IMPROVED SAMPLING OF CONFIGURATION SPACE OF BIOMOLECULES USING SHADOW HYBRID MONTE CARLO

A Thesis

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Computer Science and Engineering

by

Scott S. Hampton, B.S.

Jesús A. Izaguirre, Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana
March 2004

IMPROVED SAMPLING OF CONFIGURATION SPACE OF BIOMOLECULES USING SHADOW HYBRID MONTE CARLO

Abstract

by

Scott S. Hampton

Sampling the configuration space of complex biological molecules is an important and formidable problem. One major difficulty is the high dimensionality of this space, roughly $3N$, with the number of atoms N typically in the thousands. This thesis introduces shadow hybrid Monte Carlo (SHMC), a propagator through phase space that enhances the scaling of sampling with space dimensionality. SHMC is a biased variation on the hybrid Monte Carlo algorithm (HMC) that uses an approximation to the modified Hamiltonian to sample more efficiently through phase space. The overhead introduced is modest in terms of time, involving only dot products of the history of positions and momenta generated by the integrator. We present the derivation of SHMC, along with: proof that it preserves microscopic reversibility; analysis of the asymptotic speedup of SHMC over HMC, which is shown to be $O(N^{1/4})$ when using Verlet integrators; and results evaluating correctness and efficiency.

CONTENTS

FIGURES

TABLES

CHAPTER 1

INTRODUCTION

The sampling of the configuration space of complex biological molecules is an important and formidable problem. One major difficulty is the high dimensionality of this space, roughly $3N$, with the number of atoms N typically in the thousands. Other difficulties include the presence of multiple time and length scales, and the rugged energy hyper-surfaces that make trapping in local minima common, cf. [2]. This thesis introduces the shadow Hybrid Monte Carlo (SHMC), a propagator through phase space[1] that enhances the scaling of sampling with space dimensionality.

Sampling of configuration space can be done with Markov chain Monte Carlo methods (MC) or using molecular dynamics (MD). MC methods are rigorous sampling techniques. However, their application for sampling large biological molecules is limited because of the difficulty of specifying good moves for dense systems [4] and the large cost of computing the long range electrostatic energy, cf. [31, p. 380]. In addition, MC methods are usually limited to single particle perturbations in order to keep the probability of accepting a move feasible. MD, on the other hand, can be readily applied. It enables relatively large steps in phase space and allows global updates of all the positions and momenta in the system. Nevertheless, the numerical implementation of MD introduces a bias due to the finite step size in the numerical integrator used to solve the equations of motion.

[1] Phase space is a $6N$-dimensional hyper-space where $3N$ dimensions correspond to the positions and $3N$ dimensions correspond to the momenta for a system of N particles.

Hybrid Monte Carlo (HMC), introduced in [8], uses MD to generate a global MC move and then uses a technique known as the Metropolis criterion to accept or reject the move. HMC rigorously samples the canonical[2] distribution and eliminates the bias of MD due to finite step size. As we will see in Chapter 2, it is sufficient that the numerical integrator for MD be reversible and preserve volume in phase space to ensure detailed balance.

Unfortunately, the rejection rate of HMC increases exponentially with the system size N due to fluctuations in the energy introduced by the numerical integrator, cf. [6, 16]. The fluctuations can be reduced by using higher order integrators for the MD step, as was attempted for lattice-gauge simulations in [7]. Unfortunately, higher order integrators are not an efficient alternative for MD for two reasons. First, because the evaluation of the force is very expensive, and these integrators typically require more than one force evaluation per step. Second, because the higher accuracy in the trajectories is not needed in MD, where statistical errors and errors in the force evaluation are very large.

SHMC is a biased variation on HMC, which uses a smooth approximation to the shadow Hamiltonian to sample more efficiently through phase space. The shadow Hamiltonian is exactly conserved by the numerical integrator, and a cheap and arbitrarily accurate approximation has been proposed in [33]. SHMC samples a non-canonical distribution defined by high order approximations to the shadow Hamiltonian, which greatly increases the acceptance rate of the method. A reweighting of the observable to be sampled is performed in order to obtain proper canonical averages, thus eliminating the bias introduced by the shadow Hamiltonian. The overhead introduced by the method is modest in terms of time, involving only dot products of the history of positions and momenta generated by the integrator. There is moderate extra storage needed to maintain this history.

[2]The canonical ensemble can be thought of as a probability distribution function for configuration space where the number of atoms, volume and temperature are all held constant.

We present the derivation of SHMC, along with: (i) a proof that it preserves microscopic reversibility, which makes it a rigorous sampling method; (ii) an analysis of the asymptotic speedup of SHMC over HMC, which is shown to be $O(N^{1/4})$ when using Verlet/leapfrog or r-RESPA/Impulse as the integrator; (iii) and results of evaluating correctness and efficiency of sampling in a number of molecular systems: butane, decalanine, and BPTI, ranging in size from 4 to 1,101 atoms. We evaluate correctness by computing: (i) the average torsion energy for butane and (ii) the average potential energy for decalanine and BPTI. We evaluate efficiency of sampling for the molecular systems by computing the computer time per new conformation visited. We also show the acceptance rate and the number of new conformations discovered per simulation.

1.1 Sampling problem

The problem of sampling can be thought of as estimating expectation values for a function $A(\Gamma)$ with respect to a probability distribution function (p.d.f.) $\rho(\Gamma)$, where Γ is a state variable $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_n\}$, and $\gamma_i = (x_i, p_i)$, where x_i are positions and p_i the momenta for the i^{th} atom. For the case of continuous components of Γ,

$$\langle A(\Gamma) \rangle_\rho = \int A(\Gamma)\rho(\Gamma)\mathrm{d}\Gamma. \tag{1.1}$$

Examples of observables A whose averages we may want to compute include potential energy, pressure, free energy, and distribution of solvent molecules in vacancies[14]. For the sampling of configuration space of biological molecules, ρ typically corresponds to a constant temperature T and volume V ensemble (canonical ensemble),

$$\rho_{\mathrm{NVT}}(\Gamma) \propto \exp(-\mathcal{H}(\Gamma)/(k_{\mathrm{B}}T)), \tag{1.2}$$

where $\mathcal{H}$ is the Hamiltonian or total energy of the system and k_{B} is Boltzmann's constant. Even better, because it can be compared to experiments, is the constant T and pressure P

ensemble

$$\rho_{\mathrm{NPT}}(\Gamma, V) \propto \exp(-(\mathcal{H}(\Gamma) + PV)/(k_{\mathrm{B}}T)), \tag{1.3}$$

where the volume can fluctuate, $0 < V < +\infty,$ and the positions x belong to a box scaled to have volume $V.$ The momenta of the system p are typically drawn from a well known distribution, such as a Gaussian.

1.2 Results and contributions of this thesis

This thesis contains the derivation and proof that SHMC is a rigorous sampling method. It is also shown that SHMC has an asymptotic speedup of $N^{(1/4)}$ over HMC. An approximation to SHMC, denoted SHMC*, has been implemented and tested. SHMC* is nearly equivalent to SHMC in the limit that c approaches 0. SHMC* maintains many of the performance characteristics of SHMC but is simpler to implement. Several small molecules and simple protein systems have been tested. A nearly 8-fold speedup is observed in the efficiency of sampling of SHMC* for a system containing 1101 atoms. We analytically predict speedups over HMC approaching an order of magnitude for typically sized proteins of 10 - 30 thousand atoms. Experimental data shows evidence of this to be true. Bias was not detected in the computation of the average torsion energy for n-butane, which was verified analytically, nor in the average potential energy for decalanine and BPTI. In addition, several metrics have been extended for determining the efficiency of sampling for biomolecular systems.

CHAPTER 2

DERIVATION OF SHADOW HYBRID MONTE CARLO

We first present the fundamental sampling methods MD, MC, and HMC. Next, we introduce SHMC by showing its derivation along with a proof that it satisfies microscopic reversibility. This chapter is concluded with a discussion of the theoretical performance of SHMC.

2.1 Molecular dynamics as a sampling method

As previously mentioned, MD is an important sampling method for biomolecules. It can be readily applied as long as one has a "force field" description of all the atoms and interactions among atoms in a molecule. MD finds changes over time in conformations of the molecule (semi-stable geometric configurations, to be defined more precisely later).

MD typically solves Newton's equations of motion, a Hamiltonian system of equations,

$$\dot{\Gamma}(t) = J\mathcal{H}_{\Gamma}(\Gamma(t)), \qquad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \tag{2.1}$$

where $\Gamma = (x, p)$, with a Hamiltonian

$$\mathcal{H}(x, p) = \frac{1}{2} p^{\mathrm{T}} M^{-1} p + U(x), \tag{2.2}$$

where M is a diagonal matrix of masses, $U(x)$ is the potential energy of the system, and

$p = M\dot{x}$. A typical form of $U(x)$ for biological molecules is

$$\begin{aligned} U &= U^{\text{bonded}} + U^{\text{non-bonded}}, \\ U^{\text{bonded}} &= U^{\text{bond}} + U^{\text{angle}} + U^{\text{dihedral}} + U^{\text{improper}}, \\ U^{\text{non-bonded}} &= U^{\text{electrostatic}} + U^{\text{Lennard-Jones}}. \end{aligned} \tag{2.3}$$

Equation (2.1) can be rewritten as

$$\dot{x}(t) = M^{-1}p(t), \qquad \dot{p}(t) = F(x(t)), \tag{2.4}$$

where $F(x(t))$ are conservative forces and are defined to be the negative gradient of the potential energy, $-\nabla U(x)$.

Numerical integrators for MD generate a solution $\Gamma^n \approx \Gamma(n\delta t)$, where δt is the step size or time step used in the discretization. It should be noted that Γ^n is only an approximation to $\Gamma(n\delta t)$. Higher order integrators help to reduce the error. Typical integrators can be expressed as

$$\Gamma^{n+1} = \Psi(\Gamma^n), \tag{2.5}$$

such that the values at time step $n+1$ are a function of the values at time step n. In this work we will use the common Verlet or leapfrog discretization of Equation (2.4), which using the form expressed above can be written as:

$$\begin{aligned} x^{n+1} &= x^n + \delta t M^{-1} p^n - \frac{1}{2}\delta t^2 M^{-1} F(x^n), \\ p^{n+1} &= p^n - \frac{1}{2}\delta t \left(F(x^n) + F(x^{n+1})\right). \end{aligned} \tag{2.6}$$

If the system is ergodic (defined below), modifications to the equations of motion for MD can give the correct averages for sampling from ρ_{NVT}, as for example Nosé thermostat methods (cf. [3]), or from ρ_{NPT}, such as the Langevin piston method (cf. [10]).

In any case, the use of a finite time step δt introduces a bias in the estimate of samples using MD. Also, given the presence of many local minima and energy barriers in typical force fields to describe $U(x)$ and $F(x)$, MD can get easily trapped. Furthermore, there are indications that low frequency correlations are under sampled in MD simulations of biological macromolecules [5].

2.2 Monte Carlo Markov chain

We explore next the possibility of using Monte Carlo methods, which are rigorous sampling methods, and also use stochasticity to avoid trapping. We can sample from $\rho_{\mathrm{x}}(x)$, the desired probability density function (p.d.f.) for configuration space, by simulating a Monte Carlo Markov Chain, where moves are generated according to $\rho_{\mathrm{s}}(x'|x)$, the conditional p.d.f. for the new configuration X' given that the previous configuration is $X = x$. This is implemented as Algorithm 1. Hasting (1970, as quoted in [26, p. 137]) introduced this generalization of Metropolis Monte Carlo to non-symmetric proposal distributions.

Algorithm 1 HASTINGS MARKOV CHAIN MONTE CARLO

Given X :

(1) Generate X'

(2) Accept X' with probability

$$\min\left\{1, \frac{\rho_{\mathrm{x}}(X')\rho_{\mathrm{s}}(X|X')}{\rho_{\mathrm{x}}(X)\rho_{\mathrm{s}}(X'|X)}\right\}$$

(3) If rejected, choose X.

Definition 1. *For Metropolis MC, detailed balance means that the proposal distribution is symmetric,* $\rho_{\mathrm{s}}(x|x') = \rho_{\mathrm{s}}(x'|x)$.

Definition 2. *In general, microscopic reversibility means that* $\rho_{\mathrm{s}}(x|x')\rho_{\mathrm{x}}(x')\rho_{\mathrm{s}}(x'|x)\rho_{\mathrm{x}}(x)$, *cf. [1, p. 116].*

Convergence of a Markov Chain requires (1) ergodicity—except for a set of measure zero of initial configurations, all configurations are reached arbitrarily closely infinitely

often; and (2) that $\rho(x)$ does not vary with time, i.e. stationary. For an ergodic chain there can only be one stationary p.d.f.

Proposition 1. *Microscopic reversibility implies $\rho_{\mathrm{x}}(x)$ is stationary.*

Proof. If $\rho_{\mathrm{x}}(x)$ is the p.d.f. for X, then the p.d.f. for X' is

$$\rho_{\mathrm{x}}(x') = \int \rho(x'|x)\rho_{\mathrm{x}}(x)\mathrm{d}x = \int \rho(x|x')\rho_{\mathrm{x}}(x')\mathrm{d}x = \rho_{\mathrm{x}}(x'),$$

since $\rho(x'|x)$ is a p.d.f. with respect to x. ■

Proposition 2. *Algorithm 1 satisfies microscopic reversibility.*

Proof. Enough to show it for $x' \neq x$:

$$\rho(x'|x) = \min\left\{1, \frac{\rho_{\mathrm{x}}(x')\rho_s(x|x')}{\rho_{\mathrm{x}}(x)\rho_{\mathrm{s}}(x'|x)}\right\}\rho_s(x'|x).$$

Then

$$\rho(x'|x)\rho_{\mathrm{x}}(x) = \min\left\{\rho_{\mathrm{s}}(x'|x)\rho_x(x), \rho_{\mathrm{s}}(x|x')\rho_{\mathrm{x}}(x')\right\},$$

which is clearly symmetric in x and x'. ■

2.3 Hybrid Monte Carlo (HMC)

MD and MC complement each other in their ability to explore phase space: MD can take long steps in phase space, whereas MC can exhibit random walk behavior. However, MC can escape more easily from local minima due to the stochasticity built into the method, plus the rejection step makes it exact. Hybrid Monte Carlo (HMC) combines the long steps in phase space of an MD trajectory, with an MC step that introduces stochasticity and also eliminates inaccuracies due to finite time step and other numerical artifacts. See Algorithm 2.

Let

$$\Psi = \begin{pmatrix} \Psi_1 \\ \Psi_2 \end{pmatrix}$$

be an MD integrator,

$$\gamma = \begin{pmatrix} x \\ p \end{pmatrix}$$

be a point in phase space, and

$$\Gamma = \begin{pmatrix} X \\ P \end{pmatrix}$$

be the set of all points in phase space. Assume that the integrator is volume preserving, $\det \Psi'(\gamma) = 1$, and reversible, $\Psi^{-1}(\gamma) = R\Psi(R\gamma)$, where

$$R = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}.$$

Algorithm 2 Hybrid Monte Carlo (HMC)

Given X :

(1) Generate P from $\rho_{\mathrm{p}}(p)$

(2) $X' = \Psi(X, P)$

(3) Accept X' with probability

$$\min\left\{1, \frac{\rho(\Psi(X,P))}{\rho(X,P)}\right\},$$

where $\rho(X, P) = \rho_{\mathrm{x}}(x)\rho_{\mathrm{p}}(p)$

(4) If rejected, choose X.

