To the Graduate Council:

I am submitting herewith a thesis written by Thomas Donald Hahn entitled "Dynamic Monte Carlo Simulations of Isolated Polymer Chains Using Lattice Models: 1) Chain Terminally Attached to an Infinite Plane, 2) Shape Fluctuations." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Chemistry.

Jeffrey D. Kovac,
Major Professor

We have read this thesis and recommend its acceptance:

Accepted for Council:

Vice Provost and
Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the library shall make it available to borrowers under rules of the library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying for use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Thomas D. Hall

Date August 15, 1989

DYNAMIC MONTE CARLO SIMULATIONS OF ISOLATED POLYMER CHAINS USING LATTICE MODELS: 1) CHAIN TERMINALLY ATTACHED TO AN INFINITE PLANE, 2) SHAPE FLUCTUATIONS

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Thomas Donald Hahn

December 1989

ACKNOWLEDGMENT

The author would like to express his gratitude to Dr. J. Kovac for his support and understanding during his studies at the University of Tennessee.

A great debt is owed to Mr. Don Broach, without whose assistance with the VAX and IBM computer systems, this research would not have succeeded.

The author must also mention his friends E. Wheeler Conover, Jerome G. May, and Gary M. Finniss whose companionship made his brief stay at UT bearable.

Dedication of the author's thesis work is to one person, Nancy Jane Becker, who patiently waited in Philadelphia during their two year separation.

ABSTRACT

Conformational and dynamic properties of isolated polymer chains were studied under two distinct conditions. Under one condition, conformational changes were studied for a chain terminally attached to a flat, impenetrable surface. The second condition looked at a free chain viewed as ellipsoidal in shape. Conformational fluctuations of the ellipsoid were studied by separating the motions into three rigid rotations (one for each axis) and three dilation and contraction motions (one for each axis).

Both studies were Monte Carlo computer simulations of polymer chains with different molecular weights, by using either a simple cubic or face centered cubic lattice model. In both cases, the polymer chains were assumed to be a collection of connected statistical segments called beads, each representing several repeat units.

The terminally attached chains were found to be somewhat enlarged if the beads were assumed to occupy space (like hard spheres) and they were found to have significantly lower mobility than corresponding free

chains. However, both size and mobility of terminally attached chains did seem to have the same molecular weight dependence as corresponding free chains.

Using the ellipsoidal model for free chains showed the chain to be highly anisotropic and that the order of increasing size was the order of decreasing mobility with respect to dilations and contractions, for each of the axes of the ellipsoid. These dilation and contraction motions were shown to be closely related to the first three normal modes for free chains.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

CHAPTER I

LATTICE SIMULATIONS OF POLYMER CHAIN DIMENSION AND MOBILITY: MOTIVATION AND DESCRIPTIVE FUNCTIONS

A. Motivation

Computer simulations of polymer chains using lattice models serve two basic functions. They model the dimension and mobility of polymer chains. The quantities used to monitor the dimension and motion of the chain depend, in part, on the interests of the researcher. In Chapter II of this thesis, dimension is measured by the mean square end-to-end distance $\langle R^2 \rangle$, and mobility is measured by an autocorrelation function of the time-dependent end-to-end vector $\mathbf{R}(t)$ for a polymer chain terminally attached to an infinite, impenetrable surface. The autocorrelation function allows us to calculate a characteristic terminal relaxation time of the chain. In Chapter III, we study the fluctuations in the instantaneous shape of an isolated polymer chain by considering it to be

ellipsoidal (Figure 1) in shape. Dimension is measured by the mean square extension of the polymer chain ($\langle L_1 \rangle$, $\langle L_2 \rangle$, and $\langle L_3 \rangle$) along the three principal axes of the ellipsoid. Conformational fluctuations of the ellipsoid are studied by separating the motions into three rigid rotations (one for each axis) and three dilation and contraction motions (one for each axis). They are measured in terms of autocorrelations of each $\langle L_i \rangle$ (expansion and contraction) and of appropriate functions of the angles which principal axes make with respect to a fixed laboratory coordinate system. The functions probe the two kinds or relaxation processes of a non-rigid ellipsoid, changes in extension along the principal axes and rigid body rotation. These functions will be discussed in greater detail in section B of this chapter.

This thesis work has some short term motivations as well as some long term goals for future work. The intent of Chapter II is to study the physical properties of a polymer chain terminally attached to an impenetrable, flat surface. The size, measured as $\langle R^2 \rangle$, has been calculated by Monte Carlo simulations for free chains in lattice systems.[1] These values can be

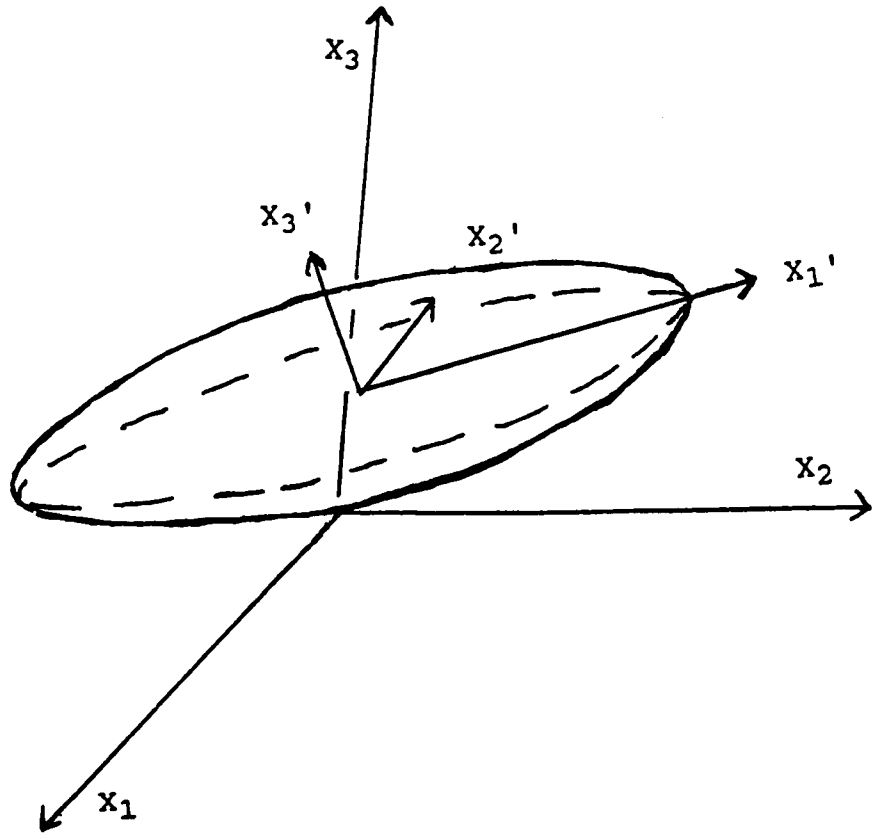

Figure 1. An ellipsoid and its three principal axes (x_1',x_2',x_3').

compared to $\langle R^2 \rangle$ measured in Chapter II to see the effects of the surface on the size of the chain. Likewise, relaxation times for R(t) for free chains have been previously published. These values can be compared to T_R values listed in Chapter II in order to see how the surface affects the mobility of the polymer chain. This is, however, only a starting point with many possible routes of investigation.

One route is to extend the investigation to systems with increasing surface coverage. By adding more chains to the simulation and changing the distances between adsorption sites, the effect of surface coverage can be studied. This can be extended even further by looking at two parallel surfaces with polymer chains attached to the two opposing surfaces. Properties such as extent of interpenetration, average chain size, and mobility can be measured as a function of the distance between surfaces. These studies are of considerable relevance to problems such as adhesion and colloid stabilization.

The study of the end adsorbed chain can be extended in another direction to study some aspects of polymer crystallization. Growing polymer crystals requires chain folding. The size of the loops between folds as

well as where a chain reenters a crystal are matters of some controversy. The question is whether a chain will fold back on itself and continue growing at an adjacent site along the crystal (adjacent reentry) or whether other polymer chains will interfere and crystallize along sites adjacent to the chain in question (switch board). This could be studied by monitoring a chain tail extending from the edge or corner of an impenetrable surface. Comparing the extent to which the tail comes in contact with that surface at sites adjacent to the edge or corner with the extent to which other chains adsorb at those sites can give some indication of the tendency for adjacent reentry in preference to switch board.

In Chapter III, the interest is focused on shape fluctuations. The mean squared extensions along the principal axes can be sorted by size ($\langle L_1 \rangle \geq \langle L_2 \rangle \geq \langle L_3 \rangle$) and averaged over time. They show that the polymer chain is far from spherical in shape at any moment in time. Each of the $\langle L_i \rangle$ can be studied for chain length dependence of both size and mobility. Rotational relaxation times can also be determined from the autocorrelations of an angular function (Y_i^2) described

in Chapter III. The main focus of the chapter is the relationship between these properties and how they relate to measurements reported in other publications. Specifically, it is possible that the three $<L_i>$ relaxation times can be related to the relaxation times of the first three normal modes by a simple relationship. As will be seen in Chapter III, a simple relation does seem to exist.

Shape and orientation of a polymer chain could have a significant effect on properties which are size dependent. For instance, the viscosity of a polymer chain which has its long axis oriented perpendicular to the flow direction is greater than the viscosity of a chain whose long axis is oriented in the flow direction. This suggests that the rate of reorientation as well as the tendency to orient in a particular direction will have a profound effect on the overall viscosity.

Likewise, separation techniques which depend on size of polymer chains, will also depend to some extent on the axial orientation. A popular technique used by polymer scientists is size exclusion chromatography (SEC), which is also known as gel permeation chromatography (GPC). SEC is a technique used to

separate large molecules, namely polymers, by size. The equipment consists of a series of columns containing beads of crosslinked polystyrene or silica. Each of these columns will have cavities with a distribution of sizes. As the polymer chains pass through the column they only enter the cavities large enough to accommodate them. All polymer samples have some molecular weight distribution. SEC is used to separate polymer samples by molecular size, which is related to molecular weight. If a chain is larger than the largest cavity size, it will pass right through the column at the rate of elution. If the chain is small enough to enter the smallest cavity, it will enter every cavity it approaches. In other words, the polymer molecular weight of the eluent will decrease with increasing elution time. Once again, the orientation of the polymer chain will also affect its ability to flow through the column because orientation affects its ability to enter the cavities. In this type of separation, if the elution time is comparable to the relaxation time for reorientation or shape relaxation, shape and orientation effects will be observable in measurements of elution times.

Shape studies can be extended to include concentrated systems. The influence of other chains on chain orientation and shape is of some importance. The orientation of chains in bulk systems will influence the systems' overall physical properties. The overall shape can be related to the average separation between chains.

Shape studies can also be extended to other types of polymer chains, namely nonlinear polymer chains. Presence and location of branches will influence the overall shape of the chain. In nonlinear polymer systems, the shape fluctuations may be the best way to study internal molecular motions.

B. Descriptive Functions

The size of a polymer chain is normally calculated as its characteristic mean square end-to-end distance $\langle R^2 \rangle$ or mean square radius of gyration $\langle S^2 \rangle$. For a random coil, the two are approximately related by the simple relationship

$$\langle S^2 \rangle = \langle R^2 \rangle / 6. \qquad (1)$$

Both are expected to scale as

$$\langle R^2 \rangle \sim (N-1)^{2\nu} \qquad (2)$$

where (N-1) is the number of bonds joining N beads. A bead is a statistical segment containing several repeat units whose size depends on chain stiffness. The scaling exponent (2ν) is expected to have the value 1.0 for a chain with no excluded volume and 6/5 for a chain with excluded volume, also called a self avoiding walk. For the terminally attached chain studied in Chapter II, $\langle R^2 \rangle$ is calculated, giving some indication of the effect of the barrier on the overall chain size.

In chapter III, the mean square extension along each of the three principal axes are calculated and they are related to $\langle S^2 \rangle$ by

$$\langle S^2 \rangle = \langle L_1 \rangle + \langle L_2 \rangle + \langle L_3 \rangle. \qquad (3)$$

Each $\langle L_i \rangle$ can be determined by diagonalizing the "radius of gyration" matrix $\underline{X}$ originally defined by Solc,[2] giving the eigenvalue matrix $\underline{L}$. Each element of the matrix $\underline{X}$, X_{ij}, is defined as

$$X_{ij} = (1/N)\sum_{k} X_i^k X_j^k - (1/N)^2 \sum_{k} X_i^k \sum_{k} X_j^k \qquad (4)$$

with i and j representing any of the three fixed axes and index k representing the kth unit of the chain. Each $\langle L_i \rangle$ is a diagonal element in $\underline{L}$ and, therefore, an eigenvalue of $\underline{X}$. These eigenvalues are a measure of the chain shape and its deviation from spherical symmetry.

As mentioned above, mobility of the polymer chains in this thesis is measured by the decay of various autocorrelation functions. The autocorrelation function ($p_f(t)$) is defined as:

$$p_f(t) = (\langle f(t)f(0)\rangle - \langle f\rangle^2)/(\langle f^2\rangle - \langle f\rangle^2) \qquad (5)$$

in which f is the property being monitored, t is the time interval, and <> means the equilibrium average value. For terminally attached chains studied in Chapter II, f is the end-to-end vector **R**. In Chapter III, f is either an eigenvalue L_i, described above, or a rotational function Y_i^2 described in that chapter. As the time interval increases, the first term in the numerator of equation (5) decays to some limiting value

which, at infinite time, is $\langle f \rangle^2$. Thus, as the time interval increases, $p_f(t)$ decreases and approaches zero asymptotically at infinite time. This decay can often be related to a characteristic relaxation time (T_f).

Figures 2 through 9 are semilog plots of the autocorrelation functions versus time. As they show, the **R** and the $\langle L_i \rangle$ semilog plots are linear after an initial induction time, but the Y_i^2 semilog plots never become linear. This linear region implies that, at long times, the autocorrelation function is an exponential of the form $p_f(t) = A\exp(-t/T_f)$. T_f can be calculated by a linear least squares fit of the long time decay of $\ln p_f(t)$ versus t. The slope is equal to $(-1/T_f)$. For a nonlinear plot, a fairly simple relationship can be used to calculate a characteristic relaxation time. For the Y_i^2 autocorrelations, a T_f is calculated as the time at which

$$\ln p_f(t) = -1. \qquad (6)$$

These T_f values can give some indication of the overall chain mobility when different chains and different simulation conditions are compared.

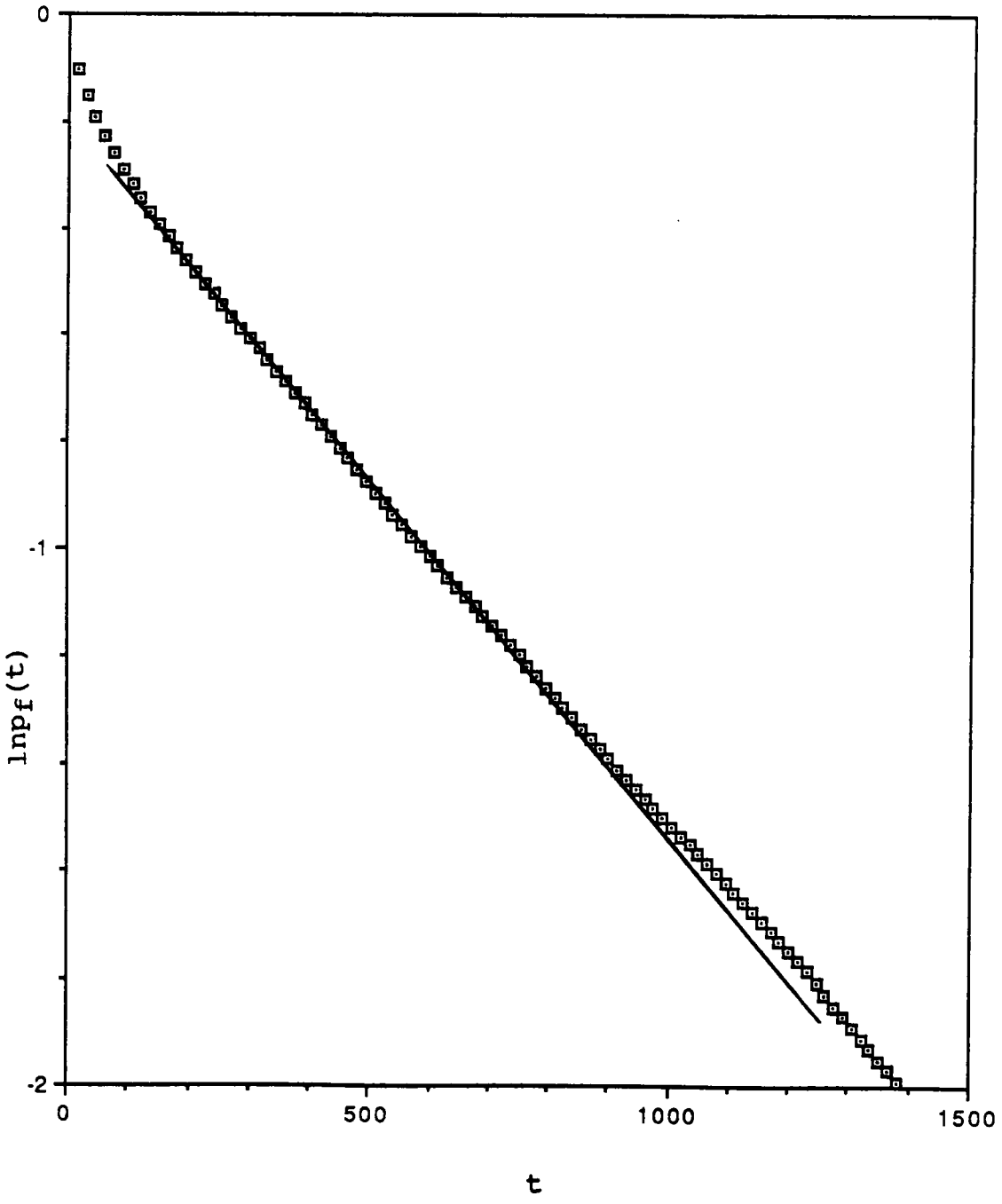

Figure 2. Sample semilog plot of $p_f(t)$ (f = R) versus time for a terminally attached 23 length chain with excluded volume.

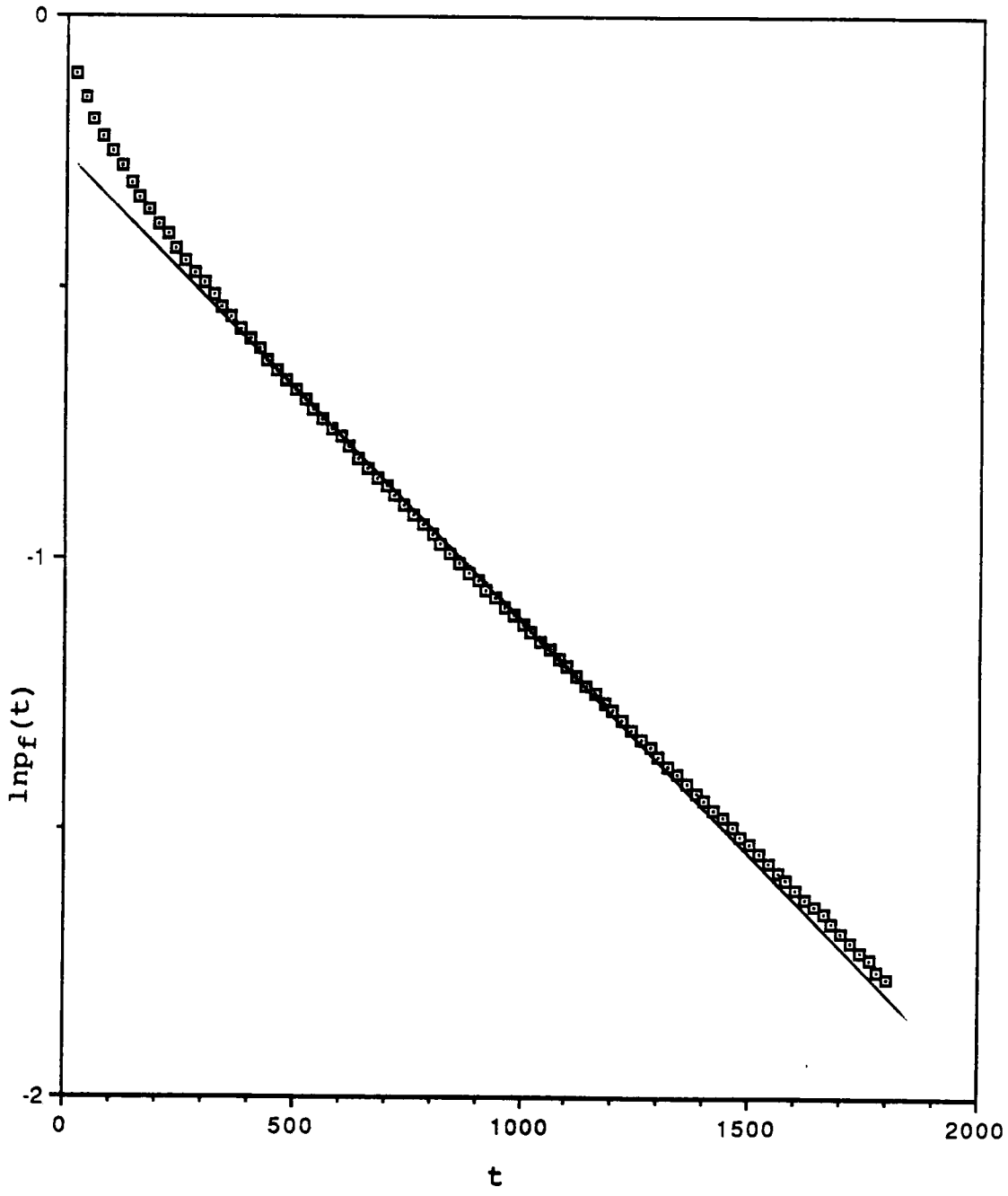

Figure 3. Sample semilog plot of $p_f(t)$ (f = R) versus time for a terminally attached 47 length chain with no excluded volume.

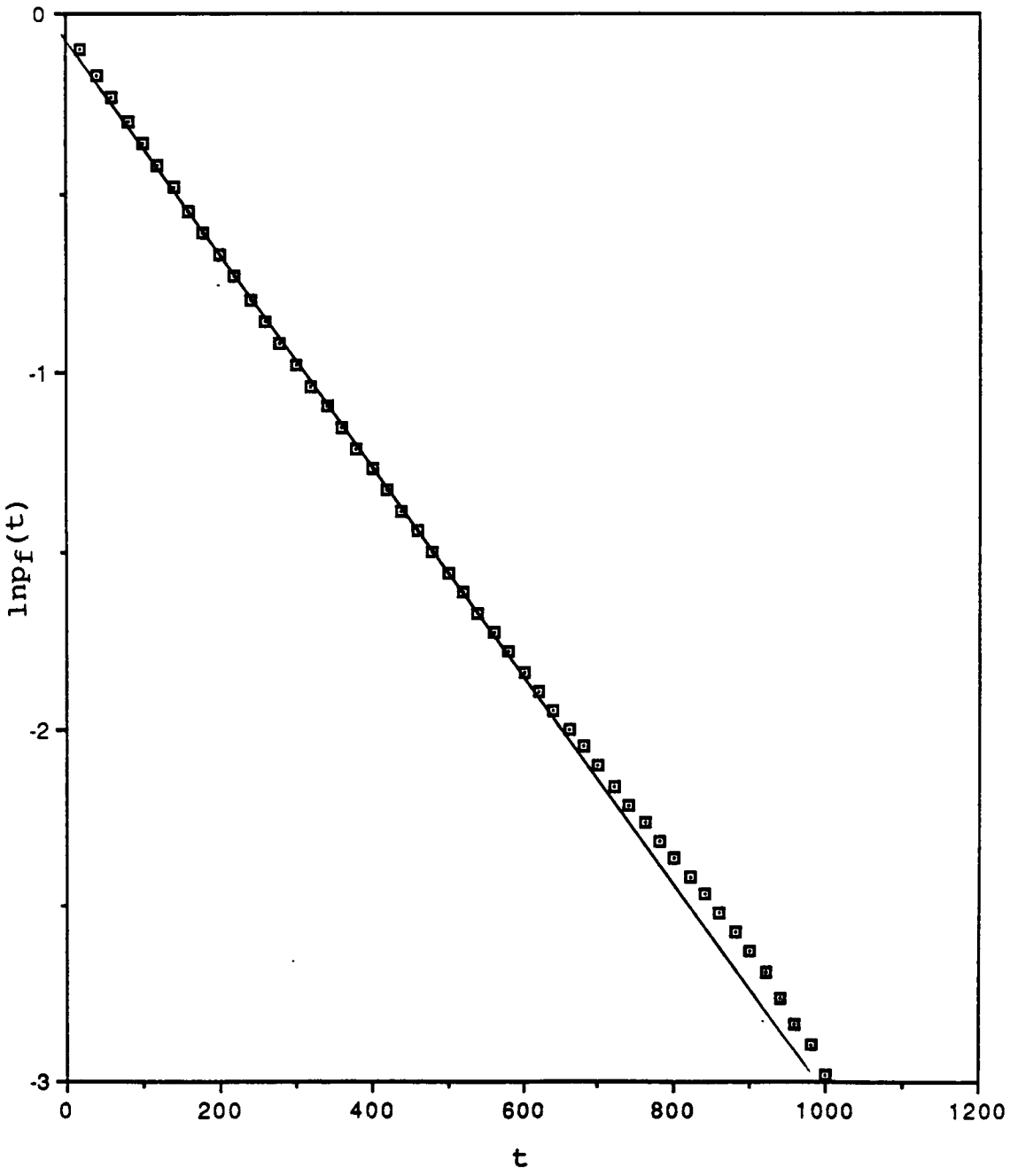

Figure 4. Sample semilog plot of $p_f(t)$ ($f = L_1$) versus time for a free 59 length chain with excluded volume.

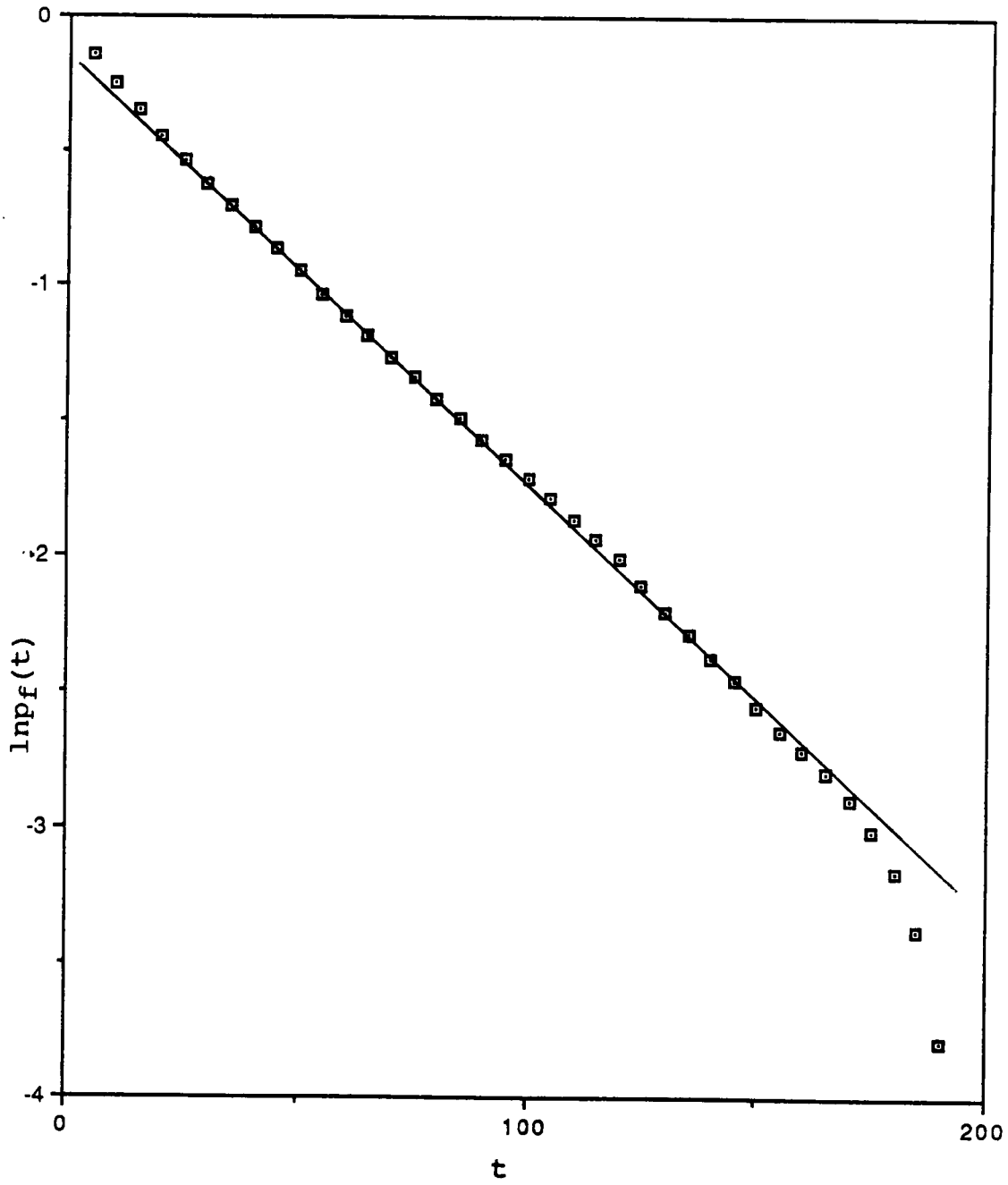

Figure 5. Sample semilog plot of $p_f(t)$ ($f = L_2$) versus time for a free 71 length chain with no excluded volume.

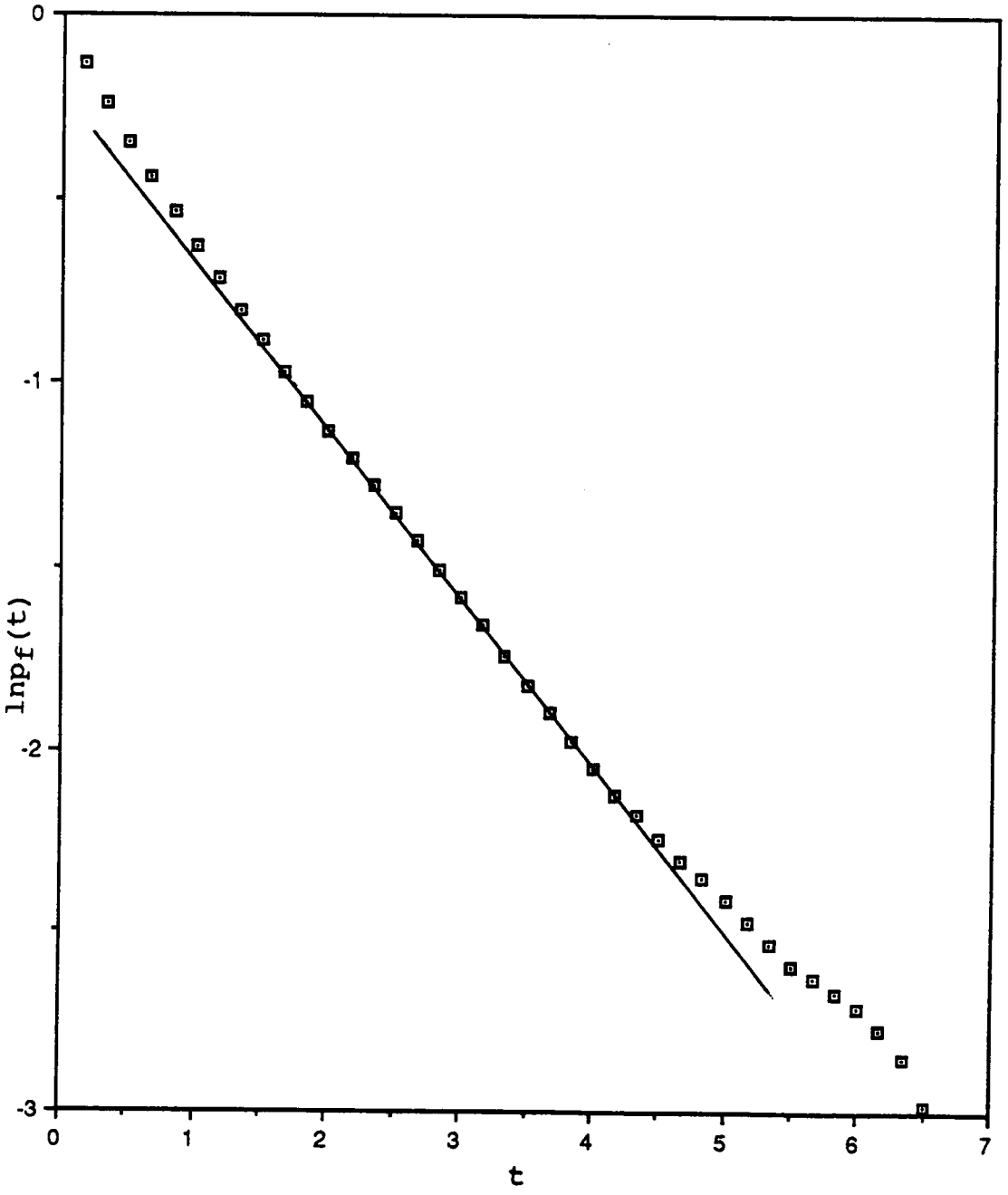

Figure 6. Sample semilog plot of $p_f(t)$ ($f = L_3$) versus time for a free 23 length chain with no excluded volume.

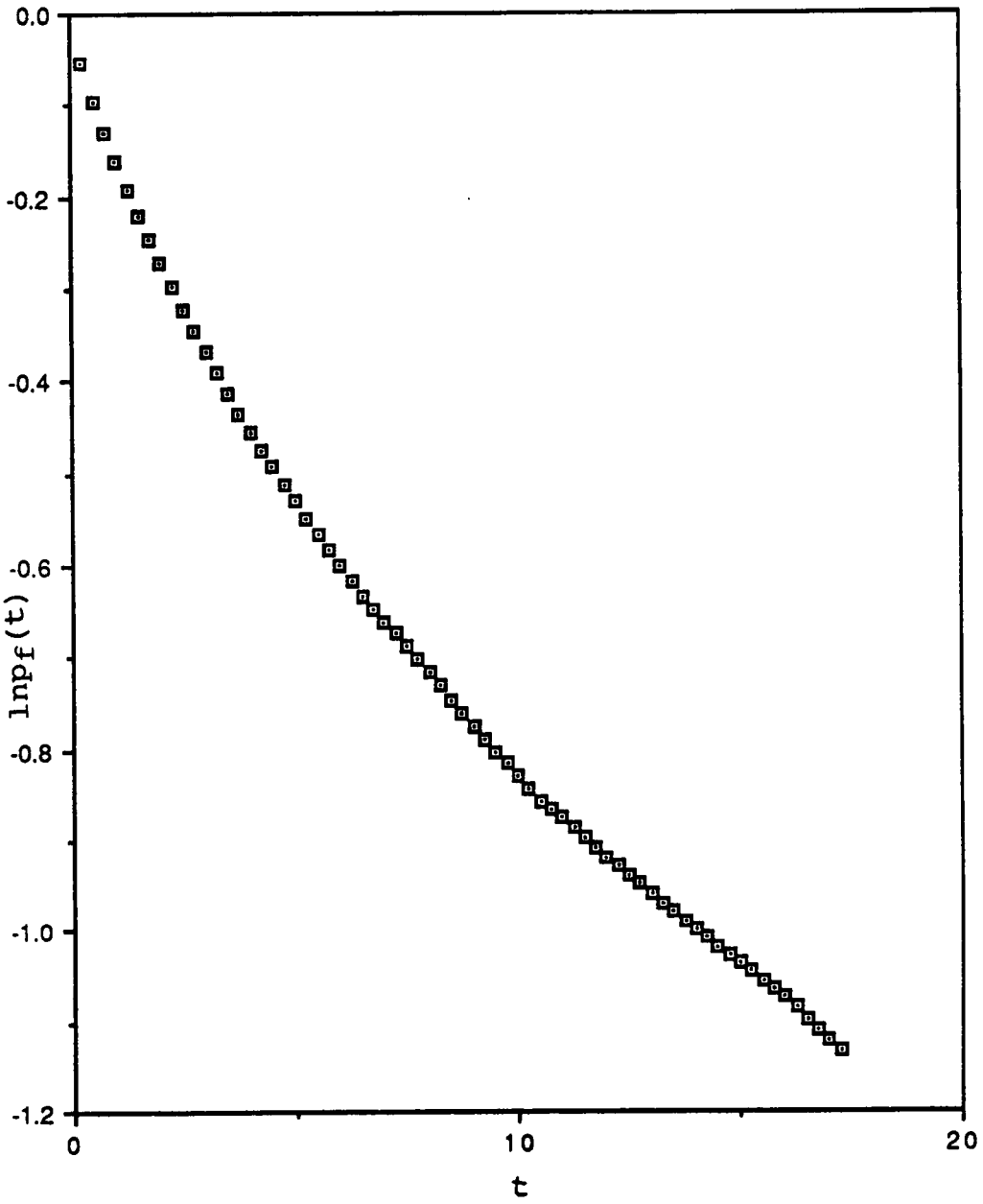

Figure 7. Sample semilog plot of $p_f(t)$ ($f = Y_1^2$) versus time for a free 47 length chain with no excluded volume.

