Reducing the Time Complexity of Hybrid Monte Carlo Simulation through Randomized Selection

Brian Thorndyke and Paul Fishwick

Department of Computer and Information Sciences and Engineering

University of Florida

301 CSE, P.O. Box 116120, Gainesville, FL 32611

e-mail: thorndyb@cise.ufl.edu

February 10, 1997

Abstract

Hybrid Monte Carlo provides an efficient means of sampling from the canonical ensemble, by using dynamical methods to propose transition states, and accepting these states based on Metropolis acceptance rules. The dynamical methods reduce the random walk associated with pure Monte Carlo, while the acceptance rules ensure the accuracy of the simulation. In this article, the dynamics of hybrid Monte Carlo are modified by the introduction of "randomized selection," where only subsets of particles are updated each dynamics timestep. It is shown that randomized selection can improve the efficiency of the simulation, provided there is coupling between the Hamiltionian variables. For a system of strongly coupled harmonic oscillators, the efficiency is seen to increase by up to 150%. Even in the case of a weakly-coupled system of Lennard-Jones particles, there is an increase in efficiency of up to 25%. This indicates that randomized selection can be used effectively not only on systems with short-range potentials, but also on those with long-range (for example, Coulombic) interactions.

Keywords: *Molecular Dynamics, Hybrid Monte Carlo, Particle Simulation, Randomized Selection*

1 Introduction

Particle science is the study of physical systems which can be described by particles and their interactions. These systems range from the microscopic (e.g. colloidal suspensions) to the astronomical (e.g. galactic formation), and involve extremely large numbers of particles. Their experimental and theoretical predictions are typically macroscopic, in that measured quantities involve averages over the states of the entire system. For this reason, particle simulations require large numbers of particle interactions to reproduce physically interesting quantities. New algorithms or improvements to existing algorithms, which decrease the computational complexity of particle simulations, are very important to the particle simulation community.

One class of algorithms uses timesteps to advance the system according to Newton's laws of motion. This method is known as molecular dynamics (MD). While the discretization of the equations of motion leads to imprecision in the particle paths, results for short time periods accurately reflect the dynamics of the system. Furthermore, by periodically adjusting the positions and momenta of the particles, long-term system averages can be observed. The major drawback of MD is the large amount of time it takes to accurately represent long-term statistics. Major factors include the number of timesteps necessary for good statistics, as well as the complexity of each timestep.

Another approach is to sample the states of the system stochastically, and derive the statistical measurements directly. This procedure falls under the broad category of Monte

Carlo (MC) simulation. Particle simulations usually involve a specific kind of MC, developed by Metropolis *et. al* [1] several decades ago, which uses *a priori* knowledge of the relative state distribution (e.g. the Boltzmann distribution). Metropolis Monte Carlo (MMC) has the benefit that each step has at most linear complexity in the number of particles, and can incorporate heuristics specific to the problem domain to reduce the total number of steps required for accurate results. A principal disadvantage is that dynamical information is lost, but as far as retrieving statistical information about particle systems, MMC has proven to be very effective.

Over the past few years, hybrid models have emerged, in which the properties of MC are combined with those of MD [3]. Guarnieri and Still [4] developed an alternating MMC / MD scheme, which was found to sample a system of n-pentane molecules significantly faster than MMC or MD alone. Clamp *et al.* [5] combined MMC and MD into a hybrid Monte Carlo (HMC) algorithm, and found that the HMC sampled a system of Lennard-Jonesium an order of magnitude faster than pure molecular dynamics.

Although these algorithms have been shown to work very well, it is not at all clear that they represent optimal combinations of dynamic and stochastic moves. In this paper, we introduce "randomized selection" to the dynamics of HMC (henceforth referred to as "randomized hybrid Monte Carlo", or RHMC), in an effort to further increase the efficiency of the algorithm, and to reduce its computational complexity. Over the last decade, there has been much literature concerning the use of randomized algorithms in other areas of computer

science. These algorithms make decisions based on random number generation, and have proven to be substantially quicker than their sequential counterparts in areas such as sorting and classification [6]. We carry the concept of randomized selection to HMC, in the form of randomized particle selection [7]. Rather than update the entire set of positions and momenta each dynamic run, we choose to update random subsets of these variables. Although this restricts the state space volume accessible to the system for any given step, uniform selection of the subset elements ensures that the volume is explored without bias.

In Section 2, we examine some technical details of particle simulation, with emphasis on MD and MC. In Section 3 we explore the HMC algorithm. In Section 4 we introduce our modified HMC algorithm, and demonstrate that the randomized version generates correct statistics. In Section 5 we examine the effect of the coupling between state variables on the efficiency of the randomized scheme. Using a test system of coupled harmonic oscillators, the qualitative exploration of state space by randomized dynamics is compared to that obtained by non-randomized dynamics. These results are followed by quantitative comparisons of efficiency, derived from long RHMC simulations of the oscillator system in the canonical ensemble. Finally, RHMC simulations of Lennard-Jonesium are presented in Section 6, and the paper is concluded.

2 Particle Science Simulations

Typical particle simulations model a physical system as point (i.e. 0-dimension) particles. Each of these particles contributes to the potential field, through some function of distance. For example, charged particles have a Coulomb potential which varies as $1/r$, so that the potential contribution decreases with distance. This is the case with all real systems, but the rate at which the potential decreases varies greatly depending on the particles and their media.

Each particle responds to the potential field by accelerating in proportion to the gradient of the field (the force). As the particles change position, the field is continuously updated. This interplay of acceleration and force gives rise to Newton's second law ($\vec{F} = m\vec{a}$) or equivalently, Hamilton's equations of motion (described in the next section). At any point in time, the system's state is described by the $3N$ position coordinates and $3N$ momenta coordinates of all the particles, where N is the number of particles in the system. It is convenient to think of the system as moving through *phase space*, formed by the cross product of all position and momenta axes. In this case, the properties of the system are completely specified by the movement of the system through phase space.

