COMPUTER SIMULATION OF POLYMER DYNAMICS

A thesis

Presented to

The Faculty of the Department of Chemistry

The College of William and Mary in Virginia

In Partial Fulfillment

Of the requirements for the degree of

Master of Arts

---

by

Rebecca L. Smithson

1980

APPROVAL SHEET

This thesis is submitted in partial fulfillment of
the requirements for the degree of

Master of Arts

Rebecca L Smithson

Author

Approved, April 1980

David E Kranbuehl

David E. Kranbuehl

Robert A. Orwoll

Robert A. Orwoll

Eric Herbst

Eric Herbst

Table of Contents

List of Tables

List of Figures

ABSTRACT

The concentration dependence of the equilibrium and dynamic properties of random coil polymer chains was studied using Monte Carlo simulations. Chains of 10, 20, and 30 beads were studied at densities ranging from 0.04 to 0.80. Simulations were performed using two types of bead motions. The equilibrium properties sampled were the second and fourth moments of the end-to-end length and the diffusion constant. Relaxation behavior of the chains was studied by sampling the autocorrelation functions of end-to-end length and square end-to-end length.

The effect of concentration on the equilibrium properties agrees with previous studies and theoretical predictions. The faster relaxation modes of the chain appear to be less affected by concentration than the slower modes. The critical density, at which the onset of entangled behavior begins, was found to decrease with increasing concentration. The dependence of relaxation on polymer volume fraction agrees with the free volume theories.

COMPUTER SIMULATION

OF POLYMER DYNAMICS

CHAPTER 1

The Rouse Model

The Rouse Model has been used successfully to describe linear viscoelastic properties of random coil polymer chains in dilute solution[1]. The polymer chain is pictured as being composed of a number of submolecules. Each submolecule is a part of the polymer chain which must be of sufficient length to approximate a gaussian distribution for the separation of end-to-end length. Hydrodynamic interactions, which are the effects of one segment on other segments caused by its motion through the viscous medium, are not included in this model. Excluded volume effects, or the obstruction of a segment by any other segments are also not included. As a result, the Rouse Model is valid only in solutions where these effects are small.

Model of the Polymer Chain

Each polymer molecule is divided up into N identical submolecules. The probability that the end of one of these submolecules is located in the region x to x+dx, y to y+dy, and z to z+dz given that the other end of the molecule is fixed at the origin of this coordinate system is

$$\Psi(x,y,z)\,dxdydz = (\beta/\pi)^{3/2}\exp(-\beta(x\ +y\ +z\ ))dxdydz. \qquad (1.1)$$

The constant β is given by $(3/2l^2)$ where l^2 is the average square length of the submolecules[1]. The number of configurations available to the chain is proportional to $\Psi(x,y,z)dxdydz$. The relative number of configurations thus depends only on $x\ +y\ +z\ = r$ and is given by

$$\exp(-\beta r^2). \qquad (1.2)$$

This model is equivalent to one of N beads connected by N-1 Hookean springs. Treloar[2] shows this by calculating the amount of work required to move an end of the submolecule from r to r+dr. For reversible processes it is known from thermodynamics that at constant temperature and volume

$$dE = TdS + dW \qquad \text{and} \qquad dA = dE - TdS.$$

Combining these two equations gives dA = dW.

The Helmholtz free energy, A, may be found by the equation A = E - TS, where E is the internal energy of the polymer. Therefore, at constant volume and temperature

$$dW/dr = (\partial E/\partial r) - T\partial S/\partial r. \qquad (1.3)$$

By considering a freely jointed chain, or one that is subjected to no internal energy barriers, $dE/dr = 0$, since the internal energy will be the same for all configurations. With this, equation 1.2 becomes

$$dW/dr = -T(S/dr)_{T,V} \tag{1.4}$$

The entropy, S, may be found by Boltzmanns equation, $S = k \ln \Omega$, where k is the Boltzmann constant and Ω is the number of possible configurations. Using equation 1.2, together with the Boltzmann equation gives

$$S = c - k\beta r^2 . \tag{1.5}$$

The constant c includes the size of the volume element dxdydz and is used to indicate that only differences in entropy are of interest, but not absolute values of entropy.

Substituting equation 1.5 into 1.4 gives $dW = 2kT\ rdr$. Because work had to be done to move the end of the subchain from r to r+dr, there is a restoring force acting on the chain such that $dW = fdr = 2kT\beta rdr$, or $f = 2kT\beta r$. The tension is proportional to the length, each submolecule may be thought of as a Hookean spring of equilibrium length zero and spring constant $2kT = 3kT/l^2$ since $\beta = (3/2)l^2$.

In all forthcoming discussion, polymer molecules will be thought of as consisting of N beads connected by N-1 Hookean springs with these properties.

The 3N coordinates of the system will be obtained by assigning a Cartesian Coordinate system to each bead. The coordinate system assigned to the jth bead is (x_j, y_j, z_j) where x_j is the displacement in the x direction of the jth bead from its equilibrium position. The coordinates y and z describe similar displacements in the y and z directions.

Motion of the Polymer Molecule.

The equations of motion for the polymer molecules will be set up in a manner analogous to the method used by Zimm[3], but without his inclusion of hydrodynamic interactions. The general equation used to describe the forces acting on the jth bead of the polymer chain upon application of an external force is the Langivin equation[4],

$$m(dv_{x_j}/dt) = -\rho v_{x_j} + x + A(t) . \qquad (1.6)$$

Similar equations may be written for the y and z dimensions.

The right hand side of the Langevin equation is the sum of the forces acting on the beads. These are

(1) A frictional force, ρv_{x_j}, with friction constant ρ caused by motion of the bead through the solvent. The velocity of the solvent in the x-direction at the jth bead is v_{x_j}.

(2) the term X represents the external, systematic forces exerted on bead j by the springs. The force exerted on bead j by the (j-1)st spring is $-(3kt/l^2)(x_j - x_{j-1})$. The force exerted on this bead by the jth spring is $(-3kt/l^2)(x_j - x_{j+1})$. The total force from the springs on the jth bead is

$$-(3kt/l^2)(-x_{j-1} + 2x_j - x_{j+1}) \text{ for } 0<j<N. \qquad (1.7a)$$

Because beads 1 and N are only affected by springs 1 and N -1 the force on these beads is[5]

$$-(3kt/l^2)(x_1 - x_2) \text{ for } j = 0 \qquad (1.7b)$$

and $$-(3kt/l^2)(x_N - x_{N-1}) \text{ for } j = N . \qquad (1.7c)$$

(3) The force given by A(t) in the Langevin equation is a fluctuating force resulting from the Brownian motion of the solvent. This brownian motion causes the beads to diffuse from regions of high concentration to regions of low concentration. The driving force behind this movement may be written as the change in the Helmholtz free energy[6], $\partial A/\partial x_i$. Expanding this gives $A(t) = \partial A/\partial x = \partial/\partial x_i(E - TS) = T(\partial S/\partial x_i)$. The entropy may be written according to Boltzmanns relations, $S = k\ln\Psi$ with k being Boltzmanns constant. The unknown function Ψ is a function of the coordinates $x_1, x_2 \ldots z_N$. It is to be interpreted as the probability of finding each bead with coordinates between x_i to $x_i + dx_i$, y_i to $y_i + dy_i$ and z_i to $z_i + dz_i$, so is proportional to the number of possible configurations of the system . This leads to

$$A(t) = -kT\,\partial \ln\Psi/\partial x_i . \qquad (1.8)$$

The quantity f/m is a very high frequency on the order of 10^{13} sec^{-1}. Since only the low frequency response of the chain is of interest and $m(dv_{xj}/dt) << f v_{xj}$ the left hand side of equation (1.6) may be set to zero.

Substituting equations (1.7) and (1.8) into equation (1.6) gives

$$\dot{x}_1 = v_{x_1} - D\,\partial \ln\Psi/\partial x_1 - \sigma(x_1 - x_2)\,,$$

$$\dot{x}_j = v_{x_j} - D\,\partial \ln\Psi/\partial x_j - \sigma(-x_{j-1} + 2x_j - x_{j+1}) \quad \text{for} \quad 1 < j < N,$$

$$\dot{x}_N = v_{x_N} - D\,\partial \ln\Psi/\partial x_N - \sigma(x_N - x_{N-1})\,,$$

where $D = kT/\rho$ and $\sigma = 3kT/l^2\rho$.

These equations may be written more compactly in matrix form

$$\partial \underset{\sim}{x}/\partial t = \underset{\sim}{v}_x - D\,(\partial/\partial \underset{\sim}{x})\,\ln\Psi - \sigma A \underset{\sim}{x}\,. \tag{1.9}$$

In this equation

$$\underset{\sim}{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \qquad v_x = \begin{pmatrix} v_{x_1} \\ v_{x_2} \\ \vdots \\ v_{x_N} \end{pmatrix} \qquad \left(\frac{\partial}{\partial \underset{\sim}{x}}\right) = \begin{pmatrix} \partial/\partial x_1 \\ \partial/\partial x_2 \\ \\ \partial/\partial x_N \end{pmatrix}$$

and

$$A = \begin{pmatrix} -1 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & \cdots & 0 & 0 & 0 \\ & & & & \cdots & & & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix}$$

Equations for the y and z dimensions may be written in a similar way.

These equations may now be written in terms of Ψ by use of the equation of continuity,

$$\partial\Psi/\partial t = -\operatorname{div}\Psi[(\partial\underset{\sim}{x}/\partial t) + (\partial\underset{\sim}{y}/\partial t) + (\partial\underset{\sim}{z}/\partial t)]$$

with the divergence operator given by

$$\operatorname{div} = \left(\frac{\partial}{\partial\underset{\sim}{x}}\right)^T + \left(\frac{\partial}{\partial\underset{\sim}{y}}\right)^T + \left(\partial/\partial\underset{\sim}{z}\right)^T.$$

The superscript T is used to indicate the transpose of a vector or matrix. This equation is a statement of the conservation of mass, that the rate at which beads enter some volume element must be equal to the rate at which they leave[8]. Substitution into the equation of continuity for $\partial\underset{\sim}{x}/\partial t$, $\partial\underset{\sim}{y}/\partial t$, and $\partial\underset{\sim}{z}/\partial t$ as given by equation 1.9 yields

$$\partial\Psi/\partial t = \sum_{q=x,y,z}\{-(\partial\Psi/\partial\underset{\sim}{q})^T\cdot\underset{\sim}{v}_q - \Psi\cdot(\partial/\partial\underset{\sim}{q})^T\cdot\underset{\sim}{v}_q + D(\partial/\partial\underset{\sim}{q})^T\cdot(\partial\Psi/\partial\underset{\sim}{q})$$
$$+\sigma(\partial\Psi/\partial\underset{\sim}{q})^T\cdot A\cdot\underset{\sim}{q} + \sigma\cdot\Psi(\partial/\partial\underset{\sim}{q})^T\cdot A\cdot\underset{\sim}{q}. \qquad (1.10)$$

Transformation to Normal Coordinates

Solution of equation 1.10 is simplified by a transformation to normal coordinates. These new coordinates are defined by the equations

$$\underset{\sim}{u} = R^{-1}\underset{\sim}{x}, \qquad (1.11a)$$

$$\underset{\sim}{v} = R^{-1}\underset{\sim}{y}, \text{ and} \qquad (1.11b)$$

$$\underset{\sim}{w} = R^{-1}\underset{\sim}{z}. \qquad (1.11c)$$

The matrix R in these equations is the orthogonal matrix which diagonalizes the A matrix, or

$$R^{-1} AR = \Lambda = (\lambda_p \delta_{pq}). \qquad (1.12)$$

The pth eigenvalue of A is given by λ_p and δ_{pq} is the Kronecker delta.

The rules for transformation of partial derivatives are found by use of the chain rule[8]:

$$\partial/\partial x_j = \sum_K (\partial u_K/\partial x_j)(\partial/\partial u_K) = \sum_K \sum_l [\partial/\partial x_j (R^{-1}_{Kl} x_l)] \partial/\partial u_K$$

$$= \sum_K \sum_l [\partial/\partial x_j (R_{lK} x_l)] \partial/\partial u_K$$

$$= \sum_K \sum_l [R_{Kl} \delta_{jl} \, \partial/\partial u_K]$$

$$= \sum_K R_{jK} \partial/\partial u_K .$$

In matrix notation this becomes

$$(\partial/\partial \underset{\sim}{x}) = R (\partial/\partial \underset{\sim}{u}) . \qquad (1.13)$$

It may be shown in a similar manner that

$$(\partial/\partial \underset{\sim}{u}) = R^{-1} (\partial/\partial \underset{\sim}{x}). \qquad (1.14)$$

The transformation of partial deivatives in the y and z dimensions are of the same form.

Before it is possible to transform the first term of equation 1.10 to normal coordinates, it is necessary to know more about the velocities $\underset{\sim}{v}_q$. Application of a shearing stress to the solution in the x y plane causes the velocity of the solvent to be non zero only in the x direction, as shown in Figure 1.1. Therefore,

$$v_{xj} = \alpha z_j \quad , \quad v_{yj} = v_{zj} = 0. \tag{1.15a}$$

The shear rate α is

$$\alpha = \alpha_0 \exp(i\omega t). \tag{1.15b}$$

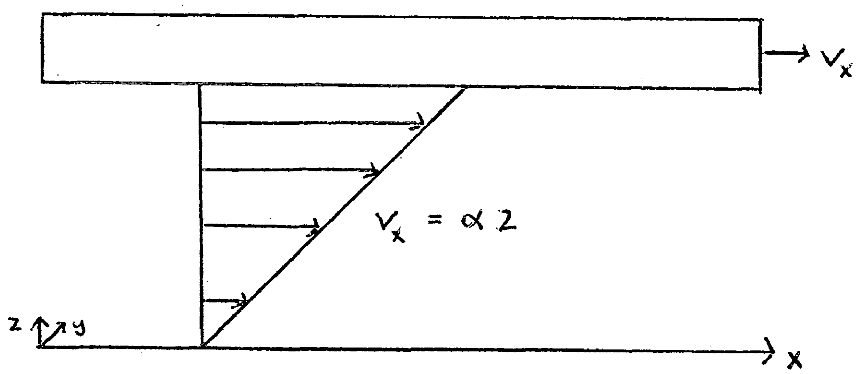


Figure 1.1.
Velocity gradient caused by a shearing motion in the x-y plane.

The velocity gradient is assumed to be constant over the dimensions of a molecule. This assumption will be good except for solvents of low viscosities at high frequencies[1].

The first two terms of equation 1.10 now become

$$-(\partial\Psi/\partial \underset{\sim}{x})^T \cdot \underset{\sim}{v}_x - \Psi(\partial/\partial \underset{\sim}{x})^T \cdot \underset{\sim}{v}_x = -\alpha(\partial\Psi/\partial x)^T \cdot \underset{\sim}{z}_j \quad .$$

This may be written in terms of the coordinates u,v,w by multiplying by $R^{-1}R = E$, where E is the unit matrix as follows :

$$-\alpha(\partial\Psi/\partial \underset{\sim}{x})^T \cdot \underset{\sim}{z} = -\alpha(\underset{\sim}{z})^T \cdot (\partial\Psi/\partial \underset{\sim}{x})$$

$$= -\alpha \underset{\sim}{z}^T \cdot RR^{-1} \cdot (\partial\Psi/\partial \underset{\sim}{x}) = -\alpha \underset{\sim}{z}^T \cdot R \cdot (\partial\Psi/\partial \underset{\sim}{u}).$$

Taking the transpose of both sides of equation 1.11c gives[9]

$$(\underset{\sim}{w}^T) = (R^{-1}\underset{\sim}{z})^T = z^T R^{-1T}.$$

Because R is an orthogonal matrix, $R^{-1} = R^T$ giving

$$(\underset{\sim}{w})^T = \underset{\sim}{z}^T \cdot R, \tag{1.16}$$

with similar equations for $(\underset{\sim}{u})^T$ and $(\underset{\sim}{v})^T$. Using equation 1.13 gives the first term as

$$-\alpha \underset{\sim}{w}^T \cdot (\partial\psi/\partial\underset{\sim}{u}) = -\alpha \sum_i w_i\, \partial\psi/\partial u_i. \tag{1.17}$$

The third term of equation 1.10, the Brownian motion term, may be transformed as follows :

$$D(\partial/\partial\underset{\sim}{x})^T \cdot (\partial\psi/\partial\underset{\sim}{x}) = (\partial/\partial\underset{\sim}{x})^T \cdot RR^{-1} \cdot (\partial\psi/\partial\underset{\sim}{x})$$

$$= (\partial/\partial\underset{\sim}{x})^T (\partial\psi/\partial\underset{\sim}{x})$$

$$= D\sum_i \partial^2\psi/\partial u_i^2 \tag{1.18}$$

Transformation of the spring terms is similar :

$$\sigma[(\partial\psi/\partial\underset{\sim}{x})^T \cdot A \cdot \underset{\sim}{x} + \psi(\partial/\partial\underset{\sim}{x})^T \cdot A \cdot x]$$

$$= \sigma[(\partial\psi/\partial\underset{\sim}{x})^T \cdot RR^{-1} \cdot A \cdot RR^{-1} \cdot \underset{\sim}{x} + \psi(\partial/\partial\underset{\sim}{x})^T \cdot RR^{-1} \cdot A \cdot RR^{-1} \cdot \underset{\sim}{x}]$$

$$= \sigma[(\partial\psi/\partial\underset{\sim}{u})^T \cdot \lambda_k \cdot \underset{\sim}{u} + \psi(\partial/\partial\underset{\sim}{u})\lambda_k \underset{\sim}{u}]$$

$$= \sigma[(\partial\psi/\partial\underset{\sim}{u})^T \cdot \lambda_k \cdot \underset{\sim}{u} + \sigma\lambda_k\psi]$$

$$= \sum_i \sigma\lambda_i [u_i\, \partial\psi/\partial u_i + \psi].$$

(1.19)

Combination of equations 1.17, 1.18, 1.19 and their counterparts for the y and z dimensions gives the diffusion equation for the polymer in the coordinates $\underset{\sim}{u}$, $\underset{\sim}{v}$, and $\underset{\sim}{w}$, as

$$\partial\Psi/\partial t = \sum_i \{-\alpha\ w_i\ \partial\Psi/\partial u_i + \sum_{q=u,v,w} [D\partial^2\Psi/\partial q_i^2 + \sigma\lambda_k [q_i\,\partial\Psi/\partial q_i + \Psi)]\}. \qquad (1.20)$$

Solution of the Differential Equation

Equation 1.20 is easily solved when no shearing motion is involved. In this case, the first term is zero, and the unnormalized solution, Ψ_0 is

$$\Psi_0 = \exp(-\sigma/\partial D)[\sum_{k=1}^{N}(u_k^2 + v_k^2 + w_k^2)]. \qquad (1.21)$$

Differentiating gives

$$\partial\Psi_0/\partial t = (-\sigma/D)\,q_k\,\Psi_0 \qquad q = u, v, w. \qquad (1.22)$$

The solution of 1.20 may be written as a power series in α,

$$\Psi = \Psi_0 \sum_{\mu=0}^{\infty} \psi_\mu \alpha^\mu, \qquad (1.23)$$

$\psi_0 = 1$ and α given by equation 1.15b. Substituting 1.19 into 1.16 and equating the coefficients of α^N gives the recursion equations for ψ_N

$$i\omega\mu\psi_\mu = \sum_{k=1}^{N} [(\sigma/D)\,u_k\,v_k\,\psi_{\mu-1} - w_k\partial\psi_{\mu-1}/\partial u_k]$$

$$+ \sum_{q=u,v,w} (D\partial^2\psi_\mu/\partial q_k^2) - \sigma\lambda_k\,q_k\,\partial\psi_\mu/\partial q_k \qquad (1.24)$$

Solving this equation for ψ_1 will then permit use of equation 1.25 as a recursion relation from which $\psi_2, \psi_3, \psi_4 \ldots$ may be found.

The equation for ψ_1 is

$$i \nu \omega \psi_\nu = \sum_{K=1}^{\nu} \{ (\sigma / D) \upsilon_K \omega_K + \sum_{q=\upsilon,\nu,\omega} \times (D \partial^2 \psi / \partial q_k^2 - \sigma \lambda_k q_k \partial \psi_1 / \partial q_k) \} .$$

A solution for ψ_1 is given by

$$\psi_1 = \sum_{K=1}^{\nu} c_k \upsilon_k \omega_k \qquad , \text{ with}$$

$$c_K = (1/2 D)_k) (1 + (i\omega / 2\sigma\lambda) .$$

For small α, only this first term will be of importance and the relaxation times for the normal modes of vibration of the polymer are then given by

$$\tau_K = 1/(2\sigma\lambda_K) = l^2 P / 6 K T \lambda_K . \quad (1.27)$$

The λ_K are the eigenvalues of the A matrix. These are shown in Appendix A to be

$$\lambda_K = 4 \sin^2 (\pi k/2N) \qquad \text{for } k = 0,1,\ldots N-1. \qquad (1.28)$$

Each of the normal coordinates represents a mode of motion of the polymer chain. These modes are independent of one another, so that superposition or combination of them gives vibrations that are not normal vibrations. All possible vibrations of the molecule can be represented by combinations of the normal vibrations.

Different types of experiments will excite certain modes of motion. For example, mechanical measurements will predominantly excite the lower modes of motion, whereas dielectric type experiments will excite higher frequencies[10].

CHAPTER 2

Monte Carlo Studies of Random Coil Polymer Chains.

A second approach used in investigating both the equilibrium and dynamic properties of polymers is the computer simulation of lattice model polymer chains based on Monte Carlo techniques. This approach makes it possible to study the effects of excluded volume conflicts which are not easily treated analytically.

The Model

A polymer chain is represented in this model as a series of points or beads on a cubic lattice. Each bead lies on one of the cubes vertices and they are connected along the cube sides. Each connection, called a bond, is of unit length. The initial chain configurations are generated randomly. Movement of the chain due to Brownian motion is then simulated by picking either a bead or a bond at random and moving it according to the rules to be given later. Each movement of a bead or a bond is called a bead cycle. Without excluded volume more than one bead may occupy a lattice site at a given time. The excluded volume condition is realized by allowing only one bead per lattice site at any one time.

The beads of a chain are indexed from 1 to N. Let σ_j be the vector pointing from bead j to bead j + 1. The movement of a bead that is not an endbead is accomplished by an interchange of vectors. This interchange for bead j is

$$\sigma_j' = \sigma_{j-1} \quad \text{and} \quad \sigma_{j-1}' = \sigma_j \quad \text{for } 2 \leq j < N-1, \tag{2.1}$$

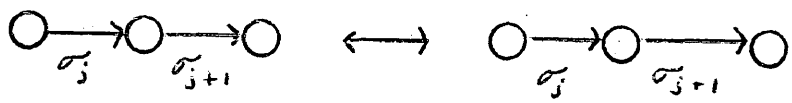
where the prime denotes the vector after the move. This type of vector exchange, which involves the movement of only one bead is pictured in Figure 2.1. It is called a single bead movement. The endbeads, numbered 1 and N are moved by replacing σ_1 and σ_{N-1} by a vector chosen at random from one of the six vectors originating at beads 2 or N-1, respectively.

The second type of bead movement, pictured in Figure 2.2, is called a crankshaft bead movement. It results in the movement of two beads instead of just one. The bond connecting bead j and bead j + 1 is moved by the following

Figure 2.1

The possible Single Bead Moves for Midbeads in the Presence of Excluded Volume.

a) no move possible if the angle between beads is 180°.



b) move across the diagonal if angle is 90° between beads.

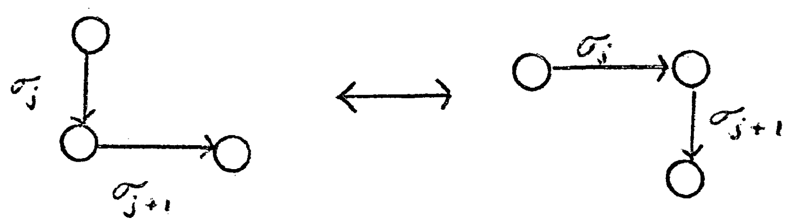
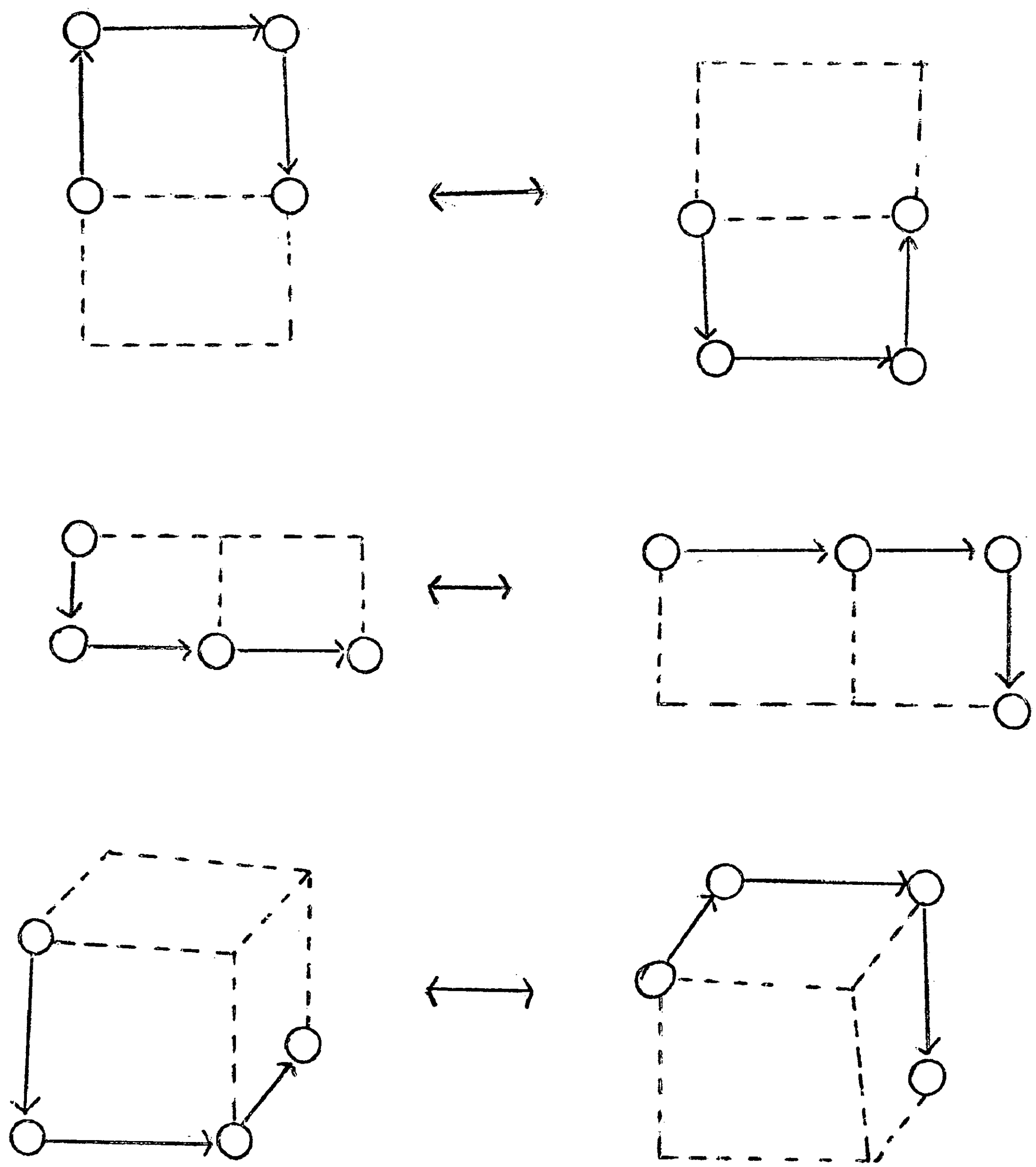


Figure 2.2
The Possible Crankshaft bead moves for midbonds in the Presence of Excluded Volume.



rules :

$$\underset{\sim}{\sigma}'_{j-1} = \underset{\sim}{\sigma}_{j+1} ,$$

$$\underset{\sim}{\sigma}'_{j} = \underset{\sim}{\sigma}_{j} ,$$

$$\underset{\sim}{\sigma}'_{j+1} = \underset{\sim}{\sigma}_{j-1} \qquad \text{for } 1 \leq j \leq N-1. \qquad (2.2)$$

The subscripts 0, 1, N and N - 1 label vectors that are chosen at random from the six possible vectors.

The time interval, in bead cycles, over which the dynamics of the chains are observed is called a frame. Each frame is further divided into samples. The number of bead cycles in each sample varies, and is controlled by input to the program.

At the start of each frame the initial end-to-end length, l_0, of each chain is calculated, as well as l_0^2 and the position of the center of mass. For each subsequent sample, l_t, l_t^2, l_t^4, and the position of the center of mass are again sampled. The products $l_0 \cdot l_t$ and $l_0^2 \cdot l_t^2$ are calculated. This procedure is repeated for some number of frames that is defined by input to the program. This group of frames is called a run.

The products $l_0 \cdot l_t$ and $l_0^2 \cdot l_t^2$ are summed over all the frames for each time, t. The values of l, l^2 , l^4 and the change in the position of the center of mass from the proceeding sample,d ,are added into running sums at the end of each sample. At the end of a run the sums are averaged to the quantities

$\langle l_0 \cdot l_t \rangle$, $\langle l_0^2 \cdot l_t^2 \rangle$, $\langle l^2 \rangle$, $\langle l^4 \rangle$ and $\langle d^2 \rangle$.

From this the correlation functions for each chain as a function of time may be computed as

$$\rho(1,1,t) = \langle l_0 \cdot l_t \rangle / \langle l^2 \rangle,$$

$$\rho(l^2, l^2, t) = (\langle l_0^2 \cdot l_t^2 \rangle - \langle l^2 \rangle^2) / (\langle l^4 \rangle - \langle l^2 \rangle^2). \qquad (2.3)$$

This procedure is repeated for a number of runs, again defined by input. The final configurations of each run are used as the intitial configuration of the next. Final averages are obtained over all runs for each chain, and then over all the chains. The standard deviation of the final averages is used as a measure of the uncertainty in the results.

It is evident that this model emphasizes the chain connectivity properties of a real polymer chain. It is not possible to look at effects due to short range forces within the polymer. Its utility comes, however in looking at the long range, longtime cooperative motions of the chain[11].

CHAPTER 3

Comparison of Monte Carlo Techniques with the Rouse Model

model polymer chains without excluded volume have shown remarkable agreement with the Rouse statistical bead model[12,13]. In particular, a lattice model chain of N beads behave svery much like a Rouse chain of N statistical segments. Theoretical explanations for this have been given by various arthors[14,15,16]. The method used by Verdier[16] will be used here to show the similarities in the relaxation times and correlation functions for the two models. Although only single bead moves are considered here, this method may be extended to include either pure crandshaft or a mixture of crankshaft and single moves[17].

The Relaxation Times

For single bead moves there are two types of bond exchanges which result in bead moves. The first is given by equation 2.1 which is a simple exchange of vectors. The second is one in which an end bond is lost and replaced by one created at random. Calculation of the relaxation of quantities linear in the bond vectors may be accomplished by finding the expectation $\langle \sigma_j(t) \rangle$. This is just the expectation that the ith vector in the initial set of vectors $\{ \sigma_1(0), \sigma_2(0) \cdots \sigma_{N-1}(0) \}$ will migrate in t moves to the jth position. The angular brackets are used to denote the average over all possible motions of the chain for t steps. Any vector that is absorbed by being replaced at the end, or any vector created at the end will not contribute to this expectation. It can be written then as

$$\langle \sigma_j(t) \rangle = \sum_{i=1}^{N-1} \sigma_i(0)\, p_{ij}^{(t)} \qquad (3.1)$$

The $p_{ij}^{(t)}$ are the probibilities of a vector moving from position i to position j in t bead cycles.

Calculation of the $p_{ij}^{(t)}$ for this case is very similar to that of the t-step transition probabilities for a random walk on a line with two absorbing barriers[18]. These t-step probabilities may be calculated with only the knowledge of the one step transition probilities, $p_{ij}^{(1)} = p_{ij}$.

The probability that an interior bond will move one step to the right or one step to the left is given by $p_{j,j+1} = 1/N$ or $p_{j,j-1} = 1/N$. The probability that the bond will not move is $p_{ii} = (N-2)/N$. The matrix of these transition probabilities then looks like

$$\Pi = \begin{pmatrix} K & 1/N & 0 & \cdots & 0 & 0 & 0 \\ 1/N & K & 1/N & \cdots & 0 & 0 & 0 \\ 0 & 1/N & K & \cdots & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1/N & K & 1/N \\ 0 & 0 & 0 & \cdots & 0 & 1/N & K \end{pmatrix}$$

Here the constant K is used to denote (N-2)/N, and Π is a matrix of order N-1.

Diagonalization of this matrix yields the t step transition probablilties. Then, by expressing the bond vectors as normal coordinates which are similar to the Rouse normal coordinates, Verdier[16] shows that the relaxation of these normal coordinates is simple exponential and independent of initial configuration.

The relaxation times are therefore of the form $\exp(-1/\tau_{NK})$ where

$$\tau_{NK} = -\ln(1 - (4/N)\sin^2(k\pi/2N)). \qquad (3.2)$$

Comparison of the relaxation times for the two models requires rewriting equations 1.23 and 3.2 so that they can be expressed in terms of the same unit of time. This is done by computiong the translational diffusion constant, D_{cm}, for both models. For the lattice model, Einsteins relation gives $D_{cm} = 1/6 \cdot \nu \sigma^2$, where ν is the number of times a bead is moved per unit time and σ is the rms magnitude of each displacement[19]. Only one bead is moved each bead cycle, so $\nu = 1$ and $\sigma = 2/N$. The diffusion constant is then $D_{cm} = 2/N^2 \cdot 1/6 = 1/3N^2$.

For the Rouse Model, the diffusion constant of a single bead, D, in equation 1.5 is equal to kT/ρ. This is related to the diffusion of the center of mass by $D = D_{cm}N$.

The relaxation times for the two models may now be written as $\tau_{NK} = (3N^2 D_{cm} \ln(1 - (4/N)\sin^2(k\pi/2N))))$ (3.3) for the lattice model, and

$$\tau_{NK} = [24 N D_{cm} \sin^2(K\pi/2N)]^{-1} \quad K = 0, 1, \ldots N-1 \quad (3.4)$$

for the Rouse Model. In writing 3.4 the mean square extension of a submolecule, l^2 is taken as unity.

Both expressions exhibit the same general behavior. The longest relaxatiion time is for k = 1. Spacing between relaxation times is large for small values of k, but decreases as k increases. The relaxation times for both models approach the same value for large N. Even for N = 8 the difference in the longest relaxation times for the two models is less than 3%[16].

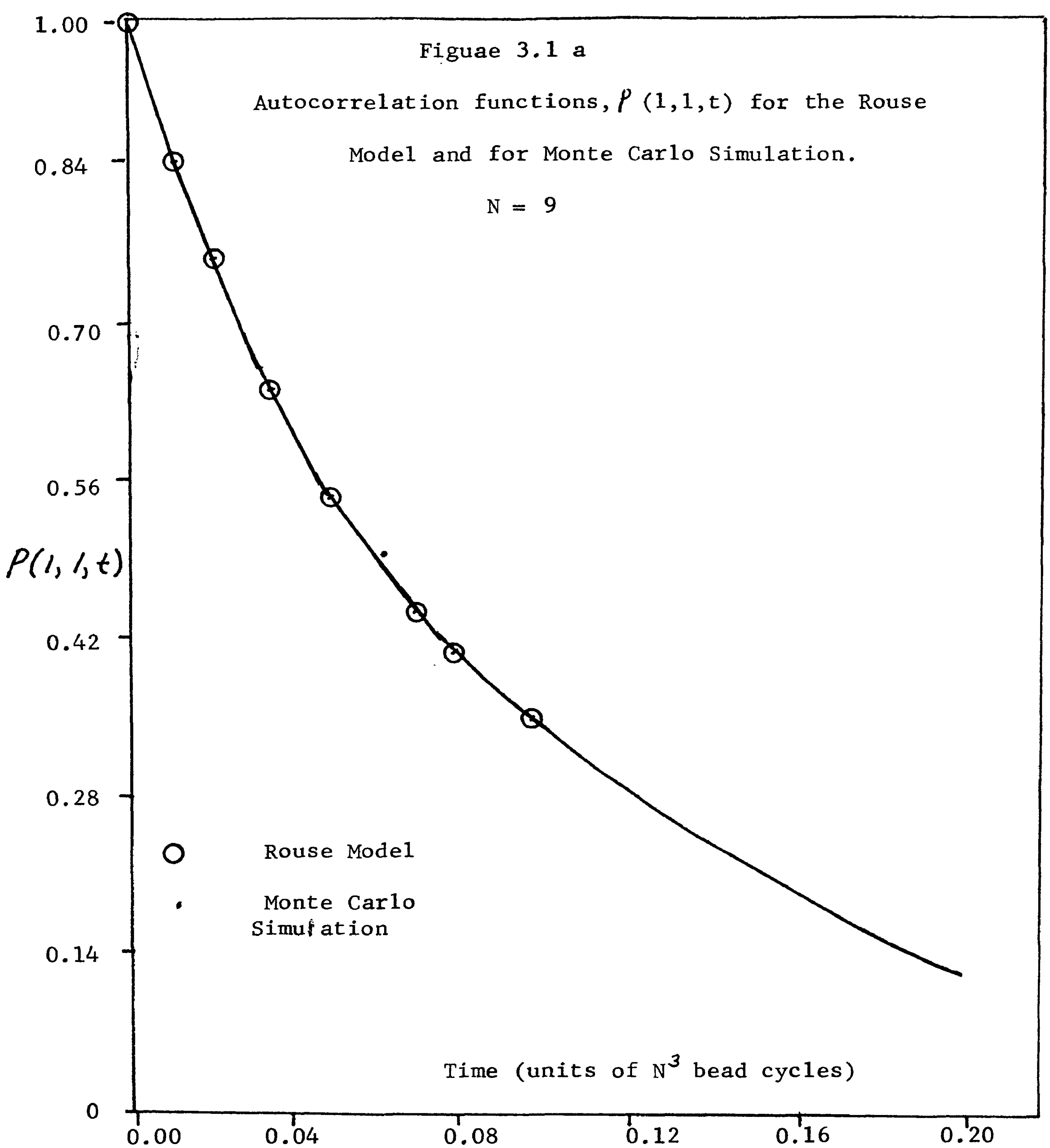
The Correlation Functions

Direct comparison between the Rouse model and computer simulations may be made by calculating the correlation functions of equations 2.3 for the Rouse model. In particular, $\rho(l,l,t) = \langle l_o \cdot l_t \rangle / \langle l^2 \rangle$ may be calculated as follows.