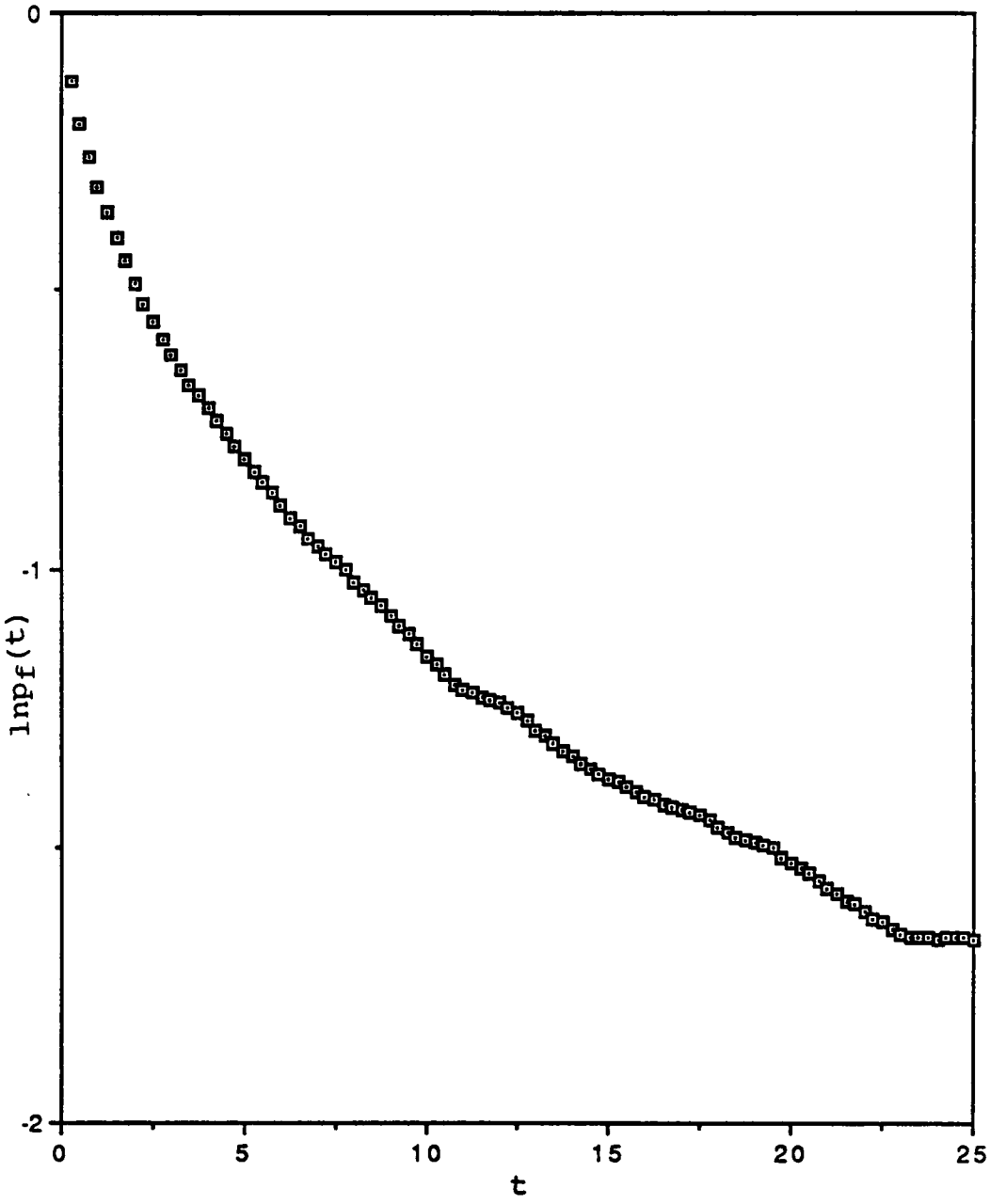

Figure 8. Sample semilog plot of $p_f(t)$ ($f = Y_2^2$) versus time for a free 47 length chain with no excluded volume.

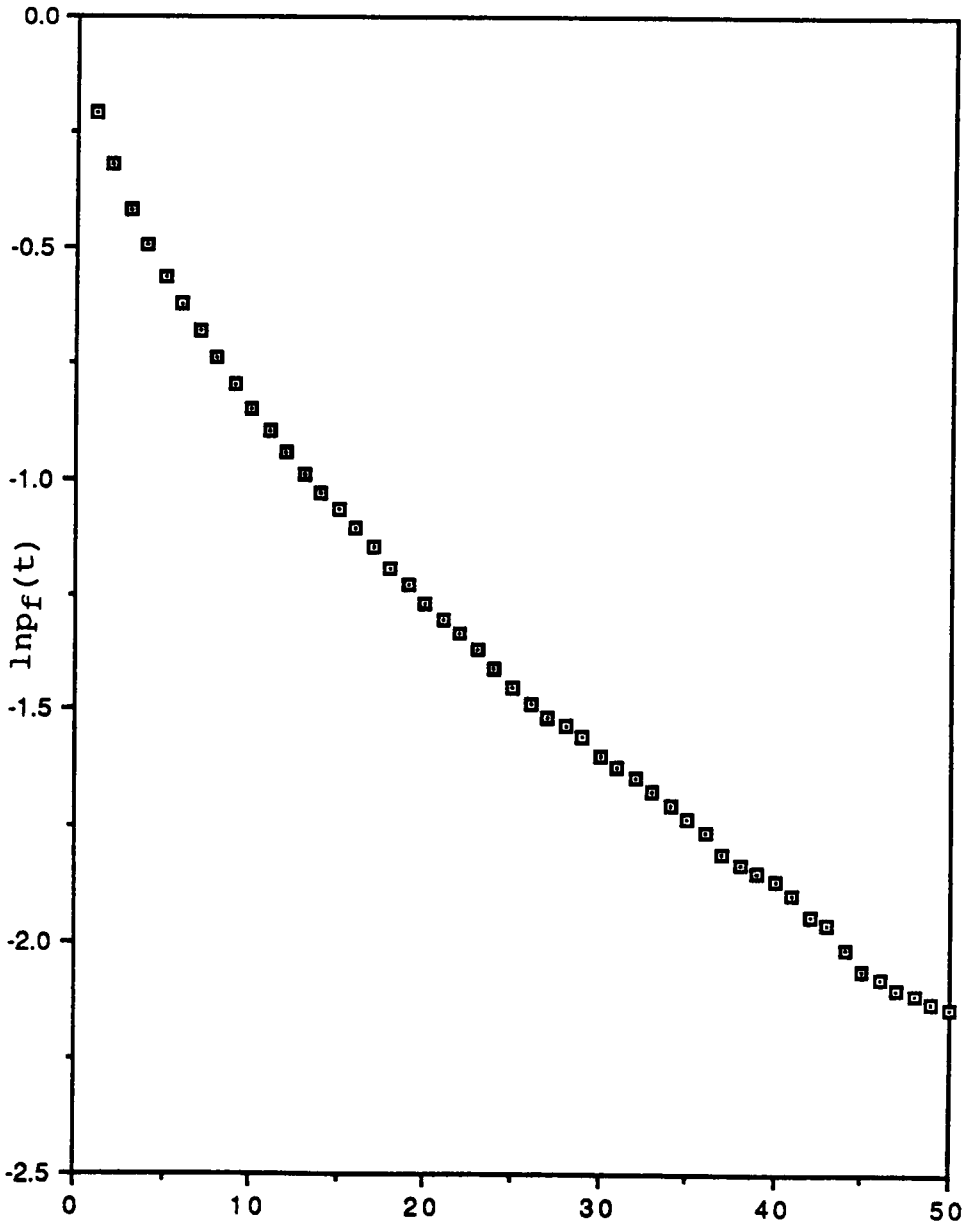

Figure 9. Sample semilog plot of $p_f(t)$ ($f = Y_3^2$) versus time for a free 35 length chain with no excluded volume.

CHAPTER II

SIMULATIONS OF SINGLE CHAINS TERMINALLY ADSORBED ON AN IMPENETRABLE FLAT SURFACE

A. A review of adsorption studies

The properties of polymer chains on surfaces make them useful for numerous applications, including adhesion, colloid stabilization for water purification and other uses, liquid and gas chromatography, and microencapsulation. Empirical studies can improve on these properties, but understanding the relationship between chain structure and properties will help to intelligently and efficiently manipulate the properties.

This chapter studies the nature of terminally adsorbed polymer chains. It focuses on isolated chains grafted to impenetrable surfaces. Section A reviews the literature and describes theories, simulations, and experiments used to study adsorbed chains. Part B explains the simulation model used. Section C reports the simulation model results and Part D summarizes the results and suggests further studies.

Several approaches, both experimental and theoretical, have been used in the past twenty years to study the properties of polymers adsorbed on a surface. This review is divided into two sections: theoretical calculations and experimental methods. Studies which can be best related to the work described in this thesis are emphasized.

Theoretical studies have looked at several properties of adsorbed chains, such as extent and effect of surface coverage, mean square end-to-end distance $\langle R^2 \rangle$ and mean square radius of gyration $\langle S^2 \rangle$, density profiles, and critical adsorption potentials. Prior to this thesis there have been no known simulations for dynamics of terminally adsorbed chains.

An isolated chain terminally adsorbed on a flat impenetrable surface in a good solvent (with excluded volume) is expected to occupy roughly a half sphere.[3] The radius of that sphere, known as the Flory radius, is

$$R_F = aN^{3/5} \tag{7}$$

with 3/5 being one half of the scaling exponent (2ν) defined in equation (2).

In 1974 Lax published several papers which calculated some static properties of terminally attached, self avoiding walk chains.[4,5] He calculated the scaling exponent and found reasonable agreement (2ν = 1.23) with Flory's prediction.[4] In another publication,[5] he attempted to determine the chain length dependence of the number of adsorbing units at different surface attraction potentials (E/kT). At very low potentials(E/kT<0.40) he found the extent of adsorption to be nearly independent of chain length, but when E/kT was above 0.4, the chain length dependence increased with increasing potential.

In 1978 Clark and Lal published a study of a Monte Carlo simulation on a tetrahedral lattice relating surface coverage to configurational properties of adsorbed chains.[6] They measured properties such as mean square end-to-end distance, $\langle R^2 \rangle$, and mean layer thickness, $\langle t \rangle$, as a function of the surface attraction potential and the surface coverage. For the athermal chain, they found that both $\langle R^2 \rangle$ and $\langle t \rangle$ changed little until the surface coverage reached about 1/4 of the squared maximum extension of the chain. Then $\langle R^2 \rangle$ and $\langle t \rangle$ increased dramatically, rapidly approaching the

fully extended configuration. When attractive potentials were included, $\langle R^2 \rangle$ and $\langle t \rangle$ started out smaller than in the athermal case at low surface coverage. As the surface coverage increased, $\langle R^2 \rangle$ reached a minimum before increasing again, but $\langle t \rangle$ behaved similarly to the athermal case.

In 1980, de Gennes calculated the concentration profile for isolated, terminally attached chains, as well as terminally attached overlapping coils.[7] For the isolated chains he found that the concentration (C_z) at a given distance (z) from the surface is given by

$$C_z = s(z/a)^{2/3} \tag{8}$$

where s is the fraction of grafted points. For overlapping coils, the distance between grafted sites is much less than the Flory radius ($D \ll R_F$). The chain is highly stretched perpendicular to the surface and the mean thickness is proportional to the chain length. Parallel to the surface plane, the chain is expected to be a random coil. The average spread parallel to the surface plane is found to be proportional to $N^{1/2}$.

Cosgrove did exact enumeration calculations of

terminally attached, self-avoiding walks on a tetrahedral lattice.[8] He calculated the fraction of polymer segments adsorbed as a function of attractive potential (E/kT). Like Lax, he found little chain length dependence for small potentials. He also found a critical potential at which adsorption becomes significant.

Experimental techniques such as ESR and NMR have been used to study both static[9-13] and dynamic[14,15] properties of end adsorbed polymer chains. The distribution of segments between low mobility regions near the surface and high mobility regions in tails can be determined because their signals can be separated. Temperature dependence of adsorption can then be monitored. Mobility of the chain can also be measured by ESR and NMR. The measured relaxation processes, however, are dominated by the fastest relaxations. In other words, the relatively slow relaxation of an entire chain will not be seen in the paramagnetic relaxation in ESR and NMR. Instead, these relaxations are likely to be dominated by short range motions along the chain, which do not exhibit molecular weight dependence.

The lack of literature on the overall mobility of a

terminally adsorbed polymer chain can make this thesis work of some interest. It can fill the gap by supplying information on the overall mobility of an end adsorbed chain. As mentioned in Chapter I, it can also be a starting point for studies of several chains terminally adsorbed on a surface, and simulations to study the folding behavior of polymer chains involved in crystallization. The procedure for the simulation and the results are described in what follows.

B. The Face Centered Cubic Lattice Model

In this section the specifics of the simulation method used are discussed. Since it is desirable to have a model in which the lattice will cause minimum hindrance to polymer chain mobility near the adsorption surface, the face centered cubic lattice (FCC) was chosen for the simulations. A justification for its preference over the simple cubic (SC) lattice as well as a description of the basic movement rules are discussed below. The programs were written in FORTRAN. The programs and the details of their usage are listed in Appendix I.

In any lattice model the polymer chain is represented as a series of connected units called beads. These beads represent statistical segments of a polymer chain, each representing several repeat units. The basic simulation involves a chain with one end fixed in a flat, infinite surface. If the plane of the wall can be represented by the Cartesian y-z plane (x=0) then the fixed end will be at the coordinates (x,y,z) = (0,v,w). In words, the end is fixed in the plane. No other units of the chain are permitted to be in this plane. So, the second member of the chain will be at (u'>0,v',w'). In simple cubic (SC) and face centered cubic (FCC) lattices these beads can be represented by simple integer coordinates with connected beads a unit distance apart for ease of representation in the program arrays. The advantage of a FCC lattice over a SC lattice for this particular investigation is the larger coordination number of the FCC lattice. In a SC lattice, the first unit being in the plane of the wall would freeze the second unit at the coordinate (1,v,w). However, in a FCC lattice there are three possible locations on one side of the wall which are a unit distance from the adsorbed end. The result is that no unit other than the

adsorbed end is frozen in place allowing for decreased end effects in the FCC lattice compared to the SC lattice. Smaller end effects are desirable because they allow relatively short chain lengths to be used in simulations to emulate the properties of high molecular weight polymers. Shorter chains means shorter simulation times and less memory requirements on the computer systems used.

The chain moves are executed as described in the following algorithm, which is the one developed by Downey, Crabb, and Kovac.[1] First, a bead is randomly chosen from the movable beads. The adsorbed bead is ignored. If the ungrafted end bead is chosen, the program randomly selects among the eleven other locations that are a unit distance from the second bead from the end. If the program uses excluded volume, a check is made to see if the proposed site is already occupied. If it is, the move is aborted. If no excluded volume is considered, all moves are allowed except those which would involve adsorption of the bead onto the wall.

If the bead chosen is an internal bead, the move is attempted as follows. First, the two beads adjacent to

the chosen one are determined. Second, all sites which are a unit distance from each of the two beads are determined. From among these sites, the ones that coincide are selected as possible locations for the chosen bead move. The current location of the bead is removed from the selection and the bead is randomly moved to one of the remaining sites. The number of choices varies depending on the bond angles (Figures 10 and 11, zero and 180 degrees not shown). This angle can be zero (only possible in the no excluded volume case), 60, 90, 120, or 180 degrees. In the zero degree case, there are eleven choices because the two adjacent beads occupy the same coordinates. In the 60 and 90 degree cases, there are four sites that coincide (three move choices). At 120 degrees, only one move site is available and at 180 degrees there are no move choices. If excluded volume is considered, the move attempt is checked for whether or not the site is occupied. If it is occupied the move is aborted. In the no excluded volume case, like for the end bead, all moves are allowed except those in which the bead would adsorb onto the wall.

The bead cycle is then complete, even if the move

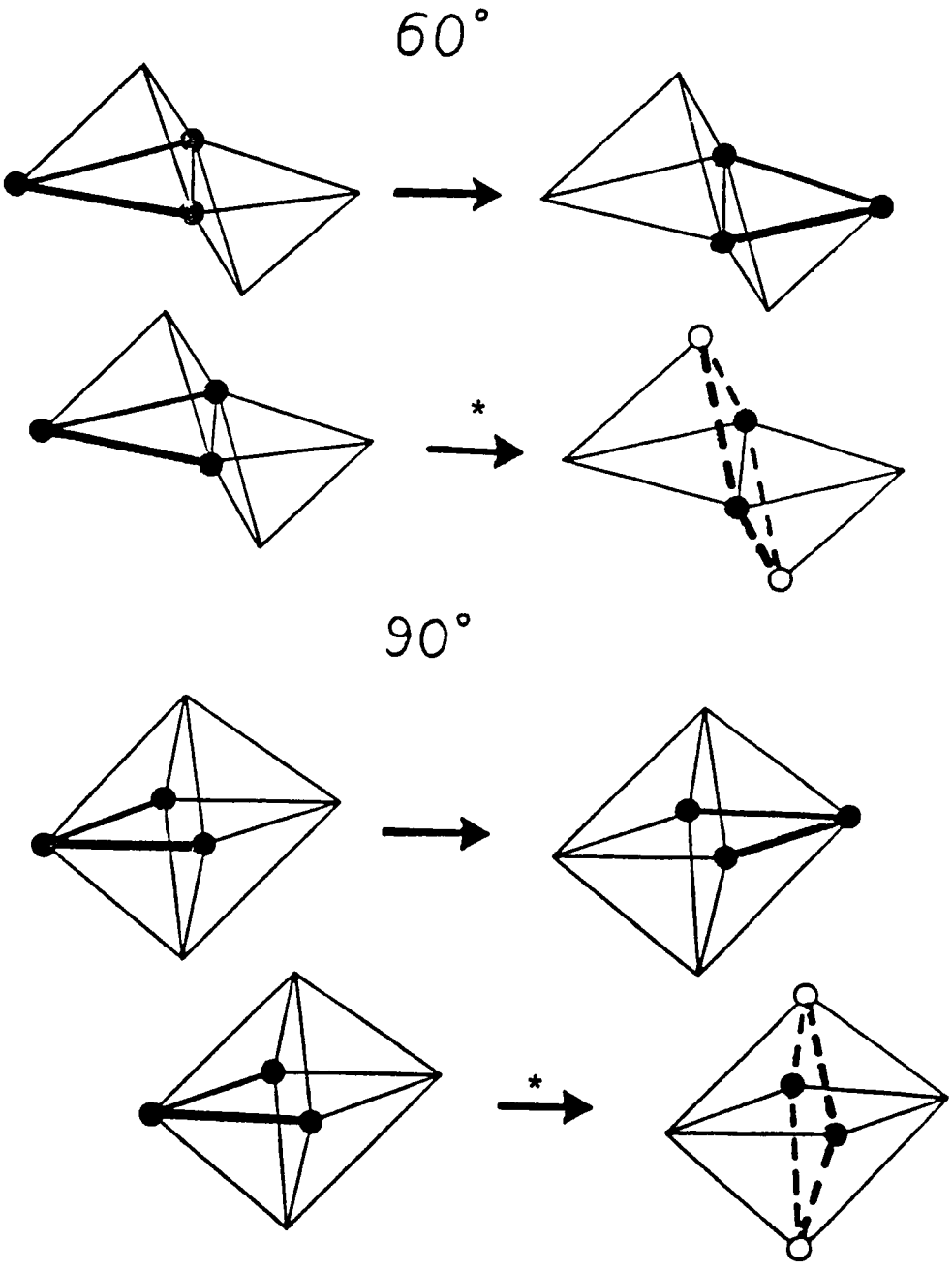

Figure 10. Move choices for 60 and 90° bonds. * denotes out-of-plane motions.

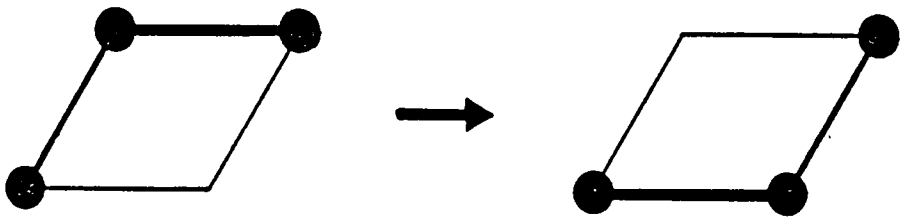

Figure 11. Move choice for 120^{o} bonds.

is aborted, and another begins. Periodically the chain coordinates are stored in the program memory. Once a sufficient number of coordinates are recorded, the file of chain coordinates, as a function of time, is temporarily stored on disk and the last set of chain coordinates is stored to be used as the initial coordinates in another run. Completing the output is the last step of the first program.

A second program is then run immediately. It reads the sets of coordinates from disk and calculates the end-to-end vector, R, for each set. Each run consists of 26000 samples taken at I(N-1) bead cycles apart (I = interval, and N = number of beads). A list of interval values for each N value are given in Table I. These R values are used to calculate the mean square end to end distance $\langle R^2 \rangle$ and autocorrelation values as defined in equation (5) in which f(t) = R(t). At the end of the second program, the autocorrelation values are stored for the use of the final program.

In the last program the log autocorrelation values are read from disk and used with calculated time values to determine the relaxation times. These relaxation times are calculated by a linear least squares fit of

TABLE I

LIST OF INTERVALS BETWEEN RECORDED COORDINATES FOR EACH CHAIN LENGTH

Number of beads (N)	Interval (no excluded volume)	Interval (excluded volume)
24	3	10
36	6	20
48	9	30
60	14	40
72	10	65

the log autocorrelation versus time interval as described in Chapter I. The slope is equal to $-1/T_R$. The time unit is N bead cycles. This is the time unit used by essentially all workers in this field. This result is the final step of the last program. This three program series completes the desired production of data for this project.

C. Results for the Adsorbed Chain

Simulations were done for chains with 24, 36, 48, 60, and 72 beads. The results of the $\langle R^2 \rangle$ and T_R calculations are listed in Tables II, and III respectively. Plots of data in Tables II and III are in Figures 12 through 16. Tables II and III list the $\langle R^2 \rangle$ and T_R values for the grafted chain and the free chain both with and without excluded volume. All simulation values were an average of at least five runs.

As Table II shows, the size of the grafted chain with no excluded volume is not significantly different from the free chain case. However, in the case of the grafted chain with excluded volume, the chain size does increase significantly. This increase in the size of

Table II

LIST OF MEAN SQUARE END-TO-END DISTANCES FOR GRAFTED AND FREE CHAINS BOTH WITH AND WITHOUT EXCLUDED VOLUME

	$\langle R^2 \rangle$			
	grafted chains		free chains*	
(N-1)	excluded volume	no excluded volume	excluded volume	no excluded volume
23	45.6 (1.4)	26.4 (0.7)	39.3	27.6
35	77.2 (2.0)	41.3 (1.1)	65.6	42.0
47	108.4 (2.6)	54.9 (1.6)	98.8	56.4
59	146.6 (3.8)	71.3 (1.8)	121.1	70.8
71	182.5 (3.3)	89.7 (3.0)	154.5	85.2

* Note: Values for $\langle R^2 \rangle$ for the free chain no excluded volume are calculated using the exact relationship shown by Domb and Fisher[16] to be $\langle R^2 \rangle = 1.2(N-1)$ for a free chain in a lattice with coordination number 12. The values for a chain with excluded volume are taken from Downey, Crabb, and Kovac.[1]

Table III

LIST OF RELAXATION TIMES FOR GRAFTED AND FREE CHAINS BOTH WITH AND WITHOUT EXCLUDED VOLUME

	T_R			
	grafted chains		free chains*	
(N-1)	excluded volume	no excluded volume	excluded volume	no excluded volume
23	810 (130)	275 (13)	249	84.1
35	2010 (290)	606 (51)	643	187
47	3630 (300)	1077 (87)	1190	316
59	6341 (1180)	1740 (150)	2030	533
71	9152 (1240)	2440 (500)	----	-----

* Note: Relaxation time values for free chains are taken from Downey, Crabb, and Kovac.[1]

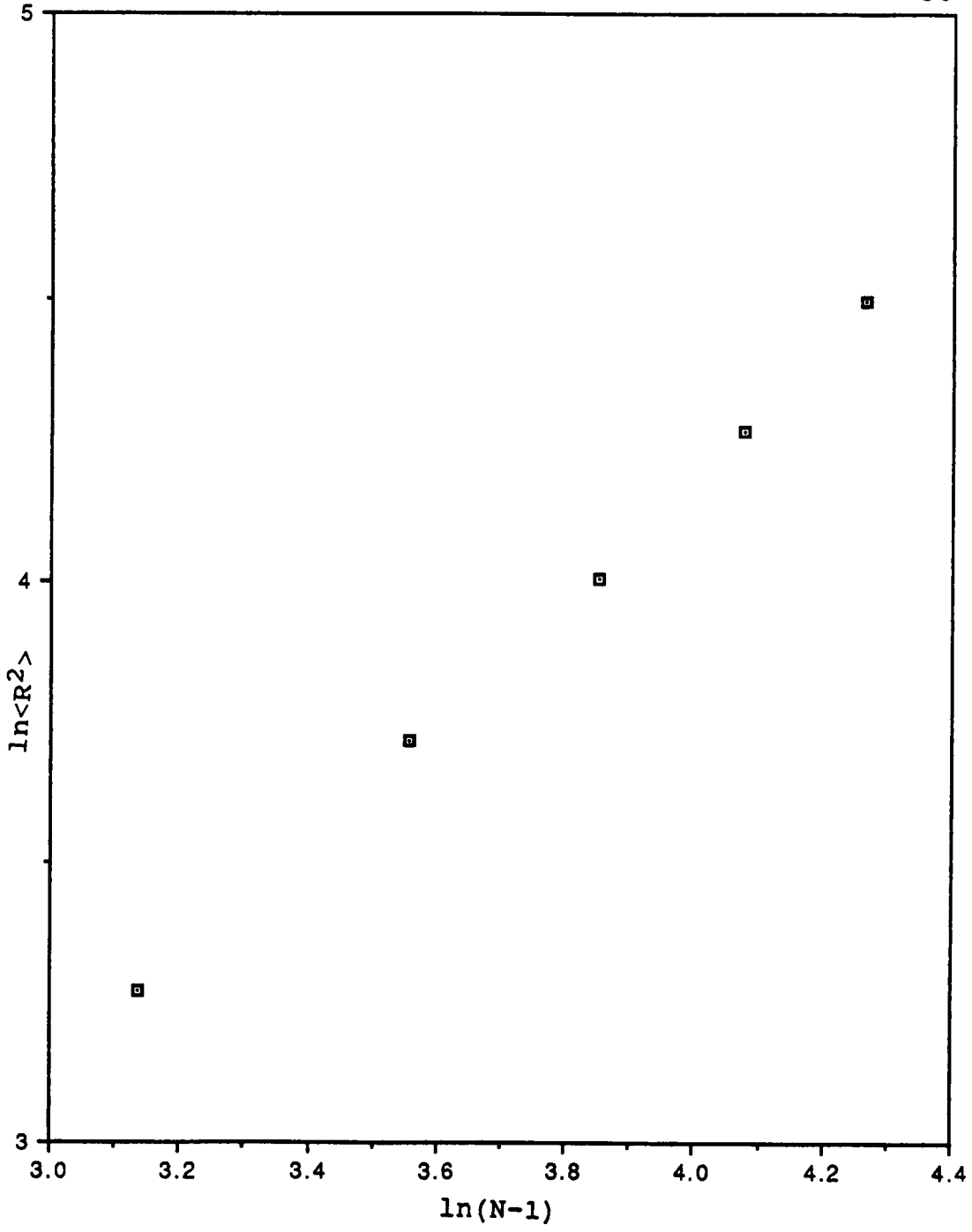

Figure 12. Log-log plot of $\langle R^2 \rangle$ versus (N-1) for a grafted chain with no excluded volume.

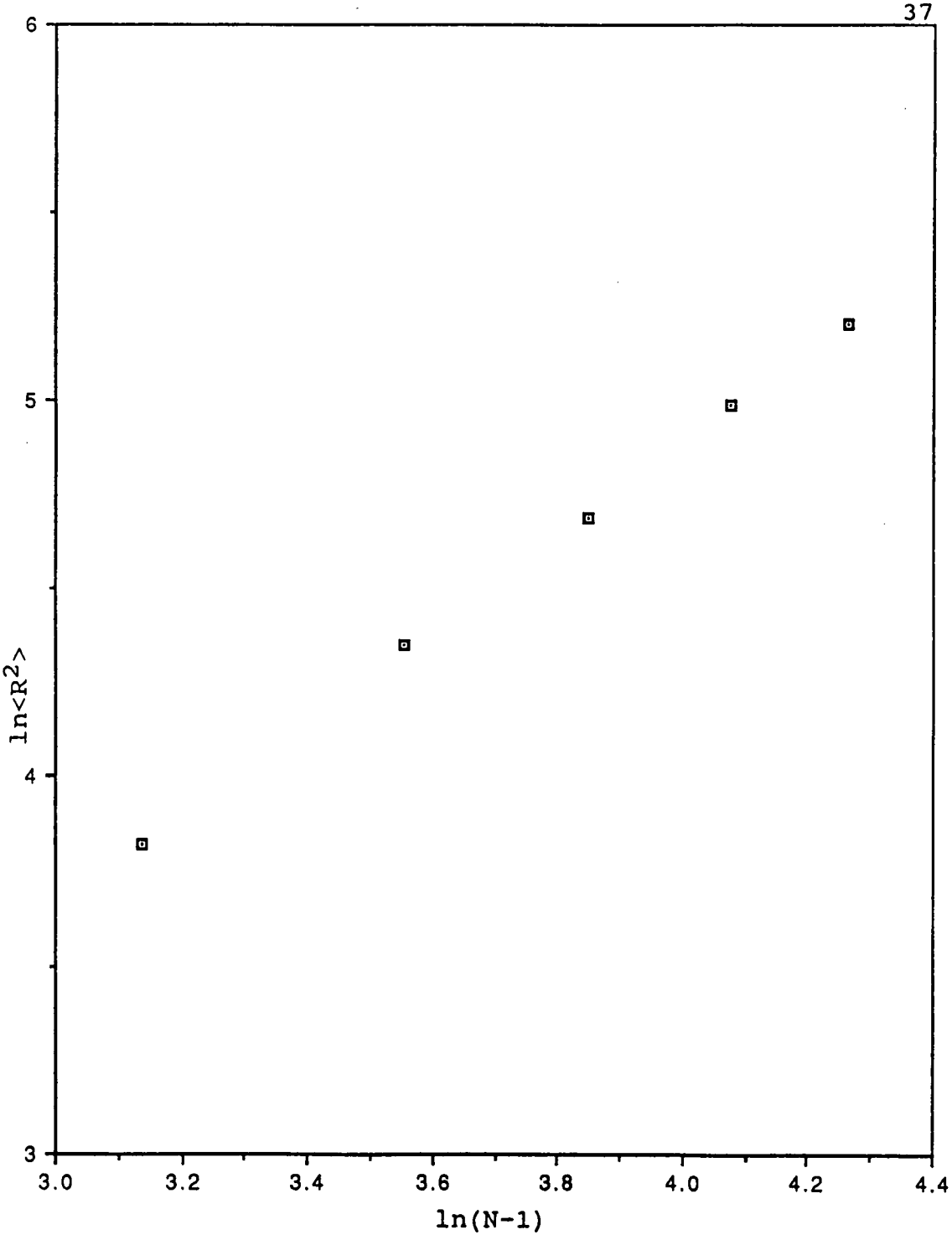

Figure 13. Log-log plot of $\langle R^2 \rangle$ versus (N-1) for a grafted chain with excluded volume.

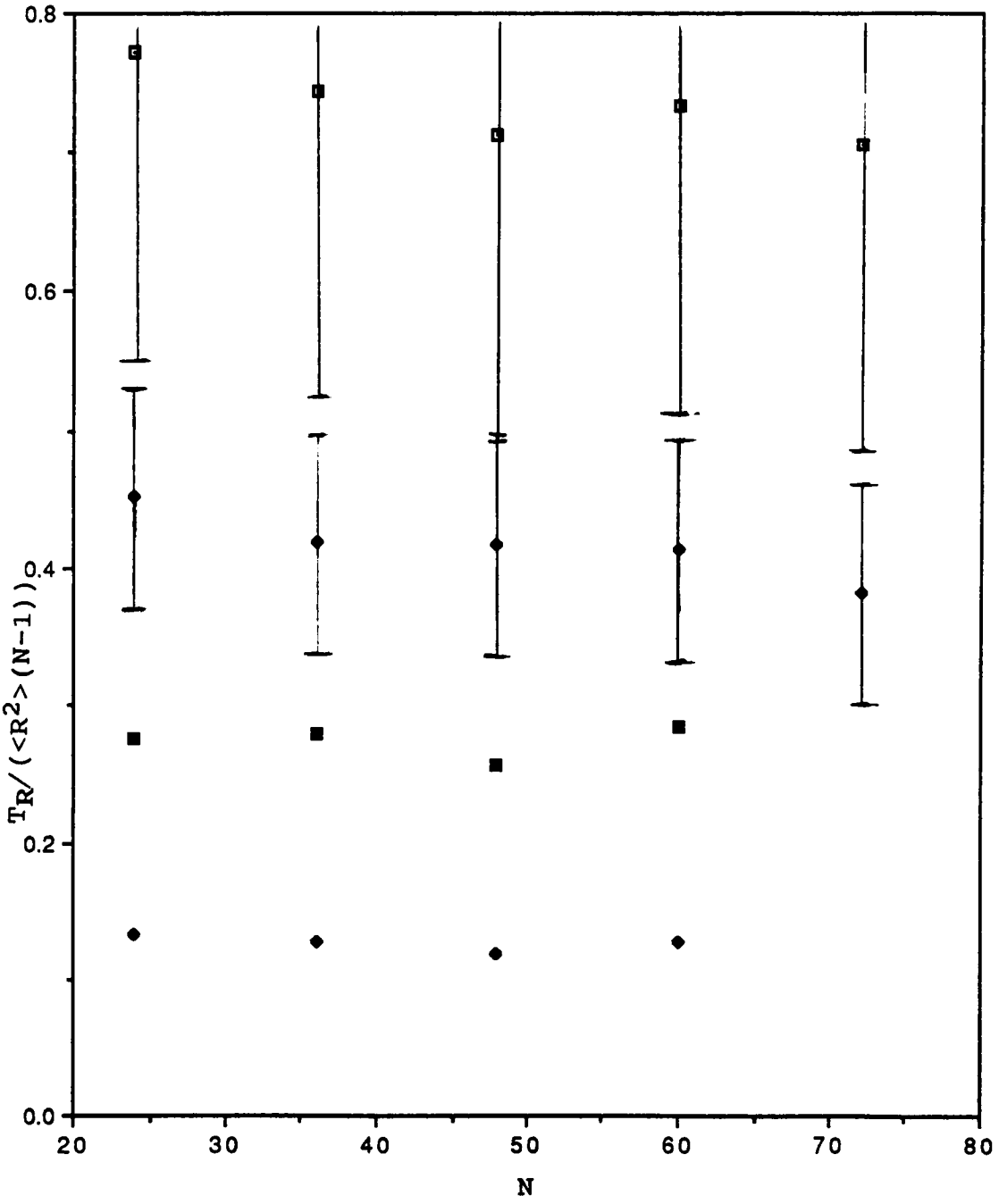

Figure 14. $T_R/(<R^2>(N-1))$ versus number of beads-grafted chains (open points); free chains (filled points): (■) excluded volume; (◆) no excluded volume.

