

A Monte Carlo Algorithm For Closed Polymer Chains
With an Application to the Cyclization Problem

by

Monica Lynn Shaw, B.S.

Report

Presented to the Faculty of the Graduate School
of the University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Computational and Applied Mathematics

The University of Texas at Austin

August 2003

A Monte Carlo Algorithm For Closed Polymer Chains
With an Application to the Cyclization Problem

APPROVED BY SUPERVISING COMMITTEE:

Oscar Gonzalez

Clint Dawson

Contents

1	Introduction	1
2	Markov Chain Monte Carlo (MCMC) Methods	2
2.1	Monte Carlo Integration	2
2.2	Markov Chain Monte Carlo	3
3	MCMC Sampling Methods	5
3.1	The Metropolis Hastings Algorithm (MH)	5
3.2	The Gibbs Sampler	5
4	The Rigid Body Chain Model	7
5	Configuration Variables	8
5.1	Absolute variables (Q^a, r^a)	8
5.2	Junction variables (Λ^a, v^a)	8
5.2.1	Tilt-Roll-Twist Angles	9
5.2.2	Bend-Wrench-Turn Angles	9
6	Energy of Chains	11
6.1	Potential energy	11
6.2	Kinetic energy, momentum	12
7	Chain Statistics	13
7.1	Hamiltonian, distribution functions	13
8	The Cyclization Problem	14
8.1	Probability Gradient	14
8.1.1	Proposition 1	14
8.1.2	Proposition 2	15

9	MCMC Methods	16
9.1	Sampling Methods	16
9.1.1	Direct sampling	16
9.1.2	Transformation methods	17
9.1.3	Sampling Open Polymer Chains	17
9.2	Simulating Closed Chains	18
9.3	The Metropolis-Hastings Step	19
10	Numerical Results	21
10.1	Goals of MCMC Simulations	21
10.2	Chain Parameters	21
10.3	Calculation of $\langle \frac{\partial \langle E \rangle}{\partial \phi} \rangle$	22
10.4	Monitoring MCMC Convergence	24
10.4.1	CUSUM Convergence	24
10.4.2	Sub-Averaging	25
11	Conclusion	27
12	Appendix	28
12.0.3	statechange() pseudocode	28
12.0.4	chainangles() pseudocode	28
12.0.5	Markov Chain Monte Carlo pseudocode	29

1 Introduction

Markov Chain Monte Carlo algorithms have been used extensively in polymer statistical mechanics. Of particular interest is its application to the study of the thermodynamic and structural properties of DNA. DNA is a stiff double helical polymer made of two antiparallel strands of deoxyribonucleotide chains. Each DNA strand is comprised of a sugar-phosphate backbone and attached bases. The two strands are connected by hydrogen bonding with paired bases: adenine(A) with thymine(T), and cytosine(C) with guanine(G). We can assume that the macroscopic structure of DNA can be modeled by a single rigid body chain where each body represents a basepair. It is desirable to obtain a set of these chain configurations for study. In this way, Markov chains provide a useful method for modeling these chains. A Markov chain represents a set of random variables X_t having the property that, given the present, the future is conditionally independent of the past. Note that a "polymer chain" refers to a single polymer configuration, while a "Markov chain" refers to a set of these polymer chains. We will see that Monte Carlo integration proves to be a powerful tool for tackling some difficult problems in polymer modelling.

One such problem is the question of cyclization: what is the probability that an open chain becomes closed? This probability depends on the material properties of a polymer chain and MCMC can be used to characterize how changes in a polymer chain's properties affect the probability of closure.

In this report, we will give a brief overview of MCMC methods, sampling techniques, and implementation issues. We will discuss the geometry of polymer chains and the probability distribution that describes them. Additionally, we will give an outline of the cyclization problem and how MCMC can be applied to this problem. Lastly, we will present the results of my numerical simulations for this problem.

2 Markov Chain Monte Carlo (MCMC) Methods

2.1 Monte Carlo Integration

Our goal is to estimate equilibrium averages and event probabilities of the form

$$F = \int_Q f(u)\rho(u) du \quad (2.1)$$

where $\rho(u) \in \mathbf{R}$ is a probability distribution on the variable $u \in Q$, Q is an m -dimensional domain in \mathbf{R}^m and $f(u) \in \mathbf{R}^n$ is a given function.

Monte Carlo (MC) has proven to be a powerful numerical method for estimating integrals of this type. The basic idea behind MC is as follows:

We first generate a sample set $S_N = \{u_1, \dots, u_N\}$ from $\rho(u)$, $u \in Q$. S_N is a *sample set* from ρ if for any cube or cell $C \subset Q$

$$\frac{\# \text{ samples } u_i \text{ in } C}{N} \approx \int_C \rho(u) du$$

or, alternatively, if

$$\rho[S_N](u) \approx \rho(u)$$

where $\rho[S_N](u)$ is a piecewise constant probability distribution or “normalized histogram” defined by S_N .

We suppose the method for generating the sample set is consistent in the sense that

$$\frac{\# \text{ samples } u_i \text{ in } C}{N} \rightarrow \int_C \rho(u) du \quad \text{as } N \rightarrow \infty$$

for any fixed C .

We then approximate F using S_N ; for example,

$$F = \int_Q f(u)\rho(u) du \approx \int_Q f(u)\rho[S_N](u) du \approx F_N \quad (2.2)$$

where

$$F_N = \frac{1}{N} \sum_{i=1}^N f(u_i), \quad u_i \in S_N \quad (2.3)$$

is the average or mean of f over the sample set.

F_N is just one example of an *estimator* for approximating F from S_N . A family of estimators is given by

$$F_{w,N} = \sum_{i=1}^N w_i f(u_i), \quad u_i \in S_N \quad (2.4)$$

where $w = (w_1, \dots, w_N)$ are any constants such that $\sum_i w_i = 1$.

The integral F is a definite, fixed quantity. In contrast, the estimator F_N is a random quantity; its value depends on the sample set S_N .

2.2 Markov Chain Monte Carlo

In many statistics problems, it is impossible to directly simulate an independent sample S_N from $\rho(u)$. Fortunately there are sampling methods available to simulate a *Markov chain* $\{X_1, \dots, X_N\}$ which converges in distribution to our target distribution, $\rho(u)$.

Simply put, a Markov Chain, X , is a discrete time stochastic process $\{X_0, X_1, \dots\}$ with the property that the distribution of X_t given events X_0, X_1, \dots, X_{t-1} depends only on X_{t-1} :

$$P(X_t \in A | X_0, X_1, \dots, X_{t-1}) = P(X_t \in A | X_{t-1}) \quad (2.5)$$

for any set A .

The probability on the right describes the transition between two states. Because the probability of a transition depends on A , it is natural to define the chain in terms of its transition kernel density, $K(X_t, X_{t+1})$ which has the property

$$P(X_{t+1} \in A | X_t) = \int_A K(X_t, dX). \quad (2.6)$$

A typical example of a Markov chain is a random walk process in which $X_{n+1} = X_n + \epsilon_n$ in which ϵ_n is generated independently of X_n, X_{n-1}, \dots [1].