The vector end-to-end length, l may be written as $\underset{\sim}{l} = \underset{\sim}{r}_N - \underset{\sim}{r}_1$ where $r = (x_i, y_i, z_i)$ for bead i. Also, let $\underset{\sim}{u}_m$ be the normal coordinates of the Rouse Model. Then by a simple extension of equation 1.11 the $\underset{\sim}{r}_i$ may be written as $\underset{\sim}{r}_i = \sum_{m=0}^{N-1} R_{jm} \underset{\sim}{u}_m$. Since R is an orthogonal matrix, its inverse is equal to its transpose and $R^{-1}_{jm} = R_{mj} = (2/N)^{1/2}\cos(n\pi(j - 1/2)/N)$. Let $B_m = R_{Nm} - R_{om} = (2/N)^{1/2}\cos(\pi n(N - 1/2)/N) - (2/N)^{1/2}\cos(-1/2n\pi/N)$. The correlation functions for vector l may now be written as

$\rho(l,l,t) = \langle l(t)\cdot l(0)\rangle / \langle l^2\rangle = \sum_{i=0}^{N-1}\langle B_i^2 \underset{\sim}{u}_i(t)\underset{\sim}{u}_i(0)\rangle / \sum_{i=0}^{N-1}\langle B_i^2 \underset{\sim}{u}_i(0)\rangle$.

The normal coordinates, $\underset{\sim}{u}_K$ may be written as $\underset{\sim}{u}_K = \langle u_K^2\rangle \exp(-t/\tau_K)$ where τ_K is the relaxation time of the kth normal mode of vibration and $\langle u_n^2\rangle_e$ is the mean square of the coordinate at equilibrium. Verdier shows that $\langle u_i^2\rangle_e = (1/4)N\sin^2(i\pi/2N) = 1/\lambda_i$. This gives $\rho(l,l,t) = \sum_{i=0}^{N-1}(B_i^2 / \lambda_i^2)\exp(-t/\tau_i) / \sum_{i=0}^{N-1}(B_i^2 / \lambda_i^2)$. These correlation functions were calculated by computer. The graphical results of these calculations for N = 9 and 15 are shown in figures 3.1a and b. Also shown are the results for Monte Carlo simulations for the same chain lengths.



Figuae 3.1 a

Autocorrelation functions, ρ (1,1,t) for the Rouse Model and for Monte Carlo Simulation.

N = 9

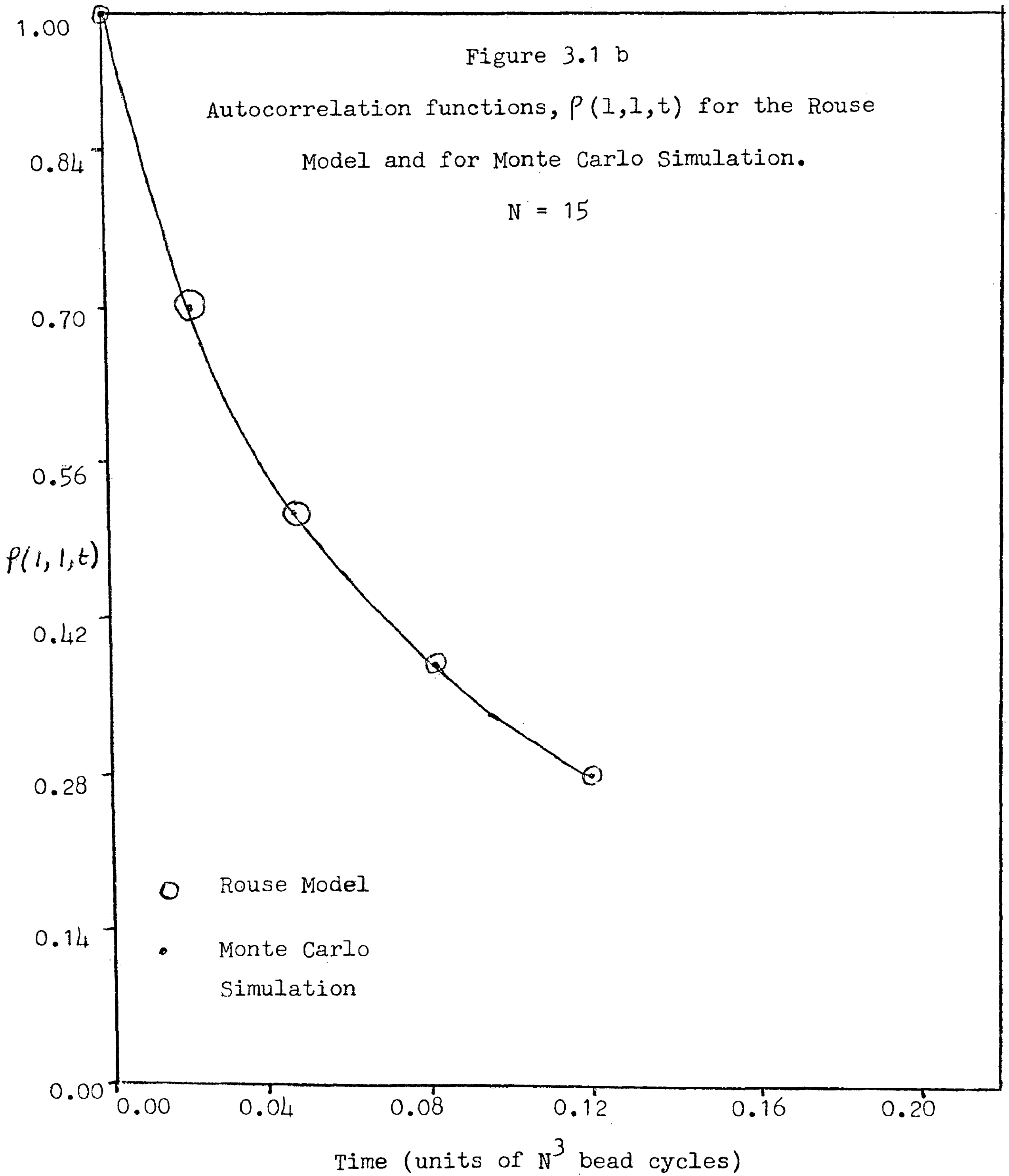


Figure 3.1 b
Autocorrelation functions, ρ(1,1,t) for the Rouse
Model and for Monte Carlo Simulation.
N = 15
1.00
0.84
0.70
0.56
ρ(1,1,t)
0.42
0.28
0.14
0.00
Rouse Model
Monte Carlo
Simulation
0.00
0.04
0.08
0.12
0.16
0.20
Time (units of N3 bead cycles)

CHAPTER 4

Computer Simulation of the Equilibrium and Dynamic Properties of Random Coil Polymer Chains.

The concentration dependence of both the equilibrium and dynamic properties of polymer chains was studied. Computer experiments for chain lengths of 10, 20 and 30 were performed with concentrations ranging up to 0.8. Both single bead movement rules and mixed rules were used. In order to minimize box edge effects, boxes the size of the chains were used in as many simulations as possible. Smaller boxes were used in some of the runs at higher concentrations because of limitations imposed by computer time. Previous experiments have been performed by Bellemans for equilibrium properties only, and for chains of the same lengths. Kranbuehl and Schardt studied the concentration dependence for both the equilibium and dynamic properties for chain lengths of 10 and 20 for concentrations of up to 0.6 using single bead movement rules and smaller boxes than were used in this study.

Equilibrium Properties

The equilibrium data sampled in each simulation are $\langle l^2 \rangle$, $\langle l^4 \rangle$ and $\langle d^2/\text{frame} \rangle$. Also of interest is the expansion factor, α. This quantity may be defined by the equation

$$\langle l^2 \rangle = \alpha^2 \langle l_0^2 \rangle .$$

The average end-to-end length squared of the unperturbed chain is given by l_0^2. By unperturbed is meant the dimensions of the random flight or guassian chain, in which the end-to-end length is affected only by bond angles and bond lengths. A polymer molecule will assume unperturbed dimensions at the theta temperature, where deviations from ideality vanish. However, in a good solvent at non-theta conditions the attraction of the polymer to the solvent is greater than the attraction of the polymer segments for each other. The polymer will tend to expand so as to minimize these less favorable contacts. Similarly, in a poor solvent at non-theta conditions polymer-polymer contacts are favored over polymer-segment contacts and the chain will contract[20].

An expression for $\langle l_0^2 \rangle$ may be obtained from equation 1.1 by performing the integration as described by Flory[20]. The result is $\langle l_0^2 \rangle = x$, where x is the number of bonds in the chain, and $\langle l_0^2 \rangle$ is expressed in units of the bond length. This gives $\alpha = \langle l^2 \rangle / (N - 1)$, where N is the number of beads in the chain. The expansion factor is expected to be unity under ideal conditions. Fixman and Peterson[21] have derived an analytical expression which describes the concentration dependence of α in a good solvent. Initially, α will be greater than one. As the concentration increases in dilute solutions, α decreases rapidly. This decrease slows at higher concentrations, and α approcaches unity as ρ^{-1} where ρ is the concentration of the solution.

Studies by Bishop et al[22], Wall et al[23], Okamoto[24], Kranbuehl and Schardt[25], De Vos and Bellemans[26,27], and Curro[28,29,30], have shown that for chains in dilute solution $\langle l^2 \rangle \propto (N - 1)^{1.2}$. As the solution becomes more concentrated the chains contract and the polymer sizes approach a proportionalality of $(N - 1)^{1.0}$ in the bulk phase. This implies that $\alpha > 1$ for dilute solution and becomes one in bulk. This is in line with Flory's reasoning that in the bulk phase the inter-chain repulsions are exactly balanced by the intra-chain repulsions. Therefore the chain will not contract or expand, and $\langle l^2 \rangle$ will be proportional to $(N - 1)^{1.0}$. An $(N - 1)^{1.0}$ dependence in the bulk phase has been observed by Cotton et al[31] by neutron scattering experiments on bulk polystyrene.

The average squared distance traveled per frame, $\langle D^2 \rangle$/frame, is another property sampled by the program. This is used to calculate the translational diffusion constant, D_t, given by $D_t = (1/6)\langle D^2 \rangle / t$. The time t is the number of bead cycles in a frame divided by the total number of beads cubed. This time was chosen because a frame length, whenever possible was one relaxation time. It is not known if this time interval is of sufficient length for the longest range motions of the chain to occur. Data for diffusion over longer time intervals would help to answer this question.

The actual calculation of the chains diffusion became complex because the chain was allowed to penetrate the box walls. This means that when a bead hit a box wall it was moved to the other side of the box. The continuity of the chain was always preserved. Each such hit had to be recorded and used to calculate the diffusion for each chain. Details of this procedure are discussed in Appendix B.

Two factors were found to affect the diffusion of the chains. The first of these is the frame length, the diffusion being greater for greater frame lengths. The second is concentration. Figures 4.1a and 4.1b show the effect of concentration on diffusion for 20 and 30 bead chains. These are graphs of the probability distribution of the average square diffusion of the center of mass of the chain with diffusion measured in the number of lattice sites traveled. As can be seen in these plots, increasing concentration causes the distribution to become sharper and the chains to travel less distance.

Exact calculation of diffusion was not possible using our algorithm if the number of beads in the chain was greater than the box length. However, because of computer time restrictions it became necessary to go to smaller boxes for the higher concentrations of the 20 and 30 bead chains. The diffusion of chains in these smaller boxes was calculated by assuming that the chain would travel no more than half the box length. Selection of a box size for which this condition would be met was made using Figures 4.1a and 4.1b.

The approach used in calculating the box size and frame length for these smaller boxes is exemplified by the method used in calculating these quantities for concentrations of 0.5 and greater for the 20 bead chains. Analysis of Figure 4.1a indicates that by choosing a box size of 10 and not increasing the frame length no chain should travel more than 5 lattice sites when the density is increased to 0.5.

This box size of 10 was used for all concentrations of 0.5 and higher. The frame length for the density of $\rho = 0.6$ was chosen by multiplying the frame length used at $\rho = 0.5$ by the quantity $D(\rho = 0.4)/D(\rho = 0.5)$. This allows the frame length to increase within safe limits by taking into account the fact that as density increases the diffusion will be less. Frame lengths for densities of 0.7 and 0.8 were chosen in the same way. A similar analysis of Fig. 4.1b led to the the use of a box size of 14 for 30 bead chains.

Dynamic Properties

As a measure of the dynamics of the system, the autocorrelation function,

$$\rho(1,1,t) = \langle 1_o \cdot 1_t \rangle / \langle 1^2 \rangle, \qquad \text{and}$$

$$\rho(1^2, 1^2, t) = (\langle 1_o^2 \cdot 1_t^2 \rangle - \langle 1^2 \rangle^2)/(\langle 1^4 \rangle - \langle 1^2 \rangle^2)$$

were obtained from the program. The time, t, is measured in N bead cycles, where N is the total number of beads. These autocorrelation functions are normalized so that they have a value of one at t=0 and decrease to zero as time increases. This relaxation behavior of the autocorrelation functions is characteristic of the overall relaxation behavior of the chain. It is not simple exponential. However, for times after which the autocorrelation function has dropped to about 1/2, it is nearly simple exponential and may be fitted to a function of the form $\rho(t) = \exp(-t/\tau_s)$. This part of the graph is fitted to a function of the form $\ln \rho(1,1,t) = \ln A - t/\tau_s$. The relaxation time, τ_s, may then be calculated from the limiting slope. This relaxation time should be representive of the longest relaxation time found in the Rouse Model. It will therefore involve the long range motions of the chain.

A second relaxation time, $\tau_{1/e}$, is defined as as the time it takes $\rho(1,1,t)$ to relax to a value of 1/e. This relaxation time is characteristic of the overall relaxation of the chain.

Results and Discussion

The results for the equilibrium properties are shown in tables 4.1 and 4.2. Previous studies of the concentration effects on the equilibrium dimensions were made by De Vos and Bellemans[26,27]. For low concentration Bellemans used the same model as in the present study, with crankshaft bead movement rules. Samples of $\langle l^2 \rangle$ were made about every 200 bead moves. Because this model has a large number of excluded volume conflicts at high concentrations resulting in large amounts of computer time needed, a different approach was taken. Bellemans approach was to use single bead rules but to pick an empty lattice site at random. If an adjacent bead could move to this site it was moved. Bellemans data is summarized in Table 4.3. Figure 4.2 shows a graphical comparison of our model to Bellemans. Mixed bead movement results are shown, the results of the single bead movements are very similar. All points are within one standard deviation of Bellemans results. This data shows that $\langle l^2 \rangle$ decreases with increasing density. The last four points fit better to $\rho^{-.32}$ than ρ^{-1}.

The dependence of end-to-end length on chain length may be found by taking the slope of lines generated by a log-log plot of N - 1 vs $\langle l^2 \rangle$. The results are shown in Table 4.4

The parameter a is the exponent in the equation $\langle l^2 \rangle \propto (N - 1)^{\alpha}$. It is simply the slope of the plot of $\log\langle l^2 \rangle$ vs $\log(N - 1)$. Results compare with those of Bishop *et al*[22] and Bellemans[26,27]. They show a decrease in the exponent a from 1.2 for a single chain to a value approaching 1 in bulk.

That chains in concentrated solutions show the behavior of $\langle l^2 \rangle \propto (N - 1)^{1.0}$ means that they behave as Gaussian or ideal chains. This does not mean that the complete distribution of chain segments is gaussian, since the distribution is uniquiely ditermined by all even moments, not just the second.

The average end-to-end length squared vs density is plotted in Fig 4.3. The results have been extrapolated to a density of 1. The theoretical results for $\langle l^2 \rangle$ from second order (5 choice) random wald calculations on a cubic lattice at this density are guven by Curro[30] as

$$\langle r^2 \rangle = ((1 + \delta)/(1 - \delta))n - 2 \quad (1 - \delta^m)/(1 - \delta^m)$$

where $\delta = 1/(q - 1)$ and q is the coordination number of the lattice (6 for a cubic lattice). The extrapolated results are close for all three chain lengths, as expected. No exact expression was readily available for $\langle l^4 \rangle$, but the theoretical values for $\langle l^4 \rangle / \langle l^2 \rangle^2$ were obtained from Curro as 1.47, 1.51 and 1.55 for chain lengths of 10, 20 and 30 respectively. Plots of $\langle l^4 \rangle / \langle l^2 \rangle^4$ for all three chain lengths vs density are given in Fig 4.4. They are again extrapolated to a density of 1, and the agreement is good with the results for a gaussian chain.

The distribution function W(r) is completely determined by the even moments. It is a slowly converging series, however, so it can only be written in terms of just the second and fourth moments as a rough approximation. Curro gives this as

$$W(r) \sim \pi^{-3/2}\exp(-\rho^2)((1 + 15g_4) - 20g_4\rho^2 + 4g\rho^4)$$

$$\text{where } \rho^2 = 3r^2/2\langle r^2\rangle \text{ and}$$

$$g = 0.125((3/5)\langle r^4\rangle/\langle r^2\rangle^2 - 1).$$

The concentration dependence of $\langle l^2\rangle$ and $\langle l^4\rangle/\langle l^2\rangle^2$ with the ideal chain values suggests that this approximate distribution is the same for all chain lengths for the bulk system and the gaussian chain. Data for the higher even moments would be necessary to show with a higher certainty that the two distributions are the same.

The concentration dependence of $\langle l^2\rangle$ is predicted by theory[31] to be $\langle l^2\rangle \sim \rho^{-\gamma}$ with $\gamma = 0.25$ in three dimensions. Least squares fits of the plots of $\log\langle l^2\rangle$ vs $\log\rho$, shown in Figure 4.5, gave values of γ to be 0.07 $\pm$ 0.005, 0.12 $\pm$ 0.05 and 0.13 $\pm$ 0.04 for chain lengths of 10, 20, and 30. Table 4.4b shows a comparison of those of Curro[30] and Bishop _et al_[22]. Our results are within one standard deviation of both sets of data. The exponents for the small chain lengths are less than predicted by theory. For the longer chain lengths used by Bishop et al[22], the exponents did approach the expected result of 0.25.

Plots of D_t vs. bead density are shown in Fig. 4.6a for mixed bead movements and in Fig. 4.6b for single bead movements. The diffusion constant appears to be independent of chain length. D_t decreases linearly with increasing concentration. This decrease might be expected to be less pronounced because frame lengths of less than one relaxation time were used for the higher concentrations of the 20 and 30 bead chains. This length of time is most likely not sufficient for the long range motions of the chain to have occured, with the result that D_t is too small .

Plots of ln $\rho(1,1,t)$ vs. time are shown in Figures 4.7a through 4.7h. The time is measured in N^3 bead cycles, with N being the total number of beads in the box. The error bars correspond to one standard deviation of the mean.

The Dynamic results are shown in tables 4.5 and 4.6. Figure 4.8 is a plot of τ_s vs. bead density. The behavior of τ_s is different for the mixed and single bead movement rules. It is the same, however, for the three chain lengths using a given rule. τ_s varies much more slowly at low concentrations for the mixed plots than for the single ones. Behavior is similar for the two bead movement rules for concentrations above about 0.5.

For the mixed moves, τ_s changes very little at the low densities, but increases very rapidly at the high densities. This effect is much less pronounced for the single moves. A behavior was seen by Kranbuehl and Verdier[17] in the study of the effect of bead movement rules on isolated chains of various lengths. They found that τ_s increased much more slowly for mixed moves at small chain lengths (10 to 20 beads) than for either pure single or pure crankshaft moves. They also found that the increase in τ_s with chain length was similar for all bead movement rules at longer chain lengths.

The smaller dependence of τ_s on density at low densities than at high densities may be explained as follows. At low densities each chain is isolated from the others, each chain forming a cluster. As chains are added this situation does not change until some critical density, ρ^* is reached. At this point the chains become entangled and the relaxation is slowed by these interchain excluded volume effects. At this density τ_s begins to rise sharply with increasing density.

Figures 4.9a and b are log-log plots of τ_s vs density. The concentration, ρ^* ,at which entanglement begins is the concentration at which the two lines intersect. These concentrations are given in Table 4.7. Schaefer, Joanny and Pincus have predicted that ρ^* is proportional to $1/(N-1)^{4/5}$. As expected, the critical density decreases with increasing chain length. It does not decrease as rapidly as expected, however. Little can be seen from the runs using single bead moves because runs were not taken to high enough densities.

Figures 4.10a and 4.10b are plots of ln τ_s vs. ln(N) for the various densities. Table 4.8 shows the slopes for each of these lines. For both types of bead movement rule the dependence on chain length increases with increasing density. This dependence is much larger in the single bead movement rules than for the mixed ones.

Figures 4.10c and 4.10d are similar plots of ln $\tau_{1/e}$ vs. ln(N). The overall relaxation of the chain shows little change in its N-dependence with increasing densities until a density of ρ =0.6, at which point it also begins to increase.

The dimensionless quantity $\tau_s D_t / \langle 1^2 \rangle$ has been found to be nearly linear, as shown in Table 4.5. This gives

$$D_t \propto \tau_s / \langle 1^2 \rangle . \tag{4.1}$$

Einsteins equation relates D to the viscosity, η , by the equation

$$D_t \propto 1/\eta . \tag{4.2}$$

Combining equations 4.1 and 4.2 gives

$$\eta \propto \langle 1^2 \rangle / \tau_s . \tag{4.3}$$

Cohen and Turnbull derived a relation between the diffusion and free volume by considering transport in a liquid of hard spheres. The molecules are pictured as traveling in a cage consisting of their nearest neighbors.

Diffusion occurs when a hole large enough for the molecule to move through opens up in the cage and another molecule occupies its original position before it is able to return. Using this approach they found that

$$D = D(v^*)\exp(-\gamma\ v^*/v_f\ ). \tag{4.4}$$

In this equation v* is a critical volume just large enough to allow another molecule to jump in after the displacement. The average free volume per molecule, v_f , is the volume within the cage minus the volume of the molecule. The contribution of the molecule to the diffusion coefficient is D(v*), as D(v) is zero if v < v*. D(v*) is equal to ga*u with a* being the molecular diameter, u the velocity of the molecule and g is a geometrical factor. The final quantity, γ is a numerical factor introduced to include the overlap of free volume between molecules and should lie between 1/2 and 1.

An emperical relation of the same form, $\eta = A\exp(Bv_f/v_o)$ was found by Doolittle for simple hydrocarbons[36] . Here v_f is the free volume of the molecular unit or bead and v_o is the volume of the bead.

Combining equations 4.2, 4.3, and 4.4 gives

$$\langle l^2\rangle / \tau_s \ \propto \ D(v^*)\exp(\gamma\ v^*/v_f\ ). \tag{4.5}$$

The quantity γ v* must be nearly equal to the molecular volume, giving

$$\langle l^2\rangle / \tau_s \ \propto \ \exp(1/v_f).$$

The free volume, v_f , is the total free volume divided by the number of beads, N. Letting V be the total volume, $v_f = (V - N)/N = (1-\rho)/\rho$. Therefore $\langle l^2\rangle/\tau_s \propto (\rho/1-\rho)$. A plot of $\ln\langle l^2\rangle/\tau_s$ vs. $\rho/(1-\rho)$ is given in Figure 4.11. As predicted it is a linear graph. Also $\langle l^2\rangle/\tau_s$, which is proportional to D_t is independent of chain length.

Table 4.9 shows the values for $\tau_{1/e}(l^2)$ and $\tau_s(l^2)$ for the correlation functions $\rho(l^2, l^2, t)$. Plots of $\ln \rho(l^2, l^2, t)$ vs time are shown in Figures 4.12a and b. The dependence of $\tau_s(l^2)$ on concentration is shown in figure 4.13. There appears to be no difference based on chain lengths. Figure 4.14 shows that a break point occurs at a density of 0.46. This is comparable to the density at which the break point occured in τ_s for the longest chain length studied.

The effect of concentration on equilibrium properties agrees with previous studies. $\langle l^2\rangle$ is proportional to $(N - 1)^a$ where the exponent a is near 1.2 for the single chain and approaches an exptrpolated value of 1.0 in bulk. Agreement of $\langle l^2\rangle$ and $\langle l^4\rangle / \langle l^2\rangle^2$ at high concentrations with random walk calculations in bulk is good. This suggests that the distribution of end-to-end length for the bulk system is consistant with that of the gaussian chain. The exponent in the equation $\langle l^2\rangle \propto (N - 1)^a$ was found to be lower than predicted by theory. The exponent agrees with previous simulations, which show that the theoretical value is approached as chain length is increased. Diffusion decreases linearly with concentraiton and is not dependent on chain length.

The behavior of the longtime relaxation, τ_s, is different for mixed and single bead movement rules at low concentration. Behavior is similar at high concentrations. The critical density was found to decrease with increasing chain length, but at a much slower rate than predicted by theory. The N dependence of τ_s increases with increasing density. This increase appears to be stronger for the single bead movement rules than for the mixed ones. The slope of $\ln(\tau_s)$ vs. N increases by a factor of 1.33 for the mixed moves vs. 1.73 for the single moves in the concentration range 0.04 to 0.5. The N dependence of $\tau_{1/e}$ is less pronounced for low concentrations than high ones. Again this increase is larger for single moves. For single moves the slope of $\ln(\tau_{1/e})$ vs. N increases by a factor of 1.45 vs. an increase of 1.08 for the mixed moves in the concentration range 0.04 to 0.5. The overall increase in the N dependence of the long-time relaxation is less than that of the overall relaxation. This suggests that the short range, faster relaxation modes are less affected by concentration changes than the long range coperative relaxation modes.

The quantity $\ln \langle l^2 \rangle / \tau_s$ vs $(\rho / 1-\rho)$ is linear as predicted by the free volume theories of Doolittle and

Williams, Landel, and Ferry. The free volume theory provides a good description of the dependence of the relaxation time on polymer volume fraction.

Table 4.1

Mixed Bead Movements. Equilibrium Data.

Length	#Runs	#Frames	Box	Dens	$\langle D^2 \rangle$	Std	$\langle D_+ \rangle$	$\langle L^2 \rangle$	Std	$\langle L^4 \rangle$	Std	α^2	Std
9	40	200	32	----	2.95	0.17	3.28	13.31	0.39	224	12	0.41	0.04
10	50	100	10	.04	2.38	0.12	3.30	14.70	0.09	297	3	1.63	0.01
10	10	100	10	0.20	2.58	0.10	2.39	14.29	0.20	280	8	1.59	0.02
10	8	100	10	0.40	2.46	0.13	1.33	13.87	0.21	286	7	1.54	0.02
10	6	50	10	0.50	2.20	0.17	0.93	13.63	0.54	256	19	1.49	0.06
10	10	20	10	0.60	1.93	0.26	0.36	13.43	0.54	253	19	1.49	0.06
10	6	20	10	0.70	1.44	0.22	0.34	13.10	0.68	244	24	1.46	0.08
10	8	8	10	0.80	1.53	0.26	0.12	13.30	0.63	252	21	1.48	0.07
15	40	100	32	----	5.00	0.52	4.63	25.23	1.1	887	69	1.80	0.08
20	10	25	20	0.04	5.86	0.59	4.65	36.1	0.9	1848	81	1.90	0.05
20	6	10	20	0.20	4.36	1.40	2.91	34.7	2.0	1727	287	1.69	0.17
20	4	4	20	0.40	4.09	1.13	1.79	31.9	3.2	1506	287	1.68	0.17
20	12	15	20	0.50	4.32	0.28	1.19	31.8	1.3	1447	110	1.67	0.07
20	12	10	20	0.60	4.57	8.00	1.26	30.5	2.0	1389	1.61	1.61	0.11
20	4	15	20	0.70	1.56	0.17	0.42	30.5	3.5	1425	331	1.61	0.18
20	4	15	20	0.80	0.27	0.13	0.05	28.6	13	1212	564	1.51	0.68
32	20	25	32	----	12.5	3.08	6.94	66.01	5.7	6239	1068	2.13	0.18
30	4	20	30	0.06	9.00	1.90	5.74	58.90	3.0	4994	498	2.03	0.10
30	8	15	30	0.20	8.06	3.06	3.60	55.61	7.0	4569	1105	1.92	0.24
30	6	6	14	0.40	5.22	0.69	2.07	51.60	5.0	4040	814	1.78	0.17
30	4	4	14	0.50	4.19	0.50	1.44	50.39	4.5	3817	690	2.68	0.16
30	4	5	14	0.60	2.83	0.44	0.86	49.20	6.3	3633	837	1.70	0.22
30	3	4	14	0.70	2.02	0.38	0.60	52.14	12	4259	2049	1.80	0.41
30	2	5	14	0.80	1.24	0.28	0.19	56.19	15	4845	2708	1.94	0.52

Table 4.2

Single Bead Movements. Equilibrium Data.

Length	#Runs	#Frames	Box	Dens	$\langle D^2 \rangle$	Std	D_t	$\langle L^2 \rangle$	Std	$\langle L^4 \rangle$	Std	α^2	Std
9	30	90	32	----	2.53	0.28	1.28	12.92	0.88	3.07	475	1.62	0.22
*10	10	500	8	0.02	----	----	0.93	14.81	0.21	297	8	1.654	0.02
*10	10	500	6	0.19	----	----	0.51	14.37	0.19	282	7	1.597	0.02
*10	10	400	6	0.42	----	----	0.26	13.89	0.28	267	9	1.543	0.03
10	8	20	10	0.50	3.47	0.40	0.32	13.74	0.38	264.60	14	1.53	0.40
10	6	20	10	0.50	----	----	0.24	13.55	0.32	257	20	1.50	0.06
*10	10	400	6	0.60	----	----	0.24	13.55	0.32	257	11	1.506	0.06
*20	10	200	8	0.04	----	----	1.45	35.75	0.90	1082	74	1.892	0.05
*20	5	100	8	0.20	----	----	0.92	34.07	1.45	1661	135	1.793	0.08
*20	5	100	7	0.41	----	----	0.50	31.63	1.53	1446	129	1.665	0.08
*20	5	100	7	0.52	----	----	0.32	30.69	1.65	1387	132	1.613	0.09
15	10	100	32	----	4.32	0.63	1.38	25.02	0.86	879	51	1.79	0.06
20	8	25	20	0.04	5.40	0.64	1.36	35.90	0.74	1829	68.5	1.89	0.04
20	2	4	20	0.20	3.49	2.82	0.59	35.44	4.90	1791	282	1.87	0.20
20	2	2	10	0.40	2.71	7.50	0.50	31.53	4.00	14.67	349	1.66	0.21
20	3	20	10	0.50	3.63	0.40	0.28	31.10	2.50	1438	213	1.64	0.13
20	3	6	10	0.60	4.32	0.70	0.16	31.54	1.73	14386	386	1.66	0.09
32	20	25	32	----	10.35	4.81	1.44	67.25	6.4	6467	1214	2.17	0.21
30	20	25	14	0.04	6.61	0.64	1.41	59.17	0.6	5056	109	2.04	0.02
30	4	4	14	0.20	4.27	2.00	0.76	53.95	7.0	4216	1080	1.86	0.24
30	2	2	14	0.40	4.33	1.70	0.33	49.87	9.9	3665	1482	1.72	0.34
30	2	2	14	0.50	4.81	2.12	0.18	50.96	12	4023	1840	1.76	0.41

Table 4.3

Concentration dependence of the end to end Length. Data from A. Bellemans and E. De Vos

N=10			N=20			N=30		
ρ	$\langle l^2 \rangle$	α^2	ρ	$\langle l^2 \rangle$	α^2	ρ	$\langle l^2 \rangle$	α^2
0.0000	14.7806	1.642	0.0000	36.32 ± .03	1.910	0.0000	60.40 ± .05	2.083
0.0930	14.587 ± .027	1.621	0.1006	35.37 .24	1.860	0.1111	57.40 .19	1.979
0.2015	14.312 .020	1.590	0.2138	34.22 .07	1.801	0.1901	55.94 .37	1.929
.2933	14.111 .023	1.568	0.2975	33.35 .11	1.755	0.3053	52.88 .29	1.823
0.3608	13.988 .018	1.544	0.3975	32.49 .24	1.710	0.3884	52.17 .19	1.799
0.4506	13.786 0.034	1.532	0.4975	31.73 .10	1.670			
0.4858	13.707 0.013	1.528	0.5975	30.71 .20	1.616	0.4999	48.812 ± .32	1.683
0.5729	13.519 0.018	1.502				0.5997	49.17 .14	1.696
0.5867	13.515 .034	1.502	0.5957	31.06 ± .32	1.635	0.6995	47.76 .26	1.647
0.7262	13.218 .033	1.476	0.6982	30.57 .32	1.609			
0.7438	13.199 .037	1.467	0.6982	30.36 .30	1.598	1.0000	44.70 ± .97	1.541
0.8695	13.018 .040	1.446	0.7910	29.50 .45	1.553			
			0.7910	29.13 .20	1.533			
0.7983	13.122 ± .090	1.458	0.8495	28.17 .29	1.483			
0.8423	13.084 0.53	1.545	0.8496	30.79 .27	1.621			
0.8986	12.896 .057	1.433	0.8984	27.04 .37	1.530			
0.9497	12.924 .088	1.436						
			1.0000	28.34 ± .19	1.492			
1.0000	12.79 ± .02	1.421						

--The first set of data at each chain length (up to the first blank line) was obtained using a computer model the same as in the present studies.
--The second set of data was obtained by use of a model in which empty lattice sites are moved instead of beads.
--The data given for polymer volume fraction of one was obtained by extrapolation of the computer results.

Table 4.4

a. The dependence of $\langle l^2 \rangle$ on chain length

density	a
0.04	1.18
0.20	1.16
0.40	1.11
0.50	1.12
0.60	1.06
0.70	1.12
0.80	1.17

$\langle l^2 \rangle \propto (N-1)^a$

b. The dependence of $\langle l^2 \rangle$ on density

l	γ	γ(Curro)	γ (Bishop)
5			$.06 \pm .01$
10	$.07 \pm .005$		$.13 \pm .02$
15		.076	
20	$.12 \pm .005$	.077	$.163 \pm .02$
30	$.13 \pm .04$		
32			$.20 \pm .03$
50			$.22 \pm .02$
70			$.23 \pm .02$

$\langle l^2 \rangle \propto \rho^{-\gamma}$

Table 4.5

Mixed bead movements. Dynamic Properties.

