11-16-1988

A Fractal Analysis of Diffusion Limited Aggregation

Cliff Myers
Portland State University

AN ABSTRACT OF THE THESIS OF Cliff Myers for the Master of Science in Physics presented November 16, 1988.

Title: A Fractal Analysis of Diffusion Limited Aggregation.

APPROVED BY MEMBERS OF THE THESIS COMMITTEE:

[redacted]

David I. Paul, Chair

[redacted]

Erik Bodegom

[redacted]

Jack S. Semura

[redacted]

Eugene A. Enneking

A modified Witten-Sander algorithm was devised for the diffusion-limited aggregation process. The simulation and analysis were performed on a personal computer. The fractal dimension was determined by using various forms of a two-point density correlation function and by the radius of gyration. The results of computing the correlation function with square and circular windows were analyzed. The correlation function was further modified to exclude the

edge from analysis and those results were compared to the fractal dimensions obtained from the whole aggregate. The fractal dimensions of 1.67 ± .01 and 1.75 ± .08 agree with the accepted values. Animation of the aggregation process elucidated the limited penetration into the interior and the zone of most active deposition at the exterior of the aggregate.

A FRACTAL ANALYSIS OF DIFFUSION LIMITED AGGREGATION

BY

CLIFF MYERS

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
PHYSICS

Portland State University

1988

TO THE OFFICE OF GRADUATE STUDIES

The members of the Committee approve the thesis of Cliff Myers presented November 16, 1988.

[redacted]

David I. Paul, Chair

[redacted]

Erik Bodegom

[redacted]

Jack S. Semura

[redacted]

Eugene A. Enneking

APPROVED:

[redacted]

David I. Paul, Chair, Department of Physics

[redacted]

Bernard Ross, Vice Provost for Graduate Studies

ACKNOWLEDGEMENTS

I would like to thank my friends who stood by me and offered support and encouragement during the time it has taken to complete this 'vast project'.

I would like to especially extend my gratitude toward Michael E. Sullivan for his expertise and collaboration on the creation of the computer programs and for teaching me that Basic is much more than peeks and pokes, without whom the programs would lack their efficiency and elegance.

To the following members of the the faculty who participated in this project I want to thank:

Dr. Bachhuber for his patient and gentle prodding, to only 'kill the flea on the tail of the lion' and for his enthusiasm for computer simulation and encouragement to study fractals.

Dr. Bodegom for his ideas, insights, and effective time devoted to this project.

Dr. Paul for lighting a fire under me.

And to all these people I want to express my appreciation for consistently availing themselves to me.

Finally, I would like to thank B. Jennifer Selliken, who came to my rescue with her proficient editing and helped transform these pages into something I can be proud of.

TABLE OF CONTENTS

PAGE

ACKNOWLEDGEMENTS iii

LIST OF TABLES vi

LIST OF FIGURES vii

CHAPTER

I INTRODUCTION 1

II BACKGROUND MATERIAL 4

The Fractal Dimension 4

Density Scaling 5

The Correlation Function 7

The Diffusion-Limited Aggregation Model .. 9

The Laplace Equation 12

Experimental Realizations of the Model ... 14

Electrodeposition
Hydrodynamics
Dielectric Breakdown

III IMPLEMENTATION OF THE MODEL 17

IV SIMULATION RESULTS AND DISCUSSION 20

Numerical Results 20

Correlation Function Results
Radius of Gyration Results

Graphical Results 33

V CONCLUSION 47

REFERENCES CITED 52

APPENDIX

A THE COMPUTER PROGRAMS 53

B NUMERICAL DATA 120

C GRAPHICAL DATA 137

D ADDITIONAL RADIUS OF GYRATION ANALYSIS 153

E CONSIDERATIONS FOR FURTHER WORK 160

LIST OF TABLES

TABLE		PAGE
I	Average Fractal Dimensions	22
II	Fractal Dimension Data for Individual Small Aggregates	120
III	Fractal Dimension Data for Individual Large Aggregates	121
IV	Corrected Radius of Gyratiion Results for Aggregate Number 20	156
V	Corrected Radius of Gyration Results for Aggregates Numbers 1 to 20, Inclusive ...	157

LIST OF FIGURES

FIGURE PAGE

1. Scale Invariance of a Fractal Aggregate 5
2. Sierpinski Gasket 6
3. Correlation Dependence on 'Window' Size 27
4. Fractal Dimension vs. Aggregate Number 28
5. Radius of Gyration Dependence on Number of Deposits for Small and Large Aggregates ... 32
6. Aggregate Number 20 34
7. Quartile Stages of Growth of Aggregate Number 20 35
8. Radius of Gyration Dependence on Number of Deposits for Aggregate Number 20 36
9. Radial Mass Distribution for Aggregate Number 20 36
10. Angular Mass Distribution for Aggregate Number 20 (Smoothed) 37
11. Mass Distribution in X for Aggregate Number 20 38
12. Mass Distribution in Y for Aggregate Number 20 39
13. Upper Percentiles of Deposition for Aggregate Number 20 40

14. Cumulative Probability Distribution in X and Y 41

15. Cumulative Angular Mass Distribution of the 30 Large Aggregates (Smoothed) 43

16. Cumulative Mass Distribution in X for the 30 Large Aggregates 44

17. Cumulative Mass Distribution in Y for the 30 Large Aggregates 45

18. Cumulative Radial Mass Distribution for the 30 Large Aggregates 46

19. Persistence of Growth Trends 46

20. Demonstration Program Flowchart 55

21. Ln(R_g) vs. Ln(N) for Aggregate Number 1 122

22. Ln(R_g) vs. Ln(N) for Aggregate Number 2 122

23. Ln(R_g) vs. Ln(N) for Aggregate Number 3 123

24. Ln(R_g) vs. Ln(N) for Aggregate Number 4 123

25. Ln(R_g) vs. Ln(N) for Aggregate Number 5 124

26. Ln(R_g) vs. Ln(N) for Aggregate Number 6 124

27. Ln(R_g) vs. Ln(N) for Aggregate Number 7 125

28. Ln(R_g) vs. Ln(N) for Aggregate Number 8 125

29. Ln(R_g) vs. Ln(N) for Aggregate Number 9 126

30. LN(R_g) vs. Ln(N) for Aggregate Number 10 126

31. Ln(R_g) vs. Ln(N) for Aggregate Number 11 127

32. Ln(R_g) vs. Ln(N) for Aggregate Number 12 127

33. Ln(R_g) vs. Ln(N) for Aggregate Number 13 128

34. Ln(R_g) vs. Ln(N) for Aggregate Number 14 128

35. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 15 129

36. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 16 129

37. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 17 130

38 $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 18 130

39. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 19 131

40. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 20 131

41. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 21 132

42. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 22 132

43. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 23 133

44. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 24 133

45. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 25 134

46. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 26 134

47. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 27 135

48. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 28 135

49. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 29 136

50. $Ln(R_g)$ vs. $Ln(N)$ for Aggregate Number 30 136

51. Corrected Radius of Gyration Dependence on Number of Deposits for Aggregate Number 20. 156

52. Corrected Radius of Gyration Dependence on Total Number of Deposits for 26 Small and Large Aggregates 158

CHAPTER I

INTRODUCTION

Many complex forms in nature are products of some kind of growth process. There are growth processes ranging from the formation of galaxies to polymers, from the structure of snowflakes to that of living systems. It is hoped that insight into the underlying mechanisms of growth and the formation of structure can be gained from exploration of more tractable models than the direct study of these complicated physical systems. Researchers have been recently encouraged by the intricate patterns and scaling relations that can be produced by computer simulations. By using few and simple growth rules it is suggested that the computer models can elucidate some of the essentials of the mechanisms of growth.

Many everyday forms have the property of self-similarity, that is, the appearance of the structure is invariant under change of length scale. Familiar examples include coastlines, rivers, and lightning. The quantitative description of the structure of these forms, which had been until recently regarded as too complicated, has been facilitated by the concept of the fractal dimension, which was primarily developed by Mandelbrot in 1975. It has provided the tool for understanding a diverse variety of processes which lead to similar fractal geometries. Aside from scientific considerations, structures with fractal geometries are found in many processes and products of technological importance, such as, aggregates and fluid flows.

The other development which has stimulated much recent research is the Witten-Sander model of diffusion-limited aggregation (1981). The fractal graphical output produced by the computer simulation bears a striking resemblance to actual structures and patterns found in nature, examples of these include; cathodic deposition, dielectric breakdown, and viscous fingering. These physical growth processes and the stochastic growth rules of the simulation can be related to a potential field described by Laplace's equation. Moreover, computation of the fractal dimension has been verified by direct experimental measurement. This suggests that the model provides a basis for understanding previously unrelated processes and that computer simulation can serve as a bridge between theory and experiment.

I have devised a modified Witten-Sander algorithm for the diffusion-limited aggregation process and performed the simulation and analysis on an Atari 1040ST personal computer. After generating the patterns, the fractal dimension was computed by using a two-point density correlation function and compared to that obtained using the radius of gyration. The method of computing the correlation function was modified to study edge effects. Frequency histograms were obtained for various coordinate systems to investigate any defects in the simulation. Animation programs were written to demonstrate the active zone of deposition and to better illustrate the deposition process.

After presentation of background material and details of the model, the method of simulation and programming details are then discussed. Following that, the graphical and numerical results are analyzed and compared to similar theoretical and experimental studies. Concluding remarks are then offered in support of the accepted fractal dimension for diffusion-limited aggregation. Additionally, comments are presented to address the differences between the methods for computing the fractal dimension.

CHAPTER II

BACKGROUND MATERIAL

THE FRACTAL DIMENSION

Mandelbrot has extended the application of geometrical constructs to the natural sciences by generalizing the scaling relationships found in certain mathematical functions and geometric patterns. These had been previously disregarded as pathological, to the forms common in nature. He recognized that fractal forms could serve as tools for analyzing physical phenomena. Fractal geometry may become better suited to deal with the real world of intricacies and irregularities than the Euclidean idealizations of abstract regular forms of smooth curves and surfaces.

The concept of fractal dimension, subsequently referred to in this thesis as D, is demonstrated by considering the diffusion-limited aggregate grown by the simulation in the embedding Euclidean dimension, $d = 2$, as having a fractional dimension such that $1 \leq D \leq d$ (Figure 1.). The aggregate is not a compact surface punctured with holes, nor is it a meandering line, it is a fractal (except on the scale of pixels). The irregularities are not without order in that fractals have an intrinsic symmetry, the property of self-similarity, although for random

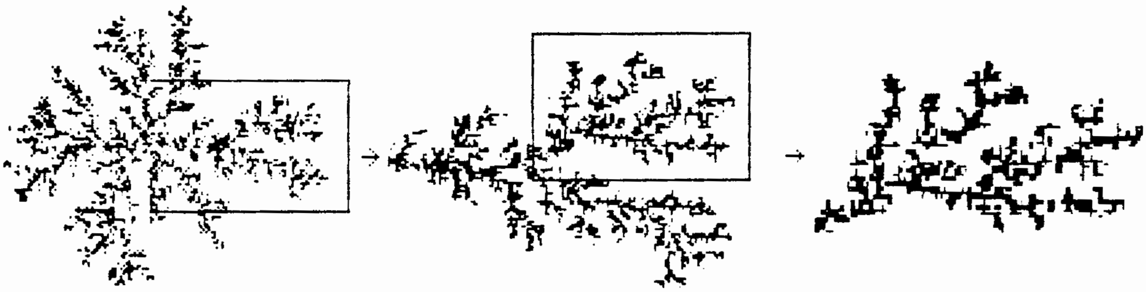


Figure 1. Scale invariance of a fractal aggregate.

fractals this dilation symmetry is statistical.

Although the structure is grown by a random process, it is not random. As the sections of the structure are magnified the pattern is recognizable so that similar structure exists on all scales between an upper cut off, nearly the size of the aggregate and a lower cut off, on the order of a pixel diameter. Thus, there exist 'holes' at all length scales. A purely random pattern would not show this scaling of 'holes'. As a consequence of having 'holes' of all sizes, the pixel density decreases with increasing length scale. This can be contrasted with a homogeneous object of Euclidean geometry where the density is independent of the length scale on which it is measured.

DENSITY SCALING

The fractal dimension is a measure of how density approaches zero as the length over which it is measured increases (assuming that there is no upper cut off). The functional equation, $M(\lambda L)=\lambda^{d}M(L)$ with $\lambda > 0$, describes how the mass of Euclidean objects scale with length. This is analogous to regular fractal objects such as Sierpinski gaskets. These can also be described by $M(\lambda L)=\lambda^D M(L)$ with $D < d$ (D is also called the similarity dimension since it describes how the mass changes after a change of scale, λ.) (Figure 2.) The solution for the fractal mass dependence on size is obtained by use of $\lambda = L^{-1}$ and $M(1) = 1$ and is

$$M(L) = L^D. \tag{1}$$

The density, ρ, given by $\rho = M/L^d$ for exact fractals is

$$\rho = L^{D-d}. \tag{2}$$

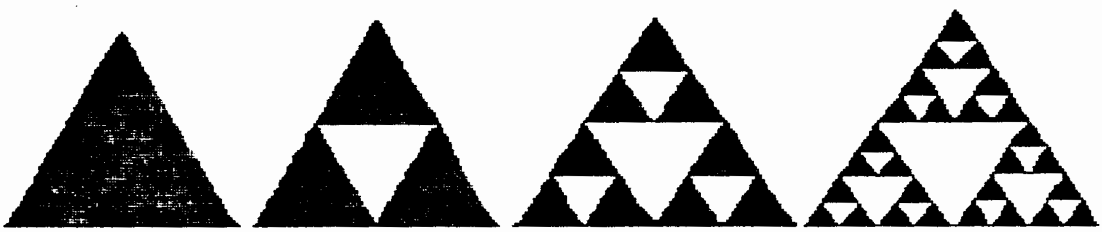


Figure 2. Sierpinski gasket.

For the Sierpinski gasket of Figure 2, the mass scales according to $M(2L) = 3M(L) = 2^D M(L)$ and $D = \ln 3/\ln 2 \cong 1.585$. Although, for exact fractals such as Sierpinski gaskets the fractal dimension can be calculated due to their deterministic construction rules; the fractal dimension for diffusion-limited aggregates grown with a stochastic process can only be measured.

The fractal dimension, as introduced, corresponds to the mass dimension in physics and any characteristic length such as the radius of gyration can be used to relate an aggregate's mass to its size during the process of growth.

In a general way, the fractal dimension can be defined by:

$$N(r) = (r/r_o)^D \tag{3}$$

where N(r) is the quantity obtained by measuring a fractal medium with a gauge r_o. Forrest and Witten (1979) first obtained for aggregated smoke particles that $M(L) = L^{1.6}$ and concluded that there were long range correlations in the particle density. There is another, less globally defined formulation for the fractal dimension, it is the correlation function, C(r), which must also reflect the scale invariance.

THE CORRELATION FUNCTION

The correlation function, C(r), may be defined as the average density of an aggregate at a length r from occupied sites and, as such, it is a local measure of the average environment of a site, $C(r) = N^{-1}\Sigma\ \delta(r_i+r)\delta(r_i)$ summed over the occupied sites, r_i, $i = 1,...,N$. The correlation function thus describes the probability that a site within a length r is occupied. The probability of occupancy is the ratio of occupied sites to the total sites of possible occupancy. Using equation (2), the correlation function is:

$$C(r) = r^D\ r^{-d} = r^{D-d} = r^{\alpha}. \tag{4}$$

Witten and Sander (1981) first noticed that the correlation function for diffusion-limited aggregates was consistent with a power law, and found $C(r) = r^{-0.343}$. The correlation function is scale-invariant in that $C(\lambda r) = \lambda^{\alpha}C(r)$.

Although, globally, the density of the aggregate decreases as it grows, (due to the corresponding growth in the 'hole' size distribution) locally, these unoccupied sites between the extending tenuous arms do not affect the correlation function if $r \ll L_{MAX}$. It is the screening effect of these growing arms that allows for fractal, as opposed to compact growth. That is, it allows for the long range correlations in the pattern, and the decrease in aggregate density.

Aggregation processes can be roughly classified into three regimes. The first of these is when an object grown near equilibrium, such as a crystal, which has only short range correlations. This correlation length or resemblance distance is on the order of the unit cells of the crystal. When the system is driven away from equilibrium, growth is in the second regime. For example, in supercooled solidification, the morphology becomes that of dendritic pattern formation where the structure may still be regarded as compact. The lengths associated with the steady-state growth of the intricate patterns of snowflakes are much longer than the crystalline lattice spacing (see Langer, 1980). The third regime, applies to diffusion-limited aggregation in which the growth process is irreversible and its growth is even farther from equilibrium. It has long range density correlations and no natural length scales, evident by its having holes of all sizes.

THE DIFFUSION-LIMITED AGGREGATION MODEL

In the Witten-Sander model for diffusion-limited aggregation or DLA, pixels are added one at a time to the growing aggregate, via random walk trajectories on a lattice. The process is started with a single seed at the lattice origin. Subsequent pixels are introduced from random points sufficiently distant so that their flux is isotropic. They then undergo simulated Brownian motion until a site adjacent to the aggregate is reached, where they irreversibly 'stick' without rearrangement.

Various improvements and extensions to this process have been developed, beginning with the work of Meakin (1983a). Meakin injected the random walkers from a random point on a circle of radius five lattice spacings greater than the distance from the seed to the most distant pixel on the growing aggregate, $R_{INJECT} = R_{MAX} + 5$. The random walker was also 'killed' if $R > R_{KILL} = 3R_{MAX}$.

With an average aggregate size of 9700 pixels, Meakin obtained fractal dimensions, of 1.68 ± .04 and 1.68 ± .07 taken from calculations using the radius of gyration and a correlation function, respectively.

In order to investigate lattice effects, the sticking rules were modified. The particle was incorporated into the aggregate if it reached a next-nearest neighbor position and did not stick if it was at the nearest neighbor position.

The corresponding dimensions of, 1.69 ± .07 and 1.70 ± .07 were obtained for aggregates with an average size of 5900 pixels.

In order to investigate the effects of the 'sticking' probability on the fractal dimension, the probability was set at 0.25 for nearest neighbor sites and 0.0 for the next-nearest neighbors. The aggregates, with an average size of 16,300 pixels, yielded fractal dimensions of, 1.71 ± .055 and 1.73 ± .13 respectively. Setting the probabilities at 0.0 for nearest neighbor sites and 0.1 for the next-nearest neighbors, Meakin further obtained the fractal dimensions of, 1.74 ± .03 and 1.73 ± .04 respectively, for aggregates with an average size of 9,800 pixels.

Later improvements in the simulation algorithm include those by Meakin (1983b) where the aggregation rate was increased by scaling the step size of the random walk to the distance from the aggregate. The step size was increased to two lattice units if the random walker was at a distance greater than r_{MAX} + 5 lattice units from the center seed, four units, if greater than r_{MAX} + 10 units, four, if greater than r_{MAX} + 20, eight if greater than r_{MAX} + 40, and sixteen if r_{MAX} + 80. The correlation function was calculated for $5 \leq r \leq 50$ and gave a fractal dimension of, 1.68 ± .05. The radius of gyration gave a fractal dimension of, 1.73 ± .06. These results were obtained from aggregates whose average size was 8,585 pixels.

It can be seen that, for these relatively small aggregate sizes (Meakin states that these aggregate sizes reached the practical limit for the VAX-11/780 computer which was used), the fractal dimension obtained by radius of gyration calculations agreed well with those that were based on the correlation function. Furthermore, the results were not significantly changed by the described modifications in the simulation process.

The diffusion-limited aggregation model was developed to provide a simple model for a broad class of growth processes in which diffusion limits the rate of irreversible growth. The reason that the model produces fractal growths and not non-symmetric amorphous blobs can be qualitatively explained by the interplay of noise and growth. Consider the random deposition of a few nearby particles; tiny bumps and 'holes' will be formed due to noise of the Brownian process. The bumps will grow faster than the interior of the 'holes' because the probability that the random walking particles will arrive at the bumps, is greater. (This is demonstrated by the lightning rod effect in electrostatics.) As the bumps become steeper, the deposition probability decreases for the interior of the 'holes'. The bumps grow larger due to this screening effect and tiny bumps, in turn, begin to form on them, then subsequent splitting occurs and this gives rise to the ramified fractal structure. This evident growth instability is similar to the Mullins-Sekerka instability of solidification processes. The association between diffusion-limited aggregation and certain processes of electrostatics (electrolytic deposition and dielectric breakdown), thermal-mass transport (dendritic solidification), and hydrodynamics (viscous fingering) is more than similar growth instabilities, or structure. Although these processes apparently do not involve diffusing 'particles', the 'particles' are conserved and under appropriate conditions they can all be described by harmonic functions which satisfy Laplace's equation.

THE LAPLACE EQUATION

That the random walkers diffuse can be understood by noting that the probability that the $\underline{x}$ site is reached on the k+1 step is: (following Witten and Sander, 1983)

$$u(\underline{x},k+1) = 1/4\ \Sigma u(\underline{x}+\underline{1},k), \qquad (5)$$

where the summation over $\underline{1}$ runs over the 4 neighbors of $\underline{x}$ and is simply the previous mean value of the neighboring sites. Without boundaries to distort the probability field, the random walk will eventually diffuse everywhere (In the simulations, it is hoped that the random walker has no preferred direction.) In the continuum limit, this becomes the diffusion equation for the probability distribution of an incomimg particle (equivalent to the average concentration if many were simultaneously diffusing), with B as the diffusion constant:

$$\partial u/\partial t = B\nabla^2 u. \quad (6)$$

The boundary conditions for DLA are given by the simulation rules: because the particles deposit on the growing aggregate u = 0 on the perimeter and because the particles approach isotropically $u = u_\infty$ for $\underline{x} \to \infty$. Because only one walker arrives at a time, they 'see', essentially a steady-state; that is, each deposit's perturbation of the field relaxes instantaneously. Thus, the diffusion equation reduces to Laplace's equation, outside the aggregate:

$$\nabla^2 u = 0. \quad (7)$$

More formally, the probability distribution is analogous to a potential field, the gradient of which, is proportional to the diffusion flux of random walkers. Because the walkers are absorbed only on the perimeter, the flux, $\underline{v}$, has zero divergence ($\underline{v} \propto \nabla u$, $\nabla \cdot \underline{v} = \nabla^2 u = 0$). The growth of the aggregate is given by the flux at its surface.

The varied physical systems of; solidification, electrodeposition, fluid-fluid displacement, and aggregation, under appropriate approximations, all share similar interfacial growth equations and morphologies. The corresponding control variables for these systems are; undercooling, applied voltage, pressure, and concentration. For example, in electrodeposition, the potential is the electric potential, V, where the growth rate is proportional to the electric field, $\underline{E}$, at the surface of the deposit ($\underline{E} \propto -\nabla V$, $\nabla \cdot \underline{E} = 0$, and $\nabla^2 V = o$).

EXPERIMENTAL REALIZATIONS OF THE MODEL

Electrodeposition

Using a polymer to raise the viscosity of the copper sulfate electrolyte so as to inhibit the mixing of the sulfate ions by convection, and an added excess of sodium sulphate to screen the electric field, Brady and Ball (1984) deposited copper in which growth was limited by diffusion of Cu^{2+} ions. The radius of deposit was proportional to the diffusion-limited current and the mass was obtained from Faraday's law. The inferred fractal dimension obtained was 2.43 ± .03 which is in agreement with three dimensional simulations of DLA.

Two dimensional zinc leaves were grown by Matsushita *et al.* (1984) and their two-point correlation function was obtained by digitized image analysis. The deposits grew in an interfacial layer between a zinc sulphate solution and a covering of n-butyl acetate. Because the applied voltage was low, the growth process was controlled by the electrical potential field, obeying Laplace's equation. The fractal dimension obtained was 1.66 ± .03.

Hydrodynamics

Hele-Shaw cells consisting of two parallel plates where a low viscosity fluid, is injected into a high viscosity fluid have been used as analogs for fluid flow through homogeneous porous media. By Darcy's law, the local fluid velocity is proportional to the pressure gradient, and for an incompressible fluid, the fluid potential field obeys Laplace's equation. Paterson (1984) was the first to point out the similarities between the viscous fingers produced by the Saffman-Taylor instabilities and the patterns of DLA. He speculated that they should also scale like DLA.

Daccord *et al.* (1986) used water as the driving fluid and a high viscosity polymer for displaced fluid. The boundary conditions agreed with those of DLA because the viscosity of the water was negligible which allowed the approximation that the interface be isobaric. However, the polymer was non-Newtonian and its shear thinning introduced a non-linearity which was accounted for by using a power function of the pressure gradient. The fractal dimension was measured using various methods which produced consistent results of, 1.70 ± .05.

Dielectric Breakdown

Lichtenberg figures are the electrical discharge patterns formed by the conduction channels during dielectric breakdown. Niemeyer (1984) assumed that the breakdown channel is a good enough conductor to be regarded as an equipotential and that further breakdown or growth of the breakdown channel is proportional to the surrounding electric field (or the gradient of the electric potential). Under these crude approximations the electric potential obeys Laplace's equation with similar boundary conditions as

DLA. In compressed SF_6 gas, the surface discharge on a plate of glass was analyzed and a fractal dimension of 1.7 was found from digitized photographs.

CHAPTER III

IMPLEMENTATION OF THE MODEL

Various modifications to Meakin's improvements on the original Witten-Sander model were made due to machine limitations and the desire to have real-time graphics display. (For more extensive discussion of these modifications see the Appendix A.) The most notable of these is the modification of the interfacial boundary conditions. In consideration of memory and speed limitations, the growth interface or exterior perimeter was not stored separately from the aggregate as it was grown. Consequently, the deposition rules at the interface were changed so that the pixel was deposited only when it attempted to 'jump' into the aggregate and not when it was on its interface. Thus interfacial transport was allowed and the deposition probability as a function of the velocity relative to the interface, P(v), was as follows:

$$P(-v_{NORMAL}) = 1$$
$$P(+v_{NORMAL}) = P(\pm v_{TANGENTIAL}) = 0. \qquad (8)$$

Deposition occurred at the site from where it attempted to 'jump' into the aggregate. As the pixel was only allowed to single step while inside the deposition zone, $R \leq R_{MAX} + 5$, and because the steps were along the orthogonal lattice directions, the possibility of the pixel 'jumping' over a deposit filament was eliminated.

In Meakin's model the deposition forces acted over a distance of one pixel diameter, since deposition occurred as soon as the pixel entered the one pixel thick perimeter. This is in contrast to the contact forces of the model used in this study, which allowed the pixel to move tangentially along the interface until an attempted 'jump' caused the centers of the pixels to coincide. In this sense, the present study deals with aggregation of points and ignores the excluded volume effect, whereas Meakin's model aggregated extended pixels of one lattice spacing in diameter. Consequently, the surface variations on the order of a lattice spacing were not smoothed over, which was an effect of the overlapping of the surrounding perimeter layer in Meakin's model. Thus, pixels could enter into cavities with entrances of one pixel in diameter and there be deposited. However, this modification did not significantly change the fractal dimension, which is a measure of the local deposit density or compactness.

The growing aggregation was surrounded by a 'birthing' circle which injected the random-walking pixels at a distance of $R_{INJECT} = R_{MAX} + 5$ lattice spacings away from the initial center seed. The release was randomized over half-degree increments around this circle. If the pixel was outside of this circle the step size was scaled as follows:

if $10 \cdot 2^N < R - R_{MAX} < 10 \cdot 2^{N+1}$ then stepsize = 2^{N+1}. The random walk was continued until deposition occurred or until the pixel was terminated on the 'killing' circle of radius $R_{KILL} = 2 \cdot R_{MAX} + 5$. This modification was made to expedite the deposition process.

To complete the description of the model, it should be noted that, although, there were toriodial boundaries (remnants from a previous demonstration program, from which the simulation program evolved), they were never reached because the growth terminated when the aggregate reached a radius of 200 lattice spacings. This constraint was devised to insure that the whole aggregate could be displayed. The center seed was located at (200,200) in the screen space. The coordinates of the seed in the simulation space (a Boolean array in main memory) were (408,408) with boundaries at 3 and 812 in both x and y. Although, larger aggregates could have been grown, their growth times would have been excessive and it would have been necessary to partition their displays. (For a more complete discussion of the memory and time constraints, see Appendix A.)

Initially, 26 small aggregates were grown using the demonstration program which stopped growth when the 'birthing' circle reached the edge of the screen at R = 200 lattice spacings. These small aggregates were then used as 'seeds' in the simulation program which allowed for larger growth. A total of 30 large aggregates were grown.

CHAPTER IV

SIMULATION RESULTS AND DISCUSSION

NUMERICAL RESULTS

The output from the simulation program consisted of two files which were stored on disk. The spatial deposit array was stored as a sequential file in the order of deposition. The screen buffer was also stored as a binary file so that screen sites could be later checked for deposition. These files were processed by programs to obtain the fractal dimension from the correlation function and the radius of gyration. (For more extensive discussion of these programs see Appendix A.)

The correlation program actually consisted of three separate programs, each of which calculated the correlation function using circular and square 'windows', and from its dependence on the 'window' size, the fractal dimension was determined for each aggregate. The first of these programs used circular 'windows' which accumulated the enclosed pixel area by a polygonal approximation which in effect included the pixel area as either inside or outside the 'window'. This approximation technique affected only those pixels which were on the perimeter of the 'window'. This correlation function was evaluated at all the deposits comprising the aggregate. The second and third programs excluded those pixels located at radii, $R > R_{MAX} = 32.5$ lattice spacings as, 32.5 was the largest window size. Because the edge of the growth was where deposition was most active, it was thought that by excluding the edge from consideration, the fractal dimension obtained would be more representative of the complete aggregate. The third correlation program utilized a look-up table of the exact areas for those pixels that were bisected by the perimeter of the circular 'window'. The 'window' sizes for all the programs were $2^N + .5$ lattice spacings, $N = 0,1,2,3,4,5$. All the correlation programs were tested for accuracy by evaluation of the fractal dimension of compact Euclidean figures.

The radius of gyration program used the lattice origin and not the center of mass of each aggregate to compute the radius of gyration. The calculation of the center of mass at each deposition would have greatly increased the process time. Furthermore, it was assumed that any offset would not be appreciable. If it was appreciable, it would distort the numerical results in a complicated manner.

Correlation Function Results

For each aggregate, the results of the dependencies of $Ln(C(r))$ on $Ln(r)$, and $Ln(R_g)$ on $Ln(N)$ were analyzed by linear regression to give the corresponding fractal dimensions. The individual results are given in Appendix B.

Each of the 26 small aggregates served as a seed for the growth of the large aggregates. The correlation results of all the individual aggregates were averaged by a separate least squares analysis of the average results of each 'window'. The average fractal dimension, as determined from the radius of gyration, was determined by processing a composite of all individual growths. (This composite was also utilized in the determination of the frequency histograms, which are discussed below under Graphical Results.) These results are listed in the following table.