Simulation of a Markov chain requires an initial state X_1 . Subject to regularity conditions, a Markov chain will gradually "forget" this initial state and its distribution will converge to $\rho(u)$. Several conditions must be satisfied in order to achieve a convergent Markov chain. First of all, the Markov chain must be irreducible, that is, from any X_1 , the chain must be able to reach any point in the sample space with positive probability in a finite number of iterations. The Markov chain must also be aperiodic: it cannot repeat itself after a finite number of steps. Lastly, the chain must be positive recurrent: if X_1 is sampled from $\rho(u)$, then eventually all subsequent iterates must be distributed according to $\rho(u)$ [5].

Markov Chain Monte Carlo refers to the procedure of simulating Markov chains and using these chains to draw conclusions about $\rho(u)$. There are several methods available for generating a Markov chain, the most common of which are the Metropolis-Hastings method and the Gibbs sampler.

3 MCMC Sampling Methods

The two most common MCMC algorithms are the Metropolis-Hastings method and the Gibbs sampler. In the following discussion, let $X_t \in A \subset \mathbf{R}^k$ denote the (k -dimensional) state of a Markov chain at time t .

3.1 The Metropolis Hastings Algorithm (MH)

MH algorithms generate Markov chains that converge to the target distribution, $\rho(u)$, by successively sampling from an arbitrary transition kernel density, K , and imposing a random rejection step at each transition. Let $\alpha(x, y)$ be a function taking values in $[0, 1]$. If the chain is in state $X_t = x$ at time t , MH generates a transition candidate y from the distribution $K(x, \cdot)$ for the next state, X_{t+1} . This candidate is accepted as the new state so that $X_{t+1} = y$ with probability $\alpha(x, y)$, otherwise it is rejected and $X_{t+1} = x$.

It has been shown that the optimal form of the acceptance function is given by

$$\alpha(x_t, x_{t+1}) = \min\left\{1, \frac{\rho(x_{t+1})K(x_{t+1}, x_t)}{\rho(x_t)K(x_t, x_{t+1})}\right\}. \quad (3.7)$$

This choice of $\alpha(x_t, x_{t+1})$ rejects suitable candidates less often than other forms [8]. When the transition density satisfies $K(x_{t+1}, x_t) = K(x_t, x_{t+1})$, we get

$$\alpha(x_t, x_{t+1}) = \min\left\{1, \frac{\rho(x_{t+1})}{\rho(x_t)}\right\}. \quad (3.8)$$

In this case, we need only know $\rho(u)$ up to a normalizing constant in order to evaluate $\alpha(x_t, x_{t+1})$. We will make extensive use of this algorithm later in polymer chains sampling.

3.2 The Gibbs Sampler

Although we will not be applying the Gibbs sampler to our study of polymer chains, it is worth noting this widely used sampling method. The Gibbs sampler is a special case of the Metropolis-Hastings algorithm. It splits the k -dimensional parameter space into $p \leq k$ blocks.

Let $\rho(x) = \rho(x_1, \dots, x_p)$, $x \in \mathbf{R}^k$ denote the joint density of x and $\rho(x_i|x_{-i})$ denote the full conditional density for the i th element, x_i , of x , given each of the other $p - 1$ components, where $x_{-i} = \{x_j : j \neq i\}$ for $i = 1, \dots, p$. The Gibbs sampler proceeds as follows.

Pick an arbitrary starting point, x^1 and generate the next state x^2 by successively sampling from the full conditional distribution $\rho(x_i|x_{-i})$ by sampling x_i^2 from $\rho(x_i|\tilde{x}_{-i})$, $i = 1, \dots, p$ where $\tilde{x}_{-i} = \{\tilde{x}_j (j \neq i) \mid \tilde{x}_j = x_j^2 (j < i), \tilde{x}_j = x_j^1 (j > i)\}$. We repeat the process to produce a sequence x^1, \dots, x^t, \dots of states from a Markov chain with transition probability

$$K(x, y) = \prod_{l=1}^k \rho(y_l \mid y_j (j < l), x_j (j > l))[3]. \quad (3.9)$$

This is the product of the conditional densities of the individual steps required to produce an iteration of the Gibbs sampler.

4 The Rigid Body Chain Model

Polymer chains can be modeled with links whose position and direction are determined by a coordinate reference frame. Here we consider a linear polymer chain in which each chain unit is modeled as a rigid link of length $h > 0$.

A chain of $n + 1$ units corresponds to a list of links indexed by $a = 0, \dots, n$. Each link is defined by a vector r^a and an orthonormal frame, $\{\mathbf{d}_i^a\}$. The vector r^a describes the position of the link reference point, while the frame $\{\mathbf{d}_i^a\}$ is fixed in the link and describes its orientation relative to a frame $\{e_i\}$ fixed in the lab. In terms of DNA, the d_1 and d_2 axes lie in the plane of the base pairs and d_2 lies along the long axis of the dinucleotide base pair. The d_3 axis is normal to the aforementioned plane and parallel to the helix axis.

We assume the links of a chain are always joined end-to-end, and without loss of generality, we suppose $r^{(0)} = 0$ and $d_i^{(0)} = e_i$. The configuration of a chain is completely determined by the n orthonormal frames $\{\mathbf{d}_i^a\}$. In particular, a configuration is completely defined by n rotation matrices $A^a \in SO(3)$ which will be discussed in detail.

5 Configuration Variables

The configuration of a rigid body chain is said to be known when the configuration of each body B_a is known with respect to a fixed origin O_{fixed} and fixed frame $\{\mathbf{e}_i\}$. Configurations can be defined using either absolute or junction variables.

5.1 Absolute variables (Q^a, r^a)

These variables specify the configuration of each body in the chain relative to the fixed origin O_{fixed} and fixed frame $\{\mathbf{e}_i\}$, namely

$$\left. \begin{aligned} \mathbf{d}_j^a &= \sum_i Q_{ij}^a \mathbf{e}_i, & Q^a &= \varphi(\theta^a) \in SO(3), & \theta^a &\in \mathbf{R}^3 \\ \mathbf{r}^a &= \sum_i r_i^a \mathbf{e}_i, & \mathbf{r}^a &\in \mathbf{R}^3, \end{aligned} \right\} (a = 0, \dots, n) \quad (5.10)$$

where \mathbf{r}^a is the vector from O_{fixed} to O_a and φ is a coordinate map for $SO(3)$.