Proposition 3. *HMC satisfies microscopic reversibility in configuration space.*

Proof. It is sufficient to consider $x' \neq x$. Microscopic reversibility here means that the probability of transition from somewhere in a region A to a region B of configuration space is the same as that of the reverse transition (cf. [26, p. 43]).

Let π_{AB} be the probability of $X \in A$ and $X' \in B$, assuming X has p.d.f. $\rho_{\mathrm{x}}(x)$. Then

$$\pi_{AB} = \int_A \int_B \rho(x'|x)\rho_{\mathrm{x}}(x)\mathrm{d}x'\mathrm{d}x.$$

It suffices to show that $\pi_{AB} = \pi_{BA}$ for arbitrary $A \cap B = \phi$. Now,

$$\pi_{AB} = \int\int 1_A(x)1_B(\Psi_1(x,p)) \min\left\{1, \frac{\rho(\Psi(x,p))}{\rho(x,p)}\right\} \rho_{\mathrm{p}}(p)\rho_{\mathrm{x}}(x)\mathrm{d}p\mathrm{d}x,$$

where 1_A and 1_B are indicator, or characteristic, functions. Then,

$$\pi_{AB} = \int\int 1_A(x)1_B(\Psi_1(x,p)) \min\left\{\rho(x,p), \rho(\Psi(x,p))\right\} \mathrm{d}p\mathrm{d}x. \tag{2.7}$$

It is enough to show that Equation (2.7) satisfies $\pi_{BA} = \pi_{AB}$. Replacing (x,p) by $\Psi^{-1}(x,p) = R\Psi(x,-p)$, we obtain

$$\pi_{AB} = \int\int 1_A(\Psi_1(x,-p))1_B(x) \min\left\{\rho(R\Psi(x,-p), \rho(x,p)\right\} \mathrm{d}p\mathrm{d}x.$$

Replacing p by $-p$, flipping limits, and using evenness of ρ, we derive:

$$\pi_{AB} = \int\int 1_A(\Psi_1(x,p))1_B(x) \min\left\{\rho(\Psi(x,p), \rho(x,p)\right\} \mathrm{d}p\mathrm{d}x = \pi_{BA}.$$

■

Remark 1. *Random p and coupling of x to p imply that HMC is ergodic. In practice, people believe that MD of biological molecules is ergodic as well.*

2.4 Shadow hybrid Monte Carlo (SHMC)

We will see soon that HMC's performance degrades when δt or N grow. This is related to the fluctuations in the energy, which increase with δt and N and cause an extremely high rejection rate. Here we present a method that samples using a much smoother energy function. In this generalization of HMC, we do sampling in all of phase space as opposed to configuration space alone.

Let $\rho^{\mathrm{M}}(x,p)$ be the modified target density of SHMC, where

$$\rho^{\mathrm{M}}(x,p) \propto \exp\left(-\beta\mathcal{H}^{\mathrm{M}}(x,p)\right),$$

and

$$\mathcal{H}^{\mathrm{M}}(x,p) = \max\left\{\mathcal{H}(x,p) - c, \mathcal{H}^{\mathrm{S}}(x,p)\right\}.$$

Here, $\mathcal{H}^{\mathrm{S}}(x,p)$ is the much smoother shadow Hamiltonian, defined in Section 3.3, and c is an arbitrary constant that limits the amount by which $\mathcal{H}^{\mathrm{S}}$ is allowed to depart from $\mathcal{H}$. We also assume that $\mathcal{H}^{\mathrm{M}}(x, Rp) = \mathcal{H}^{\mathrm{M}}(x,p)$.

Algorithm 3 Shadow Hybrid Monte Carlo (SHMC)

---

(1) **MC Step:**

Given X, generate P with p.d.f. $\rho^{\mathrm{M}}(X,p)$. For example, one may proceed as follows:

(a) Generate P having p.d.f. $\rho_{\mathrm{p}}(p)$

(b) Accept with probability

$$\min\left\{1, \exp(-\beta(\mathcal{H}^{\mathrm{S}}(\Gamma) + c - \mathcal{H}(\Gamma))\right\}$$

(c) Repeat (1a) - (1b) until P is accepted.

(2) **MD Step:**

Given Γ :

(a) $\Gamma' = R\Psi(\Gamma)$ (where Ψ nearly conserves $\mathcal{H}^{\mathrm{S}}$)

(b) Accept Γ' with probability

$$\min\left\{1, \frac{\rho^{\mathrm{M}}(\Gamma')}{\rho^{\mathrm{M}}(\Gamma)}\right\}$$

(c) If rejected, choose Γ.

(3) **Reweighting Step:**

Given $\{A, \Gamma\}$, reweight observable A using $\rho(\Gamma)/\rho^{\mathrm{M}}(\Gamma)$ before computing averages. For example, to obtain proper canonical distributions:

$$\langle A \rangle_{\rho_{\mathrm{NVT}}} = \frac{1}{m}\sum_{i=1}^{m} w_i A_i$$

where the expected value of $\frac{1}{m}\sum_{i=1}^{m} w_i = 1$, and

$$w_i = \frac{\exp\left(-\beta\mathcal{H}(\Gamma_i)\right)}{\exp\left(-\beta\mathcal{H}^{\mathrm{M}}(\Gamma_i)\right)}.$$

Appropriate values for the constant c are proposed next. Consider the probability of acceptance $P_{\rm acc}$ of the MC step 1(a) of Algorithm 3 as $\mathcal{H}$ departs from $\mathcal{H}^{\rm S}$. Let $\Delta\mathcal{H} = \mathcal{H}^{\rm S} - \mathcal{H}$. There are two cases: If $\Delta\mathcal{H} \geq 0$, then $c \leq 0$, and

$$P_{\rm acc} = \begin{cases} 1 & \text{if } 0 \leq \Delta\mathcal{H} \leq -c, \\ \exp(-\beta(\Delta\mathcal{H} + c)) & \text{if } \Delta\mathcal{H} > -c. \end{cases} \tag{2.8}$$

Else, if $\Delta\mathcal{H} < 0$, then $c \geq 0$, and

$$P_{\rm acc} = \begin{cases} 1 & \text{if } -c \leq \Delta\mathcal{H} < 0, \\ \exp(-\beta(\Delta\mathcal{H} + c)) & \text{if } \Delta\mathcal{H} < -c. \end{cases} \tag{2.9}$$

Negative c should be used when $\mathcal{H}^{\rm S} \geq \mathcal{H}$, and positive c when the opposite holds. Experiments suggest that $\Delta\mathcal{H}$ is predominantly positive in MD simulations. Two criteria are also proposed to determine numerical values of c. The first criterion bounds the weights w_i from below or above, depending on the sign of $\Delta\mathcal{H}$. These weights are exponentially proportional to the value of $\Delta\mathcal{H}$. The second determines c that yield acceptable $P_{\rm acc}$ in the MC step. This can be obtained from a histogram of $\Delta\mathcal{H}$ based on short MD simulations, either at the beginning of the simulations, or periodically if $\Delta\mathcal{H}$ changes significantly during exploration of phase space.

Suppose a maximum weight $w_{\max} \propto N$ is desired when $\Delta\mathcal{H} > 0$, and a minimum weight $w_{\min} = 1/(e \cdot N)$ when $\Delta\mathcal{H} < 0$. Values of c that satisfy these constraints on average can be found from

$$\ln w = -\beta c,$$

because the MD step of SHMC tends to reject moves where $|\Delta\mathcal{H}| > c$. These values are $c \propto \frac{\ln N}{\beta}$ and $c \propto \frac{\ln N}{\beta}$ when $\Delta\mathcal{H} > 0$ and $\Delta\mathcal{H} < 0$, respectively. The bounds for the weight could also be chosen to balance the systematic and statistical error, for example.

Proposition 4. *The MD step of SHMC satisfies microscopic reversibility.*

Proof. Sufficient to consider $\gamma' \neq \gamma$. Let π_{AB} be the probability of $\Gamma \in A$ and $\Gamma' \in B$, assuming Γ has p.d.f. $\rho^{\mathrm{M}}(\gamma)$. Then

$$\pi_{AB} = \int_A \int_B \rho(\gamma'|\gamma)\rho^{\mathrm{M}}(\gamma)\mathrm{d}\gamma'\mathrm{d}\gamma.$$

It suffices to show that $\pi_{AB} = \pi_{BA}$ for arbitrary $A \cap B = \phi$. Now,

$$\pi_{AB} = \int \int 1_A(\gamma)1_B(R\Psi(\gamma)) \min\left\{1, \frac{\rho(\Psi(\gamma))}{\rho(\gamma)}\right\} \rho(\gamma)\mathrm{d}\gamma,$$

where 1_A and 1_B are indicator, or characteristic, functions, and then

$$\pi_{AB} = \int \int 1_A(\gamma)1_B(R\Psi(\gamma)) \min\left\{\rho(\gamma), \rho(\Psi(\gamma))\right\} \mathrm{d}\gamma.$$

Replacing γ by $\Psi^{-1}(\gamma) = R\Psi(R\gamma)$, we get

$$\pi_{AB} = \int \int 1_A(R\Psi(R\gamma))1_B(R\gamma) \min\left\{\rho(R\Psi(R\gamma)), \rho(\gamma)\right\} \mathrm{d}\gamma.$$

Replacing γ by $R\gamma$, we finally get:

$$\pi_{AB} = \int \int 1_A(R\Psi(\gamma))1_B(\gamma) \min\left\{\rho(\gamma), \rho(\Psi(\gamma))\right\} \mathrm{d}\gamma = \pi_{BA.}$$

■

Proposition 5. *The MC step generates* $\rho^{\mathrm{M}}(x, p)$.

Proof. According to Von Neumann (1951; as quoted in [1, p. 349]), this method generates a random number from an arbitrary complex distribution, which in this case is given by

$$\rho^{\mathrm{M}}(x, p) = f(x)\rho_{\mathrm{p}}(p) \min\left\{1, \exp\left(-\beta(\mathcal{H}^{\mathrm{S}}(\gamma) + c - \mathcal{H}(\gamma))\right)\right\}.$$

■

Proposition 6. *The MC step of SHMC satisfies microscopic reversibility.*

Proof. Since $\rho^{\mathrm{M}}(x,p)$ and $\rho^{\mathrm{M}}(x',p')$ are independent, the probabilities of going from one to the other are clearly symmetric. ■

Corollary 1. *SHMC satisfies microscopic reversibility, since both the MD and MC steps separately satisfy it.*

2.5 Performance of SHMC

The cost of HMC as a function of system size N and time step δt has been investigated in [6]. HMC drives (X,P) towards an equilibrium with a coupled probability

$$\rho(X,P) = \rho_{\mathrm{x}}(x)\rho_{\mathrm{p}}(p) \propto \exp(-\mathcal{H}(\gamma)).$$

Computing the expected values over this distribution, one finds that

$$\langle \exp(-\delta\mathcal{H}(x,p)) \rangle_{\rho(X,P)} = 1, \tag{2.10}$$

where $\delta\mathcal{H} = \mathcal{H}(x',p') - \mathcal{H}(x,p)$. We take the $\log$ of both sides of Equation (2.10):

$$\log \langle \exp(-\delta\mathcal{H}(x,p)) \rangle_{\rho(X,P)} = 0.$$

Henceforth, we drop the parameters of the Hamiltonian $\mathcal{H}$ and the density ρ. Since $\exp$ is a convex function, we use Jensen's inequality to write

$$\langle \delta\mathcal{H} \rangle \geq 0, \tag{2.11}$$

with equality possible only if the MD integrator Ψ exactly conserves energy.

For small $\delta\mathcal{H}$, we expand Equation (2.10) up to a third order term,

$$\langle \exp(\delta\mathcal{H}) \rangle = 1 + \langle \delta\mathcal{H} \rangle + \frac{1}{2} \langle \delta\mathcal{H}^2 \rangle + O(\delta\mathcal{H}^3) = 1,$$

and thus

$$\langle \delta\mathcal{H} \rangle = \frac{1}{2} \langle \delta\mathcal{H}^2 \rangle + O(\delta\mathcal{H}^3). \tag{2.12}$$

For an MD integrator Ψ that is $O(\delta t^m)$ accurate, Equation (2.12) becomes

$$\langle \delta \mathcal{H} \rangle = O(\delta t^{2m}). \tag{2.13}$$

We want to know the expected value of the probability of acceptance in HMC. Because of Equation (2.11), updating N variables together is expected to cause an increase in $\mathcal{H}$, that is, positive $\delta\mathcal{H}$. Thus the acceptance rate falls as

$$\exp(-\beta N \delta t^{2m}). \tag{2.14}$$

This equation means that to keep the acceptance rate constant as N increases, and for a fixed β, one needs to change $\delta t \propto N^{-1/(2m)}$.

Let L be the MD trajectory length needed to produce an uncorrelated sample. Assuming L is fixed, the cost of producing uncorrelated samples increases as

$$T^{\mathrm{HMC}}_{\mathrm{MD\_step}} N^{1/(2m)} + T^{\mathrm{HMC}}_{\mathrm{MC\_step}}, \tag{2.15}$$

where $N^{1/(2m)}$ is the number of MD steps to achieve a trajectory of length L. $T^{\mathrm{HMC}}_{\mathrm{MD\_step}}$ is the cost of each MD step, which depends on the cost of the force evaluation, and will be anywhere from $O(N)$ for cutoff computation to $O(N^2)$ for all pairs evaluation, or more typically $O(N \log N)$ for tree methods or FFT-based methods. $T^{\mathrm{HMC}}_{\mathrm{MC\_step}}$ is the cost of generating P, basically the generation of $O(N)$ random numbers from a Gaussian distribution.

