Diffusion in Amorphous Media

Thesis by

Mihail S. Iotov

In Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

1998

(Submitted December 1, 1997)

Acknowledgment

First and foremost I would like to thank my advisor Professor William Goddard for providing continual guidance and advice. Many thanks also go to Siddharth Dasgupta, whose stimulating discussions have kept me focused on the diffusion project. I express gratitude to other researchers from Dr. Goddard's: collaborators Guanghua Gao, Michael Belmarez, Tahir Cagin, Vaidehi Nagarajan, Seichi Kashihara and friends Francesco Faglioni, Chris Kankel, Ersan Dermilap and Hao Li.

Finally, special thanks to my wife, Dobrina, for being extremely supportive throughout my stay at Caltech.

Abstract

The goals of this research are twofold: First, to develop methods and tools for studying problems in chemistry, material science and biology, as well as accurate prediction of the properties of structures and materials of importance to those fields. Second, use those tools to apply the methods to practical problems. In terms of methodology development this thesis focuses on two topics: *One*: Development of a massively parallel computer program to perform electronic, atomic, molecular levels simulations of problems in chemistry, material science and biology. This computer program uses existing and emerging hardware platforms and parallel tools and is based on decades long research in computer modeling and algorithms. We report on that development in Chapter 3. *Two*: Development of tools for Molecular Dynamics simulation and methods and tools for course-grained meso-scale modeling of transport properties and especially diffusion of gas penetrants in polymers. We have formulated a *new method* for extracting coarse-grained information from short (0.2-0.5 nanoseconds [ns]) MD simulations and use this in a meso-scale simulation to calculate diffusion constants in polymer matrices. This is a grid-based method, which calculates the average probability of each grid point of being a void and performs constrained and biased Monte Carlo (MC) dynamics to reach much longer time regimes than possible in MD. The MC method mimics the three regimes of mean square deviation (MSD) behavior seen in MD, thus accounting for the proper mobility of the voids and the compressibility of the polymer matrix. Theoretical discussions and justification for the method is presented in chapter 6. Initial results on He diffusion in a low-density polyethylene (PE) matrix are

presented in chapter 7. The behavior at different temperatures follows closely the trend observed from calibrating long term MD for this particular system.

Table of Contents

CHAPTER 6 THE MONTE CARLO VOID METHOD

Chapter 7

DIFFUSION OF GASES IN AMORPHOUS POLYMERS:

THE MONTE CARLO VOID METHOD BASED ON MOLECULAR DYNAMICS

APPLICATION TO HE DIFFUSION IN POLYETHYLENE

List of Illustrations and Tables

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Figure 10a. **Einstein Diffusion dependence on radius of channel connecting r=20 spheres at 100 distance from each other.**

Figure 10b. **Anomalous exponents for the same structure.**

Figure 11a. **80 units long finite linear stretch.**

Figure 11b. **infinitely long 1D line.**

Figure 11c. **80 units wide infinite strip (a direct product of 11a and 11b)**

Figure 12. **20x20 infinite rectangular cube.**

Figure 13a. **Square displacement from the beginning (not averaged) for a system similar to the one in figure 2 with channel with of two.**

Figure 13b. **Square displacement from the beginning (not averaged) of an MD run of He in PE at 450K.**

Figure 14a. **Random walk Bias at q=0.5;**

Figure 14b. **Random walk Bias at q=0.0 (no bias);**

Figure 14c. **Random walk Bias at q=1.0 (no backward motion);**

Figure 14d. **Random walk Bias at q=0.5 and the term squared to point even sharper in forward direction.**

Figure 15. **MD short-range trajectory.**

Figure 16. **Fast momentum vs. no momentum. The ballistic region is apparent for the fast momentum case and is lacking for the no momentum case.**

Chapter 7

Chapter 1 Introduction

1.0 Polymers as example of amorphous media

The most common example of amorphous media considered in this thesis is a polymer.

1.1 Why Diffusion in Polymers

Various industrial applications of polymers involve diffusion of gases through polymers[i]. Membrane separation of gases in the gas and oil industry has become as wide spread as more traditional methods based on absorption, pressure-swing adsorption or cryogenics. The membrane process has certain benefits compared to the cryogenic process, for example lower investment cost and easier operation. Very important applications in the food industry involve impeding the diffusion of gases through thin polymer films, commonly used as packaging food wrappers or protective coatings. More applications are anticipated in the burgeoning field of biotechnology, such as biosensors, controlled release and bioreactors. Moreover, the effects of the interaction between glassy polymers and small molecules are of practical interest to chemical engineers, due to the inherent sorption and transport of gases present in most processes they encounter. The design of new polymer materials would benefit greatly from reliable predictions of the diffusion properties from atomic models. Both experimental and theoretical data have been unreliable in the past. Experimental data have not been consistently reproduced between different reporting laboratories. Theoretical predictions have achieved an order of magnitude accuracy at best. Moreover, theoretical methods that are not based on molecular dynamics do not account for the atomistic details that distinguish one polymer

from another. On the other hand, molecular dynamics methods, which take into account atomistic details, can not achieve the time scales of simulation necessary to compute diffusion reliably. We have addressed this problem in this thesis and offer a solution to the challenge of large time scale simulation of gas diffusion in polymers.

1.2 The need for coarse-grained methods

The polymers typically exist in one of two states: glassy and rubbery. The underlying difference between those two states comes from the amount of rotation around the chain axis, which the polymer segments experience. In the glassy state there is very limited amount of rotation going on. Glassy polymers are usually dense, hard structures with very little internal void space through which a penetrant can move about. Diffusion rates are usually very small, but depend largely on the size of the penetrant.

In rubbery state, the polymer chains exhibit a lot more mobility and are thought to participate in the diffusion process themselves. More free volume is available to potential penetrants to move through. Thus diffusion rates are generally higher in rubbery state than in glassy state. At sufficiently high temperatures a gas molecule in the polymer melt might behave similarly to being in a liquid.

In molecular dynamics (MD) simulations we usually place a particle in the polymer and track its motion. The small diffusion rates in the glassy state mean that it will take a very long time for the penetrant particle to diffuse a significant distance away. Without the tracked particle moving from its initial position about 100 angstroms ($\overset{o}{A}$,) it is difficult to obtain reliable statistics. This requires simulation times of the order of microseconds to milliseconds. Using state-of-the-art simulation programs on today's

supercomputers allows up to tens or hundreds of nanoseconds of simulation. Thus, it is desirable to develop a method to simulate the motion of the tracked particle on a much larger timescale. This method will, by necessity, involve a course-grain description of the underlying polymer, extracting only features, which are relevant to the penetrant motion. This reflects the specifics of the gas diffusion process in polymers, which differs a lot from the diffusion process in gases or liquids in that there is very low concentration of penetrant particles which move among the very different, relatively rigid polymer atoms. This difference in the underlying structure leads naturally to attempts to find methods for course-graining of the system. One such method is the *Monte Carlo Void Method*, being described in this thesis. It is briefly described in the next section.

1.3 The Monte Carlo void method

The basic idea behind the *Monte Carlo Void Method* is to replace the description of the motion of the polymer with description of the dynamics of the free volume inside that polymer. During that coarse-graining some information about the polymer motion will be lost. However, the free volume information that is extracted will be sufficient and relevant in providing an input to a framework for describing the penetrant motion alone via a much faster method, thus allowing much longer time scales to be achieved.

1.4 The challenges

Attempts have been made to explore the motion of a particle within the free volume of a polymer. De Gennes[ii] compared the motion of a tracer particle in a lattice

where a number of the sites are blocked by other particles with the random walk of an ant in a labyrinth. Using molecular mechanics it is possible to extract the free volume of a polymer matrix and perform the ant-in-a-labyrinth kind of motion in it. It is easy to see that this approach lacks dynamics. The thermal motion of the polymer is not taken into account. In that case, a finite size penetrant often gets blocked in the polymer chains without performing any long-range motion (see description of failed attempts, which considered a rigid polymer matrix in Section 5.2.2.) Our idea is to account for the polymer motion by introducing proper averaging of the free volume over the dynamic evolution of the polymer. A few of the questions that need to be answered are:

- What time scale should we use for averaging the polymer motion?
- How often shall we sample the polymer?
- How do we extract the free volume of static snapshot of the polymer?
- What is the exact procedure for averaging the above free volumes?
- How do we use that average to jumpstart a random walk (e.g., what are the probabilities for the transitions)?
- How do we translate back the random walk step to real time?
- How do we account for differences in particles that have same geometry but differ in electric charge?

The rest of the thesis addresses the above issues. The most difficult issue is about translating the random walk step to time. We define a conversion timescale, in which one random walk step corresponds to a certain amount of time. To determine its value we develop a framework of matching random walk simulations to short term molecular dynamics simulation, based on comparing the mean-square deviation functions computed

for the penetrants. We are able to develop a random walk simulation, which matches very closely the MD simulation. An example of mean-square deviation plots from MD and from Monte Carlo simulations is shown in Figure 1.1. What is remarkable is that the shape of one of the curves agrees very well with the MD curve. The curve labeled MD is obtained from MD simulations for 10ns, which took about 3 months. The other curves are obtained through Monte Carlo simulations at various probe radii (the one labeled 07 at 0.7 10^{-10} m, etc.) Those runs took usually a couple of hours. We see that the 1.3 curve agrees with the MD curve everywhere (e.g. both in the initial anomalous region and in the linear Fickian region; these plots do not show ballistic area.) The other curves do not agree with the MD curve in any region. This implies that we have a method for obtaining the MD 10ns curve, without running MD for all this time. First we run MD for about 400 ps, to obtain equilibration (200 ps) and enough sampling (another 200 ps) to extract the void data. Then on the extracted voids we run Monte Carlo Dynamics (MCD) with various probe radii in the range (0.7-1.8 10^{-10} m.) For each of those we compute the mean-square-deviation curves (MSD) One of the MCD MSD curves will match the MD one very closely in shape. From this match we obtain the correct probe radius, the conversion scale in terms of time per MC step and the predicted diffusion coefficient.

1.5 Summary of the other sections

A reader who is interested in the applications can go directly to Chapter 7. Chapter 2 reviews very briefly the Molecular Dynamics methods of computer simulation. It also describes the parameters that were used in the simulation model. Chapter 3

describes the development of the general-purpose computational tools for molecular dynamics. Chapter 4 describes the graphics user interface toolkit for diffusion simulation and analysis of light gases in polymers. Chapter 5 contains the theoretical background and brief review of theoretical models for diffusion. Chapter 6 is a self-contained paper which explores, through the use of random walks on idealized model artificially generated free volumes, the relevance of random walks through the generalized free volume of a polymer for prediction of diffusion properties. Chapter 6 also defines in more detail the Monte Carlo void method being developed in this thesis. Chapter 6 is a multiauthor paper and was exclusively written by the thesis author with help from Siddharth Dasgupta and William A. Goddard III. Chapter 7 is another self-contained paper, which describes in more detail the application of the methodology developed in this thesis to diffusion of gases in amorphous polymers and more specifically to Helium diffusion in polyethylene.

Chapter 7 is a result of collaboration between the author of the thesis, other graduate students and staff members under the supervision of Professor W. A. Goddard. Guanghua Gao did the development of the torsional forcefields for Polyethylene (PE). Michael Belmarez generated PE structures and performed many molecular dynamics simulations of PE and especially studied the torsional flips in the polymer chains. Seiichi Kashihara brought a practical approach to the problem by analyzing industrial polymers, copolymers and mixes. Most of the method development and the particular analysis of He diffusing in PE was performed by the thesis author with help from Siddharth Dasgupta.

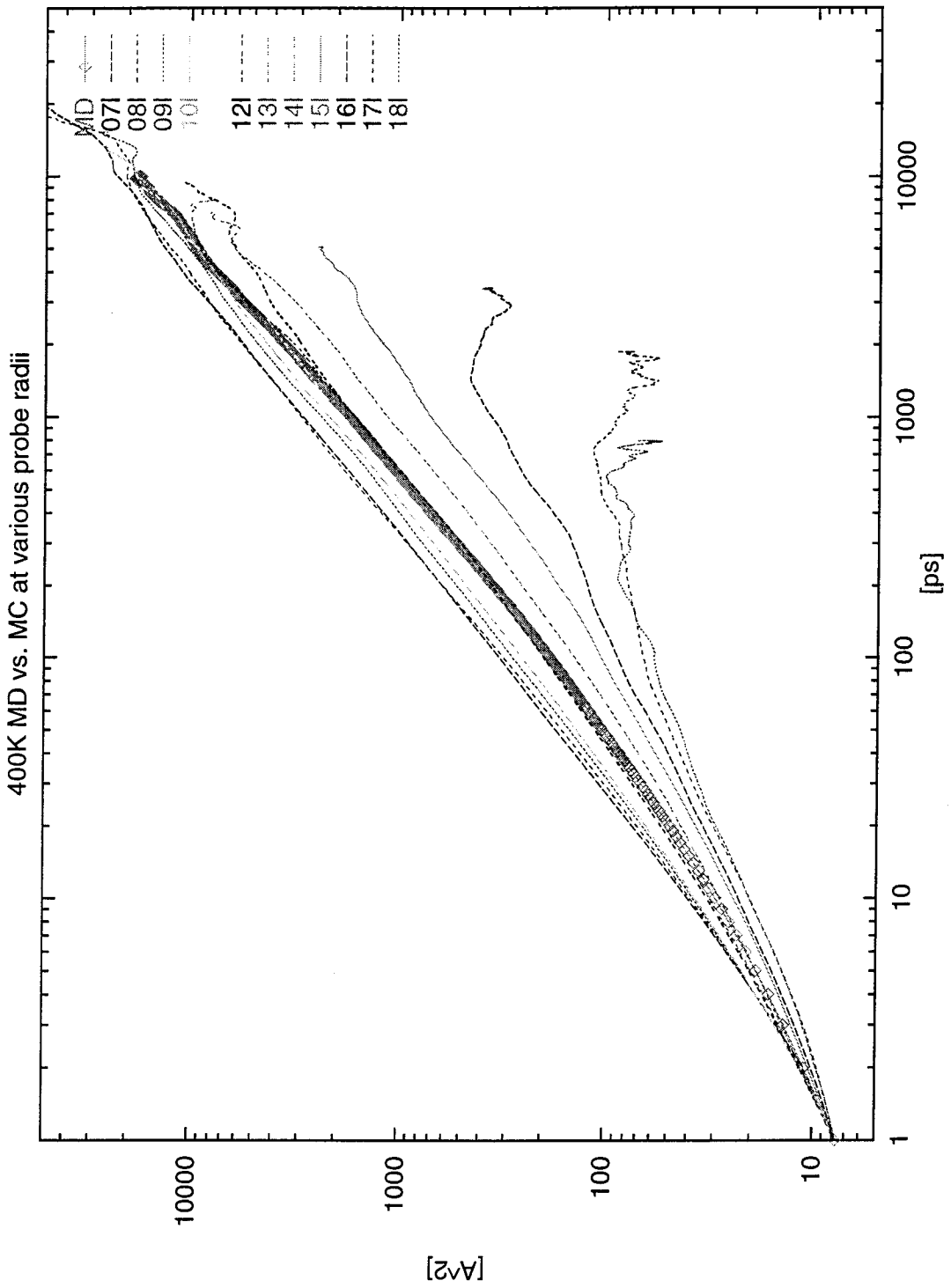


Figure 1.1. Mean square displacement curves obtained from a MD run at 400K at MC runs at various temperatures.

---

[i] Vieth, W.R., *Diffusion In and Through Polymers*, Hanser Publishers, 1991

[ii] De Gennes, P.G., La Recherce, **7**, 919 (1976)

Chapter 2 Molecular Dynamics

2.1 Introduction

Molecular dynamics has become a powerful tool for simulating chemical and biological systems of sizes much larger than what is possible with first principle quantum mechanics calculation. Systems of up to millions of atoms have been simulated in atomistic detail. Forcefields are usually fitted to experimental data (e.g., vibrational frequencies) and detailed results from quantum mechanical calculations. With the increase of available computational power and the development of faster quantum mechanical simulation methods (e.g., Jaguar[i],) it is possible to obtain better than ever calibrated forcefield parameters optimized for polymers by fitting to quantum mechanics computations on chains involving a few monomers. More powerful methods (e.g., CMM[ii], Ewald summation[iii]) for non-bond, long-range interactions (e.g., electrostatic and van der Waals) have also become widely implemented in simulation software. On the other hand, better methods have been developed for numerical integration of the equations of motion. By introducing coupling with external heat or pressure bath, it is possible to obtain more realistic simulations leading to generation of constant volume and temperature and Gibbs (constant volume and pressure) ensembles[iv] in addition to the microcannonical ensemble (constant energy and volume.) All of the above simulations also hold the number of particles constant. The constant energy and pressure ensemble (known as NTP or TPN) is of especial importance, since it corresponds to the conditions in which most experiments are conducted. Molecular Dynamics methods have been described extensively[v,vi,vii] so we will only touch on the relevant points for this study.

2.2 Force fields

All of our simulations use standard forcefields, such as DREIDing[viii], AMBER[ix], or the Universal Force Field (UFF[x]). For detailed description of the forcefield for polyethylene, see Section 2.2 in Chapter 7. For polyvynildenechoride we used the UFF forcefield for the bond and angle parts, DREIDing II for the van der Waals parts and the torsional potentials were again determined from quantum chemical calculations. Forcefields for mixtures of two polymers, such as PVC and PVDC, also used torsions calibrated from quantum mechanic computations on lower mass alkanes with selected hydrogen atoms replaced by Cl.

2.3 Integration parameters

All of our simulations used 1 fs as a timestep, because of the presence of explicit hydrogen atoms. Using united atom approach in which the hydrogen in the methyl group is collapsed onto the carbon atom allows for larger timesteps, but it leads to about two orders of magnitude overestimation of the diffusion constant, primarily because of overestimation of the free volume. This is discussed further in Chapter 5, Section 5.2.2. We used the Nose-Hoover method for heat bath coupling to keep the system at the desired temperature. The heat bath coupling parameter, which controls the rate of transfer of kinetic energy from the system to the bath and vice versa and thus the rate of equilibration, was set to 10 times the timestep, or 10 fs. Previous studies have shown that this value is small enough to assure fast equilibration (of the order of a few hundred

steps), while by not being too small allows the system to explore different states in phase space, thus ensuring a degree of ergodicity[xi].

2.4 Generation of polymers

All polymers were generated using the *Amorphous Polymer Builder* module of Cerius2[xii], which implements the Rotational Isomeric State theory to select the torsional distributions from specified gauche-trans barriers. This build procedure is followed by minimization to remove occasional bad contacts, which lead to huge non-bond energies between atoms that happen to be too close. Then the structure is equilibrated for at least 100ps. If the initial structure was simulated at 400K temperature, when we lower the temperature to 350K we again equilibrate for 100ps, before doing a production run. Then we take the structure from the end of the production run and equilibrate it at 300K, proceeding in similar manner until we obtain production runs over a range of temperatures.

---

[i] http://www.psgvb.com/publications.html, Schrodinger, Inc., Portland, OR

[ii] Ding H.Q., Karasawa N. and Goddard III W.A., *J.Chem.Phys.*,**97**, 4309-4315(1992);
Ding H.Q., Karasawa N. and Goddard III W.A., *Chem. Phys. Lett.*, **196**, 6-10 (1992)

[iii] Karaswa N. and Goddard III W.A., *J.Phys.Chem.*, **93(21)**, 7320-7327 (1989)

[iv] Nose S., *Progress of Theoretical Physics Supplement*, **103**, 1-46(1991)

[v] Allen M.P. and Tildesley D.J., *Computer Simulation of Liquids*, Oxford University Press, 1987

[vi] Biograf 3.2.1 Reference Manual, Molecular Simulations, Inc., Burlington, MA, 1993

[vii] Lim, K.T., Brunett S., Iotov M., McClurg R., Vaidehi N., Dasgupta S., Taylor S. and Goddard, W.A. III, *Journal of Computational Chemistry*, **18**(4) 501-521 (1997)

[viii] Mayo S.L., Olafson B.D., Goddard III W.A., *J.Phys.Chem.* **v94**(26), 8897-8909(1990)

[ix] Weiner S.J., Kollman P.A., Nyuyen D.T and Case D.A., *J.Comp. Chem.*, **106**, 230-252(1986)

[x] Rappe A.K., Casewit C.J., Colwell K.S., Goddard III W.A. and Skiff W.M., *J.Am.Chem.Soc.*,**114**(25), 10024-10035(1992)

[xi] Lim, K.T., Ph.D. Thesis, Caltech, 1995

[xii] http://www.biosym.com, BioSym, San Diego, CA

Chapter 3 Massively Parallel Molecular Simulation Program

3.1 Introduction

The work on developing our primary simulation tool builds up on many years of work in the Materials and Process Simulation Center at Caltech[i] and particularly the programming efforts of Kian-Tat Lim[ii] and is more extensively described by Lim et al.[iii]

More recently, the MPSim program has been used for a wide variety of other general purpose molecular dynamics simulations, including study of nanotubes by Gao[iv] and study of the mechanism of phosphoglycerate kinase (PGK) enzyme by Vaidehi et al.[v]

3.2 Structure of the MPSim program

The basic structure of the MPSim program is shown on Figure 3.1. Figure 3.2 depicts the structure of the energy evaluator.

3.3 Message-passing implementation

The original design of the message passing implementation was to use active messages. Active messages were the preferred model for programming the experimental fine grain J- and M-machines from MIT. The low latency of the active messages programming model, along with its natural, asynchronous, multithreaded style of programming made this model a prime candidate for preferred programming model for the then-upcoming generations of supercomputers.

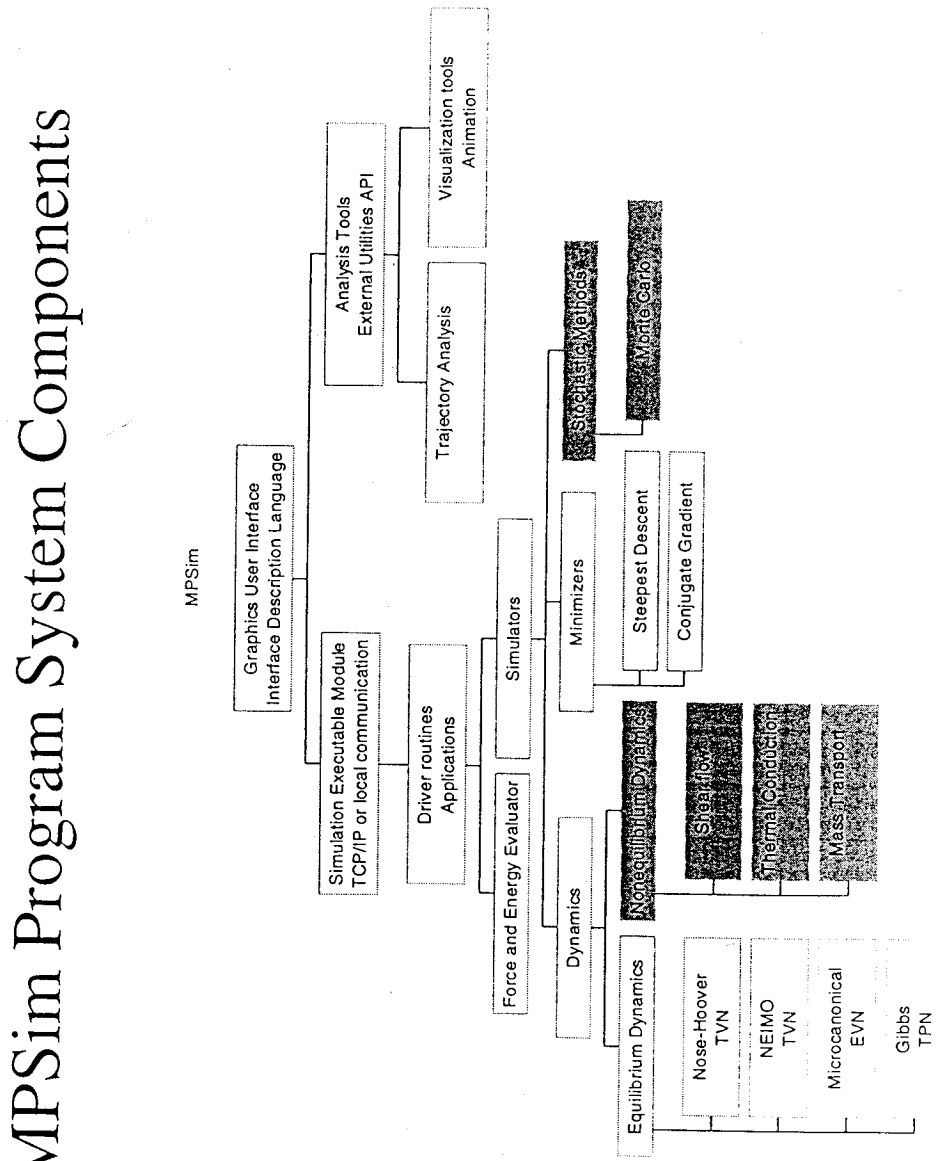


Figure 3.1 MPSim program system components.

Energy and Force Components

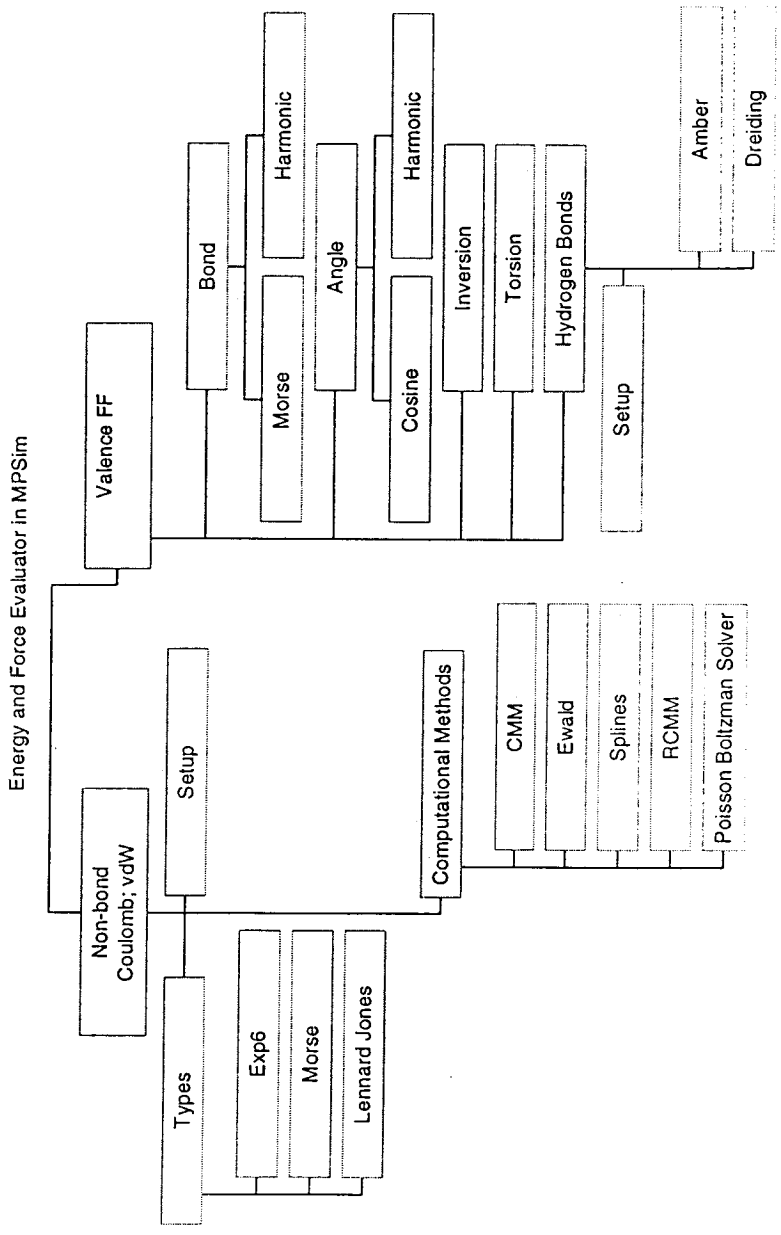


Figure 3.2 MPSim program energy and force components.

On the other hand, recent development in parallel programming has placed emphasis on more traditional message-passing models, such as PVM, NX, and MPI. MPI especially has been adopted as the *de facto* standard with support from all vendors. Native active-message libraries did not materialize for machines to which we have access (Cray T3D, Intel Paragon, and Silicon Graphics). Therefore, we have decided to proceed with using the MPI message-passing library. This ensures high degree of portability and allows for future optimization by using more advanced collective operations provided by the MPI library.

As a first implementation we have preserved the active-message paradigm and especially the "pull" strategy as described in Lim et al. Here is an example of using the "pull" strategy to compute the kinetic energy of a system.

For each node compute its local kinetic energy.

If I am not node 0, then {

Send my local kinetic energy to node 0 and

Wait for receiving back the global kinetic energy.

}

Else {

Start a loop for receiving from N-1 nodes a double number and add them together.

Broadcast the global energy to all nodes.

}

The above algorithm can be optimized by using the MPI global operations, such as MPI_Reduce. The flow chart for the current implementation using the simplest MPI calls is given in Appendix A.

3.4 Shared memory implementation

Portable shared memory programming model

Shared memory programming models have the advantage of greatly simplifying the initial parallel-programming task. However, attention must be paid to the actual location of data, to ensure high parallel efficiency without extensive rewriting of code. Another challenge comes from the fact that there is no standard for shared memory programming. In contrast there are many standards for message-passing programming, e.g., MPI or PVM. Since portability is extremely important for us, we have attempted to formulate a simple shared memory programming model which presents to the program standardized interface and hides low level machine specific features from the rest of the program. The idea is that the standardized interface consists in a number of machine independent function calls, which then provide the necessary machine dependent code in the body of the function. A shared memory programming model requires two kinds of software support: locks and barriers. Barriers are generally provided by all thread libraries, so we just implement a *do_sync()* function which does barrier check in and immediately follow it by check out. Locks are handled differently on different machines. For example, on SGI architectures they need to be allocated from arena space, whereas on KSR machines any processor can lock any 128-byte subpage of memory.

To ensure portability we define a set of macros:

lock_structure(a)

unlock_structure(a)

lock_variable(a)

unlock_variable(a)

Then on each machine we expand those macros to do the proper machine-dependent code. The exact expansion is given in Appendix B.

Care must be taken also in dealing with private data, which is generally used to compute global sums. (Each CPU computes its own value for the data, which are later added together globally.) The following macros are defined to help with global sums:

```
def_local (Type,a);/*defines a of type Type and local for each CPU*/
local(a) = 5 ;      /* assigns our local copy of a to 5 */
global_sum (g,a) ; /* sums all local a's globally */
```

Efficiency considerations for shared memory on various machines

Just as for sequential programs it is important to keep data local in order to avoid cache thrashing (also known as false cache sharing.) On the KSR computer 128-byte subpages tend to migrate to the processor that needs them most often. That means we should make sure we don't have two processors accessing the same 128-byte subpage with approximately the same frequency.

On the SGI Power Challenge and Origin architecture a memory page is 16K. This presents a particular problem with the implementation of private data. On the SGI machine there is no "private" primitive so we implement private variables as arrays. Our implementation of the *"local(private_a)"* macro expands to *private_a [my_cpu]*, where *private_a* is defined as an array of dimension the maximum number of CPUs. To ensure efficient memory management it is necessary that each element of the array be of size 16K (one page.) Thus it is more efficient to group all private variables in a structure and then have just one array of structures.

3.5 Implementation on machines that provide one-sided communication

Some machines, like the Cray T3D, provide one-sided communication (shared memory get and put in Cray's parlance). This programming model is somewhere in the middle between the shared memory and message passing programming models. The algorithm presented in Figure 3.3 was designed to perform non-local updates. It can be shown that this algorithm terminates for all processors and produces the correct updated value of X. Moreover, for most of the cases when there is no contention for simultaneous update from two processors, this algorithm is much faster than the traditional, message passing approach which would require sending, receiving and processing a request.

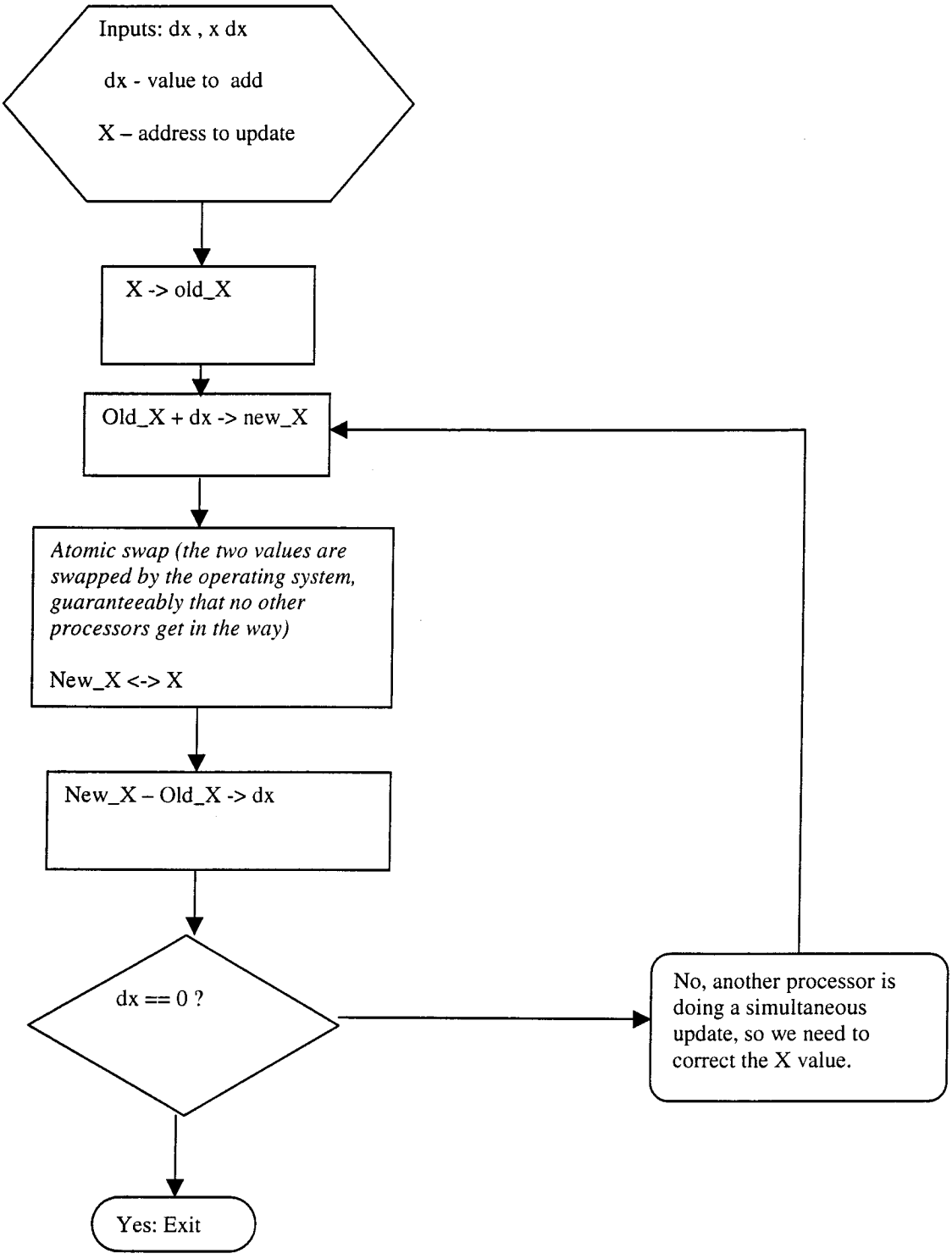


Figure 3.3 Common location update algorithm for architectures that provide one-sided communication.

3.6 Performance results

3.6.1 Performance on shared-memory architectures

The scaling performance of the MPSim program on the Kendall Square Research shared memory computer is shown in Figure 3.4a.

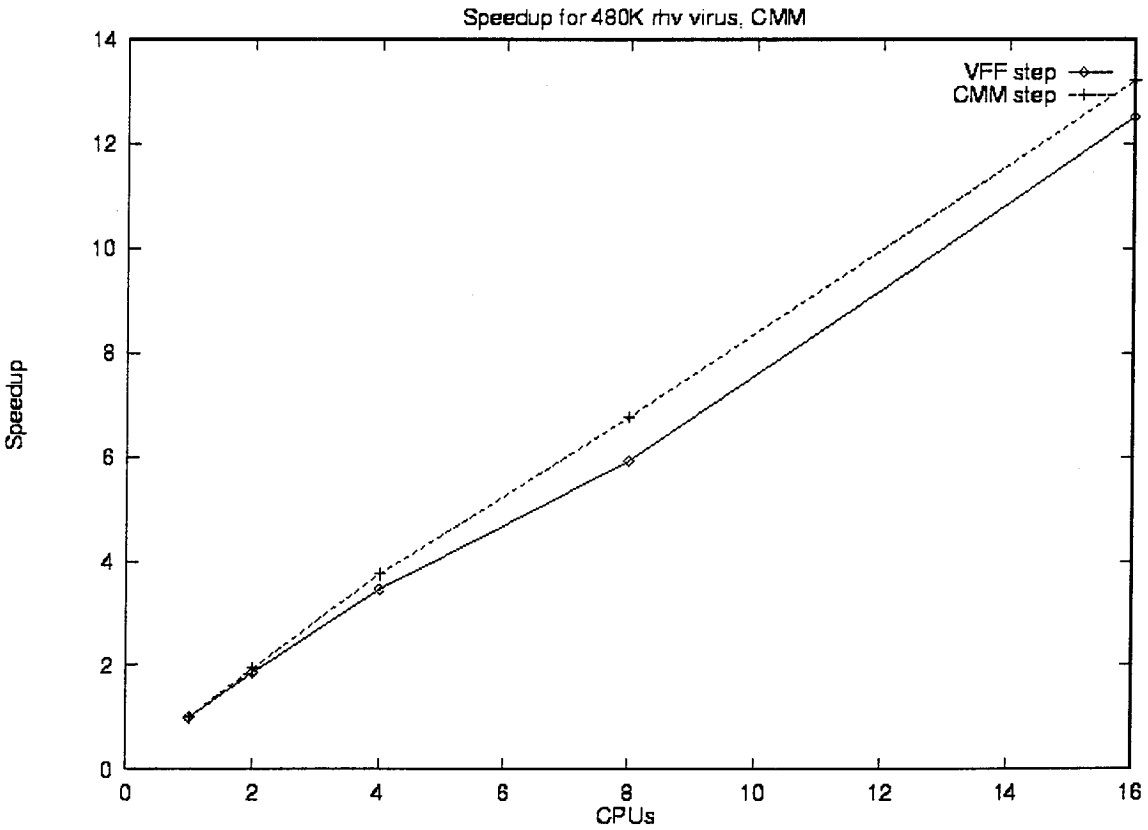


Figure 3.4a Performance on the Kendall Square computer.