The assumption that the links are always joined end-to-end implies

$$\mathbf{r}^a - \mathbf{r}^{a-1} = h \mathbf{d}_i^{a-1}. \quad (5.11)$$

5.2 Junction variables (Λ^a, v^a)

For purposes of defining internal potential energies for chains, it is convenient to describe a chain configuration in terms of junction variables. These variables specify the configuration of each body in the chain relative to its neighbor, namely

$$\left. \begin{aligned} \mathbf{d}_j^a &= \sum_i \Lambda_{ij}^a \mathbf{d}_i^{a-1}, & \Lambda^a &= \varphi(u^a) \in SO(3), & u^a &\in \mathbf{R}^3 \\ \mathbf{r}^a &= \mathbf{r}^{a-1} + \sum_i h \mathbf{d}_i^{a-1} \end{aligned} \right\} (a = 1, \dots, n) \quad (5.12)$$

with

$$\left. \begin{aligned} \mathbf{d}_j^0 &= \sum_i A_{ij}^0 \mathbf{e}_i, & A^0 &= \varphi(u^0) \in SO(3), & u^0 &\in \mathbf{R}^3 \\ \mathbf{r}^0 &= \sum_i h^0 \mathbf{e}_i. \end{aligned} \right\} \quad (5.13)$$

Notice that A^a ($a = 1, \dots, n$) is the rotation matrix for $\{\mathbf{d}_i^a\}$ relative to $\{\mathbf{d}_i^{a-1}\}$, not $\{\mathbf{e}_i\}$. The rotation matrices A^a can be parameterized in different ways. Here we outline two different parameterizations in terms of Euler angles.

5.2.1 Tilt-Roll-Twist Angles

The rotation matrix A^a for the Tilt-Roll-Twist model is given by the Euler angle map:

$$A(u) = \begin{pmatrix} c_2 c_3 - s_1 s_2 s_3 & -c_1 s_3 & s_2 c_3 + s_1 c_2 s_3 \\ c_2 s_3 + s_1 s_2 c_3 & c_1 c_3 & s_2 s_3 - s_1 c_2 c_3 \\ -c_1 s_2 & s_1 & c_1 c_2 \end{pmatrix}, \quad (5.14)$$

$$c_i = \cos(u_i), \quad s_i = \sin(u_i),$$

$$u \in \mathcal{A} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times [-\pi, \pi] \times [-\pi, \pi] \subset \mathbf{R}^3,$$

$$\left. \begin{aligned} u_1^a &= \text{tilt} \\ u_2^a &= \text{roll} \\ u_3^a &= \text{twist} \end{aligned} \right\} \text{relative rotations of } \{\mathbf{d}_i^a\} \text{ wrt } \{\mathbf{d}_i^{a-1}\}.$$

Tilt and Roll provide two independent measures of bending while Twist provides an overall measure of twisting between $\{\mathbf{d}_i^a\}$ and $\{\mathbf{d}_i^{a-1}\}$.

5.2.2 Bend-Wrench-Turn Angles

The rotation matrix A^a for the Bend-Wrench-Turn model is given by:

$$A(\mathbf{u}) = \begin{pmatrix} s_2 s_3 - c_1 c_2 c_3 & c_2 s_3 + s_2 c_1 c_3 & s_1 c_3 \\ -s_2 c_3 - s_3 c_1 c_2 & s_2 s_3 c_1 - c_2 c_3 & s_1 s_3 \\ c_2 s_1 & -s_1 s_2 & c_1 \end{pmatrix}, \quad (5.15)$$

$$c_i = \cos(u_i), \quad s_i = \sin(u_i),$$

$$\mathbf{u} \in \mathcal{A} = [0, \pi] \times [-\pi, \pi] \times [-\pi, \pi] \subset \mathbf{R}^3,$$

$$\left. \begin{array}{l} u_1^a = \text{bend} \\ u_2^a = \text{wrench} \\ u_3^a = \text{turn} \end{array} \right\} \text{relative rotations of } \{\mathbf{d}_i^a\} \text{ wrt } \{\mathbf{d}_i^{a-1}\}.$$

Bend provides an overall measure of bending while Wrench and Turn provide two independent measures of twisting between $\{\mathbf{d}_i^a\}$ and $\{\mathbf{d}_i^{a-1}\}$.

Tilt-Roll-Twist angles are commonly used in DNA modeling whereas Bend-Wrench-Turn angles are not. However, because our goal is to study overall bending effects in polymer chains while ignoring twisting effects, we find BWT angles to be most appropriate.

6 Energy of Chains

6.1 Potential energy

For our BWT chain we consider an internal potential energy of the form

$$E(u) = E_1(u^1) + \cdots + E_n(u^n) \quad (6.16)$$

where $u^a \in \mathcal{A} = [0, \pi] \times [-\pi, \pi] \times [-\pi, \pi] \subset \mathbf{R}^3$ and E_a is a general quadratic energy associated with the junction between bodies a and $a - 1$. In particular, we have

$$E_a(u^a) = \frac{1}{2}[u^a - \hat{u}^a] \cdot \mathbf{K}^a[u^a - \hat{u}^a] \geq 0 \quad (6.17)$$

where

$$\begin{aligned} \mathbf{K}^a &\in \mathbf{R}^{3 \times 3} && \text{stiffness matrix} \\ \hat{u}^a &\in \mathbf{R}^3 && \text{unstressed or relaxed shape parameters.} \end{aligned}$$

Our goal is to study overall bending effects in polymer chains while ignoring twisting effects. Thus we assume \mathbf{K}^a is of the form

$$\mathbf{K}^a = \begin{pmatrix} K_a & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, K_a > 0 \quad (6.18)$$

In this case the junction energies take the form

$$E_a(u^a) = \frac{1}{2}K_a(\phi_a - \hat{\phi}_a)^2 \quad (6.19)$$

where $\phi_a = u_1^a$ is the bending angle, K_a is the bending stiffness and $\hat{\phi}_a$ is the relaxed shape parameter associated with the junction between links a and $a - 1$.

6.2 Kinetic energy, momentum

By definition, the total kinetic energy of the chain is

$$\Phi \stackrel{\text{def}}{=} \sum_{a=1}^n \frac{1}{2} m^a |\nu^a|^2 + \frac{1}{2} \omega^a \cdot \Gamma^a \omega^a \quad (6.20)$$

where for each link

- m^a = total mass,
- ν^a = velocity of mass center,
- Γ^a = rotational inertia matrix wrt/mass center,
- ω^a = angular velocity.

Using (5.12), (5.14) and the relations

$$\dot{A}^a = \Lambda^a[\omega^a \times], \quad \omega^a = S(u^a) \dot{u}^a \quad (6.21)$$

we find that the total kinetic energy can be written in the form

$$\Phi = \frac{1}{2} \dot{u} \cdot M(u) \dot{u} \quad (6.22)$$

where $\dot{u} = (\dot{u}^1, \dots, \dot{u}^n) \in \mathbf{R}^{3n}$. $S(u^a)$ is a structure matrix associated with the Euler angle map (5.14) and $M(u) \in \mathbf{R}^{3n \times 3n}$ is the chain mass matrix (symmetric, positive-definite).

When we introduce the canonical momentum variable

$$p \stackrel{\text{def}}{=} M(u) \dot{u} \quad (6.23)$$

the kinetic energy takes the form

$$\Phi = \frac{1}{2} p \cdot [M(u)]^{-1} p. \quad (6.24)$$

7 Chain Statistics

7.1 Hamiltonian, distribution functions

An equilibrium statistical model for thermal fluctuations can now be outlined.

Consider a polymer chain with configuration space \mathcal{A}^n and potential energy $E : \mathcal{A}^n \rightarrow \mathbb{R}$.

In particular, let

- $(u, p) \in Q \times \mathbb{R}^m$, $Q = \mathcal{A}^n \subset \mathbb{R}^m$ ($m = 3n$),
- $H(u, p) = E(u) + \frac{1}{2}p \cdot [M(u)]^{-1}p$.