Most experimental quantities are macroscopic, involving phenomena like temperature, pressure and volume. With these quantities, the dynamical path of the system is not of interest, but rather the statistical configuration of the particles. Rather than discuss the paths in phase space, we consider the *probability distribution* of the points in phase space.

If this distribution is known, any macroscopic property can be calculated. It turns out that if a system is in thermal equilibrium with its surrounding, it is said to be in the *canonical ensemble*, and its phase space is distributed according to the Boltzmann distribution (to be described in more detail in the next section).

If a system is described by the canonical ensemble, a popular method of exploring its phase space is through MMC. The MMC algorithm moves the system through phase space by randomly perturbing the system variables, and then accepting or rejecting these moves based on the Boltzmann distribution. As long as the sampling is ergodic (so that any two states can eventually be connected through a series of transitions), MMC is guaranteed to converge to the proper canonical distribution. However, because the sampling is essentially a random walk through phase space, its expected distance after N steps is $\leq \sqrt{N}$. As a result, it can take a long time for distant areas of phase space to be sampled, and convergence can be extremely slow.

An alternative approach, stochastic dynamics (SD), moves a system dynamically through phase space according to Hamilton's equations of motion, as in MD, but occasionally resampling the velocities according to a Gaussian distribution. The random walk behavior of MMC is avoided, but the discretization of Hamilton's equations introduces an intrinsic error to the simulation ($\bigcirc(\epsilon^2)$ in the case of leap-frog discretization, where ϵ is the size of the time step) [2]. Care must be taken to use time steps which are small enough to ensure stability and accuracy, which inevitably slows convergence.

3 Hybrid Monte Carlo

In the canonical ensemble, the states $\{\Gamma\}$ of a system S are distributed according to the Boltzmann probability distribution [8],

$$\varrho(\Gamma) \;=\; \mathcal{Z}^{-1}\exp(-\beta H(\Gamma)), \tag{1}$$

where $\beta = (k_B T)^{-1}$, H is the Hamiltonian, and $\mathcal{Z}$ is the partition function,

$$\mathcal{Z} \;=\; \int_{\Gamma\in S}\exp(-\beta H(\Gamma))d\Gamma. \tag{2}$$

In order to sample from this ensemble, the probability distribution must be taken into account. This is the idea behind the MMC algorithm. A random walk is made through phase space, and the steps are accepted or rejected in a manner consistent with the Boltzmann distribution.

However, for a given energy, there may be a hyperplane of states, all of which are equally accessible. Unfortunately, the random walk nature of MMC doesn't accomodate motion along such a hyperplane. A better way of sampling these regions is by specifying the system in terms of its independent variables $\vec{q}$ and their conjugates $\vec{p}$ (e.g. position coordinates and momenta). The hyperplane can be dynamically explored by following Hamilton's equations of motion [9],

$$\frac{\partial q_i}{\partial t} \;=\; \frac{\partial H}{\partial p_i}, \tag{3}$$

$$\frac{\partial p_i}{\partial t} \;=\; -\frac{\partial H}{\partial q_i}. \tag{4}$$

Equations (3) and (4) conserve energy, keeping the system on the same hyperplane. For example, if a system has N independent coordinate variables, we have

$$\frac{\partial H}{\partial t} = \sum_{i=1}^{N} \left(\frac{\partial H}{\partial p_i} \frac{\partial p_i}{\partial t} + \frac{\partial H}{\partial q_i} \frac{\partial q_i}{\partial t} \right) \tag{5}$$

$$= \sum_{i=1}^{N} \left(\frac{\partial q_i}{\partial t} \frac{\partial p_i}{\partial t} - \frac{\partial p_i}{\partial t} \frac{\partial q_i}{\partial t} \right) \tag{6}$$

$$= 0. \tag{7}$$

In practice, Hamilton's equations must be cast in a discrete form, using a small timestep to advance the motion of the system. In the limit of infinitely small time steps, the system would not deviate from the hyperplane, but stay at constant energy. However, the discretization leads to unavoidable error proportional to the size of the timestep (or some power thereof), and to the number of dynamic steps taken. In order to ensure stability, the timesteps must remain below a certain threshold, and the momenta must be periodically rescaled to restore the original energy.

HMC introduces this dynamic motion through phase space, within the context of MMC. At the beginning of each HMC cycle, the momenta coordinates are redistributed according to a Gaussian distribution consistent with the temperature of the system. This moves the system to a state $\Gamma = \Gamma(\vec{q}, \vec{p})$ on the hyperplane of some energy $H(\Gamma)$. At this point, a series of dynamic steps are taken, either forward or backward in time with equal probability (the reason for this to be explained in the next section), arriving at state $\Gamma' = \Gamma(\vec{q'}, \vec{p'})$ with energy $H(\Gamma')$. The accumulated error ($\Delta H = H(\Gamma') - H(\Gamma)$) is kept in check by accepting the final positions and momenta according to the Metropolis acceptance rule: the run is accepted with

a probability

$$\mathcal{P}(\text{accept}) \quad = \quad \min(1, \exp(-\beta \Delta H)). \tag{8}$$

If the run is not accepted, the positions and momenta at the beginning of the run $(\vec{q}, \vec{p})$ are restored. Figure 1 illustrates the difference between MMC and HMC.

HMC is a variant of MMC, and will sample the canonical ensemble correctly provided that *detailed balance* is maintained. In the next section, we will show explicitly that detailed balance is satisfied for our RHMC. However, we refer the reader to Neal[2] for a proof that the original HMC algorithm satisfies this requirement.