Length	Density	Fr.Len.	#Frames	#Chains	$\tau_{1/e}$	$\Delta\rho$	τ_s	$\ln \tau_s$	$\tau_s D_t/\langle l^2\rangle$
9	----	0.150	8000	1	0.14	0.035	0.15 ±.02	-1.88	0.04
10	0.04	0.120	5000	4	0.15	0.002	0.19	-1.66	0.04
10	0.20	0.180	1000	20	0.19	0.01	0.21	-1.56	0.04
10	0.40	0.310	800	40	0.30	0.02	0.36	-1.02	0.03
10	0.50	0.395	300	50	0.41	0.04	0.50 ±.08	-0.70	0.03
10	0.60	0.650	200	60	0.59	0.04	0.68 ±.31	-0.39	0.03
10	0.70	0.700	120	70	1.00	0.05	1.20 ±.21	0.25	0.03
10	0.80	2.100	128	80	2.13	0.04	3.23 ±.36	1.17	0.03
15	----	0.18	4000	1	0.16	0.05	0.23	-1.46	0.04
20	0.04	0.210	250	16	0.20	0.05	0.26 ±.01	-1.35	0.03
20	0.20	0.25	60	80	0.27	0.06	0.30 ±.05	-1.20	0.03
20	0.40	0.38	32	160	0.45	0.10	0.65 ±.06	-0.43	0.04
20	0.50	0.606	180	25	0.65	0.04	1.02 ±.18	0.02	0.05
20	0.60	0.696	120	30	1.04	0.05	1.43 ±.15	0.36	0.06
20	0.70	0.623	120	35	1.90	0.07	3.09 ±.32	1.13	0.04
20	0.80	0.720	240	40	4.61	0.07	7.46 ±3	2.01	0.01
32	----	0.300	500	1	0.26	0.06	0.32	-1.13	0.03
30	0.06	0.36	80	36	0.29	0.06	0.41 ±.01	-0.89	0.04
30	0.20	0.450	120	17	0.37	0.04	0.45	-0.80	0.03
30	0.40	0.420	36	18	0.565	0.07	0.83	-0.19	0.01
30	0.50	0.487	32	46	0.734	0.07	1.13	0.12	0.03
30	0.60	0.540	40	55	1.239	0.09	1.79 ±.38	0.58	0.03
30	0.70	0.676	24	64	2.86	0.1	4.30 ±1.4	1.469	0.40
30	0.80	1.10	20	73	4.91	0.1	11.1 ±4.9	2.411	0.04

Table 4.6

Single bead movements. Dynamic Properties

Length	Density	Fr.Len.	#Frames	#Chains	$\tau_{1/e}$	$\Delta\rho$	τ_s	$\ln\tau_s$	$\tau_s D_t / \langle l^2 \rangle$
9	----	0.33	2700	1	0.33	0.04	0.37	-0.99	0.04
*10	0.02	1.00	5000	1	0.33	----	0.44	-0.82	0.036
*10	0.19	1.00	5000	4	0.44	----	0.56	-0.58	0.036
*10	0.42	1.00	4000	9	0.73	----	0.96	-0.04	0.035
10	0.50	1.80	160	50	1.28	0.04	1.28	0.25	0.03
10	0.60	1.40	120	60	1.36	0.50	1.31	0.27	0.02
*10	0.60	1.25	400	13	1.36	----	1.76	0.57	0.034
*20	0.04	1.25	2000	1	0.71	----	1.13	0.12	0.046
*20	0.20	1.25	500	5	1.00	----	1.45	0.37	0.039
*20	0.41	1.25	500	7	1.57	----	2.06	0.72	0.032
*20	0.52	1.25	500	9	2.15		2.77	1.02	0.029
15	----	0.50	1000	1	0.50	0.04	0.65	-0.43	0.04
20	0.04	0.71	200	16	0.70	0.04	0.91±.17	-0.09	0.03
20	0.20	1.00	8	80	0.95	0.17	1.32	0.28	0.02
20	0.40	1.57	4	160	1.40	0.14	2.13	0.76	0.04
20	0.50	2.15	60	25	2.34	0.07	4.20±.9	1.44	0.04
20	0.60	4.80	18	30	3.43	0.07	7.58±.12	2.03	0.04
32	----	1.20	500	1	1.18	0.06	1.40±.12	0.40	0.03
30	0.04	0.788	300	4	1.08	0.02	1.55	0.44	0.04
30	0.20	1.30	24	18	1.38	0.10	2.63±1	0.97	0.04
30	0.40	2.80	8	37	2.93	4.57	4.17	1.43	0.03
30	0.50	4.60	8	46	4.69	7.69	7.14±3	1.97	0.03

Table 4.7

Densities at which chain entanglements begin
Mixed Moves

Chain Length	ρ^*
10	0.54
20	0.49
30	0.42

Ratio 1 : .91 : .86

Predicted Ratio 1 : .6 : .4

Table 4.8

The dependence of relaxation times on chain length

Density	Slope of Ln τ_s vs LN(N)		Slope of LN$\tau_{1/e}$ vs LN(N)	
	Mixed	Single	Mixed	Single
1 chain	0.58	1.18	0.50	1.09
0.04	0.68	1.17	0.58	1.13
0.2	0.67	1.40	0.60	1.39
0.4	0.76	1.18	0.57	1.31
0.5	0.77	2.04	0.54	1.58
0.6	0.90		0.70	2.09
0.7	1.13		0.95	
0.8	1.16		0.79	

Table 4.9

Relaxation of $\rho(L^2,L^2,T)$

Length	Density	$T_{1/e}$	$\Delta\rho$	T_s	ΔT_s
10	0.04	0.03	0.04	0.04	0.01
10	0.20	0.04	0.04	0.05	0.01
10	0.40	0.07	0.07	0.11	0.02
10	0.50	0.07	0.11	0.17	0.05
10	0.60	0.09	0.15	0.13	0.08
10	0.70	0.17	0.15	0.23	0.14
10	0.80	0.26	0.32	0.56	0.15
20	0.04	0.03	0.80	0.05	0.01
20	0.20	0.03	0.21	0.08	0.01
20	0.40	0.04	0.43	0.06	0.01
20	0.50	0.09	0.13	0.09	0.02
20	0.60	0.11	0.35	0.16	0.16
20	0.70	0.26	0.22	0.52	0.19
20	0.80	0.41	0.26	0.74	0.25
30	0.04	0.04	0.24	0.07	0.01
30	0.20	0.03	0.17	0.07	0.02
30	0.40	0.07	0.30	0.13	0.13
30	0.50	0.08	0.21	0.21	0.19
30	0.60	0.04	0.25	0.21	0.20
30	0.70	0.12	0.19	0.26	0.20
30	0.80	0.20	0.42	0.65	0.25

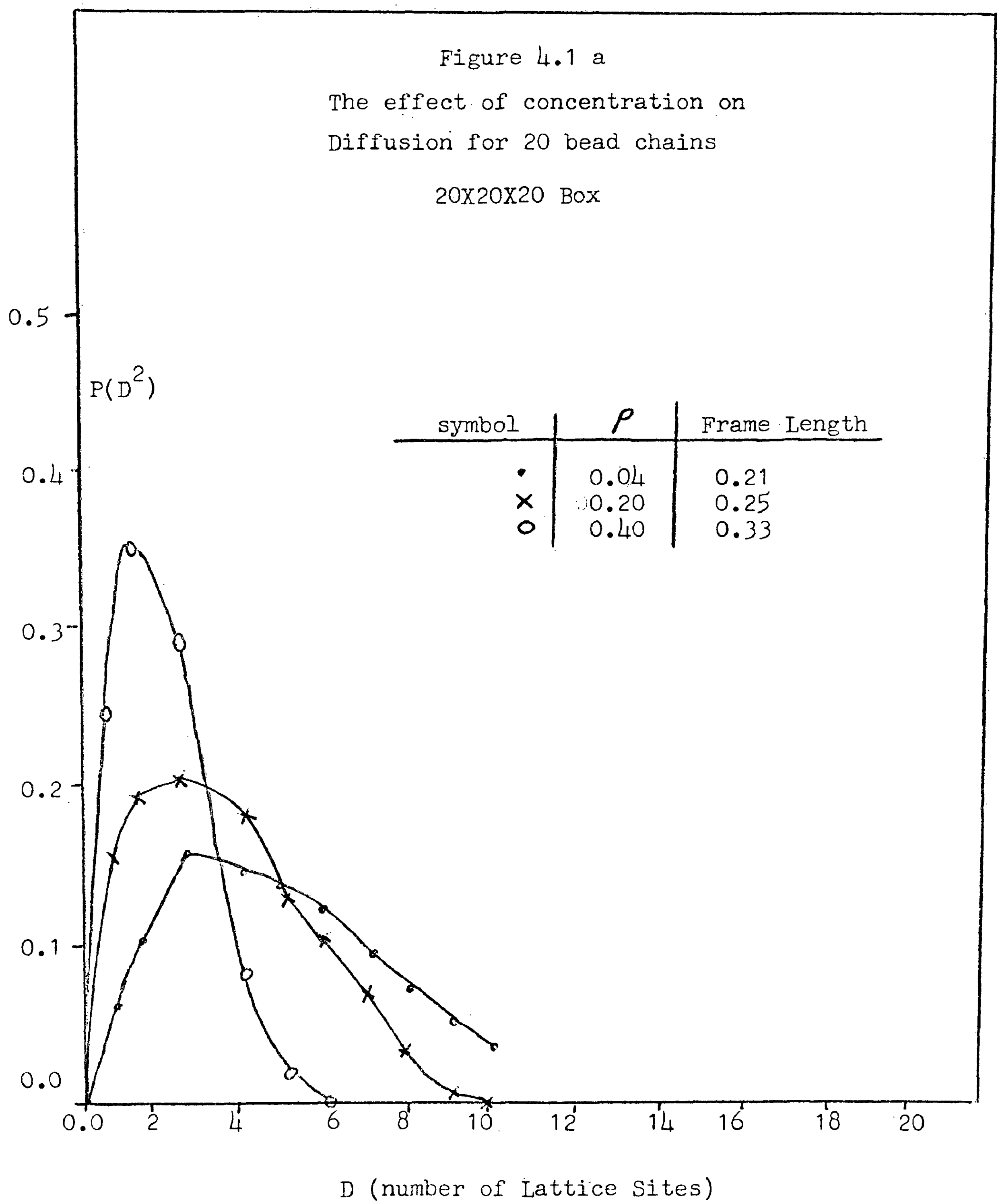


Figure 4.1 a
The effect of concentration on
Diffusion for 20 bead chains
20X20X20 Box
P(D^2)
symbol ρ Frame Length
• 0.04 0.21
× 0.20 0.25
o 0.40 0.33
0.5
0.4
0.3
0.2
0.1
0.0
0.0
2
4
6
8
10
12
14
16
18
20
D (number of Lattice Sites)

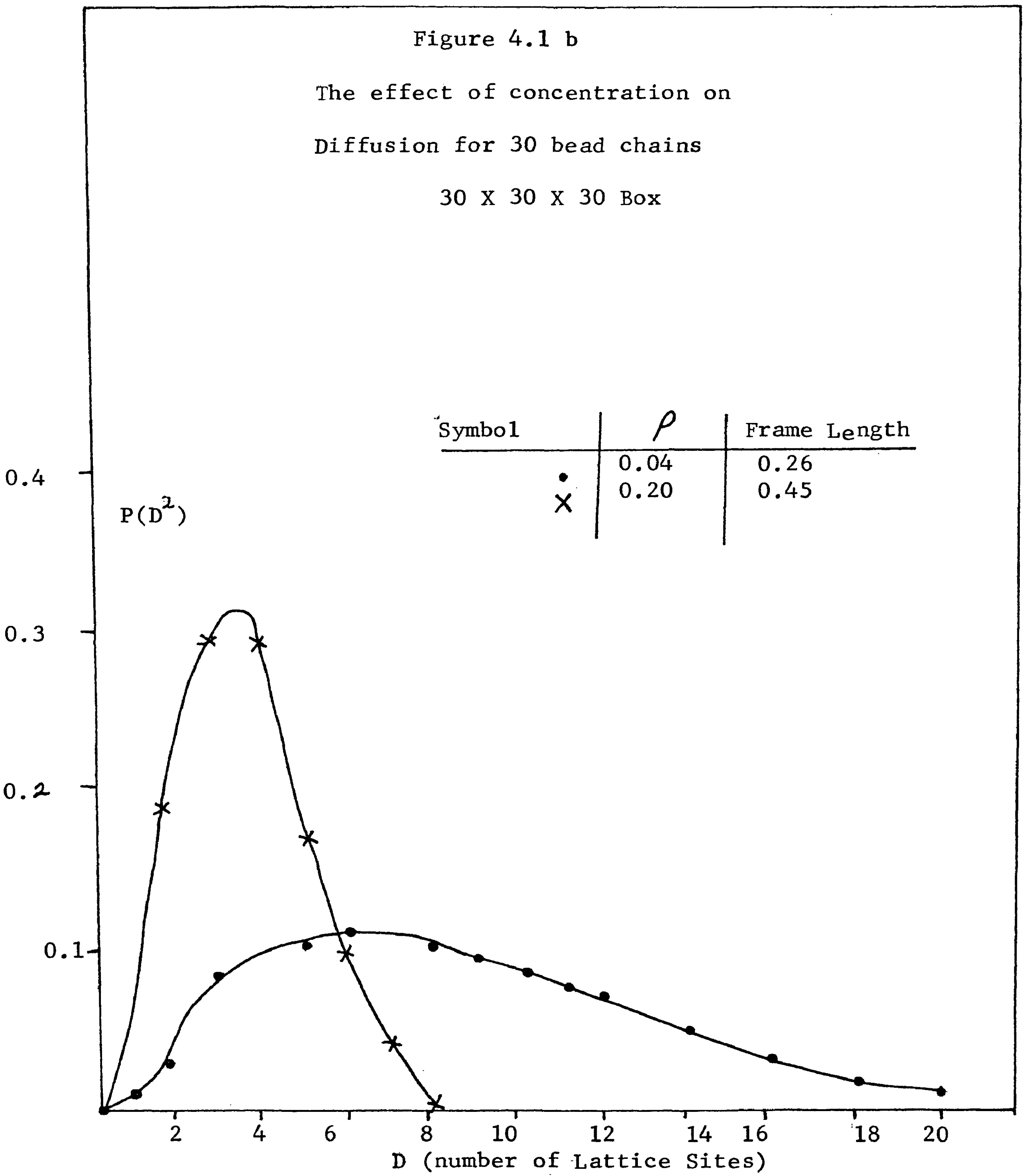


Figure 4.1 b
The effect of concentration on
Diffusion for 30 bead chains
30 X 30 X 30 Box
Symbol
ρ
Frame Length
•
0.04
0.26
X
0.20
0.45
P(D^2)
0.4
0.3
0.2
0.1
2
4
6
8
10
12
14
16
18
20
D (number of Lattice Sites)

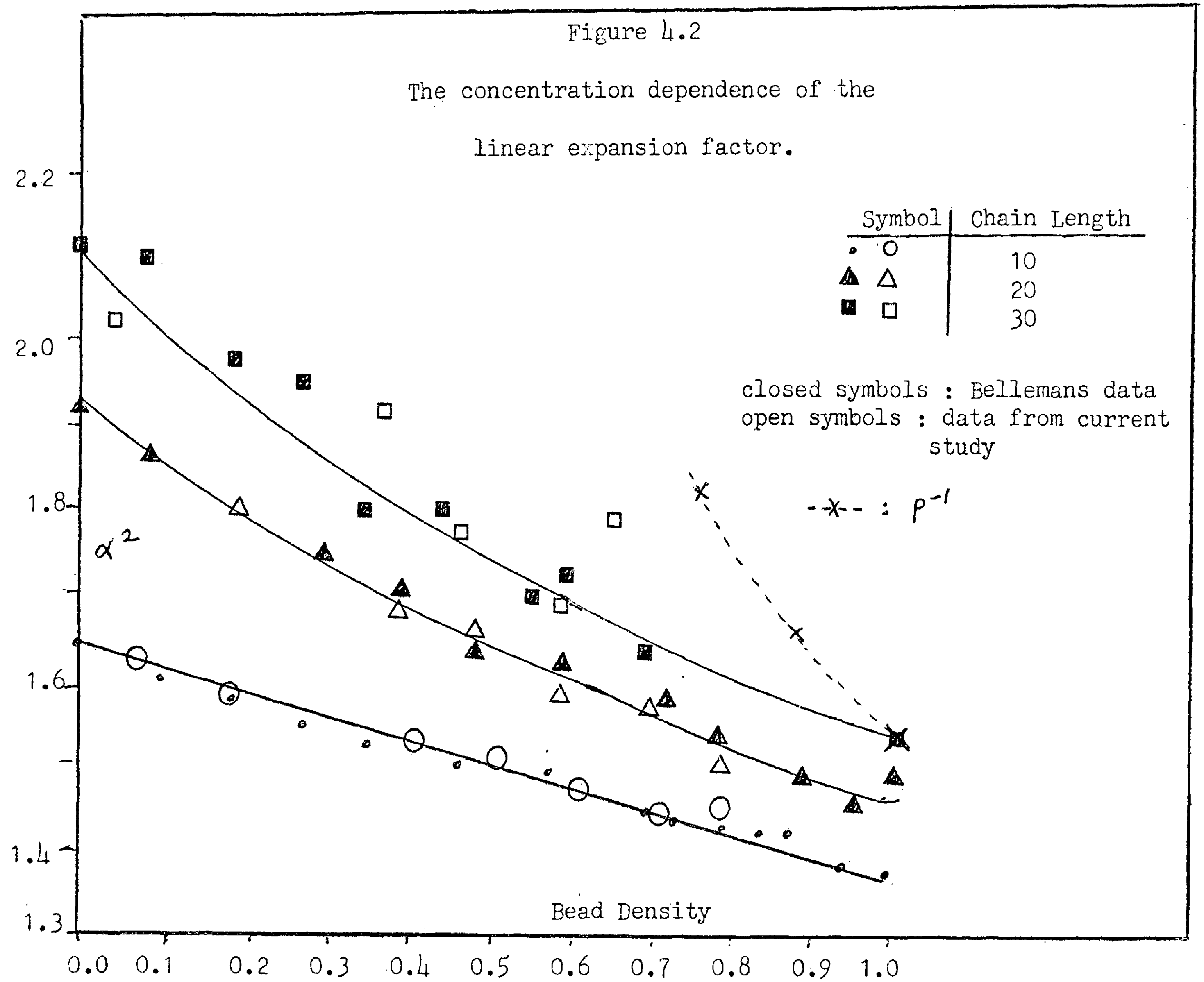


Figure 4.2

The concentration dependence of the linear expansion factor.

Figure 4.3

$\langle l^2 \rangle$ as a function of density.

Chain Length	x
10	12.87
20	27.87
30	42.87

Symbol	Chain Length
●	10
▲	20
■	30

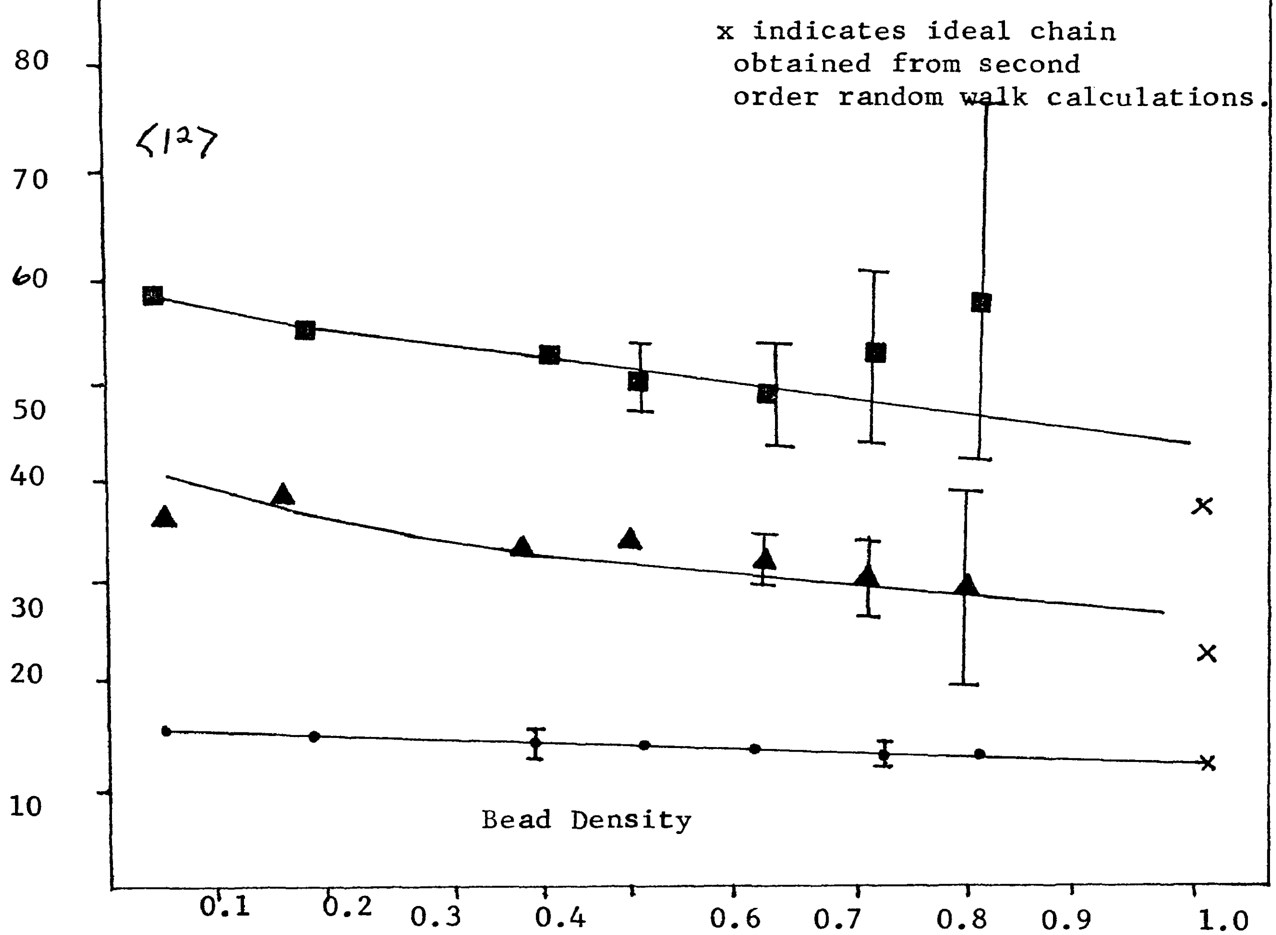
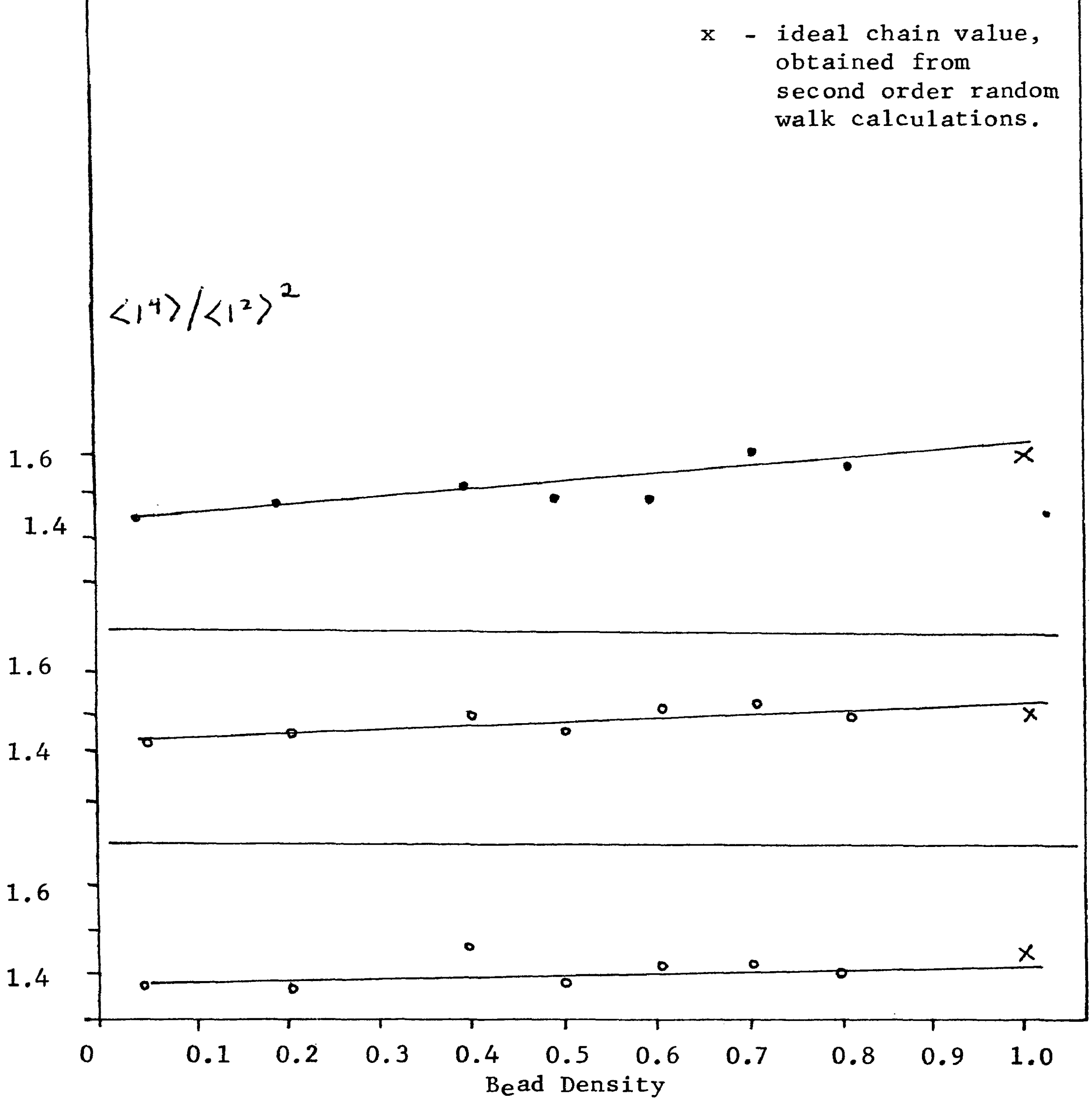


Figure 4.4

$\langle l^4 \rangle / \langle l^2 \rangle^2$ as a function of density.



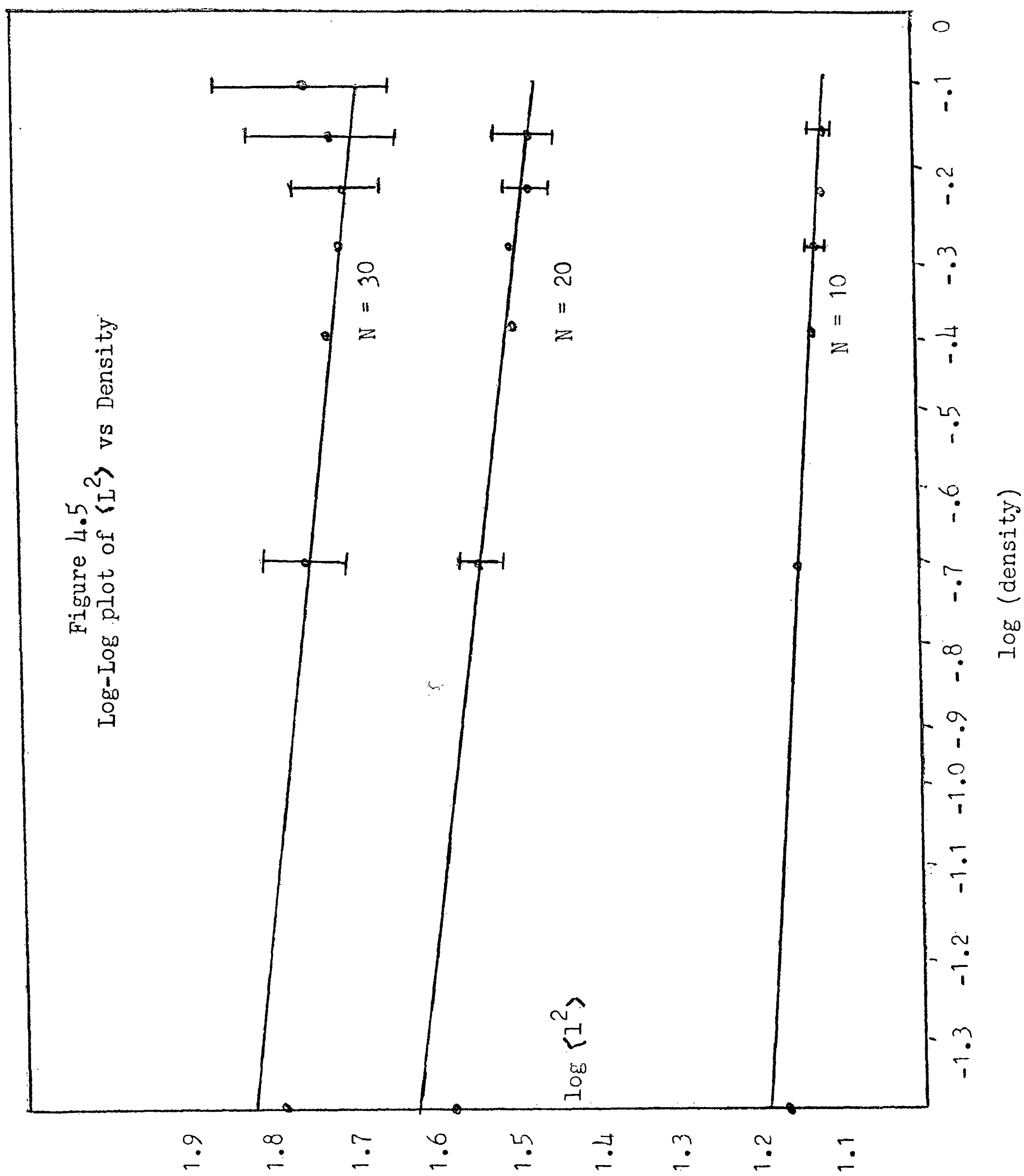


Figure 4.5
Log-Log plot of $\langle L^2 \rangle$ vs Density

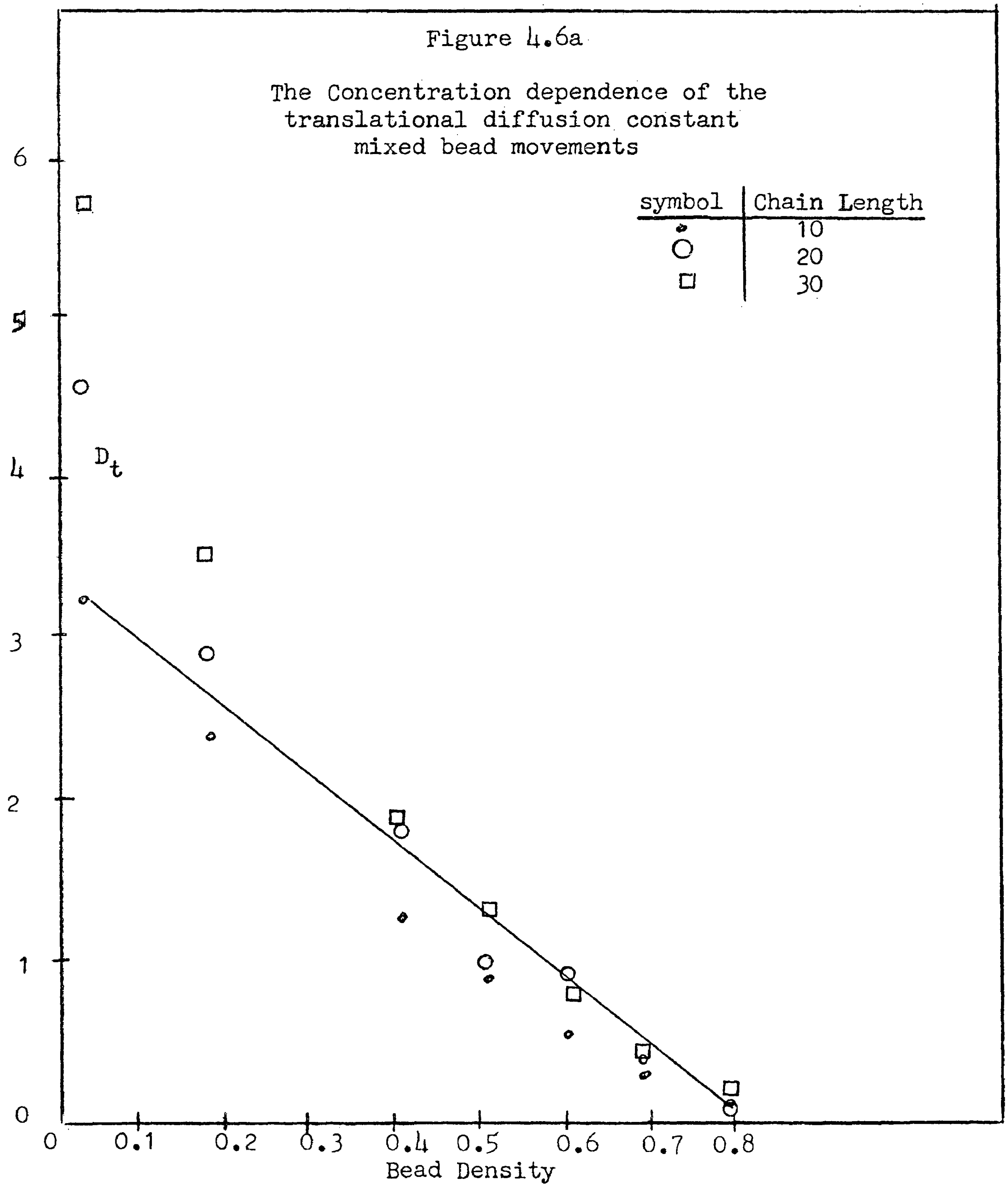
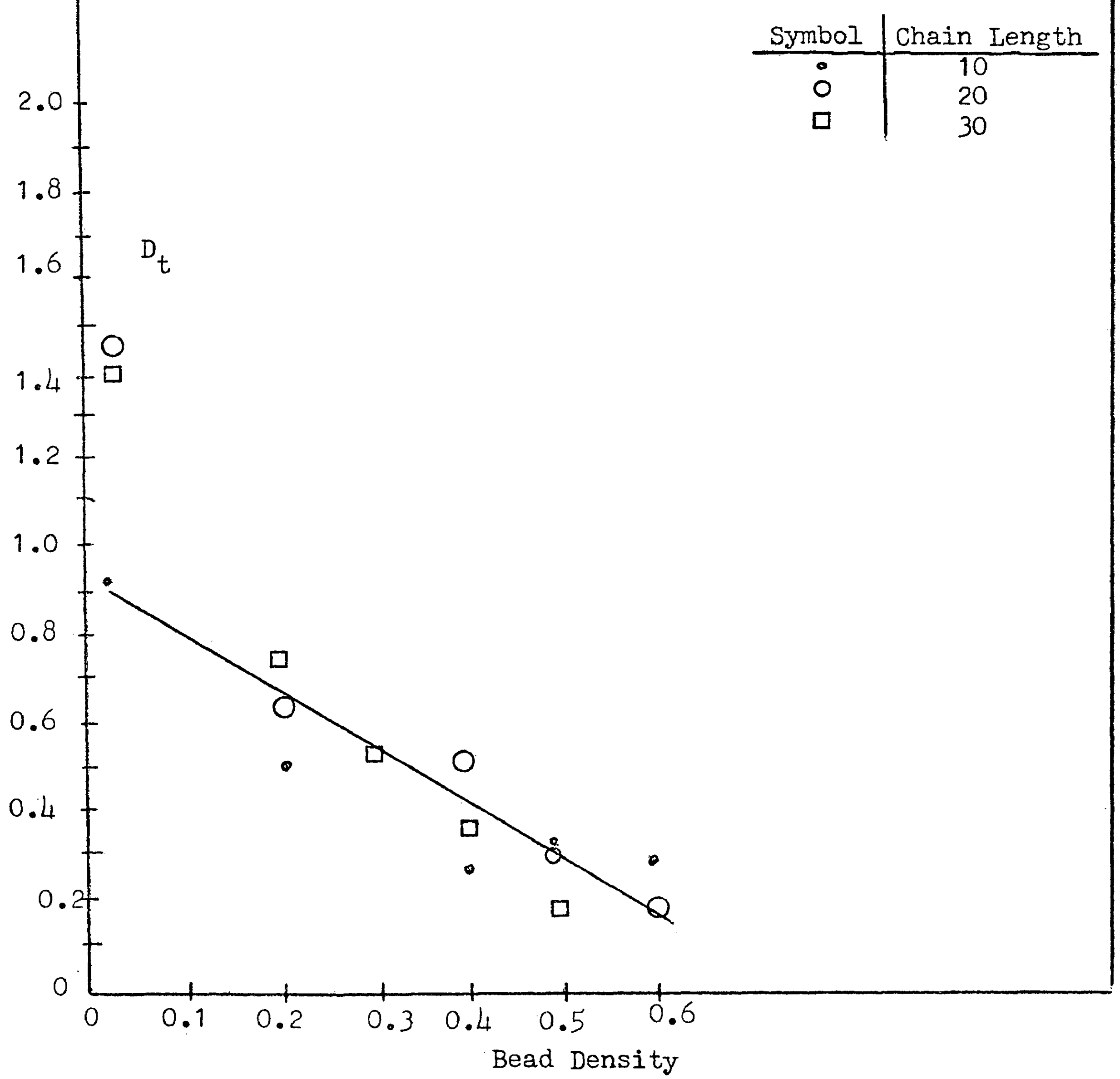


Figure 4.6a
The Concentration dependence of the translational diffusion constant mixed bead movements
symbol | Chain Length
10
20
30
D_t
0
1
2
3
4
5
6
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Bead Density

Figure 6.6 b

The concentration dependence of the translational diffusion constant single bead movements