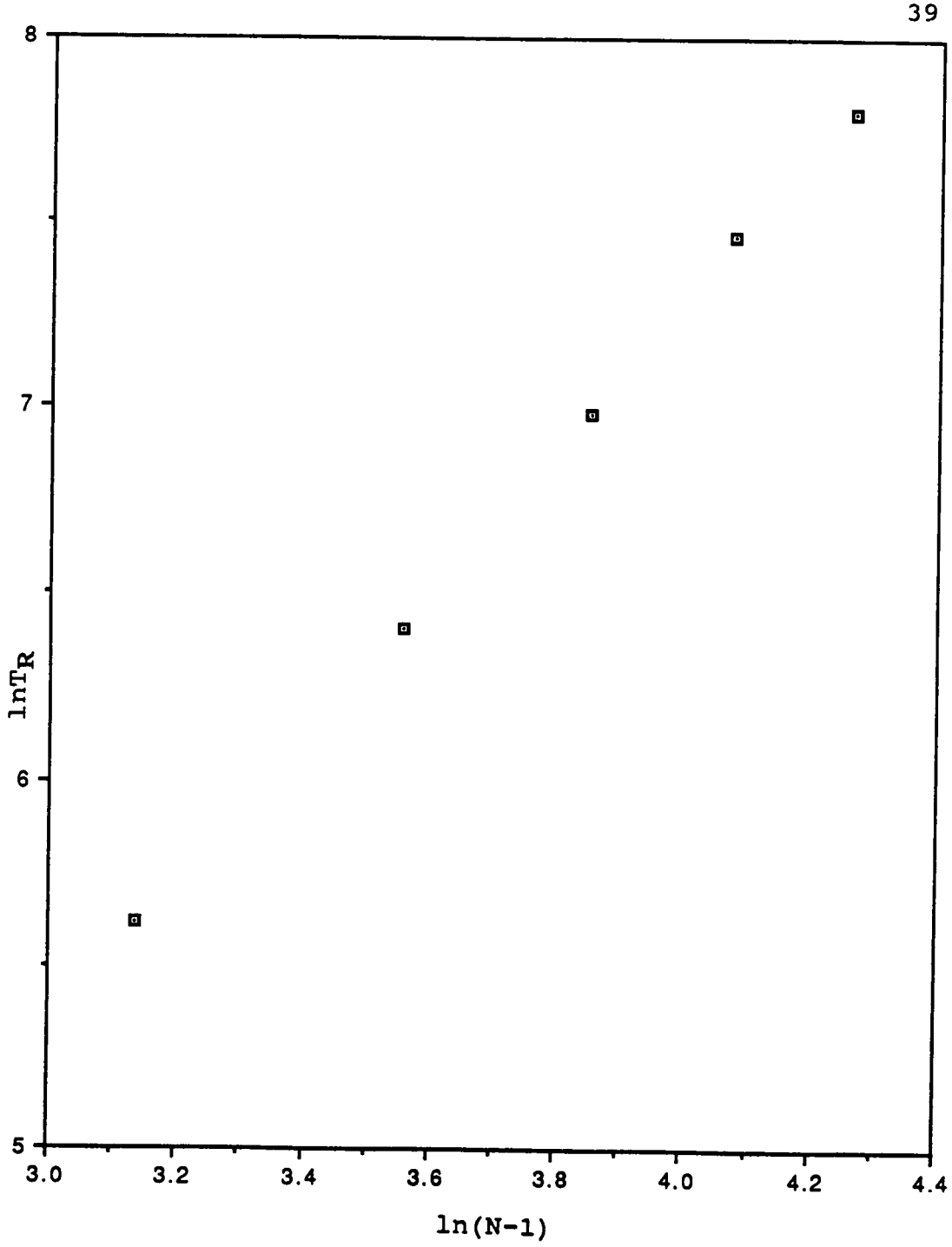

Figure 15. Log-log plot of T_R versus (N-1) for a grafted chain with no excluded volume.

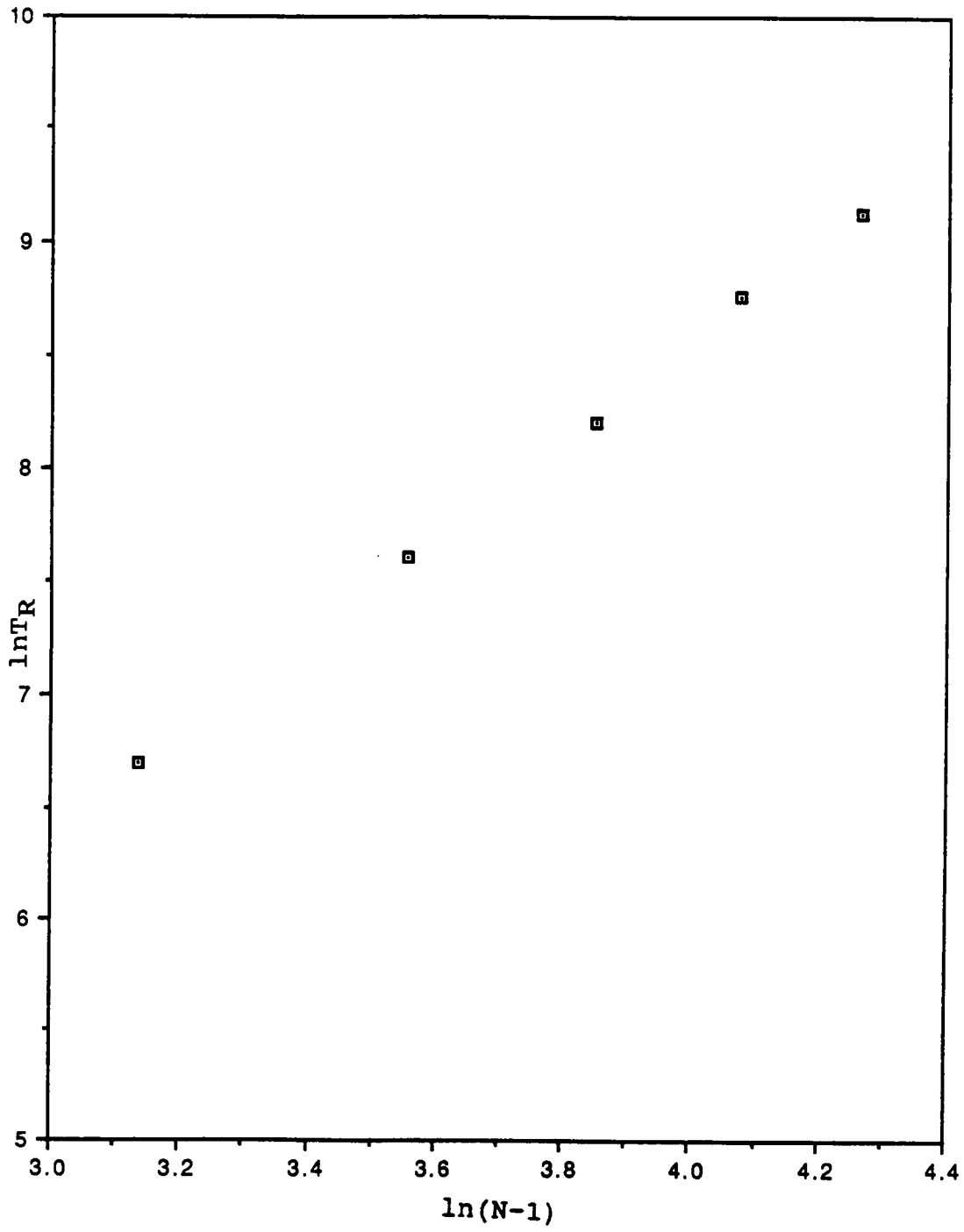

Figure 16. Log-log plot of T_R versus (N-1) for a grafted chain with excluded volume.

the self avoiding walk chain can be explained by the elimination of possible sites of the chain due to the surface of the plane. In order to maintain the excluded volume condition, the chain must extend further from the surface and thus further from the adsorbed end.

Log- log plots of $\langle R^2 \rangle$ versus (N-1) are in Figures 12 and 13. From Figure 12 the slope can be calculated to give a 2ν value of 1.07 for the no-excluded-volume grafted chain. For excluded volume (Figure 13) 2ν is 1.23. Both of these exponents agree with the expected values for a free chain. The grafted chain with excluded volume agrees with the value of 6/5 predicted in Flory[3] and it agrees with the results of Lax.[5] It seems reasonable that a grafted chain has the same chain length dependence as the free chain because the chain is still a self avoiding walk after a few steps away from the wall.

Of more interest are the end-to-end vector autocorrelations which probe the chain dynamics. Typical semilog plots of p(t) versus time are given in Figures 2 and 3 in Chapter I. At short times there is a definite nonlinear region. After this induction time, the plot becomes linear until statistical noise begins

to obscure the data. The relaxation times were calculated as (-1/slope) in the linear region. The linear region was chosen to be the same for all sets of runs for the sake of consistency. This range was chosen as $-0.5 \leqslant \ln p(t) \leqslant -1.2$ which appears to be linear in all plots. The effects of deviations at both ends of the range are minimized by averaging the relaxation time values of at least five runs per chain length. Table III compares relaxation time values of the grafted chain with those of the free chain taken from Downey, Crabb, and Kovac,[1] both with and without excluded volume. The ratio of relaxation times for the grafted to the free chain is in all cases approximately 3.2. This suggests the obvious, that the constrained chains have a lower mobility than free chains and thus much larger relaxation times.

The dynamic scaling hypothesis suggests that the relaxation time will be proportional to $\langle R^2 \rangle (N-1)$. If it is, the function

$$f = T_R / (\langle R^2 \rangle (N-1)) \qquad (9)$$

will be a constant. This has been shown to be true for

free lattice chains and it is of some interest to see if the hypothesis is true for grafted chains. If f is constant, f values for the free chains and terminally adsorbed chains can be compared. Any differences indicate chain length independent influences of the grafted chain on the relaxation times. Figure 14 is a plot of f against chain length. As Figure 14 shows, the function is either constant or has a small molecular weight dependence. As expected, chains with excluded volume have much larger relaxation times and therefore lower mobilities than those without excluded volume. The increase in the value of f for the grafted chain compared to the free chain is clearly due to restricted mobility induced by the flat, impenetrable surface. The chain length dependence of the relaxation time gives the same scaling for the grafted chains as for the free chains (Figures 15 and 16). For the excluded volume case, the relationship gives $T_R = a(N-1)^{2.16}$ and for the no excluded volume case, the relationship gives $T_R = a(N-1)^{1.95}$. This fits well with simulation scaling values for the free chain supplied by Downey and Kovac. They found values for the exponent to be 2.25 and 1.98 for the excluded volume and no excluded volume cases

respectively.

D. Conclusions about Grafted Chains

The chain length dependence of the mean square end-to-end distance $\langle R^2 \rangle$ and the relaxation times (T_R) for the grafted chains exhibits the same scaling behavior as free chains. In the no excluded volume case, the mean square end-to-end distance is approximately the same for both the grafted and free chain. For the excluded volume case, $\langle R^2 \rangle$ is significantly larger for the grafted chain than for the free chain. The large increase in $T_R/(\langle R^2 \rangle (N-1))$ for the grafted chains as compared to the free chains indicates that the slowing of the relaxation time is predominantly from the presence of the impenetrable barrier rather than any increase in $\langle R^2 \rangle$.

As described in Chapter I, some possible future studies are the following: (1) Study the static and dynamic influence of multiple chains adsorbed on the surface, as related to separation; (2) Monitor chains adsorbed on parallel walls as a function of separation; (3) Study the folding behavior of polymer chains in the

crystallization process. These studies are some of the next possible steps in simulations to relate them to chromatography, adhesion, and other surface science problems.

CHAPTER III

SIMULATIONS OF SHAPE FLUCTUATIONS OF SINGLE FREE CHAINS

A. Importance and Review

The time-averaged shape of a polymer chain is spherical, but at any instant it is better represented as an ellipsoid. This asymmetry influences properties such as viscosity, birefringence, dielectric relaxation, sedimentation, and diffusion. Clearly an ellipsoid flowing through a column will have a larger viscosity if the largest axis of the ellipsoid is oriented perpendicular to the flow rather than parallel. A better understanding of polymer shape and how shape affects its mobility can be interesting both to the experimentalist and the theoretician.

This chapter studies the nature of the shape and orientation of a polymer chain. This section reviews related literature. It focuses on size distribution of the principal axes and the dynamic properties as related to shape and orientation in both theoretical and experimental studies. Section B discusses the

simulation model used and section C discusses the results of the work done for this thesis and relates it to past literature.

Numerous theoretical studies of the size and shape of polymer chains are reported in the literature. In 1971 Solc and Stockmayer[17] published the ratio of the principal components (L_1,L_2,L_3) of the mean square radius of gyration for an unrestricted random walk on a simple cubic lattice, as studied by Monte Carlo methods. For $L_1<L_2<L_3$ the ratio is

$$\langle L_3\rangle:\langle L_2\rangle:\langle L_1\rangle = 11.80:2.70:1.00.$$

from which they claimed that a random-flight chain is more the shape of a cake of soap than the shape of a tennis ball. Such a ratio suggests a much larger asphericity than might be intuitively expected.

Later that same year, Solc[2] defined the symmetrical radius of gyration tensor $\underline{X}$ such that the elements of $\underline{X}$ are defined by equation (4) repeated here:

$$X_{ij} = (1/N) \sum_k x_i^k x_j^k - (1/N^2) \sum_k x_i^k \sum_k x_j^k. \quad (4)$$

The trace of $\underline{X}$ ($tr(\underline{X})$) gives the instantaneous radius of gyration S^2. $\underline{X}$ can be diagonalized to give sorted,

orthogonal principal components of S^2, $\underline{L} = D \underline{X} D^{-1}$, as described above, in which

$$\underline{L} = \begin{bmatrix} L_1 & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & L_3 \end{bmatrix} \tag{10}$$

is the eigenvalue tensor and

$$D = \begin{bmatrix} \cos\phi\cos\gamma-\sin\phi\cos\theta\sin\gamma & \sin\phi\cos\gamma+\cos\phi\cos\theta\sin\gamma & \sin\theta\sin\gamma \\ -\cos\phi\sin\gamma-\sin\phi\cos\theta\cos\gamma & -\sin\phi\sin\gamma+\cos\phi\cos\theta\cos\gamma & \sin\theta\cos\gamma \\ \sin\phi\sin\theta & -\cos\phi\sin\theta & \cos\theta \end{bmatrix} \tag{11}$$

is the eigenvector tensor.

An appropriate question would be how do different conditions affect asphericity? Additional work by Solc[18] published in 1973 suggested that rings are closer to being spherical in shape than linear chains. The same paper discussed stars and indicated that the asymmetry decreases with an increasing number of arms. Mazur, Gutman, and McCrackin[19] also published

information on chain shape. Their work studied the influence of excluded volume and interaction potential on both the simple cubic (SC) and face centered cubic (FCC) lattices. Their ratios of orthogonal components were found to be 14.8:3.1:1 in the SC lattice at zero potential and 13.9:2.9:1 in the FCC lattice at zero potential.

In 1973 Kranbuehl, Verdier, and Spencer published their first attempt to study the dynamics of shape in a polymer chain.[20] They investigated the fluctuations of the shape using a Monte Carlo simulation with no excluded volume. They produced results for the principal component ratios of two chain lengths (N = 63 and 9). These results reinforced the predicted values mentioned above (N = 63, 11.8:2.7:1.0; N = 9, 12.3:2.9:1.0). They claimed that the larger dimensions (R^2, and L_3) relaxed exponentially at approximately the same rate. This would suggest that the relaxation of R^2 is dominated by the relaxation of the largest principal component of S^2, which would be the slowest relaxation mode. L_2 relaxes at about 1/5 and L_3 at 1/10 the rate of the larger dimensions. Their work on chains with excluded volume is discussed later.

In 1975 and 1977 Rubin and Mazur presented a different measure of the asymmetry of the polymer chain.[21,22] They resolved the instantaneous radius of gyration onto three fixed axes (x,y,z) and sorted the axes as (u,v,w) such that u>v>w is always the case. They calculated the ratios of the first and second moments of the random walks of these chains for both unrestricted and self avoiding (SAW) walks. The second moment values (2.71:1.60:1 unrestricted, and 3.08:1.71:1 SAW) are clearly much smaller than those for the calculated principal components of S^2. This difference is simply explained. The principal axes of a random chain will be randomly oriented and are not likely to coincide with fixed axes. As a consequence, measurements of size distribution along each fixed axis will only be a projection of the extensions along the principal axes. This projection is some combination of the three principal axes. Therefore, the fixed axes are not an indication of the true principal axis lengths. The eigenvalues L_1, L_2, and L_3 are aligned along the three principal axes. As a result, they are much more indicative of the size distribution of the randomly oriented chain.

Kranbuehl and Verdier attempted to determine the shape relaxation of a random coil polymer chain on a simple cubic (SC) lattice. Unfortunately, they were using a simulation on the SC lattice which did not include a crankshaft motion. This motion was later determined to be necessary in order to be able to use a SC lattice for the simulation of polymer chain dynamics with excluded volume. Therefore, their dynamic values are not valid and are not presented here. However, they also calculated the static ratios of the principal components for a series of chain lengths both with and without excluded volume. They agreed well with the ratios calculated by Solc and coworkers and by Mazur and coworkers listed above.

Minato and Hatano determined the expansion factor using the perturbation series in the excluded volume parameter z for the orthogonal components of S^2 along the principal axes of inertia. They calculated the first order[23] and second order[24] expansion factors for the expansion series in z. The result was

$$r_3 = \langle L_3 \rangle / \langle L_3 \rangle_o = 1 + 1.4079z - 3.2842z^2 \qquad (12a)$$

$$r_2 = 1 + .9791z - 2.6833z^2 \qquad (12b)$$

$$r_1 = 1 + .7365z - 2.1986z^2. \qquad (12c)$$

Their resulting calculations of the unperturbed and perturbed (SAW) component ratios (9.22:2.27:1, and 11.3:2.47:1 respectively) were smaller than those calculated by the Monte Carlo method mentioned earlier. They attribute these differences to the approximate nature of their work. Nevertheless, their results do qualitatively agree with previous results, because they show the high asymmetry and the increase of asymmetry with excluded volume.

The effect of chain interaction was considered by Olaj and Zifferer. They generated ensembles of 50 segments for SAW chains and studied the influence of separation of the two chains on the average values of the principal components. Their results agree with intuition. They found that the largest principal component of S^2 increased with decreasing separation. They observed the same result, only to a lesser extent, for the second component. The smallest eigenvalue decreased with decreasing separation.

In a recent publication, Bishop and Clark[25] generated values of asphericity (A_3) defined by

$$A_3 = \langle (\sum_{i>j} L_i - L_j)^2 \rangle / 2 \langle (\sum_{i} L_i)^2 \rangle . \qquad (13)$$

If the chain is a sphere, $A_3 = 0$, and if the chain is a rod, $A_3 = 1$. They showed that star polymers are less asymmetrical than linear chains. They also showed that both star and branch chains approach spherical symmetry in the collapsed chain ($A_3 << 1$).

Relaxation times for dilute polymer solutions can be determined by methods such as NMR,[26-29], dielectric relaxation,[29,30] and depolarized Rayleigh spectroscopy.[31] In some cases, molecular weight dependence was observed, but in other cases it was not. Those studies which showed molecular weight dependence were either for systems with low molecular weight polymers or with relatively stiff backbones. In NMR studies by Stockmayer and coworkers,[26-29] molecular weight independent relaxation times were observed. They dominated the spin lattice relaxation processes because the relaxation of small side groups or segments of the backbone, which are molecular weight independent, were fastest and most readily observable.

In dielectric relaxation processes, both molecular

weight independent and molecular weight dependent relaxation processes were observed. If the relaxation process is for a relatively long or flexible chain, or the dipole is positioned on a side group extending from the backbone, the relaxation is most likely to be molecular weight independent, as was the case in one study by Matsuo, Stockmayer, and Mashimo.[29] In another study by Stockmayer and Matsuo, molecular weight dependence was observed, but only for low molecular weight materials. As the molecular weight became very large, the relaxation times levelled off and became molecular weight independent. This suggests that the local motions relaxed faster than the overall chain at very large molecular weights.

Good molecular weight dependence for the relaxation time of solvated polystyrene was observed by Bauer, Brauman, and Pecora,[31] even at high molecular weights. They used depolarized Rayleigh spectroscopy to measure the relaxation of polystyrene (PS) in solution. They found the relaxation time to be proportional to the product of the weight average molecular weight, the intrinsic viscosity, and the solvent viscosity as was predicted by Zimm.[32] In any case, however, only one

molecular weight dependent relaxation time has been observed.

It is difficult to see shape in polymer solution because of the rapid motions of the polymer chains in the midst of the solvent. In the solid state, however, polymer chains are essentially frozen in place. Aharoni illustrated that the individual chains lack spherical symmetry[33] by imaging iodized polystyrene (IPS) in a continuous polystyrene (PS) matrix using the electron microscope. The IPS has a higher electron density than the PS matrix and will reflect the electron beam more. As a result aggregates of IPS and even individual IPS molecules are observable. From the pictures supplied in Aharoni's paper, it is clear that the polymer chains are highly anisotropic.

This concludes the discussion of the literature on the shape and dynamics of polymer chains. It is now desirable to discuss the simulation used in the thesis.

B. The Simple Cubic Lattice Model

In this section, details of the simulation method used to model shape and shape fluctuations are

explained. A simple cubic (SC) lattice model is used here. Justification for using the SC lattice and basic movement rules for the SC lattice are discussed below.

Lattice models represent the polymer chain as a series of connected units called beads. Each bead is a statistical segment representing several repeat units in the polymer chain. The basic simulation performed here involves a chain whose members are represented by integer coordinates in a simple cubic lattice. For example, if the first bead is represented by the integer coordinates (n_x,n_y,n_z) then the second bead can be located at any of six coordinates a unit distance from the first one. Choosing the SC lattice in preference to any other lattice, such as the face-centered cubic lattice used in chapter II, is based on one simple reason. Many simulations have been done on the SC lattice before, so the results obtained here can be compared with those in the literature. The chain movement algorithm and data analysis are described in the following algorithm. The simulation model is the one developed by Gurler et al.[34]

The entire simulation and basic analysis is contained in three FORTRAN programs listed in Appendix

II. The first program does the actual simulation as described here. First a bead is randomly chosen. If it is an end bead, the program randomly selects among the five other sites that are a unit distance from the bead connected to the chosen end. If the program uses excluded volume, a check is made to see if the site is already occupied. If it is, the move is aborted. If no excluded volume is considered, all moves are allowed.

If the chosen bead is an internal bead, three types of moves are possible depending on the configuration of the chain and on whether or not excluded volume is considered. The moves available to both cases are discussed first. The coordinates of the two beads adjacent to the chosen bead are determined. If these three beads are in a straight line, the chosen bead cannot be moved (Figure 17(a)). It is possible for the bead bonds to be at 90 degrees (Figure 17(b)). In this case, only one other site is a unit distance from the two adjacent beads. It is chosen for the move attempt. If no excluded volume is required, the move is always allowed. If excluded volume is required, the move is checked for whether or not the site is occupied. If it is occupied, a crank shaft move, shown in Figure 18, is

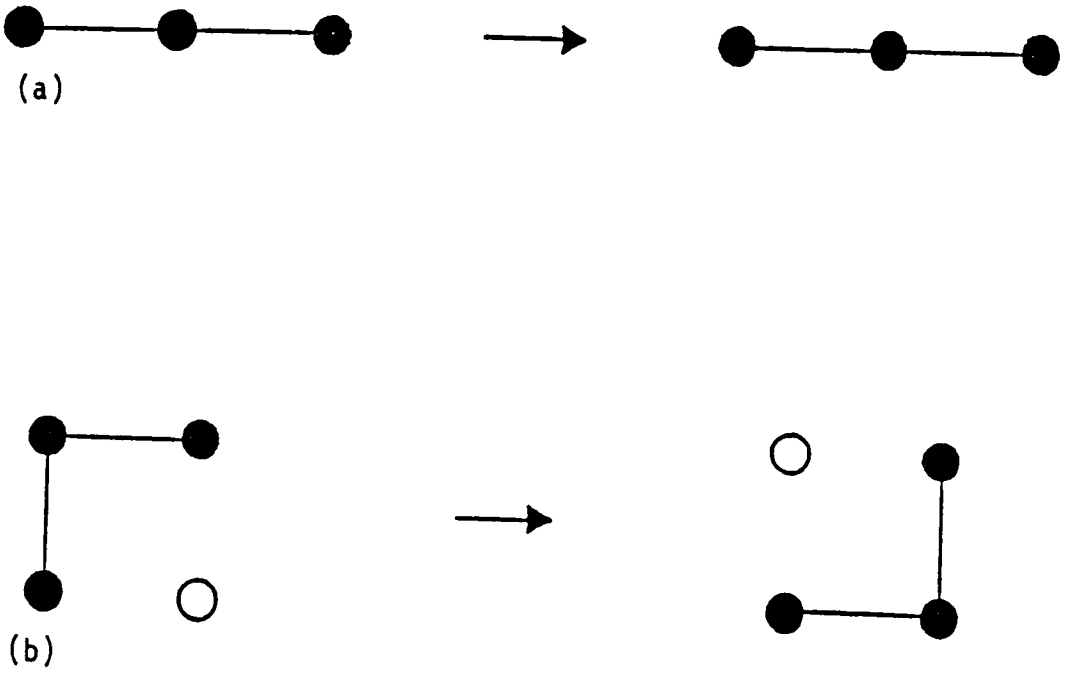

Figure 17. Possible configurations for (a) 180° and (b) 90° bonds in a simple cubic lattice.

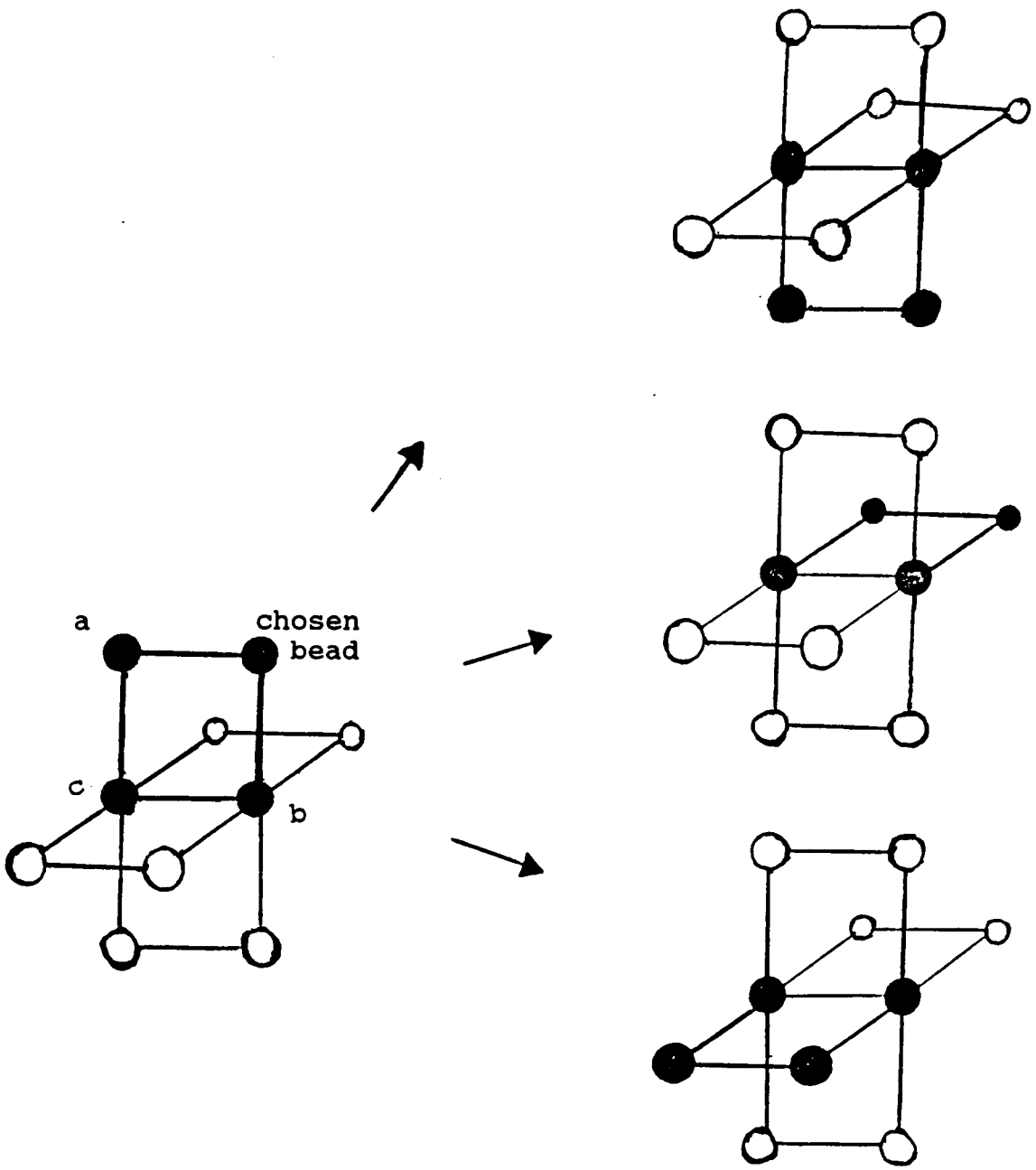

Figure 18. Possible crank shaft motions.

attempted.

The crank-shaft movement attempt is determined as follows. As Figure 18 shows, the fourth member of the crank (member c) must be connected to one of the two beads adjacent to the chosen bead. One of the other three crank-shaft orientations is randomly chosen for the move. If one of the two locations of the new orientation is occupied, the move is aborted.

In the no excluded volume case, a spike configuration is allowed. If a spike configuration is determined, there are five possible new locations for the chosen bead. The move is treated like an end bead move.

The bead cycle is complete even if the move is aborted, and another begins. Periodically the chain coordinates are stored as a function of time in the program memory. Once a sufficient number of these coordinates are recorded, the set of chain coordinates is temporarily stored on disk and the last set of coordinates is stored for use as the initial configuration for another run. Completing the output is the last step of the first program and the end of the simulation.

A second program is run immediately. This program monitors the anisotropy effects on size and dynamics. It follows the motion of the three principal axes of the chain by separating them into three breathing modes (eigenvalue relaxations) and three rotation modes (spherical harmonics relaxations). It reads the sets of coordinates from disk and calculates the radius of gyration tensor ($\underline{X}$) given by equation (4). The program then diagonalizes $\underline{X}$ and sorts the resultant eigenvalues (L_i, i=1,2,3) from largest to smallest in the matrix ($\underline{L}$). Each run consisted of 30000 samples taken at IXN bead cycles apart (I = interval, and N = number of beads). These L_i values are used to calculate the average principal components $\langle L_i \rangle$ of the mean-square radius of gyration and the autocorrelation values of equation (5) in which $f(t) = L_i(t)$. Also, time correlations of the squared spherical harmonics (described in part C) are calculated with $f(t) = Y_i^2(t)$. The spherical harmonics functions (Y_i) are prone to sign changes. This is a result of the fact that the diagonalization algorithm can choose between two possible orientations, 180 degrees to each other, for each of the three principal axes. Since both

orientations are valid choices, the easiest solution is to square Y_i and calculate $p_f(t)$ for $f = Y_i^2$. This eliminates the non-physical jumps in the data due to the sign changes. One of these sets is chosen to be the output (see Appendix II) onto the disk as $\ln(p_f(t))$ to be used by the last of the three programs. Then, the second program ends.

In the last program, the log autocorrelation values are read from disk and used with calculated time values to determine the relaxation times. As explained in Chapter II, the relaxation time for each of the eigenvalues can be determined from a linear least squares fit of $\ln p_f(t)$ versus time($f = L_i$). The slope is equal to $(-1/T_f)$. For the rotational autocorrelations ($f = Y_i^2$), a four point quadratic interpolation is used to find T_f at $\ln(p_f(t)) = -1$. This result is the final step of the last program.

This three program series completes the production of data for this project. The results and discussion follow.

C. Results and Discussion for Chain Shape

Simulations were done on chains with 24, 36, 48, 60, and 72 beads. At least five runs for each chain length, for each eigenvalue measure, and for each rotational relaxation time measure were averaged. The ratios of the averaged principal components $\langle L_i \rangle$ of the mean-square radius of gyration, $\langle S^2 \rangle$, as well as the actual sizes are given in the order $\langle L_1 \rangle \geq \langle L_2 \rangle \geq \langle L_3 \rangle$ in Tables IV and V. The relaxation times for the eigenvalues and for the rotational functions are given in Tables VI and VII. Ratios of relaxation times for the eigenvalues and the corresponding normal modes (taken from Dial, Crabb, Crabb, and Kovac[35] are given in Table VIII. Table IX is a list of the function $f_i = T_{Li}/(\langle L_i \rangle (N-1))$. Ratios of the rotational relaxation times to the corresponding eigenvalue relaxation times are given in Table X. Log-log plots of the eigenvalues against number of bonds (N-1) are shown in Figures 19 through 24. Log-log plots of the eigenvalue relaxation times relaxation times are plotted in Figures 25 through 30. Figures 31 and 32 are plots of the function $f_i = T_{Li}/(\langle L_i \rangle (N-1))$ versus (N-1) for the three eigenvalues.