4 Randomized HMC

4.1 Algorithm

We modify the HMC algorithm by randomly selecting subsets of variables to be updated during the dynamic run. All momenta are redistributed, as in the original version, but only a subset S_D of position variables (and their conjugate momenta) are changed according to Hamilton's equations. This is equivalent to replacing the stationary variables in the Hamiltonian by constants, equal to the variables' values after redistribution of momenta, say at time $t = t_x$:

$$\forall i \notin S_D, \quad q_i(t + t_x) \quad = \quad q_i(t_x), \tag{9}$$

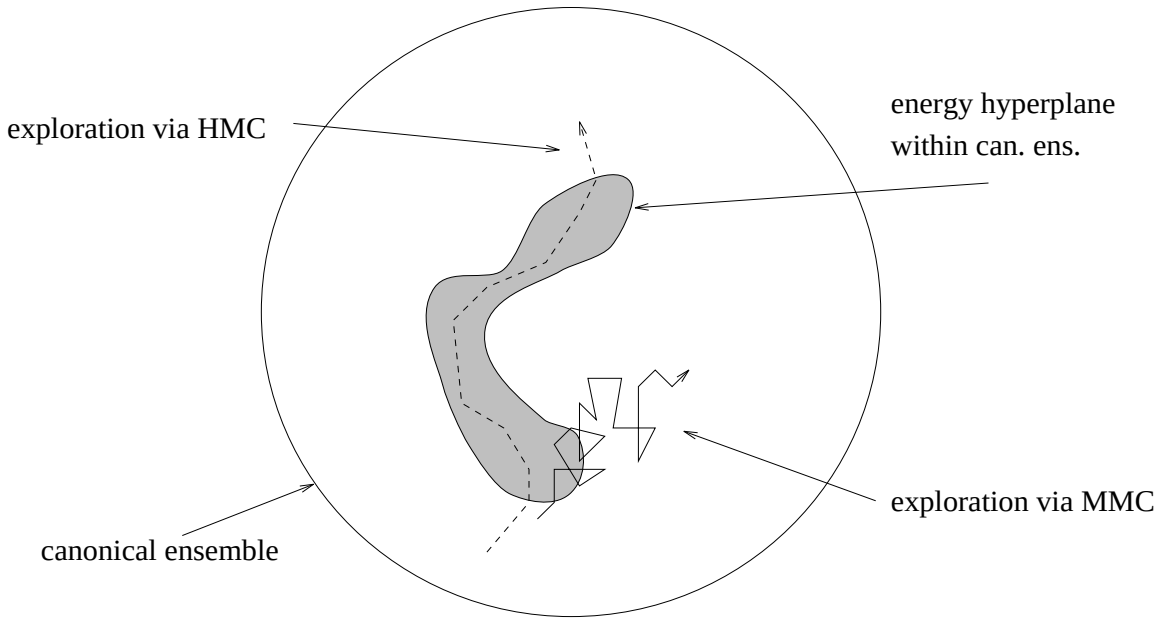

Figure 1: The difference between MMC (solid line) and HMC (dashed line). Grey area represents an energy hyperplane within the canonical ensemble. MMC explores the canonical ensemble via a random walk, ignoring such hyperplanes. HMC exploits the hyperplanes by moving across them using MD, thereby reducing the random walk.

$$p_i(t + t_x) \;=\; p_i(t_x). \tag{10}$$

Eqns. (5)-(7) are still valid for this modified Hamiltonian. As such, the equations of motion will move the system through phase space along a reduced set of axes, and sample a projection of the full energy hyperplane. However, if the variables in S_D are chosen randomly each RHMC cycle, the hyperplanes will be explored without bias.

Here is the randomized HMC pseudo-code:

```
while (simulation not finished) {
   redistribute all momenta;
   calculate energy H0;
   choose subset Sd of variables;
   for (length of dynamics run) {
      for (each variable in Sd)
         update according to Hamilton´s equations, with modified Hamiltonian;
   };
   calculate energy H1;
   accept with probability(-(H1-H0)/kT) {
      case ´accept´:  keep new variables
      case ´reject´:  restore old variables before dynamics run
   };
}
```

If the variables are coupled (so that updating a single variable depends on many or all other variables), updating their values each dynamics step is the most time consuming task of the algorithm. RHMC cuts the complexity of the dynamics run by requiring fewer updates each cycle. If only $\rho\%$ of the particles are moved each run, then we can expect an asymptotic time savings of $(100 - \rho)\%$.

4.2 Detailed balance

Although RHMC conserves energy during the dynamical moves, we must show that the algorithm leaves the Boltzmann distribution invariant. In other words, we need to satisfy detailed balance between any two regions of phase space $d\Gamma$ and $d\Gamma'$,

$$\mathcal{P}(d\Gamma \to d\Gamma') \quad = \quad \mathcal{P}(d\Gamma' \to d\Gamma). \tag{11}$$

The RHMC algorithm can be broken into two stages: momenta redistribution and dynamics run. Each stage can be seen to satisfy detailed balance separately. In the canonical ensemble, the momenta are distributed according to a Gaussian distribution. When the momenta are redistributed at the beginning of each RHMC cycle, they are assigned a Gaussian distribution once again. As such, detailed balance holds trivially for the first stage of RHMC.