3.6.2 Performance on message-passing architectures

The performance of the MPSim program on a number of message passing architectures is shown in Figure 3.4b.

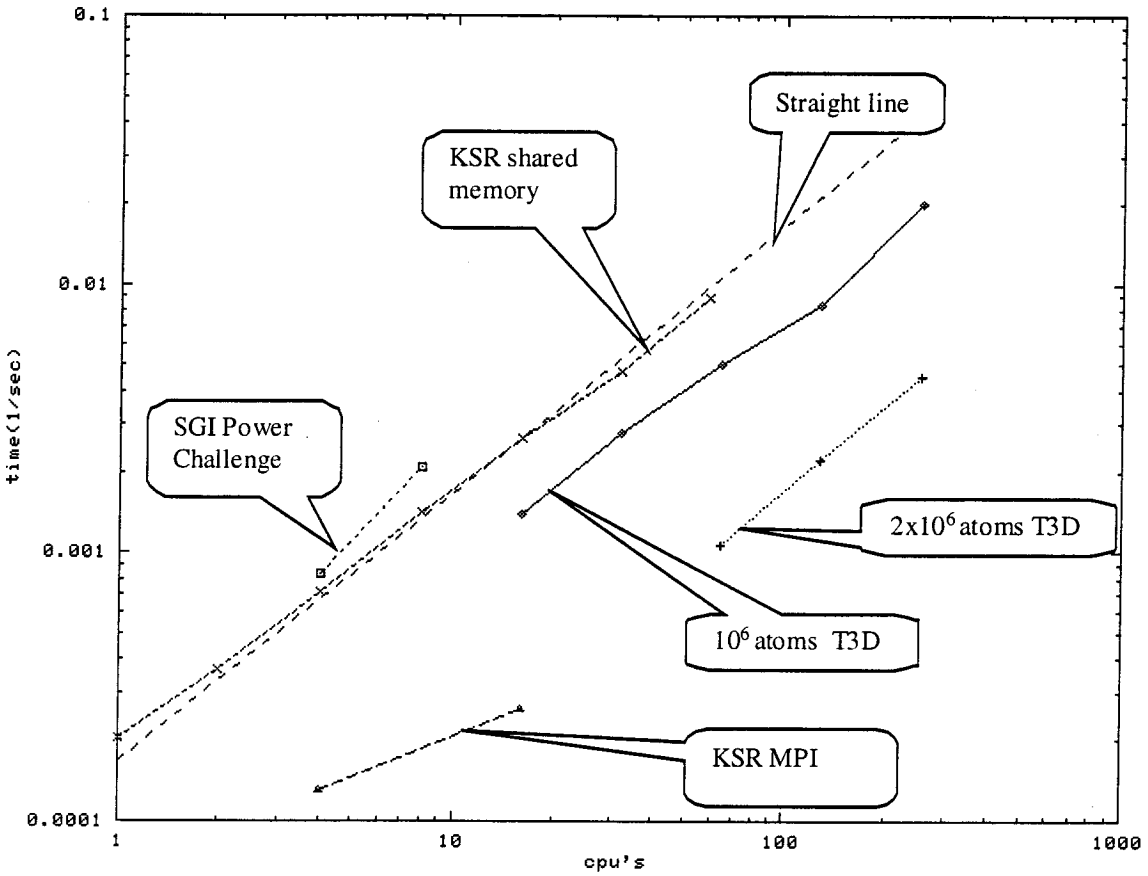


Figure 3.4b. Comparative performance on various architectures using message-passing libraries.

3.7 Programming interface description for addition of new modules

The program provides general-purpose force and energy evaluation engine and a set of bookkeeping utilities, so that other modules can be added with relative ease. The flowchart on Figure 3.5 is an example how a driver that implements a different integration method (these are sometimes called *movers* since they move the atoms to new positions). A pseudo-code implementing this algorithm is given in Appendix C. It is also possible to interface to the program via a macro language, which is described in Appendix D.

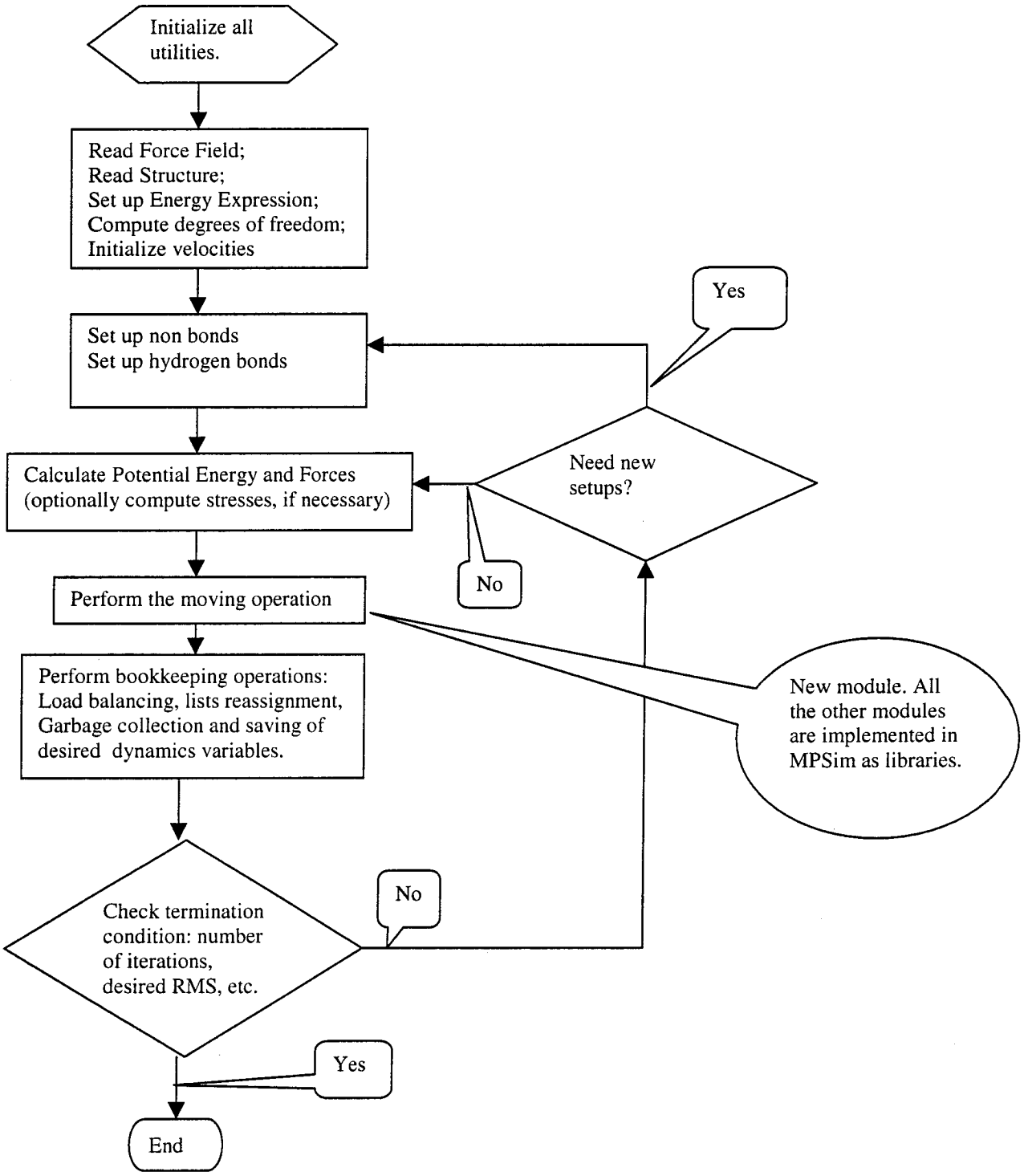


Figure 3.5 Flowchart for the API interface to the MPSim program.

[i] http://www.wag.caltech.edu/research.html; see in particular the annual report sections.

[ii] Lim K.T., Ph.D. Thesis, Caltech, 1995

[iii] Lim K.T., Brunett S., Iotov M., McClurg R., Vaidehi N., Dasgupta S., Taylor S. and Goddard III W.A., *Journal of Computational Chemistry*, **18**(4) 501-521 (1997)

[iv] Gao G., Ph.D. Thesis, Caltech, 1998

[v] Vaidehi N., Goddard II W.A., manuscript in preparation

Chapter 4 Suite of Programs for Diffusion Simulation

This chapters documents the suite of programs for analysis and visualization of the free volume in a polymer as well as performing Monte Carlo diffusion simulations with it. Those programs are unified in a graphics user interface for the X windows system. Description of the command line interface is given in Appendix E.

4.1 File formats

Table 4-1 summarizes the most commonly used file formats.

Table 4-1 File formats used by the void manipulation utilities

Extension	created by	description
.bgf	POLYGRAF	biograf coordinate file format
.par	POLYGRAF	biograf force field file format
.voids	vgen, voids	description of voids in structure.
.track	MPSim	tracks specific atoms more often
.snap	MPSim	tracks all atoms but less often
.cvoid	cvoid	description of clusters in .voids file

The .par and .bgf files are described in the Polygraf Reference Manual. The .track and .snap files are described in another chapter of this thesis.

The void files describe generalized free volume (described in Chapter 6)

Utilities are available for manipulation of void files as scalar fields.

4.1.1 Generalized free volume format file

The format is one line per void point, listing the x, y and z coordinates of the point and then an integer describing the void multiplicity at this point. To obtain the representation described in Chapter 5, this integer needs to be rescaled by the number of snapshots that were used in obtaining that file. This and other useful information can be found in the trailer of the file (in order to preserve compatibility with existing software a trailer, rather than a header, was chosen.)

Here is an example of the trailer:

```
n_points = 28186,  total volume = 615.444
Created by iotov on teijin on Fri Feb  7 03:05:45 PST 1997
by extracting from MD run data.
FF              /net/ksr2/temp1/iotov/400K/harmpethmcxxlj-He.ff
BGF             /net/ksr2/temp1/iotov/400K/PEdynRealHe1ns.bgf
Penetrant Diameter      2
Center Mode             2
Snapfiles used  He_400K.snap9804999 He_400K.snap9809999
He_400K.snap9814999 He_400K.snap9819999 He_400K.snap9824999
He_400K.snap9829999 He_400K.snap9834999 He_400K.snap9839999
He_400K.snap9844999 He_400K.snap9849999 He_400K.snap9854999
BGF file was not considered.
First 2408 atoms processed.
```

```
nfiles  10
```

The nfiles field shows how many files were processed. Center mode specifies if soft core or hard core potential is being used. n_points can be used for parsing where the void data ends. The volume is computed taking into account the unit cell size. Since we have used 100x100x100 grids, in this case, this translates into about 28% free volume.

4.2 Void extraction facility

The void extraction program creates voids from a single file structure or multiple snapshots obtained from dynamics simulation of the same structure. The output is a generalized free volume format file. The inputs are the names of the files to process, the hypothetical penetrant diameter and a center mode flag, which described what kind of potential is used for the van der Walls repulsion interaction (soft or hard core.)

4.3 Void creation utilities

These utilities create artificial voids useful in modeling. Voids can be in spherical or cylindrical form. The format of the description file is:

```
VOID  index       center_x center_y center_z  radius
RATE  ind1        ind2  radius    disp_x disp_y disp_z
```

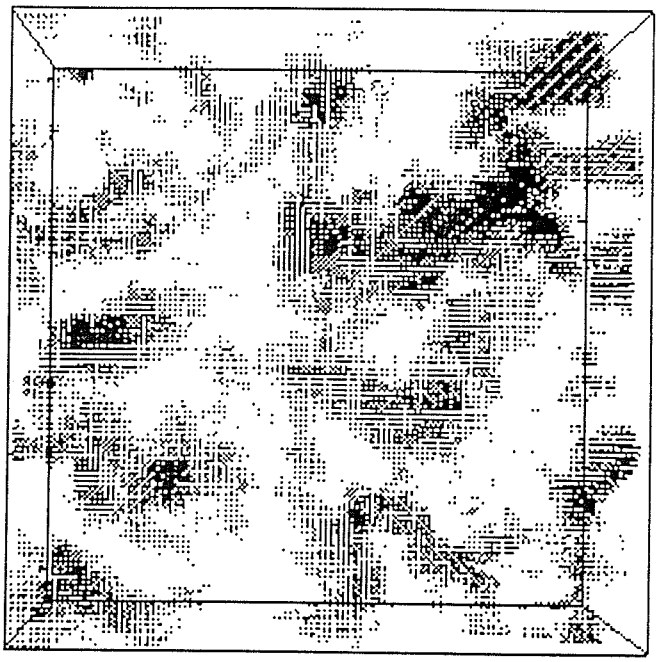


Figure 4.1 Example of void visualization.

VOID defines a sphere with the given center and radius.

RATE defines a cylinder of the given axial radius going through the centers of voids ind1 and ind2 and of height equal to the distance between those two centers. The displacement shows that the cylinder is to be connected to the translated image of void ind2.

4.4. Void analysis utility

This utility takes as input a generalized free volume file. It separates the voids into connected clusters. It also prints statistics for them, checks for percolation, and outputs a cluster void file, in which the number of the void they belong to labels the points. That file can be visualized later. An example is shown in Figure 4.1.

4.5 MSD and SD computation

Mean Square Deviation (MSD) computation proceeds according to Section 2.3 in Chapter 6. This is inherently an $O(n^2)$ process and thus is the bottleneck of the computation for large number of Monte Carlo steps with processing trajectory of 100,000 steps taking as long as 1.5 hours on a 250 Mhz Silicon Graphics R4010 processor. Square Deviation (SD) without averaging is computed in the same way.

4.6 Angle analysis

To verify that a penetrant has explored enough of the available space we computed the angular statistics of all pairwise vectors between positions of the penetrant for all timesteps. Then we mapped the space angles onto spherical coordinates. If the particle is moving the same amount in all directions, which is what we want to verify, we expect the φ dependence to be uniform and the θ dependence to be proportional to $\frac{1+\sin(\theta)}{2}$. Plots are smoothed via filtering through a density estimation routine. To obtain a qualitative measure if the obtained probability densities conform to our expected formula we feed them into a Kolmogorov-Smirnov[i] probability estimation routine, thus obtaining a numerical value for the quality of the random walk sampling.

4.7 Force Visualization

An AVS module was developed which produces 'hedgehog' type visualization of the potential field and its gradient. The field and its gradient are computed by inserting a

hypothetical probe particle on a 100x100x100 grid and varying its location to scan all 1 million points. Another way to compute the field, the gradient and its derivative is to get them directly from the CMM algorithm. In this case the octtree decomposition for the CMM implementation mandates using a grid size of a power of two. Either 64x64x64 or 128x128x128 (corresponding respectively to CMM Levels 6 and 7) are quite feasible for systems of a few tens to a few hundreds thousands of atoms.

4.8 Void visualization

A perl script that produces Open Inventor ™ 3D scene representing the voids is listed in Appendix F. It is possible to visualize either cluster *.cvoid* file or generalized volume file *.void* format file. When visualizing a cluster *.cvoid* file, the colors of the clusters corresponds to their index, e.g. different clusters are different colors. When visualizing a generalized volume *.void* file the color corresponds to the void density at this point, e.g. red means this point belongs to a void with high probability. Examples is shown in Figure 4-1.

4.9 Track file visualization

A perl scripts was developed that produce Open Inventor ™ 3D scene description file for either static scene or animation of the particle trajectories.

4.10 MCD simulation utility

The Monte Carlo Dynamics Utility performs random walk on the generalized free volume format of Section 4.1.1. The probability for the jump to a neighbor is determined

by the void values on the current position, its neighbors and the history (where the particle comes from.) The exact formula and comments are given in Chapter 6, Section 2.1.

The input parameters are shown in table 4-2.

Table 4-2 Control Parameters for the Random Walk Algorithm

STEPS	Integer	number of steps to run for
PENETRANTS	Integer	number of penetrants to simulate
SELF	Yes/No/ Double	are self transitions allowed and should we double the self-probability ? This is necessary for 1d-parts of 2d random walks.
MOMENTUM	cos/no/fast	how biased the random walk is.
DIRECTIONS	6 or 27	nearest neighbors only or include next nearest ones, too.
TRACK	Integer	tracking frequency.
PROJECT	Name	base name for output files.
ABSOLUTE	Yes/No	renormalize to global maximum
START	three numbers	user placement of the penetrants
PERCOLATE	Yes/No	detect percolation

The flowchart for the algorithm is shown on Figure 4.2. A practical introduction to random walks is given in Binder and Heermann[ii]. Our modifications include adding a bias, which mimics free particle motion, experimenting with different number of neighbors and considering jump rates that are function of the particle position. It is exactly in the jump rates, which are function of the grid location that the connection to the physical system being modeled is made. Another significant development is including

consideration of the rotational degrees of freedom, a method for which is proposed in Section 4.11.

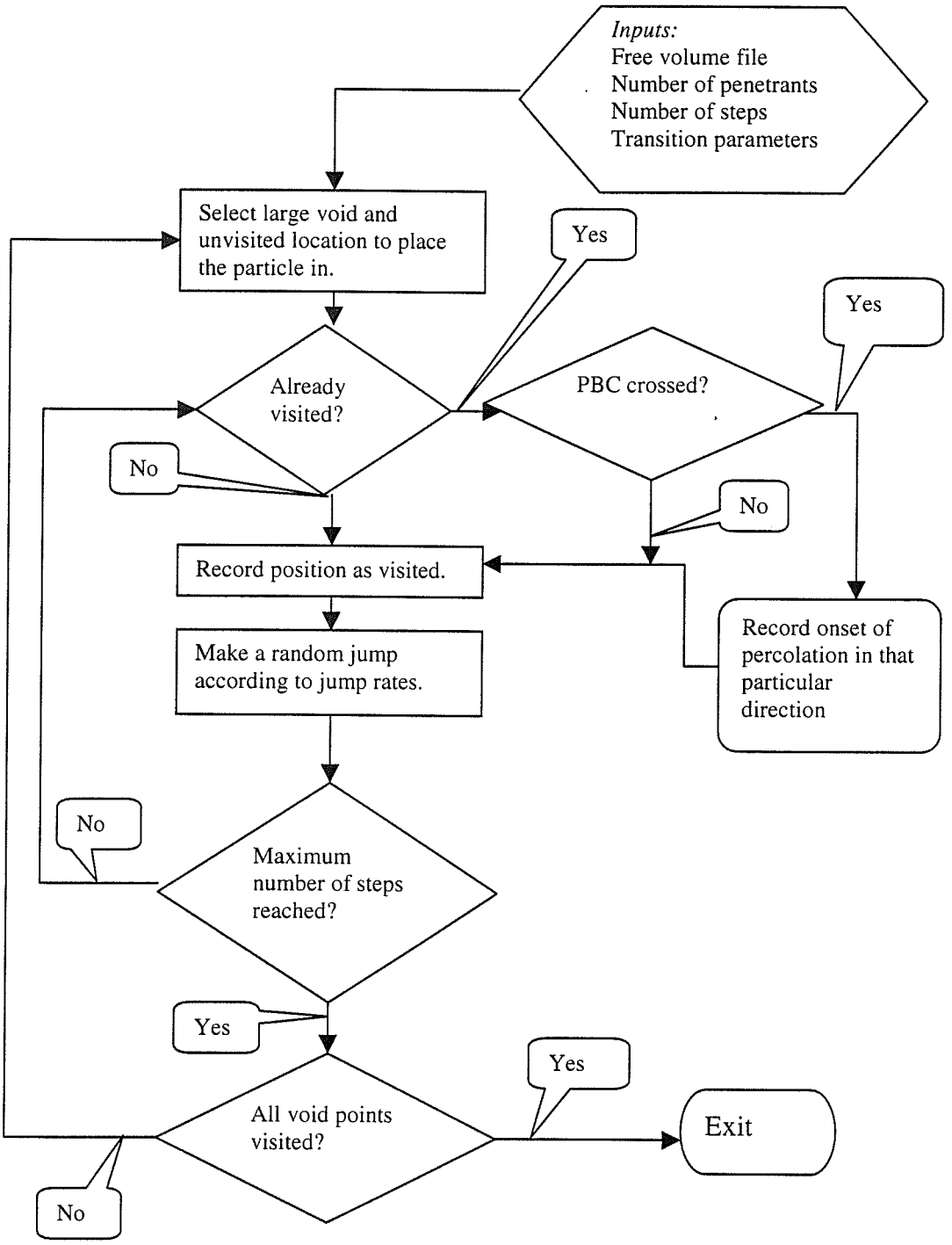


Figure 4.2 flow-chart for the Monte Carlo Dynamics algorithm

4.11 Extension of the random walk to electrically charged penetrants

4.11.1 Case of a point-like particle

We bias the random walk jump probabilities by taking into account the electrostatic force acting on the random walker. There are a few different ways we could do that. One way is by adding a term to the jump probability, proportional to $\alpha e - \frac{W}{kT}$. Here W is the particle energy in the electric field produced by the polymer. This is the traditional Boltzmann method. Another way is to bias all terms with respect to the angle that they subtend to the field gradient on the grid point $\beta \cos(\vec{r} - \vec{r}_0).\vec{E})$. In both cases we need to come up with a reasonable value for the parameters α and β. We chose the first way since that allows us also to treat diatomic molecules, as described in the next subsection.

4.11.2 Case of a diatomic molecule (one with rotational degrees of freedom)

MD studies have shown that the rotational degree of freedom is independent of the translational one[iii], with each getting half of the kinetic energy. The vibrational degrees of freedom are not activated at temperatures below 1000K. If the molecule has non-zero dipole (or quadrupole, etc. moment) it will interact with the field gradient, introducing additional force terms into its equations of motion. For example, if the electrostatic potential is $\Phi(\vec{r})$ and the electric field is

$$\vec{E}(\vec{r}) = -\nabla\Phi(\vec{r})$$

the first few terms in the energy expression for the molecule are[iv]:

$$W = q\Phi(0) - \vec{p}.\vec{E}(0) - \frac{1}{6}\sum_{i,j=1}^{3} Q_{ij}\frac{\partial E_j}{\partial x_i}(0)$$

where q is the charge of the molecule p is its dipole moment and Q is its quadrupole moment. To correct the translational motion we can then add the Boltzmann factor $\alpha e - \frac{W}{kT}$. Additionally the interaction of the molecule with the electric field exerts a torque on the molecule. In first order the torque is given by the expression $\vec{p} \times \vec{E}$. However, we can use the same expression for W as above and perform a Boltzmann type of jump, which we do on the joint set of coordinates, translational and rotational. We would need to introduce a grid over the rotational degree of freedom. We do so by introducing discrete steps on the unit sphere. We chose to work with the spherical coordinate theta and phi. Then the configurational space for the Monte Carlo motion has 6 x 4 = 24 possible steps, since we use possible jumps to 6 neighbors and any of them can be combined with 4 jumps = 2 x 2 jumps in the spherical coordinates φ and θ. It is also possible to introduce a geodesic grid on the sphere, which does not exhibit singularities around the poles. To be consistent we need to make sure that the step of the rotational

motion corresponds to the step of the translational motion. We calibrate the two steps to each other by monitoring the rotational and the translational kinetic energies and requiring that they be the same. We fix the parameter α by computing the kinetic energy and requiring that it correspond to the specified temperature.

---

[i] Press W.H., Teukolsky S.A., Vetterling W.T. and Flannery B.P., *Numerical Recipes in C*, Cambridge, University Press , 1992

[ii] Binder K., Heermann, D.W., *Monte Carlo Simulation in Statistical Physics, Springer Series in Solid-State Sciences,* **80**, Springer-Verlag, 1992

[iii] Gusev, A.A., Mullerplathe F., Vangunsteren W.F., Suter, U.W., *Advances in Polymer Science*, **116** 207-247(1994)

[iv] Jackson, J., *Classical Electrodynamics*, New York, Wiley, 1975

Chapter 5 Methods for studying diffusion in Polymers

Greenfield and Theodorou[i] give a review of the methods for prediction of the diffusivity of penetrants in various polymers. Another review is given by Gusev et.al[ii]

5.1. Phenomenological models

Phenomenological models employ coarse-graining to correlate diffusivity with physical parameters extracted from the molecular ones. These models typically are analytical and do not involve extensive computer simulations. They can provide some understanding and explanation of diffusion results, but do not allow for a priori prediction of transport properties as such prediction would require detailed molecular and atomic information for the structure of the polymer. A detailed description of the atoms and their interactions is incompatible with many simplifications usually necessary for analytical treatment.

5.1.1 Free volume methods

Free volume is the fraction of the volume not occupied by the polymer. This concept has found uses in discussing physical properties such as viscosity, diffusion in liquids, viscoelasticity, electrical conductivity, the glass transition and plastic yielding. Early attempts to predict diffusion consist in trying to relate the diffusion coefficient (D) of a hypothetical penetrant to the fraction of the volume that is free

volume (voidage fraction) and possibly some other information obtained from the detailed structure, in addition to the pure quantity, of the free volume. The free volume theory implies that the voidage fraction is the dominant content as D is proportional to both the kinetic velocity and the probability of finding enough free volume for the hopping of the penetrant. (Cohen and Turnbull[iii].) An interesting early result from Wakao and Smith[iv] for diffusion in porous media is that in a random distribution of pores the diffusion coefficient is proportional to the square of the voidage.

More recent models have tried to take into account the distribution of the free volume and even its dynamical redistribution as the polymer evolves performing thermal motion with time. Takeuchi and Okazaki[v] have suggested a simple integral expression of the cluster size distribution that shows very good correlation with the self-diffusion coefficient of penetrant particles of He and O_2 for their hypothetical polymer models. However, the lack of further interest in that expression by the same or other authors, probably indicates that their model was oversimplified or unrealistic in some other way.

Free volume distributions have been studied theoretically by using kinetic theories and molecular dynamics. Experimental probes for free volume at molecular and atomic scales are possible using small-angle diffraction, fluorescent spectroscopy and positron annihilation lifetime spectroscopy, which has emerged as the unique method providing resolution of a few Angstroms in size directly. Chow[vi] has modeled the size and volume distribution of holes in amorphous polymers and has provided a

good description of the non-Gaussian spreading observed very recently in position annihilation experiments.

5.1.2 Series of Activated jumps

For those kind of models the diffusive process is represented as a sequence of activated hopping motions; a specific molecular mechanism is assumed and the resulting transition dynamics is calculated. This involves some way of determining the voids as well as the transition rates for transitions between them. The transition state theory (TST) mentioned below is one way to do that. Another possible way is by computing residence times and transition rates from our random walk algorithm described in Section 6. An algorithm for extracting residence times and transition rates is given in Appendix G.

5.2. First principle methods

5.2.1 Potential energy along a diffusion path

As early as 1973 Jagodic et al.[vii] attempted a computer calculation of the diffusion coefficient from the potential energy along a diffusion path. Their progress was, however, seriously impaired by the lack of methods at that time to generate realistic polymer structure and to compute realistic potential energy barriers.

5.2.2 Molecular dynamics methods

With the increase in computing power available in recent years, numerous molecular dynamics studies have been performed for both polymer melts (above the polymer glass transition temperature) and for glassy polymers. It is a lot easier to predict the diffusion coefficients for the melts, since the mobility of the polymer and hence of the penetrant is high enough, so that penetrant motion can be observed within the limits for computational simulation, approximately of the order of 10 ns.

Various attempts have been made to speed up MD computations. Gusev et.al.[viii] employed transition-state theory (TST) to study the dynamics of light gases dissolved in rigid microstructures of glassy polycarbonate and rubbery polyisobutylene modeled in atomistic detail. They concluded that the rigid-matrix approach is inadequate for studying the dynamics of light gases in dense polymers, except He. The rigidity of the polymer was a drawback of a study by Gusev et.al[ix], which considered model polymers with atoms bound harmonically to their equilibrium positions, about which they execute independent vibrations. Again prediction was in the correct order of magnitude for melt, but orders of magnitude too small for glassy structures.

In the early attempts to utilize MD for predicting diffusivity considerable popularity was enjoyed by the united force field approaches, which collapse a hydrogen onto the carbon atoms, thus eliminating the small mass of the hydrogen from the dynamic equations and allowing for much larger timestep in the simulations. This however leads to orders of magnitude overestimation of the diffusion

coefficients, which can be explained as follows. The free volume in the polymer consists of large cavities and narrower channels which connect them. Using united atom forcefields does not change significantly the voids, however it makes the channels much wider, since there are no hydrogen atoms sticking out of the polymer chains. Thus the penetrant motion through the channels proceeds largely unobstructed.

5.3. Free volume computation

Various methods have been used for computation of the free volume of a particular structure. We have used the following method similar to the one utilized by Rigby and Roe[x]. The unit cell is divided in 100x100x100 small cubes (approximate resolution 0.3 A) Then we scan all polymer atoms and mark as occupied all cubes, which are within a Van der Waals radius of the polymer atom. After all polymer segments are exhausted what remains is the free volume. To take into account the finite size of the penetrant molecule the Van der Waals radius is modified to reflect the interaction between the penetrant and the polymer atom. The exact modification depends on whether soft or hard core interaction is chosen. Another method for computing free volume is Delaunay tessellation, which partitions points in space into irregular tetrahedra such that the circumsphere of each tetrahedron is devoid of points except those lying on the vertices. An algorithm for computing the tetrahedra is described by Tanemure et.al.[xi]

5.4 Other approaches

Takeuchi and Okazaki[xii] have considered computing other correlation functions. First they compute the Van Hove self-correlation function. They see a shoulder or bump at about 3 $\overset{o}{A}$, which they infer to be the order of the cage size. They also compute the density autocorrelation function and deduce the average instance between two adjacent cages to be 6 $\overset{o}{A}$. They conclude that anomalous diffusion is seen at spatial scale between those two spatial characteristics 3A and 6 $\overset{o}{A}$. These are very interesting approaches and in our opinion they are worth applying on polymers modeled with realistic forcefield.

---

[i] Greenfield M.L. & Theodorou,D.N. *Macromolecules*, **26**,5461-5472(1993)

[ii] Gusev A.A., Muller-Plathe F., van Gunsteren W.F and Suter U.W., *Advances in Polymer Science*, **116**, 207(1994)

[iii] Cohen M.H. and Trunbull D., *J.Chem.Phys.*, **31**, 1164(1959)

[iv] Wakao N. and Smith J.M. , *Chem.Engng. Sci.*, **17**, 825-834(1962)

[v] Takeuchi H. and Okazaki K., *Makromol.Chem., Macromol.Symp.*, **65**, 81-88(1993)

[vi] Chow T.S., *Macrom. Theory Simul.*, **4**, 397-404 (1995)

[vii] Jagodic F., Borstnik B. and Azman A., *Makroml.Chem.*, **173**, 221-231(1973)

[viii] Gusev A.A., Arizzi S., Suter U.W, *J.Chem.Phys.*, **99**(3), 2221(1993)

[ix] Gusev A. A. and Suter U.W., *J.Chem.Phys.*,**99**,2228-2234(1993)

[x] Rigby D. and Roe R.J., *Macromolecules,* **23**, 5312 (1990)

[xi] Tanemura M., Ogawa T. and Ogita N., *J. Comput. Phys.*, **51**, 191(1983)

[xii] Takeuchi-H, Okazaki-K, *Molecular Simulations*, **16** (1996)

Chapter 6 Diffusion of Gases in Amorphous Polymers: The Monte Carlo Void Method

Mihail Iotov, Siddharth Dasgupta,* and William A. Goddard III*

Materials and Process Simulation Center, Beckman Institute (139-74)
Division of Chemistry and Chemical Engineering
California Institute of Technology, Pasadena, California 91125

Abstract

We propose a method for studying diffusion in amorphous structures based on biased random walk in the free volume extracted from a polymer (*"the Monte Carlo Void Method"*). We analyze a number of simple free volume structures not derived from realistic polymers and show that the biased random walk method is offering intuitively realistic description of the particle motion and present a framework for computing diffusion coefficients.

1.0 Introduction

The diffusion of gases in polymers is of major importance to the polymer industry[i]. For example, various applications in the food industry involve impeding the diffusion of gases (such as H_2, O_2, CO_2, H_2O, and CH_4) through thin polymer films[ii]. The design of new copolymers, new monomers, or new blends to selectively impede the diffusion of some gases while allowing others (O_2 versus N_2, CO_2 versus O_2, H_2O versus O_2) underlies a number of potential applications. The design and optimization of the selectivity of polymer membranes for diffusion would be greatly facilitated if reliable predictions of diffusivity could be made rapidly in advance of

synthesis and experiment. The time scale for diffusion (μsec to msec) is far too long for routine applications of molecular dynamics (MD). This has made it difficult to obtain reliable diffusion data from theory[iii]. In addition, it is not easy to obtain reproducible data on gas diffusion in polymers from experiment. As a result there is little in the way of reliable predictions on how to design polymer films to achieve specific diffusion properties.

The permeability (P) of a gas through a membrane can be written as[iv]

$$P = D\,S \tag{1}$$

where D is the diffusion coefficient and S is the solubility. Generally, only P is available experimentally, but D and S depend differently on the various design parameters. Thus, it is useful to be able to independently measure each. In this paper we will focus on D.

In the long time limit for a three-dimensional system, Einstein showed that the total distance $\boldsymbol{R}$ traveled in a time (t) is given by Equation (2) (the Einstein relation)

$$\langle R^2 \rangle = 6\,D\,t \quad \text{as } t \rightarrow \infty \tag{2}$$

where < > designates averaging over the ensemble of starting and ending points for the given time interval t. To use theory to predict D for a gas X in a polymer, it is necessary to carry out a number of MD calculations (each starting with X in various sites) for times sufficiently long that (2) is obeyed.

The problem with using MD to study gaseous diffusion is the time scale. It is

necessary for the calculation to proceed for a time t sufficiently long that the Einstein relation (2) is valid. Even for simple gases in polymers, the relevant times are often on the order of microseconds (μs) or longer. Since the MD time step is ~1 femtosecond (fs), a total time of μsec requires ~1 billion steps. For a polymer of realistic molecular weight, this requires very long times on even the largest supercomputer. Consequently, it is usually not practical to use MD to obtain reliable values for D.

It is desirable, therefore, to develop another framework for simulating the motion of the penetrant particle, one that will be faster and will concentrate on the diffusion relevant motion only, using a coarse-grain description of the surrounding system. For well-ordered media (e.g., crystals) the motion of the diffusing penetrant is usually modeled in terms of jumps between vacancies of the crystal structure. No ordered structure exists for amorphous system, but it is still possible to study the free volume in such a system – the locus of all points accessible to a hypothetical penetrant of a given radius. This is usually a complex irregular geometric shape with rich structure in which features like cavities, channels, dead-ends, etc., can be discerned.

Thus, we consider a Monte Carlo (MC) process in which a pseudo particle moves unrestricted, or partially restricted, within the free volume and is prohibited from leaving it (i.e., it encounters an infinite potential outside of it). In this paper, we will study model systems that are not based on realistic polymers, but are better suited for simulation and analysis.

Section 2.0 defines the generalized free volume concept. Section 2.1 presents

the Monte Carlo void diffusion method (MCVD). Sections 3.1-3.4 apply the method to study diffusion in simple model systems. Section 3.5 makes a connection to MD. Section 3.6 attempts to explain the origin and the behavior of the anomalous diffusion. Section 4.0 discusses how those results might be applied for studying realistic polymers.

2.0 Generalized free volume concept

The free volume is defined as the collection of points accessible to a penetrant of a given radius. Thus it can be viewed as a scalar field defined on the unit cell and which has value 1 when the point is considered to belong to the free volume and value 0 otherwise.

We can perform operations like addition or subtraction on such scalar fields and multiplication by a number.

When we take consecutive snapshots of the simulated structure and add together the free volume scalar fields for each of them and divide by their total number, we obtain the *generalized free volume*, which is again a scalar field valued between 0 and 1, but now it can accept also values other than 0 and 1. This average free volume contains not only static but also dynamic information about the simulated structure. It can be argued that the value of the free volume is related to the potential energy of the penetrant in that the higher the free volume value, the lower the potential energy is. At points where the free volume field is 0, the penetrant experiences infinite potential and is prohibited from going there.

2.1 The Monte Carlo Void Diffusion (MCVD) Model

In our approach we compute the free volume fields on a 100x100x100 rectangular grid. It appears, then, that this free volume field can be viewed as a probability grid for performing a biased random walk. Given that void probability grid we will consider a random walk (in which a particle on a grid point moves to an adjacent grid point. We assume the probability to be proportional to the void weight (1 if always a void to 0 if never a void). Figure 15 shows a MD trajectory within a void (at 1fs resolution). In order for a random walk process to mimic this, we include a bias in the jump probability with an angular dependence based on the last previous step direction. We find that a simple cosine term (of the angle between the last step and the next jump directions) leads to a smooth linear trajectory within a cavity with the particle changing direction mostly only on reaching the edge of the void. This mimics the MD trajectory as indicated in Figure 15. This bias helps mimic the initial ballistic region of MD where r^2 is proportional to t^2. Non-biased random walk yields r^2 proportional to t. Figure 16 demonstrates how the r^2 MSD curves differ for two simulation runs, one of which is ran without momentum and the other with a fast momentum term. The initial ballistic region is apparent in the latter case and missing for the non-momentum case. Fast momentum term means the bias shown on Figure 14d. The normal momentum term curve will fall somewhere between the two curves.

A second issue concerns which neighboring points are allowed for the jump. Simplest choices might be

a) only the 6 nearest neighbor points (forward, backward, up, down, right, left),

b) the 18 points also including next nearest neighbors, or

c) the 26 points also including third nearest neighbors.

We chose case c since it efficiently spans the solid angle choices, allowing channels in diagonal direction to be found. Thus, starting at point $G(l_x,l_y,l_z)$ with the previous point at $P(l_x^-, l_y^-, l_z^-)$ (where $|l_i - l_i^-| = 0,1$ but at least one is nonzero), then we calculate the probability for

$$G(l_x^+,l_y^+,l_z^+) = W(l_x^+,l_y^+,l_z^+)\left[q\cos(\underset{\sim}{l}^+ - \underset{\sim}{l}) \cdot (\underset{\sim}{l}^o - \underset{\sim}{l}^-) + (\underset{\sim}{1} - q)\right]$$

where q can be taken from 0 to 1 (we used $q = ½$).