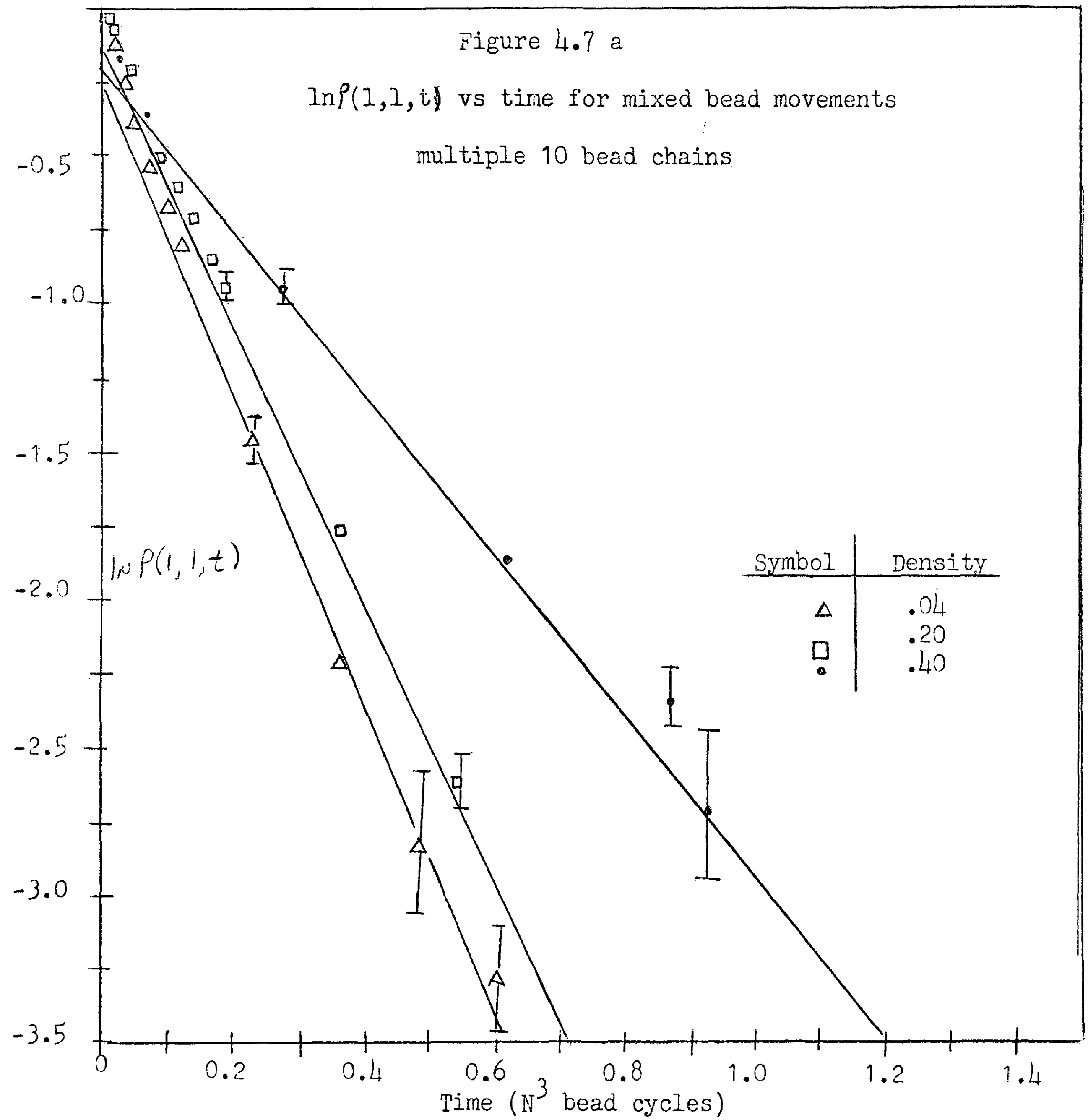


Figure 4.7 a

ln ρ(1,1,t) vs time for mixed bead movements

multiple 10 bead chains

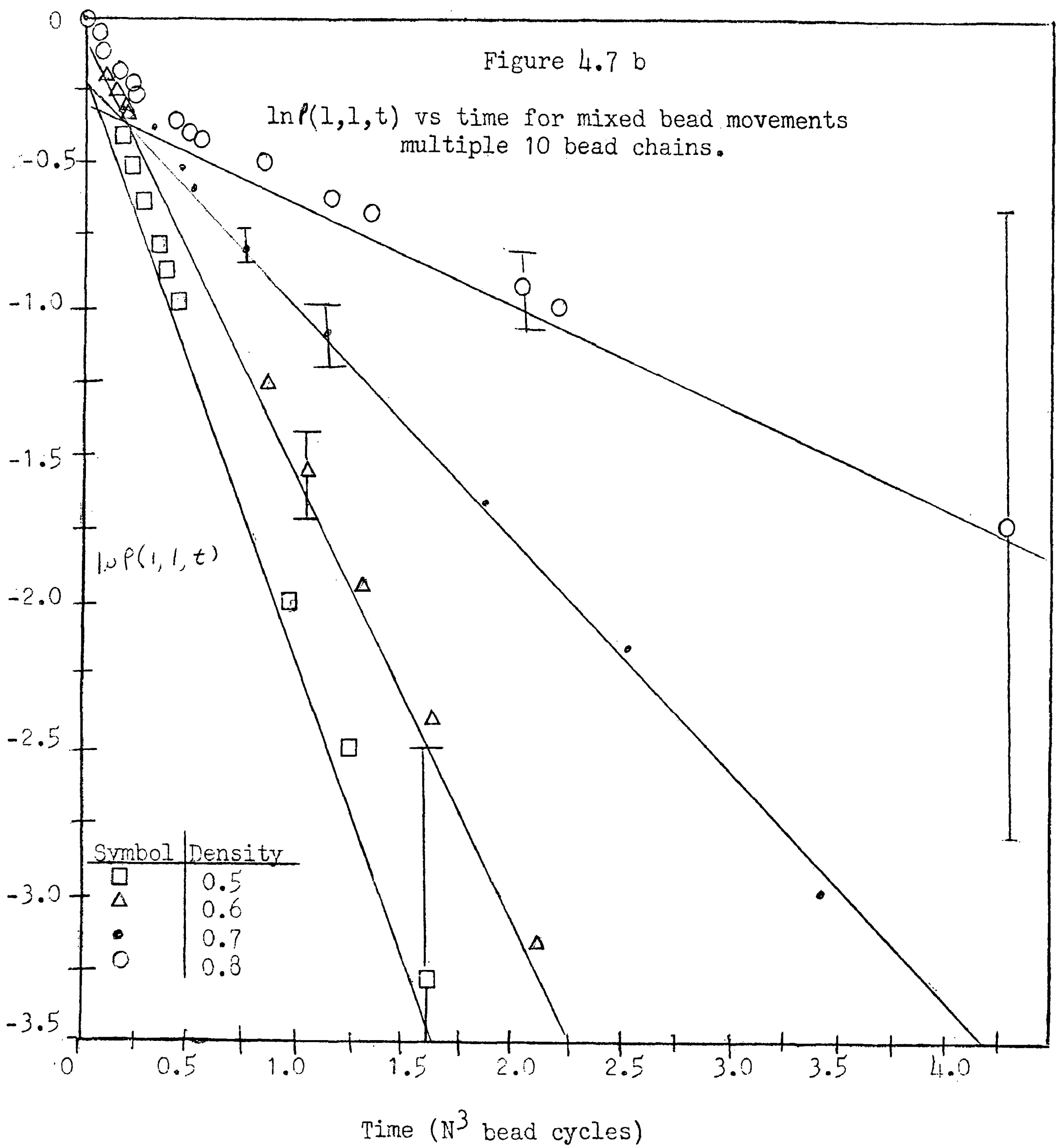


Figure 4.7 b

$\ln \rho(1,1,t)$ vs time for mixed bead movements multiple 10 bead chains.

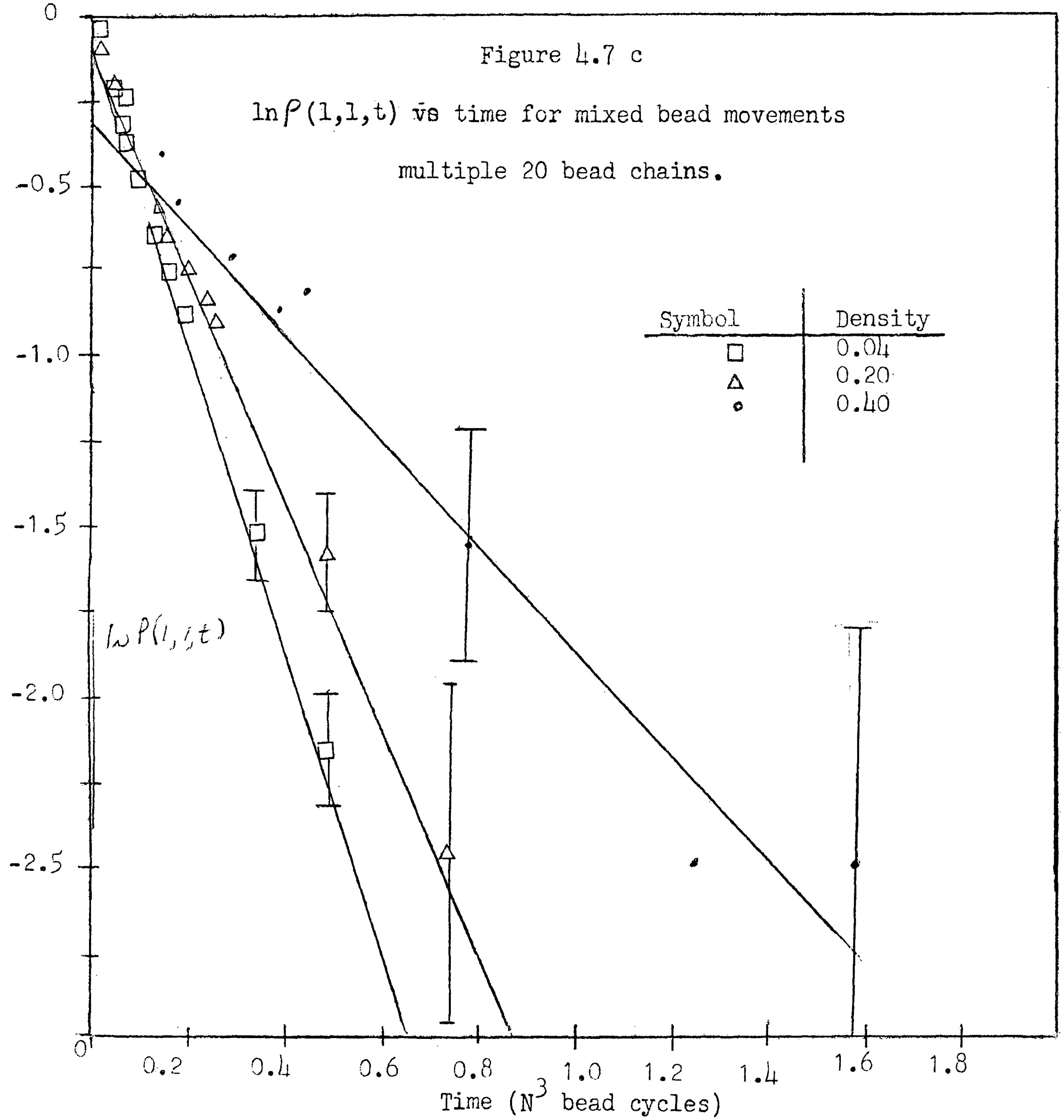


Figure 4.7 c

$\ln \rho(1,1,t)$ vs time for mixed bead movements multiple 20 bead chains.

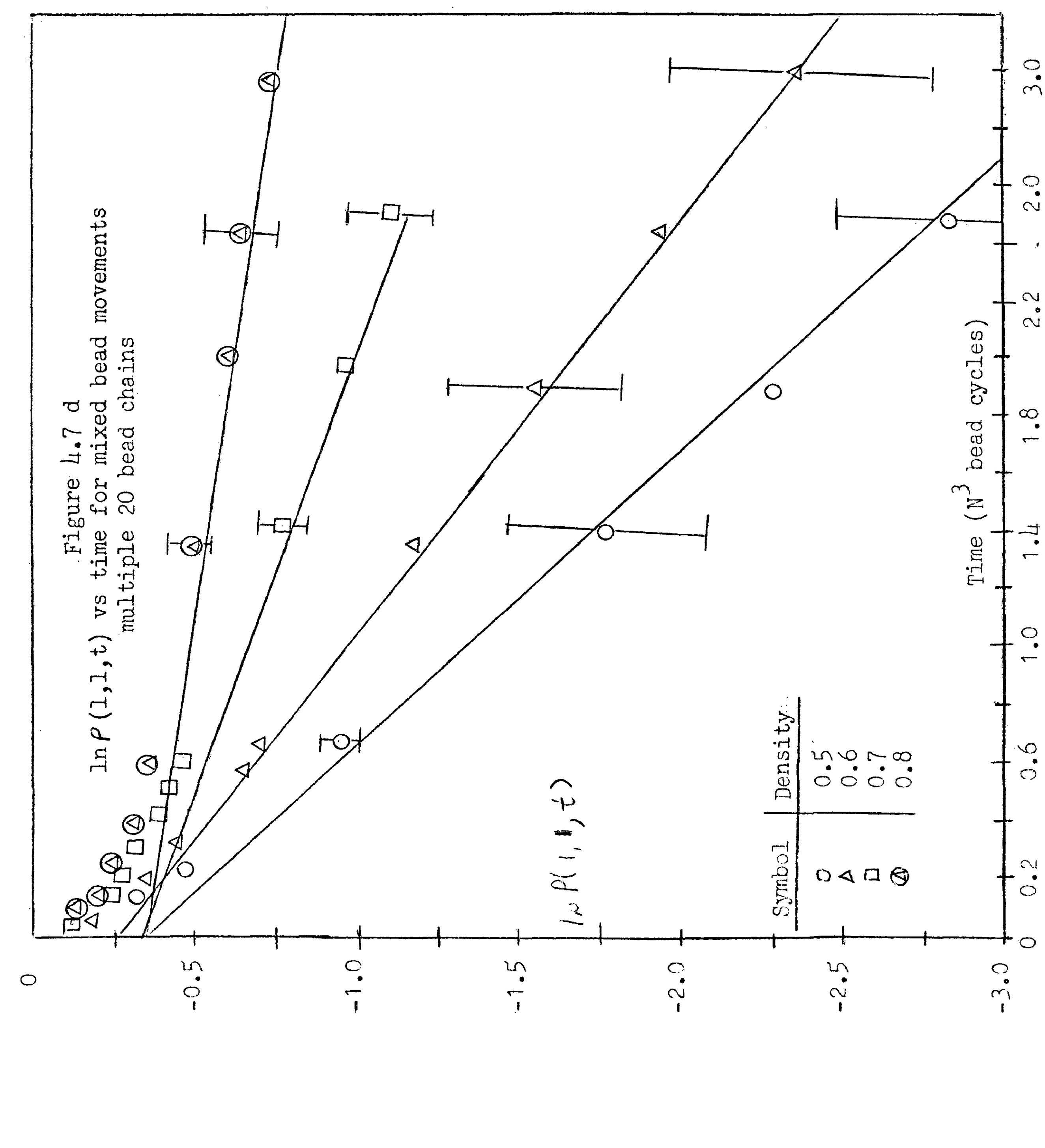


Figure 4.7 d
lnρ(1,1,t) vs time for mixed bead movements
multiple 20 bead chains

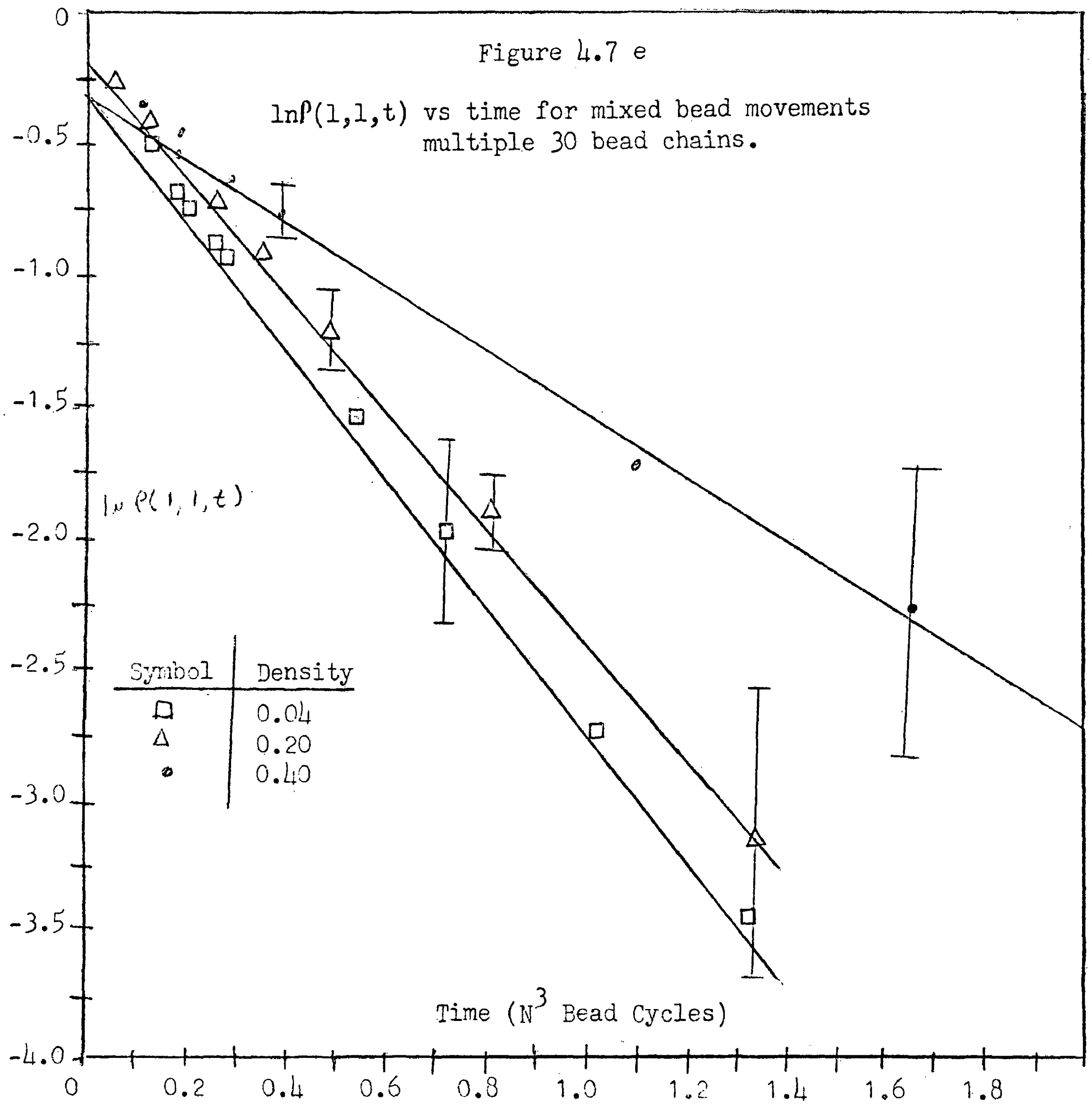


Figure 4.7 e

$\ln \rho(1,1,t)$ vs time for mixed bead movements multiple 30 bead chains.

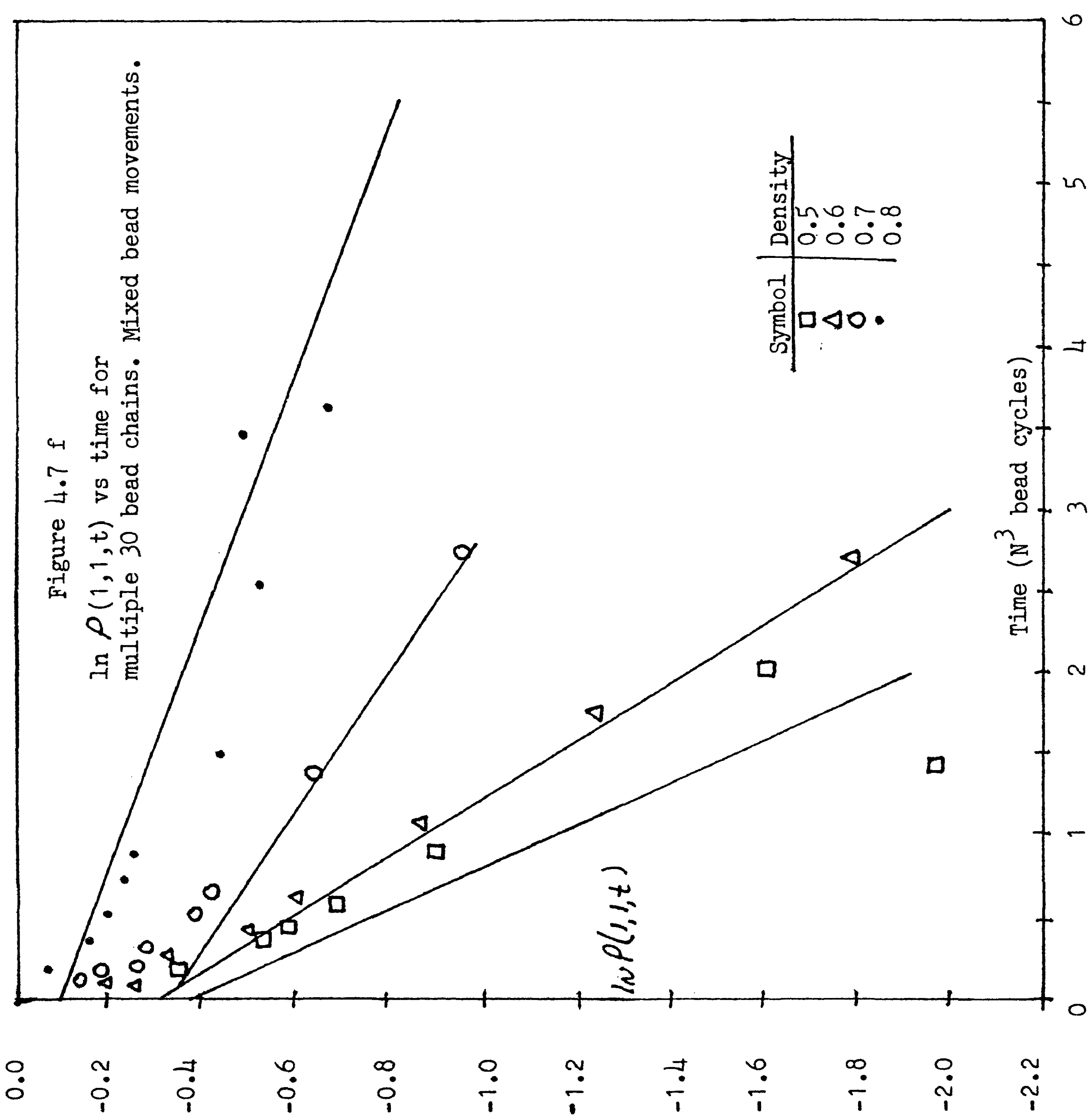


Figure 4.7 f
ln $\rho(1,1,t)$ vs time for multiple 30 bead chains. Mixed bead movements.

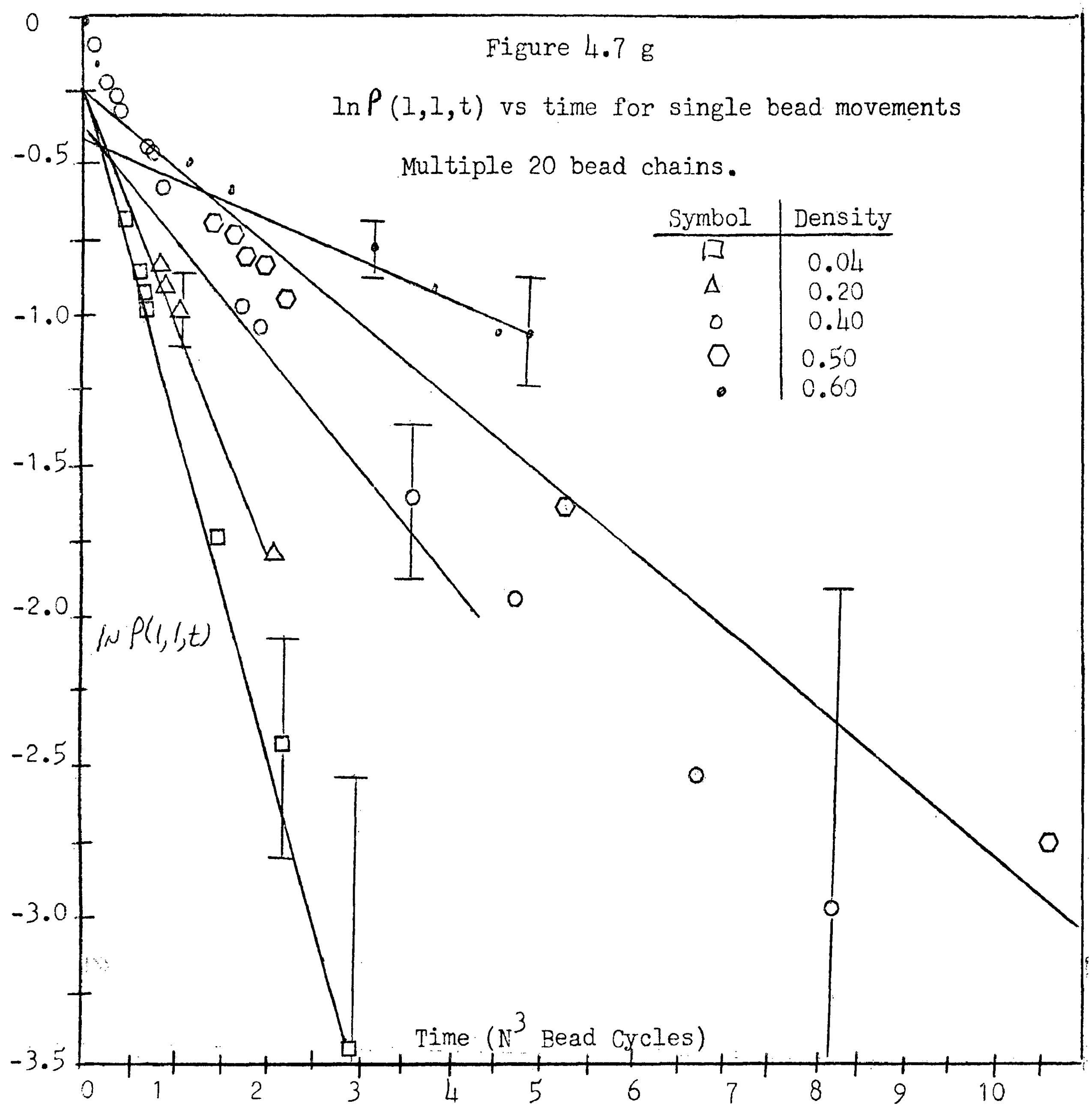


Figure 4.7 g

$\ln \rho(1,1,t)$ vs time for single bead movements

Multiple 20 bead chains.

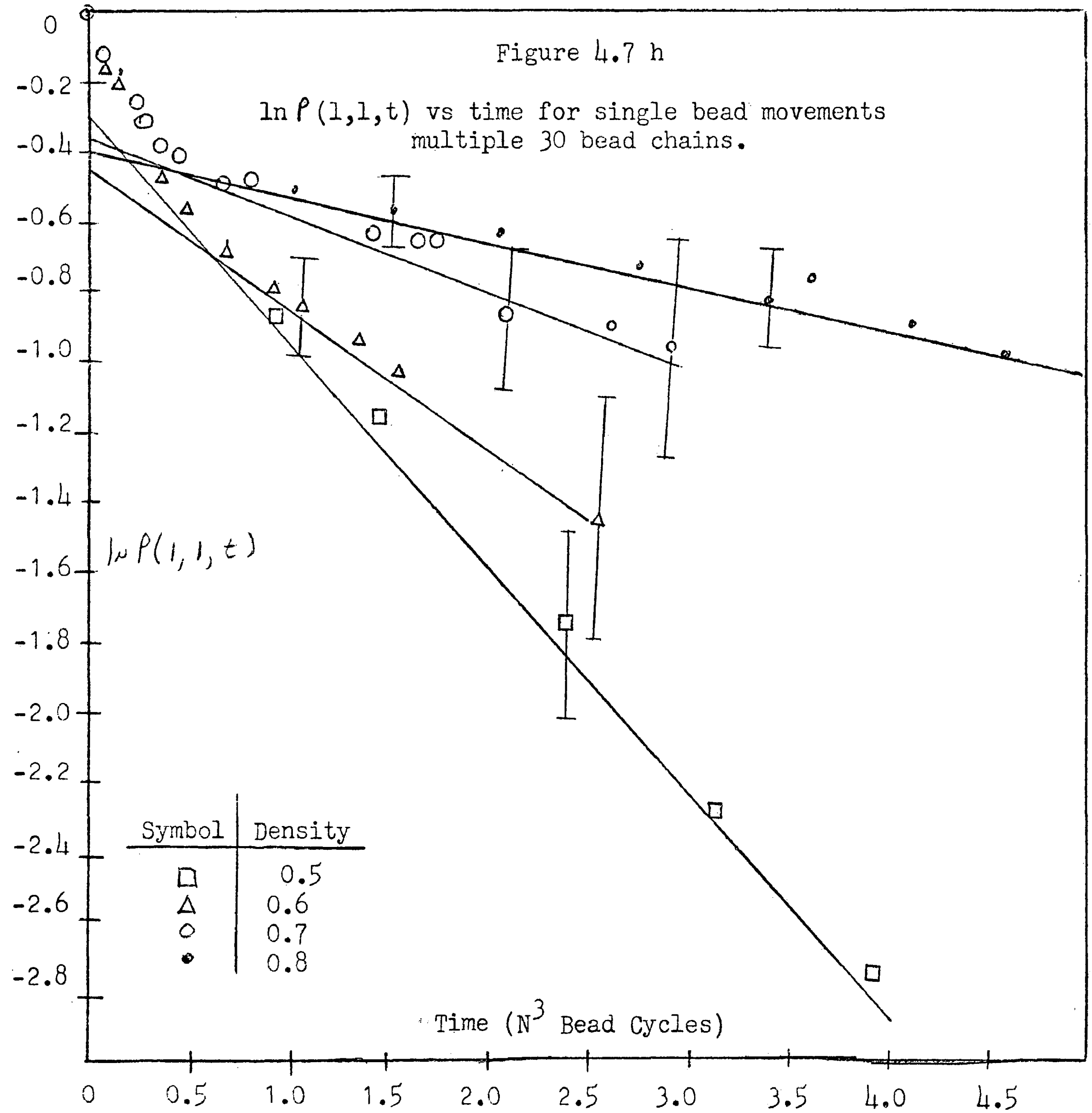


Figure 4.7 h

ln ρ(1,1,t) vs time for single bead movements multiple 30 bead chains.

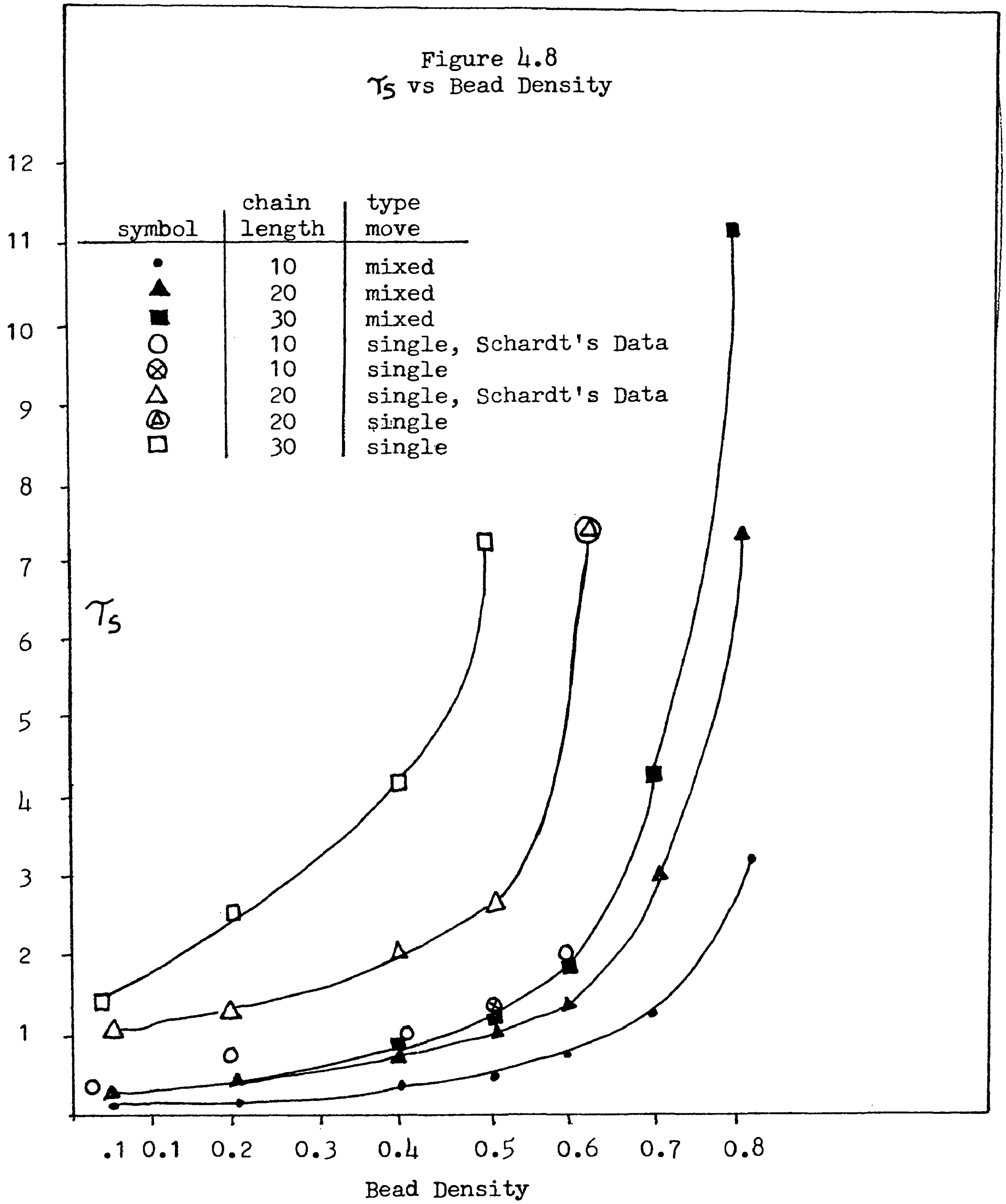


Figure 4.8
τ_5 vs Bead Density
symbol | chain length | type move
• 10 mixed
▲ 20 mixed
■ 30 mixed
○ 10 single, Schardt's Data
⊗ 10 single
△ 20 single, Schardt's Data
Ⓐ 20 single
□ 30 single
12
11
10
9
8
7
6
5
4
3
2
1
τ_5
.1 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
Bead Density

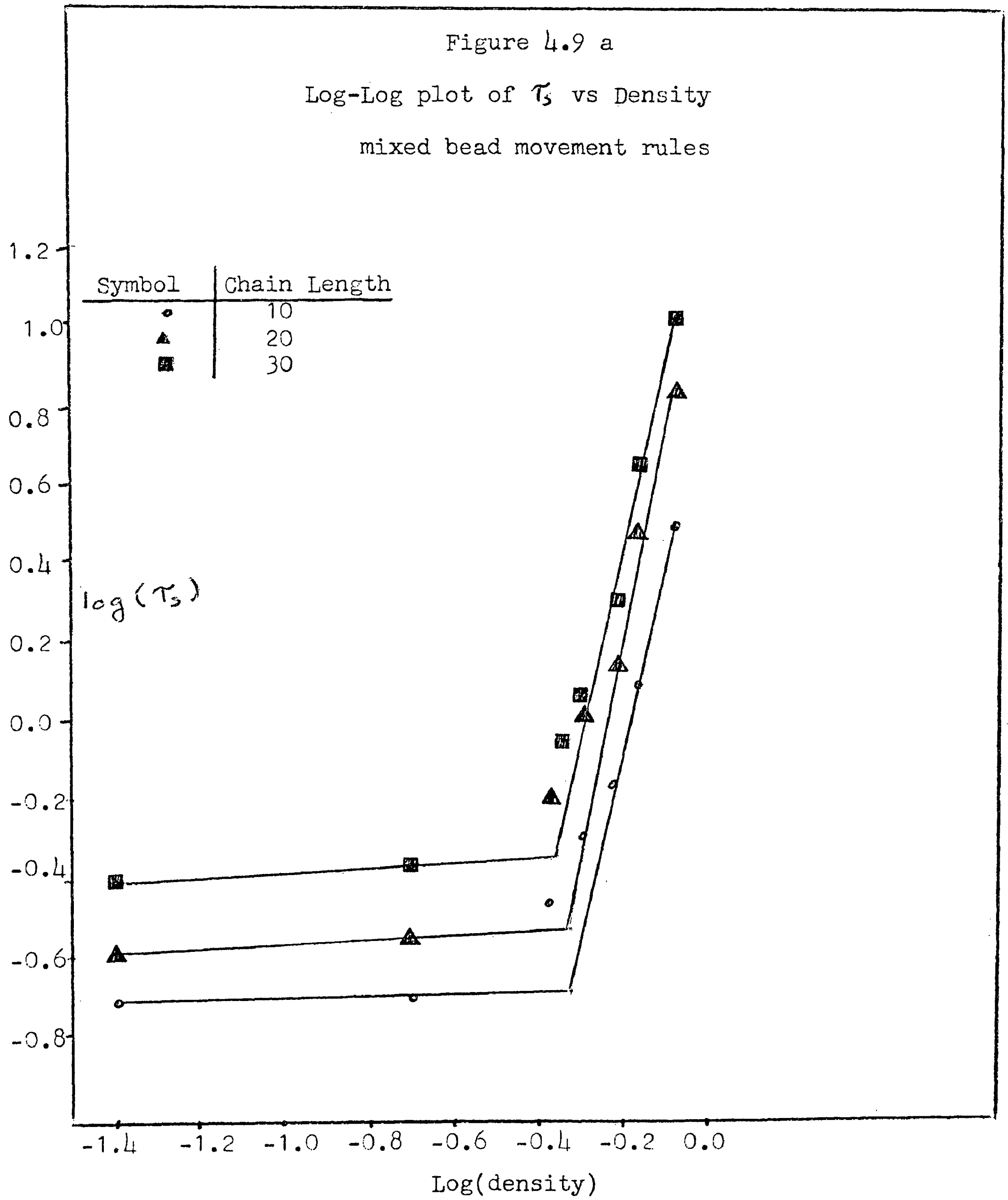


Figure 4.9 a
Log-Log plot of τ_s vs Density
mixed bead movement rules
Symbol | Chain Length
10
20
30
log(τ_s)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.4
-1.2
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
Log(density)

