MONTE CARLO SIMULATIONS FOR THE STATIC PROPERTIES OF POLYMERS UNDER EXTERNAL FIELDS AND AGGREGATION MEDIATED BY POLYMERS

by

YONG CHEN

A dissertation submitted to the Graduate Faculty in Chemistry in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

2004

This manuscript has been read and accepted for the Graduate Faculty in Chemistry in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

4/26/04
Date

Chair of Examining Committee

4-29-04
Date

Executive Officer

Marilyn Gunner

James Batteas

Chwen-Yang Shew

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

MONTE CARLO SIMULATIONS FOR THE STATIC PROPERTIES OF POLYMERS UNDER EXTERNAL FIELDS AND AGGREGATION MEDIATED BY POLYMERS

by

Yong Chen

Advisor: Professor Chwen-Yang Shew

Theoretical studies were conducted to elucidate the interplay between polymeric materials and external fields. External fields had a broad sense in this dissertation, including applied electric fields, confining boundaries, and medium induced potentials. Several polymer models have been investigated to understand field induced conformational deformation for flexible chains, orientation ordering for rigid rods, and aggregate formation in the presence of polymers. The essence of this research was to explore the effects of the local architecture of different polymer models on the static polymer properties. In each project, the local structure of a chosen model was modified to different extent, and the properties predicted by each level of model were investigated.

It was first shown that modification of a small portion of polymer molecules slightly deformed chain conformation, but would not affect the fundamental physics behind their conformational behavior. Such a result has been observed when the external field was weak or absent. When the strength of applied potentials (electric fields and confining potentials) was increased, the local polymer structure became more important because the effect of intrinsic monomer characteristics associated with the monomer chemical architecture was amplified. As two models were compared, the dividing line to separate the regimes of weak and strong fields could be defined at the field strengths where the local polymer structure starts to influence their polymer properties. Also, the results illustrated the general trend regarding how the local structure of polymers should be altered to maximize their response to external fields. These findings will facilitate the future development of applications and the design of more useful materials. Furthermore, aggregation of smaller particles in templated polymers was investigated. The chain connectivity prevented complex formation between the dispersed particles and polymers when their interactions were effectively attractive. As a result, the probability of forming larger aggregates increased. At the practical standpoint, these studies provided the insight into mechanisms for polymer mediated nanoparticle formation, which has been widely employed to manufacture nanoparticles of different geometry and size.

Acknowledgements

I would like to extend my gratitude to my mentor Professor Chwen-Yang Shew for his enthusiasm, patience and encouragement throughout my dissertation research. I would also like to thank my dissertation committee - Professor Marilyn Gunner at City College and Professor James Batteas at the College of Staten Island, for their sound advice and warm support through all stages of my dissertation. I would like to thank Professor Nan-Loh Yang for his continued support for my academic progress and career development.

I would also like to thank Professor Themis Lazaridis at City College, for his help in my acquaintance with molecular modeling of biomolecules and his unrelenting efforts to conquer problems.

Thanks also go to all of the faculty and staff of the chemistry department who have contributed to an extraordinarily friendly and helpful working environment. In particular, I want to thank Mr. Tai Park for his help on my computer skills, Ms. Diane Paladino, Susan Chew and Angela La Duca for their efforts in running the chores for the department, lab technicians Jo Ann Calascibeta, Soa Dang, James Saccardo Abraham Malz for their firm support and professionalism. I also want to thank my fellow students Chong Cheng, Shiying Tian, Weimin Wang, Chang Xu and others for their help in these years.

And last but not least, I want to thank my parents and my dearest wife Yang Li for their steadfast support and unconditional love.

Table of Contents

Chapter 3

Theoretical Studies of Conformational Behavior of Chain Molecules Containing Polar Groups and Electric Fields Effects

List of Tables

Chapter 2

Chapter 3

List of Illustrations

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

- Chapter 1 -

Monte Carlo Simulation in Studies of Polymers

Introduction

This dissertation project investigated the interplay between polymeric materials and external fields using Monte Carlo simulations, aiming to understand the role of the local structure of polymer models on their static properties. In addition to physical fields, such as electric fields, the external fields studied in this dissertation encompassed a variety of environmental constraints in materials, including grafting surfaces, cavities with flexible boundaries and medium induced solvation cages in polymer/particle blends. Various static properties were calculated to address the effects of applied fields, including conformational deformation, orientation ordering, and aggregate size distribution. The polymer models were chosen either at coarse-grained or atomistic levels. These studies provided insights into how the level of simplifications involved in different models affected the investigated properties.

The dissertation is organized as follows. The first chapter reviewed Monte Carlo simulations and polymer models. Chapter two to five will presented Monte Carlo simulations of polymers in different types of external fields, as mentioned above. The computer source code for some important algorithms

was reported in the Appendix.

1.1 Overview of Computer Simulations

Computer simulation is by now an established tool in many branches of science, such as nuclear physics, meteorology, economics, etc.[1] In physics, computer simulations have become an effective tool to enhance our understanding for many-body problems, with numerous applications in materials science, such as multiple component systems and polymeric materials. To better understand the role of computer simulations in modern theoretical development, the interplay of simulations with experiment, modeling and theory is presented in **Fig. 1.1**.

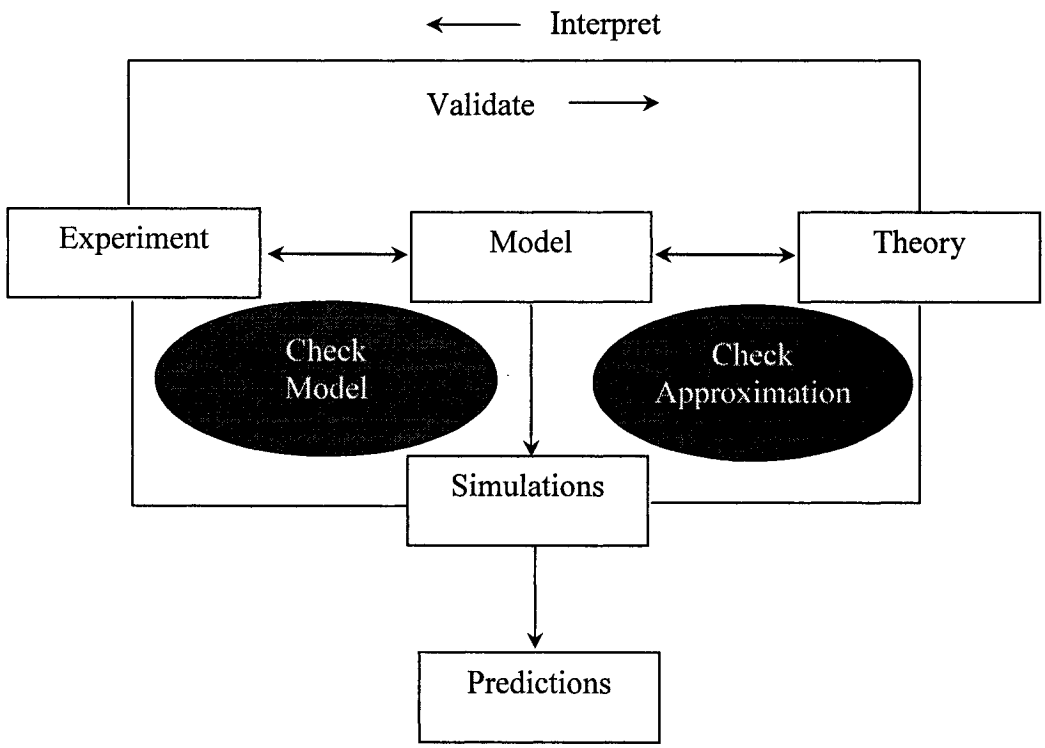


Fig. 1.1 Connections among experiment, model, theory and computer simulation.

To account for the experimental observations, the starting point lies in creating an appropriate model. For a given model, the corresponding physical properties can be calculated through theory and computer simulations, which are needed to compare with experiments. Both approaches serve the same goal of making predictions about physical properties from a model, which may lead to useful applications. In a theory, approximations are introduced to simplify the mathematics required to calculate the corresponding properties of a model. Very often, when the theoretical predictions disagree with experiments, one cannot discern whether the disagreements arise from model or theory. Since computer simulations calculate these properties exactly through robust procedures without any state-of-the-art approximations involved, they are useful to test the accuracy of a theory. In addition, computer simulations become even more attractive when the model consists of detailed molecular architectures.

Recently, a great deal of interest has been devoted to devising sophisticated molecular models, for instance force fields from the first principle calculations, to approach the level of atomistic accuracy. Solving the thermodynamic properties of these atomistic models analytically is almost impossible by human hands. However, with computer simulations, the macroscopic properties of a molecular system can be predicted, and direct comparison with experimental measurements, for example the density of polymeric materials, becomes possible. Additionally, computer simulations provide some other

properties, such as microscopic chain conformation, that cannot be easily obtained from experiments. However, computer simulations cannot replace theories completely. The approximations behind the theory make the contribution of the most important physical factors stand out. Also, an accurate theory prioritizes the factors required to understand a realistic system, and manifest the underlying basic physics laws.

Polymeric materials are typical many-body systems, in which monomer-monomer interactions govern the material properties. Even a single polymer is a complex statistical mechanics system because of large number of chain conformations. Since this dissertation addresses the many-body problems related to polymer systems, the following discussion will be focused to the context regarding the applications of computer simulations in statistical mechanics of polymers. Statistical mechanics serves as a bridge between microscopic interactions and macroscopic properties. Microscopic interactions are attainable from the first-principle calculations, such as quantum mechanics. Macroscopic properties, on the other hand, represent experimentally measurable thermodynamic quantities. To understand how these observables are obtained from computer simulations, a brief introduction about different simulation schemes will be given first.

Computer simulations can be roughly divided into deterministic and stochastic methods. Even though the two methods carry out calculations differently (the former is based on time evolution and the latter is based on

ensemble average), they are expected to produce the same statistical averages for any mechanical properties according to the fundamental hypothesis of statistical mechanics on the equivalence between ensemble average and time averages. Molecular Dynamics simulation is a widely used deterministic method, in which the time evolution of the particle trajectories in a system is calculated via Newton's Second Law. The mechanical observables are obtained by averaging over the time-dependent trajectories. In addition to static properties, Molecular Dynamics simulations provide dynamical information. However, the accessible time scale (on the order up to a few hundred nanoseconds) is far from adequate to reproduce the slow dynamics involved in systems with long time-scale relaxations, such as helix-coil conformational transition. Also, for systems consisting of multiple local minima, this method encounters considerable difficulty, since the possibility of being trapped in metastable states increase the statistical error in statistical averaging. Moreover, the dynamics produced by Molecular Dynamics simulations is basically in a (N,V,E) ensemble where energy is conservative. When this method is extended to other ensembles, extra constraints are needed, which impedes acquisition of real-time trajectories. In contrast, Monte Carlo simulation relies on the stochastic approach, in which the system evolves through a sequence of random transitions. Each probabilistic transition is equivalent to changing the state of the system. The collection of all the states sampled in the simulation is similar to an ensemble in statistical mechanics, and desired observables are obtained

from ensemble average. The moves in Monte Carlo simulations are probabilistic and do not conform to physical laws. Therefore, no true dynamic information can be obtained as in the time trajectories of Molecular Dynamics, but those moves can be designed in such a way to facilitate the barrier crossing involved in local energy minima. As a result, we anticipate Monte Carlo simulations to be more flexible in terms of diverse systems and computational efficiency. Moreover, Monte Carlo methods are also extendable to different ensembles without introducing extra constraints as in Molecule Dynamics. In this dissertation, the primary focus is to elucidate the static properties of polymer systems. Monte Carlo simulation is sufficient for this purpose, and the method is employed throughout the course of the dissertation research. In the following sections, the background and practice of the Monte Carlo method will be briefly reviewed. Applications of the Monte Carlo method to statistical mechanics studies of polymers and the limitations of the method will also be summarized.

1.2 Introduction to Monte Carlo Methods

Monte Carlo method is a numerical technique that involves a deliberate use of random numbers to solve a problem. In a many body problem, the Monte Carlo method is particularly useful in dealing with multivariable integrals needed for computing averaged mechanical properties. As will be shown later,

the Monte Carlo method converts an integral into an arithmetic average. For illustration purpose, we will begin with a simple example to show how an integral can be calculated through the Monte Carlo method, and the essential difference between the Monte Carlo method and the conventional numerical method based on Riemann summation.

Suppose one needs to evaluate the area enclosed by a circle of radius R, as shown in **Fig. 1.2**. Note for comparison, a square of side length 2R is also drawn in the figure. The calculation of the two-dimensional area involves an integral of two variables. Besides the exact integration, we may apply the conventional numerical procedure (Riemann summation) as shown in **Fig. 1.2 (a)**, to compute the area. In such a method, one may divide the area into many small uniform intervals and approximate each interval by a rectangle (or trapezoid) with a well-defined area. By summing up the areas of all intervals, the total area of the two-dimensional circle is obtained. Alternatively, we can use a random process to solve the problem. This process can be pictured as shooting dots randomly within the enclosing rectangle of side length 2R. If the number of shots is large enough, the area of circle is proportional to the number of shots within the enclosure of the circle. Considering the number of shots falling within the circle is m, then the ratio of M and the total number of shootings N approaches the ratio of the two areas. The area (A) of the circle is now calculated as $A = m \times 4R^2/n$. The statistics of the result can be improved if the number of shots N is increased.

In practice, the random shooting process can be programmed as follows. First, we prepare a sequence of uncorrelated random numbers uniformly distributed in the range [-R, R], and pick a pair (ξ_1, ξ_2) to sample for x and y in each trial or "shooting". Each ξ_1 , ξ_2 is compared with the function of the circle corresponding to ξ_1. Suppose the function of the circle is given by $y = f(x)$, then when $\xi_2 \leq |f(\xi_1)|$, this trial is counted as "on target". In this procedure, the only knowledge needed *a priori* is the function for the circle. The area of the circle (i.e., the integral of the function) is obtained from a robust stochastic process.

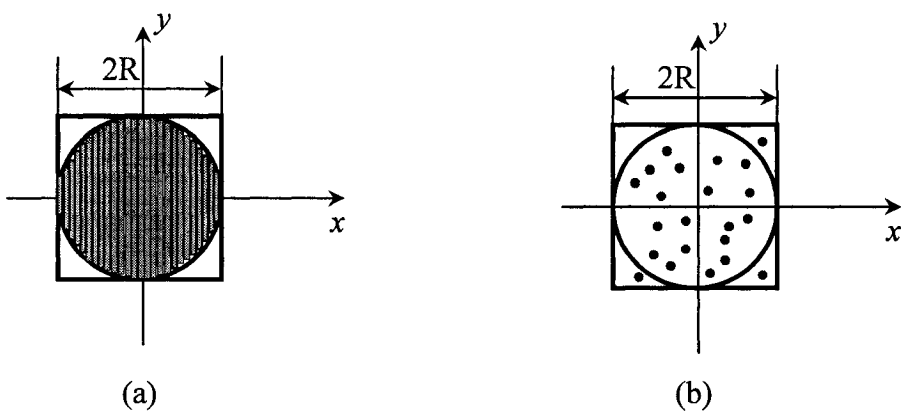


Fig. 1.2 An example of evaluating a definite integral: an area enclosed by a circle of radius R, by conventional numerical approach in (a), and by the Monte Carlo method in (b).

This simple example illustrates that the Monte Carlo method is capable of solving problems that are not inherently probablistic. This robut stochastic procedure can be further extended to handle problems involving multivariable

integrals. Since all the statistical mechanics observables for a many-body system are calculated by multivariable integrals in configurational space, the Monte Carlo method provides an efficient way of sampling the many-body high-dimension systems. In contrast, conventional numerical calculations will become computationally intractable for such systems by discretizing each variable uniformly in configurational space.

1.3 Importance Sampling

As discussed above, the essential use of the Monte Carlo method in a many-body system is to evaluate multivariable integrals. We now want to show how we should devise a sampling method to carry out this numerical process. Our focus is to achieve high sampling efficiency in a finite simulation. To this end, we need to introduce the concept of importance sampling, which biases the sampling toward the most significant component of the integrand.

Consider the a simple function $g(x)$, where x is in the range [a,b], the integral I is given by

$$I = \int_a^b g(x)dx \tag{1.1}$$

The average value of $g(x)$ is evaluated according to the mean-value theorem of calculus as follows:

$$< g(x) >= \frac{I}{b-a}. \tag{1.2}$$

If the integral is computed by selecting n points randomly in [a,b] on a

uniform distribution, the integral can be approximated as

$$I = \int_a^b g(x)dx \approx \frac{b-a}{n}\sum_{i=1}^{n} g(x_i) \qquad (1.3)$$

The validity of Eq. (1.3) comes from the law of large numbers, which guarantees that a sufficiently large sample makes the arithmetic average converges to the true value of the integral I.[2]

We may notice that this straightforward way to calculate the integral is similar to the conventional numeric method, except that now the integrand is sampled according to a uniform distribution function rather than by uniform discretization. But since this sampling plan does not discriminate between the contribution of different portions of the integrand, it results in a lower sampling efficiency especially when the integrand has a sharp distribution in the given range of x. This is because the sampling which occurs in the regime of smaller $g(x)$ is carried out with an equal probability as regimes of larger $g(x)$. To accelerate the numertical calculation, we can modify our sampling method, based on the concept of importance sampling, by which the region where the integrand is large is sampled more frequently than other regions.

Here we utilize the same integral $I = \int_a^b g(x)dx$ to illustrate the concept of importance sampling. Instead of a uniform distribution, we now choose the points x_i using a weighing probability $f(x)$. The probability distribution function $f(x)$ (≥ 0) obeys the normalization:

$$\int_a^b f(x)dx = 1 \tag{1.4}$$

So the integral I can be reformulated as

$$I = \int_a^b g(x)dx = \int_a^b \frac{g(x)}{f(x)} f(x)dx \tag{1.5}$$

This probability distribution is incorporated into the integral acording to

$$< g(x) > \approx \frac{1}{n} \sum_{i=1}^{n} \frac{g(x_i)}{f(x_i)}. \tag{1.6}$$

Where the variance of the average $< g(x) >$ is given by

$$\sigma^2 = \int_a^b \left[\frac{g(x)}{f(x)} \right]^2 f(x)dx - \left[\int_a^b \frac{g(x)}{f(x)} f(x)dx \right]^2 \tag{1.7}$$

We wish to select a $f(x)$ which makes σ a minimum (which means the best sampling efficiency). This may be achieved when the $f(x)$ is chosen as

$$f(x) = g(x) / \int_a^b g(x)dx \tag{1.8}$$

σ is reduced to zero in Eq. (1.7). Even this is only realizable when the quantity of $\int_a^b g(x)dx$ is known *a priori*, we expect the variance of the integral will be reduced if we choose a probability distribution $f(x)$ that looks similar to the original integrand and then sample the transformed integrand function (i.e., $g(x)/f(x)$) according to this $f(x)$.

Importance sampling is an extremely important concept in computing statistical mechanics properties via Monte Carlo simulations in that it increases the computation efficiency tremendously by sampling low energy states more

than high energy states. In statistical mechanics, mechanical property (observable) A is computed from averaging over all possible configurations for a many-body system, given by

$$< A >= \int A(X)P(X)dX \tag{1.9}$$

where X is a point in a multiple dimension configurational space, $P(X)$ is the probability distribution function of X. Such an integral can be transformed into an arithmetic averaging, as

$$< A >= \frac{\int A(X)P(X)dX}{\int P(X)dX} = \sum_{i=1}^{n} \frac{A(X_i)P(X_i)}{f(X_i)} / \sum_{i=1}^{n} \frac{P(X_i)}{f(X_i)} \tag{1.10}$$

If we choose the weighing probability as

$$f(X_i) = P(X_i) / \sum_{i=1}^{n} P(X_i) \tag{1.11}$$

the observable can be computed as

$$< A >= \frac{1}{n} \sum_{i=1}^{n} A(X_i) \tag{1.12}$$

In a canonical ensemble, $P(X)$ will take the simple form of

$$P(X) = \frac{\exp[-U(X)/k_B T]}{Z} \tag{1.13}$$

where k_B is the Boltzmann constant, T is the temperature, U is configurational energy, Z is the partition function given by

$$Z = \int \exp[-U(X)/k_B T]dX \tag{1.14}$$

If we use $P(X)$ as the weight function, any observable A computed has a zero

variance according to Eq. (1.7) and (1.8). However, since the partition function itself is not known *a priori*, we do not have the exact expression for $P(X)$. Nevertheless, we can make use of a Markov Chain process which follows this weight function in a simulation, as explained below.

Markov Chain is referred to as a sequence of transitions for evolution of a given system. The transition probability from state X_{n-1} to state X_n is associated only with these two adjacent states but not with the previous history of the process, i.e.,

$$W(X_n, T_n \mid X_{n-1}, T_{n-1}; \ldots; X_1, T_1) = W(X_n, T_n \mid X_{n-1}, T_{n-1}) \qquad (1.15)$$

the transition probability is only related to the state at time T_{n-1} and T_n. To ensure the states are ultimately distributed according to $P(X)$, there are four mathematical criteria for selecting the transition probability:

(i) Connectivity or ergodicity: there exists $W(X \to X') \neq 0$ for the whole configuration;

(ii) Positivity: $W(X \to X') \geq 0$ for all X and X';

(iii) Conservation: $\sum_{X'} W(X \to X') = 1$ for all X;

(iv) $\sum_{X'} W(X' \to X) P(X') = P(X)$ for all X.

Note that the equation in condition (iv) requires that the limiting distribution be the equilibrium distribution. This can be understood from the fact that the transition probability of a Markov Chain process is a conditional probability in

statistics, i.e., $\sum_{X'} W(X' \to X)P(X') = \sum_{X'} \frac{P(X',X)}{P(X)} P(X) = \sum_{X'} P(X',X)$

$= P(X)$. Actually, in a Markov chain, the evolution of the probability $P(X)$ has a kinetic interpretation, which can be described by the Master equation

$$\frac{dP(X,T)}{dT} = -\sum_{X'} W(X \to X')P(X,T) + \sum_{X'} W(X' \to X)P(X',T) \qquad (1.16)$$

Eq. (1.16) is a rate equation to account for the probability flow of the system moving in and out of state X. The first term describes the rate of all transitions out of the considered state, whereas the second term describes the rate of transitions into the considered state. The stationary solution (at equilibrium condition) to the Master equation is given by

$$\sum_{X'} W(X \to X')P(X) = \sum_{X} W(X' \to X)P(X') \qquad (1.17)$$

Using the detailed balance or microscopic reversibility (a more restrictive requirement than condition (iv)), one possible solution for Eq. (1.17) is

$$W(X \to X')P(X) = W(X' \to X)P(X') \qquad (1.18)$$

Hence, we arrive at the ratio of the transition probability for the forward and reverse processes in Canonical Ensemble, which reads

$$\frac{W(X \to X')}{W(X' \to X)} = \frac{P(X')}{P(X)} = \exp[-\delta U(X \to X')/k_B T] \qquad (1.19)$$

Eq. (1.19) is the key to develop an appropriate and a robust simulation procedure. The partition function is eliminated from the expression, and the transition probability depends only on the energy difference between the two successive states without any other knowledge involved. As shown by

Metropolis et al,[3] one possible solution for W is

$$W(X \to X') = T(X \to X')\exp(-\delta U / k_B T) \quad \text{if } \delta U > 0 \qquad (1.20)$$

$$= T(X \to X') \qquad \text{if } \delta U < 0$$

where T is a trial probability to transform a system for state X to X', and in the Metropolis method, $T(X \to X') = T(X' \to X)$. Besides, $W(X,X) = 1 - \sum_{x \neq x'} W(X,X')$. These formalisms can be abbreviated as $W(X \to X') = \min\{1, \exp[-\delta U / k_B T]\}$. These formalisms for W satisfy the basic criteria of positivity and conservation for the transition probability. Actually, the desired Boltzmann factor to achieve importance sampling in Eq. (1.14) is embedded in this solution, and the Boltzmann distribution will be obtained from a long enough simulation. Likewise, the ensemble averages can be simply calculated as algebraic averages of the all sampled states along the Markov chain. By using this transition probability, the probability distribution function is expected to be reproduced when the number of samples is large enough (i.e, $n \to \infty$).

In **Fig. 1.3**, we use a flow chart to illustrate how a Metropolis Monte Carlo simulation is conducted in practice for computing the an observable in canonical ensemble. The starting point is to generate an initial conformation which is physically reasonable. The next step is to generate a new randomly chosen configuration. The energy difference between the old and the new configuration is calculated, and so is the transition probability. If the trial

configuration has an energy is lower than that of old configuration, it is accepted unconditionally; however, if the trial configuration increases the energy, a random number is chosen to compare with the transition probability. If the random number is smaller than the transition probability, the new configuration is accepted; otherwise the system stays at the old configuration. The process is repeated until the calculated properties become convergent.

1.4 Monte Carlo Simulation in Different Ensembles

In the previous section, we have introduced Monte Carlo simulations of a **Canonical Ensemble** (*N, V, T*). This corresponds to the condition where the temperature, volume and particle numbers are fixed. A Monte Carlo simulation can be carried out in other ensembles. Here we will briefly summarize the transition probability for the Monte Carlo method in these ensembles. The formulations of transition probability in different ensembles share the same origin from Eq. (1.20). The fundamental differences lie in the probability distribution function associated with each ensemble.

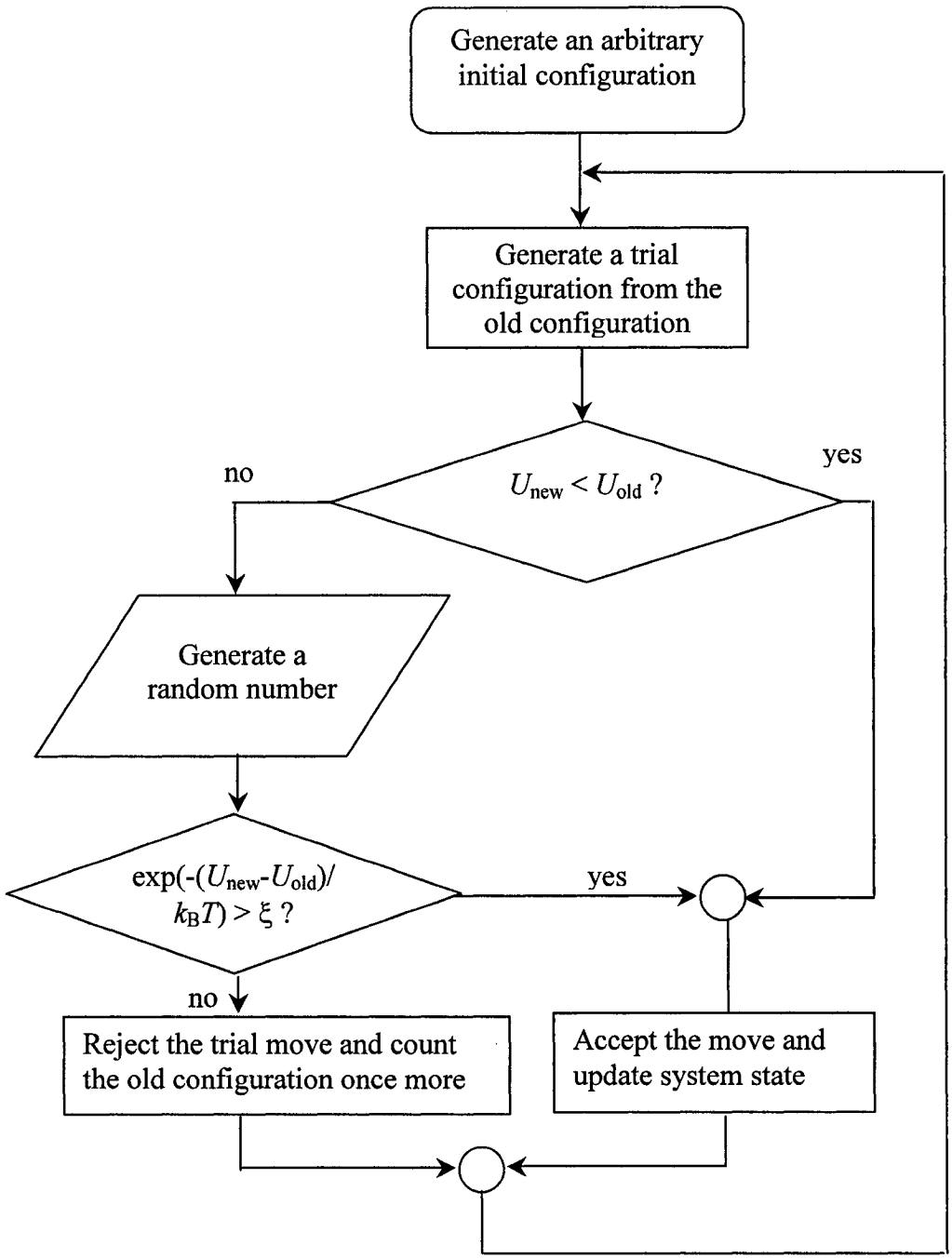


Fig. 1.3 Flow chart for a Monte Carlo simulation using Metropolis criterion.

In the **Microcanonical Ensemble**, a system consists of constant (*N, V, E*). The partition function of microcanonical ensemble is to measure the total number of states at energy *E*, given by

$$Z = \int \delta(U(X) - E)dX \qquad (1.21)$$

where E is the chosen energy of the system and $U(X)$ is the potential energy of configuration X. Any observable A can be calculated as

$$< A >= \frac{1}{Z} \int A(X)\delta(U(X) - E)dX \qquad (1.22)$$

Due to the constant energy, all states have *a priori* equal weight. With the energy constraint, the ideal simulation method will be Molecule Dynamics in which every time step is constrained by the constant energy. In the Monte Carlo method, it is difficult to keep the system energy constant because the method relies on random transitions. The random moves make the system deviate from the fixed system energy. In spite of its difficulty, we can employ a method to sample the fixed energy within a narrow window, i.e., $E - \varepsilon < U(X) < E + \varepsilon$ (ε being a small fluctuation in energy), and to introduce a demon with some extra energy (always positive) to facilitate the relaxation of configurations. The demon will transform the partition function into

$$Z = \sum_{X} \sum_{E_D} \delta(U(X) + E_D - E). \qquad (1.23)$$

From the above two partition functions, the distribution function of demon energy $P(E_D)$ can be evaluated as $P(E_D) \propto \frac{\partial(\ln Z)}{\partial E_D} \approx \exp(-\frac{E_D}{k_B T})$ because for

small E_D, $\ln Z(E^0 - E_D) \approx \ln Z(E^0) - \frac{E_D}{k_B T}$ where $E^o = U(X) + E_D$, a combination of the system energy in Eq. (1.22) and demon energy. Since the energy of demon should follow the Boltzmann distribution, the contribution of potential energy $U(X)$ in Eq. (1.22) can be singled out. To achieve the Boltzmann distribution for demon energy, the sampling is designed based on energy conservation as follows. (1) In the simulation process, a trial move that lowers the system energy (within the allowed energy window) is always accepted, and the demon energy is reset to $E_D - \delta U$, where δU is the energy change of the trial move; (2) A trial move that increases the system energy will only be accepted when the demon carries enough energy, i.e., when $\delta U - E_D < 0$. In this case, the energy of the demon is reset to $E_D - \delta U$. After repeating the above sampling processes, the energy of the demon follows a Boltzmann distribution, from which the system temperature can be defined. Actually, we may conceptualize the demon as a thermometer (i.e., kinetic energy) and the system acts as a reservoir and thermalizes the demon. It is noticeable that the method adopts the two fundamental elements in the Monte Carlo method, including detailed balance and utilization of random numbers. The random numbers are not applied to determine if a move is accepted as in Metropolis method for Canonical Ensemble, but are used to randomize our choices of a new state in the energy window. Restrictively speaking, the Microcanonical Monte Carlo method keeps the potential energy constant, instead of the total energy of the system as in the original definition of this ensemble.

In the **Isothermal-Isobaric Ensemble** (*N, P, T*), the number of particles, pressure and temperature are fixed, but the volume of the system is allowed to fluctuate. The partition function for this ensemble is

$$Z = \int_0^\infty Z_N(V,T)\exp(-PV/k_BT)dV \tag{1.24}$$

where Z_N is the partition function of the canonical ensemble with N particles (Eq. (1.14)). Therefore, for any observable A, we have

$$< A >= \frac{1}{Z}\int_0^\infty \int A(X)\exp(-[PV+U(X)]/k_BT)dV \tag{1.25}$$

We can still use a Markov process to sample the configuration space. In a cubic volume (with side length L) under periodic boundary condition, if the process of sampling the volume change is selected, the reduced positions ρ_i are the same but the box length L is changed. The reduced position is defined as

$$\rho_i = r_i / L, \tag{1.26}$$

where r_i is the real position of the i-th particle. Then the partition function Z becomes

$$Z = L(2\pi m\beta/h^2)^{3N/2}(1/N!)\int\exp(-\beta PV)dV\int\exp(-\beta U(r^N))dr^N \tag{1.27}$$

where $\beta = 1/k_BT$ (k_B being Boltzmann constant and T the temperature), h is the Planck constant, V is the volume of the system and $U(r^N)$ is the total interaction of the N particles in the system. In terms of reduced position, Eq. (1.27) gives rise to the probability distribution, which is proportional to

$$\prod(V) \propto \exp[-\beta PV - \beta V(\rho^N, L) + N\ln V] \tag{1.28}$$

where $\prod(V)$ is the probability of a system configuration at volume V. $N \ln V$ comes from the argument dr^N when it is transformed to the reduced coordinate in Eq. (1.27) and $V(\rho^N, L)$ is equal to $V(r^N)$. If microscopic reversibility or detailed balance is employed, the transition probability for the change in volume can be chosen as

$$W(X \to X') = \min\{1, \exp[-\beta P \Delta V - \beta \Delta V(\rho^N, L) + N \Delta \ln V]\} \qquad (1.29)$$

where X, X' represent two system states, ΔV and $\Delta \ln V$ the volume and logarithm volume change from X to X', respectively; and $\Delta V(\rho^N, L)$ the energy difference of the transition. If we define a new variable containing pair interactions and P-V work as

$$U' = P\Delta V + \Delta V(\rho^N, L) - (N \Delta \ln V)/\beta \qquad (1.30)$$

we can then carry out the energy calculation in similar manners to the Monte Carlo algorithm in the Canonical Ensemble (presented in flow chart of **Fig. 1.3**) for the moves that change system volume. The transition probability can be chosen as

$$W(X \to X') = \min\{1, \exp(-\beta \Delta U')\} \qquad (1.31)$$

For moves that do not involve volume changes, the original Metropolis criterion (Eq. (1.20)) for the Canonical Ensemble can be used.

In the **Grand Canonical Ensemble**, the parameters to be specified are the chemical potential μ, the volume V and the temperature T. The particle numbers in the system are allowed to fluctuate, therefore this ensemble is useful for

simulating systems near a phase boundary. In a Grand Canonical Ensemble, the partition function

$$Z = \sum_N \left(\frac{a^N}{N!} \right) \int \exp(-U(X^N)/k_B T) dX^N \tag{1.32}$$

and observable A is calculated by

$$< A >= \frac{1}{Z} \sum_N \left(\frac{a^N}{N!} \right) \int A(X^N) \exp(-U(X^N)/k_B T) dX^N \tag{1.33}$$

where a is defined as

$$a = \lambda (h^2 / 2\pi m k_B T)^{-3/2} \tag{1.34}$$

h being Planck's constant and the prefactor resulting from the momentum-space integration. The absolute activity is given by

$$\lambda = \exp(\beta \mu) \tag{1.35}$$

A viable GCMC algorithm can have three steps: 1) configuration changes the existing particles; 2) adding new particles; 3) deleting existing particles. For the configuration changes, we can use exactly the same method as in the canonical ensemble method. For adding a particle in a random place to the volume V containing N particles, the probability of having N+1 particles is

$$P(X^{N+1}) = [a^{N+1}/(N+1)!] \exp[-\beta U(X^{N+1})]/Z \tag{1.36}$$

Similarly, the probability of deleting one particle from a system containing N particle will be

$$P(X^{N-1}) = [a^{N-1}/(N-1)!] \exp[-\beta U(X^{N-1})]/Z \tag{1.37}$$

In the spirit of Eq. (1.18) we can create the transition probability of addition

and deletion of particles as the following,

$$W(X^N \rightarrow X^{N+1}) = \min\{1,[a/(N+1)]\exp(-\beta(U(X^{N+1})-U(X^N))\} \quad (1.38)$$

$$W(X^N \rightarrow X^{N+1}) = \min\{1,[N/a]\exp(-\beta(U(X^{N-1})-U(X^N))\} \quad (1.39)$$

In the GCMC simulation, each of these three steps may be carried out with equal probability.

These examples are employed to illustrate how the transition probabilities are chosen in the Monte Carlo method. In this research, simulations were conducted primarily using Canonical Ensemble Monte Carlo method. Below will be given a brief review of some polymer chain models and algorithms commonly used in Monte Carlo simulations for polymer systems.

1.5 Polymer Models and Algorithms Used in Monte Carlo Simulations

In Monte Carlo simulations, polymer chains can be modeled in various ways. Different models simplify the structure of chain molecules to different extents. The research for these models is twofold: (1) they allow us to crosscheck those universal properties independent of detailed chemical architectures; (2) the effects of local structure on polymer properties can be examined. Also, different models can be used in different simulation methods to facilitate the calculations of static and/or dynamical properties, depending on the characteristics of the chosen model. The existing models can be roughly divided into atomistic models and coarse-grained models.

Building a molecular model to incorporate the detailed chemical structure of polymer chains is a desirable approach because the obtained properties can be used to make a direct comparison with experiments. However, such a model suffers from two major problems: (1) intense computation is required; (2) it is impossible to pinpoint the decisive factor (such as the dominant interaction) to understand the investigated system.

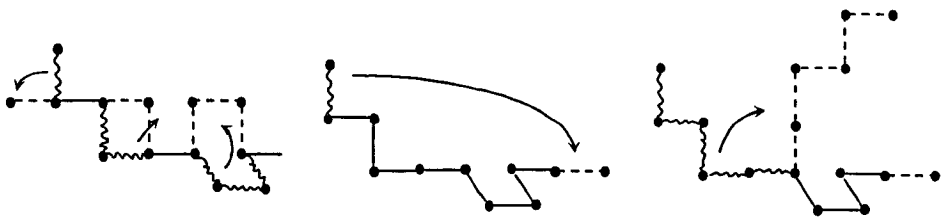
Owing to the intense computation required, models require large computing power facilities, and are limited to single chain problems or multiple short chain systems. For the sake of computational time and physical understanding, simulations are carried out using more simplified models that coarse-grain part of the repeat unit or several repeat units on a polymer molecule into more geometrically simple units, such as blobs. This type of model provides the physics at different length scales, and facilitates our search for possible universal polymer properties insensitive to polymer chemical architectures.

Coarse-grained models are more tractable in terms of computational time, but there is no robust way to create those models. Selection of a polymer model remains at the level of state-of-the-art, and depends heavily on the physical problems of our interest. The other challenging problem is how to parameterize these simplified models to generate quantitative information for realistic materials. By and large, there are two types of models: lattice and off-lattice models. The former ones are computational more efficient but the latter ones more closely represent real structures of polymers. In the following, some

representative polymer models and available Monte Carlo algorithms associated with each of these models were to be presented.

Lattice models are the most simplified representation of polymer chains. They were initially employed to develop polymer theories, for example, by Flory for polymer solutions and polymer melts. In addition to theory, lattice models have been studied extensively in Monte Carlo simulations.[4] The most widely used model in this category is Self Avoiding Walk (SAW) model because it accounts for the excluded volume in realistic polymer molecules. In a SAW polymer chain, the polymer chain is grown sequentially from monomer units, but no two polymer units can occupy at the same lattice site. Considering the greatly reduced degrees of freedom in a SAW chain, the model is computationally quite efficient. A variety of Monte Carlo algorithms have been devised for simulating lattice polymer models, such as the original Verdier-Stockmayer algorithm,[5-8] reptation algorithm[9,10] and pivot algorithm,[11-13] as shown in **Fig. 1.4**. The Verdier- Stockmayer algorithm in **Fig. 1.4(a)** is quite self-explanatory, but it has the drawback of non-ergodicity and have very small acceptance rate at high polymer volume fraction. In the reptation algorithm, as shown in **Fig. 1.4(b)**, one cuts one unit on one side of the chain and grows another unit on the other end of the chain in an arbitrary orientation. The advantage of this algorithm is that it only needs one vacant site near a chain end for a successful move and thus can be used for significantly denser systems. In the pivot algorithm (**Fig. 1.4(c)**), a randomly chosen

segment of chain is rotated around a pivotal point. This algorithm allows one to generate new configurations rapidly, which are not dynamically correlated to their predecessors. But it is not suitable for dense polymer system, where the acceptance probability of this type of drastic sweeping move is extremely small.



(a) (b) (c)

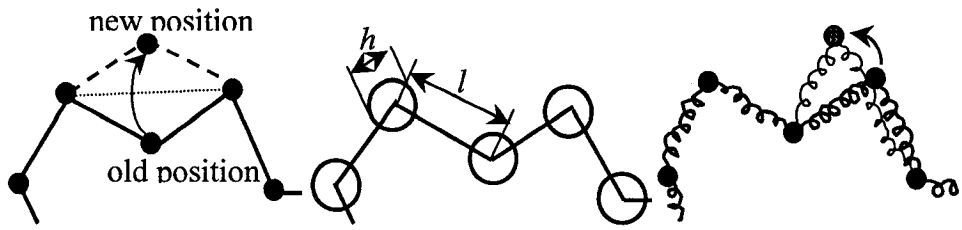
Fig. 1.4 Various examples of Monte Carlo algorithms for SAWs: sites taken by beads are shown by dots, and bonds connecting the bead are shown by lines. Bonds that are moved are shown as wavy lines (before the move) or broken lines (after the move), while bonds that are not moved are shown as solid lines. Verdier-Stockmayer algorithm in (a), reptation algorithm in (b), and pivot algorithm in (c).[4]

The above lattice models are based on a cubic lattice, and have difficulties in accurately depicting the orientation of chain molecules. Also the discrete space may induce artifacts for calculating some properties such as the structure function for scattering experiments. To improve the model, several modifications have been suggested. For example, other kinds of lattices, such

as diamond lattice, have been devised to represent certain structures of polymers, which can partially relieve the restricted spatial constraints for cubic lattice models.[14-16] Also the lattice model can be modified by incorporating some off-lattice features. One such successful model for the simulation of many-chain system is the bond fluctuation model.[17-20] This model is between the simple SAW model of **Fig. 1.4** and the off-lattice model (as in **Fig. 1.5**, to be presented next), because the vector that connects two monomers can take 36 tabulated values rather than 6 in a 3-D cubic lattice. The constraint of polymer units on lattice is relaxed, but the computational time is increased only moderately.

Off-lattice models offer a more realistic representation of polymer chains and thus can be applied to elucidate non-zero density polymer systems. Some off-lattice models include the freely jointed chain model, pearl necklace model and the bead-spring model, shown in **Fig. 1.5**. In a freely jointed chain, each polymer is modeled by a succession of N rigid bonds of length L joined together at arbitrary angles. The Monte Carlo algorithm to relax the chain conformations in this model can use a crankshaft move, with a random rotation of a randomly chosen bead along the axis connecting its two neighboring beads, as shown in **Fig. 1.5 (a)**. In the pearl necklace model, shown in **Fig. 1.5 (b)**, the polymer chain is modeled as beads of finite size connected by bonds. The advantage of this model is that exclusion volume can be adjusted readily by changing the ratio of bead size h versus bond length l.[21] When the bead size is equal to the

bond length with hard core repulsions incorporated, the model is known as "Tangent Hard Sphere Model". The algorithm used for the freely jointed chain can also be used for this model. When the hard core repulsion is imposed between beads, an additional procedure is required to check for overlap of the beads. The bead-spring model, shown in **Fig. 1.5(c)**, consists of beads connected by springs, is somewhat more versatile and can be used in Molecular Dynamics simulation as well as in Monte Carlo simulation. Because of the elasticity of bonds, this model needs an additional mode to relax the chain conformation, e.g., random displacement of a bead.



(a) (b) (c)

Fig. 1.5 Some off-lattice models for polymer chains. Freely jointed chain in (a), pearl necklace chain in (b) and bead spring chain in (c).[4]

Besides these algorithms, in order to deal with problems arising from polymers of high packing densities, more sophisticated algorithms have been developed to increase the sampling efficiency, such as configurational bias Monte Carlo (CMBC) by Frenkel et al.[22-24] These advanced algorithms rely on

the idea of choosing a biased trial move mode so that high statistical weight conformation have a higher chance of being sampled and accepted. In the presence of a bias move, we need to rewrite Eq. (1.18) as

$$P(X)T(X \to X')W(X \to X') = P(X')T(X' \to X)W(X' \to X) \qquad (1.40)$$

where $T(X \to X')$ is trial probability, meaning the probability of attempting a move of a particle from X to X'. For the simple crankshaft trial move, $T(X \to X')$ is $1/2\pi$, and so is the reverse move $T(X' \to X)$. Therefore, the trial probability is cancelled in Eq. (1.18). However, if we use trial moves that do not have the same probability for forward and reverse direction, the trial probability will be appear explicitly and will affect the selection of transition probability. For example, if we want to generate trial configurations with a probability that depends on the potential energy of that configuration:

$$T(X \to X') = f[V(X')] \qquad (1.41)$$

For the reverse move, we have

$$T(X' \to X) = f[V(X)] \qquad (1.42)$$

If we want to sample the canonical ensemble, we have to generate configurations with a Boltzmann distribution. From Eq. (1-40), the transition probability can be chosen as

$$W(X \to X') = \min\{1, \frac{f[V(X')]}{f[V(X)]}\exp[-\beta(V(X') - V(X))]\} \qquad (1.43)$$

This derivation shows that we can introduce an arbitrary biasing function $f(X)$ in the sampling scheme, but need to be counter-balanced in the

calculation of the transition probability. In a lattice polymer model consisting of many chains, the outline for the CMBC scheme would be:

(i) Generate a trial conformation (by growing bead by bead) using Rosenbluth scheme[25] and compute its Rosenbluth weight $w(X')$. In the Rosenbluth scheme of a lattice model, we first randomly insert a polymer unit, and its energy is denoted by $V_1(X')$ and its weight $w_1(X') = k\exp[-\beta V_1(X')]$, where k is the coordination number of the lattice (for cubic lattice $k = 6$). There are k ways to generate the next polymer unit on the same chain with index i, and we select a direction j with a probability

$$p_i(X') = \frac{\exp[-\beta V_i(j)]}{w_i(X')} \tag{1.44}$$

where $w_i(X')$ is defined as

$$w_i(X') = \sum_{j=1}^{k} \exp[-\beta V_i(j)] \tag{1.45}$$

The interaction $V_i(j)$ denotes all interactions of unit i (for the trial direction j) with other molecules in the system and with units 1 through i-1 of the same molecule. In this way, the entire chain (with length l) is grown to completion bead by bead, and eventually the Rosenbluth factor of configuration X' is determined as

$$w(X') = \prod_{i=1}^{l} p_i(X'); \tag{1.46}$$

(ii) Retrace the conformation of an arbitrarily selected existing chain

(which is to be replaced by the trial chain) and determine its Rosenbluth weight $w(X)$ in a similar way;

(iii) Accept the trial move with a probability

$$W(X \to X') = \min\{1, w(X')/w(X)\} \tag{1.47}$$

It is noticeable that the implementation of the CMBC method is quite mathematically involved, and is lack of efficiency (for each trial move a whole chain segment is regrown and weighting factors need to be calculated at each step). It is mostly useful in a very dense system with a very low acceptance probability in a simulation when other methods encounter great difficulties. Also, the method is useful when insertion of an entire polymer chain into a dense system is needed, such as in Grand Canonical Ensemble Monte Carlo. This model is chosen to make a comparison with those algorithms used in my simulations. In my dissertation, all the simulations are carried out in canonical ensemble for systems mostly consisting of single polymer chains or lattice model for multiple short chains in polymer solutions. Therefore, the simple moves such as reptation and crankshaft are sufficient to sample polymer conformations efficiently.

1.6 Challenges for Monte Carlo Simulation of Polymers

Though the Monte Carlo method has its unique advantages in studying complex system, it inevitably encounters some difficulties, many of which are shared by other types of simulations as well. The difficulties primarily arise

from the following problems:

1) Only systems of $N = 10^3 \sim 10^6$ units are accessible for computer simulations, which are much smaller than any realistic systems. In the simulations, we intend to understand the system of bulk solutions or bulk materials through a finite simulation. Appropriate boundary conditions need to be applied for a system of non-zero polymer densities. The conventional boundary condition for a finite system is the periodic boundary condition in the simulations, such as cubic cells. Under this condition, the surrounding cubic cells are assumed as simply replica of the central simulation cell. In the calculations of energy, for each particle, we draw an imaginary box of same size as the simulation cell. The distance between this particle and other particles is calculated from the minimum image convention, i.e., with the nearest image of the other particle, as shown in **Fig. 1.6**.This convention is utilized to compute the short-ranged interaction. For longer ranged interactions, one either needs to truncate the potential[26] or more accurately, use Ewald Summation method.[27] The finite size effects in the simulations, which may hinder the accuracy of results, cannot be totally avoided. Increasing the size of the simulation cell is one possible way to test the results, but is usually compromised by the lack of computing power.

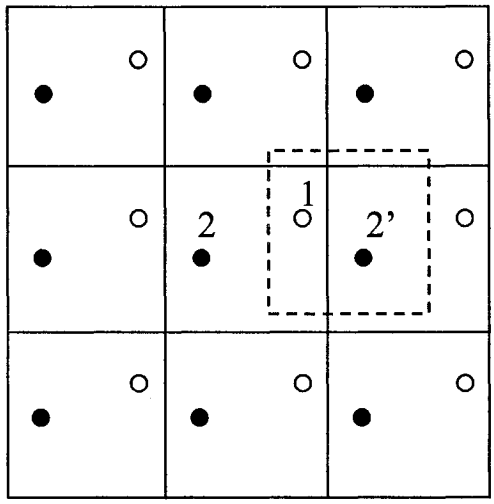


Fig. 1.6 Periodic boundary condition and minimum image convention. The configuration in the central box is duplicated to all its neighboring boxes. When calculating interaction energy for the particle 1, only the nearest image of the other particle, i.e., 2' is considered.

2) The finite number of simulation moves may not be sufficient when a property is calculated. This problem is related to the hypothesis of ergodicity for the simulation algorithm. The finite simulation may not sample the entire phase space enough, and results in an erratic average. Also, the convergence criterion of a simulation is somehow *ad hoc*, and may cause an inaccuracy, especially, when the simulation has a low acceptance probability, for example, at low temperatures or with multiple metastable states.

3) Dynamic properties are not accessed by Monte Carlo method because of those unphysical moves used in the simulations. One may argue that the Monte

Carlo simulations represent the long time dynamics of a system because at long time scale, the velocity correlation diminishes. Each move in the Monte Carlo simulations can be viewed as a long enough time unit, during which the short length-scale correlation between local monomers is cancelled. However, such an argument has not been fully validated.

For a polymer system in particular, some of these difficulties become even more severe. The long chain nature of a polymer adds the complexity of condensed matters to another dimension. Besides the inter-particle correlation, the multiple intramolecular conformations increase the difficulty level for the simulations due to the problems of intrinsic multiple length scales. In addition to multiple length scales, the amounts of degrees of freedom slow down the efficiency of a simulation. Also, sophisticated interactions (bonding, angular, non-bonded, electrostatic, etc) are computationally demanding. Nevertheless, simulations provide the solution when no theory is available for the chosen model. To increase the reliability of simulations, different algorithms and different theoretical models should be utilized to conduct cross-examination for an investigated problem.

1.7 Motivations and Objectives

Characterization of polymer properties advances our understanding of polymer materials, and results in useful applications. Among available techniques, the methods associated with an external electrical field are

commonly used to explore the property of a material through its response to the applied field. For example, electrophoresis — the detection of the mobility of charged nucleic acid or protein fragments under an applied electric field, has become a common tool in biomedical analysis.[28] Moreover, polymer materials display interesting behavior under the external field, such as electric field induced fibers-type assembly on the surface morphology of piezoelectric materials,[29] and field induced the surface structure distortion of materials.[30] The fundamental problem lies in whether the observed morphological change is due to conformational transition and/or packing of polymer chains. Hence, one of the objectives of this dissertation is to investigate the influences of electric fields on chain conformation.

In addition to electric or magnetic fields, external fields have a broader sense, such as confining boundaries in composite materials. Local confinement is a basic physical feature found in this type of materials, for example core-shell structures. The novel core-shell structures were constructed based on self-organization of the mixture of neutral/polyelectrolyte copolymers and oppositely charged surfactants.[31] The neutral/polyelectrolyte copolymers form a larger micellar shell to trap smaller surfactant micelles of spherical or cylindrical geometry in the core. In polymer/liquid crystal composites, polymer matrices induced confinement to liquid crystal molecules.[32-35] Park et al have developed a novel method to encapsulate liquid crystals in monodisperse micron-sized polymer particles.[36] Vorflusev and coworkers have prepared the

phase separated composite films, with adjacent layers of liquid crystal and polymer, which greatly enhance the performance of liquid crystal displays.[37] Also, lasers can be used to create optical trapping forces to confine spherical and non-spherical objectives. This technique has been widely developed as optical tweezers.[38-40] Modeling of these confining boundaries has attracted my attention, and study of the polymer properties under confinement is one of the objectives of this research.

In this dissertation, an extension of the concept of composite materials to single chain level was also attempted. In reality, such a model system mimics a polymer grafted onto a surface, and our primary interest was on curved surfaces, because curved surfaces allows us to model polymers labeled with nanoparticles. Experiments have been devised to utilize nanoparticle labeled macromolecules to elucidate biological systems, such as the binding mechanism of proteins.[41] In a more general view, the model for nanoparticle labeled macromolecules shares many features similar to chromophores labeled on polymers, and drug bound to DNA. Development of a general model to study these systems has been considered in this research.

Simulations beyond single chain problems were also attempted in my research. These simulations are motivated by recent experiments for the synthesis of nanoparticles in templated polymers.[42-44] Experimentally, templated polymers were introduced to regulate the growth of nanoparticles and control their size and geometry. However, our understanding of the

mechanism behind templated polymer synthesis was still in its infancy. These systems consist of at least three components, including precursors, templated polymers and solvents. In our view, templated polymers may form cavities to accommodate the precursors of nanoparticles. After the reaction of precursors is triggered, the units of nanoparticles may start to aggregate and grow within the cavities. To control the size and geometry, those confined cavities induced by polymers may serve as reaction vessels and molds at nanometer scale. Meanwhile, it is known that the thus obtained nanoparticles nucleate onto the templated polymers, a mechanism that prevents nanoparticles from further aggregation. In this case, polymers may be pictures as an external driving force to manipulate aggregate formation.

Here is a brief description of the studies to be presented in the following chapters.

- *Monte Carlo simulations for a fluctuating sphere labeled on a flexible polymer chain in good solvent*

  In this project, a novel fluctuating sphere model was devised and studied through simulations. The fluctuating sphere was an elastic ball whose size was allowed to fluctuate based on the Hook's law. Therefore, the exlcuded volume interaction arose from the luctuating sphere. This model was found to be very useful for describing many systems. First, the flutuating sphere resembles a vibrational probe under constant vibrational motion. Second, the model is similar to polymers grafted to nanoparticles or

fluoresence probes. Third, the fluctuating spheres account for a size fluctuation for a flexible chain segment on a polymer. This point is particularly important from the standpoint of chemical engineering because polymers are usually pictured as flexible dumbbells. Each blob in the dumbbell represents the half segment of a flexible polymer molecule. The fluctuating sphere model allowed us to elucidate some fundamental questions, including (1) how the labeled probes affected chain conformation; (2) how the conformational fluctuation influneced a vibrational probe; (3) how to parameterize a realistic polymers.

- *Theoretical studies of conformational behavior of chain molecules containing polar groups and electric fields effects*

Polar polymers contain intrinsic dipoles and respond to electric fields, which leads to many applications such as actuators, sensors and artificial muscles.[45] However, how polar polymers respond to electric fields on molecular level has received little attention. In this study two questions were to be addressed: (1) the role of dipolar interactions within a polar polymer on its conformational behavior, and (2) how external electric fields influence the conformations of polar polymer. Poly(vinylidene fluoride) (PVDF), $(CH_2CF_2)_n$ was chosen as a representative example of polar polymers. In the first part of the study, an all-atom model was employed with a realistic force field from literature. In the second part, two polar polymer models were simulated for comparison: one derived from PVDF

model with all atoms and original partial charges, the other one obtained from coarse-graining the methylene group and fluoride methylene group into alternating positive and negative charge sites on a chain backbone. These simulations were carried out at weak coupling limits in which the monomer-monomer interactions were ignored, so that existing theories can be tested by our simulations. The subtle difference between the two models can be useful to disclose the effects of chemical architectures on field induced conformational deformation. Also the results could serve as a guide for designing appropriate coarse-grained level models for future studies.