The larger the value or q, the more strongly the next movement is biased to be in the direction of the last movement - thus more close resembling dynamics motion at short distances. For q=0, we have an unbiased random walk, and for q=1, we have 0 probability of going backward on any step, so all back turns will be by necessity smooth curvilinear trajectories. Thus, q can be thought of as a smoothing factor. Figure 14 a,b,c,d, shows the bias in a polar diagram where the radius is proportional to the probability to select that direction, values of q=0,0.5,1 and also a case in which the whole term in the braces is squared to assure even more forward-biased walk.

3. Model problems

Unless otherwise specified, all of our simulation runs are done with the following settings. Unit cell size 100 A. The free volume structures are simple, e.g., only 0 and 1 values are allowed, either void or not. When visualizing them, the void

part is shown, i.e. this is the part that is accessible to the particles. Grid size was 100x100x100. Run time was 1,000,000 steps. Sampling positions were done every 1000 steps. For Sections 3.1 to 3.5, transitions were made in 26 neighboring directions. Five particles were placed randomly in the voids and their motion was followed, then mean-squared deviation was computed for each of them and an average of the five was taken. For all the later sections, no momentum was used and a single particle was followed. Not using momentum bias allows for easier theoretical analysis.

3.1 Finite sphere

Figure1 shows a simulation run of a finite sphere. This simulations involves 10,000 steps and is sampled every timestep. As expected after the ballistic region, the mean-squared deviation saturates to a value of the order of the diameter of the sphere. It can be shown analytically that the square root of the saturated mean-square deviation is equal to twice the radius of gyration of the sphere or sqrt(6/5) ~= 1.095 of the radius of the sphere, for constrained Brownian motion without a momentum term. The value from numerical simulations is actually about 1.7% lower, which is probably a finite size effect, since the analytical value of 1.095 is computed for sphere of infinite radius. In the actual simulation on Figure 1, a momentum term is present which leads to an even higher value, since more rapid movement away from previous positions is favored.

3.2 Sphere with periodic connections

Figures 2,3 and 4 show simulations of the same sphere, but when connected to its own translated image across the periodic boundary. The difference between them is that in Figure 2 there is a connection in only 1 dimension, in Figure 3 in two dimensions and in Figure 4 in all three dimensions. The diffusion constant increases respectively from 110 to 225 and 320, as might be expected for diffusion through independent channels, e.g., the diffusion in three dimensions is three times as fast as the one in one dimension (those diffusion constants are in A^2/iteration units).

3.3 Fine-grained simulation

Figure 5 shows a simulation run for the same system as in Figure 4, but sampled every timestep. This more fine grained sampling shows the three distinct regions of diffusion: the initial ballistic region, the "anomalous" region in which the mean-squared deviation is not proportional to the time, but to a fractional power of it (from around 25 steps to around 300 steps) and the Fickian region, when linear proportionality is achieved.

3.4 Cylinders

Figures 6, 7, 8, 9 show the results from simulation runs for longitudinally and transversely connected cylinders in one and two dimensions. Just as expected the diffusion coefficient is highest in the longitudinally connected cylinders and smaller

in the one dimensionally transversely connected case, in which case the cylinder acts as a capacitor which slows down the moving particle.

3.5 Connection to Molecular Dynamics

It is interesting to look at a plot of the displacement of a particle from its initial position as a function of the simulation time. Figure 13a shows such a plot for a system similar to the one in Figure 2 but with channel width of two. The typical behavior of the penetrant is spending quite some time in one of the spheres and then rather quickly moving through a channel to another sphere. Figure 13b shows a similar plot obtained from a molecular dynamics simulation of a He particle in PE at 450k. We find it very interesting that the two plots in Figures13 a and b look so similar. It can be concluded then that the underlying motion of the He penetrant in PE consists of rambling in a cavity for a while and then quickly moving to another one. It looks quite plausible that that motion can be modeled through a random walk as described above on an appropriately defined void structure extracted from the structure of the polymer.

3.6 Origin and behavior of the anomalous diffusion region

In recent years, a lot of attention has been directed toward study of the anomalous diffusion, in which the mean r^2 is proportional to a fractional power of t, like $t^{0.5}$. This has been demonstrated in long MD simulations[v] [vi] and also

experimentally observed in single-molecule microscopy on model membranes[vii].

Some of the possible explanations have attributed the anomalous diffusion to single file diffusion for which evidence have been observed in zeolites[viii].

We will demonstrate, however, that the anomalous diffusion is due to the restricted geometry of the particle motion.

3.6.1. Spheres connected via cylindrical channels.

To see how the anomalous diffusion region arises, we studied the system on Figure 2, and we varied the width of the channel. Thus we had on one end spheres connected with very thin channels, and at the other end the channel was encompassing the spheres totally and the system looked like an infinite cylinder of radius 20. The results are summarized in the following table:

Table 1. Results from simulation of spheres connected via cylindrical channels.

Channel width (in A)	Einstein diffusion (A^2/step)	Ballistic rate (A^2/step)	Anomalous intersect (steps)	anomalous exponent	Crossover time (steps)	Crossover distance [A^2]
2	0.70	35	250	.15	10	450
4	2.2	35	160	.30	7	750
8	7	34	70	.58	5	2500
12	12	35	62	.65	5	1000
20	13	36	70	.7	10	4000

The first crossover term is for transition from ballistic to anomalous diffusion, and the second one is for transition from anomalous to linear Einstein diffusion. We see that the crossover to Einstein diffusion happens well before 10000 A^2, which is the square size of the unit cell. This would imply that the use of unit cell does not interfere with the diffusion rate computation.

Figure 10a shows the diffusion coefficient as a function of the channel radius. Figure 10b shows the anomalous diffusion exponent as a function of the channel radius.

3.6.2 Infinite cylinders

To make sure that the anomalous diffusion observed is not an artifact of the underlying periodicity of the structure imposed by using a repeated unit cell, we also studied diffusion in cylinders. Since they are similar to each other, it does not matter in what size unit cell they are embedded. The following table shows the results for infinite cylinders of different radii.

Table 2. Results from simulation of infinite cylinders.

Width of Channel (in A)	Diffusion Rate (A^2/step)	Ballistic rate (A^2/step)	Anomalous Rate (A^2/step)	Anomalous exponent	Crossover time (steps)	Crossover distance (A)
2	15	-	20	.7	-	30
4	13	-	25	.7	-	150
8	12	33	40	.7	2	50
12	13	35	50	.7	4	100
20	13	36	70	.7	10	300
30	14	36	80	.7	15	300

As expected the anomalous exponent is independent of the width of the cylinder. Apparently it is a property inherent to the cylindrical shape. The crossover to the Einstein diffusion increases with the size of the cylinder, since the particle needs time to reach one edge from the other. For the first two cases the cylinder is so narrow that no ballistic region is observed.

3.6.3 Infinite strip

To further explore the hypothesis that anomalous diffusion is due to obstructions, we consider the simplest possible system: 80-unit wide infinite planar strip. This can be viewed as the product of an infinite line and a finite 80-unit long linear stretch. The MSD plots for those three structures are shown in Figures 11 a, b, and c. As expected the anomalous region is well present in the case of the infinite planar strip. It is spatially located between 750 and 10000 A^2, which correlates favorably with the size of the stretch in the finite, dimension 80, the square of which falls in the middle of that region. On Figure 11a, the saturation point for the finite stretch is twice the radius of gyration of the finite stretch, which is the square of its length divided by six. Figure 12 shows the same behavior for a 20x20 rectangular tube.

In an attempt to analyze the above case, we try to model the saturation curve of Figure 11a, with a simple ramp-like function. Since $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle$ we need to add the saturation curve and a straight line to see how the infinite linear strip MSD

function would look. The results are shown in Figure 17. While the transition curve in the anomalous region is not really linear, its character is obviously due to the saturation of the y^2 MSD function. It is our believe that more realistic modeling of the saturation curve with a concave curve more like the one in Figure 11a, will lead to a linear function in the anomalous region.

4.0 Discussion and Conclusions

We conclude that the described biased random walk method can be used to coarse-grain molecular dynamics and simulate motion of penetrants in amorphous polymers. We have shown also that the anomalous region appears for even very simple systems and appears to be due to the restrictions that the void matrix imposes on the motion in some directions. In general, the more restrictions there are, the lower the anomalous exponent that we observe. In the case of free space there is no anomalous exponent (e.g., it reaches one). It is very interesting also to observe that anomalous region is observed also in a system practically identical to a Zeolite, namely a tube, where the restriction is perpendicular to the axial direction. For an arbitrary system, it will be extremely interesting to see how the fraction by which the diffusion coefficient is reduced and the anomalous exponent depend on the void structure. It might be possible to find a numeric relationship expressing the above two quantities in terms of geometric quantities pertaining to the void matrix, for example, porosity, tortuosity or fractal dimension.

[i] Schlotter N. E. and Furlam P. Y., *Polymer*, **33**, 3323(1992); Wessling M., Boomgaard T. v. d., Mulder M. H. V. and Smolders C. A. *Makromol. Chem., Macromol. Symp.*, **70/71**, 379(1993)

[ii] Koros W. J., ed. *Barrier Polymers and Structures, ACS Symp. Ser.*, 423(1990); Mohr J. M., and Paul D. R., *J. Appl. Polym. Sci.*, **42**, 1711(1991)

[iii]Gusev A.A., Arizzi S., Suter U.W., *J.Chem.Phys.*, **99**(3), 2221(1993);

Gusev A. A. and Suter U.W., *J.Chem.Phys.*,**99**,2228-2234(1993)

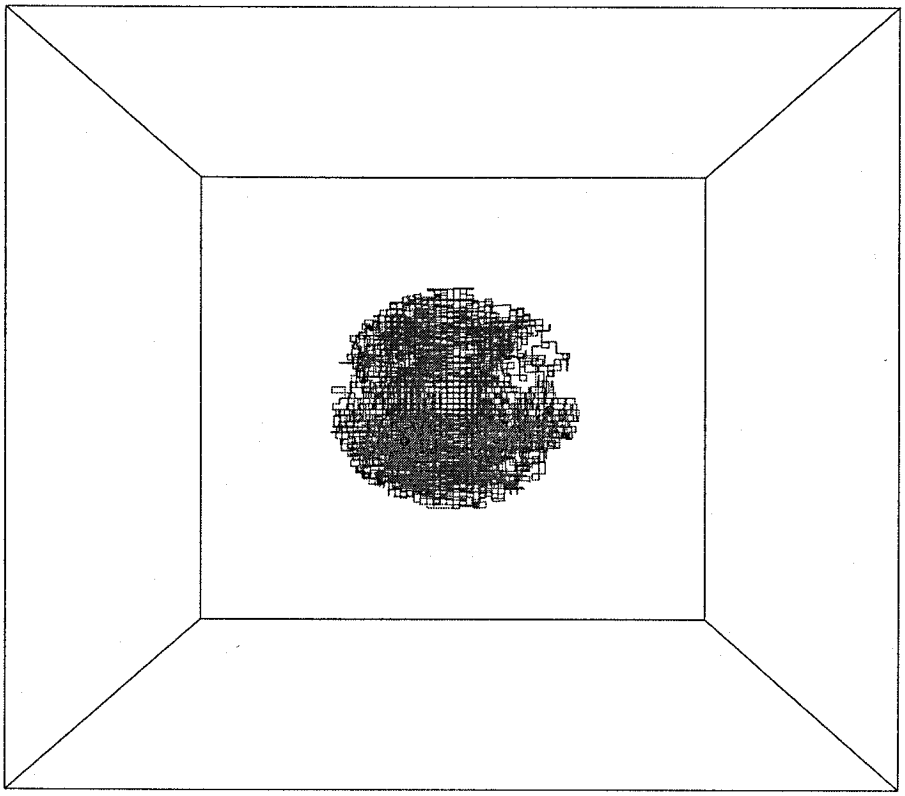
[iv] Crank J., *The Mathematics of Diffusion*, Oxford, Clarendon, 1989

[v] Iotov M., Ph.D. Thesis, Caltech, 1998

[vi] Takeuchi H. and Okazaki K., *Makromol.Chem.,Macromol.Symp.*,**65**, 81-88(1993)

[vii] Schutz G.J., Schindler H. and Schmidt T., *Biophysical Journal*, **73**(2),1073-1080(1997)

[viii] Gupta V., Nivarthi S.S., Keffer D., Mccormick A.V. and Davis H.T., *Science*, **274**, 164-164(1996)



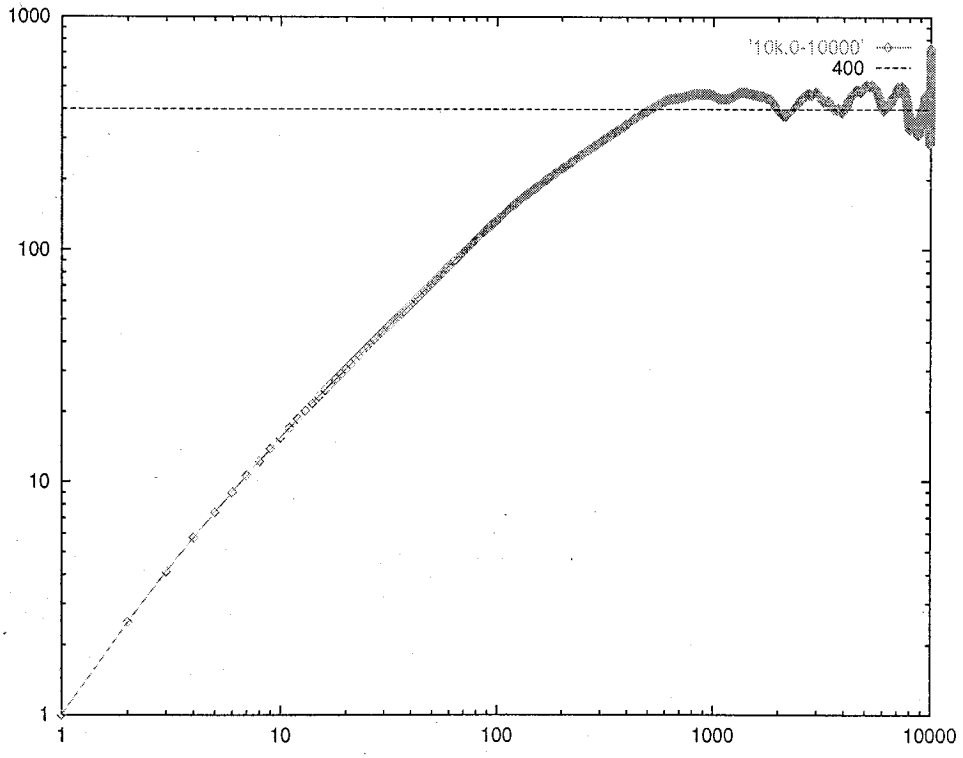
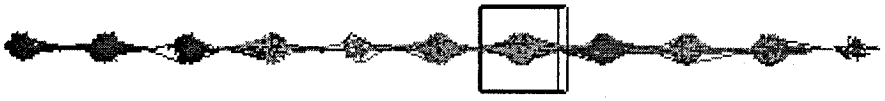


Figure 1. Finite sphere.



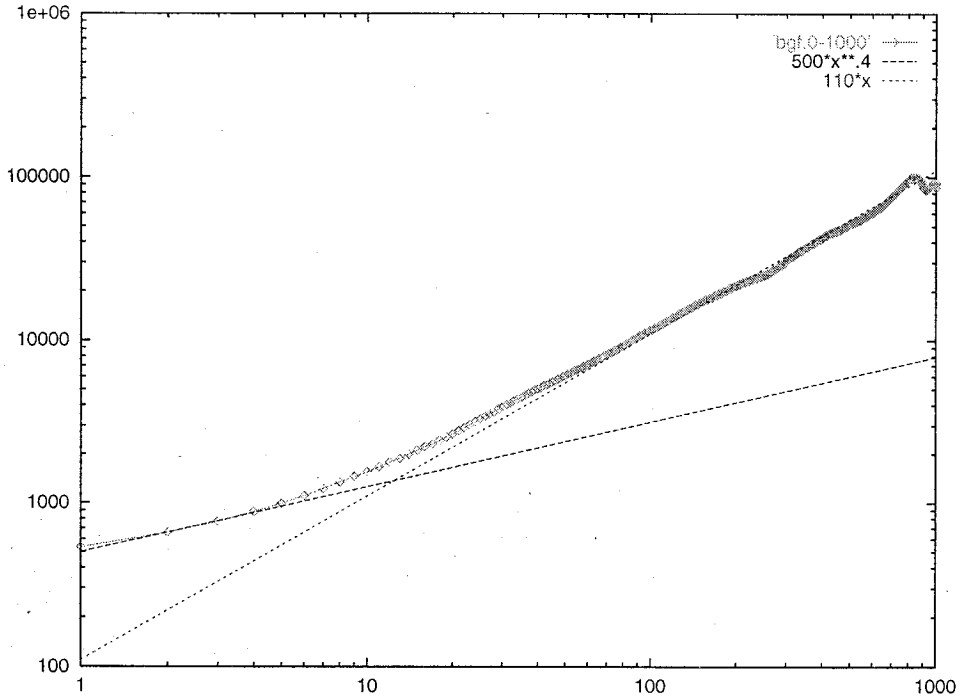
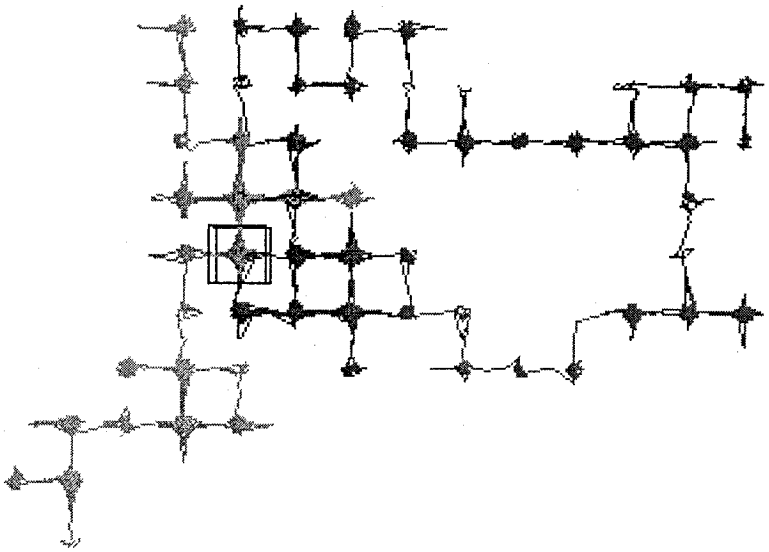


Figure 2. 1D PBC connected sphere.



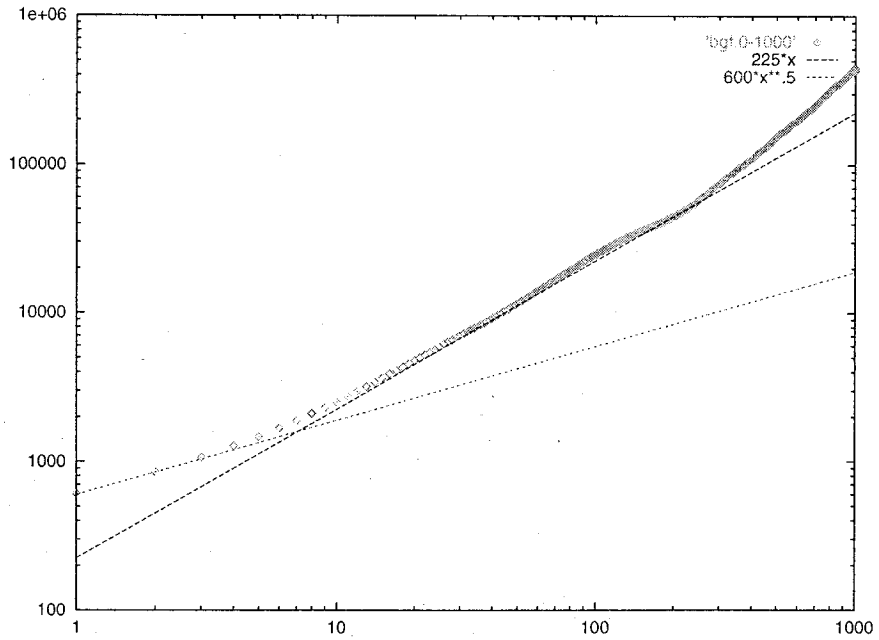
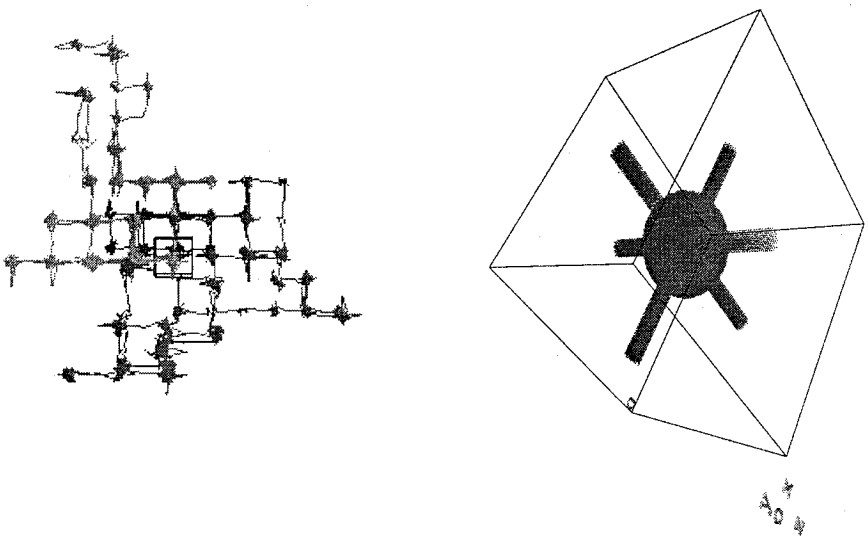


Figure 3. 2D PBC connected sphere.



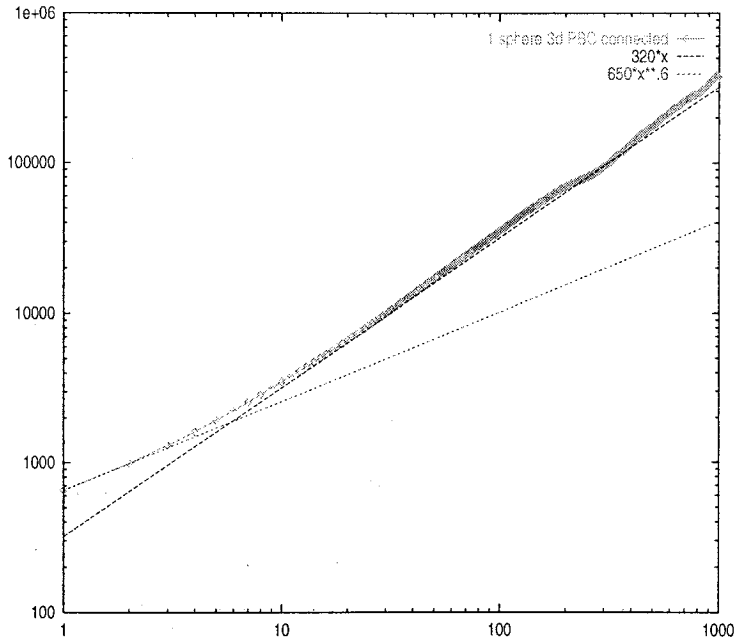


Figure 4. 3D PBC connected sphere.

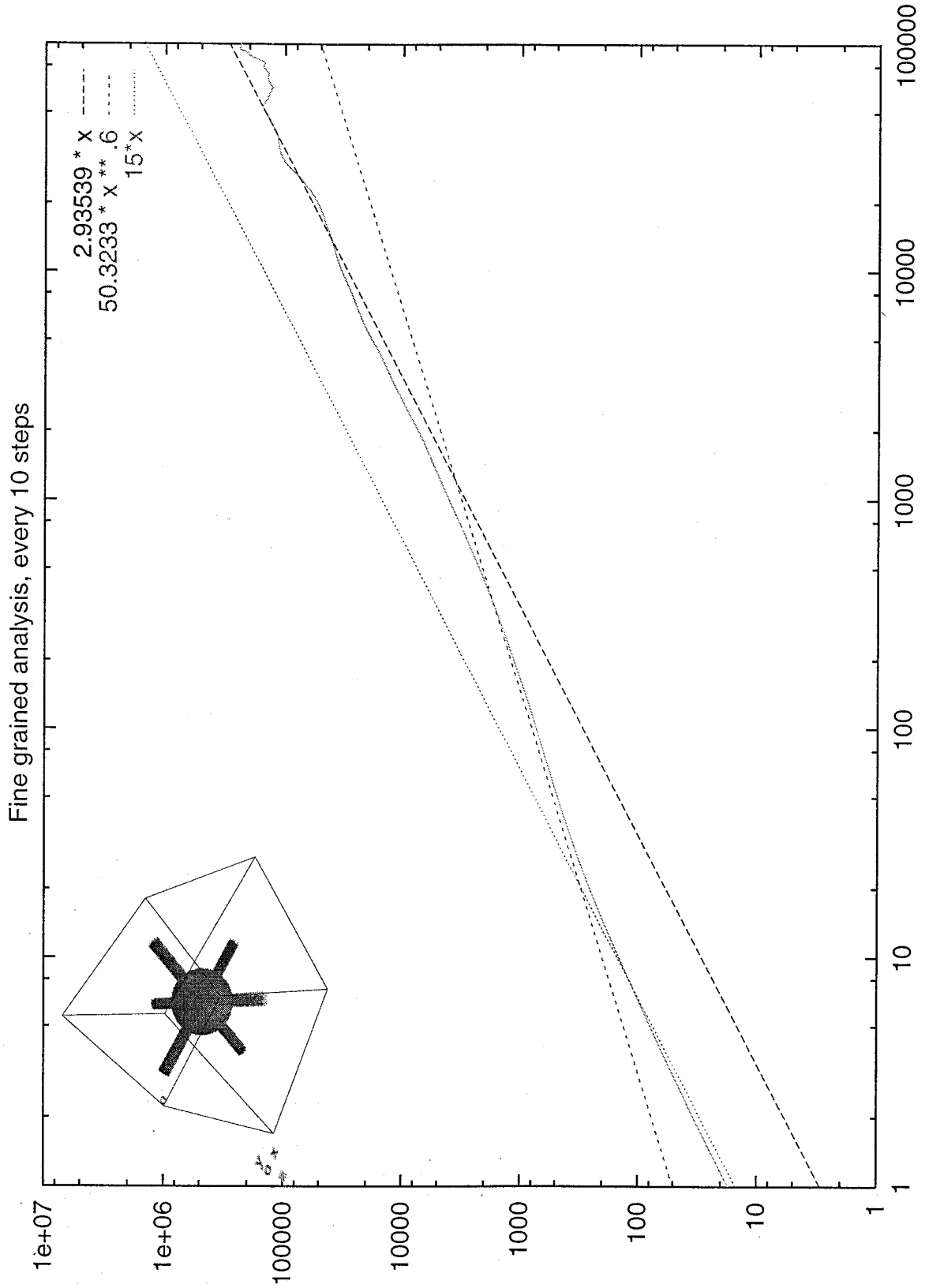
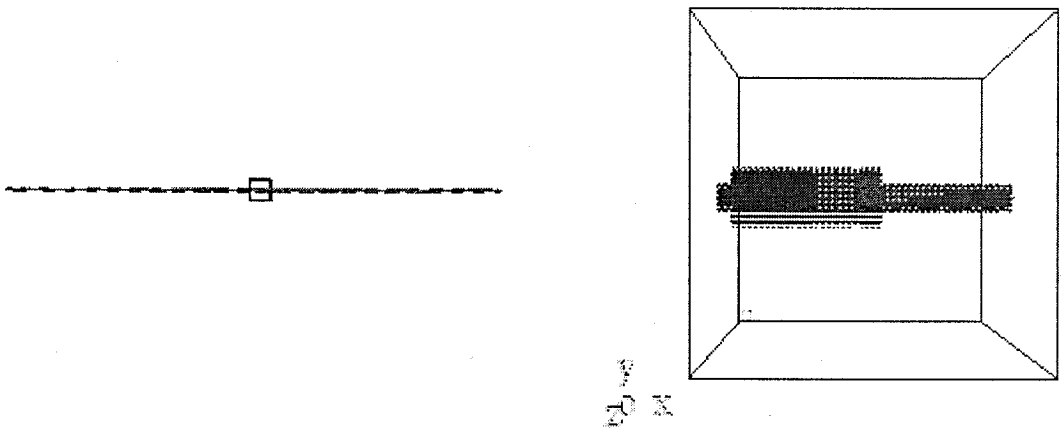


Figure 5. 3D PBC sphere - fine grain run.



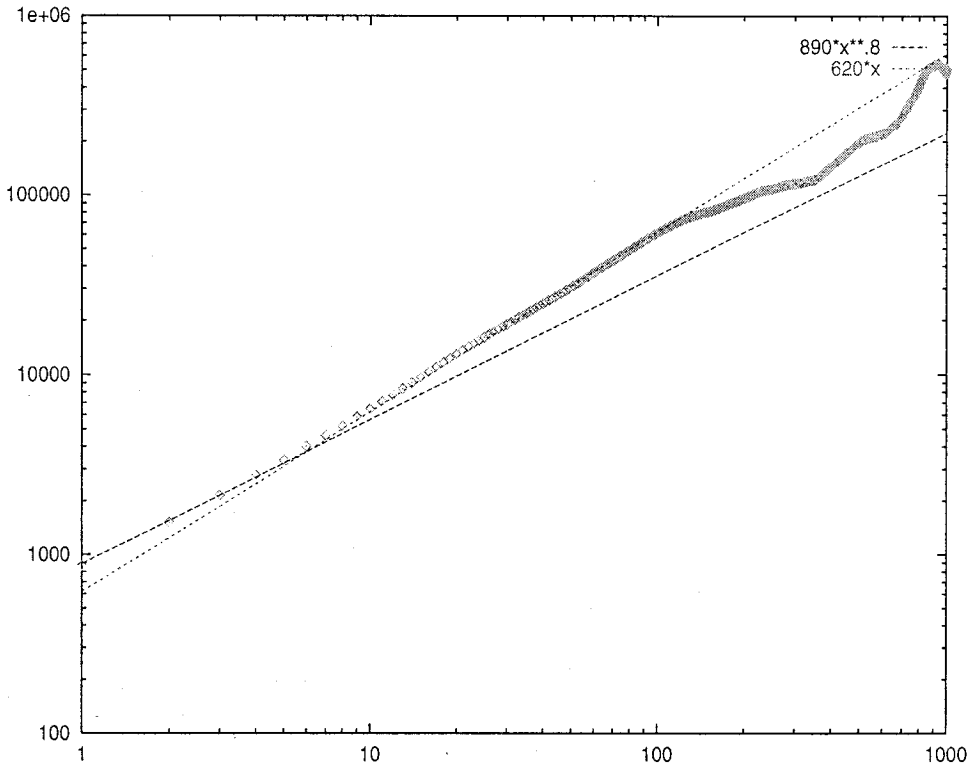
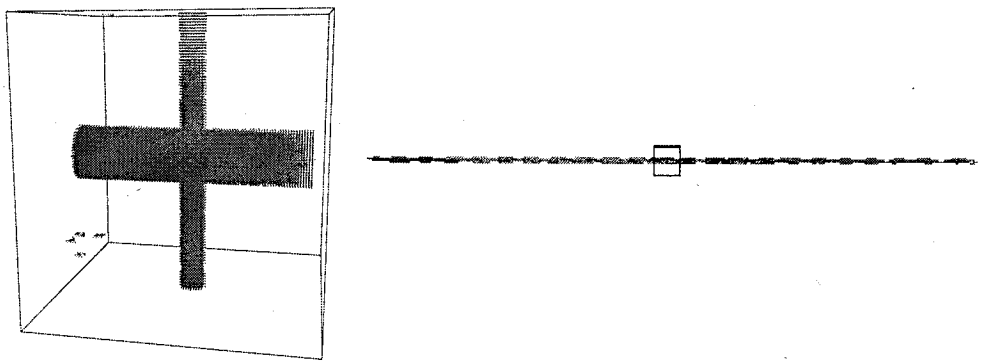


Figure 6. 1D PBC longitudinally connected cylinder.



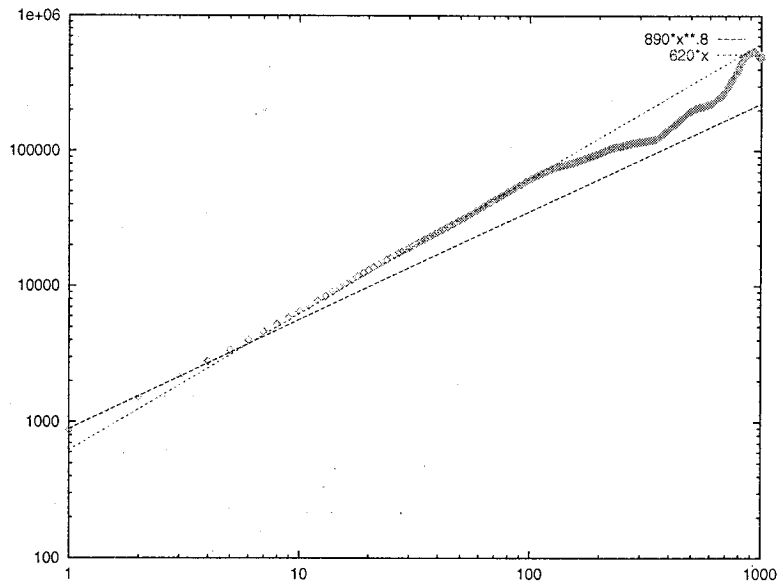


Figure 7 . 1D PBC longitudinally connected cylinder.

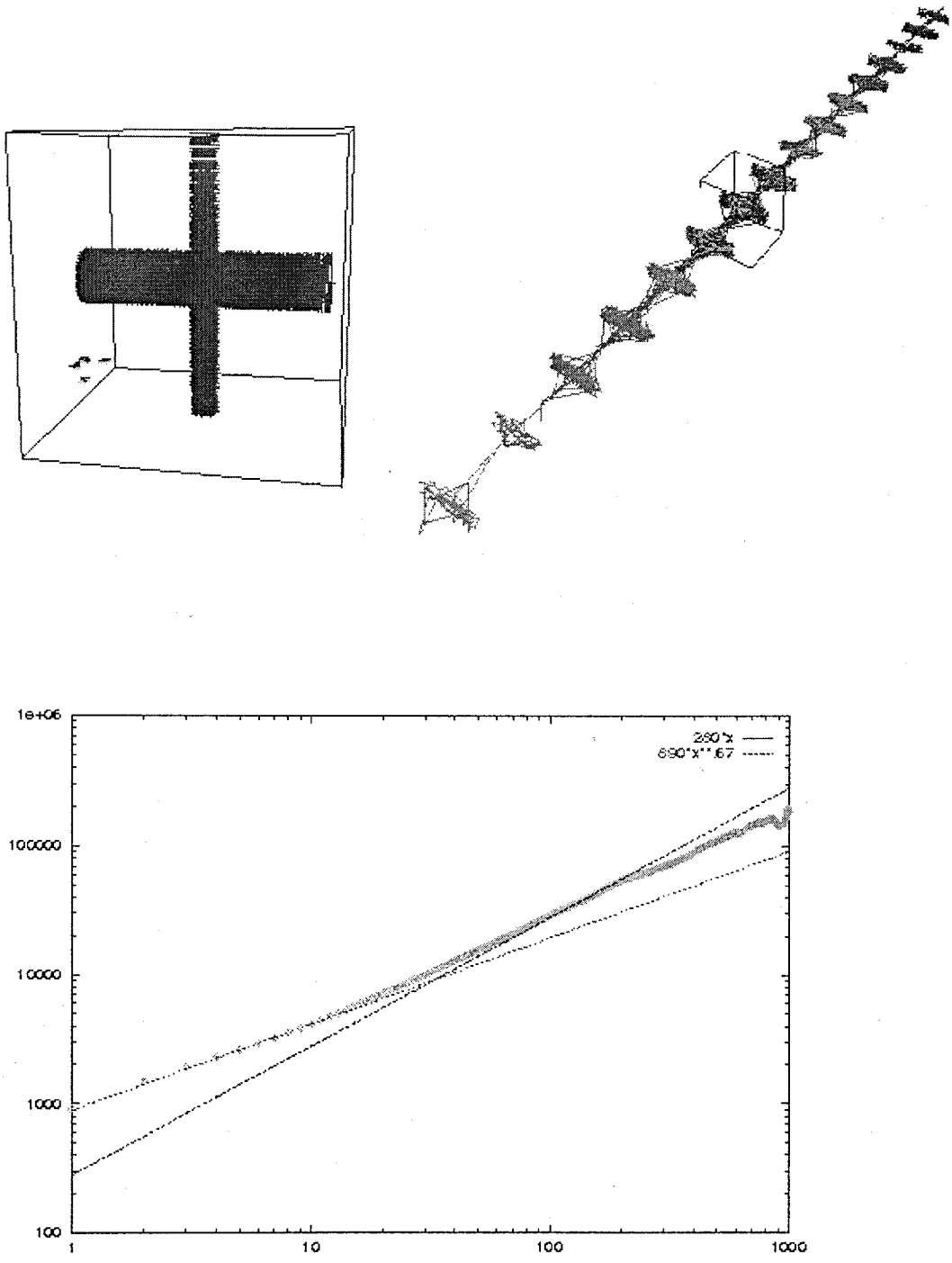


Figure 8. 1D PBC transversely connected cylinder.