TABLE IV

LIST OF EIGENVALUE RATIOS FOR BOTH NO EXCLUDED VOLUME AND EXCLUDED VOLUME CASES

Number of bonds (N-1)	$\langle L_1 \rangle : \langle L_2 \rangle : \langle L_3 \rangle$ (no excluded volume)	$\langle L_1 \rangle : \langle L_2 \rangle : \langle L_3 \rangle$ (excluded volume)
23	11.95:2.77:1.00	15.23:3.05:1.00
35	12.05:2.77:1.00	15.23:3.04:1.00
47	12.85:2.70:1.00	15.09:3.03:1.00
59	13.19:2.73:1.00	15.07:3.03:1.00
71	13.42:2.77:1.00	14.92:3.04:1.00
average	12.77:2.75:1.00	15.11:3.04:1.00

TABLE V

THE PRINCIPLE COMPONENTS (EIGENVALUES) OF THE RADIUS OF GYRATION BOTH WITH AND WITHOUT EXCLUDED VOLUME

	No Excluded Volume			With Excluded Volume		
(N-1)	$\langle L_1 \rangle$	$\langle L_2 \rangle$	$\langle L_3 \rangle$	$\langle L_1 \rangle$	$\langle L_2 \rangle$	$\langle L_3 \rangle$
23	2.999	0.722	0.2610	5.853	1.1723	0.3842
35	4.687	1.038	0.3890	9.720	1.9425	0.6384
47	6.666	1.399	0.5187	13.729	2.7611	0.9101
59	8.637	1.776	0.6500	18.051	3.6339	1.1981
71	10.497	2.166	0.7821	22.455	4.5792	1.5067

TABLE VI

RELAXATION TIMES FOR THE THREE PRINCIPLE COMPONENTS OF THE RADIUS OF GYRATION AND THE CORRESPONDING SPHERICAL HARMONICS WITHOUT EXCLUDED VOLUME*

	Relaxation Times					
(N-1)	L_1	L_2	L_3	Y_1^2	Y_2^2	Y_3^2
23	31.0	7.458	2.12	3.321	2.112	2.789
	(7.9)	(0.55)	(0.23)	(0.34)	(0.34)	(0.12)
35	86.6	16.823	5.13	7.243	5.352	5.874
	(20)	(2.3)	(1.3)	(1.0)	(0.92)	(0.25)
47	151.6	27.953	7.91	13.804	10.344	11.077
	(20)	(4.0)	(2.4)	(2.0)	(1.4)	(0.83)
59	237.1	48.153	13.26	23.852	15.772	17.508
	(40)	(5.3)	(1.2)	(1.9)	(4.0)	(1.3)
71	310.3	70.694	19.95	33.942	23.203	24.145
	(60)	(13)	(2.3)	(4.1)	(6.6)	(2.6)

* Note: The values parentheses are the absolute standard deviations for the values immediately above them.

TABLE VII

RELAXATION TIMES FOR THE THREE PRINCIPLE COMPONENTS OF THE RADIUS OF GYRATION AND THE CORRESPONDING SPHERICAL HARMONICS WITH EXCLUDED VOLUME*

	Relaxation Times					
(N-1)	L_1	L_2	L_3	Y_1^2	Y_2^2	Y_3^2
23	39.86	10.93	4.13	7.636	5.267	5.290
	(8.7)	(2.2)	(0.72)	(0.52)	(0.95)	(0.32)
35	104.38	25.72	9.71	19.759	10.922	13.048
	(12)	(2.3)	(0.71)	(3.1)	(2.4)	(1.1)
47	216.72	47.83	19.24	38.095	28.943	26.824
	(35)	(8.9)	(2.3)	(4.9)	(13)	(3.4)
59	343.99	89.60	32.66	56.368	42.932	42.019
	(35)	(12)	(3.8)	(5.2)	(9.2)	(3.8)
71	471.73	119.23	47.03	111.159	58.755	64.953
	(88)	(13)	(1.7)	(22)	(8.1)	(13)

* Note: The values in parentheses are absolute standard deviations of the values immediately above them.

TABLE VIII

RATIOS OF EIGENVALUE RELAXATION TIMES TO CORRESPONDING NORMAL MODES (T_{Li}/T_i, i = 1, 3) BOTH WITH AND WITHOUT EXCLUDED VOLUME*

	No excluded Volume			With excluded Volume		
(N-1)	1	2	3	1	2	3
23	0.397	0.407	0.286	0.220	0.287	0.317
35	0.494	0.395	0.287	0.246	0.254	0.256
47	0.488	0.361	0.235	0.268	0.256	0.245
59	0.473	0.398	0.257	0.262	0.288	0.263
71	---	---	---	---	---	---

* Note: The normal modes used were taken from Dial, Crabb, Crabb, and Kovac.[35]

TABLE IX

THE FUNCTION $f_i = T_{Li}/(\langle L_i \rangle (N-1))$ $(i = 1, 3)$ BOTH WITH AND WITHOUT EXCLUDED VOLUME

	No excluded Volume			With excluded Volume		
(N-1)	1	2	3	1	2	3
23	0.4327	0.4491	0.3523	0.2961	0.4054	0.4678
35	0.5280	0.4629	0.3768	0.3068	0.3783	0.4346
47	0.4839	0.4285	0.3244	0.3359	0.3686	0.4499
59	0.4654	0.4595	0.3458	0.3230	0.4179	0.4620
71	0.4163	0.4597	0.3593	0.3062	0.3667	0.4396

TABLE X

RATIO OF THE SPHERICAL HARMONICS TO CORRESPONDING EIGENVALUE RELAXATION TIMES
(T_f/T_g, $f = Y_i^2$, and $g = L_i$, $i = 1, 3$)

	No excluded Volume			With excluded Volume		
(N-1)\i	1	2	3	1	2	3
23	0.1070	0.2831	1.3187	0.1916	0.4819	1.2796
35	0.0836	0.3181	1.1450	0.1893	0.4247	1.3438
47	0.0911	0.3700	1.3951	0.1758	0.6051	1.3940
59	0.1006	0.3275	1.3204	0.1638	0.6788	1.2866
71	0.1094	0.3282	1.2107	0.2356	0.4928	1.3811

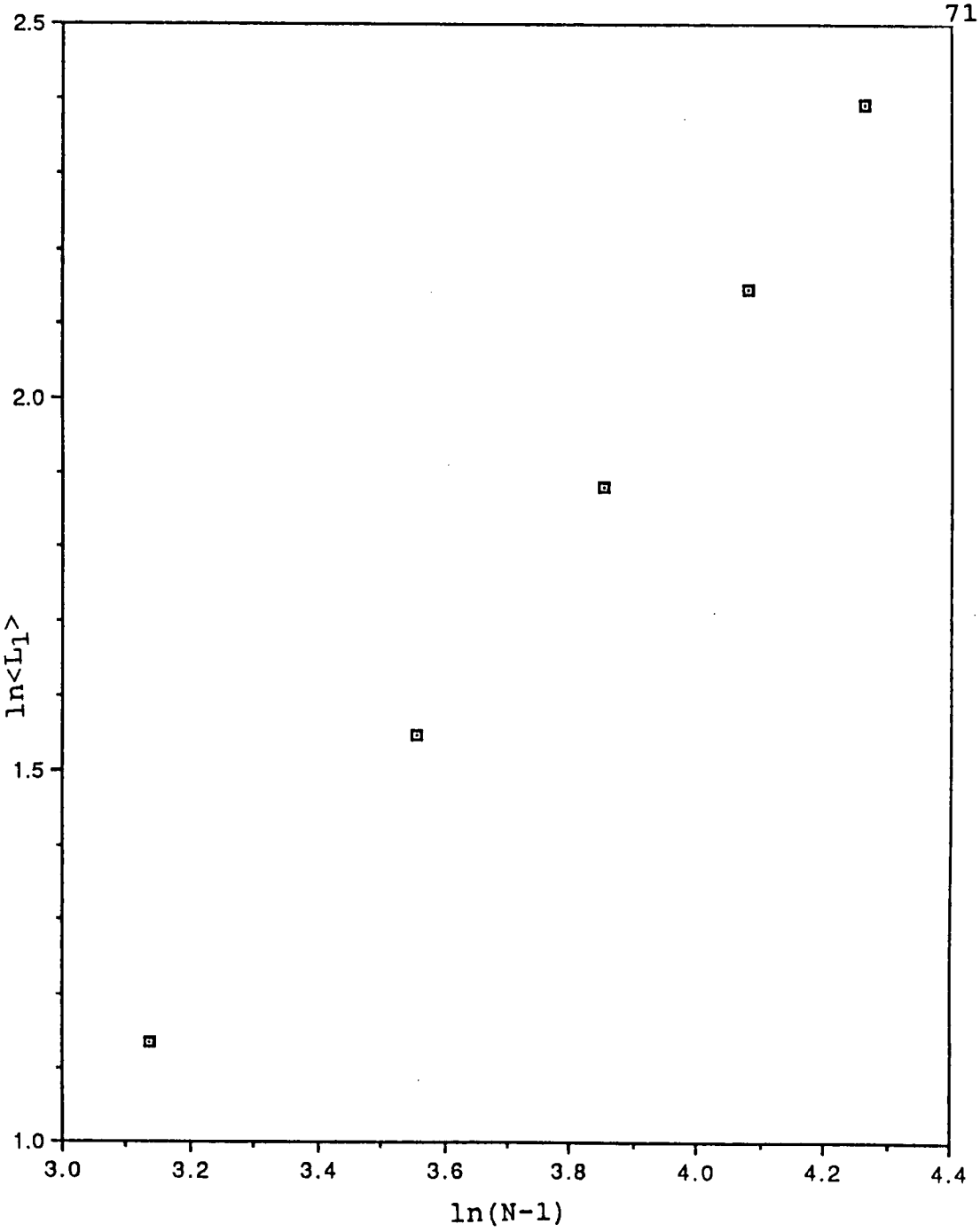

Figure 19. Log-log plot of $\langle L_1 \rangle$ versus (N-1) with no excluded volume.

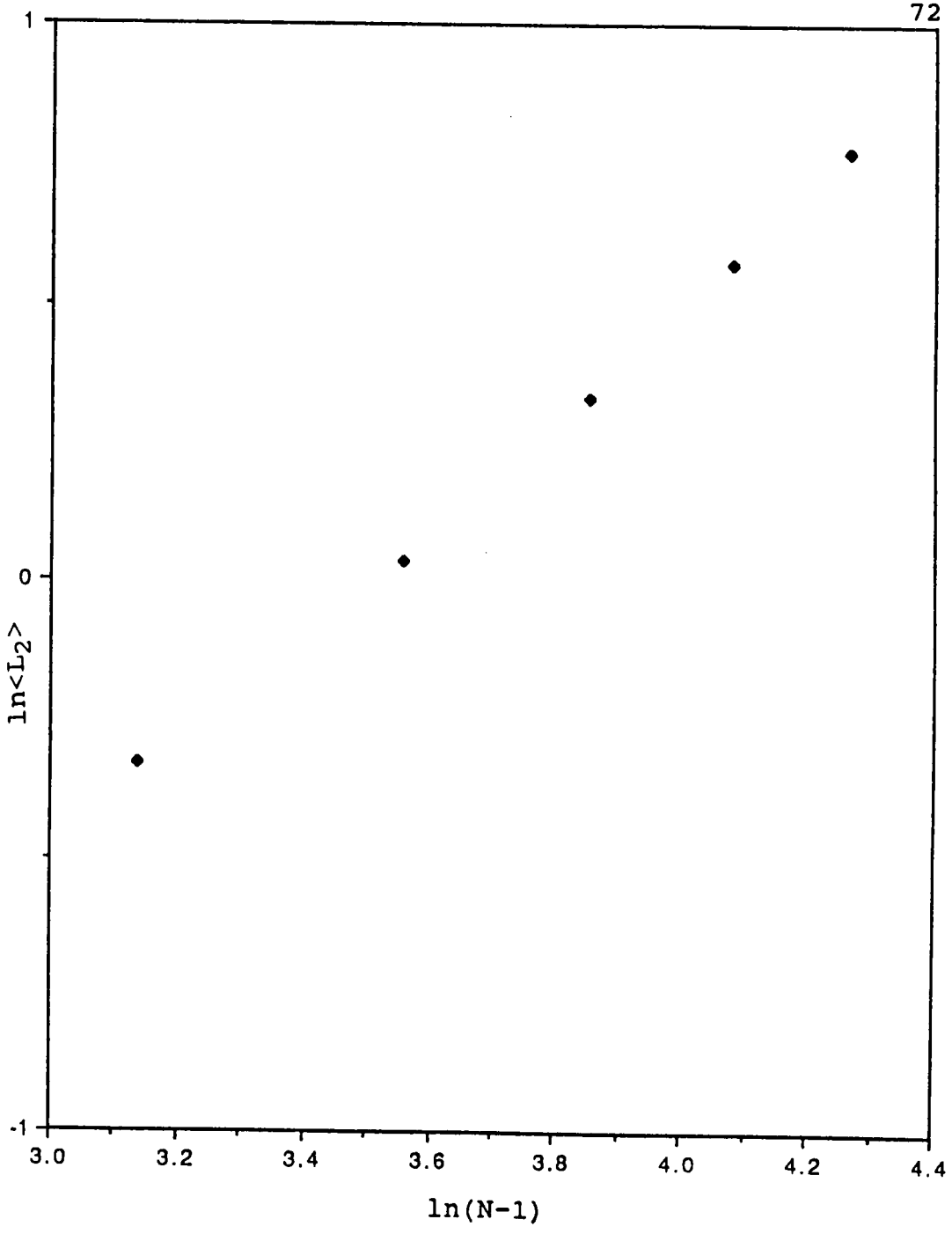

Figure 20. Log-log plot of $\langle L_2 \rangle$ versus (N-1) with no excluded volume.

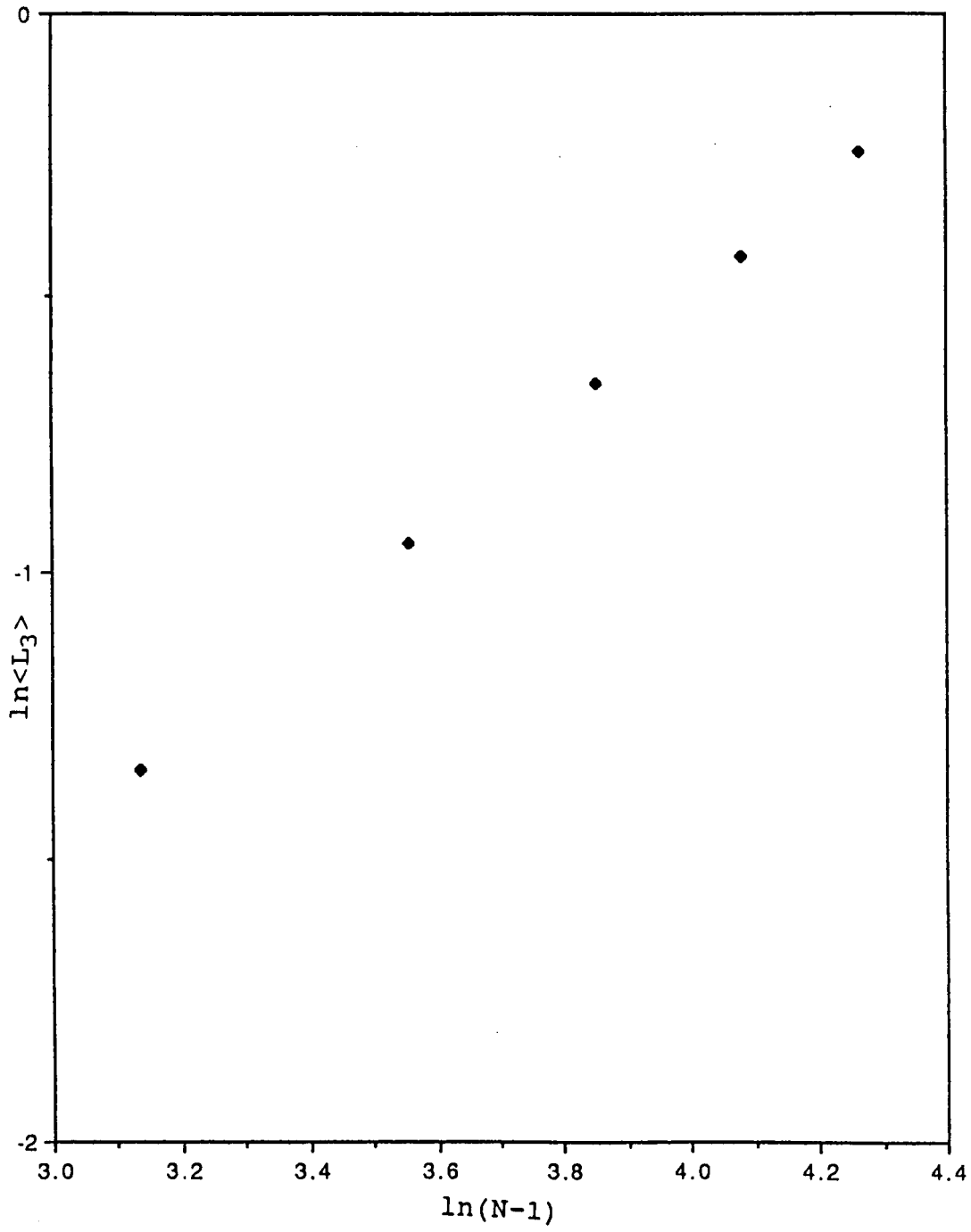

Figure 21. Log-log plot of $\langle L_3 \rangle$ versus (N-1) with no excluded volume.

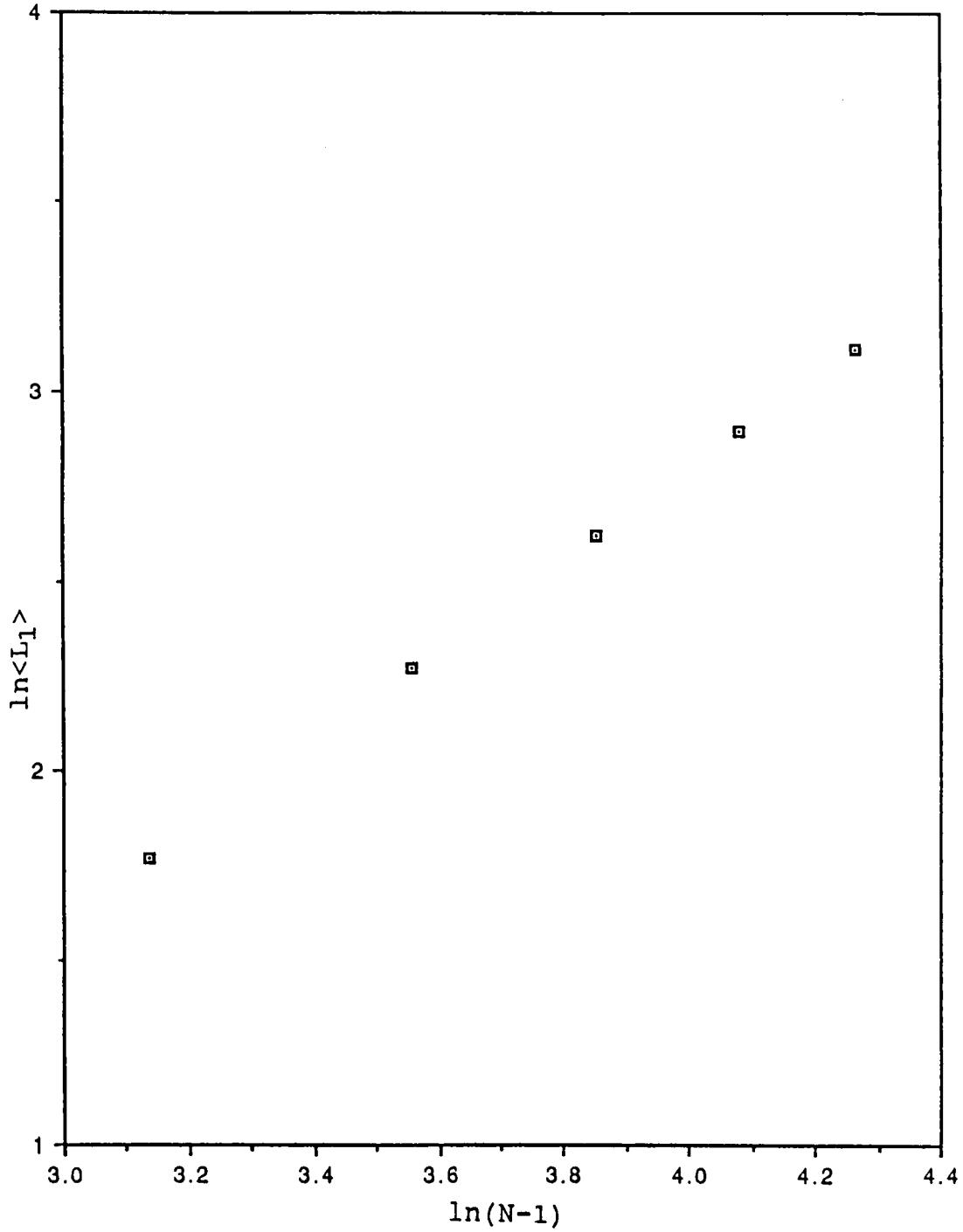

Figure 22. Log-log plot of $\langle L_1 \rangle$ versus (N-1) with excluded volume.

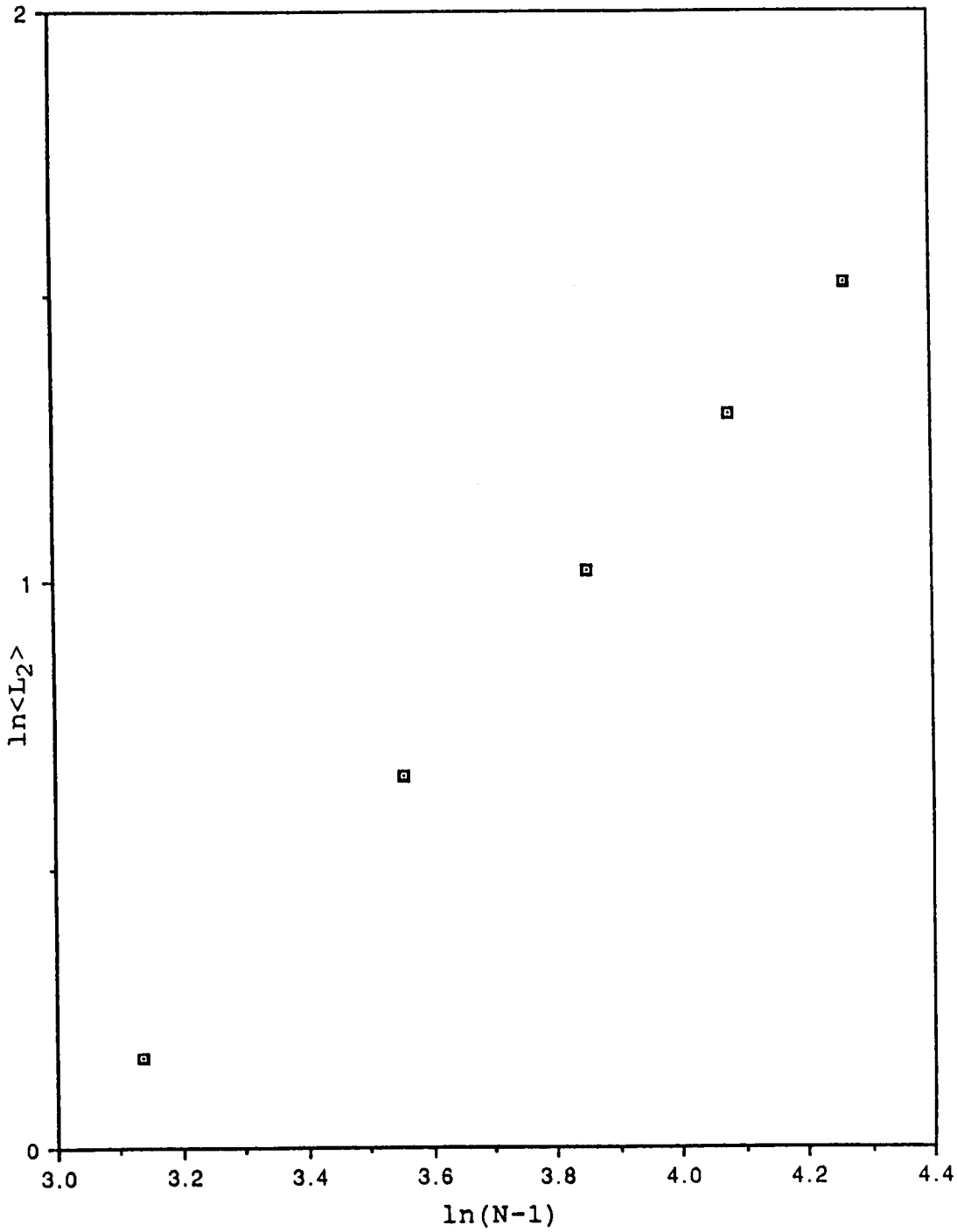

Figure 23. Log-log plot of $\langle L_2 \rangle$ versus (N-1) with excluded volume.

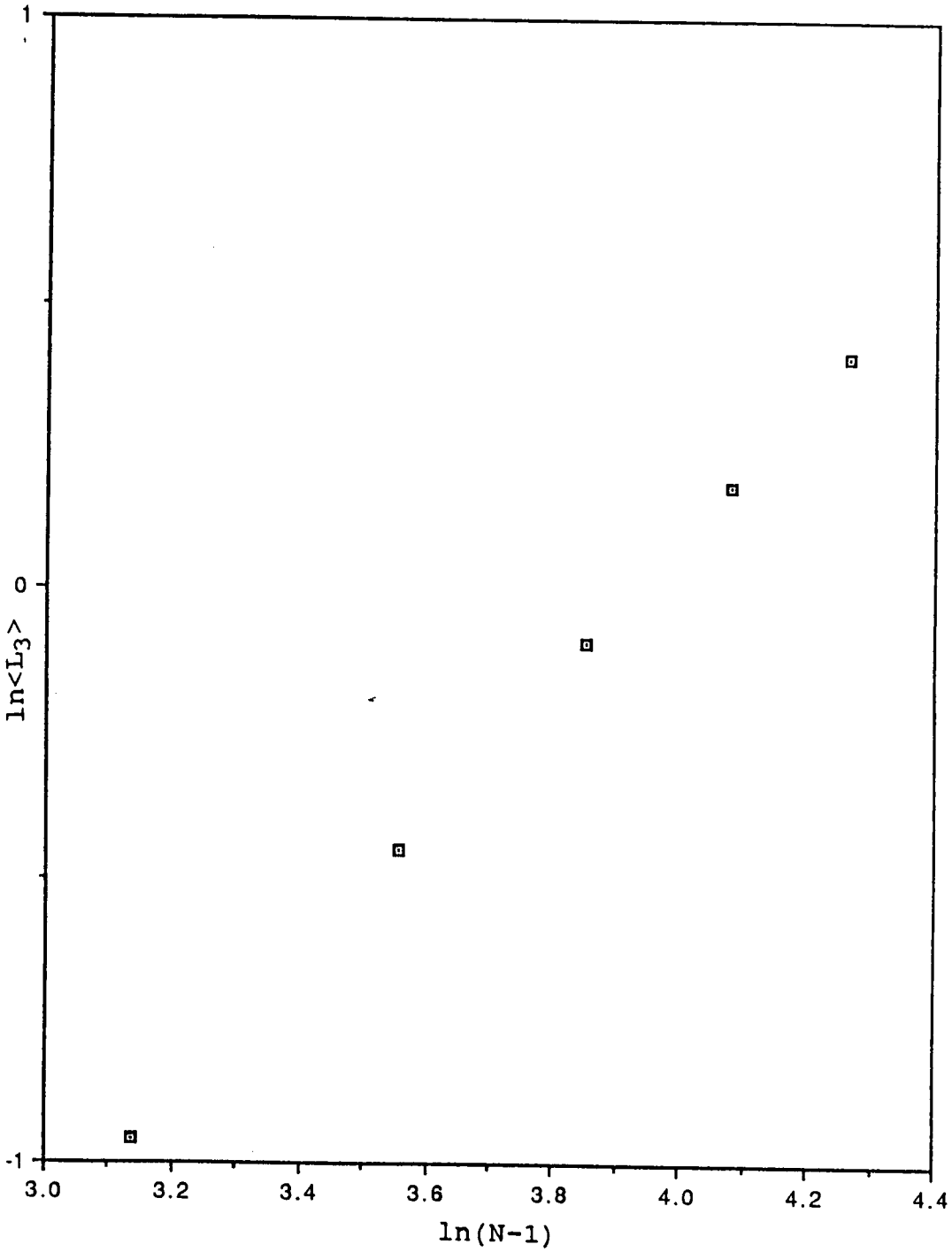

Figure 24. Log-log plot of $\langle L_3 \rangle$ versus (N-1) with excluded volume.

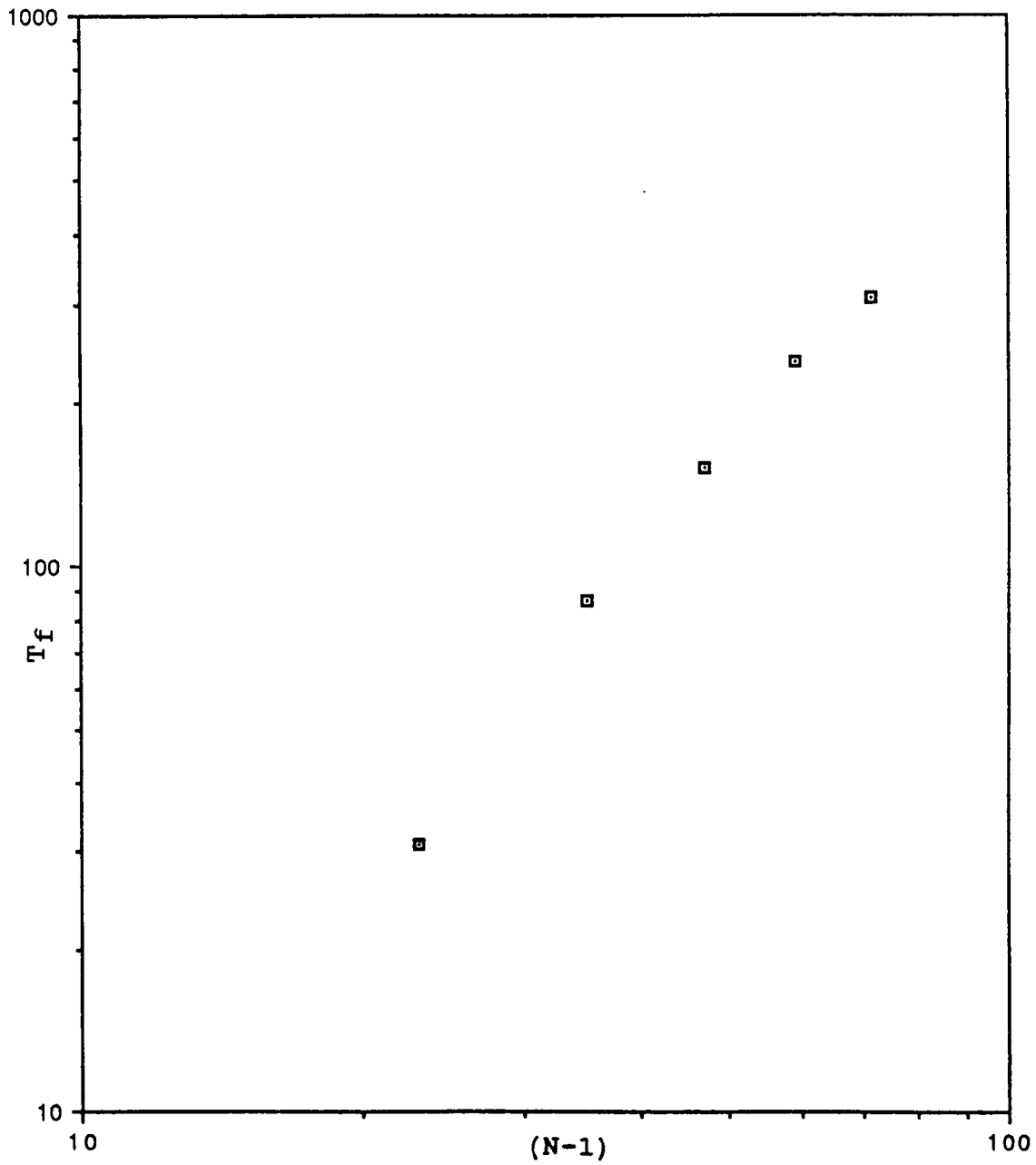

Figure 25. Log-log plot of T_f (f = L_1) versus (N-1) with no excluded volume.

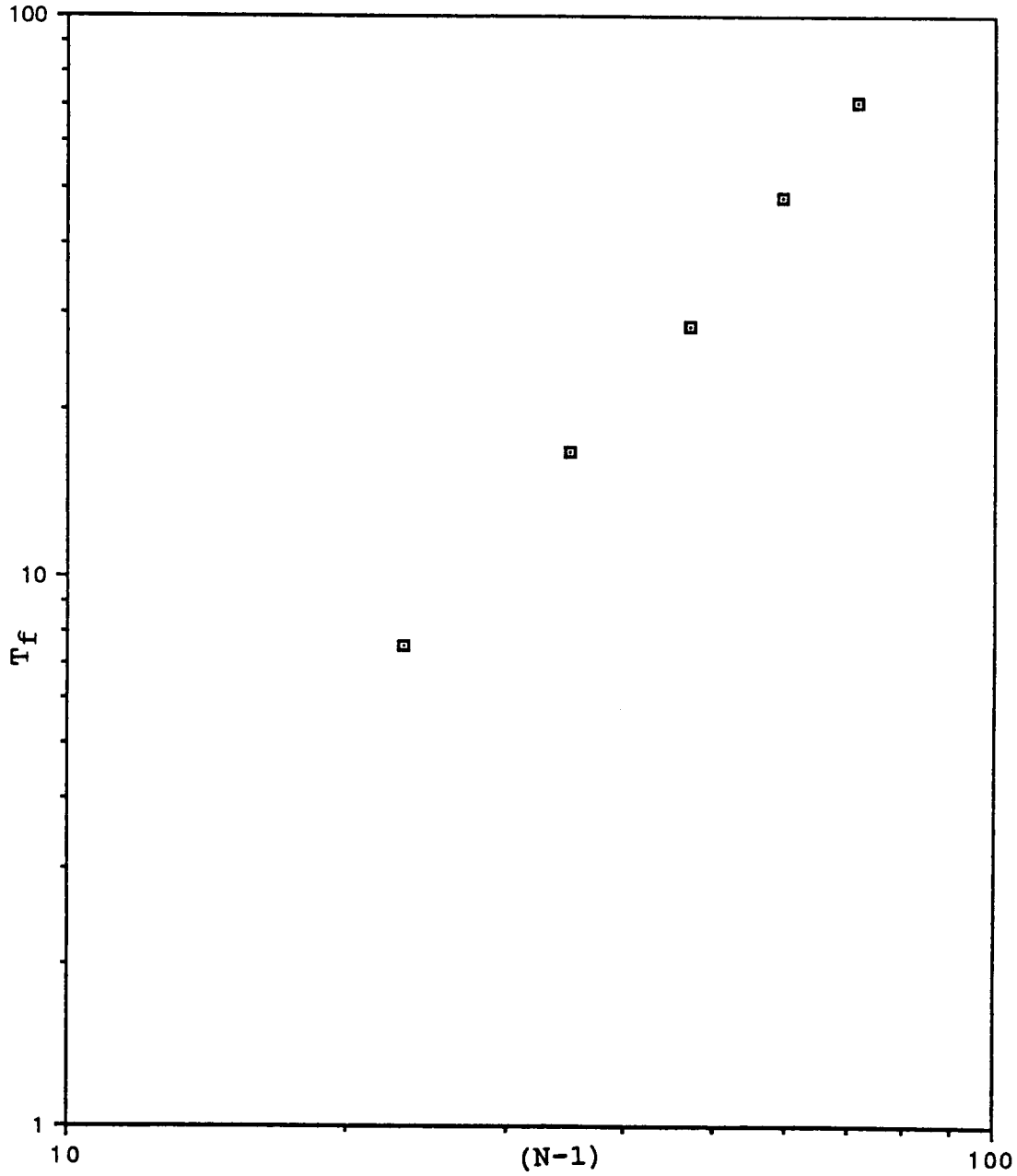

Figure 26. Log-log plot of T_f (f = L_2) versus (N-1) with no excluded volume.