Suppose the chain is placed in a heat bath at temperature $T > 0$. Then the canonical (Gibbs) distribution function is defined by

$$\rho_G(u, p) = \frac{1}{Z_G} e^{-H(u, p)/kT}, \quad Z_G = \int_{Q \times \mathbb{R}^m} e^{-H(u, p)/kT} du dp \quad (7.25)$$

where

- $T =$ fluid temperature (absolute),
- $k =$ Boltzmann constant.

Assuming $\det M(u)$ is constant and/or slowly varying compared to $E(u)$, the associated configurational distribution may be approximated by

$$\rho(u) = \frac{1}{Z} e^{-E(u)/kT}, \quad Z = \int_Q e^{-E(u)/kT} du. \quad (7.26)$$

8 The Cyclization Problem

A classic problem in linear polymer chains is studying the probability that an open chain becomes a closed chain. Assuming the two ends A, B of a chain will form a bond when brought sufficiently close together, the probability of cyclization can be modeled by the function

$$P_{\Omega_\epsilon} = \int_{\Omega_\epsilon} \rho(u) du \quad (8.27)$$

where $\Omega_\epsilon = \{u \in \mathcal{A}^n : |A - B| \leq \epsilon\}$ and $\epsilon > 0$ is a closure parameter that defines the maximum distance between A and B at which a bond will form. Our goal is to study how P_{Ω_ϵ} depends on the fixed shape parameters $\hat{\phi}_a$ of the chain. We wish to characterize the changes in $\hat{\phi}$ that produce the most significant changes in P_{Ω_ϵ} . In particular, given two chains I and II with parameter sets $\hat{\phi}_I, \hat{\phi}_{II}$ we seek to develop methods for determining which cyclization probability, $P_{\Omega_\epsilon}(\hat{\phi}_I)$ or $P_{\Omega_\epsilon}(\hat{\phi}_{II})$, is larger.

8.1 Probability Gradient

Consider a polymer chain with potential energy $E(u, \hat{\phi}) \in \mathbb{R}$ where $u \in \mathcal{A}^n$ are the configuration variables and $\hat{\phi} \in \mathbb{R}^n$ are the given relaxed shape parameters. For a given closure parameter $\epsilon > 0$ the cyclization probability is

$$P_{\Omega_\epsilon} = \int_{\Omega_\epsilon} \rho(u) du = \frac{\int_{\Omega_\epsilon} e^{-E(u, \hat{\phi})/kT} du}{\int_{\mathcal{A}^n} e^{-E(u, \hat{\phi})/kT} du} \quad (8.28)$$

where $k, T > 0$ are constants. The following result provides a convenient characterization of the gradient of P_{Ω_ϵ} .

8.1.1 Proposition 1

For any $\hat{\phi} \in \mathbb{R}^n$ we have

$$\frac{\partial P_{\Omega_\epsilon}}{\partial \hat{\phi}}(\hat{\phi}) = P_{\Omega_\epsilon}(\hat{\phi}) \left[\left\langle \frac{\partial E}{\partial \hat{\phi}} \right\rangle_{\mathcal{A}^n} - \left\langle \frac{\partial E}{\partial \hat{\phi}} \right\rangle_{\Omega_\epsilon} \right] (\hat{\phi}) \in \mathbb{R}^n \quad (8.29)$$

where

$$\langle f \rangle_D := \frac{\int_D f(u, \hat{\phi}) e^{-E(u, \hat{\phi})/kT} du}{\int_D e^{-E(u, \hat{\phi})/kT} du} \quad (8.30)$$

for any function $f(u, \hat{\phi})$ and subset $D \in A^n$.

8.1.2 Proposition 2

Using the probability gradient we can provide a local characterization of the cyclization properties of a polymer chain.

Consider two chains defined by the shape parameters $\hat{\phi}_I, \hat{\phi}_{II} \in \mathbf{R}^n$ and suppose $\frac{\partial P_{\Omega_\epsilon}}{\partial \hat{\phi}}(\hat{\phi}_I) \neq 0$. Then

- $P_{\Omega_\epsilon}(\hat{\phi}_{II}) > P_{\Omega_\epsilon}(\hat{\phi}_I)$ for all $\hat{\phi}_{II}$ such that $\frac{\partial P_{\Omega_\epsilon}}{\partial \hat{\phi}} \cdot [\hat{\phi}_{II} - \hat{\phi}_I] > 0$ with $|\hat{\phi}_{II} - \hat{\phi}_I|$ sufficiently small.
- The largest (positive) change in $P_{\Omega_\epsilon}(\hat{\phi}_I)$ occurs when $[\hat{\phi}_{II} - \hat{\phi}_I] \in \mathbf{R}^n$ is parallel to $\frac{\partial P_{\Omega_\epsilon}}{\partial \hat{\phi}}(\hat{\phi}_I) \in \mathbf{R}^n$.

So the gradient $\frac{\partial P_{\Omega_\epsilon}}{\partial \hat{\phi}}(\hat{\phi}_I)$ characterizes those directions from $\hat{\phi}_I$ along which P_{Ω_ϵ} increases (or decreases) the fastest. We need only know the direction of $\frac{\partial P_{\Omega_\epsilon}}{\partial \hat{\phi}}(\hat{\phi}_I)$ in order to apply the above results. This direction is parallel to

$$V(\hat{\phi}) = [\langle \frac{\partial E}{\partial \hat{\phi}} \rangle_{A^n} - \langle \frac{\partial E}{\partial \hat{\phi}} \rangle_{\Omega_\epsilon}](\hat{\phi}). \quad (8.31)$$

9 MCMC Methods

We wish to use the Monte Carlo method to approximate the gradient direction vector $V(\hat{\phi})$ for small closure parameters $\epsilon > 0$. Our primary task is estimating the average

$$\langle f \rangle_D := \frac{\int_D f(u, \hat{\phi}) e^{-E(u, \hat{\phi})/kT} du}{\int_D e^{-E(u, \hat{\phi})/kT} du} \quad (9.32)$$

where $f(u, \hat{\phi})$ is a given function and D is a given subset of the chain configuration space A^n . In particular, we are interested in $f(u, \hat{\phi}) = \frac{\partial E}{\partial \phi}$ and the subsets $D = A^n$ and $D = \Omega_\epsilon$. Because our $E(u, \hat{\phi})$ and Ω_ϵ depend only on the bend angles ϕ_a , we need only consider the d_3 vectors. In particular, we need only track the centerline of the polymer chain, and not the twist of the chain about the centerline. For our numerical simulations we consider the limit case $\epsilon = 0$. In this case, $D = \Omega_o$ corresponds to the set of all closed chains whereas $D = A^n$ corresponds to the set of all open chains.

9.1 Sampling Methods

There are two basic methods available for generating a sample set S_N from a given distribution $\rho(u)$, $u \in Q \subset \mathbf{R}^m$:

- Direct sampling, transformation methods (“Direct MC methods”),
- Rejection methods (“Metropolis MC methods”).