At the beginning of the second stage, assume the system finds itself in an infinitesimal region $d\Gamma$ with Boltzmann probability density $\varrho(\Gamma) = \mathcal{Z}^{-1} \exp(-\beta H(\Gamma))$. A subset S_D of the total N variables is chosen, each variable being an equally likely candidate. Denoting the

cardinality of S_D by $|S_D|$, we find the probability of any subset to be

$$\mathcal{P}(S_D) \;=\; \binom{N}{|S_D|}. \tag{12}$$

At this point, the direction of time is chosen randomly, and a series of MD steps take the system from $d\Gamma$ to a new region $d\Gamma'$. If the dynamic motion follows Hamilton's equations, Liouville's theorem [10] asserts that the volume of phase space will remain constant, $d\Gamma' = d\Gamma$. Finally, the new configuration is accepted according to equation (8), giving

$$\begin{aligned}
\mathcal{P}(d\Gamma \to d\Gamma') &= d\Gamma \mathcal{Z}^{-1} \exp(-\beta H(\Gamma)) \binom{N}{|S_D|} \cdot \frac{1}{2} \cdot \min\left(1, \exp(-\beta(H(\Gamma') - H(\Gamma))\right) \\
&= d\Gamma' \mathcal{Z}^{-1} \exp(-\beta H(\Gamma')) \binom{N}{|S_D|} \cdot \frac{1}{2} \cdot \min\left(1, \exp(-\beta(H(\Gamma) - H(\Gamma'))\right) \\
&= \mathcal{P}(d\Gamma' \to d\Gamma).
\end{aligned} \tag{13}$$

Both stages of RHMC satisfy detailed balance, and the Boltzmann distribution is left invariant.

4.3 Phase space exploration

Although RHMC cuts the computational complexity of the original algorithm, we must be concerned as well with its ability to effectively explore phase space. If the randomized procedure demands proportionally more samples to represent the system at equilibrium, then we will have gained nothing overall. However, if the number of required samples remains unchanged, our savings in computational complexity translates directly into increased efficiency. For reasons to be seen later, coupling between the variables has a crucial effect on

the overall efficiency of the algorithm. In order to better examine this effect, we turn to simulation.

5 Harmonic Oscillator Simulation

5.1 System details

Our test system consisted of 100 harmonic oscillators, with a variable coupling λ between the coordinate degrees of freedom:

$$H(\vec{q},\vec{p}) = \sum_{i=1}^{N}\left(\frac{q_i^2}{2\sigma_i^2}+\frac{p_i^2}{2}\right)+\lambda\sum_{i=1}^{N}\sum_{j=i+1}^{N}\left(\frac{q_iq_j}{\sigma_i\sigma_j}\right), \tag{14}$$

where σ_i is the standard deviation of the potential well of oscillator i, in the absence of coupling terms. The standard deviations were randomly assigned values distributed uniformly between 0.2 and 1.2. The entire system was taken to have a temperature of 1.0, so that the momenta were distributed according to Gaussians with mean $\overline{p}_i = 0$, and standard deviation $\sigma_{p_i} = 1.0$.

5.2 Qualitative view

Consider the case where $\lambda = 0$. Hamilton's equations become

$$\frac{\partial q_i}{\partial t} = p_i, \tag{15}$$

$$\frac{\partial p_i}{\partial t} = -\frac{q_i}{\sigma_i^2}. \tag{16}$$

Each conjugate pair of variables is updated independently of the other variables. On the other hand, with λ non-zero, Hamilton's equations become

$$\frac{\partial q_i}{\partial t} = p_i, \tag{17}$$

$$\frac{\partial p_i}{\partial t} = -\frac{q_i}{\sigma_i^2} - \lambda \frac{q_i}{\sigma_i} \sum_{j \neq i} \frac{q_j}{\sigma_j}, \tag{18}$$

and each variable's evolution depends on every other variable.

The exploration of an energy hyperplane can be expected to proceed in markedly different ways, depending on the value of λ. Without any coupling, a single timestep moves each pair of conjugate variables a single independent step along their path. If only the subset S_D is updated, it will take $N/|S_D|$ as long to move the entire set of variables over the same space. The paths will be essentially identical, and the time saved by updating only partial sets of variables will be cancelled by the greater number of steps required for similar results. If, on the other hand, $\lambda > 0$, the paths will vary substantially, and it is not clear that the path updating all variables will reach independent points in phase space faster than the path updating only partial variables.

In order to demonstrate these dependencies, we simulated the dynamical trajectories of the oscillator system. In order to visualize the whole of phase space, we plotted the sum of coordinate variables against the sum of momentum variables as the simulation progressed. The results are shown in Figures 2 and 3. We see that when the coupling is absent ($\lambda = 0$), the simulation with randomized dynamics ($|S_D| = 50$) follows roughly the same path as the non-randomized dynamics ($|S_D| = 100$), but requires twice as many timesteps to reach

the same point. However, when the coupling is present ($\lambda > 0$), the paths are very much different, and span the same region after equal number of steps. The only difference is that the randomized path requires half the computational time (asymptotically) as the original code.

5.3 Sampling efficiency

A system can be described by its potential energy distribution. In the canonical ensemble, this distribution is a Gaussian, with characteristic mean $< U >$ and variance $< \delta U^2 >$. With n experimental data points, one only has access to the *population* mean ($< \delta U >_n$) and *population* variance ($< \delta U^2 >_n$), where

$$< U >_n = \frac{1}{n} \sum_{i=1}^{n} U_i, \tag{19}$$

$$\text{and } < \delta U^2 >_n = \frac{1}{n} \sum_{i=1}^{n} (U_i - < U >)^2. \tag{20}$$

If the data points are independent, the variance of $< U >_n$ is inversely proportional to the number of sample points, and similarly for the variance of $< \delta U^2 >_n$. However, if there is a correlation between the points (as is common with a Markov chain, like RHMC), these variances in population statistics will increase proportionally to the length s of the correlation [8]:

$$\sigma^2_{<U>_n} = \frac{s}{n} < \delta U^2 >_n, \tag{21}$$

$$\text{and } \sigma^2_{<\delta U^2>_n} = \frac{s}{n} < \delta U^2 >_n^2 . \tag{22}$$