TABLE I

AVERAGE FRACTAL DIMENSIONS

<u>Fractal Dimension from Average Correlation 'Window' Data</u>

	Including Edge			Excluding Edge	
	Squares	'Circles'	Squares	'Circles'	Circles
Small Aggregates					
D	1.66410592	1.610013451	1.6953093637	1.6393097109	1.6962591969
s.d.	.0082032213497	.0079478124734	.012463779431	.011922381434	.012101511144
Large Aggregates					
D	1.6668462298	1.6107480877	1.6725249781	1.6160897292	1.672937113
s.d.	.0053549107253	.0050211512165	.0058509194604	.0056517456068	.0057074275171

<u>Fractal Dimension from Composite of all Aggregates based on Radius of Gyration</u>

<u>Small Aggregates</u> 1.8452894007 <u>Large Aggregates</u> 1.8120055785

<u>Average Aggregate Size</u>

<u>Small Aggregates</u> N = 4510 ± 702 pixels <u>Large Aggregates</u> N = 16298 ± 2159 pixels

Polygonal approximation of the circular 'windows' was utilized to expedite implementation. Circular 'windows' which computed the exact areas were justified in so far as the correlation function utilized the Euclidean metric. Furthermore, in a statistical sense, the aggregates tended to have a circular symmetry. It had been for computational convenience that Forrest and Witten used square 'windows' to determine the correlations of smoke particles. However, the underlying square lattice geometry also suggests the utilization of the more natural square 'windows'. In the absence of an adequate discussion of this issue in the literature, it will now be discussed as to whether these computational schemes yielded significant differences of the resulting fractal dimension.

The average fractal dimensions which were obtained by using the correlation function with circular 'windows' and by excluding the edges of the aggregates, were, as follows: for the small aggregates, polygonal approximation gave results of $D_{\cdot c \cdot} = 1.639 \pm .012$ and exact calculation yielded results of $D_c = 1.696 \pm .012$. For the large aggregates, results were, $D_{\cdot c \cdot} = 1.616 \pm .006$ and $D_c = 1.673 \pm .006$, respectively. Therefore, the polygonal approximation is not justified.

Comparison of the results obtained from the correlation function by using exact circular and square 'windows' and by excluding the edges of the aggregates, indicates that the choice of method is arbitrary. Specifically, the fractal dimensions which were obtained for the small aggregates were, for circular and square 'windows'; $D_c = 1.696 \pm .012$ and $D_s = 1.695 \pm .012$, respectively, and for the large aggregates the dimensions were identical, $D_c = D_s = 1.673 \pm .006$. Whether structural symmetry or the underlying lattice geometry alter the fractal dimension, as determined by this correlation function, can not be decisively concluded on the basis of this analysis. Other correlation functions and scaling relations could be formulated to address this issue more conclusively.

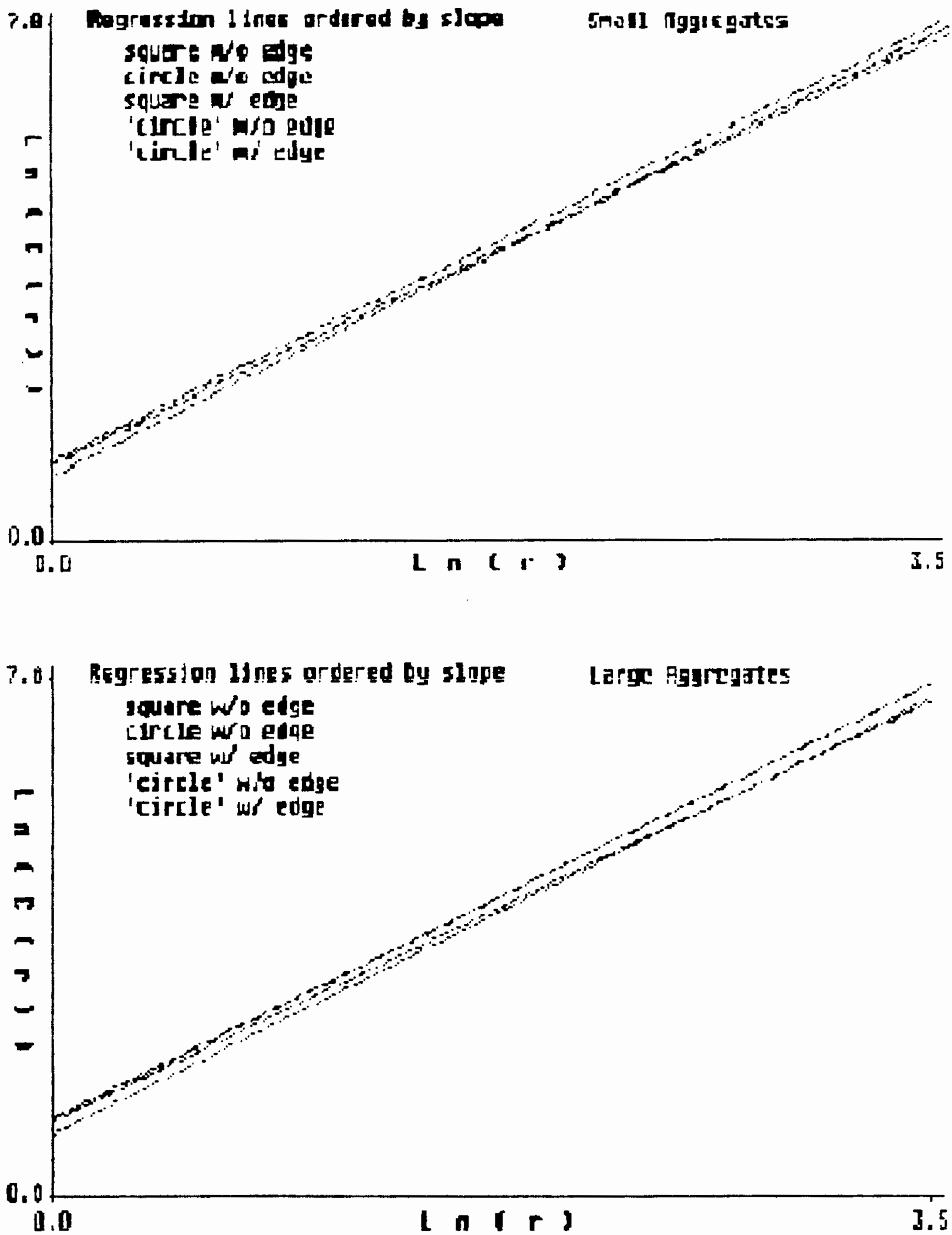
The effect of screening on deposition is evident by the decrease of the average fractal dimensions, computed where edges are excluded, as the aggregates become larger. Comparison of the corresponding average fractal dimensions between the small and large aggregates must take into account that the individual large aggregates were grown from individual small aggregate seeds and not independently, each with a particular fractal dimension and growth trend based on its structure. However, because the analysis is based upon the average fractal dimensions, (which suppress any particular trend that an individual aggregate may have in terms of its fractal dimension), it is valid for comparing the change in the fractal dimension between the average small aggregate and the average large aggregate. Because the excluded edge is 32.5 lattice spacings for both the small and the large aggregates, the proportion of the region of active deposition that is excluded, is greater for the small aggregates than for the large aggregates. Conversely, proportionately more of the inactive interior region (which is more compact and thus has a greater fractal dimension) is used in the correlation calculation that excludes the edge for the small aggregates rather than for the large aggregates. (Screening, and the active deposition zone, are more fully discussed in the Graphical Results section.)

The average fractal dimensions computed by not excluding the edges of the aggregates and by using the correlation function using square 'windows' are; for the small aggregates, $D_s = 1.664 \pm .008$, and for the large aggregates, $D_s = 1.667 \pm .005$. The difference in these fractal dimensions is not significant, and is not inconsistent with the above analysis. Furthermore, it suggests that the active zone also scales as a fractal.

The sequence, of the average fractal dimensions, obtained by using the various correlation function schemes, (presented in Table I), is consistent between the small and large aggregates. This is illustrated in Figure 3, on both the graphs for the small and large aggregates, where the slopes of the regression lines are listed in decreasing order. The regression line, for the rejected scheme using polygonal approximation, is skew to those regression lines for the exact schemes. The coincident regression lines for the exact schemes; where the edge is excluded, are parallel to the regression line for the scheme using exact squares, where the edge is included; is true for large aggregates and not for the small aggregates. The regression lines have different intercepts simply because edge deposits were excluded. The average fractal dimensions, calculated by the exact schemes, for the large aggregates, yield the fractal dimension of D = 1.67 ± .01. However, the corresponding results, for the small aggregates, do not agree within statistical uncertainty. Further analysis of the average dimensions, between the small and large aggregates, of all the exact schemes, indicates a convergence, as the aggregates become larger, toward the results given by the scheme using squares, and where the calculations included the edge. This convergence is also supported by the agreement between the average fractal dimensions of the small and large aggregates, which are produced by the scheme where the edge is included and the correlation function utilizes squares. This agreement also yields D = 1.67 ± .01. This suggests that, to fully characterize a growing aggregate, an additional fractal dimension for the zone of active deposition could be utilized.

The sequence of the fractal dimensions, obtained by the various correlations schemes, is further illustrated in

Figure 4. The graphs of the results for the individual small and large aggregates do not intersect, indicating that the consistency of the schemes is not dependent upon the averaging process.



<u>Figure 3</u>. Correlation dependence on 'window' size.

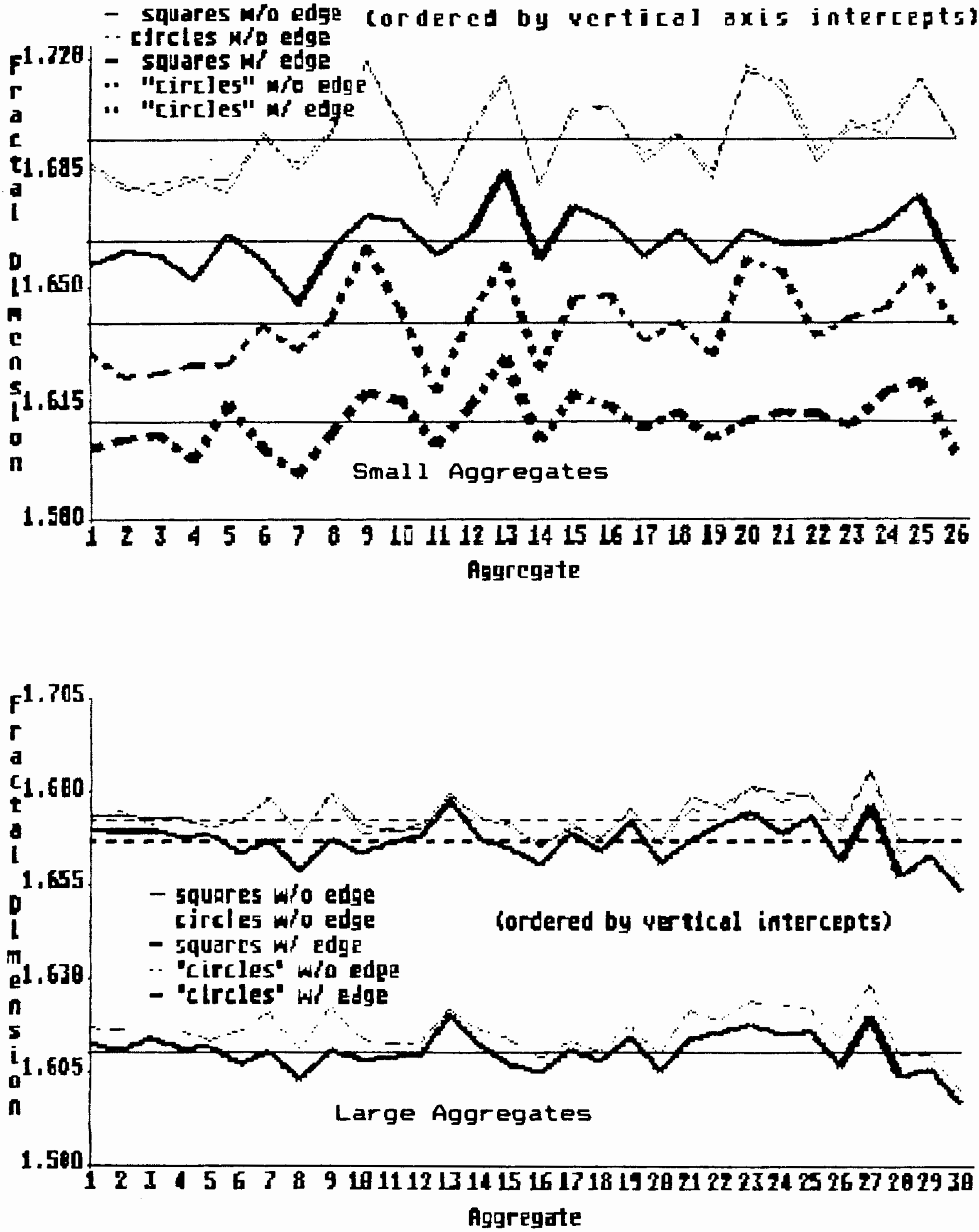


Figure 4. Fractal dimension vs. aggregate number.

<u>Radius of Gyration Results</u>

The results from the radius of gyration, R_g, dependence on the number of deposits, N, reported in Table I, are not in immediate agreement with the results discussed above concerning the correlation function, C(r), dependence on the 'window' size. In further contrast, are the fractal dimensions reported by Meakin, which do agree. (These were similarly related to the slopes of the graphs of $Ln(R_g)$ vs. Ln(N) and Ln(C(r)) vs. Ln(r).) The fractal dimensions, calculated from the reciprocals of slopes of the graphs of $Ln(R_g)$ vs. Ln(N), were determined from composites of all the small and large aggregates, over the entire ranges of N. Time did not allow for an estimation of the statistical uncertainties associated with the listed fractal dimensions, even though this would have required only minor modifications to the least squares routine in order to obtain the standard deviation of the regression coefficient. However, inspection of any of the $Ln(R_g)$ vs. Ln(N) graphs in Appendix B, indicates that the graphs for the individual aggregates are not initially linear and only appear to asymptotically become so with increasing N. However, due to the condensed size of the graphs, this interpretation may not be valid. The non linear region of the graphs, for small values of N, indicates that the aggregates are initially random, and that their structure stabilizes and becomes fractal with more deposition. This corresponds to the apparent linear portions of the graph. As an aggregate becomes larger, a deposit's perturbation of the global geometry is diminished. With the average large aggregate size of only N = 16298, it is unknown whether the fractal dimension also has an upper cut off, above which the aggregate becomes non-fractal, or its dimension approaches another value. It was hoped that the averaging of the individual aggregates into a composite would damp the initial transients and the graph would be linear over its entire range. Indeed, at a first glance, the graphs in Figure 5, appeared to indicate this result. However, when the regression was parameterized by a lower cut off, the resulting fractal dimensions did not stabilize, in fact, the results, as shown in the chart overlaid on the graphs, indicate that the graphs are actually slightly concave. This is in accord with the effect of screening by the perimeter. As the aggregate grows the perimeter effectively leaves behind it a region 'frozen' at an intermediate fractal dimension. Deposition, when penetration is restricted, tends to increase the radius of gyration more because it occurs, on the average, at a greater distance. A more thorough study of this concavity and asymptotic growth would require an analysis of the scaling properties of the zone of active deposition. The results which suggest the concavity may lack statistical significance, as the maximum graphical error for the graph of the large aggregates is only ≅ 2%. Furthermore, the curve tends to oscillate, which indicates that the graph can be regarded as linear. The use of the upper endpoint, with the parameterized lower cut offs in the linear regression, may not accurately determine the fractal dimension for the average mid region of the aggregates because it tends to attach more statistical weight to the active zone. A separate correlation function analysis of the active zone would determine whether the active zone had a smaller or greater local density than the mid region of the aggregate. Even without this separate analysis, it may be inferred that the active zone had a smaller local density than the mid region of the aggregate. This inference is drawn from an analysis of the results of correlations over the entire aggregate, between those which exclude and those which include, the edge. (These results are listed in Table I.) The question arises, of whether the reported results should represent just the global properties of a stabilized and relatively large aggregate, or whether they should also include the residual effects of its incipient growth. Utilizing the results for an average 'mature', yet growing aggregate, the fractal dimensions are, for small aggregates, $D = 1.799$, and for large aggregates, $D = 1.773$. In acknowledgement of the uncertainties involved, and of the apparent inverse nature of the growth of the aggregate and its fractal dimension, the final result, using the radius of gyration is, D = 1.78 ± .01. This does not agree with the correlation function results. The relative discrepancy is ≅ 6.6%. The radius of gyration program could be flawed, as there is no obvious explanation for the discrepancy between the two methods (The averages of the individual aggregates,

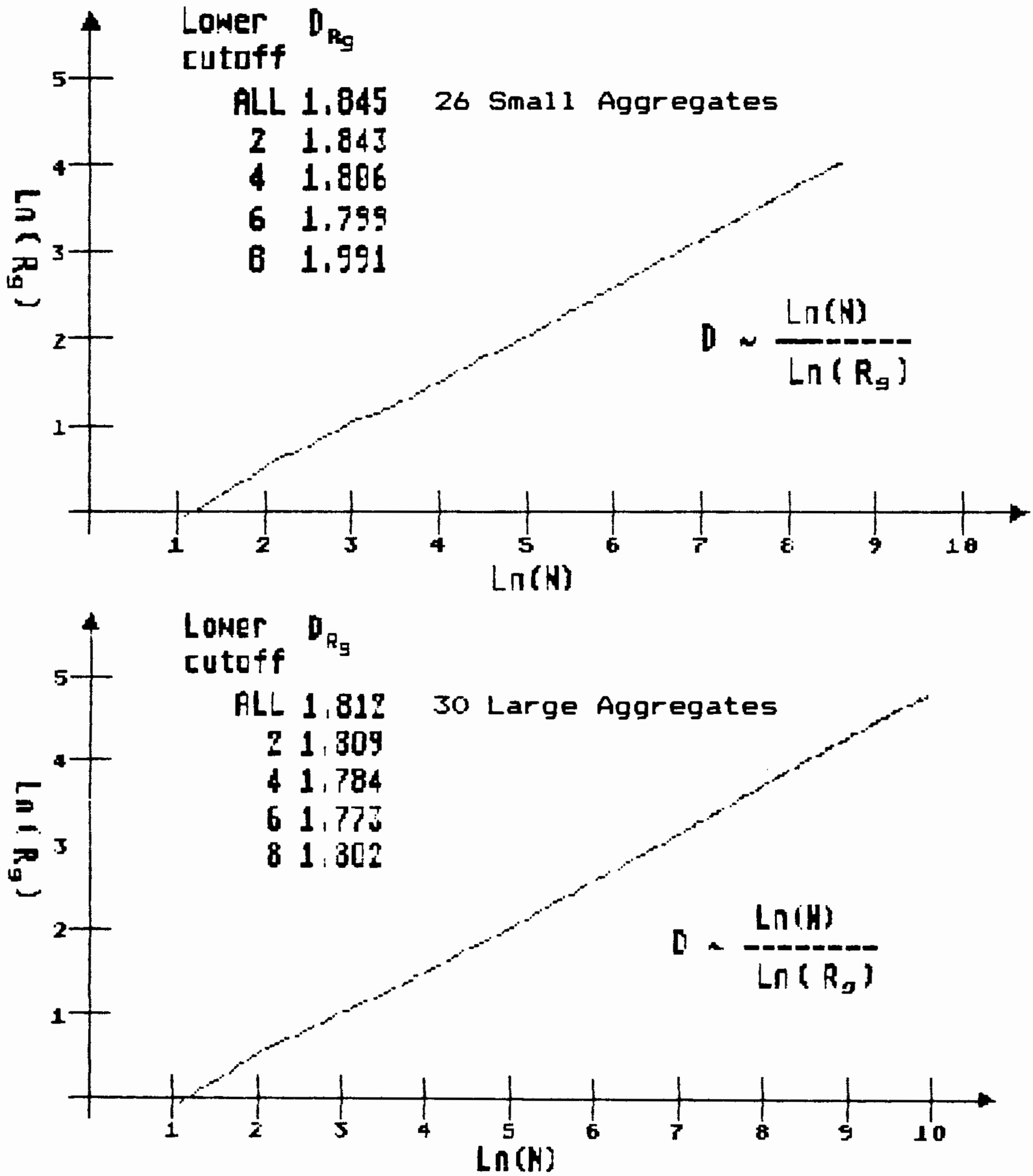


Figure 5. Radius of gyration dependence on number of deposits for small and large aggregates.

without cut off, are, for small aggregates, $D = 1.84 \pm .07$, and for large aggregates, $D = 1.80 \pm .05$.)

GRAPHICAL RESULTS

This section discusses the graphical depiction of the aggregates. The graphical output for all the aggregates are found in Appendix C. It is evident that the aggregates represent a diversity of structure, yet a recognizable pattern is discernable. However, without the fractal dimension, only a qualitative description of this pattern is possible. However, aside from the pattern, other characteristics can be demonstrated. Symmetries and anisotropies were investigated by the use of frequency histograms. The dynamics of growth were studied by use of animation programs, the results of which were distilled into the series of images depicting the evolution of growth. Additionally, the animation programs were used to construct a sequence displaying the depth of penetration at varying stages of screened growth. Aggregate number 20 was selected as a representative aggregate and its characteristics are presented (Figure 6.). A similar presentation follows for the composite of all the large aggregates. The extent that subsequent growth depends upon initial conditions and the persistence of growth trends are studied by the comparison between two of the large growths, which were grown from the same small growth.

The most salient features are the radial symmetry and the similarity of branching structure ramified over different orders of magnitude. Predicting its occurrence and structure in terms of natural ratios of characteristic lengths, such as arm diameters and interarm distances, unfortunately, was not relevant to the present study, although it certainly merits further study.

Examination of the growth stages of aggregate number 20, in Figure 7, indicates that the initial pattern of the

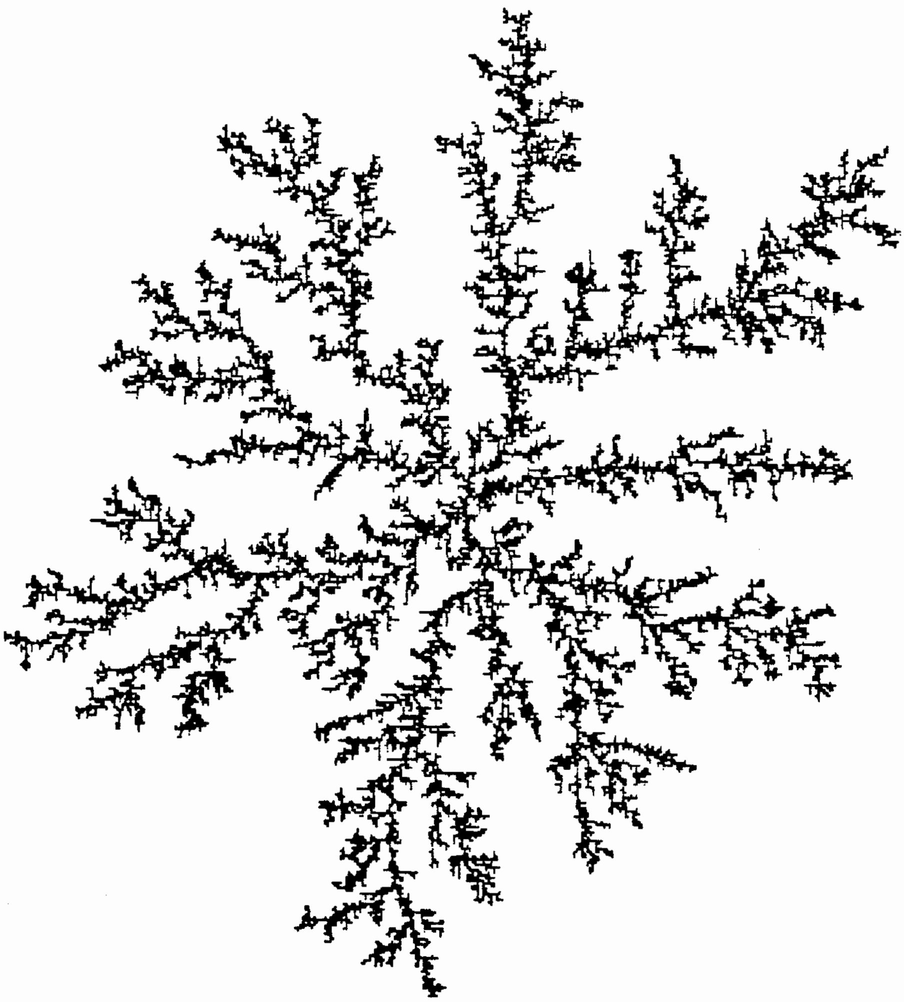


Figure 6. Aggregate number 20.

main branches is propagated, and persists in, the more intricate stages of later growth. The $Ln(R_g)$ vs. $Ln(N)$ graph for this aggregate is presented in Figure 8. The transients of the initial growth are visible in the oscillations of the lower portion of the graph. The frequency histogram of the radial mass distribution is presented in Figure 9. The presence of 'holes' is indicated by the increasing portion of the histogram. Growth was terminated before uniformity in the distribution for the mid region of the aggregate could be ascertained.

The radial symmetry is manifest in the outward growth of the arms. The angular distribution, as shown in its frequency histogram in Figure 10, indicates that the arms 'sweep up' the incident flux of random walkers. The flux is assumed to be uniform and isotropic. (The unsmoothed data for aggregate number 20 is given Appendix C.)

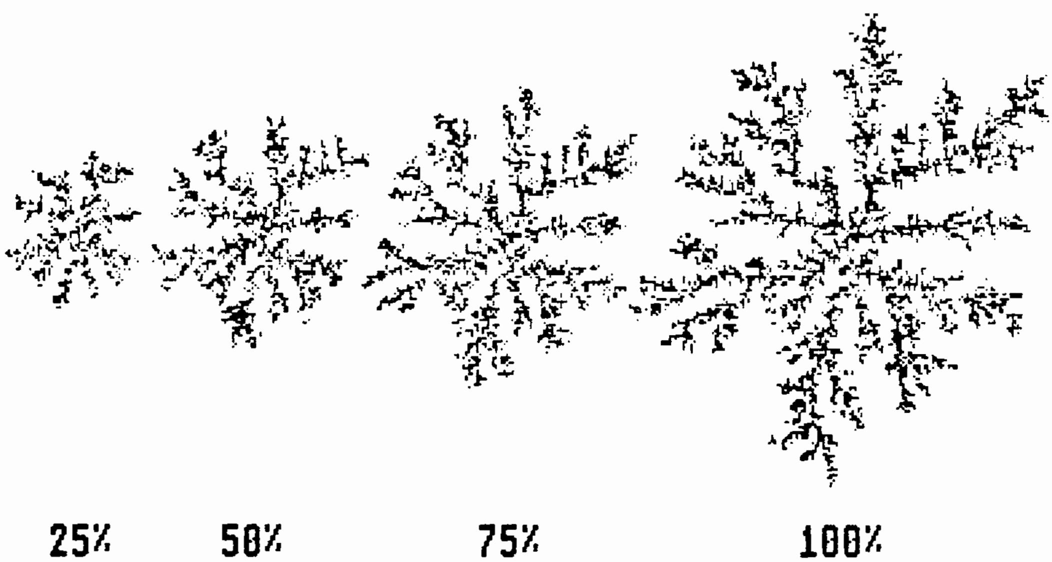


Figure 7. Quartile stages of growth of aggregate number 20.

Because the deposit's diameter, lattice spacing, and step size, prior to deposition, are identical, it is improbable that any periodicities in the X and Y directions would be detected in the histograms for these coordinates.

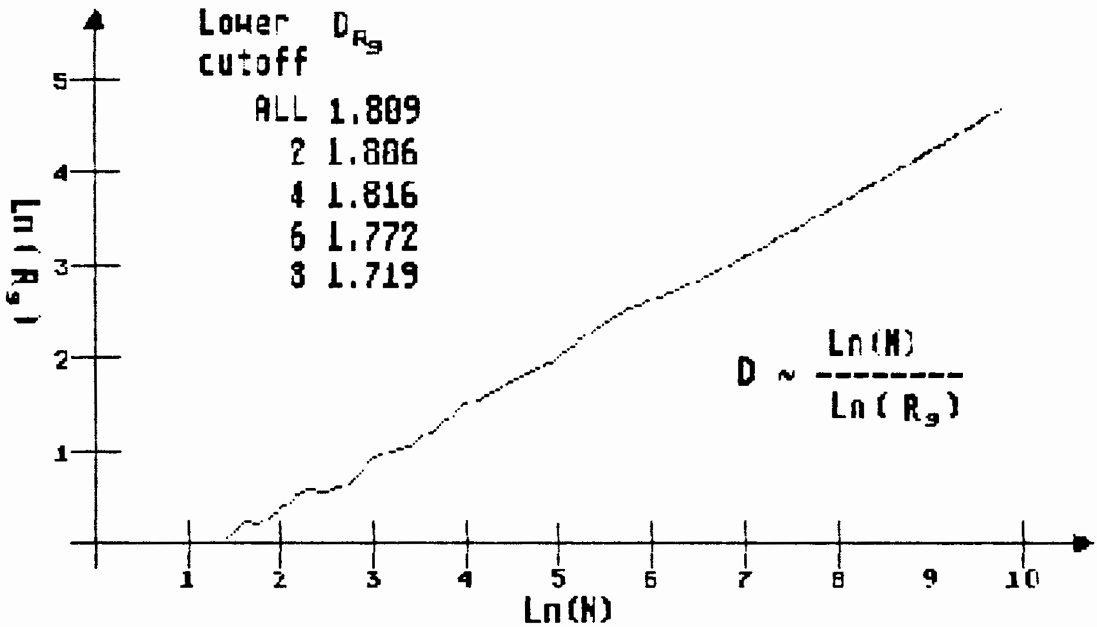


Figure 8. Radius of gyration dependence on number of deposits for aggregate number 20.

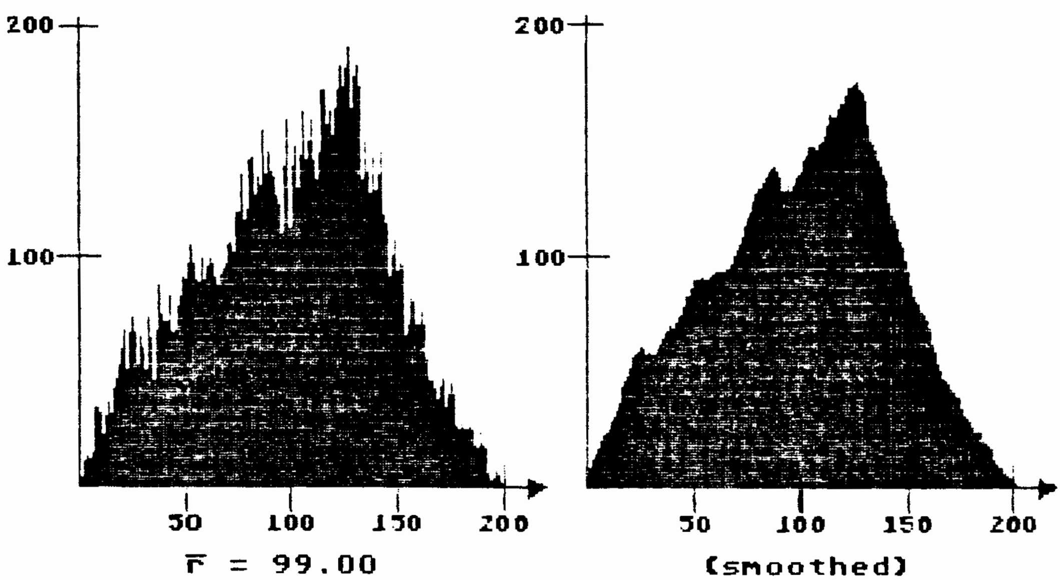


Figure 9. Radial mass distribution for aggregate number 20.