A similar argument can be made for the acceptance rate of SHMC, but now with respect to the equilibrium p.d.f. ρ^{M}. Thus, $\left\langle \delta \mathcal{H}^{\mathrm{M}} \right\rangle = O(\delta t^{2k})$, where k is the order of the shadow Hamiltonian $\mathcal{H}^{\mathrm{S}}_k$, and the cost of producing uncorrelated samples increases as

$$T^{\mathrm{SHMC}}_{\mathrm{MD\_step}} N^{1/(2k)} + T^{\mathrm{SHMC}}_{\mathrm{MC\_step}}. \tag{2.16}$$

The asymptotic speedup of SHMC over HMC is given by the quotient of Equations (2.15) and (2.16). Note that $T^{\mathrm{SHMC}}_{\mathrm{MD\_step}} \gg T^{\mathrm{SHMC}}_{\mathrm{MC\_step}}$ and $T^{\mathrm{HMC}}_{\mathrm{MD\_step}} \gg T^{\mathrm{HMC}}_{\mathrm{MC\_step}}$. In addition, $T^{\mathrm{SHMC}}_{\mathrm{MD\_step}}$, $T^{\mathrm{HMC}}_{\mathrm{MD\_step}}$, $T^{\mathrm{SHMC}}_{\mathrm{MD\_step}}$, and $T^{\mathrm{SHMC}}_{\mathrm{MC\_step}}$ have the same asymptotic complexity. Thus, the speedup is given by the ratio

$$\eta = \frac{N^{1/(2m)}}{N^{1/(2k)}} = N^{\frac{1}{2}\frac{k-m}{mk}}. \tag{2.17}$$

Typical values of N and k are given next, assuming that one is using the Verlet/leapfrog or r-RESPA/Impulse, which is second order accurate, $m = 2$. In this case the asymptotic speedup of SHMC over HMC is $= N^{1/4}$. These values are in good agreement with the numerical results presented in the next section. There are implementations available of the $8th$ and $24th$ order shadow Hamiltonians, hence the choice of $k = 8$ and $k = 24$.

TABLE 2.1. ASYMPTOTIC SPEEDUP OF SHMC OVER HMC FOR INCREASING N

N	$k = 8$	$k = 24$	η
10^3	3.7	4.9	5.6
10^4	5.6	8.3	10.0
10^5	8.7	14.0	17.8
10^6	13.3	23.7	31.6

CHAPTER 3

IMPLEMENTATION OF SHMC*

In the previous section, we derived a precise version of SHMC. However, we initially chose to implement an approximation to SHMC, denoted SHMC*. SHMC* differs from exact SHMC by simplifying the MC step and eliminating the parameter c from the Metropolis acceptance step. This simplified version reduces the complexity of the code and eases testing of the algorithm. At the same time, SHMC* is useful in determining the proper values of c that avoid damaging the variance of reweighted observables when using SHMC. These minor changes enable a quicker implementation while providing similar results.

3.1 SHMC* class

SHMC* was implemented in PROTOMOL, a framework whose purpose is to aid in the design and testing of molecular simulation algorithms. PROTOMOL was built from scratch using an object oriented (OO) approach. The framework is composed of the three component layers shown in Figure 3.1.

The front-end and back-end layers deal mostly with setup and operation of the simulation. The middle layer contains the integrator hierarchy where SHMC* is actually implemented. Figure 3.2 contains a very simplified version of the integrator hierarchy of PROTOMOL written in a UML-like fashion. Only those classes relevant to the implementation of SHMC* as an integrator have been included in the figure. PROTOMOL

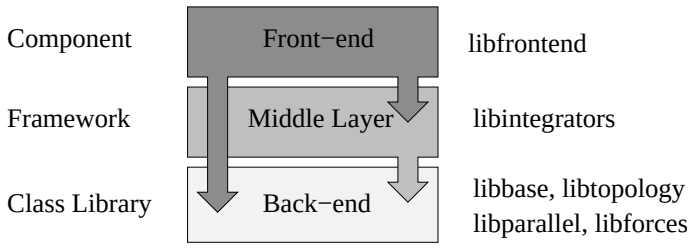

Figure 3.1. The component based design of PROTOMOL. Image courtesy T. Matthey [23].

contains many more integrators than those shown here. Likewise, instead of listing all data members and methods, a representative set has been chosen. The `Integrator` and `StandardIntegrator` classes form the basis for all integrators by defining a basic implementation. Each integrator must have access to necessary data such as positions, velocities, forces and energies. In addition, it is required that each integrator implement a *run()* method. This method is called once per simulation step and is responsible for *integrating* the system.

However, both of these classes contain pure virtual functions. Since it is not possible to instantiate a class containing a pure virtual function, actual integrators must be derived from these classes. The next level in the hierarchy differentiates between single time stepping (STS) and multiple time stepping (MTS) integrators. STS integrators are the most commonly used and involve a full force evaluation step during each cycle. MTS integrators split the force evaluations among differing levels in order to reduce computation time. MTS integrators form a chain analogous to a set of nested `for` loops. Each "loop" is a MTS integrator except for the innermost one, which must be an STS integrator. The forces are distributed among the differing levels of the chain such that the *fastest* varying forces are evaluated most often and the *slowest* varying forces are evaluated the least. `HMCIntegrator` and `ShadowHMCIntegrator` are not true MTS integrators in the sense that they do not actually *integrate* the system, but rather they use a Metropolis

criteria to determine system moves. However, they fit well into the MTS hierarchy because they require a STS integrator in the same way that MTS integrators do.

Much of the functionality of HMC and SHMC* is the same, so the class `ShadowHMCIntegrator` was implemented as a derived class of `HMCIntegrator`. Only the `run()` method and a few initialization items needed to be overloaded by SHMC*. This exhibits the power of OO programming as well as the quality of the design of the integrator hierarchy. The most obvious difference in the two methods is that SHMC* bases its Metropolis acceptance on the change in the shadow energy $\delta\mathcal{H}^S$ and not the change in the total energy $\delta\mathcal{H}$. This design works well with the overall OO structure of PROTOMOL and produces very little overhead. Unlike HMC, which can use an existing STS integrator, the internal MD integrator of SHMC* must be designed for calculating the shadow Hamiltonian. There are several structures of previous values that must be maintained as well as the implementation of the formulas for calculating the shadow. In addition, a β-term must be propagated along with the integration of the system. This particular implementation of the shadow Hamiltonian is built around Leapfrog, although there are other integrators that could be used. As was the case with SHMC*/HMC, most of the existing structure of `LeapfrogIntegrator` was reusable so `ShadowLeapfrogIntegrator` was created as a subclass.

For a shadow Hamiltonian of order $O(\delta t^{2k})$, k values of the positions, velocities and β-term must be available in addition to the current ones. For purposes of storage, we need a data structure that supports inserting elements at both the "beginning" and at the "end" as well as efficient random access to all of the elements. For these reasons, the STL class `Deque`, a double ended queue, was chosen as the primary data structure to hold the values. In order to calculate $\mathcal{H}^{\mathrm{S}}$, $k/2$ previous values, the current values and $k/2$ forward values are needed. When SHMC* starts a step, it first calls a `reverse()` method in `ShadowLeapfrogIntegrator` that runs the system backwards $k/2$ steps.

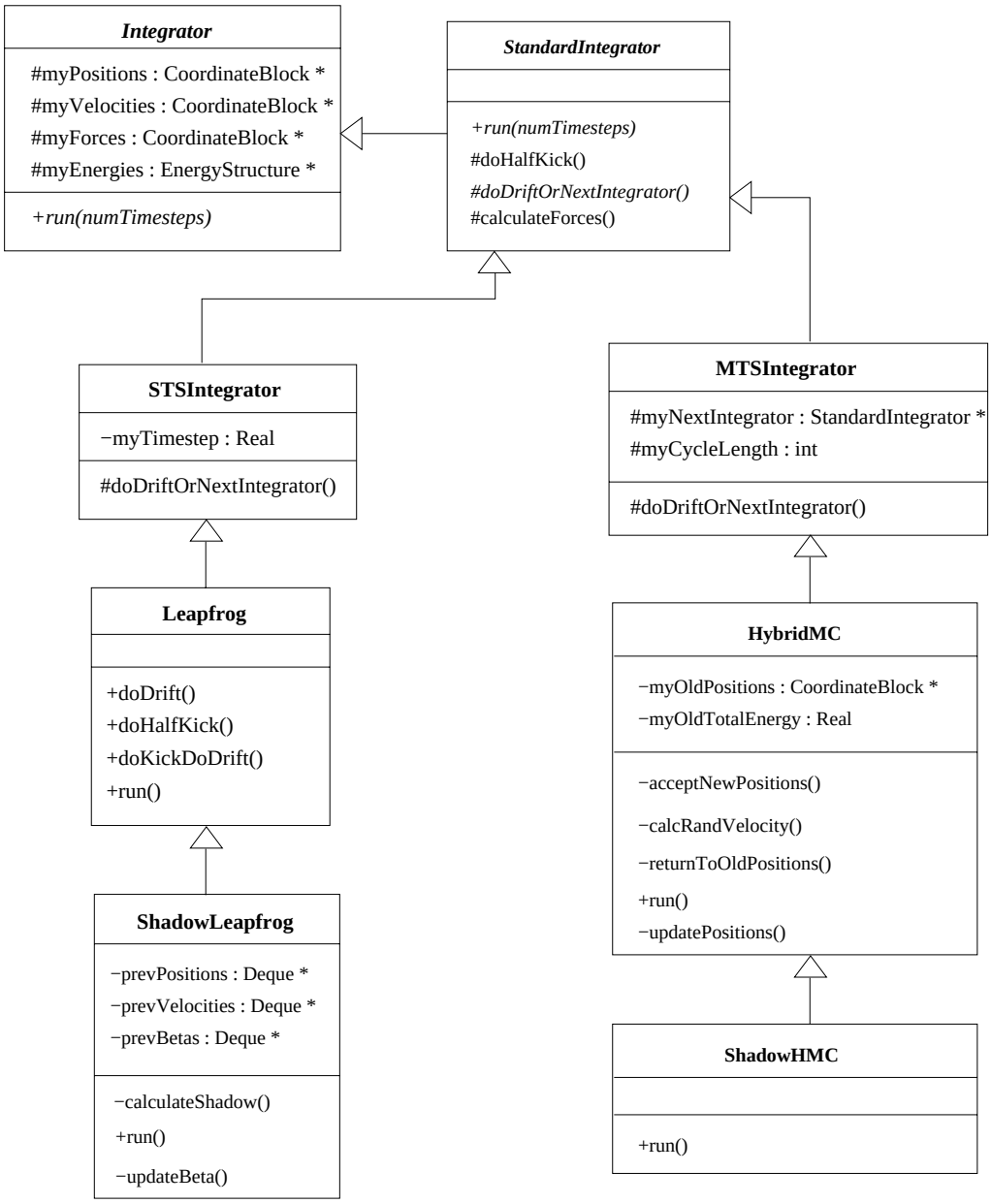

Figure 3.2. A simplified version of the integrator hierarchy in PROTOMOL.

This is valid since Leapfrog is a time-reversible integrator. After this, the `run()` method is called as normal. When $k/2$ additional steps have executed, the value of the $\mathcal{H}^{\mathrm{S}}$ at t_0 is available. At the end of L steps the shadow is only available for step $L-2$ so, $k/2$ additional steps must be run to be able to compute the shadow at step L.

3.2 SHMC* and HMC implementations

Algorithms 4 and 5 describe how HMC and SHMC* are implemented in PROTOMOL respectively.

Algorithm 4 Hybrid Monte Carlo Algorithm (HMC)

Given positions X, and $\beta = 1/(k_{\mathrm{B}}T)$ where k_{B} is Boltzmann's constant and T is the temperature:
(1) **MC Step:**
- (a) Generate new random momenta P from a Gaussian distribution
- (b) Generate MD trajectory length L from uniform distribution $[0.7L_0, 1.3L_0]$.

(2) **MD Step:**
Given time step δt:
- (a) Compute energy $\mathcal{H}_0(X, P)$
- (b) Run MD algorithm for $n_{\mathrm{md}} = L/\delta t$ steps to produce (X', P')
- (c) Compute new energy $\mathcal{H}_1(X', P')$
- (d) Compute change in energy $\delta\mathcal{H} = \mathcal{H}_1 - \mathcal{H}_0$
- (e) Choose a uniform random number, r, between $[0, 1]$
- (f) Accept new positions X' if $r < \exp(-\beta\delta\mathcal{H})$
- (g) If new positions X' are rejected, restore old positions X.

(3) **Sampling Step:**
Compute observable $A(x)$ at accepted positions.

Algorithm 5 Approximate Shadow Hybrid Monte Carlo Algorithm (SHMC*)

Given positions X, and $\beta = 1/(k_{\mathrm{B}}T)$ where k_{B} is Boltzmann's constant and T is the temperature:
(1) **MC Step:**
  (a) Generate new random momenta P from a Gaussian distribution
  (b) Generate MD trajectory length L from uniform distribution $[0.7L_0, 1.3L_0]$.
(2) **MD Step:**
  (a) Compute modified shadow energy $\mathcal{H}_0^{\mathrm{M}}(X, P)$
  (b) Run MD algorithm for $L/\delta t$ steps to produce (X', P')
  (c) Compute new modified shadow energy $\mathcal{H}_1^{\mathrm{M}}(X', P')$
  (d) Compute change in modified shadow energy $\delta\mathcal{H}^{\mathrm{M}} = \mathcal{H}_1^{\mathrm{M}} - \mathcal{H}_0^{\mathrm{M}}$
  (e) Choose a uniform random number, r, between $[0, 1]$
  (f) Accept new positions X' if $r < \exp(-\beta\delta\mathcal{H}^{\mathrm{M}})$
  (g) If new positions X' are rejected, restore old positions X.
(3) **Sampling Step:**
  (a) Compute observable $A(x)$ using accepted positions
  (b) Reweight $A(x)$ by $\mathcal{H}(x, p)/\mathcal{H}^{\mathrm{s}}(x, p))$.

3.3 Approximation to the modified Hamiltonian

The modified equations of a system of differential equations are exactly satisfied by the approximate discrete solution of the numerical integrator used to solve them. These equations are usually defined as an asymptotic expansion in powers of the discretization time step. If the expansion is truncated, there is excellent agreement between the modified equations and the discrete solution [13].

In the case of a Hamiltonian system,

$$\gamma'(t) = J\mathcal{H}_\Gamma(\gamma(t)), \qquad J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \tag{3.1}$$

with a Hamiltonian $\mathcal{H}(\gamma)$, $\Gamma = (x, p)$, symplectic integrators conserve exactly (within roundoff errors) a modified Hamiltonian $\mathcal{H}^{\delta\mathrm{t}}$. The integrator is symplectic if $\Psi\gamma J^{\mathrm{T}}\Psi\gamma \equiv J$. For short MD simulations (such as in HMC) $\mathcal{H}^{\delta\mathrm{t}}$ stays close to the true Hamiltonian, cf. [30, p. 129–136]. Work by Skeel & Hardy [33] shows how to compute an arbitrarily accurate approximation to the modified Hamiltonian integrated by symplectic integrators

based on splitting. The idea is to compute

$$\mathcal{H}^{\mathrm{S}}_{2k}(x,p) = \mathcal{H}^{\delta \mathrm{t}}(x,p) + O(\delta t^{2k}). \tag{3.2}$$

$\mathcal{H}^{\mathrm{S}}_{2k}$ is the shadow Hamiltonian of order $2k$, and is assembled from k copies of available positions and momenta generated by the MD integration. It is also necessary to propagate an extra degree of freedom β along with the momenta. For example,

$$\mathcal{H}^{\mathrm{S}}_{8}(\Gamma^n) = \Lambda^{n-2}\bar{\Gamma}^{n-2} + \Lambda^{n-1}\bar{\Gamma}^{n-1} + \Lambda^{n}\bar{\Gamma}^{n} + \Lambda^{n+1}\bar{\Gamma}^{n+1} + \Lambda^{n+2}\bar{\Gamma}^{n+2},$$

where the Λ are diagonal matrices, and $\bar{\Gamma} = (\Gamma, \beta)$ generated by the integration. This is basically a linear combination of trajectory information. By construction, $\mathcal{H}^{\mathrm{S}}$ is exact for quadratic Hamiltonians, which are very common in MD. Details can be found in the original reference. The 8^{th} order shadow, $\mathcal{H}^{\mathrm{S}}_{8}$, is currently implemented in our software. Implementations of higher order shadow Hamiltonians (up to $\mathcal{H}^{\mathrm{S}}_{24}$) will be made available by Skeel & Hardy.