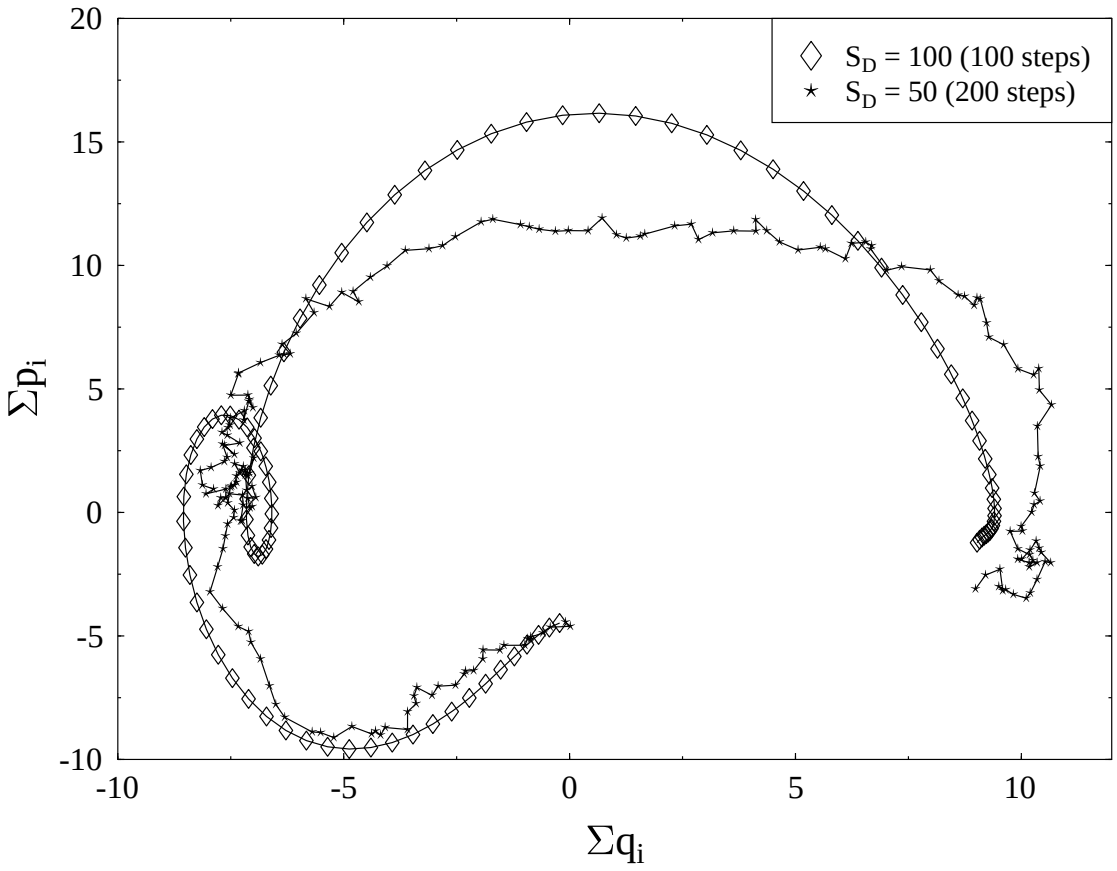

Figure 2: Qualitative comparison between randomized and non-randomized MD paths, for 100 harmonic oscillators with $\lambda = 0$: The randomized MD ($|S_D| = 50$) requires twice as many steps as the non-randomized MD ($|S_D| = 100$) to cover the same path.

