systems such as polymer melts and lipid membrane systems that are not intended to be careful detailed representations of reality. Although LAMMPS will run systems for example that are described by the CHARMM force field, LAMMPS does not have any facilities to create or manage such simulations.

Analysis of the results has turned out to be problematic. We are not convinced that the simulations are correct at this point, as pointed out previously in the discussion of the Ramachandran plots for the $\phi - \psi$ angle distributions. We do not yet have a good technique for clustering the results and it may be that there are no significant clusters for the chosen problem. This is certainly the conclusion that we obtained from the R_g vs. R_e analysis. We are currently obtaining results from some peptide chains that form loops, helices, and β-sheet configurations, which we expect to show much more distinct "folded" states. Another possible idea that we have not yet explored is to classify configurations based on a contact map. A folded state is well defined by such a map, whereas an open state has a very sparse map and the collapsed but not folded states have indefinite maps.

Summary

There are some other possible ways to utilize PTMC that we have not examined. For example, the method can be used with any parameter in the probability function. Pressure or volume are some obvious examples, although a force field parameter would also work. Thus one can imagine a simulation for an array of temperatures and pressures that would compute a complete phase diagram for a material in one go.

As mentioned above the umbrella sampling technique might be applied to improve the use of the state samples that are generated. In this approach we use the configurations at nearby temperatures as samples for a different temperature. The other samples must be reweighted to give them the proper probability for the different temperature, however, and this can be difficult.

Finally we note that in the case of protein folding we can use PTMC to look both at low lying (folded) states as well as the generation of pathways for getting to folded states.

[1] Steve Plimpton. `LAMMPS`, a parallel molecular dynamics code. See `http://www.cs.sandia.gov/~sjplimp/docs/lammps/README.html`

[2] E. Marinari and G. Parisi, "Simulated Tempering, a New Monte Carlo Scheme". *Europhysics Lett.*, 19(6), pp. 451-458 (1992)

[3] Bernd A. Berg and Thomas Neuhaus. "Multicanonical Ensemble: A new Approach to Simulate First Order Phase Transitions". *Phys. Rev. Lett.*, 68(1), pp. 9-12 (1992)

[4] A. P. Lyubartsev, A. A. Martsinovski, S. V. Shevkunov, and P. N. Vorontsov-Velyaminov, "New Approach to Monte Carlo Calculation of the Free Energy: Method of Expanded Ensembles". *J. Chem. Phys.*, 96(3), pp. 1776-1783 (1992)

[5] Koji Hukushima and Koji Nemoto, "Exchange Monte Carlo Method and Application to Spin Glass Simulations". *J. Phys. Soc. Jap.*, 65(6), pp. 1604-1608 (1996)

[6] Ulrich H. E. Hansmann, "Parallel Tempering Algorithm for Conformational Studies of Biological Molecules". *Chem. Phys. Lett.*, 281, pp. 140-150 (1997)

[7] K. Y. Sanbonmatsu and A. E. Garcia, "Structure of Met-enkephalin in Explicit Aqueous Solution Using Replica Exchange Molecular Dynamics". *Proteins*, 46, pp. 225-234 (2002)

[8] Marcus Martin, `TOWHEE` Computer Code. See `http://www.cs.sandia.gov/projects/towhee/index.html`

[9] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, Jr., D. M. Ferguson, D. C. Sellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman. *J. Am. Chem. Soc.*, 117, p. 5179 (1995)

[10] F. A. Momany and R. Rone, "Validation of the General Purpose quanta(r)3.2/charmm(r) Force Field". *J. Comp. Chem*, 13(7), pp. 888-900 (1992)

[11] V. Barone, G. Capecchi, Y. Brunel, M.-L. D. Andries, and R. Subra, "Development and Validation of Force-Field Parameters for Molecular Simulations of Peptides and Proteins Containing Open-shell Residues. *J. Comp. Chem.* 18(14), pp. 1720-1728 (1997)

Distribution:

5	MS	0310	Mark Rintoul, 9212
5		0310	Steve Plimpton, 9220
1		0310	Mike Brown, 9212
1		0316	John Aidun, 9235
1		0316	Paul Crozier, 9235
1		0316	Sudip Dosanjh, 9233
1		0316	Laura Frink, 9212
1		0316	Marcus Martin, 9235
5		0316	Mark Sears, 9235
1		0316	Alex Slepoy, 9235
1		0316	Aidan Thompson
1		0318	Paul Yarrington, 9230
1		1411	Amalie Frischknecht, 1834
1		1411	Mark Stevens, 1834
1		9951	Joe Schoeniger, 8130
1		9951	Malin Young, 8130
1		9018	Central Technical Files, 8945-1
2		0899	Technical Library, 9616