Direct sampling is useful for simulating open chains, however rejection methods are necessary for closed chains.

9.1.1 Direct sampling

Random number generators have been developed for sampling from various standard distributions. The two main examples are:

- Standard uniform distribution on unit cube in \mathbf{R}^m :

$$\rho(u) = 1, \quad u \in Q = [0, 1]^m \subset \mathbf{R}^m. \quad (9.33)$$

- Standard Gaussian distribution on \mathbf{R}^m :

$$\rho(u) = \frac{1}{(2\pi)^{m/2}} e^{-|u|^2/2}, \quad u \in Q = \mathbf{R}^m. \quad (9.34)$$

9.1.2 Transformation methods

Transformation methods provide a means to construct new sample sets from old ones. In particular, consider a

- sample set $S_N = \{u_1, \dots, u_N\}$ from $\rho(u)$, $u \in Q \subset \mathbf{R}^m$,
- change of variable $\tilde{u} = T(u) = Au + c$, $\det[A] \neq 0$.

Then

$$\tilde{S}_N = \{\tilde{u}_1, \dots, \tilde{u}_N\}, \quad \tilde{u}_i = T(u_i)$$

is a sample set from

$$\tilde{\rho}(\tilde{u}) = \rho(T^{-1}(\tilde{u})) \left| \det \frac{\partial T^{-1}}{\partial \tilde{u}} \right|, \quad \tilde{u} \in \tilde{Q} = T(Q)$$

where

$$T^{-1}(\tilde{u}) = A^{-1}(\tilde{u} - c).$$

9.1.3 Sampling Open Polymer Chains

We can use the transformation methods developed above to create a sample of open chains from the distribution $\tilde{\rho}(\tilde{u}) = \frac{1}{Z} e^{-K(\phi - \hat{\phi})^2/2}$.

Consider

- $S_N = \{u_1, \dots, u_N\}$, $\rho(u) = \frac{1}{(2\pi)^{1/2}} e^{-u^2/2}$, $u \in Q = \mathbf{R}$
- $\tilde{u} = au + c$, $a \neq 0$, $u = a^{-1}(\tilde{u} - c)$.

Then

$$\tilde{S}_N = \{\tilde{u}_1, \dots, \tilde{u}_N\}$$

is a sample set from

$$\tilde{\rho}(\tilde{u}) = \frac{1}{(2\pi a^2)^{1/2}} e^{-(\tilde{u}-c)^2/2a^2}, \quad \tilde{u} \in \tilde{Q} = \mathbf{R}.$$

Thus a sample set from a non-standard Gaussian distribution can be constructed from the standard one through an appropriate change of variable. For instance, if we let

$$u = \sqrt{K}(\phi - \hat{\phi})$$

we can sample the open chain directly from the distribution

$$\tilde{\rho}(\tilde{u}) = \frac{1}{Z} e^{-K(\phi - \hat{\phi})^2/2}.$$

9.2 Simulating Closed Chains

We wish to simulate a statistically correct set of closed chains comprised of n free-joint segments of unit length. We utilize the "random hedgehog" algorithm originally proposed by Klenin et. al [6].

For the initial chain configuration, randomly choose n (even number) vectors on the unit sphere. This is done by computing the points

$$\begin{aligned} x &= \sqrt{1-u^2} \cos\theta \\ y &= \sqrt{1-u^2} \sin\theta \\ z &= u \end{aligned}$$

with $\theta \in [0, 2\pi)$ and $u \in [-1, 1]$ chosen so that $e_i = (x, y, z)$ are uniformly distributed. For each (x, y, z) on the sphere, let $e_{i-1} = -e_i$ so that $\sum_{i=0}^n e_i = 0$. When we assemble these vectors in an arbitrary order we obtain a closed chain whose edges have length 1. If we desire edges of a different length, h , we can scale the vectors obtained by the procedure above by h before assembling them.

We perturb the state of the polymer chain by performing the following moves. Pick two vectors in the set randomly and rotate them around their bisectrix by θ degrees. We use Rodrigues' Rotation formula [7] to compute the new vectors: the rotation matrix $Q \in SO(3)$ corresponding to a rotation by an angle $\theta \in \mathbf{R}$ about a fixed axis specified by the unit vector $w = (w_1, w_2, w_3) \in \mathbf{R}^3$ is given by

$$Q = I + \hat{w} \sin \theta + \hat{w}^2 (1 - \cos \theta) \quad (9.35)$$

where \hat{w} denotes the skew symmetric matrix

$$\hat{w} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix}. \quad (9.36)$$

Let the resulting vectors be the new configuration for the originally chosen vectors. Then let an arbitrary arrangement of the new set of vectors determine the new chain.

9.3 The Metropolis-Hastings Step

As we have seen, the MH algorithm uses random moves to create a set of chain configurations. We will use this procedure to arrive at a set of polymer chain configurations. For our purposes, we assume $K(x_{t+1}, x_t) = K(x_t, x_{t+1})$ so that

$$\alpha(x_t, x_{t+1}) = e^{-(E(x_t) - E(x_{t+1}))}. \quad (9.37)$$

- Let State-0 be the current state of the chain as computed above having Energy-1. Use the random hedgehog method to randomly change the chain to State-1 having Energy-2. Save State-1 and State-0.
- Accept or reject State-1 according to the following conditions:

- If $E2 < E1$, accept State-1.
- If not, take a random number R between 0.0 and 1.0.
 - * If $e^{-(E1-E2)} > R$, accept State-1.
 - * If not, reject State-1.
- To continue, return to Step 1.

In an MCMC procedure, it is desirable to accept approximately 50 percent of the attempted trials. In order to achieve this in our scheme, we had to adjust the range from which we chose our random rotation angle in the random hedgehog algorithm. For both the open and closed chains, we use $\theta \in [-\theta_o, \theta_o]$ where θ_o is a constant chosen to achieve an acceptance ratio of approximately 50%.

10 Numerical Results

10.1 Goals of MCMC Simulations

We wish to use Markov Chain Monte Carlo to estimate the probability gradient direction vector $V(\hat{\phi})$ for two different polymer chain configurations. Knowledge of $V(\hat{\phi})$ allows us to use Proposition 2 to explore the dependence of the cyclization probability on the location of interior bends. We will use these simulations to determine what changes in $\hat{\phi}$ lead to the largest increase in P_Ω for each polymer chain. Also, we will observe if there is any difference between the results obtained for the two different chain configurations.

10.2 Chain Parameters

Our initial chain configuration consists of the chain having minimum energy rather than that proposed by Klenin [6]. However, the random hedgehog algorithm is perfectly useful for generating random moves on the minimum chain. Each $\hat{\phi}$ contains a 90 degree bend dispersed over 10 of its edges. The first $\hat{\phi}$ contains a bend between edges 35 and 45 (represented by "35-45"), the second between 55 and 65 (see Figure 1).