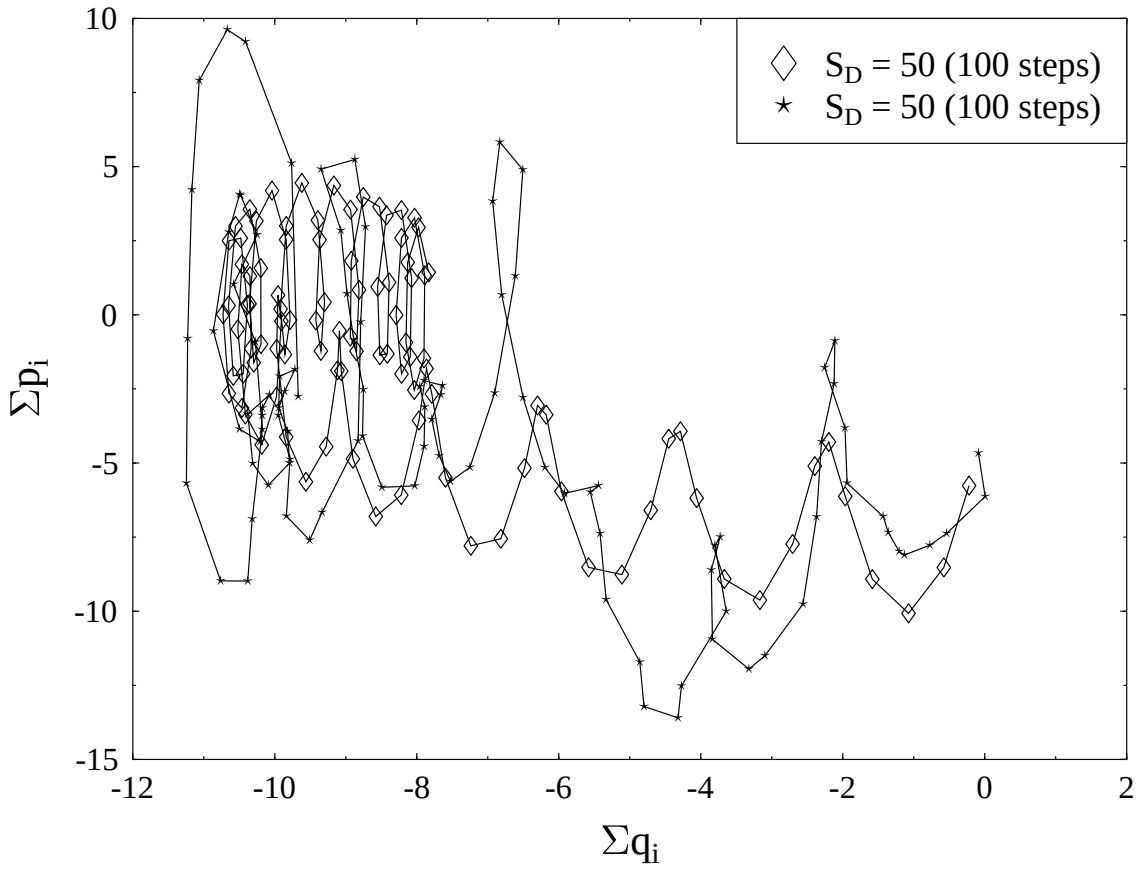

Figure 3: Qualitative comparison between randomized and non-randomized MD paths, for 100 harmonic oscillators with $\lambda = 0.9$: The randomized MD ($|S_D| = 50$) requires the same number of steps as the non-randomized MD ($|S_D| = 100$) to cover the same path.

To compare efficiencies of two simulations A and B, we proceed as follows. The correlation lengths of each simulation are determined, say s_A and s_B. Then A requires (s_A / s_B) as many steps as B to reach an independent point. However, each simulation may require different amounts of computation each step, say c_A and c_B. We define the relative efficiency of A to B as

$$\text{(rel. efficiency)} \quad = \quad \frac{c_A \cdot s_B}{c_B \cdot s_A}. \tag{23}$$

To compare the efficiency of RHMC and HMC, the harmonic oscillators were set in motion for 210,000 RHMC cycles. The first 10,000 were discarded as "burn-in" cycles, while the remaining 200,000 values contributed to the potential energy measurements. Several coupling strengths were used, the efficiency of RHMC (with $|S_D| = 50$) versus HMC ($|S_D| = 100$) being compared for each λ. To put the simulations on equal footing, the timestep for each value of λ was chosen such that the Metropolis acceptance rate was $\approx 50\%$. The results of the simulations are shown in Table 1 and Figure 4.

As expected, without any coupling, RHMC has no advantage over HMC. However, when the coupling is introduced, RHMC becomes more efficient than HMC. These results agree with the qualitative arguments of Section 5.2. Among the coupling strengths tested, $\lambda = 0.5$ provided the greatest increase in efficiency (approximately 150%). A possible explanation of the decrease in efficiency for $\lambda > 0.5$ may be the nature of the simulation - as λ approaches 1.0, the harmonic oscillator system becomes numerically instable. This is reflected by the increased deviation of the population mean and variance.

λ	$\lvert S_D \rvert$	$< U >$	$< \delta U^2 >$	Δt^*	$\%accept$	s
0.00	50	49.969 ± 0.096	7.06 ± 0.68	0.3668	51.9	37.0 ± 1.5
	100	49.912 ± 0.069	7.18 ± 0.50	0.3140	52.6	18.5 ± 0.8
0.05	50	50.025 ± 0.083	7.02 ± 0.58	0.3495	54.1	27.7 ± 1.1
	100	50.004 ± 0.071	7.11 ± 0.51	0.2958	53.7	20.0 ± 1.0
0.10	50	49.973 ± 0.086	7.04 ± 0.61	0.3225	52.1	30.2 ± 1.3
	100	49.972 ± 0.083	7.08 ± 0.59	0.2570	48.5	27.8 ± 1.3
0.50	50	50.064 ± 0.056	7.07 ± 0.39	0.1783	50.3	12.4 ± 0.9
	100	50.167 ± 0.063	7.10 ± 0.45	0.1272	51.5	15.7 ± 0.9
0.90	50	49.973 ± 0.087	7.09 ± 0.62	0.1356	50.5	30.0 ± 2.0
	100	49.958 ± 0.075	7.07 ± 0.53	0.0957	51.7	22.7 ± 1.4
0.95	50	49.910 ± 0.121	7.10 ± 0.86	0.1330	48.4	58 ± 4
	100	49.860 ± 0.109	7.11 ± 0.77	0.0925	54.5	47 ± 3

Table 1: Results from RHMC simulations of 100 coupled harmonic oscillators. Each simulation had an equilibrium period of 10,000 RHMC cycles, followed by 200,000 measured RHMC cycles. Each cycle contained 10 MD timesteps.

6 Lennard-Jonesium Simulation

6.1 Test system details

In order to evaluate RHMC under more common circumstances, we simulated 100 Lennard-Jones particles in a periodic box, with reduced density 0.8 and reduced temperature 1.0. Each simulation began with 10,000 RHMC "burn-in" cycles, followed by 200,000 measured RHMC cycles. Each cycle contained 10 MD steps, and the timestep was adjusted in each case to give an acceptance rate of $\approx 50\%$.

6.2 Sampling efficiency

Following the procedure used to analyse the harmonic oscillators, we measured the relative efficiency of each RHMC simulation ($|S_D| < 100$), to the HMC simulation ($|S_D| = 100$). The results are shown in Table 2 and Figure 5. We see that although the performance is poor for most subset sizes, the efficiency is increased somewhat for $|S_D| = 75$, where RHMC is about 1.25 times as efficient as HMC. This is reasonable, as the Lennard-Jones particles interact only at short distances, the majority of particles moving essentially independently of one another.