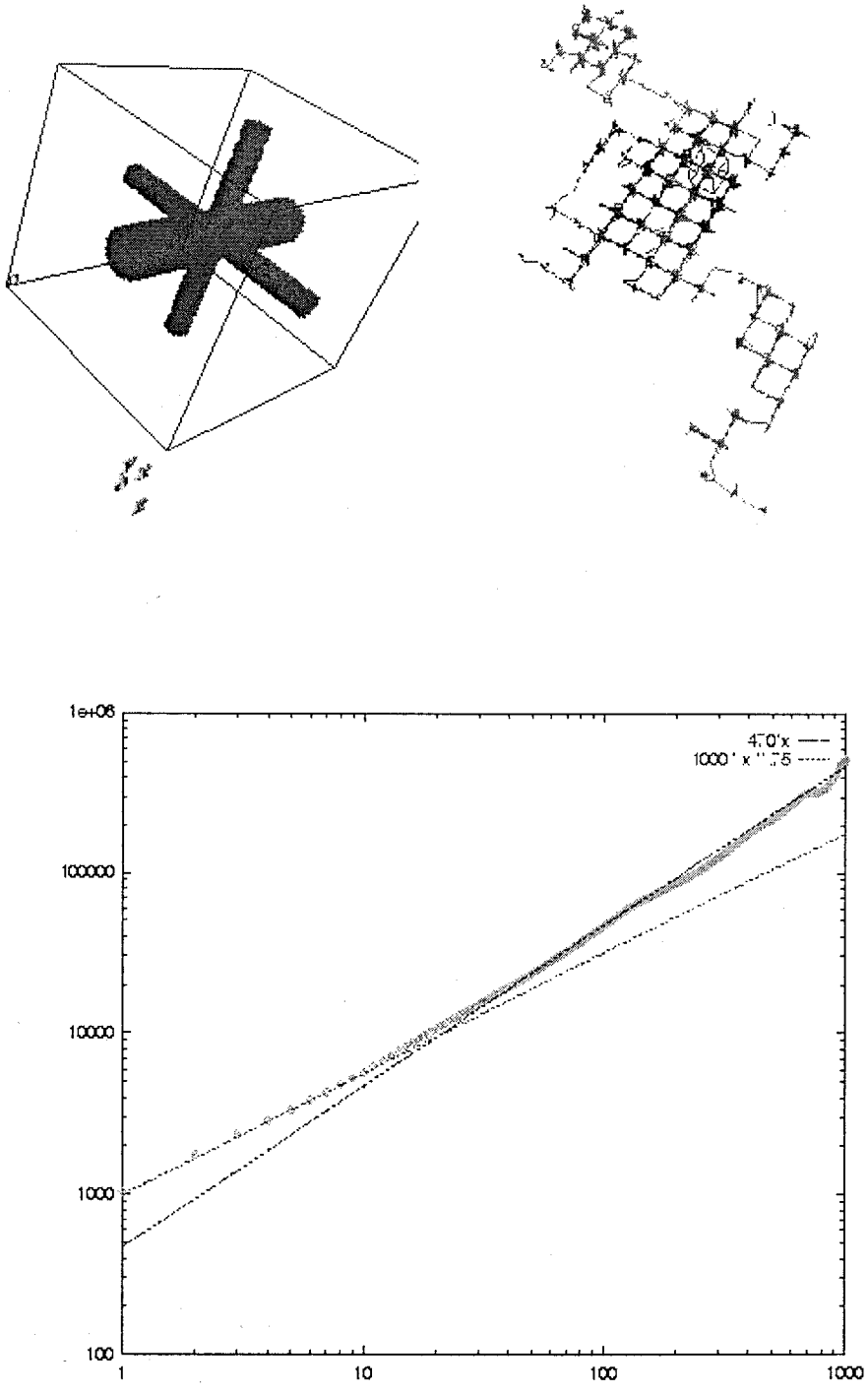


Figure 9. 2D PBC transversely connected cylinder.

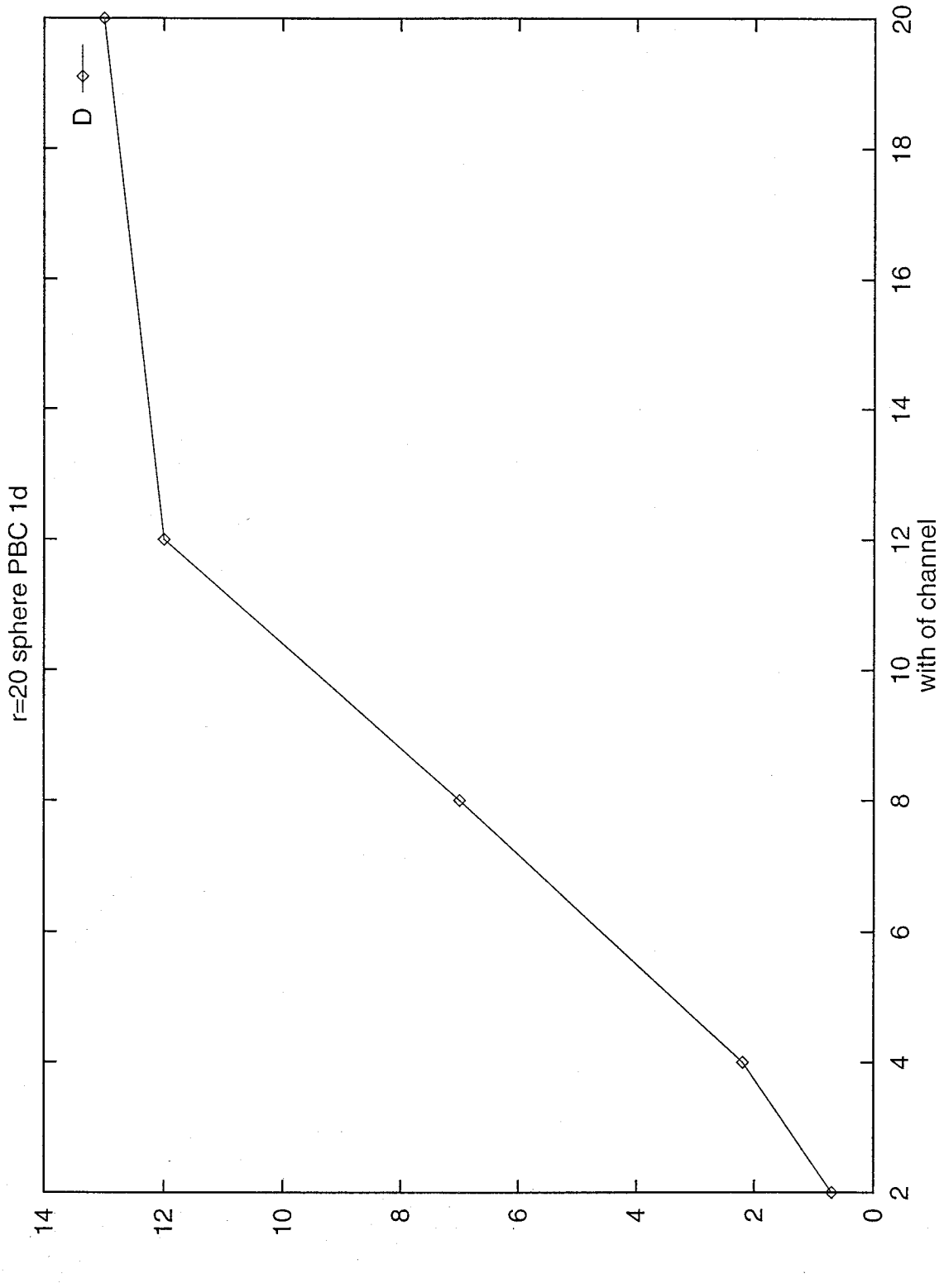


Figure 10a. Einstein Diffusion dependence on radius of channel connecting r=20 spheres at 100 distance from each other.

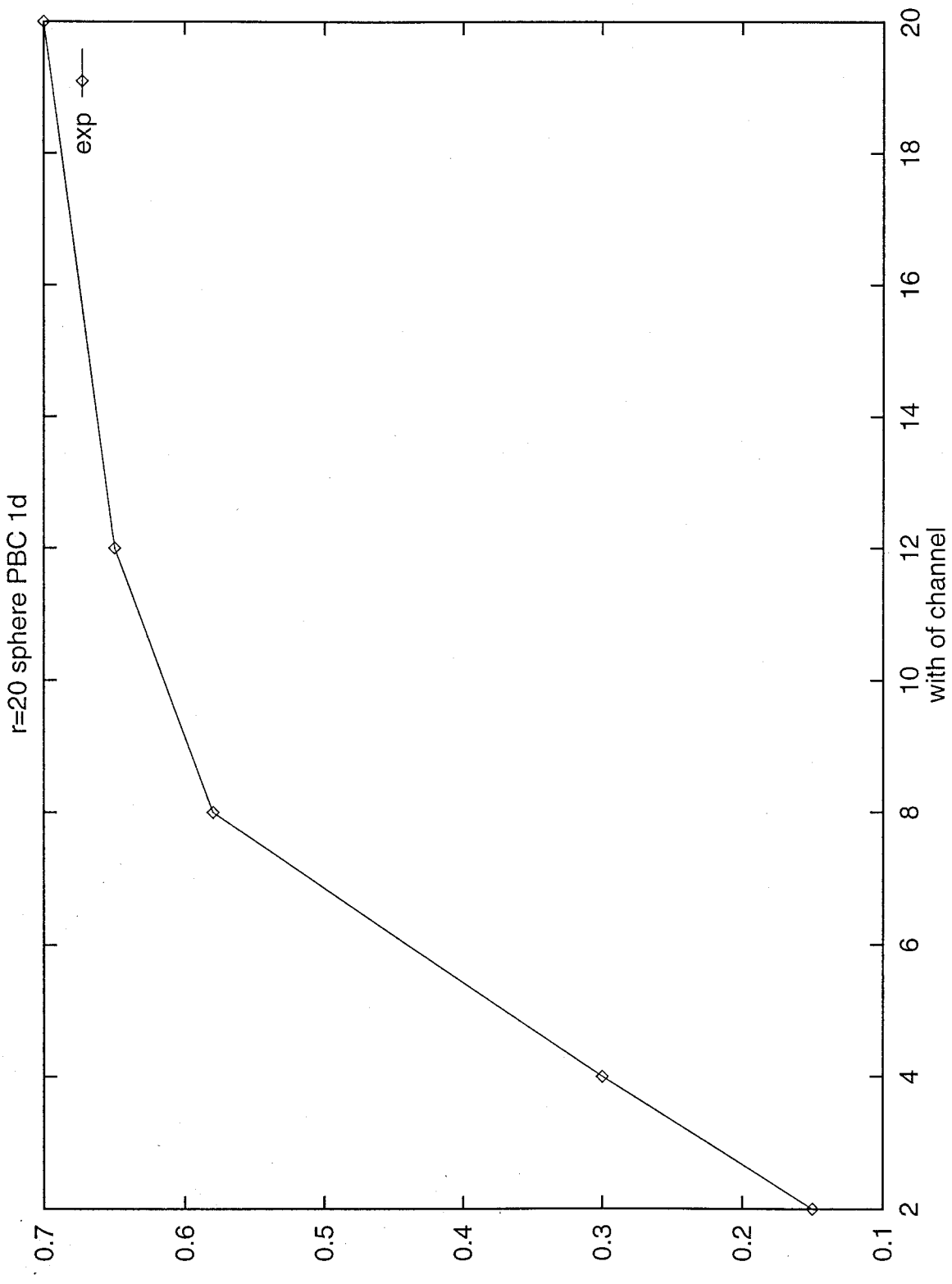


Figure 10b. Anomalous exponents for the same structure.

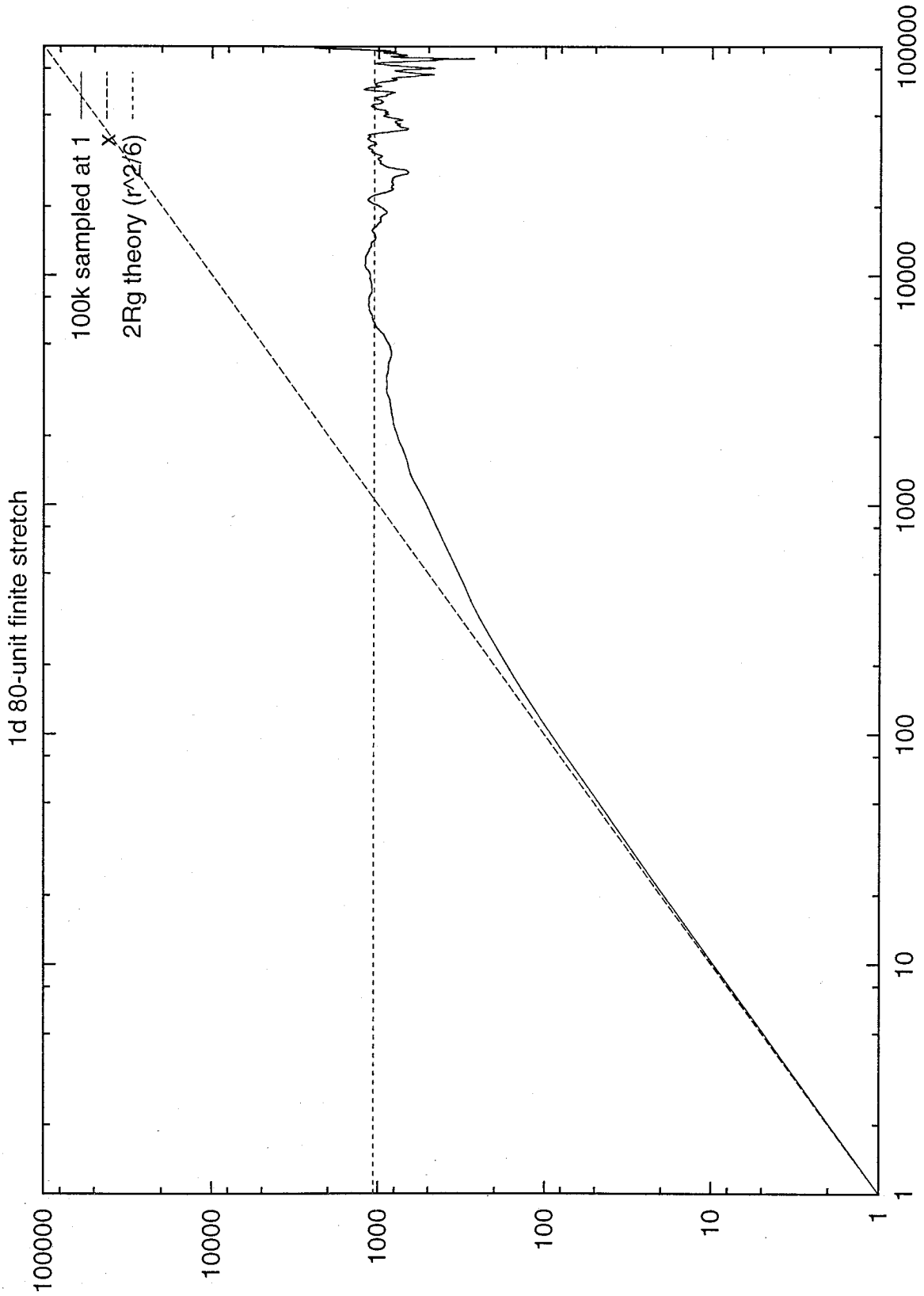


Figure 11a. 80 units long finite linear stretch.

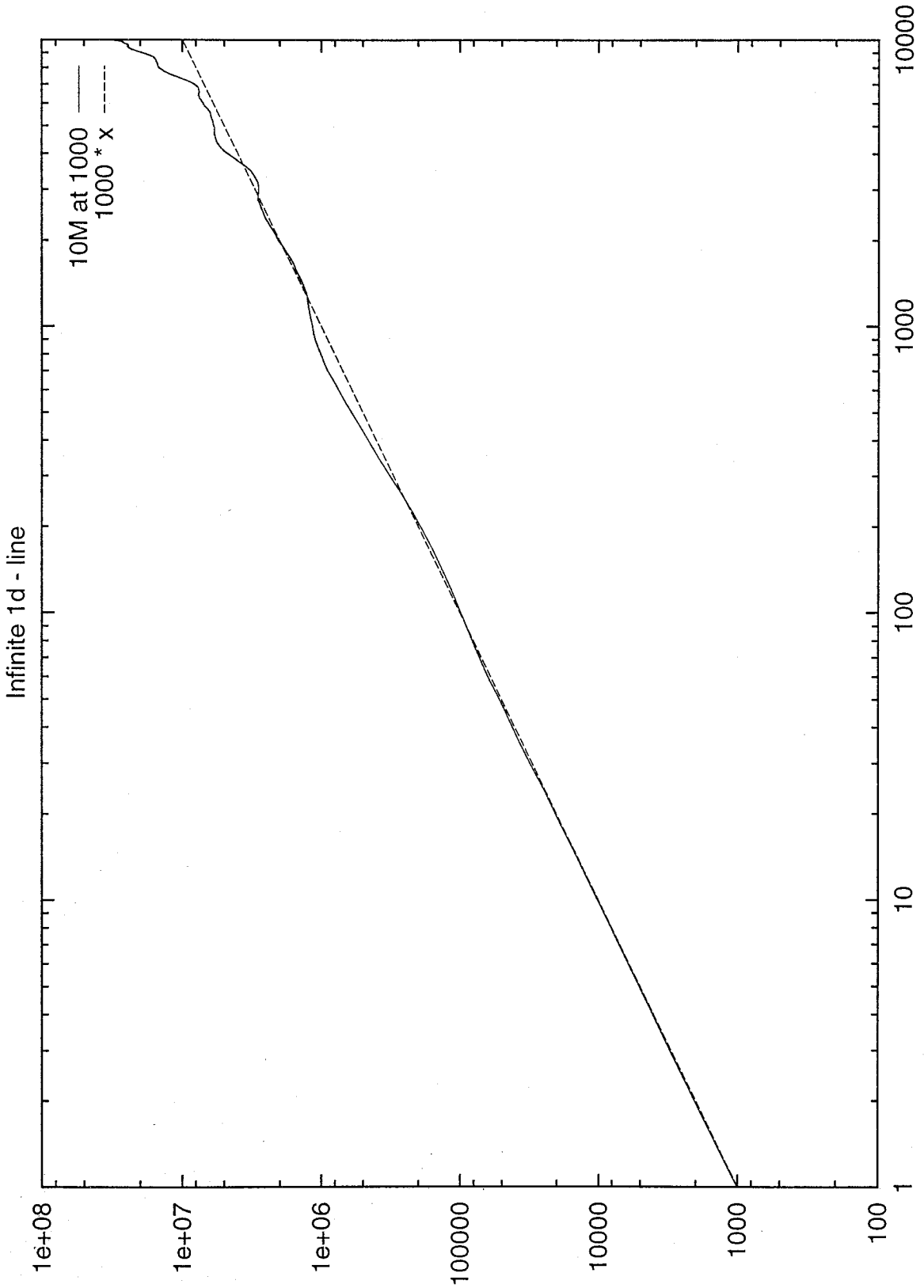


Figure 11b. Infinitely long 1D line.

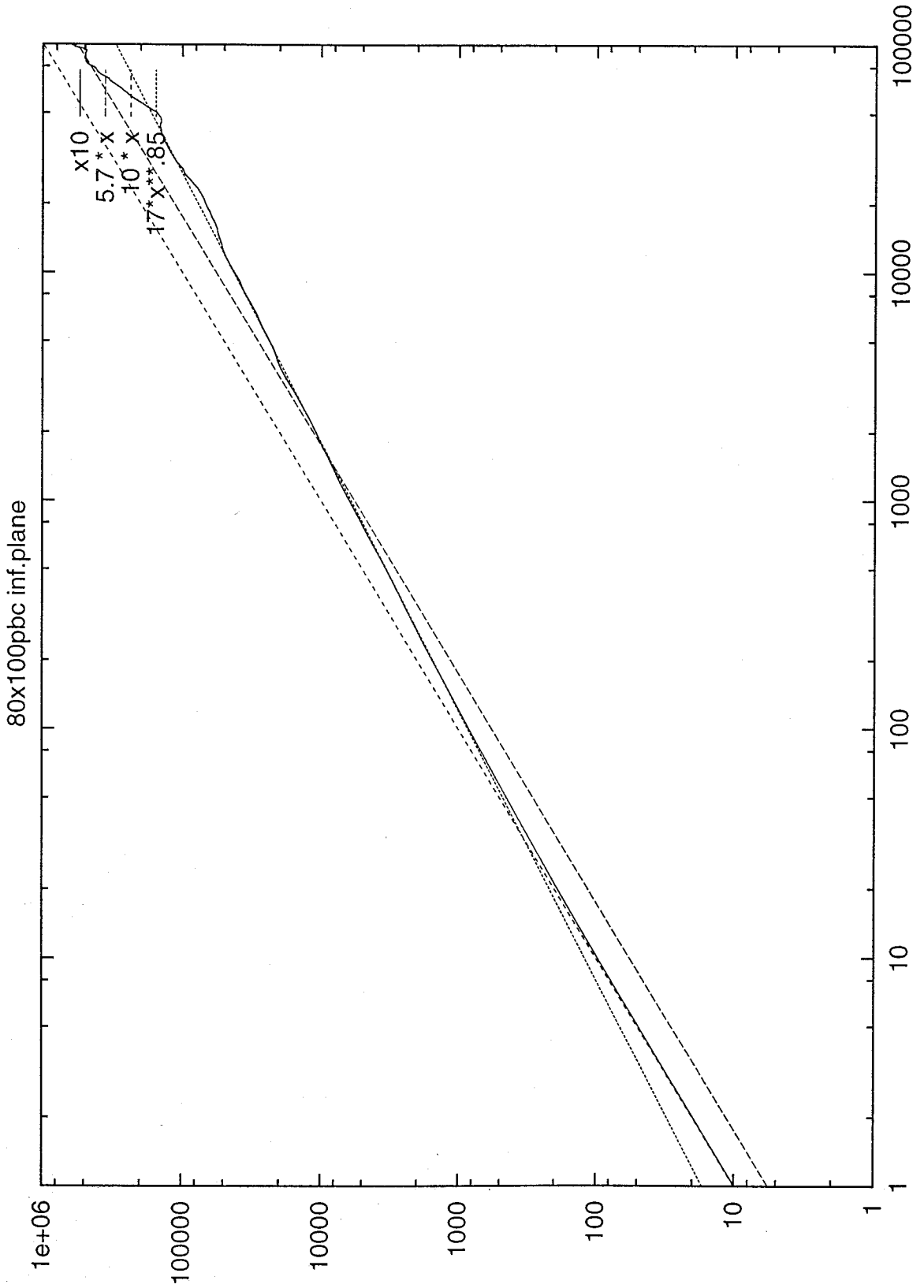


Figure 11c. 80 units wide infinite strip (a product of 11a and 11b).

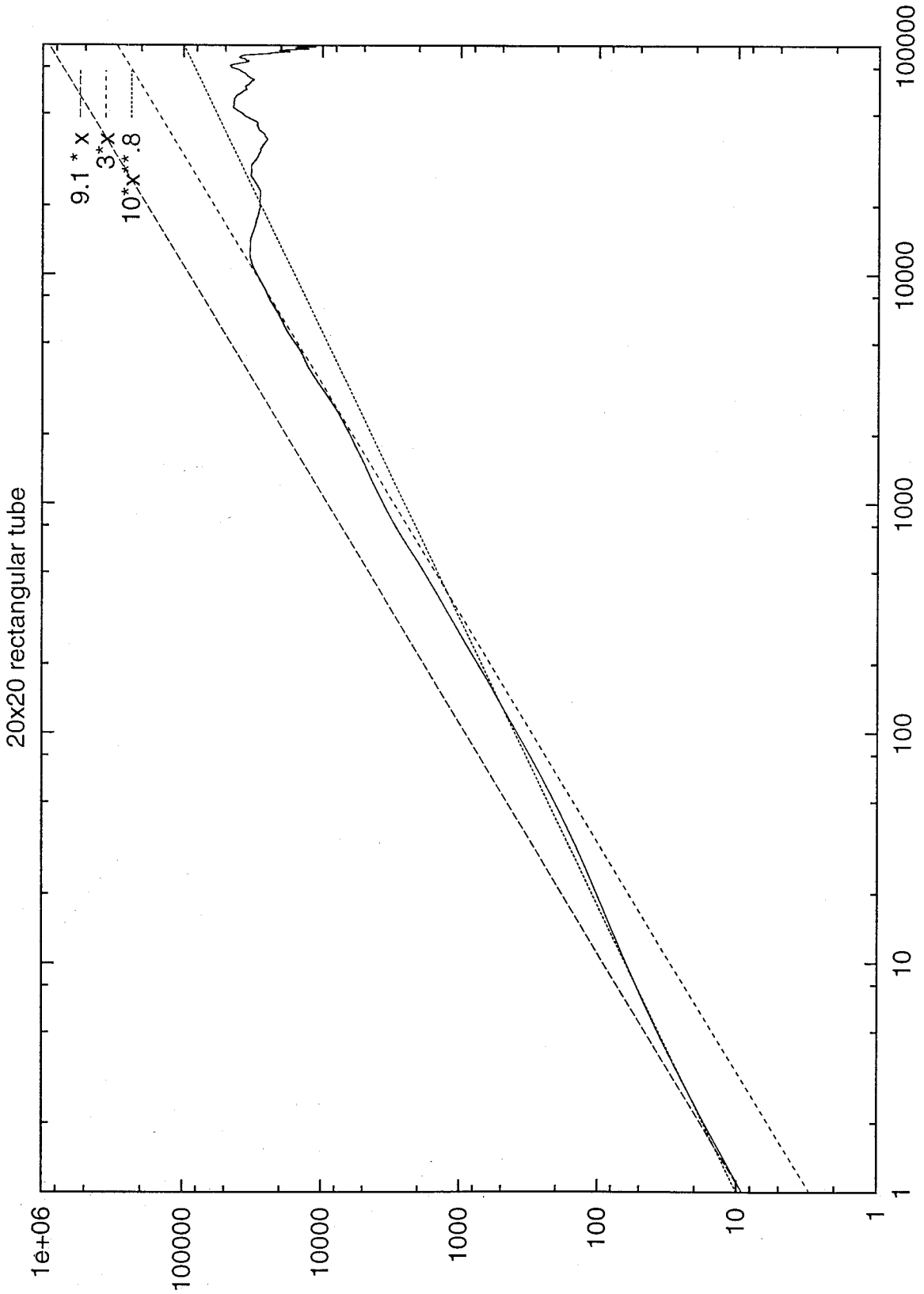


Figure 12. 20x20 infinite rectangular cube.

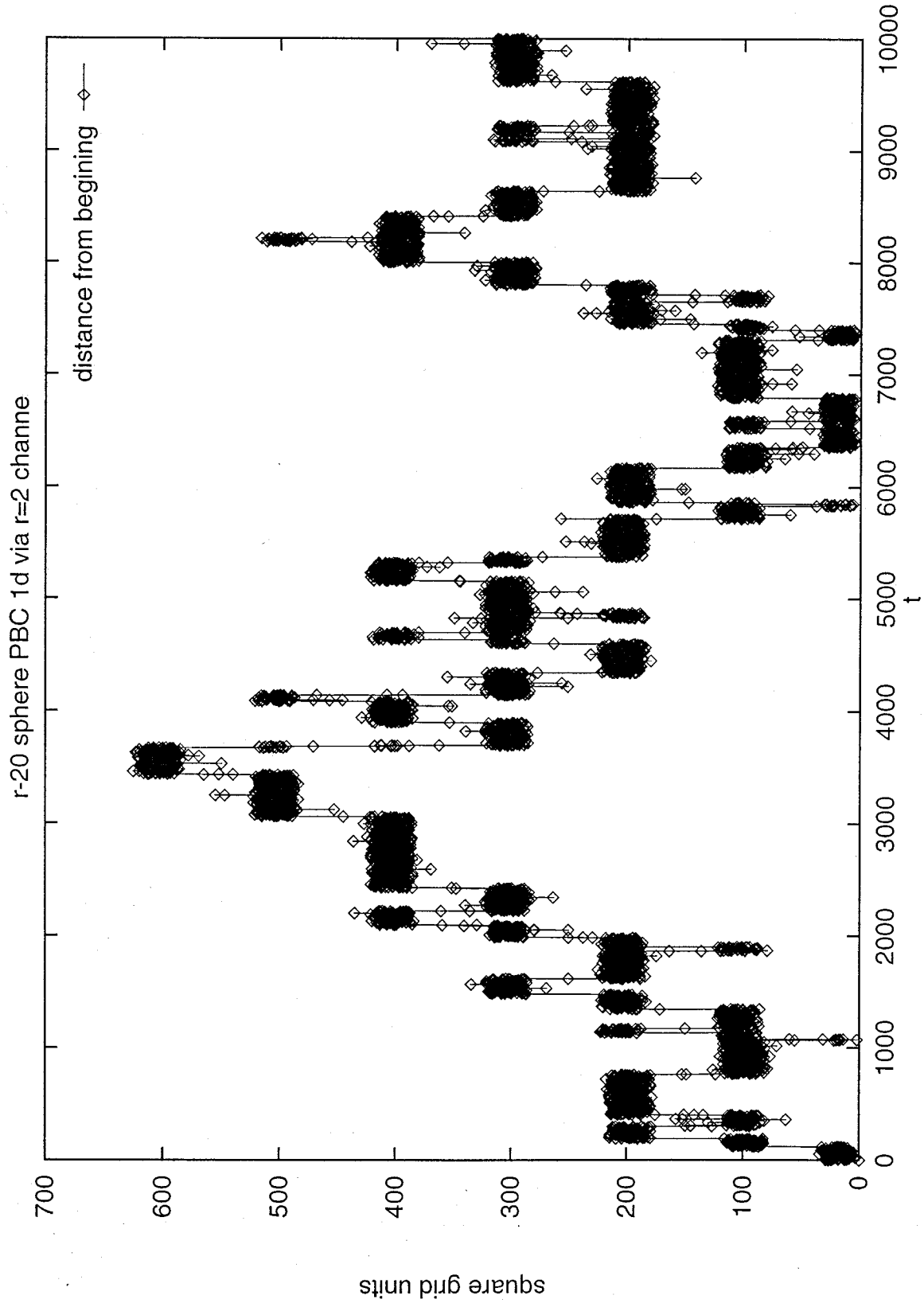


Figure 13a. Square displacement from the beginning (not averaged) for system similar to the one in Figure 2 with channel width of two.

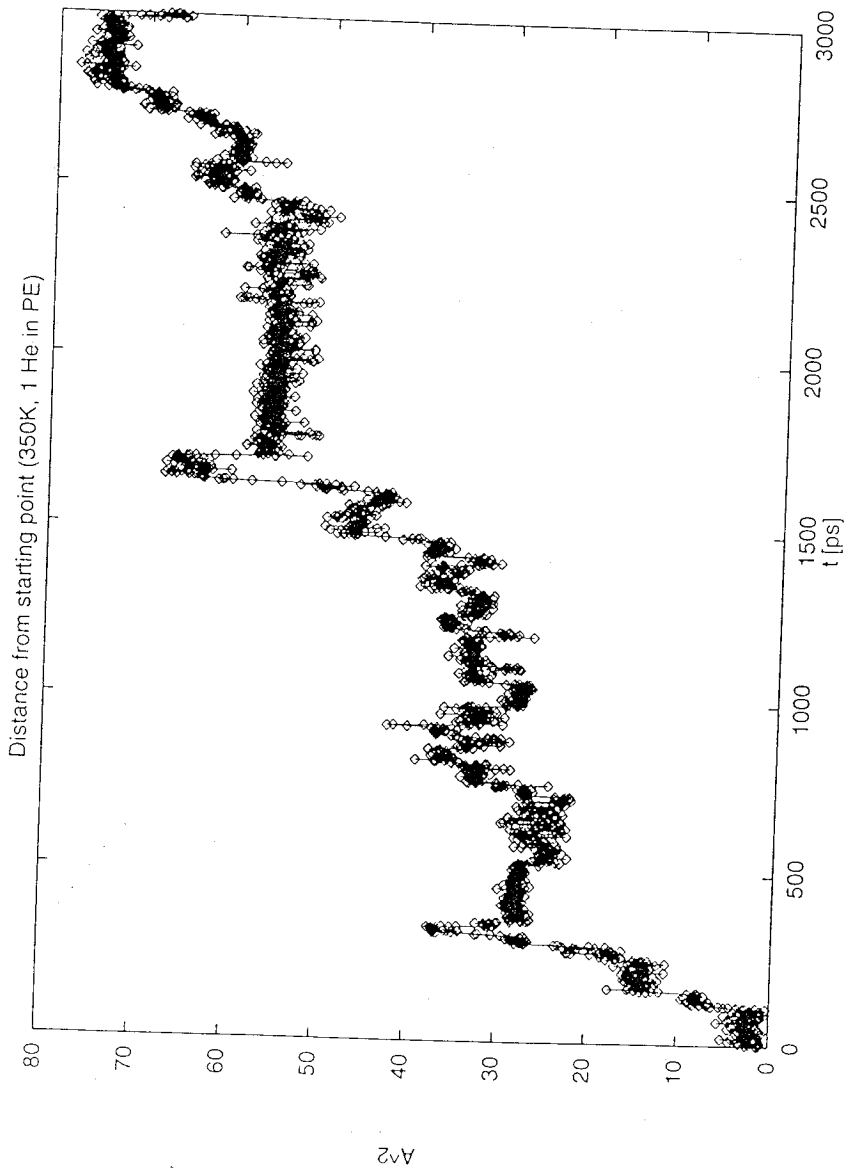


Figure 13b. Square displacement from the beginning (not averaged) of an MD run of He in PE at 350K.

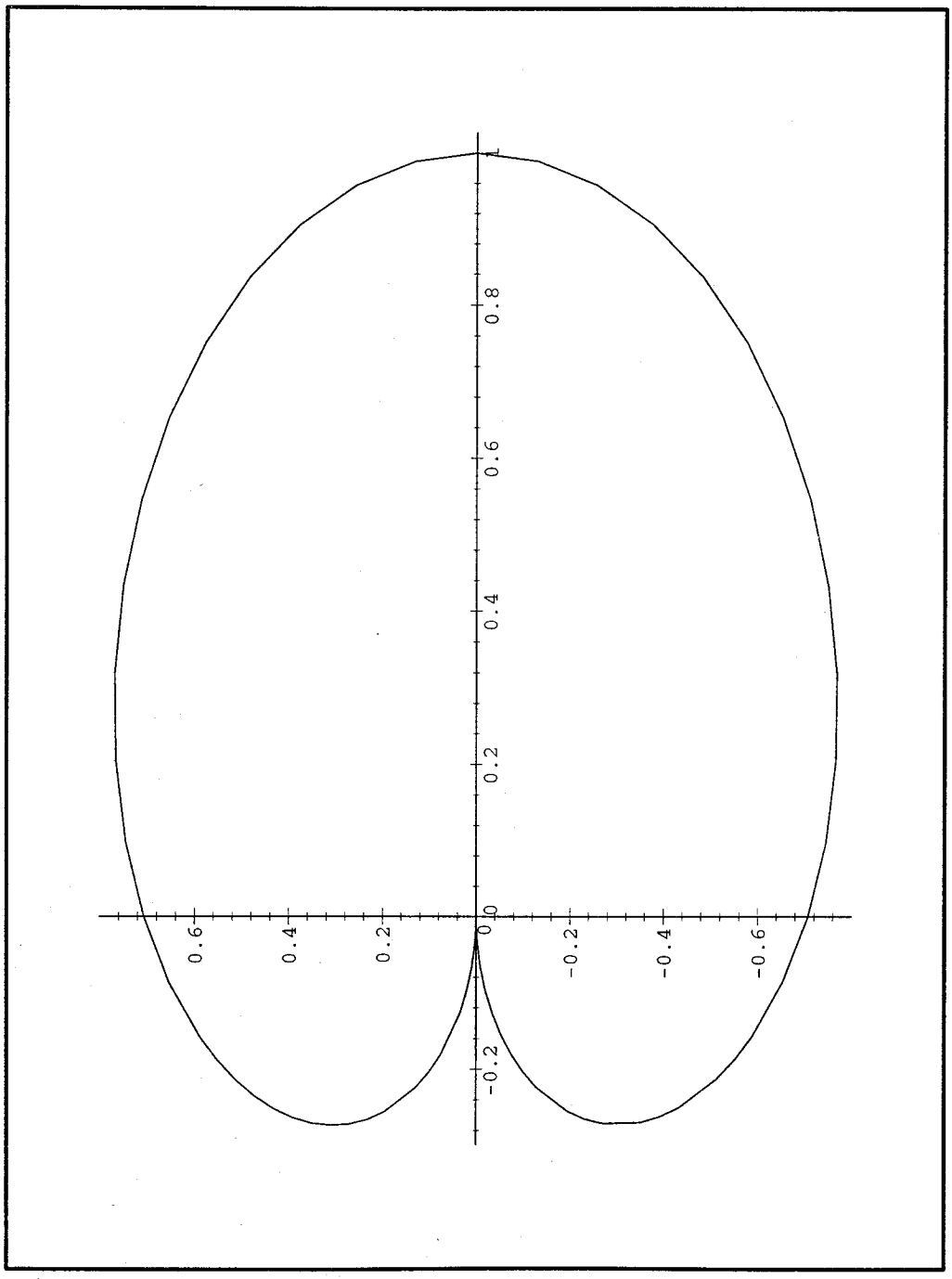


Figure 14a. Random walk Bias at q=0.5.