These distributions, presented in Figures 11 and 12, are not uniform due to the interaction between the arms and the deposition process. (Comments concerning the averages of these distributions are presented below under the discussion of the cumulative distribution of the large aggregates.)

The effect of screening on the growth is depicted in Figure 13. The ultimate N % of the total deposits are illustrated, for N = 10,...,90. On the average, the deposition occurs in the outer and more active shell. However, occasionally, screening is incomplete and a random walker wanders deeply into a 'fjord' before coming to rest,

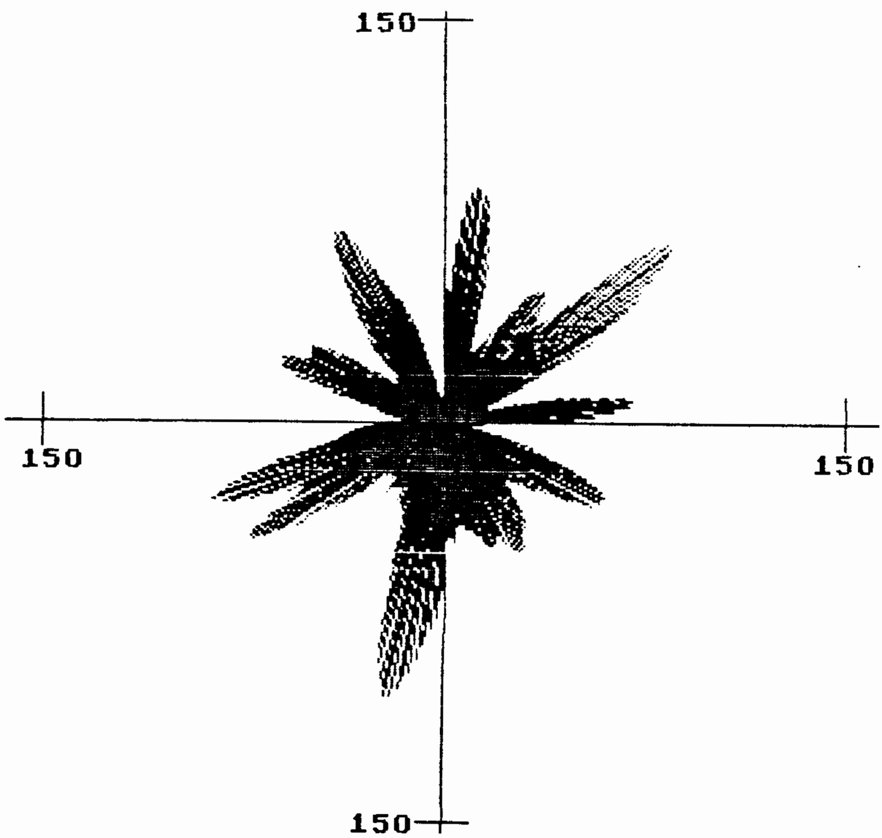


Figure 10. Angular mass distribution for aggregate number 20 (smoothed).

as shown by the stray deposits which have penetrated the interior. This screening process limits the 'filling in' of the interior, and growth continues in the outer shell. Subsequently, this active shell extends, by virtue of the deposition occurring there, leaving behind the incompletely 'filled in' interior of the aggregate, which is a fractal, rather than a compact structure.

Figure 14 examines the sample space of the cumulative probability distribution of the large aggregates for uniformity and isotropy of deposition. The suggestion of underlying arms, most discernable in those images labeled

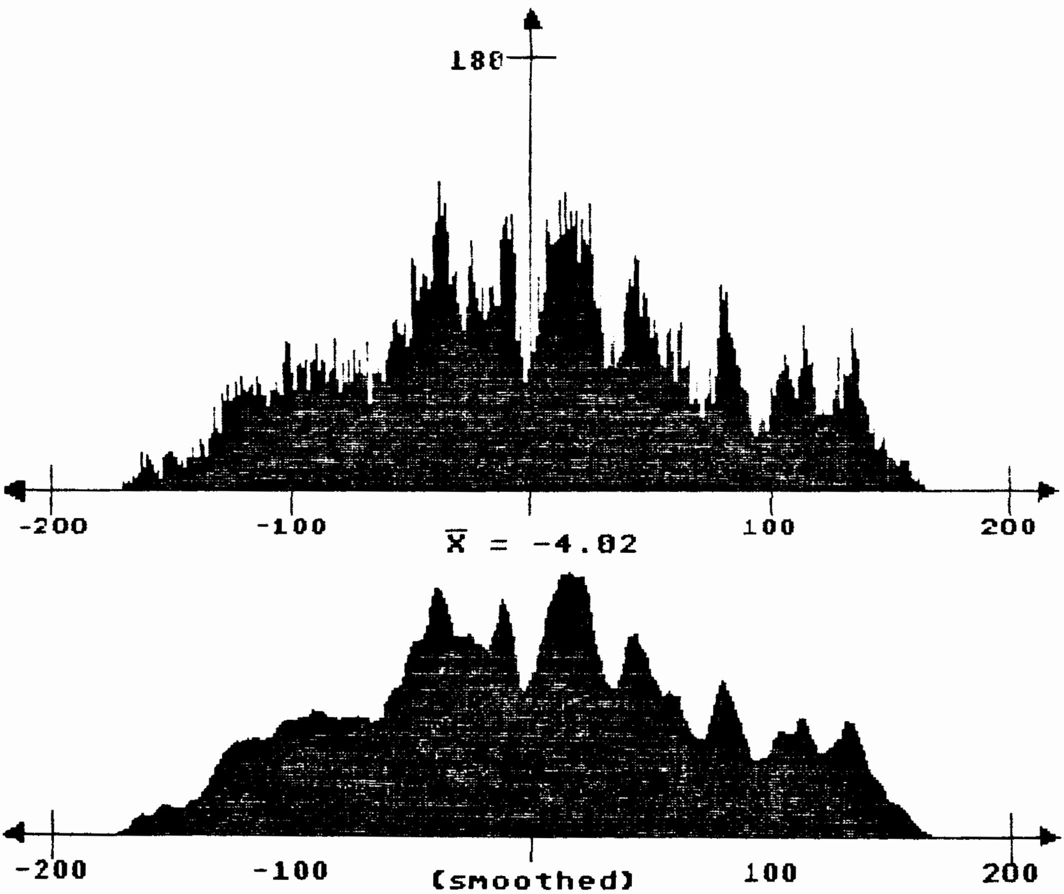


Figure 11. Mass distribution in X for aggregate number 20.

30% and 40%, (which are projections of the deposition distribution onto the XY plane, for $P(\underline{X}) \geq .30$ and .40) and the corresponding modes in the angular mass distribution of the large aggregates, which is presented in Figure 15, could be an effect of the lattice, if deposition was most probable along the orthogonal and diagonal directions of the lattice. Moreover, there does not appear to be any pattern associated with those sites which have not been deposited, except that they tend to be between those arms. The averaged growth appears to be uniform and radial because the perimeters of Figures 14 and 15 can be regarded as circular.

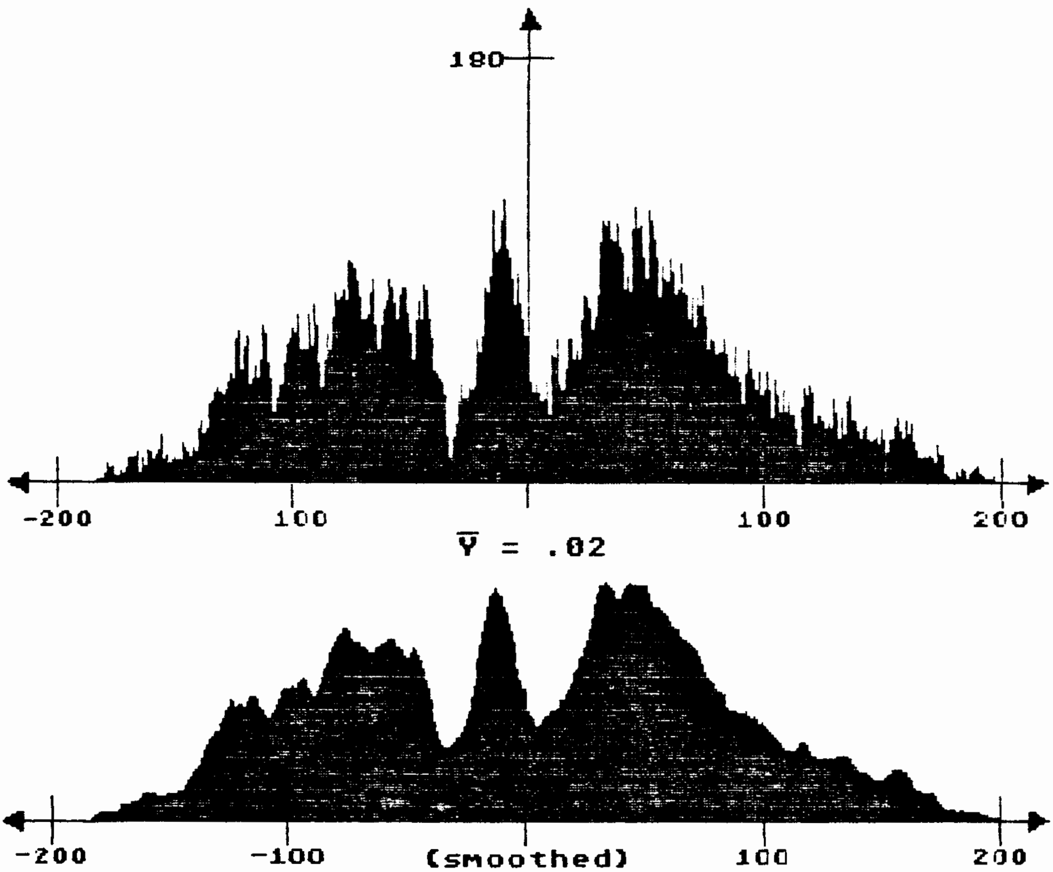


Figure 12. Mass distribution in Y for aggregate number 20.

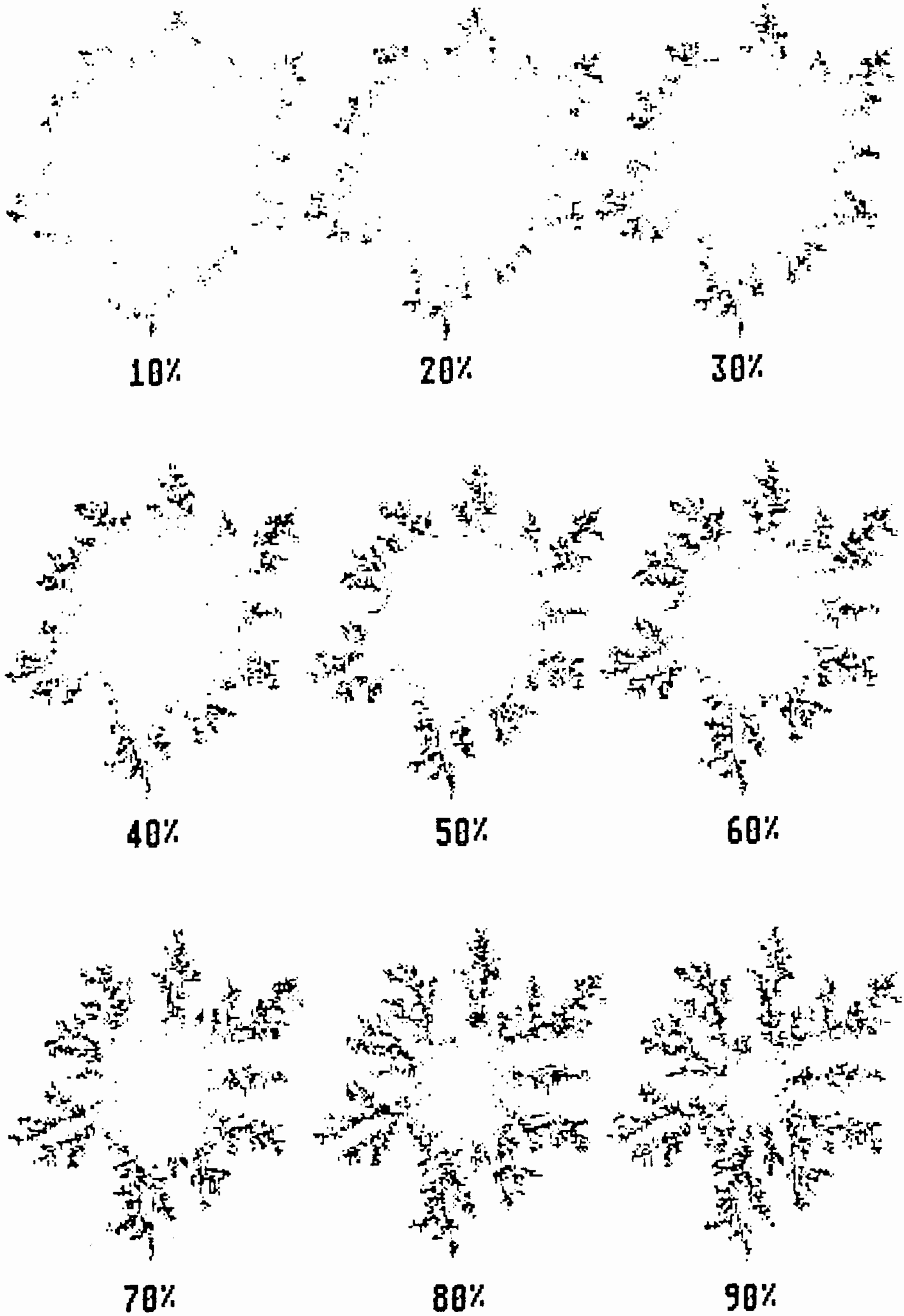


Figure 13. Upper percentiles of deposition for aggregate number 20.

The frequency histograms for the cumulative distributions in X and Y are displayed in Figures 16 and 17, respectively. The center of deposition is located at (3.47,-5.37). The center is 6.4 lattice spacings from the origin of the simulation. This result exposes a possible source of error in the fractal dimension based on the radius of gyration and is discussed at length in the Conclusion and Appendix D. Factors which might influence the displacement of the average center of mass, as accumulated over the

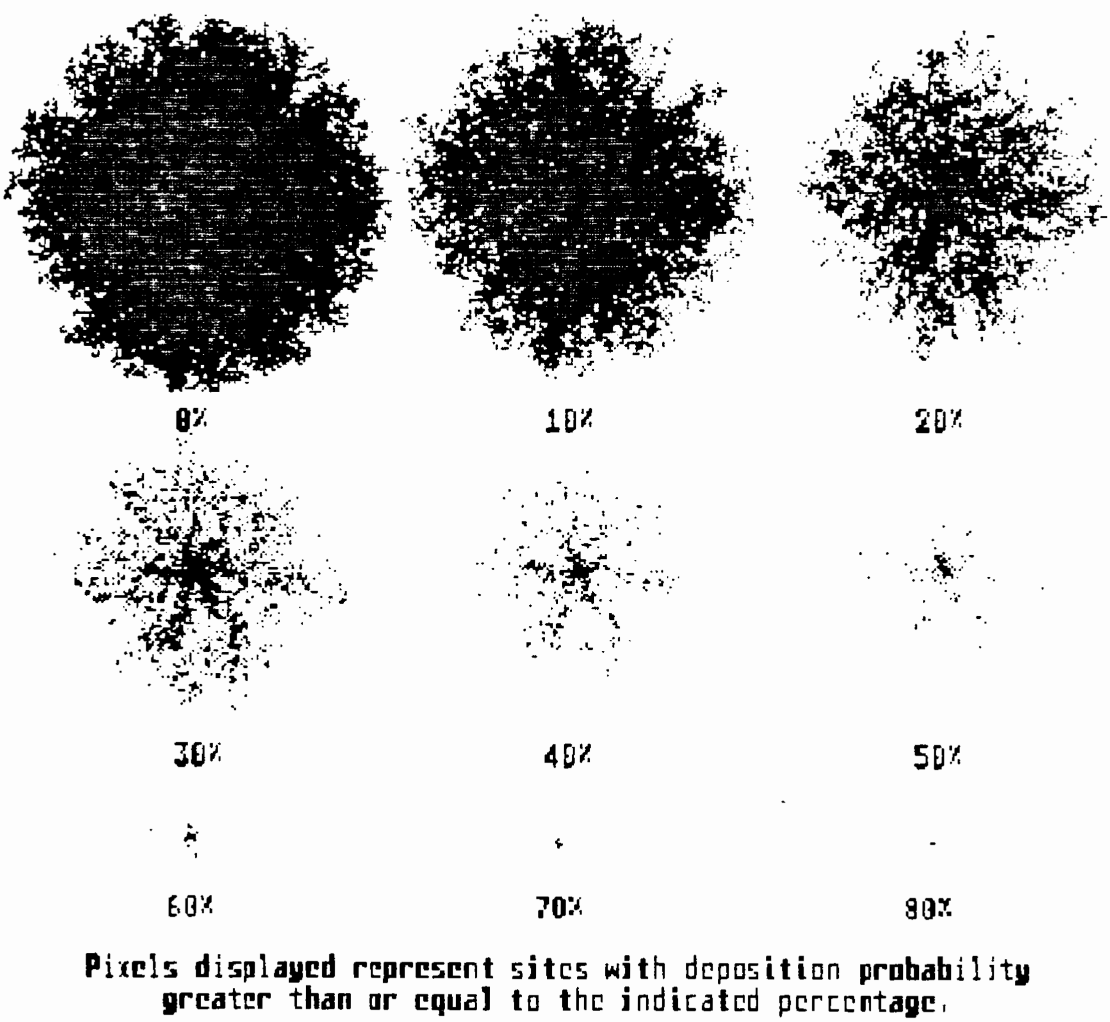


Figure 14. Cumulative probability distribution in X and Y.

relatively large sample of aggregates, are that the incident flux is not isotropic, that the deposition is preferential to certain orientations, or that growth is restricted in some directions. (The center of mass for any particular aggregate is expected to be displaced.) Because the graphics screen was dimensioned by even, and not odd integers, the lattice origin was slightly eccentric to the screen boundaries. Consequently, growth was terminated slightly more often when the maximum radius was in the fourth quadrant. However, this would explain the location of center of deposition in the second, and not in the fourth quadrant. Possibly, this asymmetry was caused by non-uniformity of the random number generator function. If it was biased towards higher values, the 'birthing' circle would have released a greater flux of random walkers into the fourth quadrant. Unfortunately, time did not allow for analysis of the random number generator. (This bias also would have caused anisotropy in the Brownian motion, which could have countered the above effect, because the leeward side of the aggregate would have obstructed movement and collected more deposition. However, not knowing the shape of the random number distribution, it is impossible to predict how the 'jump' procedures, which direct the movement, would have responded to the anisotropy.) The radial symmetry is indicated by the joint symmetry in X and Y, as shown in the histograms.

The frequency histograms for the radial distribution of the large aggregates, shown in Figure 18, are included for comparison to Figure 9. Because uniformity of deposition would imply that the aggregates would not be fractal, it is not to be expected. If the large aggregates are fractal, then the increasing portion of the histogram should exhibit power law dependence, specifically, r^D. That it departs from this is most probably due to occasional penetration into the interior. The decreasing portion of the histogram indicates that growth is incomplete and possibly that the active zone of deposition has different

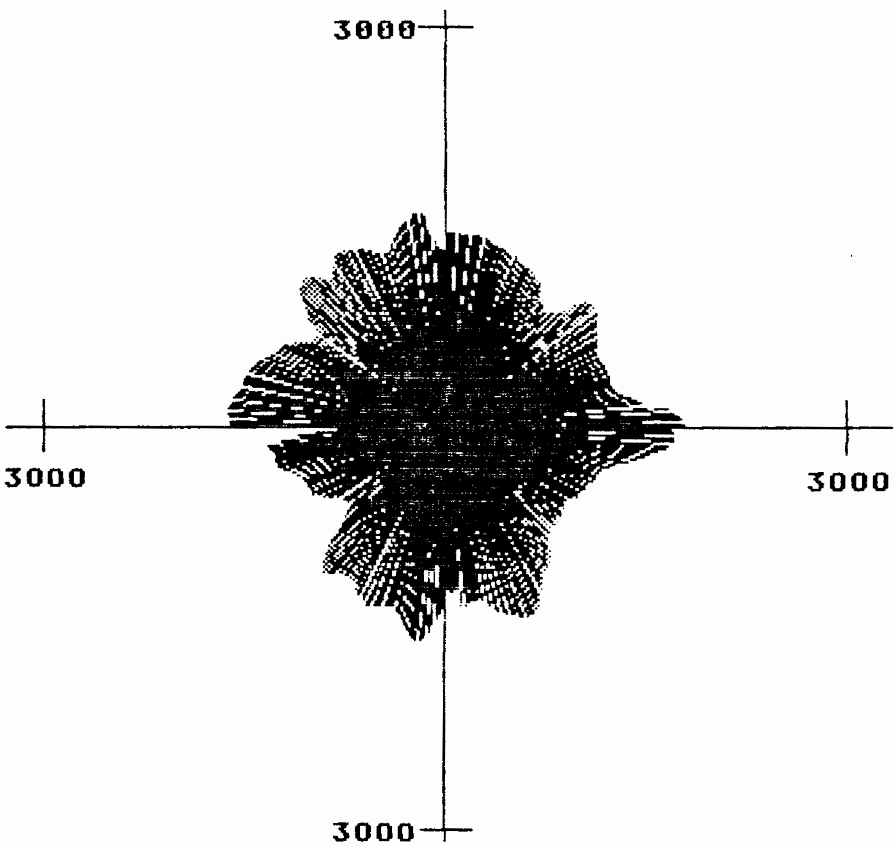


Figure 15. Cumulative angular mass distribution of the 30 large aggregates (smoothed).

scaling properties than the more complete interior region. However, its decreased inclination, as compared to Figure 9, is most probably the result of the averaging which occurred when the histogram was constructed from a composite of all the large aggregates.

Figure 19 depicts the dependence that subsequent growth has on initial conditions. The large aggregates, numbers 23 and 27, were each grown from the small aggregate, number 23. Even though the large aggregates are more than three times the size of the seed aggregate, the small aggregate seems to have imparted a general growth trend.

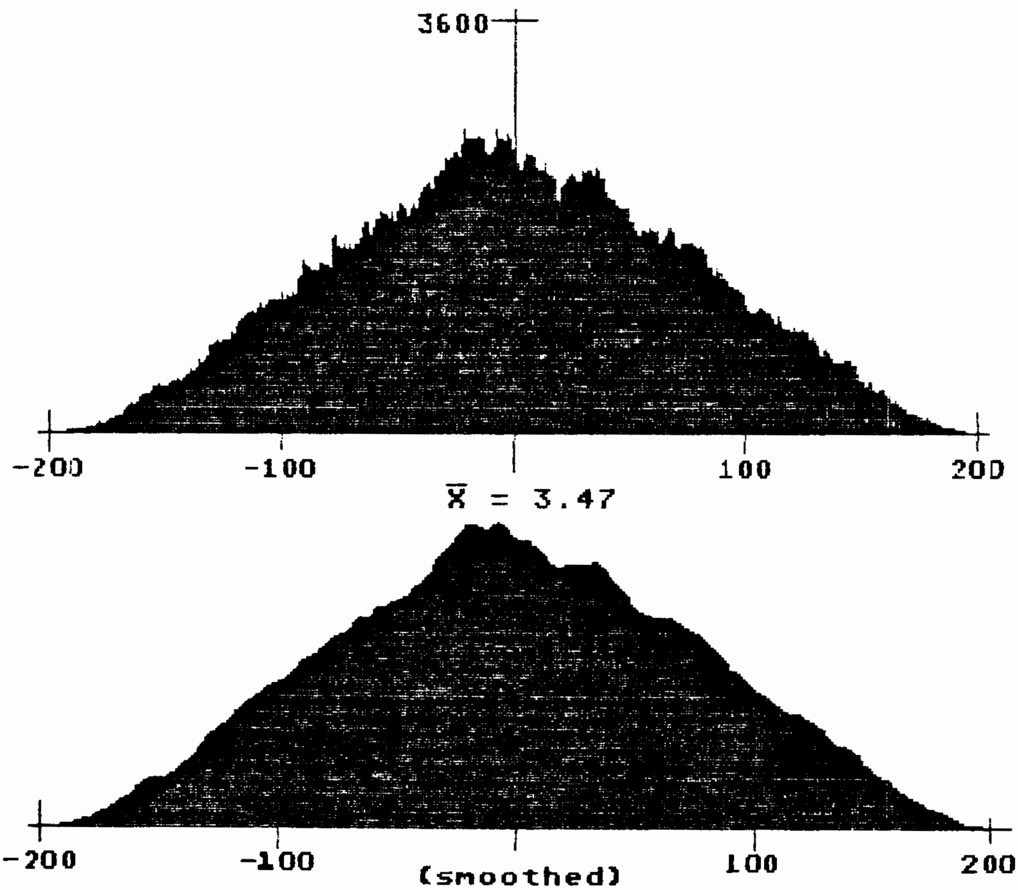


Figure 16. Cumulative mass distribution in X for the 30 large aggregates.

This similarity of structure between the two large aggregates persisted, even into regions beyond the scale of the original aggregate. The large aggregates were grown to sizes of 16464 and 19056 deposits, respectively. An investigation of the divergence of their morphologies with further growth was not performed.

All of the small aggregates were grown from a single featureless seed. Yet, each of the aggregates developed distinctly, with its own characteristic structure. The

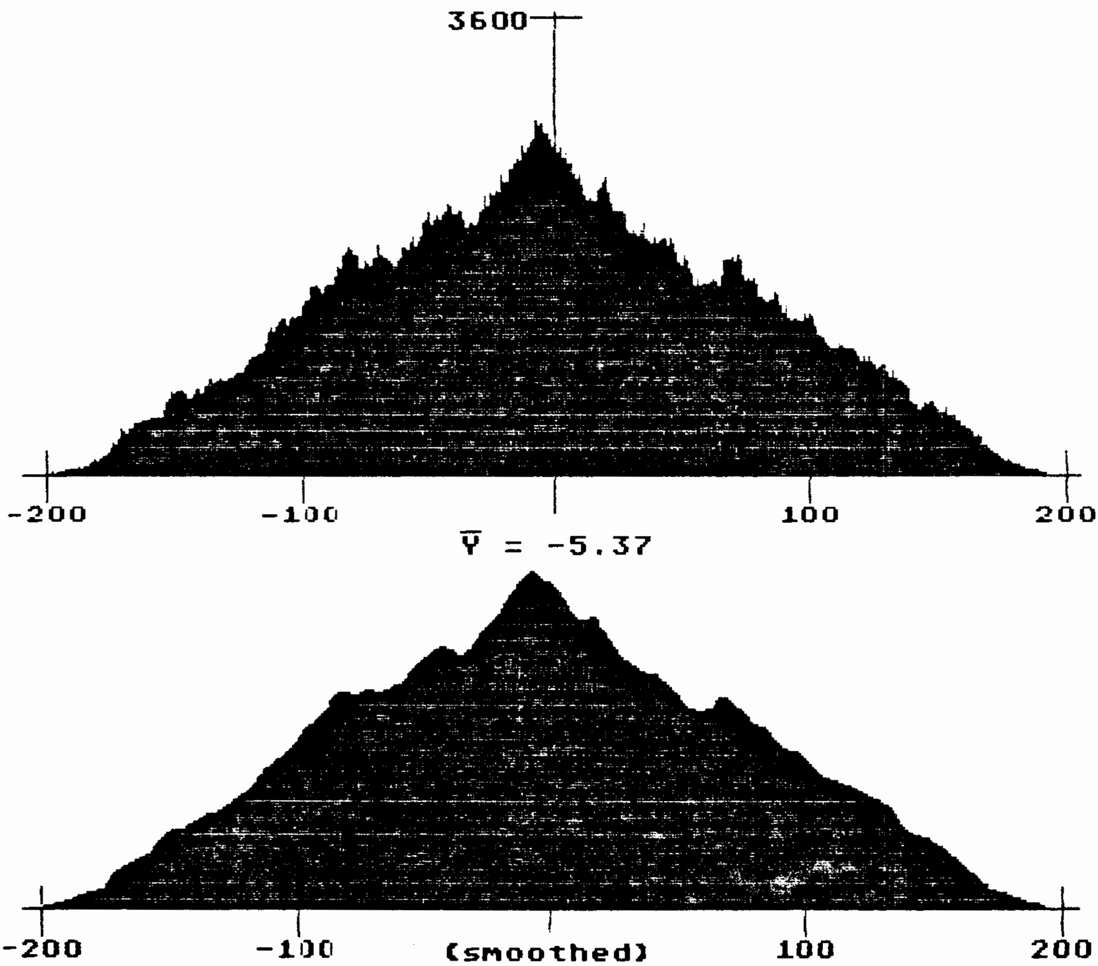


Figure 17. Cumulative mass distribution in Y for the 30 large aggregates.

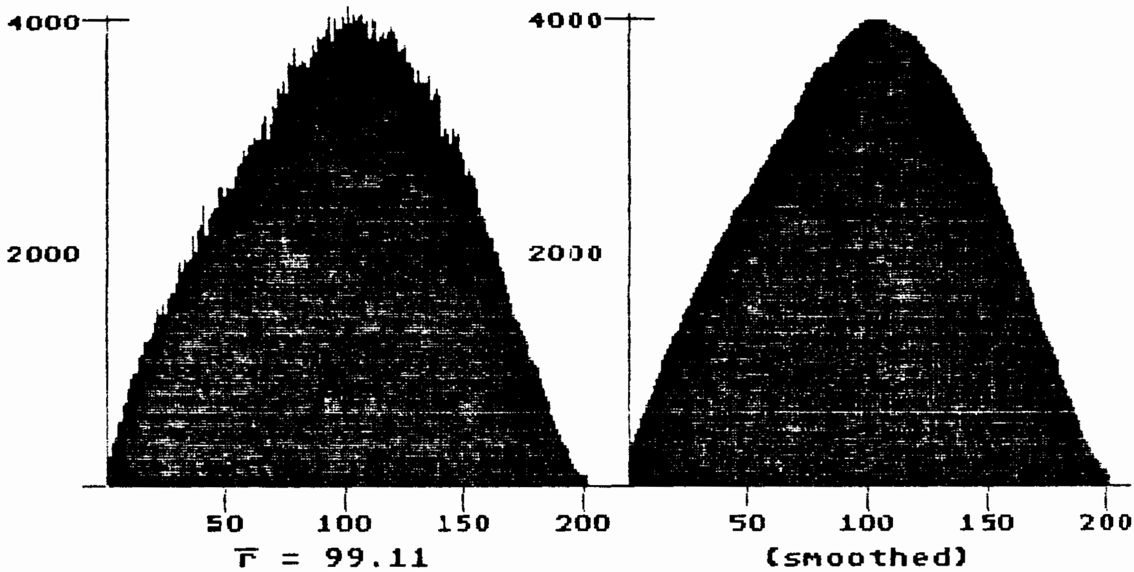


Figure 18. Cumulative radial mass distribution for the 30 large aggregates.