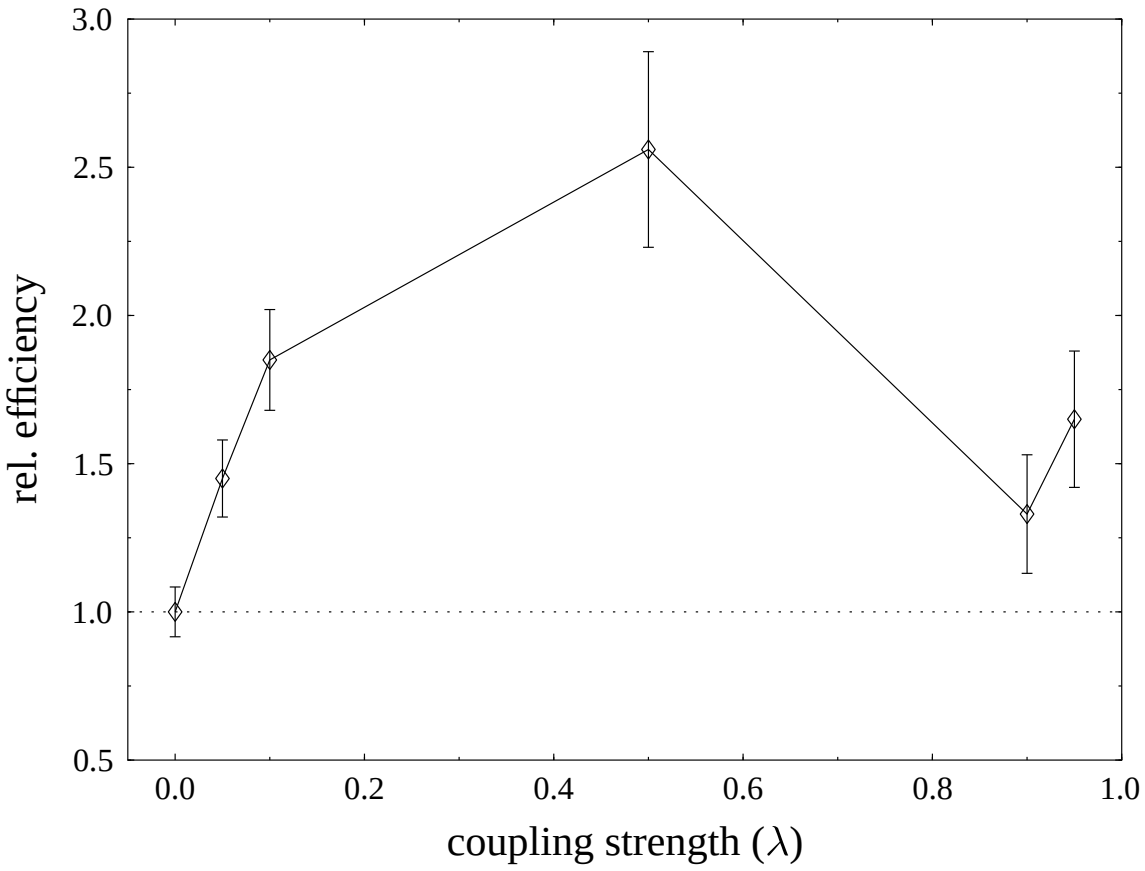

Figure 4: Relative efficiency of RHMC ($|S_D| = 50$) to HMC ($|S_D| = 100$), depending on λ. Without coupling ($\lambda = 0$), there is no increase in efficiency. With coupling ($\lambda > 0$) RHMC is significantly more efficient than HMC, up to 150% more efficient in the case of $\lambda = 0.50$.

$\|S_D\|$	$< U >$	$< \delta U^2 >$	Δt^*	$\%accept$	s
10	-217.75 ± 0.17	7.6 ± 1.3	0.0431	50.9	105 ± 10
25	-217.42 ± 0.10	7.53 ± 0.76	0.0345	52.5	36.0 ± 2.5
50	-217.528 ± 0.063	7.56 ± 0.47	0.0293	52.1	13.8 ± 0.8
75	-217.528 ± 0.046	7.51 ± 0.34	0.0266	52.4	7.4 ± 0.3
100	-217.542 ± 0.044	7.55 ± 0.33	0.0249	52.0	6.7 ± 0.3

Table 2: Results from RHMC simulations of 100 Lennard-Jonesium particles in a periodic box, with $\rho^* = 0.8$ and $T^* = 1.0$. Each simulation had an equilibrium period of 10,000 RHMC cycles, followed by 200,000 measured RHMC cycles. Each cycle contained 10 MD timesteps.

7 Discussion

The HMC algorithm has been found by other researchers to sample the canonical ensemble more efficiently than either MC or MD alone. We have extended HMC by introducing randomized selection to the dynamic part of the algorithm, where only subsets of the variables are updated during the dynamic steps. Using a testbed of 100 coupled harmonic oscillators, it was seen that RHMC is more efficient than HMC for all non-zero couplings. The efficiency was increased by up to 150%, in the case of $\lambda = 0.5$.

As a second test, we carried out a simulation of 100 Lennard-Jones particles in a periodic box. Because of the limited range of the Lennard-Jones force, only neighboring particles are effectively coupled, while the others move independent of one another. Nonetheless, we were able to increase the efficiency of HMC by approximately 25%, using a subset size of 75%.

While the Lennard-Jones potential is essentially short-range, many particle systems in molecular modeling and astrophysics involve long-range interactions (e.g. Coulomb and gravitational forces), where every body has a significant influence on every other body. From the results of this paper, it is likely that the efficiency of these long-range simulations can be significantly increased by using RHMC rather than HMC.