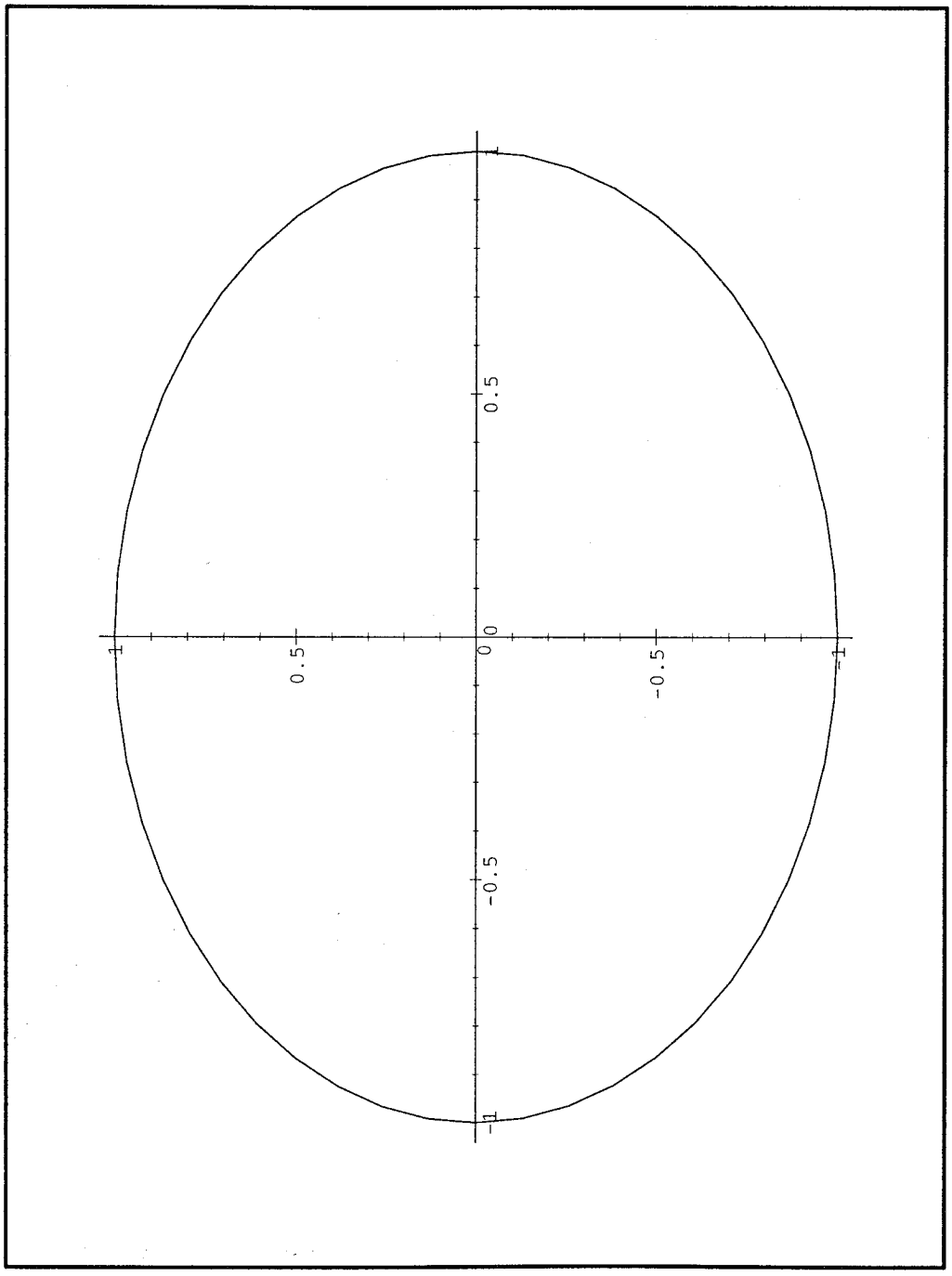


Figure 14b. Random walk Bias at q=0.0 (no bias).

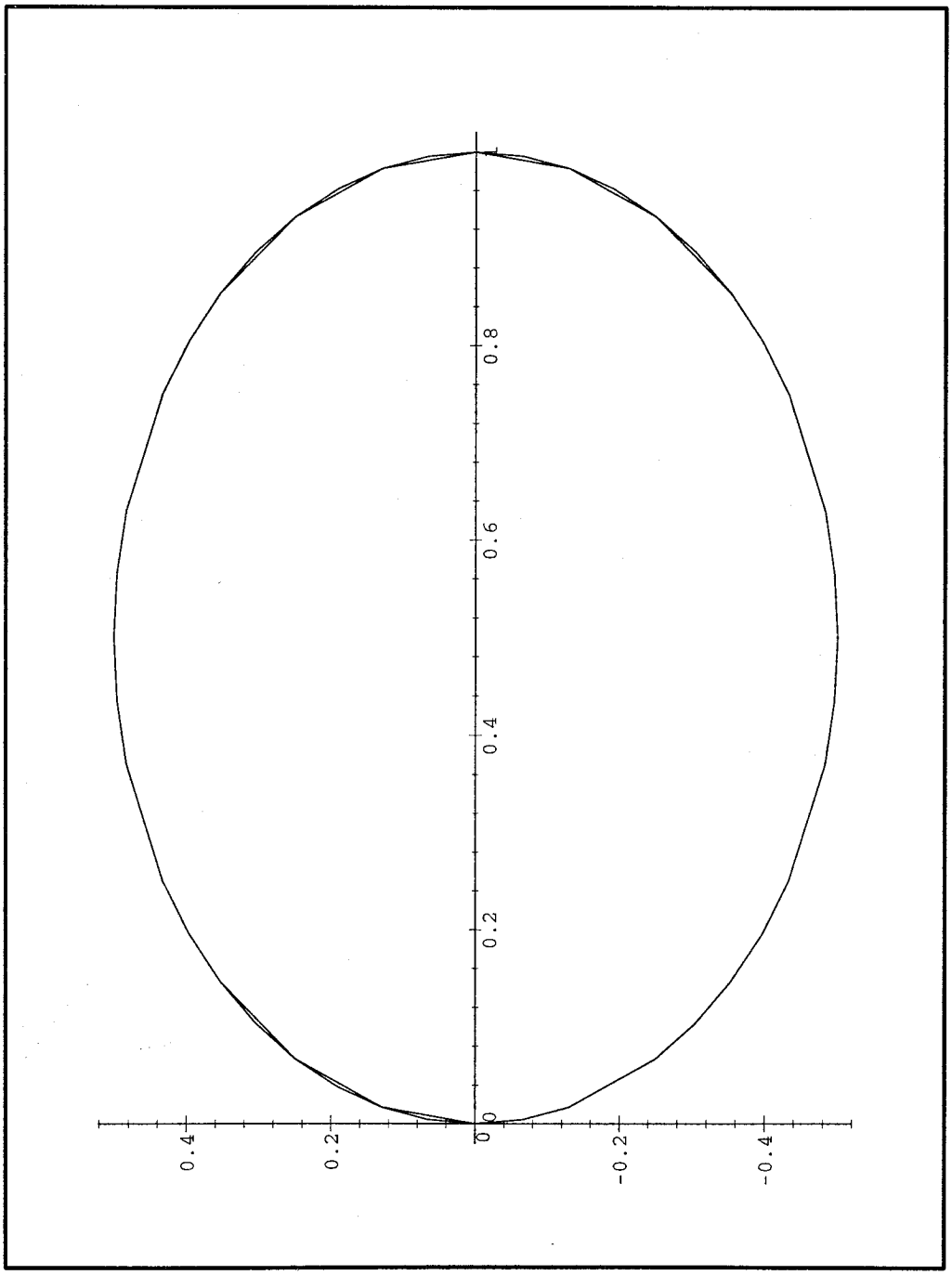


Figure 14c. Random walk Bias at q=1.0 (no backward motion).

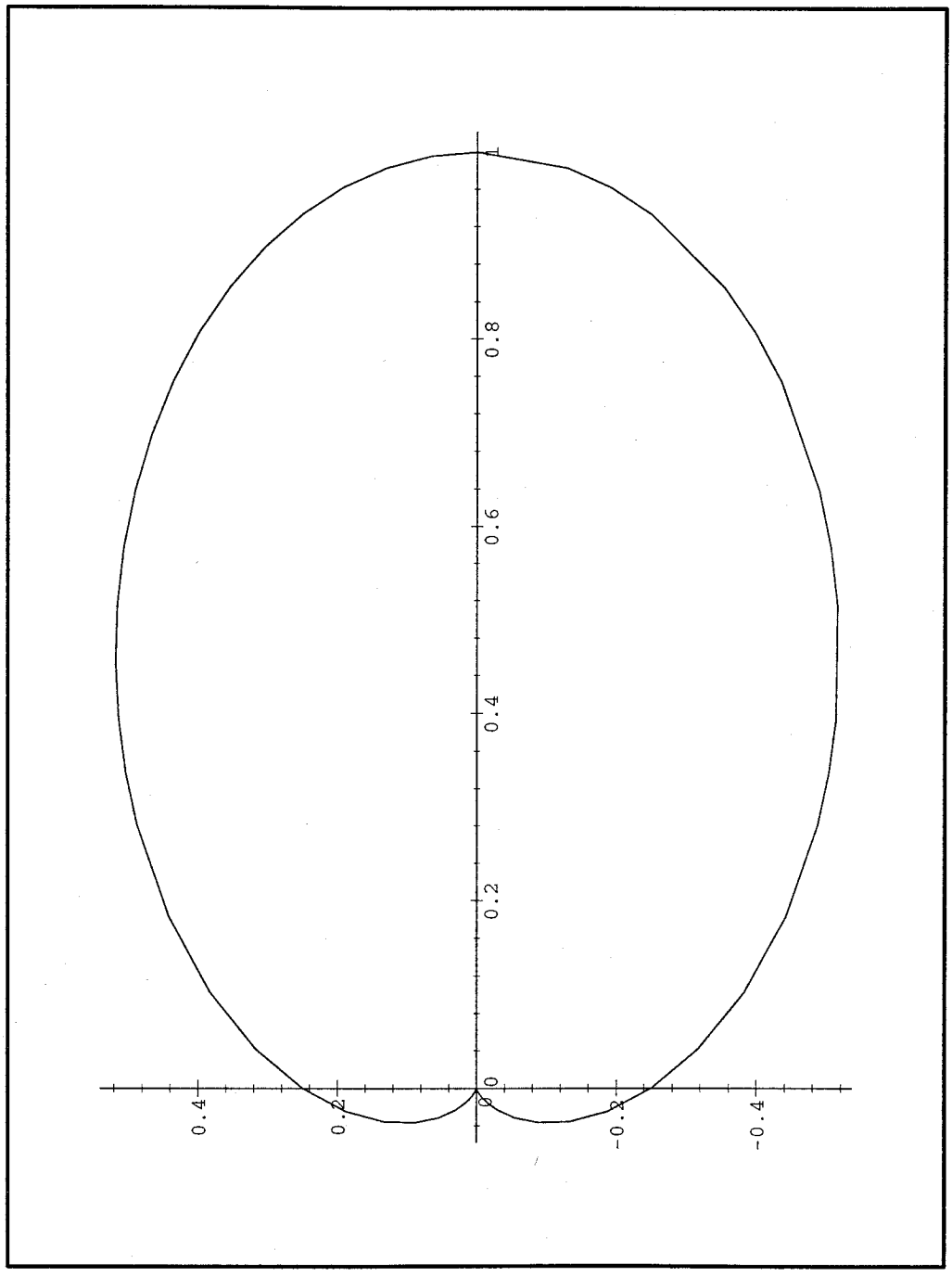


Figure 14d. Random walk Bias at q=0.5 and the term squared to point even sharper in forward direction.

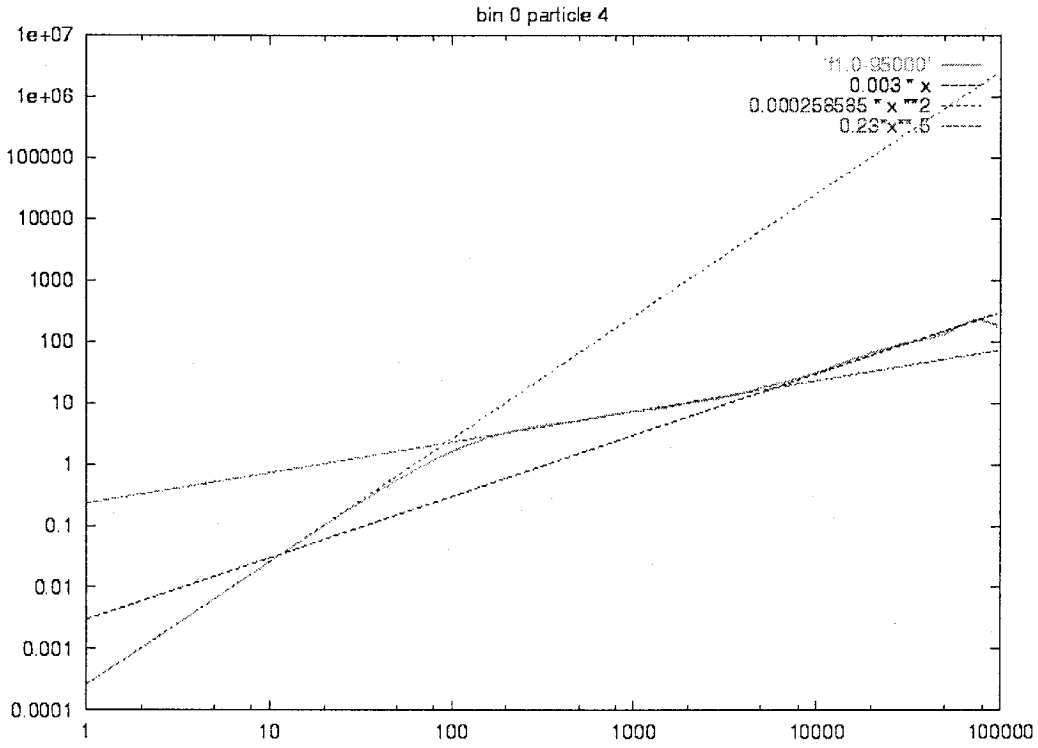


Figure 15. MD short range trajectory.

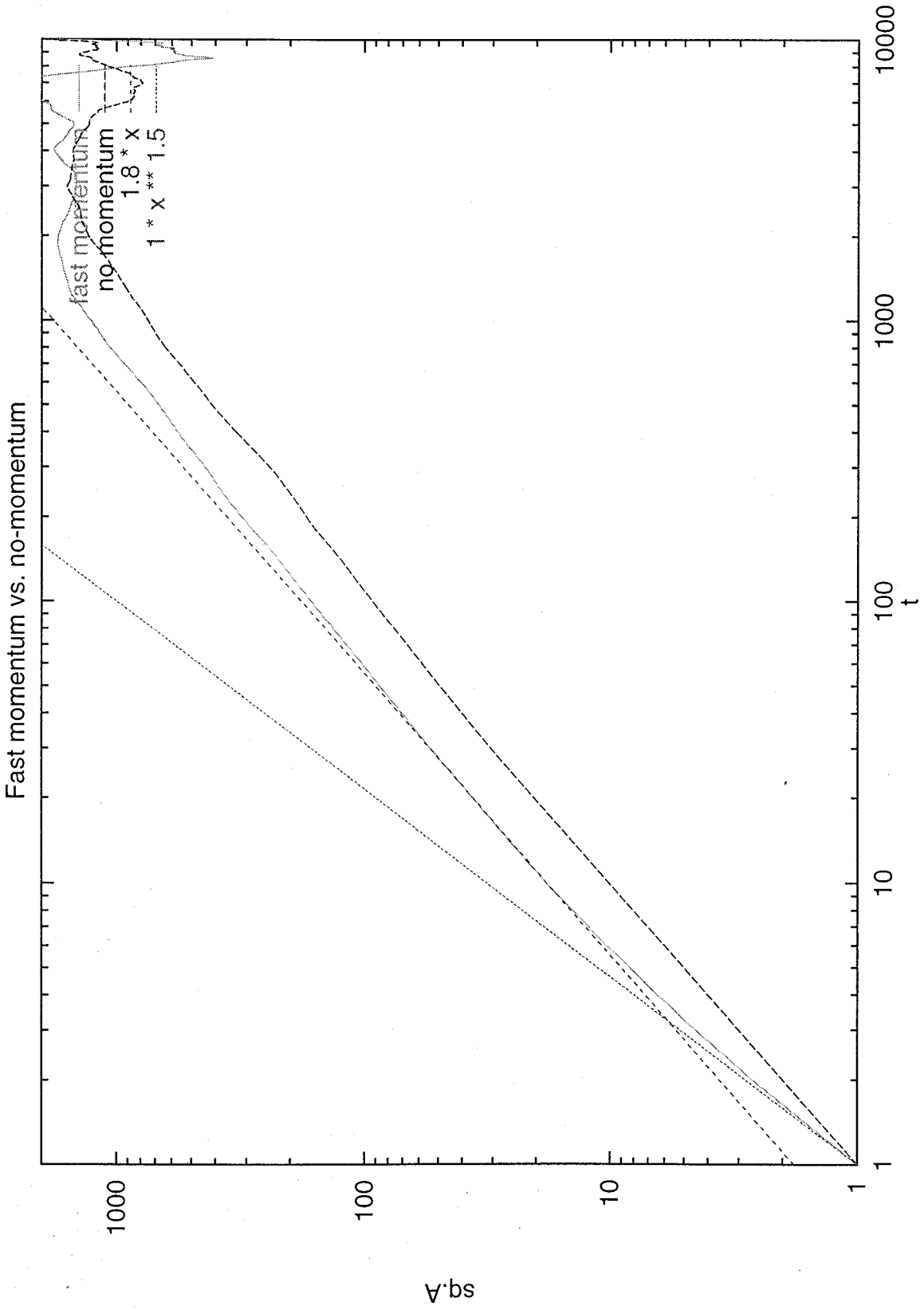


Figure 16. Fast momentum vs. no momentum. The ballistic region is apparent for the fast momentum case and is lacking for the no momentum case.

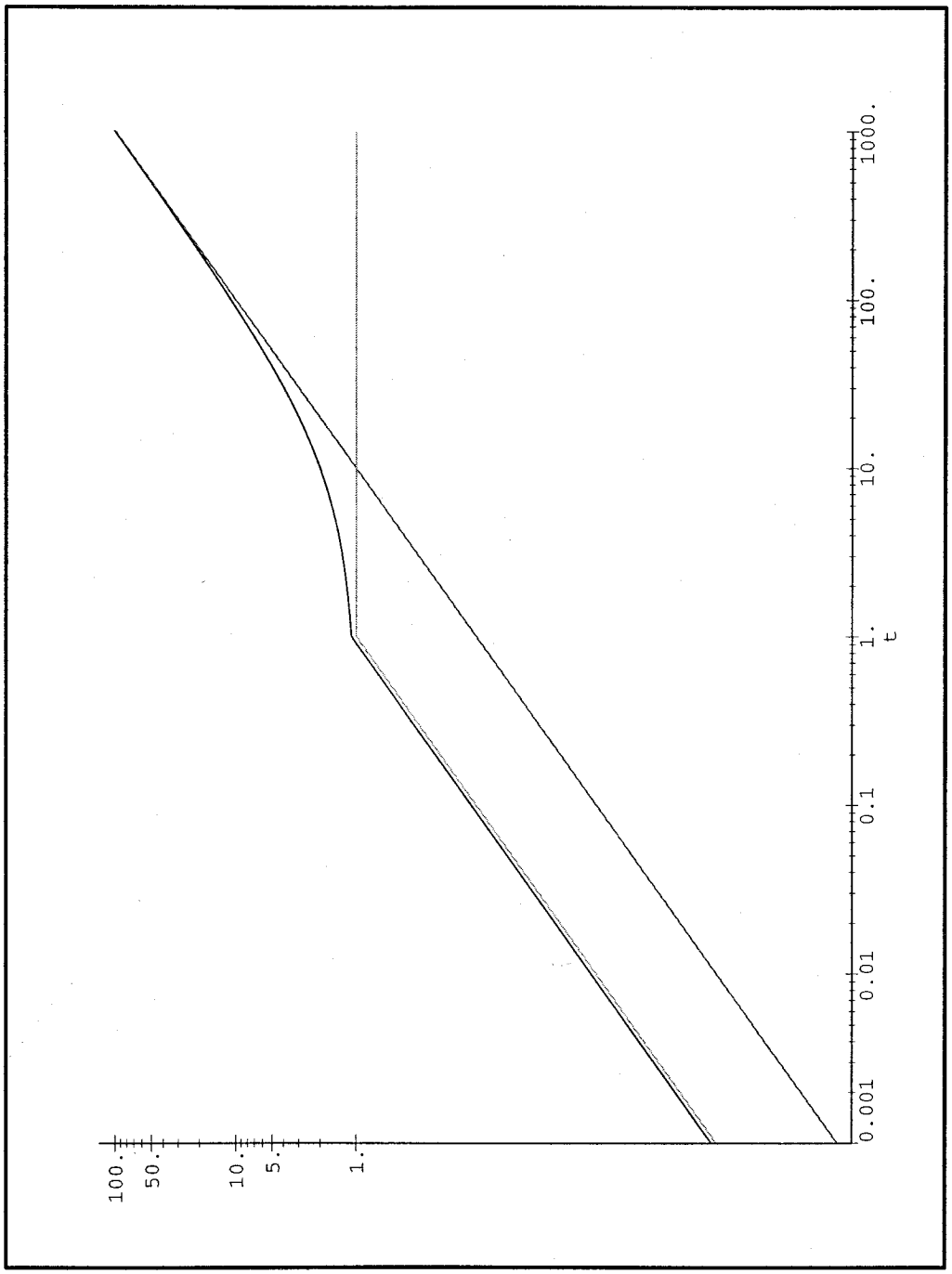


Figure 17. Modeling the anomalous diffusion zone from combining two one-dimensional cases.

J. Chem. Phys.
(to be submitted)

Diffusion of Gases in Amorphous Polymers: The Monte Carlo Void Method Based on Molecular Dynamics; Application to He Diffusion in Polyethylene

Seiichi Kashihara,[#] Mihail Iotov, Siddharth Dasgupta,* Guanghua Gao, Michael Belmares, and William A. Goddard III*

Materials and Process Simulation Center, Beckman Institute (139-74)
Division of Chemistry and Chemical Engineering
California Institute of Technology, Pasadena, California 91125

Abstract

The diffusion of small molecules in polymer matrices involves time scales (microseconds [μs] to milliseconds [ms]) far too long for routine molecular dynamics (MD) simulations. We have formulated a *new method* to extract coarse-grained information from short (1-2 nanoseconds [ns]) MD simulations and use this in a mesoscale simulation to calculate diffusion constants in polymer matrices. It uses a grid to calculate the average probability of each grid point of being a void and does constrained Monte Carlo (MC) dynamics to reach much longer time regimes than possible in MD. The MC method mimics the three regimes of mean square deviation (MSD) behavior seen in MD, thus accounting for the proper mobility of the voids and the compressibility of the polymer matrix. Initial results on He diffusion in a low-density polyethylene (PE) matrix are presented. The behavior at different temperatures follows closely the trend observed from calibrating long term MD for this particular system. Details of the methodology are discussed in this paper.

[#]Permanent Address: Asahi Chemical Industry Co., Ltd., 2-1 Samejima, Fuji, Shizuoka 416, Japan

*Authors to whom correspondence should be sent: e-mail: sdg@wag.caltech.edu and wag@wag.caltech.edu.

1.0 Introduction

The diffusion of gases in polymers is of major importance to the polymer industry.[1] For example, various applications in the food industry involve impeding the diffusion of gases (such as H_2, O_2, CO_2, H_2O, and CH_4) through thin polymer films.[2] The design of new copolymers, new monomers, or new blends to selectively impede the diffusion of some gases while allowing others (O_2 versus N_2, CO_2 versus O_2, H_2O versus O_2) underlies a number of potential applications. The design and optimization of the selectivity of polymer membranes for diffusion would be greatly facilitated if reliable predictions of diffusivity could be made rapidly in advance of synthesis and experiment. The time scale for diffusion (μs to ms) is far too long for routine applications of molecular dynamics (MD). This has made it difficult to obtain reliable diffusion data from theory.[3] In addition, it is not easy to obtain reproducible data on gas diffusion in polymers from experiment. As a result there is little in the way of reliable predictions on how to design polymer films to achieve specific diffusion properties.

The permeability (P) of a gas through a membrane can be written as[4]

$$P = D\,S \tag{1}$$

where D is the diffusion coefficient and S is the solubility. Generally only P is available experimentally, but D and S depend differently on the various design parameters. Thus, it is useful to be able to independently measure each. In this paper we will focus on D.

In the long time limit for a three-dimensional system, Einstein showed that the total distance (R) traveled in a time (t) is given by Eq. (2) (the Einstein relation)

$$\langle R^2 \rangle = 6\,D\,t \quad \text{as } t \rightarrow \infty \tag{2}$$

where < > designates averaging over the ensemble of starting and ending points for the given time interval t. To use theory to predict D for a gas X in a polymer, we carry out a number of MD calculations (each starting with X in various sites) for times sufficiently long that (2) is obeyed.

Some typical results are shown in Figure 1. Figure 1a shows that for He in PE a 1.5 ns run leads to a reliable D while Figure 1b shows that for O_2 in a PVC/PVDC copolymer, 1.5 ns of MD is not even close to satisfying Eq. (2). (We estimate that this latter case requires a μs time scale.) The problem with using MD to study gaseous diffusion is the time scale. It is necessary for the calculation to proceed for a time t sufficiently long that the Einstein relation (2) is valid. Even for simple gases in polymers, the relevant times are often on the order of μs or longer. Since the MD time step is ~1 femtosecond (fs), a total time of μs requires ~1 billion steps. For a polymer of realistic molecular weight this requires very long times on even the largest supercomputer. Consequently, it is usually not practical to use MD to obtain reliable values for D.

In order to gain insight about the features of an amorphous polymer dominating the diffusional properties, we carried out MD studies of various gases diffusing through various polymers. Typical results are shown in Figure 2. Figure 2a shows the trajectories for 5 He atoms diffusing for 1.5 ns in PE. Here the atoms of the polymer are not shown. We see that there are three-dimensional regions (referred to here as *felicitons*) in which the gases spend significant times (≈20-70 picoseconds [ps]) separated by pseudo one-dimensional channels where they spend shorter times (≈5 to 20 ps). On the other hand, Figure 2b shows the trajectory for $5O_2$ diffusing for 1.5 ns in PVC/PVDC. Here we see that each O_2 remains within a feliciton, with no channels connecting them.

The felicitons are regions that better accommodate the diffusing particle - because this region has more available free volume or a higher compressibility. Thus, one can think of the polymer as consisting of felicitons connected by channels with the diffusing particle hopping from feliciton to feliciton through the connecting channels. We should emphasize that this feliciton/channel concept of diffusion is dynamic. At any instant there are no large voids in the polymer. However, averaged over the time scale that a particle can diffuse through a region of the polymer, there are some regions - felicitons and channels - better able to accommodate the particle than others. This framework of felicitons and channels then controls the diffusion. If insufficient channels connect the felicitons, there may not be the percolation necessary for long-range diffusion.

Numerous calculations on different gas molecules diffusing in various amorphous polymers give results such as in Figure 2. We also find that the regions occupied by the diffusing particles are defined by the dynamics of the polymer even

without the particle! Thus, Figure 3 shows the dynamical void distributions in the two polymers of Figure 1 and Figure 2. Here we partitioned the unit cell into one million cells (100×100×100) and examined the voids over a period of 200 ps. Every 5 ps we examined whether a probe of 1Å radius would contact any part of the polymer. The color code in Figure 3 indicates the fraction of the times that there was a void of at least 1Å radius at that site. Here we see that the void analysis leads to an excellent match with the felicitons and channels extracted from the MD.

Given that the topology of felicitons and channels can be determined independent of the diffusing particle, we asked whether we might estimate the diffusion constants without actually considering the diffusing particle! Thus, we consider a Monte Carlo process in which a pseudo particle migrates from one void site to an adjacent void site. Indeed, we find that such Monte Carlo Void Diffusion (MCVD) procedure lead to a diffusion trajectory very similar to the MD calculations, as shown in Figure 4.

Such MCVD calculations are many orders of magnitude faster than MD, allowing us to easily consider time scales long enough to see diffusion even in O_2/PVC-PVDC. However, to obtain the actual diffusion constant from MCVD, we must convert implicit time of a MC jump to the physical time. This is done by comparing the distance vs. MC step behavior from MCVD with the distance vs. real time behavior from MD. We find that reliable MC step to time conversion can be obtained from even short time MD. This procedure provides a practical means of predicting diffusion properties of gases in polymers. In this paper we will focus on He/PE. Subsequent papers will consider other penetrants (O_2, N_2, H_2O, CH_4, C_6H_6, etc.) as well as other polymer matrices - polypropylene, polyisobutane, polystyrene, polyvinyl chloride, polyvinylidene chloride, etc.

Section 2.0 describes the diffusion of He in PE using atomistic MD. Section 3.0 describes the void analysis procedures. Section 4.0 presents the Monte Carlo Void Diffusion method. The diffusion of He atoms in polyethylene is analyzed in Section 5.0.

2.0 Molecular Dynamics

2.1 The Amorphous Polymers

We are interested in studying the diffusion of gases in amorphous polymers. A system such as PE will often have a high percentage of crystalline lamella. In

semi-crystalline polymers, the most widely accepted view[2] of diffusion is the electrical analog of porous media (amorphous region). According to this view, the crystalline region of the polymer is impermeable. Therefore, the effective diffusion can be derived from the diffusivity of the gas in the pure amorphous region. We consider using different approaches to equilibrate randomized conformations of polymers in order to provide the ensemble of structures mimicking amorphous polymers. We have found the following procedure to be effective and used it in the calculations reported here.

For MD calculations on amorphous polymers, we use periodic boundary conditions (PBC) so that a finite number of independent atoms can describe an infinite polymer system without surfaces. Our calculations allowed four independent chains of PE per unit cell (each with 200 carbons, a molecular weight of 2802 Dalton/chain). We started by allowing each chain to have a random conformation but with the requirement that the chain-chain interactions be reasonable.

To do this we built the initial structure[5] using Rotational Isomeric State (RIS) theory[6] to select the torsional distribution, assuming the trans state to be 0.8 kcal/mol more stable than gauche, for a temperature of 400K. At the end of the growth process, we used energy minimization to relax the structure removing initial bad contacts. We than carried out MD to anneal the system using the MPSim[7] program developed at Caltech. These calculations used 8 cells of the KSR1 parallel computer. An initial MD at 500K was run for 100ps to remove bad contacts. The unit cell is a cube of volume 21835 $Å^3$ (27.95 Å sides), leading to a final density of 0.854 gm/cc, which matches the density of the amorphous component of PE.[8]

Figure 5 shows a typical amorphous system built using this procedure consisting of 4 chains of PE with 100 monomers (200 backbone C atoms) per chain. For purposes of clarity, the H atoms have been removed and the various chains are color coded to follow the chain conformation and entanglements.

2.2 Placement of the He Atoms

To locate the best places to start the penetrant He atoms, we calculated the voids in the system. We divided the cell into a million cells (100×100×100 grid) and at each grid point checked for the presence of nearby polymer atoms. Here we count a microvoid at a potential grid point if there was no polymer atom whose van der Waals radius was within 1.0Å - the probe radius - of the grid point. The contiguous microvoids were collected together to form a macrovoid which is characterized by the

centroid and radius of the largest single sphere that fits within the macrovoid. The program reports the centroids of these voids in descending order of size, Table 1. A probe radius of 1.0 Å leads to the voids shown in Figure 3. We then placed He atoms near centers of the voids. The simulations used 5 He atoms per unit cell, but we turned off the He-He interactions. This allows us to obtain good statistics while avoiding artifacts that might arise from the He-He interactions.[9]

MD simulations were then carried out at various temperatures using MPSim. Since explicit H atoms are described, a time step of 1 fs was used for the MD. The trajectory of the PE chains and the He atoms was saved at 1ps intervals.

We used the MS force field (FF) for PE,[10] which gives excellent agreement[11] with experimental data for crystalline PE. However, the torsional potential about the C-C bond was determined from quantum chemical calculations on n-pentane.[12] The typical calculation on 8 cells of the KSR1 leads to 50 ps of MD per day.

2.3 Analysis of the Diffusion

Figure 6 shows the track of the 5 He atoms in a 10 ns simulation. It is quite clear that the He atoms spend most of their time traversing the felicitons, occasionally hopping through a channel to an adjacent feliciton.

To calculate the mean square displacement (MSD) with time, we considered each ps step of the 10 ns trajectory to be a possible starting point. Thus,

$$R^2(\tau) = \sum_{t_o=0}^{T-\tau} |R(t_o + \tau) - R(t_o)|^2 \Big/ \sum_{t_o=0}^{T-\tau} 1$$

where T is the total time. This leads to a sampling size that decreases from 10^4 at τ = 1 ps to 1 at τ = 10 ns. Thus, the value of $\langle R^2 \rangle(\tau)$ is not expected to be reliable for $\tau \gg 1/2$ T. To determine whether the system is in the diffusing region where the Einstein relation (2) holds, we plot log $\langle R^2 \rangle$ vs. log τ, which should have a slope of unity. Indeed, Figure 7 shows diffusive behavior for $\tau \geq 0.1$ ns averaged over all 5 He atoms. The intercept gives 6D which leads to D = 1.77 A^2/ps = 29.5 × 10^{-6} cm^2/sec.

The diffusion coefficient of He in LDPE (ρ = 0.914 g/cc) at 25°C is 6.8×10^{-6} cm^2/sec and in HDPE (ρ = 0.964 g/cc) at 25°C is 3.07×10^{-6} cm^2/sec.[13] Allowing for the difference in density, polymer length, system size and temperature, our calculated value is in the expected range.

Figure 7 shows that the Einstein relation (2) is satisfied above times of ~100 ps, corresponding to diffusion length of $\sqrt{< R^2 >} \sim 13$Å (indicated by the line with long dashes). The shorter time scale region is asymptotic to the short dashed line where $<R^2>$ is proportional to $t^{1/2}$. The analysis in Figure 7 used trajectory data that was saved only at 1ps intervals during the 10 ns MD run. Consequently, this does not allow an analysis of the early time behavior below ~5 ps. To examine the early region, we ran the MD for 95 ps and saved the trajectory every fs. This leads to Figure 8a. Up to $<R^2> \approx 0.25$ Å^2 and times up to ~0.03 ps, the distance increases rapidly with time ($<R^2> \propto t^2$ or $R = vt$ where $v \approx 506$ Å/ps is the velocity). This is the *ballistic regime*, before the particle hits a wall. From $<R^2> \sim 4$Å to 12Å and time from ~0.25 ps to ~4 ps, the trajectory fits the $<R^2> \propto t^{1/2}$ line quite well. This is the *anomalous diffusion region*, corresponding to intra-feliciton motion. During this time the motion is primarily within felicitons, suggesting an average feliciton size of 12Å diameters. We find such an anomalous diffusion region (where $<R^2> \propto t^{1/2}$) for diffusion in all amorphous polymers.

3.0 Void Analysis

As discussed in the Introduction, our MD studies of diffusion always show well-defined feliciton regions separated by channels in which the gas particle spends considerable time. We can define void analysis procedures that lead to the same regions in space that the gas particle samples during diffusion.

The question is what size to use for the probe radius in defining the void region. In Figure 9 we show the void analysis for various sized probes from 0.3 to 1.3Å. This analysis can be compared to the diffusion trajectory in Figure 2a, which has the same orientation. We see that probe radius of 0.9Å leads to a definition of the felicitons that matches the dynamics trajectory well. By running MD for 500 ps (the first 300 ps to adequately equilibrate the system and then 200 ps for sampling), we extract snapshots at 5 ps intervals to give 40 snapshots spread over 200 ps. Then we calculate whether there is a void larger than the probe radius at each of the 1,000,000 grid points. Each grid point is then weighted by the fraction of the 40 snapshots that it is a void. This weight embodies atomistic information about the polymer matrix like compressibility, porosity, and dynamical behavior.

4.0 The Monte Carlo Void Diffusion (MCVD) Model

Given the void probability grid from Figure 3 we will consider a random walk in which a particle on a grid point moves to an adjacent grid point. We assume the probability to be proportional to the void weight - 1 if always a void to 0 if never a void. Figure 10 shows a MD trajectory within a void at 1 fs resolution. In order for a random walk process to mimic this, we include a bias in the jump probability with an angular dependence based on the last previous step direction. We find that a simple cosine term, of the angle between the last step and the next possible jump direction, leads to a smooth linear trajectory within a feliciton with the particle changing direction mostly on reaching the edge of the feliciton. This mimics the MD trajectory as indicated in Figure 8b.

A second issue concerns which neighboring points are allowed for the jump. Simple choices are

a) only the 6 nearest neighbor points (forward, backward, up, down, right, left),
b) the 18 points also including next nearest neighbors, or
c) the 26 points also including third nearest neighbors.

We chose case c since it efficiently spans the solid angle choices, allowing channels in diagonal direction to be found. Thus, starting at point $G(l_x,l_y,l_z)$ with the previous point at $P(l_x^-, l_y^-, l_z^-)$ (where $|l_i - l_i^-| = 0,1$ but at least one is nonzero), we then calculate the probability as

$$G(l_x^+,l_y^+,l_z^+) = W(l_x^+,l_y^+,l_z^+)\left[q\cos(\underset{\sim}{l}^+ - \underset{\sim}{l})\cdot(\underset{\sim}{l}^o - \underset{\sim}{l}^-) + (1-q)\right]$$

where q can have values from 0 to 1 (we use q = 1/2).

5.0 Results: He/PE

Our model system was 4 chains of PE, each with 200C, MW=2802 Dalton, in a PBC box. Figure 7 shows the MD diffusion of 5 He atoms for 10 ns. This can be compared with the MCVD results in Figure 4, which parallels the MD simulation quite closely. The MD analysis showed three distinct regimes of diffusion

- early *ballistic* regime where the penetrant fills up the voids very rapidly
- middle *anomalous* regime where the penetrant occasionally finds channels to adjacent voids, but is still quite directional
- late *Fickian* regime where the jumps between voids are isotropic in nature

This behavior can be better seen in Figure 8A from MD and Figure 8B from MCVD. It is particularly gratifying to note that the anomalous regime and the Fickian regime exhibit similar time dependence in both simulations.

The last problem concerns converting the MCVD step to an actual time. There are 3 possibilities for doing this

1. If MD for sufficiently long times is available, then one can obtain the diffusion constant from the Fickian regime of the MSD plot. By matching this diffusion constant to that from MCVD gives the conversion from MC step to actual time. This is equivalent to matching the intercepts of the Fickian regimes.
2. One can match the crossover point from the anomalous regime to the Fickian regime.
3. One can match the intercepts of the anomalous regime from MD and MCVD.

The problem with the first method is that if MD can be done long enough to get a reliable diffusion constant, then there is not much need for a faster method. In most cases, it is not practical to do the MD for sufficiently long times. If there is a clean crossover from MD and MCVD, then one can also use the second method. The third method is the easiest to access from the MD simulations since on the ps to ns time scale, most polymer systems show the anomalous behavior. For He/PE we have sufficiently long dynamics to use all 3 methods and we show in Table 2 the MD to MCVD time constant for all 3 cases. We see that while the diffusion constant changes by a factor of 15 between the temperature extremes (1.77 @ 400K → 0.12 @ 200K) the time conversion factor remains relatively constant within a factor of 2. The time conversion is most reliable by matching the slopes of the Fickian regime, but as noted above, this time domain will not be readily accessible for most cases of interest, where either the system will be too large for sufficiently long dynamics, or it will be too rigid for showing much diffusion within the MD run. The crossover point is difficult to define unless a clear Fickian regime can be observed. The cleanest operational procedure is to use the slopes of the anomalous regime, which is easily accessed from doing short (~ 500 ps) dynamics. As the last column in Table 2 shows, the ratio of the time conversion from this regime to the Fickian regime is relatively constant. Under ideal circumstances, it would have been exactly the same at all temperatures. For the purposes of extracting an order of magnitude diffusion constant, and more importantly understanding the physical parameters controlling diffusion, it is sufficiently accurate to determine the conversion factor from the

anomalous regime. Consequently we use the third method which leads to a time conversion of 3.6fs/MCVD step.