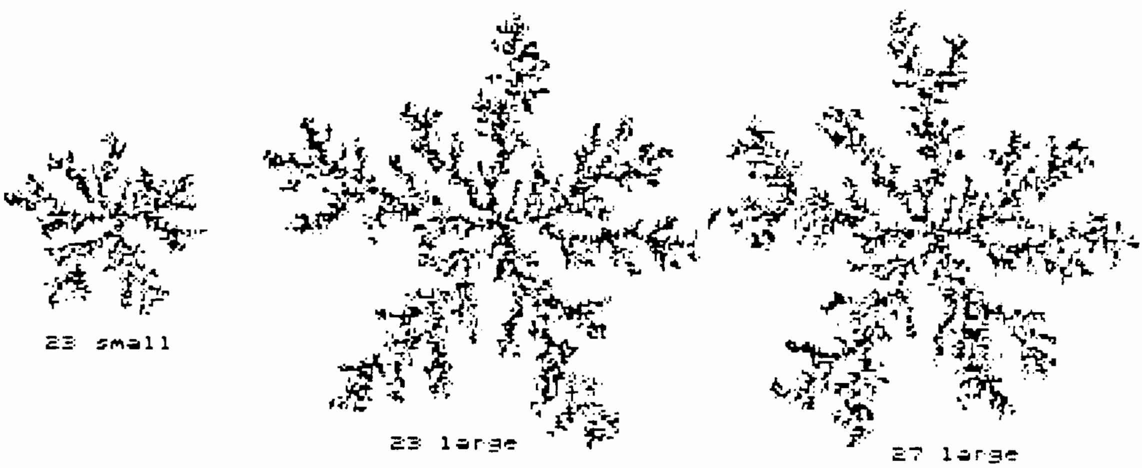


Figure 19. Persistence of growth trends.

fractal dimension not only describes how its density scales, both locally and globally, but also the resemblance noticeable in those characteristic structures due to the scale invariance, or self-similarity.

CHAPTER V

CONCLUSION

The aggregates were grown by a random process yet their structure is not entirely random. Their structure is symmetric under changes of scale, from lengths of a few pixels to that on the order of the size of the aggregate itself. A consequence of their self-similarity (or scale-invariance of their patterns) is that their density decreases as their size increases. By contrast, a two dimensional Euclidean disk with homogeneous mass density, which is compact within its perimeter, has constant density regardless of its size. Consequently, as the density of a fractal aggregate decreases to zero the perimeter becomes infinite. (Another formularization for the fractal dimension is, $(\text{perimeter})^{1/D} \propto (\text{area})^{1/2}$, see Mandelbrot, 1983.) The ramification of the structure of an aggregate contributes to this increase in the aggregate's perimeter. The screening effect which causes the arms to grow out more than interior to fill in, contributes to the decrease in density. The diffusion-limited aggregation mechanism operates on the microstructure using local growth rules, the effects of which are mediated through the fractal property of self-similarity and affect the resulting macrostructure.

Mass/length scaling relationships associated with the aggregates were analyzed to obtain a measure of the fractal dimension. The dependence of the radius of gyration on aggregate mass yielded a dimension related to global properties of the aggregate while the density-density correlation function gave a dimension more associated with local properties. The agreement between these two methods is due to the fractal property of scale invariance.

The various modifications of the correlation function indicated that the shape of the correlation 'window' is not pertinent to the evaluation of an aggregate with radial symmetry and which is grown on a square lattice. However, the results given by the method using both square 'windows' and the inclusion of the edge, more quickly attained the value to which the results of the other methods appeared to converge, as the average size of the aggregates increased. It should be noted however, that the method which would have used exactly circular 'windows' together with inclusion of the edge was not performed so that this value could be due to only the inclusion of the edge, independent of the shape of the 'window'. The methods which excluded the edge did provide additional information about the screening effect. Furthermore, the results of these methods which utilized square 'windows' and circular 'windows' did not differ significantly. The fractal dimension as calculated over the entire aggregate essentially remained constant as the size of the aggregate increased. When the edge was excluded from the correlation analysis, the correlation function indicated that the interior of the aggregate had a greater fractal dimension than the entire aggregate. However, the interior did not become compact indicating that the outer edge was screening the interior. (See Appendix E for possible modifications of the edge analysis.) The fractal dimension using the correlation function is $D_c = 1.67 \pm .01$.

After finalizing the analysis and discussion of the graphical results, it became evident that the offset in the location of the center of deposition from the lattice origin was, in fact, appreciable. Consequently, the approximation used in the radius of gyration calculations was not justified and the results had a systematic error. This offset, L, enters into the radius of gyration calculation in a complicated manner. Although, utilization of the parallel axis theorem could correct the radius of gyration for each deposition, N, it would require the functional dependence, L(N). However, the dependence that the offset has on N is non-trivial and depends on the interaction of the growing structure with the random mechanisms of the simulation. Further discussion of the approximations used in the recalculation of the fractal dimension based on the corrected radius of gyration is given in Appendix D. It is noted there that the concavity in the graphs, mentioned above, may be due, in part, to this error. The error, also indicates that 'radius of gyration', as measured from the lattice origin, is not as characteristic of the aggregate as the true radius of gyration. The fractal dimension based on the radius of gyration dependence is, D_{R_G} = 1.75 ± .08.

The correlation function results using 'windows' of 1.5 to 32.5 lattice spacings of 1.67 ± .01 are in agreement with the accepted results of 1.68 ± .05, as reported by Meakin (1983b), where 'windows' of 5 to 50 lattice spacings were utilized. The radius of gyration results of 1.75 ± .08 are in precise agreement with the accepted results reported there.

The differences with Meakin's model do not give significantly different numerical results. The slight difference in the boundary conditions, which might allow pixels to more completely fill cavities with entrances of one pixel in diameter, could give slightly different graphical results. The aggregates could be analyzed for the presence of 'lakes', which would indicate that occasionally a pixel could close off the opening of a 'fjord'. However, this analysis was not performed, in part, because Meakin's graphical results were not available.

The graphical results demonstrated the diversity in the morphologies of the aggregates as well as the symmetry property of self-similarity. The animation programs clearly demonstrated the decreasing penetration into the interior of the aggregates by the random walkers as the aggregates grew larger. The perimeter of an aggregate screens the interior and grows preferentially. Intricacies in the perimeter are enhanced by the growth mechanism and tend to be extended. Thus, the patterns of the large aggregates resemble the patterns of their predecessors.

The morphology of a diffusion-limited aggregate resembles the fractal structures of those physical processes such as electrodeposition and fluid-fluid displacement. The measured fractal dimensions for these processes, as previously stated in Chapter II, are 1.66 and 1.70, respectively. This supports the contention that diffusion-limited aggregation belongs to the same universality class of physical behavior.

REFERENCES CITED

Brady, R. and R. Ball. "Fractal Growth of Copper Electrodeposits," Nature, 309, 225, (1984).

Daccord, D., J. Nittmann, and H. Stanley. "Radial Viscous Fingers and Diffusion-Limited Aggregation: Fractal Dimension and Growth Sites," Phys. Rev. Lett., 56, 336, (1986).

Forrest, S. and T. Witten. "Long Range Correlations in Smoke Particles," J. Phys. A: Math. Gen., 12, L109, (1979).

Langer, J.. "Instabilities and Pattern Formation in Pattern Growth," J. Rev. Mod. Phys., 52, 1, (1980).

Mandelbrot, B.. The Fractal Geometry of Nature, (Freeman, New York, 1983).

Matsushita, M., M. Sano, Y. Hayakawa, H. Honjo, and Y. Sawada. "Fractal Structures of Zinc Metal Leaves by Electrodeposition," Phys. Rev. Lett., 53, 286, (1984).

Meakin, P.. "Diffusion-Controlled Cluster Formation in Two, Three, and Four Dimensions," Phys. Rev. A, 27, 604, (1983a).

Meakin, P.. "Diffusion-Controlled Cluster Formation in 2-6 Dimensional Space," Phys. Rev. A, 27, 1495, (1983b).

Niemeyer, L., L. Pietronero, and H. Wiesmann. "Fractal Dimension of Dielectric Breakdown," Phys. Rev. Lett., 52, 1033, (1984).

Paterson, L.. "Diffusion-Limited Aggregation and Two-Fluid Displacements in Porous Media," Phys. Rev. Lett., 52, 1621, (1984).

Witten, T. and L. Sander. "Diffusion-Limited Aggregation, a Kinetic Critical Phenomenon," Phys. Rev. Lett., 47, 1400, (1981).

Witten, T. and L. Sander. "Diffusion-Limited Aggregation," Phys. Rev. B, 27, 5586, (1983).

APPENDIX A

THE COMPUTER PROGRAMS

The selection of this thesis topic was, in part, motivated by the desire to demonstrate the feasibility of performing credible physics research on a personnel computer. Many student researchers do not have access to mainframe computers, especially those with graphics capabilities. Although, it could be said that fractal geometry is one of the computer viruses of the 1980's. The computer programs developed in this project can serve as a basis for further research by students interested not only in the fractal patterns they generate, which resemble many patterns found in nature; but more importantly, by the apparent generality of the model to natural and technological processes.

Initially, the simulation was attempted on a Commodore C-64 computer as it was a very popular and inexpensive system. However, with only 64K bytes of random access memory, a slow (1Mhz) 8 bit microprocessor, small maximum array size (32K), and a graphics screen of only 320 pixels by 200 pixels at 'high' resolution, it was abandoned as soon as larger and faster machines became available. The Atari 1040ST was selected because it had the most advanced technology at that time (1986), although, since then it has been superseded by other systems, preferred by researchers, because these systems are more technically supported.

The Atari 1040ST with its 16/32 Motorola 68000 microprocessor operating at 8 Mhz with 1 Megabyte of random access memory is still a respectable system. However, the basic language interpreter supplied by Atari had 'bugs' in the integer arithmetic routines and could not even use 32K of memory for arrays. With this memory limitation, simulations could not be done which would realize the potential of the 640 pixels by 400 pixels graphics display. Fortunately, GFA Basic was developed by GFA-Systemtechnik (which has become the system standard for the Atari, especially in Europe, where Atari is on par with IBM or Apple computers). The following computer programs were written in GFA Basic version 2.0.

The following short demonstration program was the prototype of more complicated and extensive programs and is included, with comments, to offer insight into the structure and coding of the simulation. It models DLA in a toroidal geometry on a two dimensional square lattice. The simulation space is a 400 by 400 lattice. The deposits are stored sequentially in an integer array using ten bit packed words; at the termination of the program the core image is dumped to a binary sequential file on disk.

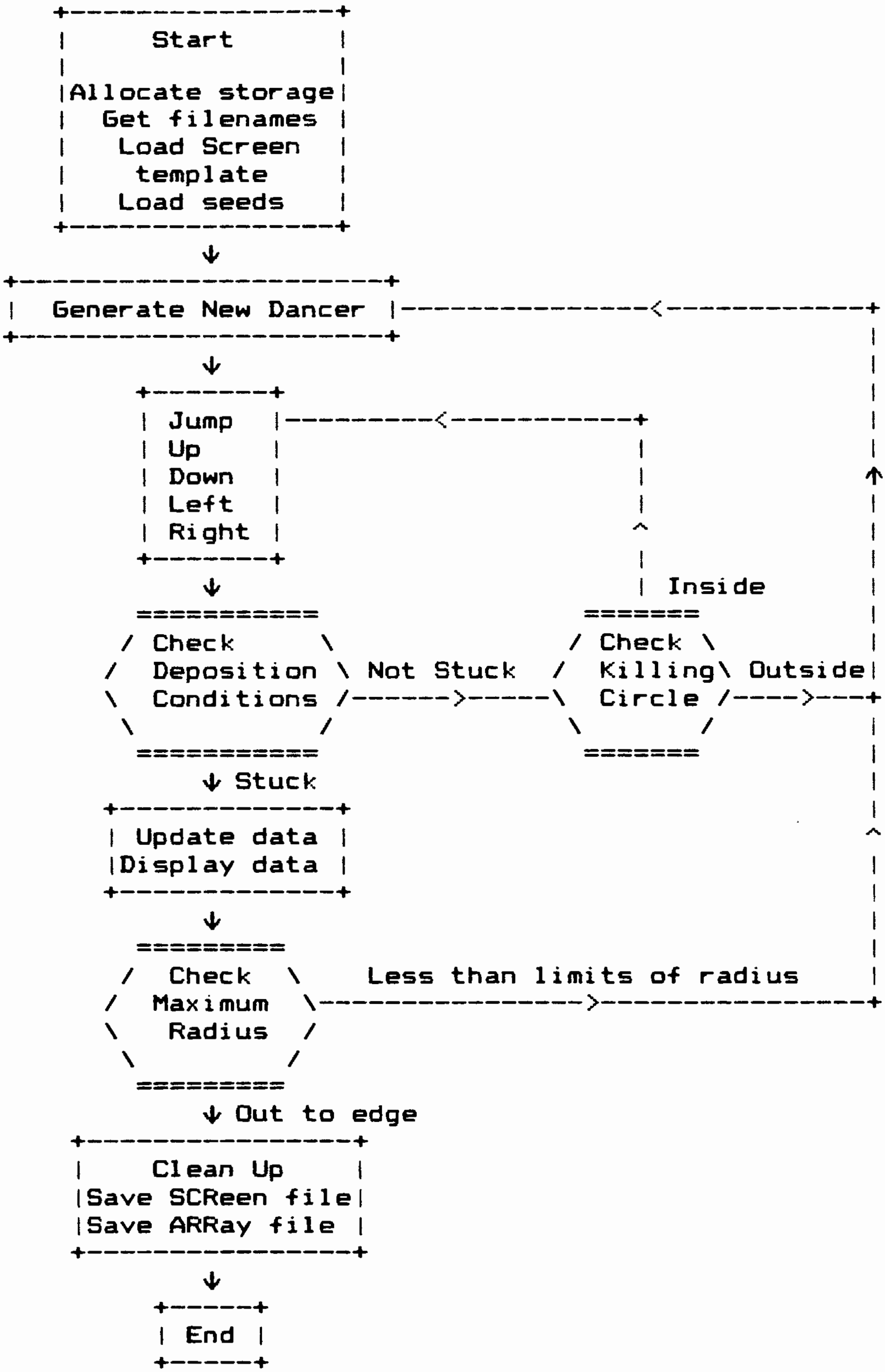


Figure 20. Demonstration program flowchart.

```
Cls  ' Clears the screen.
Graphmode 3
' 3 is complement mode, so plot(x,y) alternately sets and clears (x,y).
Deftext 1
' Standard text mode for Text command.
Color 0
' Plot color is white (for white dot on black background).
On Break Gosub Breakhandler
' Control-Shift-Reset vectors through this cleanup routine.
On Error Gosub Errorhandler
' Any errors vector through this cleanup routine.
Print "Starting seed filename:"
Fileselect "\*.SCR","SEED.SCR",A$
' Selects a filename (or NULL for none) to act as the seed.
Print At(1,1);"Storage filename:       "
Do
  Fileselect "\*.SCR",Mid$(A$,2),B$
  ' Selects filename to save work.
  Exit If B$<>"" And B$<>"\"
  ' Won't accept null filenames, a place is needed to save work;
  ' Loops until a vaild filename is obtained.
Loop
If Instr(B$,"SCR")=0 Then
  ' If the SCReen extension isn't there...
  If Instr(B$,".")=0 Then
    ' checks for a period;
    B$=B$+"."
    ' adds it if it's not there,
  Endif
  B$=B$+"SCR"
  ' then adds SCReen extension.
Endif
Hidem
Dim Order%(30000)
' Allocates storage for the array of deposit coordinates.
Order%(0)=1
' (0) is location for the number of deposits, n=(0)+1, since (0) and (1) are occupied.
' That is, first deposit is in Order%(2).
Order%(1)=0
' (1) is the maximum radius of the growth from the center of the screen.
If A$="" Or A$="\" Then
  ' If 'CANCEL' was selected for "Starting Seed", then sets up standard screen.
  Cls
  Deffill 1,1
  ' Sets fill as solid black, and
  Fill 320,200
  ' fills it up from the center out.
  Plot 200,200
  ' Starting point (seed).
  Order%(2)=205000 ! 205000 = 200 * 1024 + 200
  Order%(0)=2
  ' Put the seed as the first element of the array.
```

```
  Line 400,0,400,400
  ' Right boundary.
  Line 401,301,639,301
  ' Dividing line beween title and data sections.
  Text 408,16,"Simulation of Diffusion-"
  Text 408,32,"Limited Aggregation by"
  Text 408,48,"single particle migration."
  Text 408,64,"Diffusion space: 2-D planar"
  Text 408,80,"  square lattice"
  Text 408,96,"Deposit space: 2-D planar"
  Text 408,112,"  square lattice"
  Text 408,128,"Trajectories:"
  Text 408,144,"  collision layer: unit steps"
  Text 408,160,"  diffusion zone: orthogonal"
  Text 408,176,"    steps; scaled to R"
  Text 408,192,"(R = maximum radius; dynamic)"
  Text 408,208,"Initial seed: central pixel"
  Text 408,224,"Generating geometry: circle;"
  Text 408,240,"  radius = R + 5"
  Text 408,256,"Killing geometry: annulus;"
  Text 408,272,"  minimum radius = 2R + 5"
  Text 408,288,"Sticking probability = 1.0"
  ' Data section of screen starts here:
  Text 408,316,"Deposits:"
  Text 408,332,"Maximum growth radius:"
  Text 408,348,"Angle of maximum radius:"
  Text 408,364,"Data on Last Dancer"
  Text 408,380,"R:      Θ:"
  Text 408,396,"Number of jumps:"
Else
  ' Else if a filename was selected for a seed, load the
  Bload A$,Xbios(2)
  ' screen portion into the screen memory and the
  Bload Left$(A$,Instr(A$,"."))+"ARR",Lpeek(Arrptr(Order%()))
  ' array portion into the previously allocated array.
Endif
Jump%=1
' Jump% is the number of spaces a dancer can jump, depending on how close it is to the deposition zone
Njumps%=0
' Njumps% is the number of jumps dancer(s) have made since last depostion.
Do
  ' Main loop of program.  Loops until deposit reaches the edge.
  Stuck=False
  ' Starts out with dancer unstuck, so it can move.
  Jump%=1
  Gosub Newdancer
  ' Generates a new particle.
  Repeat
    ' Actual dancing loop. This makes the dancer move.
    Xold%=X%
    Yold%=Y%
    ' Saves old location of dancer for comparison,
```

```
    ' or to leaves particle there if deposition conditions are satisfied.
    On Random(4)+1 Gosub Up,Down,Left,Right
    ' Random number 1 through 4. 1 goes up, 2 down, etc.
    Inc Njumps%
    ' A jump was made, so count it.
    On Jump% Gosub Check
    ' If Jump%=1 (ie. in depostion zone) then checks deposition criteria.
    If Not Stuck Then
      ' If the criteria was not met then
      Plot Xold%,Yold%
      ' erases the old dancer pixel,
      Plot X%,Y%
      ' and draws the new one at the new coordinates.
    Endif
    Rd%=Int(Sqr((X%-200)^2+(Y%-200)^2))
    ' Calculates the distance from the center of the deposit.
    If Rd%>2*Order%(1)+5 Then
      ' If the dancer gets outside the killing circle at 2 Rmax + 5...
      Stuck=True
      ' artificially sticks it (so it gets replaced with a new dancer)
      Plot X%,Y%
      ' and erases it from the screen.
    Endif
    If Rd%>Order%(1)+5 Then
      ' If outside depostion zone, scales the jumping distance; larger jumps will economize run time.
      Jump%=2^Int(1.442695*Log((Rd%-Order%(1))/5))
    Else
      Jump%=1
      ' Inside the deposition zone, jumping is single-stepped: the deposit
      ' can't be jumped over and contact is normal.
    Endif
  Until Stuck
  ' Repeats dancing with this dancer until it's stuck (deposited or killed).
  Exit If 2*Order%(1)+5>200
  ' Exits the main loop if growth is big enough,if the killing circle reaches the edge of the screen.
Loop
Gosub Cleanup
' Cleans up the mess before finishing the program.
End
' Procedure Library:
Procedure Newdancer
  ' Makes a new particle to deposit.
  X%=Random(720)
  ' Radial location in half degrees, 0 to 719.
  Y%=200+Int((Order%(1)+5)*Cos(X%*Pi/360))
  ' Generating circle is Rmax+5, so y=RCos(theta) and
  X%=200+Int((Order%(1)+5)*Sin(X%*Pi/360))
  ' theta=(halfdegrees x pi)/360.
  Plot X%,Y%
  ' Puts the new dancer on the screen.
Return
Procedure Up
```

```
  Sub Y%,Jump%
  ' Jump up, so y coordinate is decremented by the distance to jump.
  If Y%<0 Then
    ' If jump is off the screen, wraps around to the other edge,
    ' (never satisfied with killing circle present; dancer dies first).
    Add Y%,400
  Endif
Return
Procedure Down
  Add Y%,Jump%
  ' Likewise, only jump is downward (increasing y coordinate).
  If Y%>399 Then
    Sub Y%,400
  Endif
Return
Procedure Left
  Sub X%,Jump%
  ' As above, only decrease x.
  If X%<0 Then
    Add X%,400
  Endif
Return
Procedure Right
  Add X%,Jump%
  If X%>399 Then
    Sub X%,400
  Endif
Return
Procedure Check
  ' Checks to see if deposition conditions are satisfied. If they are then, stick, Stuck=True.
  If Not -Point(X%,Y%) Then
    ' If the point jumped to is already occupied, then collision is detected
    Stuck=True
    ' and stick at prevoius coordinates (Xold%,Yold%).
    Inc Order%(0)
    ' Records the number of deposits as being one greater.
    Order%(Order%(0))=Xold%*1024+Yold%
    ' Encodes and saves the coordinates of the deposited particle.
    Print At(62,20);Using "#####",Order%(0)-1;
    ' Displays the position
    Ra%=Sqr((Xold%-200)^2+(Yold%-200)^2)
    Print At(55,24);Using "###",Ra%;
    ' and the radius of the deposit. Then calculates the angle from the center.
    Angle%=Atn((Yold%-200)/(Xold%-200+0.01))*57.3
    Theta%=Angle%
    ' This calculates the true angle from the arctan function, which gives
    ' angles from -90 to +90 degrees, instead of 0 to 359 degrees.
    If Angle%<0 Then
      Theta%=360+Angle%
    Endif
    If Xold%<200 Then
      Theta%=180+Angle%
```

```
    Endif
    If Ra%>Order%(1) Then
      ' If this is a maximum radius deposit, then
      Order%(1)=Ra%
      ' updates Rmax and
      Mangle%=Theta%
      ' reports the angle of the maximum radius of the deposit.
    Endif
    ' Prints it all out...
    Print At(75,21);Using "###",Order%(1);
    Print At(77,22);Using "###",Mangle%;
    Print At(63,24);Using "###",Theta%;
    Print At(69,25);Using "#####",Njumps%;
    ' Makes a beep to indicate deposition.
    Sound 1,15,1,8,1
    Sound 1,0
    Njumps%=0
    ' Resets Njumps for the new dancer which will be generated.  It's here
    ' so Njumps% is only reset between deposits, not when a dancer is killed
    ' and replaced: if it were in newdancer, it would count jumps only for that dancer.
  Endif
Return
Procedure Breakhandler
  ' If Control-Shift-Reset is key-stroked, comes here and clean up.
  Gosub Cleanup
  ' Does the clean up routine,
  On Break
  ' resets basic language's default Break handler,
  End
  ' and ends the program.
Return
Procedure Errorhandler
  ' If an error happens, comes here.
  Gosub Cleanup
  ' Cleans up the mess,
  Err$="Error # "+Str$(Err)+" occurred.|Data dumped to disk."
  ' makes a message telling what happened,
  Alert 1,Err$,1,"Return",X%
  ' and displays it. Then...
  On Error
  ' resets error handler to basic's regular one,
  End
  ' and ends the program.
Return
Procedure Cleanup
  ' This does the actual work of cleaning up.
  If Point(Xold%,Yold%)=0 Then
    ' If there's a dancer on the screen at an old coordinate
    Plot Xold%,Yold%
    ' erases it so that it doesn't appear in the SCR file.
  Endif
  If Point(X%,Y%)=0 Then
```

```
    ' Likewise if it's at the new coordinates.
    Plot X%,Y%
  Endif
  ' Binary saves the screen contents to the save filename,
  Bsave B$,Xbios(2),32000
  ' binary saves the Order array to a file with an ARR extension.
  Bsave Left$(B$,Instr(B$,"."))+"ARR",Lpeek(Arrptr(Order%())),Order%(0)*4+4
  ' and announces the saving.
  Text 80,64,"Data saved to file "+B$
Return
```

In order to display the whole aggregate on the screen at once, it was necessary to limit the maximum size of the aggregate to 30,000 deposits. If a partitioned display had been utilized, the constraints would have been upon the limitations of the computer memory and the amount of time available to run the simulation. The average time to grow the small aggregates was approximately 8 hours and it took 30 hours to grow the large aggregates. If time had not been a factor, then the memory requirements of the Boolean array simulation space and the integer array deposit space, would have allowed for a maximum of approximately 75,000 deposits. For the large version of the simulation program, the simulation was moved from the screen buffer into the main memory. Additionally the deposit array was a changed from a real number array with nine bit packed words consisting of; the x and y coordinates and the number of 'jumps' taken from a pixel's 'birth', to its deposition, into an integer array with ten bit packed words consisting of; the x and y coordinates of each deposit. (The encoding of the of the coordinates saved memory space, allowing the simulation spaces to be larger. In order to have the coordinates of the large simulation space to be greater than 512 the coordinates required ten bits.) Although, the simulation space needed four times as much memory as the deposit space, in order to allow for the diffusion zone enclosed in the 'killing' circle, the deposit space could be larger than the memory locations of the deposit array because the deposition was fractal and not compact. Integer arrays require 4 bytes of memory for each element, floating point arrays 6 bytes, and Boolean arrays need only 1 bit for each element.

In order to more quickly execute the simulation, deposition was determined by checking the spatial array of the simulation space, rather than the sequential deposit array and then only when the stepsize was a unit step. In the large simulation, the information concerning the 'dancer' or random walker was deleted; the 'dancer' or random walker was not plotted, the number of 'jumps' was not counted, and its polar coordinates at deposition were not calculated. Implementation of a smaller 'killing' circle rather than Meakin's, ($2R_{MAX.}$ vs. $3R_{MAX.}$), reduced the time a pixel would be in the diffusion zone, this effectively increased the rate of deposition. (The agreement of the fractal dimension supports this modification. Further analysis was not conducted to investigate whether this simulation was, in fact, less diffusive than Meakin's.) Various look-up tables were used to decrease the run time. Examples are the jump table which gave the lengths of the jumps that the random walker took when in the diffusion zone (instead of using the exponential function), and the Pythagorean array which gave radial distances (rather than taking the square root).

Among the programs developed for this research, the more salient are presented below. They are menu driven and are provided with 'Help' screens. The Correlate Program calculates the correlation function using exact circles and squares. It is representative and the most developed of the three correlation programs. It provides additional data such as the number of excluded pixels in the edge and the run time, (approximately 24 hours). (The number of excluded pixels was computed with the intention of additional analysis; to determine the connection between the aggregate's geometry, the correlation function results, and the number of excluded pixels.) The look-up table of partial areas is given for only one octant and by employing symmetry, is used for the whole circle.

The Radius of Gyration Program utilizes a running average as it evaluates the deposit array. It also includes the special procedure which corrects for the previously mentioned error and calculates the radius of gyration from the center of mass.