- *Molecular alignment of rigid rods in non-rigid spherical pores*

In this study, the orientation ordering of two rid rod chains confined by spherically harmonic potentials were investigated in great details. The rigid rods were modeled as Shish-Kebab chains, containing tangent hard spheres aligned along in the same axis. The harmonic potential was chosen to model non-rigid cavities, media induced local confinement or the trapping potential of optical tweezers. The problems of our interest included (1) effects of applied potentials on orientation ordering of rods; (2) development of asymptotic analysis to obtain the limiting order parameters for two Shish-Kebab chains confined to the limiting weak and strong fields; (3) comparison of different rod models to explore the effects of local rod smoothness on the orientation ordering of confined rods; (4) alingment of multiple rods confined to spherically harmonic potentials. Together with

the PVDF in electric fields, we intended to find the correlation between static properties and the local chemical architecture of polymers in external fields.

- *Theoretical study of the effects of templated materials on aggregate formation*

A simple blend model was investigated in which the dispersed small particles are solvated in the medium of templated particles. And the simulations were carried out on the two-dimensional lattice. Two types of templated particles were compared: templated polymers and templated monomers (without chain connectivities between monomers). Despite its simplicity, this two-dimensional lattice model facilitates the simulations and increases the statistical accuracy for our results. It also helped reveal the fundamental mechanism for the polymer mediated aggregation. In this project, we consider the following issues: (1) search of an effective method to quantify the aggregate size in the simulations; (2) effects of the interaction (attractive or repulsive) between templated particles and units of aggregates on chain conformation and aggregate size distribution; (3) effects of chain connectivity on aggregation because templaed polymers and templated monomers may induce cavities (solvation cages) of different sizes in blends; (4) effects of an confined environment on polymer mediated aggragation. The study approached the thermodynamic limits for aggregation, and was highly beneficial to identify the origin of polymer

mediated aggregation.

In the following chapters, each topic were to be elaborated in detail to introduce models, simulation methods, results and discussions. In addition to independent conclusions given in each chapter, a comprehensive summary will be presented at the end of this dissertation.

- Chapter 2 -

Monte Carlo Simulations for a Fluctuating Sphere Labeled on a Flexible Polymer Chain in Good Solvent

2.1 Introduction

Polymer chains are complex molecules due to multiple length scales present in a chain molecule. To probe their conformational and dynamical behaviors, a great deal of work has been devoted to develop spectroscopic techniques, via fluorescence probe labeling. This approach can be applied in probing the conformational behavior of polymers and copolymers,[46,47] diffusion and viscosity of polymers,[48-50] and protein folding mechanisms,[51] etc. The labeled chromophores vary from small fluorescent molecules to nanoparticles.[52] An important system similar to the probe labeling problem was the drug-DNA binding in biomedical research, in which bio-active molecules were bound to polymeric molecules.[53,54] Further, not only does a small molecule bind with polymeric molecules, but also a chain segment, such as the protein molecule in the enzyme-RNA polymerase.[55] Recently, a novel experimental method has been developed to facilitate the crystallization of RNA molecules via labeling a bulky protein segment onto a long chain RNA.[56]

Labeled probes and bound chain segments can be rigid or flexible, i.e., with or without size fluctuation. Monte Carlo simulations were conducted to investigate a fluctuating sphere model, which mimicked the size fluctuation of a labeled chromophore and a bound chain segment. In a dilute solution, the polymer molecule can be modeled as a chain composed of a string of blobs in which each blob contains several repeated monomer units. The blob model represents a molecular model based on the length scale of monomers,[57] and the blob size can be ascribed to the balance between the porosity and excluded volume of blobs. In contrast to the blob model, the Flory mean field modeled coarse-grains the chain molecule to a large blob, and treated mean chain size as a variable.[58] Frankel and co-workers extended the Flory's model to incorporate the fluctuation of chain size by assuming the fluctuation was governed by a Hookean elastic potential.[59] They argued such a model was particularly useful for the dynamics of flexible polymer chains in dilute solution because it accounted for the stress of the chain molecule and hydrodynamics interactions under chain size fluctuation. Inspired by the model of Frankel et al., in this work, the bound chain segment on a polymer molecule was coarse-grained to a sphere with a variable σ_e, where $\sigma_e \sim N^{3/5}$ in good solvents (N was the chain length of the bound chain segment) and was close to the mean chain size. The instantaneous size of the segment was σ_e (≥ 0), and the segment was subject to restoring forces arising from its conformational entropy. Following the work by Frankel et al., the fluctuating energy of the sphere size was depicted by a

harmonic-like potential $k(\sigma - \sigma_e)^2$ where the effective spring constant of the sphere was k. However, in this model, the sphere and the attached polymer were assumed to interact through hard-core repulsions. In terms of length scale, the investigated model was at a level between the Frankel's (or Flory's) and blob models. Further, the harmonic-like potential could also be employed to model the vibrational energy of a chromophore, and was useful to explore the labeled probe (or drug) on a polymer molecule. From this model were elucidated two specific physical problems: one was the origin of the confinement effect on the labeled probe (or a drug) induced by a polymer molecule; the other was the feasibility of treating a bound chain segment as a fluctuating hard sphere.[60] The inception of our work was the simulations of a fluctuation sphere in the presence of a grafted polymer, the simulated mean sphere size displayed a non-monotonic behavior as the force constant of the fluctuating sphere was decreased, i.e., the mean size decreased first and then increased again after passing a minimum. When the intrinsic sphere size was large and the size fluctuation was also large, the distribution of instantaneous sphere size became bimodal. Rationales of the simulation results were attained through a greatly simplified model composed of a fluctuating sphere and a tethered dimer. When the force constant of the sphere was small enough, the groove in the dimer exhibited a pronounced confinement effect on the fluctuating sphere and resulted in two distinct regions. The sphere could be either confined in the groove between two neighboring monomers or freely vibrate outside the

groove. When the chain length of the grafted polymer molecule was increased, the confinement effect was enhanced due to coiled chain conformations. As a result, the bimodal distribution became more pronounced. These findings provided physical insights regarding the confinement effect induced by a flexible chain polymer, and might be useful in reasoning the role of the long chain DNA molecule on drug-DNA binding. [53,54] Also, the behavior of the vibrational spectroscopy of a labeled chromophore subject to a grafted bound segment was summarized at the end to illustrate the confinement effect on a labeled molecule.

The feasibility of modeling the bound chain segment as a fluctuating sphere was also tested. The flexible chain polymer model was divided into two chain segments: a main chain segment and a bound chain segment. In the fluctuating sphere model, the bound chain segment in the flexible chain model was replaced with a fluctuating sphere. The size distribution of the bound chain segment was obtained from simulations, and was fitted with the fluctuating sphere model. By adjusting the two parameters (k and σ_e) in the fluctuating sphere model, a good fit can be arrived for the distribution function of the size of a bound chain segment, and the calculated sizes of the main chain segment were found to be in good agreement between these two models. The results indicated that the simple two-parameter model captures the size fluctuation of a bound chain segment. Also, this simple model was extended to parameterize a polymer chain to a dumbbell. Since the dumbbell model has been widely

applied to polymer dynamics,[61] this model would be useful to simplify the chain structure in reasoning the dynamical properties of dilute polymer solutions.

The influences of the attached ball on the conformation of the polymer chain were also studied. The results show that polymer chain was slightly expanded by the elastic ball labeled at one chain end, but the scaling behavior of the polymer dimension against chain length was not changed. The chain dimension was also affected by the position of the label, which could be attributed to different hard sphere repulsion effects.

2.2 Model

An isolated tangent hard sphere chain was considered, with N monomers labeled with a fluctuating sphere with an instantaneous size σ on one chain end, as shown in the schematic plot of **Fig. 2.1**. [Note, in the rest of the manuscript, bead and sphere were referred to as monomer and chromophore (or bound chain segment), respectively.] In the model, the beads were allowed to move freely, but the neighboring beads remained contact at one point. The bond length d was set equal to the diameter of monomers b. Like monomers, the fluctuating sphere also remained contact with the neighboring bead of one chain end, in which the instantaneous sphere radius was σ. The hard core repulsions were incorporated to account for the excluded volume of particles, and the interactions between particles $u(r_{ij})$ were given by

$$u(r_{ij}) = \begin{cases} 0, & r_{ij} \geq \dfrac{d_i + d_j}{2} \\ \infty & r_{ij} < \dfrac{d_i + d_j}{2} \end{cases} \tag{2.1}$$

where r_{ij} was the distance between particles i and j; d_i was the diameter of particle i, which was equal to d for monomers and $2\sigma_e$ for the sphere of fluctuating size. The vibrational energy $V(\sigma)$ of the sphere was assumed to be governed by a harmonic potential with a cutoff at $\sigma = 0$,

$$\beta V(\sigma) = \begin{cases} k(\sigma - \sigma_e)^2, & \sigma > 0 \\ \infty & \sigma \leq 0 \end{cases} \tag{2.2}$$

where k was the reduced force constant; σ_e was the intrinsic sphere radius (or size) at the minimum of the vibrational energy; $\beta = 1/k_BT$; T was the temperature; k_B was the Boltzmann constant. For simplicity, β was set to 1 and b= 1, i.e., the vibrational energy and length were considered in units of k_BT and bead diameter b, respectively.

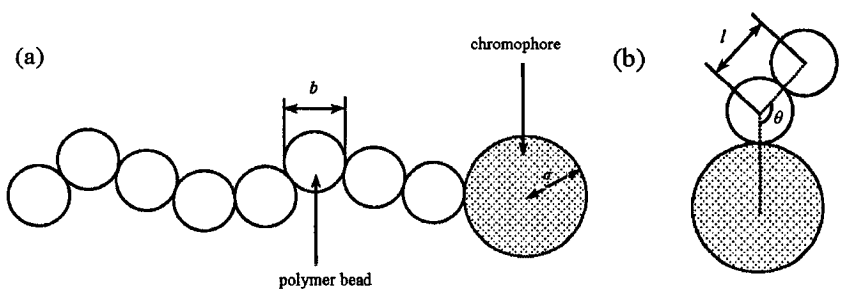


Fig. 2.1 Schematic plots for a fluctuating sphere (denoted by the shaded sphere), with an instantaneous size σ, bound to a flexible chain in (a), and for a dimer of bond length l grafted onto a fluctuating sphere, with an instantaneous angle θ between the bond of l and the bond connecting the end bead and sphere

in (b).

2.3 Monte Carlo Simulations

To relax chain conformation and sphere size simultaneously in the simulations, different Monte Carlo moves were combined, including reptation, crankshaft and sphere vibration moves. The weighing probabilities of the above three moves were P_1, P_2, and P_3, respectively, and $P_1 + P_2 + P_3 = 1$. For each simulation move, one of the above three algorithms was randomly chosen according to their weighing probabilities. The moves were carried out as follows: (1) **Reptation move**. A bead was cut at either chain end and regrown at the other end, and then the sphere was grown at either chain end without changing its size. [Note the particles (beads or sphere) were randomly placed next to the neighboring bead.] (2) **Crank shaft move**. A bead was chosen randomly and the axis connecting its two neighboring beads was determined. The chosen bead was rotated around this axis randomly, and the fluctuating sphere was regrown on either end without changing its size. A reptation move or a crank-shaft move was discarded if the chosen bead or the sphere overlapped with other beads; otherwise the new conformation was accepted. (3) **Vibration move**. The sphere was allowed to change its size while maintaining its contact with the attached end bead. The sphere would swell or shrink along the axis connecting the sphere and the attached end bead. The variation of the sphere size was randomly chosen in the range between $-\delta$ and δ, where δ was

discarded if the size of the sphere became negative or the sphere overlapped with other beads. If the overlap does not occur, the trial move was accepted based on the Metropolis[62] acceptance probability Min{1, exp(-β($V(\sigma_f)$-(σ_i)))}, where σ_i and σ_f were the sphere radius before and after a Vibration move; V was the vibrational energy defined in Eq. (2.2). The algorithms were tested by varying the vibration step size δ and the weighing probabilities (P_1, P_2, and P_3), and the results were found independent of the choices of these parameters.

A total of 5×10^7 moves were carried out for each simulation, and 20 moves were skipped before a new equilibrated conformation and sphere size was called for. The simulated results were obtained by averaging over 2.5×10^6 equilibrated conformations and sphere sizes. The acceptance ratio of the simulations ranged from 40 to 60% depending on the chain length and sphere characteristics. The chain conformation and the sphere size were equilibrated simultaneously using the combination algorithm.

2.4 A Dimer Grafted onto a Fluctuating Sphere

The following was the investigation of a simplified model in reasoning the confinement effect arising from a grafted segment. **Figure 2.1(b)** displayed the schematic plot of the model with a dimer end-grafted onto a fluctuating sphere (i.e., $N = 2$ in the tangent sphere chain model). The fluctuating sphere was not allowed to overlap with either bead on the dimer, thus the sphere was subject to the geometric constraints due to the excluded volume of the two beads. The

angle θ in **Fig. 2.1(b)** depicted the angle between the molecular axis of the dimer and the bond between the fluctuating sphere and attached bead. For a given sphere sizes, the bond connecting the fluctuating sphere and the attached bead could freely move from $\theta = \pi$ to a minimum angle θ_c which was given by

$$\theta_c = \cos^{-1}\left(\frac{l}{b+2\sigma}\right), \tag{2.3}$$

where b was the diameter of the beads on the dimer, and σ was the instantaneous sphere size. Different bond lengths l for the dimer (ranging from l = 0 to d) were investigated. Variation of the bond length allowed for adjustment of the free space between two neighboring monomers, such that the local environment of a groove could be mimicked. The partition function Z of a dimer with a bond length l grafted onto a fluctuating sphere took the following form:

$$Z = \int_0^\infty d\sigma \int_{\theta_c}^{\pi} \sin\theta \exp[-k(\sigma-\sigma_e)^2] d\theta \tag{2.4}$$

The calculated mean sphere size $<\sigma>$ and distribution function of the sphere size $P(\sigma)$ were, respectively, given by

$$<\sigma>=\frac{1}{Z}\int_0^\infty d\sigma\sigma\left(\left[1+\frac{l}{b+2\sigma}\right]\exp[-k(\sigma-\sigma_e)^2]\right) \tag{2.5}$$

and

$$P(\sigma)=\frac{1}{Z}\left\{\left(1+\frac{l}{b+2\sigma}\right)\exp[-k(\sigma-\sigma_e)^2]\right\} \tag{2.6}$$

This model was reduced to the limiting case of a free fluctuating sphere by letting l = 0 and b = 0, therefore

$$\frac{<\sigma>_0}{\sigma_e} = \frac{\exp[-(\sqrt{k}\sigma_e)^2] + (\sqrt{k}\sigma_e)\sqrt{\pi}[1 + erf(\sqrt{k}\sigma_e)]}{(\sqrt{k}\sigma_e)\sqrt{\pi}[1 + erf(\sqrt{k}\sigma_e)]}, \tag{2.7}$$

and

$$\sigma_e P_0(\sigma) = \frac{2(\sqrt{k}\sigma_e)\exp[-k(\sigma - \sigma_e)^2]}{\sqrt{\pi}[1 + erf(\sqrt{k}\sigma_e)]}. \tag{2.8}$$

Eq. (2.7) and (2.8) indicated that the parameters k and σ_e were directly coupled for a free fluctuating sphere, i.e., they were not independent variables. However, such a coupling was weakened while a grafted segment was present. The expressions for the mean sphere size $<\sigma>$ and $P(\sigma)$ were, respectively, given by

$$<\sigma> = \left(<\sigma>_0 + \frac{l+b}{2}\right)\frac{Z_0}{Z} - \frac{b}{2}, \tag{2.9}$$

and

$$P(\sigma) = P_0(x)\frac{Z}{Z_0} + \frac{\sqrt{k}l}{2(B+x)}\frac{\exp(-x^2)}{Z} \tag{2.10}$$

where $Z_0 = \int \exp[-k(\sigma - \sigma_e)^2]d\sigma = [\sqrt{\pi}/(2\sqrt{k})](1 + erf(\sqrt{k}\sigma_e))$,

$B = \sqrt{k}(b + 2\sigma_e)/2$, and $x = \sqrt{k}(\sigma - \sigma_e)$. Eq. (2.10) had implications on the vibrational spectroscopy for a chromophore bound to a segment, and the discussion was given in the Final Note. After some expansion of Eq. (2.4), the partition function Z could be written as

$$Z = \int_0^\infty d\sigma\left(1 + \frac{\exp[-k(\sigma - \sigma_e)^2}{b + 2\sigma}\right) = Z_0 + Z_1(B) + Z_2(a, B) \tag{2.11}$$

$$Z = \int_0^\infty d\sigma \left(1 + \frac{\exp[-k(\sigma - \sigma_e)^2}{b + 2\sigma} \right) = Z_0 + Z_1(B) + Z_2(a, B) \qquad (2.11)$$

where $a = \sqrt{k}\sigma_e$, $Z_1 = \frac{l}{2} \int_B^\infty \frac{1}{x} \frac{\exp(-x^2)}{(1 + \frac{B}{x})} dx$, and $Z_2 = \frac{l}{2B} \int_a^B \frac{\exp(-x^2)}{(1 + \frac{x}{B})} dx$.

The Z_1 and Z_2 could, respectively, be reformulated as follows:

$$\begin{aligned} Z_1(B) = & \frac{l}{2} \sum_{n=1,3,5,\ldots}^{\infty} B^{n-1} \left\{ \frac{E_1(B^2)}{2n \times (n-1)!} + \sum_{m=0}^{\infty} \frac{(n-m-1)}{2n \times (n-1)!} \exp(-B^2) \right\} \\ & - \frac{l}{2} \sum_{n=2,4,6,\ldots}^{\infty} B^{n-1} \left\{ \begin{array}{l} \frac{(-1)^n 2^{n-1} B^{2n-1} \sqrt{\pi}}{(2n-1)!!} \times [1 - erf(B)] + \\ \exp(-B^2) \sum_{m=0}^{\infty} \frac{(-1)^m 2^m B^{2m}}{(2n-1)(2n-3)\cdots(2n-2m-1)} \end{array} \right\} \end{aligned} \qquad (2.12)$$

and

$$Z_2(a, B) = \frac{l}{4B} \left\{ \begin{array}{l} \sum_{n=1,2,\ldots}^{\infty} \left[\frac{-B^{n-1} \exp(-B^2) + (-a)^{n-1} \exp(-a^2)}{B^n} \right] + \\ \sum_{n=2,4,6,\ldots}^{\infty} \frac{(n-1)!!}{(\sqrt{2}B)^n} \left[(erf(a) + erf(B)) \sqrt{\pi} - \sum_{m=1}^{n/2-1} \frac{2^m (B^{2m-1} \exp(B^2) + a^{2m-1} \exp(a^2))}{(2m-1)!!} \right] + \\ \sum_{n=1,3,5,\ldots}^{\infty} \frac{\left(\frac{n-1}{2}\right)!}{B^n} \left[(-\exp(-B^2) + \exp(-a^2)) - \sum_{m=1}^{(n-3)/2} \frac{(B^{2m} \exp(-B^2) + a^{2m} \exp(-a^2))}{m!} \right] \end{array} \right\} \qquad (2.13)$$

Although the direct numerical evaluation of the partition function Z was more convenient than its expansion, this analysis disclosed the role of sphere and the characteristics of a grafted segment on the properties of a fluctuating sphere. In contrast to the free fluctuating sphere, the properties of the sphere [$<\sigma>$ and $P(\sigma)$] were not the function of the coupled variable $\sqrt{k}\sigma_e$ alone if the grafted segment was incorporated. The force constant k was also coupled with

the geometry parameters of the grafted segment (i.e., bond length l and the monomer length b of a dimer). Therefore, the force constant k was selected as the primary variable in the calculations of the sphere properties.

2.5 Results and Discussion

2.5.1 Effect of force constant on mean sphere size

The properties of the sphere were determined by its intrinsic characteristics and the grafted segment. **Fig. 2.2** depicts the variation of the average sphere size with the force constant, k, for different chain lengths, as marked, for the intrinsic sphere size $\sigma_e = 10$, where the curves for $N = 0$ and 2 (denoted by dashed lines) were obtained from Eqs. (2.7) and (2.9), respectively; the simulation results were denoted by symbols, and solid lines were meant to guide our eyes. For a free sphere without a tethered chain ($N = 0$), the mean sphere size $<\sigma>$ was a monotonic decreasing function of k. As the sphere was attached to a polymer molecule, the $<\sigma>$ showed a non-monotonic behavior for chain length $N > 1$, i.e., as k was increased, the average sphere size decreased first, and then increased again after passing a minimum. When the chain length, was increased, the average sphere size became smaller, and the minimum moved toward smaller k. In **Fig. 2.2(b)**, the mean sphere size $<\sigma>$ was plotted as a function of k for different intrinsic sizes σ_e, as marked, for $N = 21$. The mean sphere size increased, and the minimum shifted slightly from $k \approx 0.07$ to 0.03

when σ_e was increased from 6 to 10. When chains were long enough, the $<\sigma>$ collapsed into same value for larger k (data not shown). The results suggested the tethered chain had a significant effect on the mean size of the fluctuating sphere of smaller force constants.

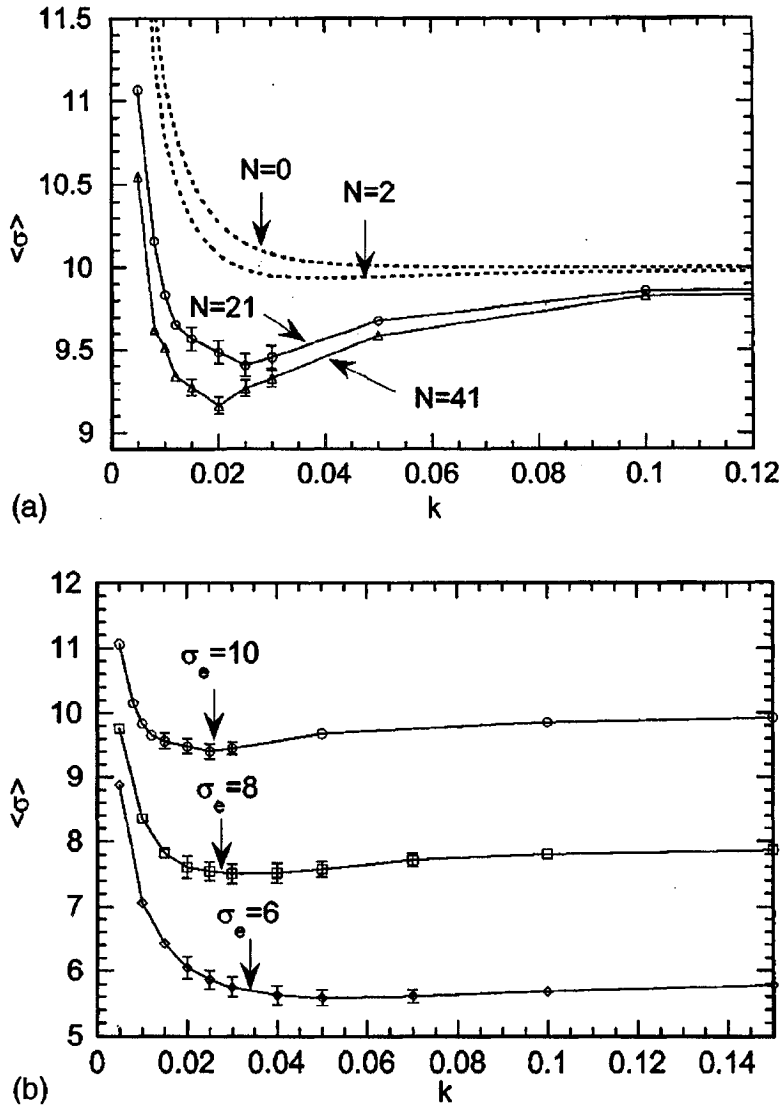


Fig. 2.2 Comparison of the reduced mean sphere size $<\sigma>$ as a function of k for different chain lengths, as marked, for $\sigma_e = 10$ in (a), and for different intrinsic sphere sizes, as marked, for $N = 21$ in (b), where symbols and dashed lines

denote the simulation results and theoretical calculations, respectively, and solid lines were meant to guide our eyes.

2.5.2 Distribution of sphere size

The abovementioned nonmonotonic behavior of mean sphere size was dependent on the surroundings of the fluctuating sphere, and could be reasoned from the sphere size distribution function. In **Fig. 2.3**, the distribution function were plotted for two different force constants, k = 0.01 (denoted by solid line and open symbols) and 0.05 (denoted by dashed line and solid symbols) for different chain lengths N for σ_e = 10, as marked, in which solid and dashed lines were obtained from Eq. (2.6). For k = 0.05, the size fluctuation of the sphere was less pronounced such that the sphere size showed a broad symmetric distribution near the intrinsic sphere size, and the distribution function was not sensitive to the chain length. When the force constant was decreased to k = 0.01, the distribution of sphere size became bimodal, i.e., one maximum appeared near σ = 0, and the other near the intrinsic sphere size σ_e .The population of the distribution near σ = 0 was enhanced as the chain length was increased, which may be ascribed to the confinement effect of a coiled chain that restrained the sphere from expansion. The results indicated the fluctuating sphere was either confined within the chain coil, or freely vibrate outside. However, such a confinement effect did not account for the bimodal distribution of the dimer, N = 2, in which the grafted segment was too short to

form a coil. This might be caused by an additional confinement mechanism at a smaller length scale.

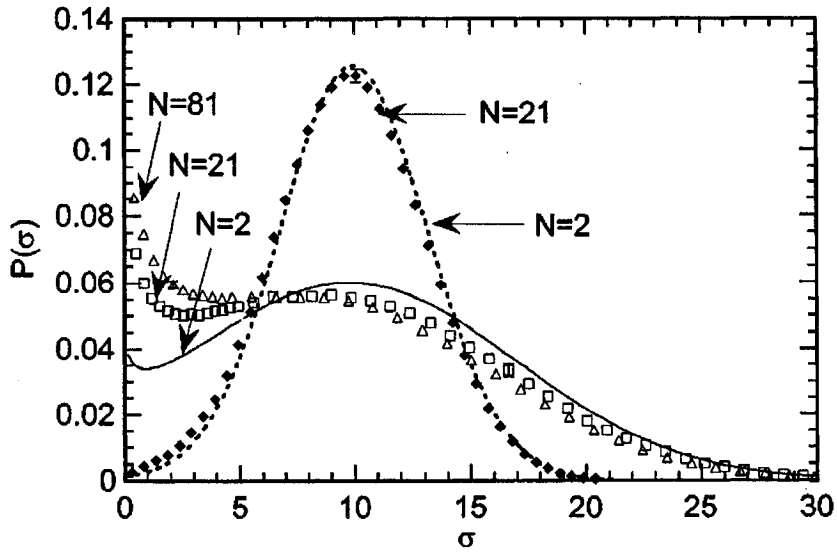


Fig. 2.3 The distribution function $P(\sigma)$ of sphere size for different chain lengths, as marked, for $\sigma_e = 10$ and for $k = 0.01$ (denoted by open symbols and a solid line) and $k = 0.05$ (denoted by solid symbols and a dashed line), where solid and dashed lines denote theoretical calculations.

2.5.3 Local confinement effect

To explore the additional confinement effect, a simplified model composed of a fluctuating sphere and a grafted dimer was used. The local structure of the dimer was varied to change the geometry of the grafted segment on the fluctuating sphere. **Fig 2.4(a)** depicted the average sphere size as a function of k for different bond lengths l ranging from $l = 0$ to d, as marked, for

σ_e = 10. The average sphere size displayed a non-monotonic behavior, i.e., as k was increased, the average sphere size decreased first and then increased again after passing a minimum. In **Fig. 2.4(b)**, the distribution function of sphere size for different bond lengths was plotted, as marked, for k = 0.01 and σ_e = 10. When the force constant was small enough (i.e., k < 0.02 for σ_e =10), it was found that the bimodal distribution became more pronounced as the bond length was increased, indicating that the sphere could be trapped in the confined space of the groove between the two neighboring beads or freely move outside the groove.

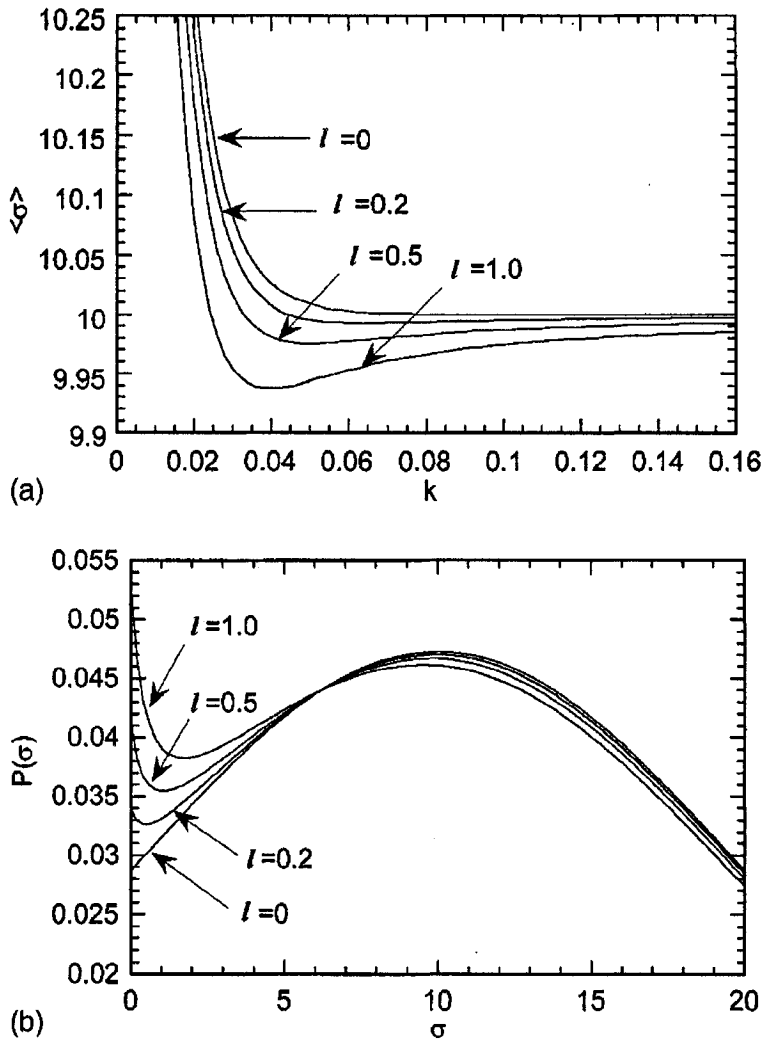


Fig. 2.4 Plot of the mean sphere size in (a) and the distribution of sphere size in (b) for different bond lengths of the dimer, as marked, for $\sigma_e = 10$.

Moreover, in **Fig. 2.5**, the reduced size distribution $[\sigma_e P(\sigma)]$ of the fluctuating sphere with a grafted dimer of bond length $l = 1$ was plotted as a function of $\sqrt{k}\sigma_e$ for $k = 0.01$ (solid lines) and $k = 0.16$ (dashed lines), as marked. For $k = 0.01$, $P(\sigma)$ was a broad bimodal distribution, but this bimodal behavior became less pronounced when the σ_e was decreased. In contrast to $k =$

0.01 and σ_e = 10, the $\sigma_e P(\sigma)$ for k = 0.16 and σ_e = 2.5 (with a same coupling parameter $\sqrt{k}\sigma_e$ = 1 as k = 0.01 and σ_e= 10) displayed appreciable discrepancy due to the additional couplings between the sphere and the grafted segment geometry, as shown in Eq. (2.9). The discrepancy was also observed in $<\sigma>/\sigma_e$ between the fluctuating sphere with and without the bound segment (data not shown). For a free vibrating sphere without the grafted segment, the sphere properties were the function of the coupled variable $\sqrt{k}\sigma_e$, i.e., it was equivalent to varying k or s*e*, as shown in Eqs. (2.7) and (1.8). When a segment, e.g., dimer, was grafted onto the sphere, such a coupling was interfered by the couplings between the sphere and grafted segment, as shown in Eqs. (2.9) and (2.10). These additional couplings became more significant and complex for long chain polymers, and were responsible for the observed behaviors in **Fig. 2(a)** and **2(b)**.

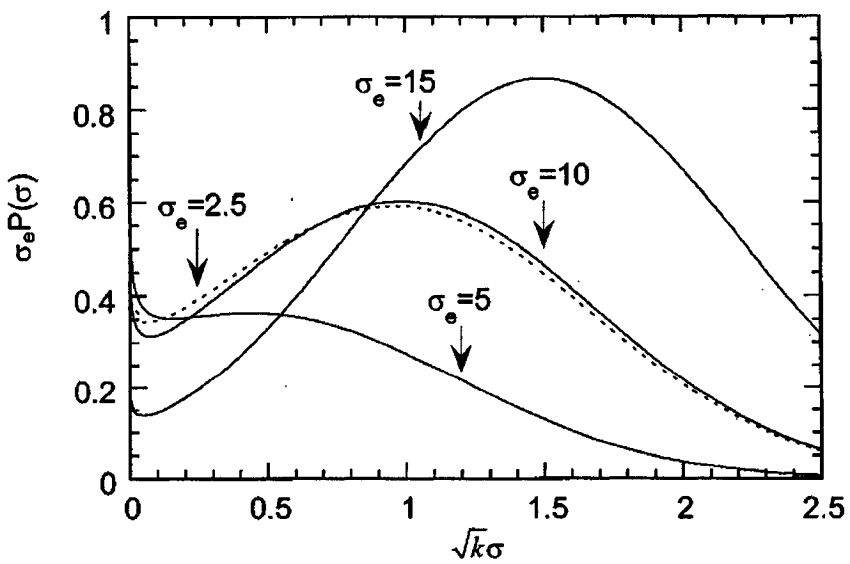


Fig. 2.5 Distribution of the reduced sphere size in the presence of a grafted dimer for $k = 0.01$ (solid lines) and $k = 0.04$ (dashed lines) for different σ_e, as marked.

These results can be understood as follows. The properties of a fluctuating sphere bound to a grafted chain were governed by the interplay of two factors, i.e., characteristics of the sphere itself and the coupling between the sphere and grafted segment. In fact, contraction of the sphere will increase the entropy of the segment because the sphere exerts a smaller spatial constraint on the segment. The coiled chain conformations for long chain polymer molecules play a similar role as the confinement arising from the grooves on the chain backbone, and result in more pronounced confinement, as shown in **Fig. 2.3**.

Although the spheres with very small force constant may be less realistic, from theoretical viewpoints, these limiting cases allowed us to find of the confinement effects induced by a polymer molecule. In the Final Note, some preliminary calculations for the behavior of the vibrational spectroscopy of a labeled chromophore was given to exemplify the influence of the confinement effect.

2.5.4 Extension of the model to a bound flexible segment

To test the feasibility of modeling a bound chain segment as a fluctuating sphere, two different models were compared. In model I, a flexible chain

polymer molecule was divided into a main chain segment and a bound chain segment, and in model II, the bound chain segment in model I was replaced with a fluctuating sphere.

In the simulations, the bound chain segment of chain length N_b = 41 were fixed for model I. The length of main chain segment N_m for the two models was chosen to be the same. The distribution function of the size of the bound chain segment in model I with the fluctuating sphere size in model II was first compared. By adjusting the two variables of the fluctuating sphere, the size distribution of the bound chain segment of model I, characterized by the root mean squared radius of gyration R_g, can be fitted with model II. In **Fig. 2.6**, the distribution function $P(R_g)$ (denoted by solid symbols) was compared with the sphere size distribution of model II $P(\sigma)$ for N_m = 41 for different sets of k and σ_e in model II, as marked (denoted by solid lines). It was found that when σ_e = 4.25 and k = 0.85, the distribution function $P(\sigma)$ displayed the best fit for the entire profile of $P(R_g)$. The small discrepancy at smaller length scales might arise from the lower bound of the chain size in model I (or the lower bound of the sphere size in model II). Nevertheless, the fluctuating sphere model displayed the essential behavior for the size fluctuation of the bound chain segment. Further, different N_m (= 21 and 61) was simulated for the same bound chain segment N_b = 41. The above procedure was repeated to determine the correspondent k and σ_e in model II for each N_m. The k and σ_e obtained from the fit for N_m = 21 and 61 were the same as N_m = 41, and were insensitive to the

chain length of main chain segment N_m (noting for N_m = 0, the fitted k was about the same, but the σ_e was slightly smaller, i.e., the σ_e quickly became leveled off when the grafted chain was long enough).

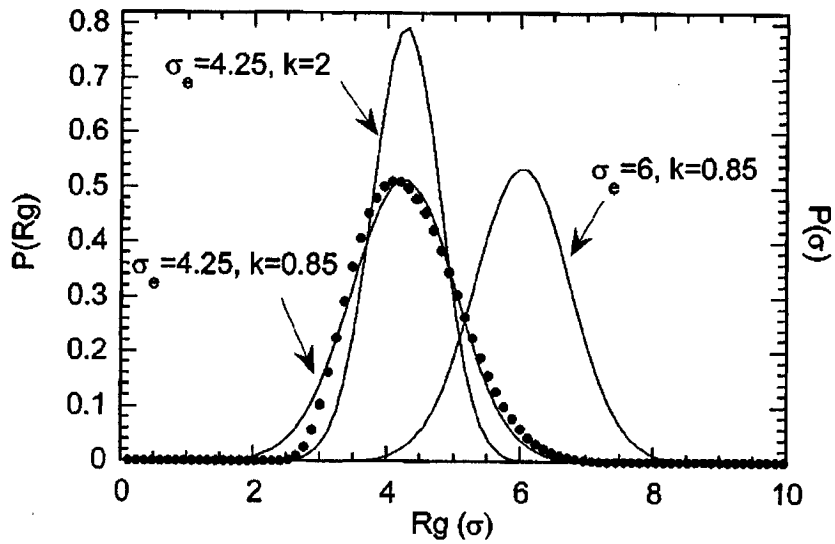


Fig. 2.6 Comparison of the distribution of the size (characterized by R_g) of a bound flexible chain segment of length N_b = 41 in model I (solid symbols) with the distribution of sphere size of model II for different k and σ_e (solid lines), as marked, in the presence of a main chain segment of length N_m = 41.

With these variables, the properties of the main chain segment of model II were computed, and the results were found in good agreement with model I. In **Table 2.1**, the mean chain size between model I and model II were compared, where $< R_g >_b^I$ and $< R_g >_m^I$ were the mean size of bound and main chain segments, respectively, for model I; $< R_g >_m^{II}$ was the mean size of the main chain segment in model II; $< R_g >$ was the mean size of a free polymer chain

(without a bound chain segment or a sphere) whose chain length was equal to N_m in models I and II. In model I, the calculated $< R_g >_b^I$ were insensitive to N_m (for $N_m > 21$), and were close to the mean sphere size of model II (noting $<\sigma>$ = 4.28 ± 0.02, close to σ_e 4.25); so were the mean sizes of the main chain segment for the two models ($< R_g >_m^I$ and $< R_g >_m^{II}$). The main chain segment tended to expand in the presence of a bound sphere or segment, and the extent of expansion for these two models was very close. The chain expansion could be attributed to the interpenetration and excluded volume between two chain segments. These two effects were somewhat equivalent to the contraction and expansion of a vibrating sphere, respectively, thus the simple fluctuating sphere model well represented the interactions due to a bound flexible chain.

Table 2.1 Comparison of chain properties for model I and model II

Bound Seg.	Main Seg.	Model I	Model I	Model II	Free Chain
N_b	N_m	$< R_g >_b^I$	$< R_g >_m^I$	$< R_g >_m^{II}$	$<R_g>$
41	21	4.27 ± 0.01	2.78 ± 0.01	2.80 ± 0.01	2.75 ± 0.01

41	41	4.28 ± 0.02	4.28 ± 0.02	4.31 ± 0.03	4.23 ± 0.02
41	61	4.29 ± 0.02	5.52 ± 0.02	5.53 ± 0.05	5.44 ± 0.02

2.5.5 A homopolymer and a flexible dumbbell model

The dumbbell model was a widely studied model for polymer dynamics due to its simplicity. In the following, the fluctuating sphere model was further extended to coarse-grain a polymer molecule to a dumbbell. A polymer chain composed of two chain segments of same chain length was considered, and the chain length of each segment were chosen to be N_s. In **Fig. 2.7**, the distribution functions $P(R_d)$ (probability of finding the center-of-mass of the two segments at distance R_d) was plotted for N_s = 41 and 61, denoted by open symbols, and the $P(R_d)$ could be fitted by a Gaussian function $\{C\exp[-(K/2)(R_d-\overline{R_d})^2\}$ (denoted by solid lines), where C was the normalization constant; K and $\overline{R_d}$ were the effective spring constant and equilibrium bond length of a flexible dumbbell model. The best fit was obtained when K and $\overline{R_d}$ were equal to 0.11 and 9.6 for N_s = 41, and 0.069 and 12.0 for N_s = 61, respectively. A longer chain became more flexible due a smaller effective spring constant. In addition to the bond length and force constant, the size of the two lobes of the dumbbell could be estimated from the mean size of the two flexible chain segments. These calculations provided a means to parameterize a polymer chain to a flexible dumbbell or a rigid one with an effective bond length, and the two

blobs of the dumbbell could be viewed as two effective rigid spheres or fluctuating spheres.

In terms of dynamics, a polymer chain structure could be simplified to different levels, such as a blob with a hydrodynamic radius, a dumbbell with two blobs, or a bead–spring chain model with many small blobs. Compared to the model by Frankel *et al.* (with one blob),[59] the dumbbell model (with two blobs) incorporated the size fluctuation of a polymer molecule (and its hydrodynamics interactions) in different ways, and was useful, e.g., for the flow field when the dynamics was coupled with the molecular orientation. The investigated model could serve to estimate the blob size of a dumbbell. The chain dynamics predicted by different models can be tested further in the future.

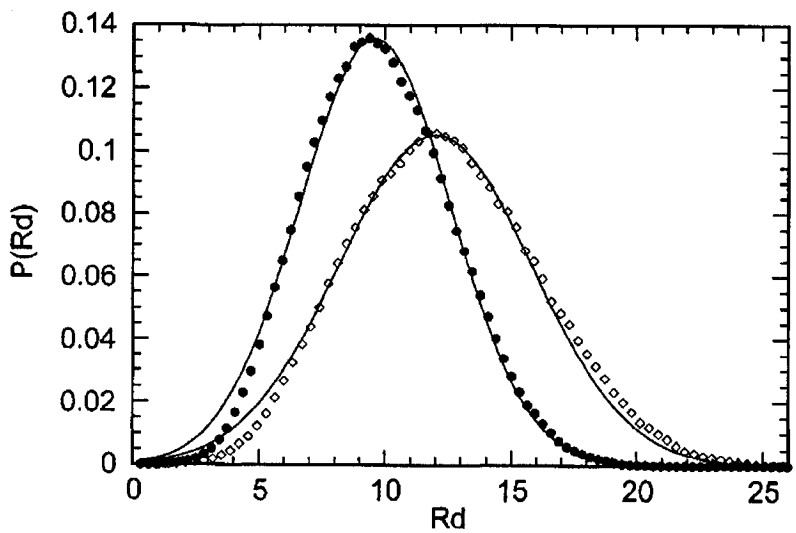


Fig. 2.7 The distance distribution of the center-of-mass of two jointed chain segments of same length N_s for $N_s = 41$ (solid symbols) and 61 (open symbols) where solid lines were the best fit using the Gaussian function.

2.5.6 Effects of the labeled chromophore on the chain conformations

Here the effect of the attached ball on the conformational behavior of the tethered polymer chain was investigated. **Fig 2.8(a)** and **(b)** depicted the variation of the mean squared end-to-end distance for $N = 21$ with intrinsic ball size σ_e for different k and with ball stiffness k for different σ_e, respectively. Similar trends were observed for different chain lengths ranging from $N = 21$ to 101 (only data for $N = 21$ was shown here). As the intrinsic ball size σ_e was increased, the mean squared end-to-end distance $< R^2 >$ increased slowly first with a continuous transition followed, as shown in **Fig. 2.8**. For the small stiffness $k = 0.01$, the chain was more expanded at smaller σ_e, and the transition occurred at the larger σ_e compared to $k = 10$ and ∞. For $k = 10$, the ball was stiff enough, and the corresponding chain size was always close to that of $k = \infty$ for the entire range of σ_e. When the σ_e was small (= 0.1), the chain size was close to that of a free chain without the attached ball, i.e. $< R^2 > \approx 51.6$, for $k = 10$ and ∞. When the σ_e was greater than 10 (20 times greater than the size of a monomer), the expansion of the chain conformation slowed down, and the $< R^2 >$ showed little difference for different k. The chain conformation tended to expand in the presence of the elastic ball, and was sensitive to the ball characteristics.

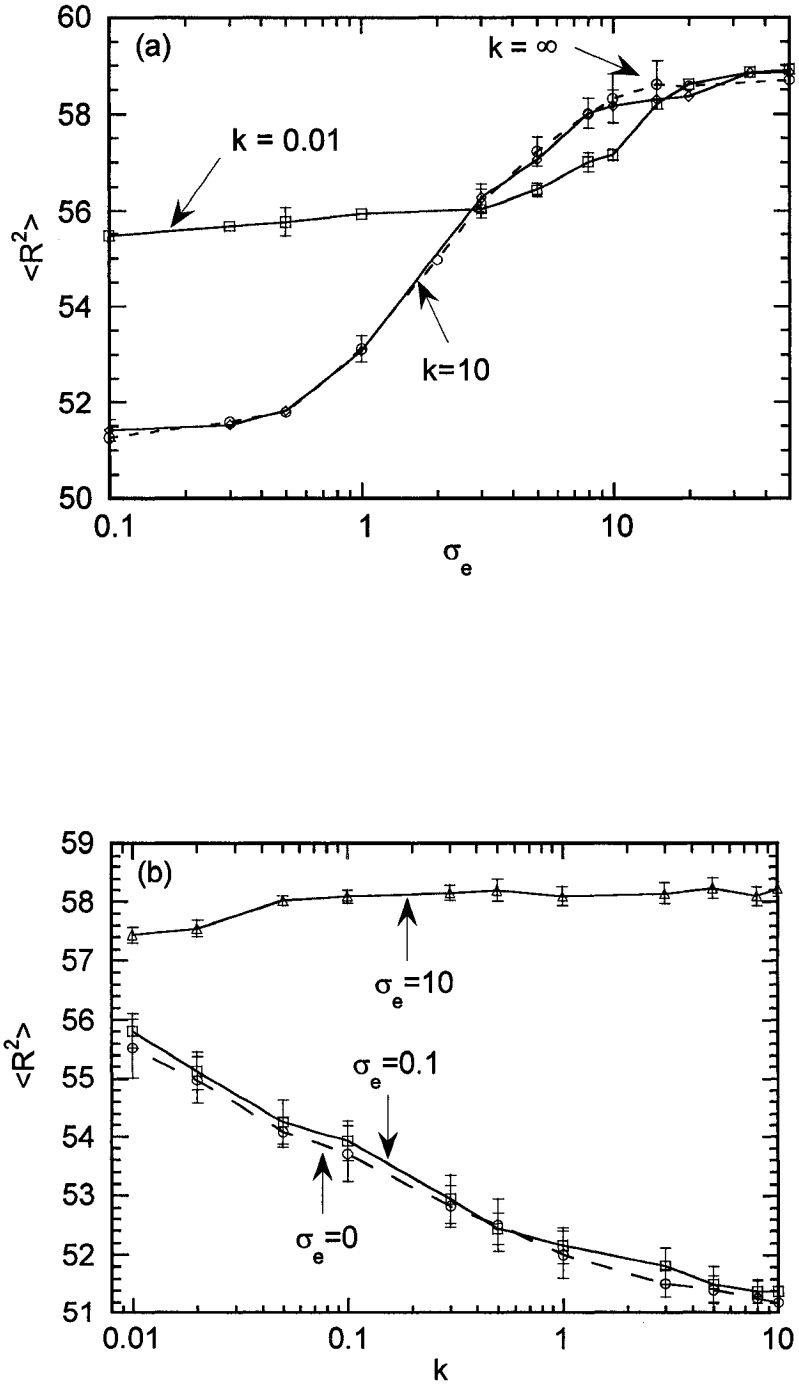


Fig. 2.8 Variation of mean squared end-to-end distance with intrinsic ball size

σ_e for different k in (a), and with ball stiffness k for different σ_e, in (b), for N = 21. Lines were meant to guide our eyes.

These results could be attributed to the exclusion effect of the attached ball and the symmetry of the elastic potential. The elastic ball impeded the approach of the polymer chain near its surface, and resulted in chain expansion. When the intrinsic ball size (or the stiffness of the ball) was very large, the elastic potential was deep and roughly symmetric with respect to the σ_e (except for very small k or σ_e). The shrunk and expanded ball would exclude the chain differently due to their distinct sizes, but these effects were roughly cancelled during vibration. For smaller σ_e (or k), the elastic potential became asymmetric because the imposed constraint prevented the ball size from shrinking to negative values. As a result, the elastic ball was biased on expansion, which leads to the increase of chain size.

The above argument held for most cases, but the exception appeared when the elastic ball had a large σ_e and a small k. In **Fig. 2.8 (b)**, for smaller intrinsic ball sizes such as $\sigma_e = 0$ and 0.1, their mean squared end-to-end distances were similar for the whole range of k, and the chain size decreased continuously as the ball elasticity was increased. For very large k, the chain size approached the limiting value of a free chain without the attached ball, i.e. $< R^2 > \approx 51.2$. This trend was roughly consistent with the above picture, i.e. as the ball stiffness k was increased, the elastic potential became more symmetric, and the exclusion

effects due to ball expansion and shrinkage cancelled out each other. In contrast to $\sigma_e = 0$ and 0.1, a somewhat opposite trend was observed for a large ball like $\sigma_e = 10$. When k was increased, the mean squared end-to-end distance increased slightly, and became leveled off quickly at very small k. Even though the increase was not pronounced, it could be differentiated in the simulations. This result could be attributed to the confinement of the grooves between neighboring monomers or tight chain coils, as discussed earlier sections.

2.5.7 Scaling behaviors of the polymer chain

Fig. 2.9 was the log-log plot of the mean squared end-to-end distance $< R^2 >$ against the chain length for the polymer molecules with and without the attached ball, denoted by open squares and circles, respectively. In the plot, a rigid and large ball was used, i.e. $k = 10$ and intrinsic ball size $\sigma_e = 10$. The attached ball increased the chain size significantly (near 15%), compared to the simulation error about 1% for all chain lengths. The scaling exponents for $< R^2 >$ with N were 1.26 ± 0.01 and 1.27 ± 0.01 for the chain molecules with and without the attached ball, respectively, in **Fig. 2.9**. These two scaling exponents were similar and close to the accurate value, 1.18. Similar results were observed for other conditions ranging from $k = 0.1$ to 10 (data not shown). Note that a large ball like $\sigma_e = 10$, the percent increase of the chain size was always near 15% for different k because the corresponding average ball size $<\sigma>$ was always large. This result indicated the ball tended to expand the chain

size, and the percent increase of the chain size was insensitive to the chain length. These results were encouraging because, in experiments, the labeled chromophore on a polymer molecule may expand the chain conformation, but does not alter the underlying physics of chain conformation.

2.5.8 Influences of attaching position of the elastic ball on polymer chain conformation

The position of the attached ball was of significance for the chain conformation. In the most simplistic way, the chain conformation for the case

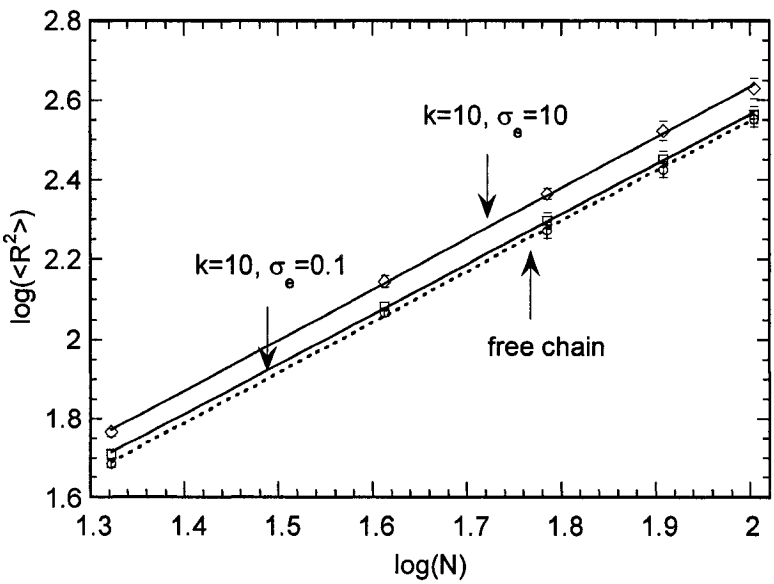


Fig. 2.9 Log-log plot of variation of mean squared end-to-end distance with chain length for free chains and chains attached with very rigid balls.

with the ball labeled at one end of the chain with that labeled on the central bead of the chain was compared, as shown in **Fig. 2.10**. When the ball was large and stiff, the two cases showed significant differences. This could be understood as differences in hard repulsion effects imposed by the labeled ball. When the ball was attached at one chain end, it repelled the other end of the chain. But since the chain was flexible, the other end does not come close to the attached ball very frequently, so the chain expansion was insignificant. However, when the ball was attached at the middle of the chain, both chain ends would experience the hard repulsion from the ball, and as a result, two chain ends approach each other to avoid this repulsion. The degree of chain size shrinkage in this case was more significant compared to that for the chain expansion in the former case, which may arise from the fact that the chain was split into two segments, each having only half the original chain length, so the repulsion from the ball was more severe.

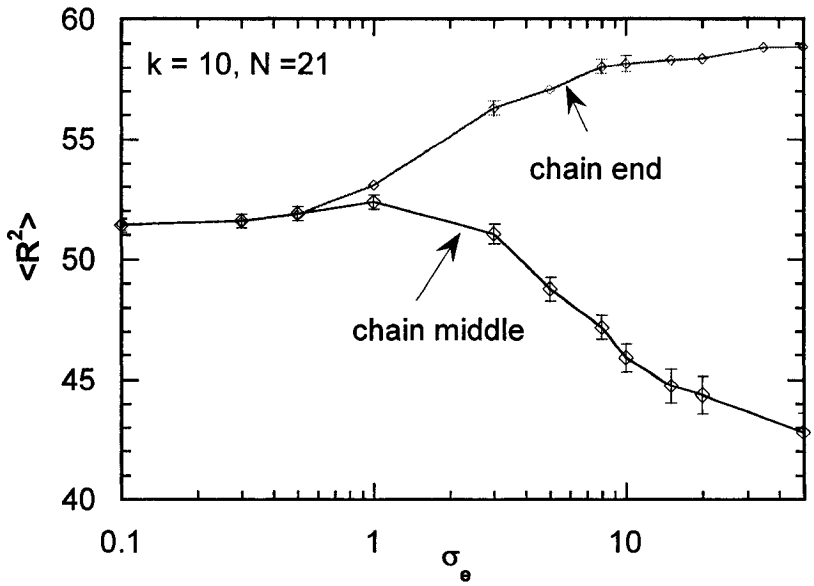


Fig 2.10 Chain size as function of attached ball size for different labeling positions, for $N = 21$ and $k = 10$.

2.6 Conclusions

A simple model composed of a flexible chain polymer grafted onto a fluctuating sphere in good solvents was investigated to mimic a polymer chain labeled by a chromophore or a bound flexible chain segment. The polymer chain was modeled as a tangent sphere chain, and monomers interacted via hardcore repulsions. The sphere size was assumed to remain positive, and the sphere was not allowed to overlap with monomers on the polymer chain. The vibrational energy of the fluctuating sphere took the form similar to the harmonic potential consisting of two parameters: a force constant and an

intrinsic sphere size. The force constant was to account for the vibration motion of the labeled molecule or the restoration forces arising from the conformational entropy of the flexible chain segment. When a bound chain segment was modeled, the intrinsic sphere size was roughly equal to the mean chain size of the bound segment. In this work, two problems related to this model were investigated, including the confinement effect induced by a flexible polymer chain, and the feasibility of treating a bound chain segment as a fluctuating sphere.