6.0 The Discussion of Voids

6.1 Free Volume Concepts

Given the behavior of the penetrant molecule in the polymer matrix from MD, can we model this by the mobility and dynamics of the voids itself? The free volume approaches essentially treat diffusion as the redistribution of space unoccupied by the polymer matrix.[14] An alternative approach called the Transition State Theory (TST)[15] is to treat the process as a sequence of activated hopping between adjacent sorption sites. While Suter and co-workers were successful in simulating the diffusion of helium in glassy polycarbonate and melt polyisobutylene,[16] the method failed for hydrogen and larger penetrants, primarily due to the rigidity of the imposed polymer network. They did MD to extract amplitude of short time vibrational motion, but the atoms essentially were fixed in space with small amplitude motions about their mean positions. Greenfield and Theodorou[17] analyzed the geometric properties of the sorption sites and their connectivities as a prelude to a more atomistically correct input to the TST model. They observed that free volume rapidly rearranged in the melt, while they were much more rigid in the glass with occasional channel openings in between them. Our idea is based on the fact that if one can do MD for times longer than the characteristic times for void fluctuations, whether they be redistributions in the melt, or thermal fluctuations in the glassy state, then one can extract this information averaged over many static snapshots of the system and use this information to coarsen the model. The simplest way to measure fluctuations is to measure all the voids and average them. Figure 11a shows the voids in the PE as a function of MD time at 1ns intervals. Each distinct void is color coded for clarity. At 400K, in the melt the voids are redistributed rapidly and carry the penetrant molecule with it. At the lower temperature of 150K, similar analysis shows that the voids do not redistribute as rapidly but show fluctuations about their mean position, Figure 11b.

6.2 Probe Radii

At 0.0Å probe radius the free volume consists mainly of one big percolative void, whereas at larger probe radii isolated voids begin to emerge. If the probe size is

too large, then there are very few voids, which stay isolated with no channel openings and closings. Obviously, there is an ideal probe size for extracting the void information and it has to be related to the percolation properties of the system. Takeuchi showed[18] that if one measures the free volume as a function of the decreasing probe size, then at approximately the probe size that leads to 5.5% free volume, the system shows percolative channels. Greenfield and Theodorou reached similar conclusions.[17] We have analyzed the free volume as a function of probe radii, which is shown in Figure 12a. As expected, the free volume is a smoothly decreasing function of increasing probe size. 5.5% free volume occurs at approximately 0.75Å probe radius. In addition to analyzing a static snapshot, we have also analyzed a collection of snapshots from the MD and can measure what fraction of them are percolative at a specific probe radius, which is shown in Figure 12b. The change from all snapshots being percolative to none being percolative is quite sharp, between 0.6 and 1.0 Å. Another view of this is shown in Figure 12c where the fraction of percolative snapshots is plotted against the free volume available in them. Corroborating the results of Takeuchi, we find that at above 5.5% free volume, 50% of the snapshots show percolative channels.

6.3 Percolation

We have two ways of analyzing percolation. One method is to start in the largest cavity and perform the grid walk algorithm, marking each visited point and counting transitions across the unit cell boundary. If we visit a marked point a second time after a transition across a cell boundary that implies a percolative channel exists. If percolation does not happen within a large number of steps, typically 10M, we assume that the probability of a percolative channel existing is quite small. There can be very narrow channels of percolation, but these will be insignificant in the diffusion process. Now we start the percolation analysis in another cavity until all major cavities are analyzed. This algorithm is simple and very fast for porous systems with well-defined percolative channels provided the initial placement is in the right cavity. If repeated starts are required in many cavities then this method slows down substantially. It is also non-deterministic since random grid walk is performed and the maximum number of steps is also arbitrary and can affect the outcome. Although this determinism is physically insignificant as discussed above, we have devised a second algorithm, which is deterministic with a well-defined termination. In this recursive algorithm, similar to finding a path in a labyrinth, we explore all neighbors

that have not been marked as visited. If the algorithm gets stuck then it recurses back to the previous branch point and takes a different branch. If all branches from the last branch point is marked visited, then the recursion steps further back to the earlier branch point. If this leads to the initial starting point, then a new starting point is chosen from among the points that have not been marked. Using the number of unmarked points as a metric, it is easy to show that this algorithm will terminate after a fixed number of steps. This number of steps might be quite large for porous systems with large free volume. Both algorithms produce track files, which can be visualized to show the actual trajectory of the probe.

Percolative channels are one criterion by which to select the probe radius of our voids. However, as the preceding analysis shows, below a particular probe radius, most of the snapshots will show percolation. At very small probe radius, the whole polymer will be percolative and a MC dynamics will treat this essentially as one infinite cavity and only ballistic behavior will be observed. On the opposite end, with too large a probe radius, there will be no percolation at all. Since we want to include dynamical effects, such as the compressibility of the polymer, and more importantly, channel openings and closings, in our void data, percolation is not a sufficient criterion for choosing the probe size. A probe radius (say 1.3Å) may not show percolation in a static snapshot but the conjoint set of many snapshots might show percolation at this large probe radius. After several initial trials, we selected a second criterion to adopt the probe radius. Our coarsened physical model must reflect the atomistic behavior of the MD. The coarsening of the model is at the level of selecting voids and their mobility and compressibility. In the MSD plot, this is reflected in the anomalous regime, where the gas molecule finds occasional channel openings and closings between adjacent voids. The MSD from MC should mimic the MD behavior in terms of the slope of the anomalous region. This gives a narrow range of penetrant size where this physical relationship holds. For the cases reported here, 0.9Å was the probe radius chosen for mimicking the behavior of He atoms in PE.

6.4 Length of MD for Calibration of MCVD

The final questions that needs to be answered about the spatial grid is how long should the MD run be and how frequently does one need to take snapshots of the system for the purpose of extracting void information. In a polymer, bonds and angles constitute stiff degrees of fluctuations because typically the well depth for a

bond is about 70-180 kcal/mol. While a similar well depth cannot be described for an angle, the force constant (or curvature at the bottom of the well) for an angle is 70kcal/mol (force constants for bonds are about 700-1000 kcal/mol). In contrast, the torsional potentials are relatively shallow with barriers being on the order of 5-10 kcal/mol and van der Waals and hydrogen bonds also constitute softer degrees of freedom. In the case of PE, the barrier between trans and gauche conformers for the model n-butane is about 4.02 kcal/mol. Consequently, in MD, the fluctuations are significantly smaller in the bonds and angles than in the dihedrals. This is graphically shown in Figure 13, where for 1 chain of PE, for a 200ps MD run the fluctuations in selected bonds and angles are shown along with the fluctuations of all the dihedrals along this chain. While the bond fluctuates between 1.50-1.56Å and the angle fluctuates between 102-115° there are quite a few dihedrals that undergo frequent flips between gauche$^+$(60°), trans (180°) and gauche$^-$(300°) conformations.

The plot in Figure 13 shows two specific dihedrals, one of which stays constant at gauche$^+$ while the other undergoes rapid flips. While the behavior of each dihedral is dependent on a number of factors, on the average, the overall motion of chains is due to rotations of dihedrals. Rearrangement of these dihedrals is the primary cause of the void mobility because these entail motions of large segments of the polymer chain. The dihedral motion constitutes the characteristic time/frequency of the system under study. In the case of PE, we observe dihedral flips on the average of 1/10ps. This indicates that one needs to do MD for sufficiently long time to cover many such dihedral flips and we have found 200ps sufficient to sample the dynamical motion of PE. For a polymer with a more constrained dihedral motion, like polyenes, the characteristic time will be different. For this paper, we have run longer MD to also inspect the longer-term diffusive behavior from MD, but in general this will not be necessary.

The issue of how many snapshots are needed to average the properties is still an open question. Channel openings and closings have characteristic times of 10ps from Takeuchi MD simulations[19] even though he had to run 1ns MD to observe this. In our work, we have found that 40 snapshots, at 5ps intervals are optimum for our void extraction procedure. If too many snapshots are averaged, then for the melt, almost all of the space will likely be a void at some instant in time. Too few snapshots will not span the dynamical nature of the polymer matrix adequately. However, this question is not settled and we are continuing to investigate it.

7.0 Discussion

The replacement of MD on gases diffusing through a real polymer with MCVD on particles migrating through negating voids (obtained from the MD) corresponds to a coarse gaining of the gas-phase system. In this process the details in the atomistic description of the voids is replaced by the void probability distribution. This is analogous to going from the quantum mechanical wavefunction and its detailed information on electrons and nucleus to the more averaged molecular mechanics FF comprising of balls and strings. In the case of diffusion, we are going from the atomistic MD to a procedure based on the mesoscopic length and time scale by averaging the voids and their mobility information. The anomalous regime in the MD is precisely this time and length scale, and thus it is reasonable to match the slopes for this regime.

To be useful the MD to MCVD time conversion should be the same for different temperatures. In order to check, we carried out MD on PE at various temperatures from 400-200K in steps of 50K. Table 3 lists the diffusion constants we get from the slope of the Fickian part of the msd plot. We then performed MCVD on the set of voids derived from the MD at various temperatures. Using the time conversion factor reported above (derived at 400K), we then derived the MCVD diffusion constants over the full range of temperatures for the MD simulations. The normalized diffusion constants from both MD and MCVD are plotted in Figure 14. The agreement between the MD and MCVD values is within 33%.

Thus, MCVD affords a remarkably simple method to convert short time (ns) MD trajectories to a mesoscopic simulation (μsec) using only the voids to simulate small molecule diffusion in a polymer matrix. This should be useful above and below the glass transition temperature T_g.

Acknowledgements

We thank Asahi Chemical for allowing one of the co-authors (SK) to have an extended stay at the MSC. This work was supported by DOE-AICD. The facilities of the MSC are supported by grants from NSF-GCAG (ACS-92-17368), NSF-Chemistry (91-00284), DOE-BES, Allied-Signal, Asahi Chemical, Asahi Glass, BP America, Chevron, BF Goodrich, Gore Corp., and Xerox. The MSC KSR-1 and KSR-2 machines were purchased with money from the NSF-GCAG grant.

Figure Captions

Figure 1. (a) MSD from 1.5 ns MD of 5 He atoms in PE; (b) MSD from 1.5 ns of MD of 5 O_2 molecules in a PVC/PVDC copolymer.

Figure 2. (a) trajectories for 1.5 ns MD of 5 He atoms in PE corresponding to the MSD of Figure 1a; (b) trajectories for 1.5 ns of MD of 5 O_2 molecules in a PVC/PVDC copolymer corresponding to the MSD of Figure 1b.

Figure 3. The dynamical void distributions in the two polymers of Figures 1 and 2. The unit cell is partitioned into one million cells (100×100×100) and at intervals of 5 ps in the MD trajectory, a probe of 1Å radius is used to analyze the polymer structure. The color code indicates the fraction of the times that there was a void at that site.

Figure 4. Monte Carlo Void Diffusion (MCVD) trajectory for He in PE.

Figure 5. Typical amorphous system used in this work, consisting of 4 chains of PE with 100 monomers (200 backbone C atoms) per chain. For purposes of clarity, the H atoms have been removed and the various chains are color coded to follow the chain conformation and entanglements.

Figure 6. Track of 5 He atoms in PE for a 10 ns MD.

Figure 7. MSD from 10 ns MD for 5 He atoms in PE.

Figure 8. (a) MSD from 95 ps of MD of 5 He atoms in PE with trajectories saved at each 1 fs interval; this plot clearly shows the three different regimes of behavior – early *ballistic,* middle *anomalous,* late *Fickian* (b) MSD from MCVD at finer resolution, showing three regimes similar to the MSD from MD.

Figure 9. Void analysis of 4 chain PE for various probe sizes ranging from 0.1 to 1.2Å.

Figure 10. MD trajectory of a single He atom in PE showing smooth behavior of the particle within a feliciton.

Figure 11. (a) Voids in PE at several time steps in the 400K MD showing rapid redistribution of the voids in the melt; (b) Same as in (a), but at 150K, below the T_g= 220K showing slow redistribution in the glassy state.

Figure 12. (a) Free volume as a function of probe radii; (b) fraction of percolative snapshots as a function of probe radius; (c) fraction of percolative snapshots as a function of free volume.

Figure 13. Fluctuations in two bonds, two angles, all torsions for 1 chain and two selected torsions of PE from a 200ps MD run. The bond fluctuates between 1.50-1.56 Å and the angle fluctuates between 102-115°.

Figure 14. Normalized diffusion constants plotted against inverse temperature from MD and MCVD.

Table 1. Void analysis.

#	Size	Area	Asphericity	Radius of gyration	Centroid		
					x	y	z
1	67674	62312	7.76	298.3	14.7828	14.2896	12.8502
2	1488	1691	2.68	43.20	22.2629	10.2332	15.1200
3	1095	1113	2.17	36.24	5.0075	22.3579	9.1546
4	541	764	2.38	11.40	6.0731	22.7761	16.6048
5	540	445	1.39	6.60	20.3239	14.1350	19.5393
6	253	307	1.59	5.62	14.8523	16.8209	23.7709
7	205	319	1.90	7.31	9.7171	9.4090	8.2283
8	169	220	1.49	4.79	9.4088	20.0497	13.5401
9	169	235	1.59	43.35	1.5331	4.1198	13.1631
10	160	220	1.54	4.75	10.5145	9.8926	20.6866

Table 2. Time conversion from different regimes.

Temperature	Diffusion Constant[a]	Time conversion (fs/MCVD step)			Ratio
		Crossover	Anomalous	Fickian	$t_{anomalous}/t_{Fickian}$
400	1.77	4.23	8.03	8.47	0.95
350	1.30	10.00	10.00	9.23	1.08
300	0.60	8.91	8.91	5.83	1.53
250	0.20	9.97	9.97	6.00	1.66
200	0.12	7.84	7.84	8.33	0.94

[a]In units of $Å^2$/ps – the He atom displacements were recorded at each 1 ps interval from the saved trajectory.

Table 3. Diffusion constants for He in PE from MD and MCVD. The diffusion constants are obtained from the slopes of the Fickian regime of the respective MSD plots and have different units.

Temperature	Molecular Dynamics		Monte Carlo Void Dynamics	
	Absolute	Normalized	Normalized	Absolute
400	1.77	1.00	1.00	15.00
350	1.30	0.73	0.80	12.00
300	0.60	0.34	0.23	3.50
250	0.20	0.11	0.08	1.20
200	0.12	0.07	0.07	1.00

[1] Schlotter N. E. and Furlam P. Y., *Polymer*, **33**, 3323(1992); Wessling M., Boomgaard T. v. d., Mulder M. H. V. and Smolders C. A., *Makromol. Chem., Macromol. Symp.*, **70/71**, 379(1993)

[2] Koros W. J., ed. *Barrier Polymers and Structures, ACS Symp. Ser.* , 423(1990); Mohr, J. M., and Paul D. R., *J. Appl. Polym. Sci.*, **42**, 1711(1991)

[3] Gusev A.A., Arizzi S., Suter U.W., *J.Chem.Phys.*, **99**(3), 2221(1993);

Gusev A. A. and Suter U.W., *J.Chem.Phys.*,**99**,2228-2234(1993)

[4] Crank J., *The Mathematics of Diffusion*, Oxford, Clarendon 1989.

[5] Using the Amorphous Builder Module in Cerius2 from Molecular Simulations Inc. (San Diego, Calif.).

[6] Flory P. J., Jackson C. J. and Wood J., *Statistical Mechanics of Chain Molecules* (New York; Oxford University Press, 1989), 49-94

[7]Lim K.T., Brunett S., Iotov M., McClurg R., Vaidehi N., Dasgupta S., Taylor S. and Goddard III W.A., *Journal of Computational Chemistry*, **18**(4) 501-521 (1997)

[8] Mohr J. M. and Paul D. R., *J. Appl. Polym. Sci.*, **42**, 1711(1991)

[9] This gives a sorption density higher than experiment. The actual solubility of He in PE is 0.054×10^{-6} $(cm^3/cm^3)^x$ which converts to 3.18×10^{-8} molecules in our simulation volume of 21835 $Å^3$.

[10] Karasawa N., Dasgupta S. and Goddard III W. A., *J. Phys. Chem.*, **95**, 2260(1991). The MSXX FF is used for accurate intrachain vibrations and not required for these applications. Thus, we used the MS FF but avoid the more complete description of the torsions.

[11] Lacks D. J. and Rutledge G. C., *J. Phys. Chem.*, **98**, 1222(1994)

[12] Gao G., Goddard III,W. A., manuscript in preparation.

[13] *Polymer Handbook*, New York, Wiley, 1989

[14] Vrentas J. and Duda J., *J. Polym. Sci., Polym. Phys. Ed.*, **15**, 403(1977); ibid, 417; ibid, 441

[15] Pace R. and Datyner A., *J. Polym. Sci., Polym. Phys. Ed.*, **17**, 437(1979); ibid, 453; ibid, 465

[16] See reference 3.

[17] Greenfield M. L. Theodorou D. N., *Macromol.*, **26**, 5461 (1993)

[18] Takeuchi H, Okazaki K., Makromol.Chem.,Macromol.Symp., **65**, 81-88 (1993)

[19] Takeuchi H., *J. Chem. Phys.*, **93**, 2062 (1990); see also the previous reference.

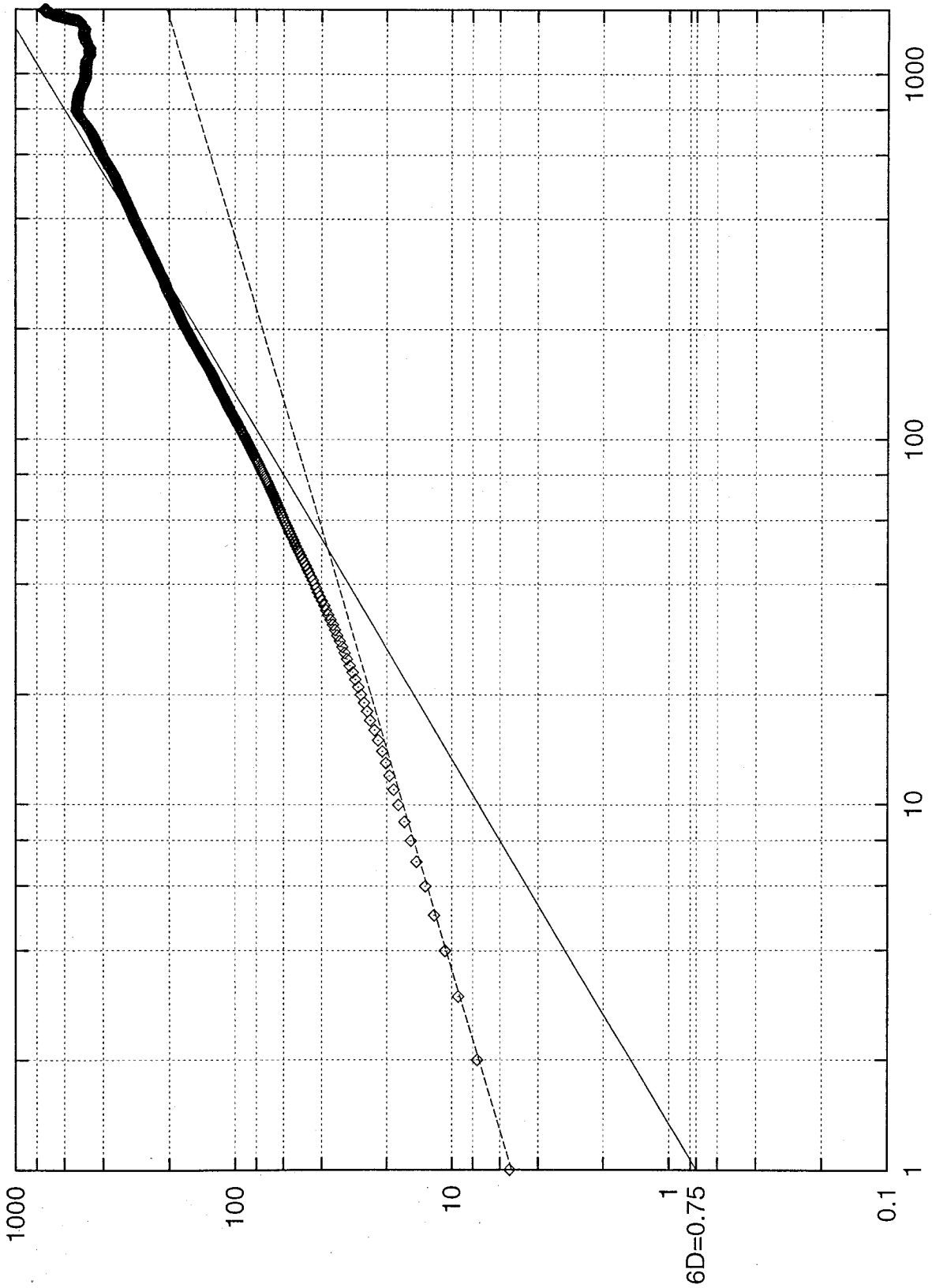


Figure 1. (a) MSD from 1.5 ns MD of 5 He atoms in PE. Units: x-axis – time in ps; y-axis MSD in A^2.

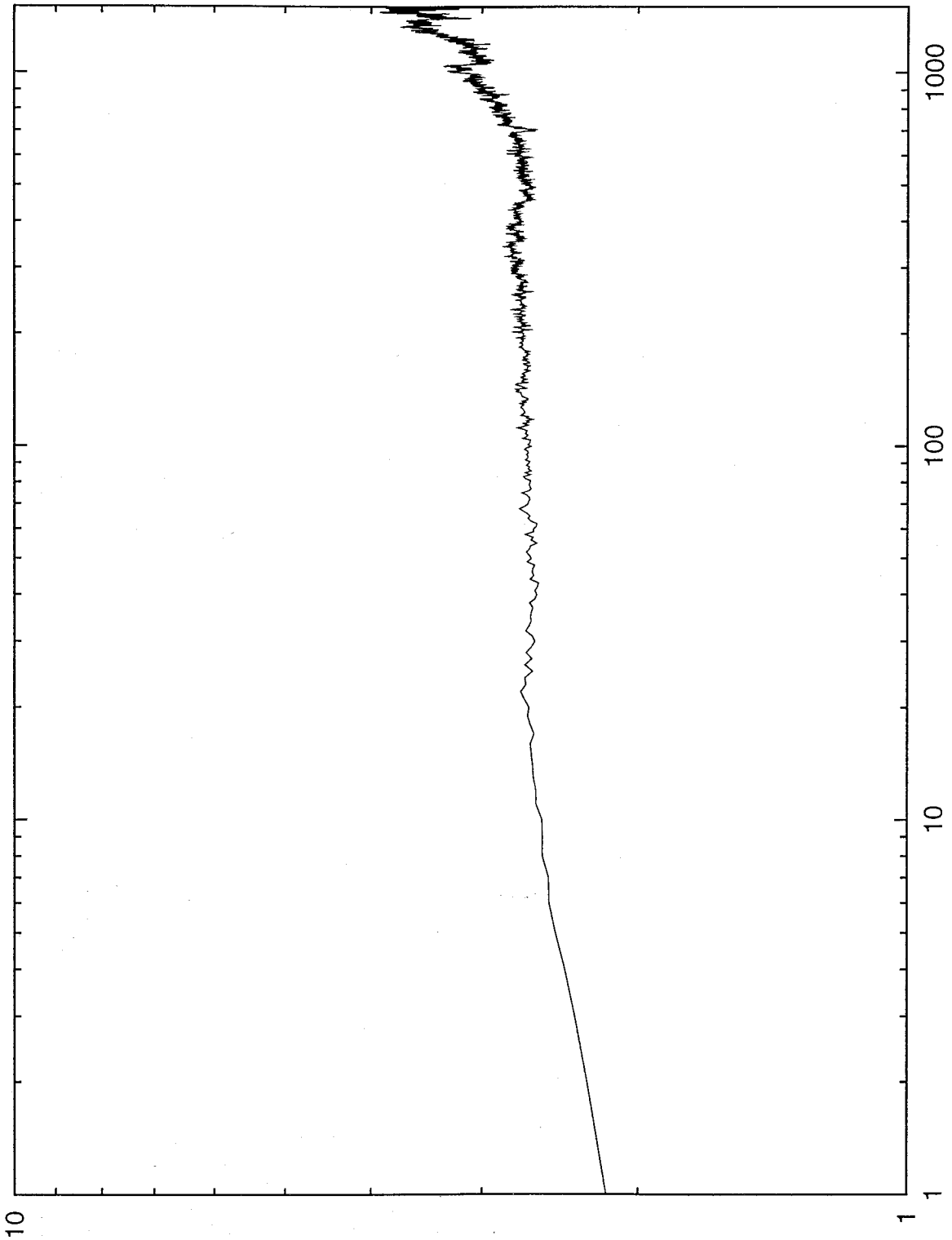


Figure 1. (b) MSD from 1.5 ns of MD of 5 O_2 molecules in a PVC/PVDC copolymer

Units: x-axis – time in ps; y-axis MSD in A^2.

Figure 2. (a) Trajectories for 1.5 ns MD of 5 He atoms in PE corresponding to the MSD of Figure 1a.

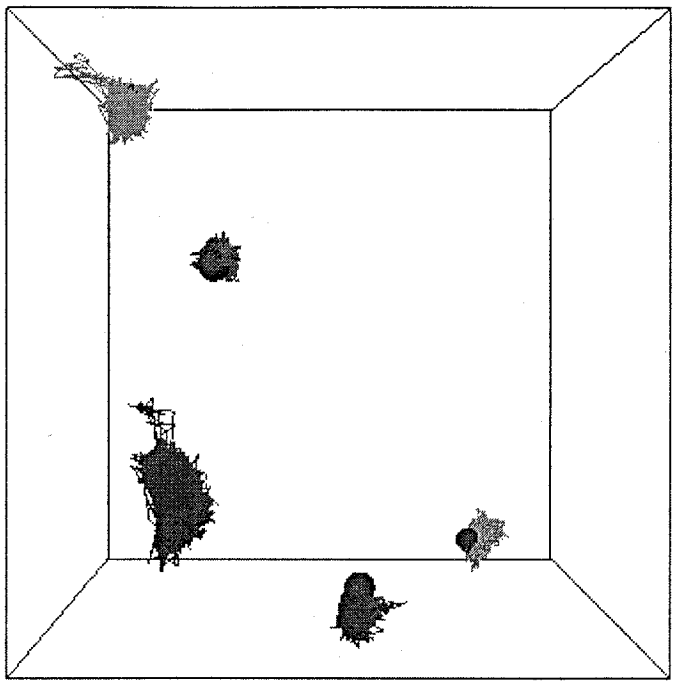
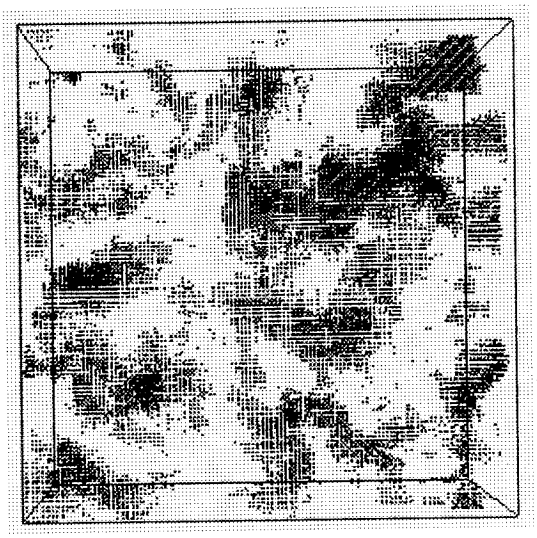


Figure 2. (b) trajectories for 1.5 ns of MD of 5 O_2 molecules in a PVC/PVDC copolymer corresponding to the MSD of Figure 1b.



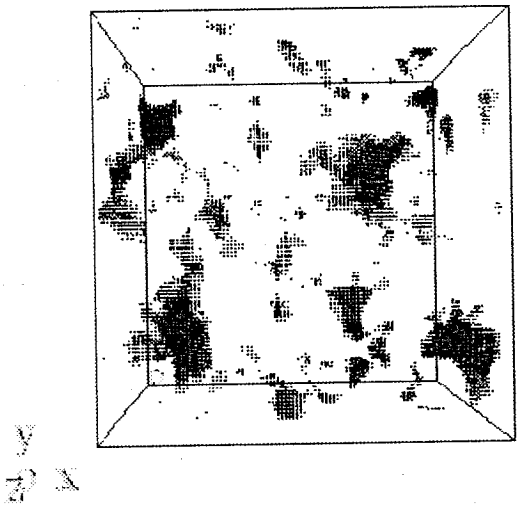


Figure 3. The dynamical void distributions in the two polymers of Figures 1 and 2. The unit cell is partitioned into one million cells (100×100×100) and at intervals of 5 ps in the MD trajectory, a probe of 1Å radius is used to analyze the polymer structure. The color code indicates the fraction of the times that there was a void at that site.

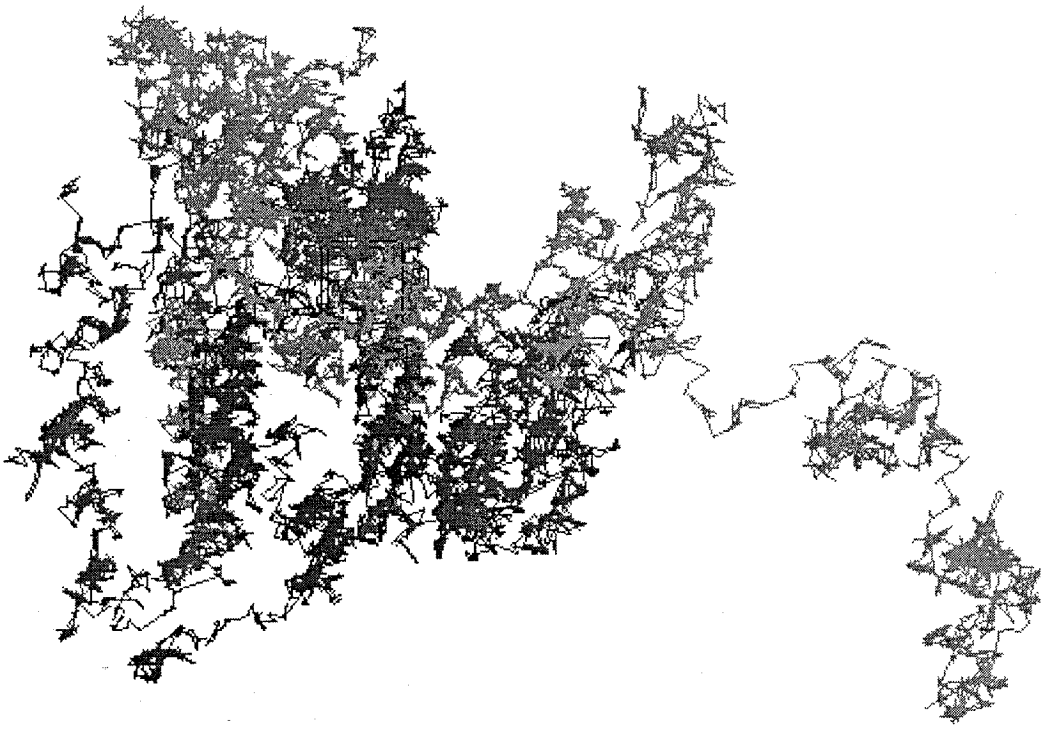


Figure 4. Monte Carlo Void Diffusion (MCVD) trajectory for He in PE.

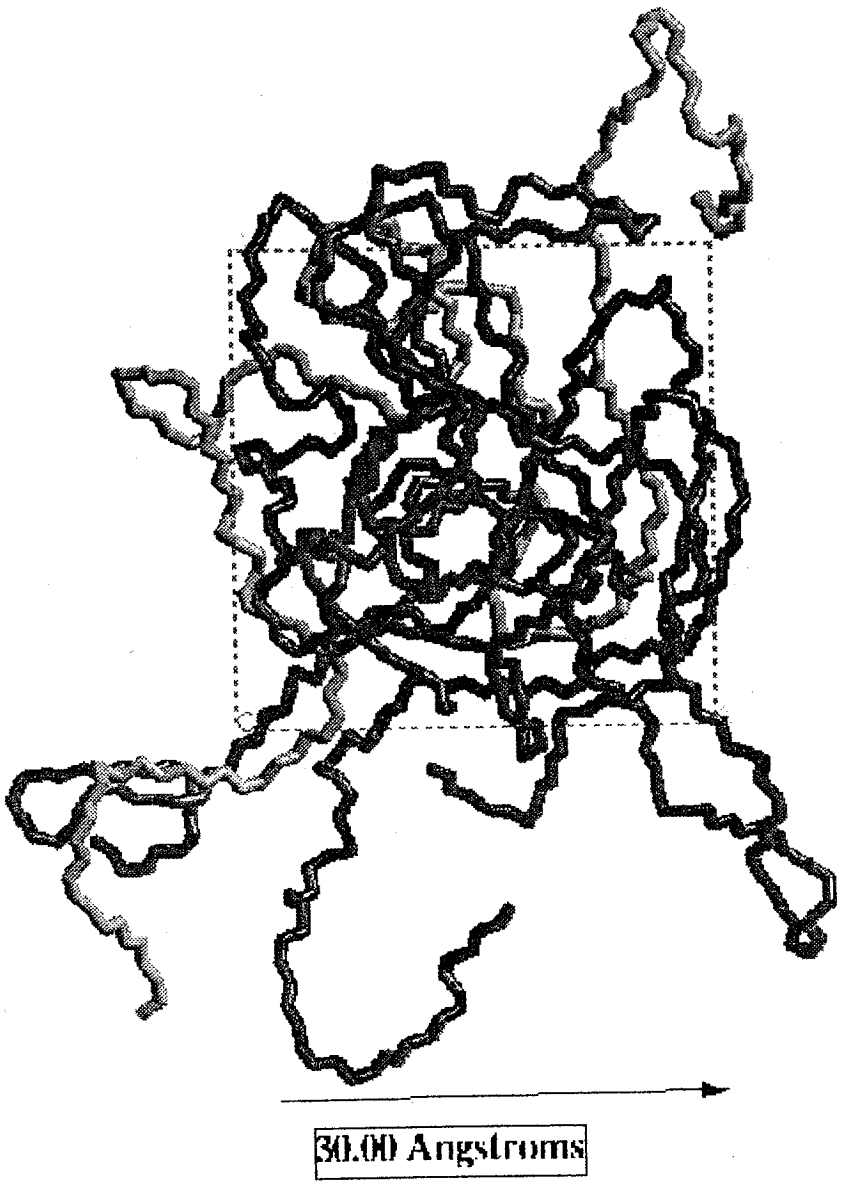


Figure 5. Typical amorphous system used in this work, consisting of 4 chains of PE with 100 monomers (200 backbone C atoms) per chain. For purposes of clarity, the H atoms have been removed and the various chains are color coded to follow the chain conformation and entanglements.

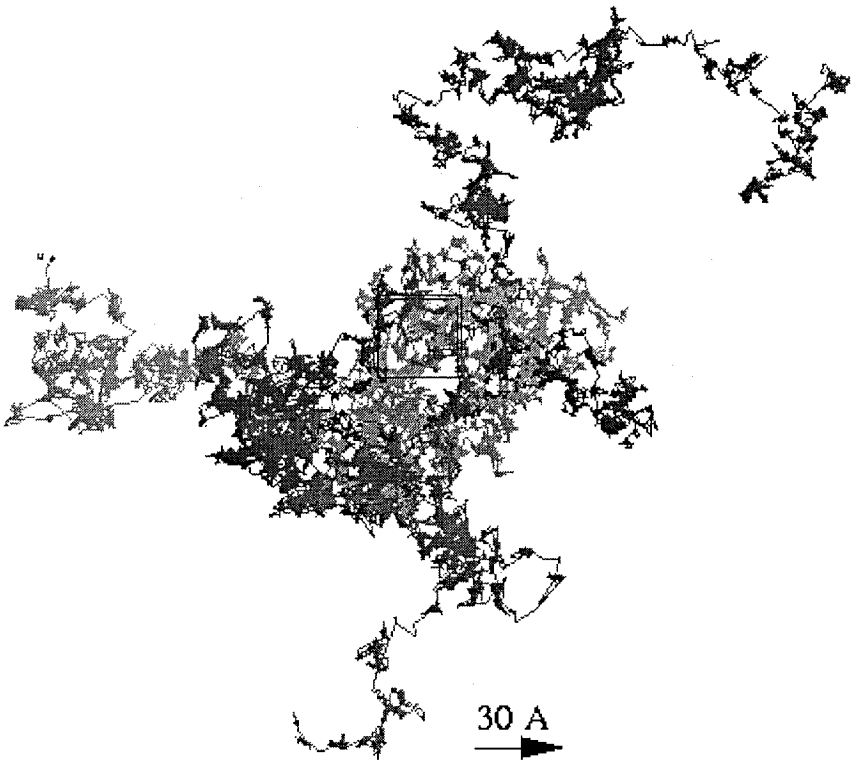


Figure 6. Track of 5 He atoms in PE for a 10 ns MD.