The following programs provide graphical output and analysis; Megamenu is the animation and file maintenance program, Coremenu determines the various mass distributions for single aggregates and composites, and the Deposition Frequency Histogram Program also compiles the composites, in addition to, 'slicing' the cumulative deposition probability distribution, at any arbitrary deposition probability.

```
' Correlation Program
Version=6.1
Revdate$="13 Jun 88"
Dim Order%(30000)
Dim Pythagoras(100,100)
Dim Power(6,3)
Dim Include(32,32)
Cls
Print "Automatic Correlation Calculator, version"'Version;",''Revdate$
Print "Determines the fractal dimension by least squares slope"
Input "Number of windows of increasing length (2 to 6)";Limit%
Print "Setting lookup table:"
X%=0
Repeat
  If (X% And 7)=7
    Print At(23,4);Using "X=##",X%
  Endif
  Y%=0
  Repeat
    A=Sqr((50-X%)^2+(50-Y%)^2)
    Pythagoras(X%,Y%)=A
    Pythagoras(Y%,X%)=A
    Pythagoras(100-X%,Y%)=A
    Pythagoras(100-Y%,X%)=A
    Pythagoras(Y%,100-X%)=A
    Pythagoras(X%,100-Y%)=A
    Pythagoras(100-X%,100-Y%)=A
    Pythagoras(100-Y%,100-X%)=A
    Inc Y%
  Until Y%>X%
  Inc X%
Until X%>50
Print "Reading pixel integration table"
Y%=0
Repeat
  X%=Y%
  Repeat
    Read Include(X%,Y%)
    Let Include(Y%,X%)=Include(X%,Y%)
    Inc X%
  Until X%>32
  Inc Y%
Until Y%>32
Ndx%=1
```

```
Radius%=2^(Limit%-1)
Repeat
  Power(Ndx%,1)=Radius%+0.5
  Inc Ndx%
  Div Radius%,2
Until Ndx%>Limit%
Do
  Cls
  Showm
  Print "Choose Mode of Operation: Type number or click on selection."
  Print
  Print "1  Automatic processing of all .ARR files on disk"
  Print
  Print "2  Use already created directory of filenames (CORELATE.DIR)"
  Print
  Print "3  Process single file"
  Print
  Print "4  Helpful hints and instructions"
  Print
  Print "5  Exit"
  Graphmode 3
  Deffill 1,1
  Ptrvertpos%=Mousey
  If Frac(Ptrvertpos%/32)<0.5 Then
    Gosub Inbox(Ptrvertpos%)
  Else
    In%=0
  Endif
  Do
    Repeat
      Ptrvertpos%=Mousey
      If (In%>0) And (Frac(Ptrvertpos%/32)>0.5) Then
        Gosub Outbox(Ptrvertpos%)
      Endif
      If (In%=0) And (Frac(Ptrvertpos%/32)<0.5) Then
        Gosub Inbox(Ptrvertpos%)
      Endif
      Switch%=Mousek
      If Switch%>0 Then
        If In%>0 Then
          Switch%=(Ptrvertpos%\32)-2
        Else
          Switch%=0
          Sound 1,15,6,7,5
          Sound 1,0
        Endif
      Endif
      Key$=Inkey$
    Until Key$<>"" Or Switch%
    If Switch% Then
      Key$=Str$(Switch%)
    Endif
```

```
    Exit If Val(Key$)>0 And Val(Key$)<6
    Sound 1,15,6,7,5
    Sound 1,0
  Loop
  Cls
  Graphmode 1
  On Val(Key$) Gosub Auto,Existingfile,Single,Help,Exit
  In%=0
  Switch%=0
Loop
End
Procedure Inbox(Ht%)
  Ht%=32*(Ht%\32)
  If Ht%>16 And Ht%<192 Then
    Pbox -1,Ht%,500,Ht%+16
    In%=Ptrvertpos%\32
  Endif
Return
Procedure Outbox(Ht%)
  Ht%=32*In%
  Pbox -1,Ht%,500,Ht%+16
  In%=0
Return
Procedure Exit
  Edit
Return
Procedure Help
  Cls
  Print "  This program can run in automatic mode.  The requirements are that"
  Print "it must be given a disk with a series of .ARR files with their"
  Print "associated .SCR files. There can be no other .ARR files on the disk."
  Print "If there are no .ARR files in the current disk or directory, a bus"
  Print "error (two bombs) will result."
  Print "  To use the pre-existing directory mode (eg. to do only some of"
  Print "the .ARR files on a disk), create a text file named CORELATE.DIR,"
  Print "containing the filenames of then .ARR files you wish to process."
  Print "Each filename should appear on a single line in the file."
  Print "  In both these cases, the results go into a file called CORELATE.DAT"
  Print "in a tabular form, with the filename at the top, followed by lines"
  Print "with three numbers separated by commas. These represent R, Mdisk(R),"
  Print "and Msquare(R) for each R processed (Mdisk is the average pixel"
  Print "density in a disk of radius R).  The slopes of the best-fit power"
  Print "curves for each technique are printed on the next two lines.  These"
  Print "slopes  are the fractal dimensions as determined by the two-point"
  Print "correlation function over disks and squares respectively.  The total"
  Print "number of deposits and the number of pixel excluded to eliminate edge"
  Print "effects are printed on the last two lines."
  Print "  The Single File mode allows you to process a single file on the"
  Print "disk, which can be entered from a Fileselect box. The results do not"
  Print "go into a file, but are just printed on the screen."
  Print "                         Hit any key to continue"
  Repeat
```

```
  Until Inkey$<>""
Return
Procedure Single
  Gosub Loader
  If File$<>"" Then
    Time=Timer
    Gosub Process(File$)
    Cls
    Gosub Secs_to_hms((Timer-Time)/200)
    Print "Running time:"'Hms$
    Ndx%=1
    Repeat
      Power(Ndx%,2)=Power(Ndx%,2)/(Power(0,1))
      Power(Ndx%,3)=Power(Ndx%,3)/(Power(0,1))
      Print Power(Ndx%,1);",";Power(Ndx%,2);",";Power(Ndx%,3)
      Inc Ndx%
    Until Ndx%>Limit%
    Power(0,0)=Limit%
    Gosub Power
    Print "Fractal Dimension(disk)=";Sloped
    Print "Fractal Dimension(square)=";Slopes
    Print "Total Number of Deposits=";Order%(0)-1
    Print "Number of excluded pixels=";Power(0,2)
    Print "Hit any key to continue"
    Repeat
    Until Inkey$<>""
  Endif
Return
Procedure Auto
  Dir "*.ARR" To "CORELATE.DIR"
  Gosub Existingfile
Return
Procedure Existingfile
  Open "I",#0,"CORELATE.DIR"
  If Eof(#0) Then
    Goto Escape
  Endif
  Repeat
    Gosub Open_file_for_output_or_append("CORELATE.DAT",1)
    Input #0,File$
    If File$="" Then
      Print "       Directory file is empty: either no .ARR files on current"
      Print "           directory, or you forgot to fill the .DIR file."
      Print "                     Hit any key to continue."
      Repeat
      Until Inkey$<>""
      Goto Escape
    Endif
    Gosub Load(File$)
    Time=Timer
    Gosub Process(File$)
    Gosub Secs_to_hms((Timer-Time)/200)
```

```
    Print #1,File$
    Print #1,"Running time:"'Hms$
    Ndx%=1
    Repeat
      Power(Ndx%,2)=Power(Ndx%,2)/(Power(0,1))
      Power(Ndx%,3)=Power(Ndx%,3)/(Power(0,1))
      Print #1,Power(Ndx%,1);",";Power(Ndx%,2);",";Power(Ndx%,3)
      Inc Ndx%
    Until Ndx%>Limit%
    Power(0,0)=Limit%
    Gosub Power
    Print #1,"Fractal Dimension(disk)=";Sloped
    Print #1,"Fractal dimension(square)=";Slopes
    Print #1,"Total Number of Deposits=";Order%(0)-1
    Print #1,"Number of excluded pixels=";Power(0,2)
    Close #1
  Until Eof(#0)
  Escape:
  Close
Return
Procedure Process(File$)
  Deffill 0,1
  Pbox 401,0,639,399
  Deffill 1,1
  Rwindomax%=Int(Power(1,1))
  Rdepositmax%=Order%(1)
  Ndx%=1
  Power(0,1)=0
  Power(0,2)=0
  Repeat
    Power(Ndx%,2)=0
    Power(Ndx%,3)=0
    Inc Ndx%
  Until Ndx%>Limit%
  Print At(53,3);"File:";'File$
  Print At(53,5);"N=    0"
  Print At(53,7);Using "Out of ##### total deposits",Order%(0)-1
  Print At(53,9);"Excluded pixels=     0"
  N%=2
  Repeat
    Xw%=Order%(N%)\1024
    Yw%=Order%(N%) And 1023
    If Abs(Sqr((Xw%-200)^2+(Yw%-200)^2))+Rwindomax%<=Rdepositmax% Then
      Inc Power(0,1)
      X%=Xw%-Rwindomax%
      Repeat
        Y%=Yw%-Rwindomax%
        Repeat
          If Point(X%,Y%) Then
            Rpix=Pythagoras(X%-Xw%+50,Y%-Yw%+50)
            Ndx%=1
            Repeat
```

```
            Exit If Abs(Xw%-X%)>Power(Ndx%,1) Or Abs(Yw%-Y%)>Power(Ndx%,1)
            Inc Power(Ndx%,3)
            Exit If Rpix>Power(Ndx%,1)+0.70710678119
            If Rpix<Power(Ndx%,1)-0.70710678119 Then
              Inc Power(Ndx%,2)
            Else
              Corner=Sgn(Power(Ndx%,1)-Rpix)*0.5
              Rcnr=(Abs(Abs(X%)-Abs(Xw%))+Corner)^2
              Add Rcnr,(Abs(Abs(Y%)-Abs(Yw%))+Corner)^2
              Rcnr=Sqr(Rcnr)
              If Power(Ndx%,1)>Min(Rpix,Rcnr) And Power(Ndx%,1)<Max(Rpix,Rcnr)
                Add Power(Ndx%,2),Include(Abs(Xw%-X%),Abs(Yw%-Y%))
              Else
                If Power(Ndx%,1)>Rpix Then
                  Inc Power(Ndx%,2)
                Endif
              Endif
            Endif
            Inc Ndx%
          Until Ndx%>Limit%
        Endif
        Inc Y%
      Until Y%>Yw%+Rwindomax%
      Inc X%
    Until X%>Xw%+Rwindomax%
  Else
    Inc Power(0,2)
    Print At(70,9);Using "#####",Power(0,2)
  Endif
  Print At(53,5);Using "N=#####",N%-1
  Inc N%
  Option "U1"
 Until N%>Order%(0)
Return
Procedure Loader
 Print At(1,3);"Select array:      "
 Fileselect "\*.ARR","SEED.ARR",File$
 If File$<>"" Then
   Gosub Load(File$)
 Endif
Return
Procedure Load(File$)
 Hidem
 Arrayfill Order%(),0
 Bload File$,Lpeek(Arrptr(Order%()))
 Gosub Parsefilename(File$)
 Bload Pathname$+"\"+Left$(File$,Instr(File$,"."))+"SCR",Xbios(2)
Return
Procedure Parsefilename(Fn$)
 Local First%,Last%,X%
 Pathname$=Left$(Fn$,Instr(Fn$,":"))
 First%=Instr(Fn$,"\")
```

```
  For X%=Len(Fn$) Downto 1
    If Mid$(Fn$,X%,1)="\"
      Last%=X%
    Endif
    Exit If Mid$(Fn$,X%,1)="\"
  Next X%
  Pathname$=Pathname$+Mid$(Fn$,First%,Last%-First%)
  File$=Mid$(Fn$,Last%+1)
Return
Procedure Open_file_for_output_or_append(File$,Chanl%)
  If Not Exist(File$) Then
    Open "O",#Chanl%,File$
  Else
    Open "A",#Chanl%,File$
  Endif
Return
Procedure Power
  Local I%,N%,Sumofx,Sumofy,Sumofz,Sumofproducts,Sumofprod2,Sumofsquares
  N%=Power(0,0)
  Sumofx=0
  Sumofy=0
  Sumofz=0
  Sumofproducts=0
  Sumofprod2=0
  Sumofsquares=0
  For I%=1 To N%
    Add Sumofx,Log(Power(I%,1))
    Add Sumofy,Log(Power(I%,2))
    Add Sumofz,Log(Power(I%,3))
    Add Sumofproducts,(Log(Power(I%,1)))*(Log(Power(I%,2)))
    Add Sumofprod2,(Log(Power(I%,1)))*(Log(Power(I%,3)))
    Add Sumofsquares,(Log(Power(I%,1)))^2
  Next I%
  Sloped=(N%*Sumofproducts-Sumofx*Sumofy)/(N%*Sumofsquares-Sumofx^2)
  Slopes=(N%*Sumofprod2-Sumofx*Sumofz)/(N%*Sumofsquares-Sumofx^2)
Return
Procedure Secs_to_hms(Secs)
  Local H,M,S
  Hms$=""
  H=Secs\3600
  M=(Secs Mod 3600)\60
  S=(Secs Mod 3600) Mod 60
  If H>0 Then
    Hms$=Str$(H)+" hours, "
  Endif
  Hms$=Hms$+Str$(M)+" minutes, "+Str$(S)+" seconds"
Return
Data 1,.97173982736,.98323187634,1,.99072351790,0,0,0,.99509549182
Data 0,0,0,0,0,0,1,.99747439951,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.99871790316
Data .54540604028,.76932502669,1,.87746746419,0,0,1,.93596316353
Data 0,0,0,0,0,0,1,.96712950448,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.9833278216
Data .13685659153,1,.51818108335,0,0,1,.75601286272
```

```
Data 0,0,0,0,0,0,1,.87575702090,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.93711375142
Data .79041291337,.040939641236,0,0,1,.44699616090
Data 0,0,0,0,0,0,1,.72232444282,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.85994358826
Data 0,0,1,.92966414755,.063188476255,0,0,0,0,0,0,1,.50504463958
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.75159505196
Data 1,.99978095027,.36478676634,0,0,0,0,0,0,0,1,.22126674792
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.61175243929
Data .50713675836,0,0,0,0,0,0,0,1,.86285867404,.0044426875975
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.44000194594
Data 0,0,0,0,0,0,0,1,.43815088971,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
Data .23582534771
Data 0,0,0,0,0,1,.88988358922,.037209702437,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Data 1,.96754775755,.031044089235
Data 0,0,0,1,.99993904191,.32508603414,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
Data .72754780948,0
Data 0,0,1,.61938262358,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
Data .42180377617,0
Data 1,.76123760181,.031030323298,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
Data .98595157766,0.094366951496,0
Data .05568853789,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.70187921077,0,0
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.28507660143,0,0
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.82387863618,.0043956641457,0,0
Data 0,0,0,0,0,0,0,0,0,0,0,0,0,1,.32956993207,0,0,0
Data 0,0,0,0,0,0,0,0,0,0,0,1,.782780131,.0039658586902,0,0,0
Data 0,0,0,0,0,0,0,0,0,1,.98980506335,.20740175794,0,0,0,0
Data 0,0,0,0,0,0,0,0,1,.55789679661,0,0,0,0,0
Data 0,0,0,0,0,0,1,.8302864983,.034906049143,0,0,0,0,0
Data 0,0,0,0,1,.95098614268,.16377227806,0,0,0,0,0,0
Data 0,0,1,.99069313362,.31065568299,0,0,0,0,0,0,0
Data 1,.99880149052,.41972986399,0,0,0,0,0,0,0,0
Data .45948678866,0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0,0
Data 0,0,0,0,0,0,0
Data 0,0,0,0,0,0
Data 0,0,0,0,0
Data 0,0,0,0
Data 0,0,0
Data 0,0
Data 0

' Radius of Gyration Program
Version=1.7
Revdate$="29 Oct 88"
Dim Order%(30000)
Dim Radii(1,400)
Dim Pythagoras(100,100)
Do
  Cls
  Showm
  Print "Automatic Radius of Gyration Calculator, version"'Version;","'Revdate$
  Print "Choose Mode of Operation: Type number or click on selection."
```

```
Print "1  Automatic processing of all .ARR files on disk"
Print
Print "2  Use already created directory of filenames (GYRATE.DIR)"
Print
Print "3  Process single file"
Print
Print "4  Helpful hints and instructions"
Print
Print "5  Exit"
Print "6  Special processing of single file"
Graphmode 3
Deffill 1.1
Ptrvertpos%=Mousey
If Frac(Ptrvertpos%/32)<0.5 Then
  Gosub Inbox(Ptrvertpos%)
Else
  In%=0
Endif
Do
  Repeat
    Ptrvertpos%=Mousey
    If (In%>0) And (Frac(Ptrvertpos%/32)>0.5) Then
      Gosub Outbox(Ptrvertpos%)
    Endif
    If (In%=0) And (Frac(Ptrvertpos%/32)<0.5) Then
      Gosub Inbox(Ptrvertpos%)
    Endif
    Switch%=Mousek
    If Switch%>0 Then
      If In%>0 Then
        Switch%=(Ptrvertpos%\32)
      Else
        Switch%=0
        Sound 1,15,6,7,5
        Sound 1,0
      Endif
    Endif
    Key$=Inkey$
  Until Key$<>"" Or Switch%
  If Switch% Then
    Key$=Str$(Switch%)
  Endif
  Exit If Val(Key$)>0 And Val(Key$)<7
  Sound 1,15,6,7,5
  Sound 1,0
Loop
Cls
Graphmode 1
On Val(Key$) Gosub Auto,Existingfile,Single,Help,Exit,Special
In%=0
Switch%=0
Loop
```

```
End
Procedure Inbox(Ht%)
  Ht%=32*(Ht%\32)
  If Ht%>16 And Ht%<224 Then
    Pbox -1,Ht%,500,Ht%+16
    In%=Ptrvertpos%\32
  Endif
Return
Procedure Outbox(Ht%)
  Ht%=32*In%
  Pbox -1,Ht%,500,Ht%+16
  In%=0
Return
Procedure Exit
  Edit
Return
Procedure Help
  Cls
  Print "This program can run in automatic mode.  The requirements are that"
  Print "it must be given a disk with a series of .ARR files.  If there are no"
  Print ".ARR files on the disk an error (two bombs) will result."
  Print "To use the pre-existing directory mode (eg. to do only some of"
  Print "the .ARR files on a disk), create a text file named GYRATE.DIR,"
  Print "containing the filenames of the .ARR files you wish to process."
  Print "Each filename should appear on a single line in the file."
  Print "The Single File mode allows you to process a single file on the"
  Print "disk, which can be entered from a Fileselect box."
  Print "In all these cases, the results go into a file called <FILENAME>.GYR"
  Print "Type 'Y' If You Have Inserted An Expendable Disk"
  Repeat
    Answer$=Inkey$
  Until Answer$="Y" Or Answer$="y"
Return
Procedure Single
  Gosub Loader
  If File$<>"" Then
    Time=Timer
    Gosub Process(File$)
    Cls
    Gosub Secs_to_hms((Timer-Time)/200)
    Print "Running time:"'Hms$
    Print "Hit any key to continue"
    Repeat
    Until Inkey$<>""
    Cls
    Gosub Drawaxes(100,300,0,450,250,0,40,30)
    For X%=1 To Radii(0,0)-1
      Depvar%=Log(Radii(0,X%))*40+100
      Indvar%=Log(Radii(1,X%))*30
      Dv2%=Log(Radii(0,X%+1))*40+100
      Iv2%=Log(Radii(1,X%+1))*30
      Draw Depvar%,300-Indvar% To Dv2%,300-Iv2%
```

```
    Next X%
    Repeat
    Until Inkey$<>""
  Endif
Return
Procedure Auto
  Dir "*.ARR" To "GYRATE.DIR"
  Gosub Existingfile
Return
Procedure Existingfile
  Open "I",#0,"GYRATE.DIR"
  If Eof(#0) Then
    Goto Escape
  Endif
  Repeat
    Input #0,File$
    If File$="" Then
      Print "       Directory file is empty: either no .ARR files on current"
      Print "           directory, or you forgot to fill the .DIR file."
      Print "                     Hit any key to continue."
      Repeat
      Until Inkey$<>""
      Goto Escape
    Endif
    Gosub Load(File$)
    Gosub Process(File$)
  Until Eof(#0)
  Escape:
  Close
Return
Procedure Process(File$)
  Cls
  Line 400,0,400,399
  Print At(53,3);"File:";'File$
  Print At(53,5);"N=    0"
  Print At(53,7);Using "Out of ##### total deposits",Order%(0)-1
  Sum=0
  Radii(0,0)=1
  Avex=0
  Avey=0
  Dest%=Int(((Radii(0,0)+10)^2.4)/82)
  N%=2
  Repeat
    Xpixel%=Order%(N%)\1024
    Ypixel%=Order%(N%) And 1023
    Avex=(Avex*(N%-2)+Xpixel%)/(N%-1)
    Avey=(Avey*(N%-2)+Ypixel%)/(N%-1)
    Add Sum,(Avex-Xpixel%)^2+(Avey-Ypixel%)^2
    Plot Xpixel%,Ypixel%
    If (N%-1)=Dest% Then
      Radii(1,Radii(0,0))=Sqr(Sum/(N%-1))
      Radii(0,Radii(0,0))=Dest%
```

```
      Inc Radii(0,0)
      Dest%=Int(((Radii(0,0)+10)^2.4)/82)
    Endif
    If (N%-1) Mod 100=0 Then
      Print At(53,5);Using "N=#####",N%-1
    Endif
    Inc N%
    Option "U1"
  Until N%>Order%(0)
  Dec Radii(0,0)
  Gosub Parsefilename(File$)
  File$=Pathname$+"\"+Left$(File$,Instr(File$,"."))+"GYA"
  Bsave File$,Lpeek(Arrptr(Radii())),(Radii(0,0)+1)*12+8
Return
Procedure Loader
  Print At(1,3);"Select array:      "
  Fileselect "\*.ARR","SEED.ARR",File$
  If File$<>"" Then
    Gosub Load(File$)
  Endif
Return
Procedure Load(File$)
  Hidem
  Arrayfill Order%(),0
  Arrayfill Radii(),0
  Bload File$,Lpeek(Arrptr(Order%()))
Return
Procedure Parsefilename(Fn$)
  Local First%,Last%,X%
  Pathname$=Left$(Fn$,Instr(Fn$,":"))
  First%=Instr(Fn$,"\")
  For X%=Len(Fn$) Downto 1
    If Mid$(Fn$,X%,1)="\"
      Last%=X%
    Endif
    Exit If Mid$(Fn$,X%,1)="\"
  Next X%
  Pathname$=Pathname$+Mid$(Fn$,First%,Last%-First%)
  File$=Mid$(Fn$,Last%+1)
Return
Procedure Power
  Local I%,N%,Sumofx,Sumofy,Sumofz,Sumofproducts,Sumofprod2,Sumofsquares
  N%=Power(0,0)
  Sumofx=0
  Sumofy=0
  Sumofz=0
  Sumofproducts=0
  Sumofprod2=0
  Sumofsquares=0
  For I%=1 To N%
    Add Sumofx,Log(Power(I%,1))
    Add Sumofy,Log(Power(I%,2))
```

```
    Add Sumofz,Log(Power(I%,3))
    Add Sumofproducts,(Log(Power(I%,1)))*(Log(Power(I%,2)))
    Add Sumofprod2,(Log(Power(I%,1)))*(Log(Power(I%,3)))
    Add Sumofsquares,(Log(Power(I%,1)))^2
  Next I%
  Sloped=(N%*Sumofproducts-Sumofx*Sumofy)/(N%*Sumofsquares-Sumofx^2)
  Slopes=(N%*Sumofprod2-Sumofx*Sumofz)/(N%*Sumofsquares-Sumofx^2)
Return
Procedure Secs_to_hms(Secs)
  Local H,M,S
  Hms$=""
  H=Secs\3600
  M=(Secs Mod 3600)\60
  S=(Secs Mod 3600) Mod 60
  If H>0 Then
    Hms$=Str$(H)+" hours, "
  Endif
  Hms$=Hms$+Str$(M)+" minutes, "+Str$(S)+" seconds"
Return
Procedure Drawaxes(Originx%,Originy%,Lendx%,Rendx%,Upendy%,Loendy%,Hashx%,Hashy%)
  Defline 1,1,1,1
  If Lendx%=0 Then
    Defline 1,1,0,1
  Endif
  If Rendx%=0 Then
    Defline 1,1,1,0
  Endif
  Draw Originx%-Lendx%,Originy% To Originx%+Rendx%,Originy%
  Defline 1,1,1,1
  If Upendy%=0 Then
    Defline 1,1,0,1
  Endif
  If Loendy%=0 Then
    Defline 1,1,1,0
  Endif
  Draw Originx%,Originy%-Upendy% To Originx%,Originy%+Loendy%
  Local A%,Length%
  Length%=10
  Defline 1,1,0,0
  If Hashx%<>0 Then
    For A%=Originx% To Originx%-Lendx% Step -Hashx%
      Draw A%,Originy%-Length% To A%,Originy%+Length%
    Next A%
    For A%=Originx% To Originx%+Rendx% Step Hashx%
      Draw A%,Originy%-Length% To A%,Originy%+Length%
    Next A%
  Endif
  If Hashy%<>0 Then
    For A%=Originy% To Originy%-Upendy% Step -Hashy%
      Draw Originx%+Length%,A% To Originx%-Length%,A%
    Next A%
    For A%=Originy% To Originy%+Loendy% Step Hashy%
```

```
      Draw Originx%+Length%,A% To Originx%-Length%,A%
    Next A%
  Endif
Return
Procedure Centerofmass(P.array,Lmt%)
  N%=0
  Swap *P.array,Avearray%()
  Avex=0
  Avey=0
  Do
    Inc N%
    Exit If N%>Lmt%
    Avex=(Avex*(N%-1)+(Avearray%(N%+1)\1024))/N%
    Avey=(Avey*(N%-1)+(Avearray%(N%+1) And 1023))/N%
  Loop
  Swap *P.array,Avearray%()
Return
Procedure Special
  Gosub Loader
  If File$<>"" Then
    Do
      Print "Input number of deposits to include in Rg (up to ";Order%(0)-1;", 0 to quit)";
      Input Limit%
      Exit If Limit%=0
      Gosub Centerofmass(*Order%(),Limit%)
      Print "Center of mass ="'Avex-200;",''200-Avey
      Print "Distance Center of Mass to Origin ="'Sqr((Avex-200)^2+(Avey-200)^2)
      Gosub Specialprocess(File$)
      Print "Ln(# of deposits) ="'Log(Limit%)
      Print "Ln(Rg) ="'Log(Sqr(Sum/(Limit%)))
    Loop
  Endif
Return
Procedure Specialprocess(File$)
  Sum=0
  N%=2
  Repeat
    Xpixel%=Order%(N%)\1024
    Ypixel%=Order%(N%) And 1023
    Add Sum,(Avex-Xpixel%)^2+(Avey-Ypixel%)^2
    Inc N%
    Option "U1"
  Until N%>Limit%
Return

' Megamenu Program
Version=4.3
Revdate$="29 Jun 88"
Dim Order(30000)
Dim Order%(30000)
Dim Menu$(50)
Let Menu$(0)="Desk"
```

```
Let Menu$(1)=" Utilities info"
Let Menu$(2)="--------------------"
For I=3 To 9
  Let Menu$(I)=Str$(I)
Next I
Do
  Inc I
  Read Menu$(I)
  Exit If Menu$(I)="*"
Loop
Data "Exit"," Quit  ","","Utilities"," Invert"," Display SCR file  "
Data "  Dump to printer","  Strip data lines","  View array file",""
Data "Animation","  Load ARR file  ","-----------------","  Animate"
Data "  Involute","  Zonal growth","","","*"
Menu Menu$()
On Menu  Gosub Handle_it_for_me
Print At(1,3);"Menu Program Version"'Version;","'Revdate$
Do
  On Menu
Loop
End
Procedure Handle_it_for_me
  Cls
  If Menu(0)=1 Then
    Gosub Give_info
  Else
    On Menu(0)-11 Gosub Quit,Dummy,Dummy,Invert,Disp,Prscreen,Strip,Viewarr
    If Menu(0)>19 Then
      On Menu(0)-19 Gosub Dummy,Dummy,Loader,Dummy,Animate,Involute,Zonal
    Endif
  Endif
  Menu Menu$()
  Print At(1,3);"Select function:         "
Return
Procedure Give_info
Return
Procedure Quit
  Menu Kill
  Edit
Return
Procedure Invert
  Print At(1,3);"File to invert: "
  Fileselect "\*.SCR","SEED.SCR",A$
  If A$<>"" Then
    Hidem
    Bload A$,Xbios(2)
    For X%=Xbios(2) To Xbios(2)+31998 Step 2
      Dpoke X%,Not Dpeek(X%)
    Next X%
    Bsave A$,Xbios(2),32000
    Showm
  Endif
```

```
Return
Procedure Disp
  Print At(1,3);"File to display: "
  Fileselect "\*.SCR","SEED.SCR",A$
  If A$<>"" Then
    Hidem
    Bload A$,Xbios(2)
    Repeat
    Until Inkey$<>""
    Showm
  Endif
Return
Procedure Prscreen
  Print At(1,3);"File to print:   "
  Fileselect "\*.SCR","SEED.SCR",A$
  If A$<>"" Then
    Hidem
    Bload A$,Xbios(2)
    Sdpoke 1262,0
    Showm
  Endif
Return
Procedure Strip
  A$="File must be in normal video|mode (black on white) to strip.|"
  A$=A$+"If in doubt, check with|display function."
  Alert 3,A$,2,"go ahead|cancel",A%
  If A%=1 Then
    Fileselect "\*.SCR","SEED.SCR",A$
    If A$<>"" Then
      Hidem
      Bload A$,Xbios(2)
      Print At(52,23);"                           ";
      Print At(52,24);"                           ";
      Print At(52,25);"                           ";
      Gosub Invert_window(488,304,527,319)
      Gosub Invert_window(592,320,615,335)
      Gosub Invert_window(608,336,631,351)
      Bsave A$,Xbios(2),32000
      Showm
    Endif
  Endif
Return
Procedure Invert_window(X%,Y%,X1%,Y1%)
  Color 0
  Graphmode 3
  For A%=Y% To Y1%
    For B%=X% To X1%
      Plot B%,A%
    Next B%
  Next A%
Return
Procedure Viewarr
```

```
Begin:
Local Start%,Line%,Len%,A$,Old!,Optr%,Nptr%,Changed!
Changed!=False
Optr%=Lpeek(Arrptr(Order()))
Nptr%=Lpeek(Arrptr(Order%()))
Lpoke Optr%,30001
Lpoke Nptr%,30001
Arrayfill Order(),0
Arrayfill Order%(),0
Print At(1,3);"Array file to view:";
Fileselect "\*.AR?","SEED.ARR",Arr$
If Arr$<>"" Then
  Bload Arr$,Optr%
  Old!=True
  Len%=Order(0)
  If Order(1)<>Int(Order(1)) Then
    Bmove Optr%,Nptr%,8
    If Order%(0)<30001 Then
      Bmove Optr%,Nptr%,Order%(0)*4+8
    Else
      Bmove Optr%,Nptr%,120008
    Endif
    Old!=False
  Endif
  If Not Old! Then
    Len%=Order%(0)
  Endif
  If Instr("23456789",Right$(Arr$,1))=0 Then
    Glen%=Len%
    Seg%=1
    Segment%=0
  Else
    Open "R",#1,Left$(Arr$,Len(Arr$)-1)+"R",4
    Field #1,4 As Buf$
    Get #1,2
    Glen%=Cvl(Buf$)
    Close #1
    Seg%=Val(Right$(Arr$,1))
    Segment%=29999*(Seg%-1)
  Endif
  Gosub Viewarrscreen
  Do
    For Line%=Start% To Start%+23
      If Old! Then
        If Line%=0 Then
          Print At(1,2);"        N = ";Order(Line%)-1;"          ";
        Endif
        If Line%=1 Then
          Print At(1,3);"        Rmax = ";Order(Line%);"        ";
        Endif
        If Line%>1 And Line%<30001 Then
          Print At(1,Line%-Start%+2);Using "#####  ",Line%-1;
```

```
      Print At(9,Line%-Start%+2);Using "#####    ",Order(Line%)\262144;
      Print At(18,Line%-Start%+2);Using "###,",Order(Line%) Mod 262144\512;
      Print At(22,Line%-Start%+2);Using "### ",Order(Line%) Mod 512;
    Endif
  Else
    If Line%=0 Then
      Print At(1,2);"       N = ";Order%(Line%)-1;"         ";
    Endif
    If Line%=1 Then
      Print At(1,3);"       Rmax = ";Order%(Line%);"        ";
    Endif
    If Line%>1 And Line%<30001 Then
      Print At(1,Line%-Start%+2);Using "#####   ",Line%+Segment%-1;
      Print At(9,Line%-Start%+2);Using "    ###,",Order%(Line%)\1024;
      Print At(17,Line%-Start%+2);Using "### ",Order%(Line%) And 1023;
    Endif
  Endif
  If Line%>30000 Then
    Print At(1,Line%-Start%+2);"                          ";
  Endif
Next Line%