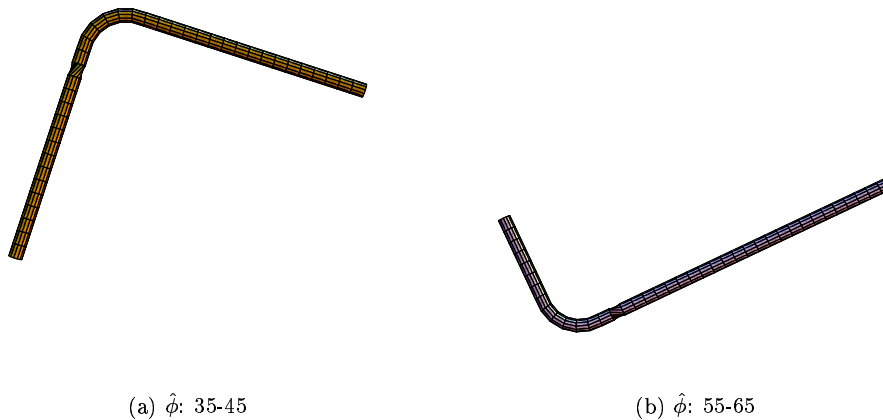


Figure 1: Chains with two different $\hat{\phi}$ parameters: (a) parameters for bend between edges 35 – 45, (b) parameters for bend between edges 55 – 65.

Tables 1 and 2 give a list of the parameters used for our simulation of these values of $\hat{\phi}$

for the open and closed chain. The remaining variables correspond to the following:

- Sample Size = number of accepted MH steps,
- K = stiffness constant,
- θ range = the degree range of vector rotation in the hedgehog simulation,
- n = number of links in the polymer chain,
- h = length of each link.

$\hat{\phi}$: 35-45		
Parameters	Open Chain	Closed Chain
Sample Size	61,479	102,206
K	80/3	80/3
θ range	$(\frac{\pi}{70}, -\frac{\pi}{70})$	$(\frac{\pi}{70}, -\frac{\pi}{70})$
n	80	80
h	1/80	1/80

Table 1.

$\hat{\phi}$: 55-65		
Parameters	Open Chain	Closed Chain
Sample Size	148,511	33,944
K	80/3	80/3
θ range	$(\frac{\pi}{70}, -\frac{\pi}{70})$	$(\frac{\pi}{70}, -\frac{\pi}{70})$
n	80	80
h	1/80	1/80

Table 2.

10.3 Calculation of $\langle \frac{\partial \langle E \rangle}{\partial \hat{\phi}} \rangle$

The results of the individual $\langle \frac{\partial E}{\partial \hat{\phi}} \rangle$ and $V(\hat{\phi}) = \langle \frac{\partial E}{\partial \hat{\phi}} \rangle_{open} - \langle \frac{\partial E}{\partial \hat{\phi}} \rangle_{closed}$ calculations are presented in Figures 2 and 3. The calculations were made over each angle components of the chains.

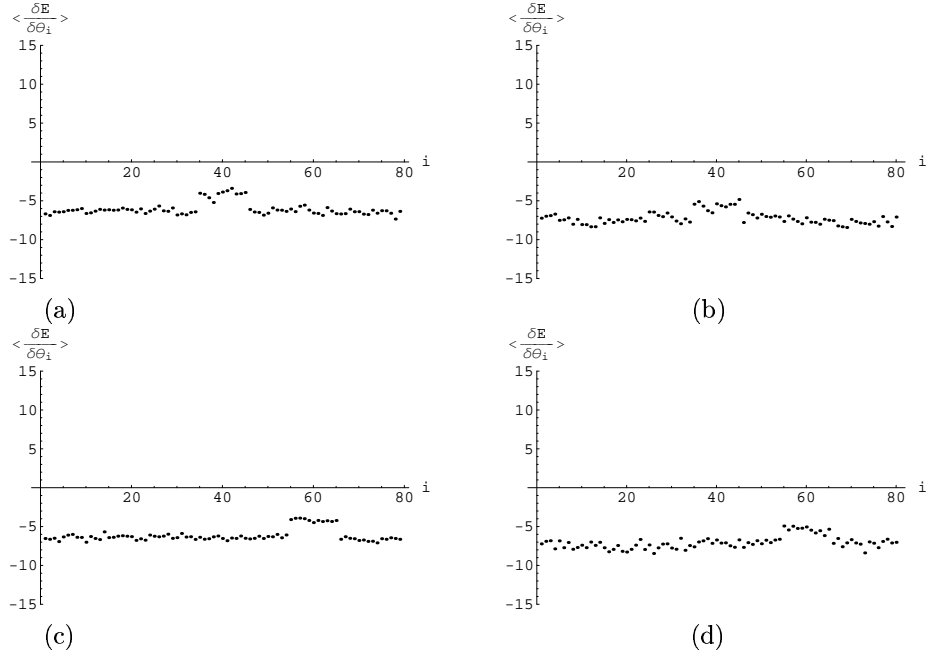
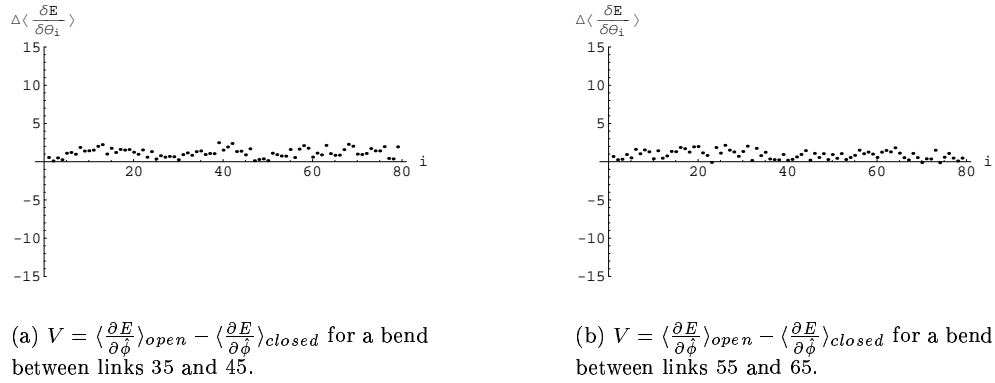


Figure 2: $\langle \frac{\partial E}{\partial \theta} \rangle$ calculations: (a) open chain with a 90 degree bend between links 35 and 45, (b) closed chain with a 90 degree bend between links 35 and 45, (c) open chain with a 90 degree bend between links 55 and 65, (d) closed chain with a 90 degree bend between links 55 and 65.



(a) $V = \langle \frac{\partial E}{\partial \phi} \rangle_{open} - \langle \frac{\partial E}{\partial \phi} \rangle_{closed}$ for a bend between links 35 and 45.

(b) $V = \langle \frac{\partial E}{\partial \phi} \rangle_{open} - \langle \frac{\partial E}{\partial \phi} \rangle_{closed}$ for a bend between links 55 and 65.

Figure 3: $V = \langle \frac{\partial E}{\partial \phi} \rangle_{open} - \langle \frac{\partial E}{\partial \phi} \rangle_{closed}$.