Our results showed that when the force constant of the fluctuating sphere was decreased, the average sphere size displayed a non-monotonic behavior, i.e., the mean sphere size decreased first and then increased again after passing a minimum. Meanwhile, the distribution function of instantaneous sphere size can be bimodal. To understand the bimodal distribution, we resorted to a simplified model composed of a fluctuating sphere and a tethered dimer. It was found that the groove in the dimer can exhibit pronounced confinement effect on the fluctuating sphere and results in two distinct regions. The sphere could be either confined in the groove between two neighboring monomers or freely vibrate outside the groove. When the chain length of the grafted polymer molecule was increased, the confinement effect was enhanced due to coiled chain conformations. In other words, the fluctuating sphere was coupled with the geometry of the grafted segment when a grafted segment was present. These findings provided physical insights regarding the confinement effects induced

by a flexible chain polymer, and might be useful in reasoning the role of a DNA molecule on drug-DNA binding,[53,55] or a polymer molecule on the labeled probe.

To further test the feasibility to model a bound chain segment with a fluctuating sphere, the fluctuating sphere model was employed to fit a flexible chain polymer model. The size distribution of the bound chain segment could be fitted with the fluctuating sphere containing a grafted chain. The calculated chain sizes of the main chain segment were found in good agreement between the two models. Moreover, this simple model provided a means to parameterize a polymer chain to a flexible dumbbell. The investigated model was in the level between the Flory's mean field model and the blob model, and could be employed to simplify the chain structure to different levels for polymer dynamics.

The attached fluctuating probe also affected the conformation of the polymer chain. When the probe was attached at one chain end, it expanded the polymer chain. The extent of the expansion depended on the characteristics of the probe, i.e., softer and larger probes tended to have more significant effects. However, the attached probe essentially did not change the scaling law of polymer size with respect to chain length. When the probe was attached at central bead of the polymer, the chain conformation was contracted because both chain ends were repelled from the labeled probe.

The purpose of this study was twofold: (1) parameterization of a flexible

chain polymer to a dumbbell; (2) rigorous test for the conformational behavior of polymers labeled by fluorescence or vibrational probes. Dumbbells have been the widely used model to elucidate the dynamics of polymer chains in experiments. This work has successfully created a robust procedure for parameterization of a flexible chain polymer into a dumbbell. Meanwhile, labeling of chain molecules (such as DNA and proteins) with probes has been one of the well accepted experimental techniques employed to examine the chain properties[46-50]. However, the effects of labeled probes on chain conformational behavior were not clear. Theoretical modeling in this work has validated such an experiment technique because the labeled probe was found not to affect the fundamental physics law for the conformational behavior of a chain molecule.

Final Note: Effect of the bound segment on the vibrational spectroscopy of the chromophore

The fluctuating sphere model can be viewed as a simplified model for a labeled chromophore, and the harmonic-like potential was equivalent to the molecular potential of a chromophore with a single vibrational degree of freedom. The size distribution corresponded to the thermal distribution of the vibrational energy levels in classical mechanics. Here we focus on a chromophore bound to a dimer. After some rearrangement, Eq. (2.10) became

$$\frac{d^2\beta(V+W)}{d\sigma^2}\Big|_{\sigma=\sigma_0} = k - \frac{8l}{(b+2\sigma_e)^3}, \qquad (2.A4)$$

where the second term on the right-hand side was greater or equal to zero. With the grafted segment, the model predicted that the curvature of the effective molecular potential decreased near the minimum. The result indicated that a red shift would be observed in the vibrational spectroscopy of a chromophore when the chromophore was attached to a polymer chain, provided that the force constant of the chromophore was not affected by chemical environments for instance covalent or hydrogen bonding (i.e., same force constant for the chromophore with and without the bound segment). Such a red-shift could be attributed to the confinement effect induced by the grafted segment which essentially produced an effective attractive force on the chromophore.

- Chapter 3 -

Theoretical Studies of Conformational Behavior of Chain Molecules Containing Polar Groups and Electric Fields Effects

3.1 Simulations of a Poly(Vinylidene Fluoride) Model in the Absence of Electric Fields

3.1.1 Introduction

Polar bonds were commonly found in a variety of polymer molecules, and play a crucial role on the chain conformation of biological macromolecules, such as proteins consisting of polar groups and/or hydrogen bonding.[63] Our understanding of the properties of biopolymers, such as protein folding, was hampered by their heterogeneous compositions and the multiple length-scaled interactions. Unlike most biopolymers, synthetic polymers may have regular chemical compositions. With careful design, some of synthetic polymers can serve as model systems and enable us to elucidate the influences of molecular interactions on chain conformations. From the standpoint of theoretical models, polymers containing strong polar bonds are between the limiting cases of neutral and fully charged ionic polymers. However, the effects of electrostatic dipolar interactions arising from the strong polymer polar bonds on chain conformation have received less attention.

Several recent theoretical and experimental studies have involved chain length dependent conformational behavior, which were related to protein folding problems. In the theoretical studies, short coarse-grained chain models (containing 3 – 30 monomers) were chosen, in which monomers, without detailed chemical structure, interact through simplified potentials such as square-well and Lennard-Jones potentials.[64-66] Since the simulations involving strong attractive interactions were more difficult, such coarse-grained level models could be employed to increase computational efficiency and statistical accuracy, and to provide the fundamental insights into complex polymer systems. Experimentally, short peptides have been designed in finding a critical chain length to form a specific secondary structure.[67] Also, the folding process of a short peptide, consisting of 20 amino acids, has been characterized,[68-70] and would be used to rigorously test theoretical predictions obtained from molecular modeling. Inspired by these studies, here, we utilized Monte Carlo (MC) simulation to investigate a model chain molecule containing polar bonds to address the issues regarding the chain conformational behavior under different environmental conditions, including temperature and dielectric constant, and for different chain lengths.

Among synthetic polymers, poly(vinylidene fluoride) (PVDF) was known to contain pronounced polar bonds, which has a regular chemical structure with methylene and fluoro-methylene groups alternating on the chain backbone. In addition to standard molecular interaction potentials (vibrational

energy, bending energy, nonbonded energy, and torsional energy), the recently developed molecular models consist of empirically modified atomic partial charges to account for strong dipolar interactions in a PVDF molecule.[71,72] In our view, PVDF was an ideal molecule to examine the general conformational behavior of polymer chains containing polar groups.

Recently, two PVDF molecular models have been devised. Karasawa and coworkers first developed molecular modeling through quantum calculations to investigate the properties of crystalline PVDF.[71] Byutner et al have extended this study to develop a more generalized molecular model for PVDF,[72] applicable to different condensed phases, by using high level quantum mechanics calculations along with empirical modifications to improve the accuracy of the force field. To test the force field, they conducted Molecular Dynamics simulations for short chain molecules.[73] The bulk material density and chain characteristic ratio were extracted through extrapolation procedures, and the results by Byutner et al showed good agreement with experimental results for bulk amorphous polymeric materials of 1000 monomers. Despite the success of the model, detailed chain properties, such as chain conformation, have not been investigated.

In this work, the PVDF model Byutner et al to was modified by fixing the bond length and angle. Monte Carlo (MC) simulation was employed to investigate the chain conformation of this simplified model because dipolar interactions were effectively attractive and might cause chain to collapse.

Compared to Molecular Dynamics simulations, MC simulations were a better approach to deal with systems trapped in local energy minima arising from attractive interactions,[74] because more drastic moves can be implemented into MC simulations to force the chain molecule to cross the local energy barriers of metastable conformations.[75-77]

To characterize chain conformational behavior, mean chain size and chain size distribution were the two widely used statistical properties. The latter property has been exploited to identify the complex conformational transition experimentally and theoretically,[78-81] because the distribution function was directly related to the free energy as a function of chain size. With this property, Yamasaki et al found the discontinuous (sudden) conformational transition of DNA.[78] Near the transition point, the chain size distribution made the transition from a single-modal to a bimodal function due to the coexistent globular and coiled conformational state, corresponding to two thermodynamic states.

In this work, atomistic Monte Carlo simulations were carried out to investigate the conformational behavior of a modified PVDF model in which bond lengths and bond angles were fixed. The simulations were conducted at the single chain level, and the condensed phase environment was treated as a dielectric continuum (for instance a single polymer in solvents or a tagged polymer in bulk materials). Such a model with explicit polar bonds and more realistic molecular interactions incorporated was beyond coarse-grained chain

models. In the calculations, two environmental parameters, i.e., temperature and dielectric constant were used to tune the strength of interaction potentials. The temperature was varied to change the strength of overall molecular interactions, while the dielectric constant was varied to adjust the strength of electrostatic interactions. The simulation algorithm was first tested, mean chain size and chain size distribution for different chain lengths were then calculated. The chain length dependent conformational transition was summarized in a schematic phase diagram in terms of temperature and chain length. To further verify the observed conformational transition, the chain size distribution was expanded as a function of chain size and the corresponding conformation energy to correlate the chain size with conformational energy. In addition to temperature and chain length, the effects of polar bonds on chain conformation were drawn from these studies.

3.1.2 Model

In this work, the all-atom force field developed by Byutner et al[72] was chosen to investigate the conformational transition of a PVDF molecule. Since bond length and bond angle were quite rigid in a PVDF polymer molecule, the vibrational potential and bending potential had little influence on chain conformation.[82] The model was modified by assuming bond lengths and bond angles were fixed, and their equilibrium values were listed in **Table 3.1.1**.

Table 3.1.1 Equilibrium bond lengths and angles and partial charges in our MC simulations.

Bond Lengths (Å)		Bond Angles (deg)		Partial Atomic Charges	
C-C	1.534	F-C_F-F	105.27	C_H	-0.5202
C-H	1.085	F-C_F-C_H	107.74	C_F	0.6120
C-F	1.357	C_H-C_F-C_H	118.24	H	0.1807
		H-C_H-H	109.27	F	-0.2266
		H-C_H-C_F	108.45		
		C_F-C_H-C_F	118.24		

With these simplifications, the total conformational energy of a molecule was composed of three types of interactions, including electrostatic dipolar V_{el}, Van der Waals V_{vdW}, and torsional potential V_{tor}, respectively, given by

$$V_{el} = \sum_{i>j} \frac{q_i q_j}{\varepsilon r_{ij}} \tag{3.1.1}$$

$$V_{vdW} = \sum_{i>j} A_{ij} \exp(-B_{ij} r_{ij}) - \frac{C_{ij}}{r_{ij}^6} \tag{3.1.2}$$

$$V_{tor} = \sum_{i,j,k,l} \sum_{n} \frac{1}{2} K(n)_{ijkl} [1 - \cos(n\phi_{ijkl})] \tag{3.1.3}$$

where q_i was the partial atomic charge on atom i, ε was the dielectric constant, and r_{ij} was the distance between atoms i and j. A_{ij}, B_{ij} and C_{ij} were Buckingham exponential parameters for Van der Waals interactions, ϕ was the torsional angle formed by bonded atoms i, j, k and l. V_{el} and V_{vdW} applied to all nonbonded atom pairs that were separated by at least three bonds, and the 1-4 interactions (atoms separated by three bonds) were not scaled specially.

The parameters for the above potentials were obtained from the same force field. Note, that Van der Waals interactions and electrostatic interactions account for instantaneous induced dipole-dipole interactions and permanent dipole-dipole interactions in a molecule, respectively.

In addition to equilibrium bond lengths and bond angles, **Table 3.1.1** also summarized the partial atomic charges of different atoms for the model. In the calculations, the same partial charge q for the carbons of the same type were chosen (i.e. q = -0.5202 for all methylene carbons and 0.6120 for all fluoro-methylene carbons). Each repeated unit (CH_2CF_2) satisfied electroneutrality, and the two chain ends were terminated by -CF_3 (next to –CH_2-) and -CH_3 (next to –CF_2-), respectively. Note, that the large difference of the partial charges between two neighboring carbons indicated that the dipolar interactions within in a PVDF molecule arose from other bonds other than C-F bond alone. To simplify the model further, the partial charges of one of the three hydrogen (and of fluorine atoms) on chain ends were neglected. In fact, other choices to model chain ends were tested, for example, by incorporating partial charges into the H and F atoms of both ends, but the results were not affected by these variations due to insignificant contribution from chain ends.

In the Van der Waals potentials, Byutner et al made use of exponential functions, but noticeably, these exponential functions produced unphysical energy wells at the very small distances. **Fig. 3.1.1** plotted the Van der Waals potentials between two nonbonded carbons, in which a well emerged for r_{ij} <

1.2 Å. Such an unphysical potential well was also observed for the Van der Waals potential between other atoms. Although the chance of crossing the barrier into this energy well was essentially zero in the Molecular Dynamics simulations by Byutner et al,[73] this unphysical well could be frequently sampled in our MC simulations, and would result in the abnormal collapse of chain conformation. To rectify this problem, a cut-off distance on each Van der Waals potential was imposed slightly to the right of the peak of the corresponding original Van der Waals potential to prevent two atoms from falling into this regime. **Table 3.1.2** summarized the cut-off distances of the Van der Waals potentials between different atoms for our modified model. To validate these modifications, different cut-off distances (by up to ± 10% of the chosen cut-off values) were tested, and the simulation results were independent of chosen cut-off values.

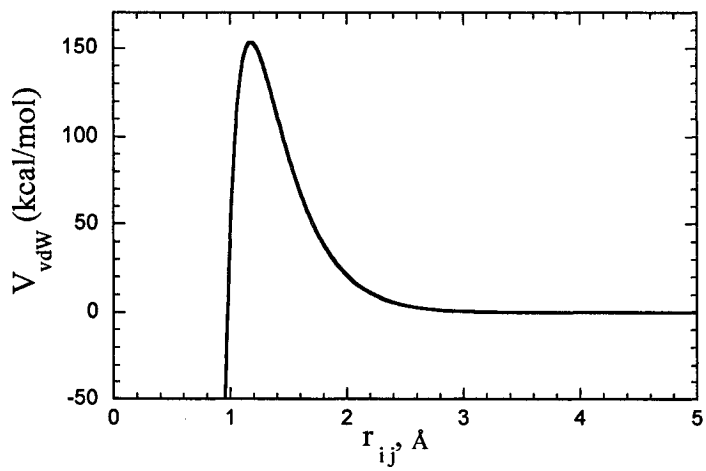


Fig. 3.1.1 Potential function of the original Van der Waals interaction between two nonbonded carbon atoms.

Table 3.1.2 Cut-off distances for the Van der Waals potentials between different atoms in MC simulations.

Atom pair	Cut-off distances (Å)	Atom pair	Cut-off distances (Å)
(C, C)	1.2	(H, H)	1.0
(C, H)	1.2	(H, F)	1.2
(C, F)	1.5	(F, F)	1.5

3.1.3 Monte Carlo simulation

In the MC simulations, chain conformation was sampled by changing torsional angles on the PVDF backbone. For each simulation step, a C-C bond on the PVDF backbone was randomly selected and a new torsional angle randomly between $-\pi$ and π was tried. Such a simulation move was the same as that was used in the RMMC algorithm.[83,84] In the RMMC algorithm, a cut-off was introduced in the calculation of energy, and only those interactions within a few monomers near the bond chosen for rotation were considered. However, in this study, the energy of the entire molecule with all interactions was considered. As in the RMMC algorithm, a trial move was accepted based on the probability P_{accept} of the Metropolis criterion, given by

$$P_{accept} = \min(1, -(E_{new}-E_{old})/k_B T) \qquad (3.1.4)$$

where E_{new} and E_{old} were the energies of the entire molecule for the new and the old conformation, k_B was the Boltzmann constant and T was the temperature.

In the simulations, the chain length was varied from $M = 4$ to 20 where M was the number of repeated units (monomers) of a PVDF molecule to investigate the chain length dependent conformational behaviors. The chosen chain lengths were short compared to those in industrial PVDF materials, but these chain lengths fell into the range of recent experimental and theoretical interests for short chain peptides and polymers.[64-69] Moreover, temperature T was varied between 200 K and 500 K, and the dielectric constant of the medium was chosen in the range between $\varepsilon = 1$ and 8. Note, that in the work of Byutner et al,[73] all atoms were explicitly incorporated in the simulation of bulk PVDF materials and ε was chosen to be 1. The dielectric constant played two roles: 1) to adjust the strength of electrostatic interactions; 2) to represent the condensed phase environment (i.e., dipolar interactions mediated by the medium). For example, for $\varepsilon = 1$, dipolar interactions reached the maximum strength for this model system, and $\varepsilon = 8$ approximated the value of amorphous PVDF material. The chosen temperatures and dielectric constants were based on experimentally relevant conditions.[85]

First, the chain conformation, which was characterized by mean squared end-to-end distance $<R^2>$ of the carbon backbone of a PVDF molecule (between the two end carbon atoms), was computed. A total of 1×10^8 moves were conducted for each simulation, and $2M$ moves were skipped to allow the molecule to relax before a new equilibrated conformation was sampled. Namely, a total of $5 \times 10^7/M$ equilibrated conformations were used to

calculate simulated mean properties. The simulations were tested by calculating the cumulative $<R^2>$ with respect to Monte Carlo steps τ, as well as the autocorrelation function of normalized end-to-end vector (the vector $\boldsymbol{r}_{ee}$ between the two end carbons) $C(\tau)$, defined as

$$C(\tau)=<r_{ee}(\tau)\cdot r_{ee}(0)> \;=\frac{1}{\tau_{max}}\sum_{\tau_0=1}^{\tau_{max}} r_{ee}(\tau+\tau_0)\cdot r_{ee}(\tau_0) \qquad (3.1.5)$$

where τ_{max} was the number of time steps (simulation moves in MC) of ensemble average of each time origins.[86]

In addition to chain conformation, the distribution of the instantaneous end-to-end distance r_{ee} (between the two end carbon atoms) was monitored,

$$P(r_{ee})= H(r_{ee})/\Delta r_{ee} \qquad (3.1.6)$$

where $H(r_{ee})$ was the histogram obtained from simulations and Δr_{ee} was the grid size of histogram. r_{ee} *was* set to be $0.025 \times M$ Å in the calculations. To better understand the effect of interaction energies on chain conformation, the single variable distribution function $P(r_{ee})$ was expanded to a two-variable distribution function of instantaneous end-to-end distance and energy, given by

$$P(r_{ee},E)=H(r_{ee},E)/\Delta r_{ee}/\Delta E \qquad (3.1.7)$$

where ΔE was set to be 0.1 kcal/mol.

3.1.4 Results and discussion

3.1.4.1 Test of algorithms

First, the performance of the simulation algorithm was tested by checking the cumulative average of mean squared end-to-end distance and the autocorrelation function of the end-to-end vectors, as shown in **Fig. 3.1.2**. **Fig. 3.1.2(a)** plotted the cumulative mean squared end-to-end distance as the number of simulation moves (i.e., time τ in Eq. (3.1.5)) for $M = 8$ and 15 for $\varepsilon = 8$, and for comparison, $M = 15$ and $\varepsilon = 1$ were also plotted. Good convergence were obtained for all the cases (different temperatures and dielectric constants), even for the longer chain ($M = 15$), except at the very low temperature (200 K). **Fig. 3.1.2(b)** displayed the autocorrelation function $C(\tau)$ (from Eq. (3.1.5)) for the same parameters as in **Fig. 3.1.2(a)**. Again it was found that the end-to-end vectors were randomized in a sufficiently short time period except for the long chain case ($M = 15$) at the low temperature (200 K). Note that tests for the relaxation of the middle segments were tested, and fast relaxation was observed. In other words, the algorithm offered reasonable performance for chain length up to $M = 15$ and temperature down to near 200 K. (Note that for each simulation, 1×10^8 moves were conducted to achieve good statistics.)

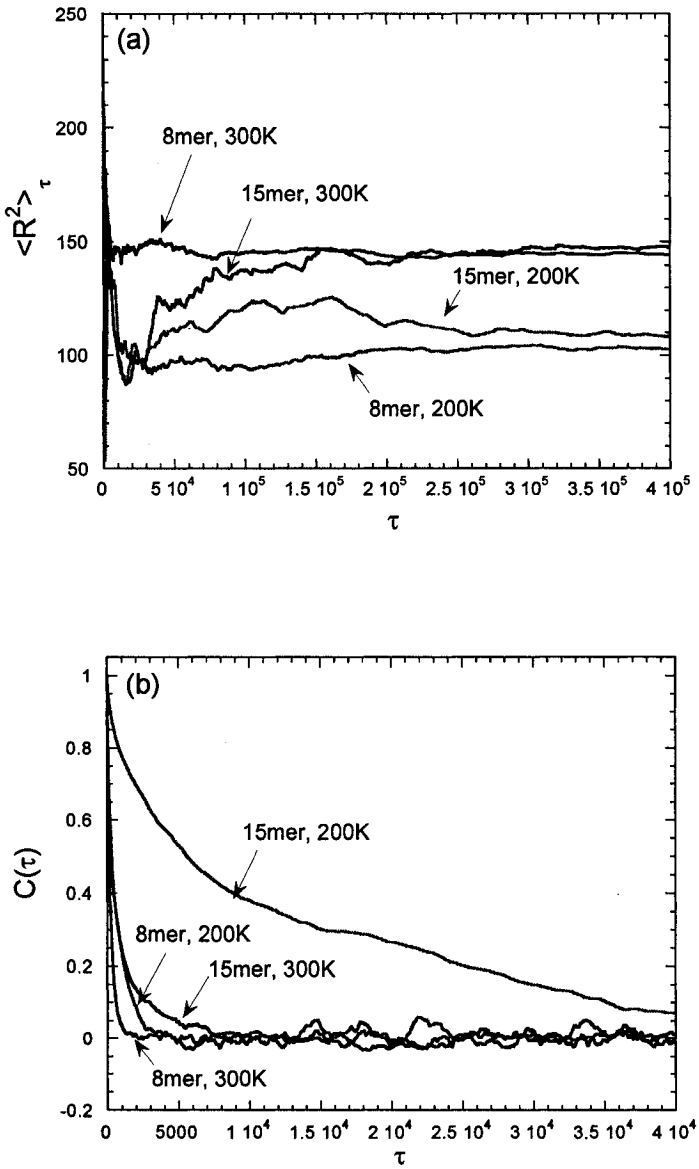


Fig. 3.1.2 Plots of cumulative $<R^2>$ against number of simulation moves τ for 8-mer and 15-mer for different temperatures and $\varepsilon = 8$, (and for 15-mer and $\varepsilon = 1$ for comparison), as marked, in (a) (top), and autocorrelation function $C(\tau)$ (defined in Eq. (3.1.4)) in (b) (bottom) for the same parameters as in (a).

3.1.4.2 Effects of the strength of interactions on mean chain size

The two environmental parameters, temperature and dielectric constant were used in the simulations to fine-tune the strength of the overall interactions and to single out the contribution of dipolar interactions, respectively. **Fig. 3.1.3** plotted the mean chain size $<R^2>$ as a function of temperature for different chain lengths, M = 8 and 12, and for two dielectric constants, ε = 1 and 8, as marked. In all cases, chain size increased with increasing temperature, which suggested that the overall intramolecular interactions of a PVDF chain molecule were attractive. For a given chain length, the chain dimension tended to be more contracted at smaller dielectric constant, which might be attributed to the increased strength of attractive intramolecular dipolar interactions.

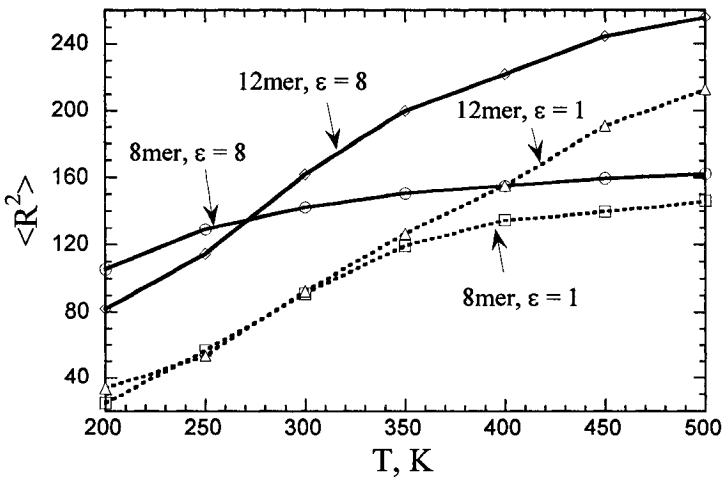
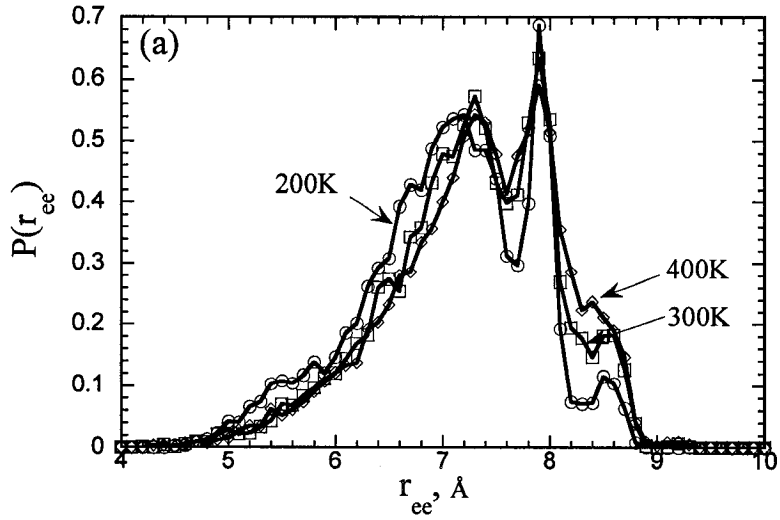


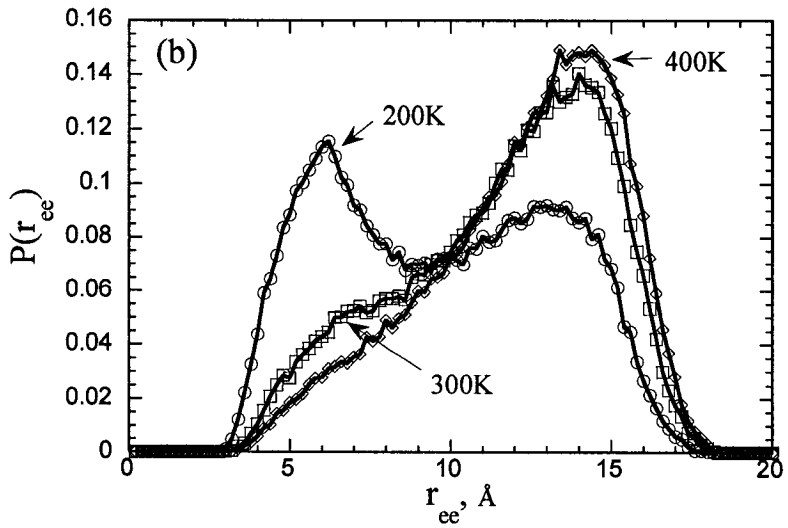
Fig. 3.1.3 Variation of $<R^2>$ with temperature for different dielectric constants ε = 1 and 8, denoted by dotted and solid lines, respectively, and for different chain lengths M = 8 and 12, as marked. Lines were meant for eye guide.

3.1.4.3 Effect of the strength of interactions on chain size distribution

To elucidate the effects of the strength of overall interactions on conformational transition, the size distribution $P(r_{ee})$ for different temperatures was computed. **Fig. 3.1.4** plotted $P(r_{ee})$ for chain lengths $M = 4$ in (a), $M = 8$ in (b) and $M = 15$ in (c), for dielectric constant $\varepsilon = 8$ and for different temperatures (ranging from 200 K to 400 K for $M = 4$ and 8, and from 300 K to 500 K for $M = 15$). When the chain length was short ($M = 4$) in **Fig. 3.1.4(a)**), the chain molecule displayed some features similar to very short conformers, i.e., the $P(r_{ee})$ consisted of multiple peaks and many small bumps for each peak (by noting that simulation errors were about the size of symbols). These distinct peaks and bumps, particularly for the lowest temperature in **Fig. 3.1.4(a)** ($T = 200$ K), could be attributed to the distinct energy difference for different conformations, as seen for very short conformers ($M < 4$). Note that multiple peaks became more pronounced below $T = 200$ K (data not shown). When the temperature was increased from 200 K up to 400 K, the small bumps for each peak diminished, and the third peak near $r_{ee} = 8.4$ Å at 200 K became a shoulder of the peak located at $r_{ee} = 7.9$ Å at 400 K. Namely, the $P(r_{ee})$ was in the form of a bimodal-like distribution (with two primary peaks). As chain length was increased to $M = 8$, the distribution of $P(r_{ee})$ can be a single-modal and bimodal function, but the distribution was quite smooth without small bumps present for all temperatures. At the low enough temperature, $P(r_{ee})$ was a bimodal function,

and this bimodal distribution persisted at the lowest temperature in our simulations for $\varepsilon = 8$. In the bimodal distribution, the two peaks emerged at near $r_{ee} = 6.2$ Å and 13.6 Å, corresponding to the coexistent coiled state and globular state for chain conformations. When the temperature was increased to $T = 300$ K, the peak near $r_{ee} \approx 6.2$ Å diminished, and $P(r_{ee})$ became a single-modal function with a shoulder present near $r_{ee} \approx 6.7$ Å. When the temperature was increased to $T = 400$ K, the $P(r_{ee})$ showed a single peak located near 13.8 Å, due to the predominant coiled state. The transformation of bimodal distribution to single-modal distribution was an indication of discontinuous globule-coil conformational transition, as has been shown in previous theoretical[77,80,87] and experimental[78,79] studies in literature. Thermodynamically, the two conformational states corresponded to one metastable and one stable state, depending on their free energies. At the transition temperature, the free energy of the two states was the same. Furthermore, as the chain length was increased to $M = 15$, the $P(r_{ee})$ was a single-modal distribution function for all simulated temperatures. The peak position moved continuously toward larger r_{ee} as the temperature was increased, suggesting a continuous conformational transition from globular state to coiled state.





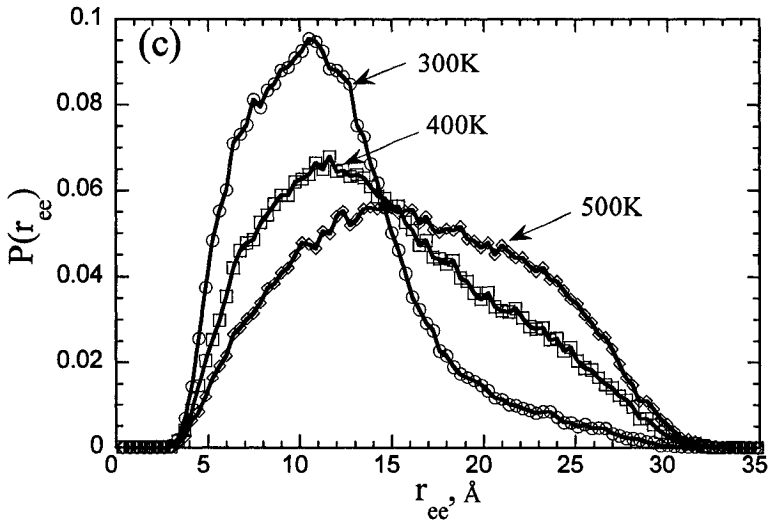


Fig. 3.1.4 Chain size distribution for different chain lengths $M = 4$ in (a), $M = 8$ in (b), and $M = 12$ in (c) for $\varepsilon = 8$ for different temperatures, as marked. Lines were meant for eye guide.

In **Fig. 3.1.5**, the chain size distribution function was plotted for $M = 8$ and $\varepsilon = 1$ for different temperatures, as marked. In contrast to **Fig. 3.1.4(b)** (for $M = 8$ and $\varepsilon = 8$), the distribution function also showed a bimodal profile, but the statistical weight for globular state became greater. When the temperature was high (around 500 K), single-modal distribution was observed, and chain tended to be in the more stretched coil state. For a lower temperature, the distribution became bimodal (with coexistent globular and coiled conformations). As the temperature was decreased further, the bimodal behavior diminishes, and eventually, the single peak emerged, corresponding

to globular conformational state. The results indicated that dipolar interactions induce conformational contraction, and favored globular conformations.

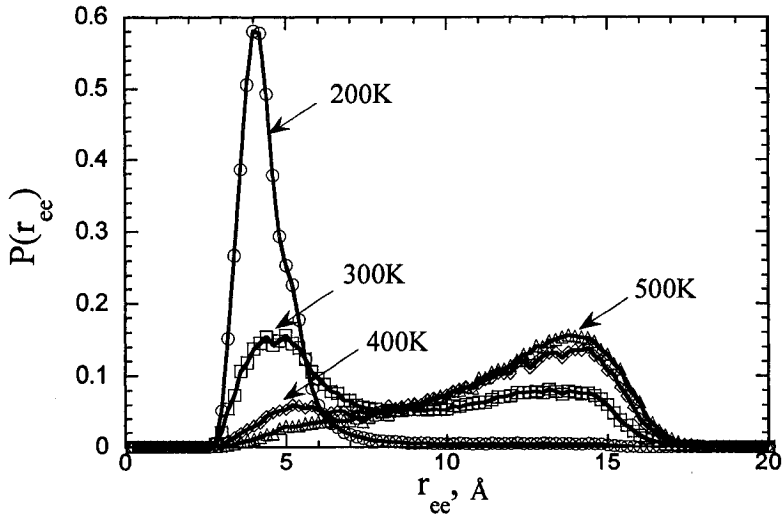


Fig. 3.1.5 Chain size distribution for chain length $M = 8$ and $\varepsilon = 1$ for different temperatures, as marked. Lines were meant for eye guide.

3.1.4.4 Chain length dependence of discontinuous globule-coil conformational transition

Phase behavior of a thermodynamic system could be obtained from its free energy. Although free energy was not calculated directly in our simulations, the simulated distribution function $P(r_{ee})$ was related to the free energy (proportional to $-\ln(P(r_{ee}))$ as a function of chain size. In the following, the transition temperature was defined, from one conformational state to the other, as the temperature at which the two peaks of the bimodal distribution had approximately the same height. In **Fig. 3.1.6** $P(r_{ee})$ was plotted for chain

length $M = 8$, 10 and 12 for $\varepsilon = 8$ in (a) and for $\varepsilon = 1$ in (b) when the two peak heights of the bimodal distribution were about the same. **Fig. 3.1.6(a)** showed that as the chain length was increased, the transition occurred at higher temperatures and the bimodal profile became less pronounced. For long enough chains, the bimodal behavior of $P(r_{ee})$ diminished completely and became a single-modal distribution, as seen in **Fig. 3.1.4(c)**. In **Fig 3.1.6(b)**, it was found that decrease of dielectric constant results in increase of transition temperature, suggesting that the chain molecule was in favor of globular conformational state as dipolar interactions were increased.

3.1.4.5 Effects of dipolar interactions on chain size distribution and conformational transition

To rationalize the influence of dipolar interactions, the dielectric constants were varied to adjust the interaction strength arising from polar bonds. **Fig. 3.1.7** plotted the distribution function $P(r_{ee})$ for $M = 8$ and $T = 300$ K in (a) and for $M = 12$ and $T = 400$ K in (b) for different dielectric constants, as marked. Note that for dielectric constant $\varepsilon > 8$, it was found the $P(r_{ee})$ was almost identical to that of $\varepsilon = 8$; namely the contribution from dipolar interaction became negligible at this temperature. In **Fig. 3.1.7(a)**, for $\varepsilon = 8$, there was only one peak observed in $P(r_{ee})$, corresponding to more stretched coiled chain conformations. As the dielectric constant was decreased, the $P(r_{ee})$ becomes bimodal, indicating that the collapsed globular conformations coexisted with the more stretched coiled conformations. In contrast to $\varepsilon = 2$,

the statistical weight of globular state for $\varepsilon = 1$ increased significantly because of the stronger dipolar interactions for the smaller dielectric constant. In **Fig. 3.1.7(b)**,

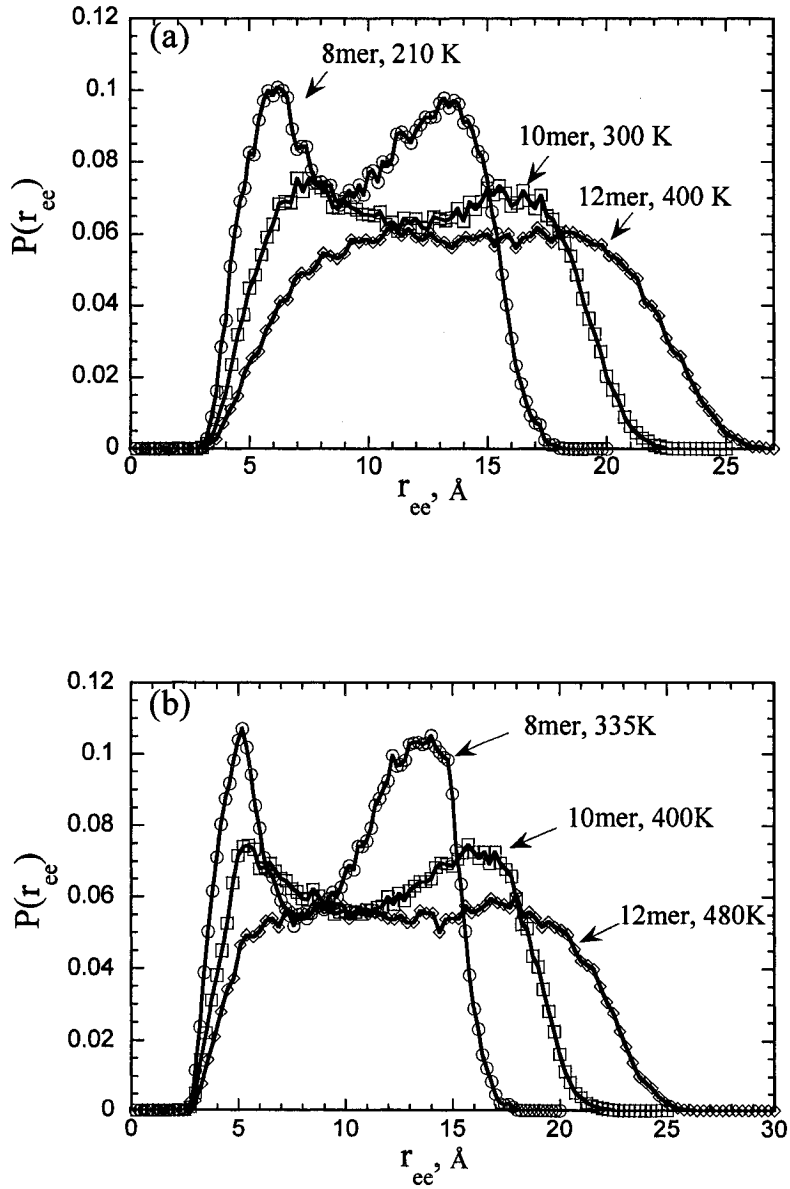


Fig. 3.1.6 Chain size distribution near the transition temperature for different chain lengths, as marked, for $\varepsilon = 8$ and $\varepsilon = 1$. Lines were meant for eye guide.

Similar features were observed for the longer chain M = 12 as dielectric constant was varied. These results showed that dipolar interactions induced chain contraction, and favored more compact globular structures. Increase of dielectric constant, indeed, reduced the contribution from dipolar interactions, and when the dielectric constant was increased to 8, dipolar interactions were greatly reduced with less contribution on the chain conformations. (Note that the observed bimodal behavior also occurred when the monomers interacted through Van der Waals and torsional interactions alone without dipolar interactions.) From the results of different dielectric constants, it could be argued that dipolar interactions enhanced the attractive part of total interactions and collapsed chain conformation further.

3.1.4.6 Schematic phase diagrams

The observed conformational behavior could be summarized in terms of schematic phase diagrams as a function of temperature and chain length, as shown in **Fig. 3.1.8** for $\varepsilon = 8$ in (a) and $\varepsilon = 1$ in (b). The phase diagrams could be roughly divided into three regimes, including short chain, intermediate chain and long chain. (Note that the shaded areas stood for the transition from one regime to the other.) For short chains and long chains, $P(r_{ee})$ exhibited multiple peaks and single peak, respectively. For the intermediate chain length, the dividing lines, denoted by dashed lines, indicated that the transition temperature of the two conformational states. Actually, these findings were analogous to the electronic orbital energy observed in quantum chemistry.

When the number of atoms in a molecule increased, such as in conjugated chain molecules, the energy levels could be in the form of discrete states, band gaps or continuous energy states, depending on chain lengths.[65,66,88]

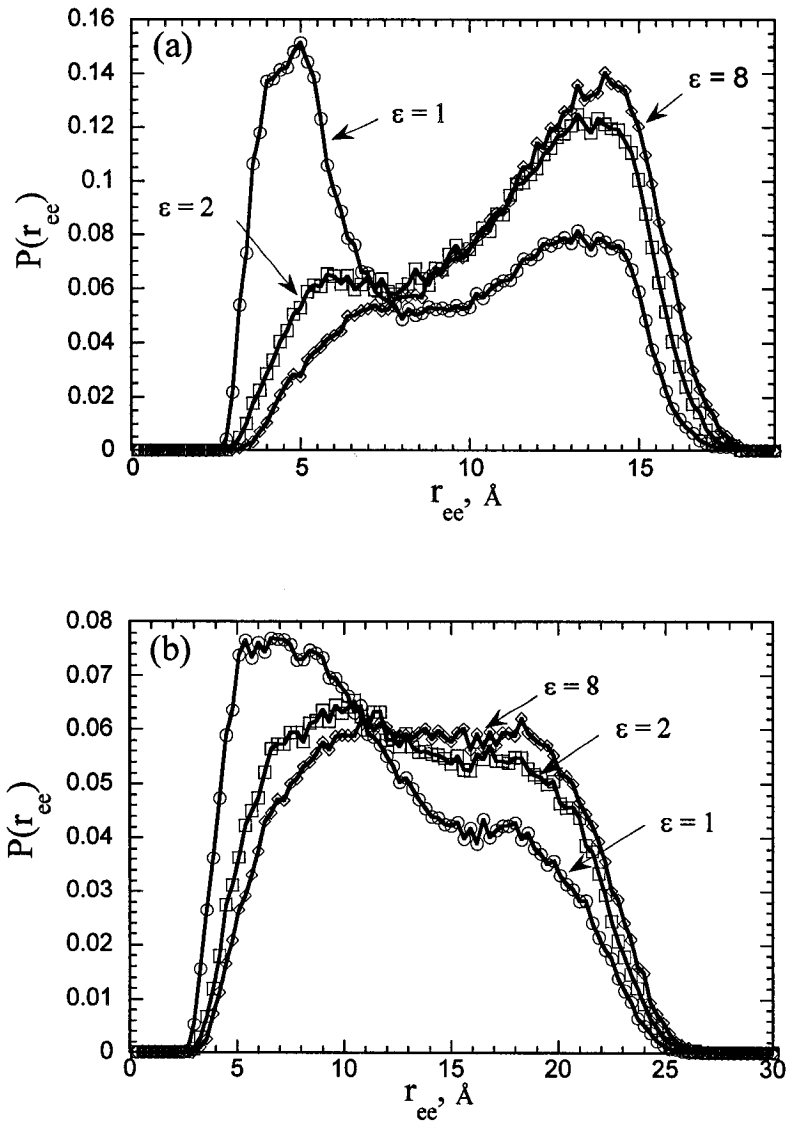


Fig. 3.1.7 Chain size distribution for $M = 8$ and $T = 300$ K in (a) and for $M = 12$ and $T = 400$ K in (b) for different dielectric constants, as marked. Lines were meant for eye guide.

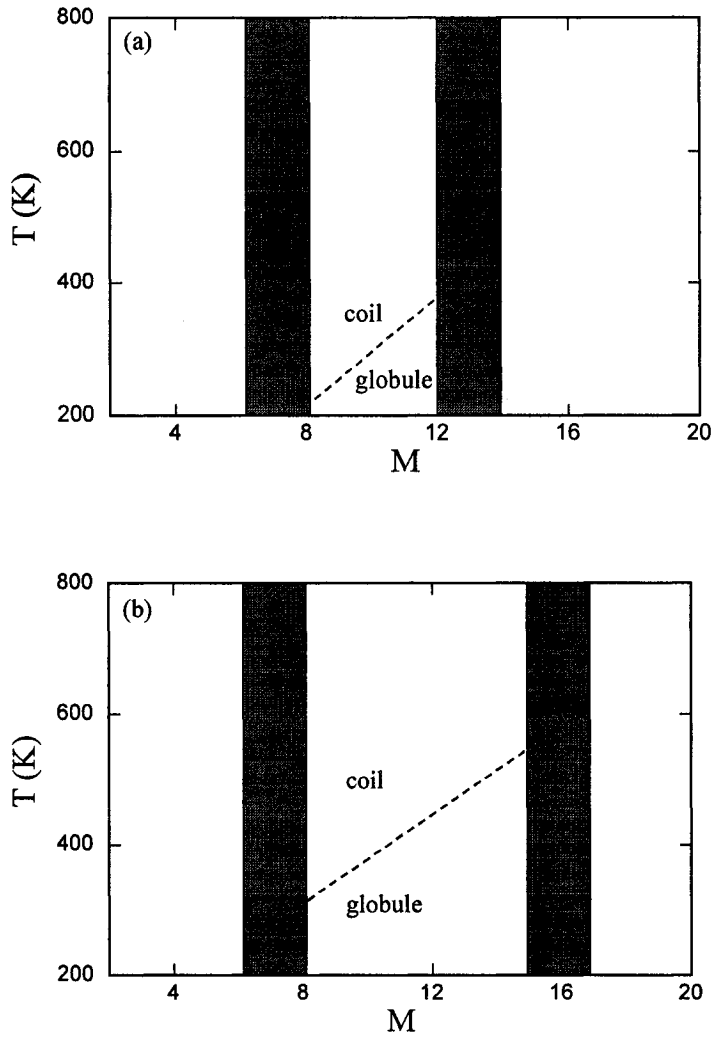


Fig. 3.1.8 Schematic phase diagrams for $\varepsilon = 8$ in (a) and $\varepsilon = 1$ in (b); the transition temperature of discrete conformational transition denoted by dotted curves.

Compared to $\varepsilon = 8$, it was found that the general features of the phase diagram remains similar for $\varepsilon = 1$, and the boundary to separate short chains from intermediate chain length was roughly the same. However, the boundary to divide the intermediate chain length and long chain tended to move toward

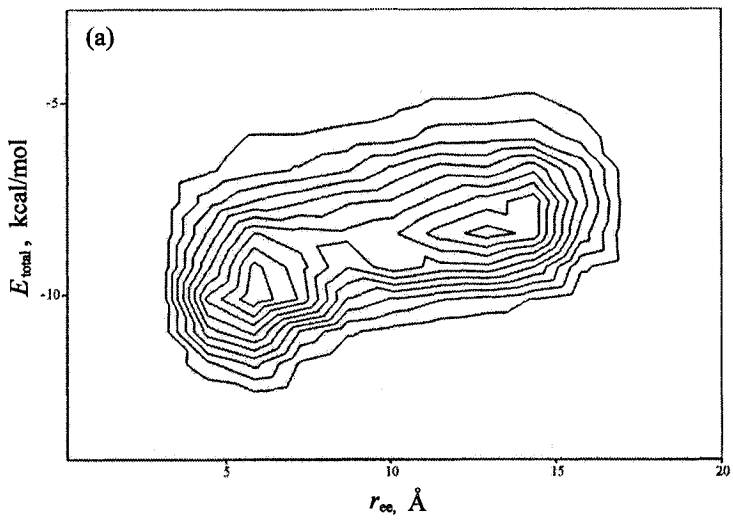
larger chain lengths. Also, for a given chain length, the transition temperature increased when ε was decreased to 1.

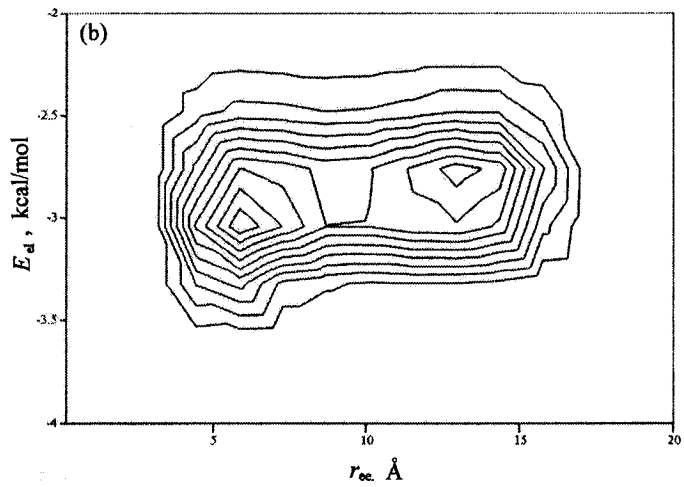
These results might be understood from the perspectives of chain flexibility and effective intramolecular attractive interactions. As pointed out by Khokhlov,[89] the discontinuous conformational transition, as seen for the intermediate chain lengths of PVDF molecules, was sensitive to chain stiffness. For smaller chain lengths, the effective intramolecular attractions were not sufficient to contract a stiff short PVDF chain. For intermediate chain lengths, the chain became slightly more flexible and could be deformed in the presence of attractive interactions. As a result, the chain molecule could either elongate or contract, with different mechanisms to lower the conformational free energy. Namely, the elongated chain molecule gained conformational entropy, and the contracted chain molecule gained internal energy. For longer chains, the chain molecule was effectively more flexible and gave rise to more different conformations; the density states of conformational energy and conformational transition became continuous. From the standpoint of energy, the strong dipolar interactions (at smaller dielectric constants) induced more attraction to stabilize globular conformations at higher temperatures, and counterbalanced the large conformational entropy for longer chains to maintain the distinct globular (energy driven) and coiled (entropy driven) state.

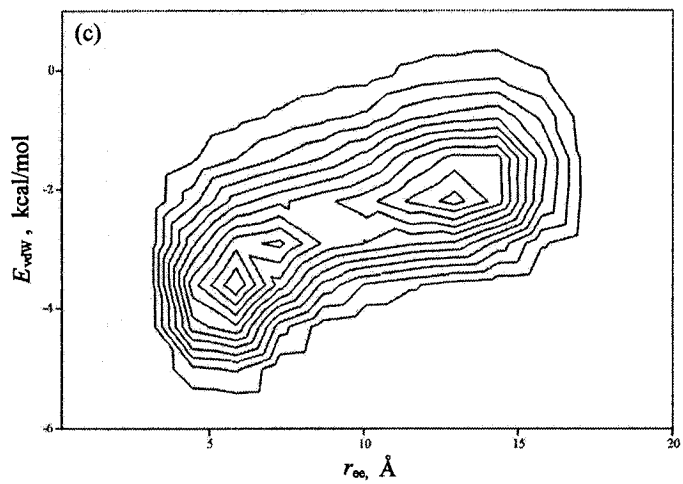
3.1.4.7 More detailed energetic analysis

The PVDF molecule of intermediate chain lengths displayed distinct conformational states, and could be used to discern the effects of different molecular interactions on chain conformations. The chain size distribution $P(r_{ee})$ was first expanded to a function of two variables $P(r_{ee}, E)$, instantaneous chain size (r_{ee}) and energy (E), for such a function depicted the correlation between chain size and interaction energy, and could be used to verify the presence of discrete conformational transition. **Fig. 3.1.9** displayed the contour plots of $P(r_{ee}, E_{total})$ for total energy in (a), $P(r_{ee}, E_{el})$ for electrostatic energy in (b), $P(r_{ee}, E_{VdW})$ for Van der Waals energy in (c), and $P(r_{ee}, E_{tor})$ for torsional energy in (d) for 8-mer for $\varepsilon = 8$ and $T = 200$ K. Note that the reported energies were relative to the energy of the totally stretched zig-zag conformation. The chosen parameters for the calculations in **Fig. 3.1.9** were the same as the curve for $T = 200$ K in **Fig. 3.1.4(b)**. In **Fig. 3.1.9(a)**, $P(r_{ee}, E_{total})$ had the two distinct peaks of approximately same statistical weight, and the energy difference between two peaks was about 2 kcal/mol. These two peaks corresponded to the coiled conformational state of higher energies and the globular conformational state of lower energies. In **Fig. 3.1.9(b)**, the electrostatic energy difference between the two peaks of $P(r_{ee}, E_{el})$ was rather small (only about 0.3 kcal/mol). Meanwhile, the torsional energy difference between two conformational states was negligible, as shown in **Fig. 3.1.9(d)**. However, in **Fig. 3.1.9(c)**, the Van der Waals energy resulted

in a much greater energy difference between two peaks (about 1.7 kcal/mol), which accounted for the primary energy difference between two conformational states. From the contour plots of different types of energy, it was noticed that the energy levels of two distinct conformational states might reveal the nature of interaction potentials. For weak or short-ranged molecular potentials, such as weak dipolar interactions for ε = 8 and short-ranged torsional interactions, the two conformational states in the two distinct peaks had a similar potential energy level. For stronger molecular interactions, such as Van der Waals interactions, the energy level was significantly different between the two conformational states. Despite the fact that dipolar interactions and Van der Waals interactions had comparable length scales, for ε = 8, dipolar interactions were screened out significantly, and Van der Waals interactions were crucial in governing chain conformations.







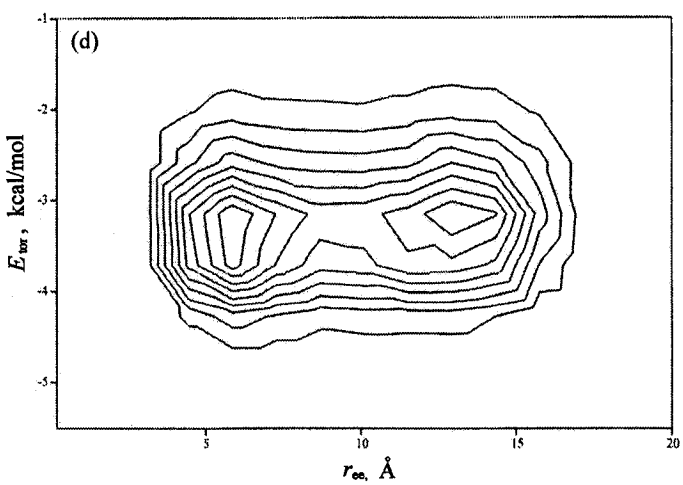


Fig. 3.1.9 Contour plots for $P(r_{ee}, E_{total})$ for total energy in (a), $P(r_{ee}, E_{el})$ for electrostatic energy in (b), $P(r_{ee}, E_{VdW})$ for Van der Waals energy in (c), and $P(r_{ee}, E_{tor})$ for torsional energy in (d) for 8-mer for $\varepsilon = 8$ and $T = 200$ K.

Moreover, the contour plot was also helpful to identify the effects of dipolar interactions. **Fig. 3.1.10** displayed the contour plots of $P(r_{ee}, E_{total})$ in (a) and $P(r_{ee}, E_{el})$ in (b) for 8-mer for $T = 300$ K and $\varepsilon = 1$, by noting that the contour plots of Van der Waals energy and torsional energy were roughly same as those in **Fig. 3.1.9**. The chosen temperature corresponded to **Fig. 3.1.7(a)** with distinct bimodal distribution. In **Fig. 3.1.10**, $P(r_{ee}, E_{total})$ and $P(r_{ee}, E_{el})$ showed similar features, i.e., the energy levels of the two distinct conformational states were quite different. Unlike **Fig. 3.1.9**, the electrostatic interactions in **Fig. 3.1.10** had significant contribution to the conformational energy (the energy difference between two conformational states was 1.8 kcal/mol for electrostatic interactions). Namely, for the smaller dielectric constant $\varepsilon = 1$, the globular conformations were substantially stabilized by electrostatic interactions. Meanwhile, the energy difference of Van der Waals interactions (or torsional interactions) between the two conformational states was insensitive to the presence of dipolar interactions.

3.1.4.8 Representative snapshots of chain conformation

In addition to average characteristic properties, here, a few typical snapshots were presented here to illustrate the molecular structure of chain

conformational states. For illustration purpose, the snapshots obtained from the simulations for $M = 12$ and $T = 400$ K were reported. **Fig. 3.1.11** displayed the snapshots (with carbon backbone sketched only for clarity) for globular conformation for $\varepsilon = 8$ in (a), coiled conformation for $\varepsilon = 8$ in (b), and the globular conformation for $\varepsilon = 1$ in (c). **Fig. 3.1.11 (a)** and **(b)** corresponded to the globular and coiled chain conformation in **Fig. 3.1.7(b)**, in which the chain molecule adopted a partially folded middle segment and a stretched coil, respectively. As for dielectric constant $\varepsilon = 1$, the simulated conformations were not very different from those seen for dielectric constant $\varepsilon = 8$. However, in the simulations, the hairpin-like structures in globular conformations (with more compact loops than (a)) were more frequently encountered, as shown in **Fig. 3.1.11(c)**, due to stronger dipolar interactions.