We reproduce here the $\mathcal{H}^{\mathrm{S}}_{8}$ order shadow as it is coded in PROTOMOL:

$$\mathcal{H}^{\mathrm{S}}_{8} = \frac{1}{2\delta t}(210\mathrm{A}_{10} - \frac{2}{7}\mathrm{A}_{12} - \frac{19}{210}\mathrm{A}_{14} + \frac{5}{42}\mathrm{A}_{30} + \frac{13}{105}\mathrm{A}_{32} - 315\mathrm{A}_{34}) \tag{3.3}$$

If we define the i^{th} centered difference formula to be $\delta\omega^{[i]}$. So, for example, $\delta\mathbf{x}^{[2]}$ would represent the 2^{nd} centered difference of the positions:

$$\delta\mathbf{x}^{[2]} = \mathbf{x}^{n+1} - 2\mathbf{x}^{n} + \mathbf{x}^{n-1}. \tag{3.4}$$

Now, define A_{ij}:

$$\mathrm{A}_{ij} = \begin{cases} \delta\mathbf{x}^{[i]} \cdot \delta\mathbf{p}^{[j]}\mathrm{M} - \delta\mathbf{x}^{[j]} \cdot \delta\mathbf{p}^{[i]}\mathrm{M} - \delta\beta^{[i]} & : \quad j = 0 \\ \delta\mathbf{x}^{[i]} \cdot \delta\mathbf{p}^{[j]}\mathrm{M} - \delta\mathbf{x}^{[j]} \cdot \delta\mathbf{p}^{[i]}\mathrm{M} & : \quad j \neq 0 \end{cases} \tag{3.5}$$

Finally, the β term propagated by Leapfrog is:

$$\beta = -\delta t(\mathbf{x}^n \cdot F^n + 2U(\mathbf{x}^n)), \tag{3.6}$$

M is a diagonal matrix containing the mass of each atom. A superscript denotes values at a different timestep where the current timestep is considered to be n.

CHAPTER 4

TESTING OF SHMC*

The goal for the testing of the approximate version of SHMC (SHMC*) was to determine the validity of the following hypotheses:

(H1) SHMC* samples with reasonable efficiency as the system size and the time step are increased.

(H2) SHMC* has an asymptotic speedup over HMC of $O(N^{\frac{1}{4}})$.

(H3) The overhead of computing the shadow Hamiltonian is moderate.

(H4) Any bias introduced by sampling from the shadow Hamiltonian can be removed by a reweighting of sampled values.

(H5) Any bias can be made negligible.

The first hypothesis is tested by varying the time step and system size. Slight performance degradation is expected as the system size increases due to greater truncation and round off error. Similarly, as the time step approaches the upper bound, set both by instabilities in the MD integrator and the fastest motions of the system, performance will be affected. However, we show that these effects are negligible for reasonable values. More information on measuring efficiency of sampling can be found in Section 4.3.3.

In Section 2.5, it was proven that SHMC has an asymptotic speedup of $O(N^{\frac{1}{4}})$ over HMC. This analysis also applies to SHMC* and we find reasonably close scaling during our testing. Using wall clock time and efficiency of sampling, our tests show that the overhead for computing the shadow is small, although not negligible (3-10% overhead

over HMC for typical method parameters). Several optimizations are being explored to bring this overhead to about 1%, and are discussed as future work.

Section 2.4 shows that SHMC removes the bias introduced into the system by sampling from the shadow Hamiltonian. SHMC* can also be expected to remove the bias. Reweighted averages are nearly indistinguishable from correct estimates obtained from HMC.

4.1 Test systems

SHMC* was implemented and tested within PROTOMOL [23], a generalized framework for molecular simulations. SHMC* was tested on a variety of molecules from the simple alkane n-butane with only 4 (united) atoms to a more complex solvated protein, BPTI, with 1101 atoms. While n-butane is a relatively simple molecule, it was specifically chosen because it has been previously well documented and many of the results can be confirmed analytically [11]. Table 4.1 lists the test molecules and the corresponding number of atoms.

TABLE 4.1. MOLECULES USED FOR TESTING SHMC

Molecule	Number of Atoms
n-butane	4
Decalanine	66
BPTI	1101

Testing was done on a Beowulf cluster [27] administered by the College of Chemistry and Biochemistry at the University of Notre Dame. Each node contains 2 1.7 GHz Xeon processors, 1 GB RDRAM, 40 GB HD and a Gigabit Ethernet card.

4.2 Simulation parameters

In this section we discuss the parameters common to all of our simulations and then describe those that are particular to the hybrid methods.

4.2.1 Common simulation parameters

Each simulation was run with similar input but there are some parameters, such as time step, that must be tailored to the molecule being simulated. The parameters common to all simulations are as follows:

- Random initial velocities at a temperature of 300 K
- Exclude 1-3 interactions, which ignores molecules connected by bonds and angles when calculating non-bonded forces
- Center of mass motion is removed, to keep the center of mass fixed
- Periodic boundary conditions, except for butane, which is isolated
- Forces include angle, bond, Coulombic, dihedral, improper, and Lennard-Jones as in Equation (2.3)
- Leapfrog is the MD integrator used in both HMC and SHMC*
- CHARMM 22 parameters for butane/alkanes [9] and proteins [21, 22] are used

The method for evaluating the electrostatic forces varied among molecules and was determined using an automatic recommender system called MDSimAid [19]. MDSimAid attempts to heuristically determine optimal parameters for the fast electrostatic force evaluation. Given input files containing positional and structural information, MDSimAid will suggest a force evaluation method to use and it will also give the necessary parameters for use with PROTOMOL. The remaining system options have been summarized in Table 4.2.1.

TABLE 4.2. SIMULATION PARAMETERS ACCORDING TO MOLECULE

	n-Butane	Decalanine	BPTI
`timestep`δt (fs)	1, 3, 6, 8	0.25, 0.5, 1, 1.25, 2	0.05, 0.1, 0.2, 0.3, 0.4
Trajectory length L (fs)	72, 630	20, 50, 80, 100	6, 12, 18, 24
Electrostatics. Alg.	none	PME	PME

4.2.2 HMC parameters

HMC has several parameters that affect its performance, including the random number generator, the integrator chosen for the molecular dynamics, the time step δt, and trajectory length L.

Because HMC is a stochastic method, the (pseudo)random number generator is an essential component. During the course of a typical HMC step, PROTOMOL generates 36 uniform random numbers for every atom in the system. It could well happen that a long simulation would exceed the period of the random number generator. Even worse than exceeding the period are the effects of choosing a poorly designed generator. A common family of random number generators are the linear congruential generators (LCG). In these methods, a sequence is recursively defined as follows:

$$x_{i+1} = (ax_i + c) \bmod M$$

a, c, and M must be chosen carefully to maximize performance [18, p. 41]. Even when properly defined, this class of generators is still known to produce lattice structures instead of random structures [31, p. 355]. Our implementation currently uses `drand48()`, but we are contemplating using a method with a longer period and better "randomness".

Any time reversible and volume preserving integrator can be used for HMC. In SHMC and SHMC*, there is the additional constraint that it should be based on splitting to be able to compute $\mathcal{H}^{\mathrm{s}}$. Our implementation uses the Verlet/leapfrog discretization shown in Equation (2.7), which satisfies the constraints for both propagators.

In order to compare the efficiency among different runs, it was decided to fix the expected value of the trajectory L, such that $E[L] = L_0$. In this way, direct comparisons can be made between simulations with different δt.

The choices of L_0 and δt have dramatic effects on performance of HMC, SHMC, and SHMC*. L_0 should be long enough so that the longest correlation times of interest are sampled during an MD step, and thus the random walk behavior of MC be avoided. One way of approximating the correlation times in a molecular system is to compute the normal modes through a linearization of the interaction forces of interest, and take the maximum period $\tau_{\max}$ as a desirable value for L_0. Ideally, δt should also be no larger than $\tau_{\min}$, and thus $n_{\mathrm{md}} \equiv \tau_{\max}/\tau_{\min}$. In practice, however, numerical artifacts like instability and resonance force $\delta t \ll \tau_{\min}$ and $L_0 \ll \tau_{\max}$.

One way to bring L closer to $\tau_{\max}$ in HMC is to randomize n_{md}. Based on suggestions in [20], we choose a value of n_{md} from a distribution $[0.7L_0/\delta t,\, 1.3L_0/\delta t]$, with δt fixed by stability limits of Verlet/leapfrog.

4.3 Test metrics

We have implemented several techniques in order to test and gauge SHMC* against similar methods.

4.3.1 Acceptance rate

One of the most convenient metrics to compare between HMC methods is the acceptance rate. If you recall Algorithm 4, at the end of a HMC step a Metropolis acceptance check is made to determine if the current set of positions will be accepted or not. If the change in energy is too high, then the new positions are rejected and we restore the old positions. The ratio of accepted moves to total moves attempted is defined to be the acceptance rate (AR).

There is no direct relationship between the AR and the quality of the samples being generated. For example, a simulation that alternates between exactly 2 points in phase space can maintain perfect acceptance but is not generating characteristic data. However, there is certainly a correlation between the AR and sample quality. Without an at least moderately successful algorithm, we can not hope to explore phase space sufficiently. What constitutes a good or bad AR depends on the problem at hand and is somewhat subjective. We feel that the best AR is the one that samples configuration space more quickly, cf. [31, p. 376]

If the samples generated by the propagator are sufficiently uncorrelated, then performance of the algorithm $\propto$ probability of acceptance $\times$ cost of trial move. If the correlation between samples and the cost per trial for two methods are roughly the same, such as HMC and SHMC*, we would like the AR to be as high as possible.

4.3.2 Sampling rate

The AR is a start in determining the viability of a new sampling method, but it is not a completely sufficient measure. What is needed is a metric that directly tests the sampling ability of the algorithm. If there existed a method for counting the number of "new" conformations, then sampling rate (SR) could be defined as the ratio of the number of new conformations discovered to the number of possible conformations. A method that discovered a new conformation at every step would have a 100% SR.

We extend a method presented in [17, 28] based on the dihedral angles in the system. In this method, each dihedral angle is assigned a label based on its current value, usually listed in radians from $[-\pi, \pi]$. By studying the strings generated at the end of an MD step, we can determine when a particular conformation[1] is first visited. Using this labeling notation, we can calculate the rate at which new conformations are being generated.

[1]The strict definition of a conformation states that *any* change in a dihedral angle constitutes a new conformation. For purposes of sampling we denote a new conformation only when a dihedral angle moves between local maxima.

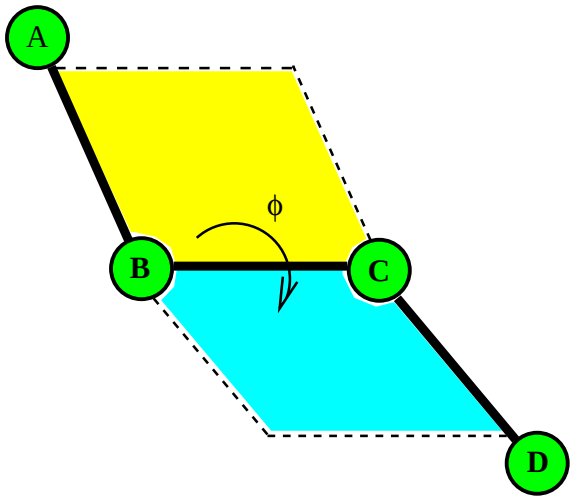

Figure 4.1. Define ϕ as the angle between the two planes formed from atoms (a, b, c) and (b, c, d).

Dihedral angle description

A dihedral angle occurs when, within a molecule, there are four atoms connected linearly end-to-end. The first three atoms form a plane, as do the last three atoms. The dihedral angle is the angle between the two planes [2], as shown in Figure 4.1. The potential energy for a single dihedral angle is defined in Equation (4.1).

$$U^{\text{dih}}(\phi) = \sum_{i=1}^{m} \frac{1}{2} f_i (1 + \cos(n_i \phi - \delta_i)). \tag{4.1}$$

Associated with each term in the potential energy for a dihedral angle ϕ is a force constant f, a periodicity n, a phase-shift δ and a multiplicity m. Many dihedral angles are defined with $m = 1$, but there are some that can only be constructed as a combination of multiple terms. The dihedral angle formed from the central carbons of n-butane in Figure A.1 is a common example. For instance, Ryckaert and Bellemans [29] use an equation

[2] An alternate definition has the dihedral defined to be the angle between the normals to the two planes.

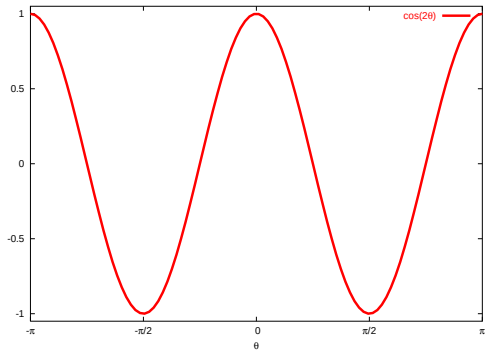

Figure 4.2. A simple example: $\cos(2\theta)$.

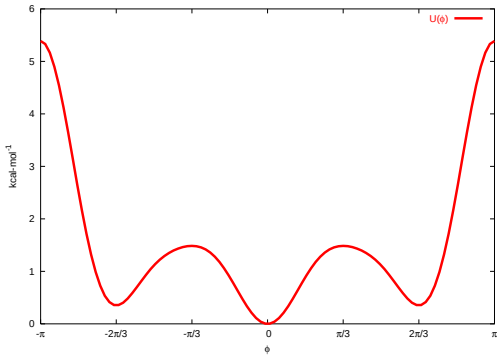

Figure 4.3. A more complex example: U^{dih} for n-butane.

with $m = 6$ to accurately describe this dihedral:

$$U^{\text{dih}}(\phi) = 1.5105 - 1.1815\cos(\phi) + 0.789\cos(2\phi) - 1.27575\cos(3\phi) \quad (4.2)$$
$$+0.3945\cos(4\phi) - 0.23675\cos(5\phi).$$

Example: Counting conformations

The function $\cos(2\theta)$ has been plotted in Figure 4.2. Notice how there are two wells formed between the local maxima. Any randomly chosen value of θ must be located in one of these two wells. Let us define the well between $[0, \pi)$ as w_1 and the well between $[\pi, 2\pi)$ as w_2. It is now possible to roughly describe the position of θ using the name of the well it occupies.