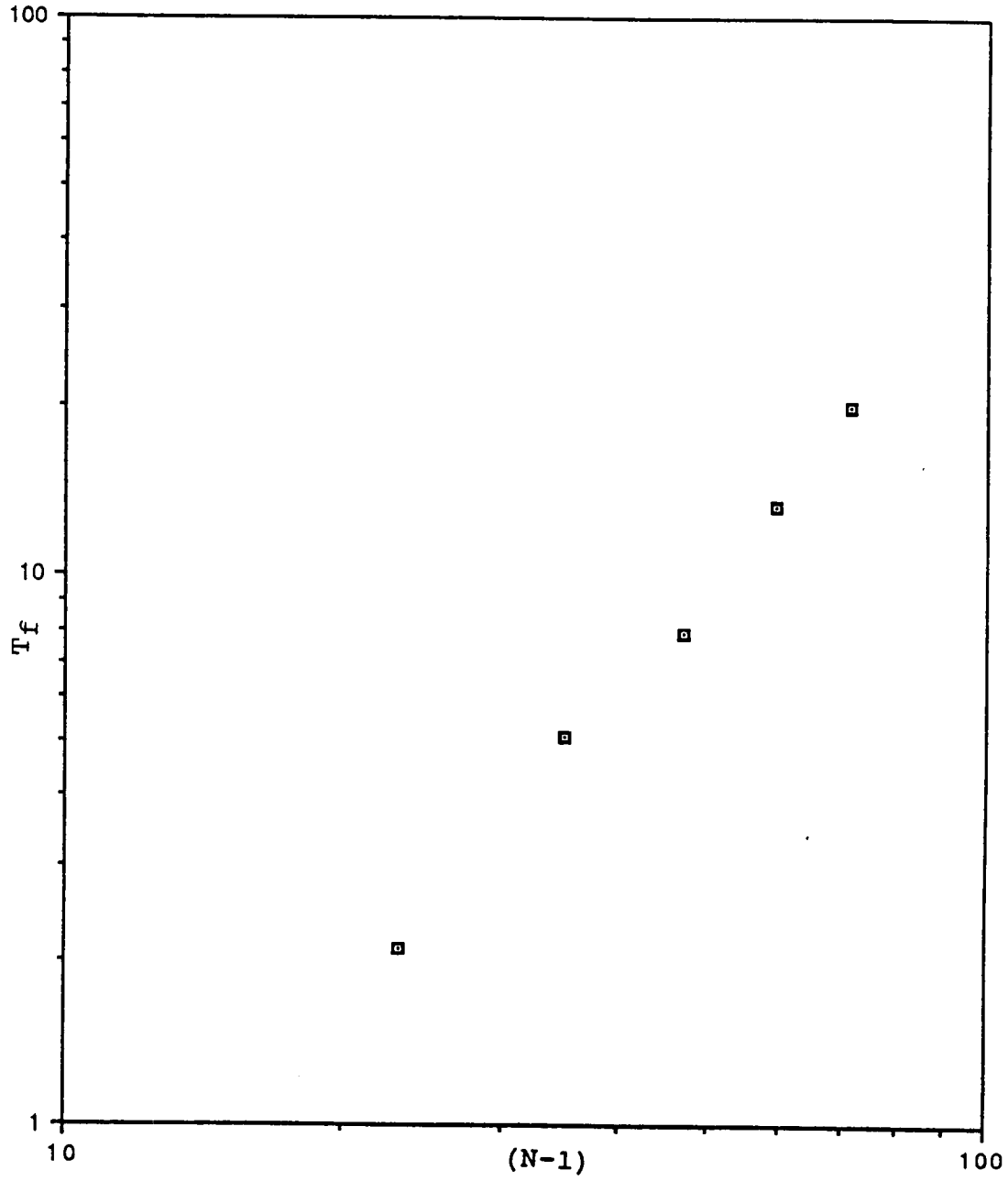

Figure 27. Log-log plot of T_f (f = L_3) versus (N-1) with no excluded volume.

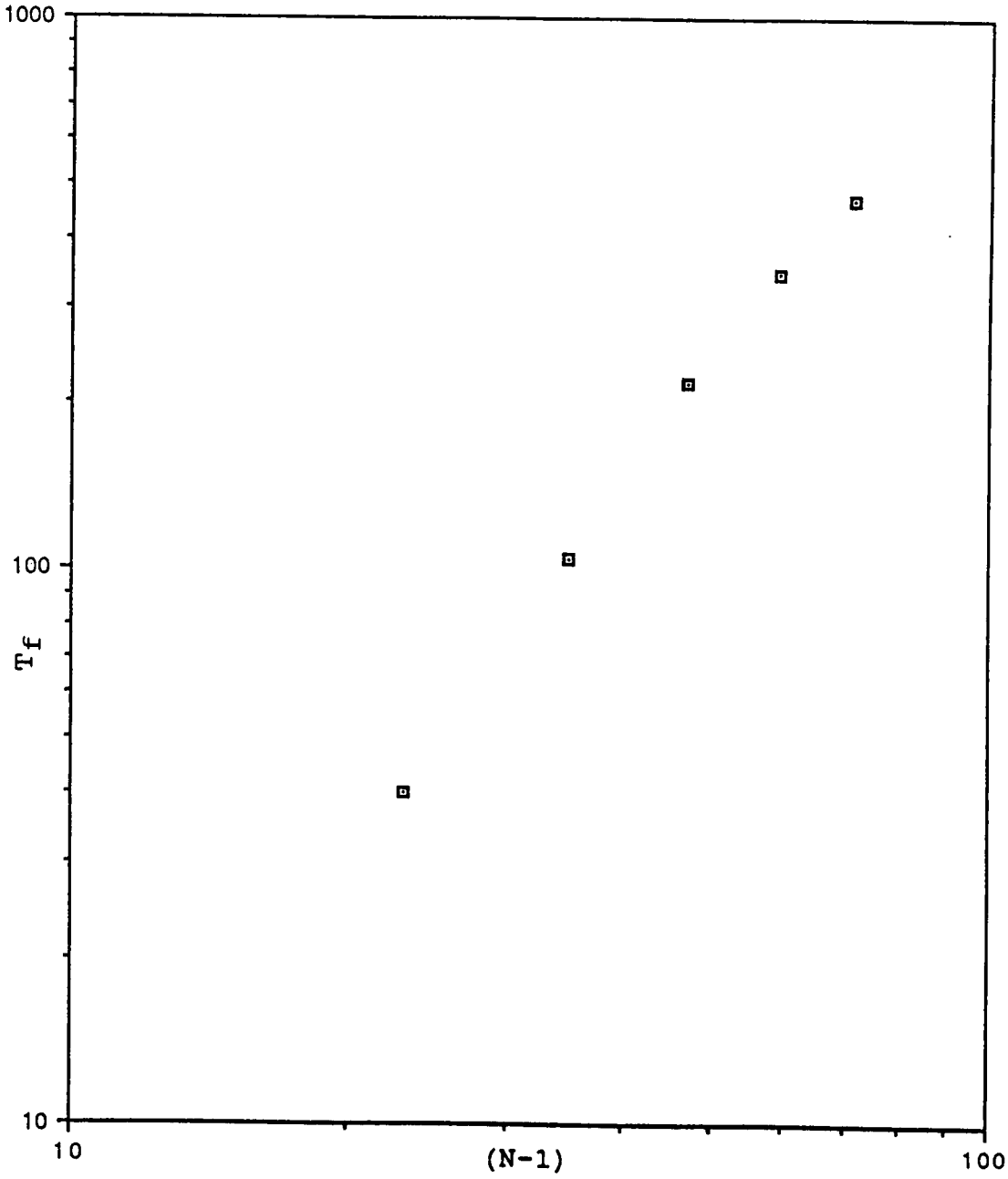

Figure 28. Log-log plot of T_f (f = L_1) versus (N-1) with excluded volume.

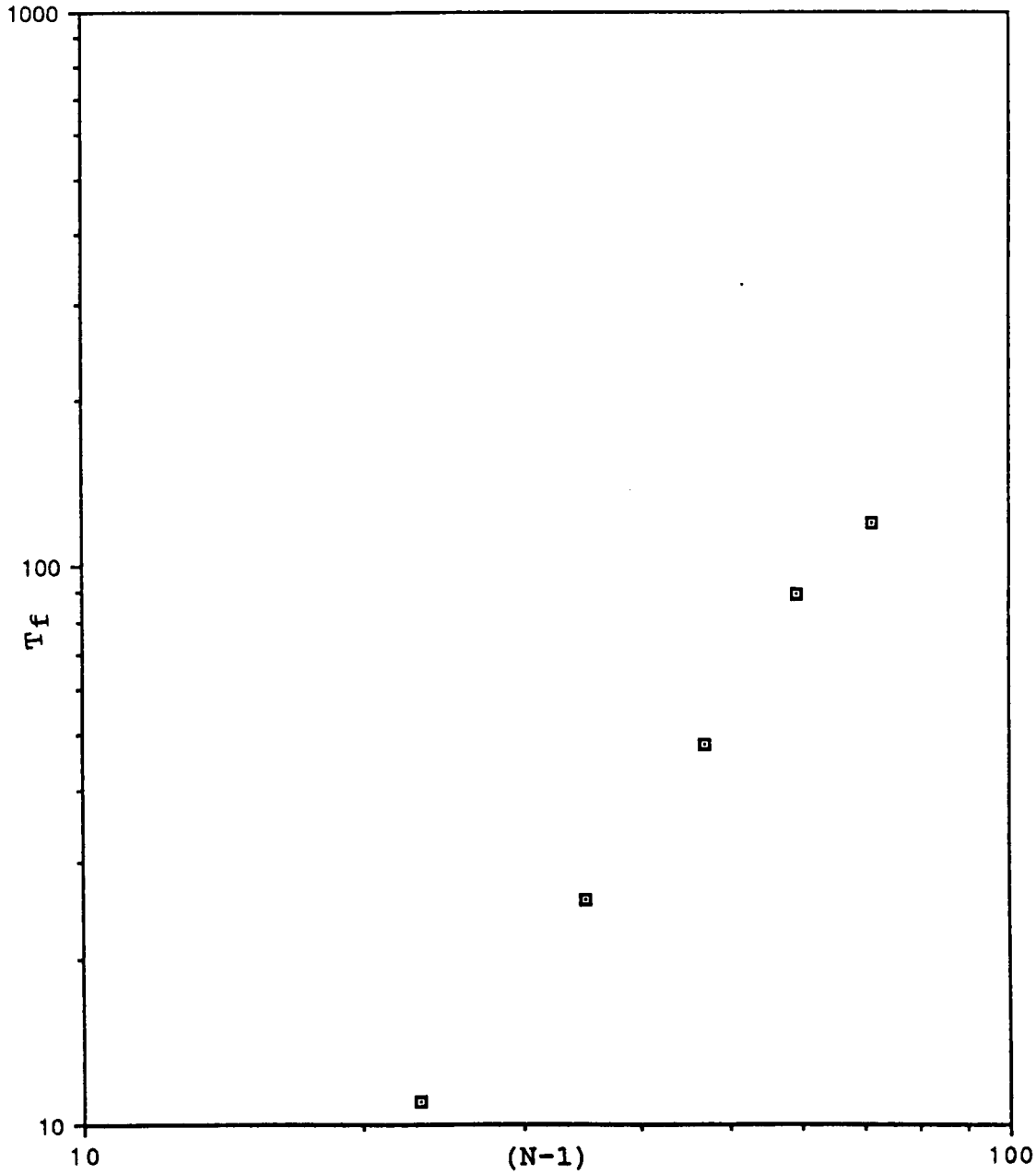

Figure 29. Log-log plot of T_f (f = L_2) versus (N-1) with excluded volume.

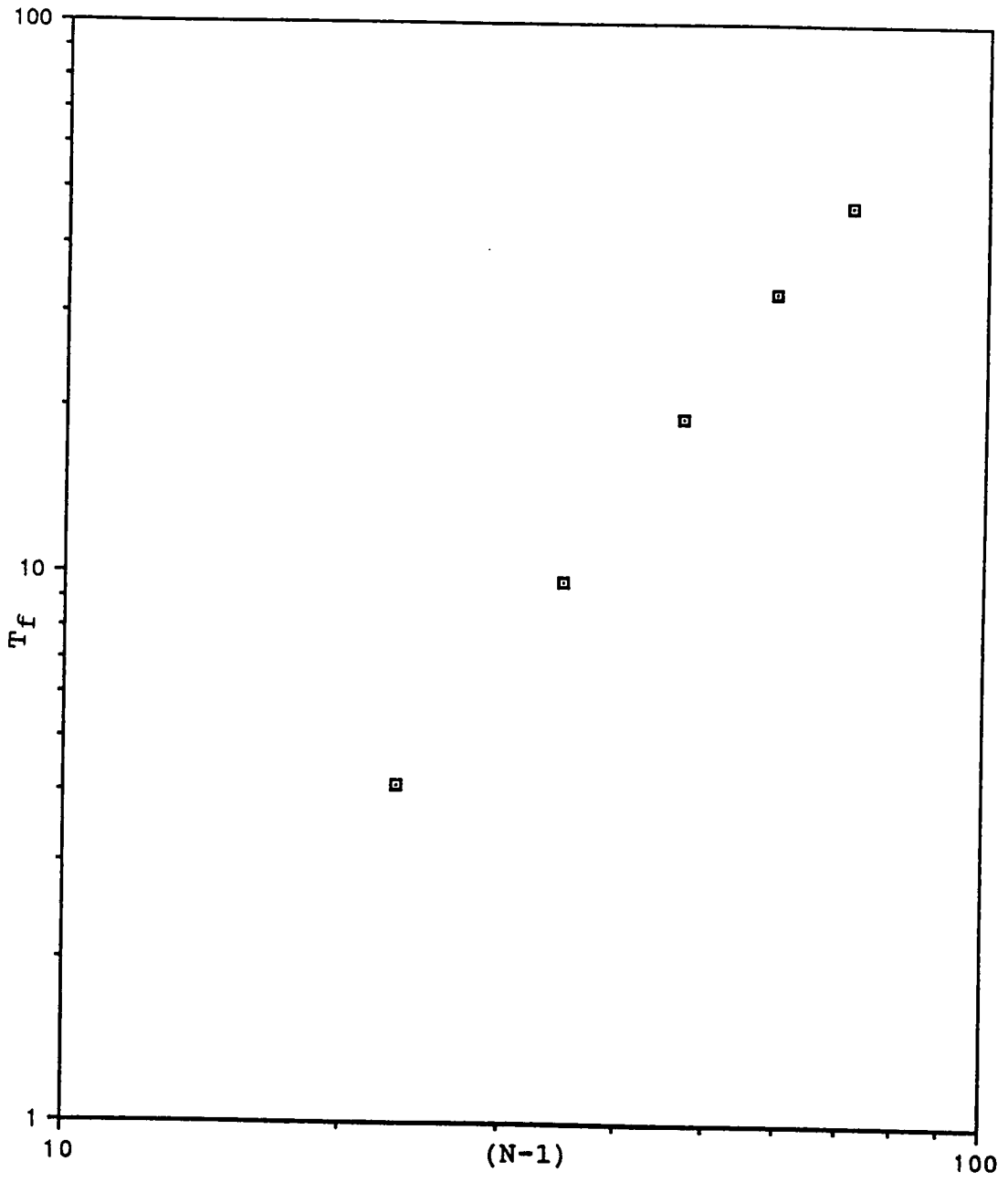

Figure 30. Log-log plot of T_f (f = L_3) versus (N-1) with excluded volume.

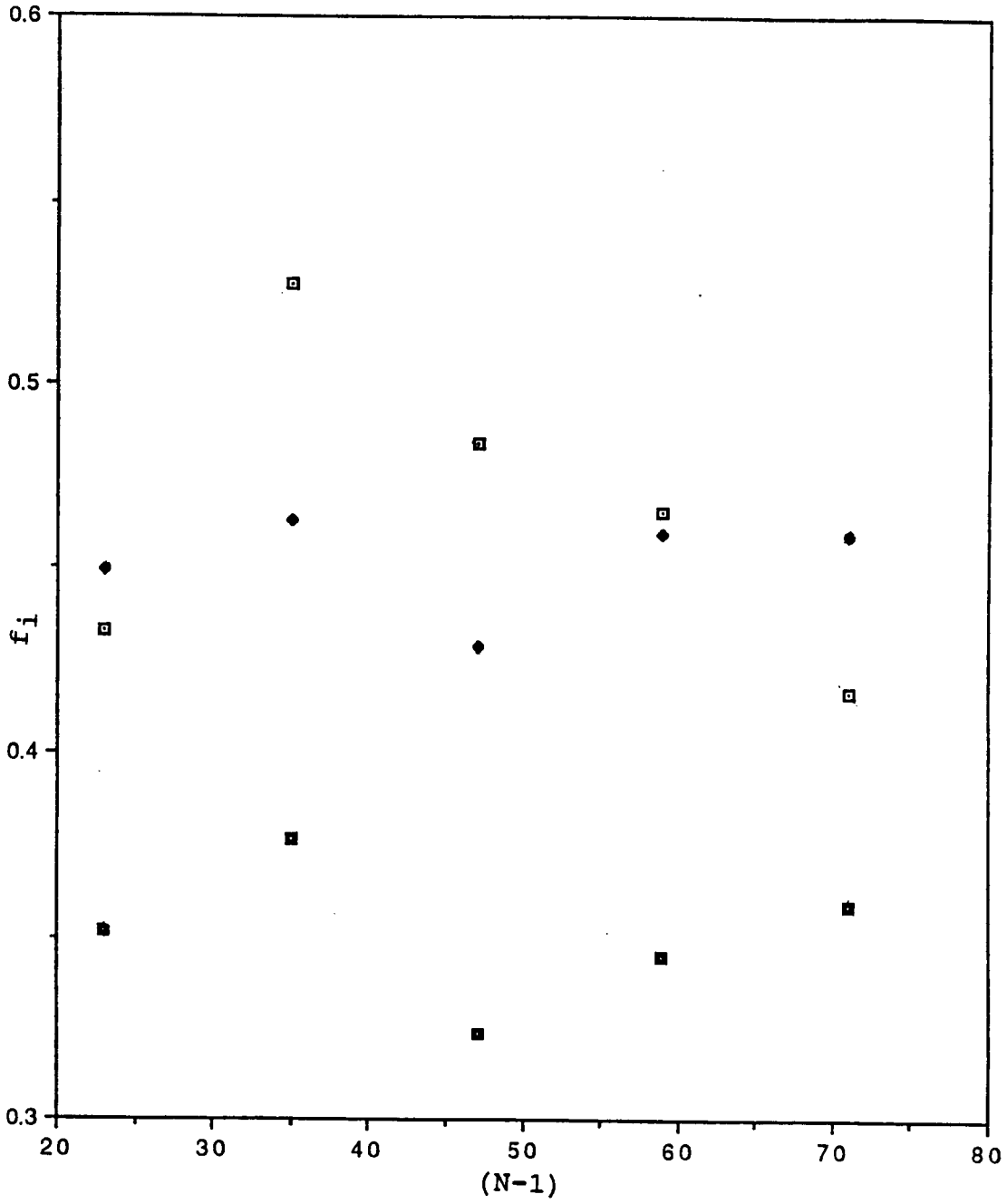

Figure 31. Plot of $f_i = T_{Li}/(<L_i>(N-1))$ versus (N-1) with no excluded volume (i = 1 (▫); 2 (◆); 3 (▪)- standard deviations of .10, .06, and .07 respectively).

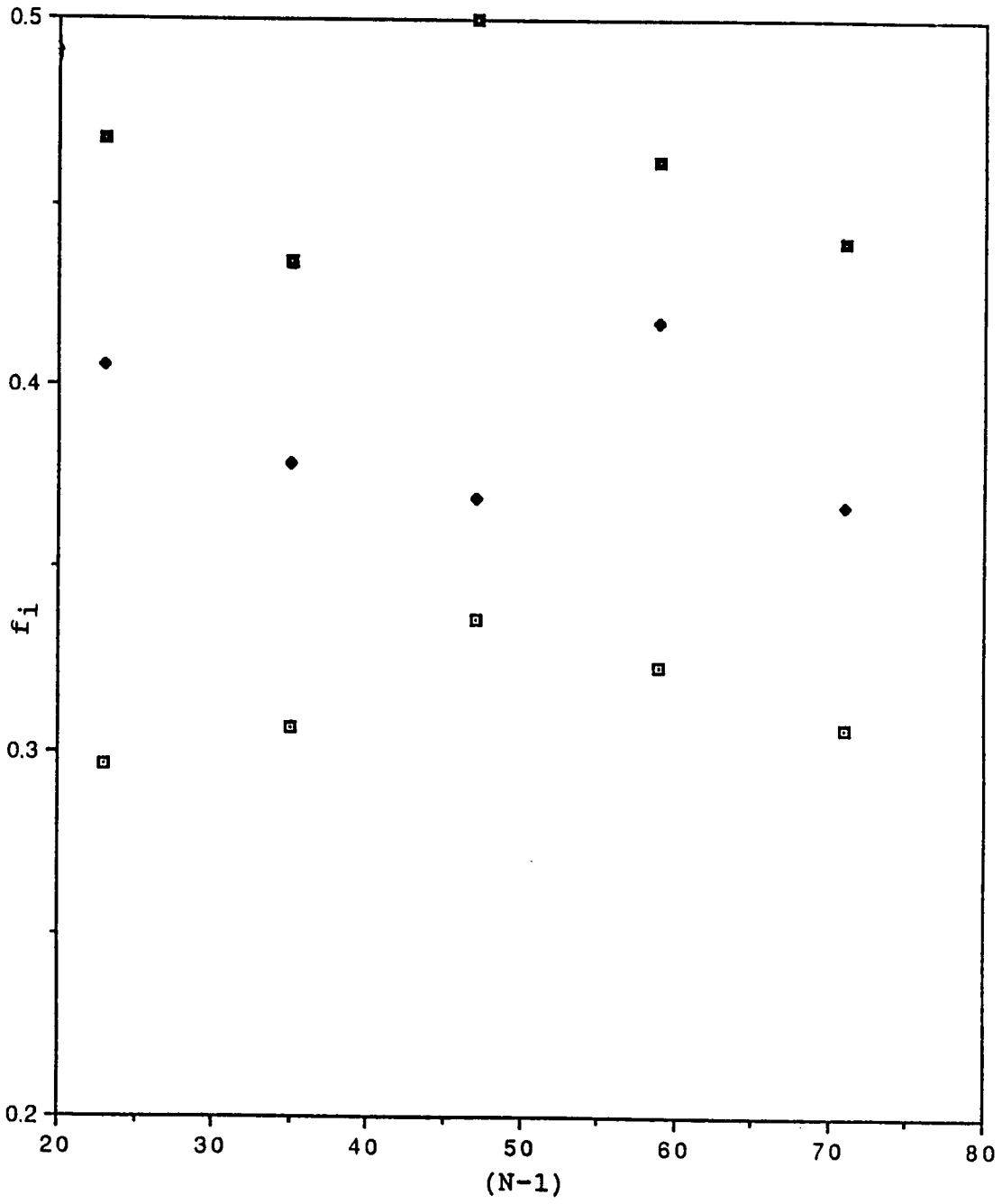

Figure 32. Plot of $f_i = T_{Li}/(<L_i>(N-1))$ versus (N-1) with excluded volume (i = 1 (▫); 2 (♦); 3 (■)-standard deviations of .05, .06, and .05 respectively).

Log-log plots of spherical harmonics relaxation times are plotted in figures 33 through 38. Plots of the ratio T_f/T_g ($f = Y_i^2$ and $g = L_i$) for the three spherical harmonics and the corresponding eigenvalues are in Figures 39 and 40. These tables and figures are discussed below.

The ratios of eigenvalues (Table IV) for the no excluded volume case seem to show slightly greater asymmetry than that found by Solc and Stockmayer[17] or by Kranbuehl, Verdier, and Spencer,[20] but the excluded volume ratios seem to agree very well with those found by Mazur, Gutman and McCrackin.[19] The ratio $\langle L_1\rangle:\langle L_2\rangle:\langle L_3\rangle$, defined such that $L_1 \gtrsim L_2 \gtrsim L_3$ was given by Solc and Stockmayer as 11.8:2.70:1.00 with no excluded volume. This compares to the average ratio 12.71:2.75:1.00 obtained here. All values of $\langle L_1\rangle:\langle L_3\rangle$ listed in this thesis are slightly larger than 11.8 given by Solc and Stockmayer, but the values of $\langle L_2\rangle:\langle L_1\rangle$ they obtained lie within the range of the calculated ratios listed in Table IV. With excluded volume, Mazur et al found the ratio 14.8: 3.1: 1.0. This fits closely to the average ratio in this thesis of 14.9:3.0:1.0. It is important to note that some

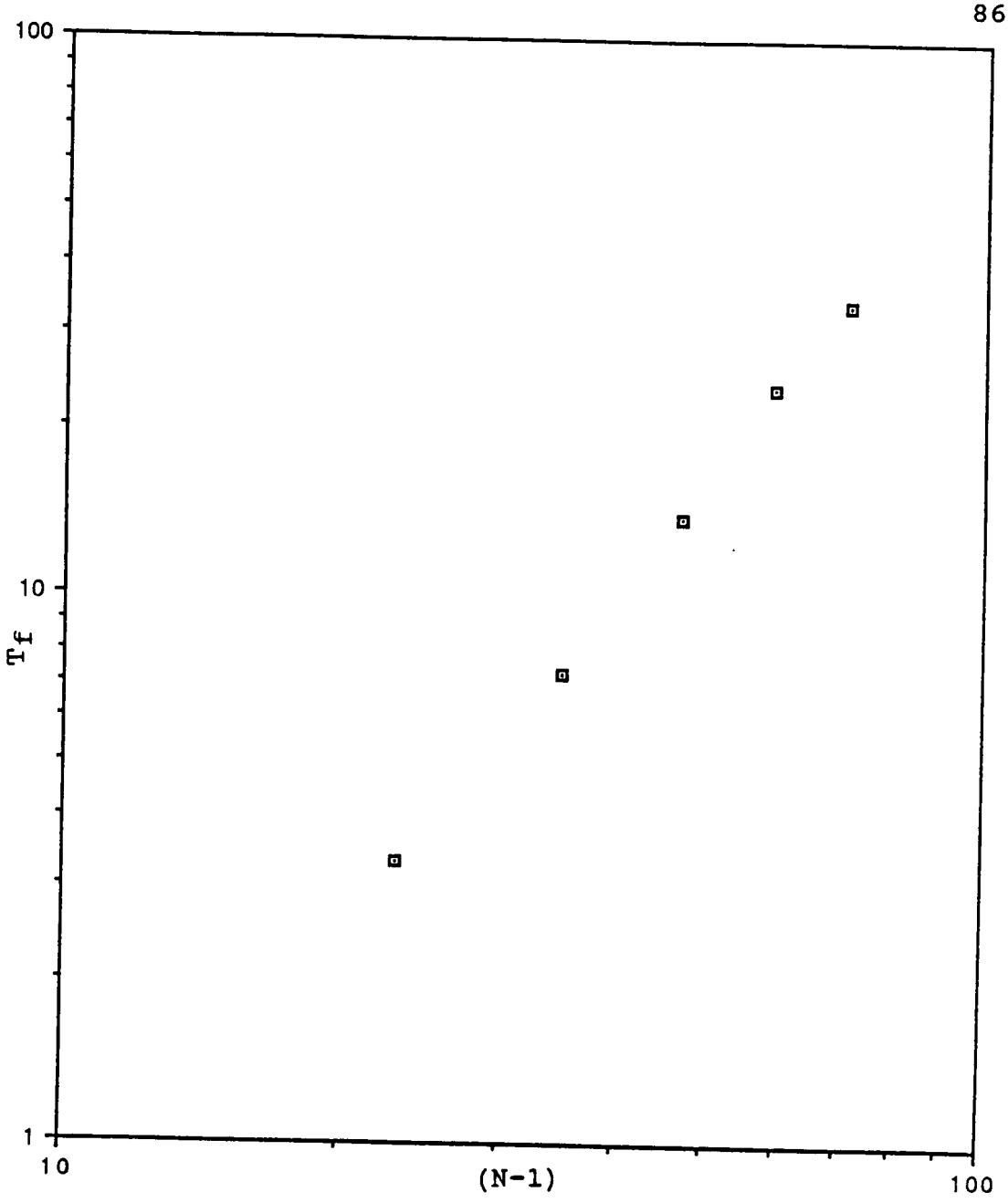

Figure 33. Log-log plot of T_f ($f = Y_1^2$) versus (N-1) with no excluded volume.

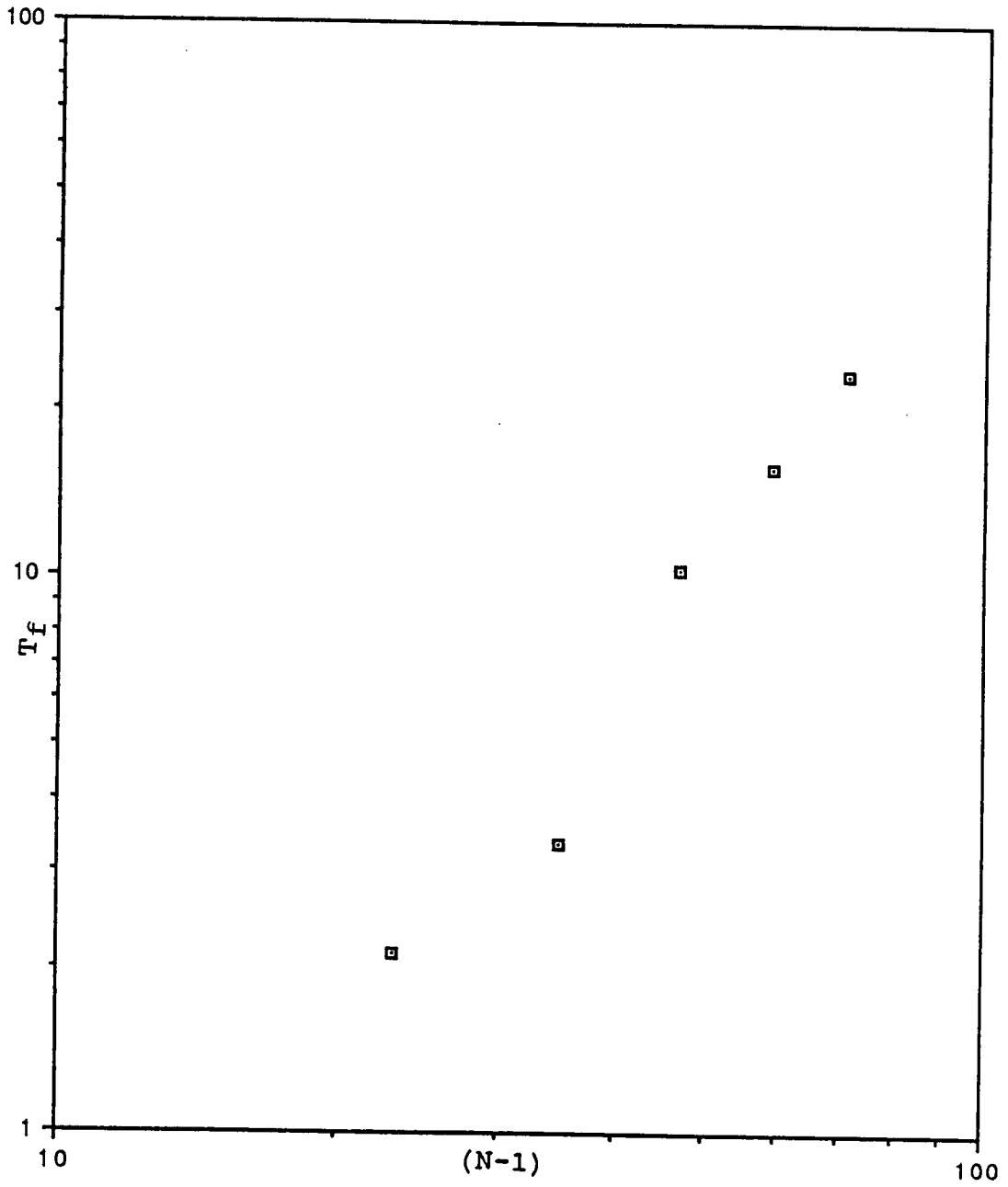

Figure 34. Log-log plot of T_f ($f = Y_2^2$) versus (N-1) with no excluded volume.

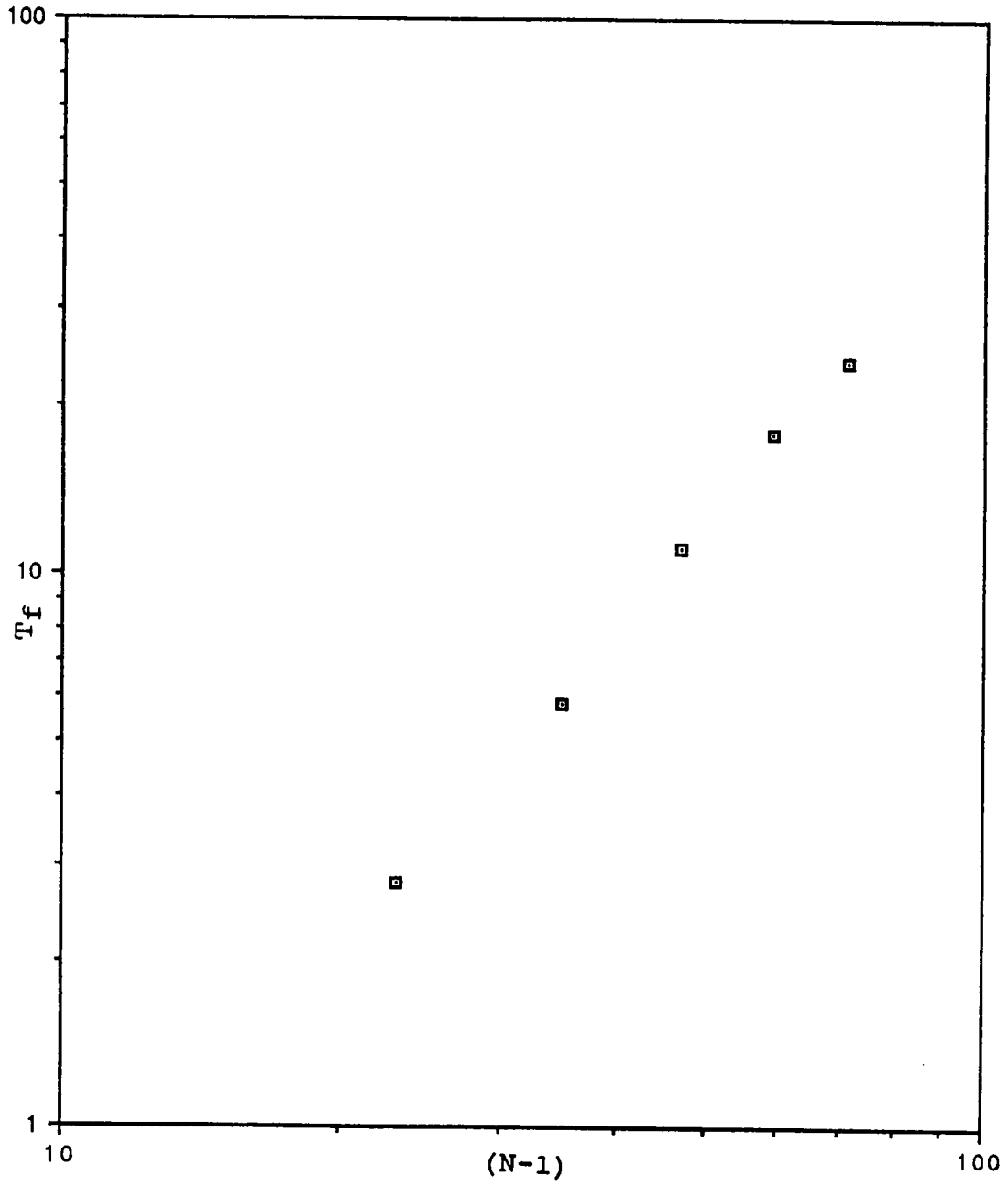

Figure 35. Log-log plot of T_f ($f = Y_3^2$) versus (N-1) with no excluded volume.