The PVDF model consisted of the general molecular interactions found in many other polymeric molecules, and could serve as a model system to elucidate the conformational behavior for more complex copolymers or biopolymers. To our best knowledge, the chain conformation of single PVDF molecules has not been systematically studied although the molecule may exhibit significant vibrational-circular dichroism signals due to strong polar bonds.[90] Nevertheless, an increase of interest has been devoted to understanding the chain length dependent single peptide conformation, such as oligomeric N-substituted glycines or "peptoids" reported by Wu et al through CD spectra.[91] Such a peptide was unique because it was believed that

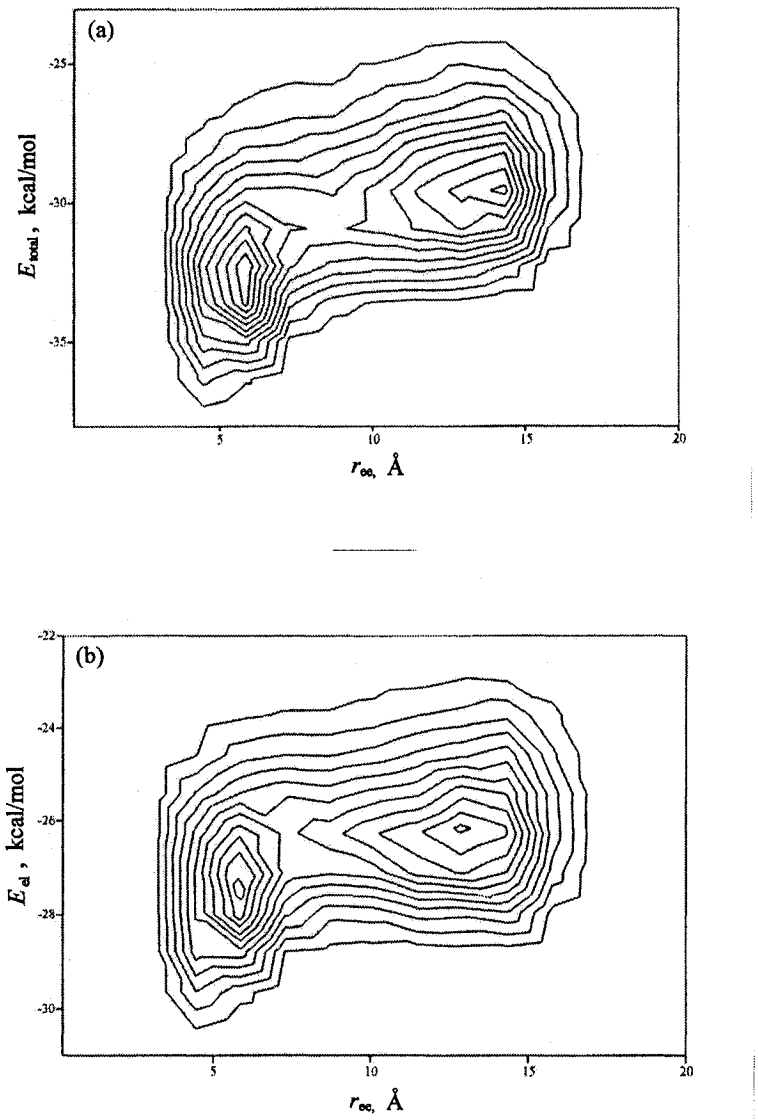


Fig. 3.1.10 Contour plots of $P(r_{ee}, E_{total})$ in (a) and $P(r_{ee}, E_{el})$ in (b) for 8-mer for $T = 300$ K and $\varepsilon = 1$.

hydrogen bonding was not important for its secondary structure. Their findings suggested that the conformation of these peptides was sensitive to chain length. For example, for very short peptides (5 ~ 8 amino acids), the peptide molecules of different chain lengths shared similar spectral

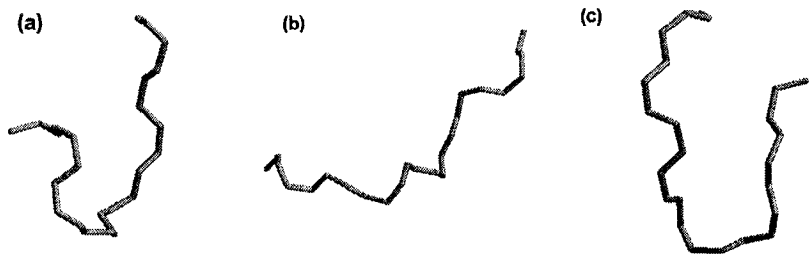


Fig. 3.1.11 Representative snapshots for the globular conformation for 12-mer at 400 K, ε = 8 in (a), the coiled conformation for ε = 8 in (b), and the globular conformation for ε = 1 in (c), with carbon backbone sketched.

characteristics, indicating that their conformations were roughly the same. When the chain length of the peptide was increased to the range between 8 ~ 13 amino acids, the chain conformation was strongly affected by chain length, and these peptides show two major spectral bands of different intensities for these chain lengths. It could be speculated that the chain-length dependent conformational behavior might be caused by coexistent globular and coiled conformational states (observed within the regime near the discrete globule-coil transition).

3.1.5 Conclusions

Atomistic Monte Carlo simulations have been conducted to investigate the conformational behavior of a modified PVDF model in which bond lengths and bond angles were fixed. The simulation was conducted at the single chain level, and the condensed phase environment was treated as a dielectric continuum. The aim of the study was to investigate the roles of dipolar

interactions among different molecular interactions. By varying temperature, the strength of overall interactions was tuned; and by varying dielectric constant, the contribution of dipolar interactions from other types of interactions could be distinguished.

In the simulations, mean chain size and chain size distribution were investigated. For a given chain length, the mean chain size increased as the temperature was increased. However, the chain size distribution was sensitive to the chain length. For short chains, the distribution function exhibited multiple peaks, even for higher temperatures, consistent with conformer behaviors. For intermediate chain lengths, the chain conformation distribution made a transition between single-modal distribution and bimodal distribution as the temperature was changed. Such a behavior could be recognized as discrete conformational transition. To further verify the presence of discrete conformational transition, the distribution was expanded as a function of chain size and energy, and contour plots of these variables were obtained. Such a discrete conformational transition could be identified from the bimodal distribution in the contour plot. For longer chains, the distribution displayed a broad peak and conformational transition became continuous.

Polar bonds (C-C, C-H, C-F) induced dipolar interactions that were effectively attractive. It was found that as dipolar interactions were increased, mean chain size decreased, and the globular conformational state became more pronounced. Meanwhile, the discrete conformational transition for

intermediate chain lengths shifted toward higher temperatures and/or longer chain lengths. From the analysis of different types of interaction energy, it was found that Van der Waals and torsional interactions were responsible for the coexistent conformational states, and the dipolar interactions induced by polar bonds further enhanced the attractive part of molecular interactions and the stability of globular states.

In addition to aforementioned chain properties, snapshots for the simulated conformations were systematically monitored. The globular conformation was characterized by folded middle segment, whereas the coiled conformation was much more stretched out. When the strength of dipolar interactions was increased, a hairpin-like structure with more compact loops was more frequently sampled.

3.2 Conformational Behavior of Polar Polymer Models Under Electric Fields: Investigation of a Simplified PVDF Model and a Polyampholyte Model

3.2.1 Introduction

Polymeric materials with polar or polarizable segments display many interesting properties in the presence of electric fields. A number of experiments have been initiated to probe polymer materials subject to electric fields in the form of thin films, crystals and liquids. A recent neutron scattering experiment has revealed the morphologic change of a copolymer

thin film (PS-co-PMMA) under a strong field.[92] Meanwhile, transformation of the crystalline forms of poly(vinylidene fluoride) in the electric field has been observed.[93] Further, it was known that a spherical droplet of polymer solutions was distorted to be ellipsoid in an electric field.[94] For the immiscible polymer blends such as urethane-modified polypropylene glycol (UPPG)/ dimethylsiloxane, electric fields could induce the phase inversion.[95] In addition, the effect of electric field has been found in natural occurring biopolymers. For example, a DNA molecule end-grafted onto a gold surface could rotate in response to an applied field.[96] One fundamental problem elicited from these experiments was that how the conformation, molecular orientation and structure of polymers were influenced in an electric field.

A great deal of efforts have been devoted to probe the local segment motion of polymers using birefringence and dielectric relaxation.[97-99] All these experiments were designed to measure the response of polymer materials to electric fields. Simplified models such as elastic dumbbells, multi-inverted polar chain, and rotation isomeric state models (RIS) have been developed in reasoning these experiments.[100-102] The primary focus of these studies was on the local motion of polymer segments, but calculations of equilibrium conformation received less attention. Extension of these models to investigate many chain systems was impractical because they were either oversimplified, e.g. dumbbell and lattice models, or complicated by detailed atomic structures. The aim of this work was to find proper ways to simply polymer chain

structure so that its equilibrium properties under external electric fields could be addressed, with the potential to extend to the study of many chain systems.

As discussed in the last part of the chapter, dipolar interactions played important roles in determining the polymer chain conformations. Meanwhile, the charges on the chain molecule that gave rise to dipolar interactions were responsible for deformation (response) of chain molecules under external electric fields. The pattern of charge distribution on the chain was an important factor contributing to the chain response behavior. For a realistic chain molecule, local dipoles undergo frequent fluctuation both in magnitude and orientation, making it necessary to model all charges explicitly. In an external field, however, under the condition that dipole coupling is weak compared to field-dipole interactions and internal thermal motion, one can simplify the chain structure further into what was called polyampholyte, whose conformational behavior has received increasing attention in recent years.[103-109] Polyampholytes are known to have important industrial applications such as modifiers and stabilizers for interfacial properties. Also, polyampholytes have been employed as a model system to mimic more complex biological macromolecules, such as proteins, with heterogeneous chemical structures.[110-112] According to the spatial distribution of the two types of charged monomers, polyampholytes can be divided into random, block, and alternating charged chain molecules.[106,113] Under an external electric field, a polyampholyte molecule undergoes conformational

deformation in response to the applied field, caused by the alignment of intrinsic dipoles arising from opposite charges within polyampholyte molecules.

Theoretical models have been devised to investigate the conformational properties of polymers in presence of electric fields.[107,114-118] Schiessel et al[115] and Winkler et al[118] have solved the freely jointed-chain and Gaussian-chain polyampholyte models, respectively, subject to uniform electric fields in the weak coupling limit, in which the interactions between charged monomers were neglected. These models predicted that the applied fields induce asymmetric conformational deformation, i.e., the chain molecule elongated along the field direction, but contracted in the direction perpendicular to the field. When the strength of the electric field (along the Y-axis) was small, the conformational deformations for the neutral alternating polyampholyte model had analytical solutions as follows.

$$< R^2 >=< R_0^2 > +\alpha_1 q^2 \lambda^2 \qquad (3.2.1)$$

$$< R_z^2 >= \frac{1}{3} < R_0^2 > -\alpha_2 q^2 \lambda^2 \qquad (3.2.2)$$

where $< R^2 >$ was the mean squared end-to-end distance; $< R_z^2 >$ was the z-component of $< R^2 >$; subscript 0 denoted the condition in the absence of electric fields; q was the magnitude of monomer charge; λ was the coupling strength, equal to $E_0 eb/kT$ (E_0 was the strength of electric field strength, e was the unit charge, b was the unit length of 1 Å, k was the Boltzmann constant,

and T was the temperature); coefficients α_1 and α_2 were the functions of chain length, given by

$$\alpha_1 = \beta_1 M^2 \tag{3.2.3}$$

$$\alpha_2 = \beta_2 M \tag{3.2.4}$$

where M was the number of repeated monomer units; β_1 and β_2 were numerical factors depending on the chosen chain model. In fact, the behavior of the y-component of mean chain size determined the overall chain deformation, in which the mean vector sum of $<R_y>$ was simply the square root of the second term of Eq. (3.2.1). However, to our best knowledge, these formalisms have not been tested rigorously by computer simulations.

Of special interest was PVDF, an electroactive polymer, whose chemical structure can be viewed as neutral alternating polyampholytes. PVDF consists of alternating methylene and fluoro-methylene on the backbone, carrying net positive and net negative charges, respectively. At the level of united atom models, the methylene and fluoro-methylene groups were reduced to two different types of charged sites, as the positive and negative charged groups in alternating polyampholytes.

Here, two different models derived from the PVDF model developed by Byutner et al.[72,73] were investigated. In the first model, the alternating methylene and fluoro-methylene groups in a PVDF molecule were collapsed into united atoms of different charges, equivalent to the limiting neutral polyampholyte model with alternating charged monomers on the polymer

backbone. In the second model, all the constituent atoms and their partial charges were treated explicitly. The purpose to investigate these models was threefold: (i) to test analytical theories of Schiessel et al and Winkler et al; (ii) to elucidate the influences of theoretical models on the conformational deformation of polymers in electric fields; (iii) and to explore possible ways to coarse-grain polymer models containing strong dipoles.

3.2.2 Models and Monte Carlo Simulation

In this work, two models were investigated, modified from the PVDF model devised by Byutner et al.[72,73] In model I, the methylene and fluoro-methylene groups were coarse-grained to united atoms, such that the chain carries alternating positive and negative charged sites on the backbone. The charges for positive and negative charged sites were estimated from summing up the partial charges of all constituent atoms of a methylene (-0.159) and of a fluoro-methylene group (0.159), respectively. Model II was an all-atom model with explicit atoms and partial atomic charges obtained from the force fields by Byutner et al.[73] In the two models, the bond lengths (1.534 Å) and bond angles (118.24 degree) were fixed at their equilibrium values, as in the first part of this chapter. The chain conformations were determined by the torsional angles of the carbon backbone alone. In absence of electric fields, the two investigated models were reduced to the free flight chain model with fixed bond lengths and angles. To compare with analytical theories, here, the weak coupling limit (without intramolecular interactions involved) was considered,

and only the interaction, V_E, between the charge sites and the electric field was taken into account. Following the notations from Eqs. (3.2.1 – 3.2.4), V_E, was given by

$$V_E = \sum_{i=1}^{N} \lambda q_i \vec{r}_i \vec{E} \qquad (3.2.5)$$

where $\lambda = E_0 eb/kT$ as defined in Eq. (3.2.1), q_i was the charge of the i-th atom, N was the total number of atoms in a polymer molecule; $\vec{r}_i$ was the position vector of each monomer, $\vec{E}_i$ was the unit vector along the electric field direction.

In the simulations, the pivot algorithm was employed to alter chain conformation in torsional space,[74] in which a C-C bond was chosen randomly as a rotating axis and part of the chain on one side of this bond was rotated by a random angle between -π and π. The acceptance of a move was subject to Metropolis criterion. In the simulations, the mean chain size, characterized by mean squared end-to-end distance $< R^2 >$ and the component perpendicular to the field direction $< R_z^2 >$ were calculated. Also, the orientation of the backbone C-C bond vectors with respect to the external field and its distribution function $P(\theta)$ were computed, where θ was the angle between the vector and field direction.

3.2.3 Results and discussion

First, the simulation results of Model I were compared with the analytical theories by Schiessel et al and Winkler et al,[115,118] **Fig. 3.2.1** plotted the mean

squared end-to-end distance $< R^2 > / b^2$ and its component perpendicular to the field $< R_z^2 > / b^2$ against field strength for different chain lengths $M = 40$, 60, 80 and 100, as marked; lines were obtained from fitting the simulation results with Eq. (3.2.1) and (3.2.2). Excellent fits with analytical theories were obtained for $< R^2 >$ and $< R_z^2 >$ in the range of investigated field strengths. Note that the simulations deviated from quadratic forms when field strength exceeded the maximum value in **Fig. 3.2.1** and **3.2.2**. Furthermore, from **Fig. 3.2.1**, the coefficients α_1 and α_2 defined in Eqs. (3.2.1) and (3.2.2) were extracted. These results were plotted against chain length M in **Fig. 3.2.2**, and were fitted with Eqs. (3.2.3) and (3.2.4). The simulation results agreed well with the analytical theories, and these calculations provided a rigorous test between simulations and theories. Furthermore, this test was also important for our simulations of Model II in which the chain backbone had the same structure as in Model I except that additional methylene and methylene fluoride groups were incorporated.

In the following, the chain properties of $M = 20$ were investigated to elucidate the influences of chain models on the conformational behavior under electric fields. **Fig 3.2.3** displayed the variation of mean chain size $< R^2 > / b^2$ (circles) and its z-component $< R_z^2 > / b^2$ (squares) with the coupling strength for Model I in log-log scale, denoted by open symbols, and Model II, denoted by solid symbols, as marked; lines were obtained from

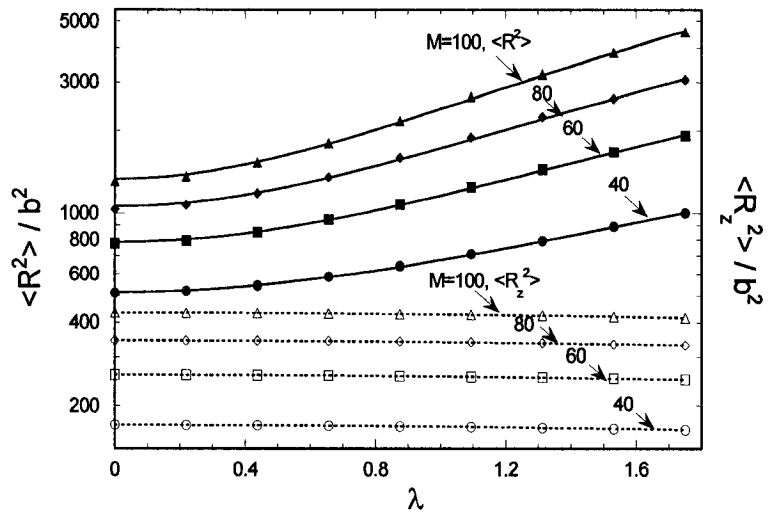


Fig. 3.2.1 Variation of $< R^2 > / b^2$ (denoted by solid symbols) and $< R_z^2 > / b^2$ (denoted by open symbols) with field strength for Model I for different chain lengths, as marked; lines were obtained from fitting the simulation results with Eqs. (3.2.1) and (3.2.2).

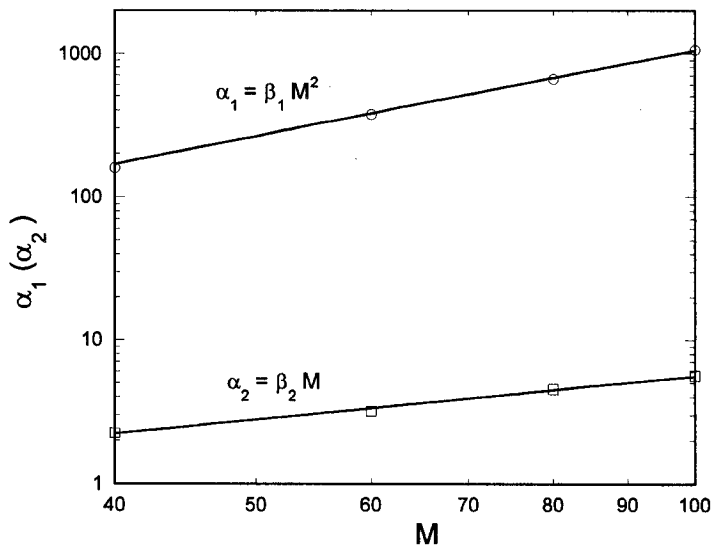


Fig. 3.2.2 Variation of coefficients α_1 and α_2 with chain length M; lines were obtained from Eqs. (3.2.3) and (3.2.4).

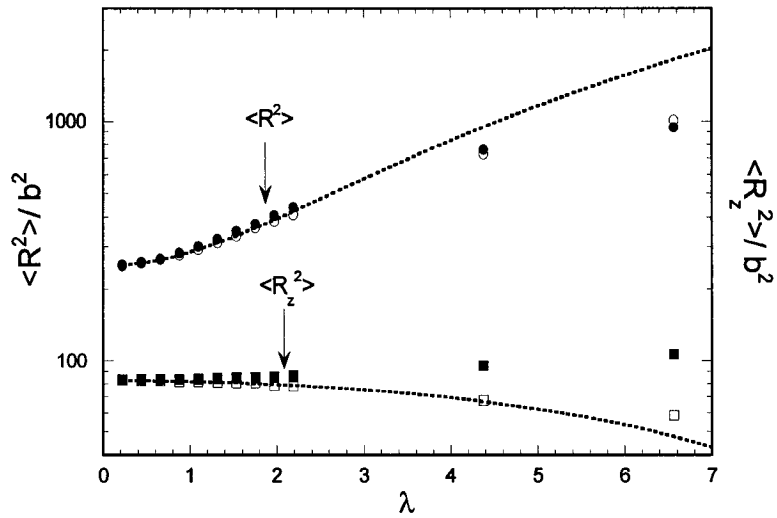


Fig. 3.2.3 Variation of $< R^2 > / b^2$ (circles) and $< R_z^2 > / b^2$ (squares) with coupling strength for Model I in log-log scale, denoted by open symbols, and Model II, denoted by solid symbols; lines were obtained from fitting the simulation results for Model I with Eqs. (3.2.1) and (3.2.2).

both models at low coupling strengths. This good agreement persisted up to around λ = 2.2, suggesting that the effective charge used for Model I was a reasonable choice to reproduce the overall chain size of Model II. This result may be attributed to the spatial average of the charged groups (CH_2 and CF_2) for the all-atom model under frequent internal rotation. However, it was noticeable that when λ became greater than 2.2, the mean chain size for both models deviated from the analytical theories. Despite the good agreement between the two models at smaller λ, these models at larger λ displayed different conformational behaviors. As λ was increased, $< R_z^2 >$ decreased

for Model I, but increased for Model II. Such a difference was associated with the local chemical structures of the two models.

To elucidate the effects of the local chemical structure of a chain model, the distribution of the orientation of C-C vectors for odd (positive charge pointing to negative charge) and even bonds (negative to positive) on the chain backbone were investigated. **Fig. 3.2.4** displayed the angular distribution of the unit vector $P(\theta)$ of the odd bonds and even bonds, as marked, of Model I for $\lambda = 1.09$ and 6.54; the inset illustrated the definition of bond vectors. At weak fields ($\lambda = 1.09$), the probability of odd bond vectors and even bond vectors orienting toward the field direction was roughly the same. However, for a stronger field, ($\lambda = 6.54$), the peak of $P(\theta)$ of the odd bonds appeared near 0 degree, but for the even C-C bonds, the peak was located at near 62 degree, close to the angle between the neighboring odd bond vector and even bond vector due to the bond angle constraint. As pointed out by Schiessel et al and Winkler et al,[115,118] in a neutral alternating polyampholyte as Model I, the total dipole of a chain molecule could be viewed as a series of dipoles due to odd bonds. The odd bonds aligned along the field direction, whereas the even bonds did not contribute to the total dipole moment but just maintaining chain connectivity. (Hence, an even bond can freely rotate around a neighboring odd bond.) In other words, the field induced chain elongation of Model I can be attributed to the alignment of odd bonds toward the field direction, and such an elongation suppressed the bond

fluctuation in the direction perpendicular to the field so that $< R_z^2 >$ decreased as λ was increased.

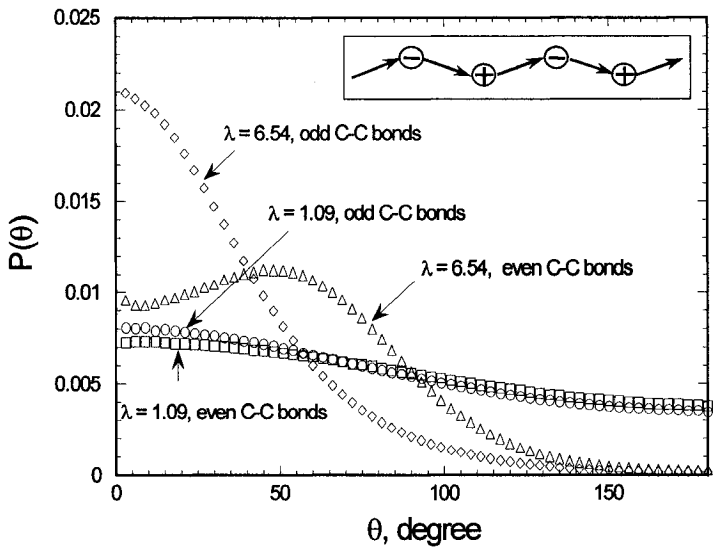


Fig. 3.2.4 Angular distribution of the unit vector $P(\theta)$ of odd $C-C$ bonds and even $C-C$ bonds, as marked, for Model I for λ = 1.09 and 6.54; the inset illustrated the definition of bond vectors.

In **Fig. 3.2.5**, the angular distribution of the odd and even C-C bonds for Model II was presented, with same coupling strengths as in **Fig. 3.2.4**; the inset defines bond vectors. For odd bonds, the vector was sketched from CH_2 to CF_2 within a monomer, and for even bonds, the vector was from CF_2 to CH_2. For the smaller field strength (λ = 1.09), the odd and even bonds tended to orient toward the field direction. In contrast to Model I, a significant difference for the orientation between odd and even bonds in Model II was

observed even for the weak coupling strength. Such a result arose from the net dipole of a monomer. In Model I, the net dipole was in the direction of the C-C bond, and its magnitude was 1.29 Debye. However, in Model II, the net dipole of a CH_2-CF_2 monomer was around 97 degree tilted from the C-C bond, provided the CH_2-CF_2 conformer was in its trans position, with a dipole moment roughly twice as large as that of Model I. Hence, the monomers (odd-bonds) of Model II responded to electric fields more strongly.

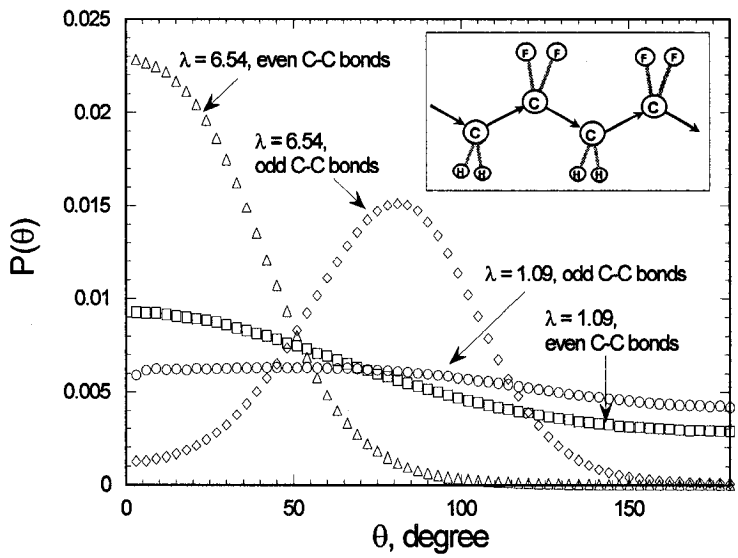


Fig. 3.2.5 Angular distribution of odd $C-C$ bonds and even $C-C$ bonds for Model II for λ = 1.09 and 6.54; the inset illustrated the definition of bond vectors.

Moreover, as the coupling strength was increased to 6.54, the $P(\theta)$ exhibited distinct peaks located at around $\theta = 80$ and 0 degree for the odd and even bonds, respectively, in **Fig. 3.2.5**. Unlike Model I, the difference of these two angles was not directly related to bond angle because the net dipole was subject to fluctuation due to internal rotation within a monomer. (For even larger field strengths, internal rotation was gradually hindered, and the peak positions of the odd and the even bond shifted to larger angles; their difference approached the limiting 62 degrees due to the constraint of bond angles.) In spite of internal rotation at weaker fields, it was found that the trans conformer was the preferred monomer structure, and this structure was enhanced by electric fields. In such a structure, an odd C-C bond tended to align in the direction more perpendicular to the field, whereas an even C-C bond of Model II oriented more toward the field direction. Namely, under a strong electric field, the elongated chain molecule for Mode II was tilted against field direction. As a result, the mean chain size perpendicular to the field direction $< R_z^2 >$ increased with increase of λ.

At even larger field strength, this local structure difference was further magnified. Snapshots of both models at extremely large fields were shown in **Fig. 3.2.6**. It could be observed that the fluorine atoms (shown as darker gray) were rather random in Model I, but were predominantly located at the same side of the chain for Model II. The asymptotic values of mean squared end-to-end distance for the Model II reached that of their completely extended zigzag

conformation, but only reached 80% of their fully extended conformation for the model I. Meanwhile, the orientation of the two models was very different: the average orientation (end-to-end vector orientation) of Model I was toward the field direction, but the fully stretched chain of Model II was tilted against the field direction, forming an angle of approximately 67 degrees. This was due to the differences in the rotational freedom between the two models. For Model I, the even-odd-even bond dihedrals were not restricted even in the strongest fields; therefore it retained this mode of motion despite the strong alignment on the odd numbered bonds. However, for Model II, all the backbone dihedrals were strongly restricted by the external field, since any bond rotation would result in lowering the favorable (attractive) interaction between the field and dipoles on chain branches and interaction. As a result, in extreme large fields, the chain was fully extended and tilted to maximize dipole-field interaction.

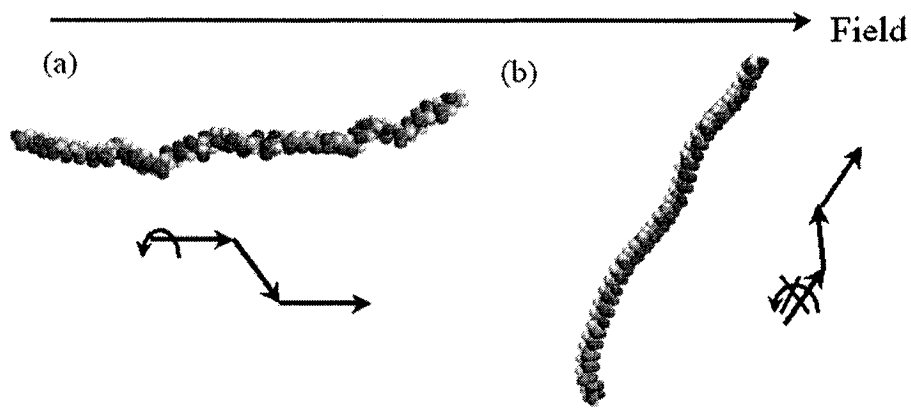


Fig. 3.2.6 Illustration of Model I (a) and Model II (b) in extremely large fields. Only the carbon (light gray) and fluorine (darker gray) atoms were shown. The connected arrows were to show whether the bond rotation was allowed.

3.2.4 Conclusions

Monte Carlo simulations have been conducted to investigate two polar polymer models that were derived from the all-atom poly(vinylidene fluoride) (PVDF) model, to elucidate the electric field induced chain elongation. In the first model, the methylene and methylene fluoride groups of a PVDF molecule were treated as united atoms, similar to neutral polyampholytes containing alternating positive and negative charges. The charge of each charged site was obtained by summing up the partial charges of the constituent atoms for that group. In the second model, the constituent atoms and partial atomic charges were explicitly incorporated.

To compare with the analytical theories, the weak coupling approximation was considered without intramolecular interactions involved. The simulated

mean chain size for Model I could be depicted quantitatively by the analytical theories at weak fields. For Model II, the overall mean chain size agreed well with that of Model I for weak fields. However, the component of the mean chain size perpendicular to the field direction displayed different behaviors for the two models, and the deviation became pronounced at stronger fields. As the field strength was increased, the magnitude of this component decreased for Model I, but increased for Model II. Such a difference could be attributed to the subtle spatial arrangement of charged sites within a monomer in different models. In Model I, the effective dipole of a monomer lied on the C-C bond and oriented toward the field direction. However, the effective monomer dipole of Model II was tilted against the C-C bond, such that the chain molecule aligned more perpendicular to the field direction.

This project was initiated by (1) development of a coarse-grained polyampholyte-like model to depict the conformational behavior of polar polymers, such as PVDF; (2) understanding of the chain properties of PVDF in the polling process to manufacture piezoelectric materials. The findings indicated that the simple polyampholyte-like model, with alternating positive and negative charges, would capture the conformational behavior of chain elongation of PVDF in weak electric fields. In stronger fields, the results suggested that the local dipole moment within a monomer were crucial in governing the orientation of an elongated chain molecule in an applied field. The findings may be linked with the electric poling of PVDF films, in which

the uniaxial stretching resulted in the β-phase of PVDF crystallites with PVDF molecules adopting all-trans conformation. As a result, the dipole arising from a monomer (CH_2-CF_2) displayed a tendency to align along the applied electric field. This observation has been consistent with the result of the investigated model with explicit atoms incorporated under a strong field.

- Chapter 4 -

Molecular Alignment of Rigid Rods in Non-rigid Spherical Pores

4.1 Introduction

An increase of interest has been devoted to synthesis of composite materials consisting of molecules trapped within a controllable confined environment. Recently, some novel core-shell structures based on self-organization of the mixture of neutral/polyelectrolyte copolymers and oppositely charged surfactants have been reported. The neutral/polyelectrolyte copolymers formed a larger micellar shell to trap smaller surfactant micelles in the core. From the standpoint of structure, these micellar shells were of ideal sub-micron or nano-scaled containers, and the shell structure could be viewed as confined boundaries. One question drawn from these systems was how the core materials distribute inside the shell. For cylindrical micelles, their orientation was another important property to be addressed, which inspired the work presented in this chapter.

In addition to core-shell structures, local confinement was a basic physical feature found in numerous composite materials.[119-122] In polymer/liquid crystal composites, polymer matrices induced confinement to liquid crystal molecules. Several examples for polymer/liquid crystal composites have been reported in literature. Park et al have developed a novel method to encapsulate liquid crystals in monodisperse micron-sized polymer particles.[123] Vorflusev and coworkers have prepared the phase separated composite films, with adjacent layers of liquid crystal and polymer, which can greatly enhance the performance of liquid crystal displays.[124] In the above two examples, polymeric materials can be viewed as confining boundaries to rodlike liquid crystals molecules.

The concept of confinement in composite materials can be further extended to biological systems. In vivo, cellular membranes separate the cellular interior from outside, and play the role to confine "core" materials, such as chromosomes, in biological cells.[125] Meanwhile, most of membrane proteins were orientated in cellular membrane in certain fashion, e.g., Rhodopsin.[126] In our view, the alignment of helical coils in Rhodopsin may be induced by the confined environment within a membrane.

An alternative way to induce confined environments was based on optical trapping, for instance optical tweezers. Such a method has been applied to trap spherical or non-spherical objects including liquid droplets or long chain DNA molecules, etc.[127-129] The trapping force was generated by strong confocal

laser beams of which light intensity displays Gaussian-like distribution. Namely the confocal area yielded a spatial gradient similar to the harmonic potential, and served as a flexible cavity to particles confined inside. In this work, harmonic potential was utilized to mimic the confining cavities, such as the interior of optical tweezers. In addition, harmonic potential was a simple model to depict the flexible boundaries of the confinement environments in materials and biological cells. Despite its simplicity, this model allowed us to address the basic physics associated with non-rigid pores.

The effect of various external fields on chain conformation for single polymers has been reported. Field induced conformational deformation and molecular alignment was sensitive to the local chemical structure of a polymer molecule, as shown in Chapter 2. In the presence of spherically harmonic potentials, the chain was compressed to globule,[130] and under the two-dimensional harmonic potentials, the chain becomes elongated.[131] These features observed in harmonic potentials were similar to polymer chains confined in spherical cavities and tubes.

In this chapter, Monte Carlo simulations and geometrical analysis were employed to elucidate the orientation ordering of two rigid rods confined by a harmonic potential. The rigid rod was primarily modeled as a Shish-Kebab chain, with tangent hard spheres aligned in the same axis. Shish-Kebab chains were a widely investigated model for the liquid crystal ordering of rigid rods.[132-135] The aims of this work were to explore the influences of harmonic

potential and local rod smoothness on the molecular alignment of two confined rigid rods. The geometrical analysis enabled us to examine the excluded volume interactions of Shish-Kebab chains and the asymptotic values of orientation order parameters in the limiting weak and strong fields. To understand the role of local rod structure, Shish-Kebab chains were compared with compressed Shish-Kebab chains and spherocyliners in the simulations. In these models, Shish-Kebab chains were the bumpiest, and spherocyliners were the most smooth. The compressed Shish-Kebab chains were in between these two extremes. Furthermore, this study would serve as a starting point to understand the alignment for more complex multiple rods in non-rigid confined potentials.

4.2 Models and Simulations

The molecular alignment of two Shish-Kebab chains confined by harmonic potentials was the primary focus of this work. The Shish-Kebab chain consisted of tangent hard spheres aligned along the same axis, and was a model widely chosen in literature. The diameter of each bead in the chain model was σ (=1), and was also the length unit in our calculations. Actually, the Shish-Kebab chain model exhibited pronounced grooves between adjacent tangent spheres. To test the influences of the local structure of the chosen model, two other rigid rod models were also investigated in the simulations, including hard spherocylinders and compressed Shish-Kebab chains. **Fig. 4.1** was the schematic plot of the chosen rigid rod models. All three models

consist of N interacting beads which were subject to the applied potential. These interaction sites were separated by a distance equal to the bead diameter, σ. The applied potential, same for the three models, was given by

$$\frac{V}{k_B T} = \sum_{i=1}^{M \times N} \frac{1}{2} k \mathbf{r}_i^2 \tag{4.1}$$

where k was the strength of the applied potential; k_B was the Boltzmann constant; T was the temperature; N was the number of interacting sites on a bead; M was the number of rods confined to the applied potential; $\mathbf{r}_i$ was the vector of the i-th bead measured from the center of the applied field.

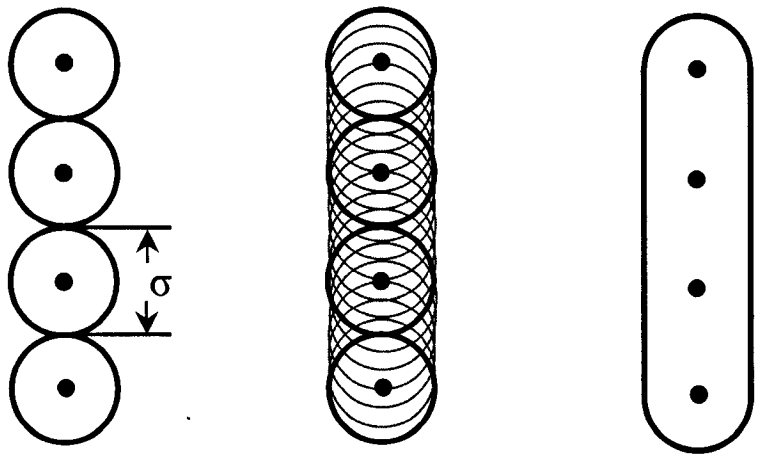


Fig. 4.1 Schematic plots of Shish-Kebab chain model in (a), compressed Shish-Kebab chain model in (b), and spherocylinder model in (c), with the sites interacting with the applied field denoted by dots.

The difference among the three models lied in their intermolecular excluded volume interactions. Hard spherocylinders were the smoothest, with a cylindrical surface capped with hemispheres on either end. The excluded volume interactions of two hard spherocylinders were checked through the shortest distance between two line segments of finite length, as the algorithm suggested by Allen et al.[136] In contrast, Shish-Kebab chains were the bumpiest, with tangent hard spheres aligned in a straight molecular axis. Between these two cases, the compressed Shish-Kebab model had additional beads fused between the tangent spheres of the Shish-Kebab. In the compressed Shish-Kebab model, additional N_p beads were incorporated and were distributed uniformly between any two adjacent beads of a Shish-Kebab model. These additional beads (i.e., non-interacting beads) experienced no interactions from the applied potential, but did have hard core repulsions with beads on the other chain. The size of all beads was assumed the same. For Shish-Kebab chains and compressed Shish-Kebab chains, the excluded volume interactions took a general form,

$$U(r_{ij}) = \begin{cases} \infty, & r_{ij} < \sigma \\ 0, & r_{ij} \geq \sigma \end{cases} \tag{4.2}$$

where r_{ij} was the distance between any two beads on the two different rods. (Note to test the algorithm, the simulations of hard spherocylinders were compared with the compressed Shish-Kebab chains at near smooth rod condition, with a sufficiently large number of non-interacting beads

incorporated into the chain model. A good agreement between the two models was found.)

In the simulations, the order parameter was defined as $< S_2 > = -2\lambda_0$, as suggested by Eppenga et al,[137] where λ_0 was the middle eigenvalue of the ordering matrix **Q** (by noting that $\lambda_1 > \lambda_0 > \lambda_{-1}$ for three eigenvalues). The tensor elements of **Q** were given by

$$Q_{\alpha\beta} = \frac{1}{2N}\sum_{i}^{N}(3e_{i\alpha}^{s}e_{i\beta}^{s} - \delta_{\alpha\beta}) \tag{4.3}$$

where α and β represented orthogonal coordinates x, y or z; e_i^s was the i-th molecular axis. In addition to order parameters, the average distance between the center-of-mass $< r >$ of two rods and the angular distribution between two rods were also computed.

4.3 Asymptotic analysis of two Shish-Kebab chains in harmonic potentials

In this section, discussion of the mathematical analyzed for the properties of Shish-Kebab chains in spherically harmonic potentials were given. The starting point was to calculate the energy of a single Shish-Kebab chain in a harmonic potential. The asymptotic analysis for weakly coupled chains then followed. For the further analysis, a function related to the angular dependent excluded volume interaction was first defined. This function will be unitized

to calculate the asymptotic order parameter for two Shish-Kebab chains confined in the limiting strong field.

4.3.1 Interaction energy of single rods in a spherically harmonic potential

Under a harmonic potential with spatial gradient, the interaction imposed on a rod was related to its position and orientation. However, as would be shown in the following, the angular component of a rod, indeed, was canceled in a spherically harmonic potential.

For the Shish-Kebab chain composed of odd-numbered beads, the center-of-mass was located at the central bead of the rod. Suppose that the position of the center-of-mass of the rod was (x_0, y_0, z_0), and the directional (unit) vector of the rod $\mathbf{v}$ $(= (\mathbf{r}_N - \mathbf{r}_1)/|\ \mathbf{r}_N - \mathbf{r}_1\ |)$ was (v_x, v_y, v_z). Thus the position of the j-th (non-central) bead would be $(x_0 + jv_x, y_0 + jv_y,\ z_0 + jv_z)$. Since each bead had a mirror bead with respect to the center-of-mass along the molecular axis, the coordinate of the mirror bead was $(x_0 - jv_x, y_0 - jv_y, z_0 - jv_z)$. The total interaction energy of a rod molecule was expressed as

$$\frac{V}{k_B T} = \frac{1}{2} k\{(x_0^2 + y_0^2 + z_0^2) + \sum_{i=1}^{(N-1)/2} [(x_0 + jv_x)^2 + (y_0 + jv_y)^2 + (z_0 + jv_z)^2$$

$$+(x_0 - jv_x)^2 + (y_0 - jv_y)^2 + (z_0 - jv_z)^2]\} \qquad (4.4)$$

After some algebra, the interaction energy was reduced to the following form

$$\frac{V(odd)}{k_B T} = \frac{1}{2} k\{N(x_0^2 + y_0^2 + z_0^2) + \frac{N(N^2-1)}{12}\} \qquad (4.5)$$

For a Shish-Kebab chain of even-numbered beads, the center-of-mass was located at the middle of the central groove. But, the interaction energy took the same form as in Eq. (4.5).

To this end, a non-trivial result was obtained. For a given chain length, the interaction energy of a Shish-Kebab chain in a spherical harmonic potential was a function of the position of its center-of-mass only, and was independent of the angular parts. It greatly simplified the calculation of interaction energy arising from the applied potential because the orientation of a rod could be neglected. This result also implied that the spatial distribution of a single rod or multiple non-interacting rods in a spherically harmonic potential was basically a Gaussian function. Namely, the applied potential played the role of regulating the position of a rod molecule in space.

4.3.2 Two uncorrelated rods

Here the order parameter for two uncorrelated Shish-Kebab chains was considered. Such a limiting case corresponded to the two rods in an extremely weak applied potential, for which the two rods were far apart and their excluded volume interaction was negligible. The order parameter was to be solved analytically for this limiting case. Without losing generality, it was assumed that the molecular axis of the first rod was placed along the z-axis,

thus the directional (unit) vector of the first rod reads $(0,0,1)$. The center of the second rod was placed at xz plane, with coordinate $(rsin\psi,0,rcos\psi)$, where r was the distance between the center-of-mass of the two rods and ψ was the polar angle. The directional vector of the second rod was $(\sin\theta\cos\phi,\sin\theta\sin\phi,\cos\theta)$ in polar coordinate. A schematic plot was shown in **Fig. 4.2(a)**. For this case, the ordering matrix **Q** in Eq. (4.3) took the following form,

$$\mathbf{Q}=\begin{bmatrix} \frac{3\sin^2\theta\cos^2\phi}{4}-0.5 & \frac{3\sin^2\theta\cos\phi\sin\phi}{4} & \frac{3\sin\theta\cos\theta\cos\phi}{4} \\ \frac{3\sin^2\theta\cos\phi\sin\phi}{4} & \frac{3\sin^2\theta\sin^2\phi}{4}-0.5 & \frac{3\sin\theta\cos\theta\sin\phi}{4} \\ \frac{3\sin\theta\cos\theta\cos\phi}{4} & \frac{3\sin\theta\cos\theta\sin\phi}{4} & \frac{3\cos^2\theta}{4}+0.25 \end{bmatrix} \tag{4.6}$$

The eigenvalues of this matrix were obtained by solving the cubic equation of $\det[\mathbf{Q}-\lambda\mathbf{I}]$. After some algebra, the analytic form of the three eigenvalues was obtained,

$$\lambda_1 = 0.25+0.75\cos\theta \tag{4.7}$$

$$\lambda_0 = 0.25-0.75\cos\theta \tag{4.8}$$

$$\lambda_{-1} = -0.5 \tag{4.9}$$

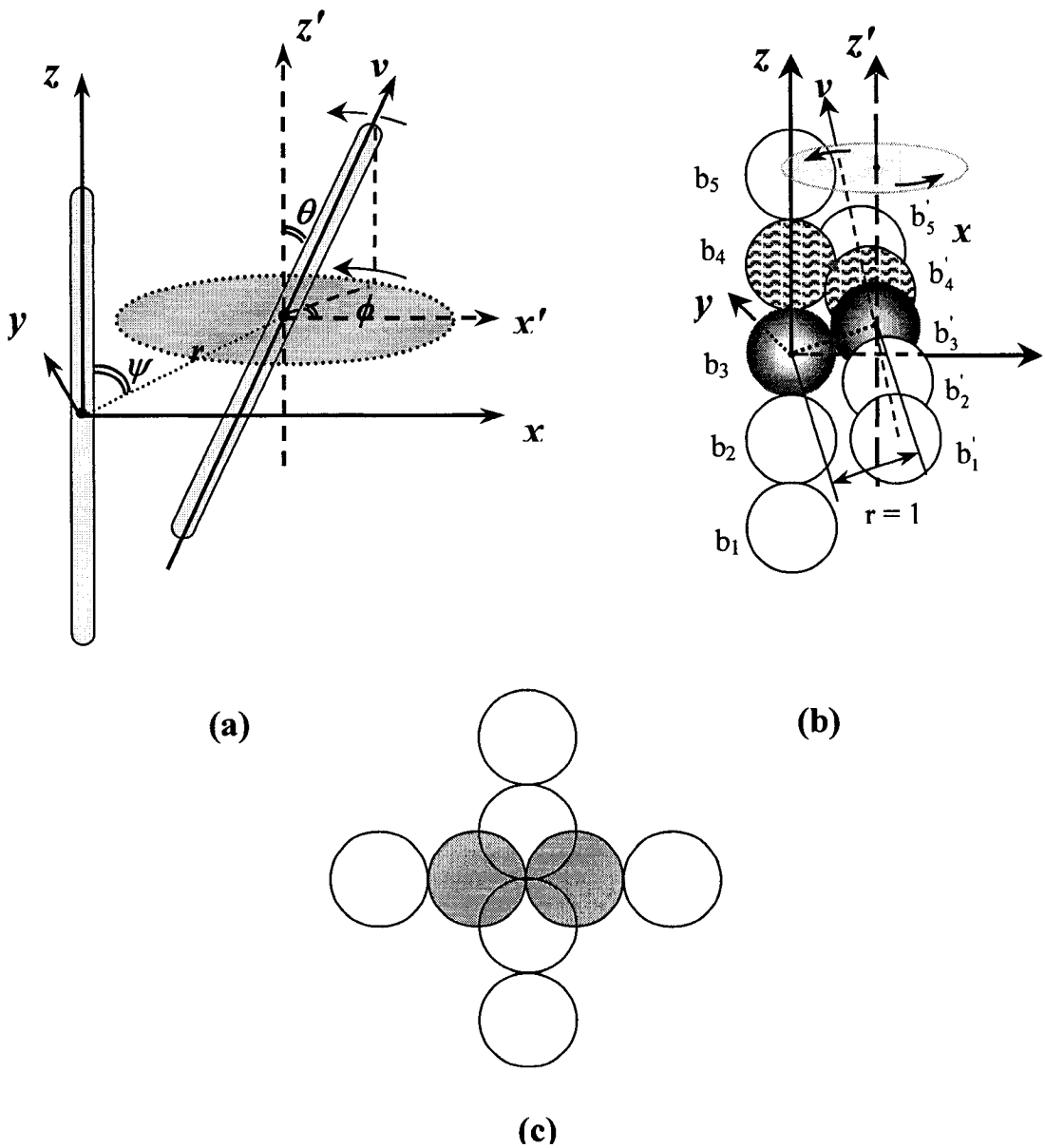


Fig. 4.2 Schematic plots for the coordinates and geometric variables of two rods in (a); two Shish-Kebab chains of odd-numbered beads under strong confinement, with the minimum distance between the two center-of-mass equal to σ in (b); two perpendicular Shish-Kebab chains of even-numbered beads in (c).

As suggested by Eppenga et al, $-2\lambda_0$ (where λ_0 was the middle eigenvalue and has a greater sensitivity for angular ordering) was used as the order parameter. Thus, the angular (θ) dependent order parameter S_2 was

$$S_2 = 1.5\cos(\theta) - 0.5 \tag{4.10}$$

With the formulation of S_2, the mean order parameter $< S_2 >$ was calculated for the case of two uncorrelated rods. The angular distribution function $P(\theta)$ for θ ranging from 0 and $\pi/2$ would scale with $\sin(\theta)$ in 3D spherical coordinate, and was symmetric to the range $[\pi/2, \pi]$. So,

$$< S_2 >= \frac{\int_0^{\pi/2} (1.5\cos\theta - 0.5)\sin\theta d\theta}{\int_0^{\pi/2} \sin\theta d\theta} = 0.25 \tag{4.11}$$

This result indicated that when the two rods were weakly coupled, their order parameter $< S_2 >$ had a non-zero asymptotic value 0.25. This value was independent of the local structure of the chosen rigid rod models.

4.3.3 Useful functions for angular dependent excluded volume interactions

In a strong confining potential, the two rods were restricted in smaller space and their collision occurs for the majority of configurations. A geometric analysis of the angular dependent excluded volume interactions disclosed the orientation ordering at the limiting strong fields. The following

analysis was conducted through examining the configurations of two rods free of collisions.

In **Fig. 4.2(a)**, the coordinate of each rod was defined. The first rod was aligned along the z-axis in molecular coordinate (x,y,z), and its center-of-mass was located at the origin. For the second rod, the origin of its molecular coordinate (x',y',z') was shifted to the center of this rod by keeping (x',y',z') parallel to (x,y,z). In terms of spherical coordinate, ψ and r were the polar angle and the distance for the center-of-mass of the second rod measured from the molecular coordinate (x,y,z), respectively. Note r was also the distance between the center-of-mass of the two rods. To depict the orientation of the second rod, the polar angle θ and the azimuthal angle ϕ were described in its own molecular coordinate. θ was measured from the z'-axis. ϕ was chosen in such a way that at $\phi = \pi/2$, the second rod was parallel to the yz plane. Since no collision was possible between two rods at this configuration, it would be used as the reference point. And ϕ was to be rotated anticlockwise in a complete circle to identify collisions.

In the following, a function to quantify the angular dependence excluded volume interaction was defined. Prior to our introduction, the range of each angle for the calculations when r was fixed needed to be specified. Due to the symmetric nature of rods, the angles to be considered for the second rod were in the following ranges: $\theta \in [0,\pi/2]$ and $\psi \in [0,\pi/2]$. For given θ and ψ, $\phi_1(\theta,\psi)$ was the boundary for the occurrence of the first collision in the range

$[\pi/2,\pi]$ when ϕ was rotated anticlockwise in **Fig. 4.2(a)**. The fraction q_1 for ϕ free from collision in the range $[\pi/2,\pi]$ (same for $[\pi,3\pi/2]$) was

$$q_1(\theta,\psi)=\frac{\phi_1-\pi/2}{\pi/2} \tag{4.12}$$

With a similar argument, q_2 was defined for ϕ in the range $[3\pi/2,2\pi]$ (same for $[0,\pi/2]$) as

$$q_2(\theta,\psi)=\frac{\phi_2-3\pi/2}{\pi/2} \tag{4.13}$$

The total allowed fraction for the rotation of ϕ in $[0,2\pi]$ was then

$$q(\theta,\psi)=\frac{q_1(\theta,\psi)+q_2(\theta,\psi)}{2} \tag{4.14}$$

$$f(r,\theta)=\int_0^{\pi/2} q(\theta,\psi)\sin\psi\, d\psi \tag{4.15}$$

Solving $f(r,\theta)$ analytically for any r and θ was a very difficult task because the site-site interactions involved in Shish-Kebab chains. For the rod of N beads, the total number of pair interactions was N^2. To accurately calculate the overlap (collision) boundary, it was necessary to solve N^2 equations. Nevertheless, $f(r,\theta)$ became mathematically tractable when $r\to 1$ because the number of pair interactions needed to be considered was greatly reduced under distance constraints. This result was applied to find the asymptotic values of order parameters under strong fields in the next section.

Despite its complexity, a numerical method could be devised to compute $f(r,\theta)$ for a given r. In the calculations, the values of ϕ_1 and ϕ_2 were identified as follows. The range of ϕ angle was divided uniformly into small intervals. For each ψ and θ, we scan ϕ, by increasing ϕ with small increments, in the two separate regions: $[\pi/2,\pi]$ and $[3\pi/2,2\pi]$. The rotation of ϕ was started at $\pi/2$ for $[\pi/2,\pi]$ and $3\pi/2$ for $[3\pi/2,2\pi]$ to search for the occurrence of the first collision in each region. The criterion of collision was based on the distance for all possible pairs. For a given ϕ value, the first collision was defined as the point that any one of these distances becomes less than 1. The collision ϕ angle was marked as ϕ_1 if $\phi \in [\pi/2,\pi]$, and as ϕ_2 if $\phi \in [3\pi/2,2\pi]$. With ϕ_1 and ϕ_2, the allowed fraction (free from collision) of ϕ could be determined for given ψ and θ using Eqs. (4.12), (4.13) and (4.14). After integrating over ψ, $f(r,\theta)$ was obtained in Eq. (4.15). The pilot calculations were to be presented in the results and discussion.

4.3.4 Two rods under strong confinement

In an extremely strong field, the two rods were very close to each other. The lowest energy configuration occurred at the point where the separation of the center-of-mass of two Shish-Kebab chains was minimized. Meanwhile, the field center was located at the mid-point along the axis connecting the center-of-mass of two rods. The center-of-mass of a Shish-Kebab chain of

odd-numbered beads and even numbered-beads was located at the central bead and the central groove, respectively. Due to this difference, their orientation order under strong confinement would be different, and it was instructive to investigate the asymptotic order parameters for both cases.

4.3.4.1 Two rods of odd numbered beads

For Shish-Kebab chains of odd-numbered beads, the smallest distance between the center-of-mass of two rods was $r = 1$. Such a condition could be reached at limiting strong fields. The schematic plot of the model was shown in **Fig. 4.2(b)**. Only the five central beads were plotted for each rod. Together with the definition in **Fig. 4.2(a)**, the coordinates of the five central beads, $b_1 - b_5$, of the first rod can be generalized to $(0,0,m)$ for bead b_{m+3}, where m was the integer between -2 to 2. It was assumed that the central bead b_3' of the second rod was located at xz plane with coordinate $(\sin\psi, 0, \cos\psi)$, and the directional (unit) vector of this rod was $(\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$, where θ and ϕ were the polar angle and the azimuthal angle defined in **Fig. 4.2(a)**. In the five central beads of the second rod, the coordinate of b'_{j+3} was $(\sin\psi + j\sin\theta\cos\phi,\ j\sin\theta\sin\phi,\ \cos\psi + j\cos\theta)$, where j was the integer between -2 to 2.