For a more complex example I have graphed Equation (4.3), describing the dihedral potential energy of n-butane, from Figure 4.3. Although this equation has a multiplicity of 6, notice that there are only 3 wells.

An obvious question to ask is how the parameters of the energy function, Equation (4.1), affect the definition of the wells. The most important parameter is n because it determines the maximum number of wells. f determines the maximum height of a well,

while δ literally shifts the graph. In the case where there is only one term describing the energy, each root will either be at a maximum or a minimum value. When m is greater than one, there will be local maxima and minima due to the summation of multiple periodic terms. In general, the phase shift is usually 0 or π. There are some dihedral energy functions that have a zero n term. In this case a constant term is added to the function which changes the values of the maxima and minima, but not the graph itself. It is safe to assume that at least one term will have a non-zero n. Periodicity is defined to be a non-negative integer, but it is usually in the range 1–6.

Conformational search algorithm

We determine the conformation of a molecule by looking at the wells that its dihedral angles occupy. Since there are not typically more than a few wells for each dihedral angle, we can create a string of digits by numbering the wells. Once the wells are numbered, it is then feasible to follow the progress of the system as it is simulated by watching the changes in the strings being generated. When a new string is generated, the string is stored along with the time at which it was generated. We now define sampling rate as the rate at which new conformations are discovered. One feature of this method is that it can be generalized to all dihedral angles. More importantly, it is independent of the propagator used to actually generate new conformations. This method is justified by the fact that once a dihedral angle is within the well formed between two local maxima, the natural behavior of the angle is to approach the local minima.

As the number of atoms increases, the possible states of these strings increases exponentially. Even a small protein such as BPTI has more than 2000 dihedral angles. The majority of these contain hydrogen atoms and are immediately discounted from analysis due to the fast motion of the hydrogen. Even so, there are still hundreds of dihedrals each with multiple wells. With roughly 200 dihedrals from BPTI, say with 3 wells each, there

would be a state space of 3^{200} different strings. In practice, there are many non-physical states (e.g., overlapping configurations) and inaccessible states due to high local energy barriers. The rugged nature of the energy hyper-surface can not be alleviated by a clever propagator along the lines of SHMC, but requires methods such as potential smoothing or multi-canonical ensembles[2]. A method to generate the dihedral strings has been implemented in PROTOMOL and is listed as Algorithm 6.

Algorithm 6 Method for generating strings used in determining the sampling rate.

Preprocessing:
(1). Remove dihedrals containing H and multiple terms, keeping essential dihedrals
(2). Find maxima of Equation (4.1) using its derivative
(3). If the phase shift is nonzero, shift the critical points accordingly
(4). Enumerate wells for essential dihedrals left from (1)

Matching:
(1). Determine which well each essential dihedral occupies
(2). Form *conformation* string based on wells of essential dihedrals
(3). Update counter and time step for conformation string

Currently, when doing this analysis, any dihedral that has multiplicity greater than one is ignored. The reason these are ignored is that it is computationally non-trivial to determine the roots of an arbitrary function. Recall Figure 4.3 where the formula for the dihedral was of order 6, yet only 3 individual wells were formed. Now consider that there might be many such essential dihedrals per biological molecule. This computation would best be done in a preprocessing step for sets of dihedrals in a force field, and is left for future work.

4.3.3 Sampling efficiency

We now have several tools to help in determining whether our new sampling method is viable or not. What is needed now is a plan for comparing numbers across different methods. If a particular algorithm samples extremely well, but is one hundred times more computationally expensive than an existing algorithm, then it might not be a suitable

solution. In order to determine this, define sampling efficiency as the (computational) cost per new conformation. This value is calculated by dividing the running time of the simulation by the number of conformations discovered,

$$\text{Cost of sampling (CS)} = \frac{\text{Execution time (ET)}}{\text{Unique conformations (C)}}. \tag{4.3}$$

Obviously, comparisons of this value are only valid for simulations run on the same platform and the same input. However, this is fair metric when comparing different sampling methods, since it takes care of the overhead of more sophisticated trial moves, and any other effects on the quality (or lack thereof, e.g., correlation) of samples produced by different sampling techniques.

4.4 Observables

4.4.1 Average torsion energy

For n-butane, we are going to calculate the average torsion[3] energy. This can be obtained analytically at any temperature T where $\beta = 1/k_B T$ by

$$\left\langle U^{\mathrm{dih}}(\phi) \right\rangle_\beta = \frac{\int_0^{2\pi} U^{\mathrm{dih}}(\phi) \exp(-\beta U^{\mathrm{dih}}(\phi)) \mathrm{d}\phi}{\int_0^{2\pi} \exp(-\beta U^{\mathrm{dih}}(\phi)) \mathrm{d}\phi}. \tag{4.4}$$

We compare the analytical result at $T = 300\,K$ against averages obtained using HMC and SHMC*. We use the simple parameters for the butane dihedral in CHARMM 22:

$$U^{\mathrm{dih}}(\phi) = 1.6(1 + \cos(3\phi - \pi)) + 0.6(1 + \cos(\phi - \pi)), \tag{4.5}$$

where $U^{\mathrm{dih}}(\phi)$ has units of $\mathrm{kcal\ mol^{-1}}$ ($1\,\mathrm{kcal\ mol^{-1}} = 4.1868\ \mathrm{kJ\,mol^{-1}}$). Substituting Equation (4.5) into Equation (4.4), and evaluating this integral numerically for $\beta = 1/(k_{\mathrm{B}} 300\,K)$, where $k_{\mathrm{B}} = 0.00198719\ \mathrm{kcal\ mol^{-1}\,K^{-1}}$ we obtain

$$\left\langle U^{\mathrm{dih}}(\phi) \right\rangle_{\beta = 1/(k_{\mathrm{B}} 300\,K)} = 0.628\,48\ \mathrm{kcal\,mol^{-1}}. \tag{4.6}$$

[3] Torsion energy is the term used to describe the energy of a dihedral angle.

This expectation value is correct for butane because the torsion angle coordinate decouples from the rest of the internal coordinates (bonds and angles), and the functional determinant from Cartesian to internal motion is 1, cf. [11].

4.4.2 Average potential energy

For decalanine and BPTI, we calculate the average potential energy as listed in Equation (2.4). The potential for bonded interactions is described by a simple harmonic spring.

$$U^{\text{bond}} = \frac{1}{2} k_B \left(||\vec{x}_{ij}|| - l \right)^2 , \tag{4.7}$$

where k_B is Boltzmann's constant, $\vec{x}_{ij} = \vec{x}_j - \vec{x}_i$ is the distance between atoms i and j, and l is the equilibrium bond length between the atoms i and j. The potential for angle interactions is described by an angular bond between three atoms

$$U^{\text{angle}} = \frac{1}{2} k_A \left(\theta_{ijk} - \theta_0 \right)^2 , \tag{4.8}$$

where k_A is an angular force constant, θ_{ijk} the current angle, and θ_0 is the reference angle. The potential for dihedral and improper forces was given in Equation (4.1). The potential for an electrostatic pair-wise interaction is given by

$$U_{ij}^{\text{electrostatic}} = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{||\vec{x}_{ij}||}, \tag{4.9}$$

where ϵ_0 is a constant, and q_i and q_j are the charges of atoms i and j. The potential for the Lennard-Jones pair-wise interaction is

$$U_{ij}^{\text{Lennard-Jones}} = \frac{A_{ij}}{||\vec{x}_{ij}||^{12}} - \frac{B_{ij}}{||\vec{x}_{ij}||^{6}}, \tag{4.10}$$

where $A_{ij} \geq 0$ and $B_{ij} \geq 0$ are the LJ parameters for atoms i and j. Parameters for all of these equations are taken from CHARMM 22.

CHAPTER 5

RESULTS

In this chapter results and analysis from our simulations are given. In the first section we describe the tests on n-butane and show results of computing the average torsion energy. The following sections describe the results for decalanine and BPTI respectively.

5.1 n-butane

We ran simulations of 4 united-atom butane ($CH_3-CH_2-CH_2-CH_3$) using the dihedral energy of Equation (4.5) and 3 bond and 2 angle parameters coming from CHARMM 22. The mass of CH_3 is 15.035 u and CH_2 is 14.027 u. We ran 16 simulations of total length 114 ns at $T = 300\,K$. We tested values for the expected MD trajectory length $L_0 = \{630 \text{ fs}, 450 \text{ fs}, 234 \text{ fs}, 72 \text{ fs}\}$, where 630 fs $\approx \tau_{\max}$ for butane, using time steps $\delta t = \{8 \text{ fs}, 6 \text{ fs}, 3 \text{ fs}, 1 \text{ fs}\}$, where 8 fs is close to the stability limit of leapfrog for butane. The values computed from the simulations for the shortest and longest L are shown in Table 5.1

The error bar was estimated as twice the standard deviation computed using the block averaging method of Flyvbjerg & Petersen as explained in [12, p. 530]. It can be seen that all the values agree with the analytical result of Equation (4.6). This indicates that there is no bias in the methods. This is expected for HMC, but is what we want to show for SHMC*.

TABLE 5.1. EXPECTED VALUE OF THE TORSIONAL ENERGY U^{dih} FOR n-BUTANE

	L=72 fs			
	HMC		SHMC*	
δt (fs)	$\left\langle U^{\text{dih}}(\phi)\right\rangle$	AR	$\left\langle U^{\text{dih}}(\phi)\right\rangle$	AR
1	0.64$\pm$0.02	(100%)	0.62$\pm$0.02	(100%)
3	0.63$\pm$0.02	(96%)	0.62$\pm$0.02	(100%)
6	0.62$\pm$0.02	(79%)	0.63$\pm$0.02	(100%)
8	0.65$\pm$0.03	(51%)	0.65$\pm$0.08	(99%)
	L=630 fs			
	HMC		SHMC*	
δt (fs)	$\left\langle U^{\text{dih}}(\phi)\right\rangle$	AR	$\left\langle U^{\text{dih}}(\phi)\right\rangle$	AR
1	0.62$\pm$0.02	(100%)	0.64$\pm$0.02	(100%)
3	0.63$\pm$0.02	(96%)	0.63$\pm$0.02	(100%)
6	0.63$\pm$0.02	(79%)	0.64$\pm$0.02	(100%)
8	0.65$\pm$0.03	(51%)	0.67$\pm$0.02	(99%)

5.2 Decalanine

Figure 5.1 shows the acceptance rate of HMC and SHMC* for $L = 100$ on decalanine. Here we see that the acceptance of HMC suffers severely as the timestep is increased. SHMC*, on the other hand, shows no sign of decay. Data for the remaining values of L can be found in Table A.2.1. For low values of δt the percentages are virtually indistinguishable regardless of L. However, by the time $\delta t = 1.0$ there is already a significant decrease in acceptance for HMC even though this is not a particularly large timestep for this system. By the time $\delta t = 2.0$, it is apparent that SHMC* is much better at accepting moves than HMC for this set of inputs.

As was stated in Section 4.3.2, the AR is only part of the picture. Figure 5.2 shows the total number of new conformations discovered by HMC for decalanine. The x-axis displays the time step δt and the y-axis is the MD length L. The number of conformations decreases as δt increases. This is to be expected since the acceptance rate of HMC decreases dramatically with increasing step size. In Figure 5.3, on the other hand, as δt

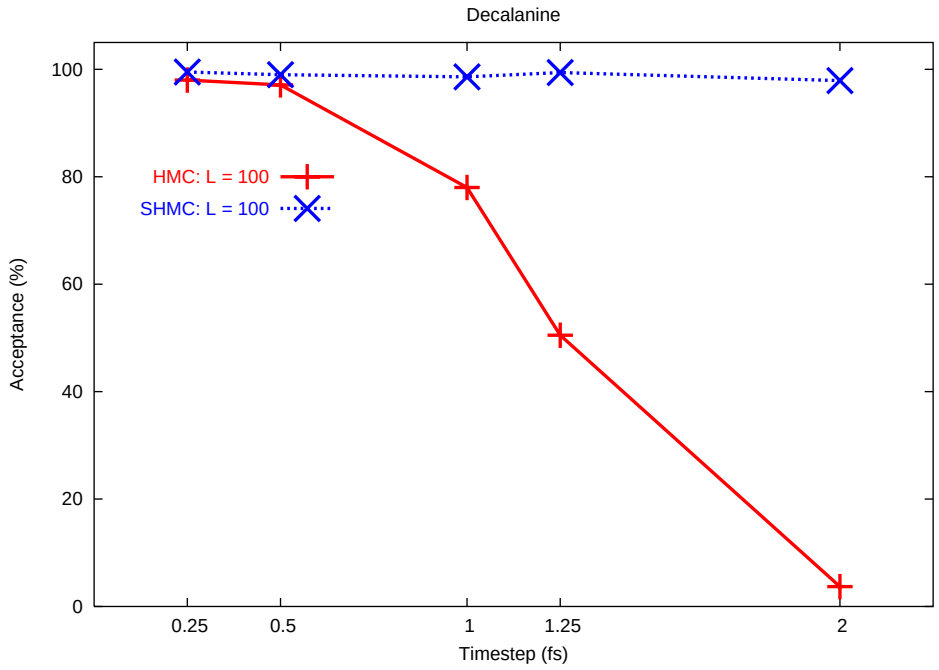

Figure 5.1. Acceptance rate of HMC and SHMC* for decalanine.

increases, there appears to be a parabolic shape to the percentages. There are several hypotheses as to why this is happening. One possible reason is that SHMC* is getting into a local minimum for which it is hard to get out of. The fact that there is similar behavior across all δt lends credence to this theory.