10.4 Monitoring MCMC Convergence

10.4.1 CUSUM Convergence

Several methods are available to assess the convergence of MCMC using the Metropolis-Hastings sampler. One such method, proposed by Yu and Mykland [9], is based on cumulative sums (CUSUM). Suppose we've acquired the Markov chain $\{X_1, \dots, X_T\}$. For arbitrary $h(X^{(t)})$, we graph the partial differences

$$D_T^i = \sum_{t=1}^i [h(X^{(t)}) - S_T] \quad (10.38)$$

where

$$S_T = \frac{1}{T} \sum_{t=1}^T h(X^{(t)}) \quad (10.39)$$

for $i = 1, \dots, T$.

The result is an evaluation of the mixing speed of the chain and the correlation between the $X^{(t)}$'s. When the mixing is high (that is, the chain has a fast pace of exploration of the stationary distribution), the graph of D_T^i is highly irregular and concentrated around 0. If the mixing is slow, the graph will display long excursions away from 0.

The results of our CUSUM convergence calculations are shown in Figure 4. For our evaluations, we let $h(X^{(t)}) = E(X^{(t)})$. It is difficult to tell from these plots whether or not our MCMC algorithm has converged. One can improve the CUSUM method, in terms of its subjectivity, but adding a "benchmark" CUSUM plot based upon a sequence of *iid* normal variates with mean and variance matched to estimate moments of the h sequence. This plot should approximate, to the second order, the "ideal" CUSUM path for an *iid* sequence from the target distribution. A favorable comparison of the two plots would indicate that the chain is mixing well [2]. Unfortunately it is difficult to arrive at a "benchmark" plot for our purposes. Instead we rely on a "sub-averaging" method to test convergence.

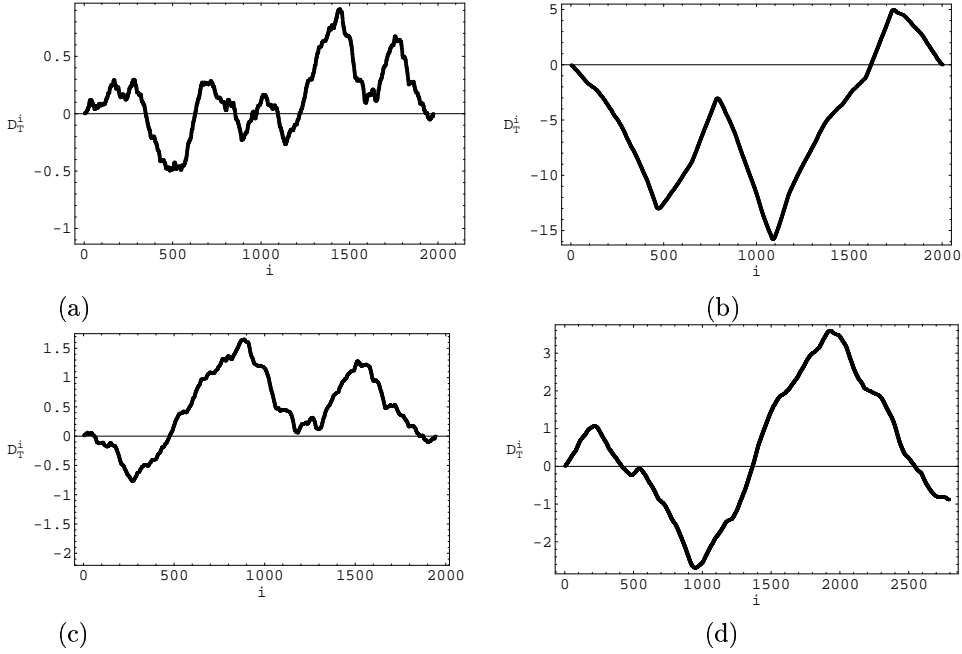


Figure 4: CUSUM convergence calculations: (a) open chain with a 90 degree bend between links 35 and 45, (b) closed chain with a 90 degree bend between links 35 and 45, (c) open chain with a 90 degree bend between links 55 and 65, (d) closed chain with a 90 degree bend between links 55 and 65.

10.4.2 Sub-Averaging

We can also test the convergence of the MCMC algorithm by comparing the average of successive lists of results compared to the average of every other result in that list. That is, let $\langle V_T \rangle$ be the average value of the gradient direction vector V over T MCMC iterations, and let $\langle V_{T/2} \rangle$ be the average over every other iteration. We then compute

$$G_T^i = \frac{\|\langle V_i \rangle - \langle V_{i/2} \rangle\|}{\|\langle V_i \rangle\|} \quad (10.40)$$

for $i = 1, \dots, T$. The idea is that this test will converge to zero when the MCMC run converges in the sense that averages become independent of sample size.

The results of this test for each chain configuration are shown in Figure 5.

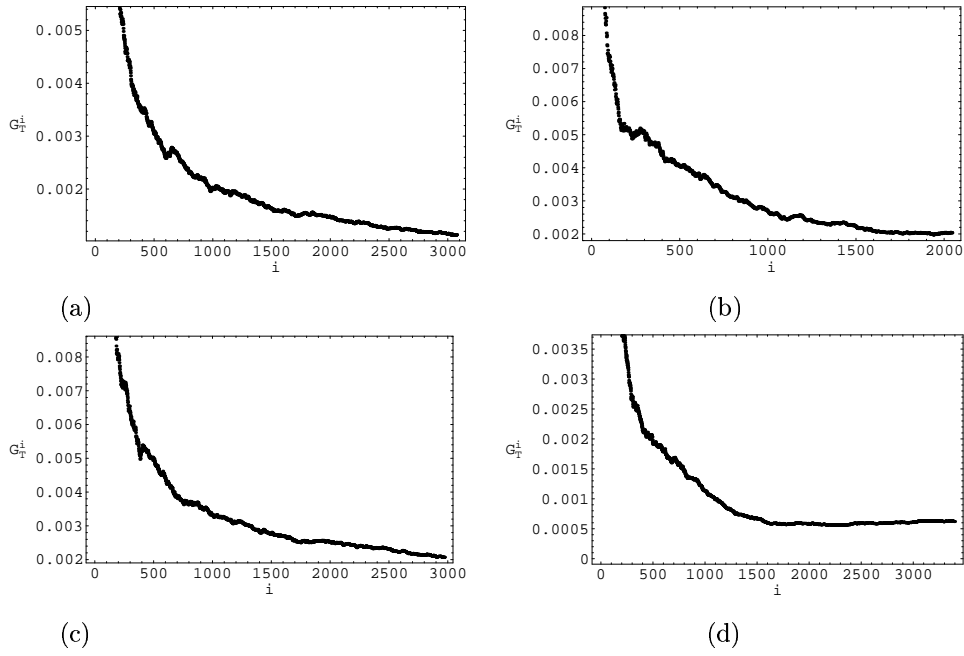


Figure 5: Sub-averaging convergence calculations: (a) open chain with a 90 degree bend between links 35 and 45, (b) closed chain with a 90 degree bend between links 35 and 45, (c) open chain with a 90 degree bend between links 55 and 65, (d) closed chain with a 90 degree bend between links 55 and 65.