Owing to the distance constraint $r = 1$, b_3 and b_3' were kept next to each other, but b_3' was allowed to move around the surface of b_3. With excluded

volume interactions, for $\theta = 0$, only three configurations were allowed: (1) both b_3 and b_3' were situated on x-axis; (2) b_3' shifts upward by $\sigma/2$ along z'-axis, and was parallel to the center of the neighboring groove above b_3; (3) same as (2) but shifting b_3' downward by $\sigma/2$. These configurations arose from $\psi = 0$, $\pi/3$ and $-\pi/3$. Rotating the azimuthal angle ϕ of the second rod for these three configurations was equivalent to rotating its molecular axis, and did not change their configurations. As θ was increased, the amounts of allowed configurations increase because the allowed range of ψ and ϕ free from collision was increased. For the further analysis of angular dependent excluded volume, the collision (or overlap) boundary was defined as follows. As ϕ was rotated in $[\pi/2, \pi]$, the first collision occurs between b_3 and b_4' or between b_4 and b_4' at $\phi = \phi_1$. This could be pictured from **Fig. 4.2(b)** or from the geometric argument. Similarly, for the rotation of ϕ in $[3\pi/2, 2\pi]$, the first collision of the two rods occurred either $b_3 - b_2'$ or $b_2 - b_2'$ at $\phi = \phi_2$. Considering the symmetry of the system, these two cases would covered all the possible relative configurations between the two strongly confined Shish-Kebab chains.

From the above discussion, the number of pair interactions needed to be considered was greatly reduced. To obtain ϕ_1 value, only two distances would be focused on: for $b_3 - b_4'$, $\quad d_1 = (\sin\psi + \sin\theta\cos\phi)^2 + (\sin\theta\sin\phi)^2$

$+(\cos\psi+\cos\theta)^2$, and for b_4-b_4', $d_2=(\sin\psi+\sin\theta\cos\phi)^2+(\sin\theta\sin\phi)^2$ $+(\cos\psi+\cos\theta-1)^2$. Upon collision, either d_1 or d_2 would be equal to 1 whereas the other would be larger or equal to 1. Solving these two equations, ϕ_1 in Eq. (4.12) was obtained for $\phi\in[\pi/2,\pi]$ as follows

$$\cos\phi_1=\begin{cases}\dfrac{-1-2\cos\psi\cos\theta}{2\sin\psi\sin\theta}, & \cos\theta+\cos\psi<0.5\\ \dfrac{\cos\theta+\cos\psi-1-\cos\psi\cos\theta}{\sin\psi\sin\theta}, & \cos\theta+\cos\psi\geq 0.5\end{cases} \tag{4.16}$$

Similarly, in $\phi\in[3\pi/2,2\pi]$, the value of ϕ_2 in Eq. (4.13) was obtained by considering the two distance constraints, b_3-b_2' and b_2-b_2', given by

$$\cos\phi_2=\begin{cases}\dfrac{1-2\cos\psi\cos\theta}{2\sin\psi\sin\theta}, & \cos\theta-\cos\psi<0.5\\ \dfrac{-\cos\theta+\cos\psi+1-\cos\psi\cos\theta}{\sin\psi\sin\theta}, & \cos\theta-\cos\psi\geq 0.5\end{cases} \tag{4.17}$$

Using Eqs. (4.12), (4.13), (4.14) and (4.15), the expression for the allowed fraction $f(r=1,\theta)$ was obtained, defined in Eq. (4.15), as a function of θ. Because r was invariant, the angular ordering distribution function $P(\theta)$ was formulated as

$$P(\theta)=\frac{f(\theta)\sin\theta}{\int_0^{\pi/2}f(\theta)\sin\theta d\theta} \tag{4.18}$$

The order parameter was then calculated as

$$<S_2>=\int_0^{\pi/2}(1.5\cos\theta-0.5)P(\theta)d\theta. \tag{4.19}$$

By numerical integration, the asymptotic order parameter was obtained $< S_2 > = 0.1310$ for two strongly confined Shish-Kebab chains of odd-numbered beads.

4.3.4.2 Two rods of even number of beads

For rods of even-numbered beads, the center-of-mass of a rod fell into the center of its central groove. Intercalation of the two rods perpendicularly through their central groves minimized the total energy because the smallest distance between the center-of-mass of each rod and the center of the applied field could be reached. A schematic plot of this configuration was shown in **Fig. 4.2(c)**. Since the central four beads on the two rods would form a regular tetrahedron of side length 1, the distance between the two center-of-mass was the distance between the two opposing edges in a tetrahedron, i.e., $\sqrt{2}/2$. With limited rotational freedom, the asymptotic value of the order parameter for this scenario was $< S_2 > = 1.5\cos\theta - 0.5 = -0.5$.

4.4 Results and Discussions

First, the simulation results for the orientation ordering of two confined Shish-Kebab chains were to be presented. **Fig. 4.3** displays the variation with field strength k of the order parameter $< S_2 >$ for different chain lengths $N =$ 5, 6, 9 and 12, and the average distance between the two rods $< r >$, for $N = 5$ and 6, respectively, as marked. The simulation data could be roughly

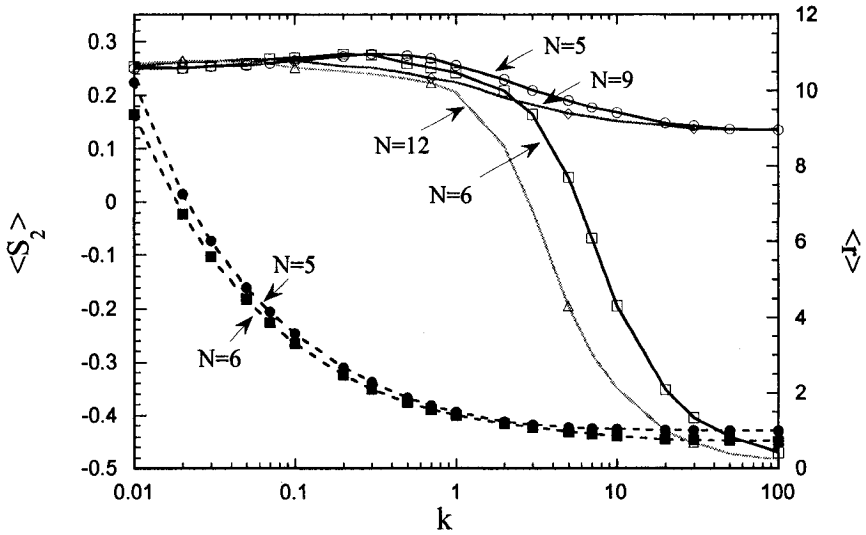


Fig. 4.3 Variation with field strength k of the order parameter $< S_2 >$ for different chain lengths N = 5, 6, 9 and 12 (open symbols), and the average distance between the two Shish-Kebab chains $< r >$ (solid symbols), for $N = 5$ and 6, as marked. Lines are meant to guide our

divided into three regimes, weak, intermediate and strong fields. At weak fields, the two rods were far apart within the applied potential, and the order parameter $< S_2 >$ was close to the asymptotic value 0.25 from Eq. (4.11) of two uncorrelated (isotropic) rods, as discussed in section 4.3.2. In intermediate field strengths, the average rod separation $< r >$ continued to decreases. Unlike $< r >$, $< S_2 >$ showed a perceivable non-monotonic behavior: it increased slightly first and decreased again after passing a maximum. Such a

non-monotonic behavior diminished as chain length was increased. For strong fields, $< r >$ and $< S_2 >$ converged to some asymptotic values, depending on the parity of chain length. For two odd-numbered rods, $N = 5$ and 9, the convergent values for $< S_2 >$ and $< r >$ were equal to 0.131 and 1.0, respectively, whereas for two even-numbered rods, $N = 6$ and 12, they were equal to -0.5 and 0.71. The convergent values of order parameters for 0.131 and -0.5 agreed well with the asymptotic values predicted by sections 4.3.3.1 and 4.3.3.2 for Shish-Kebab chains of odd-numbered beads and even-numbered beads, respectively. Also, other chain lengths have been tested (data not shown), and the parity dependent convergent values were found to hold for all Shish-Kebab chains regardless of rod length *N*. In the following, a further discussion was given for the behaviors observed in intermediate and strong fields.

To better understand the alignment in intermediate and strong field strengths, $f(r,\theta)$ defined in Eq. (4.15) was calculated for two Shish-Kebab chains. This function was relevant to the orientation dependent excluded volume between the two rods, and served as a map to tabulate the excluded volume interactions in terms of r and θ. In **Fig 4.4**, $f(\theta)$ was plotted for different r for $N = 5$, denoted by solid lines, and for $N = 6$, denoted by dotted lines. Note the cusps observed in the figure were due to the discrete hard spheres in a Shish-Kebab chain. At $\theta = 0$, $f(r,\theta)$ was identical between $N = 5$ and 6 for all r.

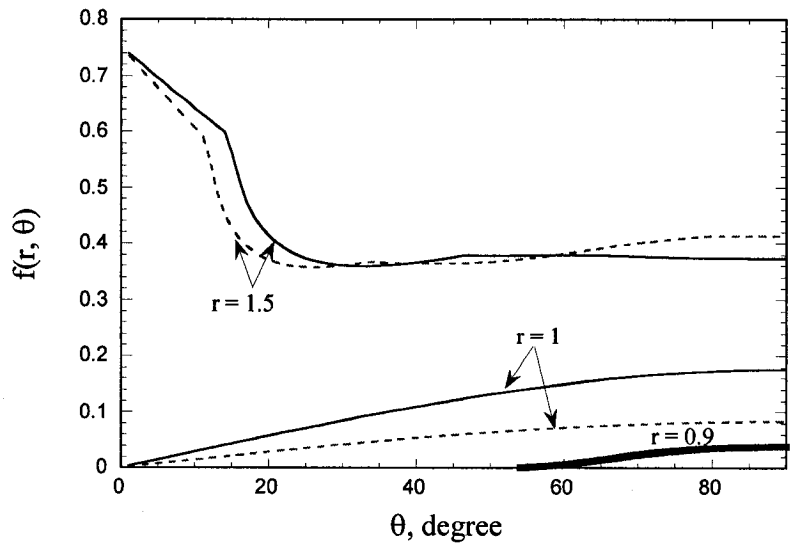


Fig. 4.4 Plot of $f(r,\theta)$, the allowed fraction of the rod rotation relative orientation defined in Eq. (4.15), for two Shish-Kebab chains of $N = 5$ and 6 for different r.

When $\theta > 0$ and r was less than 1, $f(r,\theta) = 0$ for the rods of odd-numbered beads, but for the rods of even-numbered beads, the non-zero $f(r,\theta)$ emerged at near $\theta = 90$ degrees. Actually, when r was close to and less than 1, $f(r,\theta)$ was an increasing function. This could be understood from the fact that two parallel rods were subject to collision when they were tilted to each other. As θ was increased, the contact area between the two rods decreased along with a decrease of the collision probability and an increase of free volume for θ angle rotation. For a large enough r, such as

$r = 1.5$, $f(r,\theta)$, in general, tended to decrease as θ was increased. This was because the two shorter chains were allowed to tilt without triggering collision for smaller tilted angles. However, an increase of the tilted angle enhanced the chance of collision between two rods. When r was increased further, the two rods experienced less collision, which resulted in an increase of $f(r,\theta)$. For a longer chain ($N = 6$), $f(r,\theta)$ quickly dropped below that of the shorter chain ($N = 5$). This result was consistent with the above picture that the two longer rods experience more collisions for smaller tilted angles.

In intermediate field strengths where $< S_2 >$ displayed a maximum, the average separation between two rods was greater than 1 but smaller than the chain length, such as $r = 1.5$ in **Fig. 4.4**. Namely, two rods of finite chain lengths exhibited the preference for parallel configuration, as our discussion above. However, such a configuration was impeded as chain length was increased. Therefore, as chain length N was increased, the maximum of $< S_2 >$ shifted toward a smaller k and eventually diminishes for large N.

At very strong fields, the dependence of chain length parity for $< S_2 >$ and $< r >$ could be understood as follows. For the rods of odd numbered beads, the limiting values of $< S_2 > = 0.131$ and $< r > = 1$ arose from the average of multiple-angle configurations. However, the limiting values of $< S_2 > = -0.5$ and $< r > = \sqrt{2}/2$ for the rods of even numbered beads indicated the two rods

were perpendicular under a strong enough field. The difference lied in the subtle differences in local rod structure.

The above observations for different chain length parities were associated with energy. To reduce repulsions from the applied potential, the center-of-mass of Shish-Kebab chains tended to distribute around the center of the applied potential, as discussed in section 4.3.1. For a Shish-Kebab chain of odd-numbered beads, its center-of-mass was situated at the central bead. Under very strong fields, the lowest configurational energy occurred when the two central beads make a contact at $< r > \approx 1$ with the field center located in between the two central beads. More rotation was allowed for the two rods even under strong fields, which gave rise to an $< S_2 > = 0.131$, between 0.25 for uncorrelated rods and −0.5 for perpendicular rods. This value indicated that some rotational degrees of freedom were hindered. Intercalation of the two Shish-Kebab chains of odd-numbered beads accounted for the rotational hindrance, which occurs when the central bead of a rod snapped into to either of the central grooves of the other rod. In contrast, for the Shish-Kebab chains of even-numbered beads, the center-of-mass was located at the center of a rod molecule, i.e., in the middle of the central groove. To reach the lowest energy configuration, the two rods packed in a perpendicular fashion through intercalation of the two central grooves of the two rods, as shown in **Fig. 4.2(c)**. Such a configuration with $r < 1$ minimized the repulsive energy from the applied potential. The asymptotic value for $< r > = 0.71$ was consistent

with **Fig. 4.2(c)**, as was discussed in section 4.3.4.2. This value corresponded to the distance between two opposing edges in a regular tetrahedron with the edge length ($\sqrt{2}/2$). Meanwhile, this lowest-energy configuration constrained the rods from free rotation, and was responsible for the limiting value $< S_2 >$ = -0.5 for rods of even-numbered beads.

To further test the effect of local rod structure, the simulations of compressed Shish-Kebab chains and spherocylinders were compared with those of Shish-Kebab chains. In the compressed Shish-Kebab model, $N_p = 5$ (non-interacting beads) was chosen. As the grooves in a Shish-Kebab chain were partially filled with non-interacting beads, the rod molecules approached spherocylinders with a totally smooth surface. **Fig 4.5** plotted the $< S_2 >$ against k for $N = 5$ and N = 6, for different models, as marked. For weak fields, the three models had the same value $< S_2 >$ = 0.25, similar to two uncorrelated rods. In intermediate field strengths, the $< S_2 >$ displayed a maximum for the three models investigated. Despite their pronounced difference in local rod structure, the maximum $< S_2 >$ was located at approximately the same k for different models of same chain length, which suggested a weak correlation between the order parameter and the local structure of rods.

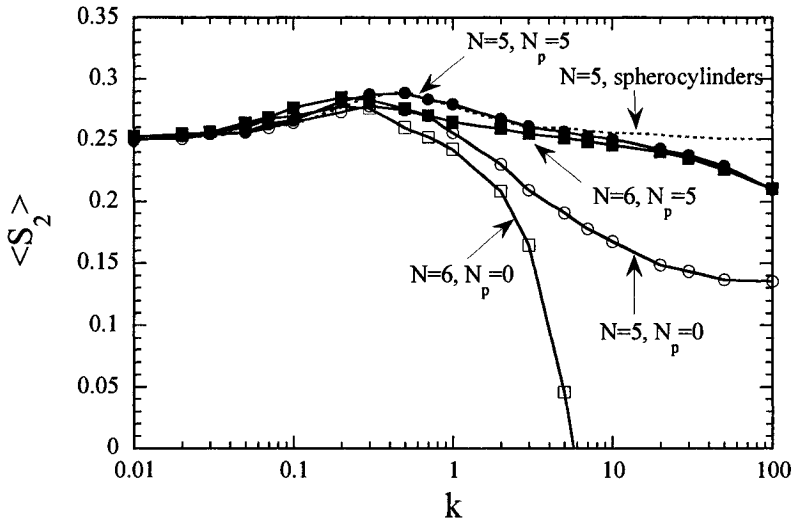


Fig. 4.5 Variation of the $< S_2 >$ with k for $N = 5$ and $N = 6$, for Shish-Kebab chains, compressed Shish-Kebab chains with $N_p = 5$ and spherocylinders.

For multiple rods, the local rod structure played an important role on orientation ordering as in the case of two rods. **Fig. 4.6** displayed the variation of order parameter $< S_2 >$ with the number of rods M for the Shish-Kebab chains for $N = 5$ and 6, and the compressed Shish-Kebab chains for $N = 5$ and $N_p = 5$ for $k = 2$. The order parameter of the Shish-Kebab chains was always smaller than that of the compressed Shish-Kebab chains because of the pronounced difference between their local structures. For the Shish-Kebab chains, the effect of the chain length parity diminished after the third rod was inserted. As M was increased, the order parameter decreased first and then

increased again after passing a minimum. There was a substantial overlap of order parameter for $N = 5$ and 6 in the range between $M = 3$ and 6. Such a result arose from the fact that for fewer rods, the non-parallel alignment, which was less sensitive to chain lengths, becomes more favorable. The increase of the order parameter with larger M could be attributed to the decrease of the rotational entropy of multiple rods. This argument was also applicable for the slight increase when N was increased from 5 to 6 because the longer chains experienced more excluded volume repulsions and suppressed the rotational entropy.

To help visualize the configuration of multiple rods, **Fig. 4.7** presented some of typical snapshots, for $M = 3$ in (a), $M = 4$ in (b), $M = 5$ in (c), at field strength k = 2. For M = 3, the three rods tilted and formed a triangle configuration. Such a configuration cancelled the orientation ordering. For M = 4, the four rods tended to form perpendicular arrangement to each other, which was unfavorable for orientation ordering. For M = 5, the configuration displayed the formation of a twisted bundle, and the orientation ordering was increased compared to $M = 3$ and 4.

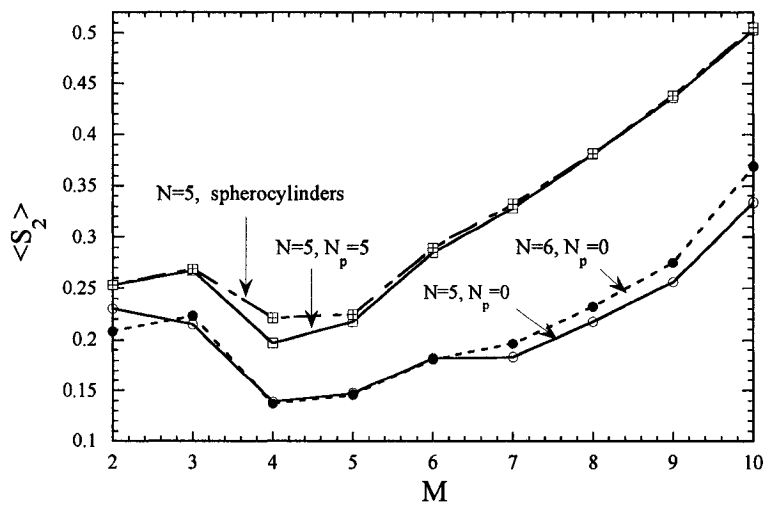
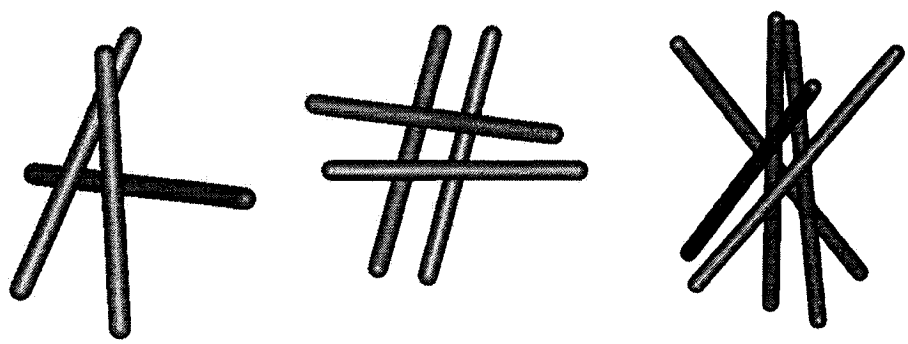


Fig. 4.6 Variation of the order parameter with the number of rods, for Shish-Kebab chains, compressed Shish-Kebab chains ($N_p = 5$) and spherocylinders, as marked, for $N = 5$ and $k = 2$.



(a) **(b)** **(c)**

Fig. 4.7 Snapshots of multiple rods at $k = 2$ for $M = 3$ in (a), $M = 4$ in (b), and $M = 5$ in (c).

4.5 Conclusions

The orientation ordering of two Shish-Kebab chains confined by harmonic potentials were investigated by Monte Carlo simulations and asymptotic analysis. The Shish-Kebab chains consisted of adjacent hard spheres aligned in the same axis. The first analysis was carried out to examine a single Shish-Kebab chain in a spherically harmonic potential. It was found that the interaction energy between the rod and the applied potential depended only on the position of the center-of-mass of the rod, not on its orientation. This was a non-trivial result because it showed that the spherically harmonic potential might smear out the angular information of single non-spherical objects. Also, the applied potential played no role on the rod orientation. Nevertheless, the external potential controlled the separation of rods, and affected the alignment of two confined rods through excluded volume interactions.

To characterize the orientation ordering, we follow the formulation suggested by Eppenga et al based on the middle eigenvalue of the ordering matrix in Eq. (4.3). The case for two uncorrelated rods were solved analytically, and a limiting value 0.25 for order parameter was obtained. With our best knowledge, such a limiting case has not been investigated systematically. When the applied potential was weak, the two rods were essential uncorrelated, and the order parameter approached this asymptotic value. For stronger fields, the angular dependent excluded volume affectd the orientation ordering of two Shish-Kebab chains. A probability distribution

function was defined to depict the angle range in which the two rods were free from collision (or overlap) in Eq. (4.15). This function enabled us to compute the order parameter for two strongly coupled rods of odd-numbered beads. It was found that the two Shish-Kebab chains of odd-numbered beads displayed an asymptotic value 0.131 for the order parameter. For the rods of even-numbered beads, the asymptotic value of the order parameter was -0.5. These results agreed well with the simulations at strong fields. For rods of odd-numbered beads and even-numbered beads, their center-of-mass were located at their central bead and central groove, respectively. At strong fields, the two rods of odd-numbered beads exhibited more rotational degree of freedom. However, the rods of even-numbered beads tended to form perpendicular configuration through intercalation of their central grooves. The dependence of chain length parity was attributed to the fact that these two types of chains could exploited different ways to reduce the repulsive energy from the applied potential.

In addition to the parity of chain length, the Shish-Kebab chains illustrated the importance of the local smoothness of rods on molecular alignment. In this work, the simulation results of Shish-Kebab chains were compared with those for compressed Shish-Kebab and spherocylinders. The compressed Shish-Kebab chains with extra non-interacting beads (without interactions from the applied potential) incorporated modified the local rod smoothness. The dependence of chain length parity diminished after extra non-interacting beads

were added to the Shish-Kebab chains. Meanwhile, the orientation ordering of compressed Shish-Kebab chains approached that of spherocylinders. The effects of local roughness of compressed Shish-Kebab chains emerged only when the applied field becomes strong enough.

The confining harmonic potential chosen in this project was a simple model to mimic non-rigid cavities. One of the primary conclusions was that the confining boundaries would enhance the orientational ordering of multiple rods. This result was consistent with the liquid crystal ordering of cell materials induced by cellular crowding, suggested by Hertzfeld.[138] This work also made a prediction regarding the minimum number of rods needed to form liquid crystal ordering for confined rod molecules in a non-rigid cavity. This finding might result in a new physical picture for those cellular macromolecules which lack liquid crystal ordering, such as microtubules.[139]

- Chapter 5 -

Theoretical Study of the Effects of Templated Materials on Aggregate Formation

5.1 Introduction

Synthesis of nanoparticles has attracted a great deal of attention over recent years because these novel materials exhibit potential applications to reduce the dimension of future devices. A variety of synthetic methods have been developed to refine the geometry and size of nanoparticles. Among available methodology, template syntheses have become widely used approaches, in which precursors were incorporated into pre-designed cavities at nano-scale, such as micelles or network gels.[140] The nanoparticles of different geometries, such as spherical or cylindrical nanoparticles, were obtained after the reaction of precursors within spherical or cylindrical micelles was initiated. The dimension of spherical nanoparticles could be adjusted via the interior cavity size of a micelle or a cross-linked gel.

More sophisticated techniques have recently been devised to manufacture non-spherical nanoparticles, such as AFM (atomic force microscopy). In the experiment, the monomers were carefully aligned and written on the surface of a substrate using the AFM tip, and rod-like nano-wires, up to 50 nm in length, were obtained after electrochemical polymerization conducted by

applying a voltage between the AFM tip and substrate.[141] To develop more tractable synthetic procedures, Tang and coworkers have devised a simple scheme in making CdTe nanowires by using self-organization of smaller nanocrystals.[142] In their experiment, they first prepared nanocrystals stabilized by thioglycolic acids. After removal of the stabilizer, thioglycolic acids, aggregation of smaller nanocrystals was initiated and nanowires were formed. The length and diameter of a nanowire can be tuned by controlling the rate and the amount of stabilizers removed.

Despite much progress, for the purpose of mass production, it would be instructive to search for some chemicals that can serve as templated materials and stabilizers simultaneously in a blend. Recently, Chauhan et al has successfully devised such a model system to fabricate nanoparticles by using the blends of Pd containing precursors and silicon containing polymers, polysiloxanes.[143] It has been shown that the geometry of Pd nanoparticles was sensitive to the chemical architecture of polysiloxanes, and can be in form of spheres or rods. Experimental evidence also discloses the dual nature of polysiloxanes as templated materials and stabilizers in Pd nanoparticle formation. This novel scheme provides a convenient route to synthesize stable nanoparticles of various geometries.

From theoretical perspectives, the mechanism of nanoparticle formation has not been systematically studied. In our view, the entire process could be divided into several stages, including pre-reaction, formation of small

nanocrystals, and large aggregation of nanocrystals. In the pre-reaction phase, precursors were mixed with templated materials. During the formation of nanocrystals, the tiny particles were dispersed and stabilized in the solution by stabilizers. As was observed for nanocrystal-nanowire transition by Tang et al,[142] removal of stabilizers resulted in the formation of necklace-like chains, the intermediate state, at the stage of nanocrystal formation. In the final stage, the desired nanoparticles were formed. The interactions between the reacting particles and template materials (or stabilizers) could be crucial to determining the final forms of nanoparticles, but theoretical understanding of the influences of templated polymers on nanoparticle formation was still lacking.

In this work, Monte Carlo simulations were conducted to investigate a greatly simplified two-dimensional lattice model for blends of smaller reacting particles and linear polymers, composed of three species: homopolymer chains, reacting particles and solvents. In the model, chain molecules were utilized to mimic templated polymers. The reacting particles could be viewed as precursors before the reaction was triggered in the blend, or metal atoms (or smaller nanocrystals) before the formation of larger aggregates. The excluded volume of particles was taken into account so that no two particles were allowed to occupy at the same site. The reacting particles interacted with the monomers of templated polymers through neighboring short-ranged interactions when they were located at neighboring sites, in addition to the excluded volume interaction. Both attractive and

repulsive short-ranged interactions were considered. In the simulations, the mean chain size was computed as a function of the strength of the short-ranged interaction. Furthermore, the probability distribution of different sizes of clusters aggregated from smaller reacting particles were considered in the presence of templated polymers and templated monomers to reveal the influence of the chemical architecture of templated materials on aggregation. Despite the simplicity of this model, the results could serve as the very first step to target on the theoretical investigation of nanoparticle formation in a blend containing templated materials. This study represented the thermodynamic limit of physical aggregation mediated by templated materials.

5.2 Lattice Monte Carlo Simulations

In this work, a greatly simplified model was considered for a blend composed of three types of species: flexible polymers (to model template materials or stabilizers), reacting particles (to model precursors, or metal atoms (or smaller nanocrystals), and solvents, to mimic the reaction system in the polymer template synthesis. Also, templated polymers (containing chain connectivity through monomers on a polymer molecule) were compared with templated monomers (containing no chain connectivity) to elucidate the effect of chain length of templated polymers on aggregation of smaller particles in the blend. Note in the rest of the manuscript, templated monomers were referred to the monomeric templated materials (as opposed to templated

polymers), but monomers had a collective meaning which referred to a unit of templated materials (i.e., a monomer in a templated polymer or a templated monomer itself). This model could be viewed as the blend of precursors and template materials in the pre-reaction phase, or the mixtures of metallic atoms (or smaller nanocyrstals) and stabilizers before further aggregation to larger nanoparticles. In the model, monomers were placed on the squared lattice points, and the neighboring monomers of a chain molecule were connected by a chemical bond whose length was set to be the distance of two neighboring lattice points. Also, each precursor or solvent particle occupied one lattice point; namely, particles of different species had the same size as monomers. To account for the excluded volume of particles, placing two particles at the same lattice site was forbidden. In addition to the excluded volume interaction, it was assumed that the monomer and reacting particle interact through a short-ranged neighboring interaction of strength equal to E. A dimensionless parameter $\varepsilon = E/kT$ was used in the calculations, where k was the Boltzmann constant and T was the temperature. The simulation cell consisted of 20×20 sites, and two different chain lengths N were investigated: $N = 8$ (templated polymers) and 1 (templated monomers). The total particle density (or volume fraction), including monomers and reacting particles, was set to be 0.8 (i.e., 80% of the total lattice sites were occupied). A periodical boundary condition was employed to account for the finite density. Here, three different compositions with ratio γ (the number of monomers to that of reacting

particles) equal to 3:1, 1:1 and 1:3 were investigated. Simulations for such a dense system were quite difficult, but the simplified two-dimensional model reduced intense computation, and enabled us to explore the fundamental physics for this complex system.

In the simulations, reptation move and random move were combined to relax chain conformation and the positions of reacting particles, respectively.[144] In the reptation move, a monomer was cut at either end, and was regrown on the other end. In the random move, a randomly chosen reacting particle was moved randomly across the simulation cell based on a step size, i.e., the maximum displacement of the particle on one dimension. In the limiting case, when templated polymers were substituted with templated monomers, reptation move was replaced with random walk for templated monomers. To test the simulation results, the statistical weights of the above two types of moves and step sizes were varied, and the results were found to be independent of these parameters. The Metropolis criterion was used to determine whether an attempted move was accepted,[145] and the acceptance ratio of the simulations was 20-40 % depending on the interaction strength ε ranging from –2 to 2) and composition.

In the simulations, mean squared end-to-end distance $<R^2>$ was used to characterize average chain conformation. The common approach to obtain the cluster size from simulations was based on a chosen threshold distance between smaller particles as the criterion to define a cluster. However, such a

calculation encountered percolation problems, and the thus obtained cluster sizes were sensitive to the chosen threshold distance and the size of the finite simulation cell.[146] Hence, in our simulations, a distribution function $P(m)$ was resorted to, which was defined as the probability of finding m adjacent reacting particles aligned along the same axis, as an alternative to address the issue related to the aggregation probability of smaller reacting particles in a blend. This function provided the information to rationalize the physics behind aggregate formation, whose physical picture was given in the following section. An equilibrated configuration (new particle locations and chain conformations) was sampled every 200 moves and a total of 5×10^7 moves were used for each simulation. The number of moves skipped was chosen arbitrarily and our tests showed that the results were independent of the chosen number. The mean properties were obtained from averaging over 2.5×10^5 equilibrated configurations.

5.3 Results and Discussion

5.3.1 Chain conformation of templated polymers

First, the effects of the interaction strength between reacting particles and templated polymers were considered, as well as the blend composition on chain conformation. **Fig. 5.1** plotted the variation of the mean squared end-to-end distance of templated polymers as a function of the coupling strength ε of the short-ranged interaction between a monomer and a reacting particle for

different compositions γ, 1:3 (□), 1:1 (○) and 3:1 (△). When the interaction was repulsive (i.e. $\varepsilon > 0$), the reacting particles tended to collapse the chain conformation of the templated polymers. In contrast to the conformation for $\varepsilon = 0$, the mean chain size decreased and became leveled-off roughly after ε was increased to 1. For attractive interactions ($\varepsilon < 0$), the trend was opposite, and chain dimension increased when the magnitude of the coupling strength was increased. Moreover, the results showed that as the composition of reacting particles was increased (from $\gamma = 3{:}1$ to 1:3), mean chain size further decreased and increased for repulsive interactions and attractive interactions, respectively. The conformational deformation induced by the short-ranged interaction could be understood as follows.

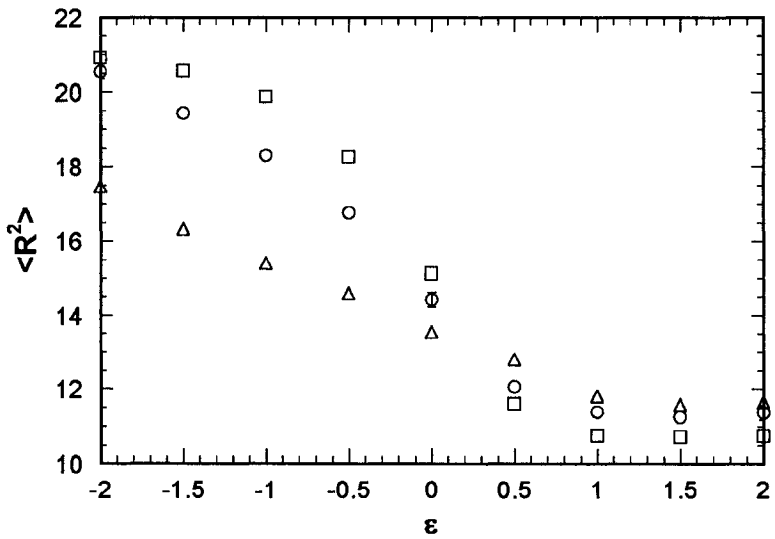


Fig. 5.1 Variation of the mean squared end-to-end distance as a function of the coupling strength ε between monomers and reacting particles, for monomer-reacting particle ratio γ = 1:3 (□), 1:1 (○) and 3:1 (△).

Under a repulsive short-ranged interaction, contraction of a polymer chain reduced the total repulsions arising from reacting particles. However, when this short-ranged interaction became attractive, the "sticky" reacting particles could bridge either intramolecular or intermolecular monomers. In the former case, a polymer chain was expected to contract. On the contrary, if intermolecular bridging was to occur, chain dimension would expand because of the overall excluded volume repulsion between two bridged polymer molecules. The simulation results showed that as $\varepsilon < 0$, the mean chain size of templated polymers was greater than that of chain molecules at ε = 0, indicating that intermolecular bridging were more favorable than intramolecular one because elongation of a chain molecule increased the total contact area between chain molecules and reacting particles. Note a similar behavior has also been observed in Chapter 1 when a chain segment was bound to the other segment. In terms of conformational entropy, intermolecular bridging resulted in more favorable elongated coiled chains than the collapsed globular conformation induced by intramolecular bridging. Furthermore, when the composition of reacting particles was increased (i.e., from γ = 3:1 to 1:3), the deformation of chain conformation became more

significant because templated polymers experienced more interactions from surrounding reacting particles.

5.3.2 Distribution of adjacent reacting particles along the same axis

In this subsection, the blend of templated polymers and reacting particles were investigated, and the results were summarized from **Fig. 5.2** to **Fig. 5.5**.

The definition of the distribution function of $P(m)$ was demonstrated in **Fig. 5.2**, a typical simulation snapshot for the blend of templated polymers and reacting particles for ε= -2, in which solids spheres denoted the monomers of templated polymers and open squares denoted reacting particles. Note blank sites were occupied by solvent particles. Owing to the strong attractive interaction between a monomer and a reacting particle, these two types of particles tended to mix together. Within the circle of **Fig. 5.2** were five reacting particles located. According to our definition of $P(m)$, these five reacting particles produced 3 possible ways to arrange two adjacent reacting particles in the same axis, i.e. m = 2 (2 in x-direction and 1 in y-direction), and one possible way to arrange three consecutive adjacent reacting particles in the same axis, i.e., m = 3. This distribution function $P(m)$ was essentially the probability of finding m adjacent reacting particles along the same axis, whose physical meaning will be elaborated below.

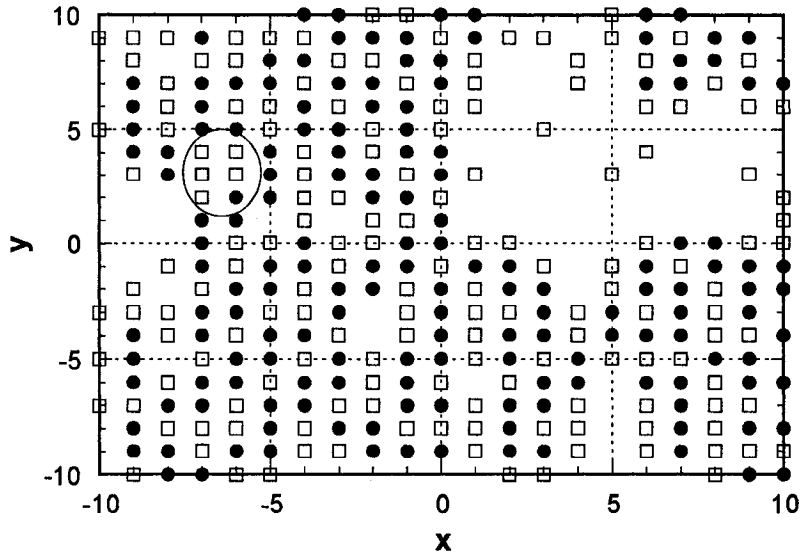


Fig. 5.2 Illustration of the definition of $P(m)$, a typical simulation snapshot for the blend of templated polymers and reacting particles for $\varepsilon = -2$, in which solids spheres (●) denote sites taken by the monomers of templated polymers and open squares (□) denote reacting particles

Fig. 5.3 displayed the plot of the distribution function $P(m)$ (in log scale) for different coupling strengths for γ = 1:1, as marked, where m was the number of consecutive adjacent reacting particles aligned in the same axis; lines were obtained from the linear regression of $log(P(m))$. It was found that, generally, for a larger m ($m \geq 3$), the magnitude $P(m)$ decreased as the interactions strength ε is decreased from positive to negative values. When $\varepsilon >$

0, the value of m varied in a wide range, and the maximum m could even be greater than the side length of the simulation cell. In other words, the cluster size of aggregates obtained from such a system displayed a broad distribution. When the short-ranged interaction was turned off, at $\varepsilon = 0$, $log(P(m))$ decreased linearly as m was increased, and for a larger m ($m \geq 3$), the magnitude of $P(m)$ was smaller than that for $\varepsilon > 0$. For attractive short-ranged interactions ($\varepsilon < 0$), the calculated distribution function $P(m)$ were smaller than that of $\varepsilon = 0$, indicating that the system favored smaller clusters; namely, reacting particles became more dispersed in the blend.

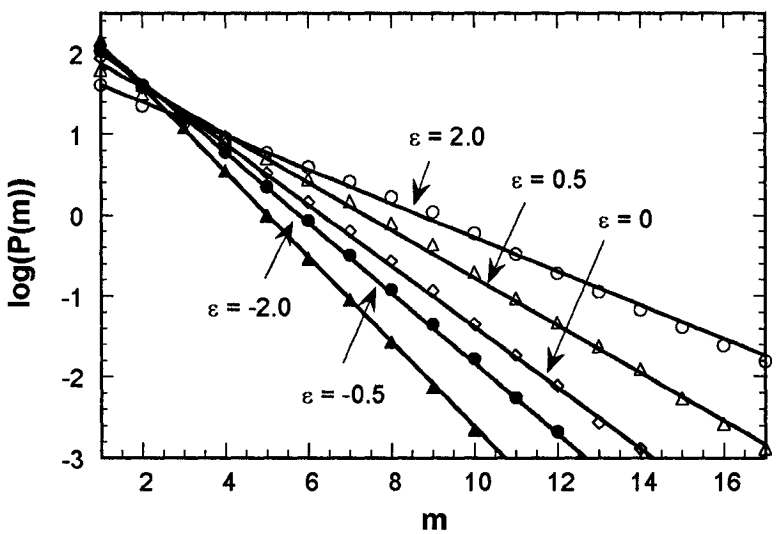


Fig. 5.3 Plot of $P(m)$ (in log scale) for different coupling strengths, for monomer-reacting particle ratio γ = 1:1, as marked.

Moreover, as shown in **Fig. 5.3**, the linear behavior of $log(P(m))$ against m indicated that the distribution function $P(m)$ follows the exponential distribution, i.e., $P(m)$~constant $\times$ $10^{-\kappa m}$ where κ was the probability strength of **NOT** finding a neighboring reacting particle.[147] In the investigated system, $P(m)$ could be correlated with $P(m + dm)$, the probability of finding m and $m + dm$ adjacent particles aligned along the same axis, respectively, by

$$P(m + dm) = (1 - \kappa m)\, P(m) \qquad (5.1)$$

in which $(1-\kappa m)$ was the probability of placing dm particles next to an aggregate with m adjacent particles aligned along the same axis.

The exponential decay of $P(m)$ indicated that the nature of a random process in formation of clusters.[147] **Fig. 5.4** plotted the variation of κ, obtained from the linear regression of $log(P(m))$ in **Fig. 5.3**, as a function of ε. As ε was decreased from a positive to a negative value, the parameter κ increased; namely, the slope of $\log(P(m))$ vs. m became greater. This result suggested that the probability of finding m adjacent reacting particles aligned along the same axis decreased. Note that the number of isolated reacting particles increased because of their participation in bridging different chain molecules; as a result, reacting particles became more dispersed in a blend. Also, our results suggested that an attractive interaction may screen out larger clusters (like filters), such that the possibility of finding m reacting particles adjacent to each other along the same axis was decreased.

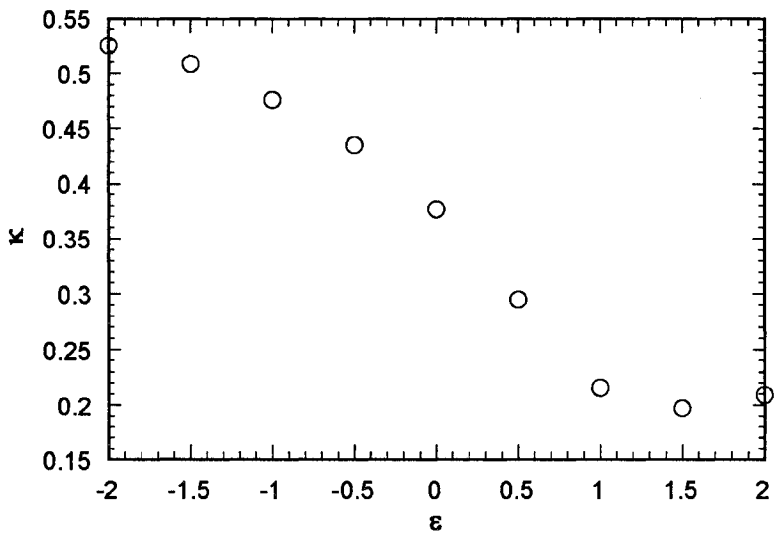


Fig. 5.4 Variation of exponent κ (obtained from **Fig. 5.3**) as a function of ε for monomer-reacting particle ratio γ = 1:1.

Although κ in **Fig. 5.4** roughly decayed monotonically, it was noticed that a slight non-monotonic behavior appears near $\varepsilon = 2$, i.e., κ decreased first and then increased again after passing the minimum near $\varepsilon = 1.5$. To test such a deviation, in **Fig. 5.5**, $log(P(m))$ was plotted in the presence of templated polymers for different compositions, as marked, for $\varepsilon = -1$ in (a) and $\varepsilon = 1.5$ in (b); lines were obtained from the linear regression of $log(P(m))$. These results suggested that in both cases, increase of γ (i.e., more templated polymers) resulted in the reduction of larger aggregates and facilitates the dispersion of

smaller reaction particles in a blend. For attractive interactions (e.g., ε = -1), the linear behavior of $log(P(m))$ persisted for different compositions γ. However, for repulsive interactions (e.g. ε = 1.5), the deviation of $log(P(m))$ from the linear behavior became significant as the composition of reacting particles was increased (from γ= 3:1 to 1:3) .

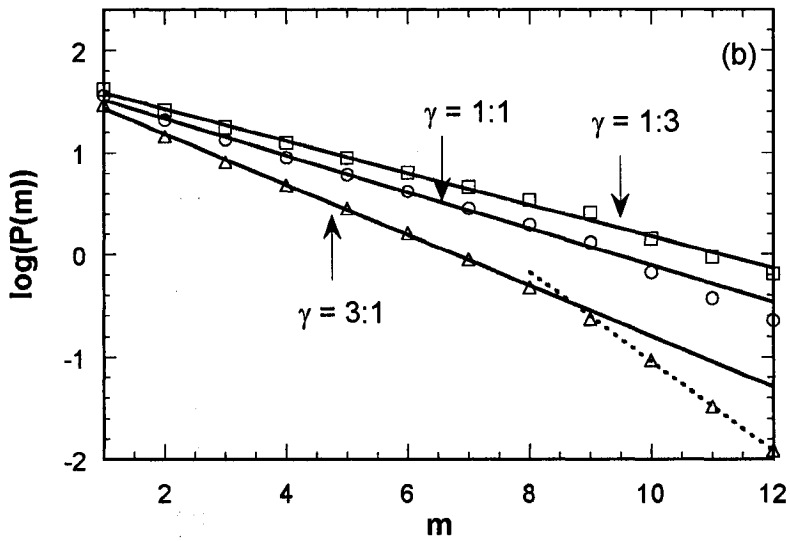


Fig. 5.5 Plot of $P(m)$ (in log scale) in presence of templated polymers for different compositions, as marked, for ε = -1 in (a) and ε = 1.5 in (b); lines were obtained from linear regression.

Incorporation of attractive interactions between two different species favored their miscibility, and the linear $log(P(m))$ in **Fig. 5.5(a)** indicated that

the system was quite homogeneous because the aggregate size of reacting particles followed an exponential distribution function, due to spatial randomization of reacting particles in the blend. On the other hand, **Fig. 5.5(b)** showed that the $P(m)$ for $\gamma = 3{:}1$ was smaller than the other two compositions, and the non-linear $log(P(m))$ could roughly be divided into two regimes, each of which being fitted with a linear regression equation (solid and dotted line). These results might be attributed to some extent of segregation and immiscibility induced by the repulsive interactions between monomers and reacting particles. Since such an interaction could induce phase separation between two different species in liquid,[148] we speculated that the formation of aggregates in reacting particle-rich (polymer-deficient) domain was different from that in reacting particle-deficient (polymer-rich) domain. If these domains were small enough, then reacting particles may form nanoparticles. According to these results, useful applications could be developed to tune the size of nanoparticles through adjusting the interactions between monomers and reacting particles as well as the composition of these two species.

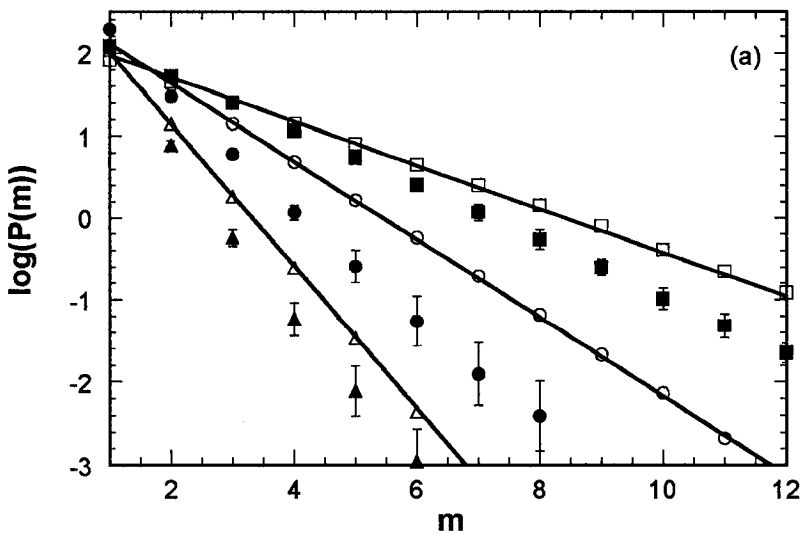
5.3.3 Aggregation in presence of templated monomers

To better understand the effects of the chain length of templated polymers on the formation of aggregates of reacting particles, a limiting model was investigated, in which templated polymers were replaced by templated monomers (without chemical connectivity between monomers). Such a model

reduced to a simple mixture consisting of two types of spheres interacting through the excluded volume interaction and a neighboring interaction of strength ε.

Fig. 5.6 compared the simulated $P(m)$ in the presence of templated polymers, denoted by open symbols, and of templated monomers, denoted by solid symbols, respectively, for different compositions γ, as marked, for ε = -1 in **(a)** and for ε = 1.5 in **(b)**; lines were obtained from **Fig. 5.5** for templated polymers. Note that the statistical errors of data points without error bars displayed were less than the size of symbols. In **Fig. 5.6(a)**, when the monomer-reacting particle ratio was less than and equal to one (γ = 1:3 and 1:1), the plot of $log(P(m))$ of templated monomers against m was nearly linear, but their magnitude was smaller than that of templated polymers. This result indicated that reacting particles were more miscible with templated monomers than with templated polymers when the interaction between the monomer and reacting particle was attractive. Actually, templated monomers (of coordination number equal to 4) could fully interact with reacting particle whereas templated polymers were constrained by chain connectivity and had a smaller total contact area to interact with reacting particles (on the basis of per monomer). As the monomer-reacting particle ratio γ was increased from 1:3 to 3:1, the simulated $P(m)$ between templated polymers and templated monomers showed less difference. This result suggested that the composition of templated materials might be more important than the effect of their chemical

architectures on the cluster formation. Also, it was noticed that the $log(P(m))$ of templated monomers slightly deviated from linear behavior for large attractive interactions (such as $\gamma = -1$), and the deviation became more obvious as γ was increased from 1:1 to 3:1. This deviation was not seen in the presence of templated polymers, which might be attributed to the formation of stoichiometric complex between a reacting particle and the templated monomers.



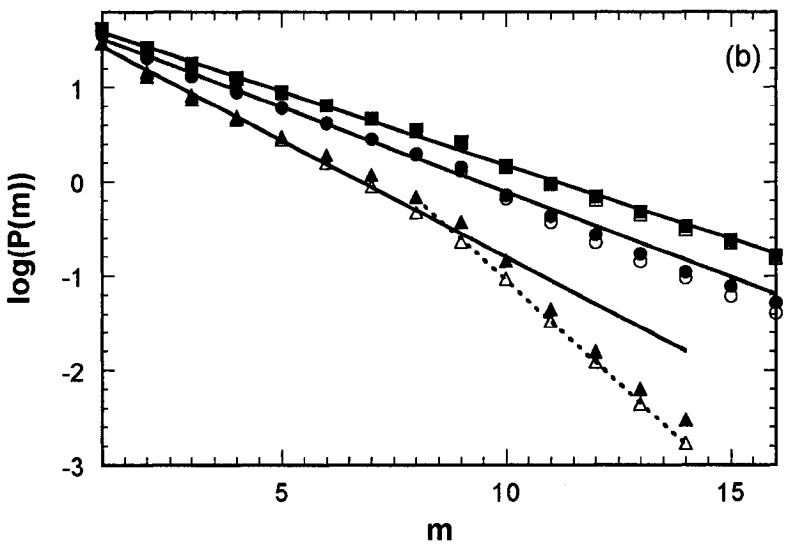


Fig. 5.6 Plot of $P(m)$ (in log scale) in the presence of templated polymers, denoted by open symbols, and templated monomers, denoted by solid symbols, for different monomer-reacting particle ratio γ = 1:3 (□/■), γ = 1:1 (○/●) and γ = 3:1 (△/▲), for ε = -1 in (a) and ε = 1.5 in (b).

When the interaction between the monomer and reacting particle was repulsive, the plot of $log(P(m))$ was found to deviate from linear behavior significantly as the composition of template materials (for both templated polymers and templated monomers) was increased. This result was probably due to the segregation of templated materials from reacting particles to avoid repulsions. Moreover, in **Fig. 5.6(b)**, the difference of $log(P(m))$ between

templated monomers and templated polymers was small for all γ (compared to **Fig. 5.6(a)**), indicating that the simulated $P(m)$ became less sensitive to the chemical architecture of templated materials in the presence of repulsive interactions. The $P(m)$ of templated monomers had a tendency to be slightly greater than that of templated polymers for larger m, especially when γ was large enough (e.g. 3:1). This result was consistent with the picture that the templated monomers had a larger interaction area per monomer and interacted with reacting particles slightly more strongly.

The results in this subsection could be summarized as follows. The effect of interactions on the cluster formation was similar for both templated polymers and templated monomers. When the effective interaction between the reacting particle and templated material was attractive ($\varepsilon < 0$ in **Fig. 5.6(a)**) the miscibility of reacting particles and templated materials increased because the probability that multiple reacting particles were simultaneously located in neighboring positions on the same axis decreases. On the contrary, the repulsive interactions between two different species tended to induce phase segregation, and the probability of forming larger aggregates was increased (e.g. comparison of **Fig. 5.6(a)** and **(b)** for m = 12 between attractive and repulsive interactions). In addition to interactions, the chemical architecture (i.e., with or without chain connectivity between monomers in our model) affected the effective interacting area between the templated materials and reacting particles, and templated monomers interacted with reacting particles

more effectively, particularly in the presence of attractive interactions. Hence, templated monomers enhanced the dispersion of reacting particles in a blend under such a condition. Furthermore, for strong enough attractive interactions, templated monomers had a greater tendency to form a stoichiometric complex with reacting particles (than templated polymers with larger excluded volume). As a result, the random nature of aggregate formation diminished, and the $log(P(m))$ of templated monomers in **Fig. 5.6(b)** deviated from the linear behavior, compared to the linear $log(P(m))$ of templated polymers. For repulsive interactions, the effect of chemical architecture on aggregation became less significant because segregation was more favorable to lower the energy of the blend. Moreover, our findings showed that the composition of templated materials should also be a crucial variable to control the size of aggregates. As the templated material composition was increased, the probability of forming larger aggregates decreased because reacting particles were more dispersed in the blend.

So far, only the important physics of aggregate formation elicited from the behavior of $P(m)$ was analyzed. Actually, this simple function also provided technical advantages to rigorously test the convergence of the simulated cluster sizes. In **Fig. 5.6(a)**, for γ = 3:1, the $log(P(m))$ of templated monomers displayed larger error bars than that of templated polymers, and the magnitude of the error bars increased as m was increased. The larger statistical errors might arise from local energy minima in the presence of attractive interactions

between two different species.[149] In **Fig. 5.7**, such a situation was illustrated with two configurations: (a) three reacting particles were separated, and two of them were surrounded by three templated monomers and one was surrounded by two templated monomers; (b) three adjacent reacting particles aligned in the same axis and surrounded by templated monomers. The energy of the two configurations was the same. Nevertheless, **Fig. 5.7(a)** was more entropically favorable (more different ways to arrange them in a simulation cell), which partially accounts for why smaller clusters (i.e., smaller m) were more populated. The configuration in **Fig. 5.7(b)** occurred infrequently in the simulation, but was more likely to be trapped during simulation when it emerged, because its energy was as low as that in the configuration of **Fig. 5.7(a)**. This type of configurations would increase the computational time to obtain equilibrated cluster sizes in the simulation. Actually, it was found that the rate of convergence can be accelerated if templated monomers were sampled more often to facilitate the system to escape from such a configuration in **Fig. 5.7(b)**. Moreover, when templated polymers were incorporated into the simulation, the simulated $P(m)$ achieved convergence more rapidly, and produced a smaller statistical error, because the configuration shown in **Fig. 5.7(b)** was impeded due to the stronger excluded volume interaction between chain molecules.