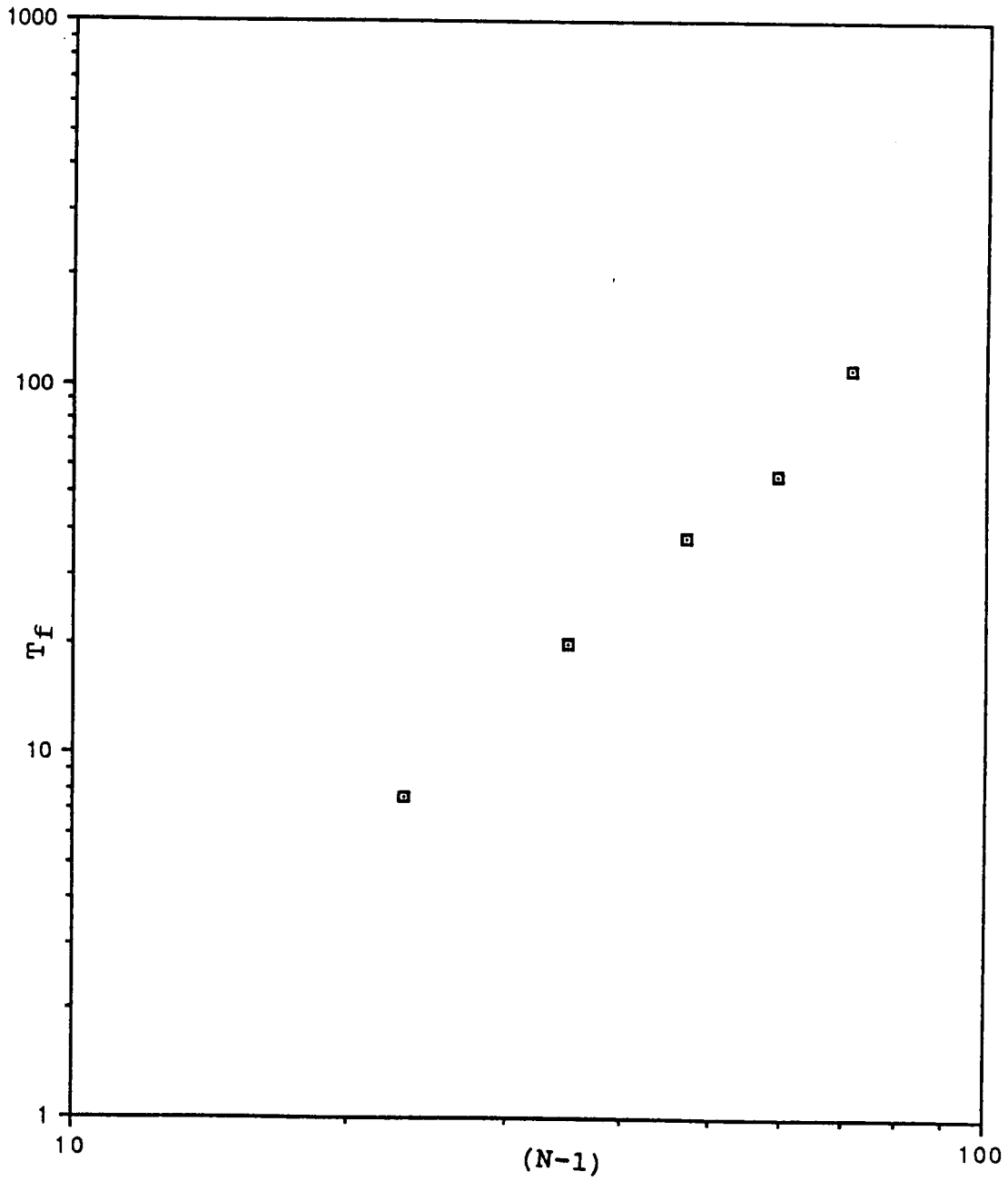

Figure 36. Log-log plot of T_f ($f = Y_1^2$) versus (N-1) with excluded volume.

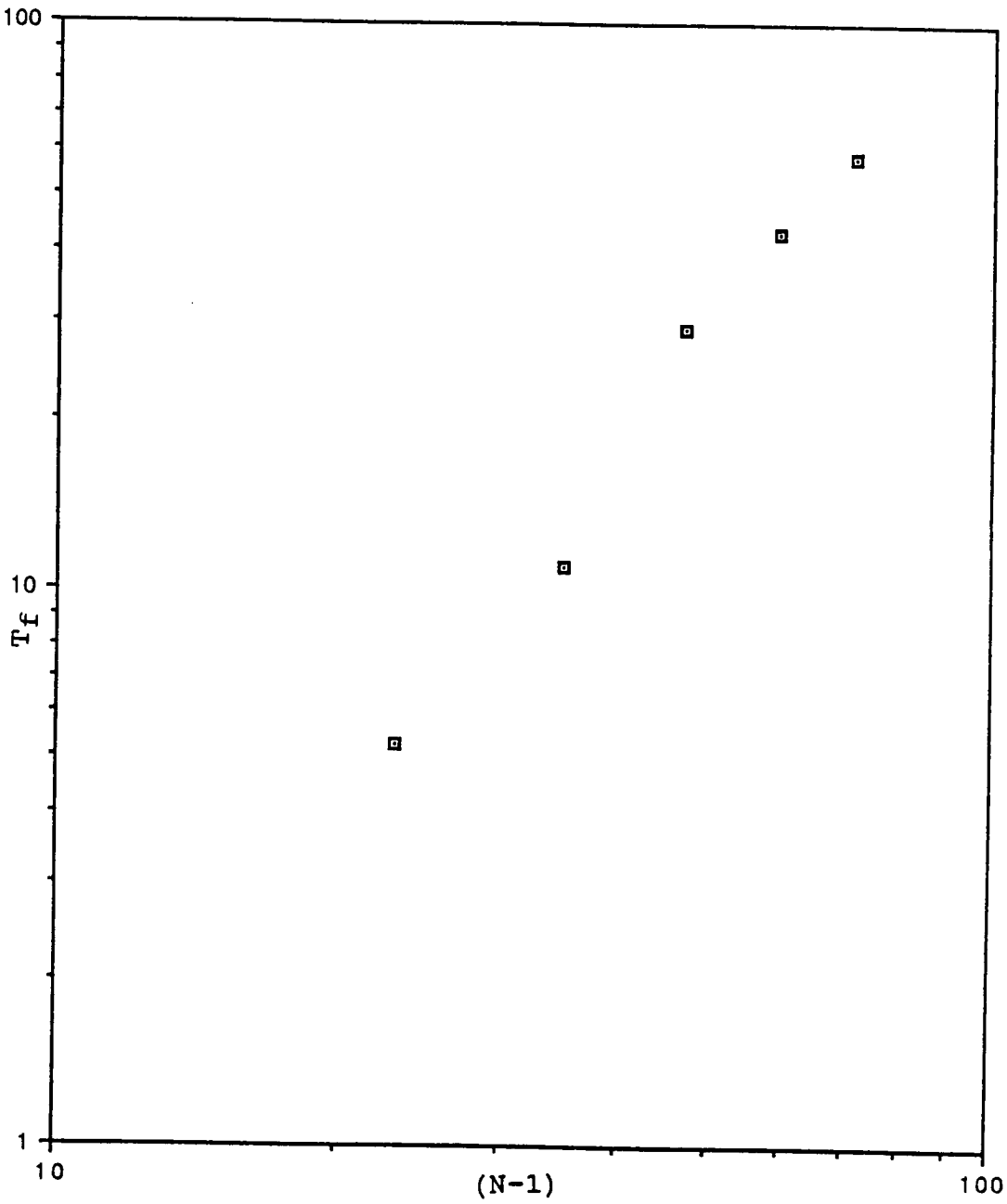

Figure 37. Log-log plot of T_f ($f = Y_2^2$) versus (N-1) with excluded volume.

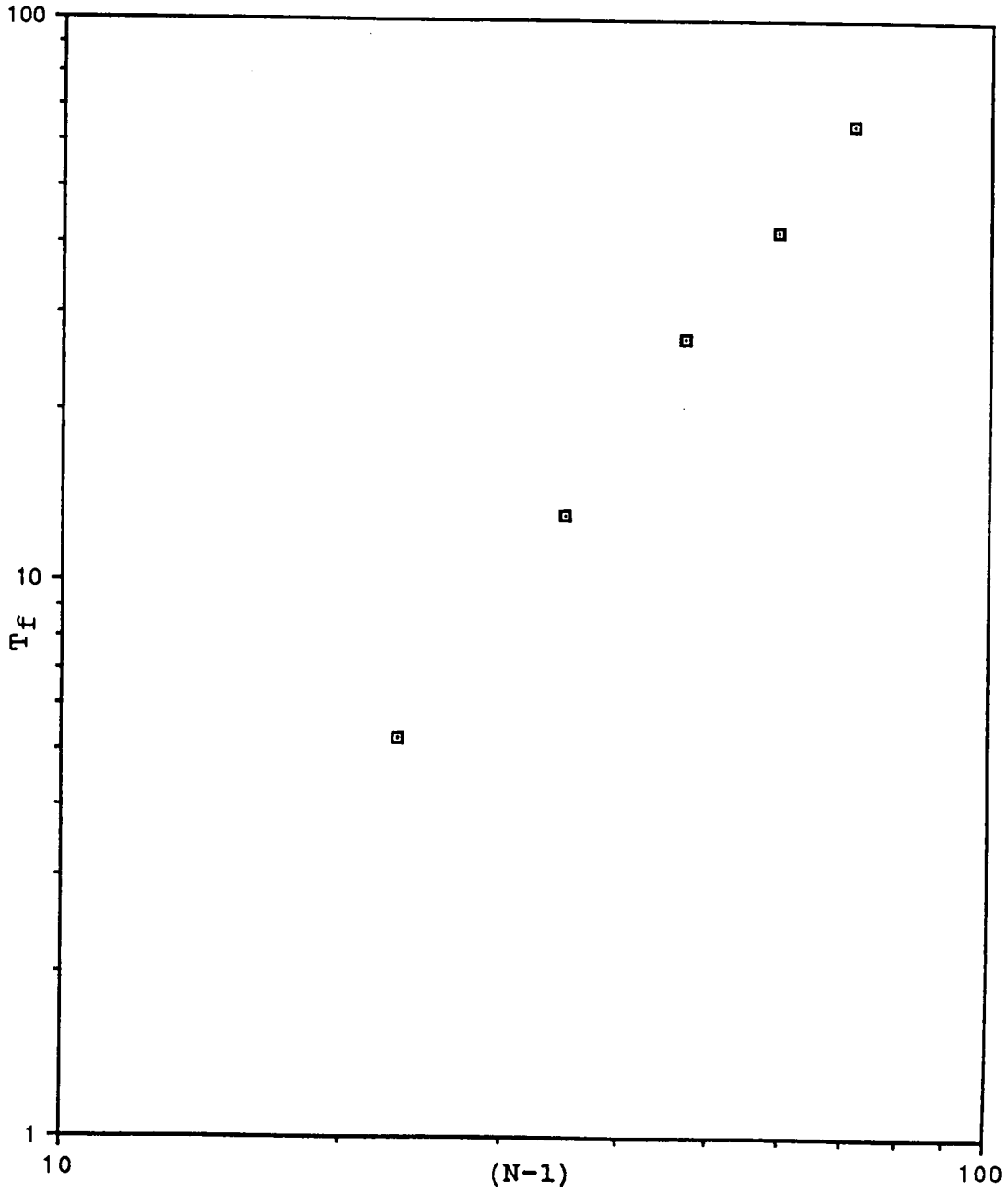

Figure 38. Log-log plot of T_f ($f = Y_3^2$) versus (N-1) with excluded volume.

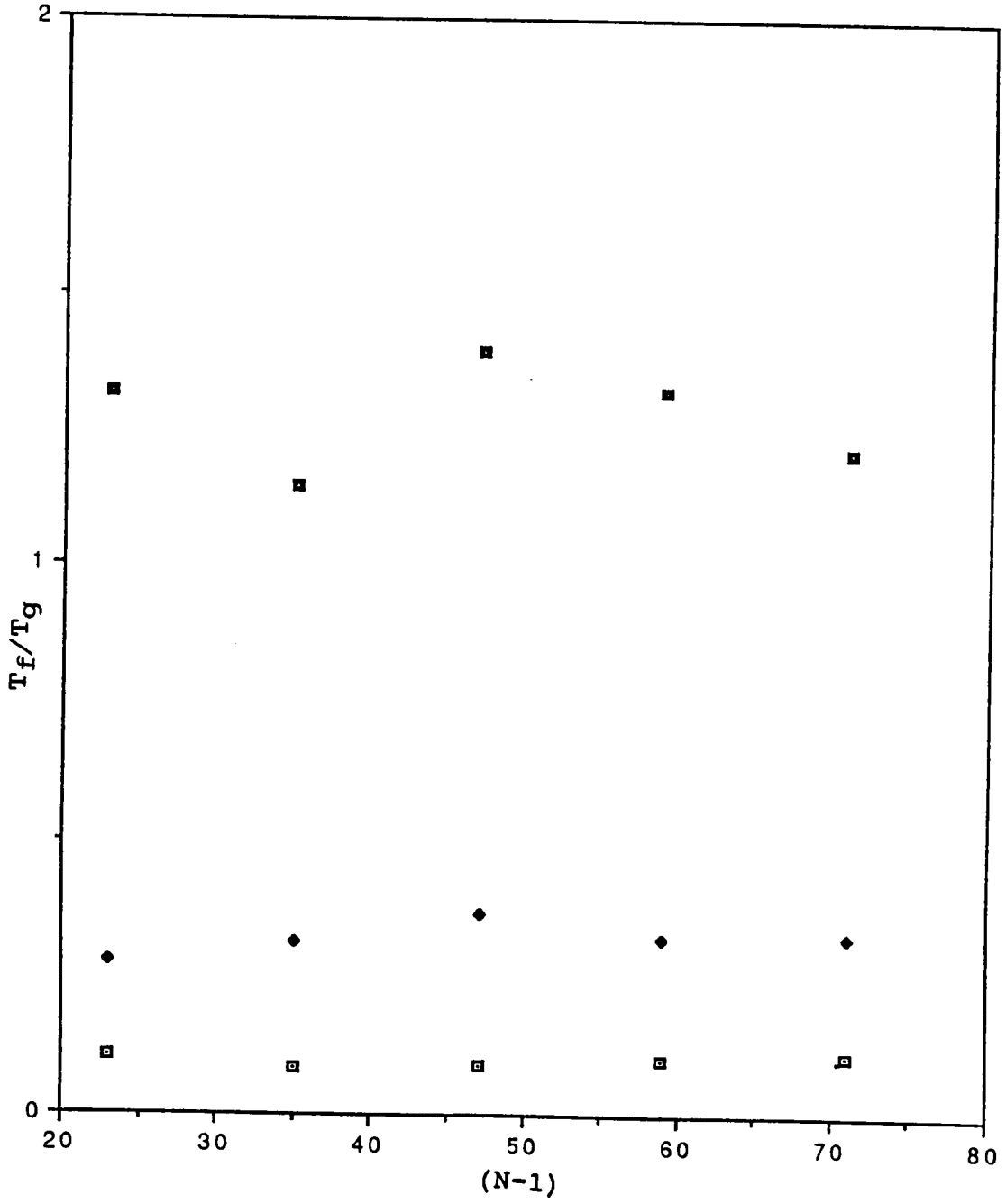

Figure 39. Plot of T_f/T_g ($f = Y_i^2$, $g = L_i$) versus (N-1) with no excluded volume (i = 1 (▫); 2 (◆); 3 (▪)- standard deviations of .02, .07, and .21 respectively).

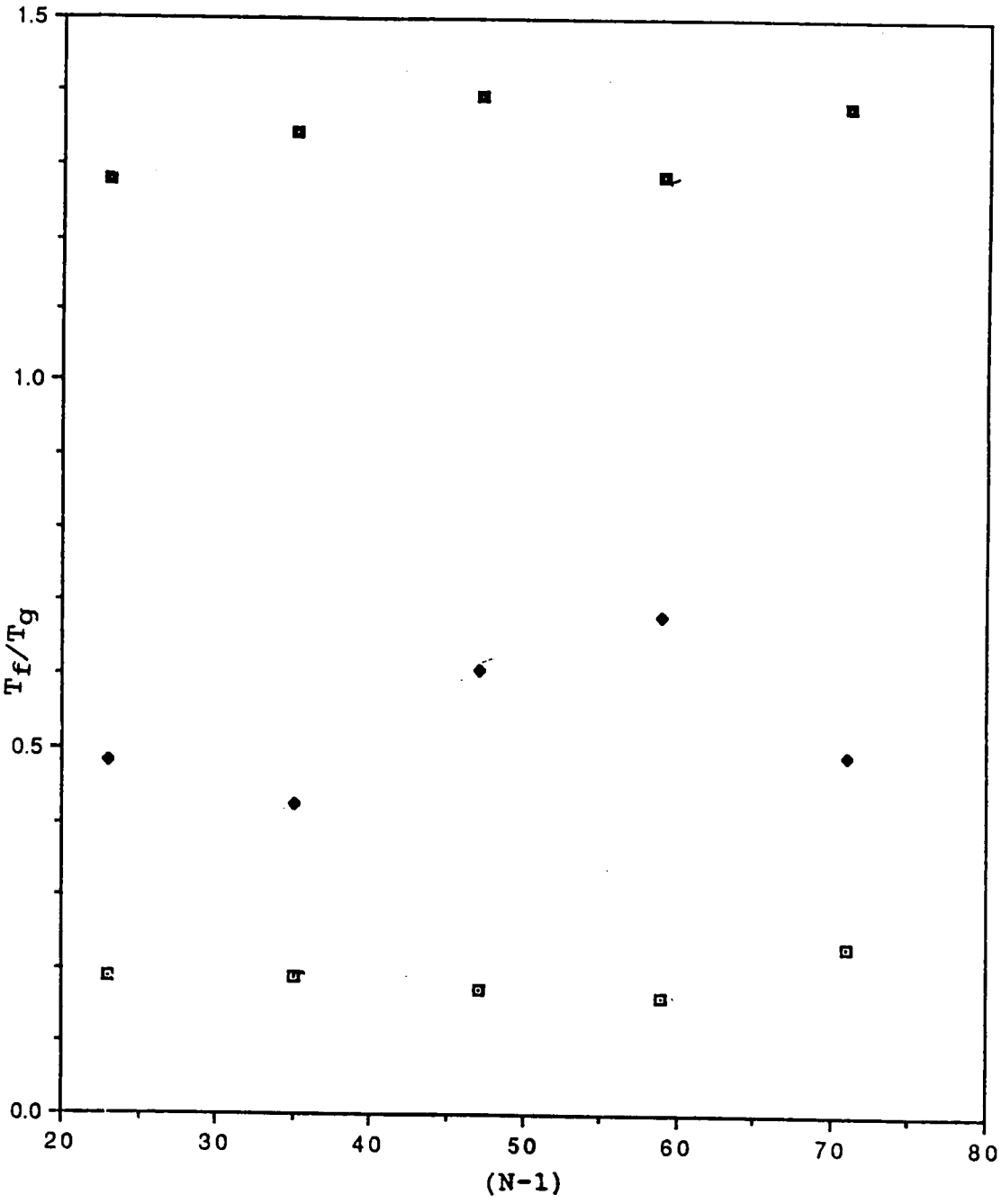

Figure 40. Plot of T_f/T_g ($f = Y_i^2$, $g = L_i$) versus (N-1) with excluded volume (i = 1 (▫); 2 (◆); 3 (■)-standard deviations of .04, .13, and .23 respectively).

distortion of the axes is possible (especially with short chains) due to existence of 180° bonds.

Figures 19 through 24 are log-log plots of $\langle L_i \rangle$ versus chain length (N-1) for each eigenvalue. From these plots the scaling exponents (2ν) for the relationship

$$\langle L_i \rangle \sim (N-1)^{2\nu} \tag{14}$$

are determined. They are found by equating the slope of the log-log plot to 2ν. For the no excluded volume case, 2ν is found to be 1.09, 0.98, and 0.97 for $\langle L_1 \rangle$, $\langle L_2 \rangle$, and $\langle L_3 \rangle$ respectively. With excluded volume it is found to be 1.19, 1.21, and 1.21 respectively. These can be compared to the 2ν values for mean-square end-to-end distances. For the no excluded volume case, the expected value of 2ν is 1.00 and for the excluded volume case 2ν is 6/5.[19] Comparison suggests that the simulation agrees well with theory. Any deviation is probably due to scatter in the data.

As mentioned in Chapter I, the eigenvalue relaxation times are found by doing a linear least squares fit of $\ln p_f(t)$ versus time (t) ($f = L_i$). A

segment of the linear region is chosen for the least squares fit. This region is the interval $-1.0 \leqslant \ln p(t) \leqslant -1.5$. The slope is $(-1/T_f)$. Also mentioned in Chapter I, the three corresponding rotational autocorrelations versus time interval are clearly nonlinear (Figures 7 through 9). Instead of a least squares fit, the relaxation time (T_f for $f = Y_i^2$) is determined as the time at which $p(t) = 1/e$ ($\ln p(t) = -1$). Calculating the relaxation time as $p(t) = 1/e$ will give a shorter time than that expected for a linear least square fit unless the graph is linear from the start of the correlation plot. In the absence of a long time exponential decay, the value of $T_f(1/e)$ does provide a measure of the angular relaxation time of the chain.

Figures 25 through 30 are log-log plots of eigenvalue relaxation times (Tables VI and VII) versus chain length. As expected the largest of the three eigenvalues of the ellipsoid relaxes slowest and the smallest eigenvalue has the fastest relaxation time for each chain length. The chain length dependence of the relaxation time for each eigenvalue is expected to scale as

$$T_f \sim (N-1)^a \qquad (15)$$

($f = L_i$). For the eigenvalues L_i the scaling exponents are found to be 2.05, 1.99, and 1.95 for i = 1, 2, and 3 respectively without excluded volume. With excluded volume they are found to be 2.23, 2.17, and 2.18 respectively. Thus the three eigenvalue relaxation time scaling exponents are approximately 2.0 with no excluded volume and 2.2 with excluded volume as predicted by scaling theory

If the three principal components of the radius of gyration $\langle S^2 \rangle$ are the slowest relaxation processes, they should correspond in a simple manner to the relaxation times of the first three normal modes. To test this hypothesis, the ratios of the corresponding relaxation times (T_f/T_g for $f = L_i$ and $g = i$) are listed in Table VIII. Looking strictly at the ratios for the no excluded volume case would suggest that the three eigenvalues do seem to correspond to the first three normal modes in a simple manner. However, the relationship between the third eigenvalue and the third normal mode seems to differ from the first and second eigenvalues and their corresponding normal modes.

Inspection of the ratios with excluded volume also seems to give a simple relationship between the eigenvalues and the normal modes. The ratios give the approximate relationship $T_{Li} = .25T_i$. This suggests that the three sorted eigenvalues ($L_1>L_2>L_3$) relax in the same manner as the first three normal modes respectively in the presence of excluded volume.

From the data in Tables VI and VII it is obvious that the order of decreasing eigenvalue size is also the order of decreasing eigenvalue relaxation times. The relaxation times, however, do not decrease proportionate to the relative eigenvalue size. This is illustrated by the data in Table IX and the plots in Figures 31 and 32 for the function $f_i = T_{Li}/(<L_i>(N-1))$. As the figures show, the smallest eigenvalue relaxes fastest for its size. The two larger eigenvalues seem to behave in a similar manner. With excluded volume, the order of sizes of f_i is the reverse of the eigenvalues. In other words, the smallest eigenvalue has the largest f_i and the largest eigenvalue has the smallest f_i. This suggests that, within a given chain, excluded volume has a greater effect for the smallest eigenvalue than for the largest eigenvalue relaxation times, if the

relaxation times are reduced by the corresponding eigenvalue. The added influence, however, only appears as a constant, chain length independent effect. This added excluded volume influence can be understood by considering the size of the principal components of the mean square radius of gyration, $\langle S^2 \rangle$. As Table V shows, the smallest of the three principal axes has a radius which is smaller than the minimum distance between two beads (1 lattice unit) for shorter chains with excluded volume and is only on the order of the distance between two beads for the longer chains (N-1 = 59 and 71). This suggests that one class of allowed moves in the unrestricted random walk case, such as beads passing through each other, is much more influential into the third eigenvalue relaxation process than the other two. Cutting off that relaxation process makes the chain much stiffer, resulting in a dramatic, chain length independent, influence on the relaxation time. This effect is likely to be reduced in a lattice system with a larger coordination number corresponding to a more flexible chain.

Chain length dependence of the rotational relaxation times are also of interest. These are

calculated by measuring autocorrelation function values for three functions (Y_1^2, Y_2^2, Y_3^2) which are related in a simple manner to spherical harmonics. Each of these functions Y_1, Y_2, and Y_3 is a measure of the rotation of the three principal axes of the of the ellipsoid with respect to a fixed axis and correspond to L_1, L_2, and L_3 respectively. These functions are:

$$Y_1 = A\sin(\theta)\cos(\theta)\sin(\phi)$$

$$Y_2 = A\sin(\theta)\cos(\theta)\cos(\phi)$$

$$Y_3 = B(3\cos^2(\theta) - 1)$$

with A and B constants and and Euler angles. They relate to spherical harmonics in the following manner:

$$Y_3 = Y_{2,0}(\theta,\phi) \text{ and}$$

Y_1, and Y_2 are linear combinations of the complex functions

$$Y_{2,+1}(\theta,\phi) = C\sin(\theta)\cos(\theta)\exp(\pm i\phi).$$

The spherical harmonics relaxation times are listed in Table VI and VII and are plotted against chain length (N-1) in Figures 33 through 38. As expected, the largest eigenvalue also has the largest rotational relaxation time. The rotational relaxation times for the second and third eigenvalues are, however, very close and possibly the same. It is of interest to

determine the scaling law for the three rotational relaxation times (T_f $(N-1)^a$ for $f = Y_i^2$). The scaling exponents for the three spherical harmonics are determined in this thesis to be 2.12, 2.13, and 1.95 without excluded volume and 2.29, 2.21, and 2.23 with excluded volume. It seems that the largest eigenvalue has a slightly greater rotational chain length dependence than the other two eigenvalues.

Essentially no literature is available for comparison of rotational relaxations, but they can be compared to the eigenvalue relaxations. From the appearance of the relaxation time data, the rotational relaxation times seem to lie within the range of the relaxation times of the second and third eigenvalues. Ratios of the spherical harmonics to eigenvalue relaxation times (T_f/T_g, $f = Y_i^2$ and $g = L_i$) are in Table X and plotted against chain length in Figures 39 and 40. Clearly the rotation of the largest axis is much faster than its breathing motion. The second axis rotation is also faster than its breathing motion, but the difference is not as large as for the largest axis. For the smallest axis, the rotational relaxation process is clearly slower than the breathing relaxation process.

The last process is even more pronounced with excluded volume. This suggests that the fastest relaxations of the three axes would be rotation, rotation, and breathing for axes 1, 2, and 3 respectively.

D. Conclusion and Suggestions for Future work

The average ratio of the three eigenvalues seems to agree well with previous Monte Carlo simulations, although the first eigenvalue seems somewhat enlarged in the no excluded volume case. Each eigenvalue relaxation time seems to scale as the overall chain relaxation time, but chain stiffness seems to have a major influence on the smallest eigenvalue relaxation. The first eigenvalue relaxes rather slowly compared to its principal axis rotation, whereas the second and third eigenvalue relaxation times are somewhat closer to their corresponding rotational relaxations.

As mentioned in Chapter I, the next step is to look at larger and interacting systems. The influence of several chains on each other can be studied as a function of concentration and separation. Ultimately, the goal is to study the anisotropic effects in

concentrated systems and under different solvent interaction conditions.

LIST OF REFERENCES

LIST OF REFERENCES

1. J. P. Downey, C. Crabb, and J. Kovac, Macromolecules, 19, 2202 (1986).

2. K. Solc, J. Chem. Phys., 55(1) 335 (1971).

3. P. Flory, "Principles of Polymer Chemistry", Cornell University Press, N. Y., 1971.

4. M. Lax, J. Chem. Phys., 60(6) (1974).

5. M. Lax, J. Chem. Phys., 60(7) (1974).

6. A. T. Clark, and M. Lal, J. Chem., Faraday Trans. 2, 74(2) 1857 (1978).

7. P. G. de Gennes, Macromolecules, 13, 1069 (1980).

8. T. Cosgrove, Macromolecules, 15, 1290 (1982).

9. I. D. Robb, and R. Smith, Polymer, 18, (1977).

10. I. D. Robb, and R. S. Smith, Eur. Polym. J., 10, 1005 (1974).

11. H. Hommel, A. P. Legrand, H. Balard, and E. Papirer, Polymer, 24, 959 (1984).

12. H. Hommel, A. P. Legrand, P. Tougne, H. Balard, and E. Papirer, Macromolecules, 17, 1578 (1984).

13. K. G. Barnett, T. Cosgrove, B. Vincent, D. S. Sissons, and M. Cohen-Stuart, Macromolecules, 14, 1018 (1981).

14. H. Hommel, L. Facchini, A. P. Legrand, and J. Lecourtier, Eur. Polym. J., 14, 803 (1978).

15. H. Hommel, A. P. Legrand, J. Lecourtier, and J. Desbarres, Eur. Polym. J., 15, 993 (1979).

16. C. Domb, and M. E. Fisher, Proc. Cambridge Philos. Soc. 54, 48 (1957).

17. K. Solc, and W. H. Stockmayer, J. Chem. Phys., 54, 2756, (1971).

18. K. Solc, Macromolecules, 6, 378 (1973).

19. J. Mazur, C. M. Guttman, and F. C, McCrackin, Macromolecules, 6, 858 (1973).

20. D. E. Kranbuehl, P. H. Verdier, and J. M. Spencer, J. Chem. Phys., 59(7) (1973).

21. R. J. Rubin, and J. Mazur, J. Chem. Phys., 63(12) 5362 (1975).

22. R.J. Rubin, and J. Mazur, Macromolecules, 10, 139 (1977).

23. T. Minato, and A. Hatano, Macromolecules, 11, 195, (1978).

24. T. Minato, and A. Hatano, Macromolecules, 11, 200 (1978).

25. M. Bishop, and J. H. R. Clarke, J. Chem. Phys., 90(11) 6647 (1989).

26. W. H. Stockmayer, A. A. Jones, and T. L. Treadwell, Macromolecules, 9, 762 (19760).

27. K. Matsuo, K. F. Kuhlmann, H. W. H. Yang, F. Geny, W. H. Stockmayer, and A. A. Jones, J. Polym. Sci., Polym. Phys. Ed., 15, 1347 (1977).