11 Conclusion

Simple arguments based on energetics suggest that the cyclization rates of our example chains might be the same because the minimum energy of two open chains with different $\hat{\phi}$'s is zero and the minimum energy of their closed states is the same. On the other hand, one can argue that the rates might be different because different open chains must overcome different end-to-end distances in order to form closed chains. Our numerical results shown in Figure 3 suggest that the probability gradient direction vector V is roughly the same for both chains. However, it is ultimately unclear if equality is in fact a property of these two chains. Furthermore, equality in V does not necessarily imply equality in cyclization rates. It merely indicates that the two chains would have the same qualitative local behavior in the sense that small changes in $\hat{\phi}$ would produce the same qualitative changes in the cyclization probability P_Ω . From the results of $V(\hat{\phi}) \in \mathbf{R}^n$ in Figure 3, we observe that $V_i > 0$ for most $i = 1, \dots, n$. This suggests that increasing the values of most of the $\hat{\phi}_i$'s would increase the probability of two chain ends finding each other and forming a closed chain. This intuitively makes sense; increasing $\hat{\phi}_i$ would result in a more compact initial chain whose ends would be closer together.

It would be interesting to extend the MCMC procedure to include twist effects in the energy. This would require working with frames $\{d_1, d_2, d_3\}$ rather than unit vectors d_3 . As a result, the MC algorithm would have to generate angles which would be used to track the twist of the d_1 and d_2 vectors about the d_3 vectors. A number of different closure conditions could be considered which not only account for the proximity of the chain end-points, but also the alignment of the chain end-frames.

It is difficult to tell whether or not our MCMC method has converged. The CUSUM results (Figure 4) are inconclusive, while the sub-averaging method is not a proven convergence test. Further study will be required to determine if the procedure has converged and our results for V are conclusive.

12 Appendix

The following is a collection of pseudocode for the various subalgorithms used to complete the MCMC algorithm.

12.0.3 statechange() pseudocode

statechange(chain) uses the "random hedgehog" method to alter the state of a current n -link chain. The following applies to closed chains:

```
1: n = length of chain
2: for  $i \leftarrow 1, 2, \dots, 30$  do
3:   link1 = first randomly chosen link
4:   link2 = second randomly chosen link (for open chains, let link2 be a randomly chosen
      vector on the unit sphere)
5:   v = bisectrix of link1 and link2
6:   theta = random real  $\in [-\theta_o, \theta_o]$ 
7:   rot1 = rotate link1 around v by theta radians
8:   rot2 = rotate link2 around v by theta radians
9:   rot1 = link1
10:  rot2 = link2
11: end for
12: rand = random number between 0 and 1
13: if rand  $\geq$  0.9 then
14:   randomly permute the link vectors
15: end if
```

12.0.4 chainangles() pseudocode

chainangles(chain) computes the θ s, the angles between each link in an n -link chain.

```
1: n = length of chain
2: thetas = array that will hold the values of  $\theta_i$ 's.
```

```

3: for  $i \leftarrow 1, 2, 3, \dots, n$  do
4:   link1 =  $i$ -th link in the chain
5:   link2 =  $(i + 1)$ -th link in the chain
6:    $\text{thetas}[i] = \arccos\left(\frac{\text{link1} \cdot \text{link2}}{|\text{link1}| |\text{link2}|}\right)$ 
7: end for

```

12.0.5 Markov Chain Monte Carlo pseudocode

```

1: mcits = number of Monte Carlo iterations to be performed
2: points = initial chain configuration
3:  $K$  = stiffness constant
4:  $\hat{\phi}$  = define desired  $\hat{\phi}$ 
5:  $\text{thetas0} = \text{chainangles}(\text{points})$ 
6:  $\text{state0} = \text{points}$ ;
7:  $E0 = \frac{K}{2} \sum_{a=1}^n (\text{thetas0}[a] - \hat{\phi})^2$ 
8: for  $i \leftarrow 2, 3, 4 \dots mcits$  do
9:    $\text{state1} = \text{statechange}(\text{state0})$ 
10:   $\text{thetas1} = \text{chainangles}(\text{state1})$ 
11:   $E1 = \frac{K}{2} \sum_{a=1}^n (\text{thetas1}[a] - \hat{\phi})^2$ 
12:  if  $E1 \leq E0$  then
13:     $E0 = E1$ 
14:     $\text{state0} = \text{state1}$ ;
15:     $\text{thetas0} = \text{thetas1}$ ;
16:  else
17:     $\text{rand} = \text{random number between } 0 \text{ and } 1$ 
18:    if  $e^{E1-E0} > \text{rand}$  then
19:       $E0 = E1$ 
20:       $\text{state0} = \text{state1}$ ;
21:       $\text{thetas0} = \text{thetas1}$ ;
22:    end if

```

23: **end if**

24: **end for**

References

- [1] Casella, G. & Robert, C. (1999) *Monte Carlo Statistical Methods*, Springer-Verlag New York, Inc.
- [2] Brooks, S P. & Roberts, G O (1998) Assessing Convergence of Markov Chain Monte Carlo Algorithms. *Statistics and Computing*. **8**, p319-335.
- [3] Geman, S. & Geman, D. (1984) Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images. *IEEE Transactions on pattern analysis and machine intelligence* **57**, 721-741.
- [4] George, E. (1999) Markov Chain Monte Carlo and the Revolution in Statistical Modeling.
- [5] Gilks, W. R., Richardson S. & Spiegelhalter, D.J. (1995) Markov Chain Monte Carlo in Practice. *Chapman & Hall*.
- [6] Klenin K.V., Vologodskii A.V., Anshelevich V.V., Dykhne A.M., Frank-Kamenetskii M.D. (1988) Effect of excluded volume on topological properties of circular DNA. *J. Biomol. Struct. Dyn.* **5**, 1173-1185.
- [7] Murray, R.M.; Li, Z.; & Sastry, S.S. *A Mathematica Introduction to Robotic Manipulation*, Boca Raton, FL: CRC Press, 1994.
- [8] Peskun, P.H. (1978) Optimum Monte Carlo Sampling Using Markov Chains. *Biometrika* **60**, 607-612.
- [9] Yu, B. and Mykland, P. (1998) Looking at Markov samplers through cusum path plots: A simple diagnostic idea. *Statistics and Computing* **80(3)**, 275-286.

Vita

Monica Lynn Shaw was born in Oak Lawn, Illinois on July 18, 1979, the daughter of Dorothy Bury Shaw and Stephen Joseph Shaw. After completing high school at Mount Assisi Academy, Lemont, Illinois, in 1997, she studied mathematics at the University of Illinois in Champaign-Urbana, Illinois. During this time, she was involved in a research project in the Astronomy Department where she studied the Large Magallenic Cloud. From May, 1999 through August 2001, she worked at Wolfram Research, Inc. as a Tester in Software Quality Assurance. In the Summer of 2000, she studied mathematical physics at Tulane University in New Orleans, Louisiana. In the Fall of 2000, she attended Pennsylvania State University in University Park, Pennsylvania. She received the degree of Bachelor of Science in Mathematics from the University of Illinois in May, 2001. In September, 2001, she entered the Computational and Applied Mathematics Graduate Program at The University of Texas in Austin, Texas.

Permanent Address:

304 Moore Boulevard
Apartment B
Austin, Texas 78705

This report was typed by the author.