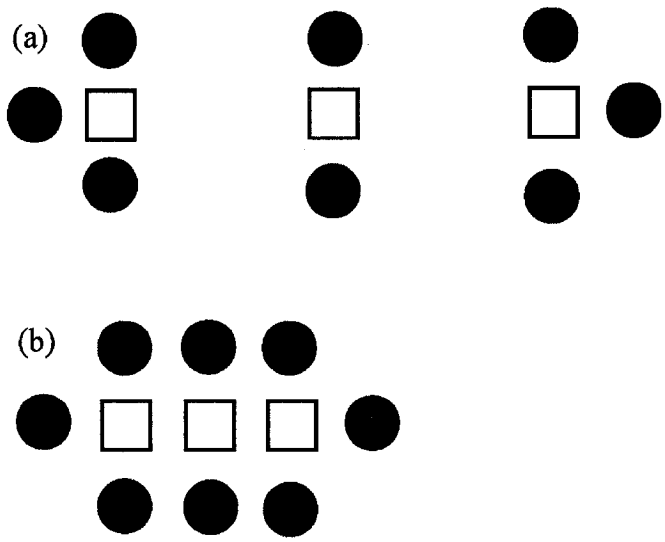


Fig. 5.7 Comparison of two illustrated configurations: (a) three reacting particles were separated, and two of them were surrounded by three templated monomers and one was surrounded by two templated monomers. The energy of the two configurations was the same; (b) three adjacent reacting particles aligned in the same axis and surrounded by eight templated monomers, where monomers were denoted by solid spheres (●) and reacting particles were denoted by open squares (□).

5.3.4 Effects of boundary condition of the simulation cell on polymer mediated aggregation

To understand the effects of the boundary conditions of a simulation cell on the simulated properties, the results under periodical boundary condition were compared with those under the closed boundary condition (confined by

hard walls). In the simulations, the closed boundary containing hard impenetrable walls were employed to confine all the particles within a simulation cell of 20 × 20 lattice sites. **Fig. 5.8** plotted the variation of the mean squared end-to-end distance of templated polymers as a function of the coupling strength ε for periodic boundary condition, denoted by open symbols, and for closed boundary condition, denoted by solid symbols, for different compositions γ, 1:3 (□/■), 1:1 (○/●) and 3:1 (△/▲). Note that the simulation error bars were about the size of symbols. The mean end-to-end distance of polymer chains under closed boundary condition was slightly smaller than their counterparts under periodic boundary condition, indicating that the closed boundary tended to compress chain molecules, consistent with the theoretical predictions by Doi and Edwards.[150]

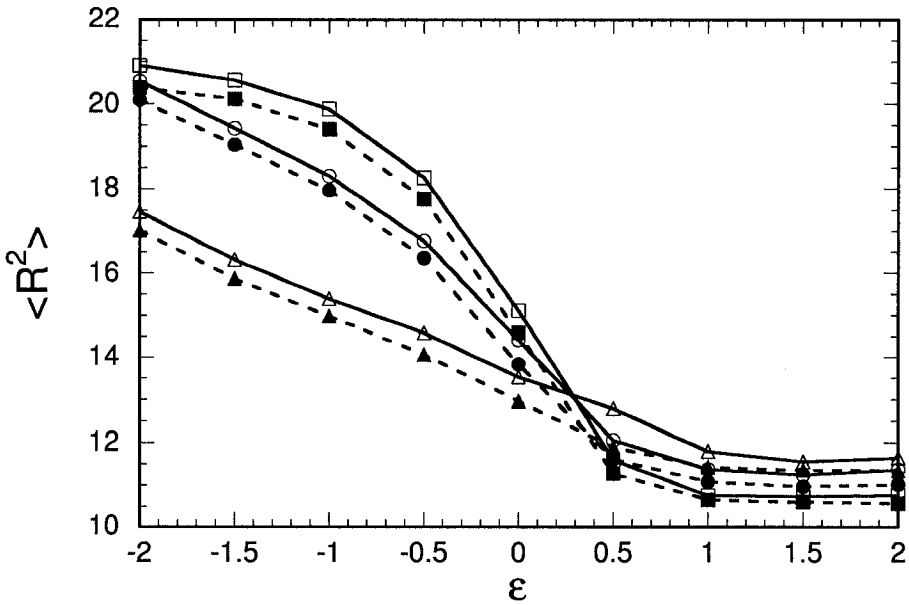


Fig. 5.8 Variation of the mean squared end-to-end distance of templated polymers as a function of the coupling strength ε for periodic boundary condition, denoted by open symbols, and for closed boundary condition, denoted by solid symbols, for different compositions, 1:3 (□/■), 1:1 (○/●) and 3:1 (△/▲).

Fig. 5.9 compared the simulated $P(m)$ in the presence of templated polymers under periodic boundary condition, denoted by open symbols, and under closed boundary condition, denoted by solid symbols, respectively, for different compositions γ and interaction parameter ε, as marked. The difference of $P(m)$ under different boundary condition was quite small, probably because the simulation cell size (20 × 20) was much larger than the

average chain size, and the hard walls did not significantly alter the polymer and reacting particle density profiles in the simulation cell. Nevertheless, the minor difference could be differentiated in the simulations. Under the closed boundary condition, $P(m)$ followed the exponential decay function as in the cases under periodical boundary condition, but the probability of larger aggregates was slightly enhanced compared to that under periodic boundary condition. The reasoning was twofold: 1) the decrease of polymer chain size under confined boundaries reduced the dispersion of smaller reacting particles among polymer chains, and enhanced the aggregation probability of reacting particles; 2) the expulsion of the polymer from the boundary created space for reacting particle to aggregate near the walls. To understand the effects of confinement induced by closed boundaries, the spatial density distribution functions for reacting particles and polymers were computed. **Fig. 5.10** displayed the spatial density profile along one dimension for reacting particles and the monomers of templated polymers, as marked, for γ = 1:3 and ε = -1; lines were meant to guide our eyes. The reacting particles were found to have a greater tendency to accumulate near the walls (x = -10 and 9), but the polymers were depleted in this regime. Note that for other γ and ε, the density profiles were qualitatively the same. This was due to the segregation induced by entropic packing as pointed out by Vakarin.[151]

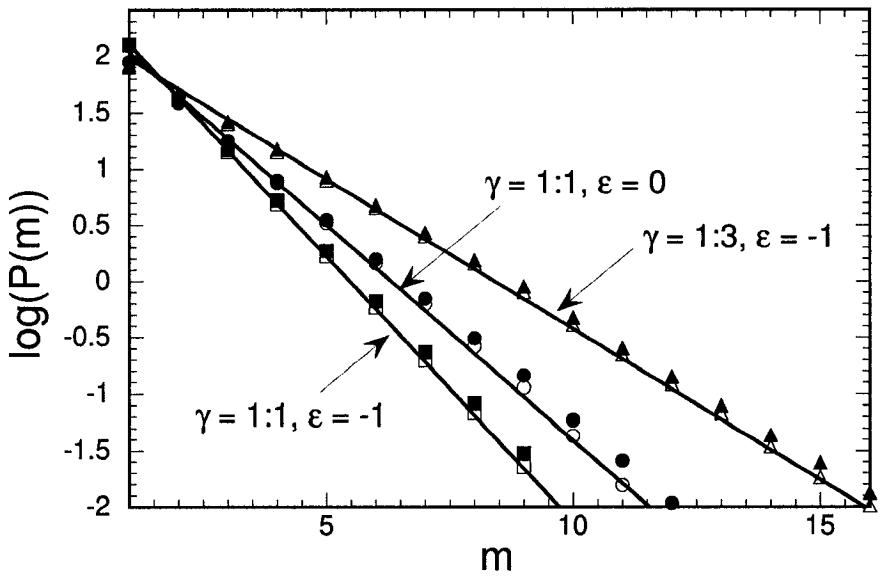


Fig. 5.9 Comparison of the simulated $P(m)$ in the presence of templated polymers under periodic boundary condition, denoted by open symbols, and under closed boundary condition, denoted by solid symbols, respectively, for different compositions γ and interaction parameters ε, as marked. Lines were linear regression for open symbols (obtained from **Fig. 5.5(a)**).

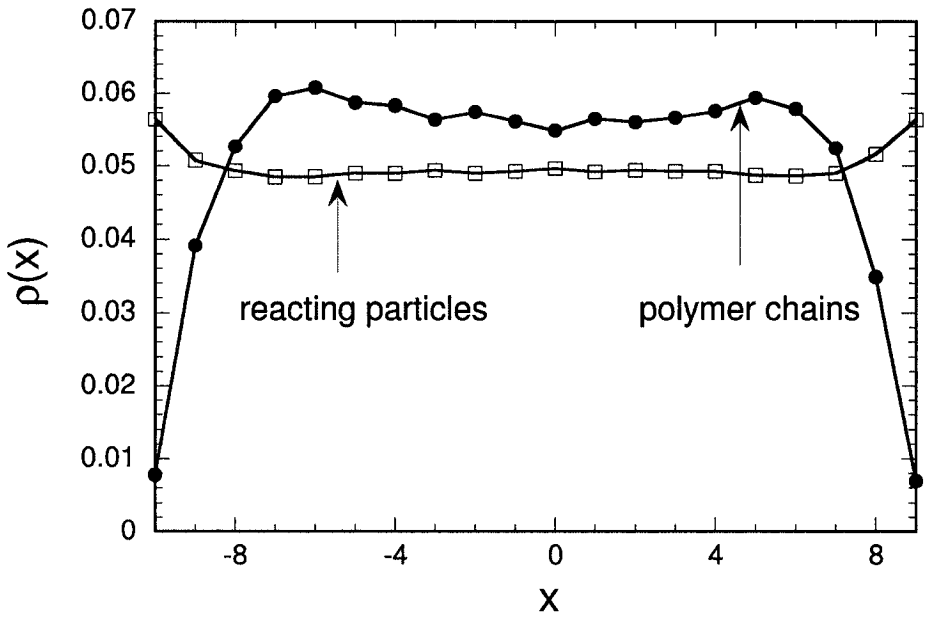


Fig. 5.10 Spatial density profile along one dimension for reacting particles and the monomers of templated polymers, as marked, for γ = 1:3 and ε = -1; lines were meant to guide our eyes.

5.4 Conclusions

Monte Carlo simulations have been carried out to investigate a greatly simplified model composed of templated materials, reacting particles and solvents on a two-dimensional lattice, aiming at understanding the role of templated polymers on the aggregation of reacting particles in the thermodynamic limit. The simulations allowed us to systematically investigate the effects of the interactions between templated materials and reacting particles and the composition of templated materials on physical aggregation

of smaller reacting particles. Furthermore, the simulation results were compared between templated polymers and templated monomers to disclose the influence of chemical architecture of templated materials on the aggregation of reacting particles.

Our findings showed that for repulsive interactions between templated polymers and reacting particles, chain conformation contracted to reduce repulsions. For attractive interactions, chain molecules elongated to maximize the attraction between templated polymers and reacting particles. Our results also implied that a "sticky" reacting particle induced intermolecular bridging rather than intramolecular bridging, probably due to conformational entropy. Also, in the simulations, the composition of templated polymers was varied. By increasing the composition of reacting particles, the deformation of chain conformation became more pronounced under stronger interactions arising from surrounding reacting particles.

To better understand the aggregation of reacting particles, the probability distribution function $P(m)$ (of finding m adjacent reacting particles aligned along the same axis) was employed. This probability distribution was found to follow the exponential distribution, i.e. $P(m) \sim 10^{-\kappa m}$ if the aggregation of reacting particles was proceeded randomly, where κ had a physical meaning equivalent to the probability strength of **NOT** finding an adjacent reacting particle to an aggregate; namely, $log(P(m))$ was a linear function against m. Generally, this function was linear for most cases. When the interaction

between monomers and reacting particles was repulsive, the plot of $log(P(m))$ deviated from the linear behavior as the composition of templated materials was large due to segregation of templated materials and reacting particles. For attractive interactions, the linear behavior of $log(P(m))$ in the presence of templated polymers persisted for different compositions γ, but deviations from linear behavior were observed in the presence of templated monomers for large γ. Such deviations might be attributed to the formation of stoichiometric complex between a reacting particle and the templated monomers. From technical perspectives, this simple function $P(m)$ could also be employed to rigorously test the convergence of different cluster sizes in the simulation.

When the effective interaction between reacting particles and templated materials was attractive, the miscibility of reacting particles and templated materials was increased because the probability that multiple reacting particles were simultaneously situated in neighboring positions along the same axis decreases. In contrast, the repulsive interaction between templated materials and reacting particles resulted in some extent of segregation, which facilitated the aggregation of reacting particles. The chemical architecture (i.e., chain connectivity at the level of our model) affected the effective interacting area between templated particles and reacting particles. Templated monomers interacted with reacting particles more effectively due to the larger interacting area per monomer. Thus, templated monomers enhanced the dispersion and segregation of reacting particles in the blend for attractive and repulsive

interaction, respectively, compared to templated polymers (especially for large γ). In addition to the above adjustable variables including interaction and chemical architecture, the composition of templated materials was a crucial variable to control the size of aggregates. As templated material composition was increased, the probability of forming larger aggregates decreased because reacting particles were more dispersed in the blend. These predictions could be tested by future experimental results.

In summary, a simple model was investigated in this project to examine the "living" aggregation mechanism of nanoparticle formation using Monte Carlo simulations. The "living" aggregation was referred to as the cluster formation under a quasi-equilibrium condition, in which the resulting cluster size distribution followed an exponential distribution. The simulation results showed that the presence of templated materials affected the mechanism of living aggregation. When the templated materials were composed of longer polymer chains, the aggregate size tended to decrease. This finding was consistent with the fundamental principle of templated polymer synthesis of nanoparticle formation, where templated polymers were utilized to manufacture smaller nanoparticles. However, the predicted size distribution of aggregates was inconsistent with the observed particle size distribution in the work of Chauhan et al[142]. Such an inconsistency may be attributed to the non-living mechanism in their work. Moreover, the other possibility was due to the limitation of the TEM technique used in particle size measurement. The

method might not able to detect the smaller nanoparticles. As a result, the distribution became non-exponential in the experiments because of instrumental cut-off.

Concluding Remarks

In this dissertation, the static properties of polymers under various external fields were investigated using Monte Carlo simulations, aiming to understand the response of polymers to external potentials. The main conclusions for this study were summarized as follows.

- *Monte Carlo simulations for a fluctuating sphere labeled on a flexible polymer chain in good solvent*

  The fluctuating sphere expanded the chain dimension when labeled at one chain end, due to excluded-volume interactions. The chain expansion was sensitive to the intrinsic characteristics of the fluctuating sphere. Generally, softer and larger fluctuating sphere tended to expand the chain more significantly. However, the scaling law of the polymer size with respect to chain length was essentially not affected by the attached fluctuating sphere. Meanwhile, the polymer chain suppressed the vibration of the fluctuating sphere due to chain-induced confinement, suggesting that the spectroscopy of a vibrational probe labeled onto a polymer chain may be altered. Both coiled chain conformations and local grooves in a polymer chain could contribute to the confining environments that trapped the fluctuating sphere label. An increase of the chain length enhanced these confinement effects. Moreover, the fluctuating sphere model was employed to fit a flexible chain polymer model. The fluctuating sphere

model provided a means to parameterize a polymer chain into a flexible dumbbell, which could be employed for further studies of polymer dynamics.

- *Theoretical studies of conformational behavior of chain molecules containing polar groups and electric fields effects*

Our simulations of the atomistic PVDF model showed that the dipolar interactions within a PVDF molecule were essentially attractive. When dipolar interactions were increased, mean polymer chain size decreased, favoring globular conformational state. Meanwhile, the conformational transition of a PVDF chain with temperature was sensitive to the chain length. For very short chains, the PVDF displayed conformer behavior. As the chain length was increased the conformational transition became discrete, and eventually, it changed to continuous transition at long enough chain lengths. Such a result might link with the effective local rigidity of a PVDF.

Under electric fields, our simulations showed a polar polymer tended to elongate, for the PVDF model (with explicit atoms incorporated) and the polyampholyte-like model (with methylene and fluoro-methylene groups coarse-grained to alternating opposite charged sites). This finding also indicated that the electric field might be utilized to tune the local rigidity of a polar polymer. In the weak-coupling limit when monomer-monomer interactions were small, the mean chain sizes of the above two

models agreed well in weak fields. These results also were in agreement with theoretical predictions. However, the two models showed pronounced differences at strong fields, including extent of elongation and chain orientation. This difference was attributed to the subtle differences in the spatial arrangement of charged sites within a monomer in the two models. The results suggested that the conformational properties of a chain molecule became more sensitive to its local chemical structure under strong electric fields. These findings provide guidelines for devising appropriate coarse-grained level models to study many-chain polar polymer materials under electric fields in the future.

- *Molecular alignment of rigid rods in non-rigid spherical pores*

In the simulations, Shish-Kebab chains confined in spherically harmonic potentials were considered. The interaction energy between a rigid rod molecule and the harmonic potential depended only on the position of the center-of-mass of the rod, and was independent of its orientation. The applied potential varied the mean separation of rods, and two strongly confined rods displayed orientation ordering due to their excluded volume interactions. It was also found that the orientation ordering of the rods became sensitive to the local rod structure (chain length parity and local smoothness) under a strong enough the applied field. Under very strong confinement, two Shish-Kebab chains of odd numbered beads retained some rotational freedom, whereas the rotation of

two rods of even numbered beads were strongly hindered because of the tight perpendicular intercalation through their central grooves. This result was, actually, associated with the different mechanisms involved for minimizing the repulsive energy from the applied potential for the rods of odd-numbered and even-numbered beads. Furthermore, the Shish-Kebab chains were compared with different rod models in the simulations, by which the effect of the local rod smoothness on molecular alignment were explored. Our findings suggested that increasing local rod smoothness enhanced the rotational degree of freedom for confined rods, and the effect of local rod roughness emerged under strong enough applied potentials.

- *Theoretical Study of the Effects of Templated Materials on Aggregate Formation*

  In a mixture of templated polymers and reacting particles, the chain conformation of templated polymers contracted when the interactions between templated polymers and reacting particles were repulsive, but expanded for attractive interactions. Increasing the fraction of reacting particles leads to more significant deformation of chain conformation. The aggregation behavior of the reacting particles was found to be sensitive to the medium induced solvation by templated polymers. A probability distribution function $P(m)$ (probability of finding m adjacent reacting particles aligned along the same axis) was employed to describe the aggregation mechanism, which followed an exponential decay with

respect to m. The deviation from the exponential function was observed in the simulations when the difference species tended to induce phase aggregation or complex formation. In addition, the chemical architecture of templated materials (via chain connectivity) affected the effective interacting area between templated particles and reacting particles. Templated monomers (without chain connectivity) interacted with reacting particles more effectively than templated polymers (with chain connectivity) due to the larger interacting area per monomer unit. Thus, compared to templated polymers, templated monomers enhanced the dispersion and segregation of reacting particles in the blend for the attractive and repulsive interactions between different species, respectively. At a given monomer density, the cages induced by templated polymers were greater so that more reacting particles could be accommodated.

The essence of this dissertation research was to explore the effects of the local architecture of different polymer models on the static polymer properties. In each project, the local structure of a chosen model was modified to different extent, and the properties predicted by each level of model were investigated. Modification of a small portion of polymer molecules (e.g., substitute a chain segment with a fluctuating sphere model) was shown to deform chain conformation, but did not affect the fundamental physics behind their conformational behavior. This has been observed in the studies when the

applied potential was weak or absent. When the strength of applied potentials (e.g., electric field for polar polymers or repulsive harmonic potential for confined rods) was increased, the local polymer structure became important because the effect of intrinsic monomer characteristics associated with the monomer chemical architecture was amplified. The dividing line to separate the regimes between weak and strong fields could be defined at the field strengths where the local polymer structure started to influence the polymer properties. Also, our results demonstrated the general trend regarding how the local structure of polymers should be altered to maximize their response to external fields. These findings could facilitate the future development of applications and the design of more useful materials. Furthermore, we investigated the case when polymers could be viewed as medium induced solvation potentials for the other species of smaller size in a blend (templated polymers for synthesis of nanoparticles). The chain connectivity prevented complex formation between the dispersed particles and polymers when their interactions were effectively attractive. As a result, the probability of forming large aggregate increased, partially because the cavities (solvation cages) induced by polymers were larger. From the practical standpoint, our studies provided insights into mechanisms for polymer mediated nanoparticle formation, which have been widely employed to manufacture nanoparticles of different geometry and size.