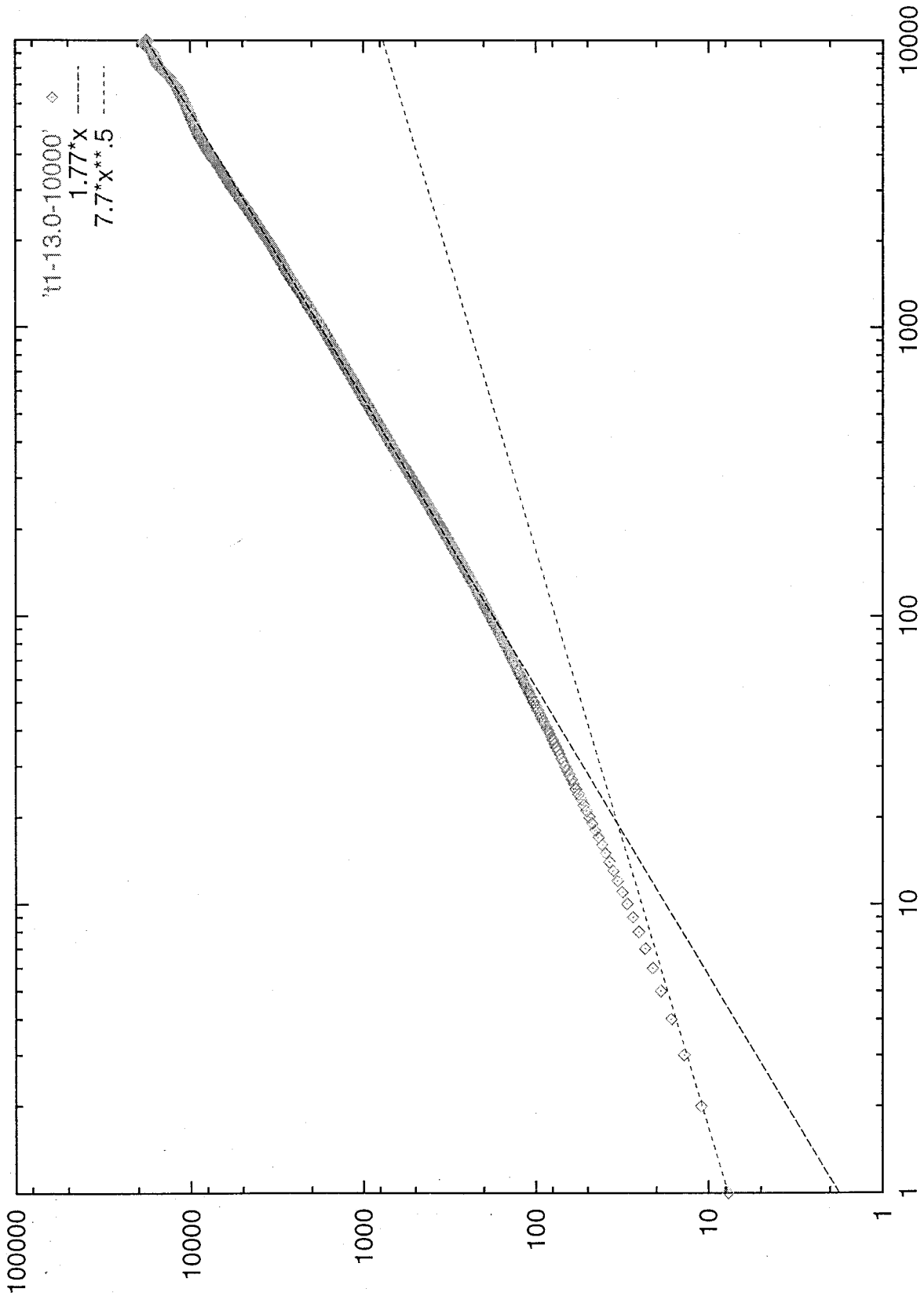


Figure 7. MSD from 10 ns MD for 5 He atoms in PE. Units: x-axis – time in ps; y-axis MSD in A^2.

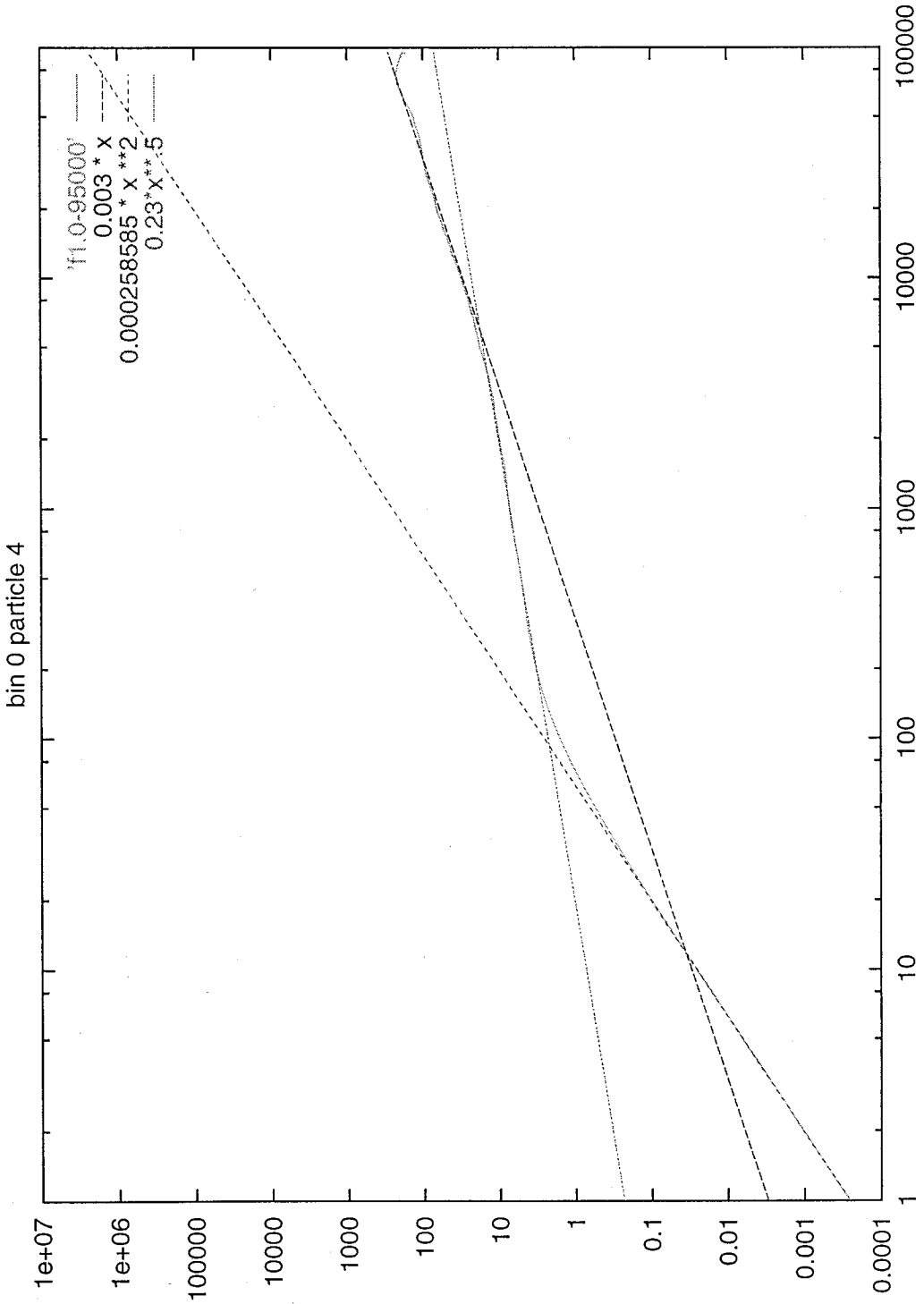


Figure 8. (a) MSD from 95 ps of MD of 5 He atoms in PE with trajectories saved at each 1 fs interval; this plot clearly shows the three different regimes of behavior – early *ballistic,* middle *anomalous,* late *Fickian*.

Units: x-axis – time in ps; y-axis MSD in A^2.

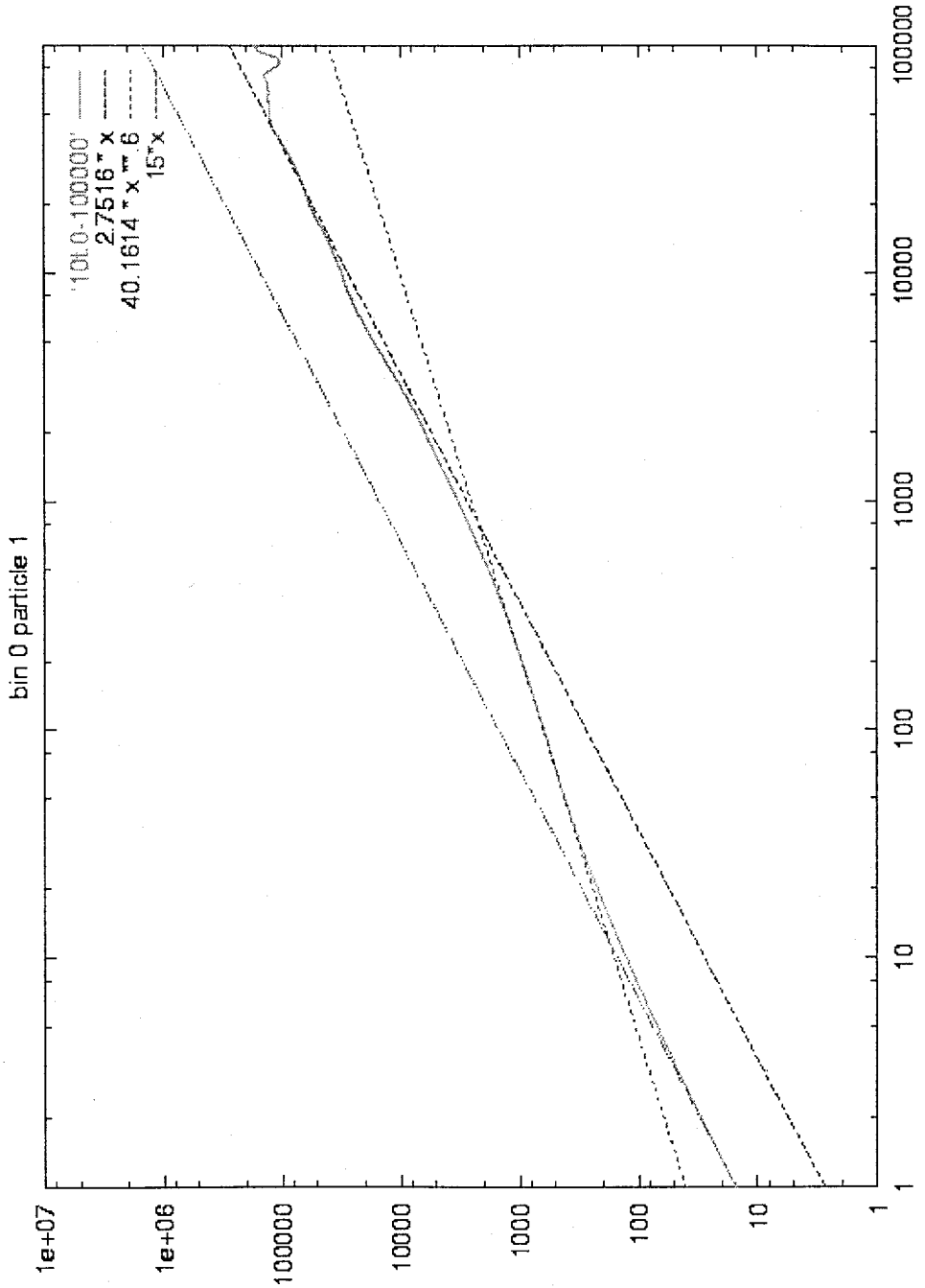


Figure 8. (b) MSD from MCVD at finer resolution, showing three regimes similar to the MSD from MD.Units: x-axis in MCVD steps, y-axis in A^2.

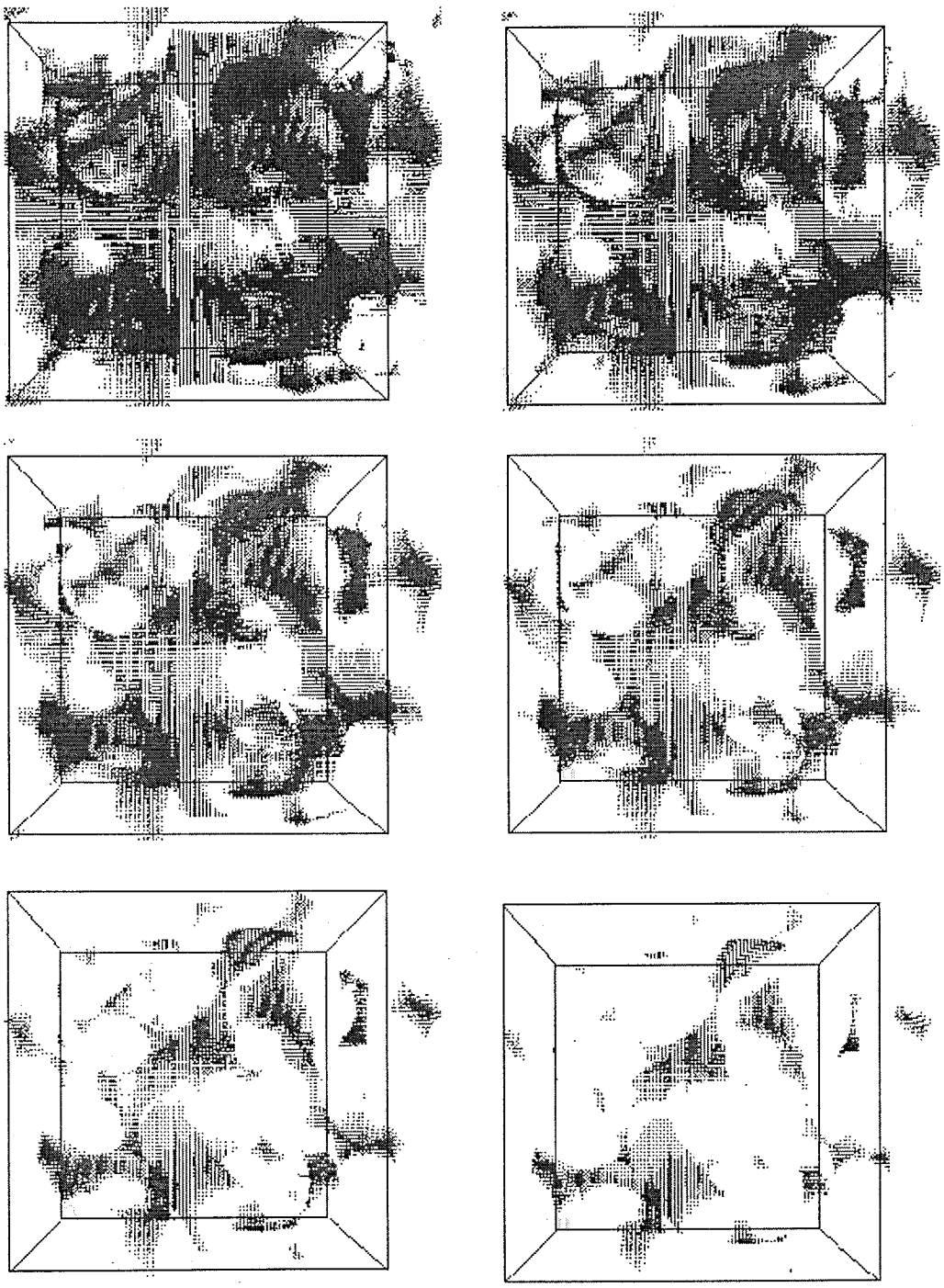


Figure 9. Void analysis of 4 chain PE for probe sizes (from top left, first right, then down) 0.3, 0.5, 0.7, 0. 9, 1.1, 1.3 A.

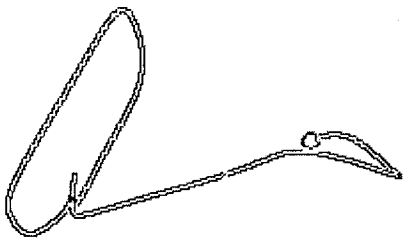


Figure 10. MD trajectory of a single He atom in PE showing smooth behavior of the particle within a feliciton.

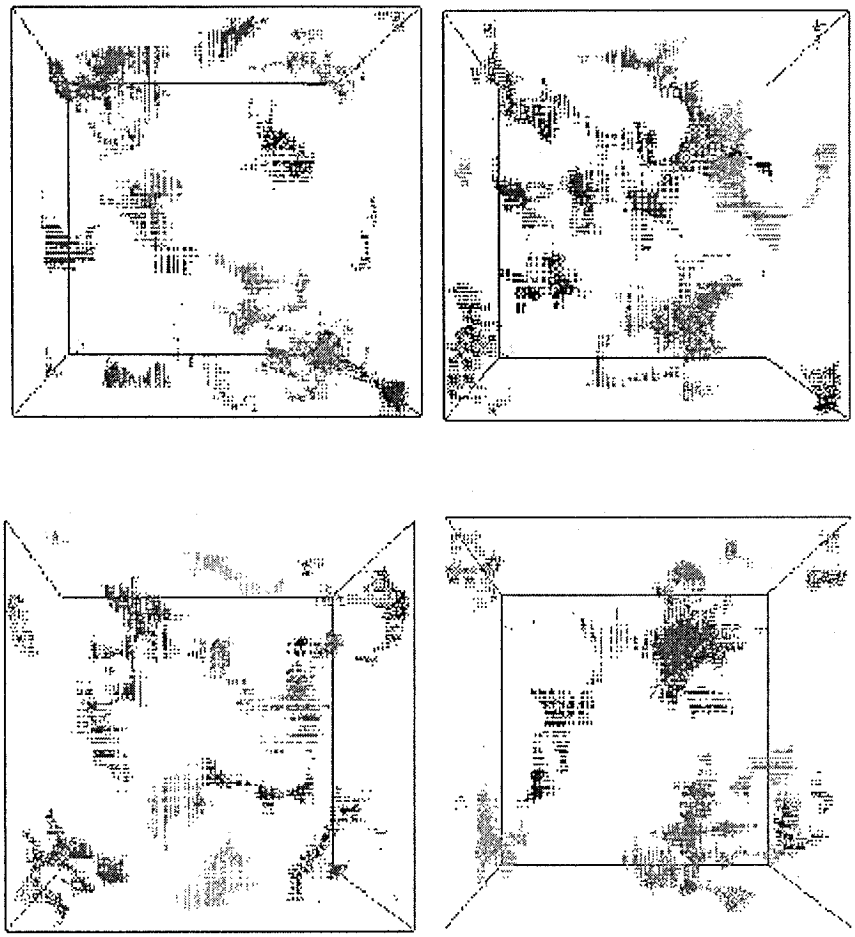


Figure 11. (a) Voids in PE at several time steps in the 400K MD showing rapid redistribution of the voids in the melt.

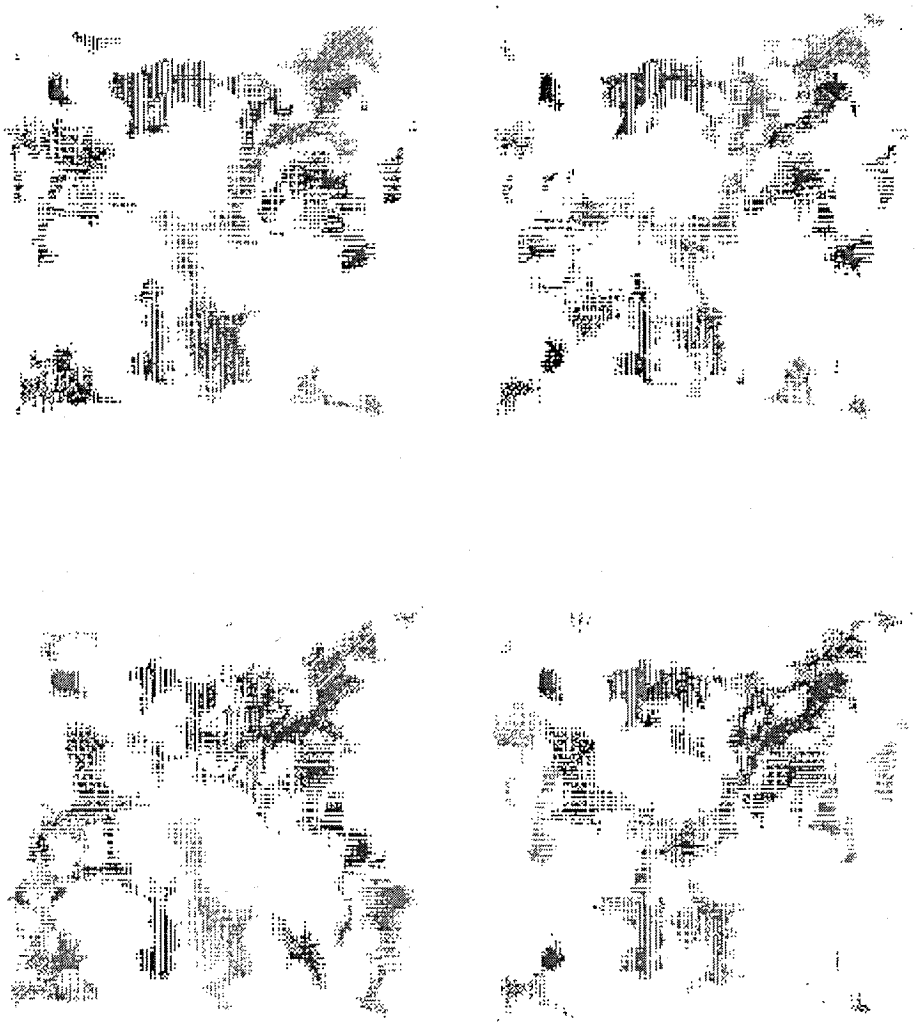


Figure 11. (b) Same as in (a), but at 150K, below the T_g= 220K showing slow redistribution in the glassy state.

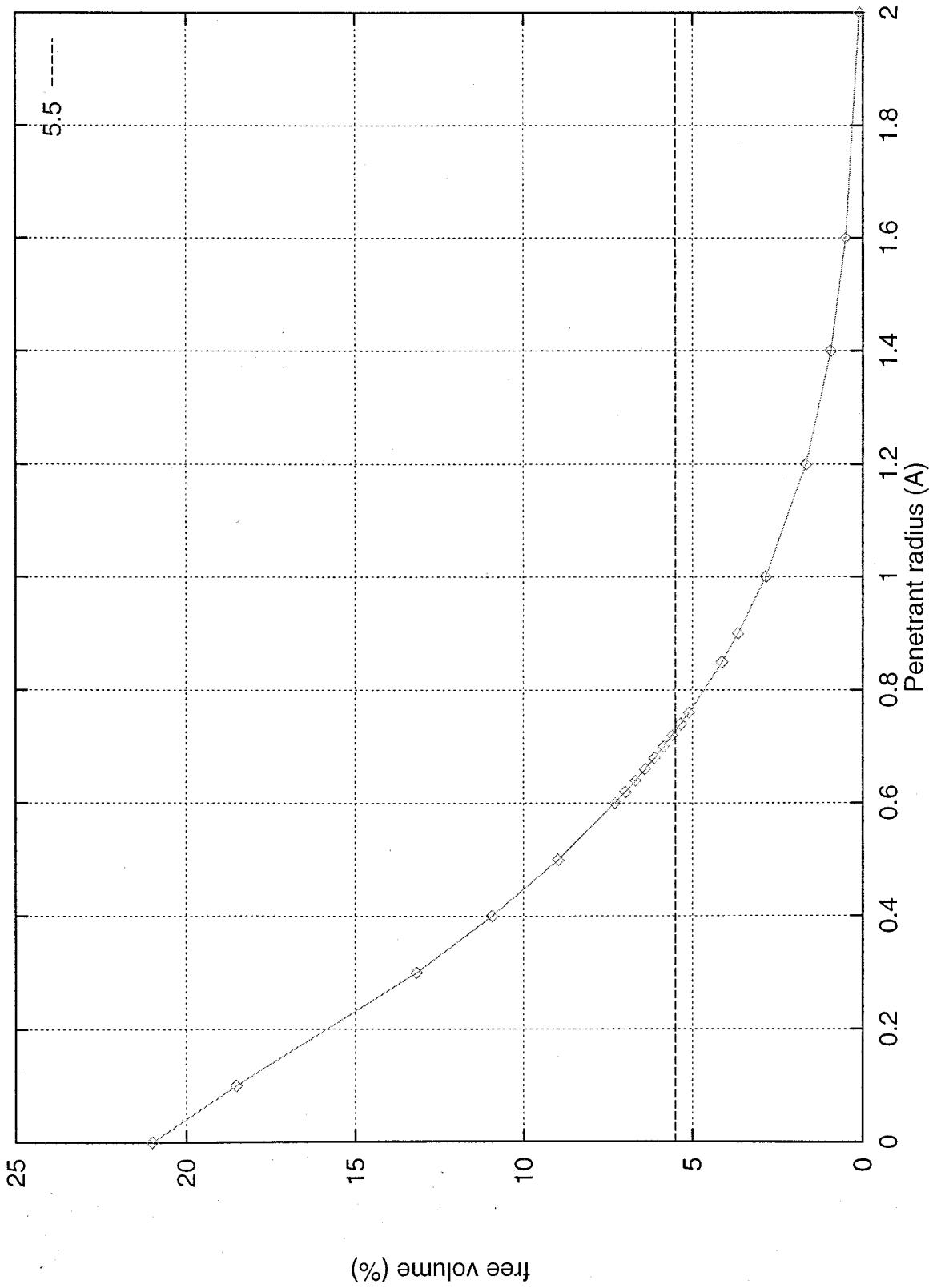


Figure 12. (a) Free volume as a function of probe radii.

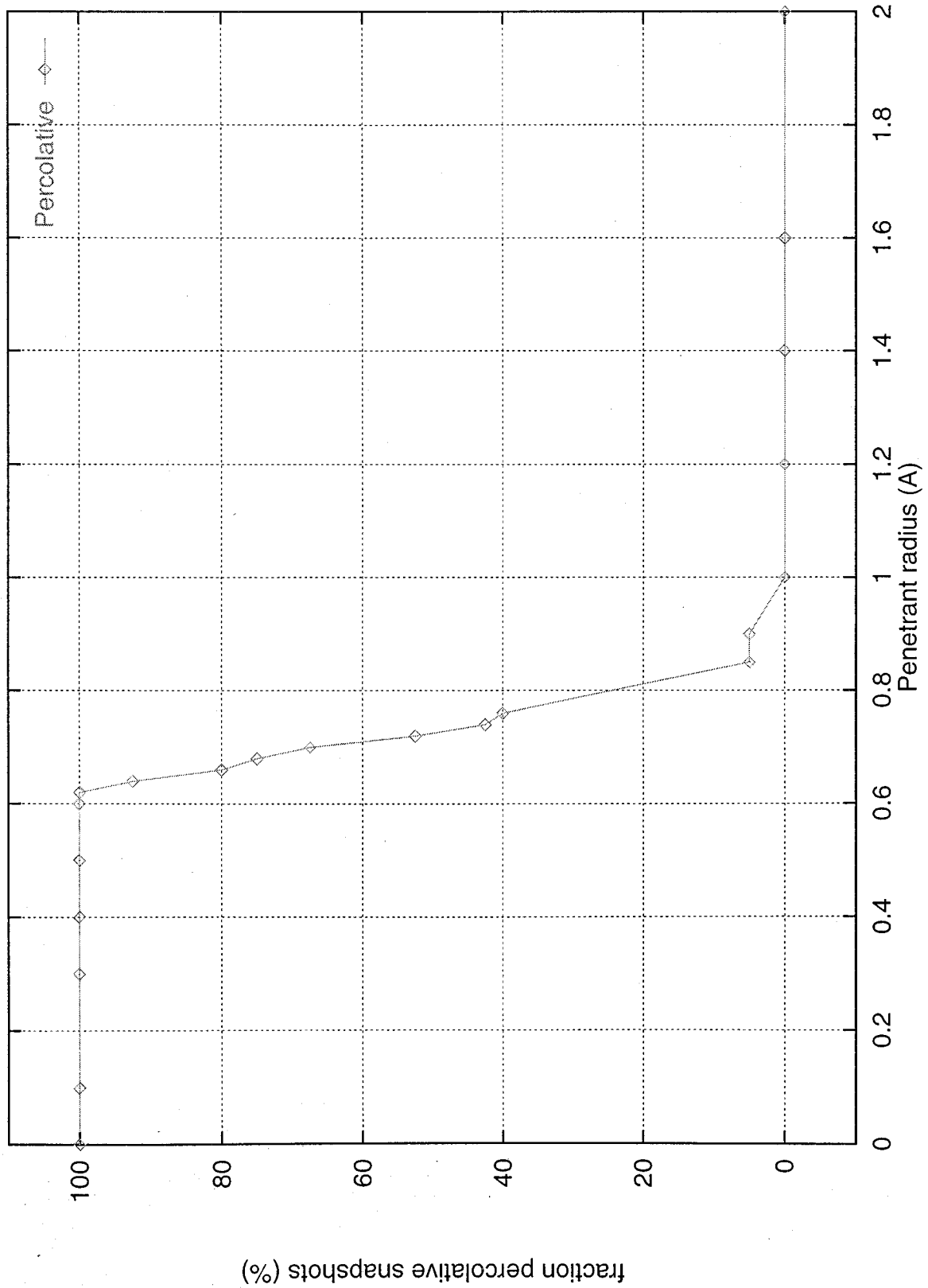


Figure 12. (b) Fraction of percolative snapshots as a function of probe radius.

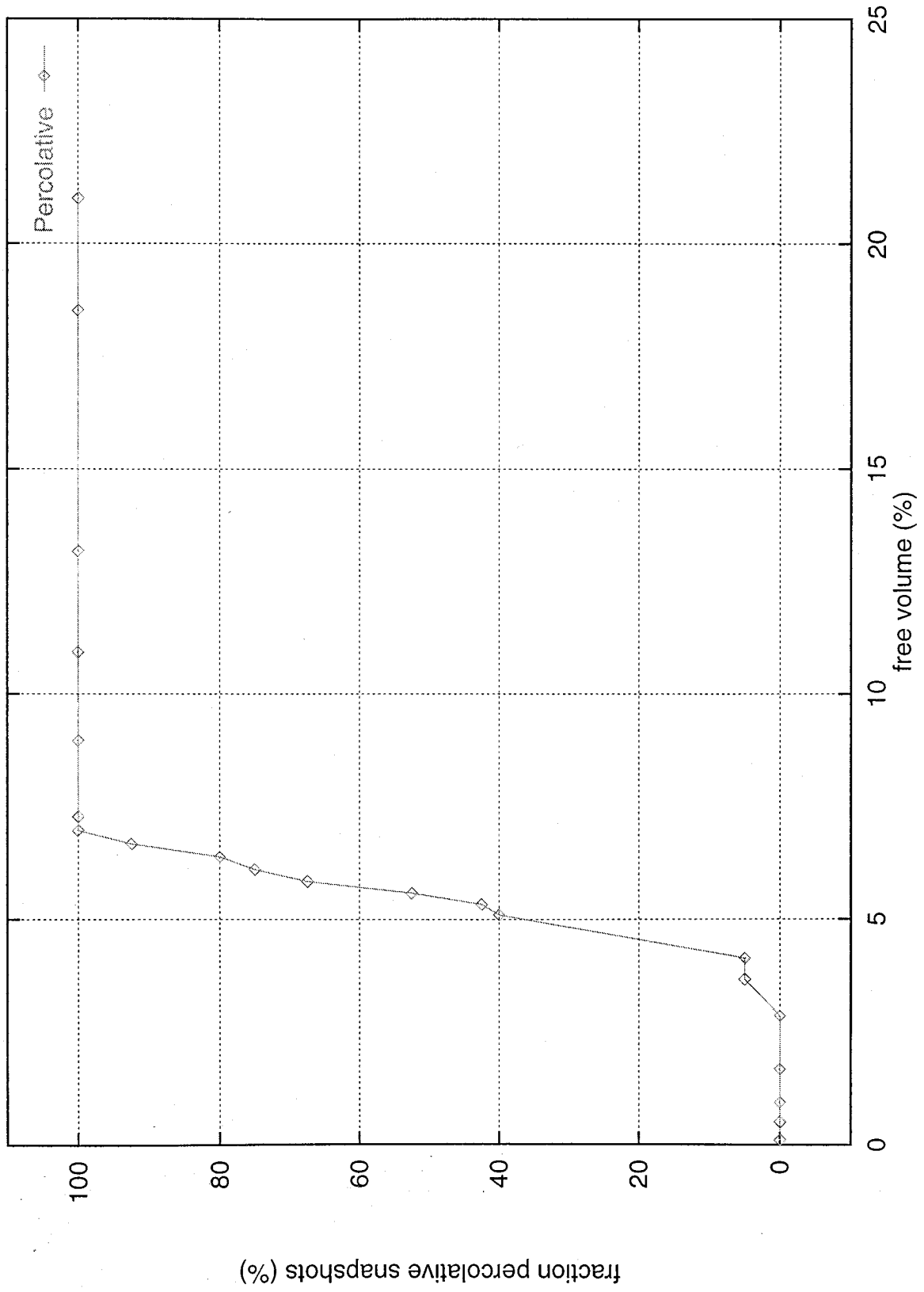


Figure 12. (c) Fraction of percolative snapshots as a function of free volume.

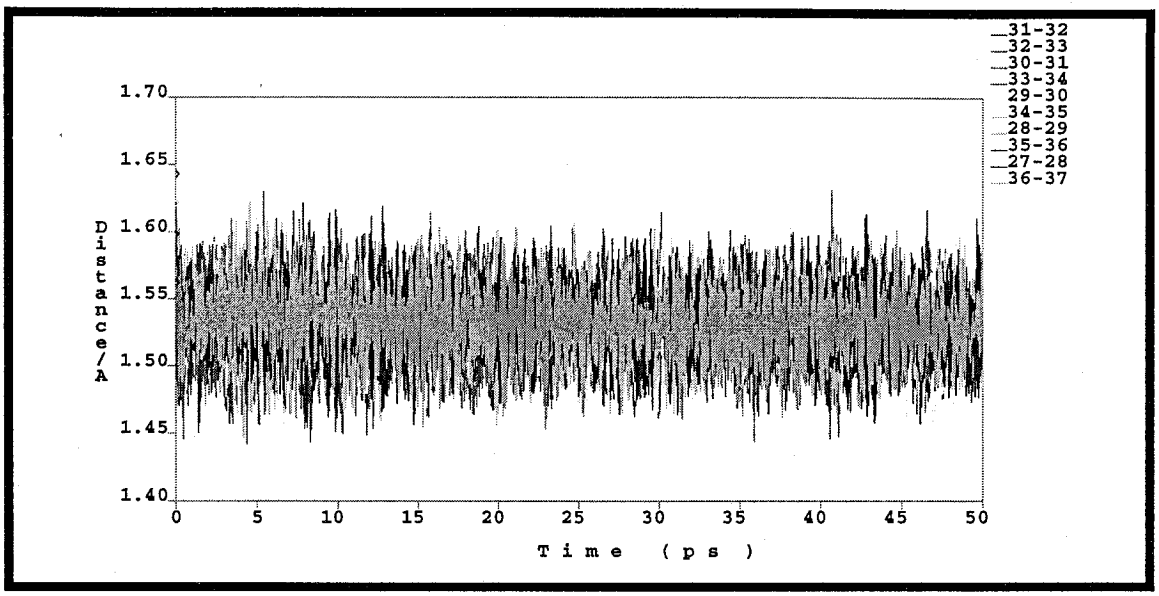


Figure 13 (a)

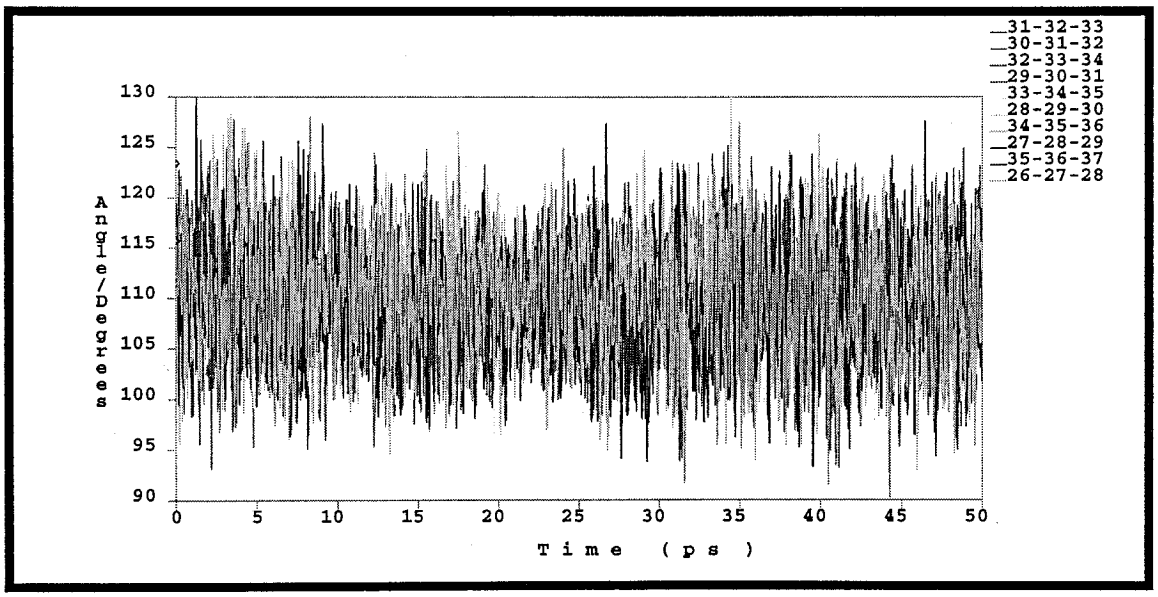


Figure 13 (b)

Figure 13. (a) (b) Fluctuations in two bonds, two angles, all torsions for 1 chain and two selected torsions of PE from a 200ps MD run. The bond fluctuates between 1.50-1.56 Å and the angle fluctuates between 102-115°.