The decrease in the number of conformations in Figures 5.2 and 5.3 as L grows is artificial due to the fact that the simulation length was fixed across all values of L. As L increases, the number of HMC steps (and hence the number of possible conformations) must also decrease. For this reason Figures 5.2 and 5.3 were recalculated using percentages of possible conformations instead of absolute numbers. Figures 5.4 corresponds to 5.2 as do Figures 5.5 and 5.3. It should be noted that the order of L has been reversed from the previous graphs in order to facilitate better viewing of the 3-dimensional bars. There are two things to take from Figure 5.4. First, as L increases so does the percentage of newly discovered conformations. This is not unexpected since a larger L gives the MD

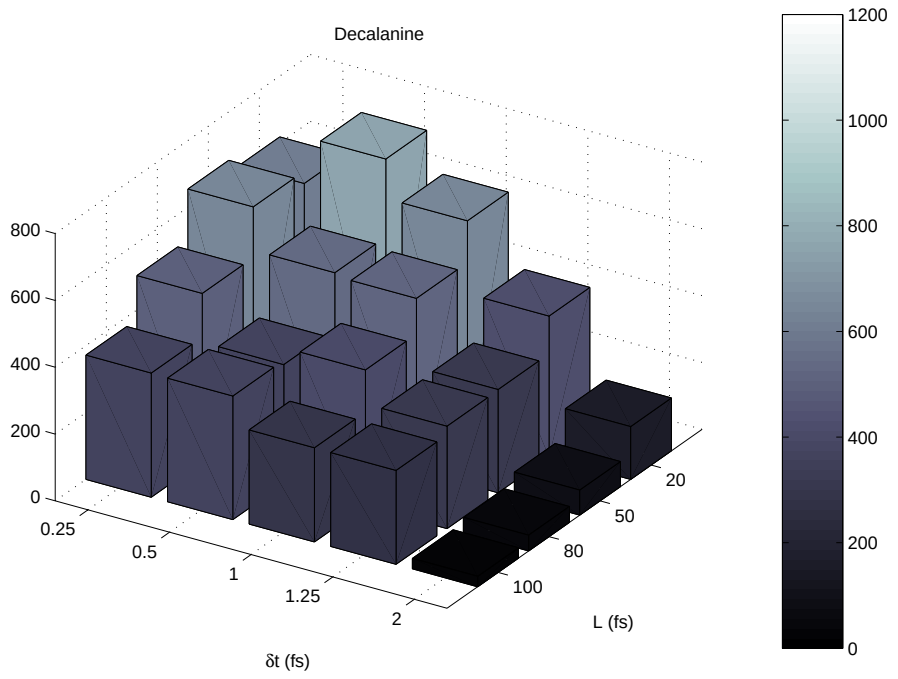

Figure 5.2. Number of new conformations discovered by HMC for decalanine.

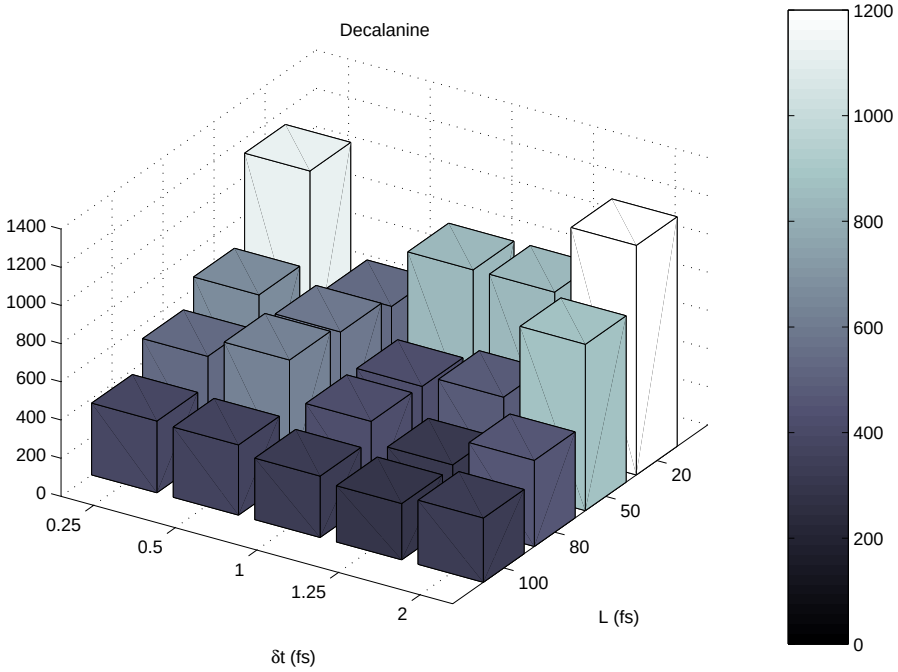

Figure 5.3. Number of new conformations discovered by SHMC* for decalanine.

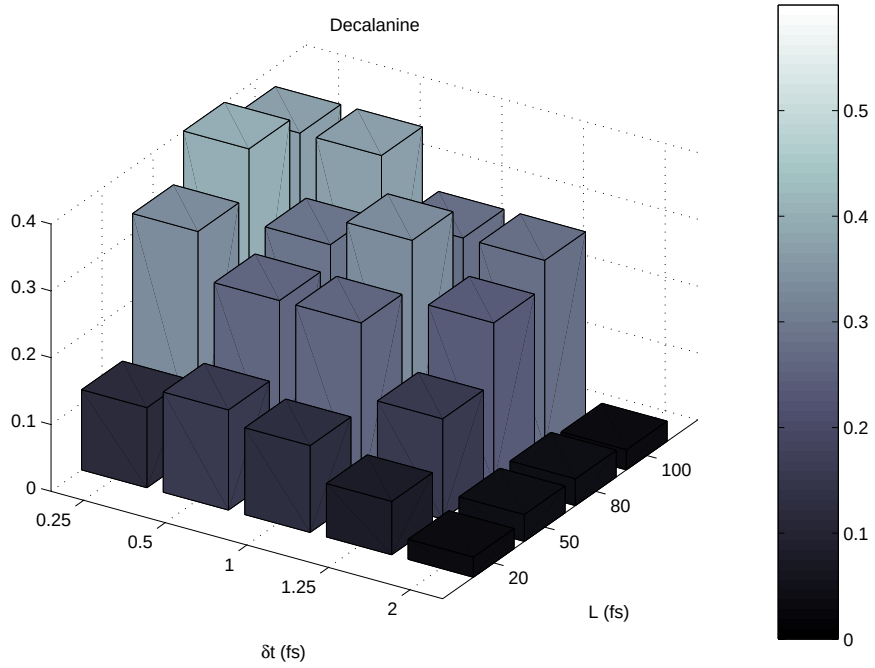

Figure 5.4. Percentage of new conformations discovered by HMC for decalanine.

time to explore further from its starting position. The second observation is that as δt increases, the percentages are decreasing. Again, not totally unexpected due to the decrease in the acceptance rate. Comparing this to Figure 5.5 we see the same general trend as L increases. On the other hand, we do not see the decrease in conformations with increasing δt. This is most likely due to the high acceptance rate of SHMC*. More successful moves are apt to produce more conformations.

Figure 5.6 has the efficiency of sampling for decalanine of both SHMC* and HMC plotted. There are four lines total with two lines for each method. The four lines correspond to the shortest and longest L values for this molecule. For the case where $L = 20$, SHMC* is slightly more efficient for all but one of the first four points. In the fifth point, however, it is apparent that HMC has begun to suffer due to its poor acceptance at this large timestep. When $L = 100$, there is virtually no difference in efficiency between the two methods for the first four points. Once again, HMC fails miserably as the timestep is

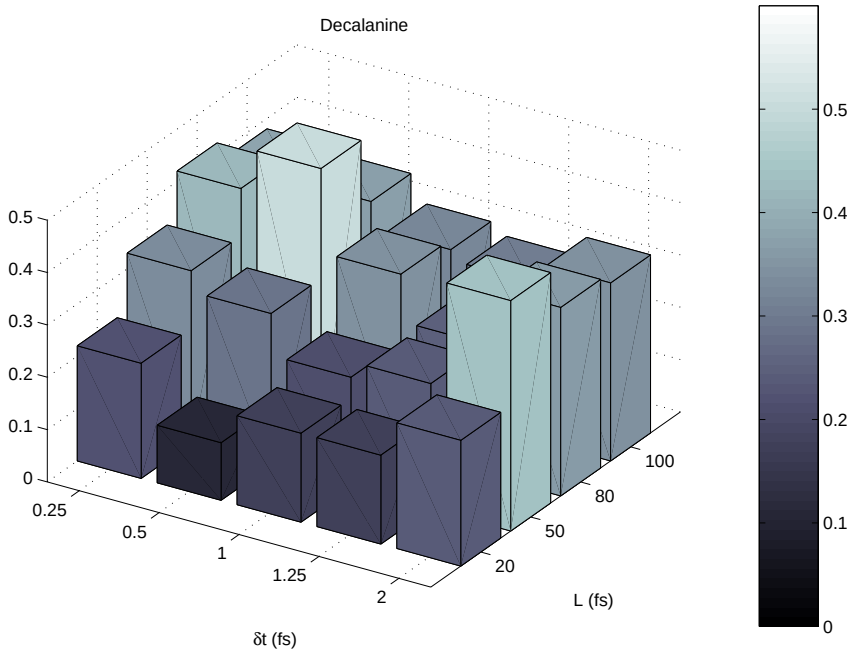

Figure 5.5. Percentage of new conformations discovered by SHMC* for decalanine.

extended. Notice, also, for both values of L that there appears to be a parabolic shape to the efficiency of HMC. This would imply that the optimal value of HMC for this molecule lies somewhere between 1 and 1.25. I am sure that SHMC* would exhibit a similar curve if we could extend the timestep further. Unfortunately, the fastest motions of decalanine and the limited accuracy in the Leapfrog integrator prevent further extension of the timestep.

Even for the simple decalanine, SHMC* shows great speedup over the traditional HMC. If we take the "best" values from both methods then we find a speedup of nearly three:

$$\frac{\text{Best efficiency HMC}}{\text{Best efficiency SHMC*}} = \frac{1.71}{0.59} = 2.89$$

Since SHMC* still appears to be decreasing as the timestep increases, we could assume that the speedup is even larger.

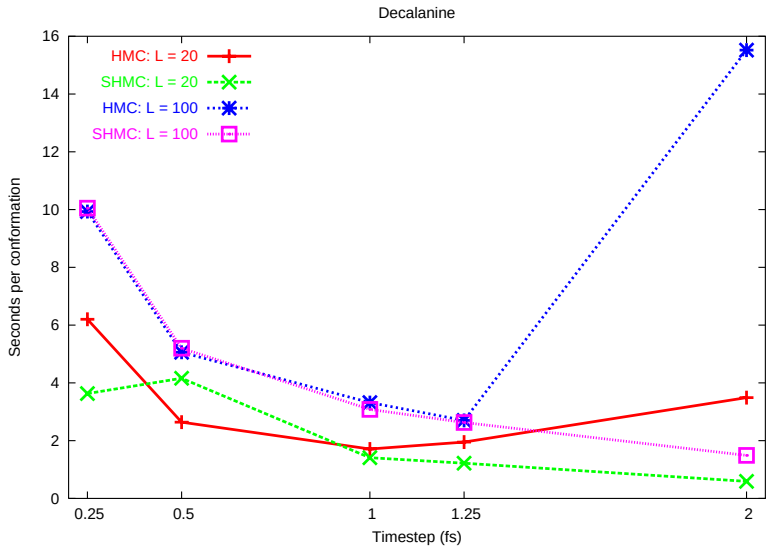

Figure 5.6. Cost per new conformation for SHMC* and HMC. Higher values denote lower efficiency of sampling.

Figure 5.7 shows the percentage overhead for SHMC* vs HMC. As can be expected, the runs that have the fewest number of n_{md} steps are those that have the highest overhead. In general though, it appears that SHMC* maintains a small percentage overhead. One particular run has an instance of HMC taking longer to complete than SHMC*. This is most likely due to external factors.

Figure 5.8 shows the unweighted average potential energy (PE) for SHMC* and HMC using the shortest and longest values of L. Since there was little difference between the two values of L for HMC, $L = 20$ was chosen as the reference point. Along with the average PE are plotted error bars corresponding to 2 standard deviations. As you can see from the plot, SHMC* still produces values that are statistically correct.

Attempts at reweighting the SHMC* PE for decalanine were unsuccessful due to the sometimes large differences between the total energy and the shadow energy. Once the actual SHMC algorithm is implemented, the parameter c will be used to keep these values within reasonable ranges.

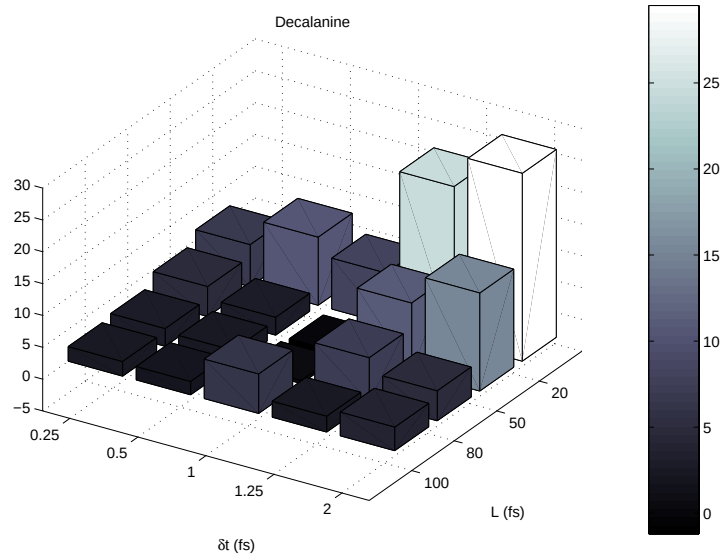

Figure 5.7. Percentage overhead for SHMC vs HMC on decalanine.

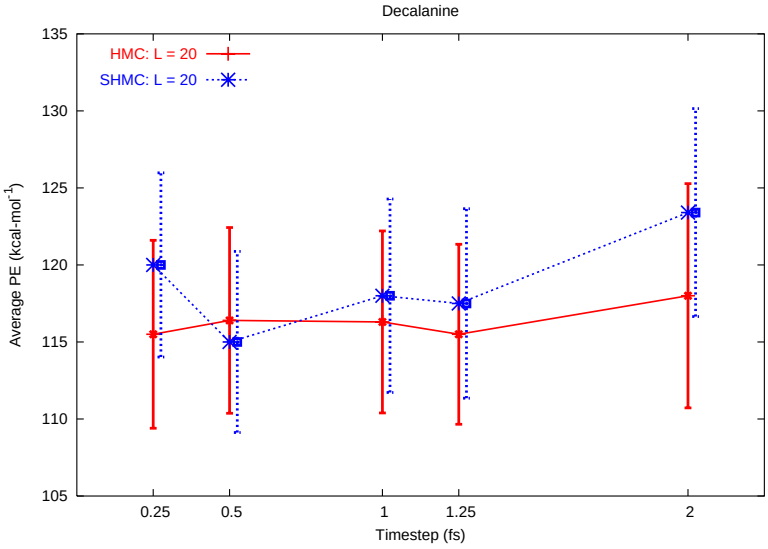

Figure 5.8. Average potential energy for decalanine with $L = 20$.

5.3 BPTI

Table 5.3 shows the acceptance rate of HMC and SHMC* for BPTI. While HMC has already started to significantly degrade, SHMC* is still accepting nearly all moves.

TABLE 5.2. ACCEPTANCE RATE OF HMC AND SHMC* FOR BPTI

	L=6		L=12		L=18		L=24	
δt	HMC	SHMC*	HMC	SHMC*	HMC	SHMC*	HMC	SHMC*
0.05	97.93	98.18	97.13	97.60	96.30	97.15	97.20	97.60
0.1	91.10	98.23	95.50	97.67	95.60	97.90	95.20	96.27
0.2	85.67	98.10	89.60	97.67	84.00	96.45	84.93	97.67
0.3	36.33	98.23	61.30	97.40	63.20	97.30	58.33	97.07
0.4	3.08	98.38	22.73	97.73	22.70	96.95	29.40	97.00

Figure 5.9 shows the number of conformations discovered by HMC. Notice the correlation in the decrease of discovered conformations as the time step increases. Figure 5.10 shows the same date for SHMC*. One interesting side effect of running SHMC* on larger molecules is that the state space of the conformations is so large, nearly every move produces a new conformation. This effect would lessen if we were to run the simulation significantly longer.