Appendix: Selected source code for simulations

```
C_____________________________PART I_____________________________C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                 C
C    Monte Carlo Simulation of a Single Polymer Chain             C
C     Attached by an Elastic Ball                                 C
C                                                                 C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       program chain_ball
       implicit double precision (a-h, o-z)
C Meaning of variables:
C n: Chain length;
C x,y,z: position of polymer beads
C xb,yb,zb: position of the elastic ball
C nsta: Number of printout;
C ndis: Number of grids for ball size distribution
C nfi: Maximum number of input files
C dr(nfi): Directory names (to save some output files)
C p(nfi): Input files.
       parameter(n=11)
       parameter(nsta=100)
       parameter(ndis=200)
       parameter(nfi=88)
       character*5 dr(nfi)
       character*3 p(nfi)

       character*8 result(NFI)
       character*6 single
       character*6 accumu
       character*5 block
       character*6 hisdis
       dimension pi(ndis)
       dimension x(n), y(n), z(n)

       real*4  ran
       common /seed/ nseed
       common /rad/ rr
       common /pos/ x, y, z
       common /pos_ball/ xb, yb, zb
       common /vibs/ r_old,r_new, r_av
       common /rec/ add
       add=1.0d-6

       single='single'
       accumu='accumu'
       block='block'
       hisdis='hisdis'

       dr(1)='fr01/'
       dr(2)='fr02/'
C More definition here
C                  .
C                  .
C                  .
C                  .
```

```
      p(1)='p01'
      p(2)='p02'
C More definition here.......
C                .
C                .
C                .
C                .
      result(1)='result01'
      result(2)='result02'
C More definition here.......
C                .
C                .
C                .
C                .

      open(unit=1,file='readfile',status='unknown')
      read(1,*) nfile1,nfile2
C nfile1 and nfile2 denotes the boundary of the current run.
      close(1)

      DO 9876 nt=nfile1,nfile2
C run input files from p(nfile1) up to p(nfile2)
          open(unit=1,file= p(nt), status='old')
C
C Definition of variables:
C   dlim1, dlim2: number between 0 and 1 to control the sampling
c   probability of reptation, crankshaft and vibration move
c   fcon: force constant (rigidity) of the fluctuating sphere
c   r0: initial sphere radius
c   r_av: intrinsic sphere radius
c   nmov: number of total monte carlo moves

        read(1,*) dlim1
        read(1,*) dlim2
        read(1,*) fcon
        read(1,*) r0
        read(1,*) r_av
        read(1,*) nmov

        close(1)
C nbg is extra number of moves used to relax system
C and to determine some boundary values for some properties
        nbg=nmov/20
        nseed = 29327

C BUILD UP AN INITIAL CONFORMATION: XB,YB and ZB are X,Y,Z
coordinates
C  of the fluctuating sphere.

        xb=0.0
        yb=0.0
        zb=0.0

C Build the initial chain conformation
        call ruv(delx,dely,delz)
        x(1)=delx*(r0+0.5)
        y(1)=dely*(r0+0.5)
        z(1)=delz*(r0+0.5)
```

```
        do k=2,n
20        call ruv(delx,dely,delz)
          x(k)=x(k-1)+delx
          y(k)=y(k-1)+dely
          z(k)=z(k-1)+delz

C         Check overlap of the newly grown bead with the ball
C         and existing beads.
          dr2=dsqrt(x(k)**2+y(k)**2+z(k)**2)
          if((dr2+add).lt.(0.5+r0)) go to 20
           do j=1, k-1
              dx1=x(j)-x(k)
              dy1=y(j)-y(k)
              dz1=z(j)-z(k)
              dr1=dx1*dx1+dy1*dy1+dz1*dz1
              if (dr1+add.lt.1.0) go to 20
           end do

        end do
        close(2)

C     dnsr1 & dnsr2 & dnsr3 are accumulation of counts of successful
C      reptation, crankshaft and vibration moves///
C     rr: instataneous ball size;
C     ra: accumulative ball size;
C     dka: number of summations;
C     eetot: accumulative squared end-to-end distance
C     rgtot: accumulative squared radius of gyration

        do i=1, ndis
         pi(i)=0
        end do

        nskip=nmov/nsta
        rr=r0
        rm=rr
        rrmax=r0
        dnsr1=0
        dnsr2=0
        dnsr3=0
        ra=0
        dka=0
        rqa=0
        eetot=0
        rgtot=0

        pen(unit=10, file=dr(nt)//single, status='unknown')
        pen(unit=11, file=dr(nt)//accumu, status='unknown')

CCC   Begin moves

        do 10000, kkk=1, nmov+nbg
            selr=ran(nseed)
          if (selr.le.dlim1)then
            call rept(kp1)
            if (kp1.eq.0)  dnsr1=dnsr1+1
          else if(selr.gt.dlim1.and.selr.le.dlim2) then
            call crank(kp2)
            if (kp2.eq.0)  dnsr2=dnsr2+1
          else
```

```
            call vib(rr, fcon)
            if (rr.eq.r_new) dnsr3=dnsr3+1
          end if


         if(mod(kkk,20).eq.0)then

          call distance(ete2,rg2)
          eetot=eetot+ete2
          rgtot=rgtot+rg2

         if(kkk.le.nbg)then
           if(rr.gt.rm) rm=rr
         else
           rmax=rm
           nsp=int(rr*ndis/rmax)+1
           if(nsp.gt.ndis)nsp=ndis
           pi(nsp)=pi(nsp)+1
         endif

          dka=dka+1
          ra=ra+rr

          rqa=rqa+(rr-r_av)*(rr-r_av)
C
         end if
C
         if(mod(kkk,nskip).eq.0.and.kkk.gt.nbg)then
           nkk=kkk/nskip
           write(10,31) nkk,rr,ete2,rg2
           write(11,31) nkk,ra/dka,eetot/dka,rgtot/dka
         end if
31      format(1x,i12,4f15.6)

10000   continue

         close(10)
         close(11)
         close(12)

         open(unit=4, file=dr(nt)//hisdis, status='unknown')
          do i=1, ndis
            q1=i*bdmax/ndis
            q2=pi(i)*ndis/(bdmax*nmov)
            write(4,744)q1, q2
744         format(1x, f8.3, 5x, f20.8)
          end do
         close(4)

            ratio_rept=dnsr1/dble(nmov)
            ratio_crank=dnsr2/dble(nmov)
            ratio_vib=dnsr3/dble(nmov)
            r_exp=ra/dka
            r_sqr=rqa/dka

          open(unit=7, file=result(nt), status='unknown')
           write(7,*) '%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%'
           write(7,*) 'This is the result', NT
           write(7,902) n,r0, r_av
902         format(1x,'chainlength:', i3,' R0:', f5.2,'  R_AV:',f5.2)
```

```
           write(7,904) dlim1,dlim2
904        format(1x,'DLIM1:',f4.2,'   DLIM2:',f4.2)
           write(7,903) fcon, nmov
903        format(1x,'FCON:',f6.3,' NMov:',i10)
           write(7,906) ratio_rept,ratio_crank,ratio_vib
           write(7,907) r_exp
906        format(1x,'acceptance ratio:rept/crank/vib',3f7.4)
907        format(1x,'R average:',f8.4)
          close(7)

          open(unit=4, file=dr(NT)//bdistr, status='unknown')
            do i=1, ndis
             write(4,744) i*(rmax/ndis), pi(i)*ndis*20/(rmax*(nmov-
nbg))
744       format(1x, f8.3, 5x, f10.5)
            end do
            close(4)

9876      continue

         stop

        end

C main program ends here.
C
C Subroutine to generate new polymer conformation using reptation
move.
C
C

        subroutine rept(nkp1)
        implicit double precision (a-h, o-z)
        parameter(n=11)
        dimension x(n),y(n),z(n)
        real*4 ran
        common /pos/ x,y,z
        common /pos_ball/ xb, yb, zb
        common /seed/ nseed
        common /rad/ rr
        common /rec/ add

           nf=0
           nkp1=1

C     Randomly flip chain end to avoid sampling bias

          if(ran(nseed).le.0.5)then
            call cvert(x,y,z)
            nf=1
          end if

C     grow a bead on the chain end
          call ruv(delx,dely,delz)
            tx=x(n)+delx
            ty=y(n)+dely
            tz=z(n)+delz

C    Check overlap of the newly grown bead
           do ki=2,n-1
```

```
            dx=tx-x(ki)
            dy=ty-y(ki)
            dz=tz-z(ki)
            dr=dx*dx+dy*dy+dz*dz
            if ((dr+add).lt.1.0)then
              if(nf.eq.1) call cvert(x,y,z)
              return
            end if
          end do

           call ruv(delx,dely,delz)
           selr2=ran(nseed)

           if(selr2.le.0.5) then

C  Grow the ball on the end of the chain
            tx1= tx + delx*(rr+0.5)
            ty1= ty + dely*(rr+0.5)
            tz1= tz + delz*(rr+0.5)
            ind = 1

           else

C  Grow the ball on bead No.2

            tx1= x(2) + delx*(rr+0.5)
            ty1= y(2) + dely*(rr+0.5)
            tz1= z(2) + delz*(rr+0.5)
            ind =0

           end if

C  Check overlap of the newly grown ball

          if(ind.eq.1)then
             do i=2,n
               dx=tx1-x(i)
               dy=ty1-y(i)
               dz=tz1-z(i)
               dr=dsqrt(dx*dx+dy*dy+dz*dz)
               if ((dr+add).lt.(rr+0.5))then
                 if(nf.eq.1)   call cvert(x,y,z)
                 return
               end if
             end do
          end if

         if(ind.eq.0)then
            dx=tx1-tx
            dy=ty1-ty
            dz=tz1-tz
            dr=dsqrt(dx*dx+dy*dy+dz*dz)
            if ((dr+add).lt.(rr+0.5))then
                 if(nf.eq.1) call cvert(x,y,z)
                 return
            end if

           do i=3,n
            dx=tx1-x(i)
            dy=ty1-y(i)
```

```
            dz=tz1-z(i)
            dr=dsqrt(dx*dx+dy*dy+dz*dz)
            if ((dr+add).lt.(rr+0.5)) then
                if(nf.eq.1) call cvert(x,y,z)
                 return
            end if
           end do
         end if

c check passed, relabel the chain
           do k=1,n-1
            x(k)=x(k+1)
            y(k)=y(k+1)
            z(k)=z(k+1)
           end do

            x(n)=tx
            y(n)=ty
            z(n)=tz

C If chain is flipped, flip back to keep the ball attach to the first
C bead.

        if(ind.eq.1) call cvert(x,y,z)

C Assign the position of ball

        xb=tx1
        yb=ty1
        zb=tz1
        nkp1=0

        return
        end

C Subroutine for crankshaft move.
       subroutine crank(nkp2)
       implicit double precision (a-h, o-z)
       parameter(n=11)
       dimension x(n),y(n),z(n)
       common /pos_ball/ xb, yb, zb
       common /pos/ x,y,z
       common /rad/ rr
       common /seed/ nseed
       common /rec/ add
       common /intac/ econ
       real*4 ran

       nkp2=1
c
c  tx,ty,tz: trial position of the bead
c  tx1,ty1,tz1:  trial position of the ball

c pick a bead to rotate
C  Three cases:
C  1)move head bead; 2)move tail bead; 3) move a middle bead
C
       j=int(n*ran(nseed))+1
C vector for new bead position
       call ruv(dx,dy,dz)
```

```
       if(j.eq.1)then
C First case
        tx=x(2)+dx
        ty=y(2)+dy
        tz=z(2)+dz

        do ik=3,n
          dr2=dsqrt((tx-x(ik))**2+(ty-y(ik))**2+(tz-z(ik))**2)
          if((dr2+add).lt.1.0)return
        end do

C Second case
       else if(j.eq.n)then
         tx=x(n-1)+dx
         ty=y(n-1)+dy
         tz=z(n-1)+dz

         do ik=1,n-2
          dr2=dsqrt((tx-x(ik))**2+(ty-y(ik))**2+(tz-z(ik))**2)
          if((dr2+add).lt.1.0)return
         end do

       else
C Third case

C Computing the rotating axis
        icutn1 = j+1
        icut0 = j-1
        delx=x(icutn1)-x(icut0)
        dely=y(icutn1)-y(icut0)
        delz=z(icutn1)-z(icut0)
        del2= dsqrt(delx**2+dely**2+delz**2)
C Pick an angle for the chosen bead to rotate

        cost=0.5d0*del2
        if(dabs(cost).ge.1.0d0) then
         sint=0.0d0
        else
         sint=dsqrt(1.0d0-cost**2)
        end if
C
        delx=delx/del2
        dely=dely/del2
        delz=delz/del2
        dot=dx*delx+dy*dely+dz*delz
        xc=dx-dot*delx
        yc=dy-dot*dely
        zc=dz-dot*delz
        cn=dsqrt(xc**2+yc**2+zc**2)
        xc=xc/cn
        yc=yc/cn
        zc=zc/cn
        tx=x(icut0)+(cost*delx+xc*sint)
        ty=y(icut0)+(cost*dely+yc*sint)
        tz=z(icut0)+(cost*delz+zc*sint)

C     Check overlap of the newly rotated bead with other beads
        do i1=1,n
          if(i1.ne.j)then
```

```
            rx=tx-x(i1)
            ry=ty-y(i1)
            rz=tz-z(i1)
            dr=rx*rx+ry*ry+rz*rz
            if((dr+add).lt.1.0) return
           end if
         end do

       end if

C Now grow the ball (on either end)
       point = ran(nseed)
       if(point.le.0.5) then
C first situation, grow the ball on the first bead
         call ruv(dx,dy,dz)
         if(j.eq.1)then
C grow on the newly grown bead
           tx1=tx + dx * (rr+0.5)
           ty1=ty + dy * (rr+0.5)
           tz1=tz + dz * (rr+0.5)
         else
           tx1=x(1) + dx * (rr+0.5)
           ty1=y(1) + dy * (rr+0.5)
           tz1=z(1) + dz * (rr+0.5)
         end if
         dis1=dsqrt((tx1-tx)**2+(ty1-ty)**2+(tz1-tz)**2)
         if(dis1+add.lt.rr+0.5) return

         do km=2,n
          if(km.ne.j) then
           some=(tx1-x(km))**2+(ty1-y(km))**2+(tz1-z(km))**2
           dis2=dsqrt(some)
           if(dis2+add.lt.rr+0.5) return
          end if
         end do

         x(j)=tx
         y(j)=ty
         z(j)=tz

       else
C  grow the ball on the chain end
         call ruv(dx,dy,dz)
          if(j.eq.n)then
c grow on the newly grown bead
           tx1=tx + dx * (rr+0.5)
           ty1=ty + dy * (rr+0.5)
           tz1=tz + dz * (rr+0.5)
          else
           tx1=x(n) + dx * (rr+0.5)
           ty1=y(n) + dy * (rr+0.5)
           tz1=z(n) + dz * (rr+0.5)
          end if

          dis1=dsqrt((tx1-tx)**2+(ty1-ty)**2+(tz1-tz)**2)
          if(dis1+add.lt.rr+0.5) return
          do km=1,n-1
           if(km.ne.j) then
           some=(tx1-x(km))**2+(ty1-y(km))**2+(tz1-z(km))**2
           dis2=dsqrt(some)
```

```
            if(dis2+add.lt.rr+0.5) return
            end if
           end do

           x(j)=tx
           y(j)=ty
           z(j)=tz

           call cvert(x,y,z)

          end if

C finish here
          xb=tx1
          yb=ty1
          zb=tz1
          nkp2=0
          return
        end

C Subroutine for the fluctuation of the elastic ball.
        subroutine vib(rib, fcon)
        implicit double precision (a-h, o-z)
        parameter(n=11)
        dimension x(n), y(n), z(n)
        real*4  ran
        common /pos/ x,y,z
        common /seed/ nseed
        common /vibs/ r_old, r_new, r_av
        common /pos_ball/ xb, yb, zb
        common /rec/ add

C  Check energy of the new size
C  Step: maximum stepsize of ball vibration
        step=0.5

        r_old=rib
        e_old = fcon*(r_old-r_av)*(r_old-r_av)
89      r_new = r_old+(2*ran(nseed)-1)*step
        if(r_new.lt.0) go to 89
        e_new = fcon*(r_new-r_av)*(r_new-r_av)
        ratio = exp(e_old-e_new)
        if (ran(nseed).gt.ratio) then
          rib=r_old
        else

C       The elastic ball change its size, while move along the vector
C       connecting to the first bead to maintain tangential contact

         vx=x(1)-xb
         vy=y(1)-yb
         vz=z(1)-zb
         r2=dsqrt(vx*vx+vy*vy+vz*vz)
         vx=vx/r2
         vy=vy/r2
         vz=vz/r2
         xt=xb-(r_new-r_old)*vx
         yt=yb-(r_new-r_old)*vy
         zt=zb-(r_new-r_old)*vz
```

```
C   Check the overlap of the ball with the polymer chain
        do k=2, n
          dx=x(k)-xt
          dy=y(k)-yt
          dz=z(k)-zt
          dr=dsqrt(dx*dx+dy*dy+dz*dz)
          if ((dr+add).lt.(r_new+0.5)) then
            rib= r_old
            return
          end if
        end do

C    All check passed, set the ball size and position to the trial
ones.
         rib=r_new
         xb=xt
         yb=yt
         zb=zt
        end if

        return
        end

C Subroutine to reverse the labeling of the chain to avoid sampling
bias
        subroutine cvert(x,y,z)
        implicit double precision(a-h,o-z)
        parameter(n=11)
        dimension x(n),y(n),z(n)
         do j = 1, n/2
          k = n - j + 1
           a = x(j)
           x(j) = x(k)
           x(k) =a
           a=y(j)
           y(j)=y(k)
           y(k)=a
           a=z(j)
           z(j)=z(k)
           z(k)=a
         enddo

         return
        end

C Subroutine to calculate squared end-to-end distance and radius of
gyration
        subroutine eterg(ete2,rg2)
        implicit double precision(a-h,o-z)
        parameter(n=11)
        dimension x(n), y(n), z(n)
        common /pos/ x,y,z
        common /pos_ball/ xb,yb,zb

         rx=x(1)-x(n)
         ry=y(1)-y(n)
         rz=z(1)-z(n)
         ete2=rx*rx+ry*ry+rz*rz

         xcm=0
```

```
        ycm=0
        zcm=0
        rg=0
        do jc=1,n
         xcm=xcm+x(jc)
         ycm=ycm+y(jc)
         zcm=zcm+z(jc)
        end do

         xcm=xcm/dble(n)
         ycm=ycm/dble(n)
         zcm=zcm/dble(n)

        do j=1,n
         rjx=x(j)-xcm
         rjy=y(j)-ycm
         rjz=z(j)-zcm
         rgsq=rjx*rjx+rjy*rjy+rjz*rjz
         rg2=rg2+rg1
        end do

        return
       end

C Subroutine to generate totally random unit vectors
       subroutine ruv(fx,fy,fz)
       implicit real*8(a-h,o-z)
       real*4 ran
       common /seed/ nseed
1       b1 = 1.0d0 - 2.0d0*ran(nseed)
        b2 = 1.0d0 - 2.0d0*ran(nseed)
        bsq = b1*b1 + b2*b2
        if(bsq.gt.1.0d0) then
c        reject
         goto 1
        else
         bh = dsqrt(1.d0 - bsq)
         fx = 2.d0 * b1 * bh
         fy = 2.d0 * b2 * bh
         fz = 1.d0 - 2.d0*bsq
        end if
        return
       end

C Subroutine for random number generators (0-1)
       function ran(nseed)
        implicit real*4(a-h,o-z)
        ix=nseed
        k1=ix/127773
        ix=16807*(ix-k1*127773)-k1*2836
        if(ix.lt.0) ix=ix+2147483647
        nseed=ix
        xx=ix*4.656612875e-10
        ran=xx
        return
       end
```

```
C_____________________________PART II_____________________________C


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C       Monte Carlo Simulation of PVDF                               C
C       With atomistic force fields by Byutner et al                 C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       program pvdf
       implicit double precision(a-h, o-z)
C maxlen is maximum chain length (a CH-CF monomer is counted as one
unit)
       parameter(maxlen=100)
       parameter(pai=3.14159265358979323383264D0)
       parameter(singlecharge=1.60217733D-19)
       parameter(nfi=100)
       parameter(ndis=100)
       parameter(ngam=60)

C define some folder and output file names
       character*5 block
       character*6 torcor
C    more here
C            .
C            .

C xch is first carbon on the monomer on which hydrogens are attached.
C xcf is second carbon on the monomer on which fluorines are
attached.
       dimension xch(maxlen),ych(maxlen),zch(maxlen)
       dimension xcf(maxlen),ycf(maxlen),zcf(maxlen)
       dimension xhy1(maxlen),yhy1(maxlen),zhy1(maxlen)
       dimension xhy2(maxlen),yhy2(maxlen),zhy2(maxlen)
       dimension xfl1(maxlen),yfl1(maxlen),zfl1(maxlen)
       dimension xfl2(maxlen),yfl2(maxlen),zfl2(maxlen)
C xttch, yttch, zttch: temporary positions used for trial moves.
       dimension xttch(maxlen),yttch(maxlen),zttch(maxlen)
       dimension xttcf(maxlen),yttcf(maxlen),zttcf(maxlen)
       dimension xtthy1(maxlen),ytthy1(maxlen),ztthy1(maxlen)
       dimension xtthy2(maxlen),ytthy2(maxlen),ztthy2(maxlen)
       dimension xttfl1(maxlen),yttfl1(maxlen),zttfl1(maxlen)
       dimension xttfl2(maxlen),yttfl2(maxlen),zttfl2(maxlen)

       real*4 ran

       common /filenumber/ nt
       common /folder/ dr

       common /pos_carbon/ xch, ych, zch, xcf, ycf, zcf
       common /pos_hydro/ xhy1, yhy1, zhy1, xhy2, yhy2, zhy2
       common /pos_fluo/ xfl1, yfl1, zfl1, xfl2, yfl2, zfl2
       common /pos_headhy/ xheadhy,yheadhy,zheadhy
       common /pos_tailfl/ xtailfl,ytailfl,ztailfl

       common /tmppos_carbon/ xttch, yttch, zttch,xttcf,yttcf,zttcf
```

```
       common /tmppos_hydro/
xtthy1,ytthy1,ztthy1,xtthy2,ytthy2,ztthy2
       common /tmppos_fluo/ xttfl1,yttfl1,zttfl1,
xttfl2,yttfl2,zttfl2
       common /tmppos_headhy/ xttheadhy,yttheadhy,zttheadhy
       common /tmppos_tailfl/ xtttailfl,ytttailfl,ztttailfl

       common /chainlength/ nch
C the energy expression is: SIGMA(A*exp(-Br)-C/r**6)
       common /vanderwaal_cc/ vcca,vccb,vccc
       common /vanderwaal_hh/ vhha,vhhb,vhhc
       common /vanderwaal_ff/ vffa,vffb,vffc
       common /vanderwaal_ch/ vcha,vchb,vchc
       common /vanderwaal_cf/ vcfa,vcfb,vcfc
       common /vanderwaal_hf/ vhfa,vhfb,vhfc

       common /charge/ qch,qcf,qhy,qfl,qhead,qtail
       common /seed/ nseed
       common /field/ fstr, epsilon, if_coneps
       common /counter/ icount
       common /flag/ if_chkint,if_static,if_chkfield,if_chkbox
       common /constants/ beta,conversion
       common /rotation/ rotset, rotlim
       common /dihedral/ dihe
       common /torsionk/ tcc,tfc
       common /box/ boxlen
       common /cutoff/ cutcc,cutch,cutcf,cuthh,cuthf,cutff
       common /discard/ vdwrange, staticrange, icutbond
       common /endtoend/ dd
       common /tottot/ tstat,tvdw,ttor,tfield,tall
       common /totelecpart/ distcc,distch,distcf,distchhy,distcffl

       include 'iofile.f'
C iofile assign explicitly folder and file names (repetitively)
C for example: backpos= 'backpos'; r2ang = 'r2ang'; p(1)='p01', etc
C Read force field parameters
        open(unit=1, file='param',status='old')
C read bond radii
        read(1,*) bondlcc,bondlch,bondlcf
C then exponential coefficients, in order of cc,hh,ff,ch,cf,hf, A/B/C
values.
        read(1,*) vcca,vccb,vccc
        read(1,*) vhha,vhhb,vhhc
        read(1,*) vffa,vffb,vffc
        read(1,*) vcha,vchb,vchc
        read(1,*) vcfa,vcfb,vcfc
        read(1,*) vhfa,vhfb,vhfc
C read torsion force constants K(1-6th order cos(theta)), only two
are considered:
C       1. CF-CH-CF-CH, 2. F-CF-CH-CF
        read(1,*) (tcc(i),i=1,6)
        read(1,*) (tfc(i),i=1,6)
C read bond angles, (but they are constrained in simu)
        read(1,*) bondcc,bondhh,bondff,bondch,bondcf
C read partial charges.
        read(1,*) eqch, eqcf, eqhy, eqfl,eqheadch,eqtailcf
        read(1,*) cutcc,cutch,cutcf,cuthh,cuthf,cutff
        read(1,*) vdwrange, staticrange
        close(1)
C the parameter file look like this
```

```
C   1.534D0, 1.085D0, 1.357D0  Bondlength of CC, CH, CF
C   14976.0 3.0900 640.80      C-C  nonbond coefficients
C   2649.6 3.7400 27.36        H-H  nonbond coefficients
C   135782.0 4.5461 106.12     F-F  nonbond coefficients
C   4320.0 3.4150 138.24       C-H  nonbond coefficients
C   45094.0 3.8181 260.77      C-F  nonbond coefficients
C   12300.0 4.1431 53.88       H-F  nonbond coefficients
C   0.790 1.440 -0.760 -0.410 0.850 -0.050  CCCC torsion
           coefficients.
C   0.710 0.690 -0.760 0.280 0.290 -0.050   FCCC torsion
coefficients.
C   118.24, 109.27,105.27,108.45,107.74  Bond Angles
C  0.5202, 0.6120, 0.1807, -0.2266, -0.6215,0.7592 Partial Charges
C  1.2, 1.2, 1.5, 1.0, 1.2, 1.5  Cutoff radius CC,CH,CF,HH,HF,FF.
C   10.0, 10.0   VDW cutoff diameter and Columb interaction cutoff



C then initialize properties to be calculated and begin major loops
C         .
C         .
C         .
C         .
         end
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                          Subroutine PIVOT                          C
C                                                                    C
C        Rotate a segment of PVDF chain around chosen bond           C
C               so that bond angle is maintained                     C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       subroutine pivot(npivot)
       implicit double precision(a-h, o-z)
       parameter(maxlen=100, pai=3.14159265358979D0)

C define coordinates for chain monomers
C xch,ych,zch: position of carbons attached by hydrogen
C xcf,ycf,zcf: position of carbons attached by fluorine
C xfl,yfl,zfl: position of fluorines
C xhy,yhy,zhy: position of hydrogens
C xtt...: means temporary positions for corresponding atom types.
C         used for trial moves.

       dimension xch(maxlen),ych(maxlen),zch(maxlen)
       dimension xcf(maxlen),ycf(maxlen),zcf(maxlen)
       dimension xhy1(maxlen),yhy1(maxlen),zhy1(maxlen)
       dimension xhy2(maxlen),yhy2(maxlen),zhy2(maxlen)
       dimension xfl1(maxlen),yfl1(maxlen),zfl1(maxlen)
       dimension xfl2(maxlen),yfl2(maxlen),zfl2(maxlen)
       dimension xttch(maxlen),yttch(maxlen),zttch(maxlen)
       dimension xttcf(maxlen),yttcf(maxlen),zttcf(maxlen)
```

```
      dimension xtthy1(maxlen),ytthy1(maxlen),ztthy1(maxlen)
      dimension xtthy2(maxlen),ytthy2(maxlen),ztthy2(maxlen)
      dimension xttfl1(maxlen),yttfl1(maxlen),zttfl1(maxlen)
      dimension xttfl2(maxlen),yttfl2(maxlen),zttfl2(maxlen)
      logical accept
      real*4 ran

C original positions
      common /pos_carbon/ xch, ych, zch, xcf, ycf, zcf
      common /pos_hydro/ xhy1, yhy1, zhy1, xhy2, yhy2, zhy2
      common /pos_fluo/ xfl1, yfl1, zfl1, xfl2, yfl2, zfl2
      common /pos_headhy/ xheadhy,yheadhy,zheadhy
      common /pos_tailfl/ xtailfl,ytailfl,ztailfl
C trial positions
      common /tmppos_carbon/ xttch, yttch, zttch, xttcf, yttcf,
zttcf
      common /tmppos_hydro/
xtthy1,ytthy1,ztthy1,xtthy2,ytthy2,ztthy2
      common /tmppos_fluo/ xttfl1,yttfl1,zttfl1,
xttfl2,yttfl2,zttfl2
      common /tmppos_headhy/ xttheadhy,yttheadhy,ztttheadhy
      common /tmppos_tailfl/ xtttailfl,ytttailfl,ztttailfl
C some other shared variables with main program
      common /chainlength/ nch
      common /seed/ nseed
      common /finalpos/ qx,qy,qz
      common /counter/ icount
      common /choice/ ipiv,natom
      common /flag/ if_chkint,if_static,if_chkfield,if_chkbox
      common /centerpos/ xce,yce,zce,xas,yas,zas

C ncrank is just a counter. 1 for successful pivot move, 0 if failed.
      ncrank=0
      add=1.0D-6

C first set all temporary positions equal to real positions
      do i = 1, nch
         xttch(i) = xch(i)
         yttch(i) = ych(i)
         zttch(i) = zch(i)
         xttcf(i) = xcf(i)
         yttcf(i) = ycf(i)
         zttcf(i) = zcf(i)
         xtthy1(i) = xhy1(i)
         ytthy1(i) = yhy1(i)
         ztthy1(i) = zhy1(i)
         xtthy2(i) = xhy2(i)
         ytthy2(i) = yhy2(i)
         ztthy2(i) = zhy2(i)
         xttfl1(i) = xfl1(i)
         yttfl1(i) = yfl1(i)
         zttfl1(i) = zfl1(i)
         xttfl2(i) = xfl2(i)
         yttfl2(i) = yfl2(i)
         zttfl2(i) = zfl2(i)
       end do

         xttheadhy = xheadhy
         yttheadhy = yheadhy
         zttheadhy = zheadhy
```

```
          xtttailfl= xtailfl
          ytttailfl= ytailfl
          ztttailfl= ztailfl

C     pick a bead on the chain, from 0 to nch+1

12321     ipiv = int(nch*ran(nseed))+1
          if(ipiv.eq.nch+1) goto 12321
C natom is a flag controlling whether a CH or CF is chosen
C   as center of rotation
          natom=2.*ran(nseed)
C thetarot is a random angle by which the segment will be rotated
          thetarot = rotstep*pai*(2.D0*ran(nseed)-1.D0)

        if(ipiv.eq.0) then
C pick head hydrogen to rotate
          xce=xcf(1)
          yce=ycf(1)
          zce=zcf(1)
          xas=xch(1)
          yas=ych(1)
          zas=zch(1)
          call rotate(xheadhy,yheadhy,zheadhy,
     +               xttheadhy,yttheadhy,zttheadhy,thetarot)
          call rotate(xhy1(1),yhy1(1),zhy1(1),
     +               xtthy1(1),ytthy1(1),ztthy1(1),thetarot)
          call rotate(xhy2(1),yhy2(1),zhy2(1),
     +               xtthy2(1),ytthy2(1),ztthy2(1),thetarot)

          else if(ipiv.eq.nch+1) then
C pick some tail hydrogen (fluorine), rotate them!
            xce=xch(nch)
            yce=ych(nch)
            zce=zch(nch)
            xas=xcf(nch)
            yas=ycf(nch)
            zas=zcf(nch)
Cf1
          call rotate(xtailfl,ytailfl,ztailfl,
     +               xtttailfl,ytttailfl,ztttailfl,thetarot)
          call rotate(xfl1(nch),yfl1(nch),zfl1(nch),
     +               xttfl1(nch),yttfl1(nch),zttfl1(nch),thetarot)
          call rotate(xfl2(nch),yfl2(nch),zfl2(nch),
     +               xttfl2(nch),yttfl2(nch),zttfl2(nch),thetarot)

        else if(ipiv.le.(nch/2)) then
C rotate first half of backbone
          if(natom.eq.0) then
C pick the carbon on which two hydrogens are attached.
            xce= xch(ipiv+1)
            yce= ych(ipiv+1)
            zce= zch(ipiv+1)
            xas= xcf(ipiv)
            yas= ycf(ipiv)
            zas= zcf(ipiv)
            call calrot(rx,ry,rz)
C rotate the carbons that connects hydrogen and hydrogen themselves.
          do i=1,ipiv
           call rotate(xch(i),ych(i),zch(i),
     +                xttch(i),yttch(i),zttch(i),thetarot)
```

```
          call rotate(xhy1(i),yhy1(i),zhy1(i),
     +                xtthy1(i),ytthy1(i),ztthy1(i),thetarot)
          call rotate(xhy2(i),yhy2(i),zhy2(i),
     +                xtthy2(i),ytthy2(i),ztthy2(i),thetarot)
         enddo

C then rotate carbons connecting fluorines and fluorines themselves.
         do i=1,ipiv-1
          call rotate(xcf(i),ycf(i),zcf(i),
     +               xttcf(i),yttcf(i),zttcf(i),thetarot)
          call rotate(xfl1(i),yfl1(i),zfl1(i),
     +               xttfl1(i),yttfl1(i),zttfl1(i),thetarot)
          call rotate(xfl2(i),yfl2(i),zfl2(i),
     +               xttfl2(i),yttfl2(i),zttfl2(i),thetarot)
         enddo

C then rotate the fluorine that attached to the rotating axis:
         call rotate(xfl1(ipiv),yfl1(ipiv),zfl1(ipiv),
     +               xttfl1(ipiv),yttfl1(ipiv),zttfl1(ipiv),thetarot)
         call rotate(xfl2(ipiv),yfl2(ipiv),zfl2(ipiv),
     +               xttfl2(ipiv),yttfl2(ipiv),zttfl2(ipiv),thetarot)


        else
C i.e., happen to choose the fluorine carbon, natom=1
           xce= xcf(ipiv+1)
           yce= ycf(ipiv+1)
           zce= zcf(ipiv+1)
           xas= xch(ipiv+1)
           yas= ych(ipiv+1)
           zas= zch(ipiv+1)
           call calrot(rx,ry,rz)

C rotate the carbons that connect hydrogen and hydrogen themselves.
C and then rotate carbons that connect fluo ...
        do i=1,ipiv
           call rotate(xch(i),ych(i),zch(i),
     +               xttch(i),yttch(i),zttch(i),thetarot)
         call rotate(xhy1(i),yhy1(i),zhy1(i),
     +               xtthy1(i),ytthy1(i),ztthy1(i),thetarot)
         call rotate(xhy2(i),yhy2(i),zhy2(i),
     +               xtthy2(i),ytthy2(i),ztthy2(i),thetarot)
         call rotate(xcf(i),ycf(i),zcf(i),
     +               xttcf(i),yttcf(i),zttcf(i),thetarot)
         call rotate(xfl1(i),yfl1(i),zfl1(i),
     +               xttfl1(i),yttfl1(i),zttfl1(i),thetarot)
         call rotate(xfl2(i),yfl2(i),zfl2(i),
     +               xttfl2(i),yttfl2(i),zttfl2(i),thetarot)
        enddo

C then rotate the hydrogens that attached to the rotating axis:
         call rotate(xhy1(ipiv+1),yhy1(ipiv+1),zhy1(ipiv+1),
     +        xtthy1(ipiv+1),ytthy1(ipiv+1),ztthy1(ipiv+1),thetarot)
         call rotate(xhy2(ipiv+1),yhy2(ipiv+1),zhy2(ipiv+1),
     +        xtthy2(ipiv+1),ytthy2(ipiv+1),ztthy2(ipiv+1),thetarot)

        end if

C finally, rotate head hydrogen
         call rotate(xheadhy,yheadhy,zheadhy,
```

```
     +              xttheadhy,yttheadhy,ztheadhy,thetarot)


       else

C @second case
C i.e., pick somewhere on the second half of the backbone.

       if(natom.eq.0) then
C pick the carbon on which two hydrogens are attached.
        xce= xch(ipiv-1)
        yce= ych(ipiv-1)
        zce= zch(ipiv-1)
        xas= xcf(ipiv-1)
        yas= ycf(ipiv-1)
        zas= zcf(ipiv-1)

C rotate the carbons that connect hydrogen and hydrogen themselves.
        do i=ipiv,nch
         call rotate(xch(i),ych(i),zch(i),
     +              xttch(i),yttch(i),zttch(i),thetarot)
         call rotate(xhy1(i),yhy1(i),zhy1(i),
     +              xtthy1(i),ytthy1(i),ztthy1(i),thetarot)
         call rotate(xhy2(i),yhy2(i),zhy2(i),
     +              xtthy2(i),ytthy2(i),ztthy2(i),thetarot)
         call rotate(xcf(i),ycf(i),zcf(i),
     +              xttcf(i),yttcf(i),zttcf(i),thetarot)
         call rotate(xfl1(i),yfl1(i),zfl1(i),
     +              xttfl1(i),yttfl1(i),zttfl1(i),thetarot)
         call rotate(xfl2(i),yfl2(i),zfl2(i),
     +              xttfl2(i),yttfl2(i),zttfl2(i),thetarot)
         end do
C then rotate the fluorine that attached to the rotating axis:
         call rotate(xfl1(ipiv-1),yfl1(ipiv-1),zfl1(ipiv-1),
     +        xttfl1(ipiv-1),yttfl1(ipiv-1),zttfl1(ipiv-1),thetarot)
         call rotate(xfl2(ipiv-1),yfl2(ipiv-1),zfl2(ipiv-1),
     +        xttfl2(ipiv-1),yttfl2(ipiv-1),zttfl2(ipiv-1),thetarot)

       else
C i.e., happen to choose the fluorine carbon, natom=1
        xce= xcf(ipiv-1)
        yce= ycf(ipiv-1)
        zce= zcf(ipiv-1)
        xas= xch(ipiv)
        yas= ych(ipiv)
        zas= zch(ipiv)

C rotate the carbons that connect hydrogen and hydrogen themselves.
         do i=ipiv+1,nch
         call rotate(xch(i),ych(i),zch(i),
     +              xttch(i),yttch(i),zttch(i),thetarot)
         call rotate(xhy1(i),yhy1(i),zhy1(i),
     +              xtthy1(i),ytthy1(i),ztthy1(i),thetarot)
         call rotate(xhy2(i),yhy2(i),zhy2(i),
     +              xtthy2(i),ytthy2(i),ztthy2(i),thetarot)
         enddo

C rotate carbons that connect fluorines and fluorine themselves
         do i=ipiv,nch
         call rotate(xcf(i),ycf(i),zcf(i),
```

```
     +              xttcf(i),yttcf(i),zttcf(i),thetarot)
          call rotate(xfl1(i),yfl1(i),zfl1(i),
     +              xttfl1(i),yttfl1(i),zttfl1(i),thetarot)
          call rotate(xfl2(i),yfl2(i),zfl2(i),
     +              xttfl2(i),yttfl2(i),zttfl2(i),thetarot)
          enddo

C then rotate the hydrogens that attached to the rotating axis:
          call rotate(xhy1(ipiv),yhy1(ipiv),zhy1(ipiv),
     +         xtthy1(ipiv),ytthy1(ipiv),ztthy1(ipiv),thetarot)
          call rotate(xhy2(ipiv),yhy2(ipiv),zhy2(ipiv),
     +         xtthy2(ipiv),ytthy2(ipiv),ztthy2(ipiv),thetarot)

        end if
C  rotate tail fluorines
           call rotate(xtailfl,ytailfl,ztailfl,
     +              xtttailfl,ytttailfl,ztttailfl,thetarot)

       endif

C check acceptance based on Metropolis Criterion
          call monte(accept,0)
          if(.not.accept) return

C new conformation accepted, assign temporary positions back
          do i=1, nch
            xch(i) = xttch(i)
            ych(i) = yttch(i)
            zch(i) = zttch(i)
            xcf(i) = xttcf(i)
            ycf(i) = yttcf(i)
            zcf(i) = zttcf(i)
            xhy1(i) = xtthy1(i)
            yhy1(i) = ytthy1(i)
            zhy1(i) = ztthy1(i)
            xhy2(i) = xtthy2(i)
            yhy2(i) = ytthy2(i)
            zhy2(i) = ztthy2(i)
            xfl1(i) = xttfl1(i)
            yfl1(i) = yttfl1(i)
            zfl1(i) = zttfl1(i)
            xfl2(i) = xttfl2(i)
            yfl2(i) = yttfl2(i)
            zfl2(i) = zttfl2(i)
           end do

            xheadhy = xttheadhy
            yheadhy = yttheadhy
            zheadhy = ztheadhy
            xtailfl= xtttailfl
            ytailfl= ytttailfl
            ztailfl= ztttailfl
           ncrank=1
         return
        end

CC  Subroutine rotation of points around a given axis
CC  based on algorithm on page
astronomy.swin.edu.au/pbourke/geometry/rotate
CC  thanks to the author Ronald Goldman, with his C code.
```

```
CC  xpt,ypt,zpt is original coordinates of the points that are to be
rotated
CC  xce,yce,zce is the origin of the rotation coordinate
CC  xas,yas,zas is adjacent point to the center, together with
CC              which defines rotation axis(previously rx,ry,rz).
CC  thetarot is the angle of rotation
CC  rx,ry,rz is the normalized vector of the rotation axis(line)
CC  qx,qy,qz is the transformed coordinates of the chosen points.

      subroutine rotate(xpt,ypt,zpt,qx,qy,qz,thetarot)
      implicit double precision(a-h, o-z)
      common /centerpos/ xce,yce,zce,xas,yas,zas

        rx=xas-xce
        ry=yas-yce
        rz=zas-zce
        ddd=rx*rx+ry*ry+rz*rz
        dsq=dsqrt(ddd)
        rx=rx/dsq
        ry=ry/dsq
        rz=rz/dsq

        px=xpt-xce
        py=ypt-yce
        pz=zpt-zce

        qx = 0.D0
        qy = 0.D0
        qz = 0.D0

        costheta = dcos(thetarot)
        sintheta = dsin(thetarot)

        qx=qx+(costheta+(1.-costheta)*rx*rx)*px
        qx=qx+((1-costheta)*rx*ry-rz*sintheta)*py
        qx=qx+((1-costheta)*rx*rz+ry*sintheta)*pz

        qy=qy+((1-costheta)*rx*ry+rz*sintheta)*px
        qy=qy+(costheta+(1.-costheta)*ry*ry)*py
        qy=qy+((1.-costheta)*ry*rz-rx*sintheta)*pz

        qz=qz+((1.-costheta)*rx*rz - ry*sintheta)*px
        qz=qz+((1.-costheta)*ry*rz + rx*sintheta)*py
        qz=qz+(costheta+(1.-costheta)*rz*rz)*pz

        qx=qx+xce
        qy=qy+yce
        qz=qz+zce

        return
        end



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                  C
C      Calculate the total energy of a conformation                C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
      subroutine totener
      implicit double precision(a-h, o-z)
      parameter(maxlen=100)
      dimension xch(maxlen),ych(maxlen),zch(maxlen)
      dimension xcf(maxlen),ycf(maxlen),zcf(maxlen)
      dimension xhy1(maxlen),yhy1(maxlen),zhy1(maxlen)
      dimension xhy2(maxlen),yhy2(maxlen),zhy2(maxlen)
      dimension xfl1(maxlen),yfl1(maxlen),zfl1(maxlen)
      dimension xfl2(maxlen),yfl2(maxlen),zfl2(maxlen)
      dimension tcc(6),tfc(6)

      common /pos_carbon/ xch, ych, zch, xcf, ycf, zcf
      common /pos_hydro/ xhy1, yhy1, zhy1, xhy2, yhy2, zhy2
      common /pos_fluo/ xfl1, yfl1, zfl1, xfl2, yfl2, zfl2
      common /pos_headhy/ xheadhy,yheadhy,zheadhy
      common /pos_tailfl/ xtailfl,ytailfl,ztailfl
      common /vanderwaal_cc/ vcca,vccb,vccc
      common /vanderwaal_hh/ vhha,vhhb,vhhc
      common /vanderwaal_ff/ vffa,vffb,vffc
      common /vanderwaal_ch/ vcha,vchb,vchc
      common /vanderwaal_cf/ vcfa,vcfb,vcfc
      common /vanderwaal_hf/ vhfa,vhfb,vhfc

      common /chainlength/ nch
      common /charge/ qch,qcf,qhy,qfl,qhead,qtail
      common /field/ fstr, epsilon, if_coneps
      common /counter/ icount
      common /flag/ if_chkint,if_static,if_chkfield,if_chkbox
      common /constants/ beta,conversion
      common /torsionk/ tcc,tfc
      common /box/ boxlen
      common /cutoff/ cutcc,cutch,cutcf,cuthh,cuthf,cutff
      common /discard/ vdwrange, staticrange, icutbond
      common /tottot/ tstat,tvdw,ttor,tfield,tall
      common /totelecpart/ distcc,distch,distcf,distchhy,distcffl

C blowup is a flag to signal if the two atoms are too close to each
C other. In this subroutine, only "good" conformation is counted, so
C blowup is always false.

      logical blowup
      blowup=.false.

       e_external=0
       e_torsion=0
       e_static=0
       e_vdw=0

C initialize all energy terms.

C first vdw terms for all different pairs.
       vccold=0
       vchold=0
       vcfold=0
       vhhold=0
       vhfold=0
       vffold=0

C then electrostatic terms for all different pairs.
       chchold=0
```

```
        chcfold=0
        chhyold=0
        chflold=0
        cfcfold=0
        cfhyold=0
        cfflold=0
        hyhyold=0
        hyflold=0
        flflold=0

C 1. torsion
       tor=0
       do 9822 i=1,nch-1
          phicc=torang(xch(i),ych(i),zch(i),
     +                 xcf(i),ycf(i),zcf(i),
     +                 xch(i+1),ych(i+1),zch(i+1),
     +                 xcf(i+1),ycf(i+1),zcf(i+1))
        do j=1, 6
         tor=tor+1./2.*tcc(j)*(1.D0-dcos(dble(j)*phicc))
        enddo

        if(i.lt.nch-1) then
          phicc=torang(xcf(i),ycf(i),zcf(i),
     +                 xch(i+1),ych(i+1),zch(i+1),
     +                 xcf(i+1),ycf(i+1),zcf(i+1),
     +                 xch(i+2),ych(i+2),zch(i+2))
        do j=1, 6
         tor=tor+1./2.*tcc(j)*(1.D0-dcos(dble(j)*phicc))
        enddo
        endif

        phicf1=torang(xcf(i),ycf(i),zcf(i),
     +                 xch(i+1),ych(i+1),zch(i+1),
     +                 xcf(i+1),ycf(i+1),zcf(i+1),
     +                 xfl1(i+1),yfl1(i+1),zfl1(i+1))
        phicf2=torang(xcf(i),ycf(i),zcf(i),
     +                 xch(i+1),ych(i+1),zch(i+1),
     +                 xcf(i+1),ycf(i+1),zcf(i+1),
     +                 xfl2(i+1),yfl2(i+1),zfl2(i+1))
        phicf3=torang(xfl1(i),yfl1(i),zfl1(i),
     +                 xcf(i),ycf(i),zcf(i),
     +                 xch(i+1),ych(i+1),zch(i+1),
     +                 xcf(i+1),ycf(i+1),zcf(i+1))
        phicf4=torang(xfl2(i),yfl2(i),zfl2(i),
     +                 xcf(i),ycf(i),zcf(i),
     +                 xch(i+1),ych(i+1),zch(i+1),
     +                 xcf(i+1),ycf(i+1),zcf(i+1))

        do j=1, 6
         tor=tor+1./2.*tfc(j)*(1.D0-dcos(dble(j)*phicf1))
         tor=tor+1./2.*tfc(j)*(1.D0-dcos(dble(j)*phicf2))
         tor=tor+1./2.*tfc(j)*(1.D0-dcos(dble(j)*phicf3))
         tor=tor+1./2.*tfc(j)*(1.D0-dcos(dble(j)*phicf4))
        enddo

9822    continue

        e_torsion=tor*conversion
        ttor=e_torsion
```

```
C end torional energy

C 2. Electrostatic
C first inside unit

       do i=1, nch
         call dist(xhy1(i),yhy1(i),zhy1(i),
     +       xfl1(i),yfl1(i),zfl1(i),5,blowup,hyfl1,vhfp1,i,i)
         call dist(xhy1(i),yhy1(i),zhy1(i),
     +       xfl2(i),yfl2(i),zfl2(i),5,blowup,hyfl2,vhfp2,i,i)
         call dist(xhy2(i),yhy2(i),zhy2(i),
     +       xfl1(i),yfl1(i),zfl1(i),5,blowup,hyfl3,vhfp3,i,i)
         call dist(xhy2(i),yhy2(i),zhy2(i),
     +       xfl2(i),yfl2(i),zfl2(i),5,blowup,hyfl4,vhfp4,i,i)
         vhfold=vhfold+vhfp1+vhfp2+vhfp3+vhfp4
         hyflold=hyflold+hyfl1+hyfl2+hyfl3+hyfl4
       enddo

C then inter unit energy
       do i=1,nch-1
        do j=i+1,nch
         call dist(xhy1(i),yhy1(i),zhy1(i),
     +       xhy1(j),yhy1(j),zhy1(j),4,blowup,hyhy1,vhhp1,i,j)
         call dist(xhy1(i),yhy1(i),zhy1(i),
     +       xhy2(j),yhy2(j),zhy2(j),4,blowup,hyhy2,vhhp2,i,j)
         call dist(xhy2(i),yhy2(i),zhy2(i),
     +       xhy1(j),yhy1(j),zhy1(j),4,blowup,hyhy3,vhhp3,i,j)
         call dist(xhy2(i),yhy2(i),zhy2(i),
     +       xhy2(j),yhy2(j),zhy2(j),4,blowup,hyhy4,vhhp4,i,j)
         vhhold=vhhold+vhhp1+vhhp2+vhhp3+vhhp4
         hyhyold=hyhyold+hyhy1+hyhy2+hyhy3+hyhy4

         call dist(xhy1(i),yhy1(i),zhy1(i),
     +       xfl1(j),yfl1(j),zfl1(j),5,blowup,hyfl1,vhfp1,i,j)
         call dist(xhy1(i),yhy1(i),zhy1(i),
     +       xfl2(j),yfl2(j),zfl2(j),5,blowup,hyfl2,vhfp2,i,j)
         call dist(xhy2(i),yhy2(i),zhy2(i),
     +       xfl1(j),yfl1(j),zfl1(j),5,blowup,hyfl3,vhfp3,i,j)
         call dist(xhy2(i),yhy2(i),zhy2(i),
     +       xfl2(j),yfl2(j),zfl2(j),5,blowup,hyfl4,vhfp4,i,j)
         vhfold=vhfold+vhfp1+vhfp2+vhfp3+vhfp4
         hyflold=hyflold+hyfl1+hyfl2+hyfl3+hyfl4

C then FL-HY interaction, j-i>=2
         call dist(xhy1(j),yhy1(j),zhy1(j),
     +       xfl1(i),yfl1(i),zfl1(i),5,blowup,hyfl1,vhfp1,i,j)
         call dist(xhy1(j),yhy1(j),zhy1(j),
     +       xfl2(i),yfl2(i),zfl2(i),5,blowup,hyfl2,vhfp2,i,j)
         call dist(xhy2(j),yhy2(j),zhy2(j),
     +       xfl1(i),yfl1(i),zfl1(i),5,blowup,hyfl3,vhfp3,i,j)
         call dist(xhy2(j),yhy2(j),zhy2(j),
     +       xfl2(i),yfl2(i),zfl2(i),5,blowup,hyfl4,vhfp4,i,j)
         vhfold=vhfold+vhfp1+vhfp2+vhfp3+vhfp4
         hyflold=hyflold+hyfl1+hyfl2+hyfl3+hyfl4

C then FL-FL interaction
         call dist(xfl1(i),yfl1(i),zfl1(i),
     +       xfl1(j),yfl1(j),zfl1(j),6,blowup,flfl1,vffp1,i,j)
         call dist(xfl1(i),yfl1(i),zfl1(i),
     +       xfl2(j),yfl2(j),zfl2(j),6,blowup,flfl2,vffp2,i,j)
```

```
          call dist(xfl2(i),yfl2(i),zfl2(i),
     +        xfl1(j),yfl1(j),zfl1(j),6,blowup,flfl3,vffp3,i,j)
          call dist(xfl2(i),yfl2(i),zfl2(i),
     +        xfl2(j),yfl2(j),zfl2(j),6,blowup,flfl4,vffp4,i,j)
          vffold=vffold+vffp1+vffp2+vffp3+vffp4
          flflold=flflold+flfl1+flfl2+flfl3+flfl4

C then C-C (include all types)

          if(j-i.ge.2) then
          call dist(xch(i),ych(i),zch(i),
     +             xch(j),ych(j),zch(j),1,blowup,chch,vccp1,i,j)
          chchold= chchold+chch

          call dist(xcf(i),ycf(i),zcf(i),
     +            xch(j),ych(j),zch(j),1,blowup,chcf,vccp2,i,j)
          chcfold= chcfold+chcf

          call dist(xcf(i),ycf(i),zcf(i),
     +           xcf(j),ycf(j),zcf(j),1,blowup,cfcf,vccp3,i,j)
          cfcfold= cfcfold+cfcf

          vccold=vccold+vccp1+vccp2+vccp3
          endif

          call dist(xch(i),ych(i),zch(i),
     +            xcf(j),ycf(j),zcf(j),1,blowup,chcf,vccp4,i,j)
          chcfold= chcfold+chcf
          vccold=vccold+vccp4

C then Ch-Hy interaction
          call dist(xch(i),ych(i),zch(i),
     +        xhy1(j),yhy1(j),zhy1(j),2,blowup,chhy1,vchp1,i,j)
          call dist(xch(i),ych(i),zch(i),
     +        xhy2(j),yhy2(j),zhy2(j),2,blowup,chhy2,vchp2,i,j)
          call dist(xch(j),ych(j),zch(j),
     +        xhy1(i),yhy1(i),zhy1(i),2,blowup,chhy3,vchp3,i,j)
          call dist(xch(j),ych(j),zch(j),
     +        xhy2(i),yhy2(i),zhy2(i),2,blowup,chhy4,vchp4,i,j)
          vchold=vchold+vchp1+vchp2+vchp3+vchp4
          chhyold=chhyold+chhy1+chhy2+chhy3+chhy4

C then Cf-Hy interaction
          if(j-i.ge.2) then
          call dist(xcf(i),ycf(i),zcf(i),
     +        xhy1(j),yhy1(j),zhy1(j),2,blowup,cfhy1,vchp1,i,j)
          call dist(xcf(i),ycf(i),zcf(i),
     +        xhy2(j),yhy2(j),zhy2(j),2,blowup,cfhy2,vchp2,i,j)
          else
              cfhy1=0
              cfhy2=0
              vchp1=0
              vchp2=0
          endif
          vchold=vchold+vchp1+vchp2
          cfhyold= cfhyold+cfhy1+cfhy2

          call dist(xcf(j),ycf(j),zcf(j),
     +        xhy1(i),yhy1(i),zhy1(i),2,blowup,cfhy3,vchp3,i,j)
          call dist(xcf(j),ycf(j),zcf(j),
```

```
     +       xhy2(i),yhy2(i),zhy2(i),2,blowup,cfhy4,vchp4,i,j)
          vchold=vchold+vchp3+vchp4
          cfhyold= cfhyold+cfhy3+cfhy4

C then CH-FL interaction
          call dist(xch(i),ych(i),zch(i),
     +       xfl1(j),yfl1(j),zfl1(j),3,blowup,chfl1,vcfp1,i,j)
          call dist(xch(i),ych(i),zch(i),
     +       xfl2(j),yfl2(j),zfl2(j),3,blowup,chfl2,vcfp2,i,j)
          vcfold=vcfold+vcfp1+vcfp2
          chflold= chflold+chfl1+chfl2

          if(j-i.ge.2)then
          call dist(xch(j),ych(j),zch(j),
     +       xfl1(i),yfl1(i),zfl1(i),3,blowup,chfl3,vcfp3,i,j)
          call dist(xch(j),ych(j),zch(j),
     +       xfl2(i),yfl2(i),zfl2(i),3,blowup,chfl4,vcfp4,i,j)
          else
            chfl3=0
            chfl4=0
            vcfp3=0
            vcfp4=0
          endif
          vcfold=vcfold+vcfp3+vcfp4
          chflold= chflold+chfl3+chfl4

C then CF-FL interaction
          call dist(xcf(i),ycf(i),zcf(i),
     +       xfl1(j),yfl1(j),zfl1(j),3,blowup,cffl1,vcfp1,i,j)
          call dist(xcf(i),ycf(i),zcf(i),
     +       xfl2(j),yfl2(j),zfl2(j),3,blowup,cffl2,vcfp2,i,j)
          call dist(xcf(j),ycf(j),zcf(j),
     +       xfl1(i),yfl1(i),zfl1(i),3,blowup,cffl3,vcfp3,i,j)
          call dist(xcf(j),ycf(j),zcf(j),
     +       xfl2(i),yfl2(i),zfl2(i),3,blowup,cffl4,vcfp4,i,j)
          vcfold=vcfold+vcfp1+vcfp2+vcfp3+vcfp4
          cfflold=cfflold+cffl1+cffl2+cffl3+cffl4

         enddo
        enddo

C add the inner unit part here: only static no vdw!!!
          chcfold=chcfold+(nch-1)*1./distcc
          chflold=chflold+2*(2*nch-1)*1./distcf
          cfhyold=cfhyold+2*(2*nch-1)*1./distch
          chhyold=chhyold+2*nch*1./distchhy
          cfflold=cfflold+2*nch*1./distcffl

C And also convert to SI system. The prefactor is for Angstrom.
C First the eletrostatic interactions.

         echch=qch*qch*chchold
         echcf=qch*qcf*chcfold
         echhy=qch*qhy*chhyold
         echfl=qch*qfl*chflold

         ecfcf=qcf*qcf*cfcfold
         ecfhy=qcf*qhy*cfhyold
         ecffl=qcf*qfl*cfflold
```

```
        ehyhy=qhy*qhy*hyhyold
        ehyfl=qhy*qfl*hyflold

        eflfl=qfl*qfl*flflold

        e_static= echch+echcf+echhy+echfl
        e_static= e_static+ecfcf+ecfhy+ecffl+ehyhy+ehyfl+eflfl

        e_static= e_static*1.D10*beta/epsilon
        tstat=e_static

C van der waal's energy

        e_vdw=vccold+vchold+vcfold+vhhold+vhfold+vffold
        e_vdw=e_vdw*conversion

        tvdw=e_vdw

C Field interactioon.
        if(if_chkfield.eq.1) then
        ext_ch=0.D0
        ext_cf=0.D0
        ext_hy=0.D0
        ext_fl=0.D0

         do i=1,nch
          ext_ch=ext_ch+ych(i)
          ext_cf=ext_cf+ycf(i)
          ext_hy=ext_hy+yhy1(i)+yhy2(i)
          ext_fl=ext_fl+yfl1(i)+yfl2(i)
         end do


C  recalcute the unit, convert everything to SI system:
        ext_ch=ext_ch*1.D-10
        ext_cf=ext_cf*1.D-10
        ext_hy=ext_hy*1.D-10
        ext_fl=ext_fl*1.D-10

C  calculate energy for external field:
       e_exttemp=fstr*(ext_ch*qch+ext_cf*qcf+ext_hy*qhy+ext_fl*qfl)

C  in terms of KbT, convert
       e_external=e_exttemp*beta
       endif

       tfield=e_external

       tall = tfield + ttor + tstat + tvdw

       return
       end

C This function is to calculate the torsion angle for the 4 connected
C atoms. needed in the previous subroutine for torsion energy.

        function torang(xcor1,ycor1,zcor1,xcor2,ycor2,zcor2,
     +                  xcor3,ycor3,zcor3,xcor4,ycor4,zcor4)
       parameter(pai=3.14159265358979323383264D0)
       implicit double precision(a-h, o-z)
```

```
              x1=xcor2-xcor1
              y1=ycor2-ycor1
              z1=zcor2-zcor1

              x2=xcor3-xcor2
              y2=ycor3-ycor2
              z2=zcor3-zcor2

              x3=xcor4-xcor3
              y3=ycor4-ycor3
              z3=zcor4-zcor3


C two direction lines for two planes
              xvec1=y1*z2-y2*z1
              yvec1=x2*z1-x1*z2
              zvec1=x1*y2-x2*y1

              xvec2=y2*z3-y3*z2
              yvec2=x3*z2-x2*z3
              zvec2=x2*y3-x3*y2


              top= xvec1*xvec2+yvec1*yvec2+zvec1*zvec2
              b1=dsqrt(xvec1*xvec1+yvec1*yvec1+zvec1*zvec1)
              b2=dsqrt(xvec2*xvec2+yvec2*yvec2+zvec2*zvec2)
              bottom=b1*b2
              ratio = top/bottom

            if(ratio.ge.1.D0) then
C this is not possible, just in case.
                    torang= 0
            elseif(ratio.le.-1.D0) then
                    torang= pai
            else
                    torang=dacos(ratio)
            endif

            return
          end

C This subroutine calculates distance and distinguish different pairs
of
C atoms.
      subroutine
dist(a1,b1,c1,a2,b2,c2,itype,blowup,dq1,vdwpart,i,j)
      implicit double precision (a-h,o-z)
      common /box/ boxlen
      common /field/ fstr, epsilon, if_coneps
      common /cutoff/ cutcc,cutch,cutcf,cuthh,cuthf,cutff
      common /discard/ vdwrange, staticrange, icutbond
      common /flag/ if_chkint,if_static,if_chkfield,if_chkbox
      common /vanderwaal_cc/ vcca,vccb,vccc
      common /vanderwaal_hh/ vhha,vhhb,vhhc
      common /vanderwaal_ff/ vffa,vffb,vffc
      common /vanderwaal_ch/ vcha,vchb,vchc
      common /vanderwaal_cf/ vcfa,vcfb,vcfc
      common /vanderwaal_hf/ vhfa,vhfb,vhfc
```

```
      logical blowup
C itype:
C 1=> cc, 2=> ch, 3=> cf, 4=> hh, 5=> hf, 6=> ff

      dq1=0.0D0
      vdwpart=0.0D0

      if(itype.eq.1) then
         cutrange=cutcc
         vcofa=vcca
         vcofb=vccb
         vcofc=vccc
      else if(itype.eq.2) then
         cutrange=cutch
         vcofa=vcha
         vcofb=vchb
         vcofc=vchc
      else if(itype.eq.3) then
         cutrange=cutcf
         vcofa=vcfa
         vcofb=vcfb
         vcofc=vcfc
      else if(itype.eq.4) then
         cutrange=cuthh
         vcofa=vhha
         vcofb=vhhb
         vcofc=vhhc
      else if(itype.eq.5) then
         cutrange=cuthf
         vcofa=vhfa
         vcofb=vhfb
         vcofc=vhfc
      else
         cutrange=cutff
         vcofa=vffa
         vcofb=vffb
         vcofc=vffc
      endif

        blowup=.false.
        vx=a1-a2
        vy=b1-b2
        vz=c1-c2

        dq2=vx*vx+vy*vy+vz*vz
        dq=dsqrt(dq2)

       if(dq.lt.cutrange) then
           blowup=.true.
           return
C Only count when two atoms are separated by two or more atoms
       else if(j-i.lt.icutbond) then
        if(if_coneps.eq.1) then
            dq1=1.0D0/dq
        else
            dq1=1.D0/dq2
        endif
        if(dq.lt.vdwrange) vdwpart=vcofa*dexp(-vcofb*dq)-
vcofc/dq2**3
       endif
```

```
      return

      end

C_____________________________PART III_____________________________C


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C         RIGID RODS IN SPHERICAL HARMONIC POTENTIALS                C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      program rods in harmonic fields
C numerical imsl libraries provide the random number generator

      use numerical_libraries
      implicit double precision (a-h,o-z)
      parameter (pi=3.141592653589793d0)
      parameter (nfile=99)
      parameter (ngam=60)
      parameter (ndis=100)
      parameter (nbmax=1000,nmax=1000)

C neal array contains flags to delete or add a rod at a time to the
C system. r2each is distance from one rod end to the beginning of the
C next rod if they are connected by a spring. x,y,z are coordinates
of
C beads; xitr,yitr,zitr are temporary coordinates for trial moves;
C ngv is number of phantom beads inserted to each grooves of tangent
C sphere chains; beps1 and beps2 are force constants of the potential
in
C radial and axial direction ; fk is spring constant, req is
equilibrium
C spring length;
      dimension neal(100)
      dimension r2each(nmax)
      dimension x(nbmax),y(nbmax),z(nbmax)
      dimension xitr(nmax),yitr(nmax),zitr(nmax)
      dimension proordp(ndis)

      common /pos/ x,y,z,xitr,yitr,zitr
      common /intvar/ nmol,n,nvl
      common /ens/ beps1,beps2,energy,eold,enew
      common /sigs/ deff
      common /corr/ pmm,pmc,pmt
      common /corr1/ cor(10,25,25)
      common /groove/ ngv
      common /groove2/ facgroove
      common /bead/ nbead
      common /steps/ steptrans,steprot
      common /linkflag/ nlink
      common /linkparam/ fk,req
      common /ssize/ r2each
      common /order/ ssave
C
      character*6 init(nfile)
      character*6 cost(nfile)
```

```
      character*6 outp(nfile)
      character*5 pos(nfile)
      character*6 ordp(nfile)
      character*6 angl(nfile)
      character*8 orddis(nfile)
      character*6 rdis(nfile)
      character*7 r2dis(nfile)

      include 'iofile.f'

c 'readfile' provides the start number and ending number of input
files.
c just for some flexibility.
c nfile1 and nfile2: a sequence of results for different input
parameters
      open(1,file='readfile',status='old')
      read(1,*) nfile1,nfile2
      read(1,*) ifreadold
      read(1,*) ifanneal
      if(ifanneal.ne.0) read(1,*) (neal(i),i=nfile1,nfile2-1)
      close(1)

c now do the big loop

       nt=nfile1-1
98777  nt = nt+1

      if((nt.eq.nfile1.and.ifanneal.ne.0)
     +    .or.(ifanneal.eq.0)) then
      open(unit=1,file=init(nt),status='old')
      read(1,*)ncon
c  ncon is # of simulations moves
      read(1,*)nskip
c  nskip is # to skip before new conf. is sampled
      read(1,*)nseed
c  seed of number
      read(1,*)nbead
c  number of beads for each rod
      read(1,*)nmol
c  number of rods
      read(1,*)dlim1,dlim2
c  control which move mode to use
      read(1,*)beps1, beps2
c field strengths: k1 and k2
      read(1,*)steptrans,steprot
      read(1,*)nlink
      read(1,*)fk,req
      read(1,*)ngv

c number of phantom beads
      close(unit=1,status='keep')
      endif

      steptrans=0.2d0/beps1
      rho=1.d0+steptrans
      r2max=1.d0+steptrans*10
      steprot=0.1


      open(7,file=outp(nt),status="unknown")
```

```
c     open(7,file="outp02",status="unknown")

      facgroove = 1./(ngv+1)
c now n is incraesed by the inserted beads.
      n=nbead+(nbead-1)*ngv

      nbtot = nmol*n
c  # of total beads
      write(6,*) 'now is run number ---', nt
      write(6,77)nbtot,n,nmol
      write(7,77)nbtot,n,nmol
      write(6,78)beps1,beps2
      write(7,78)beps1,beps2
77    format('nbtot=',i4,';   n=',i2,';   nmol=',i3)
78    format('kxy=',f10.5,';   kz=',f10.5)


      if(ifreadold.eq.0) then
c the first run will make initial guess. form a bundle along z.
       do ip = 1 , nmol
         do j = 1 , n
           imn = (ip-1)*n + j
c           x(imn)=(dble(ip)-nmol/2.)*2
c           y(imn)= 0.0d0
           ptheta=(ip-1)*2.0*pi/dble(nmol)
           cosp = dcos(ptheta)
           sinp = dsin(ptheta)
c           if(nmol.eq.2) then
c              rho = 2.0
c           else
c              rho = 1.5/dcos(pi/nmol)
c           endif
           x(imn) = 10*rho*cosp
           y(imn) = 10*rho*sinp
           z(imn)= dble((j - (n+1)/2)*facgroove)
         enddo
      enddo

c reverse direction of even rods. (make the spring link)
       if(nlink.eq.1) then
          do ip=1,nmol
          if(mod(ip,2).eq.0) call cvert(ip,icon)
          enddo
       endif

       elseif(.not.(ifanneal.ne.0.and.nt.gt.nfile1)) then
       print*
       print*,'-----------------------------'
       print*, 'read an old position file'
       print*,'-----------------------------'
       print*
        open(unit=2,file='inipos',status='old')
        do i = 1, nbtot
         read(2,*)x(i),y(i),z(i)
        enddo
        close(2)
       endif

c   numerical library to generator random number
      iopt=6
```

```
      call rnopt (iopt)
      iseed = nseed
      call rnset (iseed)

c some constants
      deff = 1.0d0 - 1.d-6

c     initialize counters
      op2 = 0
      os = 0
      ntr = 0
      tnum = 0
      nsr = 0

C there might be more depending on what properties to calculate
C          .
C          .
C          .
C          .
C          .

c     calculate initial energy
         energy=0.0
         enerfa=0.0
         enersa=0.0

       do iii=1,10000
        i = int( nmol*drnunf() ) + 1
        imol = (i-1)*n
        call rept(i,isuc,icon,1)
       enddo

c calculate initial energy
       do im = 1, nmol
        do ktry = 1 , n
         if(ngv.eq.0.or.mod(ktry,ngv+1).eq.1) then
            imol=(im-1)*n
            i2 = imol+ktry
            energy = energy + 0.5*beps1*x(i2)*x(i2)
            energy = energy + 0.5*beps1*y(i2)*y(i2)
            energy = energy + 0.5*beps2*z(i2)*z(i2)
         endif
        end do
       end do

c if rods are connected by springs
       if(nlink.eq.1)then
        do im=1,nmol-1
         imol=im*n
         rx = x(imol+1)-x(imol)
         ry = y(imol+1)-y(imol)
         rz = z(imol+1)-z(imol)
         rdd=dsqrt(rx*rx+ry*ry+rz*rz)
         energy = energy+0.5*fk*(rdd-req)**2
        enddo
       endif

c calculate the distributrion of center of mass

      if(beps1.gt.5) r2max = 2.5
```

```
c begin loop

      do 1000 icon = 1, ncon

      i = int( nmol*drnunf() ) + 1

       imol = (i-1)*n
       ntr = ntr + 1

       call move(i,isuc,icon,1)
       if(isuc.eq.0)nsr = nsr + 1

       if(mod(icon,nskip).eq.0) then
          tnum = tnum + 1.
          call omega(op2,r2t)

c order parameter distribution
          nsp=int((ssave+0.5)*ndis/1.5d0)+1
          if(nsp.ge.1.and.nsp.le.ndis)proordp(nsp) = proordp(nsp) + 1

c eners is energy of spring, enerf is for harmonic fields
         eners=0.0
         enerf=0.0
         do im = 1, nmol
          do ktry = 1 , n
           if(ngv.eq.0.or.mod(ktry,ngv+1).eq.1) then
              imol=(im-1)*n
              i2 = imol+ktry
              enerf = enerf + 0.5*beps1*x(i2)*x(i2)
              enerf = enerf + 0.5*beps1*y(i2)*y(i2)
              enerf = enerf + 0.5*beps2*z(i2)*z(i2)
           endif
          end do
         end do

       enerfa = enerfa + enerf

       if(nlink.eq.1)then
        do im=1,nmol-1
         imol=im*n
         rx = x(imol+1)-x(imol)
         ry = y(imol+1)-y(imol)
         rz = z(imol+1)-z(imol)
         rdd=dsqrt(rx*rx+ry*ry+rz*rz)
         eners = eners+0.5*fk*(rdd-req)**2
        enddo
        enersa = enersa+eners
       endif

c calculate the different properties
c        .
c        .
c        .
c        .
c        .
c        .
c        .
c        .
c        .
```

```
        endif

        if(mod(icon,ncon/20).eq.0) then
        write(6,211)icon,op2/tnum,enerfa/tnum
        write(7,211)icon,op2/tnum,enerfa/tnum
        end if

        if(mod(icon,ncon/2).eq.0) then
          open(14,file='inipos', status='unknown')
            do ji=1,nbtot
               write(14,*)x(ji),y(ji),z(ji)
            end do
          close(14)
        end if

1000  continue

        open(14,file="inipos",status="unknown")
           do i=1,nbtot
             write(14,*) x(i), y(i), z(i)
           end do
        close (14)

        open(14,file=pos(nt),status="unknown")
           do i=1,nbtot
             write(14,*) x(i), y(i), z(i)
           end do
        close (14)

        write(7,*)"acceptace ratio",float(nsr)/float(ncon)
        close (7)

211     format(i12,4f15.4)

c this output just to facilitate plot of order parameter.

        open(unit=11,file=ordp(nt),status='unknown')
        write(11,678) beps1, beps2, op2/tnum
678     format(5f15.5)
        close(11)

        open(3,file=orddis(nt),status='unknown')
          do i=1, ndis
              q1= -0.5 + i*1.5d0/ndis
              q2=proordp(i)*ndis/(tnum*1.5)
              write(3,744) q1,q2
          enddo
        close(3)

C also output other properties:
C               .
C               .
C               .
C               .
C               .

c delete a rod
       if(neal(nt).eq.-1) then
       i = int( nmol*drnunf() ) + 1
        do jj = i, nmol-1
```

```
          do kk = 1,n
           ibead = (jj-1)*n+kk
           x(ibead) = x(ibead+n)
           y(ibead) = y(ibead+n)
           z(ibead) = z(ibead+n)
          enddo
         enddo
        nmol = nmol - 1
        endif

c add a rod
        if(neal(nt).eq.1) then
         call addrod
        endif
c
        if(nt.lt.nfile2.and.nmol.ge.2) goto 98777

        stop
        end

        subroutine move(i,isuc,icon,ifener)
c
c       performs a reptation move on linear chain i
c
        implicit double precision(a-h,o-z)
        parameter (pi=3.141592653589793d0)
        parameter (nbmax=1000,nmax=1000)
        parameter (nmolmax=100)
        dimension xt(nmolmax),yt(nmolmax),zt(nmolmax)
        dimension ext(nmolmax),eyt(nmolmax),ezt(nmolmax)
        dimension x(nbmax),y(nbmax),z(nbmax)
        dimension xitr(nmax),yitr(nmax),zitr(nmax)
        common /vecvec/ xt,yt,zt,ext,eyt,ezt

        common /pos/ x,y,z,xitr,yitr,zitr
        common /box/ al,alh
        common /intvar/ nmol,n,nvl
        common /ens/ beps1, beps2, energy,eold,enew
        common /sigs/ deff
        common /groove/ ngv
        common /groove2/ facgroove
        common /steps/ steptrans,steprot
        common /linkflag/ nlink
        common /linkparam/ fk,req


        isuc = 1
c        if(drnunf().gt.0.5)call cvert(i,icon)
        imol = (i-1)*n
        enew = 0.0d0
        eold = 0.0d0
c  type i move
c  move rod and rotate it
c
        if(drnunf().lt.0.5) then
c
c translation
c
check 2; step size
        xdir = steptrans*(2.0*drnunf()-1)
```

```
      ydir = steptrans*(2.0*drnunf()-1)
      zdir = steptrans*(2.0*drnunf()-1)

      xitr(1) = x(imol+1) + xdir
      yitr(1) = y(imol+1) + ydir
      zitr(1) = z(imol+1) + zdir
      vecx=(x(imol+n)-x(imol+1))/(dble(n-1))
      vecy=(y(imol+n)-y(imol+1))/(dble(n-1))
      vecz=(z(imol+n)-z(imol+1))/(dble(n-1))
      do ichk = 2, n
      xitr(ichk) = xitr(ichk-1) + vecx
      yitr(ichk) = yitr(ichk-1) + vecy
      zitr(ichk) = zitr(ichk-1) + vecz
      end do

c
      else
c
c euler angle rotation
c
       xpiv = 0.5d0*(x(imol+1)+x(imol+n))
       ypiv = 0.5d0*(y(imol+1)+y(imol+n))
       zpiv = 0.5d0*(z(imol+1)+z(imol+n))
       alpha = (2.0d0*drnunf() - 1.0d0)*(pi*steprot)
       cosa = dcos(alpha)
       sina = dsin(alpha)
       ich = 1 + int(3.0*drnunf())
       if(ich.eq.1)then
c       rotate about x axis
        a11 = 1.0
        a12 = 0.0
        a13 = 0.0
        a21 = 0.0
        a22 = cosa
        a23 = sina
        a31 = 0.0
        a32 = -sina
        a33 = cosa
       elseif(ich.eq.2)then
c       rotate about y axis
        a11 = cosa
        a12 = 0.0
        a13 = -sina
        a21 = 0.0
        a22 = 1.0
        a23 = 0.0
        a31 = sina
        a32 = 0.0
        a33 = cosa
       elseif(ich.eq.3)then
c       rotate about z axis
        a11 = cosa
        a12 = sina
        a13 = 0.0
        a21 = -sina
        a22 = cosa
        a23 = 0.0
        a31 = 0.0
        a32 = 0.0
        a33 = 1.0
```

```
       endif

        do 150 ii = 1 ,  n
         ichk = imol + ii
         xrel = x(ichk) - xpiv
         yrel = y(ichk) - ypiv
         zrel = z(ichk) - zpiv
         xitr(ii) = xpiv +a11*xrel+a12*yrel+a13*zrel
         yitr(ii) = ypiv +a21*xrel+a22*yrel+a23*zrel
         zitr(ii) = zpiv +a31*xrel+a32*yrel+a33*zrel
 150     continue
       end if

C obtain center position of each rod and directional vector to be
used in next
C overlap calculation
      do j9=1,nmol
       imol2=(j9-1)*n
       if(j9.eq.i) then
        xt(j9)=(xitr(1)+xitr(n))/2.d0
        yt(j9)=(yitr(1)+yitr(n))/2.d0
        zt(j9)=(zitr(1)+zitr(n))/2.d0
        ext(j9)=xitr(1)-xitr(n)
        eyt(j9)=yitr(1)-yitr(n)
        ezt(j9)=zitr(1)-zitr(n)
       else
        xt(j9)=(x(imol2+1)+x(imol2+n))/2.d0
        yt(j9)=(y(imol2+1)+y(imol2+n))/2.d0
        zt(j9)=(z(imol2+1)+z(imol2+n))/2.d0
        ext(j9)=x(imol2+1)-x(imol2+n)
        eyt(j9)=y(imol2+1)-y(imol2+n)
        ezt(j9)=z(imol2+1)-z(imol2+n)
       endif
      enddo

      do i2 = 1,nmol
       if(i2.ne.i)then
c first calculate shortest distance between two line segments, if
c smaller than 0.65, overlap is for sure, otherwise check pair by
pair.
         call overline(i,i2,disij)
         if(disij.lt.0.65d0)then
             return
         elseif(disij.lt.1.0d0) then
            call overrod(i,i2)
            if(nvl.eq.1)return
         endif

        endif
      enddo

       if(ifener.eq.0) goto 8989

       do ktry = 1 , n
         if(ngv.eq.0.or.mod(ktry,ngv+1).eq.1) then
            enew = enew + 0.5*beps1*xitr(ktry)*xitr(ktry)
            enew = enew + 0.5*beps1*yitr(ktry)*yitr(ktry)
            enew = enew + 0.5*beps2*zitr(ktry)*zitr(ktry)
         endif
       end do
```

```
       do ktry = 1 , n
         if(ngv.eq.0.or.mod(ktry,ngv+1).eq.1) then
            i2 = imol+ktry
            eold = eold + 0.5*beps1*x(i2)*x(i2)
            eold = eold + 0.5*beps1*y(i2)*y(i2)
            eold = eold + 0.5*beps2*z(i2)*z(i2)
         endif
       end do

c now the spring between the rods.

      if(nlink.eq.1) then
         ileft=1
         iright=1
         if(i.eq.1) ileft=0
         if(i.eq.nmol) iright=0

         if(ileft.eq.1) then
          xndel=xitr(imol+1)-x(imol)
          yndel=yitr(imol+1)-y(imol)
          zndel=zitr(imol+1)-z(imol)
          r1new=dsqrt(xndel*xndel+yndel*yndel+zndel*zndel)
          xodel=x(imol+1)-x(imol)
          yodel=y(imol+1)-y(imol)
          zodel=z(imol+1)-z(imol)
          r1old=dsqrt(xodel*xodel+yodel*yodel+zodel*zodel)
          enew = enew + 0.5*fk*(r1new-req)**2
          eold = eold + 0.5*fk*(r1old-req)**2
         endif

         if(iright.eq.1) then
          xndel=x(imol+n+1)-xitr(imol+n)
          yndel=y(imol+n+1)-yitr(imol+n)
          zndel=z(imol+n+1)-zitr(imol+n)
          r2new=dsqrt(xndel*xndel+yndel*yndel+zndel*zndel)
          xodel=x(imol+n+1)-x(imol+n)
          yodel=y(imol+n+1)-y(imol+n)
          zodel=z(imol+n+1)-z(imol+n)
          r2old=dsqrt(xodel*xodel+yodel*yodel+zodel*zodel)
          enew = enew + 0.5*fk*(r2new-req)**2
          eold = eold + 0.5*fk*(r2old-req)**2
         endif

      endif

c acceptance check
      if(eold.le.enew)then
       boltz = (eold-enew)
        xran = drnunf()
       if(boltz.le.dlog(xran))return
      endif

8989  continue
      isuc = 0
      do ij = 1 , n
         j = (i-1)*n + ij
         x(j) = xitr(ij)
         y(j) = yitr(ij)
         z(j) = zitr(ij)
```

```
      end do
      return
      end

      subroutine cvert(i,icon)
      implicit double precision(a-h,o-z)
      parameter (nbmax=1000,nmax=1000)
      dimension x(nbmax),y(nbmax),z(nbmax)
      dimension xitr(nmax),yitr(nmax),zitr(nmax)
      common /pos/ x,y,z,xitr,yitr,zitr
      common /intvar/ nmol,n,nvl
      common /sigs/ deff

      imol = (i-1)*n

      do j = 1 , n
       k = n - j + 1
       ik = imol + k
       xitr(j) = x(ik)
       yitr(j) = y(ik)
       zitr(j) = z(ik)
      enddo

      do ij = 1 , n
        j = imol + ij
       x(j) = xitr(ij)
       y(j) = yitr(ij)
       z(j) = zitr(ij)
      enddo
      return
      end
c
c check overlap
      subroutine overrod(itry,jtry)
      implicit double precision(a-h,o-z)
      parameter (nbmax=1000,nmax=1000)
      dimension x(nbmax),y(nbmax),z(nbmax)
      dimension xitr(nmax),yitr(nmax),zitr(nmax)
      common /pos/ x,y,z,xitr,yitr,zitr
      common /intvar/ nmol,n,nvl
      common /sigs/ deff

      nvl = 1

         do i1 = 1 , nmol
          do j1 = 1 , n
             if(i1.ne.itry) then
               ijk =  (i1-1)*n + j1

               x1 = xitr(jtry)
               y1 = yitr(jtry)
               z1 = zitr(jtry)

               x2 = x(ijk)
               y2 = y(ijk)
               z2 = z(ijk)

                xt = x1 - x2
                yt = y1 - y2
                zt = z1 - z2
```

```
                rdis = xt*xt+yt*yt+zt*zt
                if(rdis.lt.deff)return
              end if

            end do
          end do
          nvl = 0
        return
        end

c generate a random vector of length 1.
        subroutine ruv(x,y,z)
        implicit double precision(a-h,o-z)

1       b1 = 1.0d0 - 2.0d0*drnunf()
        b2 = 1.0d0 - 2.0d0*drnunf()
        bsq = b1*b1 + b2*b2
        if(bsq.gt.1.0d0) then
c        reject
         goto 1
        else
         bh = dsqrt(1.d0 - bsq)
         x = 2.d0 * b1 * bh
         y = 2.d0 * b2 * bh
         z = 1.d0 - 2.d0*bsq
        endif
        return
        end

c calculate the properties of the system
        subroutine omega(op2,r2t)
        implicit double precision(a-h,o-z)
        parameter(pi=3.141592653589793)
        parameter(nbmax=1000,nmax=1000)
        dimension pmm(100), pmc(500), pmt(500)
        dimension x(nbmax),y(nbmax),z(nbmax)
        dimension xitr(nmax),yitr(nmax),zitr(nmax)
        dimension r2each(nmax)
        common /intvar/ nmol,n,nvl
        common /pos/ x,y,z,xitr,yitr,zitr
        common /groove/ ngv
        common /groove2/ facgroove
        common /bead/ nbead
        common /ssize/ r2each
        common /order/ ssave

c now r2 for linked beads
         nbtot= nmol*n
         xx=x(1)-x(nbtot)
         yy=y(1)-y(nbtot)
         zz=z(1)-z(nbtot)
         r2t=xx*xx+yy*yy+zz*zz

         do k=1,nmol-1
          ib = k*n
          xx=x(ib+1)-x(ib)
          yy=y(ib+1)-y(ib)
          zz=z(ib+1)-z(ib)
          r2each(k) = xx*xx+yy*yy+zz*zz
         enddo
```

```
C     calculate order parameter
      q11 = 0.0d0
      q12 = 0.0d0
      q13 = 0.0d0
      q21 = 0.0d0
      q22 = 0.0d0
      q23 = 0.0d0
      q31 = 0.0d0
      q32 = 0.0d0
      q33 = 0.0d0
      do ij = 1, nmol
        jm = (ij-1)*n
        ax = (x(jm+n)-x(jm+1))/dble(n-1)/facgroove
        ay = (y(jm+n)-y(jm+1))/dble(n-1)/facgroove
        az = (z(jm+n)-z(jm+1))/dble(n-1)/facgroove
       q11 = q11 + ax*ax
       q12 = q12 + ax*ay
       q13 = q13 + ax*az
       q22 = q22 + ay*ay
       q23 = q23 + ay*az
       q33 = q33 + az*az
      enddo
      q11 = 1.50d0*q11/dble(nmol)-0.50d0
      q12 = 1.50d0*q12/dble(nmol)
      q13 = 1.50d0*q13/dble(nmol)
      q22 = 1.50d0*q22/dble(nmol)-0.50d0
      q23 = 1.50d0*q23/dble(nmol)
      q33 = 1.50d0*q33/dble(nmol)-0.50d0
      q21 = q12
      q31 = q13
      q32 = q23
c     calculate eigenvalues
      a1 = -(q11+q22+q33)
      a2 = q11*q22+q11*q33+q22*q33-q23**2-q12**2-q13**2
      a3 = -(2.d0*q12*q13*q23+q11*q22*q33
     c       -q11*q23**2-q22*q13**2-q33*q12**2)
      q = (a1**2-3.0d0*a2)/9.0d0
        if(q.lt.0.d0) print*,"q-error"
      r = (2.0d0*a1**3-9.0d0*a1*a2+27.0d0*a3)/54.0d0
      dcf = q**3-r**2
      if(dcf.lt.0.0d0)then
       write(6,*)'sorry: only one real root!'
      endif
       xx=r/dsqrt(q**3)
      theta = dacos(xx)
      con = -2.0d0*dsqrt(q)
      t1 = theta/3.0d0
      t2 = (theta+2.0d0*pi)/3.0d0
      t3 = (theta+4.0d0*pi)/3.0d0
      t1 = con*dcos(t1) - a1/3.0d0
      t2 = con*dcos(t2) - a1/3.0d0
      t3 = con*dcos(t3) - a1/3.0d0
      opmin = min(t1,t2,t3)
      opmax = max(t1,t2,t3)
      if(dabs(opmin-t1).lt.1.d-8)opmid=min(t2,t3)
      if(dabs(opmin-t2).lt.1.d-8)opmid=min(t1,t3)
      if(dabs(opmin-t3).lt.1.d-8)opmid=min(t1,t2)
c      op1 = op1 + opmin
      ssave=-2.*opmid
```

```
       op2 = op2 - 2.0d0*opmid
c       op3 = op3 + opmax

50     return
       end

      subroutine addrod()
c
c     add 1 new rod to the existing system
c
      implicit double precision(a-h,o-z)
      parameter (pi=3.141592653589793d0)
      parameter (nbmax=1000,nmax=1000)
      dimension x(nbmax),y(nbmax),z(nbmax)
      dimension xitr(nmax),yitr(nmax),zitr(nmax)
      dimension xs(nmax),ys(nmax),zs(nmax)
      common /pos/ x,y,z,xitr,yitr,zitr
      common /box/ al,alh
      common /intvar/ nmol,n,nvl
      common /ens/ beps1, beps2, energy,eold,enew
      common /sigs/ deff
      common /groove/ ngv
      common /groove2/ facgroove

c generate a new rod.
      xshift=10.d0
      yshift=10.d0
      zshift=10.d0

3343  xshift= xshift+1
      yshift= yshift+1
      zshift= zshift+1

         do j = 1 , n

           xitr(j)= x(j)+xshift
           yitr(j)= y(j)+yshift
           zitr(j)= z(j)+zshift
           call over(nmol+1,j)
           if(nvl.eq.1) then
                print*, 'too far, move a bit'
                goto 3343
           endif
         enddo

c succeed. relable the rods and nmol
         do j = 1 , n
           imn = nmol*n + j
           x(imn)=xitr(j)
           y(imn)=yitr(j)
           z(imn)=zitr(j)
         enddo
       nmol=nmol + 1

c now relax the new position a little bit.
      do i=1,1000
       call rept(nmol,isuc,icon,1)
      enddo

       return
```

```
      end

      subroutine trans(isuc,icon)
c     performs a random translation for all rods
      implicit double precision(a-h,o-z)
      parameter (pi=3.141592653589793d0)
      parameter (nbmax=1000,nmax=1000)
      dimension x(nbmax),y(nbmax),z(nbmax)
      dimension xitr(nmax),yitr(nmax),zitr(nmax)
      dimension x2(nmax),y2(nmax),z2(nmax)
      common /pos/ x,y,z,xitr,yitr,zitr
      common /box/ al,alh
      common /intvar/ nmol,n,nvl
      common /ens/ beps1, beps2, energy,eold,enew
      common /sigs/ deff
      common /groove/ ngv
      common /groove2/ facgroove
      common /steps/ steptrans,steprot
      common /linkflag/ nlink
      common /linkparam/ fk,req

      isuc = 1
c      if(drnunf().gt.0.5)call cvert(i,icon)
      enew = 0.0d0
      eold = 0.0d0

      xrel = (2.*drnunf()-1.)*steptrans*nmol
      yrel = (2.*drnunf()-1.)*steptrans*nmol
      zrel = (2.*drnunf()-1.)*steptrans*nmol
check 2; step size
      do i=1,nmol
        do j=1,n
         ib = (i-1)*n + j
         x2(ib) = x(ib) + xrel
         y2(ib) = y(ib) + yrel
         z2(ib) = z(ib) + zrel
         if(ngv.eq.0.or.mod(j,ngv+1).eq.1) then
            enew = enew + 0.5*beps1*x2(ib)*x2(ib)
            enew = enew + 0.5*beps1*y2(ib)*y2(ib)
            enew = enew + 0.5*beps2*z2(ib)*z2(ib)
            eold = eold + 0.5*beps1*x(ib)*x(ib)
            eold = eold + 0.5*beps1*y(ib)*y(ib)
            eold = eold + 0.5*beps2*z(ib)*z(ib)
         endif

        enddo
      enddo



      if(eold.le.enew)then
       boltz = (eold-enew)
         xran = drnunf()
       if(boltz.le.dlog(xran))return
      endif
c     move accepted

      isuc = 0
      do i = 1 , nmol
       do j = 1 , n
```

```
            ib = (i-1)*n+j
            x(ib) = x2(ib)
            y(ib) = y2(ib)
            z(ib) = z2(ib)
          enddo
        enddo
        return
        end

        subroutine rot(isuc,icon)
c       performs a rotation for all rods
        implicit double precision(a-h,o-z)
        parameter (pi=3.141592653589793d0)
        parameter (nbmax=1000,nmax=1000)
        dimension x(nbmax),y(nbmax),z(nbmax)
        dimension x2(nbmax),y2(nbmax),z2(nbmax)
        dimension xitr(nmax),yitr(nmax),zitr(nmax)
        common /pos/ x,y,z,xitr,yitr,zitr
        common /box/ al,alh
        common /intvar/ nmol,n,nvl
        common /ens/ beps1, beps2, energy,eold,enew
        common /sigs/ deff
        common /groove/ ngv
        common /groove2/ facgroove
        common /steps/ steptrans,steprot
        common /linkflag/ nlink
        common /linkparam/ fk,req


        isuc = 1
c       if(drnunf().gt.0.5)call cvert(i,icon)
        enew = 0.0d0
        eold = 0.0d0
c  type i move
c  move rod and rotate it
c
c       if(drnunf().lt.0.25) then
         xpiv = 0
         ypiv = 0
         zpiv = 0
check 3: step size
         alpha = (2.0d0*drnunf() - 1.0d0)*(pi*steprot)
         cosa = dcos(alpha)
         sina = dsin(alpha)
         ich = 1 + int(3.0*drnunf())
         if(ich.eq.1)then
c         rotate about x axis
          a11 = 1.0
          a12 = 0.0
          a13 = 0.0
          a21 = 0.0
          a22 = cosa
          a23 = sina
          a31 = 0.0
          a32 = -sina
          a33 = cosa
         elseif(ich.eq.2)then
c         rotate about y axis
          a11 = cosa
          a12 = 0.0
```

```
        a13 = -sina
        a21 = 0.0
        a22 = 1.0
        a23 = 0.0
        a31 = sina
        a32 = 0.0
        a33 = cosa
       elseif(ich.eq.3)then
c       rotate about z axis
        a11 = cosa
        a12 = sina
        a13 = 0.0
        a21 = -sina
        a22 = cosa
        a23 = 0.0
        a31 = 0.0
        a32 = 0.0
        a33 = 1.0
       endif

        do i = 1 , nmol
          do j = 1, n
          ib=(i-1)*n + j
           xrel = x(ib) - xpiv
           yrel = y(ib) - ypiv
           zrel = z(ib) - zpiv
           x2(ib) = xpiv +a11*xrel+a12*yrel+a13*zrel
           y2(ib) = ypiv +a21*xrel+a22*yrel+a23*zrel
           z2(ib) = zpiv +a31*xrel+a32*yrel+a33*zrel
           if(ngv.eq.0.or.mod(j,ngv+1).eq.1) then
            enew = enew + 0.5*beps1*x2(ib)*x2(ib)
            enew = enew + 0.5*beps1*y2(ib)*y2(ib)
            enew = enew + 0.5*beps2*z2(ib)*z2(ib)
            eold = eold + 0.5*beps1*x(ib)*x(ib)
            eold = eold + 0.5*beps1*y(ib)*y(ib)
            eold = eold + 0.5*beps2*z(ib)*z(ib)
           endif

        enddo
       enddo

      if(eold.le.enew)then
c      boltz = dexp(eold-enew)
c      if(boltz.le.drnunf())return
       boltz = (eold-enew)
         xran = drnunf()
       if(boltz.le.dlog(xran))return
      endif
c     move accepted

      isuc = 0
      do i = 1 , nmol
       do j = 1 , n
         ib = (i-1)*n+j
         x(ib) = x2(ib)
         y(ib) = y2(ib)
         z(ib) = z2(ib)
       enddo
      enddo
```

```
      return
      end

      subroutine overline(i,j,rcl)
c given two line segments in space, with center positions r1,r2,
c direction vectors e1,e2 and segment lengths l1,l2, calculate the
c shortest distance between the two segments
      implicit double precision (a-h,o-z)
      parameter (nmolmax=100)
      dimension xt(nmolmax),yt(nmolmax),zt(nmolmax)
      dimension ext(nmolmax),eyt(nmolmax),ezt(nmolmax)
      common /vecvec/ xt,yt,zt,ext,eyt,ezt
      common /intvar/ nmol,n,nvl
      common /bead/ nbead

c first set it to big!! in case paralell, do not count it.
c see allen, evans, frenkel, mulder, adv. chem. phys. vol. lxxxvi, p.
1
c r1:
        x1=xt(i)
        y1=yt(i)
        z1=zt(i)
c r2:
        x2=xt(j)
        y2=yt(j)
        z2=zt(j)
c distance vector between centers:
        x12=x2-x1
        y12=y2-y1
        z12=z2-z1
c e1:
        ex1=ext(i)
        ey1=eyt(i)
        ez1=ezt(i)
c e2:
        ex2=ext(j)
        ey2=eyt(j)
        ez2=ezt(j)

c extra:normalize directional vectors

        dr1=dsqrt(ex1*ex1+ey1*ey1+ez1*ez1)
        dr2=dsqrt(ex2*ex2+ey2*ey2+ez2*ez2)

        ex1=ex1/dr1
        ey1=ey1/dr1
        ez1=ez1/dr1

        ex2=ex2/dr2
        ey2=ey2/dr2
        ez2=ez2/dr2

c first, find the normal vector en at minimal distance between the
c carrier lines g1 and g2:
c since en is normal both to e1 and e2 we have for any normal vector
c
c r12*e1=lam-mu*(e1*e2)
c r12*e2=lam*(e1*e2)-mu

c which yields (excluding the case e1*e2=0)
```

```
c lam0=[(e1*r12)-(e1*e2)(e2*r12)]/[1-(e1*e2)^2]
c mu0=-[(e2*r12)-(e1*e2)(e1*r12)]/[1-(e1*e2)^2]
c
        e1r=ex1*x12+ey1*y12+ez1*z12
        e2r=ex2*x12+ey2*y12+ez2*z12
        e12=ex1*ex2+ey1*ey2+ez1*ez2

        dnn=1.d0-e12*e12
        if(dnn.gt.0)then
           recipr=1.d0/dnn
        else
           return
        endif

c        denom1=e1r-e12*e2r
c        denom2=e2r-e12*e1r
c        if(denom1.lt.1.d-10.or.denom2.lt.1.d-10) return

        xla0=recipr*(e1r-e12*e2r)
        xmu0=-recipr*(e2r-e12*e1r)

c shortest perpendicular distance between carrier lines:
        xx12=x12+xmu0*ex2-xla0*ex1
        yy12=y12+xmu0*ey2-xla0*ey1
        zz12=z12+xmu0*ez2-xla0*ez1
        rnsq=xx12*xx12+yy12*yy12+zz12*zz12

        if(rnsq.ge.0d0) dd=sqrt(rnsq)


c now for the squared in-plane distance risq between two line
segments:

c rectangle half lengths h1=l1/2, h2=l2/2
        dlen1=nbead-1.
        dlen2=nbead-1.
        h1=0.5*dlen1
        h2=0.5*dlen2

c if the origin is contained in the rectangle,
c life is easy: the origin is the minimum, and
c the in-plane distance is zeroc

        if ((xla0*xla0.le.h1*h1).and.(xmu0*xmu0.le.h2*h2)) then
          risq=0.
c simple overlap

        else
c find minimum of f=gamma^2+delta^2-2*gamma*delta*(e1*e2)
c where gamma, delta are the line parameters reckoned from the
intersection
c (=lam0,mu0)


c first, find the lines gamm and delm that are nearest to the origin:
          gam1=-xla0-h1
          gam2=-xla0+h1
          gamm=gam1
          if (gam1*gam1.gt.gam2*gam2) gamm=gam2
```

```
          del1=-xmu0-h2
          del2=-xmu0+h2
          delm=del1
          if (del1*del1.gt.del2*del2) delm=del2

c now choose the line gamma=gamm and optimize delta:
          gam=gamm
          delms=gam*e12
          aa=xmu0+delms
c look if delms is within [-xmu0+/-l/2]:
          if (aa*aa.le.h2*h2) then
            del=delms
c somewhere along the side gam=gamm
          else
c delms out of range --> corner next to delmsc
            del=del1
            a1=delms-del1
            a2=delms-del2
            if (a1*a1.gt.a2*a2) del=del2
          endif
c distance at these gam, del:
          f1=gam*gam+del*del-2.*gam*del*e12

c now choose the line delta=deltam and optimize gamma:
          del=delm
          gamms=del*e12
          aa=xla0+gamms
c look if gamms is within [-xla0+/-l/2]:
          if (aa*aa.le.h1*h1) then
            gam=gamms
c somewhere along the side gam=gamm
          else
c gamms out of range --> corner next to gammsc
            gam=gam1
            a1=gamms-gam1
            a2=gamms-gam2
            if (a1*a1.gt.a2*a2) gam=gam2
          endif
c distance at these gam, del:
          f2=gam*gam+del*del-2.*gam*del*e12

c compare f1 and f2 to find risq:
          risq=f1
          if(f2.lt.f1) risq=f2
        endif

c distance of closest approach squared =
c in-plane distance squared + normal distance squared:
        rclsq=rnsq+risq

        if(rclsq.ge.0.d0) rcl=sqrt(rclsq)

       return
       end
```

```
C_____________________________PART IV_____________________________C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C          Numeric Method to Obtain Angular Dependent                 C
C     Excluded Volume Effect  of Two Rigid Shish-Kebab Rods           C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

        program track exclusion zone
        implicit double precision(a-h,o-z)
        parameter(nmax=200,nfile=200)
        parameter(npsimax=90000)
        dimension x1(nmax),y1(nmax),z1(nmax)
        dimension x2(nmax),y2(nmax),z2(nmax)
        dimension phival1(0:npsimax)
        dimension phival2(0:npsimax)
        dimension psi(0:npsimax)

        character*5 inp(nfile)
        character*6 outb(nfile)
        character*6 outf(nfile)
        logical found1,found2

C io.f contains definitions for all input and output file names.
C      and is rather repetitive.
        include 'io.f'

C 'readfile' only contain two numbers, the starting and ending file
C   number to batch process a number of input files
        open(1,file='readfile',status='unknown')
        read(1,*) nfile1,nfile2
        close(1)

C n: number of beads on each rod
C rho: distance of center of mass from origin
        do 9876 nt=nfile1,nfile2
        open(1,file=inp(NT),status='old')
        read(1,*) n
        read(1,*) rho
        close(1)

7865    continue
        pi=dacos(-1.D0)

        open(20,file=outf(NT),status='unknown')

C loop on theta from 0 to pi/2, every degree
        do 9999 jk=1,90

        theta=dble(jk)*pi/180.D0
C gridsize for phi and psi
        nphi=600
        npsi=600
```

```
C center position of the first rod
        cenx1=0.D0
        ceny1=0.D0
        cenz1=0.D0

       do 3892 ld = 1,npsi
C psi angle is scanned from 0 to pi/2
        psi(ld)=pi/2.*ld/npsi

C found1 and found2 are flags signaling occurance of collision
        found1=.false.
        found2=.false.

C center of sencond rod (on the xz plane, with polar angle psi)
        cenx2=rho*dsin(psi(ld))
        ceny2=0.D0
        cenz2=rho*dcos(psi(ld))

C set the first rod on the z axis
        do i=1,n
          x1(i)=0.D0
          y1(i)=0.D0
         if(mod(n,2).eq.0)then
          z1(i)=cenz1+dble(i-n/2)-0.5d0
         else
          z1(i)=cenz1+dble(i-n/2)-1.D0
         endif
        enddo

C rotate the second rod along its z' axis (obtained from
C shifting z axis to its center) with increasing phi value

       do k=0,nphi
        phi1=pi/2*(1.D0+dble(k)/dble(nphi))

        do i=1,n
          if(mod(n,2).eq.0) then
            dinc=dble(i-n/2)-0.5D0
          else
            dinc=dble(i-n/2)-1.D0
          endif

          x2(i)=cenx2+dinc*dsin(theta)*dcos(phi1)
          y2(i)=ceny2+dinc*dsin(theta)*dsin(phi1)
          z2(i)=cenz2+dinc*dcos(theta)

C check distance between all pairs of beads between the two rods
          do j=1,n
            xx=x2(i)-x1(j)
            yy=y2(i)-y1(j)
            zz=z2(i)-z1(j)
            dist=dsqrt(xx*xx+yy*yy+zz*zz)
            if(dist.le.0.9999999999999D0) then
              phival1(ld)=phi1
              found1=.true.
              goto 7111
            endif
          enddo

        enddo
```

```
       enddo

7111   continue

       if(.not.found1) phival1(ld)=pi

C now rotate second rod from pi*3/2 to 2*pi
       do k=0,nphi

        phi2=pi+pi/2.d0*(1.D0+dble(k)/dble(nphi))
        do i=n,1,-1
          if(mod(n,2).eq.0) then
            dinc=dble(i-n/2)-0.5D0
          else
            dinc=dble(i-n/2)-1.D0
          endif

          x2(i)=cenx2+dinc*dsin(theta)*dcos(phi2)
          y2(i)=ceny2+dinc*dsin(theta)*dsin(phi2)
          z2(i)=cenz2+dinc*dcos(theta)

          do j=n,1,-1
            xx=x2(i)-x1(j)
            yy=y2(i)-y1(j)
            zz=z2(i)-z1(j)
            dist=dsqrt(xx*xx+yy*yy+zz*zz)
            if(ifout.eq.1) write(9,888)phi2,i,j,dist
            if(dist.lt.0.99999999999999D0) then
              phival2(ld)=phi2
              found2=.true.
              goto 7222
            endif
          enddo

         enddo

        enddo
7222   continue

       if(.not.found2) phival2(ld)=2.d0*pi

3892  continue

       freezone=0
       freezone1=0
       freezone2=0
       do jj=1,npsi
         phi=phival1(jj)-pi/2.0d0
         sum1=phi*dsin(psi(jj))
         freezone1 = freezone1 + sum1
         phi=phival2(jj)-pi*3./2.0d0
         sum2=phi*dsin(psi(jj))
         freezone2 = freezone2 + sum2
       enddo

c normalization
        freezone=freezone/npsi
        freezone1=freezone1/npsi
        freezone2=freezone2/npsi
```

```
        if(freezone1.ge.1.d0) freezone1=1.d0
        if(freezone2.ge.1.d0) freezone2=1.d0
        freezone=(freezone1+freezone2)/2.d0

       write(20,789)rho,theta*180/pi,freezone
789    format(4F16.8)

9999    continue
C stepz: stepsize for increasing distance of center of the second rod
        stepz=0.5d0
        if(rho.le.dble(n)-stepz) then
           rho=rho+stepz
           goto 7865
        endif

9876   continue

       close(20)

       end



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C       Numeric Integration for Calculation of Order                  C
C          Parameter of two odd-numbered rods at                      C
C              infinitely strong fields (r=1)                         C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


       program S_2 two odd numbered rods at r1

       implicit double precision(a-h,o-z)
       parameter(ntheta=10000,npsi=10000)
       dimension t(ntheta),p(npsi),poft(ntheta)
       pi=dacos(-1.0d0)

       thetagrid=pi/2.d0/ntheta
       psigrid=pi/2.d0/npsi

       ntemp1=10000
       dtheta=pi/2.d0/dble(ntemp1)
       ntemp2=100000
       sum0=0

       open(9,file='profile',status='unknown')

       do i=1,ntemp1
        theta=pi/2.d0/dble(ntemp1)*dble(i)
        dpsi=(pi/2.d0-pi/3.d0)/dble(ntemp2)
C sum1 and sum2 are accumulation over two quadrants:
C       pi/2 ~ pi and 3pi/2 ~ 2pi, respectively.

        sum1=0.d0
        sum2=0.d0

        do j=1,ntemp2
         psi=pi/3.d0+dpsi*dble(j)
```

```
          facint1=frac(1,theta,psi)*dsin(psi)*dpsi
          sum1=sum1+facint1
          facint2=-frac(2,theta,psi)*dsin(psi)*dpsi
          sum2=sum2+facint2
         enddo

         sumsum=(sum1+sum2)/2.d0
         write(9,*) theta,sumsum
         sum01=sum01+sumsum*dsin(theta)*dtheta*(1.5*dcos(theta)-0.5d0)
         sum0=sum0+sumsum*dsin(theta)*dtheta

        enddo

        close(9)

        open(10,file='result',status='unknown')
        write(10,*) 'order parameter: ', sum01/sum0
        close(10)

        end
C end of main program

C Subroutine to calculate the allowed fraction of psi for certain
theta
C and psi
        function frac(icase,t1,p1)
        implicit double precision(a-h,o-z)
        pi=dacos(-1.0d0)

        bottom = dsin(t1)*dsin(p1)

C icase=1 for pi/2 ~ pi; icase=2 for 3pi/2 ~ 2pi
        if(icase.eq.1) then
         if(dcos(t1)+dcos(p1).lt.0.5d0) then
           ftemp=(-1.d0-2.d0*dcos(t1)*dcos(p1))/2.d0/bottom
         else
           ftemp=(dcos(t1)+dcos(p1)-1.d0-dcos(t1)*dcos(p1))/bottom
         endif

        else

         if(dcos(t1)-dcos(p1).lt.0.5d0) then
           ftemp=(1.d0-2.d0*dcos(t1)*dcos(p1))/2.d0/bottom
         else
           ftemp=(-dcos(t1)+dcos(p1)+1.d0-dcos(t1)*dcos(p1))/bottom
         endif

        endif

          if(ftemp.gt.1.d0.or.ftemp.lt.-1.d0) then
              print*,'wrong! for ftemp', ftemp
              stop
          endif

         frac=2.d0*dacos(ftemp)/pi - 1.0

        return
        end
```

```
C_____________________________PART V_____________________________C


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C       Set up an initial condition for simulation of                 C
C          a blend containing templated polymer                       C
C             and reacting particles                                  C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

     implicit double precision (a-h,o-z)
     dimension x(600),y(600),q(600)

        print*,int(1.0d-4),int(1.0010000d0)
c: side length of simulation cell
        ialh = 10
c: ratio of monomer to precursors(1:3;type:ratio=1./3.)
c: ratio of monomer to precursors(3:1;type:ratio=3.)
        ratio = 1./3.
c: total particles (monomers + precursors)
        nbtot = 320
c: # of precursors
        if (ratio.lt.1.) then
           npc = nbtot/(1.+ratio) + 1
        else
           npc = nbtot/(1.+ratio)
        endif
c: # of monomers
        nmon = nbtot-npc
        print*,"nmon",nmon
c: chain length
        n = 8
c: total # of polymer chains
        nmol = nmon/8

         jb = 0
        do iy = -ialh , ialh , 2
          do ix = -ialh ,  -ialh + 7

            jb = jb + 1
            x(jb) =  ix
            y(jb) =  iy
            q(jb) =  1.

          if(jb.ge.nmon) goto 10
          end do

          do ix = -ialh + 8 , -ialh + 15

             jb = jb +1
            x(jb) = ix
            y(jb) = iy
            q(jb) = 1.
```

```
          if(jb.ge.nmon) goto 10
          end do
        end do

        do iy = -ialh + 16, -ialh + 20
          do ix = -ialh ,  -ialh + 7
            jb = jb + 1

            x(jb)=ix
            y(jb)=iy
            q(jb)=1.

          if(jb.ge.nmon) goto 10
         end do

         do ix = -ialh + 8, -ialh+15

           jb = jb + 1
           x(jb)=ix
           y(jb)=iy
           q(jb)=1.

          if(jb.ge.nmon) goto 10
         end do
       end do

10       jb = nmon
         print*,"goto 10"

         do iy = -ialh + 1 , ialh , 2
          do ix = -ialh ,  -ialh + 15

            jb = jb + 1
            x(jb) =  ix
            y(jb) =  iy
            q(jb) =  -1.

           if(jb.gt.npc+nmon) goto 20
          end do
         end do

        do iy = -ialh + 16, -ialh + 20
          do ix = -ialh ,  -ialh + 15
            jb = jb + 1

            x(jb)=ix
            y(jb)=iy
            q(jb)=-1.

           if(jb.gt.npc+nmon) goto 20
         end do
       end do

20      continue

         open (14,file="init.pos",status="unknown")
          do jb =  1 , n*nmol + npc
            write(14,*)x(jb),y(jb),q(jb)
          end do
```

```
          close(14)
          open (14,file="init.inp",status="unknown")
            write(14,*)10000000
            write(14,*)200
            write(14,*)5137
            write(14,*)1
            write(14,*)nmol*n
            write(14,*)npc
            write(14,*)ialh*2.
c write out batt: energy parameter.
            write(14,*) -1.0
            write(14,*) 0.8,0.5, 'the step size and dlim'
          close(14)

          stop
          end



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C       Read in the initial configuration of a blend                   C
C           and continue simulation                                    C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       program tdl
c     monte carlo simulation of 2-d lattice
      implicit double precision(a-h,o-z)
      real*4 ran
c x,y,z: position of polymers
c xitr,yitr,zitr: trial position of polymers
c map: the occupation condition of the lattice
c qmap: flag for a lattice site: either a polymer or a reacting
particle
c the lattice size here is 10x10,i.e., al=10, alh=5.
c nbtot is total number of polymer units, and npc is number of
reacting particles.
c pmc is the probability function for counting the
c aggregation size of reacting particles.
      parameter (pi=3.141592653589793d0)
      parameter (nbmax=10000,nmax=100)
      dimension x(nbmax),y(nbmax),q(nbmax)
      dimension xitr(nmax),yitr(nmax),qitr(nmax)
      common /map/ map(-10:10,-10:10)
      common /qmap/ qmap(-10:10,-10:10)
      common /pos/ x,y,q,xitr,yitr,qitr
      common /intvar/ nmol,n,nbtot,npc
      common /iyes/ nvl
      common /seed/ nseed
      common /box/ al,alh
      common /ens/ eold,enew
      common /sigs/ deff
      common /corr/ cor(200),pmm(100),pmc(100)
      common /enercons/ batt
      common /stepmove/ stepm, dlim

c     generate random number seed
c     nseed = 2*int(secnds(0.0)) + 2937
```

```
c     read run parameters

      open(unit=1,file='init.inp',status='old')
c     total moves
      read(1,*)ncon
c     moves per accumulation
      read(1,*)nskip
      read(1,*)nseed
      read(1,*)n
      read(1,*)nmol
      read(1,*)npc
      read(1,*)al
c read energy constant
      read(1,*)batt
      read(1,*)stepm,dlim

      close(unit=1,status='keep')

      nbtot = nmol*n
      print*,'nbtot',nbtot,n,nmol,npc,al

      drk = al/100.
      dmm = 0.025
      open(unit=2,file='init.pos',status='old')
      do i = 1, nbtot+npc
       read(2,*)x(i),y(i),q(i)
      enddo
      close(unit=2,status='keep')

       do i1 = 1 , nmol
         do j1 = 2 , n
           kj = (i1-1)*n

           x1 = x(kj+j1) - x(kj+j1-1)
           y1 = y(kj+j1) - y(kj+j1-1)

           x1 = x1 - al*dnint(x1/al)
           y1 = y1 - al*dnint(y1/al)

           rdis = x1*x1 + y1*y1 + 1.0d-4
            if(int(rdis).ne.1) then
             print*, i1, kj+j1, jk+j1-1
            end if
        end do
       end do

      do j1 = -10, 10, 1
        do j2 = -10, 10, 1
           do jp = 1 , nbtot+npc
              xt = x(jp)
              yt = y(jp)
              xt = xt - al*dnint(xt/al)
              yt = yt - al*dnint(yt/al)
              ixt = int(xt+1.0d-4)
              iyt = int(yt+1.0d-4)
              if(j1.eq.ixt .and. j2.eq.iyt) then
                 map(j1,j2)=jp
                 qmap(j1,j2)=q(jp)
              end if
           end do
```

```
        end do
      end do

c compute some constants

      alh=0.5*al

      deff = 1.0d0

c     initialize counters
      ntr = 0
      nsr = 0
      naver = 0
      avener = 0.0d0

c     calculate initial energy

c     begin simulation
      open(16,file="pmm",status="unknown")

      do 1000 icon = 1, ncon
         rr = ran(nseed)
c make a move
       if(rr.lt.dlim) then
         i = int(nmol*ran(nseed) ) + 1
         imol = (i-1)*n
         tntr1 = tntr1 + 1.
c move a polymer
         call rept(i,isuc,icon)
         if(isuc.eq.0)tnsr1 = tnsr1 + 1.

       else
c move a reacting particle
         j = nbtot + int( npc*ran(nseed) ) + 1

         tntr2 = tntr2 + 1
         call move(j,isuc,icon)
         if(isuc.eq.0)tnsr2 = tnsr2 + 1

       end if

       if(mod(icon,nskip).eq.0) then
         tnum = tnum + 1.
         call omega(x,y,q,drk,dmm,summ,sumr2)
       end if

       if(mod(icon,ncon/20).eq.0) then
c        write(6,211)icon,sumr2/tnum/nmol,nsr1/ntr1,nsr2/ntr2
         write(6,211)icon,sumr2/tnum/nmol,tnsr1/tntr1,tnsr2/tntr2
         write(7,211)icon,sumr2/tnum/nmol,tnsr1/tntr1,tnsr2/tntr2
       end if

       if(mod(icon,ncon/2).eq.0) then

         open(14,file='pos', status='unknown')
         do ji = 1 , nbtot + npc
           xt = x(ji)
           xt = xt - al*dnint(xt/al)
           yt = y(ji)
           yt = yt - al*dnint(yt/al)
```

```
          write(14,*)xt,yt
        end do
        close(14)

c       do ji = 1, 50
c         write(15,212)(ji-1)*dmm,pmm(ji)/tnum
c       end do
        do ji = 1, 20
          write(16,211)ji,pmc(ji)/tnum
        end do

      end if

1000  continue
      close (15)
      close (16)

      open(14,file="init.pos",status="unknown")
         do i=1,nbtot+npc
           write(14,*) x(i), y(i), q(i)
         end do
      close (14)

      do i1 = 1 , nmol
        do j1 = 2 , n
          kj = (i1-1)*n

          x1 = x(kj+j1) - x(kj+j1-1)
          y1 = y(kj+j1) - y(kj+j1-1)

          x1 = x1 - al*dnint(x1/al)
          y1 = y1 - al*dnint(y1/al)

          rdis = x1*x1 + y1*y1 + 1.0d-4
           if(int(rdis).ne.1) then
            print*, i1, kj+j1, jk+j1-1
           end if
       end do
      end do

       do i1 = 1 , nbtot + npc - 1
          do  i2 = i1+1, nbtot + npc
          x1 = x(i1) - x(i2)
          y1 = y(i1) - y(i2)

          x1 = x1 - al*dnint(x1/al)
          y1 = y1 - al*dnint(y1/al)

          rdis = x1*x1 + y1*y1 + 1.0d-4
           if(int(rdis).eq.0) then
            print*, i1, i2, rdis
           end if
       end do
      end do

      write(6,*)"acceptace ratio",float(nsr)/float(ncon)

 211    format(1x,i10,f18.4,2f12.4,3f10.5)
 212    format(1x,3f18.10,2f12.4,3f10.5)
```

```
      stop
      end


c This subroutine move a polymer chain using reptation algorithm
      subroutine rept(i,isuc,icon)
c
c     performs a reptation move on linear chain i
c
      implicit double precision(a-h,o-z)
      real*4 ran
      parameter (pi=3.141592653589793d0)
      parameter (nbmax=10000,nmax=100)
      dimension x(nbmax),y(nbmax),q(nbmax)
      dimension xitr(nmax),yitr(nmax),qitr(nmax)
      common /map/ map(-10:10,-10:10)
      common /qmap/ qmap(-10:10,-10:10)
      common /pos/ x,y,q,xitr,yitr,qitr
      common /intvar/ nmol,n,nbtot,npc
      common /iyes/ nvl
      common /sigs/ deff
      common /seed/ nseed
      common /box/ al,alh
      common /ens/ eold,enew
      common /enercons/ batt

      isuc = 1
      if(ran(nseed).gt.0.5) call cvert(i,icon)
      imol = (i-1)*n
      enew = 0.0d0
      eold = 0.0d0

       do j1 = 1 , n
          xitr(j1) = x(imol+j1)
          yitr(j1) = y(imol+j1)
       end do

       call ruv(vx,vy)

       xitr(n) = x(imol+1) + vx
       yitr(n) = y(imol+1) + vy
          kj = imol + n

       call over(i,n)
       if(nvl.eq.1) return

        do j1 = nbtot+1 , nbtot + npc

           dx = xitr(n)-x(j1)
           dy = yitr(n)-y(j1)
           dx = dx - al*dnint(dx/al)
           dy = dy - al*dnint(dy/al)
           rdis = dx*dx+dy*dy + 1.0d-4
           if(int(rdis).eq.1) enew = enew + batt

        end do

        do j1 = nbtot+1 , nbtot + npc

           dx = x(kj)-x(j1)
```

```
            dy = y(kj)-y(j1)
            dx = dx - al*dnint(dx/al)
            dy = dy - al*dnint(dy/al)
            rdis = dx*dx+dy*dy + 1.0d-4
            if(int(rdis).eq.1) eold = eold + batt

         end do

        if(eold.le.enew)then
         boltz = dexp(eold-enew)
         if(boltz.le.ran(nseed))return
        endif

c       move accepted
        xt = x(kj)
        yt = y(kj)
        xt = xt - al*dnint(xt/al)
        yt = yt - al*dnint(yt/al)
        ixt = int(xt+1.0d-4)
        iyt = int(yt+1.0d-4)
        map(ixt,iyt)=0
        qmap(ixt,iyt)=0.
        xt = xitr(n)
        yt = yitr(n)
        xt = xt - al*dnint(xt/al)
        yt = yt - al*dnint(yt/al)
        ixt = int(xt+1.0d-4)
        iyt = int(yt+1.0d-4)
        map(ixt,iyt)=kj
        qmap(ixt,iyt)=q(kj)

        isuc = 0
           j = (i-1)*n + 1
           x(j)=xitr(n)
           y(j)=yitr(n)
        do ij = 2 , n
           j = (i-1)*n + ij
           x(j) = xitr(ij-1)
           y(j) = yitr(ij-1)
        end do
        return
        end

c this subroutine move a reacting particle randomly, maximum jump
size
c is specified by variable stepm.
        subroutine move(j,isuc,icon)
c
c       performs a move on precursor
c
        implicit double precision(a-h,o-z)
        real*4 ran
        parameter (pi=3.141592653589793d0)
        parameter (nbmax=10000,nmax=100)
        dimension x(nbmax),y(nbmax),q(nbmax)
        dimension xitr(nmax),yitr(nmax),qitr(nmax)
        common /map/ map(-10:10,-10:10)
        common /qmap/ qmap(-10:10,-10:10)
        common /pos/ x,y,q,xitr,yitr,qitr
        common /intvar/ nmol,n,nbtot,npc
```

```
      common /iyes/ nvl
      common /sigs/ deff
      common /seed/ nseed
      common /box/ al,alh
      common /ens/ eold,enew
      common /enercons/ batt
      common /stepmove/ stepm, dlim

      isuc = 1
      enew = 0.0d0
      eold = 0.0d0

c         call ruv(vx,vy)
          del1 = (2*ran(nseed)-1)*al*stepm
          del2 = (2*ran(nseed)-1)*al*stepm
          xitr(1) = x(j) + int(del1)
          yitr(1) = y(j) + int(del2)

       call over(0,1)
       if(nvl.eq.1) return
        do i1 = 1 , nmol
         do j1 = 1 , n

          ij = (i1-1)*n + j1

           dx = xitr(1)-x(ij)
           dy = yitr(1)-y(ij)
           dx = dx - al*dnint(dx/al)
           dy = dy - al*dnint(dy/al)
           rdis = dx*dx+dy*dy + 1.0d-4
           if(int(rdis).eq.1) enew = enew + batt

          end do
        end do

        do i1 = 1 , nmol
         do j1 = 1 , n

          ij = (i1-1)*n + j1

           dx = x(j)-x(ij)
           dy = y(j)-y(ij)
           dx = dx - al*dnint(dx/al)
           dy = dy - al*dnint(dy/al)
           rdis = dx*dx+dy*dy + 1.0d-4
           if(int(rdis).eq.1) eold = eold + batt
         end do
        end do

      if(eold.le.enew)then
       boltz = dexp(eold-enew)
       if(boltz.le.ran(nseed)) return
      endif

c     move accepted
      xt = x(j)
      yt = y(j)
      xt = xt - al*dnint(xt/al)
      yt = yt - al*dnint(yt/al)
      ixt = int(xt+1.0d-4)
```

```
      iyt = int(yt+1.0d-4)
      map(ixt,iyt)=0
      qmap(ixt,iyt)=0.
      xt = xitr(1)
      yt = yitr(1)
      xt = xt - al*dnint(xt/al)
      yt = yt - al*dnint(yt/al)
      ixt = int(xt+1.0d-4)
      iyt = int(yt+1.0d-4)
      map(ixt,iyt)=j
      qmap(ixt,iyt)=q(j)

      isuc = 0
         x(j) = xitr(1)
         y(j) = yitr(1)
      return
      end
c this subroutine revert the sequence of a polymer to avoid bias in
reptation.
      subroutine cvert(i,icon)
      implicit double precision(a-h,o-z)
      parameter (nbmax=10000,nmax=100)
      dimension x(nbmax),y(nbmax),q(nbmax)
      dimension xitr(nmax),yitr(nmax),qitr(nmax)
      common /pos/ x,y,q,xitr,yitr,qitr
      common /intvar/ nmol,n,nbtot,npc
      common /sigs/ deff

      imol = (i-1)*n

      do j = 1 , n
       k = n - j + 1
       ik = imol + k
       xitr(j) = x(ik)
       yitr(j) = y(ik)
      enddo
      do ij = 1 , n
        j = imol + ij
       x(j) = xitr(ij)
       y(j) = yitr(ij)
      enddo
      return
      end
c
c this subroutine check overlap for each move
      subroutine over(itry,jtry)
      implicit double precision(a-h,o-z)
      parameter (nbmax=10000,nmax=100)
      dimension x(nbmax),y(nbmax),q(nbmax)
      dimension xitr(nmax),yitr(nmax),qitr(nmax)
      common /pos/ x,y,q,xitr,yitr,qitr
      common /intvar/ nmol,n,nbtot,npc
      common /iyes/ nvl
      common /box/ al,alh
      common /sigs/ deff

        nvl = 1
              if(itry.ne.0) then
                ktry = (itry-1)*n + jtry
              else
```

```
                ktry = 0
              end if
            do i1 = 1 , nmol
             do j1 = 1 , n

                ijk = (i1-1)*n + j1

              if(ijk .ne. ktry) then
                x1 = xitr(jtry)
                y1 = yitr(jtry)

                x2 = x(ijk)
                y2 = y(ijk)

                 xt = x1 - x2
                 yt = y1 - y2
                 xt = xt - al*dnint(xt/al)
                 yt = yt - al*dnint(yt/al)

                rdis = xt*xt+yt*yt + 1.d-4
                if(int(rdis).eq.0) return
              end if

             end do
            end do

             do ijk = nbtot + 1, nbtot + npc
                if(jtry.ne.ijk) then
                  x1 = xitr(jtry)
                  y1 = yitr(jtry)

                  x2 = x(ijk)
                  y2 = y(ijk)

                   xt = x1 - x2
                   yt = y1 - y2
                   xt = xt - al*dnint(xt/al)
                   yt = yt - al*dnint(yt/al)
                  rdis = xt*xt+yt*yt + 1.d-4
                  if(int(rdis).eq.0) return
                end if
             end do

             nvl = 0
          return
          end


          subroutine ruv(x,y)
    c generate a random vector in (0,1),(0,-1),(1,0),(-1,0)
          implicit double precision(a-h,o-z)
          real*4 ran
          common /seed/ nseed

          if(ran(nseed).lt.0.5) then
            if(ran(nseed).lt.0.5) then
              x=1.d0
              y=0.d0
            else
              x=-1.d0
```

```
          y=0.d0
        end if
      else
        if(ran(nseed).lt.0.5) then
          y=1.d0
          x=0.d0
        else
          y=-1.d0
          x=0.d0
        end if
      end if
      return
      end

        function ran(nseed)
        implicit real*4(a-h,o-z)
             ix=nseed
             k1=ix/127773
             ix=16807*(ix-k1*127773)-k1*2836
             if(ix.lt.0) ix=ix+2147483647
             nseed=ix
             xx=ix*4.656612875e-10
           ran=xx
        return
        end

      subroutine omega(x,y,q,drk,dmm,summ,sumr2)
c calculate properties of polymer as well as the cluster size of
reacting particles
        implicit double precision(a-h,o-z)
        parameter(pi=3.141592653589793)
        parameter(nbmax=10000,nmax=100)
        common /map/ map(-10:10,-10:10)
        common /qmap/ qmap(-10:10,-10:10)
        common /corr/ cor(200),pmm(100),pmc(100)
        common /intvar/ nmol,n,nbtot,npc
        dimension x(nbmax),y(nbmax),q(nbmax)

         do i = 1  ,  nmol
            j1 = (i-1)*n + 1
            j2 = (i-1)*n + n
           xt = x(j1) - x(j2)
           yt = y(j1) - y(j2)
           sumr2 = sumr2 + xt*xt + yt*yt
         end do

        do i = -10 , 10
            if(qmap(i,-10).eq.-1.) then
             do j = 9, -9, -1
              if(qmap(i,j).eq.-1.) ica = ica+1
              if(qmap(i,j).ne.-1.) then
                 iend = j
                 go to 12
              endif
             end do
            else
              iend = 9
            end if
12      continue
          do j = -10 , iend , 1
```

```
            if(qmap(i,j).eq. -1.) ica = ica+1
            if(qmap(i,j).ne. -1.) then
               if(ica.gt.0) pmc(ica)=pmc(ica)+1
               ica = 0
            end if
          end do
        end do
        do i = -10 , 10
            if(qmap(-10,i).eq.-1.) then
             do j = 9, -9, -1
              if(qmap(j,i).eq.-1.) ica = ica+1
              if(qmap(j,i).ne.-1.) then
                 iend = j
                 go to 22
              end if
             end do
            else
              iend = 9
            end if
22      continue
          do j = -10 , iend , 1
             if(qmap(j,i).eq. -1.) ica = ica+1
             if(qmap(j,i).ne. -1.) then
                if(ica.gt.0) pmc(ica)=pmc(ica)+1
                ica = 0
             end if
          end do
        end do

50      return
        end
```