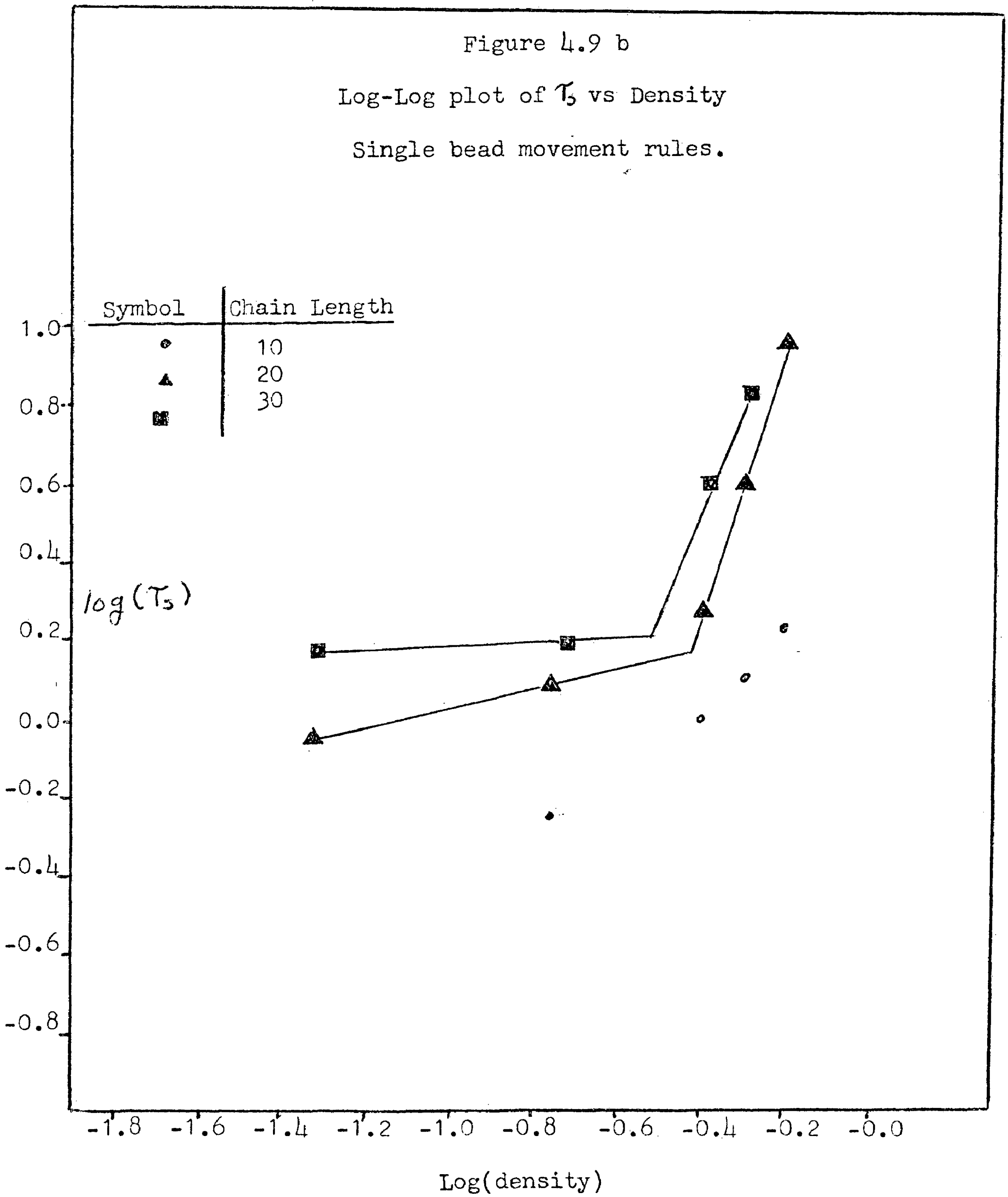


Figure 4.9 b
Log-Log plot of τ_5 vs Density
Single bead movement rules.
Symbol
Chain Length
10
20
30
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
log(τ_5)
-1.8
-1.6
-1.4
-1.2
-1.0
-0.8
-0.6
-0.4
-0.2
-0.0
Log(density)

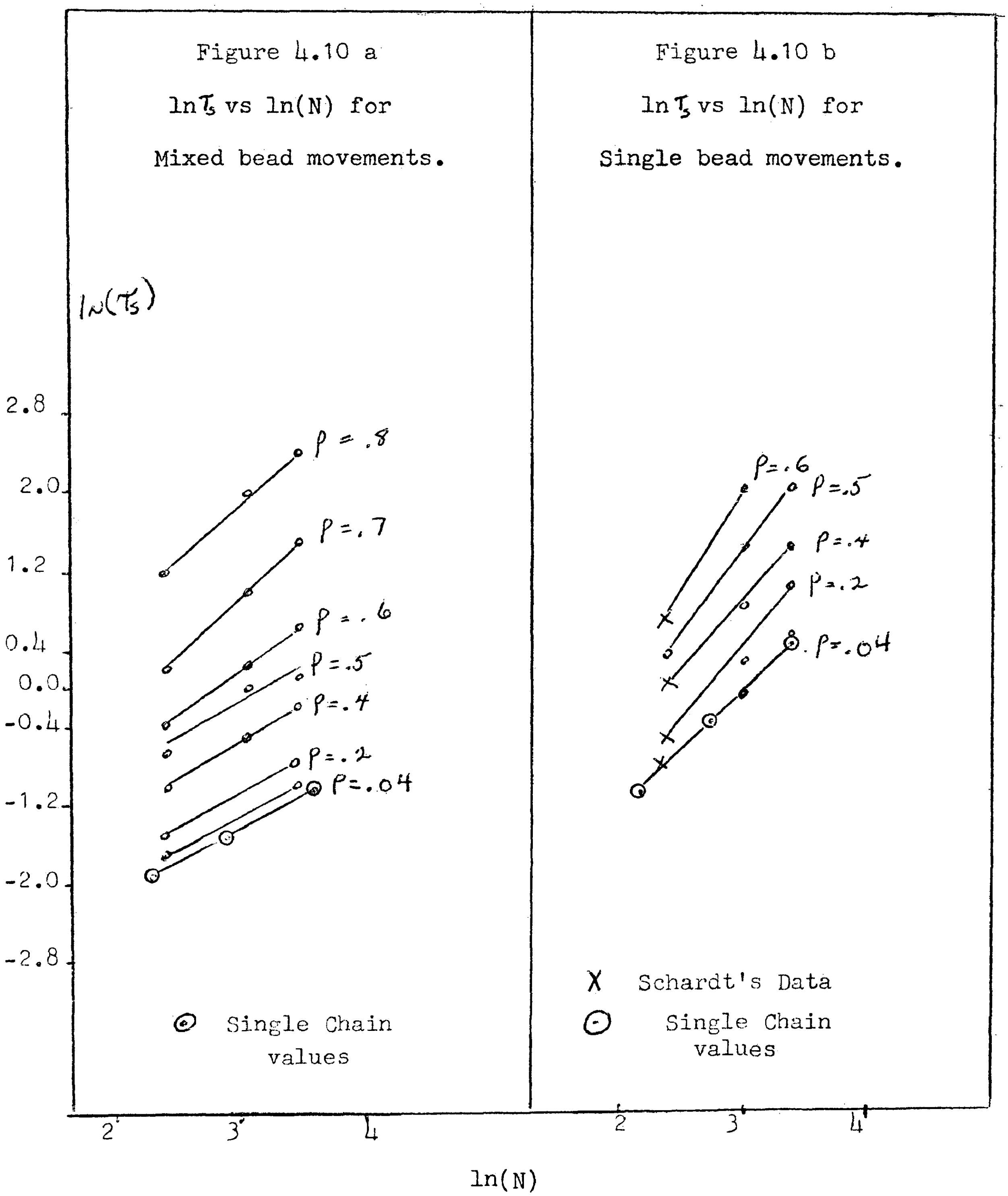


Figure 4.10 a
lnτ_s vs ln(N) for
Mixed bead movements.
Figure 4.10 b
lnτ_s vs ln(N) for
Single bead movements.
ln(τ_s)
2.8
2.0
1.2
0.4
0.0
-0.4
-1.2
-2.0
-2.8
ρ = .8
ρ = .7
ρ = .6
ρ = .5
ρ = .4
ρ = .2
ρ = .04
Single Chain values
X Schardt's Data
Single Chain values
2
3
4
ln(N)

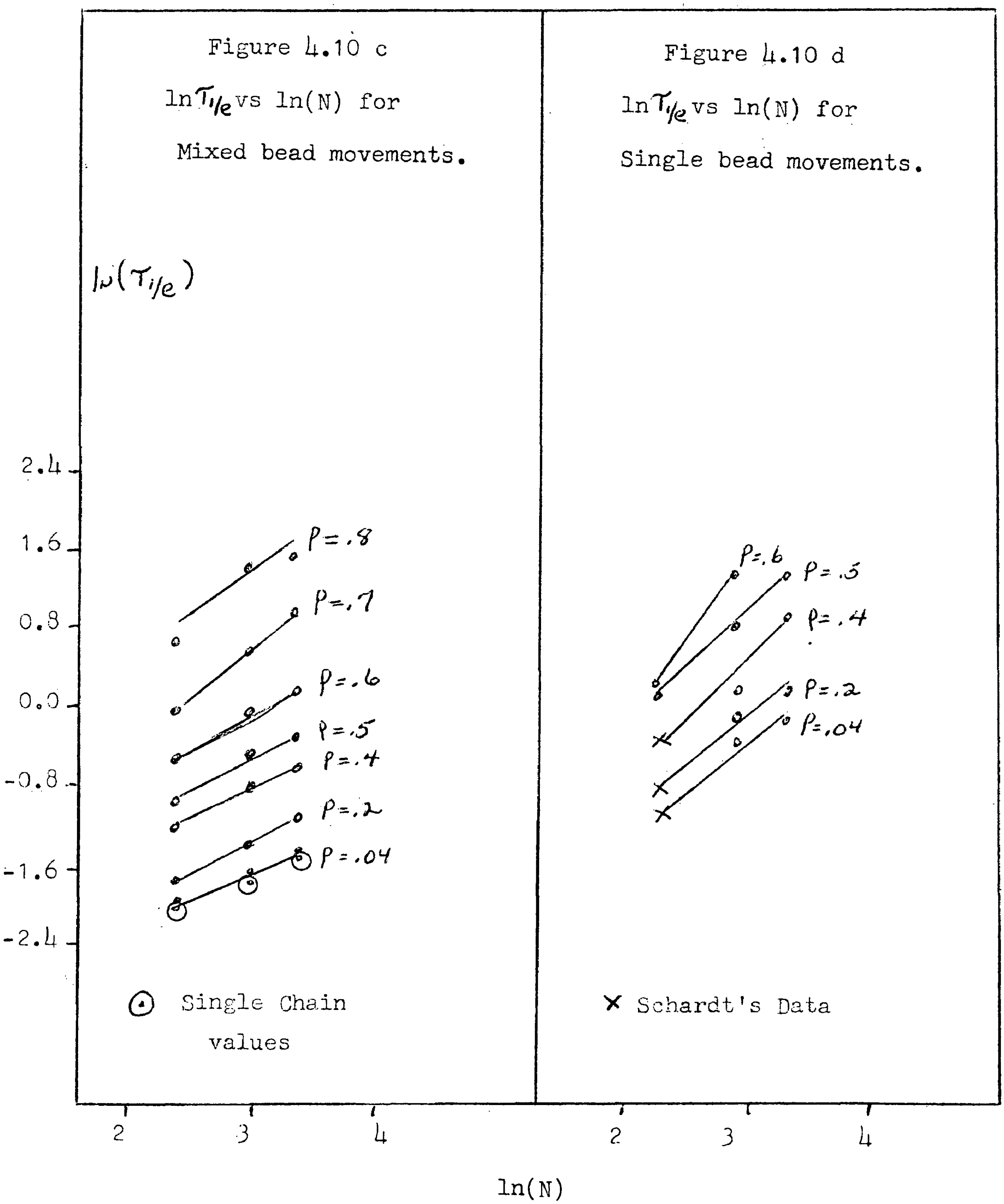
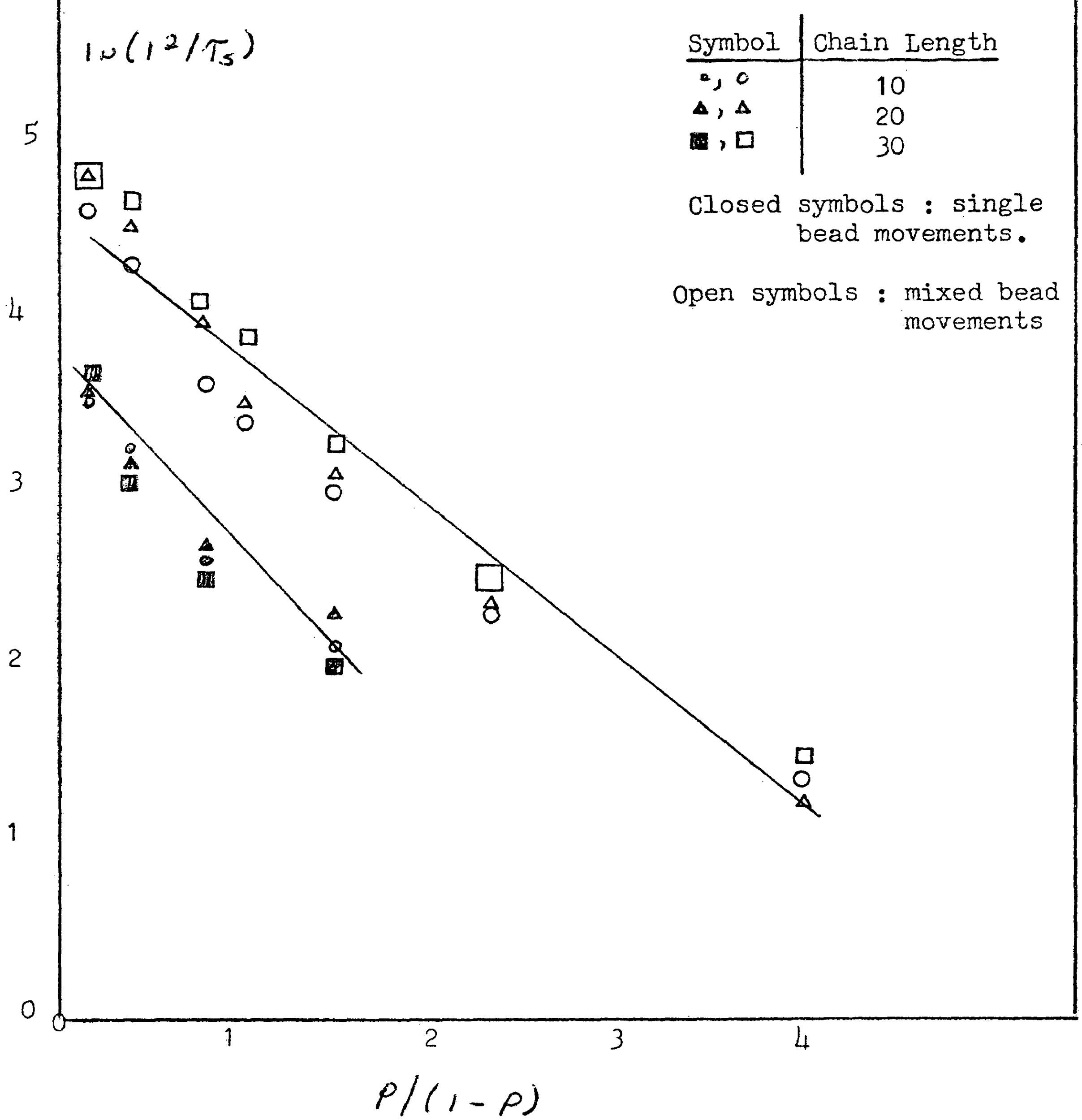


Figure 4.10 c
$\ln \tau_{1/e}$ vs $\ln(N)$ for
Mixed bead movements.
Figure 4.10 d
$\ln \tau_{1/e}$ vs $\ln(N)$ for
Single bead movements.
$\ln(\tau_{1/e})$
2.4
1.6
0.8
0.0
-0.8
-1.6
-2.4
P=.8
P=.7
P=.6
P=.5
P=.4
P=.2
P=.04
P=.6
P=.5
P=.4
P=.2
P=.04
Single Chain values
Schardt's Data
2
3
4
2
3
4
ln(N)

Figure 4.11

The dependence of $\ln(l^2/\tau_s)$ on Free Volume



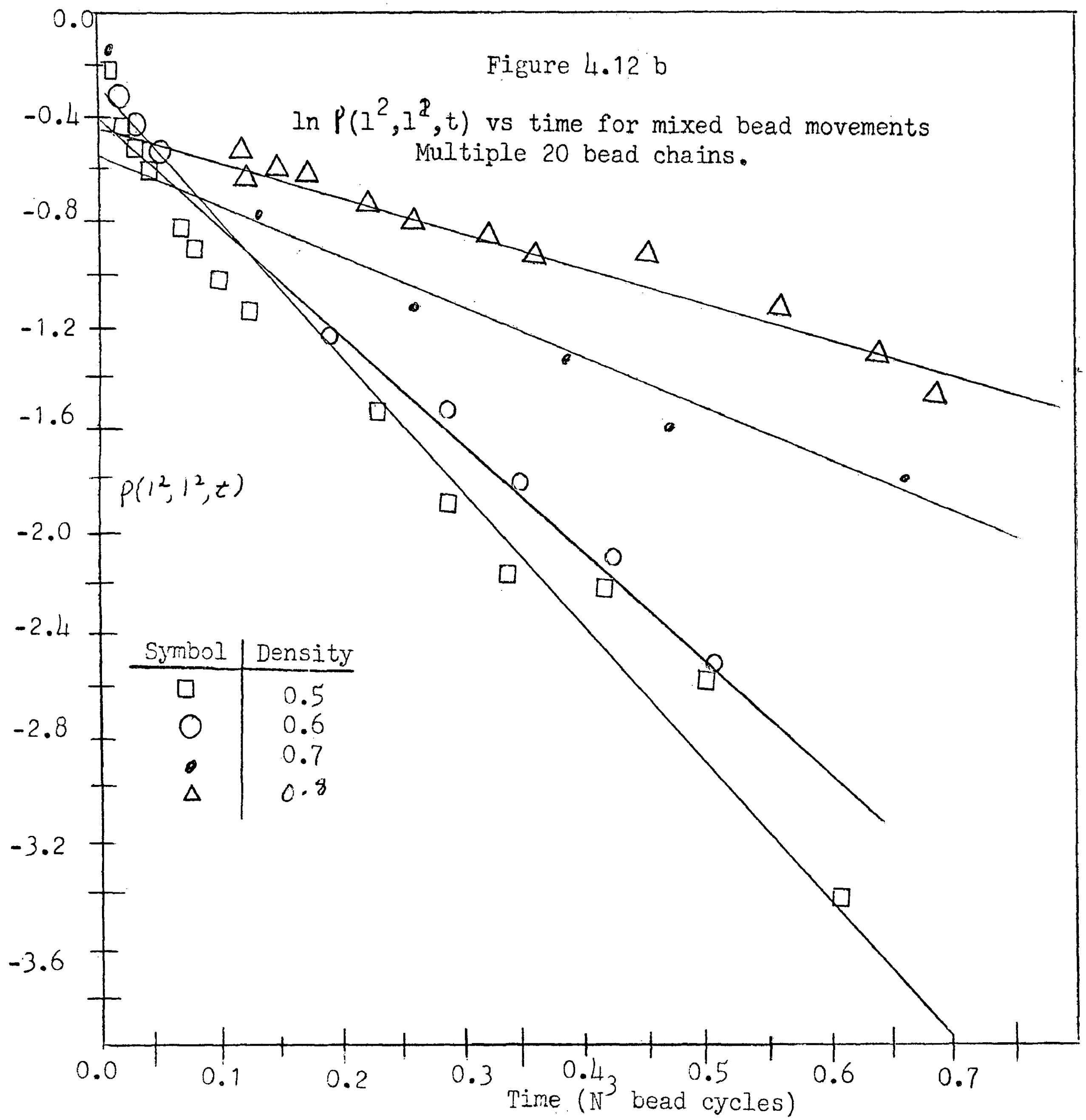


Figure 4.12 b

ln $\rho(1^2,1^2,t)$ vs time for mixed bead movements
Multiple 20 bead chains.

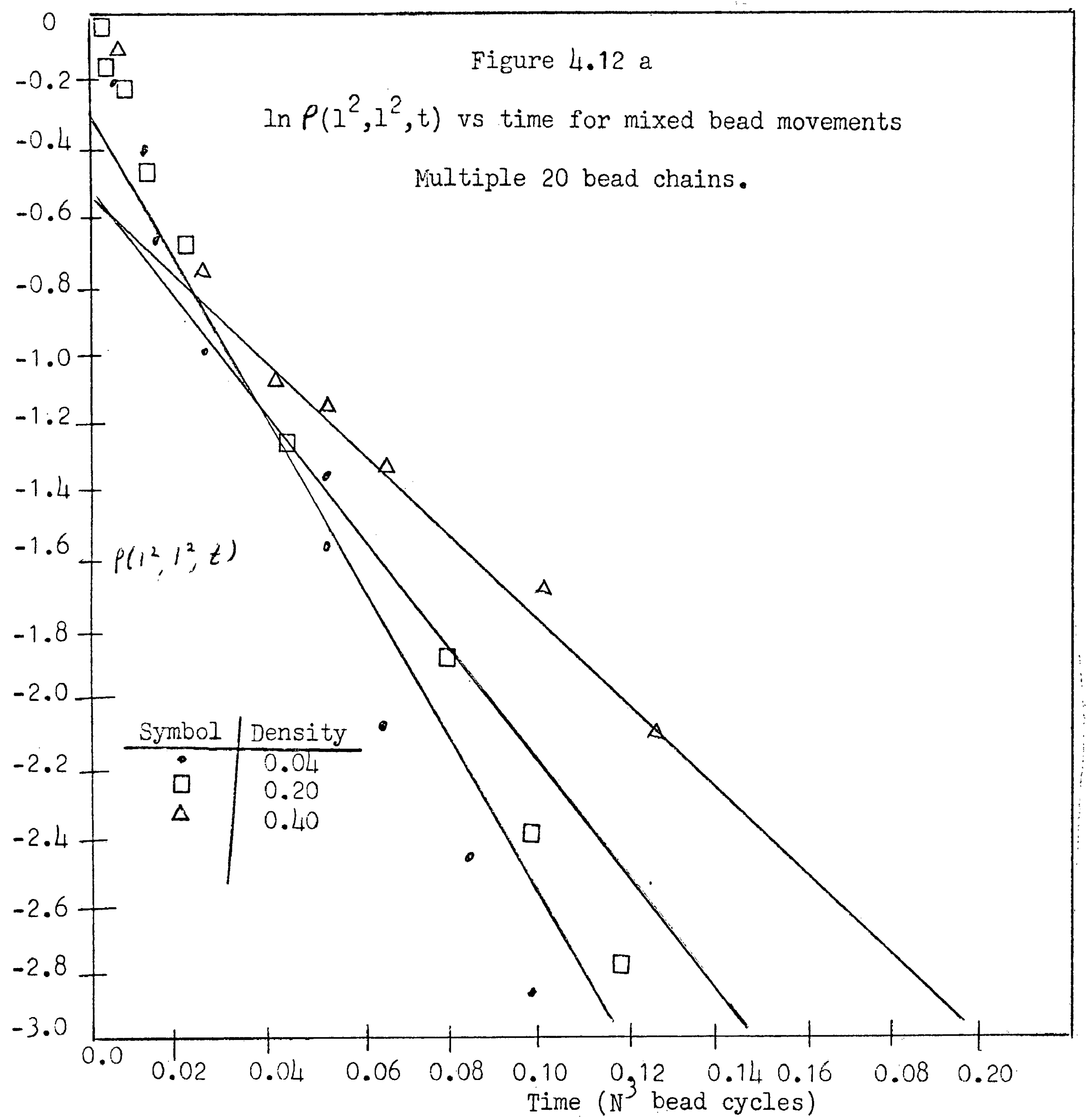


Figure 4.12 a

ln $\rho(1^2, 1^2, t)$ vs time for mixed bead movements

Multiple 20 bead chains.

Figure 4.13

τ_s (l^2) vs Density

for mixed bead movements.

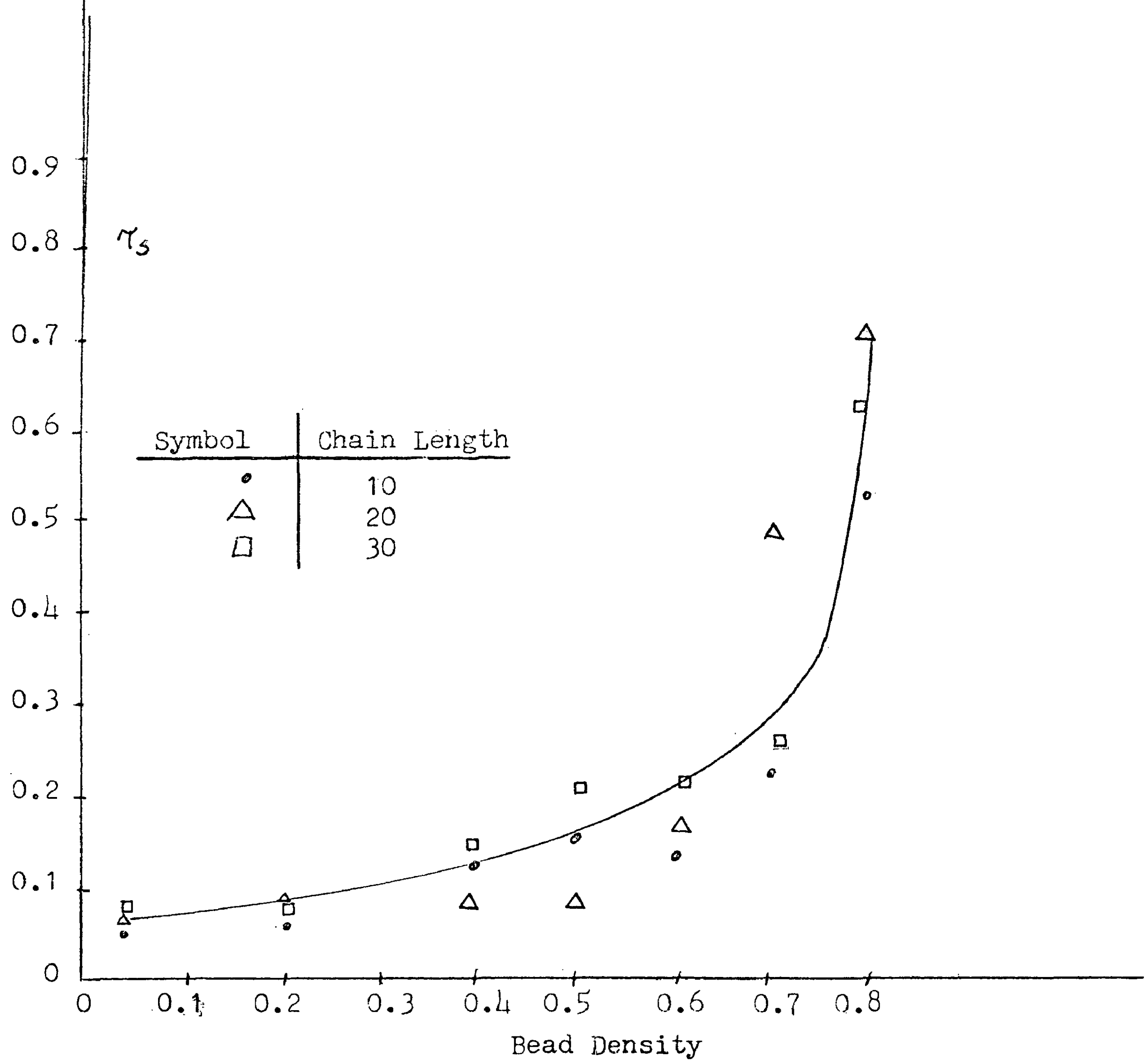
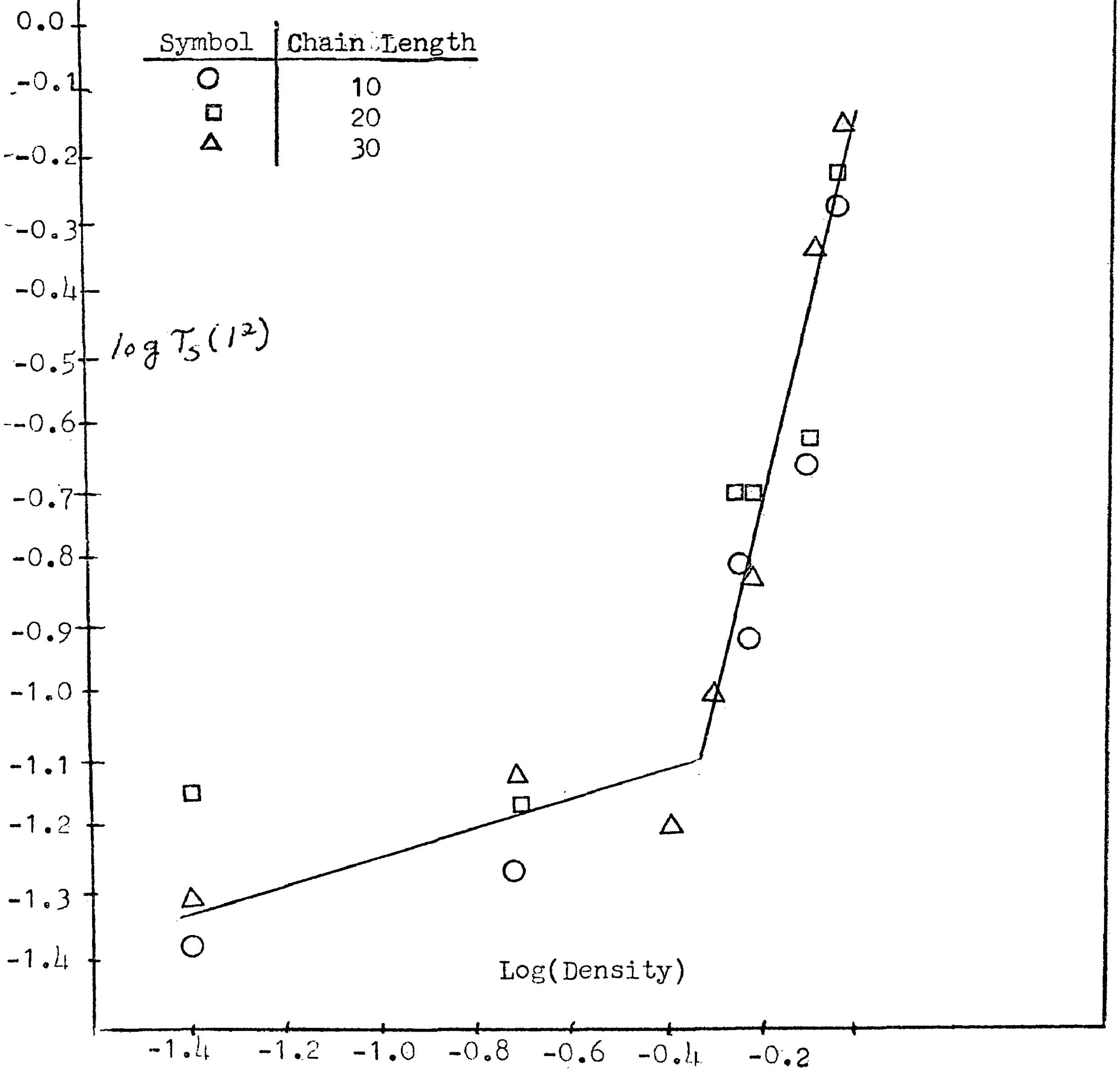


Figure 4.14

Log-Log plot of $\tau_s(1^2)$ vs Density fr

Mixed bead movements.



APPENDIX A

A Solution of the Eigenvalue Problem.

The matrix A derived in Chapter 1 may be written as

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ & & & & \cdots & & & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

by preforming a series of row and column additions on the original matrix. This matrix may be diagonalized by an orthogonal transformation,

$$R^{-1} AR = \Lambda = \lambda_i \delta_{ij}$$

where λ_p is the pth eigenvalue of A, and δ_{ij} is the Kronecker delta.

Writing A as the sum of two matrices, C and D gives

$$A = \begin{pmatrix} 0 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & \cdots & 0 & 0 & 0 \\ & & & & \cdots & & & \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \end{pmatrix} + \begin{pmatrix} 2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 2 & \cdots & 0 & 0 \\ & & & \cdots & & \\ 0 & 0 & 0 & \cdots & 2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 2 \end{pmatrix}$$

= C + D. The orthogonal marix, R, which diagonalizes C also diagonalizes D. The eigenvalues of A are related to the eigenvalues, c_i, of C by[37]

$$\lambda_i = 2 + c_i \quad . \tag{A1}$$

The eigenvalues of C are easily found by considering a more general matrix, π, which is of the save form as C,

$$\pi = \begin{pmatrix} 0 & p & 0 & 0 & \cdots & 0 & 0 & 0 \\ q & 0 & p & 0 & \cdots & 0 & 0 & 0 \\ 0 & q & 0 & p & \cdots & 0 & 0 & 0 \\ & & & & \cdots & & & \\ 0 & 0 & 0 & 0 & \cdots & q & 0 & p \\ 0 & 0 & 0 & 0 & \cdots & 0 & p & 0 \end{pmatrix}$$

Using the orthogonal transformation $B \pi B^{-1} = \lambda_i \delta_{ij}$, or $B \pi = \lambda B$ with $B = (\beta_1, \beta_2 \ldots \beta_{N-1})$, yields upon multiplication the following set of second order difference equations:

$$p \beta_{k-1} + q \beta_{k+1} = \lambda_k \beta_k \qquad (k = 0,1,2,\ldots N-1). \tag{A2}$$

The quantities β_0 and β_N are set to zero[18].

This set of equations may be solved by assuming a solution of the form[38]

$$\beta_k = \alpha r^k . \tag{A3}$$

Substituting A3 into A2 shows that this can be a solution only if r is a root of the quardratic

$$p - \lambda r + q r^2 = 0 . \tag{A4}$$

There will, in general, be two distinct, independent roots to this quardratic, r_1 and r_2. Therefore a solution to the difference equation (A2) will be in the form of a geometric progression

$$\beta_K = \alpha_1 r_1^K + \alpha_2 r_2^K . \qquad (A5)$$

Because $\beta_0 = \beta_N = 0$, it is easy to show that $\alpha_1 + \alpha_2 = 0$ and that $r_1^N = r_2^N$. Rewriting this last equation gives

$$r_1^N = r_2^N \exp(2\pi i j), \text{ or}$$

$$r_1 = r_2 \exp(2\pi i j/N) \text{ for } j = 0,1,2, .., N-1. \qquad (A6)$$

where[39] $i = \sqrt{-1}$ and $\exp(2\pi i j) = \cos 2j\pi + i \sin 2j\pi = 1 + i \cdot 0 = 1$.

By equation A6, r_1 and r_2 must be complex conjugates of each other. The absolute values of r_1 and r_2 may be found by using this fact as follows, $r^2 - \lambda/q + p/q = (r - r_1)(r - r_2) = 0$. Equating the coefficients of the two equations gives $\lambda = q(r_1 + r_2)$ and $r_1 r_2 = p/q$. The roots of A4 are

$$r = p/q \exp(\pi i j/N) \text{ and}$$

$$r_2 = \sqrt{p/q} \exp(-\pi i j/N) \quad (j = 0,1,2,..,N-1). \qquad (A7)$$

The eigenvalues of Π are $\lambda_i = q(r_1 + r_2) = -2\sqrt{pq} \cos(\pi i/N)$ $j = (1,2, .. ,N-1)$. Substituting $p = q = -1$ gives $c_i = 2\cos(\pi i/N)$ $(i = 1,.. N-1)$, and from equation A1

$$\lambda_i = 2 + 2\cos(\pi i/N) = 4\sin^2(\pi i/2N) \quad (i = 1, .. N-1) \qquad (A8)$$

The corresponding eigenvectors are given by the columns of the matrix R. The elements of R are given by Verdier[40] as R_{nj} = $\cos((n - 1/2\pi j/N)$.

APPENDIX B

Procedure used for the calculation of Diffusion Data

The exact number of lattice sites each chain moved during some time interval was calculated by the following method. This method is valid only if the chain length is less than or equal to the box size.

The subroutine performing the bead movements needed two arrays. These are IMOVE(I,J) and IWALL(I,J). The subscript I is 1,2, or 3 if the chain moved in the x,y, or z dimension, respectively. The J subscript is used to denote the chain of interest.