Repeat
  A$=Inkey$
  Let Mouse%=Mousek
  If Mouse%<>0 Then
    A$="E"
    Ptrx%=Mousex
    Ptry%=Mousey
  Endif
Until A$<>""
If A$="A" Then
  Gosub Addseed
Endif
If A$="C" Then
  Gosub Convert
Endif
If A$="E" Then
  Gosub Editarr
Endif
If A$="S" Then
  Gosub Save
Endif
If A$=Chr$(0)+Chr$(31) Then
  Gosub Changename
Endif
If A$="N" Then
  Cls
  Goto Begin
Endif
If A$=Chr$(13) Or A$=Chr$(32) Or A$=Chr$(0)+Chr$(80) Then
  Add Start%,24
  If Start%>29999 Then
```

```
        Inc Seg%
        Gosub Get_new_seg
        Start%=0
      Endif
    Endif
    If A$=Chr$(0)+Chr$(72) Then
      Sub Start%,24
      If Start%<0 And Seg%>1 Then
        Dec Seg%
        Gosub Get_new_seg
        Start%=29977
      Else
        Start%=0
      Endif
    Endif
    If A$=Chr$(0)+Chr$(71) Then
      Seg%=1
      Gosub Get_new_seg
      Start%=0
    Endif
    If A$=Chr$(0)+Chr$(119) Then
      Start%=0
    Endif
    If A$=Chr$(0)+Chr$(82) Then
      Start%=Min(29977,(Order%(0)\24)*24)
    Endif
    If A$=Chr$(0)+Chr$(77) Then
      If Glen%<30001 Then
        Start%=(Len%\24)*24
      Else
        Seg%=(Glen%-2)/29999+1
        If Seg%<>Segment%/29999+1 Then
          Gosub Get_new_seg
        Endif
        Start%=(Len%\24)*24
      Endif
    Endif
    Exit If A$=Chr$(27)
  Loop
 Endif
 Cls
Return
Procedure Viewarrscreen
  Cls
  Box 250,75,600,279
  Text 280,93,"Up arrow - Page up"
  Text 280,109,"Down arrow - Page down"
  Text 280,125,"<Space>, <CR> - same as Down arrow"
  Text 280,141,"<Home> - Top of array"
  Text 280,157,"Left arrow - Last page of array"
  Text 280,173,"<Ctrl> <Home> - First page of segment"
  Text 280,189,"<Insert> - Last page of current segment"
```

```
  Text 280,205,"<Esc> - Main menu"
  Text 280,221,"<Shift> C - Convert file"
  Text 280,237,"<Shift> A - Add seed point to file"
  Text 280,253,"<Shift> S - Save modified file"
  Text 280,269,"<Alt> S - Change filename and save"
  Start%=0
  If Old! Then
    Print At(4,1);"N";
    Print At(9,1);"Jumps";
    Print At(19,1);"X";
    Print At(23,1);"Y";
    Print At(54,1);"Old style array"
  Else
    Print At(4,1);"N";
    Print At(14,1);"X";
    Print At(18,1);"Y";
    Print At(54,1);"New style array"
  Endif
  Print At(32,1);Arr$
  If Changed! Then
    Print At(54,2);"* File Changed!!!"
  Endif
Return
Procedure Get_new_seg
  If Changed! Then
    Print At(52,20);"Writing changed segment..."
    Gosub Save
  Endif
  If Seg%=1 Then
    Arr$=Left$(Arr$,Len(Arr$)-1)+"R"
  Else
    Arr$=Left$(Arr$,Len(Arr$)-1)+Str$(Seg%)
  Endif
  Arrayfill Order%(),0
  Segment%=29999*(Seg%-1)
  Print At(32,3);"Loading segment"'Seg%;"... Please wait."
  Bload Arr$,Nptr%
  Len%=Order%(0)
  Gosub Viewarrscreen
Return
Procedure Addseed
  Local Seedlocation,A%
  Seedlocation=205000
  If Old! Then
    A%=Optr%
  Else
    A%=Nptr%
  Endif
  Cls
  Print "I'm checking the length block of ";Arr$''"=";Lpeek(A%)
  If Lpeek(A%)>30001 Then
    Print "I'm resetting the length block to 30001"
```

```
    Lpoke A%,30001
    Changed!=True
  Endif
  If Old! Then
    Seedlocation=(Seedlocation\1024)*512+(Seedlocation Mod 1024)
    If Order(2)<>Seedlocation Then
      Print "I'm seeding the array"
      Bmove A%+16,A%+22,6*(Order(0)-1)
      Order(0)=Order(0)+1
      Inc Len%
      Order(2)=Seedlocation
      Changed!=True
      Print Arr$;" has been seeded."
    Else
      Print "This file appears to be seeded, first location is ";
      Print Order%(2)\512;",";Order%(2) Mod 512
    Endif
  Else
    If Order%(2)<>Seedlocation Then
      Print "I'm seeding the array"
      Bmove A%+12,A%+16,4*(Order%(0)-1)
      Order%(0)=Order%(0)+1
      Inc Len%
      Order%(2)=Seedlocation
      Changed!=True
      Print Arr$;" has been seeded."
    Else
      Print "This file appears to be seeded, first location is ";
      Print Order%(2)\1024;",";Order%(2) And 1023
    Endif
  Endif
  Print "Hit any key to continue."
  Repeat
  Until Inkey$<>""
  Gosub Viewarrscreen
Return
Procedure Convert
  Local A$
  Cls
  If Not Old! Then
    Print "This file appears to be converted already!"
    Print "New format N=";Order%(0)-1''"Rmax=";Order%(1)
    Input "Should I convert it anyway (Y or N)? ",A$
  Else
    A$="Y"
  Endif
  If (Asc(A$) And 223)=89 Then
    Lpoke Nptr%,30001
    Arrayfill Order%(),0
    Print "N x 1000:"''
    For X=0 To Order(0)
      If X>1
```

```
        Order%(X)=Order(X) Mod 262144
        Order%(X)=(Order%(X) And 261632)*2+(Order%(X) And 511)
      Else
        Order%(X)=Order(X)
      Endif
      If X Mod 1000=0
        Print X\1000'''
      Endif
    Next X
    Old!=False
    Changed!=True
    Print
    Print Arr$''has been converted to new format"
    Print "Hit any key to continue"
    Repeat
    Until Inkey$<>""
  Endif
  Gosub Viewarrscreen
Return
Procedure Editarr
  Local Dest%,Idx%
  Ptry%=(Ptry%\16)+1
  If Ptrx%<=151 And Ptry%>1 Then
    Idx%=Start%+Ptry%-3
    If Idx%>0 Then
      Print At(55,20);"^D to delete"'Idx%
      Print At(55,21);"<TAB> to insert blank"
      Print At(55,23);"<ESC> aborts."
      If Ptrx%>=96 And Ptrx%<=151 Then
        Print At(55,22);"or type number";
        If Ptrx%<=119 Then
          Dest%=1024
          Box 95,Ptry%*16-17,120,Ptry%*16
          Print '"for X"
          Gosub Getnum(13)
        Else
          Dest%=1
          Box 127,Ptry%*16-17,152,Ptry%*16
          Print '"for Y"
          Gosub Getnum(17)
        Endif
      Else
        Dest%=0
        Box -1,Ptry%*16-17,152,Ptry%*16
        Gosub Getnum(0)
      Endif
    Else
      If Idx%=-1 Then
        Print At(55,20);"Please don't change the"
        Print At(55,21);"length directly."
        Sound 1,15,1,1,10
        Sound 1,0
```

```
      Else
        Dest%=1
        Box 119,31,144,48
        Print At(55,20);"Type new maximum radius:"
        Gosub Getnum(16)
      Endif
    Endif
    Graphmode 1
    Deffill 0,1
    Pbox 430,300,639,399
    Color 0
    Draw 0,15 To 152,15
    Color 1
    If Changed! Then
      Print At(54,2);"* File Changed!!!"
    Endif
  Endif
Return
Procedure Getnum(Destcol%)
  Local Accept$,Num$,Done!
  Let Done!=False
  Num$=""
  Accept$=Chr$(4)+Chr$(9)+Chr$(27)
  If Destcol%<>0 Then
    Accept$=Accept$+Chr$(13)+Chr$(8)+"0123456789"
  Endif
  Repeat
    Repeat
      Ans$=Inkey$
      If Ans$<>"" And Instr(Accept$,Ans$)=0 Then
        Ans$=""
        Sound 1,15,4,8,2
        Sound 1,0
      Endif
    Until Ans$<>""
    On Instr(Accept$,Ans$) Gosub Delentry,Insspace,Esc,Endnum,Delchar
    If Instr(Accept$,Ans$)>5 Then
      Gosub Num
    Endif
  Until Done!
Return
Procedure Delentry
  If Order%(0)-Idx%-1>0 Then
    Bmove Nptr%+(Idx%+3)*4,Nptr%+(Idx%+2)*4,(Order%(0)-Idx%-1)*4
  Endif
  If Order%(0)-Idx%-1>=0 Then
    Order%(Order%(0))=0
    Dec Order%(0)
    Changed!=True
  Else
    Sound 1,15,6,7,2
    Sound 0,0
```

```
  Endif
  Let Done!=True
Return
Procedure Insspace
  If Order%(0)-Idx%>0 Then
    Bmove Nptr%+(Idx%+2)*4,Nptr%+(Idx%+3)*4,(Order%(0)-Idx%)*4
    Order%(Idx%+1)=0
    Inc Order%(0)
    Changed!=True
    Let Done!=True
  Endif
Return
Procedure Esc
  Let Done!=True
Return
Procedure Endnum
  Order%(Idx%+1)=Order%(Idx%+1) And (Not (1023*Dest%))
  Add Order%(Idx%+1),Val(Num$)*Dest%
  If Idx%+1>Order%(0) Then
    Order%(0)=Idx%+1
    Len%=Order%(0)
  Endif
  Changed!=True
  Let Done!=True
Return
Procedure Delchar
  If Num$<>""
    Num$=Left$(Num$,Len(Num$)-1)
    Print At(Destcol%,Idx%-Start%+3);Using "###",Val(Num$)
  Else
    Sound 1,15,6,7,2
    Sound 0,0
  Endif
Return
Procedure Num
  If Len(Num$)<3 Then
    Num$=Num$+Ans$
    Print At(Destcol%,Idx%-Start%+3);Using "###",Val(Num$)
  Else
    Print At(55,20);"3 Digits Only                "
    Sound 1,15,6,7,2
    Sound 0,0
  Endif
Return
Procedure Parsefilename(Fn$)
  Local First%,Last%,X%
  Pathname$=Left$(Fn$,Instr(Fn$,":"))
  First%=Instr(Fn$,"\")
  For X%=Len(Fn$) Downto 1
    If Mid$(Fn$,X%,1)="\"
      Last%=X%
    Endif
```

```
    Exit If Mid$(Fn$,X%,1)="\"
  Next X%
  Pathname$=Pathname$+Mid$(Fn$,First%,Last%-First%)
  File$=Mid$(Fn$,Last%+1)
Return
Procedure Loader
  Print At(1,3);"Select array:      "
  Fileselect "\*.ARR","SEED.ARR",File$
  If File$<>"" Then
    Bload File$,Lpeek(Arrptr(Order%()))
  Endif
Return
Procedure Save
  Local A%,F1%,F2%,Tlen%
  Tlen%=Min(Len%,30000)
  A%=Nptr%
  F1%=4
  F2%=8
  If Old! Then
    A%=Optr%
    F1%=6
    F2%=10
  Endif
  Print At(32,2);"Saving array to"'Arr$
  Bsave Arr$,A%,Tlen%*F1%+F2%
  Changed!=False
  Gosub Viewarrscreen
Return
Procedure Changename
  Local Tmp$
  Tmp$=File$
  Gosub Parsefilename(Arr$)
  Print At(32,2);"File to save array to:"
  Fileselect "\*.ARR",Arr$,Arr$
  Gosub Parsefilename(Arr$)
  If Instr(File$,".")=0 Then
    Arr$=Arr$+".ARR"
  Endif
  File$=Tmp$
  Gosub Save
Return
Procedure Drawscreen
  If File$="" Then
    Gosub Loader
  Endif
  Cls
  Hidem
  Graphmode 3
  Color 1
  Line 400,0,400,399
  Deftext 0,16,0,32
  Text 455,45,"Animator"
```

```
  Text 455,375,"Animator"
  Deftext 0,0,0,13
  Print At(52,1);File$
  Text 410,77,"$ - Reverse growth direction"
  Text 410,109,"<Enter> - Continue automatic"
  Text 490,125,"growth"
  Text 410,157,". - Stop automatic growth"
  Text 410,189,"( - White background"
  Text 410,221,") - Black background"
  Text 410,253,"Any other key - Single step"
  Text 538,269,"in Stop mode"
  Text 410,301,"/ - Fill in to current pixel"
  Text 410,333,"<Undo> - Exit/abort animator"
Return
Procedure Plot(Start%,Finish%,Direction%,Width%)
  Local Wait!,S$
  X%=Start%
  Repeat
    If (Direction%>0 And X%<=Finish%) Or (Direction%<0 And X%>=Finish%) Then
      Plot Order%(X%)\1024,Order%(X%) And 1023
      If Width%>0 And X%>Width%+1 Then
        Plot Order%(X%-Width%)\1024,Order%(X%-Width%) And 1023
      Endif
      Add X%,Direction%
    Endif
    A$=Inkey$
    If Wait! Then
      Repeat
        A$=Inkey$
      Until A$<>""
    Endif
    If A$<>"" Then
      If A$="$" Then
        Plot Order%(X%)\1024,Order%(X%) And 1023
        If Width%>0 And X%>Width%+1 Then
          Plot Order%(X%-Width%)\1024,Order%(X%-Width%) And 1023
        Endif
        Swap Start%,Finish%
        Mul Direction%,-1
      Endif
      If A$="." Then
        Wait!=-1
      Endif
      If A$=Chr$(13) Then
        Wait!=0
      Endif
      If A$="(" Then
        Setcolor 0,1
      Endif
      If A$=")" Then
        Setcolor 0,0
      Endif
```

```
      If A$="/" Then
        Sget S$
        A%=X%
        Gosub Plot(2,X%,1,0)
        Sput S$
        X%=A%
        A$=""
      Endif
    Endif
  Until A$=Chr$(0)+Chr$(97)
Return
Procedure Animate
  Gosub Drawscreen
  Gosub Plot(2,Order%(0),1,0)
  Setcolor 0,1
  Showm
Return
Procedure Involute
  Gosub Drawscreen
  Gosub Plot(Order%(0),2,-1,0)
  Setcolor 0,1
  Showm
Return
Procedure Zonal
  Cls
  Print At(10,12);
  Input "Enter number of pixels to display in deposition zone";Width%
  Gosub Drawscreen
  Gosub Plot(2,Order%(0),1,Width%)
  Setcolor 0,1
  Showm
Return

' Coremenu Program
Version=5.6
Revdate$="2 Oct 88"
Dim Order%(32000) ! Make room for FHG arrays too.
Dim Results(1,400)
Dim Power(1,400)
Dim Std_graph%(12)
Dim Menu$(50)
Let Menu$(0)="Desk"
Let Menu$(1)="  Utilities info"
Let Menu$(2)="--------------------"
For I=3 To 9
  Let Menu$(I)=Str$(I)
Next I
Do
  Inc I
  Read Menu$(I)
  Exit If Menu$(I)="*"
Loop
```

```
Data "Exit"," Quit ","","ARR Funcs"," Autocorrelation Vectors"
Data " Mass Distribution in X and Y"
Data " Mass Distribution in R and Theta "
Data "","FHG Funcs"," Mass Distribution in X and Y"
Data " Mass Distribution in R and Theta "
Data "","GYR Funcs"," Prep for new file ","----------------------"
Data " View ?YA File"," Plot"," Regression","","",""
Menu Menu$()
On Menu Gosub Handle_it_for_me
Print At(1,3);"Correlation functions, Version"'Version;","'Revdate$
Do
  On Menu
Loop
Procedure Handle_it_for_me
  Cls
  If Menu(0)=1 Then
    Gosub Give_info
  Else
    On Menu(0)-11 Gosub Quit,D,D,Auto,Massxy,Massrt,D,D,Fmassxy,Fmassrt
    If Menu(0)>21 Then
      On Menu(0)-21 Gosub Dummy,D,Prep_for_new,Dummy,Viewdat,Plotya,Regression
    Endif
  Endif
  Menu Menu$()
  Print At(1,3);"Select function:"
Return
Procedure Give_info
Return
Procedure Quit
  Menu Kill
  Edit
Return
Procedure Auto
  Local A%,B%,C%,I%,J%
  Gosub Loader
  Cls
  If File$<>"" Then
    Input "Input n:";N%
    Print "Calculating Autocorrelation vectors"
    For A%=2 To Order%(0)-N%
      If A% Mod 100=0 Then
        Print At(1,6);"N=";A%
      Endif
      J%=Order%(A%+N%)
      I%=Order%(A%)
      Xj%=J%\1024
      Xi%=I%\1024
      Yj%=J% And 1023
      Yi%=I% And 1023
      Order%(A%-1)=Int(Sqr((Yj%-Yi%)^2+(Xj%-Xi%)^2))
    Next A%
    Cls
```

```
    Order%(Order%(0))=0
    Order%(Order%(0)-1)=0
    For A%=Order%(0)-N% To Order%(0)
      Order%(A%)=0
    Next A%
    Gosub Set_up(20,380,0,600,350,0,100,50,1,1)
    Gosub Axes(*Std_graph%())
    Gosub Label_hashes(*Std_graph%())
    Graphmode 1
    For Begin%=1 To ((Order%(0)\600)+1)*600 Step 600
      Text 20,396,Str$(Begin%)
      Text 580,396,Str$(Begin%+599)
      For A%=1 To 600
        Color 1
        Draw A%+20,380 To A%+20,380-Order%(A%+Begin%-1)
        Color 0
        Draw A%+20,379-Order%(A%+Begin%-1) To A%+20,0
      Next A%
      Color 1
      Repeat
      Until Inkey$<>""
    Next Begin%
  Endif
Return
Procedure Massxy
  Gosub Loader
  Cls
  If File$<>"" Then
    Print At(1,5);"Calculating Center of Mass ... Please wait"
    Gosub Centerofmass(*Order%())
    Print "Center of mass at X=";Avex''"Y=";Avey
    Print At(1,5);"Calculating X and Y density functions     "
    Print "Processed 0     of"'Order%(0)'"Points"
    For X%=2 To Order%(0)
      If X% Mod 100=0 Then
        Print At(11,6);X%
      Endif
      Inc Results(0,Order%(X%)\1024)
      Inc Results(1,Order%(X%) And 1023)
    Next X%
    Gosub Set_up(320,240,220,220,200,0,100,180,1,1)
    Gosub Dispxy
  Endif
Return
Procedure Fmassxy
  Local Coil%,Exp%,Freq%,Iter%,Sixbit%,Uncoil%,Xpixel%,Ypixel%
  Gosub Floader
  Cls
  If File$<>"" Then
    Print At(1,5);"Calculating Center of Mass ... Please wait"
    Gosub Fcenterofmass(*Order%())
    Print "Center of mass at X=";Avex''"Y=";Avey
```

```
    Print At(1,5);"Calculating X and Y density functions      "
    Print "Processed 0      of 160000 Points"
    For Iter%=1 To 32000
      Coil%=(Iter%-1)*5
      Freq%=Order%(Iter%)
      If Coil% Mod 100=0 Then
        Print At(11,6);Coil%
      Endif
      If Freq%<>0 Then
        For Sixbit%=0 To 4
          Exp%=64^Sixbit%
          Freq%=Order%(Iter%) And (63*Exp%)
          If Freq%<>0 Then
            Div Freq%,Exp%
            Uncoil%=Coil%+Sixbit%
            Xpixel%=Uncoil%\400
            Ypixel%=Uncoil% Mod 400
            Add Results(0,Xpixel%),Freq%
            Add Results(1,Ypixel%),Freq%
            Option "U1"
          Endif
        Next Sixbit%
      Endif
    Next Iter%
    Gosub Set_up(320,240,220,220,200,0,100,180,1,-20)
    Gosub Dispxy
  Endif
Return
Procedure Dispxy
  Tester:
  Cls
  Graphmode 1
  If Loaded! Then
    File$=Dat$
  Endif
  Gosub Parsefilename(File$)
  Gosub Axes(*Std_graph%())
  Gosub Label_hashes(*Std_graph%())
  Std_graph%(0)=1
  Std_graph%(3)=0
  Std_graph%(4)=399
  Std_graph%(5)=-200-Int(Avex)
  Gosub Plot(*Std_graph%())
  Lbl$="Deposit "+File$+" Mass Distribution Function in X"
  A%=40-Len(Lbl$)/2
  Print At(A%,22);Lbl$
  Print At(52,4);"Center Of Mass:"
  Print At(52,5);Avex;",";Avey
  Gosub Cmd_driver("")
  Cls
  Gosub Axes(*Std_graph%())
  Gosub Label_hashes(*Std_graph%())
```

```
  Std_graph%(0)=1+4
  Std_graph%(3)=0
  Std_graph%(4)=399
  Std_graph%(5)=-200+Int(Avey)
  Gosub Plot(*Std_graph%())
  Lbl$=Left$(Lbl$,Len(Lbl$)-1)+"Y"
  Print At(A%,22);Lbl$
  Print At(52,4);"Center Of Mass:"
  Print At(52,5);Avex;","'Avey
  Gosub Cmd_driver("")
Return
Procedure Massrt
  Local A%,B%,R%,Th%,Rav,Rm%,Rho%
  Graphmode 1
  Gosub Loader
  Cls
  If File$<>"" Then
    Print At(1,5);"Calculating Center of Mass ... Please wait"
    Gosub Centerofmass(*Order%())
    Print "Center of mass at X=";Avex''"Y=";Avey
    Print At(1,5);"Calculating R and Theta density functions "
    Print "Processed 0     of"'Order%(0)'"Points"
    For X%=2 To Order%(0)
      If X% Mod 100=0 Then
        Print At(11,6);X%
      Endif
      A%=Order%(X%)\1024
      B%=Order%(X%) And 1023
      R%=Int(Sqr((A%-Avex-200)^2+(B%+Avey-200)^2))
      Th%=Trunc(Atn((B%+Avey-200)/(A%-Avex-200+0.00001))*57.3)
      Add Th%,180
      If A%-Avex-200<0 Then
        Add Th%,180
      Endif
      If Th%>=360 Then
        Sub Th%,360
      Endif
      Inc Results(0,R%)
      Inc Results(1,Th%)
    Next X%
    Gosub Set_up(100,250,0,220,220,0,50,100,1,1)
    Gosub Disprt
  Endif
Return
Procedure Fmassrt
  Local Coil%,Exp%,Freq%,Iter%,Sixbit%,Uncoil%,Xpixel%,Ypixel%
  Gosub Floader
  Cls
  If File$<>"" Then
    Print At(1,5);"Calculating Center of Mass ... Please wait"
    Gosub Fcenterofmass(*Order%())
    Print "Center of mass at X=";Avex''"Y=";Avey
```

```
    Print At(1,5);"Calculating R and Theta density functions "
    Print "Processed 0      of 160000 Points"
    For Iter%=1 To 32000
      Coil%=(Iter%-1)*5
      Freq%=Order%(Iter%)
      If Coil% Mod 100=0 Then
        Print At(11,6);Coil%
      Endif
      If Freq%<>0 Then
        For Sixbit%=0 To 4
          Exp%=64^Sixbit%
          Freq%=Order%(Iter%) And (63*Exp%)
          If Freq%<>0 Then
            Div Freq%,Exp%
            Uncoil%=Coil%+Sixbit%
            Xpixel%=Uncoil%\400
            Ypixel%=Uncoil% Mod 400
            R%=Int(Sqr((Xpixel%-Avex-200)^2+(Ypixel%+Avey-200)^2))
            Th%=Trunc(Atn((Ypixel%+Avey-200)/(Xpixel%-Avex-200+0.00001))*57.3)
            Add Th%,180
            If Xpixel%-Avex-200<0 Then
              Add Th%,180
            Endif
            If Th%>=360 Then
              Sub Th%,360
            Endif
            Add Results(0,R%),Freq%
            Add Results(1,Th%),Freq%
            Option "U1"
          Endif
        Next Sixbit%
      Endif
    Next Iter%
    Gosub Set_up(100,250,0,220,220,0,50,100,-20,-20)
    Gosub Disprt
  Endif
Return
Procedure Disprt
  Cls
  If Loaded! Then
    File$=Dat$
  Endif
  Gosub Parsefilename(File$)
  Testing:
  Sclx%=Std_graph%(9)
  Std_graph%(9)=1
  Gosub Axes(*Std_graph%())
  Gosub Label_hashes(*Std_graph%())
  Std_graph%(0)=1+8
  Std_graph%(3)=0
  Std_graph%(4)=201
  Std_graph%(5)=0
```

```
  Gosub Plot(*Std_graph%())
  Lbl$="Deposit "+File$+" Mass Distribution Function in R"
  A%=40-Len(Lbl$)/2
  Print At(A%,22);Lbl$
  Print At(52,4);"Center Of Mass:"
  Print At(52,5);Avex;",",'Avey
  Print At(52,6);Chr$(255);"="'Ave
  Graphmode 3
  Text 408,93,"r"
  Gosub Bargraph(100+Ave,250,0,-20)
  Gosub Bargraph(100+Mode%,250,0,-28)
  Graphmode 3
  Deftext 1,0,0,6
  Text 100+Ave-3,250+20+10,"r"
  Text 100+Ave-3,250+20+9,Chr$(255)
  Text 100+Mode%-15,250+20+18,"mode"
  Deftext 1,0,0,13
  Graphmode 1
  Gosub Cmd_driver("r")
  Cls
  Gosub Set_up(220,200,180,180,180,180,150,150,Sclx%,Std_graph%(10))
  Gosub Axes(*Std_graph%())
  Gosub Label_hashes(*Std_graph%())
  Std_graph%(0)=1+4+32 !1 color, 4 upper, 32 polar
  Std_graph%(3)=0
  Std_graph%(4)=360
  Std_graph%(5)=0
  Gosub Plot(*Std_graph%())
  Lbl$=Left$(Lbl$,Len(Lbl$)-1)+"Theta"
  A%=Len(Lbl$)
  Print At(52,2);Left$(Lbl$,A%-35);
  Print At(52,3);Mid$(Lbl$,A%-34,17)
  Print At(52,4);Right$(Lbl$,17)
  Print At(52,6);"Center Of Mass:"
  Print At(52,7);Avex;",",'Avey
  Gosub Cmd_driver("")
  Let Loaded!=True
  Len%=360
Return
Procedure Cmd_driver(Char$)
  Local Char%
  Char%=-(Char$<>"")
  Do
    Gosub Pmline("S = Smoothing  M = Modes  E = Edit Screen  <ESC> aborts  <CR> stores screen")
    Repeat
      A$=Inkey$
    Until A$<>""
    Exit If A$=Chr$(27)
    If (Asc(A$) And 95)=83 Then
      On Char% Gosub Label_ave
      Gosub Smoothing
      On Char% Gosub Label_ave
```

```
    Endif
    On -((Asc(A$) And 95)=77) Gosub Modes
    On -((Asc(A$) And 95)=69) Gosub Ed
    If A$=Chr$(13) Then
      Gosub Pmline("")
      Sget Scr$
      Fileselect "\*.GRF","",Dat$
      Sput Scr$
      If Instr(Dat$,".GRF")=0 Then
        Gosub Parsefilename(Dat$)
        Dat$=Pathname$+"\"+Left$(File$+".",Instr(File$+".","."))+"GRF"
      Endif
      Bsave Dat$,Xbios(2),32000
      Print At(1,2);"Saved as"'Dat$
    Endif
    Exit If A$=Chr$(13)
  Loop
Return
Procedure Label_ave
  Graphmode 3
  Deftext 1,0,0,6
  Text 100+Ave-3,250+20+10,Char$
  Text 100+Ave-3,250+20+9,Chr$(255)
  Text 100+Mode%-15,250+20+18,"mode"
  Gosub Bargraph(100+Ave,250,0,-20)
  Gosub Bargraph(100+Mode%,250,0,-28)
  Graphmode 1
  Print At(55,6);Ave
  Deftext 1,0,0,13
Return
Procedure Smoothing
  Gosub Moving_ave
Return
Procedure Moving_ave
  Local X%,N%,Cum%,Nr%,Upper!,Split!
  Interval%=5
  Upper!=(Std_graph%(0) And 4)
  Split!=(Std_graph%(0) And 128)
  Std_graph%(0)=Std_graph%(0) And 252
  Gosub Plot(*Std_graph%())
  If Not Split! Then
    For X%=Std_graph%(3) To Std_graph%(4)
      Cum%=0
      Nr%=0
      For N%=-Interval% To Interval%
        If X%+N%>=0 And X%+N%<400 Then
          Add Cum%,Results(Abs(Upper!),X%+N%)
          Inc Nr%
        Endif
      Next N%
      Results(Abs(Upper!),X%)=Int(Cum%/Nr%)
    Next X%
```

```
  Else
  Endif
  Add Std_graph%(0),1
  Gosub Plot(*Std_graph%())
Return
Procedure Ed
  Local X,Y,K,A$
  Gosub Pmline("<Left Button> - Add Text    <Right Button> - Move Area     <ESC> exits")
  Repeat
    Mouse X,Y,K
    On K Gosub Text,Move
    If K Then
      Gosub Pmline("<Left Button> - Add Text    <Right Button> - Move Area     <ESC> exits")
      K=0
    Endif
    A$=Inkey$
  Until (A$=Chr$(13)) Or (A$=Chr$(27))
Return
Procedure Prep_for_new
  Let Loaded!=False
  Dat$=""
Return
Procedure Viewdat
  Local Start%,Line%,A$,Old!,Optr%,Nptr%
  Optr%=Lpeek(Arrptr(Results()))
  If Not Loaded! Then
    Gosub Dloader
  Endif
  If Dat$<>"" Then
    Gosub Viewdatscreen
    Do
      For Line%=Start% To Start%+23
        If Line%<401 Then
          Print At(1,Line%-Start%+2);Using "  ###  ",Line%+Base%;
          Print At(12,Line%-Start%+2);Using "#####.########",Results(0,Line%+Base%);
          Print At(28,Line%-Start%+2);Using "###.##########",Results(1,Line%+Base%);
        Endif
        If Line%>400 Then
          Print At(1,Line%-Start%+1);"                                             ";
        Endif
      Next Line%
      Repeat
        A$=Inkey$
      Until A$<>""
      If A$=Chr$(13) Or A$=Chr$(32) Or A$=Chr$(0)+Chr$(80) Then
        Add Start%,25
        If Start%>375 Then
          Start%=375
        Endif
      Endif
      If A$=Chr$(0)+Chr$(72) Then
        Sub Start%,25
```

```
        If Start%<0 Then
          Start%=0
        Endif
      Endif
      If A$=Chr$(0)+Chr$(71) Then
        Start%=0
      Endif
      If A$=Chr$(0)+Chr$(77) Then
        Start%=(Len%\25)*25
      Endif
      Exit If A$=Chr$(27)
    Loop
  Endif
  Cls
Return
Procedure Viewdatscreen
  Cls
  Box 384,75,600,185
  Text 392,93,"Up arrow - Page up"
  Text 392,109,"Down arrow - Page down"
  Text 392,125,"<Space>, <CR> - same as Down arrow"
  Text 392,141,"<Home> - Top of array"
  Text 392,157,"Left arrow - Last page of array"
  Text 392,173,"<Esc> - Main menu"
  Start%=0
  Print At(1,1);"N of"'Len%
  Print At(12,1);Zero$;
  Print At(28,1);One$;
  Print At(49,1);Dat$
Return
Procedure Plotya
  If Not Loaded! Then
    Gosub Dloader
  Endif
  If Dat$<>"" Then
    If Typ$="XYA" Then
      Gosub Dispxy
    Else
      If Typ$="RYA" Then
        Gosub Disprt
      Else
        Cls
        Gosub Axes(*Std_graph%())
        Gosub Label_hashes(*Std_graph%())
        Gosub Plot(*Std_graph%())
        Gosub Regression
        Gosub Cmd_driver("")
      Endif
    Endif
  Endif
Return
Procedure Regression
```

```
  Local Cutoff,Cutoff%
  If Not Loaded! Then
    Gosub Dloader
  Endif
  If Dat$<>"" Then
    Power(0,0)=0
    Input "Lower cutoff for regression (Ln(N) in linear region, 0 for all)";Cutoff
    If Cutoff=0 Then
      Cutoff%=1
    Else
      Cutoff%=Int((82*Exp(Cutoff))^0.41666666667)-10
    Endif
    For N%=Cutoff% To Len%
      Inc Power(0,0)
      Power(0,Power(0,0))=Results(0,N%)
      Power(1,Power(0,0))=Results(1,N%)
    Next N%
    Gosub Power
    Print " D = ";1/Slope
    Print "File = ";Dat$
  Endif
Return
Procedure Loader
  Print At(1,3);"Select array:     "
  Fileselect "\*.ARR","SEED.ARR",File$
  If File$<>"" Then
    Arrayfill Results(),0
    Arrayfill Order%(),0
    Bload File$,Lpeek(Arrptr(Order%()))
  Endif
Return
Procedure Floader
  Print At(1,3);"Select array:     "
  Fileselect "\*.FHG","LONGLIST.FHG",File$
  If File$<>"" Then
    Arrayfill Results(),0
    Arrayfill Order%(),0
    Bload File$,Lpeek(Arrptr(Order%()))
  Endif
Return
Procedure Dloader
  Local Tf$,Tp$
  Do
    Print At(1,3);"Select array:     "
    Fileselect "\*.?YA","",Dat$
    If Dat$<>"" Then
      Tf$=File$
      Tp$=Pathname$
      Gosub Parsefilename(Dat$)
      Typ$=Mid$(File$,Instr(File$,".")+1)
      Datpath$=Pathname$
      Pathname$=Tp$
```

```
    File$=Tf$
  Endif
  Exit If Instr("PYAGYAXYARYA",Typ$)>0 Or (Dat$="")
  Print At(1,1);Dat$'"is an unknown type of data file.  Please"
  Print "enter file with .PYA, .GYA, .XYA, or .RYA extension."
Loop
If Dat$<>"" Then
  Arrayfill Results(),0
  Bload Dat$,Lpeek(Arrptr(Results()))
  Let Loaded!=True
  If Instr("PG",Left$(Typ$,1))>0 Then
    Len%=Results(0,0)