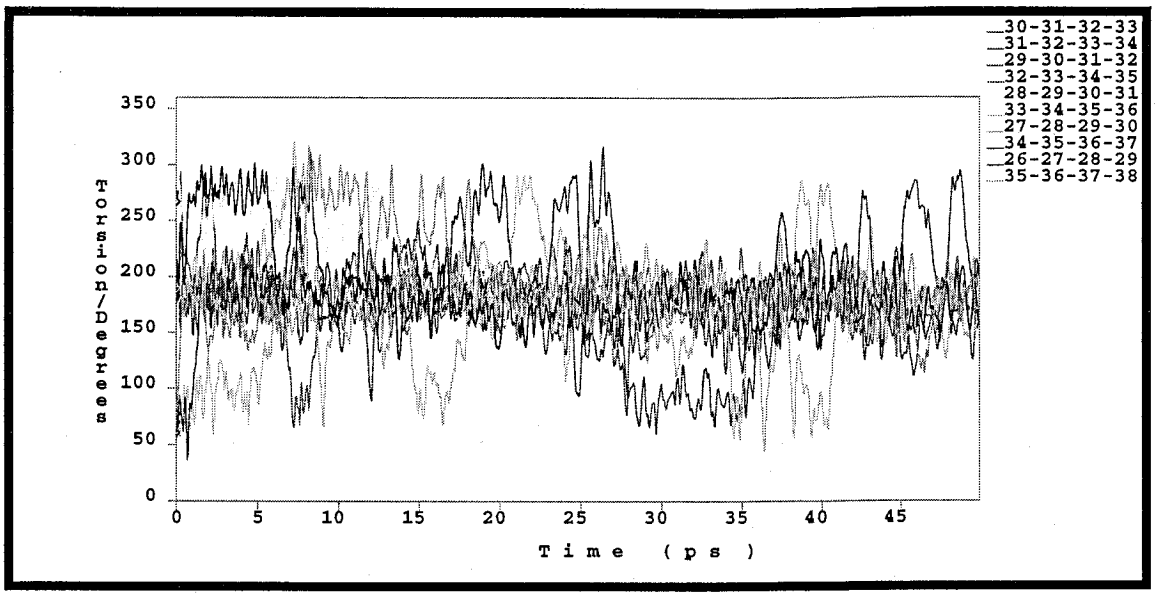


Figure 13 (c)

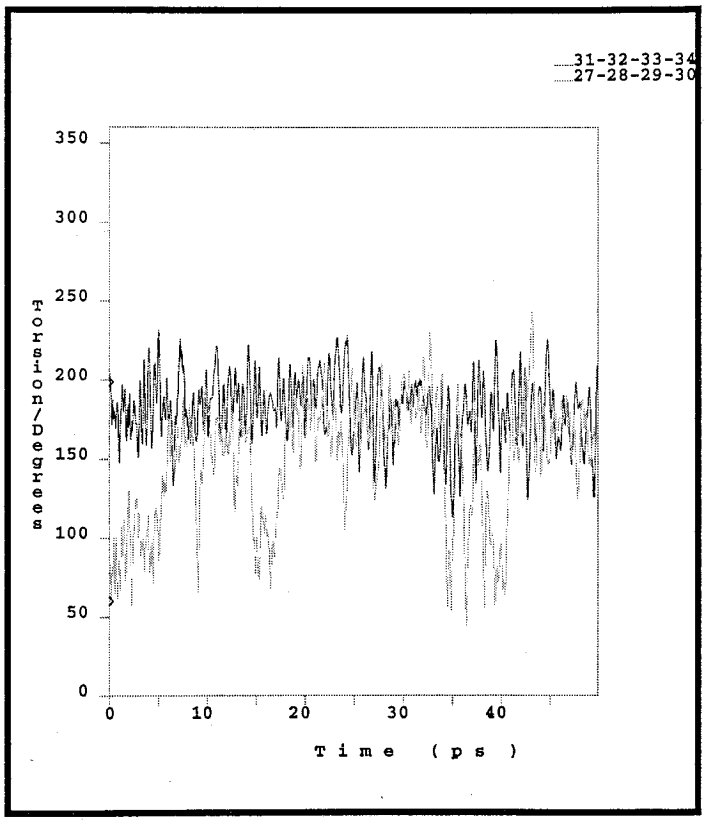


Figure 13 (d)

Figure 13. (c) (d) Fluctuations in two bonds, two angles, all torsions for 1 chain and two selected torsions of PE from a 200ps MD run. The bond fluctuates between 1.50-1.56 Å and the angle fluctuates between 102-115°.

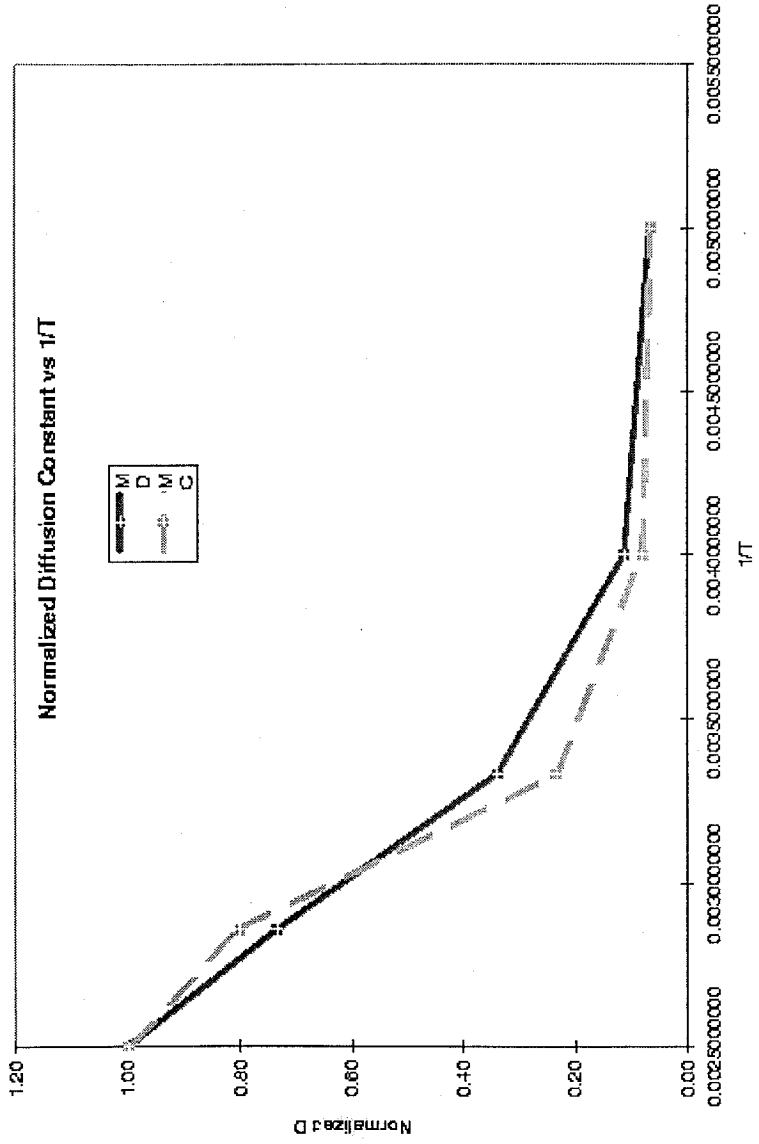


Figure 14. Normalized diffusion constants plotted against inverse temperature from MD and MCVD.

Appendix A Message-passing Algorithms

A.1 Paradigms

There are many different message types used, but they fall in roughly three categories:

Global sums, data updates and flow control.

Global sum:

Each node computes local value and sends it to node 0 for global summation.

Data update:

Each node sends a request of which data it needs from another node, then receives the data.

Flow control:

Node 0 collects done messages from all nodes, then broadcasts "go-on" message to all of them.

Note: in this way it is possible to do without barrier synchronization calls, since the J machine did not have them. In effect we implement our own barriers.

A.2 Pseudo-code

```
Main
        Read_ctl
        Setup_ee
        Master_cmm
}

master_cmm
        if not nbs setup calls dyn()
else proceeds to update multipoles.

Compute multipoles

Dyn() {

        Self_far_nearinfl;
```

Roundup_near_atom;
Unpack_near_atoms;

Calculate exclusions, torsion... no commucnication

Send_partial_forces home

Integrate1;
Compute KE
Integrate 2;

Handle_moving;

Compute farfields: is:

Calc_center_for_all cells;
Checkwindow cald, which :
Combine_calc_center for own
Send message CALC type for others.

At end broadcasts all_done_calc_type

Proceeds to :
Process CLACL_TYPE, via combine_calc_center
ALLDONECLACTYPE, by conter., via which exits.

Check_window child _pnc
Compute multipoles or call child if can,
Send CHILD_TYPE if not own
Then do update taylor of PNC_M-TYPE if not own
At end broadcasts ALL_DONE_CALC_TYPE

Case:
-child
-master_pnc
--done by counter.

Check_window parent
Combine taylor if own
Send parent type otherwise
ALL_DONE_CALC at end is broadcast
Process PARENT via parent
ALL DONE CALC via counter.
At end sends DONE CMM to 0
0 counts them and broadcasts ALL_DONE_CMM

AT ALL DONE CMM, dyn is called.

Self_far_nearinfl:

Sends DONE_CMM

Processing loop (while done):
ATOMS_M: master_atoms
DONE_CMM_TYPE;
Farinfl;selfinfl – no communicatino in those
And sends ack_near_type to start nearinfl processing.
DONE_SELF_FAR_TYPE:
Counter, at couner finished broadcast MOVE_ON_TYPE

MOVE_ON_TYPE:sets done flasg
Ack_NEAR_TYPE:check_window_ack_newar

At exit sends DONE_SELF_FAR_TYPE

Check_window_ack_near

At exit sends DONE_SELF_FAR_TYPE

Otheiwise does atoms if own
Or sends ATOMS_M_TYPE if nonlocal

Roundup_near_atoms

```
init_DestCounter(void) / * zeroed DesBuf and DestCOuer.
get_near_atoms(SEND_FORCES_TYPE,MOVE_ON_TYPE);
Counter = Numnodes-1
while (Counter) {
        crecv(SEND_FORCES_TYPE,
        expects only SEND FORCES TYPE messages and exactly one from a node.
memcpy(CellBuf[infonode()],NearMsg,(int)(sizeof(NEAR)*Size));
  dumps them directly into CellBuf with offset of the one that are received.
}
get_near_atoms(int send_type,int done_type)
        for all leaf cells on this CPU (iterated through leafhead and  ->lnext
                iterate thgour 27 neighbours.
                        If neighbour does not reside on our CPU
                                add_cell_to_buf(n,who_i_am,Dest), which does:
                                        DestCounter[dest]++;
                                        DestBuf[dest]++;
                                        And puts the cell number to
                                                DestBuf[dest]-><dest_cell

        For each cpu I, which is not ours {
                (outstanding is zero here)
                outstanding = outstanding + DestCounter[i]; destcounter
```

```
                    thus outstanding shows ;
                    how many we are sending to this I cpu;
I                   f (i != who_i_am)  {
 Csend(send_type,HeadDestBuf[i],sizeof(NEAR)*(DestCounter[i]+1),i,0);      }
DestCounter[i]=0; /* this is the basic send.

}
if we did not need to send any to other nodes:
send done_type with the number outstanding in the message.
```

Unpack_near_atoms ()

For all other CPUS.

```
send_atoms(Index); /* this generates lots of UNPACK_FORCE_TYPE
messages*/

while (Counter != 0) {
crecv(-1,RcvMsg,sizeof(DYN_MSG));

swithc processes:

case UNPACK_FORCE_TYPE:
                    casts DYN_MSG to type NEAR_MSG
                    looks up atom in local hash
                    if not updated. updates f,v,x, and where it came from,
                    sets flag updated to 1. (who clears it);

                    NEAR_MSG has another field about bond data, which is
                    updated into nbe arrays. (probably does not need that ?)
case MOVE_ON_TYPE:
          Counter--;
 break;
```

```
send_atoms(int Index);
void send_atoms(int Index)
{
double TempBuf;

  printf ("%d: send_atoms called.\n", who_i_am) ;

  while (CellBuf[Index]->dest_cell != NULL) {
     pack_atoms(CellBuf[Index],Index,0,UNPACK_FORCE_TYPE,0);
     CellBuf[Index]++;
  }
  if (Numnodes != 1)
   csend(MOVE_ON_TYPE,&TempBuf,sizeof(double),-1,0);
  csend(MOVE_ON_TYPE,&TempBuf,sizeof(double),who_i_am,0);
}
```

```
void
send_partial_forces_home(void)
{
for all atoms in localhastable (processed via atomlist[I] and lookup_atom_in hash and iterator ->hnext
        memcpy(ForceMsgMin->atom,A,sizeof(ATOM)); /* this is one atom */
```

csend(UNPACK_PARTIAL_TYPE,ForceMsgMin,sizeof(NEAR_MSG_MIN), A->sen
der,0); outstanding++;

if nothing ever was send then do csend(PARTIAL_DONE_TYPE,PartialMsg,sizeof(ATOM),0,0)

else start receiving:

case UNPACK_PARTIAL_TYPE:

update forces in hash and force_updated flag and
csend(ACK_UNPACK_PARTIAL_TYPE,&i,

case ACK_UNPACK_PARTIAL_TYPE:

- utstanding--; / decrease counter;
  - if counter is zero, csend(PARTIAL_DONE_TYPE,Pa

case PARTIAL_DONE_TYPE:

adds the message to the global PE (that's how it is send actually)b
when PARTIAL_DONE_TYPE received from all nodes,
broadcast PE back to all of them via MOVE_ON_TYPE

case MOVE_ON_TYPE:

receive the new global_pe and set done to 0, so receiving loop terminates.

Integrate()

Integrate->begin
Calc_ke caluclates local ke and sends it to 0 via csend(ENERGIES_TYPE

Collect_kes

On node zero: Receives ENERGIES_TYPE and adds them together then broadcast via
GLOBAL_ENERGIES_TYPE
On ondes other than zero: recv GLOBAL_ENERGIES_TYPE.

Integrate_force2()

Integrate->force2()

Sends MOVE_ON_TYPE to 0
Then zero receives it from all,
Adds up global kinetic energy.
Broadcasts INTEGRATE_VEL_TYPE on 0 and receives it on the other ones.

Appendix B Shared Memory Macro Definitions

B.1 Macro definitions for global sums

```
#include "locks.h"
#ifdef SGI1
#define IRIX4
#endif


#ifdef __ksr__
#define __ksr_sums
#endif

#ifdef WS
#define __ksr_sums
#endif

#ifndef WS
#ifdef __sgi
#define _USE_SGI_LOCKS
#endif
#endif

#ifdef __ksr__
/* both do_sync() calls are necessary */
#define global_sum(g,l) {\
                        do_sync() ;\
                        _gspwt(&g); \
                        g += l; \
                        _rsp(&g);\
                        do_sync();}

#define global_sum6(g,l) \
{int i; _gspwt(&g);for(i=0;i<6;i++)g[i]+=l[i];_rsp(&g);do_sync();}

#define local(a) a
#define def_local(type,a) PRIVATE  type a

/* the next four are called with pointers usually */
/* if not called for pointers, take the & in the call itself */

#define lock_atom(a) _gspwt(a);
#define unlock_atom(a) _rsp(a) ;
#define lock_other(a) _gspwt(a)
#define unlock_other(a) _rsp(a)
#define lock_otherv(a) _gspwt(&a)
#define unlock_otherv(a) _rsp(&a)
#else

#ifdef WS
#define global_sum(g,l) g += l
#define global_sum6(g,l) {int i; for(i=0;i<6;i++)g[i]+=l[i];}
```

```
#define local(a) a
#define def_local(type,a) type a
#define lock_atom(a)
#define unlock_atom(a)
#define lock_other(a)
#define unlock_other(a)
#define lock_otherv(a)
#define unlock_otherv(a)

#else

#ifdef __sgi                    /* SGI definitions */
#define paste(a,b) a##b

#define lock_atom(a) setlock(a->lock)
#define unlock_atom(a) unsetlock(a->lock)
#define lock_other(a) setlock(paste(a,_lock))
#define unlock_other(a) unsetlock(paste(a,_lock))
#define lock_otherv(a) setlock(paste(a,_lock))
#define unlock_otherv(a) unsetlock(paste(a,_lock))

#define global_sum(g,l) {do_sync();\
                        if (here == 0) {
                              int cpu;
                              for (cpu = 0 ; cpu < nthreads; cpu++) \
                              {g += l[cpu];}}\
                        do_sync();}
#define global_sum6(g,l) {do_sync();if (here == 0) {int cpu,i; for (cpu
= 0 ; cpu < nthreads; cpu++) for(i=0;i<6;i++){g[i] += l[i][cpu];}}
do_sync() ;}
#define local(a) a[here]
#define def_local(type, a)  type a[MAX_CPU]
#endif /* sgi*/
#endif /* else WS*/
#endif /* ksr */
```

B.2 Functions providing locking and mutexes

The following are examples for SGI machines using the spinlock library. Similar functions are written using the CPS library for HP Exemplar computers.

B.2.1 Header file locks.h

```
#ifndef MAX_CPU

      /* KSR */
      #ifdef __ksr__
            #define MAX_CPU 64
      #endif
```

```
      /* IRIX5 / IRIX6 / HP9000 Single Processor */
      #ifdef WS
            #undef MAX_CPU
            #define MAX_CPU 1
      #endif

      /* IRIX6 Multi Processor */
      #ifdef __sgi
            #ifndef WS
                  #define MAX_CPU 16
            #endif
      #endif

      /* HP/Convex Exemplar Multi Processor */
      #ifdef CONVEX
            #define MAX_CPU 128
      #endif

#endif

#ifdef __sgi
#include <ulocks.h>
/*
extern    ulock_t    add_force_lock;
extern    ulock_t    disp_lock;
*/

extern    ulock_t    gpole_lock ;
extern    ulock_t    add_atom_to_cell_lock ;
extern    ulock_t    movelist_lock ;
extern ulock_t      ave_sync_time_lock ;
extern double  lock_time [MAX_CPU] ;
extern double ulock_time [MAX_CPU] ;

void   init_locks (void) ;
void   setlock (ulock_t *lock) ;
void   unsetlock (ulock_t *lock) ;
ulock_t  getlock (void) ;
void finish_locks (void);
#endif

#ifdef CONVEX
      #include "clocks.h"
#endif
```

B.2.3. Implementation file locks.c

```
/* DTM:  According to Mihail, locks.c is not needed on the KSR */
#ifndef WS
#ifdef __sgi
```

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include "locks.h"
#include "sim.h"
#include "util.h"

extern int nthreads ;

ulock_t      add_atom_to_cell_lock ;
ulock_t      movelist_lock ;
ulock_t      gpole_lock ;
ulock_t      ave_sync_time_lock ;

static  char     *filename  ;
static  usptr_t  *arena ;

double  lock_time [MAX_CPU] = {0};
double ulock_time [MAX_CPU] = {0};

void  init_locks (void)
{
  int  sh_arena_size  = 100000000 ;
  int  sh_arena_users = 8 ;

  filename  = tempnam ("/usr/tmp", "lt") ;
  if (filename == 0) perror ("error setting temporary file") ;
#if 0
  printf ("Setting size to %d, old value was %d\n",
        sh_arena_size, usconfig (CONF_INITSIZE, sh_arena_size)) ;

  printf ("Setting users to %d, old value was %d\n",
        sh_arena_users, usconfig (CONF_INITUSERS, sh_arena_users)) ;
#endif

  if (usconfig (CONF_INITSIZE,  sh_arena_size) == -1) {
    perror ("usconfig CONF_INITSIZE") ;panic("usconfig") ;
  }
  if (usconfig (CONF_INITUSERS, sh_arena_users)== -1) {
    perror ("usconfig CONF_INITSIZE") ;panic("usconfig") ;
  }
  if (usconfig (CONF_ARENATYPE, US_SHAREDONLY)== -1) {
    perror ("usconfig CONF_INITSIZE") ;panic("usconfig") ;
  } /* unlink the lock file, so
                                      it does not clutter /usr/tmp */

  arena    = usinit (filename) ;
  if (arena == 0) perror ("spinlock usinit") ;
/*
  printf ("Arena size %d, max users %d\n",
       usconfig (CONF_GETSIZE,arena), usconfig (CONF_GETUSERS,arena))
;
*/

  /*  add_force_lock = getlock() ; */
  /*  disp_lock =  getlock() ; */
```

```
  add_atom_to_cell_lock = getlock() ;
  movelist_lock =  getlock() ;
  gpole_lock = getlock () ;
  ave_sync_time_lock = getlock () ;
}
/*********************************************************/
#ifndef WS
void finish_locks (void)
{
  unlink (filename) ;
  free (filename) ;
}
#endif
/*********************************************************/

void   setlock (ulock_t *lock)
{
#ifdef LOCKTIME
lock_time [here] -= dclock() ;
#endif
    if (ussetlock (lock) != 1) {
      printf ("locking failed") ;
      abort () ;
    }
#ifdef LOCKTIME
lock_time [here] += dclock() ;
#endif
}
/*********************************************************/
void  unsetlock (ulock_t *lock)
{
#ifdef LOCKTIME
   ulock_time [here] -= dclock() ;
#endif
    usunsetlock (lock) ;
#ifdef LOCKTIME
    ulock_time [here] += dclock() ;
#endif
}
/*********************************************************/
ulock_t  getlock (void)
{
  ulock_t   l = usnewlock (arena) ;

  if (l == 0) {
    perror ("spinlock getlock failed") ;
    abort () ;
  }
  return  l ;

/*  printf ("%d lock\n", m_get_myid() ) ; */
}

#endif /* __sgi */
#endif /* WS */

#ifdef CONVEX
```

```
#include "clocks.c"
#endif /* CONVEX */
```

Appendix C. API Interface to the MPsim Program

When implementing a new driver it is important to have the following features correctly:

Various setups (degrees of freedom, non-bond and bond setups; energy expression.)

Call check_move and finish_move once in a while (see example below) to make sure that atoms that physically left cells to which they were assigned are reassigned to their new cells correctly.

(Note: usually it will be necessary to call the setup nonbonds immediately after that, to initialize the nonbond structures properly. Bond structures depend on absolute atom number, and not on which cell the atom is in, so it is not necessary to call bond setups.)

Lastly, load balancing may be called once in a while, if executing on multiprocessors to redistribute the load more evenly.

Example of a main loop of a driver

```
calc_dof() ; /* degrees of freedom (dof) goes into a global variable */

/* basic loop */

while (dyn_time < dyn_steps) {

...basic loop...

setups if necessary.

calc_ef_all - computes energy and forces.
do_sync(). since calce_ef_all does not call it at the end.

Perform moving (e.g. dynamics, minimization, something else)

write trajectory entry...

Call the functions that do load balancing and reassigning atoms between
cells and cells between processors.

  do_sync() ;
  notmoving = (dyn_time + 1) % move_freq;
  check_move(!notmoving);
  if (!notmoving) finish_move();
  if (++local(loadbal_ctr) == loadbal_intvl) {
    load_balance();
    local(loadbal_ctr) = 0;
  }
  do_sync() ;


  write snap entry if desired.
```

```
if (here == 0) dyn_time ++ ;
   do_sync () ;
 }


/* Note: can not use for, since dyn_time is shared */

-------------------------------------------------
Basic loop to process all atoms in parallel.

example from computation of global kinetic energy

float
calc_global_ke(void)
{
        int i ;

      do_sync();

      local(ke) = calc_cart_ke();/* each cpu will compute it's ownKE */

      do_sync();
      if (here == 0) { /* node 0 also adds up the rigid KE */
        global_ke = calc_rigid_ke() ;
      }
      global_sum (global_ke,ke) ; /* node 0 will add them up */

     return global_ke ;
}
-----------------------------------------

static float
calc_cart_ke(void)
{
      float ke, atom_ke;
      CELL *cell;
      ATOM *atom;

      ke = 0.0;

      init_next_cell();
      while ((cell = get_next_cell()) != NULL) {
            for (atom = cell->c_atom; atom != NULL; atom = atom->cnext)
{
/*    this is the basic loop that goes over all the cells on this cpu
and the over all the atoms contatined in them */

                  if (ATOM_NMOVE(atom)) continue;
                  atom_ke = 0.0;
                  atom_ke += atom->v[0] * atom->v[0];
                  atom_ke += atom->v[1] * atom->v[1];
                  atom_ke += atom->v[2] * atom->v[2];
                  ke += atom->m * atom_ke;
            }
      }
      return (ke * 0.5);
```

```
}
/*************************************************************************/
```

Loops in which atoms are processed in sequence can be done in the following way. This has to be done on one cpu only (usually 0; cpu)

```
n = init_next_atom();

for (i = 0; i < n; i++) {
     ATOM *a = get_next_atom();

.....
}
```

All the atom and cell interfaces are defined in atom.h and cell.h

ATOM a, can be tested with the following macro predicated for specific properties:

```
ATOM_FIXED(a)          true if atom is fixed.
ATOM_RIGID(a)          true if atom is part of a rigid molecule.
ATOM_NMOVE(a)          true if atom non-movable, e.g., rigid or fixed.
ATOM_TRACK(a)          true if atom is being tracked.
```

Those properties can be set in the following way.

```
a->flags |= ATOM_FIXED_FLAG ;
a->flags |= ATOM_RIGID_FLAG ;
a->flags |= ATOM_TRACK_FLAG ;
```

The read_file biograf reader automatically sets the first one from data in the .bgf file and rigid and track are set from the corresponding keywords in the .ctl file.

ATOM *get_atom (int gnum) fetches an atom by absolute number.

```
void add_force(ATOM *const a,
               const float fx, const float fy, const float fz,
               const char *const type);

                  /* this function adds force to the
                   * atom A. It can be used to
                   * implement new terms of the forcefield.
```

The string type serves only debug-informational value.

All the cells interfaces are in cells.h

```
void assign_cell(ATOM *atom); /* determines which cell this atom
                               * belongs to and assigns the value in
atom->cell
void add_atom_to_cell(ATOM *atom); /* links the atom to its appropriate
cell */
```

Appendix D Macro Language for MPSim

Control file for MPSim

This is a template for the ctl file with description of the keywords in the control (ctl) file. The meanings of the "numbers" are explained within the file. Anything in brackets is optional. Select exactly one of anything in braces. Text in angle brackets describes a parameter (usually a number). Case and spacing are never important.

PROJECT dreid_hb
Base filename for .traj, .snap, .dbg files

This is just a name that's used to construct output filenames.For example, if the project name is PE20, the snapshot restart files are going to be saved as PE20.snapXXXX,where xxxx is the timestep; a trajectory file containing energy and rms force data is going to be saved as PE20.traj1, for the first run, PE20.traj2 for the next, etc. BGF trajectory file is going to be saved as PE20.trj if requested.

FF dreid_hb.ff
This should be the filename of the forcefield parameter file.

BBOX 0.00000 17.29713 0.00000 17.29713 0.00000 17.29713
{min x} {max x} {min y} {max y} {min z} {max z}

For finite, non-periodic cases, specify the minimum and maximum coordinates in the X, Y, and Z axes. These can, and often should, be much smaller/bigger than the actual minima and maxima in order to make the subdivided boxes large enough and provide room for the system to expand.

These values have impact on the precision and speed of the various method so it might be nice to experiment. A utility called "bbox.pl" can be used to compute the bbox dimensions like, for example, "bbox.pl structure.bgf" would give the results. Note: You only need that for non-periodic systems, for periodic ones it is the size of the unit cell.

STRUCTURE dreid_hb.xtl

This should be the name of the BIOGRAF file. Initial charges should be specified in the file. If you want to use PBC, the .xtl file should contain a CRYSTX record. This line must come after the FF line.

ACTN **[OneEnergy Dynamics Minimize Traj]**

Selects what to compute:

Dynamics runs MD (TPN, TVN, EVN, depending on selection)

Minimize runs minimization (conjugate gradient or steepest descent)

OneEnergy computes energies and forces for the

Traj - run in trajectory conversion mode (see the RDTRJ keyword then)

NB_METHOD [CMM EWALD SPLINE]

Select a method for non bond computations.
CMM is the Cell Multipole Method.
SPLINE is the spline cutt-offs.
EWALD is Ewald summation for periodic systems.

LEVEL 6

Number of octree levels. This should be a number that controls the level of subdivisionof the bounding box or unit cell. Generally more levels will provide faster run and lower precision (especially true for CMM).

The level, along with the bounding box size, will determine the average number of "atoms per leaf cell" which if too small might lead to loss of precision.

For exact non-bond calculations, use CMM with level of zero(0).

CMM_EXPANSION [centroid center qcenter]

Determines how the multipole expansion is done.

Qcenter - uses charge center
Centroid - uses the centroid
Center - uses the geometric center

EWALD_ACC 0.0001

Accuracy for ewald calculations

SPLINE_CUTOFF 8.000 8.500 9.000

These are r_spline_on, r_spline_off, and r_spline_cutoff, respectively. This should only be specified if "spline" is the chosen non-bond method.

HB_CUTOFF 4.0 4.5 5.0 65.0 75.0 90.0

These are the H-bond cutoff distances(dreid_hb_ron, dreid_hb_roff, dreid_hb_rcut) and angles (dreid_hb_angle_on, dreid_hb_angle_off, dreid_hb_angle_cut) in the DREIDII's hydrogen-bond force field.

OFTEN nbupdt_freq move_freq loadbal_freq temp_freq

Specifies update frequencies

nbupdt far-field CMM update or nonbond list rebuild

move atoms reallocation to cells.
loadbal loadbalancing across processors
temp Rescale the velocities to match the desired temperature.

INTEGRATE [TVN EVN TPN Neimo-TVN]

Specifies integration method

TVN - Constant temperature, volume and number of particles ensemble using Nose-Hoover method.

EVN - Constant energy, volume and number of particles ensemble.

TPN - Constant temperature, pressure and number of particles ensemble (Gibbs).

Neimo-TVN - TVN ensemble using only torsion angles as degrees of freedom.

DYN_STEPS 100 1

The first number is the number of dynamics steps to be done. The second number is the time step in fs.

MINMETHOD [Steepest Conjugate]
Selects a minimization method

MINPAR 0.1 500
{ rms force criteria} { number of steps}

MINCONS [Volume Pressure]
{ Constant volume constraint} { constant pressure constraint}

Specifies minimization constraints

PSSOLV [Yes No]

Select whether to assume explicit solvent

PSPARS <External Solvent Epsilon> <Solvent radius>
Poisson-Boltzman solvent parameters

Epsilon

Set the dielectric constant of the continuum. Default is 80.37, which corresponds to water at 298 Kelvin (?).

Radius

Set the probe radius of the solvent, which defines the accessibility of the solute molecule. Default value is 1.40 Angstrom, which corresponds to water.

EXTERNAL_P 1.0 1.0 1.0 0.0 0.0 0.0
{xx} {yy} {zz} {zy} {zx} {xy}
Txx Tyy Tzz Tzy Tzx Txy are 6-components of the external stress in atmosphere.

DYN_TEMP 300.0
This is the temperature in kelvin at which to run the simulation.

NOSEtau 0.1
This is the tau constant for the Nose-Hoover heat bath. It looks like it should be near the shortest frequency in the system (often 0.01 ps). If you leave it out, it defaults to 100 times the timestep. It's specified in ps, not fs.
When doing a restart you must have a
Nose tau specified in the ctl file.

GIBBS 0.1
cell mass factor for TPN dynamics

THE keyword "GIBBSpar" also works instead of the above "GIBBS" keyword as in

GIBBSpar 0.1

RIGID 1000
This number 1000 is the atom number (for example in this case it is 1000th atom).
The molecule of which the 1000th atom is a member is made rigid. So one
needs to list only one atom in any molecule which the user wants be kept
rigid. If you have more than one molecule to be made rigid (for example

all the water molecules) one has to list an atom number in each molecule
which is to be made rigid.
Exactly one atom per rigid molecule should be specified. You can put more than one atom number on each RIGID line.

This also understands things like size <= 3, which will make all molecules with
Less than 3 atoms rigid.

TRACK atom 2409 atom 2410 atom 2411 atom 2412 atom 2413 freq 1000
atom {atom number} freq {trackfile frequency}

track files are output files containing coordinates, velocities and
forces of each of the atom listed under the keyword "TRACK" as shown above.

"freq" is the frequency at which one wants the track files to be written.
if the number of track atoms is too large then you may want to write less often
and so on.

CD change working directories.

BGFtrajectory Yes/No whether to write one

BGFtrajvel Yes/No include velocities?

BGFtrajfreq number how often to write it.

For trajectory conversion mode only:

RDTRJ step number - what step was used in ps, in the simulation in which the snapfiles were produced.

RDTRJ snap file1

RDTRJ snap file2

... - lists the files that are to be processed.

RDTRJ snap filen

The following 3 keywords to run the new version of MPSim(sim_2.0.3).
SETUPEEX - sets up energy expression
DO - runs the simulation previously selected with INTEGRATE
INFO - this will print biograf style breakup in energies and other energy expression information.
EXIT exit the program. Otherwise at end of macrofile attempt will be

made to read from standard input.

Old style keywords (LEVEL FREQ ACTION ONE_EF DYNAMICS MINIMIZATION)

LEVEL 6 centroid
{number of octree levels} [{center,centroid,qcenter,spline}]

New style: use NB_METHOD and CMM_EXPANSION in addition to LEVEL with a number. For example the above will be written as :

```
LEVEL 6
NB_METHOD CMM
CMM_EXPANSION centroid
```

FREQ nbupdt 1 move 1 loadbal 1 temp 0

Same as the OFTEN keyword, except that the nbupd, move, etc,. keywords needs to be given.

ONE_EF Compute one energy. In the new style use ACTN.

Same as ACTN OneEnergy

DYNAMICS 100000 1 gibbs
{number of steps} {timestep in fs} [nose] [min] [globf][globv]

The first number is the number of timesteps. The second number is the size of the timesteps in fs. Select "nose" for Nose- Hoover dynamics, "min" for steepest-descent minimization,neither for microcanonical dynamics. "globf" causes the net global force to be reset to zero at each timestep; "globv" causes the net global velocity to be reset to zero at each timestep for Nose-Hoover or at each temperature rescaling for microcanonical; "gibbs" for gibbs dynamics.

Instead of "DYNAMICS" use "INTEGRATE" as shown above:
WHEN USING "INTEGRATE" KEYWORD USE ALSO "DYN_STEPS" KEYWORD AS SHOWN ABOVE.

The above is equivalent to

```
ACTN DYNAMICS
INTEGRATE TPN
DYN_STEPS 100000 1
```

MINIMIZATION 1 0.1 500 0

{method} {number of steps} {rms force criteria}

The first number is the method to be used in the minimization.1 stands for conjugate-gradient method, while 2 stands for steepest-descent method.(Other methods will be implemented later). The second number is the convergence criterion for the RMS forces. The third number is the maximum number of minimization steps. The fourth number denotes cell relaxation or not; 0 indicates constant volume.

For example, the above is equivalent to:

ACTN Minimize

MINMETHOD Conjugate

MINPAR 0.1 500

MINCONS Volume

VELSTART [nonmin birand truerand]

nonmin option does not double the initial kinetic energy, which is appropriate for Nose-Hoover or for minimization.

birand uses a set of initial velocities which is reproducible given the same set of input parameters. This is the same as using the same random seed for a given set of input parameters.

truerand uses a truely random set of initial velocities for a given set of parameters.

NEIMO 0 0 initvel

This option is to use NEIMO with MPSim (no waters yet!)always use 0 0 option. If the user wants the torsion angleswritten every timestep, then put 1 for the first zero.initvel is for using a set of random initial velocities.The other option readvel instead of initvel where a set of initial dihedral velocities can be read in.

ACTION same as SETUPEEX

DEFAULTS

When the program starts the defaults are as if the following MACRO has just been executed.

THREADS 1 - for workstation version

```
THREADS    CPUs          - on computer with CPUs number of available CPUs.
(No PROJECT and LEVEL defaults, so those need to be given)
EWALD_ACC .01
NB_METHOD CMM
ACTN OneEnergy
CMM_EXPANSION center
SPLINE_CUTOFF 8 8.5 9
HB_CUTOFF 4 4.5 5.0 65 75 90
EXTERNAL_P 1 1 1 0 0 0
SETUP_FREQS 5 5 5 1
INTEGRATE TPN
DYN_TEMP 300
DYN_STEPS 100 1
MINMETHOD Steepest
MINCONS Volume
MINPAR 1 500
NOSEtau 100DYN_STEPS
PSSOLV NO
PSPARS 80.37 1.40
BGFtrajectory No
BGFtrajvel No
VELSTART nonmin birand
```

Relevancy dependency of the keywords:

Keyword	only relevant if
SPLINE_CUTOFF	NB_METHOD Spline
CMM_EXPANSION	NB_METHOD CMM
EWALD_ACC	NB_METHOD Ewald
EXTERNAL_P	INTEGRATE TPN
DYN_TEMP	
DYN_STEPS	ACTN Dynamics
MINMETHOD	
MINPAR	
MINCONS	ACTN Minimize
NOSEtau	INTEGRATE TPN or TVN
CELL_MASS	INTEGRATE TPN
RDTRJ	ACTN Traj

Appendix E Command Line Interface to the Diffusion Analysis Toolkit.

The sources for those utilities are available under /ul/iotov/xgd on the MSC/WAG computers.

E.1 Void extraction

voids $parfile -$bgffile $diameter $centermode $snapfiles > $output_void_file

(option to disable '-' and thus include the bgf file will be added soon)

centermode: 2 is hard sphere option.

This feature extracts voids from snapshot files obtain via running MD simulations.

E.2 Create voids

vgen $bgf $rates > $voids

E.3 View voids

void1.pl $bgffile $voidsfile $snapshots $total > tmp.iv; ivview -q tmp.iv; rm tmp.iv

Displays voids location in blue/red mixture color. Red means more probable void, blue is less so.

The first number on the command line is the number of snapshots used to make the voidfile and the second is for the value of the first void line (e.g., the leftmost value on the first line of the void file). For voids generated with the 'Create Void' option, the values of 1 and 1 are good. Those are currently the only allowable values, too. Operating on files sorted on the 4th columns is a lot faster.

E.4 View track

genericktracksnap.pl $trackfile $bgffile > > tmp.iv; ivview tmp.iv; rm tmp.iv

Generates a picture of the movement of the atoms tracked in the $trackfile.

(Options to do animation and to select only certain atoms will be added later.)

E.5 Monte Carlo random walk simulation

gmcd <bgf> <void file> [mc.rates]

The mc.rates file format is described in Section 4.9.

E.6 Cluster void analysis

cvoid [-d] [-c size_cutoff] bgf voidfile base_outputname

-c option disables output of small voids.

Appendix F. Residence Time Extraction Algorithm

F.1 Algorithm idea

The idea of the algorithm is to collapse a feliciton to a point with the following properties:

Coordinates	x , y, z
residence time	t,i
transition rates	r,ij
transition times	t,ij

F.2 Algorithm details

The following algorithm elaborates how to extract this data from the Monte Carlo diffusion simulation:

Inputs: voids description, transition probabilities.

Output: residency time and transition rates and times for each void.

1. Simulate penetrant dynamics according to the MC diffusion model. Start in void 1.

2. Record all "leave" and "entry" events and their times.

3. Each time the following sequence of events happens:

A. Leaving void i
B. Entering void j
C. Leaving void j
D. Entering void k

Perform the following:

Add time(C) - time(B) to residency CT_j time of void j
Increment counter of CR_{jk} and total counter C_j
Add time(D) - time(C) to transition time CRT_{jk} and increment counter $_{CTjk}$

Time(A) means the step at which event A happened.

When this is done many times ($C_j >> 1$), compute

Residency times : $T_j = CT_j / C_j$

Transition rates : $R_{jk} = CR_{jk} / C_j$

Transition times: $TR_{jk} = CRT_{jk} / TC_{jk}$

F.3 Assumptions

Leaving a void is independent of which channel the particle entered from. This assumes enough friction in the void so the particle forgets where it entered from.

We expect R_{jk} to be symmetric and the residency time distribution to be Poisson.