Each time an end bead moves out of the box a one is added to or subtracted from IMOVE(I,J). If the new coordinates of the bead in the I dimension is greater than the box size then a one is added. Otherwise a one is subtracted from IMOVE(I,J). The wall through which the the bead moved is also recorded. The walls are defined as:

Wall Hit	I	IWALL(I,J)
$x < 0$	1	1
$x > Box$	1	4
$y < 0$	2	2
$y > Box$	2	5
$z < 0$	3	3
$z > Box$	3	6

Initially, IMOVE(I,J) is set to zero if the chain is connected. If it is broken then IMOVE(I,J) is set to -1 if 0 < the center of the chain < Box Size/2. If Box Size/2 ≤ the Center of mass of the chain then IMOVE(I,J) is set to +1. If the chain is broken more than once then the diffusion of the chain is not calculated for that frame. This initialization procedure also occurs before each call to the subroutine which performs bead movements. The center of mass of each chain is calculated by summing over the x, y, or z coordinates and dividing by the number of beads in the chain. These sums always start with the first bead of the chain. If the difference in coordinates between two beads is greater than one then the chain is split and a box dimension is added or subtracted to the bead coordinate before the center of mass is calculated. This is to say that the center of mass is always calculated from a fully connected chain. If the center of mass lies outside the box then a box dimension is added to or subtracted from it to put it inside the box.

The diffusion is then calculated by the procedure outlined in Figure B1. The variables in this flowchart are defined as

CMO(I,J) = the position of the center of mass of the chain the last time its properties were sampled.

CM(I,J) = the current position of the center of mass of the chain

Box = the box dimension and

DCM = the distance the chain has moved in one of the three dimensions since the last time its properties were sampled.

The function INT(A) where A is a floating point number is defined as INT(A) = largest integer $\leq |A|$.

There is one exception to this procedure which had to be accounted for. This case is outlined in Figure B2. Diffusion is considered in the x dimension only. The bead represented by the open circle is bead number 1. The arrows mean that some number of bead movements have occured to cause the chain to reach the new configuration pictured. It was reached in such a way that the variales IMOVE(I,J) and IWALL(I,J) set according to the above rules are as shown.

To calculate DCM by the procedure of Figure B1 the following choices would be made:

(1) The chain is not connected

(2) A wall was hit

(3) The center of mass, CM(I) is nearest wall 1

(4) Wall 1 waw hit.

The formula used to calculate DCM is, then

DCM = CM(1) - CMO(1,1) + INT(IMOVE(1,1) + INT(IMOVE(1,1)/2) * BOX. Substituting for these variables from Fig B2 gives

$$DCM = 2.2 - 7.3 + 0 = -5.10.$$

This is clearly wrong. The chain did not move 5.10 lattice sites to the left but 10.2 - 7.3 = 2.9 lattice sites to the right.

The correction was made to the program by the addition of the subroutine CHAIN. This subroutine checks to see if an end bead moves through a wall, and then through the opposite wall. If this type of movement is detected the IWALL(I,J) is set to zero. In the above example, the choices made in B1 are

(1) The chain is not connected

(2) no wall was hit.

Therefore, DCM = CM(I) - CMO(1,1) = 2.2 - 7.3 = -5.10 and

(3) -5.10 < BOX/2, so

$$DCM = 8.0 - 5.10 = 2.90 ,$$

which is correct.

FIGURE B1

Algorithm for calculation of diffusion

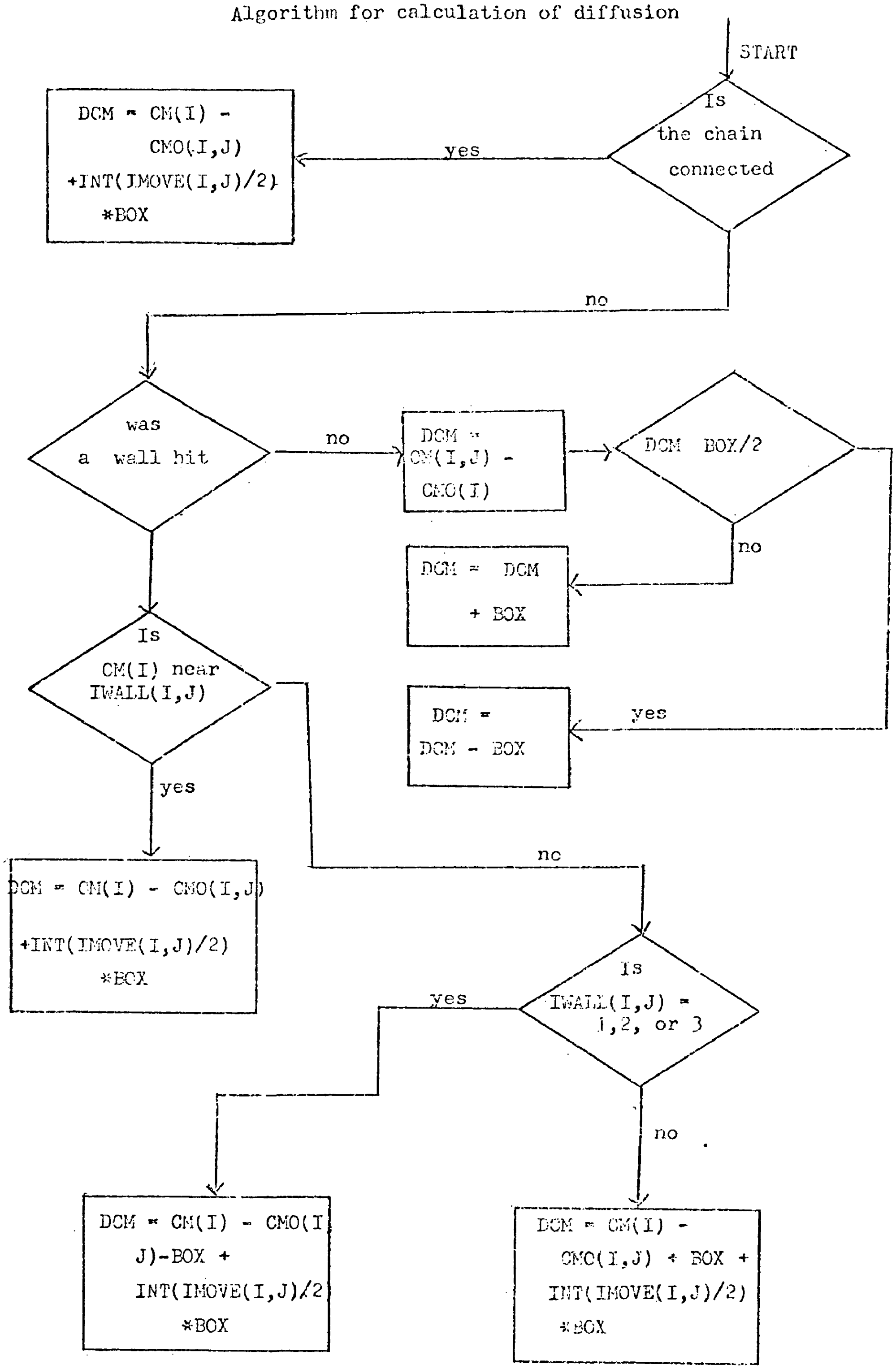
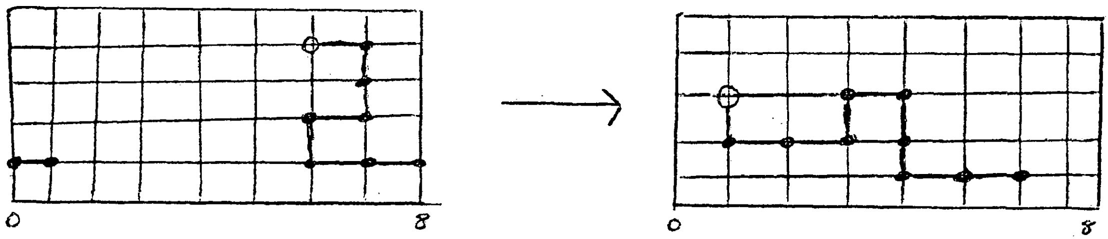