28. K. Matsuo, and W. H. Stockmayer, Macromolecules, 14, 544 (1981).

29. K. Matsuo, W. H. Stockmayer, and S. Mashimo, Macromolecules, 15, 606 (1982).

30. W. H. Stockmayer, and K. Matsuo, Macromolecules, 5 766 (1972).

31. D. R. Bauer, J. I. Brauman, and R. Pecora, Macromolecules, 8, 443 (1975).

32. B. H. Zimm, J. Chem. Phys., 24(2) 269 (1955).

33. S. M. Aharoni, Macromolecules, 11, (1978).

34. M. Gurler, C. Crabb, D. Dahlin, and J. Kovac, Macromolecules, 16, 398 (1983).

35. M. Dial, K. S. Crabb, C. Crabb, and J. Kovac, Macromolecules, 18, 2215, (1985).

APPENDIXES

APPENDIX I

PROGRAMS USED FOR TERMINALLY ATTACHED CHAIN SIMULATIONS

Included in this appendix are the programs used to simulate a polymer chain end adsorbed on a surface. The first two programs, FCRX.FOR and FCFX.FOR, are the simulation programs for the no excluded volume and the excluded volume cases respectively. The third program, FCCAUTO.FOR, calculates the mean square end-to-end distance and the autocorrelation function values used in the last program. The last program listed, LINSIM.FOR, calculates the relaxation time for the end-to-end vector and outputs it along with the mean square end-to-end distance.

```
@PROCESS DC(CMN1)
C ************************************************
C        PROGRAM NAME        FCRX.FOR
C
C       SIMULATES A SINGLE POLYMER CHAIN FACE
C       CENTERED CUBIC LATTICE, END ADSORBED ON A
C       FLAT IMPENETRABLE SURFACE.  NO EXCLUDED
C       VOLUME IS ACCOUNTED.
C
C ************************************************
      LOGICAL*1 HOME(100,100,100)
      INTEGER RCORX(26000,100), RCORY(26000,100)
      INTEGER RCORZ(26000,100)
      REAL*4 RND(10000)
      INTEGER R1X(12),R1Y(12),R1Z(12)
      INTEGER R2X(12),R2Y(12),R2Z(12),RDATX(12)
      INTEGER RDATZ(12),RXX(12),RYY(12),RZZ(12)
      INTEGER IPOINT,ISEED,LENBUF,SAMPLE, SMPMAX
      INTEGER INTVAL, BEAD, LENGTH,LENGTH1,RX,RY,RZ
C ************************************************
      INTEGER SAMPLE1, RDATY(12), RX1, RY1, RZ1
      INTEGER RX2,RY2,RZ2,RX3,RY3,RZ3,LIMIT,STEP
      INTEGER BEAD1, BEAD2, XIND, YIND, ZIND, ANGLE
      INTEGER COUNT, COUNT1, LINTVAL,ISTEP, BD
      LOGICAL*1 SAME
      DATA HOME/1000000*.FALSE./
      DATA RDATX/1,0,0,1,1,0,-1,0,0,-1,-1,0/
      DATA RDATY/0,1,0,0,-1,1,0,-1,0,0,1,-1/
      DATA RDATZ/0,0,1,-1,0,-1,0,0,-1,1,0,1/
      DATA LIMIT/100/
      DATA RXX/1,1,2,2,2,2,2,5*0/
      DATA RYY/1,2,2,2,2,3,2,5*0/
      DATA RZZ/1,1,1,2,3,3,4,5*0/
      COMMON RND, /BUFFER/ LENBUF
      COMMON /CMN1/ RCORX,RCORY,RCORZ
      SQUARE(IA,IB,IC)=IA*(IA+IB+IC)+IB*(IB+IC)+IC*IC
C ************************************************
C       INITIALIZING THE RANDOM NUMBER ROUTINE RAND
        LENBUF = 10000
        CALL INIT(IPOINT)
C ************************************************
C        READING CHAIN COORDINATES
C
      READ (10,*) INTVAL, SMPMAX, LENGTH, ISTEP
      READ (10,*) IDIV  , ICHAR
      IF(ICHAR.NE.1)WRITE(6,*)'ICHAR IS WRONG VALUE!'
        LENGTH1 = LENGTH - 1
```

```
          DO 200 BD = 1,LENGTH
         READ(10,*)RCORX(1,BD),RCORY(1,BD),RCORZ(1,BD)
C ************************************************
C      AN EXPLANATION OF THE NEXT NINE LINES CAN BE
C      FOUND WHERE THEY ARE REPEATED BELOW.
C ************************************************
       RX = RCORX(1,BD)
       RY = RCORY(1,BD)
       RZ = RCORZ(1,BD)
       RX = MOD(RX,LIMIT)
       IF (RX .LE. 0)RX = RX + LIMIT
       RY = MOD(RY,LIMIT)
       IF (RY .LE. 0)RY = RY + LIMIT
       RZ = MOD(RZ,LIMIT)
       IF (RZ .LE. 0)RZ = RZ + LIMIT
 200       CONTINUE
C ************************************************
C      THE NEXT FEW LINES CREATE A "WALL" OR SURFACE
C      WHICH THE POLYMER CHAIN CANNOT PENETRATE, AND
C      ONTO WHICH THE END OF THE CHAIN IS ADSORBED.
C ************************************************
         RX = 1
         DO 300 RZ = 1, LIMIT
         DO 400 RY = 1, LIMIT
           HOME(RX,RY,RZ) = .TRUE.
 400     CONTINUE
 300     CONTINUE
C ************************************************
C         MOVEMENT     LOOP
C
C
C ************************************************
       LINTVAL = LENGTH1*INTVAL
       DO 500 SAMPLE = 1, SMPMAX
            DO 600 STEP = 1, LINTVAL
       BEAD = RAND(IPOINT)*LENGTH1 + 2
       RX = RCORX(SAMPLE,BEAD)
       RY = RCORY(SAMPLE,BEAD)
       RZ = RCORZ(SAMPLE,BEAD)
C ************************************************
C       NOTE R?3 IS USED AT THE END OF THE LOOP TO
C       CHANGE HOME TO .FALSE.  IF THE "BEAD" IS
C       MOVED SUCCESSFULLY.
C       MOD(A,B) GIVES THE REMAINDER OF A/B.
C       IF A < OR = 0, THEN MOD(A,B) WILL BE
C      < OR = 0. IF A IS A MULTIPLE B, MOD(A,B) = 0.
C       IN EITHER CASE, THE VALUE OF R?3 NEEDS TO BE
C       PLACED IN THE RANGE FROM 1 TO LIMIT. THE
```

```
C      NEXT SIX STEPS WILL DO JUST THAT.
C ************************************************
      RX3 = MOD(RX,LIMIT)
      IF (RX3 .LE. 0)RX3 = RX3 + LIMIT
      RY3 = MOD(RY,LIMIT)
      IF (RY3 .LE. 0)RY3 = RY3 + LIMIT
      RZ3 = MOD(RZ,LIMIT)
      IF (RZ3 .LE. 0)RZ3 = RZ3 + LIMIT
C ************************************************
        IF(BEAD .NE. 1 .AND. BEAD .NE. LENGTH) THEN
      BEAD1 = BEAD - 1
      BEAD2 = BEAD + 1
C ************************************************
C
C     NOTE: IN A FCC LATTICE, THE 3 AXES ARE AT 60
C     DEGREES TO EACH OTHER.  THIS ALLOWS FOR 12
C     POSSIBLE NEAREST NEIGHBORS.
C     THE DISTANCE SQUARED BETWEEN BEAD1 AND BEAD2
C     (BOTH BONDED TO THE BEAD TO BE MOVED) IS
C     ALWAYS AN INTEGER AND HAS THE FOLLOWING
C     VALUES:
C       1        60 DEGREE BOND ANGLE
C       2        90 DEGREE BOND ANGLE
C       3       120 DEGREE BOND ANGLE
C       4       180 DEGREE BOND ANGLE.
C     THE FORMULA TO CALCULATED THE DISTANCE
C     IS GIVEN BY THE FUNCTION SQUARE() BELOW.
C ************************************************
      RX1 = RCORX(SAMPLE,BEAD1)
      RX2 = RCORX(SAMPLE,BEAD2)
      RY1 = RCORY(SAMPLE,BEAD1)
      RY2 = RCORY(SAMPLE,BEAD2)
      RZ1 = RCORZ(SAMPLE,BEAD1)
      RZ2 = RCORZ(SAMPLE,BEAD2)
        XIND = RX2 - RX1
        YIND = RY2 - RY1
        ZIND = RZ2 - RZ1
      ANGLE = SQUARE(XIND,YIND,ZIND)
        GOTO (510,510,510,600) ANGLE
        IF (ANGLE.EQ.0) GOTO 555
        WRITE(11,*)' ERROR IN COMPUTED GOTO!'
        WRITE(11,*) ANGLE
        WRITE(11,*)'BEAD1  ',BEAD1,'BEAD2  ',BEAD2
        WRITE(11,*) RX1,RY1,RZ1
        WRITE(11,*) RX2,RY2,RZ2
        STOP
510       DO 520 I = 1,12
        R1X(I) = RX1 + RDATX(I)
```

```
          R2X(I) = RX2 + RDATX(I)
          R1Y(I) = RY1 + RDATY(I)
          R2Y(I) = RY2 + RDATY(I)
          R1Z(I) = RZ1 + RDATZ(I)
          R2Z(I) = RZ2 + RDATZ(I)
 520        CONTINUE
          COUNT = 0
            DO 540 I = 1,12
          DO 530 J = 1,12
          IF(R1X(I).EQ.R2X(J))THEN
            IF(R1Y(I).EQ.R2Y(J))THEN
              IF(R1Z(I).EQ.R2Z(J))THEN
          SAME = .FALSE.
C *************************************************
C         LOOP 525 ASSURES THAT NO REDUNDANCY
C         OCCURS IN ASSIGNING MOVE CHOICES
C *************************************************
          DO 525 JCOUNT = 1,COUNT
            IF(RXX(JCOUNT).EQ.R1X(I))THEN
             IF(RYY(JCOUNT).EQ.R1Y(I))THEN
              IF(RZZ(JCOUNT).EQ.R1Z(I))THEN
          SAME = .TRUE.
              END IF
             END IF
           END IF
 525      CONTINUE
C *************************************************
C *************************************************
          IF (.NOT. SAME) THEN
          COUNT = COUNT + 1
          RXX(COUNT) = R1X(I)
          RYY(COUNT) = R1Y(I)
          RZZ(COUNT) = R1Z(I)
          END IF
C *************************************************
              END IF
            END IF
          END IF
 530      CONTINUE
 540        CONTINUE
          IF(COUNT.GT.4)THEN
C *************************************************
        WRITE(11,*)'ERROR IN COUNTING! COUNT = ',COUNT
        STOP
          END IF
        END IF
 555    IF (BEAD .EQ. LENGTH ) THEN
          BEAD1 = BEAD - 1
```

```
          DO 610 I = 1,12
        RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
        RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
        RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
610       CONTINUE
        COUNT = 12
      ELSE IF (BEAD .EQ. 1 .OR. ANGLE .EQ. 0) THEN
        BEAD1 = BEAD + 1
          DO 620 I = 1,12
        RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
        RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
        RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
620       CONTINUE
        COUNT = 12
      END IF
        COUNT1 = COUNT - 1
        COUNT1 = RAND(IPOINT)*COUNT1+ 1
        IF(RX .EQ. RXX(COUNT1))THEN
          IF(RY .EQ. RYY(COUNT1))THEN
            IF(RZ .EQ. RZZ(COUNT1))THEN
                COUNT1 = COUNT
            END IF
          END IF
        END IF
        RX = RXX(COUNT1)
        RY = RYY(COUNT1)
        RZ = RZZ(COUNT1)
        RX1 = MOD(RX,LIMIT)
        RY1 = MOD(RY,LIMIT)
        RZ1 = MOD(RZ,LIMIT)
          IF(RX1.LE.0)RX1 = RX1 + LIMIT
          IF(RY1.LE.0)RY1 = RY1 + LIMIT
          IF(RZ1.LE.0)RZ1 = RZ1 + LIMIT
      IF (HOME(RX1,RY1,RZ1)) GOTO 600
      RCORX(SAMPLE,BEAD) = RXX(COUNT1)
      RCORY(SAMPLE,BEAD) = RYY(COUNT1)
      RCORZ(SAMPLE,BEAD) = RZZ(COUNT1)
600        CONTINUE
      SAMPLE1 = SAMPLE + 1
      DO 550 J = 1, LENGTH
      RCORX(SAMPLE1,J) = RCORX(SAMPLE,J)
      RCORY(SAMPLE1,J) = RCORY(SAMPLE,J)
      RCORZ(SAMPLE1,J) = RCORZ(SAMPLE,J)
550   CONTINUE
500   CONTINUE
C ************************************************
C
C     STORING COORDINATES
```

```
C
C ***************************************************
        I = SMPMAX
       WRITE (11,*) INTVAL, SMPMAX, LENGTH, ISTEP
       WRITE (11,*) IDIV   , 1
      DO 1000 BD = 1, LENGTH
      WRITE(11,*)RCORX(I,BD),RCORY(I,BD),RCORZ(I,BD)
 1000 CONTINUE
         WRITE(11,*)'FCRX',LENGTH,' IN'
C
      WRITE(25,1976) INTVAL, SMPMAX, LENGTH, ISTEP
      WRITE(25,1976) IDIV   , 1
      DO 900 SAMPLE = 1, SMPMAX
        DO 800 BEAD = 1, LENGTH
        RX = RCORX(SAMPLE,BEAD)
        RY = RCORY(SAMPLE,BEAD)
        RZ = RCORZ(SAMPLE,BEAD)
      WRITE(25,1976) RX, RY, RZ
 800    CONTINUE
 900  CONTINUE
C ***************************************************
      STOP
 1976    FORMAT(4I5)
      END
```

```
@PROCESS DC(CMN1)
C ************************************************
C        PROGRAM NAME        FCFX.FOR
C
C      SIMULATES A SINGLE POLYMER CHAIN IN A FACE
C      CENTERED CUBIC LATTICE WHICH IS END ADSORBED
C      ON A FLAT IMPENETRABLE SURFACE.  EXCLUDED
C      VOLUME IS ACCOUNTED.
C
C ************************************************
      LOGICAL*1 HOME(100,100,100)
      INTEGER RCORX(26000,100), RCORY(26000,100)
      INTEGER RCORZ(26000,100)
      REAL*4 RND(10000)
      INTEGER R1X(12),R1Y(12),R1Z(12)
      INTEGER R2X(12),R2Y(12),R2Z(12),RDATX(12)
      INTEGER RDATZ(12),RXX(12),RYY(12),RZZ(12)
      INTEGER IPOINT,ISEED,LENBUF,SAMPLE, SMPMAX
      INTEGER RDATY(12), SAMPLE1, RX1,RY1,RZ1
      INTEGER INTVAL, BEAD, LENGTH,LENGTH1,RX,RY,RZ
      INTEGER RX2,RY2,RZ2,RX3,RY3,RZ3,LIMIT,STEP
      INTEGER BEAD1, BEAD2, XIND, YIND, ZIND, ANGLE
      INTEGER COUNT, COUNT1, LINTVAL,ISTEP,BD
      LOGICAL*1 SAME
      DATA HOME/1000000*.FALSE./
      DATA RDATX/1,0,0,1,1,0,-1,0,0,-1,-1,0/
      DATA RDATY/0,1,0,0,-1,1,0,-1,0,0,1,-1/
      DATA RDATZ/0,0,1,-1,0,-1,0,0,-1,1,0,1/
      DATA LIMIT/100/
      DATA RXX/1,1,2,2,2,2,2,5*0/
      DATA RYY/1,2,2,2,2,3,2,5*0/
      DATA RZZ/1,1,1,2,3,3,4,5*0/
      COMMON RND, /BUFFER/ LENBUF
      COMMON /CMN1/ RCORX,RCORY,RCORZ
      SQUARE(IA,IB,IC)=IA*(IA+IB+IC)+IB*(IB+IC)+IC*IC
C ************************************************
C        INITIALIZING THE RANDOM NUMBER ROUTINE RAND
        LENBUF = 10000
        CALL INIT(IPOINT)
C ************************************************
C        READING CHAIN COORDINATES
C ************************************************
      READ (10,*) INTVAL, SMPMAX, LENGTH, ISTEP
      READ(10,*) IDIV   , ICHAR
      IF(ICHAR.NE.2)  THEN
           WRITE(6,*)'ICHAR ASSIGNED WRONG VALUE'
      END IF
        LENGTH1 = LENGTH - 1
```

```
          DO 200 BD = 1,LENGTH
         READ(10,*)RCORX(1,BD),RCORY(1,BD),RCORZ(1,BD)
C ************************************************
C     AN EXPLANATION OF THE NEXT NINE LINES CAN BE
C     FOUND WHERE THEY ARE REPEATED BELOW
C ************************************************
      RX = RCORX(1,BD)
      RY = RCORY(1,BD)
      RZ = RCORZ(1,BD)
      RX = MOD(RX,LIMIT)
      IF (RX .LE. 0)RX = RX + LIMIT
      RY = MOD(RY,LIMIT)
      IF (RY .LE. 0)RY = RY + LIMIT
      RZ = MOD(RZ,LIMIT)
      IF (RZ .LE. 0)RZ = RZ + LIMIT
      HOME(RX,RY,RZ)=.TRUE.
 200      CONTINUE
C ************************************************
C     THE NEXT FEW LINES CREATE A "WALL" OR SURFACE
C     WHICH THE POLYMER CHAIN CANNOT PENETRATE, AND
C     ONTO WHICH THE END OF THE CHAIN IS ADSORBED.
C ************************************************
        RX = 1
        DO 300 RZ = 1, LIMIT
        DO 400 RY = 1, LIMIT
          HOME(RX,RY,RZ) = .TRUE.
 400    CONTINUE
 300    CONTINUE
C ************************************************
C        MOVEMENT    LOOP
C
C ************************************************
      LINTVAL = LENGTH1*INTVAL
      DO 500 SAMPLE = 1, SMPMAX
           DO 600 STEP = 1, LINTVAL
      BEAD = RAND(IPOINT)*LENGTH1 + 2
      RX = RCORX(SAMPLE,BEAD)
      RY = RCORY(SAMPLE,BEAD)
      RZ = RCORZ(SAMPLE,BEAD)
C ************************************************
C     NOTE R?3 IS USED AT THE END OF THE LOOP TO
C     CHANGE HOME TO .FALSE. IF THE "BEAD" IS
C     MOVED SUCCESSFULLY.  MOD(A,B) GIVES THE
C     REMAINDER OF A/B.  IF A < OR = 0, THEN
C     MOD(A,B) WILL BE < OR = 0. IF A IS A MULTIPLE
C     OF B, MOD(A,B) = 0.  IN EITHER CASE, THE
C     VALUE OF R?3 SHOULD BE PLACED IN THE RANGE
C     SHOULD BE PLACED IN THE RANGE FROM 1 TO LIMIT.
```

```
C     LIMIT.  THE NEXT SIX STEPS WILL DO JUST THAT.
C ************************************************
      RX3 = MOD(RX,LIMIT)
      IF (RX3 .LE. 0)RX3 = RX3 + LIMIT
      RY3 = MOD(RY,LIMIT)
      IF (RY3 .LE. 0)RY3 = RY3 + LIMIT
      RZ3 = MOD(RZ,LIMIT)
      IF (RZ3 .LE. 0)RZ3 = RZ3 + LIMIT
C ************************************************
        IF(BEAD .NE. 1 .AND. BEAD .NE. LENGTH) THEN
      BEAD1 = BEAD - 1
      BEAD2 = BEAD + 1
C ************************************************
C
C     NOTE: IN A FCC LATTICE, THE 3 AXES ARE AT 60
C     DEGREES TO EACH OTHER.  THIS ALLOWS FOR 12
C     POSSIBLE NEAREST NEIGHBORS.  THE DISTANCE
C     SQUARED BETWEEN BEAD1 AND BEAD2 (BOTH BONDED
C     TO THE BEAD TO BE MOVED) IS ALWAYS AN INTEGER
C     AND IS GIVEN  FOR EACH ANGLE BY:
C       1        60 DEGREE BOND ANGLE
C       2        90 DEGREE BOND ANGLE
C       3       120 DEGREE BOND ANGLE
C       4       180 DEGREE BOND ANGLE.
C     THE FORMULA TO CALCULATE THE DISTANCE SQUARED IS
C     IS NOT THE SAME AS FOR CARTESIAN AXES.  THE
C     DISTANCE SQUARED IS CALCULATED BY THE FUNCTION
C     SQUARE() AS SHOWN BELOW.
C ************************************************
      RX1 = RCORX(SAMPLE,BEAD1)
      RX2 = RCORX(SAMPLE,BEAD2)
      RY1 = RCORY(SAMPLE,BEAD1)
      RY2 = RCORY(SAMPLE,BEAD2)
      RZ1 = RCORZ(SAMPLE,BEAD1)
      RZ2 = RCORZ(SAMPLE,BEAD2)
        XIND = RX2 - RX1
        YIND = RY2 - RY1
        ZIND = RZ2 - RZ1
      ANGLE = SQUARE(XIND,YIND,ZIND)
        GOTO (510,510,510,600) ANGLE
        WRITE(11,*)' ERROR IN COMPUTED GOTO!'
        WRITE(11,*) ANGLE
        WRITE(11,*) 'BEAD1  ',BEAD1,'BEAD2  ',BEAD2
        WRITE(11,*) RX1,RY1,RZ1
        WRITE(11,*) RX2,RY2,RZ2
        STOP
510       DO 520 I = 1,12
        R1X(I) = RX1 + RDATX(I)
```

```
          R2X(I) = RX2 + RDATX(I)
          R1Y(I) = RY1 + RDATY(I)
          R2Y(I) = RY2 + RDATY(I)
          R1Z(I) = RZ1 + RDATZ(I)
          R2Z(I) = RZ2 + RDATZ(I)
 520        CONTINUE
          COUNT = 0
            DO 540 I = 1,12
          DO 530 J = 1,12
          IF(R1X(I).EQ.R2X(J))THEN
            IF(R1Y(I).EQ.R2Y(J))THEN
              IF(R1Z(I).EQ.R2Z(J))THEN
          SAME = .FALSE.
C *************************************************
C         LOOP 525 ASSURES THAT NO REDUNDANCY
C         OCCURS IN ASSIGNING MOVE CHOICES
C *************************************************
          DO 525 JCOUNT = 1,COUNT
            IF(RXX(JCOUNT).EQ.R1X(I))THEN
             IF(RYY(JCOUNT).EQ.R1Y(I))THEN
              IF(RZZ(JCOUNT).EQ.R1Z(I))THEN
          SAME = .TRUE.
              END IF
             END IF
           END IF
 525      CONTINUE
C *************************************************
          IF (.NOT. SAME) THEN
          COUNT = COUNT + 1
          RXX(COUNT) = R1X(I)
          RYY(COUNT) = R1Y(I)
          RZZ(COUNT) = R1Z(I)
          END IF
C *************************************************
              END IF
            END IF
          END IF
 530      CONTINUE
 540        CONTINUE
          IF(COUNT.GT.4)THEN
        WRITE(11,*)'ERROR IN COUNTING! COUNT=',COUNT
        STOP
          END IF
        ELSE IF (BEAD .EQ. LENGTH) THEN
          BEAD1 = LENGTH - 1
            DO 610 I = 1,12
          RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
          RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
```

```
        RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
610       CONTINUE
        COUNT = 12
      ELSE IF (BEAD .EQ. 1) THEN
        BEAD1 = 2
          DO 620 I = 1,12
        RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
        RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
        RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
620       CONTINUE
        COUNT = 12
      ELSE
      WRITE (11,*)'ERROR IN SELECTING BEAD CHOICE!'
      STOP
      END IF
        COUNT1 = COUNT - 1
        COUNT1 = RAND(IPOINT)*COUNT1+ 1
        IF(RX .EQ. RXX(COUNT1))THEN
          IF(RY .EQ. RYY(COUNT1))THEN
            IF(RZ .EQ. RZZ(COUNT1))THEN
                COUNT1 = COUNT
            END IF
          END IF
        END IF
        RX = RXX(COUNT1)
        RY = RYY(COUNT1)
        RZ = RZZ(COUNT1)
        RX1 = MOD(RX,LIMIT)
        RY1 = MOD(RY,LIMIT)
        RZ1 = MOD(RZ,LIMIT)
          IF(RX1.LE.0)RX1 = RX1 + LIMIT
          IF(RY1.LE.0)RY1 = RY1 + LIMIT
          IF(RZ1.LE.0)RZ1 = RZ1 + LIMIT
      IF (HOME(RX1,RY1,RZ1)) GOTO 600
      HOME(RX1,RY1,RZ1) = .TRUE.
      HOME(RX3,RY3,RZ3) = .FALSE.
      RCORX(SAMPLE,BEAD) = RXX(COUNT1)
      RCORY(SAMPLE,BEAD) = RYY(COUNT1)
      RCORZ(SAMPLE,BEAD) = RZZ(COUNT1)
600        CONTINUE
      SAMPLE1 = SAMPLE + 1
      DO 550 J = 1, LENGTH
      RCORX(SAMPLE1,J) = RCORX(SAMPLE,J)
      RCORY(SAMPLE1,J) = RCORY(SAMPLE,J)
      RCORZ(SAMPLE1,J) = RCORZ(SAMPLE,J)
550   CONTINUE
500   CONTINUE
C ************************************************
```

```
C
C     STORING COORDINATES
C
C ****************************************************
        I = SMPMAX
       WRITE (11,*) INTVAL, SMPMAX, LENGTH, ISTEP
        WRITE(11,*) IDIV   , 2
      DO 1000 BD = 1, LENGTH
      WRITE(11,*)RCORX(I,BD),RCORY(I,BD),RCORZ(I,BD)
 1000 CONTINUE
          WRITE(11,*) 'FCFX',LENGTH,' IN'
C
       WRITE(25,1976) INTVAL, SMPMAX, LENGTH, ISTEP
        WRITE(25,1976) IDIV   , 2
      DO 900 SAMPLE = 1, SMPMAX
        DO 800 BEAD = 1, LENGTH
        RX = RCORX(SAMPLE,BEAD)
        RY = RCORY(SAMPLE,BEAD)
        RZ = RCORZ(SAMPLE,BEAD)
         WRITE(25,1976) RX, RY, RZ
 800    CONTINUE
 900   CONTINUE
 1976    FORMAT(4I5)
       STOP
      END
C ****************************************************
C     FUNCTION RAND AND SUBROUTINE INIT ARE TO BE
C     APPENDED TO THIS FILE FOLLOWING THE NEXT LINE.
C ****************************************************
```

```
@PROCESS DC(CMN1)
C ************************************************
C     PROGRAM NAME                FCCAUTO.FOR
C
C     THIS PROGRAM CALCULATES THE AUTOCORRELATION
C     FUNCTIONS FOR A SINGLE CHAIN IN A FACE
C     CENTERED CUBIC LATTICE.
C ************************************************
      INTEGER*4 RCORX(26000,100), RCORY(26000,100)
      INTEGER*4 RCORZ(26000,100)
      REAL*16   COR(500), RCOUNT, RSMAX, RR2
       REAL*16    RU, RV, RW , RAX, RAY, RAZ
      REAL AUTCOR(500),AUCOR(500),R2
      INTEGER*4 INTVAL, SMPMAX, LENGTH, ISTEP
      INTEGER*4 RX, RY, RZ,I, J, K, L,  IDIV/1/
      INTEGER*4 RXK, RXL, RYK, RYL, RZK, RZL
      CHARACTER*1 TAB/Z05/
      COMMON /CMN1/RCORX,RCORY,RCORZ
      DOT2(IA,IB,IC,ID,IE,IF)=(IA+IB+IC)*(ID+IE+IF)
     +        +IA*ID+IB*IE+IC*IF
      SQUARE(IA,IB,IC)=IA*(IA+IB+IC)+IB*(IB+IC)+IC*IC
      DSQUAR(RU,RV,RW)=RU*(RU+RV+RW)+RV*(RV+RW)+RW*RW
C ************************************************
      READ(25,1976) INTVAL, SMPMAX, LENGTH, ISTEP
       READ(25,1976) IDIV , ICHAR
          RDIV = IDIV
 1976  FORMAT (4I5)
        RSMAX = SMPMAX
      DO 100 I = 1, SMPMAX
          DO 120 J = 1, LENGTH
      READ(25,1976) RCORX(I,J),RCORY(I,J),RCORZ(I,J)
 120      CONTINUE
 100  CONTINUE
        RSMAX = SMPMAX
        RR2 = 0.0
        RAX = 0.0
        RAY = 0.0
        RAZ = 0.0
      DO 150 I = 1, SMPMAX
        RX = RCORX(I,LENGTH) - RCORX(I,1)
        RY = RCORY(I,LENGTH) - RCORY(I,1)
        RZ = RCORZ(I,LENGTH) - RCORZ(I,1)
        RR = SQUARE(RX,RY,RZ)
        RR2 = RR2 + RR
        RAX = RAX + RX
        RAY = RAY + RY
        RAZ = RAZ + RZ
 150  CONTINUE
```

```
        RR2 = RR2/RSMAX
        RAX = RAX/RSMAX
        RAY = RAY/RSMAX
        RAZ = RAZ/RSMAX
         R2 = RR2
        RR2 = RR2 -DSQUAR (RAX,RAY,RAZ)
      DO 200 I = 1, ISTEP
        RCOUNT = (SMPMAX - I)*2
        J = I + 1
        COR(I) = 0.0
          DO 300 K = J, SMPMAX
        L = K - I
        RXK = RCORX(K,LENGTH) - RCORX(K,1)
        RXL = RCORX(L,LENGTH) - RCORX(L,1)
        RYK = RCORY(K,LENGTH) - RCORY(K,1)
        RYL = RCORY(L,LENGTH) - RCORY(L,1)
        RZK = RCORZ(K,LENGTH) - RCORZ(K,1)
        RZL = RCORZ(L,LENGTH) - RCORZ(L,1)
        COR(I)=COR(I)+DOT2(RXK,RYK,RZK,RXL,RYL,RZL)
300       CONTINUE
        COR(I) = COR(I)/RCOUNT
        AUTCOR(I)=(COR(I)- DSQUAR(RAX,RAY,RAZ))/RR2
200   CONTINUE
C ***************************************************
      IF(ICHAR.EQ.1)WRITE(6,*)'FCRX',LENGTH
      IF(ICHAR.EQ.3)WRITE(6,*)'FCFR',LENGTH
      IF(ICHAR.EQ.2)WRITE(6,*)'FCFX',LENGTH
      IF(ICHAR.EQ.4)WRITE(6,*)'FCC',LENGTH
      WRITE(11,*) INTVAL, SMPMAX, LENGTH, ISTEP
      WRITE(11,*)  R2 ,' = <R**2>'
      WRITE(11,*) '    TIME    ','    AUTCOR(T)'
      DO 1000 I = 1, ISTEP
       RI = INTVAL*I
            RI = RI/RDIV
      WRITE(11,1989) RI, TAB, AUTCOR(I)
1000  CONTINUE
1989   FORMAT (5X,F12.7,A1,F12.9)
1990   FORMAT (2I5,F12.7)
1991   FORMAT (F12.9)
      WRITE(13, 1976) INTVAL, SMPMAX, LENGTH, ISTEP
      WRITE (13,1990) IDIV, ICHAR, R2
      DO 2000 I = 1, ISTEP
        IF(AUTCOR(I).LT.0)THEN
      WRITE(13,1991) -999999
        ELSE
      WRITE(13,1991) LOG(AUTCOR(I))
       END IF
2000  CONTINUE
```

```
STOP
END
```

```
C ************************************************
C     PROGRAM LINSIM.FOR
C      THIS PROGRAM USES A LEAST SQUARES SUBROUTINE
C      CALLED LINEAR TO DETERMINE THE RELAXATION
C      TIMES FROM AUTOCORRELATION FUNCTION VALUES.
C      IT ALSO USES THE QUADRATIC INTERPOLATION
C      FUNCTION QDVAL TO CALCULATE T(I/E).
C ************************************************
       INTEGER INTVAL, SMPMAX, LENGTH, ISTEP,IDIV
       INTEGER ICCHAR,IS,INDEX/2000/,ICHAR
       REAL YMIN,YMAX,SLOPE,YINT,RHO,YDATA(2000)
       REAL Y(2000),X(2000),TAU,R2,R,L1,L2,L3
       REAL  FDATA(2000)
C ************************************************
       READ (10,*)  YMAX, YMIN
C ************************************************
 1976   FORMAT (4I5)
 1990   FORMAT (F12.9)
 1991   FORMAT (2I5,F12.7)
 1992   FORMAT (3F12.9)
C ************************************************
C     NOTE, THE LINES WITH READ (13, ) ARE DATA READ
C     FROM A CONTINUATION FILE ON THE IBM 3090.
C ************************************************
       READ(13,1976)INTVAL,SMPMAX, LENGTH, ISTEP
       READ(13,1976)IDIV,ICHAR,   ILI
       READ(13,1992) L1, L2, L3
C ************************************************
       DO 100 IS = 1, ISTEP
          READ (13,1990) Y(IS)
          X(IS)= FLOAT(IS*INTVAL)/FLOAT(IDIV)
      IF(Y(IS).LT.YMIN)GOTO 101
 100   CONTINUE
 101   CALL
LINEAR(X,Y,ISTEP,YMIN,YMAX,SLOPE,YINT,RHO,SUM)
       IF(SUM.EQ.0)THEN
        WRITE(6,*) 'SUM = ',SUM
        STOP
       END IF
        TAU = -1./SLOPE
C ************************************************
      WRITE(6,*)'SIMPLE CUBIC LATTICE '
      WRITE(6,*)'CHAIN SIMULATION AS A'
      IF (ICHAR .EQ. 1) THEN
      WRITE (6,*)'FREELY JOINTED RANDOM WALK'
      END IF
      IF (ICHAR .EQ. 2) THEN
       WRITE (6,*) 'SELF AVOIDING WALK'
```

```
      END IF
      IF(ILI.LE.3)THEN
      WRITE(6,*) 'FOR EIGENVALUE L',ILI
      ELSE
      WRITE(6,*)' SPHERICAL HARMONIC CALCULATION:'
      IYI = ILI-3
      GOTO(123,234,345)IYI
 123  WRITE(6,*)'Y(1)= SIN(TH)COS(TH)SIN(PH)'
      GOTO 346
 234  WRITE(6,*)'Y(2)= SIN(TH)COS(TH)COS(PH)'
      GOTO 346
 345  WRITE(6,*)'Y(3)= 3(COS(TH))**2 - 1   '
 346  CONTINUE
      END IF
      WRITE (6,*) 'LENGTH =  ',LENGTH
      WRITE (6,*) 'LI = ', L1, L2, L3
      WRITE (6,*) 'SLOPE =   ',SLOPE
      WRITE (6,*) 'CORRELATION =   ',RHO
      WRITE (6,*) 'RELAX TIME =   ',TAU
C *************************************************
      R = -1.00
      DO 200 IS = 1, ISTEP
        IF(Y(IS).LE.R)THEN
         IJ = IS
         GOTO 300
        END IF
 200   CONTINUE
 300   CONTINUE
       WRITE(6,*) IJ
       WRITE(6,*) ISTEP
         ICOUNT = IJ + 1
         ISTART = IJ - 2
       IF(ISTART.LE.0)THEN
         ICOUNT = 2*(IJ-1)
         ISTART = 1
       END IF
      DO 400 IS = ISTART, ICOUNT
         INUM = ICOUNT -2*(IS) + ISTART + 1
       IF (INUM .LT.3)GOTO 900
         ICOUNT2 = INUM + IS - 1
         NDATA = ICOUNT2 - IS + 1
      DO 500 IT = IS, ICOUNT2
         IP = ICOUNT - IT + 1
         FDATA(IP) = X(IT)
         YDATA(IP) = Y(IT)
 500  CONTINUE
       WRITE(6,*) 'TAU(1/e) = '
       WRITE(6,*) QDVAL(R,NDATA,YDATA,FDATA,CHECK)
```

```
 400    CONTINUE
 900    CONTINUE
        STOP
C ***************************************************
C ****** END OF PROGRAM !!! ************************
C ***************************************************
        END
```