Acknowledgments

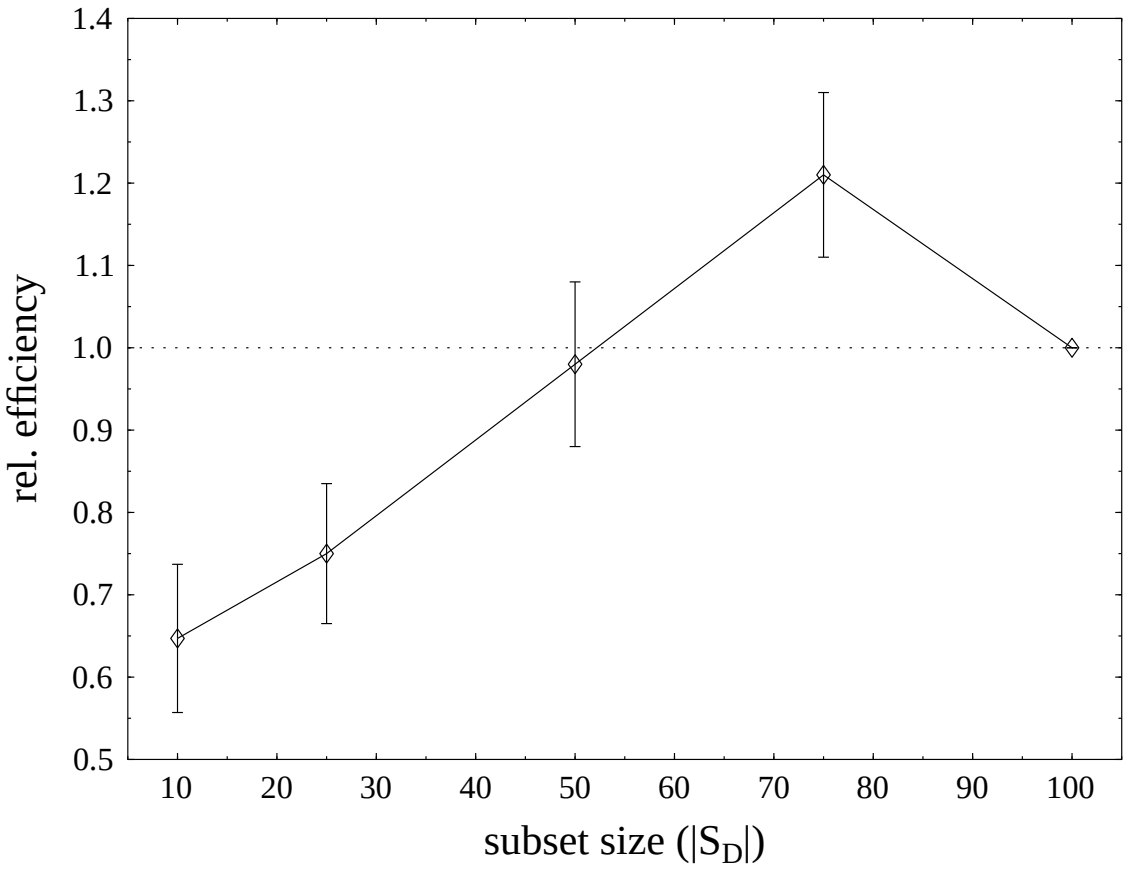

Figure 5: Relative efficiency of RHMC to HMC ($|S_D| = 100$), depending on $|S_D|$. Although the Lennard-Jones particles interact effectively with their neighors only, an efficiency increase of up to 25% was obtained, for $|S_D| = 75$.

We would like to thank N. Leporé for useful discussions regarding this work. Financial support has been provided by the Engineering Research Center (ERC) for Particle Science and Technology at the University of Florida (National Science Foundation (NSF) Grant # EEC-94-02989 and the Industrial Partners of the ERC).

References

[1] Nicholas Metropolis, Adrianna W. Rosenbluth, Marshall N. Rosenbluth, and Augusta H. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, June 1953.

[2] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods. Technical report, Department of Computer Science, University of Toronto., september 1993.

[3] A. D. Kennedy. The theory of hybrid stochastic algorithms. In P.H. Damgaard, H. Hüffel, and A. Rosenblum, editors, *Probabilistic Methods in Quantum Field Theory and Quantum Gravity*, pages 209–223. Plenum Press, New York, 1989.

[4] Frank Guarnieri and W. Clark Still. A rapidly convergent simulation method: Mixed monte carlo/stochastic dynamics. *Journal of Computational Chemistry*, 15(11):1302–1310, April 1994.

[5] M.E. Clamp, P.G. Baker, C.J. Stirling, and A. Brass. Hybrid monte carlo: An efficient algorithm for condensed matter simulation. *Journal of Computational Chemistry*, 15(8):838–846, 1994.

[6] S. Rajasekaran and J.H. Reif. "Derivation of Randomized Algorithms for Sorting and Selection." *In Parallel Algorithms Derivation and Program Transformation*, pages 187–205. Kluwer Academic Publishers, 1993.

[7] Brian Thorndyke, Paul Fishwick, and Sanguthevar Rajasekaran. A randomized approach to hybrid monte carlo simulation. In Adrian Tentner, editor, *High Performance Computing: Grand Challenges in Simulation*, pages 18–24. Society for Computer Simulation, April 1996.

[8] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids.* Oxford University Press, New York, 1992.

[9] R. K. Pathria. *Statistical Mechanics.* Elsevier Science Inc., New York, 1992.

[10] F. Reif. *Fundamentals of Statistical and Thermal Physics.* McGraw-Hill, Inc., 1965.