Figure 5.11 has the efficiency of sampling for BPTI of both SHMC* and HMC plotted. For both $L = 6$ and $L = 24$ there is very little difference in efficiency between the two methods for the first three values of δt. As was seen in Figure 5.6, SHMC* is clearly more efficient time step increases. Again we see the same parabolic shape for the HMC graphs. Clearly, the optimum value for SHMC* is much larger than that for HMC. Based on the current graph, we see a speedup of approximately 3 for the shorter L and approximately 8 for the longer L. It should be apparent that SHMC* is still decreasing, particularly for the larger value of L.

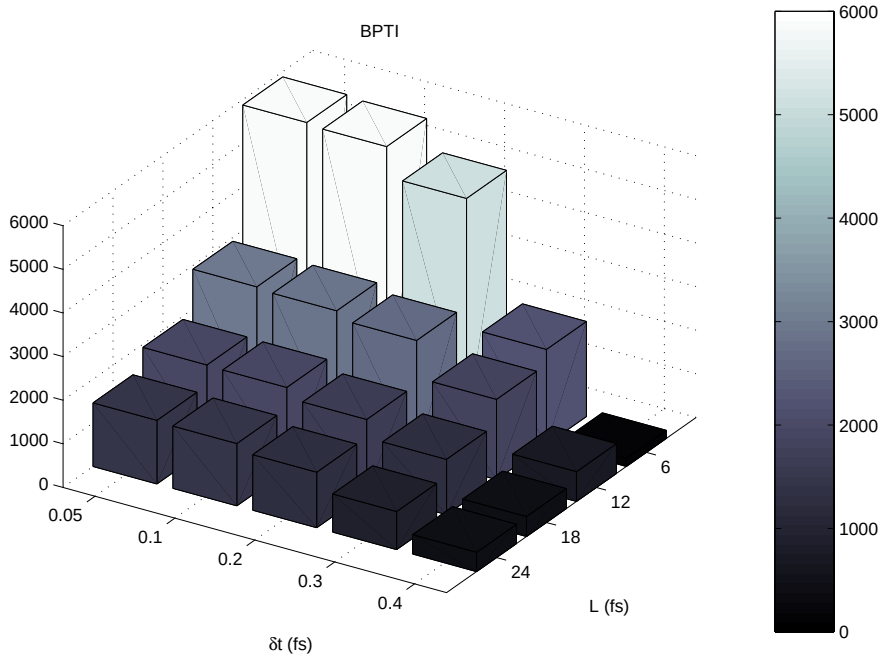

Figure 5.9. Number of new conformations discovered by HMC for BPTI.

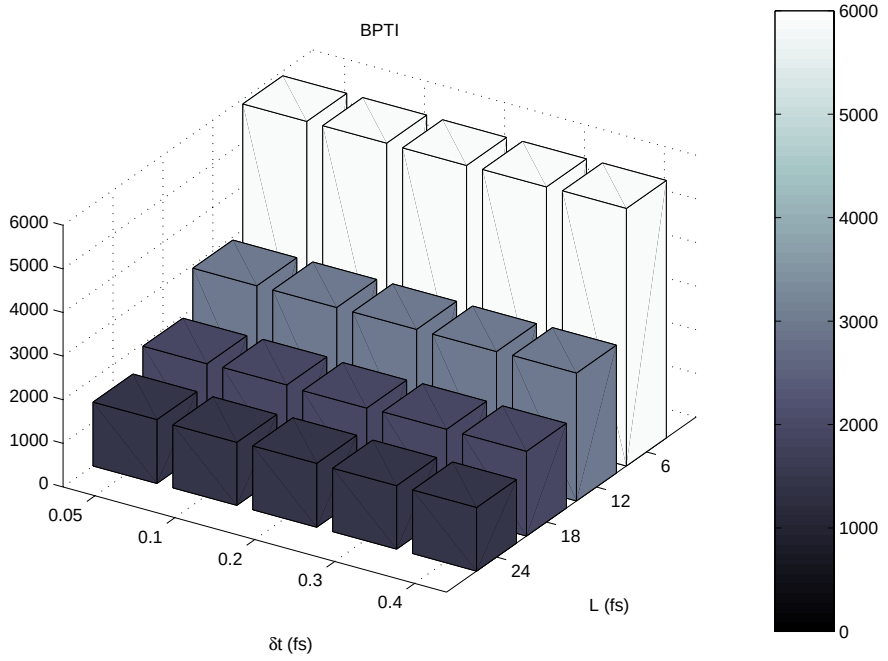

Figure 5.10. Number of new conformations discovered by SHMC* for BPTI.

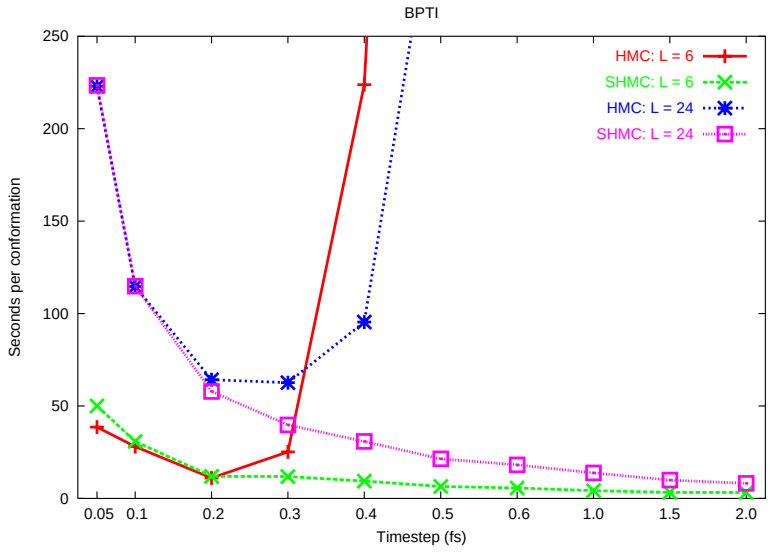

Figure 5.11. Cost per new conformation for SHMC* and HMC. Higher values denote lower efficiency of sampling.

Figure 5.12 shows the overhead of SHMC* for BPTI. There is significant overhead for all time steps within $L = 6$, but this is to be expected since there are so few MD steps n_{md}. For the rest of the values of L, the overhead is much more reasonable. There are several values less than 1%.

Figure 5.13 shows the unweighted average potential energy (PE) for SHMC* and HMC using the shortest and longest values of L. As in the previous example with decalanine, the average PE are plotted error bars corresponding to 2 standard deviations. The values show that for larger molecules, the bias in SHMC* is even more negligible.

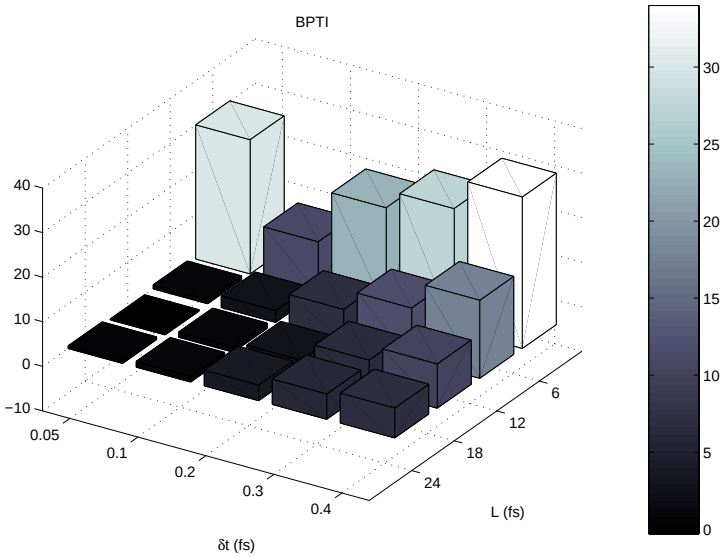

Figure 5.12. Percentage overhead for SHMC vs HMC on BPTI.

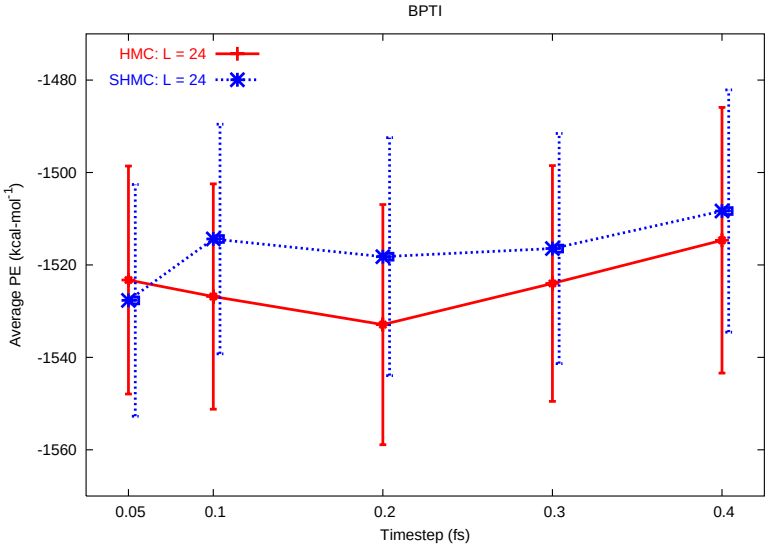

Figure 5.13. Average potential energy for BPTI with $L = 24$.

CHAPTER 6

SUMMARY AND FUTURE WORK

SHMC* has been shown to be an adequate approximation to the true SHMC method, but an implementation of the rigorous version of SHMC has recently been completed. The correct method will allow us to concentrate on optimizations since the goal of the current implementation was proof-of-concept and not speed. One of our main objectives with SHMC has been the sampling of large biomolecules, but as N grows we will undoubtedly require higher order shadow Hamiltonians to maintain the current level of competitiveness. PROTOMOL has been used as the test bed for SHMC, but before we release SHMC for public use, the code should be generalized so that a user could specify a desired accuracy of the shadow Hamiltonian at runtime. This would allow users to decide for themselves the cost/performance ratio that best suits their needs.

While not explored in this thesis, the combining of SHMC with other simulation techniques, such as multiple time stepping (MTS) methods, has begun. Not directly related to SHMC, but important nonetheless, is the recent implementation of the NPT ensemble in PROTOMOL. This extension will aid in comparison with realistic experiments. Along these same lines, we would like to see SHMC combined with methods sampling from multi-canonical ensembles so that it can be used for large scale problems such as protein folding [34] and conformational dynamics for drug design [32].

The current sampling metrics are another area for extension. All of the existing analysis was done by hand and tailored to the molecules tested herein. If this analysis could

be generalized for all molecules, it would give users of PROTOMOL more leeway and justification for deciding which method was most appropriate.

Acknowledgments

I would like to thank my advisor, Dr. Jesús Izaguirre, without whose dedication, encouragement, leadership, and advice none of this would have been possible. In addition, we would like to thank the following for their support and helpful comments:

- Jesús A. Izaguirre NSF ACI Career Award ACI-0135195
- Hong Hu – initial theoretical analysis of the performance of SHMC.
- Robert D. Skeel – Computer Science, University of Illinois Urbana-Champaign – help with the rigorous proof of SHMC.
- David Hardy – Computer Science, University of Illinois Urbana-Champaign – initial implementation of shadow Hamiltonian.
- Ed Maginn – Chemical Engineering, University of Notre Dame – suggested reweighting of values produced by SHMC.
- Gary Huber – BioEngineering, University of California San Diego – help with counting the conformations
- Thierry Matthey – Parallab, University of Bergen – help with PROTOMOL.
- Scott Hampton's support through an Arthur J. Schmitt fellowship
- Theoretical and Computational Biophysics group at UIUC
- Many of the results of this thesis were accumulated on BOB, a Linux cluster created with funds from the National Science Foundation under grant DMR-0079647.

APPENDIX A

SUPPLEMENTAL INFORMATION

A.1 Images of tested molecules

A.2 Simulation data

In this section I am listing the data used in the plots in Chapter 5 for reference.

A.2.1 Decalanine

TABLE A.1. ACCEPTANCE RATE OF HMC AND SHMC* FOR DECALANINE

	L=20		L=50		L=80		L=100	
δt	HMC	SHMC*	HMC	SHMC*	HMC	SHMC*	HMC	SHMC*
0.25	98.96	99.18	99.20	99.25	98.96	98.88	98.00	99.50
0.5	96.68	99.26	96.40	98.80	97.36	99.04	97.10	99.00
1.0	78.42	98.84	76.95	98.80	75.84	98.08	78.00	98.60
1.25	61.50	99.06	59.60	97.85	63.44	98.40	50.50	99.40
2.0	6.28	98.74	6.20	99.15	6.56	97.92	3.70	97.90

A.2.2 BPTI

TABLE A.2. DATA FOR FIGURE 5.2

Unique Conformations: HMC					
$L\backslash\delta t$	0.25	0.5	1.0	1.25	2.0
20	618	758	640	422	158
50	655	525	515	309	75
80	503	357	408	306	47
100	372	370	282	281	31

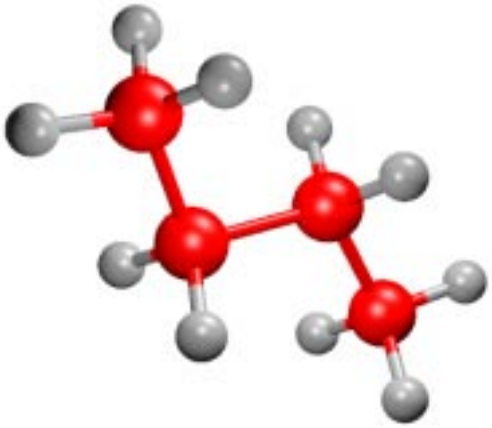

Figure A.1. A 14-atom n-butane.