Figure B2

Example of diffusion in the

x dimension for a

1 chain system



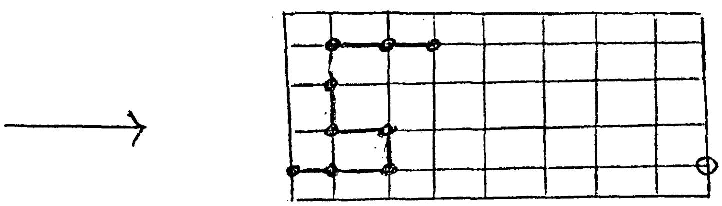
CMO(1,1) = 7.3

IMOVE(1,1) = 1

IWALL(1,1) = 0

IMOVE(1,1) = 2

IWALL(1,1) = 4



CM(1) = 10.2 - 8 = 2.2

IMOVE(1,1) = 1

IWALL(1,1) = 1

APPENDIX C

The Simulation Program

```
MEMBER NAME   MAIN
C
C        THIS PROGRAM HANDLES ONLY CHAINS THAT ARE ALL
C        THE SAME LENGTH,GREATER THAN 4 BEADS LONG
C           IT IS FOR EXCLUDED VOLUME CONDITIONS ONLY
C     DIFFUSION CALCULATIONS ARE EXACT, BUT THE BOX
C     SIZE MUST BE AT LEAST AS LARGE AS THE CHAIN
C     LENGTH FOR THE DIFFUSION TO BE CORRECT.
C
C    ****************** MAXIMUM DIMENSIONS ***********************
C     NO. CHAINS TO BE SAMPLED = 50
C     NO. BEADS = 15000
C     NO. RENS = 30
C     NO. FRAMES = 500
C     NO. SAMPLES = 35
C     MAXIMUM BOX SIZE = 32X32X32
C    ************************************************************
      LOGICAL*1 HEADER(80),SPACE
C
      COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NRUN,
     2 IRUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),NTAB(15000),MAP(4096),
     4 NSTART(15000),NEND(15000)
C
C
      COMMON/CORRFN/FLSQR(50),FLFOUR(50),NCYCS(35),TCYCS(35),TCNC(35),
     1 CORSM(2,50,35),CORFN(4,50,35),ISAMP,INCR,BOX(3),SDCM(3,50),
     2 SVDCM(50),KSB(50),KFRAME(50),D2(50),D4(50),AVGL2(2,50),
     3 AVGL4(2,50)
C
C
      COMMON/CROSS/ALF(4,50,500),CROSS(5,2,50),START(3,50),START2(50),
     2 RCROSS(5,2,50),RCRCS2(5,2,50),CCROSS(5,2),CCROS2(5,2)
C
      COMMON/FREQ/D2FR(30),NFN(50)
C
      DATA SPACE/1H /
      DIMENSION AVG(4),CORFC(4)
C
      INTEGER*4 LTAB(3)
C       READ THE RANDOM NUMBER SEED
      READ 10,IRAN
   10 FORMAT(I10)
        READ 12,IIN,IOUT
   12   FORMAT(2I10)
C   IF IIN=0  READ IN LTAB
C   IF IIN=1  READ IN NTAB
C    IF IOUT=0  WRITE FINAL BEAD CONFIGURATION
C       ONLY ON PRINTER
C    IF IOUT=1   WRITE FINAL BEAD CONFIGURATION TO DISK FILE
C       AS SPECIFIED ON FT09F001 ON JCL
      DO 20 I=1,80
   20 HEADER(I)=SPACE
      READ (5,30) HEADER
   30 FORMAT (80A1)
      PRINT 40,HEADER
   40 FORMAT(1H1,80A1)
      READ 10,NRUN
```

```
MEMBER NAME  MAIN
      READ (5,50) N,NFRAME,KCHAIN
   50 FORMAT(3I10)
      WRITE (6,60) NRUN,NFRAME
   60 FORMAT(1H0,'THE NUMBER OF RUNS= ',I3,5X,'THE
     *NUMBER OF FRAMES PER RUN= ',I3)
      READ 70,IXS,IYS,IZS
   70 FORMAT (3I10)
      BOX(1)=IXS
      BOX(2)=IYS
      BOX(3)=IZS
      PRINT 80
   80 FORMAT(1H ,'       IXS          IYS          IZS')
      PRINT 90,IXS,IYS,IZS
   90 FORMAT(1H ,I9,2I10)
C
C  WHEN MORE THAN ONE CHAIN IS USED, TREAT THEM AS ONE CHAIN
C     WITH MORE THAN TWO END READS
      NCF=0
      READ (5,100) NCHAIN,(NSTART(I),NEND(I),I=1,NCHAIN)
  100 FORMAT (16I5)
      IF (NCHAIN.EQ.1) NCF=1
      IF (NSTART(NCHAIN).LT.9000) GO TO 130
      NB=NSTART(NCHAIN)-9000
      NOC=NEND(NCHAIN)
      KS=NCHAIN
      NCHAIN=NCHAIN+NOC-1
      DO 120 I=KS,NCHAIN
      IF (NCF.EQ.1) GO TO 110
      NSTART(I)=NEND(I-1)+1
      GO TO 120
  110 NSTART(I)=1
      NCF=0
  120 NEND(I)=NSTART(I)+NB-1
  130 CALL NCYC
C
C     INITIALIZE THE ARRAYS
C
        DO 135 I = 1,30
  135   D2FF(I) = 0
      DO 140 I=1,N
  140 NTAB(I)=0
      IF (IIN.EQ.1) GO TO 515
      DO 510 J=1,N
      READ 500,(LTAB(I),I=1,3)
      CALL PACK(LTAB,J)
  500 FORMAT(3I10)
  510 CONTINUE
      GO TO 516
  515  READ 525,(NTAB(I),I=1,N)
  525  FORMAT(8(Z8,4X))
 516  PRINT 520,IRAN,IR
  520 FORMAT('1','INITIAL CONFIGURATION  NFRAME',5X,'IRAN= '
     *,I12,5X,'IR=  ',I3)
      DO 550 J=1,NCHAIN
       J1=NSTART(J)
       J2=NEND(J)
       PRINT 540,J,(NTAB(I),I=J1,J2)
  540 FORMAT(' ',3X,'CHAIN#',I4/' ',8(Z8,4X))
```

```
MEMBER NAME  MAIN
  550 CONTINUE
      IF (NCHAIN.LE.50) ISAMP=NCHAIN
      IF (NCHAIN.LE.50) INCR=1
      IF (NCHAIN.LE.50) GO TO 560
      INCR=NCHAIN/50
      ISAMP=NCHAIN/INCR
      IF (INCR.EQ.0) INCR=1
      IF (ISAMP.GT.50) ISAMP=50
  560 DO 570 J=1,ISAMP
       MEN(J) = 0
      DO 570 I=1,2
      AVGL2(I,J)=0.0
      AVGL4(I,J)=0.0
      D4(J)=0.0
      D2(J)=0.0
       IF (NFRAME.LE.5) NTIME=NFRAME
       IF (NFRAME.GT.5) NTIME=5
      DO 570 KMM=1,NTIME
      RCROSS(KMM,I,J)=0.0
  570 RCROS2(KMM,I,J)=0.0
      IFRAME=1
      CALL LDMAP
      FRAME=NFRAME
      CALL CHMAP
      CALL CTMAP
      NCYCLE=0
      NTEV=0
      DO 580 I=1,4
      DO 580 J=1,ISAMP
      DO 580 K=1,KYS
  580 CORFN(I,J,K)=0.0
      CALL PRIME
C
C    COMPUTE THE FRACTION OF EXCLUDED VOLUME CONFLICTS
      PEVC=FLOAT(NTEV)/FLOAT(NCYCLE)
C
C     COMPUTE THE BEAD DENSITY
      BDEN=FLOAT(N)/(BOX(1)*BOX(2)*BOX(3))
       IF (IOUT.EQ.0) GO TO 585
      WRITE (9,595) (NTAB(J),J=1,N)
  595  FORMAT (8(Z8,4X))
C
C
C
C     COMPUTE AND PRINT DATA FOR EACH CHAIN
C
C
 585  DO 680 J=1,ISAMP
      PRINT 40,HEADER
      PRINT 590,NRUN,NFRAME,NCYCLE,PEVC
  590 FORMAT ('0','NRUN= ',I3,'NFRAME= ',I4,'TOTAL BEAD CYCLES= ',
     *I8,'%EV CONF=',F6.4)
C
C     COMPUTE THE NUMBER OF BEADS IN JTH CHAIN
      NB=NEND(J)-NSTART(J)+1
      PRINT 600,J,IXS,IYS,IZS,N,BDEN
  600 FORMAT('0','CHAIN # =',I2,'   BOX ',3I3,'      TOTAL # BEADS='
     *,I4,'     BDEN=',F6.4)
```

```
MEMBER NAME  MAIN
C
C       COMPUTE <L2>,<L4>,AND THIER STD OVER THE N RUNS
        AVGL2(2,J)=SQRT((AVGL2(2,J)-AVGL2(1,J)**2/NRUN)/(NRUN-1))
        AVGL2(1,J)=AVGL2(1,J)/NRUN
        AVGL4(2,J)=SQRT((AVGL4(2,J)-AVGL4(1,J)**2/NRUN)/(NRUN-1))
        AVGL4(1,J)=AVGL4(1,J)/NRUN
        PRINT 610,AVGL2(1,J),AVGL2(2,J),AVGL4(1,J),AVGL4(2,J)
  610   FORMAT('0','<L2>= ',F9.3,'STD= ',F9.3,'<L4>= ',F15.3,
       *' STD ',F9.3)
C
C       COMPUTE <D**2> AND ITS STD OVER THE RUNS
C
         D3 = 0.0
         IF ((NRN(J).EQ.0).OR.(NRN(J).EQ.1)) GO TO 619
        D3=SQRT((D4(J)-D2(J)**2/NRN(J))/(NRN(J)-1))
  619   IF (NRN(J).NE.0) D2(J)=D2(J)/NRN(J)
        IF (NRN(J).EQ.0) D2(J) = 0.0
  621   PRINT 620,D2(J),D3
  620   FORMAT('0','<D**2>/FRAME=',F8.3,'STD = ',F7.3)
        PRINT 630
  630   FORMAT('0','KOUNT     T    <P(L,L,T)>      STD    LN     <P(L2,L2,T)>
       *    STD      LN     NCYCS       TCYCS')
C
C       COMPUTE <P(L2,L2,T)>,<P(L2,L2,T)>, THIER STD AND NATURAL LOGS
        DO 670 K=1,KOUNT
        CORFN(2,J,K)=SQRT((CORFN(2,J,K)-CORFN(1,J,K)**2/NRUN)
       */(NRUN-1))
        CORFN(1,J,K)=CORFN(1,J,K)/NRUN
        CORFN(4,J,K)=SQRT((CORFN(4,J,K)-CORFN(3,J,K)**2/NRUN)
       */(NRUN-1))
        CORFN(3,J,K)=CORFN(3,J,K)/NRUN
        TEMP=CORFN(1,J,K)
        IF(TEMP.LE.0.0) TEMP=.0001
        ALN1=ALOG(TEMP)
        TEMP=CORFN(3,J,K)
        IF(TEMP.LE.0.0) TEMP=.00001
        ALN2=ALOG(TEMP)
        IF (K.NE.1) GO TO 650
        PRINT 640,K,(CORFN(I,J,K),I=1,2),ALN1,(CORFN(I,J,K),I=3,4),ALN2
  640   FORMAT(1H0,I3,9X,F9.3,3X,F6.3,3X,F7.3,F9.3,F6.3,F7.3)
        GO TO 670
  650   PRINT 660,K,TCNC(K-1),(CORFN(I,J,K),I=1,2),ALN1,(CORFN(I,J,K),
       *I=3,4),ALN2,NCYCS(K-1),TCYCS(K-1)
  660   FORMAT(' ',I3,2F9.3,6X,F6.3,F7.3,F9.3,F6.3,F7.3,I8,F12.0)
  670   CONTINUE
        DO 675 KMM=1,NTIME
        DO 675 I=1,2
        RCROS2(KMM,I,J)=SQRT((RCROS2(KMM,I,J)-RCROSS(KMM,I,J)**2/NRUN)/
       *(NRUN-1))
  675   RCROSS(KMM,I,J)=RCROSS(KMM,I,J)/NRUN
        PRINT 800
  800   FORMAT('   CROSS CORRELATION TABLE')
        DO 678 KMM=1,NTIME
  678    PRINT 677,KMM,(RCROSS(KMM,I,J),I=1,2),(RCROSS(KMM,I,J),I=1,2)
  677    FORMAT(' ',I4,4F12.4)
  680    CONTINUE
C
C       CALCULATE AVERAGES OVER ALL THE CHAINS
```

```
MEMBER NAME  MAIN
C
      IF (ISAMP.LE.2) GO TO 915
      PRINT 40,HEADER
      PRINT 690,KCHAIN,NRUN,NFRAME
  690 FORMAT('0AVERAGES OF ',I2,'CHAINS FOR ',I3,'RUNS OF ',I4,
     *'FRAMES EACH')
      PRINT 700,NCYCLE,PEVC,BDEN
  700 FORMAT(1H ,'TOTAL BEAD CYCLES',I8,'%EV/100',F6.4,
     *'BEAD DENSITY ',F6.4)
      PRINT 710,IXS,IYS,IZS,N,ISAMP
  710 FORMAT(1H ,'BOX    ',I2,2('X',I2),'TOTAL # OF BEADS   ',
     *I4,'TOTAL # OF CHAINS ',I3)
      D1=0.0
      D3=0.0
      MCH=0
      DO 720 I=1,4
  720 AVG(I)=0.0
      DO 730 J=1,ISAMP
      AVG(1)=AVGL2(1,J)+AVG(1)
      AVG(2)=AVGL2(1,J)**2+AVG(2)
      AVG(3)=AVGL4(1,J)+AVG(3)
      AVG(4)=AVGL4(1,J)**2+AVG(4)
      IF(D2(J).NE.0) MCH=MCH+1
      D1=D2(J)+D1
  730 D3=D2(J)**2+D3
      AVG(2)=SQRT((AVG(2)-AVG(1)**2/ISAMP)/(ISAMP-1))
      AVG(1)=AVG(1)/ISAMP
      AVG(4)=SQRT((AVG(4)-AVG(3)**2/ISAMP)/(ISAMP-1))
      AVG(3)=AVG(3)/ISAMP
      PRINT 610,(AVG(I),I=1,4)
       IF ((MCH.EQ.0).OR.(MCH.EQ.1)) GO TO731
      D3=SQRT((D3-D1**2/MCH)/(MCH-1))
      D1=D1/MCH
  731 PRINT 620,D1,D3
      PRINT 630
      DO 760 K=1,KOUNT
      DO 740 I=1,4
  740 CORFC(I)=0.0
      DO 750 J=1,ISAMP
      CORFC(1)=CORFN(1,J,K)+CORFC(1)
      CORFC(2)=CORFN(1,J,K)**2+CORFC(2)
      CORFC(3)=CORFN(3,J,K)+CORFC(3)
      CORFC(4)=CORFN(3,J,K)**2+CORFC(4)
  750 CONTINUE
      CORFC(2)=SQRT((CORFC(2)-CORFC(1)**2/ISAMP)/(ISAMP-1))
      CORFC(1)=CORFC(1)/ISAMP
      CORFC(4)=SQRT((CORFC(4)-CORFC(3)**2/ISAMP)/(ISAMP-1))
      CORFC(3)=CORFC(3)/ISAMP
      TEMP=CORFC(1)
      IF (TEMP.LE.0.0) TEMP=.0001
      ALN1=ALOG(TEMP)
      TEMP=CORFC(3)
      IF (TEMP.LE.0.0) TEMP=.0001
      ALN2=ALOG(TEMP)
      IF (K.NE.1) GO TO 755
      PRINT 640,K,(CORFC(I),I=1,2),ALN1,(CORFC(I),I=3,4),ALN2
       IP = 0
      WRITE(8,240) IP,CORFC(1)
```

```
MEMBER NAME  MAIN
      GO TO 760
  755 PRINT 660,K,TCNC(K-1),(CORFC(I),I=1,2),ALN1,(CORFC(I),I=3,4),
     *ALN2,NCYCS(K-1),TCYCS(K-1)
       WRITE(8,940) TCNC(K-1),CORFC(1)
  940  FORMAT(' ',2F6.3)
  760 CONTINUE
       IF (NFRAME.LE.5) NTIME=NFRAME
       IF (NFRAME.GT.5) NTIME=5
      DO 840 KMM=1,NTIME
      DO 840  I=1,2
      CCROSS(KMM,I)=0.0
 840  CCROS2(KMM,I)=0.0
      DO 770  J=1,ISAMP
      DO 770  KMM=1,NTIME
      CCROSS(KMM,1)=RCROSS(KMM,1,J)+CCROSS(KMM,1)
      CCROSS(KMM,2)=RCROSS(KMM,1,J)**2+CCROSS(KMM,2)
      CCROS2(KMM,1)=RCROSS(KMM,2,J)+CCROS2(KMM,1)
 770  CCROS2(KMM,2)=RCROSS(KMM,2,J)**2+CCROS2(KMM,2)
      DO 780  KMM=1,NTIME
      CCROSS(KMM,2)=SQRT((CCROSS(KMM,2)-CCROSS(KMM,1)**2/ISAMP)/
     *(ISAMP-1))
      CCROSS(KMM,1)=CCROSS(KMM,1)/ISAMP
      CCROS2(KMM,2)=SQRT((CCROS2(KMM,2)-CCROS2(KMM,1)**2/ISAMP)/
     *(ISAMP-1))
 780  CCROS2(KMM,1)=CCROS2(KMM,1)/ISAMP
      PRINT 810
  810 FORMAT('    CROSS CORRELATION TABLE')
      PRINT 820
 820  FORMAT('   TIME P(L,L)       STD     P(L2,L2)    STD')
      DO 790 KMM=1,NTIME
 790  PRINT 830,KMM,(CCROSS(KMM,I),I=1,2),(CCROS2(KMM,I),I=1,2)
  830 FORMAT(' ',I4,4F12.4)
      PRINT 900
  900  FORMAT(' THE DISTRIBUTION FUNCTION FOR D**2 IS ')
       PRINT 910
  910  FORMAT('0      D**2       F(D**2)')
  915   DO 920 I=1,30
       D2FR(I) = D2FR(I)/NRUN
       L=I**2
  920  PRINT 930,L,D2FR(I)
  930  FORMAT(' ',I8,5X,F8.4)
 1000 STOP
      END
      SUBROUTINE NCYC
C NCYC CALCULATES THE NUMBER OF CYCLES THAT WILL ELAPSE BETWEEN SAMPLES.
C
      COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NRUN,
     2 IRUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),NTAB(15000),MAP(4096),NSTART(15000),
     4 NEND(15000)
C
      COMMON/CORRFN/FLSQR(50),FLFOUR(50),NCYCS(35),TCYCS(35),TCNC(35),
     1 CORSM(2,50,35),CORFN(4,50,35),ISAMP,INCR,BCX(3),SDCM(3,50),
     2 SVDCM(50),KSB(50),KFRAME(50),D2(50),D4(50),AVGL2(2,50),
     3 AVGL4(2,50)
C
      READ 10, FACTOR, FCUBE, FADD
```

```
MEMBER NAME  MAIN
   10 FORMAT (3F10.4)
      IN = NEND(1)-NSTART(1)+1
      TOTAL = AINT(FCUEE*FLCAT(IN*IN*N))
      TOLD = 0.0
      KYS = 0
      KOUNT=0
      IS=0
C IF MORE THAN 50 SAMPLES ARE DESIRED, COUNTER MUST BE INCREASED.
      DO 50 I=1,50
      KYS = KYS + 1
      TNEW = INT(TOLD*FACTOR) + FADD
      IF(TNEW - TOTAL) 40,30,20
   20 TNEW = TOTAL
   30 NCYCS(I) = TNEW - TOLD
      TCYCS(I) = TNEW
      TCNC(I)=FCUBE*TCYCS(I)/TOTAL
      GO TO 50
   40 NCYCS(I) = TNEW-TOLD
      TCYCS(I) = TNEW
      TCNC(I)=FCUBE*TCYCS(I)/TOTAL
 50   TOLD = TNEW
   60 CONTINUE
      PRINT 70
   70 FORMAT (1H,'NCYCS RESPONDING')
      WRITE (6,80) KYS,N
   80 FORMAT (1H0,'NUMBER OF SAMPLES PER FRAME = ',I3,5X,
     *'NUMBER OF BEADS = ',I5)
      PRINT 90, FACTOR, FCUBE, FADD
   90 FORMAT  (1H0,'FACTOR = ',F10.3,5X,'FCUBE = ',F10.3,5X,'FADD = ',
     *F10.3)
      PRINT 200,TOTAL
  200 FORMAT (1H0,' TOTAL NUMBER OF BEAD CYCLES PER FRAME = ',F7.0)
      PRINT 210
  210 FORMAT (1H0,'TNEW = INT(TOLD*FACTOR) + FADD')
      PRINT 220
  220 FORMAT (1H0,'NO. OF SAMPLE      NCYCS        TCYCS         TCYCS/N**3')
      PRINT 230,(I,NCYCS(I),TCYCS(I),TCNC(I),I=1,KYS)
  230 FORMAT (1H,I8,I12,F12.1,F15.4)
      RETURN
      END
      SUBROUTINE PRIME
C
C
C
      COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NRUN,
     2 IRUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),NTAB(15000),MAP(4096),NSTART(15000),
     4 NEND(15000)
C
      COMMON/CORRFN/FLSQR(50),FLFOUR(50),NCYCS(35),TCYCS(35),TCNC(35),
     1 CORSM(2,50,35),CORFN(4,50,35),ISAMP,INCR,BOX(3),SDCM(3,50),
     2 SVDCM(50),KSB(50),KFRAME(50),D2(50),D4(50),AVGL2(2,50),
     3 AVGL4(2,50)
C
C
C
      INTEGER*4 T,NOFR
```

```
MEMBER NAME  MAIN
      REAL*4 IFREQ(30),SUM(50)                                          00004070
C                                                                       00004080
      COMMON/CROSS/ALF(4,50,500),CROSS(5,2,50),START(3,50),START2(50),  00004090
     2 RCROSS(5,2,50),RCROS2(5,2,50),CCRCSS(5,2),CCROS2(5,2)            00004100
C                                                                       00004110
       COMMON/FREQ/D2FR(30),NRN(50)                                     00004120
      DO 1000 IRUN=1,NRUN                                               00004130
      DO 5 I=1,30                                                       00004140
    5 IFREQ(I) = 0                                                      00004150
      NEVT=0                                                            00004160
      KOUNT=KYS+1                                                       00004170
      DO 10 J=1,ISAMP                                                   00004180
      SVDCM(J)=0.0                                                      00004190
       NOFR = 0                                                         00004210
       FLSOF(J)=0.0                                                     00004220
      FLFOUR(J)=0.0                                                     00004230
      KFRAME(J)=0.0                                                     00004240
       SUM(J) = 0.                                                      00004250
      DO 91 II=1,5                                                      00004260
      CROSS(II,1,J)=0.0                                                 00004270
   91 CROSS(II,2,J)=0.0                                                 00004280
      DO 10 I=1,3                                                       00004290
      SDCM(I,J)=0.0                                                     00004300
      DO 10 K=1,KOUNT                                                   00004310
      CORSM(1,J,K)=0.0                                                  00004320
   10 CORSM(2,J,K)=0.0                                                  00004330
      WRITE (6,30) IRUN                                                 00004340
   30 FORMAT('1',15(1H*),3X,'START OF RUN NUMBER',I2,3X,15(1H*))       00004350
      DO 100 IFRAME=1,NFRAME                                            00004360
      DO 40 J=1,ISAMP                                                   00004370
   40 KSB(J)=0                                                          00004380
      KOUNT=1                                                           00004390
      CALL SAMPLE                                                       00004400
      DO 50 I=1,KYS                                                     00004410
      KKK=NCYCS(I)                                                      00004420
      IF (KKK.LE.0) STOP 32                                             00004430
      CALL CYCLE(KKK)                                                   00004440
      NCYCLE=NCYCLE+KKK                                                 00004450
      KOUNT=I+1                                                         00004460
      CALL SAMPLE                                                       00004470
   50 CONTINUE                                                          00004480
      DO 60 J=1,ISAMP                                                   00004490
      IF (KSB(J).EQ.0) KFRAME(J)=KFRAME(J)+1                            00004500
      DO 60 I=1,3                                                       00004510
       IF (KSB(J).NE.0) GO TO 60                                        00004520
      SVDCM(J)=SDCM(I,J)**2+SVDCM(J)                                    00004530
       SUM(J) = SDCM(I,J)**2 +SUM(J)                                    00004540
   60 SDCM(I,J)=0.0                                                     00004550
       DO 161 J=1,ISAMP                                                 00004560
       IF (SUM(J).NE.0) NOFR = NOFR +1                                  00004570
       DO 161 II = 1,29                                                 00004580
  161 IF ((SUM(J).LT.II**2).AND.(SUM(J).NE.0)) IFREQ(II) = IFREQ(II) +1.00004590
       IFREQ(30) = NOFR                                                 00004600
   65 PRINT 70,IFRAME,IPAN,IR,NEVT                                      00004610
   70 FORMAT(' ','FRAME',I3,2X,'IPAN=  ',I10,2X,'IR= ',I3,2X,           00004620
     *'#EV CONFLICTS= ',I12,2X)                                         00004630
      IF (IFRAME.NE.NFRAME) GO TO 100                                   00004640
      PRINT 90,IRUN,IFRAME                                              00004650
```

```
MEMBER NAME  MAIN
  80  FORMAT('0','FINAL CONFIGURATION OF RUN #',I2,'FRAME #',I3)       00004660
       WRITE(7,82) (NTAB(J),J=1,N)                                     00004670
  82   FORMAT(8(Z8,4X))                                                00004680
       DO 90  J=1,KCHAIN                                               00004690
       J1=NSTART(J)                                                    00004700
       J2=NEND(J)                                                      00004710
       PRINT 85,J,(NTAB(I),I=J1,J2)                                    00004720
  85  FORMAT(' ',3X,'CHAIN# ',I4/' ',8(Z8,4X))                         00004730
  90  CONTINUE                                                         00004740
 100  CONTINUE                                                         00004750
      DO 95  J=1,ISAMP                                                 00004760
      IF (KFRAME(J).EQ.0) GO TO 95                                     00004770
      SVDCM(J)=SVDCM(J)/KFRAME(J)                                      00004780
  95  CONTINUE                                                         00004790
       DO 101 II = 1,30                                                00004800
       IF (NDFR.EQ.0) GO TO 101                                        00004810
       IFREQ(II) = IFREQ(II)/NDFR                                      00004820
 101   D2FR(II) = D2FR(II)+IFREQ(II)                                   00004830
      PRINT 110,IRUN,NFRAME                                            00004840
 110  FORMAT('1','AVGERAGE FOR RUN #',I2,'      FRAME# ',I4)           00004850
      DO 250  J=1,ISAMP                                                00004860
      PRINT 120,J                                                      00004870
 120  FORMAT('0','AVERAGES FOR CHAIN#',I2)                             00004880
      PRINT 130,FLSQR(J),FLFOUR(J)                                     00004890
 130  FORMAT('0','LSQR SUM= ',F10.1,10X,'LFOUR SUM= ',F10.1)           00004900
      FLSQR(J)=FLSQR(J)/(NFRAME*KYS)                                   00004910
      FLFOUR(J)=FLFOUR(J)/(NFRAME*KYS)                                 00004920
      AVGL2(1,J)=FLSQR(J)+AVGL2(1,J)                                   00004930
      AVGL4(1,J)=FLFOUR(J)+AVGL4(1,J)                                  00004940
      AVGL2(2,J)=AVGL2(2,J)+FLSQR(J)**2                                00004950
      AVGL4(2,J)=AVGL4(2,J)+FLFOUR(J)**2                               00004960
      PRINT 140                                                        00004970
 140  FORMAT('0','  AVG LSQR',26X,'AVGLFOUR')                          00004980
      PRINT 150,FLSQR(J),FLFOUR(J)                                     00004990
 150  FORMAT(1H ,F10.4,25X,F10.4)                                      00005000
      PRINT 160,SVDCM(J),KFRAME(J)                                     00005010
 160  FORMAT('0','<D**2>/FRAME',F10.4,10X,'NUMBER OF FRAMES = ',I4)    00005020
      D2(J)=SVDCM(J)+D2(J)                                             00005030
      D4(J)=SVDCM(J)**2+D4(J)                                          00005040
       IF (SVDCM(J).NE.0) NRN(J) = NRN(J) + 1                          00005041
      FLG=0.0                                                          00005060
      IF (FLFOUR(J)-FLSQR(J)**2.EQ.0.0) FLG=1.0                        00005070
      DO 200 I=1,KOUNT                                                 00005080
      CORSM(1,J,I)=CORSM(1,J,I)/(FLSQR(J)*NFRAME)                      00005090
      IF (FLG.EQ.1.0) GO TO 190                                        00005100
      CORSM(2,J,I)=(CORSM(2,J,I)/NFRAME-FLSQR(J)**2)                   00005110
     * /(FLFOUR(J)-FLSQR(J)**2)                                        00005120
      GO TO 200                                                        00005130
 190   CORSM(2,J,I)=1.0                                                00005140
 200  CONTINUE                                                         00005150
      PRINT 210,CORSM(1,J,1)                                           00005160
 210  FORMAT('0','CORSM(1,J,1)=   ',F13.4)                             00005170
      PRINT 260                                                        00005180
 260  FORMAT(1H,'KOUNT    AVCORR(L*L)     AVCOR(L2*L2)   TCYCS         00005190
     *  NCYCS     ')                                                   00005200
      I=1                                                              00005210
      PRINT 220,I,(CORSM(K,J,I),K=1,2)                                 00005220
      DO 230  I=2,KOUNT                                                00005230
```

```
MEMBER NAME  MAIN
      PRINT 220,I,(CORSM(K,J,I),K=1,2),TCYCS(I-1),NCYCS(I-1)
  220 FORMAT(1H ,I3,F12.4,9X,1PE12.4,0PF11.1,I7)
  230 CONTINUE
      DO 240 I=1,KOUNT
      CORFN(1,J,I)=CORSM(1,J,I)+CORFN(1,J,I)
      CORFN(2,J,I)=CORSM(1,J,I)**2+CORFN(2,J,I)
      CORFN(3,J,I)=CORSM(2,J,I)+CORFN(3,J,I)
  240 CORFN(4,J,I)=CORSM(2,J,I)**2+CORFN(4,J,I)
         IF (NFRAME.LE.5) NTIME=NFRAME
         IF (NFRAME.GT.5) NTIME=5
      DO 280 KMM=1,NTIME
      CROSS(KMM,1,J)=ALF(1,J,KMM)*START(1,J)+ALF(2,J,KMM)*START(2,J)
     *+ALF(3,J,KMM)*START(3,J)
      CROSS(KMM,2,J)=ALF(4,J,KMM)*START2(J)
      NF=NFRAME-KMM
      IF (NF.LE.1) GO TO 265
      DO 245 T=1,NF
      CROSS(KMM,1,J)=ALF(1,J,T)*ALF(1,J,T+KMM)+ALF(2,J,T)*ALF(2,J,T+KMM)
     *+ALF(3,J,T)*ALF(3,J,T+KMM)+CROSS(KMM,1,J)
      CROSS(KMM,2,J)=ALF(4,J,T)*ALF(4,J,T+KMM)+CROSS(KMM,2,J)
   245 CONTINUE
   265 NF=NFRAME-KMM+1
      CROSS(KMM,1,J)=CROSS(KMM,1,J)/(NF*FLSQR(J))
      CROSS(KMM,2,J)=(CROSS(KMM,2,J)/NF-FLSQR(J)**2)/
     *(FLFOUR(J)-FLSQR(J)**2)
      RCROSS(KMM,1,J)=CROSS(KMM,1,J)+RCROSS(KMM,1,J)
      RCROS2(KMM,1,J)=CROSS(KMM,1,J)**2+RCROS2(KMM,1,J)
      RCROSS(KMM,2,J)=CROSS(KMM,2,J)+RCROSS(KMM,2,J)
      RCROS2(KMM,2,J)=CROSS(KMM,2,J)**2+RCROS2(KMM,2,J)
   280 CONTINUE
      PRINT 290
   290 FORMAT('0','CROSS CORRELATION TABLE')
      PRINT 300
  300 FORMAT(' ','TIME(FRAMES),  <P(L*L)>      <P(L2,L2)> ')
      DO 320 KM=1,5
  320 PRINT 310,KM,CROSS(KM,1,J),CROSS(KM,2,J)
  310 FORMAT(' ',I2,5X,2F12.4)
   250 CONTINUE
       PRINT 350
   350 FORMAT(' THE DISTRIBUTION FUNCTION FOR D**2/FRAME IS ')
       PRINT 360
   360 FORMAT('0      D**2      F(D**2)'  )
       DO 370 I = 1,30
       L = I**2
   370 PRINT 380,L,IFREQ(I)
  380  FORMAT(' ', I8,5X,F8.4)
  1000 CONTINUE
      RETURN
      END
      SUBROUTINE SIMPLE
C
C
C
C
      COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NRUN,
     2 IFUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),NTAB(15000),MAP(4096),
```

```
MEMBER NAME  MAIN
     4 NSTART(15000),NEND(15000)
C
C
      COMMON/CORRFN/FLSQR(50),FLFOUR(50),NCYCS(35),TCYCS(35),TCNC(35),
     1 CORSM(2,50,35),CORFN(4,50,35),ISAMP,INCR,BOX(3),SDCM(3,50),
     2 SVDCM(50),KSB(50),KFRAME(50),D2(50),D4(50),AVGL2(2,50),
     3 AVGL4(2,50)
C
C
C
      COMMON/CROSS/ALF(4,50,500),CROSS(5,2,50),START(3,50),START2(50),
     2 FCROSS(5,2,50),RCROS2(5,2,50),CCROSS(5,2),CCROS2(5,2)
C
C
C
      INTEGER*4 SIDE(3)
      REAL*4 LSQR,LSQR1(50),CMN(3),CMD(3,50),DCM
      DIMENSION S(3),S1(3,50),B(3)
C
C
      J=0
      DO 140 IJ=1,NCHAIN,INCR
      J=J+1
      N1=NSTART(IJ)
      N2=NEND(IJ)
         DO 10 I=1,3
         CMN(I)=0
         B(I)=(BOX(I)-1.)/2.
         ISUM(I)=0
         ISQP(I)=0
         IS(I)=0
   10 CONTINUE
      CALL UNPACK(N1,N2)
        LSQR=0
      DO 20 I=1,3
      S(I)=FLOAT(IS(I))
      LSQR=LSQR+S(I)*S(I)
   20 CMN(I)=FLOAT(ISUM(I))/FLOAT(1+N2-N1)
C  PUT CENTER OF MASS INSIDE OF THE BOX
      DO 30 I=1,3
      IF (CMN(I).LT.0) CMN(I)=CMN(I)+BOX(I)
   30   IF (CMN(I).GE.BOX(I)) CMN(I)=CMN(I)-BOX(I)
      IF (KOUNT.NE.1) GO TO 50
        LSQR1(J)=LSQR
      DO 40 I=1,3
      IF (ISEP(I).GT.1) KSB(J)=1
   40 S1(I,J)=S(I)
        IF (IFRAME.NE.1) GO TO 45
        DO 45 I=1,3
        START(I,J)=S1(I,J)
        START2(J)=LSQR
   45   CONTINUE
      GO TO 100
   50 FLSQR(J)=FLSQR(J)+LSQR
      FLFOUR(J)=FLFOUR(J)+LSQR*LSQR
C
C   CALCULATE DIFFUSION
C
```

```
MEMBER NAME  MAIN
      DO 90 I=1,3
      SIDE(I)=0
      IF (CMN(I).LE.B(I)) SIDE(I)=I
      IF (CMN(I).GT.B(I)) SIDE(I)=I+3
      IF (ISEP(I).GT.1) KSB(J)=1
      IF (KSB(J).EQ.1) GO TO 100
      IF (ISEP(I).EQ.1) GO TO 60
C     CHAIN IS NOT SPLIT IN THE END
      KAT=IMOVE(I,J)/2
      DCM=CMN(I)-CMO(I,J)+KAT*BOX(I)
      IMOVE(I,J)=0
      GO TO 80
C     CHAIN IS SPLIT IN THE END
   60 IF (IWALL(I,J).NE.0) GO TO 70
C   THE WALL WAS NOT HIT
      DCM=CMN(I)-CMO(I,J)
      IF (DCM.GT.B(I)) DCM=DCM-BOX(I)
      B(I)=B(I)*(-1)
      IF (DCM.LT.B(I)) DCM=DCM+BOX(I)
      B(I)=B(I)*(-1)
      GO TO 80
C     A WALL WAS HIT,(CHAIN SPLIT)
   70 KAT=IMOVE(I,J)/2
      IF (IWALL(I,J).EQ.SIDE(I)) DCM=CMN(I)-CMO(I,J)+
     *KAT*BOX(I)/2
      IF (IWALL(I,J).EQ.SIDE(I)) GO TO 80
C     THE CENTER OF MASS IS NOT NEAR THE LAST WALL HIT
      IF (IWALL(I,J).EQ.I+3) DCM=CMN(I)-CMO(I,J)+BOX(I)
     *+BOX(I)*KAT
      IF (IWALL(I,J).EQ.I) DCM=CMN(I)-CMO(I,J)-BOX(I)
     *+BOX(I)*KAT
   80 SDCM(I,J)=DCM+SDCM(I,J)
   90 CONTINUE
  100 CORSM(1,J,KOUNT)=CORSM(1,J,KOUNT)+S(1)*S1(1,J)+
     *S(2)*S1(2,J)+S(3)*S1(3,J)
      CORSM(2,J,KOUNT)=CORSM(2,J,KOUNT)+LSQR1(J)*LSQR
      IF (KOUNT.LE.KYS) GO TO 105
      ALF(1,J,IFRAME)=S(1)
      ALF(2,J,IFRAME)=S(2)
      ALF(3,J,IFRAME)=S(3)
      ALF(4,J,IFRAME)=LSQR
  105 DO 110 I=1,3
      IWALL(I,J)=0
  110 IMOVE(I,J)=0
      DO 130 I=1,3
      IF (ISEP(I).EQ.1) GO TO 120
      GO TO 130
  120 IF (CMN(I).GT.B(I)) IMOVE(I,J)=1
      IF (CMN(I).LE.B(I)) IMOVE(I,J)=-1
  130 CMO(I,J)=CMN(I)
       IF (J.EQ.50) IJ=NCHAIN
  140 CONTINUE
 1000 CONTINUE
      RETURN
      END
      SUBROUTINE PRTMAP
C   PRTMAP  PRINTS NON-ZERO LOCATIONS OF MAP IF CTMAP OR CHMAP DETECT
C AN ERROR IN MAP. I IS THE LOCATION IN MAP, THAT IS THE ITH WORD IN MAP
```

```
MEMBER NAME  MAIN
C NOTE: THE FIRST WORD OF MAP CORRESPONDS TO Y=0,Z=0
C
      COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NPUN,
     2 IRUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),ENDMAP(15000),MAP(4096),NSATRT(15000),
     4 NEND(15000)
C
         WRITE(6,10) IPRINT
   10 FORMAT(1H0,'THE CTMAP SUBROUTINE FOUND ',I5  ,' BEADS IN MAP'/)
      WRITE (6,20)
   20 FORMAT (1H0//'*****      PRINTOUT OF MAP       *****'//1H0,
     1'LOCATION',3X,'Y',3X,'Z',18X,'MAP'/)
      DO 50 I=1,4096
      IF (MAP(I).EQ.0) GO TO 50
      IDUM=I
      IY=IDUM/128
      IZ=IDUM-(IY*128)-1
      IF(IZ.LE.63) GO TO 30
      IZ=IZ-64
   30 CONTINUE
      WRITE(6,40) I,IY,IZ,MAP(I)
   40   FORMAT (1H ,2X,I4,2X,2I4,12X,Z8)
   50 CONTINUE
      WRITE(6,  60)
   60 FORMAT(1H0,'END OF PRTMAP')
      STOP
      END
      SUBROUTINE PRINT
C
       COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NPUN,
     2 IRUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),NTAB(15000),MAP(4096),NSTART(15000),NEND(15000)
C
C     COMMON BLOCK MUST BE IN SAME ORDER IN ALL SUBPROGRAMS
C
C
      DO 20 J=1,NCHAIN
      J1=NSTART(J)
      J2=NEND(J)
       PRINT 10,J,(NTAB(I),I=J1,J2)
   10 FORMAT(' ',3X,'CHAIN#',I4/' ',8(Z8,4X))
   20  CONTINUE
      RETURN
      END
      SUBROUTINE SKJ
C
C     THIS SUBROUTINE DETERMINES IF A CHAIN IS TO BE SAMPLED OR NOT
C    UP TO 50 CHAINS MAY BE SAMPLED.  IF THE NUMBER OF CHAINS
C    IS > 50 THEN THE CHAINS TO BE SAMPLED ARE PICKED SO THAT
C    THEY ARE EQUALLY SPACED.
C
      COMMON N,IR,IRAN,IXS,IYS,IZS,IN,ICHAIN,ISUM(3),IS(3),ISEP(3),
     1 IFRAME,NFRAME,KCHAIN,NCHAIN,NURN,
     2 IRUN,NCYCLE,NEVT,KYS,IPRINT,KOUNT,TABLE(32),IMOVE(3,50),
     3 IWALL(3,50),NTAB(1500),MAP(4096),
     4 NSTART(15000),NEND(15000)
```

```
MEMBER NAME  MAIN
C
C
      COMMON/CORREN/FLSQR(50),FLFOUR(50),NCYCS(35),TCYCS(35),TCNC(35),
     1 CORSM(2,50,35),CORFN(4,50,35),ISAMP,INCR,BOX(3),SDCM(3,50),
     2 SVDCM(50),DSB(50),DFRAME(50),D2(50),D4(50),AVGL2(2,50),
     3 AVGL4(2,50)
C
C
      IN=0
      J=0
      DO 10 IJ=1,NCHAIN,INCR
      J=J+1
      IF (ICHAIN.EQ.IJ) IN=1
       IF (ICHAIN.EQ.IJ) ICHAIN=J
      IF (J.EQ.50) IJ=NCHAIN
   10  CONTINUE
      RETURN
      END
```

```
MEMBER NAME  LDMAP
LDMAP    CSECT
         SAVE  (14,12)     INITIAL LOADING OF MAP FROM NTAB
* 'LDMAP' INITIALIZES MAP TO ZERO AND THEN LOADS MAP ACCORDING TO  LTAE
* EVERY POSSIBLE BEAD LOCATION CORRESPONDS TO A BIT IN MAP.
* A BIT IN MAP IS ONE IF THE CORRESPONDING BEAD LOCATION CONTAINS
* A BEAD AND ZERO OTHERWISE.
* EACH WORD IN MAP CORRESPONDS TO A SET OF BEAD LOCATIONS WITH A
* PARTICULAR PAIR OF Y AND Z COORDINATES AND VARYING VALUES OF X.
* THE 32 BITS IN EACH WORD CORRESPOND TO EITHER THE 32 LOCATIONS WHERE
* X IS LESS THAN OR EQUAL TO 31 OR THE 32 LOCATIONS WHERE X IS GREATER
* THAN OR EQUAL TO 32.  TO DIFFERENTIATE BETWEEN THESE TWO CASES, A
* BIT IN THE ADDRESS OF MAP IS EITHER 1 OR 0.
* HENCE THE CORRESPONDENCE BETWEEN WORDS IN MAP AND BEAD LOCATIONS IS:
*    (X,Y,Z) <=> MAP(Y*(2**7) + Z + C*(2**6))
*     WHERE C = 0  IF X<32 AND  C=1 IF X>31.
* FOR ANY PARTICULAR WORD IN MAP, IF THE NTH BIT IS 1, THEN THERE IS A
* BEAD AT (N-1,Y,Z) OR (N-1+32,Y,Z).
* FOR EXAMPLE, IF MAP(B'00000000 00000000 00000100 00000011' =
*                     B'00000001 00000000 00110000 00000000',
* THEN THERE ARE BEADS AT (X,Y,Z)= (12,4,3), (13,4,3), (24,4,3)
* IF MAP(B'00000000 00000000 00000100 01000011')  =
*                     B'00000001 00000000 00110000 00000000',
* THEN THERE ARE BEADS AT (X,Y,Z) = (44,4,3),(45,4,3),(56,4,3)
         BALR  9,0
         USING *,9
         L     12,ACOM
      USING N,12
         ST    13,SAV1+4
         LA    13,SAV1
* TABLE(32)  IS INITIALIZED.  TABLE(I)=2**(I-1)
         LA    3,4
         LA    4,1
         LA    5,31
         ST    4,TABLE
L1       SLL   4,1
         ST    4,TABLE(3)
         A     3,=F'4'
         BCT   5,L1
*    ZERO OUT ALL OF MAP
         L     15,ACOM+4
         LA    1,0
         LA    2,4
         L     3,=F'16380'
         LA    5,0
AGAIN    ST    5,0(1,15)
         BXLE  1,2,AGAIN
*    LOAD MAP    (_ MEANS UNUSED BIT, | MEANS WORD BOUNDARY)
*          NOTE: VALUES FOR X, Y, AND Z OCCUPY 6 BITS
         SLR   1,1            REG 1 - INDEX OF NTAB
         L     11,N
         L     10,=F'31'
HERE     SLR   5,5     ----------------------------------------
         L     4,NTAB(1)      X__Y__Z|  IN REG 4
         SRDL  4,6            X__Y__|Z  IN REG 4 AND REG5
         SRL   4,1            X__Y_|Z  IN REG 4 AND REG5
         SRDL  4,9            X|__Y_Z  IN REG 4 AND REG5
         SRL   5,17           X IN REG 4 AND  Y_Z| IN REG 5
         CR    4,10           COMPARE X WITH 31
```

```
MEMBER NAME  LDMAP
         BNH   DONT
         S     4,=F'32'
         O     5,TWO7         PUT 1 IN UNUSED BIT BETWEEN Y AND Z
DONT     LR    3,5
         M     2,=F'4'            INDEX OF MAP
         MH    4,=H'4'            INDEX OF TABLE
         L     7,TABLE(4)
ALL      O     7,0(3,15)
         ST    7,0(3,15)
         A     1,=F'4'        INCREMENT NTAB POINTER
         BCT   11,HERE        ---------------------------------------------
         L     13,SAV1+4
         LM    14,15,12(13)
         LM    1,12,24(13)
         MVI   12(13),X'FF'
         BR    14
SAV1     DS    19F
TWO7     DC    X'00000040'
ACOM     DC    A(N)
         DC    A(MAP)
         COM
N        DS    F
         DS    27F
TABLE    DS    32F
         DS    300F
NTAB     DS    15000F
MAP      DS    4096F
         END   LDMAP
```

```
MEMBER NAME  CHAIN
CHAIN    CSECT
*  THIS SUBPROGRAM DETERMINES IF A WALL HIT SHOULD BE COUNTED
*  AS A HIT.   FOR EXAMPLE IF A LOW X WALL WAS HIT FOLLOWED BY
*  A HIGH X WALL HIT HEN THE NET EFFECT IS THAT NO WALL WAS
*  HIT AND IWALL MUST BE SET TO EITHER 0 OR TO THE NUMBER OF
*  THE WALL HIT PREVIOUS TO THESE TWO HITS.
*
         SAVE  (14,12)
         BALR  11,0
         USING *,11                 USE R11 FOR LOCAL ADDRESSING
         L     12,ACOM              R12 = ADDRESS OF BLANK COMMON
         USING N,12                 USE R12 FOR ADDRESSING BLANK COMMON
         ST    13,SAV+4
         LA    13,SAV
*  R7 CONTAINS THE NUMBER OF WALL HIT. THIS IS PASSED IN R7 FROM CYCLE.
*  IWALL = 1 IF THE LOW  X WALL WAS PENETRATED
*  IWALL = 2 IF THE HIGH X WALL WAS PENETRATED
*  IWALL = 3 IF THE LOW  Y WALL WAS PENETRATED
*  IWALL = 4 IF THE HIGH Y WALL WAS PENETRATED
*  IWALL = 5 IF THE LOW  Z WALL WAS PENETRATED
*  IWALL = 6 IF THE HIGH Z WALL WAS PENETRATED
*  IF THIS IS THE FIRST TIME THROUGH CYCLE THEN
*  ZERO THE ARRAYS ENDB,BEGIN,AND BOTH,OTHERWISE DON'T
         ST    3,PLACE
         L     4,0(1)               R4 = ADDRESS OF PASSED ARGUMENT
         L     4,0(4)               R4 = TIME
         C     4,=F'0'              IF TIME > 0 THEN
         BNE   NO                   DON'T ZERO ARRAYS
*                                   ZERO ARRAYS
         XR    3,3
         LA    5,150
SB       XR    9,9
         ST    9,TBL1(3)
         ST    9,COUNT(3)
        ST    9,TBL(3)
         LA    3,4(3)
         BCT   5,SB
         L     3,PLACE
NO       L     5,COUNT(3)           TEST TO SEE IF A WALL IN THIS
         C     5,=F'0'              DIMENSION WAS HIT BEFORE BY THIS
*                                   CHAIN
         BE    ONE                  IF NOT GO TO ONE
         C     5,=F'1'              IF THIS WALL HAS BEEN HIT BY THIS
         BE    TWO                  CHAIN ONCE BEFORE GO TO TWO
*  SAME WALL HAS BEEN HIT TWO TIMES BEFORE THIS HIT BY THE SAME CHAIN
         L     5,TBL(3)             R5 = LAST WALL HIT
         CR    5,7                  IF SAME WALL WAS HIT
         BE    ZIP                  THIS TIME  GO TO ZIP
         LA    5,1                  CANCEL LAST HIT
         ST    5,COUNT(3)           COUNT = 1
         L     5,TBL1(3)
         ST    5,IWALL(3)           IWALL IS THE WALL HIT 2 TIMES AGO
         XR    5,5
         ST    5,TBL(3)             TBL = 0
         B     RETURN
ZIP      LA    5,1                  WALL HIT THIRD TIME
         ST    5,COUNT(3)           SAME AS IF WALL HAS BEEN HIT ONCE
         XR    5,5
```

```
MEMBER NAME  CHAIN
         ST    5,TBL(3)             TBL = 0
         ST    7,IWALL(3)           IWALL = R7
         ST    7,TBL1(3)            TBL1 = R7
         B     RETURN
TWO      L     5,TBL1(3)            R5=IWALL FOR LAST WALL THITS CHAIN HI
         CR    5,7                  IF R5 NE CURRENT WALL HIT GO
         BNE   ONE2                 TO ONE2
*  WALL HAS BEEN HIT BY CHAIN 2 TIMES IN A ROW
         ST    7,IWALL(3)           IWALL(3) = R7
         ST    7,TBL(3)             TBL(3) = IWALL(3)
         LA    5,2
         ST    5,COUNT(3)           COUNT(3) = 2
         B     RETURN
ONE      ST    7,IWALL(3)           WALL HIT FOR FIRST TIME BY CHAIN
         LA    5,1(5)               R5 = R5 + 1
         ST    5,COUNT(3)           COUNT(3) = R5
         ST    7,TBL1(3)            TBL1(3) = IWALL
         B     RETURN               FINISHED
ONE2     XR    9,9                  OPPOSITE WALL HIT, SO EFFECT
         ST    9,IWALL(3)           IS TO CANCEL OUT
         ST    9,TBL1(3)            THEREFORE ZERO IWALL,TBL1
         ST    9,TBL(3)             TBL AND COUNT
         ST    9,COUNT(3)
RETURN   L     13,SAV+4
         LM    14,15,12(13)
         LM    1,12,24(13)
         MVI   12(13),X'FF'
         BR    14
SAV      DS    18F
TBL1     DS    150F                 THE WALL HIT 2 TIMES BEFORE BY
*                                   GIVEN CHAIN
TBL      DS    150F                 THE LAST WALL HIT BY GIVEN CHAIN
COUNT    DS    150F                 NUMBER OF TIMES A CHAIN HAS
*                                   HIT A GIVEN WALL FROM 0 TO 2
TIME     DS    F                    TIME = 0 FIRST TIME BRANCH FROM
*                                   CYCLE OCCURS
PLACE    DS    F                    INDEX FOR VARIABLES IN ALL
*                                   ARRAYS IN THIS SUBPROGRAM
ACOM     DC    A(N)
         COM
N        DS    F
         DS    209F
IWALL    DS    150F
         END   CHAIN
```

```
MEMBER NAME  CYCLE
CYCLE    CSECT                      SUBROUTINE TO PERFORM BEAD MOVEMENTS
         B     12(15)               BRANCH AROUND CONSTANTS
         DC    X'7'                 LENGTH OF ENTRY POINT NAME
         DC    CL7'CYCLE'           ENTRY POINT NAME
         SAVE  (14,12)              SAVE ALL REGISTERS EXCEPT 13
         L     2,0(1)               R2 = ADDRESS OF THE PASSED ARGUMENT
         L     2,0(2)               R2 = THE NUMBER OF TIMES TO CYCLE
         BALR  12,0                 LOAD BASE ADDRESS
         USING *,12                 USE R12 IN LOCAL ADDRESSING
         L     11,ACOM
         USING N,11
         ST    13,SAV+4             SET UP SUBROUTINE SAVE AREA
         LA    13,SAV
         LA    4,0                  TIME = 0
         ST    4,TIME               FIRST TIME THROUGH CHAIN
START    XR    3,3
         ST    3,ICHAIN             ICHAIN = 0
         L     5,IRAN               ************************************
         M     4,MULT
         N     5,QQ                 GENERATE A RANDOM NUMBER OF
         ST    5,IRAN               ZERO OR ONE
         LA    3,2                  IF IT IS 0, DO A CRANKSHAFT
         SLL   3,1                  MOVE
         MR    4,3                  IF IT IS 1 DO A SINGLE
         C     4,=F'0'              BEAD MOVEMENT
         BE    STARTC
*
*              SINGLE BEAD MOVEMENT SELECTED
*
         L     5,IRAN               ************************************
         M     4,MULT
         N     5,QQ
         ST    5,IRAN               GENERATE A
         L     3,N                  RANDOM NUMBER FROM 0 TO N-1
         SLL   3,1                  R3 = R3 * 2
         MR    4,3
         ST    4,IR                 ************************************
*  DECIDE WHICH CHAIN IS MOVING.  THIS IS NEEDED FOR ADDRESSING
*  ARRAYS USED IN DIFFUSION CALCULATIONS.
         XR    7,7                  R7 = 0
         LA    3,1                  R3 = 1 INCREMENT BY 1 EACH TIME
*                                   THROUGH THE LOOP
         XR    6,6                  R6=0  USE TO INDEX NEND
         L     9,KCHAIN             R9 = NO OF CHAINS TO BE SAMPLED
         L     4,IR
         LA    4,1(4)               R4 = NO. OF MOVING BEAD FROM 1 TO N
LOOPC    L     10,ACOM+8            R10 = ADDRESS OF NEND
         L     8,0(6,10)            R8=NEND(6)
         CR    4,8                  COMPARE IR+1 TO NEND(6)
         BH    ROM                  IR+1 IS NOT LE NEND, SO CHAIN
*                                   NUMBER IS NOT ICHAIN
         ST    3,ICHAIN             IR+1 <= NEND(6) SO CHAIN NO = R3
         B     FOUND                JUMP OUT OF LOOP
ROM      LA    3,1(3)               ICHAIN = ICHAIN +1
         LA    6,4(6)               R6 = R6+4
         BCT   9,LOOPC              SEE IF CHAIN NO IS ICHAIN
FOUND    S     3,=F'1'              CHAIN NO IS BETWEEN 0 AND N-1
         SLL   3,2                  ICHAIN = ICHAIN *4
```

```
MEMBER NAME  CYCLE
         ST    4,IRE              ************************************
         C     4,=F'1'            IF IRE = 0 OR 1 THEN MOVE IN
         BNH   TRYO               Z DIMENSION
         C     4,=F'3'            IF IRE = 2 OR 3 THEN MOVE IN
         BNH   TRY1               Y DIMENSION
         L     10,=X'00000001'    IRE = 4 OR 5 SO MOVE IN
         C     4,=F'4'            X DIMENSION
         BE    ADD                IF IRE = 4 ADD 1 TO X DIMENSION
         B     SUBT               IF IRE = 5 SUBTRACT 1 FROM X DIM.
TRY1     L     10,=X'00000100'    R10 = A 1 IN Y COORD. POSITION
         C     4,=F'3'
         BL    ADD                IF IRE = 3 THEN ADD 1 TO Y POSITION
         B     SUBT               IF IRE = 2 THEN SUBTRACT 1 FROM Y
*                                 THE Y POSITION
TRYO     L     10,=X'00010000'    R10 = A 1 IN Z POSITION
         C     4,=F'0'
         BE    ADD                IF IRE = 0 THEN ADD 1 TO Z POSITION
*                                 IF IRE = 1 THEN SUBTRACT
*                                 1 FROM Z DIMENSION
SUBT     SR    8,10               MOVE AN END BEAD
         B     STSTAB
ADD      AR    8,10
STSTAB   ST    8,STAB             STAB = MOVED BEAD COORDINATES
TEST     L     4,STAB             R4 = STAB = NEW BEAD POSITION
         SRDL  4,8                X_ _Y_ _|Z IN REG 4 AND REG 5
         SRL   5,24               R5 = Z COORDINATE OF NEW BEAD COORD.
         L     3,IZS              R3 = Z DIMENSION  OF THE BOX
         C     5,=F'128'          COMPARE R5 TO -1
         BNL   TEST1              Z IS < 0
         C     5,IZS              COMPARE R5 TO IZS
         BNL   TEST2              Z IS > IZS
         SRDL  4,8                Y AT TOP OF R5
         SRL   5,24               Y AT BOTTOM OF R5
         L     3,IYS              R3 = IYS
         SLL   3,8                R3 = __IYS_
         C     5,=F'128'          COMPARE R5 TO -1
         BNL   TEST1              Y IS < 0
         C     5,IYS              COMPARE R5 TO IYS
         BNL   TEST2              Y IS > IYS
         SRDL  4,8                X AT TOP OF R5
         SRL   5,24               X AT BOTTOM OF R5
         L     3,IXS              R3 = IXS
         SLL   3,16               R3 = _IXS__
         C     5,=F'128'          COMPARE R5 TO -1
         BNL   TEST1              X IS < 0
         C     5,IXS              COMPARE R5 TO IXS
         BNL   TEST2              X IS > IXS
         B     EXVOL              X,Y,Z IN BOX
TEST1    L     5,STAB             X,Y, OR Z < 0
         AR    5,3                ADD IXS,IYS, OR IZS
         B     HIT
TEST2    L     5,STAB             X,Y, OR Z > IXS,IYS, OR IZS
         SR    5,3                SUBTRACT IXS,IYS, OR IZS
HIT      ST    5,STAB             STORE NEW BEAD COORDINATES IN STAB
EXVOL    L     4,STAB             R4 = STAB
         SRDL  4,6
         SRL   4,1
         SRDL  4,9
```

```
MEMBER NAME  CYCLE
         SRL   5,17                                              00001750
         SLL   5,2                 R3 = INDEX OF MAP             00001760
         SLL   4,2                 R4 = INDEX OF TABLE           00001770
         L     7,TABLE(4)                                        00001780
*       TEST TO SEE IF BEAD IS THERE *                           00001790
         L     10,ACOM+12                                        00001800
         N     7,0(5,10)                                         00001810
         BNZ   FINISV                                            00001820
* NO BEAD THERE SO PUT 1 IN MAP FOR NEW LOCATION                 00001830
         L     7,TABLE(4)                                        00001840
         L     10,ACOM+12                                        00001850
         O     7,0(5,10)                                         00001860
         ST    7,0(5,10)                                         00001870
*   REMOVE 1 IN MAP FOR OLD LOCATION                             00001880
         L     3,PLACE             INDEX IN NTAB OF MOVING BEAD  00001890
         L     4,NTAB(3)                                         00001900
         ST    4,OLDC              OLDC = COORDINATES BEFORE MOVE 00001910
         SRDL  4,6                                               00001920
         SRL   4,1                                               00001930
         SRDL  4,9                                               00001940
         SRL   5,17                                              00001950
         SLL   5,2                 R5 = INDEX IN MAP OF OLD NTAB(IR) 00001960
         SLL   4,2                 R4 = INDEX OF TABLE           00001970
         L     7,TABLE(4)                                        00001980
         X     7,0(5,10)                                         00001990
         ST    7,0(5,10)                                         00002000
*   UPDATE NTAB WITH NEW MOVED BEAD FROM STAB                    00002010
         L     4,STAB              NO EXCLUDED VOLUME CHECKS NEEDED 00002020
         L     3,PLACE                                           00002030
         ST    4,NTAB(3)           STORE NEW BEAD COORDINATES    00002040
*   TEST TO SEE IF ENDBEAD MOVED THROUGH A WALL                  00002050
*    USE FOR DIFFUSION CALCULATIONS                              00002060
         L     7,BEAD              IF A MIDBEAD MOVED            00002070
         C     7,=F'1'             THROUGH THE WALL IT DOSEN'T   00002080
         BNE   NEWCYC              MATTER FOR DIFFUSION PURPOSES 00002090
         L     3,ICHAIN            R3 IS USED TO CALCULATE THE ADDRESS 00002100
*                                  OF CHAIN IN IMOVE AND IWALL   00002110
         S     3,=F'1'             IF CHAIN IS TO BE SAMPLED     00002120
         MH    3,=H'12'                                          00002130
         LA    3,B(3)              ADDRESS OF CHAIN IN Z DIMENSION 00002140
         B     OUT                                               00002170
STARTC   L     5,IRAN         ********************************** 00002180
         M     4,MULT                                            00002190
         N     5,QQ                                              00002200
         ST    5,IRAN              GENERATE A                    00002210
         L     3,N                   RANDOM NUMBER FROM 0 TO N+  00002220
         A     3,KCHAIN                NCHAIN-1                  00002230
         SLL   3,1                 R3 = R3 * 2                   00002240
         MR    4,3                                               00002250
         ST    4,IR             ******************************** 00002260
*                       DETERMINE WHICH CHAIN IS MOVEING         00002270
         XR    7,7                 R7 = 0                        00002280
         ST    7,BEAD                                            00002290
         LA    3,1                 R3 = 1 INCREMENT BY 1 EACH TIME 00002300
*                                  THROUTH THE LOOP              00002310
         XR    6,6                 R6 = 0  USE TO INDEX NEND     00002320
         L     9,KCHAIN            R9 = NO OF CHAINS TO BE SAMPLED 00002330
         L     4,IR                R4 = NUMBER OF MOVING BOND    00002340
```

```
MEMBER NAME  CYCLE
LOOPCC   L     10,ACOM+8          R10 = ADDRESS OF NEND
         L     8,0(10,6)          R8 = NEND(6)
         CR    4,8                COMPARE IR TO NEND(6)
         BH    ROMC               IR IS NOT LESS THAN NEND  TRY AGAIN
         ST    3,ICHAIN           IR <=NEND  ICHAIN = R3
         B     FOUNDC             JUMP OUT OF LOOP
ROMC     LA    3,1(3)             ICHAIN = ICHAIN +1
         LA    6,4(6)             R6 = R6+4
         S     4,=F'1'            THERE ARE 1 MORE BONDS THAN BEADS
         BCT   9,LOOPCC
FOUNDC   L     4,IR
         A     4,=F'1'
         S     4,ICHAIN
         ST    4,IR               IR = IR+1 - ICHAIN
         L     4,ICHAIN
         S     4,=F'1'            R4 = CHAIN NO BETWEEN 0 AND N-1
         SLL   4,2                R4 = ADDRESS IN NSTART AND NEND
         L     6,ACOM+4         R6=ADDRESS OF NSTART
         L     7,0(6,4)          R7 = NSTART OF MOVING CHAIN
         ST    7,BEGIN             BEGIN = NSTART OF MOVING CHAIN
         L     6,ACOM+8          R6 = ADDRESS OF NEND
         L     8,0(6,4)         R8=NEND OF MOVING CHAIN
         ST    8,ENDB
         L     4,IR
         L     5,BEGIN            IR < NSTART MOVE FIRST BOND
         CR    4,5                IR = NSTART   MOVE SECOND BOND
         BL    READ1C             R5 = NEND
         BE    ONEC
         L     5,ENDB
         CR    4,5
         BE    READNC             IR = NEND MOVE BONDN
         S     5,=F'1'
         CR    4,5
         BE    ENDC               IR = NEND -1   MOVE BOND N-1
*                                 THE BOND MOVING IS NCT ON THE END
MIDBNDC  S     4,=F'1'
         SLL   4,2
         L     6,NTAB-4(4)        R6 = NTAB(IR-1)
         A     6,NTAB+8(4)        R6 = NTAB(IR-1) + NTAB(IR+1)
         ST    6,INBET            INBET = R6
         S     6,NTAB+4(4)        R6 = INBET - NTAB(IR+1)
         ST    6,STAB             STAB = R6
         L     6,INBET
         S     6,NTAB(4)          R6 = INBET - NTAB(IR)
         ST    6,STAB1            STAB1 = R6
         ST    4,PLACE            PLACE = R4
         LA    7,1
         ST    7,BEAD             BEAD = 1
         B     TESTCC
ONEC     L     4,IR               MOVE FIRST BOND IN CHAIN
         SLL   4,2
         L     8,NTAB+4(4)        R8 = NTAB(IR+1)
         LA    5,2
         ST    5,BEAD             BEAD =2
         B     STPLACEC
ENDC     L      4,IR              MOVE LAST BOND IN CHAIN
         SLL   4,2                R4 = R4 * 4
         L     8,NTAB-8(4)        R8 = NTAB(IR -2)
```

```
MEMBER NAME  CYCLE
             LA      5,3
             ST      5,BEAD                   BEAD = 3
             B       STPLACEC
*                                             FIRST BEAD IN A CHAIN SELECTED
BEAD1C       L       4,IR
             SLL     4,2                      R4 = 4*(IR - 1)      INDEX REGISTER
             L       6,NTAB(4)                R6 = ORC OF FIRST BEAD IN CHAIN
             L       8,NTAB+4(4)              R8 = ORC OF SECOND BEAD IN CHAIN
             LA      5,0
             ST      5,BEAD
             B       STPLACEC
*                                             THE LAST BEAD IN A CHAIN SELECTED
BEADNC       L       4,IR
             S       4,=F'1'
             SLL     4,2
             L       6,NTAB(4)                R6 = ORC OF LAST BEAD IN CHAIN
             L       8,NTAB-4(4)              R8=ORC OF NEXT TO LAST BEAD IN CHAIN
             LA      5,0
             ST      5,BEAD
STPLACEC     ST      4,PLACE                  PLACE=ADDRESS IN NTAB OF MOVING BEAD
RANENDC      L       5,IRAN                   **********************************
             M       4,MULT
             N       5,QQ                     GENERATE A RANDOM NUMBER FROM 0 TO 5
             ST      5,IRAN
             LA      3,12
             MR      4,3
             ST      4,IRE                    ************************************
             C       4,=F'1'                  IF IRE = 0 OR 1 THEN MOVE IN
             BNH     TRYOC                    Z DIMENSION
             C       4,=F'3'                  IF IRE = 2 OR 3 THEN MOVE IN
             BNH     TRY1C                    Y DIMENSION
             L       10,=X'0C000001'          IRE = 4 OR 5 SO MOVE IN
             C       4,=F'4'                  X DIMENSION
             BE      ADDC                     IF IRE = 4 ADD 1 TO X DIMENSION
             B       SUBTC                    IF IRE = 5 SUBTRACT 1 FROM X DIM.
TRY1C        L       10,=X'00000100'          R10 = A 1 IN Y COORD. POSITION
             C       4,=F'3'
             BL      ADDC                     IF IRE = 3 THEN ADD 1 TO Y POSITION
             B       SUBTC                    IF IRE = 2 THEN SUBTRACT 1 FROM Y
*                                             THE Y POSITION
TRYOC        L       10,=X'00010000'          R10 = A 1 IN Z POSITION
             C       4,=F'0'
             BE      ADDC                     IF IRE = 0 THEN ADD 1 TO Z POSITION
*                                             IF IRE = 1 THEN SUBTRACT
*                                             1 FROM Z DIMENSION
SUBTC        SR      8,10                     MOVE AN END BEAD
             B       STSTAVC
ADDC         AR      8,10
STSTAVC      ST      8,STAB                   STAB = MOVED BEAD COORDINATES
             XR      8,8                      R8 = 0
             L       4,PLACE                  R4 = ADDRESS IN NTAB OF MOVING BEAD
             L       5,BEAD
             C       5,=F'2'
             BL      TESTCC                   NO MORE MOVES NECESSARY
             BE      ONEMC
             L       5,STAB
             L       4,PLACE
             A       5,NTAB(4)                R5 = STAB + NTAB(IR)
```

```
MEMBER NAME  CYCLE
        S     5,NTAB-4(4)       R5 = R5 - NTAB(IR-1)
        S     4,=F'4'           PLACE = PLACE -4
        ST    4,PLACE
        ST    5,STAB1           STAB1 = R5
        L     4,STAB1
        B     TESTCC
ONEMC   L     5,STAB            MOVE FIRST BOND IN CHAIN
        L     4,PLACE
        A     5,NTAB-4(4)       R5 = STAB + NTAB(IR-1)
        S     5,NTAB(4)         R5 = R5 - NTAB(IR)
        L     6,STAB
        ST    6,STAB1           STAB1 = STAB
        ST    5,STAB            NTAB(4) = R5
        L     5,PLACE
        S     5,=F'4'
        ST    5,PLACE           PLACE = PLACE  4
TESTCC    LA    10,2
          L      4,STAB          R4 = STAB = NEW BEAD COORDINATE
STILLC    C      10,=F'2'
        BE    AC
        L     4,STAB1           TEST STAB1 SECOND TIME THROUGH
AC        SRDL  4,8              X__Y__|Z  IN REG 4 AND REG 5
        SRL   5,24              R5 = Z COORDINATE OF NEW BEAD COORD.
        L     3,IZS             R3 = Z DIMENSION  OF THE BOX
        L     9,=F'1'           R9 = 1 FOR Z DIMENSION
        C     5,=F'128'         COMPARE Z TO 128
        BNL   TESTCC1           Z IS < 0
        C     5,IZS             COMPARE Z TO IZS
        BNL   TESTCC2           Z IS > IZS
A1C       L      4,STAB          TEST STAB
        C     10,=F'2'
        BE    NNC
        L     4,STAB1           TEST STAB1 SECOND TIME THROUGH
NNC     SRDL  4,16              Y AT TOP OF R5
        SRL   5,24              Y AT BOTTOM OF R5
        L     3,IYS             R3 = IYS
        SLL   3,8               R3 =    IYS
        L     9,=F'2'           R9 = 2 FOR Y DIMENSION
        C     5,=F'128'         COMPARE R5 TO -1
        BNL   TESTCC1           Y IS < 0
        C     5,IYS             COMPARE R5 TO IYS
        BNL   TESTCC2           Y IS > IYS
A2C       L      4,STAB          TEST STAB FIRST TIME THROUGH
        C     10,=F'2'
        BE    NNNC
        L     4,STAB1           TEST STAB1 SECOND TIME THROUGH
NNNC      SRDL  4,24             X AT TOP OF R5
        SRL   5,24              X AT BOTTOM OF R5
        L     3,IXS             R3 = IXS
        SLL   3,15              R3 =   IXS
        L     9,=F'3'           R9 = 3 FOR X DIMENSION
        C     5,=F'128'         COMPARE X TO -1
        BNL   TESTCC1           X IS < 0
        C     5,IXS
        BNL   TESTCC2
        L     9,BEAD
        C     9,=F'0'
        BE    EXVOLC            ONLY ONE BEAD MOVED
```

```
MEMBER NAME  CYCLE
             B      LOPCC
TESTCC1      C      10,=F'1'
             BE     TESTCC3                 IF STAB1 IS OUT ADD BOX TO IT
             L      5,STAB                     OTHERWISE
             AR     5,3                     STAB = STAB +IXS,IYS OR IZS
             ST     5,STAB
             B      HITC
TESTCC3      L      5,STAB1
             AR     5,3                     X,Y,OR Z IS < 0 FOR STAB1
             ST     5,STAB1                 ADD IXS,IYS OR IZS TO STAB1
             B      HITC
TESTCC2      C      10,=F'1'                IF STAB1 IS OUT SUBT BOX FROM IT
             BE     TESTCC4
             L      5,STAB                  X,Y, OR Z > IXS,IYS, OR IZS
             SR     5,3                     SO SUBTRACT IXS,IYS, OR IZS
             ST     5,STAB
             B      HITC
TESTCC4      L      5,STAB1                 STAB1 = STAB1 -IXS,IYS, OR IZS
             SR     5,3
             ST     5,STAB1
HITC         LA     8,1(8)
             L      8,BEAD
             C      8,=F'0'
             BE     EXVOLC                  ONLY ONE MOVED, DON'T CHECK STAB1
             C      9,=F'2'
             BL     A1C                     TEST X AND Y
             BE     A2C                     TEST X
LOPCC        BCT    10,STILLC
EXVOLC       L      4,STAB
             SRDL   4,6
             SRL    4,1
             SRDL   4,9
             SRL    5,17
             SLL    5,2                     R5 = INDEX OF MAP
             SLL    4,2                     P4 = INDEX OF TABLE
             L      7,TABLE(4)
*            TEST TO SEE IF BEAD IS THERE   *
             L      10,ACOM+12
             N      7,0(5,10)
             BNZ    FINISV                  EV CONFLICT ENCOUNTERED FOR STAB
             L      10,BEAD
             C      10,=F'0'
             BE     PUT                      IF ONLY ONE BEAD MOVED DON'T
*                                            CHECK EVC FOR STAB1
             ST     4,TABLEID               TABLEID=INDEX IN TABLE FOR STAB
             ST     5,MAPID                 MAPID=INDEX IN MAP FOR STAB
             L      4,STAB1
             SRDL   4,6
             SRL    4,1
             SRDL   4,9
             SRL    5,17
             SLL    5,2                     R5 = INDEX OF MAP FOR STAB1
             SLL    4,2                     R4 = INDEX OF TABLE FOR STAB1
             L      7,TABLE(4)
             L      10,ACOM+12              R10 = ADDRESS OF MAP
             N      7,0(5,10)               IS A 1 THERE?  IF SO
             BNZ    FINISV                  EVC ENCOUNTERED FOR STAB1
             ST     4,TBLID1                TBLID1 = INDEX IN TABLE FOR STAB1
```

```
MEMBER NAME  CYCLE
         ST    5,MAPID1            MAPID1=INDEX IN MAP FOR STAB1
         L     4,TABLEID           R4 = TABLEID
         L     3,MAPID             R5 = MAPID
*  NO EXCLUDED VOLUME CONFLICT FOR STAB OR STAB1 SO CHANGE MAP AND
* UPDATA NTAB FOR BOTH OF THEM
*  UPDATE MAP FOR NEW LOCATION OF STAB
PUT      L     7,TABLE(4)
         L     10,ACOM+12          R10 = ADDRESS OF MAP
         O     7,0(5,10)
         ST    7,0(5,10)           MAP(5) = 1
*    REMOVE 1 IN MAP FOR OLD LOCATION FOR STAB
         L     10,BEAD
         C     10,=F'3'
         BNE   HLS                 IF LAST BOND MOVED
         L     3,PLACE             THEIN PLACE=PLACE+4
         LA    3,4(3)
         L     4,NTAB(3)
         ST    4,OLDC              OLDC=NTAB OF LAST BEAD ON CHAIN
         L     3,PLACE
         L     4,NTAB(3)           R4 = NTAB OF LAST BEAD IN CHIAN
         B     HLS1
HLS      L     3,PLACE             END BOND DID NOT MOVE
         L     4,NTAB(3)           R4 = NTAB OF MOVING BEAD
         ST    4,OLDC              OLDC = NTAB(PLACE)
HLS1     SRDL  4,6
         SRL   4,1
         SRDL  4,9
         SRL   5,17
         SLL   5,2                 R5 = INDEX IN MAP OF OLD NTAB(IR)
         SLL   4,2                 R4 = INDEX OF TABLE
         L     7,TABLE(4)
         L     10,ACOM+12          R10 = ADDRESS OF MAP
         X     7,0(5,10)
         ST    7,0(5,10)           MAP(5) = 0
         L     10,BEAD
         C     10,=F'0'            IF ONLY ONE BEAD MOVED
         BE    NOEXVOLC            UPDATE NTAB
*   PUT 1 IN MAP FOR NEW LOCATION OF STAB1
         L     4,TBLID1
         L     5,MAPID1
         L     7,TABLE(4)
         L     10,ACOM+12          R10 = ADDRESS OF MAP
         O     7,0(5,10)
         ST    7,0(5,10)           MAP(5) = 1
         L     3,PLACE
         LA    3,4(3)
         L     4,NTAB(3)           R4 = OLD NTAB OF SECOND BEAD
*                                  BEFORE MOVE
*  REMOVE 1 IN MAP FOR OLD LOCATION OF STAB1
         SRDL  4,6
         SRL   4,1
         SRDL  4,9
         SRL   5,17
         SLL   5,2
         SLL   4,2
         L     7,TABLE(4)
         L     10,ACOM+12          R10 = ADDRESS OF MAP
         X     7,0(5,10)
```

```
MEMBER NAME  CYCLE
         ST    7,0(5,10)           MAP(5) = 0
*   UPDATE NTAB WITH NEW MOVED BEAD FROM STAB
NOEXVOLC L     4,STAB              NO EXCLUDED VOLUME CHECKS NEEDED
         L     3,PLACE
         ST    4,NTAB(3)           STORE NEW BEAD COORDINATES
         L     4,BEAD
         C     4,=F'0'             IF ONLY ONE BEAD MOVED THEN
         BE    OUT                 DON'T UPDATE STAB1 IF
*                                  IF ONLY ONE BEAD MOVED
         LA    3,4(3)
         L     4,STAB1
         ST    4,NTAB(3)           STORE MOVED BEAD COORDINATES
         L     4,BEAD
         C     4,=F'1'             IF MIDBEAD MOVED NO NEED
         BE    NEWCYC              TO CALCULATE DIFFUSION
*   TEST TO SEE IF ENDBEAD MOVED THROUGH A WALL
OUT      LA    1,ADDRLIST
         L     3,NCHAIN           R3 = TOTAL NUMBER OF CHAINS
         C     3,=F'51'           IF MORE THAN 50 CHAINS ALL ARE
         BL    FEW                SAMPLED
         L     15,ADDRSUB+4       R15 = ADDRESS OF SKJ
         BALR  14,15              BRANCH TO SKJ
         L     10,IN              CHECK TO SEE IF IN = 0
         C     10,=F'0'           IF IT IS THEN CHAIN IS NOT TO BE
         BE    NEWCYC             SAMPLED.
FEW      L     3,ICHAIN            R3 IS USED TO CALCULATE THE
         S     3,=F'1'             ADDRESS OF THE CHAIN IN IMOVE ADN
         MH    3,=H'12'            IWALL IF CHAIN IS SAMPLED
         ST    3,SAMP              ADDRESS OF CHAIN IN Z DIM
         L     8,OLDC              R8 = COORDINATE BEFORE MOVE
         L     6,STAB              R6 = COORDINATES AFTER MOVE
         L     10,BEAD
         C     10,=F'3'            IF LAST BEAD MOVED CHECK STAB1
         BNE   PW                  OTHERWISE CHECK STAB
         L     6,STAB1
*  TEST TO SEE OF OLD COORD. - NEW COORD. IS GREATER THAN |2|.
*  ALL 3 DIMENSIONS MUST BE CHECKED.  IF OBC-NBC > |2| THEN
*  A BOX WALL WAS PENETRATED.  ADD ONE TO IMOVE IF A HIGH WALL WAS
*  HIT OR SUBTRACT ONE IF A LOW WALL WAS HIT.  THEN BRANCH
*  TO SUBROUTINE CHAIN AND SET IWALL.
PW       SRDL  8,8
         SRDL  6,8
         SRL   9,24                NTAB(Z) IN R9
         SRL   7,24                STAB Z IN R7
         SR    9,7                 NTAB(Z)-STAB(Z) IN R9
         C     9,=F'2'
         BH    Z1                  HIGH WALL HIT
         C     9,=F'-2'
         BL    Z0                  LOW WALL HIT
BILBOC   SRDL  8,8
         SRDL  6,8
         SRL   9,24                NTAB(Y) IN R9
         SRL   7,24                STAB(Y) IN R7
         SR    9,7                 NTAB(Y)-STAB(Y) IN R9
         C     9,=F'2'
         BH    Y1                  HIGH WALL HIT
         C     9,=F'-2'
         BL    Y0                  LOW WALL HIT
```

```
MEMBER NAME  CYCLE
BAGGINSC  SRDL   5,8
          SRL    9,24              NTAB(X) IN R9
          SRDL   6,8
          SRL    7,24              STAB(X) IN R7
          SR     9,7               NTAB(X)-STAB(X) IN R9
          C      9,=F'2'
          BH     X1                HIGH WALL HIT
          C      9,=F'-2'
          BL     X0                LOW WALL HIT
          B      NEWCYC            ALL 3 DIMENSIONS HAVE BEEN CHECKED
*  THE FOLLOWING LINES UP TO FINISFV ARE BRANCHED TO IF A WALL
*  HAS BEEN PENETRATED.  OTHERWISE THEY ARE NOT EXECUTED.
X0        L      3,SAMP
          L      7,IMOVE(3)        LOW X WALL HIT
          S      7,=F'1'
          ST     7,IMOVE(3)        IMOVE=IMOVE-1
          LA     7,1               R7 = 1, PASS TO CHAIN
          L      15,ADDRSUB        R15 = ADDRESS OF CHAIN
          BALR   14,15             BRANCH TO CHAIN
          ST     14,TIME           R14 > 0
          B      NEWCYC            MOVE NEXT BEAD
X1        L      3,SAMP
          L      7,IMOVE(3)        HIGH X WALL HIT
          A      7,=F'1'
          ST     7,IMOVE(3)        IMOVE=IMOVE+1
          LA     7,4               PASS R7=4 TO CHAIN
          L      15,ADDRSUB        R15 = ADDRESS OF CHAIN
          BALR   14,15             BRANCH TO CHAIN
          ST     14,TIME           TIME > 0
          B      NEWCYC            MOVE NEXT BEAD
Y0        L      3,SAMP
          LA     3,4(3)
          L      7,IMOVE(3)        LOW Y WALL HIT
          S      7,=F'1'
          ST     7,IMOVE(3)        IMOVE = IMOVE-1
          LA     7,2               R7 = 2 PASS TO CHAIN
          L      15,ADDRSUB        R15 = ADDRESS OF CHAIN
          BALR   14,15             BRANCH TO CHAIN
          ST     14,TIME           TIME > 0
          B      BAGGINSC
Y1        L      3,SAMP
          LA     3,4(3)
          L      7,IMOVE(3)        HIGH Y WALL HIT
          A      7,=F'1'           R7 = R7+1
          ST     7,IMOVE(3)        IMOVE = IMOVE+1
          LA     7,5
          L      15,ADDRSUB        R15 = ADDRESS OF CHAIN
          BALR   14,15             BRANCH TO CHAIN
          ST     14,TIME           TIME > 0
          B      BAGGINSC          CHECK X
Z0        L      3,SAMP
          LA     3,8(3)
          L      7,IMOVE(3)        LOW Z WALL HIT
          S      7,=F'1'           R7 = R7-1
          ST     7,IMOVE(3)        IMOVE = IMOVE -1
          LA     7,3               R7 = 3   PASS TO CHAIN
          L      15,ADDRSUB        R15 = ADDRESS OF CHAIN
          BALR   14,15             BRANCH TO CHAIN
```

```
MEMBER NAME  CYCLE
         ST    14,TIME              TIME > 0
         B     BILBOC               CHECK X AND Y
Z1       L     3,SAMP
         LA    3,8(3)
         L     7,IMOVE(3)           HIGH Z WALL HIT
         A     7,=F'1'              R7 = R7 +1
         ST    7,IMOVE(3)           IMOVE = IMOVE +1
         LA    7,6
         L     15,ADDRSUB           R15 = ADDRESS OF CHAIN
         BALR  14,15                BRANCH TO CHAIN
         ST    14,TIME              R14 > 0
         B     BILBOC               CHECK X AND Y
FINISV   L     7,NEVT               EXCLUDED VOLUME CONFLICT
         LA    7,1(7)               R7 = R7 + 1
         ST    7,NEVT               NEVT = NEVT + 1
         ST    7,NEVT
NEWCYC   BCT   2,START
EXIT     L     13,SAV+4
         LM    14,15,12(13)
         LM    1,12,24(13)
         MVI   12(13),X'FF'
         BR    14
SAV      DS    18F
ADDRSUB  DC    V(CHAIN)
         DC    V(SKJ)
ADDRLIST DC    A(TIME)
MULT     DC    F'1953125'
QQ       DC    B'01111111111111111111111111111111'
SAMP     DS    F
PLACE    DS    F                    ADDRESS IN NTAB OF MOVING CHAIN
IRE      DS    F                    RANDOM NUMBER
TWO7     DC    X'00000040'
STAB     DS    F                    NEW BEAD COORDINATES
OLDC     DS    F                    OLD BEAD COORDINATES
*                                   USED TO SEE IF BEAD WENT
*                                   THROUGH A WALL IN DIFFUSION
*                                   CALCULATIONS
TIME     DS    F                    TIME= 0 FIRST TIME THRU CHAIN
BEAD     DS    F                    BEAD= 0 FOR SINGLE MIDBEAD MOVES
*                                             AND ALL END BEAD MOVES
*                                   BEAD= 1 IF CRANKSHAFT MIDBOND
*                                               MOVE OCCURED
*                                   BEAD= 2 FOR FIRST BOND IN CHAIN MOVED
*                                   BEAD= 3 FOR LAST BOND IN CHAIN MOVED
INBET    DS    F                    INTERMEDIATE RESULTS IN CRANKSHAFT
*                                   BEAD MOVEMENTS
TABLEID  DS    F                    INDEX IN TABLE FOR STAB
TBLID1   DS    F                    INDEX IN TABLE FOR STAB1
MAPID    DS    F                    INDEX IN MAP FOR STAB
MAPID1   DS    F                    INDEX IN MAP FOR STAB1
ENDB     DS    F                    NEND OF MOVING CHAIN
BEGIN    DS    F                    NSTART OF MOVING CHAIN
STAB1    DS    F                    MOVED BEAD COORDINATES OF
*                                   SECOND BEAD IN
*                                   CRANKSHAFT BEAD MOVEMENTS
ACOM     DC    A(N)
         DC    A(NSTART)
         DC    A(NEND)
```

```
MEMBER NAME  CYCLE
         DC      A(MAP)                                          00006990
         COM                                                     00007000
N        DS      F                                               00007010
IR       DS      F                                               00007020
IFAN     DS      F                                               00007030
IXS      DS      F                                               00007040
IYS      DS      F                                               00007050
IZS      DS      F                                               00007060
IN       DS      F                                               00007070
ICHAIN   DS      F                                               00007080
         DS      11F                                             00007090
KCHAIN   DS      F                                               00007100
NCHAIN   DS      4F                                              00007110
NEVT     DS      F                                               00007120
         DS      3F                                              00007130
TABLE    DS      32F                                             00007140
IMOVE    DS      150F                                            00007150
         DS      150F                                            00007160
NTAB     DS      15000F                                          00007170
MAP      DS      4095F                                           00007180
NSTART   DS      15000F                                          00007190
NEND     DS      15000F                                          00007200
         END     CYCLE                                           00007210
```

```
MEMBER NAME  CTMAP
CTMAP    CSECT
         SAVE   (14,12)
         BALR   9,0
         USING  *,9
         L      12,ACOM
         LA     11,4095(12,0)
         LA     11,1(11)
         LA     6,4095(11,0)
         LA     6,1(6)
         USING  N,12,11,6
         ST     13,SAV+4
         LA     13,SAV
* THIS SUBPROGRAM COUNTS THE 1 BITS IN MAP AND STORES THE SUM IN IPRINT
         L      4,=F'4'
         L      5,=F'16380'
         SLR    7,7
         SLR    8,8
         L      10,ACOM+4
LOOP1    L      3,0(8,10)
L2       C      3,=F'0'
         BZ     L1
         A      7,=F'1'
         LR     2,3
         S      2,=F'1'
         NR     3,2
         B      L2
L1       BXLE   8,4,LOOP1
         ST     7,IPRINT
         C      7,N
         BE     OK
         L      15,=V(PRINT)
         BALR   14,15
         L      15,=V(PRTMAP)
         BALR   14,15
OK       L      13,SAV+4
         LM     14,15,12(13)
         LM     1,12,24(13)
         L      15,=V(PRTMAP)
         MVI    12(13),X'FF'
         BR     14
SAV      DS     18F
ACOM     DC     A(N)
         DC     A(MAP)
         COM
N        DS     F
         DS     25F
IPRINT   DS     F
         DS     15333F
MAP      DS     4096F
         END    CTMAP
```

```
MEMBER NAME  CHMAP
CHMAP    CSECT
         SAVE  (14,12)
         BALR  9,0
         USING *,9
         L     12,ACOM
         LA    8,4095(12,0)
         LA    8,1(8)
         LA    6,4095(8,0)
         LA    6,1(6)
         USING N,12,8,6
         ST    13,SAV+4
         LA    13,SAV
* THIS SUBPROGRAM CHECKS TO SEE IF THE MAP CONFIGURATION MATCHES NTAB
         L     15,ACOM+4
         SLR   1,1
         L     11,N
         L     10,=F'31'
L1       SLR   5,5
         L     4,NTAB(1)
         SRDL  4,6
         SRL   4,1
         SRDL  4,9
         SRL   5,17
         CR    4,10
         BNH   L2
         S     4,=F'32'
         D     5,TWO7
L2       LR    3,5
         M     2,=F'4'
         MH    4,=H'4'
         L     7,TABLE(4)
         N     7,0(3,15)
         BNZ   L3
         L     15,=V(PRINT)
         BALR  14,15
         L     15,=V(PRTMAP)
         BALR  14,15
         B     NO
L3       A     1,=F'4'
         BCT   11,L1
NO       L     13,SAV+4
         LM    14,15,12(13)
         LM    1,12,24(13)
         MVI   12(13),X'FF'
         BR    14
SAV      DS    18F
TWO7     DC    X'00000040'
ACOM     DC    A(N)
         DC    A(MAP)
         COM
N        DS    F
         DS    27F
TABLE    DS    32F
         DS    300F
NTAB     DS    15000F
MAP      DS     4096F
         END   CHMAP
```

```
MEMBER NAME  PACK
PACK      CSECT
*
*   THIS SUBPROGRAM PACKS THE X,Y, AND Z COORDINATES OF ONE BEAD
*   INTO NTAB.
*   NTAB(I) =(LTAB(1,I))(2**16)+(LTAB(2,K))(2**8)+(LTAB(3,I))
*
          SAVE   (14,12)
          BALR   9,0                USE R9 IN LOCAL ADDRESSING
          USING  *,9
          L      12,ACOM            R12 = ADDRESS OF BLANK COMMON
          USING  N,12
           L      2,4(1)            R2 = ADDRESS OF CHAIN NO.
*                                   ARGUMENT PASSED FROM MAIN
           L       3,0(2)           R3 = CHAIN NO. FROM 1 TO N
          S      3,=F'1'            R3 = ADDRESS IN NTAB
          SLL    3,2                OF MOVING BEAD
           L      2,0(1)            R2 = ADDRESS OF LTAB
*                                   ARGUMENT PASSED FROM MAIN
          L      4,0(2)             R4 = LTAB
          L      5,4(2)             R5=Y(I)
          L      6,8(2)             R6=Z(I)
          SRDL   6,8                Z IN TOP OF R7
          LR     6,5
          SRDL   6,8                Y,Z IN TOP OF R7
          LR     6,4
          SRDL   6,8                X,Y,Z IN TOP OF R7
          SRL    7,8                X,Y,Z IN BOTTOM OF R7
          ST     7,NTAB(3)
          LM     14,15,12(13)
          LM     1,12,24(13)
          MVI    12(13),X'FF'
          BR     14
*
*
          ENTRY UNPACK
*   UNPACK CALCULATES THE SUM OF X,Y, AND Z COORDINATES
*   AS WELL AS VECTOR END TO END LENGTH AND THE NUMBER
*   OF TIMES THE CHAIN IS SPLIT.  THESE ARE USED IN
*    SUBROUTINE SAMPLE
*
UNPACK    SAVE   (14,12)
          USING  UNPACK,15
          L      12,ACOM
          USING  N,12
           L      2,4(1)            R2 = ADDRESS OF N2
*                                   PASSED ARGUMENT FROM SAMPLE
           L       3,0(2)           R3 = N2
          S      3,=F'1'            N2 = (N2 -1)*4
          SLL    3,2                N2 IS THE ADDRESS IN NTAB OF
          ST     3,N2               THE LAST BEAD IN THE CHAIN
           L      2,0(1)            R2 = ADDRESS OF N1,PASSED FROM SAMPLE
          L      2,0(2)             R2 = N1
          S      2,=F'1'            N1 = (N1-1) *4
          SLL    2,2
          ST     2,N1               N1 IS THE ADDRESS IN NTAB OF THE
*                                   FIRST BEAD IN THE CHAIN
*              FIND OUT HOW MANY TIMES THE CHAIN IS SPLIT
*
```

```
MEMBER NAME  PACK
*  TO FIND OUT HOW MANY TIMES THE CHAIN IS SPLIT LOOK
*   AT SUCCESSIVE PAIRS OF BEADS TO SEE IF THEY ARE MORE
* THAN ONE LATTICE SITE APART.  THESE PAIRS ARE LABELED
* _X1Y1Z1 AND _X2Y2Z2 IN THE COMMENTS.
LOOP     LA    10,8                 R10 = 8
         LA    3,3                  R3 = 3
         L     4,NTAB(2)            R4 = NTAB OF FIRST BEAD = _X1Y1Z1
         L     6,NTAB+4(2)          R6=NTAB OF SECOND BEAD = _X2Y2Z2
C        SRDL  4,8                  R4 = _ _X1Y1 R5=Z1_ _ _
         SRDL  6,8                  R6 = _ _ X2Y2  R7 = Z2_ _ _ _
         SRL   7,24                 R7 = _ _ _ Z2
         SRL   5,24                 R5 = _ _ _ Z1
         SR    7,5                  R7 = Z2 - Z1
         C     7,=F'1'              IF Z2 - Z1 = 1 THEN
         BE    WHOLE                THE CHAIN IS NOT SPLIT
         C     7,=F'-1'             IF Z2 - Z1 = -1 THEN
         BE    WHOLE                THE CHAIN IS NOT SPLIT
         C     7,=F'0'              IF Z2 - Z1 = 0 THEN
         BE    WHOLE                THE CHAIN IS NCT SPLIT
         L     9,ISEP(10)           FOR X,Y, OR Z SET
         LA    9,1(9)               ISEP = ISEP +1
         ST    9,ISEP(10)
WHOLE    S     10,=F'4'
         BCT   3,C                  CHECK Y, THEN X
         LA    2,4(2)
         C     2,N2                 IF ALL BEADS HAVE BEEN
         BNE   LOOP                 CHECKED LEAVE LOOP
*
*       CALCULATE CENTER OF MASS OF CHAIN
*
* FIRST BEAD IN CHAIN DOSEN'T NEED RECONNECTING
         L     2,N1                 R2 = N1
         L     4,NTAB(2)            R4 = _X1Y1Z1, NTAB OF FIRST BEAD
         SRDL  4,8                  R4 = _ _ X1Y1   R5 = Z1_ _ _
         SRL   5,24                 R5 = _ _ _ Z1
         ST    5,LAST+8             LAST+8 = Z1
         ST    5,ISUM+8             ISUM+8 = Z1
         ST    5,FIRST+8            FIRST + 8 = Z1
         SRDL  4,8                  R4 = _ _ _ X1 R5 = Y1 _ _ _
         SRL   5,24                 R5 = _ _ _ Y1
         ST    5,LAST+4             LAST + 4 = Y1
         ST    5,ISUM+4             ISUM +4 = Y1
         ST    5,FIRST+4            FIRST +4 = Y1
         SRDL  4,8                  R4 = 0   R5 = X1 _ _ _
         SRL   5,24                 R5 = _ _ _ X1
         ST    5,LAST               LAST = X1
         ST    5,ISUM               ISUM = X1
         ST    5,FIRST              FIRST = X1
*  NOW DO REST OF CHAIN.  ADD COORD INTO ISUM FOR CENTER
*            OF MASS CALCULATIONS
*
*  BEFORE ADDING X,Y, OR Z COORDINATES OF CURRENT BEAD,
*  GIVEN BY _X2Y2Z2, CHECK TO SEE IF IT IS MORE THAN
*  ONE LATTICE SITE AWAY FROM THE LAST BEAD (AFTER
*  THE LAST BEAD WAS RECONNECTED IF CHAIN WAS SPLIT).
*  THE RECONNECTED COORDINATES OF THE PREVIOUS BEAD
*  ARE GIVEN BY _XLYLZL IN THE COMMENTS.
*
```

```
MEMBER NAME  PACK
         L      6,N1
LOOP1    LA     6,4(6)
         LA     10,3              R10 = 3
         L      4,NTAB(6)         R4 = _X2Y2Z2
LLL      SRDL   4,8               R4 = _ _X2Y2  R5 = Z2 _ _ _
         SRL    5,24              R5 = _ _ _ Z2
         LR     11,10
         S      11,=F'1'
         SLL    11,2              R11 = INDEX REGISTER
         L      7,LAST(11)        R7 = XL,YL OR ZL
         SR     7,5               R7 = XL-X1, YL-YI OR ZL-Z1
         C      7,=F'1'           IF R7 = 1 THEN CHAIN NOT SPLIT
         BE     WHOLE1
         C      7,=F'0'           IF R7 = 0 CHAIN IS NOT SPLIT
         BE     WHOLE1
         C      7,=F'-1'          IF R7 = -1 THEN CHAIN NOT SPLIT
         BE     WHOLE1
         BL     SUBT
*    SUBTRACT A BOX DIMENSION(RECONNECT THE CHAIN)
         A      5,IXS(11)         X2 = X2 +IXS,Y2=Y+IYS,OR Z2=Z2+IZS
         B      WHOLE1
*    ADD A BOX DIMENSION
SUBT     S      5,IXS(11)         X2=X2-IXS,Y2=Y2-IYS, OR Z2=Z2-IZS
WHOLE1   ST     5,LAST(11)        LAST = X2,Y2, OR Z2
         A      5,ISUM(11)        ISUM=ISUM+X2,ISUM+4 = ISUM+4+Y2
*                                 OR ISUM+8 = ISUM+8+Z2
         ST     5,ISUM(11)
         BCT    10,LLL
         C      6,N2
         BNE    LOOP1
*
*    COMPUTE VECTOR END TO END LENGTH
*
         LA     3,3
         XR     10,10
LOOP2    L      4,LAST(10)        R4 = _ XLYLZL OF LAST BEAD
         L      5,FIRST(10)       R5 = XFYFZF OF FIRST BEAD
         SR     4,5               R4 = XLYLZL - XFYFZF
         ST     4,IS(10)          IS = END TO END LENGTH
         LA     10,4(10)
         BCT    3,LOOP2
         LM     14,12,12(13)
         MVI    12(13),X'FF'
         BR     14
N1       DS     F                 ADDRESS IN NTAB OF LAST BEAD IN CHAIN
N2       DS     F                 ADDRESS IN NTAB OF LAST BEAD IN CHAIN
LAST     DS     3F                BEAD COORD OF BEAD AFTER RECONNECTION
FIRST    DS     3F                BEAD COORD. OF FIRST BEAD IN CHAIN
ACOM     DC     A(N)
         COM
N        DS     F
         DS     2F
IXS      DS     3F
         DS     2F
ISUM     DS     3F                SUM OF RECONNECTED BEAD COORDINATES
IS       DS     3F                END TO END LENGTH
ISEP     DS     3F
         DS     343F
```