APPENDIX II

PROGRAMS USED FOR CHAIN SIMULATIONS OF SHAPE DYNAMICS

Included in this appendix are the programs used to simulate a free polymer chain, and monitor its shape relaxation and rotation. The first two programs, SMP.FOR and SMFR.FOR, are the simulation programs for the no excluded volume and the excluded volume cases respectively. The third program calculates the, EIGENS.FOR eigenvalues, and the eigenvalue and spherical harmonics autocorrelation function values used in the last program. The last program listed, TAU.FOR, calculates the relaxation time from the eigenvalue or spherical harmonics autocorrelation values that were sent from the previous program.

```
@PROCESS DC(CMN1)
C ************************************************
C        PROGRAM NAME        SIMFRE.FOR
C
C        SIMULATES A POLYMER CHAIN IN A SIMPLE
C        CUBIC LATTICE WITH NO EXCLUDED VOLUME.
C
C ************************************************
      INTEGER RCORX(30000,74),RCORY(30000,74)
      INTEGER RCORZ(30000,74),CCCOUNT/0/
      REAL*4 RND(10000)
      INTEGER R1X(6),R1Y(6),R1Z(6)
      INTEGER R2X(6),R2Y(6),R2Z(6),RDATX(6),RDATY(6)
      INTEGER RDATZ(6),RXX(6),RYY(6),RZZ(6),ICN/6/
      INTEGER IPOINT,ISEED,LENBUF,SAMPLE, SMPMAX
      INTEGER INTVAL, BEAD, LENGTH,RX, RY, RZ, RX1
      INTEGER IJKOUNT/0/,SAMPLE1,IDIV/1/,RY1, RZ1
      INTEGER RX2,RY2,RZ2,RX3,RY3,RZ3,RX4,RY4,RZ4
      INTEGER LIMIT,STEP, COUNT, COUNT1,BEAD3,BEAD4
      INTEGER BEAD1, BEAD2, XIND, YIND, ZIND
      INTEGER ANGLE,LINTVAL,ISTEP,BD
      LOGICAL*1 SAME,BDEND
      DATA RDATX/1,0,0,-1,0,0/
      DATA RDATY/0,1,0,0,-1,0/
      DATA RDATZ/0,0,1,0,0,-1/
      DATA LIMIT/100/
      COMMON RND, /BUFFER/ LENBUF
      COMMON/CMN1/ RCORX,RCORY,RCORZ
      SQUARE(IA,IB,IC)=IA*IA +IB*IB +IC*IC
C ************************************************
C ************************************************
C      INITIALIZING THE RANDOM NUMBER ROUTINE RAND
        LENBUF = 10000
        CALL INIT(IPOINT)
C ************************************************
C        READING CHAIN COORDINATES
C
      READ (10,*) INTVAL, SMPMAX, LENGTH, ISTEP
      READ (10,*)  IDIV,     ILI
          DO 200 BD = 1,LENGTH
      READ(10,*)RCORX(1,BD),RCORY(1,BD),RCORZ(1,BD)
C ************************************************
C AN EXPLANATION OF THE NEXT NINE LINES CAN BE
C FOUND WHERE THEY ARE REPEATED BELOW.
C
C     RX = RCORX(1,BD)
C     RY = RCORY(1,BD)
C     RZ = RCORZ(1,BD)
```

```
C     RX = MOD(RX,LIMIT)
C     IF (RX .LE. 0)RX = RX + LIMIT
C     RY = MOD(RY,LIMIT)
C     IF (RY .LE. 0)RY = RY + LIMIT
C     RZ = MOD(RZ,LIMIT)
C     IF (RZ .LE. 0)RZ = RZ + LIMIT
 200      CONTINUE
C ************************************************
C ************************************************
C        MOVEMENT    LOOP
C
C
      LINTVAL = LENGTH*INTVAL
          LINTVAL = LINTVAL/IDIV
      DO 500 SAMPLE = 1, SMPMAX
           DO 1010 STEP = 1, LINTVAL
      BEAD = RAND(IPOINT)*LENGTH + 1
      RX = RCORX(SAMPLE,BEAD)
      RY = RCORY(SAMPLE,BEAD)
      RZ = RCORZ(SAMPLE,BEAD)
C ************************************************
       IF(BEAD .NE. 1 .AND. BEAD .NE. LENGTH) THEN
C ************************************************
C
C     THE DISTANCE SQUARED IS
C     CALCULATED BY THE FUNCTION SQUARE() BELOW.
C ************************************************
C
      BEAD1 = BEAD - 1
      BEAD2 = BEAD + 1
      RX1 = RCORX(SAMPLE,BEAD1)
      RX2 = RCORX(SAMPLE,BEAD2)
      RY1 = RCORY(SAMPLE,BEAD1)
      RY2 = RCORY(SAMPLE,BEAD2)
      RZ1 = RCORZ(SAMPLE,BEAD1)
      RZ2 = RCORZ(SAMPLE,BEAD2)
        XIND = RX2 - RX1
        YIND = RY2 - RY1
        ZIND = RZ2 - RZ1
      ANGLE = SQUARE(XIND,YIND,ZIND)
        CCCOUNT = CCCOUNT + 1
        GOTO (510,510,510,1010) ANGLE
        IF(ANGLE.EQ.0) GOTO 600
        WRITE(11,*)' ERROR IN COMPUTED GOTO!'
        WRITE(11,*) ANGLE
        WRITE(11,*) BDEND
        WRITE(11,*) 'IJKOUNT = ',IJKOUNT
        WRITE(11,*) 'CCCOUNT = ',CCCOUNT
```

```
C ************************************************
      WRITE(11,*)'BEAD1 ',BEAD1,'BEAD2 ',BEAD2
       DO 1234 BEAD = 1, LENGTH
               RX1 = RCORX(SAMPLE,BEAD)
               RY1 = RCORY(SAMPLE,BEAD)
               RZ1 = RCORZ(SAMPLE,BEAD)
       WRITE(11,*) RX1,RY1,RZ1
1234   CONTINUE
       STOP
510    COUNT = 1
       COUNT1 = COUNT
       BDEND = .FALSE.
     IF(RX.EQ.RX1)THEN
       RXX(COUNT1)=RX2
     ELSE
       RXX(COUNT1)=RX1
     END IF
     IF(RY.EQ.RY1)THEN
       RYY(COUNT1)=RY2
     ELSE
       RYY(COUNT1)=RY1
     END IF
     IF(RZ.EQ.RZ1)THEN
       RZZ(COUNT1)=RZ2
     ELSE
       RZZ(COUNT1)=RZ1
     END IF
600    CONTINUE
     END IF
     IF (BEAD .EQ. LENGTH) THEN
       BDEND = .TRUE.
       BEAD1 = BEAD - 1
         DO 610 I = 1,ICN
       RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
       RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
       RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
610      CONTINUE
       COUNT = ICN
     ELSE IF (BEAD .EQ. 1.OR.ANGLE.EQ.0)THEN
       BDEND = .TRUE.
       BEAD1 = BEAD + 1
         DO 620 I = 1,ICN
       RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
       RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
       RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
620      CONTINUE
       COUNT = ICN
     END IF
```

```
          IF(BDEND)THEN
         COUNT1 = COUNT - 1
         COUNT1 = RAND(IPOINT)*COUNT1+ 1
         IF(RX .EQ. RXX(COUNT1))THEN
           IF(RY .EQ. RYY(COUNT1))THEN
             IF(RZ .EQ. RZZ(COUNT1))THEN
                 COUNT1 = COUNT
             END IF
           END IF
         END IF
          END IF
       RCORX(SAMPLE,BEAD) = RXX(COUNT1)
       RCORY(SAMPLE,BEAD) = RYY(COUNT1)
       RCORZ(SAMPLE,BEAD) = RZZ(COUNT1)
 1010       CONTINUE
       SAMPLE1 = SAMPLE + 1
       DO 550 J = 1, LENGTH
       RCORX(SAMPLE1,J) = RCORX(SAMPLE,J)
       RCORY(SAMPLE1,J) = RCORY(SAMPLE,J)
       RCORZ(SAMPLE1,J) = RCORZ(SAMPLE,J)
 550   CONTINUE
 500   CONTINUE
C ************************************************
C
C      STORING COORDINATES
C
         I = SMPMAX
        WRITE (11,*) INTVAL, SMPMAX, LENGTH, ISTEP
        WRITE (11,*) IDIV,         ILI,      1
       DO 1000 BD = 1, LENGTH
       WRITE(12,*)RCORX(I,BD),RCORY(I,BD),RCORZ(I,BD)
 1000 CONTINUE
            WRITE(12,*) 'SMFR',LENGTH,' IN'
C
        WRITE(25,1976) INTVAL, SMPMAX, LENGTH, ISTEP
        WRITE(25,1976) IDIV,      ILI,       1
 1976     FORMAT(4I5)
       DO 900 SAMPLE = 1, SMPMAX
         DO 800 BEAD = 1, LENGTH
         RX = RCORX(SAMPLE,BEAD)
         RY = RCORY(SAMPLE,BEAD)
         RZ = RCORZ(SAMPLE,BEAD)
       WRITE(25,1976) RX, RY, RZ
 800     CONTINUE
 900   CONTINUE
C ************************************************
       STOP
       END
```

```
@PROCESS DC(CMN1)
C ************************************************
C ************************************************
C        PROGRAM NAME        SMP.FOR
C
C     SIMULATES A POLYMER CHAIN IN A SIMPLE
C        CUBIC LATTICE
C
C ************************************************
      LOGICAL*1 HOME(100,100,100)
      INTEGER RCORX(30000, 74), RCORY(30000, 74)
      INTEGER RCORZ(30000, 74)
      REAL*4 RND(10000)
      INTEGER R1X(6),R1Y(6),R1Z(6),R2X(6)
      INTEGER R2Y(6),R2Z(6),RDATX(6),RDATY(6)
      INTEGER RDATZ(6),RXX(6),RYY(6),RZZ(6),ICN/6/
      INTEGER IPOINT,ISEED,LENBUF,SAMPLE, SMPMAX
      INTEGER INTVAL, BEAD, LENGTH,RX, RY, RZ, BD
      INTEGER SAMPLE1, RX1, RY1, RZ1
      INTEGER RX2,RY2,RZ2,RX3,RY3,RZ3,RX4,RY4,RZ4
      INTEGER LIMIT,STEP, COUNT, COUNT1,BEAD3,BEAD4
      INTEGER BEAD1, BEAD2, XIND, YIND, ZIND
      INTEGER ANGLE,LINTVAL,ISTEP
      LOGICAL*1 SAME,BDEND
      DATA HOME/1000000*.FALSE./
      DATA LIMIT/100/
      DATA RDATX/1,0,0,-1,0,0/
      DATA RDATY/0,1,0,0,-1,0/
      DATA RDATZ/0,0,1,0,0,-1/
      COMMON RND, /BUFFER/ LENBUF
      COMMON /CMN1/RCORX,RCORY,RCORZ
      SQUARE(IA,IB,IC)= IA*IA + IB*IB + IC*IC
C ************************************************
C      INITIALIZING THE RANDOM NUMBER ROUTINE RAND
        LENBUF = 10000
        CALL INIT(IPOINT)
C ************************************************
C        READING CHAIN COORDINATES
C
      READ (10,*) INTVAL, SMPMAX, LENGTH, ISTEP
      READ (10,*) IDIV   , ILI
          DO 200 BD = 1,LENGTH
      READ(10,*)RCORX(1,BD),RCORY(1,BD),RCORZ(1,BD)
C ************************************************
C     AN EXPLANATION OF THE NEXT NINE LINES CAN BE
C     FOUND WHERE THEY ARE REPEATED BELOW
C
         BEAD = BD
```

```
      RX = RCORX(1,BEAD)
      RY = RCORY(1,BEAD)
      RZ = RCORZ(1,BEAD)
      RX = MOD(RX,LIMIT)
      IF (RX .LE. 0)RX = RX + LIMIT
      RY = MOD(RY,LIMIT)
      IF (RY .LE. 0)RY = RY + LIMIT
      RZ = MOD(RZ,LIMIT)
      IF (RZ .LE. 0)RZ = RZ + LIMIT
      HOME(RX,RY,RZ)=.TRUE.
 200      CONTINUE
C ************************************************
C        MOVEMENT    LOOP
C
C
C ************************************************
      LINTVAL = LENGTH*INTVAL
        LINTVAL = LINTVAL/IDIV
C ************************************************
C     LINTVAL IS THE NUMBER OF MOVE CYCLES BETWEEN
C       SETS OF RECORDED COORDINATES
C ************************************************
      DO 500 SAMPLE = 1, SMPMAX
           DO 1010 STEP = 1, LINTVAL
      BEAD = RAND(IPOINT)*LENGTH + 1
      RX = RCORX(SAMPLE,BEAD)
      RY = RCORY(SAMPLE,BEAD)
      RZ = RCORZ(SAMPLE,BEAD)
C ************************************************
C ************************************************
C  NOTE, R?3 IS USED AT THE END OF THE LOOP TO
C  CHANGE HOME TO .FALSE IF THE "BEAD" IS MOVED
C  SUCCESSFULLY.  MOD(A,B) GIVES THE REMAINDER OF
C  A/B.  IF A < OR = 0, THEN MOD(A,B) WILL BE
C  < OR = 0. IF A IS A MULTIPLE OF B, MOD(A,B)= 0.
C  IN EITHER CASE, THE VALUE OF R?3 SHOULD BE
C  PLACED IN THE RANGE FROM 1 TO LIMIT. THE
C  NEXT SIX STEPS WILL DO JUST THAT.
C ************************************************
      RX3 = MOD(RX,LIMIT)
      IF (RX3 .LE. 0)RX3 = RX3 + LIMIT
      RY3 = MOD(RY,LIMIT)
      IF (RY3 .LE. 0)RY3 = RY3 + LIMIT
      RZ3 = MOD(RZ,LIMIT)
      IF (RZ3 .LE. 0)RZ3 = RZ3 + LIMIT
C ************************************************
       IF(BEAD .NE. 1 .AND. BEAD .NE. LENGTH) THEN
C ************************************************
```

```
C
C     THE DISTANCE SQUARED IS
C     CALCULATED BY THE FUNCTION SQUARE() BELOW.
C ************************************************
C
      BEAD1 = BEAD - 1
      BEAD2 = BEAD + 1
      RX1 = RCORX(SAMPLE,BEAD1)
      RX2 = RCORX(SAMPLE,BEAD2)
      RY1 = RCORY(SAMPLE,BEAD1)
      RY2 = RCORY(SAMPLE,BEAD2)
      RZ1 = RCORZ(SAMPLE,BEAD1)
      RZ2 = RCORZ(SAMPLE,BEAD2)
        XIND = RX2 - RX1
        YIND = RY2 - RY1
        ZIND = RZ2 - RZ1
      ANGLE = SQUARE(XIND,YIND,ZIND)
        GOTO (510,510,510,1010) ANGLE
        WRITE(11,*)' ERROR IN COMPUTED GOTO!'
        WRITE(11,*) ANGLE
        WRITE(11,*) BDEND
        WRITE(11,*)'BEAD1  ',BEAD1,'BEAD2  ',BEAD2
        DO 1234 BEAD = 1, LENGTH
                RX1 = RCORX(SAMPLE,BEAD)
                RY1 = RCORY(SAMPLE,BEAD)
                RZ1 = RCORZ(SAMPLE,BEAD)
        WRITE(11,*) RX1,RY1,RZ1
1234    CONTINUE
        STOP
510     COUNT = 1
        COUNT1 = COUNT
        BDEND = .FALSE.
      IF(RX.EQ.RX1)THEN
        RXX(COUNT1)=RX2
      ELSE
        RXX(COUNT1)=RX1
      END IF
      IF(RY.EQ.RY1)THEN
        RYY(COUNT1)=RY2
      ELSE
        RYY(COUNT1)=RY1
      END IF
      IF(RZ.EQ.RZ1)THEN
        RZZ(COUNT1)=RZ2
      ELSE
        RZZ(COUNT1)=RZ1
      END IF
      ELSE IF (BEAD .EQ. LENGTH) THEN
```

```
        BDEND = .TRUE.
        BEAD1 = LENGTH - 1
          DO 610 I = 1,ICN
        RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
        RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
        RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
610       CONTINUE
        COUNT = ICN
      ELSE IF (BEAD .EQ. 1) THEN
        BDEND = .TRUE.
        BEAD1 = 2
          DO 620 I = 1,ICN
        RXX(I) = RCORX(SAMPLE,BEAD1) + RDATX(I)
        RYY(I) = RCORY(SAMPLE,BEAD1) + RDATY(I)
        RZZ(I) = RCORZ(SAMPLE,BEAD1) + RDATZ(I)
620       CONTINUE
        COUNT = ICN
      ELSE
       WRITE(11,*)'ERROR IN SELECTING BEAD CHOICE!'
      STOP
      END IF
          IF(BDEND)THEN
        COUNT1 = COUNT - 1
        COUNT1 = RAND(IPOINT)*COUNT1+ 1
        IF(RX .EQ. RXX(COUNT1))THEN
          IF(RY .EQ. RYY(COUNT1))THEN
            IF(RZ .EQ. RZZ(COUNT1))THEN
                COUNT1 = COUNT
            END IF
          END IF
        END IF
         END IF
        RX4 = RXX(COUNT1)
        RY4 = RYY(COUNT1)
        RZ4 = RZZ(COUNT1)
        RX4 = MOD(RX4,LIMIT)
        RY4 = MOD(RY4,LIMIT)
        RZ4 = MOD(RZ4,LIMIT)
          IF(RX4.LE.0)RX4 = RX4 + LIMIT
          IF(RY4.LE.0)RY4 = RY4 + LIMIT
          IF(RZ4.LE.0)RZ4 = RZ4 + LIMIT
      IF (HOME(RX4,RY4,RZ4)) GOTO 600
      HOME(RX4,RY4,RZ4) = .TRUE.
      HOME(RX3,RY3,RZ3) = .FALSE.
      RCORX(SAMPLE,BEAD) = RXX(COUNT1)
      RCORY(SAMPLE,BEAD) = RYY(COUNT1)
      RCORZ(SAMPLE,BEAD) = RZZ(COUNT1)
      GOTO 1010
```

```
C ************************************************
C     THE LINES BETWEEN 600 AND 1010 ARE FOR THE
C     CRANKSHAFT MOTION
C ************************************************
 600  IF(BDEND)GOTO 1010
         IF(BEAD.LT.3)GOTO 1020
            BEAD2 = BEAD - 2
        I=1
C ************************************************
C     IF ANY OF THE NEXT THREE STATEMENTS IS TRUE,
C     THE PROGRAM JUMPS TO 1020 TO DO AN ALTERNATE
C     CHECK
C ************************************************
      IF(RCORX(SAMPLE,BEAD2).NE.RXX(COUNT1))GOTO 1020
      IF(RCORY(SAMPLE,BEAD2).NE.RYY(COUNT1))GOTO 1020
      IF(RCORZ(SAMPLE,BEAD2).NE.RZZ(COUNT1))GOTO 1020
        BEAD4 = BEAD2
        BEAD3 = BEAD4 + 1
        BEAD2 = BEAD4 + 2
        BEAD1 = BEAD4 + 3
      GOTO 1030
 1020 IF(BEAD.GT.LENGTH-2)GOTO 1010
        BEAD2 = BEAD+2
           I = -1
C ************************************************
C     IF ANY OF THE NEXT THREE STATEMENTS IS TRUE
C     THEN THE PROGRAM JUST GIVES UP ON THE CRANK-
C     SHAFT AS BEING NOT POSSIBLE
C ************************************************
       IF(RCORX(SAMPLE,BEAD2).NE.RXX(1))GOTO 1010
       IF(RCORY(SAMPLE,BEAD2).NE.RYY(1))GOTO 1010
       IF(RCORZ(SAMPLE,BEAD2).NE.RZZ(1))GOTO 1010
        BEAD4 = BEAD2
        BEAD3 = BEAD4 - 1
        BEAD2 = BEAD4 - 2
        BEAD1 = BEAD4 - 3
C ************************************************
 1030 RX1 = RCORX(SAMPLE,BEAD1)
      RY1 = RCORY(SAMPLE,BEAD1)
      RZ1 = RCORZ(SAMPLE,BEAD1)
      RX2 = RCORX(SAMPLE,BEAD2)
      RY2 = RCORY(SAMPLE,BEAD2)
      RZ2 = RCORZ(SAMPLE,BEAD2)
      RX3 = RCORX(SAMPLE,BEAD3)
      RY3 = RCORY(SAMPLE,BEAD3)
      RZ3 = RCORZ(SAMPLE,BEAD3)
      RX4 = RCORX(SAMPLE,BEAD4)
      RY4 = RCORY(SAMPLE,BEAD4)
```

```
      RZ4 = RCORZ(SAMPLE,BEAD4)
C *************************************************
C     R? IS EITHER R?2 OR R?3 AND BOTH MOVE.
C     R?4 AND R?1 MAKE THE AXIS ABOUT WHICH R?2 &
C     R?3 MOVE (IF THE SITES ARE NOT OCCUPIED)
C     (IF THE SITES ARE NOT OCCUPIED).
C *************************************************
       COUNT = 2*RAND(IPOINT) + 1
         I = 1
      IF(COUNT .EQ. 2) I = -1
        J = I
        K = I
      IF(RX3.NE.RX1)I=0
      IF(RY3.NE.RY1)J=0
      IF(RZ3.NE.RZ1)K=0
        IF(I+J+K.NE.1)THEN
        IF(I+J+K.NE.-1)THEN
        WRITE(6,*) 'ERROR IN I + J + K'
        STOP
        END IF
        END IF
        RXX(2)=RX1 +I
        RXX(3)=RX4 +I
        RYY(2)=RY1 +J
        RYY(3)=RY4 +J
        RZZ(2)=RZ1 +K
        RZZ(3)=RZ4 +K
         RXX(1)=MOD(RXX(2),LIMIT)
         RXX(4)=MOD(RXX(3),LIMIT)
         RZZ(1)=MOD(RZZ(2),LIMIT)
         RZZ(4)=MOD(RZZ(3),LIMIT)
         RYY(1)=MOD(RYY(2),LIMIT)
         RYY(4)=MOD(RYY(3),LIMIT)
         IF(RXX(1).LE.0)RXX(1)=RXX(1)+LIMIT
         IF(RXX(4).LE.0)RXX(4)=RXX(4)+LIMIT
         IF(RYY(1).LE.0)RYY(1)=RYY(1)+LIMIT
         IF(RYY(4).LE.0)RYY(4)=RYY(4)+LIMIT
         IF(RZZ(1).LE.0)RZZ(1)=RZZ(1)+LIMIT
         IF(RZZ(4).LE.0)RZZ(4)=RZZ(4)+LIMIT
          IF(HOME(RXX(1),RYY(1),RZZ(1)))GOTO 1010
          IF(HOME(RXX(4),RYY(4),RZZ(4)))GOTO 1010
           HOME(RXX(1),RYY(1),RZZ(1))=.TRUE.
           HOME(RXX(4),RYY(4),RZZ(4))=.TRUE.
           RX2 = MOD(RX2,LIMIT)
           RX3 = MOD(RX3,LIMIT)
           RY2 = MOD(RY2,LIMIT)
           RY3 = MOD(RY3,LIMIT)
           RZ2 = MOD(RZ2,LIMIT)
```

```
            RZ3 = MOD(RZ3,LIMIT)
          IF(RX2.LE.0)RX2=RX2+LIMIT
          IF(RY2.LE.0)RY2=RY2+LIMIT
          IF(RZ2.LE.0)RZ2=RZ2+LIMIT
          IF(RX3.LE.0)RX3=RX3+LIMIT
          IF(RY3.LE.0)RY3=RY3+LIMIT
          IF(RZ3.LE.0)RZ3=RZ3+LIMIT
             HOME(RX2,RY2,RZ2) = .FALSE.
             HOME(RX3,RY3,RZ3) = .FALSE.
          RCORX(SAMPLE,BEAD2)=RXX(2)
          RCORX(SAMPLE,BEAD3)=RXX(3)
          RCORY(SAMPLE,BEAD2)=RYY(2)
          RCORY(SAMPLE,BEAD3)=RYY(3)
          RCORZ(SAMPLE,BEAD2)=RZZ(2)
          RCORZ(SAMPLE,BEAD3)=RZZ(3)
C ************************************************
C     END OF CRANK SHAFT MOTION STEPS
C ************************************************
 1010       CONTINUE
       SAMPLE1 = SAMPLE + 1
       DO 550 J = 1, LENGTH
       RCORX(SAMPLE1,J) = RCORX(SAMPLE,J)
       RCORY(SAMPLE1,J) = RCORY(SAMPLE,J)
       RCORZ(SAMPLE1,J) = RCORZ(SAMPLE,J)
 550   CONTINUE
 500   CONTINUE
C ************************************************
C
C     STORING COORDINATES
C
         I = SMPMAX
        WRITE (11,*) INTVAL, SMPMAX, LENGTH, ISTEP
        WRITE (11,*) IDIV  ,    ILI  ,   2
       DO 1000 BD = 1, LENGTH
       WRITE(12,*)RCORX(I,BD),RCORY(I,BD),RCORZ(I,BD)
 1000 CONTINUE
          WRITE(12,*) 'SMP',LENGTH,' IN'
C
        WRITE(25,1976) INTVAL, SMPMAX, LENGTH, ISTEP
        WRITE(25,1976) IDIV,     ILI, 2
 1976    FORMAT(4I5)
       DO 900 I = 1, SMPMAX
         DO 800 BD = 1, LENGTH
       WRITE(25,1976) RCORX(I,BD),RCORY(I,BD),RCORZ(I,BD)

 800     CONTINUE
 900   CONTINUE
       STOP
```

```
END
```

```
@PROCESS DC(CMN1)
C ***************************************************
C       PROGRAM NAME EIGENS.FOR
C       THIS PROGRAM CALLS TWO SUBROUTINES, TREDM,
C       AND TQLIM WHICH ARE USED TO CALCULATE
C       EIGENVALUES AND EIGENVECTORS FOR REAL
C       SYMMETRIC MATRICES.  THESE SUBROUTINES ARE
C       FOUND IN A PROGRAM FILE CALLED
C           EIGEN2.FOR
C ***************************************************
       INTEGER I,J,K,L,KK,LL,MM,INTVAL,SMPMAX
       INTEGER MMP/3/,NNP/3/,N/3/,NP/3/,BD,IS, IDIV
       INTEGER RCORX(30000,74),RCORY(30000,74)
       INTEGER RCORZ(30000,74),LENGTH, ISTEP
       REAL RSMAX,R(100,3)/300*0.0/,COR(500,3)
       REAL ECOR(500,3)
       CHARACTER*1 TAB/Z05/,CCHAR(2)/'F','P'/
       REAL*16 EVAL(30000,3),EVEC(30000,3)
       REAL*16 SUMJK(3,3)/9*0.0/,RLENGTH
       REAL*16 LAMDA(3)/3*0.0/
       REAL*16 EVC(3,3)/3.,2.,1.,2.,4.,2.,1.,2.,5./
       REAL*16 RR(3)/3*0.0/,Y(3)/3*0.0/,THREE/3.0/
       REAL*16 RE2(3)/3*0.0/,RELIM(3)/3*0.0/
       REAL*16 EVA(3),ODIA(3)/3*0.0/,ZERO/0.0/
       REAL*16 RI2(3)/3*0.0/,RA(3)/3*0.0/,TWO/2.0/
       REAL   RLL(3),RYY(3)
       COMMON /CMN1/ RCORX,RCORY,RCORZ,EVAL,EVEC
          READ(25,1976) INTVAL,SMPMAX,LENGTH,ISTEP
          READ(25,1976) IDIV,      ILI,       ICHAR
           WRITE(6,*) CCHAR(ICHAR)
 1976      FORMAT(4I5)
C ***************************************************
         CALL TREDM(EVC, N, NP, EVA, ODIA)
         CALL TQLIM(EVA, ODIA, N, NP, EVC)
C ***************************************************
              RSMAX = SMPMAX
              RLENGTH = LENGTH
       DO 1000 IS = 1, SMPMAX
        DO 1010 BD = 1, LENGTH

READ(25,1976)RCORX(IS,BD),RCORY(IS,BD),RCORZ(IS,BD)
 1010   CONTINUE
 1000  CONTINUE
C ***************************************************
C   THIS IS THE START OF THE ANALYSIS FOR
C   EIGENVALUES AND SPHERICAL HARMONICS.  TREDM AND
C   TQLIM ARE THE SUBROUTINES USED TO CALCULATE THE
C   EIGENVALUES AND EIGENVECTORS.
```

```
C   THEY ARE SUPPLIED IN FILE EIGEN2.FOR
C ************************************************
       DO 1100 IS = 1, SMPMAX
        DO 1110 BD = 1, LENGTH
       R(BD,1) = RCORX(IS,BD)
       R(BD,2) = RCORY(IS,BD)
       R(BD,3) = RCORZ(IS,BD)
 1110   CONTINUE
        DO 1170 LL = 1, NNP
         DO 1180 MM = 1, NNP
       SUMJK(MM,LL) = ZERO
          DO 1165 BD = 1, LENGTH
       SUMJK(MM,LL)=SUMJK(MM,LL)+R(BD,MM)*R(BD,LL)
 1165     CONTINUE
 1180    CONTINUE
 1170   CONTINUE
        DO 1190 LLL = 1, MMP
         RR(LLL) = ZERO
         DO 1195 BD = 1, LENGTH
           RR(LLL) = RR(LLL) + R(BD,LLL)
 1195    CONTINUE
 1190   CONTINUE
        DO 2000 K = 1, MMP
         DO 2010 J = 1, MMP
       EVC(J,K) = SUMJK(J,K) -RR(J)*RR(K)/RLENGTH
       EVC(J,K) = EVC(J,K)/RLENGTH
 2010    CONTINUE
 2000   CONTINUE
C ************************************************
         CALL TREDM(EVC, N, NP, EVA, ODIA)
         CALL TQLIM(EVA, ODIA, N, NP, EVC)
C ************************************************
C
C ------ WRITE EIGENVALUES AND EIGENVECTORS ------
C
          DO 1123 IX = 1, 3
                LAMDA(IX) = LAMDA(IX) + EVA(IX)
 1123     CONTINUE
              Y(3) = THREE*EVC(3,3)**2 - 1
C ************************************************
C     Y(3) = 3(COS(TH))**2 - 1 ..  TH MEANS THETA
C ************************************************
              Y(2) = -EVC(3,3)*EVC(3,2)
C ************************************************
C     Y(2) = SIN(TH)COS(TH))COS(PH) .. PH MEAND PHI
C ************************************************
              Y(1) =  EVC(3,3)*EVC(3,1)
C ************************************************
```

```
C     Y(1) = SIN(TH)COS(TH))SIN(PH)
C ************************************************
           DO 3000 J = 1, MMP
             EVAL(IS,J) = EVA(J)
             RI2(J)= RI2(J) + EVA(J)*EVA(J)
            RA(J) = RA(J) + EVA(J)
          RELIM(J) = RELIM(J) + Y(J)*Y(J)
              EVEC(IS,J) = Y(J)*Y(J)
          RE2(J) = RE2(J) + (EVEC(IS,J)*EVEC(IS,J))
 3000       CONTINUE
 1100      CONTINUE
C ************************************************
C  THIS IS THE END OF THE EIGENVALUE AND SPHERICAL
C   HARMONICS CALCULATIONS.
C ************************************************
C   AND THE START OF THE CORRELATION CALCULATIONS
C ************************************************
         DO 4000 K = 1,MMP
             RI2(K) = RI2(K)/RSMAX
             RA(K) = RA(K)/RSMAX
             RE2(K) = RE2(K)/RSMAX
             RELIM(K) = RELIM(K)/RSMAX
             RLL(K) = RA(K)
             RYY(K) = RE2(K)
 4000    CONTINUE
         DO 201 M = 1, NNP
         DO 202 I = 1, ISTEP
          COR(I,M) = 0.0
          ECOR(I,M) = 0.0
 202    CONTINUE
        DO 200 I = 1, ISTEP
          RCOUNT = SMPMAX - I
          J = I + 1
           DO 300 K = J, SMPMAX
             L = K - I
            COR(I,M) = COR(I,M) + EVAL(K,M)*EVAL(L,M)
          ECOR(I,M) = ECOR(I,M) + EVEC(K,M)*EVEC(L,M)
 300       CONTINUE
                COR(I,M) = COR(I,M)/RCOUNT
               ECOR(I,M) = ECOR(I,M)/RCOUNT
               ECOR(I,M) = ECOR(I,M)
               ECOR(I,M) = ECOR(I,M)/RE2(M)
               COR(I,M) = COR(I,M)/RI2(M)
 200     CONTINUE
           RA(M) = RA(M)**2/RI2(M)
           RELIM(M) = RELIM(M)**2/RE2(M)
 201    CONTINUE
               JSTEP = ISTEP
```

```
          DO 7777 J = 1, NNP
            DO 5555 I = 1, ISTEP
              COR(I,J) = COR(I,J) - RA(J)
              COR(I,J) = COR(I,J)/(1.0 - RA(J))
              ECOR(I,J) = ECOR(I,J) - RELIM(J)
             ECOR(I,J) = ECOR(I,J)/(1.0 - RELIM(J))
 5555       CONTINUE
 7777       CONTINUE
C ***************************************************
C   THIS IS THE END OF THE CORRELATION CALCULATIONS
C ***************************************************
 1989       FORMAT ( 5X,F14.9,A1, F12.9)
 1991       FORMAT ( 3F12.7)
         WRITE(13,1976)INTVAL,SMPMAX,LENGTH,ISTEP
         WRITE(13,1976)IDIV, ICHAR, ILI
        IF(ILI .LE.3)THEN
C ***************************************************
        WRITE(6,*)'L',ILI,CCHAR(ICHAR),LENGTH,'* TAU'
           WRITE(13,1991)(RLL(JLI),JLI=1,3)
        DO 1313 K = 1, ISTEP
            I = INTVAL*K
           RI = FLOAT(I)/FLOAT(IDIV)
            IF(COR(K,ILI).LE.0)THEN
          WRITE(13,1991)9.
          WRITE(11,1991)RI,TAB,9.
            ELSE
           WRITE(11,1989) RI,TAB,LOG(COR(K,ILI))
          WRITE(13,1991)LOG(COR(K,ILI))
            END IF
 1313   CONTINUE
          ELSE
           IYI = ILI - 3
        WRITE(6,*)'Y',IYI,CCHAR(ICHAR),LENGTH,'* TAU'
        WRITE(13,1991)RYY(IYI)
           DO 1343 K = 1, ISTEP
             I = INTVAL*K
                RI= FLOAT(I)/FLOAT(IDIV)
            IF(ECOR(K,IYI).LE.0)THEN
           WRITE(13,1991) -999999
           WRITE(11,1991) -999999
            ELSE
           WRITE(11,1989)RI,TAB,LOG(ECOR(K,IYI))
           WRITE(13,1991)LOG(ECOR(K,IYI))
            END IF
 1343      CONTINUE
           END IF
      STOP
      END
```

```
C ************************************************
C    PROGRAM NAME TAU.FOR
C ************************************************
      INTEGER INTVAL, SMPMAX, LENGTH, ISTEP, IDIV
      INTEGER IS, ICHAR
      REAL Y(2000),X(2000), TAU, R2 ,XDATA(100)
      REAL FDATA(100)
C ************************************************
 1976   FORMAT (4I5)
 1990   FORMAT (F12.9)
 1991   FORMAT (2I5,F12.7)
C ************************************************
C    NOTE, THE LINES WITH READ (13, ) ARE DATA READ
C    FROM A CONTINUATION FILE ON THE IBM 3090.
C ************************************************
      READ(13,1976)INTVAL,SMPMAX, LENGTH, ISTEP
      READ(13,1976)IDIV,ICHAR,   ILI
      READ(13,1990) R2
C ************************************************
      DO 100 IS = 1, ISTEP
         READ (13,1990) X(IS)
         Y(IS)= FLOAT(IS*INTVAL)/FLOAT(IDIV)
 100  CONTINUE
      IF(ICHAR.EQ.1) THEN
      WRITE(6,*)'FREELY JOINTED RANDOM WALK'
      END IF
      IF (ICHAR .EQ. 2) THEN
         WRITE (6,*) 'SELF AVOIDING WALK'
      END IF
      IF(ILI.LE.3)THEN
      WRITE(6,*) 'FOR EIGENVALUE L',ILI
      ELSE
      WRITE(6,*)'SPHERICAL HARMONIC CALCULATION:'
      IYI = ILI-3
      GOTO(123,234,345)IYI
 123  WRITE(6,*)'Y(1)= SIN(TH)COS(TH)SIN(PH)'
      GOTO 346
 234  WRITE(6,*)'Y(2)= SIN(TH)COS(TH)COS(PH)'
      GOTO 346
 345  WRITE(6,*)'Y(3)= 3(COS(TH))**2 - 1   '
 346  CONTINUE
      END IF
      WRITE (6,*) 'LENGTH =  ',LENGTH
      WRITE (6,*) '<R**2> =  ', R2
C ************************************************
      R = -1.00
       DO 200 IS = 1, ISTEP
        IF(X(IS).LE.R)THEN
```

```
            IJ = IS
            GOTO 300
           END IF
 200      CONTINUE
 300      CONTINUE
          WRITE(6,*) IJ
          WRITE(6,*) ISTEP
            ICOUNT = IJ + 1
            ISTART = IJ - 2
          IF(ISTART.LE.0)THEN
            ICOUNT = 2*(IJ-1)
            ISTART = 1
          END IF
         DO 400 IS = ISTART, ICOUNT
            INUM = ICOUNT -2*(IS) + ISTART + 1
          IF (INUM .LT.3)GOTO 900
            ICOUNT2 = INUM + IS - 1
            NDATA = ICOUNT2 - IS + 1
         DO 500 IT = IS, ICOUNT2
            IP = ICOUNT - IT + 1
            XDATA(IP) = X(IT)
            FDATA(IP) = Y(IT)
 500    CONTINUE
C ***************************************************
C     QDVAL IS AN IMSL FUNCTION WHICH USES QUADRATIC
C     INTERPOLATION TO DETERMINE TAU.
C ***************************************************
           TAU=  QDVAL(R,NDATA,XDATA,FDATA,CHECK)
         WRITE(6,*)'TAU=',   TAU
 400    CONTINUE
 900    CONTINUE
        STOP
        END
```

VITA

Thomas Donald Hahn was born in Pittsburgh, Pennsylvania, on May 12, 1965 as the seventh of eight children. He graduated from Moon Area High School in June 1983. From August 1983 to May 1987 he studied at The Pennsylvania State University. He received a bachelors of science degree from Penn State in Polymer Science. He then enrolled as a graduate student at the University of Tennessee, Knoxville in September of 1987 and completed his M.S. requirements in Chemistry in the Summer of 1989.