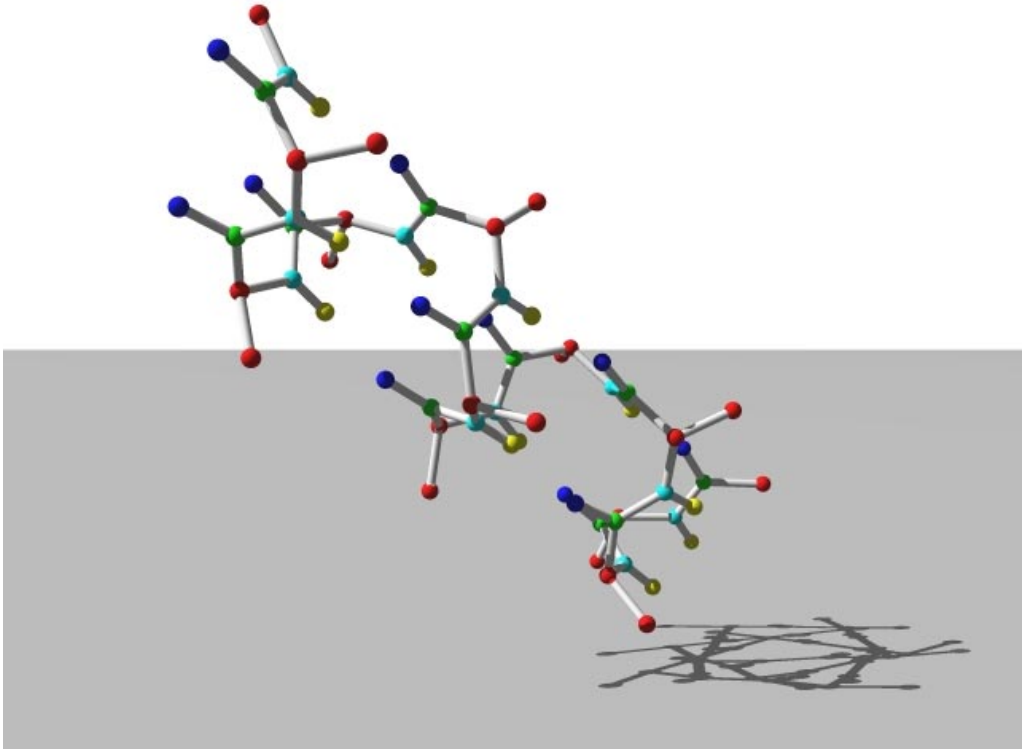

Figure A.2. A 66-atom decalanine.

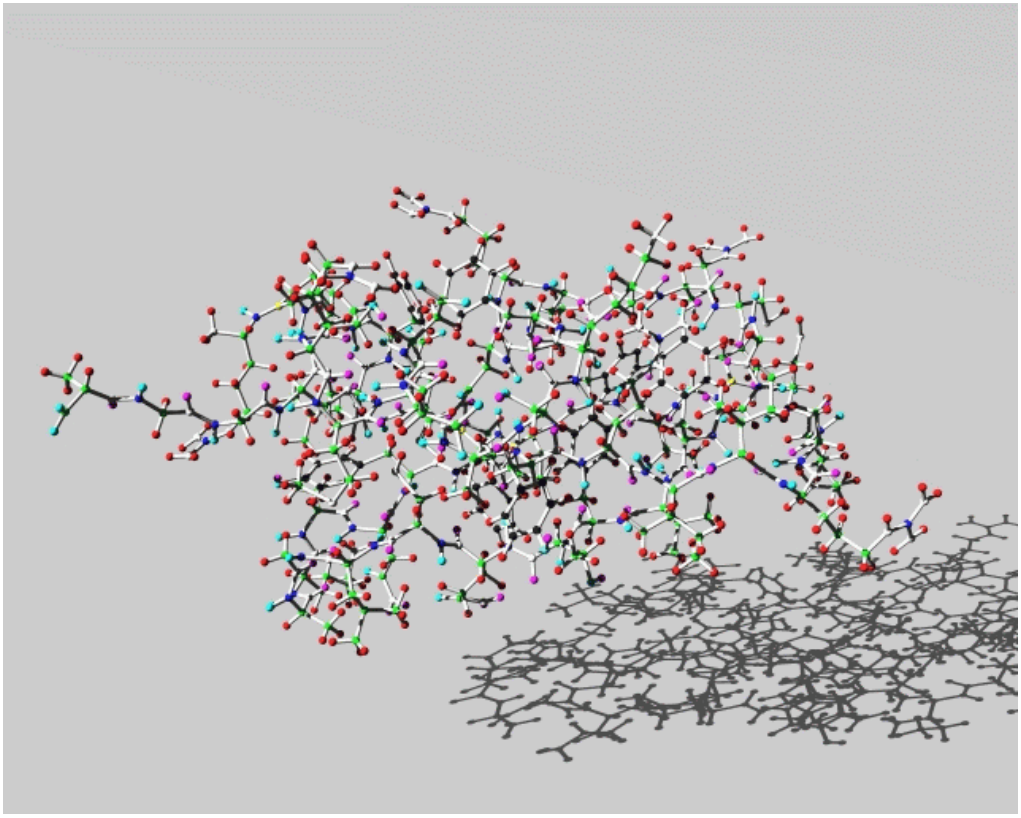

Figure A.3. An unsolvated BPTI with 882 atoms.

TABLE A.3. DATA FOR FIGURE 5.3

Unique Conformations: SHMC*					
$L\backslash\delta t$	0.25	0.5	1.0	1.25	2.0
20	1123	531	840	840	1202
50	662	585	417	477	871
80	527	625	421	310	450
100	376	368	322	295	336

TABLE A.4. DATA FOR FIGURE 5.4

Percentage Conformations: HMC					
$L\backslash\delta t$	0.25	0.5	1.0	1.25	2.0
20	0.37	0.37	0.28	0.28	0.03
50	0.40	0.29	0.33	0.24	0.04
80	0.33	0.26	0.26	0.15	0.04
100	0.12	0.15	0.13	0.08	0.03

TABLE A.5. DATA FOR FIGURE 5.5

Percentage Conformations: SHMC*					
$L\backslash\delta t$	0.25	0.5	1.0	1.25	2.0
20	0.38	0.37	0.32	0.30	0.34
50	0.42	0.50	0.34	0.25	0.36
80	0.33	0.29	0.21	0.24	0.44
100	0.22	0.11	0.17	0.17	0.24

TABLE A.6. DATA FOR FIGURE 5.6

Seconds per Conformation						
	$L\backslash\delta t$	0.25	0.5	1.0	1.25	2.0
HMC	20	6.20	2.64	1.71	1.95	3.49
	100	9.93	5.06	3.32	2.70	15.52
SHMC*	20	3.63	4.16	1.41	1.22	0.59
	100	10.05	5.20	3.08	2.63	1.49

TABLE A.7. DATA FOR FIGURE 5.7

SHMC* Overhead versus HMC					
$L\backslash\delta t$	0.25	0.5	1.0	1.25	2.0
20	6.58	10.66	7.95	24.48	29.53
50	4.71	2.81	-1.17	10.91	15.42
80	2.75	2.33	0.51	7.00	4.68
100	2.33	2.08	6.20	2.51	3.74

TABLE A.8. DATA FOR FIGURE 5.8

HMC: Average Potential Energy					
δt	0.25	0.5	1.0	1.25	2.0
L = 20	115.5	116.4	116.3	115.5	118.0
σ	6.10	6.03	5.91	5.84	7.28
SHMC*: Average Potential Energy					
δt	0.25	0.5	1.0	1.25	2.0
L = 20	120.0	115.0	118.0	117.50	123.4
σ	5.98	5.88	6.28	6.15	6.75

TABLE A.9. DATA FOR FIGURE 5.9

Unique Conformations: HMC					
$L\backslash\delta t$	0.05	0.1	0.2	0.3	0.4
6	5876	5826	5140	2180	185
12	2914	2865	2688	1839	682
18	1926	1912	1680	1264	454
24	1458	1428	1274	875	441

TABLE A.10. DATA FOR FIGURE 5.10

Unique Conformations: SHMC*					
$L\backslash\delta t$	0.05	0.1	0.2	0.3	0.4
6	5891	5894	5886	5894	5903
12	2928	2930	2930	2922	2932
18	1943	1958	1929	1946	1939
24	1464	1444	1465	1456	1455

TABLE A.11. DATA FOR FIGURE 5.11

Seconds per Conformation						
	$L\backslash\delta t$	0.05	0.1	0.2	0.3	0.4
HMC	6	38.57	28.02	11.10	25.19	223.78
	24	222.96	114.62	64.25	62.60	95.44
SHMC*	6	50.06	30.85	11.95	11.85	9.40
	24	223.46	114.82	57.84	39.75	30.86

TABLE A.12. DATA FOR FIGURE 5.12

SHMC* Overhead versus HMC					
$L\backslash\delta t$	0.05	0.1	0.2	0.3	0.4
6	0.30	0.11	0.23	0.27	0.34
12	0.01	0.03	0.07	0.12	0.17
18	0.00	0.01	0.02	0.07	0.10
24	0.01	0.01	0.04	0.06	0.07

TABLE A.13. DATA FOR FIGURE 5.13

HMC: Average Potential Energy					
δt	0.05	0.1	0.2	0.3	0.4
L = 24	-1523.26	-1526.83	-1532.91	-1524.01	-1514.66
σ	24.67	24.38	25.99	25.52	28.75
SHMC*: Average Potential Energy					
δt	0.05	0.1	0.2	0.3	0.4
L = 24	-1527.67	-1514.39	-1518.21	-1516.45	-1508.33
σ	25.07	24.85	25.75	24.91	26.21

BIBLIOGRAPHY

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Clarendon Press, Oxford, New York, 1987. Reprinted in paperback in 1989 with corrections.

[2] B. J. Berne and J. E. Straub. Novel methods of sampling phase space in the simulation of biological systems. *Curr. Topics in Struct. Biol.*, 7:181–189, 1997.

[3] S. D. Bond, B. J. Leimkuhler, and B. B. Laird. The Nosé–Poincaré method for constant temperature molecular dynamics. *J. Comput. Phys*, 151(1):114–134, 1999.

[4] A. Brass, B. J. Pendleton, Y. Chen, and B. Robson. Hybrid Monte Carlo simulations theory and initial comparison with molecular dynamics. *Biopolymers*, 33:1307–1315, 1993.

[5] J. B. Clarage, T. Romo, B. K. Andrews, B. M. Pettitt, and G. N. Philips. A sampling problem in molecular dynamics simulations of macromolecules. *Proc. Natl. Acad. Sci. USA*, 92:3288–3292, 1995.

[6] M. Creutz. Global Monte Carlo algorithms for many-fermion systems. *Phys. Rev. D*, 38(4):1228–1238, 1988.

[7] M. Creutz and A. Gocksch. Higher-order hybrid monte carlo algorithms. *Phys. Rev. Lett.*, 63(1):9–12, 1989.

[8] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Phys. Lett. B*, 195:216–222, 1987.

[9] S. E. Feller, D. Yin, R. W. Pastor, and A. D. MacKerell Jr. Molecular dynamics simulation of unsaturated lipids at low hydration: Parametrization and comparison with diffraction studies. *Biophys. J.*, 73(5):2269–2279, 1997.

[10] S. E. Feller, Y. Zhang, R. W. Pastor, and B. R. Brooks. Constant pressure molecular dynamics simulation: The Langevin piston method. *J. Chem. Phys.*, 103(11):4613–4621, 1995.

[11] A. Fischer, F. Cordes, and C. Schüette. Hybrid Monte Carlo with adaptive temperature in mixed-canonical ensemble: Efficient conformational analysis of RNA. *J. Comp. Chem.*, 19(15):1689–1697, 1998.

[12] D. Frenkel and B. Smit. *Understanding Molecular Simulation, Second Edition*. Academic Press, San Diego, 2002.

[13] E. Hairer and C. Lubich. Asymptotic expansions and backward analysis for numerical integrators. In *Dynamics of Algorithms*, pages 91–106, New York, 2000. IMA Vol. Math. Appl 118, Springer-Verlag.

[14] S. Hampton and J. A. Izaguirre. Improved sampling for biological molecules using Shadow Hybrid Monte Carlo. Accepted in *International Conference on Computational Science* (ICCS 2004), Poland, 2004.

[15] W. F. Humphrey, A. Dalke, and K. Schulten. VMD – Visual Molecular Dynamics. *J. Mol. Graphics*, 14:33–38, 1996.

[16] A. D. Kennedy and B. Pendleton. Acceptances and autocorrelations in hybrid Monte Carlo. *Nuclear Physics B (Proc. Suppl.)*, 20:118–121, 1991.

[17] P. D. Kirchhoff, M. B. Bass, B. A. Hanks, J. Briggs, A. Collet, and J. A. McCammon. Structural fluctuations of a cryptophane host: A molecular dynamics simulation. *J. Am. Chem. Soc.*, 118:3237–3246, 1996.

[18] D. E. Knuth. *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, Reading, Massachusetts, 3rd edition, 1997.

[19] A. Ko. MDSimAid: An automatic recommender for optimization of fast electrostatic algorithms for molecular simulations. Master's thesis, University of Notre Dame, Notre Dame, Indiana, USA, Dec. 2002.

[20] P. B. Mackenzie. An improved hybrid Monte Carlo method. Technical Report FERMILAB-Pub-89/100-T, Fermi National Accelerator Laboratory, 1989.

[21] A. D. MacKerell Jr., D. Bashford, M. Bellott, R. L. Dunbrack Jr., J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, I. W. E. Reiher, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. All-hydrogen empirical potential for molecular modeling and dynamics studies of proteins using the CHARMM22 force field. *J. Phys. Chem. B*, 102:3586–3616, 1998.

[22] A. D. MacKerell Jr., D. Bashford, M. Bellott, R. L. Dunbrack Jr., J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. Self-consistent parameterization of biomolecules for molecular modeling and condensed phase simulations. *FASEB J.*, A143:6, 1992.

[23] T. Matthey. *Framework Design, Parallelization and Force Computation in Molecular Dynamics*. PhD thesis, University of Bergen, Bergen, Norway, 2002.

[24] T. Matthey, T. Cickovski, S. Hampton, A. Ko, Q. Ma, T. Slabach, and J. A. Izaguirre. PROTOMOL: an object-oriented framework for prototyping novel algorithms for molecular dynamics. accepted in ACM Trans. Math. Softw., 2004.

[25] E. Merritt and D. Bacon. Raster3d: Photorealistic molecular graphics. *Methods in Enzymology*, 277:505–524, 1997.

[26] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[27] D. of Chemistry and Biochemistry. BoB: Bunch-o-boxes beowulf cluster. Available online via `http://bob.nd.edu/`, 2000–2003.

[28] M. J. Potter, P. D. Kirchhoff, H. A. Carlson, and J. A. McCammon. Molecular dynamics of cryptophane and its complexes with tetramethylammonium and neopentane using a continuum solvent model. *J. Comp. Chem.*, 20:956–970, 1999.

[29] J.-P. Ryckaert and A. Bellemans. Molecular dynamics of liquid alkanes. *Faraday Discussions*, 66:95–106, 1978.

[30] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems*. Chapman and Hall, London, 1994.

[31] T. Schlick. *Molecular Modeling and Simulation - An Interdisciplinary Guide*. Springer-Verlag, New York, NY, 2002.

[32] C. Schüette, A. Fischer, W. Huisinga, and P. Deuflhard. A direct approach to conformational dynamics based on hybrid Monte Carlo. *J. Comput. Phys*, 151(1):146–168, 1999.

[33] R. D. Skeel and D. J. Hardy. Practical construction of modified Hamiltonians. *SIAM J. Sci. Comput.*, 23(4):1172–1188, 2001.

[34] R. Zhou, B. J. Berne, and R. Germain. The free energy landscape for β hairpin folding in explicit water. *Proc. Natl. Acad. Sci. USA*, 98:14931–14936, 2001.