    Base%=1
    Split!=True
    If Typ$="PYA" Then
      Zero$="Radius of zone"
      X$="Ln R"
      Let One$="Filled Area"
      Y$="Ln C(R)"
      Gosub Set_up(100,300,0,430,230,0,50,50,50,50)
      Std_graph%(0)=1+64+128+256
      Std_graph%(3)=Base%
      Std_graph%(4)=Len%
      Std_graph%(5)=0
    Else
      Zero$="# of Deposits"
      X$="Ln N"
      Let One$="Radius of Gyration"
      Y$="Ln Rg"
      Gosub Set_up(100,300,0,430,230,0,40,40,40,40)
      Std_graph%(0)=1+64+128+256
      Std_graph%(3)=Base%
      Std_graph%(4)=Len%
      Std_graph%(5)=0
    Endif
  Else
    Base%=0
    Split!=False
    If Typ$="XYA" Then
      Len%=400
      Len0%=400
      Len1%=400
      Zero$="Density in X"
      Let One$="Density in Y"
      X$="X"
      Y$="Density in X"
      Gosub Set_up(320,240,220,220,200,0,100,180,1,1)
      Std_graph%(0)=1
      Std_graph%(3)=Base%
      Std_graph%(4)=Len%
      Std_graph%(5)=0
    Endif
```

```
      If Typ$="RYA" Then
        Len%=360
        Len0%=201
        Len1%=360
        Zero$="Density in R"
        Let One$="Density in Theta"
        Gosub Set_up(100,250,0,220,220,0,50,100,1,1)
        Std_graph%(0)=1+8
        Std_graph%(3)=Base%
        Std_graph%(4)=Len0%
        Std_graph%(5)=0
      Endif
    Endif
  Endif
Return
Procedure Parsefilename(Fn$)
  Local First%,Last%,X%
  Pathname$=Left$(Fn$,Instr(Fn$,":"))
  First%=Instr(Fn$,"\")
  For X%=Len(Fn$) Downto 1
    If Mid$(Fn$,X%,1)="\"
      Last%=X%
    Endif
    Exit If Mid$(Fn$,X%,1)="\"
  Next X%
  Pathname$=Pathname$+Mid$(Fn$,First%,Last%-First%)
  File$=Mid$(Fn$,Last%+1)
Return
Procedure Set_up(Orx%,Ory%,Lendx%,Rendx%,Tendy%,Bendy%,Hashx%,Hashy%,Sclx%,Scly%)
  Std_graph%(1)=Orx%
  Std_graph%(2)=Ory%
  Std_graph%(6)=65536*Lendx%+Rendx%
  Std_graph%(7)=65536*Tendy%+Bendy%
  Std_graph%(8)=65536*Hashx%+Hashy%
  Std_graph%(9)=Sclx%
  Std_graph%(10)=Scly%
Return
Procedure Axes(P.array)
  Local Orx%,Ory%,Lendx%,Rendx%,Tendy%,Bendy%,Hashx%,Hashy%
  Local A%,Length%
  Swap *P.array,Array%()
  Orx%=Array%(1)
  Ory%=Array%(2)
  Lendx%=Array%(6)\65536
  Rendx%=Array%(6) And 65535
  Tendy%=Array%(7)\65536
  Bendy%=Array%(7) And 65535
  Hashx%=Array%(8)\65536
  Hashy%=Array%(8) And 65535
  Gosub Drawaxes(*Array%())
  Length%=10
  Defline 1,1,0,0
```

```
  If Hashx%<>0 Then
    For A%=Orx% To Orx%-Lendx% Step -Hashx%
      Draw A%,Ory%-Length% To A%,Ory%+Length%
    Next A%
    For A%=Orx% To Orx%+Rendx% Step Hashx%
      Draw A%,Ory%-Length% To A%,Ory%+Length%
    Next A%
  Endif
  If Hashy%<>0 Then
    For A%=Ory% To Ory%-Tendy% Step -Hashy%
      Draw Orx%+Length%,A% To Orx%-Length%,A%
    Next A%
    For A%=Ory% To Ory%+Bendy% Step Hashy%
      Draw Orx%+Length%,A% To Orx%-Length%,A%
    Next A%
  Endif
  Swap *P.array,Array%()
Return
Procedure Drawaxes(P.array)
  Local Orx%,Ory%,Lendx%,Rendx%,Tendy%,Bendy%,Hashx%,Hashy%
  Swap *P.array,Array%()
  Orx%=Array%(1)
  Ory%=Array%(2)
  Lendx%=Array%(6)\65536
  Rendx%=Array%(6) And 65535
  Tendy%=Array%(7)\65536
  Bendy%=Array%(7) And 65535
  Defline 1,1,1,1
  If Lendx%=0 Then
    Defline 1,1,0,1
  Endif
  If Rendx%=0 Then
    Defline 1,1,1,0
  Endif
  Draw Orx%-Lendx%,Ory% To Orx%+Rendx%,Ory%
  Defline 1,1,1,1
  If Tendy%=0 Then
    Defline 1,1,0,1
  Endif
  If Bendy%=0 Then
    Defline 1,1,1,0
  Endif
  Draw Orx%,Ory%-Tendy% To Orx%,Ory%+Bendy%
  Swap *P.array,Array%()
  Defline 1,1,0,0
Return
Procedure Label_hashes(P.array)
  Local Hash%,Loend%,Hiend%,Scale,A%,Lbl$
  Swap *P.array,Array%()
  Deftext 1,0,0,6
  Hash%=Array%(8)\65536
  Loend%=Array%(6)\65536
```

```
  Hiend%=Array%(6) And 65535
  Scale=Array%(9)
  If Scale<0 Then
    Scale=-1/Scale
  Endif
  A%=Hash% ! To Hiend% Step Hash%
  While A%<Hiend%
    Lbl$=Str$(A%/Scale)
    Text Array%(1)+A%-Len(Lbl$)*4,Array%(2)+18,Lbl$
    If A%<Loend% Then
      Text Array%(1)-A%-(Len(Lbl$)+1)*4,Array%(2)+18,"-"+Lbl$
    Endif
    Add A%,Hash%
  Wend
  Hash%=Array%(8) And 65535
  Hiend%=Array%(7)\65536
  Loend%=Array%(7) And 65535
  Scale=Array%(10)
  If Scale<0 Then
    Scale=-1/Scale
  Endif
  A%=Hash% ! To Hiend% Step Hash%
  While A%<Hiend%
    Lbl$=Str$(A%/Scale)
    Text Array%(1)-10-Len(Lbl$)*8,Array%(2)-A%+4,Lbl$
    If A%<Loend% Then
      Text Array%(1)-10-(Len(Lbl$)+1)*8,Array%(2)+A%+4,"-"+Lbl$
    Endif
    Add A%,Hash%
  Wend
  Swap *P.array,Array%()
  Deftext 1,0,0,13
Return
Procedure Plot(P.array)
  Local Flags%,Upper!,Collect!,Xplot!,Polar!,Line!,Split!,Logs!,Count%
  Local Datum,Indep,Xbeg%,Xend%,Ybeg%,Yend%,Max%,Sum%,Nave%
  Local Sclx,Scly
  Swap *P.array,Array%()
  Flags%=Array%(0)
  Upper!=Flags% And 4
  Collect!=Flags% And 8
  Xplot!=Flags% And 16
  Polar!=Flags% And 32
  Let Line!=Flags% And 64
  Split!=Flags% And 128
  Logs!=Flags% And 256
  Sclx=Array%(9)
  Scly=Array%(10)
  If Sclx<0 Then
    Sclx=-1/Sclx
  Endif
  If Scly<0 Then
```

```
  Scly=-1/Scly
Endif
If Collect! Then
  Sum%=0
  Nave%=0
Endif
Color Flags% And 1
Graphmode (Flags% And 2)+1
For Count%=Array%(3) To Array%(4)
  If Split! Then
    Indep=Results(0,Count%)
    Datum=Results(1,Count%)
  Else
    Indep=Count%
    Datum=Results(Abs(Upper!),Count%)
  Endif
  If Logs! Then
    Indep=Log(Indep)
    Datum=Log(Datum)
  Endif
  If Not Polar! Then
    Xbeg%=Array%(1)+Abs(Xplot!+1)*(Indep*Sclx+Array%(5))
    Ybeg%=Array%(2)+Abs(Xplot!)*(Indep*Scly+Array%(5))
    Xend%=Datum*Sclx*Abs(Xplot!)
    Yend%=Datum*Scly*Abs(Xplot!+1)
  Else
    Xbeg%=Array%(1)
    Ybeg%=Array%(2)
    Xend%=Datum*Scly*Cos((Indep*Sclx-180)*Pi/180)
    Yend%=-Datum*Scly*Sin((Indep*Sclx-180)*Pi/180)
  Endif
  If Line! Then
    If Count%=Array%(3) Then
      Draw Xbeg%+Xend%,Ybeg%-Yend%
    Else
      Draw  To Xbeg%+Xend%,Ybeg%-Yend%
    Endif
  Else
    Draw Xbeg%,Ybeg% To Xbeg%+Xend%,Ybeg%-Yend%
  Endif
  If Collect! Then
    If Datum%>Max% Then
      Max%=Datum%
      Mode%=Count%
    Endif
    Add Sum%,Results(Abs(Upper!),Count%)*Count%
    Add Nave%,Results(Abs(Upper!),Count%)
  Endif
Next Count%
If Collect! Then
  Ave=Sum%/Nave%
Endif
```

```
  If ((Flags% And 1) Or (Flags% And 2))=0 Then
    Color 1
    Graphmode 1
    Gosub Drawaxes(*Array%())
  Endif
  Swap *P.array,Array%()
Return
Procedure Bargraph(Xbeg%,Ybeg%,Xend%,Yend%)
  Draw Xbeg%,Ybeg% To Xbeg%+Xend%,Ybeg%-Yend%
Return
Procedure Pmline(Txt$)
  Deftext 1,0,0,6
  Graphmode 1
  Text 0,398,Space$(80)
  If Txt$<>"" Then
    Graphmode 4
    Txt$=Space$(40-(Len(Txt$)\2))+Txt$
    Text 0,398,Txt$+Space$(80-Len(Txt$))
  Endif
  Graphmode 1
  Showm
Return
Procedure Text
  Local X,Y,K,In$,Title$,Mse$,Bigmse$,Scr$,Size%,A$
  Size%=6
  Gosub Pmline("<Left> - Locate text line  T - Toggle print Size  <ESC> aborts")
  Deftext 1,0,0,Size%
  Graphmode 1
  Mse$=Mkl$(393224)+Mkl$(65536)+Mki$(1)+Mkl$(2080412160)+Mkl$(1811949568)
  Mse$=Mse$+Mkl$(671098880)+Mkl$(671116288)+Mkl$(-1845462016)+Mkl$(0)+Mkl$(0)+Mkl$(0)
  Mse$=Mse$+Mkl$(27648)+Mkl$(268439552)+Mkl$(268439552)+Mkl$(268439552)
  Mse$=Mse$+Mkl$(1811939328)+Mkl$(0)+Mkl$(0)+Mkl$(0)
  Bigmse$=Mkl$(65537)+Mki$(1)+Mkl$(1) ! Ref at 1,1; filler; standard colors 0,1
  Bigmse$=Bigmse$+Mkl$(1065361536)+Mkl$(545267840)+Mkl$(1530968992)
  Bigmse$=Bigmse$+Mkl$(-1147108448)+Mkl$(-1079984224)+Mkl$(1598038144)
  Bigmse$=Bigmse$+Mkl$(545267840)+Mkl$(1065353216)
  Bigmse$=Bigmse$+Mkl$(7936)+Mkl$(520101632)+Mkl$(612385856)+Mkl$(1145076800)
  Bigmse$=Bigmse$+Mkl$(1077952576)+Mkl$(545267456)+Mkl$(520101632)+Mkl$(0)
  Defmouse Mse$
  Repeat
  Until Mousek=0
  Repeat
    Mouse X,Y,K
    A$=Inkey$
    If (Asc(A$) And 95)=84 Then
      Size%=-(Size%-9.5)+9.5
      If Size%=13 Then
        Defmouse Bigmse$
      Else
        Defmouse Mse$
      Endif
      Deftext 1,0,0,Size%
```

```
    Endif
    Exit If A$=Chr$(27)
    If K=1 Then
      Sget Scr$
      Gosub Pmline("Type line    <Arrows> - Direction    <CR> - ends input    <ESC> aborts")
      Deftext 1,0,0,Size%
      Title$=""
      Text X,Y,Chr$(3)+" "
      Do
        Repeat
          In$=Inkey$
        Until In$<>""
        Exit If In$=Chr$(13) Or In$=Chr$(27)
        If Asc(In$)=0 Then
          On (Asc(Right$(In$,1))-71) Gosub Up,Dum,Dum,Back,Dum,For,Dum,Dum,Dn
        Endif
        If In$>Chr$(8) Then
          Title$=Title$+In$
        Endif
        If In$=Chr$(8) Then
          Title$=Left$(Title$,Max(Len(Title$)-1,0))
        Endif
        Text X,Y,Title$+Chr$(3)+" "
      Loop
      Text X,Y,Title$+" "
      If In$=Chr$(27) Then
        Sput Scr$
        In$=Chr$(13)
      Endif
    Endif
  Until In$=Chr$(13)
  Defmouse 0
Return
Procedure Up
  Text X,Y,Space$(Len(Title$)+1)
  Deftext 1,0,900,Size%
  Text X,Y,Title$+Chr$(3)+" "
Return
Procedure Dn
  Text X,Y,Space$(Len(Title$)+1)
  Deftext 1,0,2700,Size%
  Text X,Y,Title$+Chr$(3)+" "
Return
Procedure For
  Text X,Y,Space$(Len(Title$)+1)
  Deftext 1,0,0,Size%
  Text X,Y,Title$+Chr$(3)+" "
Return
Procedure Back
  Text X,Y,Space$(Len(Title$)+1)
  Deftext 1,0,1800,Size%
  Text X,Y,Title$+Chr$(3)+" "
```

```
Return
Procedure Move
  Local X,Y,K,A$,Mse$,X0%,Y0%
  Mouse X,Y,K
  Gosub Pmline("<Right> - opens box, release records area.")
  Graphmode 3
  X0%=X
  Y0%=Y
  While K=2
    Mouse X,Y,K
    Box X0%,Y0%,X,Y
    Box X0%,Y0%,X,Y
  Wend
  Get X0%,Y0%,X,Y,Mse$
  X0%=Min(X0%,X)
  Y0%=Min(Y0%,Y)
  Put X0%,Y0%,Mse$,6
  Gosub Pmline("<Left> - Places area   C - Copies area   D - Deletes area   <ESC> aborts")
  Hidem
  Graphmode 1
  Do
    Mouse X,Y,K
    Put X,Y,Mse$,6
    A$=Inkey$
    If (Asc(A$) And 95)=67 Then
      Put X0%,Y0%,Mse$,7
    Endif
    If (Asc(A$) And 95)=68 Then
      Put X,Y,Mse$,6
      K=1
    Endif
    If A$=Chr$(27) Then
      Put X,Y,Mse$,6
      Put X0%,Y0%,Mse$,7
      K=1
    Endif
    Exit If K=1 Or Mousek=1
    Put X,Y,Mse$,6
  Loop
  Showm
  Repeat
  Until Mousek=0
Return
Procedure Centerofmass(P.array)
  N%=0
  Swap *P.array,Avearray%()
  Print "Processed 0     of"'Avearray%(0)'"Points"
  Avex=Avearray%(2)\1024
  Avey=Avearray%(2) And 1023
  Do
    Inc N%
    If N% Mod 100=0 Then
```

```
      Print At(11,6);N%
    Endif
    Exit If Avearray%(N%+1)=0 And Avearray%(N%+2)=0
    Avex=(Avex*(N%-1)+(Avearray%(N%+1)\1024))/N%
    Avey=(Avey*(N%-1)+(Avearray%(N%+1) And 1023))/N%
  Loop
  Sub Avex,200
  Avey=200-Avey
  Swap *P.array,Avearray%()
Return
Procedure Fcenterofmass(P.array)
  Swap *P.array,Avearray%()
  Print "Processed 0      of 160000 Points"
  Count%=0
  N%=0
  Repeat
    Inc N%
  Until Avearray%(N%)<>0
  Sixbit%=0
  While (Avearray%(N%) And (63*64^Sixbit%))=0
    Inc Sixbit%
  Wend
  Uncoil%=(N%-1)*5+Sixbit%
  Exp%=64^Sixbit%
  Freq%=(Avearray%(N%) And (63*Exp%))/Exp%
  Avex=(Uncoil%\400)*Freq%
  Avey=(Uncoil% Mod 400)*Freq%
  Add Count%,Freq%
  Inc Sixbit%
  Repeat
    Exp%=64^Sixbit%
    Freq%=(Avearray%(N%) And (63*Exp%))/Exp%
    If Freq%>0 Then
      Avex=(Avex*Count%+(Uncoil%\400)*Freq%)/(Count%+Freq%)
      Avey=(Avey*Count%+(Uncoil% Mod 400)*Freq%)/(Count%+Freq%)
      Add Count%,Freq%
    Endif
    Inc Sixbit%
  Until Sixbit%>4
  Inc N%
  Repeat
    Coil%=(N%-1)*5
    Freq%=Avearray%(N%)
    Uncoil%=Coil%+Sixbit%
    If Uncoil% Mod 100=0 Then
      Print At(11,6);Uncoil%
    Endif
    If Freq%<>0 Then
      For Sixbit%=0 To 4
        Exp%=64^Sixbit%
        Freq%=Avearray%(N%) And (63*Exp%)
        If Freq%<>0 Then
```

```
        Div Freq%,Exp%
        Avex=(Avex*Count%+(Uncoil%\400)*Freq%)/(Count%+Freq%)
        Avey=(Avey*Count%+(Uncoil% Mod 400)*Freq%)/(Count%+Freq%)
        Add Count%,Freq%
      Endif
    Next Sixbit%
  Endif
  Inc N%
 Until N%>32000
 Sub Avex,200
 Avey=200-Avey
 Swap *P.array,Avearray%()
Return
Procedure Power
 Local I%,N%,Sumofx,Sumofy,Sumofproducts,Sumofsquares
 N%=Power(0,0)
 Sumofx=0
 Sumofy=0
 Sumofproducts=0
 Sumofsquares=0
 For I%=1 To N%
  Add Sumofx,Log(Power(0,I%))
  Add Sumofy,Log(Power(1,I%))
  Add Sumofproducts,(Log(Power(0,I%)))*(Log(Power(1,I%)))
  Add Sumofsquares,(Log(Power(0,I%)))^2
 Next I%
 Slope=(N%*Sumofproducts-Sumofx*Sumofy)/(N%*Sumofsquares-Sumofx^2)
 Intercept=(Sumofsquares*Sumofy-Sumofx*Sumofproducts)/(N*Sumofsquares-Sumofx^2)
Return


' Deposition Frequency Histogram Program
Version=1.5
Revdate$="28 Jun 88"
Print "Deposition Frequency Histographer, version"'Version;","'Revdate$
Print "This program requires maximal memory...do not boot-up with system disk"
Print "This program will collect the frequencies of deposition over the pixel"
Print "field (x,y) for all deposits, either large or small."
Print "The output will be a frequency list (f(x,y)) called Longlist.FH6"
Print "The field will be sliced by a cutoff α; all pixels(x,y) that have a"
Print "P(deposit) greater, lower, or equal to α will be displayed."
Print "The synthesized deposit will then be stored as a standard .SCR file with"
Print "the exception that α (the cutoff), and type of region will be overlaid."
Print "The deposit coordinates are stored in a standard .ARR file"
Print "corresponding to the above .SCR file."
Print "If you have inserted an Array Disk and have ready an Empty and Formatted"
Print "disk and are ready to process .ARR files then....type 'Y'"
Print "When the new screen appears then ...type or select '1'"
Print "Come back when you hear the tones........"
Repeat
 Answer$=Inkey$
Until Answer$="Y" Or Answer$="y"
```

```
Dim Order%(30000)
Dim Longlist%(32000)
D=0
Grand%=0
Segment%=0
Arrayfill Longlist(),0
Cls
Do
  Cls
  Showm
  Print "Choose Mode of Operation: Type number or click on selection."
  Print
  Print "1  Automatic processing of all .ARR files on disk"
  Print
  Print "2  Process field array with input of α for upper slice"
  Print
  Print "3  Process field array with input of α for lower slice"
  Print
  Print "4  Process field array for frequency contours"
  Print
  Print "5  Helpful hints and instructions"
  Print
  Print "6  Exit"
  Graphmode 3
  Deffill 1,1
  Ptrvertpos%=Mousey
  If Frac(Ptrvertpos%/32)<0.5 Then
    Gosub Inbox(Ptrvertpos%)
  Else
    In%=0
  Endif
  Do
    Repeat
      Ptrvertpos%=Mousey
      If (In%>0) And (Frac(Ptrvertpos%/32)>0.5) Then
        Gosub Outbox(Ptrvertpos%)
      Endif
      If (In%=0) And (Frac(Ptrvertpos%/32)<0.5) Then
        Gosub Inbox(Ptrvertpos%)
      Endif
      Switch%=Mousek
      If Switch%>0 Then
        If In%>0 Then
          Switch%=(Ptrvertpos%\32)
        Else
          Switch%=0
          Sound 1,15,6,7,5
          Sound 1,0
        Endif
      Endif
      Key$=Inkey$
    Until Key$<>"" Or Switch%
```

```
    If Switch% Then
      Key$=Str$(Switch%)
    Endif
    Exit If Val(Key$)>0 And Val(Key$)<7
    Sound 1,15,6,7,5
    Sound 1,0
  Loop
  Cls
  Graphmode 1
  On Val(Key$) Gosub Array,Upper,Lower,Contour,Help,Exit
  In%=0
  Switch%=0
Loop
End
Procedure Inbox(Ht%)
  Ht%=32*(Ht%\32)
  If Ht%>16 And Ht%<224 Then
    Pbox -1,Ht%,500,Ht%+16
    In%=Ptrvertpos%\32
  Endif
Return
Procedure Outbox(Ht%)
  Ht%=32*In%
  Pbox -1,Ht%,500,Ht%+16
  In%=0
Return
Procedure Exit
  Edit
Return
Procedure Help
  Cls
  Print "This program has two stages; the first, the .ARR file processor"
  Print "requires a disk with a series of .ARR files.  If there are no .ARR"
  Print "files on the disk an error (two bombs) will result."
  Print "The screen during this processing is overlaid with deposits however,"
  Print "the screen density is not representative of the frequency at (x,y)."
  Print "The second stage slices the cumulative histogram at the value of α"
  Print "which is input at the prompt.  .SCR, .FHG, and .ARR files are then"
  Print "set-up after the input whether the higher or lower slices are chosen."
  Print "After viewing, these files named <Freqhist>. can be further processed"
  Print "by existing methods"
  Print "If you have inserted an Array Disk and have ready an Empty and Formatted Disk"
  Print "and are ready to process .ARR files then....type 'Y'"
  Print "When the new screen appears then ...type or select '1'"
  Print "Come back when you hear the tones........"
  Print "If you want to further process a Longlist...........then type 'Y'"
  Print "When the new screen appears then type or select '1' or '2'"
  Print "and follow the prompts...."
  Repeat
    Answer$=Inkey$
  Until Answer$="Y" Or Answer$="y"
Return
```

```
Procedure Upper
  Local Sixbit%,Exp%,Freq%,Fmin%,Coil%,Uncoil%,Iter%
  On Error Gosub Seg_array
  Gosub Checkandload
  Tmax%=Longlist%(16041) And 63 ! The center pixel (200,200) always on.
  Gosub Get_freq
  File$="UPPER"+Str$(Int(Fmin%/Tmax%*100))+".SCR"
  Cls
  Print At(52,1);"Upper slice of"
  Print At(52,2);"frequency histogram"
  Print At(52,4);"Pixels displayed"
  Print At(52,5);"represent sites with"
  Print At(52,6);"frequency α >="'Fmin%/Tmax%
  Print At(52,7);"based on"'Tmax%'"deposits"
  Print At(52,20);"  0% of screen painted"
  Order%(0)=1
  Order%(1)=201
  For Iter%=1 To 32000
    Coil%=(Iter%-1)*5
    Freq%=Longlist%(Iter%)
    If Iter% Mod 320=0 Then
      Print At(52,20);Using "###%",Iter%*100/32000
    Endif
    If Freq%<>0 Then
      For Sixbit%=0 To 4
        Uncoil%=Coil%+Sixbit%
        Exp%=64^Sixbit%
        Freq%=Longlist%(Iter%) And (63*Exp%)
        If Freq%<>0 Then
          Div Freq%,Exp%
          If Freq%>=Fmin% Then
            Xpixel%=Uncoil%\400
            Ypixel%=Uncoil% Mod 400
            Plot Xpixel%,Ypixel%
            Inc Order%(0)
            Option "U1"
            Order%(Order%(0))=Xpixel%*1024+Ypixel%
          Endif
        Endif
      Next Sixbit%
    Endif
  Next Iter%
  Hidem
  Print At(52,20);Space$(28)
  Bsave File$,Xbios(2),32000
  Print At(52,20);"Save filenames:"
  Print At(52,21);File$
  Print At(52,22);"and"
  File$=Left$(File$,Instr(File$,"."))+"ARR"
  Print At(52,23);File$
  Gosub Save_segs
Return
```

```
Procedure Lower
  Local Sixbit%,Exp%,Freq%,Fmin%,Coil%,Uncoil%,Iter%
  On Error Gosub Seg_array
  Gosub Checkandload
  Tmax%=Longlist%(16041) And 63 ! The center pixel (200,200) always on.
  Gosub Get_freq
  File$="LOWER"+Str$(Int(Fmin%/Tmax%*100))+".SCR"
  Cls
  Print At(52,1);"Lower slice of"
  Print At(52,2);"frequency histogram"
  Print At(52,4);"Pixels displayed"
  Print At(52,5);"represent sites with"
  Print At(52,6);"frequency α <="'Fmin%/Tmax%
  Print At(52,7);"based on"'Tmax%'"deposits"
  Print At(52,20);"  0% of screen painted"
  Order%(0)=1
  Order%(1)=201
  For Iter%=1 To 32000
    Coil%=(Iter%-1)*5
    Freq%=Longlist%(Iter%)
    If Iter% Mod 320=0 Then
      Print At(52,20);Using "###%",Iter%*100/32000
    Endif
    If Freq%<>0 Then
      For Sixbit%=0 To 4
        Uncoil%=Coil%+Sixbit%
        Exp%=64^Sixbit%
        Freq%=Longlist%(Iter%) And (63*Exp%)
        If Freq%<>0 Then
          Div Freq%,Exp%
          If Freq%<=Fmin% And Freq%>0 Then
            Xpixel%=Uncoil%\400
            Ypixel%=Uncoil% Mod 400
            Plot Xpixel%,Ypixel%
            Inc Order%(0)
            Option "U1"
            Order%(Order%(0))=Xpixel%*1024+Ypixel%
          Endif
        Endif
      Next Sixbit%
    Endif
  Next Iter%
  Hidem
  Print At(52,20);Space$(28)
  Bsave File$,Xbios(2),32000
  Print At(52,20);"Save filenames:"
  Print At(52,21);File$
  Print At(52,22);"and"
  File$=Left$(File$,Instr(File$,"."))+"ARR"
  Print At(52,23);File$
  Gosub Save_segs
Return
```

```
Procedure Contour
  Local Sixbit%,Exp%,Freq%,Fmin%,Coil%,Uncoil%,Iter%
  On Error Gosub Seg_array
  Gosub Checkandload
  Tmax%=Longlist%(16041) And 63 ! The center pixel (200,200) always on.
  Gosub Get_freq
  File$="CNTUR"+Str$(Int(Fmin%/Tmax%*100))+".SCR"
  Cls
  Print At(52,1);"Contour slice of"
  Print At(52,2);"frequency histogram"
  Print At(52,4);"Pixels displayed"
  Print At(52,5);"represent sites with"
  Print At(52,6);"frequency α ="'Fmin%/Tmax%
  Print At(52,7);"based on"'Tmax%'"deposits"
  Print At(52,20);"  0% of screen painted"
  Order%(0)=1
  Order%(1)=201
  For Iter%=1 To 32000
    Coil%=(Iter%-1)*5
    Freq%=Longlist%(Iter%)
    If Iter% Mod 320=0 Then
      Print At(52,20);Using "###%",Iter%*100/32000
    Endif
    If Freq%<>0 Then
      For Sixbit%=0 To 4
        Uncoil%=Coil%+Sixbit%
        Exp%=64^Sixbit%
        Freq%=Longlist%(Iter%) And (63*Exp%)
        If Freq%<>0 Then
          Div Freq%,Exp%
          If Freq%=Fmin% Then
            Xpixel%=Uncoil%\400
            Ypixel%=Uncoil% Mod 400
            Plot Xpixel%,Ypixel%
            Inc Order%(0)
            Option "U1"
            Order%(Order%(0))=Xpixel%*1024+Ypixel%
          Endif
        Endif
      Next Sixbit%
    Endif
  Next Iter%
  Hidem
  Print At(52,20);Space$(28)
  Bsave File$,Xbios(2),32000
  Print At(52,20);"Save filenames:"
  Print At(52,21);File$
  Print At(52,22);"and"
  File$=Left$(File$,Instr(File$,"."))+"ARR"
  Print At(52,23);File$
  Gosub Save_segs
Return
```

```
Procedure Checkandload
  Local Fail!,File$,Devlist$,Devcnt%
  Devlist$="ADB"
  If Longlist%(0)=0 Then
    Do
      Devcnt%=1
      Repeat
        File$=Mid$(Devlist$,Devcnt%,1)+":\LONGLIST.FHG"
        Print "Checking device"'Left$(File$,2)'"for LONGLIST.FHG"
        Exit If Exist(File$)
        Inc Devcnt%
      Until Devcnt%>Len(Devlist$)
      Exit If Devcnt%<=Len(Devlist$)
      Print "Can't find any longlist files. Please load a disk with a"
      Print "longlist at top level and hit any key. <ESC> aborts"
      Print "the program."
      Repeat
        File$=Inkey$
      Until File$<>""
      If File$=Chr$(27) Then
        Edit
      Endif
    Loop
    Print "Loading"'File$
    Bload File$,Lpeek(Arrptr(Longlist%()))
    Arrayfill Order%(),0
  Endif
Return
Procedure Get_freq
  Local Fmin$
  Print "Cutoff frequencies must be integer multiples of 1/";Tmax%;"."
  Print "Frequency will automatically be rounded to nearest 1/";Tmax%;"th."
  Do
    Input "Cutoff frequency (absolute n, or α%)";Fmin$
    If Instr(Fmin$,"%")<>0 Then
      Fmin%=Tmax%*0.01*Val(Fmin$)+0.5
    Else
      Fmin%=Val(Fmin$)+0.5
    Endif
    Exit If Fmin%<=Tmax%
    Print "Frequency can't exceed 100% or"'Tmax%'"deposits. Please reenter."
  Loop
Return
Procedure Seg_array
  Local Ecode%,Seg%,Seg$
  Ecode%=Err
  On Error Gosub Seg_array
  If Ecode%<>16 Then
    On Error
    Error Ecode%
  Endif
  Seg%=(Segment%\29999)+1
```

```
  Print At(53,21);"Segmenting .ARR file"
  Print At(53,22);"Segment"'Seg%
  Print At(53,23);"Please wait..."
  If Seg%>1 Then
    Seg$=Left$(File$,Instr(File$,"."))+"AR"+Str$(Seg%)
  Else
    Seg$=Left$(File$,Instr(File$,"."))+"ARR"
  Endif
  Dec Order%(0)
  Bsave Seg$,Lpeek(Arrptr(Order%())),Order%(0)*4+8
  Arrayfill Order%(),0
  Order%(0)=2
  Order%(1)=201
  Add Segment%,29999
  Print At(52,21);Spc(29)
  Print At(52,22);Spc(29)
  Print At(52,23);Spc(29)
  Resume
Return
Procedure Save_segs
  Local Base$
  Base$=Left$(File$,Instr(File$,"."))
  If Segment%=0 Then
    Bsave File$,Lpeek(Arrptr(Order%())),Order%(0)*4+8
  Else
    File$=Base$+"AR"+Str$(Int(Segment%/29999)+1)
    Bsave File$,Lpeek(Arrptr(Order%())),Order%(0)*4+8
    Open "R",#1,Base$+"ARR",4
    Field #1,4 As Buf$
    Lset Buf$=Mkl$(Segment%+Order%(0))
    Put #1,2
    Close #1
  Endif
Return
Procedure Array
  Print At(52,1);"Deposit Grand Total=     0"
  Print At(52,3);"File:"
  Print At(52,5);"File number=";'0
  Print At(52,7);"N=    0"
  Print At(52,9);"Out of     0 total deposits"
  Repeat
    Dir "*.ARR" To "FREQHIST.DIR"
    Open "I",#0,"FREQHIST.DIR"
    Repeat
      Gosub Loader
      Gosub Process
    Until Eof(#0)
    Repeat
      Print At(52,22);"Hit any key to continue"
      P=Trunc(125/Rnd(1)+0.5)
      Sound 1,15,#P,50
    Until Inkey$<>""
```

```
    Sound 1,0
    Print At(52,10);"If all Array Disks are done"
    Print At(52,11);"Remove the last Array Disk"
    Print At(52,13);"If all are done... Type 'D'"
    Print At(52,15);"If more Disks are to be done"
    Print At(52,16);"insert the next Array Disk"
    Print At(52,17);"into the disk drive"
    Print At(52,19);"If more to do..... Type 'M'"
    Repeat
      Repeat
        Answer$=Inkey$
      Until Answer$<>""
      Answer$=Chr$(Asc(Answer$) And 95)
      If Answer$="D" Then
        Gosub Blank
        Gosub Escape
      Endif
      If Answer$="M" Then
        Gosub Blank
        Close
      Endif
    Until Answer$="S" Or Answer$="M"
  Until Answer$="S"
Return
Procedure Escape
  Print At(52,10);"Insert a Formatted and Empty"
  Print At(52,11);"Disk into the disk drive"
  Print At(52,13);"If the drive is ready"
  Print At(52,14);"then Longlist will be saved"
  Print At(52,16);"To save.........Type 'S'"
  Repeat
    Answer$=Inkey$
    Answer$=Chr$(Asc(Answer$) And 95)
  Until Answer$="S"
  Bsave "LONGLIST.FHG",Lpeek(Arrptr(Longlist%())),128008
  Close
Return
Procedure Process
  Inc D
  Print At(58,3);Space$(22)
  Print At(58,3);File$
  Print At(65,5);D
  Print At(52,9);Using "Out of ##### total deposits",Order%(0)-1
  N%=2
  Repeat
    Xpixel%=Order%(N%)\1024
    Ypixel%=Order%(N%) And 1023
    Plot Xpixel%,Ypixel%
    Coil%=400*Xpixel%+Ypixel%
    Disp%=Coil% Mod 5
    Coil%=Coil%\5
    Add Longlist%(Coil%+1),64^Disp%
```

```
    Inc Grand%
    If (N%-1) Mod 100=0 Then
      Print At(72,1);Using "######",Grand%
      Print At(54,7);Using "#####",N%-1
    Endif
    Inc N%
  Until N%>Order%(0)
  Print At(52,1);Using "Deposit Grand Total=######",Grand%
  Longlist%(0)=Grand%
Return
Procedure Loader
  Arrayfill Order%(),0
  Input #0,File$
  Bload File$,Lpeek(Arrptr(Order%()))
Return
Procedure Blank
  Deffill 0,1
  Pbox 401,124,639,399
Return
```