Bibliography

(1) Fishman, G. S. *Monte Carlo: Concepts, Algorithms, and Applications*; Springer-Verlag: New York, 1996.

(2) Kalos, M. H.; Whitlock, P. A. *Monte Carlo Methods*; John Wiley & Sons: New York, 1986; Vol. 1.

(3) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087.

(4) Binder, K. *Monte Carlo and Molecular Dynamics Simulations in Polymer Sciences*; Oxford University Press: Oxford, 1995.

(5) Verdier, P. H.; Stockmayer, W. H. *J. Chem. Phys.* **1962**, *36*, 227.

(6) Verdier, P. H. *J. Chem. Phys.* **1966**, *45*, 2122.

(7) Verdier, P. H. *J. Chem. Phys.* **1970**, *52*, 5512.

(8) Verdier, P. H. *J. Chem. Phys.* **1973**, *59*, 6119.

(9) Kron, A. K. *Polymer Sci. (USSR)* **1965**, *7*, 1361.

(10) Wall, F. T.; Mandel, F. *J. Chem. Phys.* **1975**, *63*, 4592.

(11) Lal, M. *Molec. Phys.* **1969**, *17*, 57.

(12) Olaj, O. F.; Pelinka, K. H. *Makromol. Chem.* **1976**, *177*, 3413.

(13) MacDonald, B.; Jan, N.; Hunter, D. L.; Steinitz, M. O. *J. Phys.* **1985**, *A 18*, 2627.

(14) Kolinski, A.; Skolnick, J.; Yaris, R. *J. Chem. Phys.* **1986**, *85*, 3585.

(15) Kolinski, A.; Skolnick, J.; Yaris, R. *Proc. Natl. Acad. Sci.* **1986**, *83*, 7267.

(16) Kolinski, A.; Skolnick, J.; Yaris, R. *Biopolymers* **1987**, *26*, 937.

(17) Carmesin, I.; Kremer, K. *Macromolecules* **1988**, *21*, 2819.

(18) Carmesin, I.; Kremer, K. *J. Phys. (France)* **1990**, *51*, 915.

(19) Paul, W.; Binder, K.; Heermann, D. W.; Kremer, K. *J. Phys. II* **1991**, *1*, 37.

(20) Paul, W.; Binder, K.; Heermann, D. W.; Kremer, K. *J. Chem. Phys.* **1991**, *95*.

(21) Baumgartner, A. *Ann. Rev. Phys. Chem.* **1981**, *75*, 2994.

(22) Mooij, G. C. A.; Frenkel, D. *Mol. Phys.* **1991**, *74*, 41.

(23) Siepmann, J. I. *Mol. Phys.* **1990**, *70*, 1145.

(24) Frenkel, D. *Mol. Phys.* **1992**, *75*, 983.

(25) Rosenbluth, M. N.; Rosenbluth, A. w. *J. Chem. Phys.* **1955**, *23*, 356.

(26) Temperley, H. N. V.; Rowlinson, J. S.; G.S.Rushbrooke. *Physics of simple liquids*; North-Holland: Amsterdam, 1968.

(27) Brush, S. G.; Sahlin, H. L.; Teller, E. *J. Chem. Phys.* **1966**, *45*, 2102.

(28) Zubay, G. *Biochemistry*, 2nd ed.; MacMillan: New York, 1988.

(29) Seto, T.; Nozoye, H. *Chem. Lett.* **1997**, *1997*, 141.

(30) Kajiyama, T.; Khuwattanasil, N.; Takahara, A. *J. Vac. Sci. Technol.* **1998**, *B 16*, 121.

(31) Herve, P.; Destarac, J.-F.; Lal, J.; Oberdisse, J.; Grillo, I. *Europhys. Lett.* **2002**, *58*, 912.

(32) Drzaic, P. S. *J. Appl. Phys.* **1986**, *60*, 2142.

(33) LeGrange, J. D.; Carter, S. A.; Fuentes, M.; Boo, J.; Freeny, A. E.; Cleveland, W.; Miller, T. M. *J. Appl. Phys* **1997**, *81*, 5984.

(34) Nephew, J. B.; Nihei, T. C.; Carter, S. A. *Phys. Rev. Lett.* **1998**, *80*, 3276.

(35) Motoyama, M.; Nakazawa, H.; Ohtaa, T.; Fujisawab, T.; Nakadab, H.; Hayashib, M.; Aizawab, M. *Theor. Comp. Polym. Sci.* **2000**, *10*, 287.

(36) Park, N.-H.; Park, S.-I.; Suh, K.-D. *Colloid and Polymer Science* **2001**, *279*, 1082.

(37) Vorflusev, V.; Kumar, S. *Science* **1999**, *283*, 1903.

(38) Kubo, K.; Ichikawa, M.; Yoshikawa, K.; Koyama, Y.; Niidome, T.; Yamaoka, T.; Nomura, S.-I. M. *Appl. Phys. Lett.* **2003**, *83*, 2468.

(39) Matsuzawa, Y.; Hirano, K.; Mizuno, A.; Ichikawa, M.; Yoshikawa, K. *Appl. Phys. Lett.* **2002**, *81*, 3494.

(40) Nomura, S. M.; Harada, T.;Yoshikawa, K. *Phys. Rev. Lett.* **2002**, *88*, 93903.

(41) D'Aare, R.; Zhou, K.; Doudna, J. A. *Nature* **1998**, *395*, 567.

(42) Maynor, B. W.; Filocamo, S. F.; Grinstaff, M. W.; Liu, J. *J. Am. Chem. Soc.* **2002**, *124*, 522.

(43) Tang, Z.; Kotov, N. A.; Giersig, M. *Science* **2002**, *297*, 237.

(44) Chauhan, B. P. S.; Rathore, J. S.; Chauhan, M.; Krawicz, A. *J. Am. Chem. Soc.* **2003**, *125*, 2876.

(45) Bar-Cohen, Y. *Electroactive Polymer (EQP) actuators as Artificial Muscles -Reality, Potential and Chalenges*; SPIE Press, 2001; Vol. PM98.

(46) Chu, D. Y.; Thomas, J. K. *Macromolecules* **1984**, *17*, 2142.

(47) Tsuneda, S.; Endo, T.; Saito, K.; Sugita, K.; Horie, K.; Yamashita, T.; Sugo, T. *Macromolecules* **1998**, *31*, 366.

(48) Allen, N. S.; Schnabel, W. *Photochemistry and Photophysics in Polymers*; Elsevier Applied Science: London, 1984.

(49) Picarra, S.; Martinho, J. M. G. *Macromolecules* **2001**, *34*, 53.

(50) Gajraj, A.; Ofoli, R. Y. *Langmuir* **2000**, *16*, 8085.

(51) D'Silva, R. P.; Lala, A. K. *Protein Sci.* **1999**, *8*, 1099.

(52) Kelley, A. M.; Michalet, X.; Weiss, S. *Science* **2001**, *292*, 1671.

(53) Tanious, f. A.; Ding, D.; Patrick, D. A.; Bailly, C.; Tidwell, R. R.; Wilson, W. D. *Biochemistry* **2000**, *39*, 12091.

(54) Jadhav, V. R.; Barawkar, D. A.; Ganesh, K. N. *J. Phys. Chem. B* **1999**, *103*, 7383.

(55) Cramer, P.; Bushnell, D. A.; Fu, J. H.; Gnatt, A. L.; Davis, B. M.; Thompson, N. E.; Burgess, R. R.; Edwards, A. M.; David, P. R.; Kornberg, R. D. *Science* **1995**, *288*, 50.

(56) D'Aare, R.; Zhou, K.; Doudna, J. A. *Nature* **1998**, *395*, 567.

(57) de Gennes, P. G. *Scaling Concepts in Polymer Physics*; Cornell University Press: Ithaca, 1979.

(58) Flory, P. J. *Statistical mechanics of chain Molecules*; Interscience: New York, 1969.

(59) Frankel, I.; Mancini, F.; Brenner, H. *J. Chem. Phys.* **1991**, *95*, 8636.

(60) Doi, M.; Edwards, S. F. *The Theory of Polymer Dynamics*; Oxford Science: Oxford, 1986.

(61) Bird, R. B.; Armstrong, R. C.; Hassager, O. *Danymics of Polymeric Liquids*; Wiley: New York, 1977.

(62) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087.

(63) Dill, K. A. *Biochemistry* **1990**, *29*, 7133.

(64) Liang, H.; Chen, H. *J. Chem. Phys.* **2000**, *113*, 4469.

(65) Taylor, M. P. *J. Chem. Phys.* **2001**, *114*, 6472.

(66) Taylor, M. P. *J. Chem. Phys.* **2003**, *118*, 883.

(67) Katakai, R. *J. Am. Chem. Soc.* **1977**, *99*, 232.

(68) Neidigh, J. W.; Fesinmeyer, R. M.; Andersen, N. H. *Nat. Struct. Biol.* **2002**, *9*, 425.

(69) Gellman, S. H.; Woolfson, D. N. *Nat. Struct. Biol.* **2002**, *9*, 408.

(70) Qiu, L.; Pabit, S. A.; Roitberg, A. E.; Hagen, S. J. *J. Am. Chem. Soc.* **2002**, *124*, 12952.

(71) Karasawa, N.; Goddard III, W. A. *Macromolecules* **1992**, *33*, 7268.

(72) Byutner, O. G.; Smith, G. D. *Macromolecules* **1999**, *32*, 8376.

(73) Byutner, O. G.; Smith, G. D. *Macromolecules* **2000**, *33*, 4264.

(74) Binder, K. *Monte Carlo and Molecular Dynamics Simulations in Polymer Sciences*; Oxford University Press: Oxford, 1995.

(75) Cao, J.; Berne, B. J. *J. Chem. Phys.* **1989**, *92*, 1980.

(76) Borkovec, M.; Berne, B. J. *J. Chem. Phys.* **1987**, *86*, 2444.

(77) Shew, C.-Y.; Mills, P. *J. Phys. Chem.* **1993**, *97*, 13824.

(78) Yamasaki, Y.; Yoshikawa, K. *J. Am. Chem. Soc.* **1997**, *119*, 10573.

(79) Yamasaki, Y.; Yoshikawa, K. *Adv. Drug. Delivery. Rev.* **2001**, *52*, 235.

(80) Noguchi, H.; Yoshikawa, K. *J. Chem. Phys.* **1998**, *109*, 5070.

(81) Shew, C.-Y.; Yethiraj, A. *J. Chem. Phys.* **1999**, *110*, 676.

(82) Chang, G.; Guida, W. C.; Still, W. C. *J. Am. Chem. Soc.* **1989**, *111*, 4379.

(83) Honeycutt, J. D. *Comp. Theor. Polym. Sci.* **1998**, *8*, 1.

(84) Bicerano, J. *Comp. Theor. Polym. Sci.* **1998**, *8*, 9.

(85) Bicerano, J. *Predictions of Polymer Properties, 2nd Ed.*; Marcel Dekker: New York, 1996.

(86) Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Oxford, 1987.

(87) Takagi, S.; Tsumoto, K.; Yoshikawa, K. *J. Chem. Phys.* **2001**, *114*, 6942.

(88) Calvo, F.; Doye, J. P. K.; Wales, D. J. *J. Chem. Phys.* **2002**, *116*, 2642.

(89) Grosberg, A. Y.; Khokhlov, A. R. *Statistical Physics of Macromolecules*; American Institute of Physics: New York, 1994.

(90) Xie, P.; Zhou, Q.; Diem, M. *J. Am. Chem. Soc.* **1995**, *117*, 9502.

(91) Wu, C. W.; Sanborn, T. J.; Zuckerman, R. N.; Barron, A. E. *J. Am. Chem. Soc.* **2001**, *123*, 2958.

(92) Mansky, P.; DeRouchey, J.; Russell, T. P.; Mays, J.; Pitsikalis, M.; Morkved, T.; Jaeger, H. *Macromolecules* **1998**, *31*, 4399.

(93) Kinura, H.; Aikawa, K.; Masubuchi, Y.; J., T. *J. Chem. Phys.* **1997**, *107*, 5945.

(94) Xi, K.; Krause, S. *Macromolecules* **1998**, *31*, 3974.

(95) Costa, M. M.; Giacometti, J. A. *Appl. Phys. Lett.* **1993**, *62*, 1091.

(96) Kelley, O. S.; Barton, J. K.; Jackson, N. M.; McPherson, L. D.; Potter, A. B.; Spain, E. M.; Allen, M. J.; Hill, M. G. *Langmuir* **1998**, *14*, 6781.

(97) Frederick, E.; Houssier, C. *Electric Dichroism and Electric Birefringence*; Clarendon Press: Oxford, 1973.

(98) O'Konski, C. T. *Molecular Electrooptics, Part 1-Theory and Methods*; Marcel Dekker: New York, 1976.

(99) Krause, S. *Molecular Electro-Optics*; Plenum Press: New York, 1981.

(100) Onuki, A.; Doi, M. *J. Chem. Phys.* **1986**, *85*, 1190.

(101) Lipson, J. E. G.; Stockmayer, W. H. *J. Chem. Phys.* **1998**, *89*, 3373.

(102) Wirtz, D.; Werner, D. E.; Fuller, G. G. *J. Chem. Phys.* **1994**, *101*, 1679.

(103) Kantor, Y.; Li, H.; Kardar, M. *Phys. Rev. Lett.* **1992**, *69*, 61.

(104) Kantor, Y.; Kardar, M.; Li, H. *Phys. Rev. E* **1994**, *49*, 1383.

(105) Kantor, Y.; Kardar, M. *Phys. Rev. E* **1995**, *51*, 1299.

(106) Srivastava, D.; Muthukumar, M. *Macromolecules* **1996**, *29*, 2324.

(107) Bratko, D.; Chakraborty, A. K. *J. Phys. Chem.* **1996**, *100*, 1164.

(108) Tanaka, M.; Grosberg, A. Y.; Tanaka, T. *Langmuir* **1999**, *15*, 4052.

(109) Yamakov, V.; Milchev, A.; Limbach, H. J.; Dünweg, B.; Everaers, R. P*hys. Rev. Lett.* 2000, 85, 4305.

(110) Dobrynin, A. V.; Rubinstein, M.; Joanny, J. F. M*acromolecules* 1997, 30, 4332.

(111) Dobrynin, A. V.; Obukhov, S. P.; Rubinstein, M. M*acromolecules* 1999, 32, 5689.

(112) Khan, M. O.; Akesson, T.; Jönsson, B. Mac*romolecules* **2001**, *34*, 4216.

(113) Khan, M. O.; Akensson, T.; Jönsson, B. J. Ch*em. Phys.* **2002**, *116*, 3917.

(114) Navarro, S.; Carrasco, B.; Martinez, M. C. L.; De La Torre, J. G. J. Po*lym. Sci. B: Polym. Phys.* **1997**, *3*, 689.

(115) Schiessel, H.; Oshanin, G.; Blumen, A. *J.* Ch*em. Phys.* **1995**, *103*, 5070.

(116) Schiessel, H.; Blumen, A. *J.* Ch*em. Phys.* **1996**, *104*, 6036.

(117) Soddemann, T.; Schiessel, H.; Blumen, A. *Phys. Rev. E* **1998**, *57*, 2081.

(118) Winkler, R. G.; Reineker, P. *J. Chem. Phys.* **1997**, *106*, 2841.

(119) Drzaic, P. S. *J. Appl. Phys.* **1986**, *60*, 2142.

(120) LeGrange, J. D.; Carter, S. A.; Fuentes, M.; Boo, J.; Freeny, A. E.; Cleveland, W.; Miller, T. M. *J. Appl. Phys* **1997**, *81*, 5984.

(121) Nephew, J. B.; Nihei, T. C.; Carter, S. A. *Phys. Rev. Lett.* **1998**, *80*, 3276.

(122) Motoyama, M.; Nakazawa, H.; Ohtaa, T.; Fujisawab, T.; Nakadab, H.; Hayashib, M.; Aizawab, M. *Theor. Comp. Polym. Sci.* **2000**, *10*, 287.

(123) Park, N.-H.; Park, S.-I.; Suh, K.-D. *Colloid and Polymer Science* **2001**, *279*, 1082.

(124) Vorflusev, V.; Kumar, S. *Science* **1999**, *283*, 1903.

(125) Avila, V. L. *Biology - A Human Endeavor*; Bookmark Publishers: California, 1992.

(126) Palczewski, K.; Kumasaka, T.; Hori, T.; Behnke, C. A.; Montoshima, H.; Fox, B. A.; Trong, I. L.; Teller, D. C.; Okada, T.; Stenkamp, R. E.; Yamamoto, M.; Miyano, M. *Science* **2000**, *289*, 739.

(127) Nomura, S. M.; Harada, T.; Yoshikawa, K. *Phys. Rev. Lett.* **2002**, *88*, 93903.

(128) Kubo, K.; Ichikawa, M.; Yoshikawa, K.; Koyama, Y.; Niidome, T.; Yamaoka, T.; Nomura, S.-I. M. *Appl. Phys. Lett.* **2003**, *83*, 2468.

(129) Matsuzawa, Y.; Hirano, K.; Mizuno, A.; Ichikawa, M.; Yoshikawa, K. *Appl. Phys. Lett.* **2002**, *81*, 3494.

(130) Wenczel, R.; Shew, C.-Y. *J. Chem. Phys.* **2001**, *114*, 4717.

(131) Shew, C.-Y. *J. Chem. Phys.* **2003**, *119*, 10428.

(132) Fynewever, H.; Yethiraj, A. *J. Chem. Phys.* **1998**, *108*, 1636.

(133) Yethiraj, A.; Fynewever, H. *Mol. Phys.* **1998**, *93*, 693.

(134) Whittle, M.; Masters, A. J. *Mol. Phys.* **1991**, *72*, 247.

(135) de Gennes, P. G. *Scaling Concepts in Polymer Physics*; Cornell University Press: Ithaca, 1979.

(136) Allen, M. P.; Evans, G. T.; Frenkel, D.; Mulder, B. M. *Adv. Chem. Phys.* **1993**, *86*, 1.

(137) Eppenga, R.; Frenkel, D. *Mol. Phys.* **1984**, *52*, 1303.

(138) Herzfeld, J. *Acc. Chem. Res.* **1996**, *29*, 31.

(139) Maly, I.V.; and Borisy, G.G. *Trends in Cell Biology* **2002**, *12*, 462.

(140) Edelstein, A. S.; Cammarata, R. C. *Nanomaterials: Synthesis, Properties and Applications*; Institute of Physics Publishing, 1996.

(141) Maynor, B. W.; Filocamo, S. F.; Grinstaff, M. W.; Liu, J. *J. Am. Chem. Soc.* **2002**, *124*, 522.

(142) Tang, Z.; Kotov, N. A.; Giersig, M. *Science* **2002**, *297*, 237.

(143) Chauhan, B. P. S.; Rathore, J. S.; Chauhan, M.; Krawicz, A. *J. Am. Chem. Soc.* **2003**, *125*, 2876.

(144) Binder, K. *Monte Carlo and Molecular Dynamics Simulations in Polymer Sciences*; Oxford University Press: Oxford, 1995.

(145) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A.; Teller, E. *J. Chem. Phys.* **1953**, *21*, 1087.

(146) Privman, V. *Finite-Size Scaling and Numerical Simulation of Statistical Systems*; World Scientific: Singapore, 1990.

(147) Mackeown, P. K.; Newman, D. J. *Computational Techniques in Physics*; Adam Hilger: Bristol, 1987.

(148) Shew, C.-Y.; Yethiraj, A. *J. Chem. Phys.* **1996**, *104*, 7665.

(149) M. Muthukumar, M.; Ober, C. K.; Thomas, E. L. *Science* **1997**, *277*, 1225.

(150) Doi, M.; Edwards, S. F. *The Theory of Polymer Dynamics*; Oxford Science: Oxford, 1986.

(151) Vakarin, E. V. *J. Chem. Phys.* **1998**, *109*, 338.