References

1. P. E. Rouse, Jr., J. Chem. Phys., 21,1272(1953).

2. L. R. G. Treloar, The Physics of Rubber Elasticity, 2nd ed., Oxford University Press, 1967.

3. B. H. Zimm, Jr., J. Chem. Phys., 24, 269(1956).

4. S. Chandrasekhar, Revs.Modern Phys., 15,1(1943).

5. Grant R. Fowles, Analytical Mechanics, 2nd ed. New York, 1962.

6. Katchalsky and Curran, Nonequilibrium Thermodynamics in Biophysics, Harvard University Press, 1965.

7. M. Fixman, J. Chem. Phys., 42,3831(1965).

8. Bird, Hassager, Armstring, Curtiss, Dynamics of Polymer Liquids, Vol. 2. John Wiley and Sons, 1977.

9. Hildebrand, Methods of Applied Mathematics, 2nd ed., Prentice - Hall, 1965.

10. F. Beuche, J. Chem. Phys., 22,603(1954).

11. P. Verdier, J. Chem. Phys., 4,204(1969).

12. P. H. Verdier, J. Chem. Phys., 45,2122(1966).

13. D. E. Kranbuehl and P. H. Verdier, J. Chem. Phys., 31,45(1972).

14. R. A. Orwoll and W. H. Stockmayer, Advan.Chem.Phys., 15,305(1969).

15. K. Iwata and M. Kurata, J. Chem. Phys., 50,4008(1969).

16. P. H. Verdier, J. Chem. Phys., 52,5512(1970).

17. D. E. Kranbuehl and P. H. Verdier, J. Chem. Phys., 71,2662,(1979).

18. L. Takacs, Stochastic Processes, Methuen and Co. , Ltd, London, 1960.

19. P. H. Verdier and W. H. Stockmayer, J. Chem. Phys., 36,227(1962).

20. Paul J. Flory, Principles of Polymer Chemistry, Cornell University Press, LTD., London, 1953.

21. M. Fixman and J. M. Peterson, J.Amer.Chem.Soc., 86,3524(1964).

22. Bishop, Ceperley, Frisch, Kalos. (In Print, J.CHem.Phys.)

23. F. T. Wall, J. C. Chin and F. Mandel, J.Chem.Phys., 66,3143(1977).

24. H. Okamoto, J.Chem.Phys., 70,1690(1979).

25. D. E. Kranbuehl and B. Schardt, ACS Symposium No. 86, Computer Modeling of Matter. 1978.

26. A. Bellemans and E. DeVos, Macromolecules, 7,809(1974).

27. A. Bellemans and E. DeVos, Macromolecules, 8,651(1975).

28. J. G. Curro, J.Chem.Phys., 61,1203(1974).

29. J. C. Curro, J.Chem.Phys., 64,2496(1976).

30. J. C. Curro, Macromolecules, 12,463(1979).

31. J. P. Cotton, D. Decker, H. Benoit, B. Farnoux, J. Higgins, G. Jannink, R. Ober, C. picot, and J. des Cloizeaux, Macromolecules, 7,863(1974).

32. D. W. Schaefer, J. F. Joanny and P. Pincus (Private Communication)

33. B. Schardt, Honors Thesis in Chemistry, College of William and Mary in Virginia, May 1977.

34. M. H. Cohen and D. Turnbull, J. Chem. Phys., 31, 5(19 59).

35. A. K. Doolittle, J.Appl.Phys., 22,1471(1951).

36. A. K. Doolittle, J.Appl.Phys., 23,236(1952).

37. P. H. Verdier, J. Chem. Phys., 52,11(1969).

38. L. G. Kelly, Handbook of Numerical Methods and Applications, Addison-Wesley Publishing Co., 1967.

39. Mary Boas, Mathematical Methods in the Physical Sciences, John Wiley & Sons Inc. New York, 1966.

40. P. H. Verdier, J. Chem. Phys., 45,2118(1967).

Vita

Rebecca Louise Moss Smithson

Born in Frederick, Maryland on December 31, 1953. Graduated from Governer Thomas Johnson High School in that city in June, 1971. Entered Hood College in Frederick, Maryland in September of 1971 and in May of 1975 received a B. A. in Chemistry and Mathematics. In September of 1975 the author entered a Post-Baccalaureate program in engineering at the University of VIrginia in Charlottesville and continued in this program until December of 1975. She then worked in the Polymer Section of the NASA Langley Research Center in Hampton Virginia from Janurary of 1976 until August of that year.

In September of 1976 the author entered the graduate program in Chemistry at the College of William and Mary. In September of 1978 she entered the program in Applied Science at William and Mary. She is currently a research asistant in the Programming Techniques Branch of the NASA Langley Research Center.