APPENDIX B

NUMERICAL DATA

TABLE II

FRACTAL DIMENSION DATA FOR INDIVIDUAL SMALL AGGREGATES

			Including Edge		Excluding Edge		
			Squares	'Circles'	Squares	'Circles'	Circles
Deposit	N	D_{Rg}	D_c	D_c	D_c	D_c	D_c
1	4767	1.8355364401	1.6573104232	1.6020511402	1.6880587015	1.6303281027	1.6870086816
2	3825	1.9224613245	1.6609789001	1.6046445531	1.6810192676	1.623102596	1.6802291833
3	3899	1.8374010038	1.6599607926	1.6062148931	1.6785667926	1.6242967218	1.6817418526
4	4621	1.7910889197	1.6525600963	1.5983715229	1.6831302681	1.6267701358	1.6839210176
5	2972	1.8931199782	1.6661022793	1.6156322755	1.6790711059	1.6272517401	1.6833733409
6	4969	1.8368096251	1.658277456	1.6020197103	1.697172808	1.6389580136	1.6957571247
7	4639	2.0458284656	1.6453806026	1.5943751484	1.6857970137	1.6313605475	1.688136336
8	4354	1.807984199	1.6621301618	1.6069461333	1.6969820377	1.641131271	1.6984915942
9	5335	1.8242820754	1.6716125386	1.6187794088	1.7187509464	1.6628165381	1.7197489563
10	4512	1.7398365792	1.6704012545	1.6162811829	1.6992186088	1.6438876088	1.7012425911
11	3314	1.871951025	1.6601788582	1.6030092667	1.6769611661	1.6188760225	1.6749274763
12	4622	1.7310142902	1.6670967371	1.6151354645	1.6974291693	1.6423349234	1.699262593
13	4529	1.8186270042	1.6852599808	1.6293001878	1.7139555174	1.6575690305	1.7150032462
14	3793	1.9193907128	1.6587513458	1.6041053153	1.6808722818	1.6253296977	1.6819975035
15	5000	1.744598435	1.674623683	1.6185214248	1.7054809105	1.6467311771	1.703906851
16	5042	1.908052491	1.6698398268	1.6143191837	1.7050727751	1.6477016711	1.7047734201
17	3795	1.8193060175	1.6598486635	1.6081433871	1.688375014	1.6344183185	1.6909629697
18	4420	1.8347762916	1.6675452581	1.6127089791	1.6962384505	1.6398140939	1.696631758
19	4411	1.774914963	1.6571104783	1.6045614382	1.6832531101	1.6293954963	1.6864977259
20	5764	1.8319819071	1.6676959306	1.6104902431	1.7177188699	1.6591283631	1.7163278898
21	5518	1.9340069265	1.6632756023	1.6127396026	1.7101350582	1.6554795156	1.7130257265
22	3506	1.8022675617	1.6635810306	1.6119526325	1.6881568945	1.6351435214	1.6918002253
23	5238	1.8128935254	1.6653275148	1.6087703385	1.7004290546	1.6410850401	1.69825338
24	4132	1.8484878424	1.6692567821	1.6186259044	1.6962588726	1.6438149748	1.7007075376
25	5212	1.8311292049	1.6781571799	1.6220593534	1.7136615252	1.6562096633	1.7132077267
26	5080	1.8549047671	1.6544908435	1.6005912453	1.6962772363	1.6391177077	1.6958024112

TABLE III

FRACTAL DIMENSION DATA FOR INDIVIDUAL LARGE AGGREGATES

			Including Edge		Excluding Edge		
			Squares	'Circles'	Squares	'Circles'	Circles
Deposit	N	D_{Rg}	D_c	D_c	D_c	D_c	D_c
1	17428	1.82979542	1.6699438252	1.6126899269	1.6747005497	1.6170335607	1.6738358665
2	14872	1.8571485234	1.6695933746	1.6113824213	1.6750954405	1.6167919779	1.6737356634
3	15411	1.777909686	1.6693841613	1.614141633	1.6711044925	1.6163991096	1.6733642509
4	15243	1.7746517225	1.6682995941	1.6118757841	1.6735022653	1.6164864462	1.6733864314
5	12208	1.8329831474	1.6688731604	1.6120067888	1.6705759311	1.6138091472	1.6705589024
6	18052	1.843776623	1.6635686705	1.6073511179	1.673057665	1.6160872085	1.6727685344
7	19429	1.9543270665	1.6670483288	1.6106346917	1.6786194158	1.6212306642	1.6781842979
8	16525	1.7756269843	1.659262413	1.6031247851	1.668189111	1.6115455375	1.6682588539
9	21268	1.7853591922	1.6672958319	1.6113788372	1.679632124	1.623083186	1.6801694666
10	17189	1.7354014398	1.6635703673	1.6088319099	1.6693108723	1.6144074109	1.6711497312
11	13706	1.8248885544	1.6666992685	1.6092906897	1.6700040655	1.6128300832	1.6696128472
12	14395	1.7357458246	1.6680916443	1.610045142	1.6711784013	1.6131569843	1.6699916866
13	15320	1.78879105	1.6776510535	1.62032031	1.6798353916	1.622090424	1.6789364492
14	12854	1.8613387844	1.6675974528	1.6125763577	1.6715090894	1.6167550434	1.6734976508
15	14897	1.7333504997	1.6649833499	1.6076303434	1.6718644097	1.6139823356	1.6707704828
16	16326	1.8693838524	1.6607155269	1.6052765298	1.6647759132	1.6089647983	1.665931117
17	14944	1.7978242181	1.6689430768	1.6111271869	1.6716337209	1.6136249278	1.6705724942
18	16338	1.8051414443	1.6639643543	1.60817652	1.6667714795	1.6110268009	1.6679065033
19	15752	1.7697621464	1.6720236587	1.6144379524	1.675944359	1.6178854626	1.6750132936
20	17715	1.8098581388	1.6613071914	1.6057271884	1.6662990664	1.6105454317	1.6672416964
21	19255	1.8781901382	1.6670629371	1.6142472474	1.6752969272	1.621885911	1.678824173
22	12613	1.732079339	1.6710463412	1.6159165965	1.6745999834	1.619317032	1.6761888829
23	16464	1.7898308771	1.6746682685	1.6178779816	1.6815031254	1.6243320708	1.6811924391
24	17161	1.7829201003	1.6688924395	1.6152812764	1.6774442037	1.623302901	1.680249824
25	16907	1.8066831114	1.6733568534	1.6161727463	1.6796282729	1.6220518204	1.6788707044
26	19615	1.8434988508	1.661437214	1.6072074545	1.6695702278	1.61454239	1.6713265236
27	19056	1.7922802635	1.6761666367	1.6198860929	1.6857029235	1.6287472478	1.6857074059
28	17401	1.8293298347	1.6572971729	1.6042628967	1.663857593	1.6101775892	1.6666498788
29	15680	1.7937369861	1.6629574201	1.6063976033	1.6672379819	1.6101002107	1.6668820211
30	14909	1.8046050822	1.6536853066	1.5971666179	1.6573053395	1.6004981638	1.6573353164

The following graphs of the radius of gyration dependence on the number of deposits are based on the radius of gyration which was calculated from the lattice origin. The slopes are also listed in Table III.

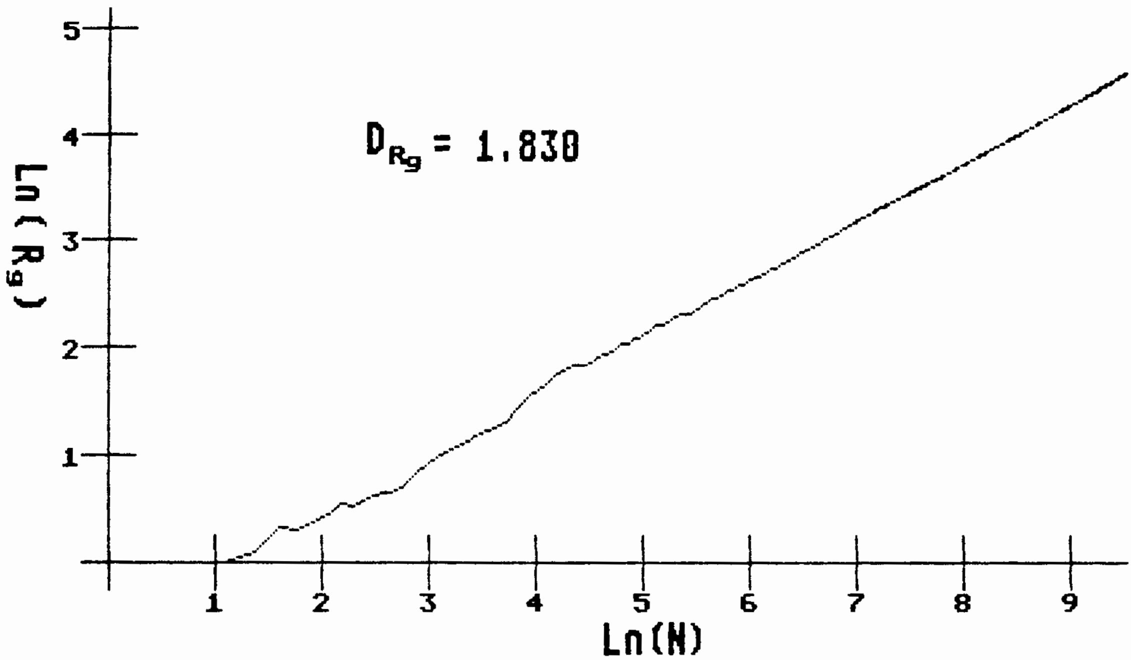


Figure 21. Ln(R_g) vs. Ln(N) for aggregate number 1.

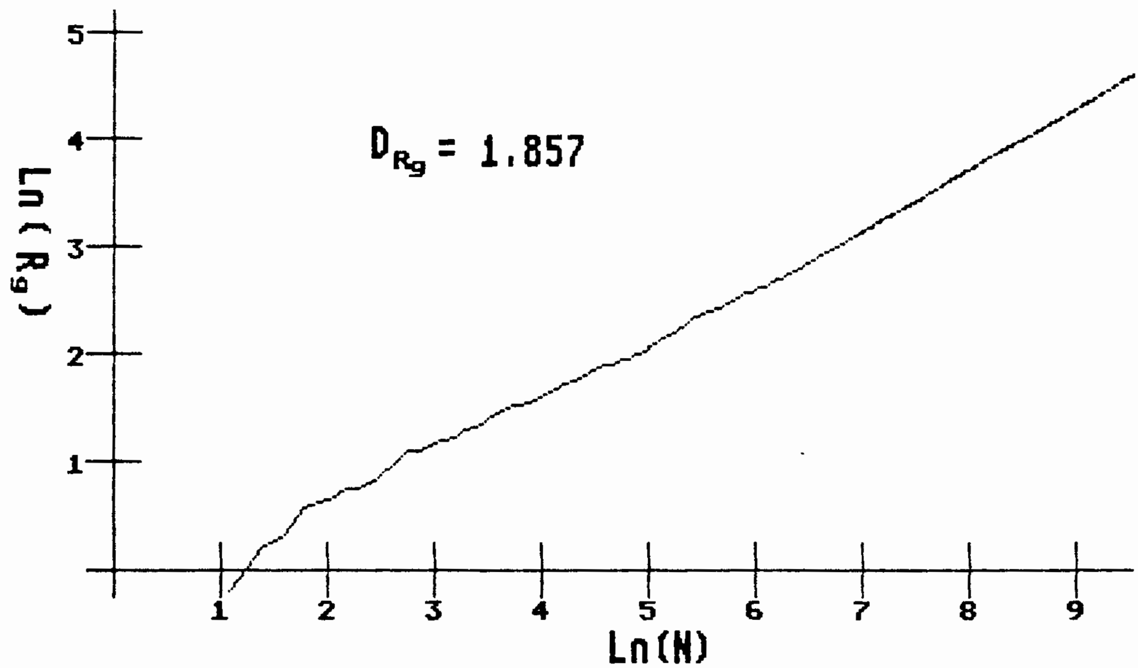


Figure 22. Ln(R_g) vs. Ln(N) for aggregate number 2.

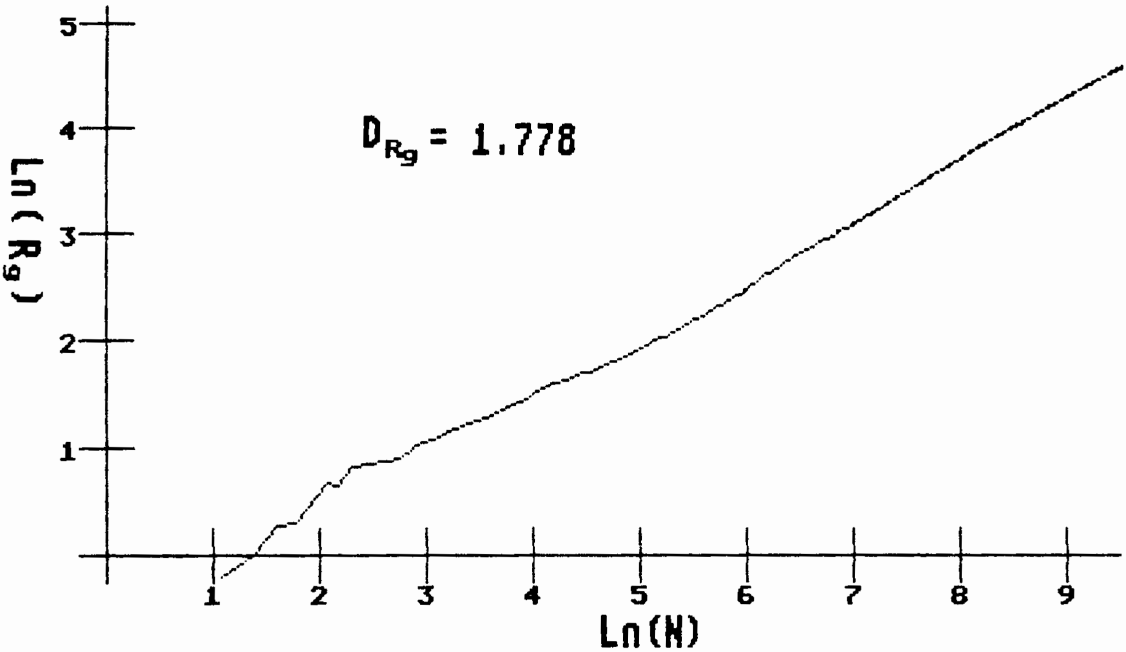


Figure 23. $Ln(R_g)$ vs. Ln(N) for aggregate number 3.

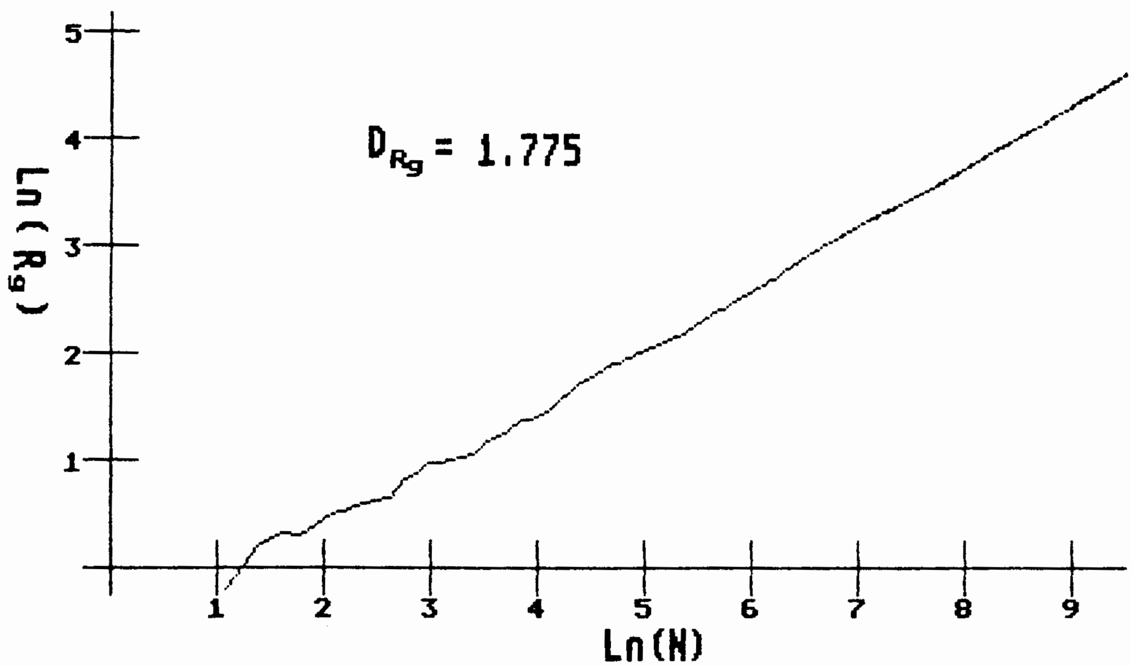


Figure 24. $Ln(R_g)$ vs. Ln(N) for aggregate number 4.

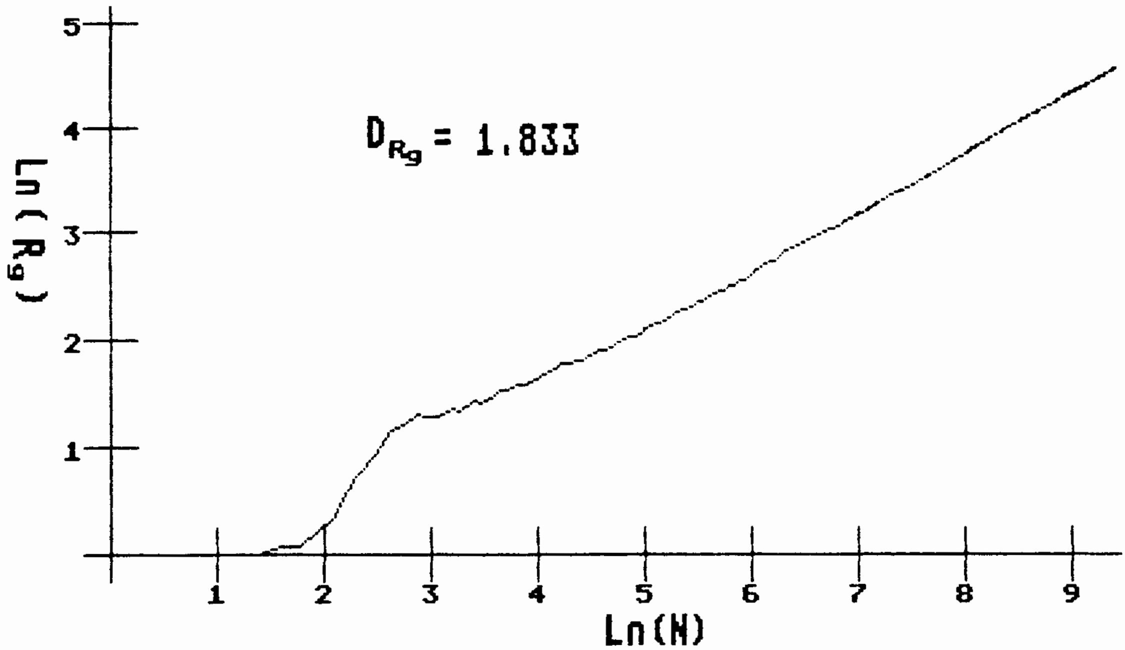


Figure 25. Ln(R_g) vs. Ln(N) for aggregate number 5.

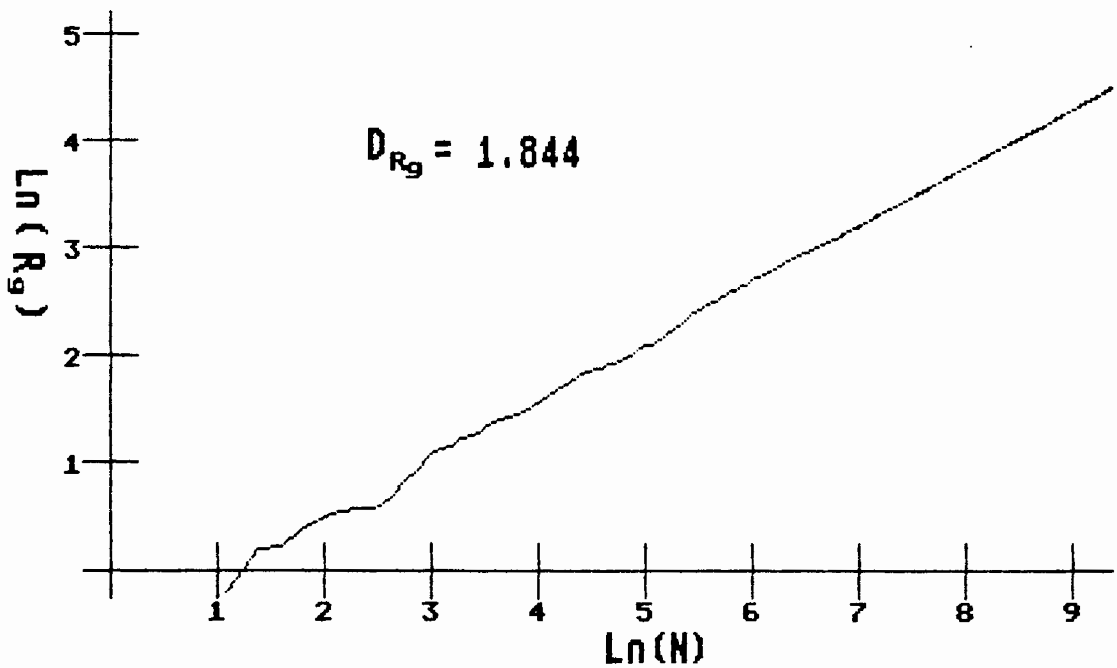


Figure 26. Ln(R_g) vs. Ln(N) for aggregate number 6.

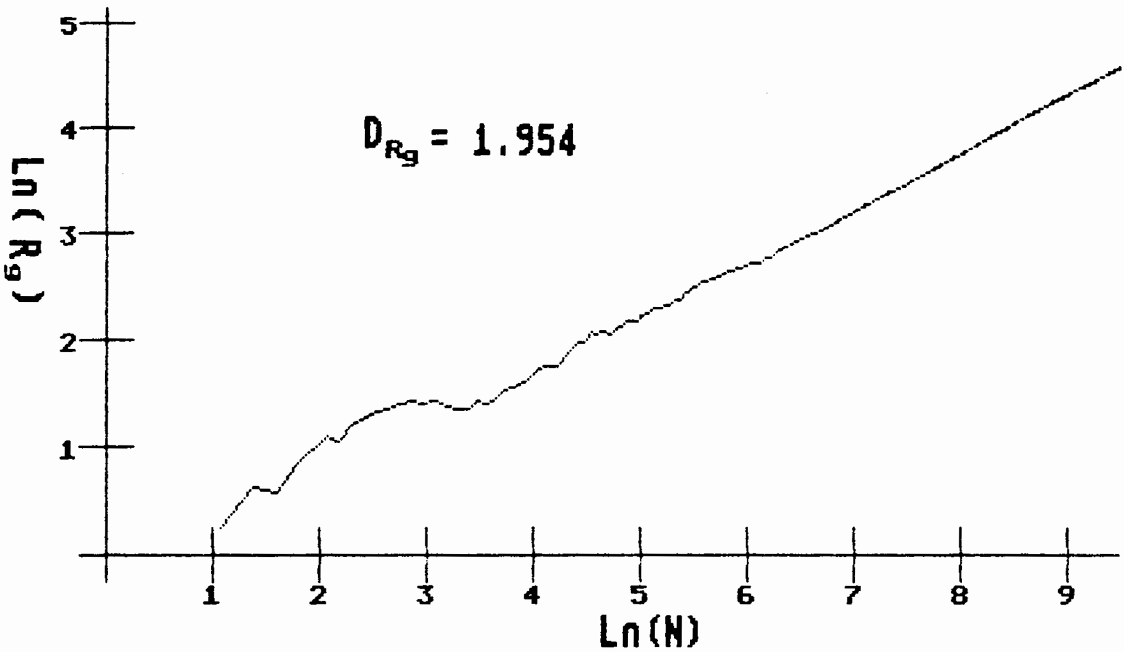


Figure 27. Ln(R_g) vs. Ln(N) for aggregate number 7.

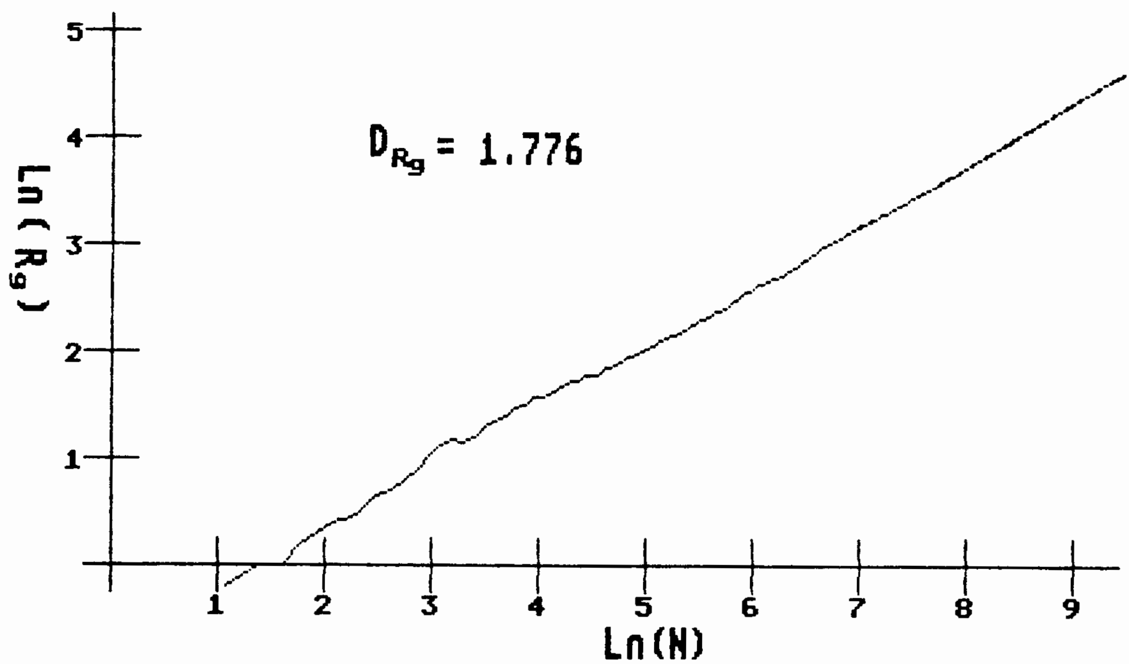


Figure 28. Ln(R_g) vs. Ln(N) for aggregate number 8.

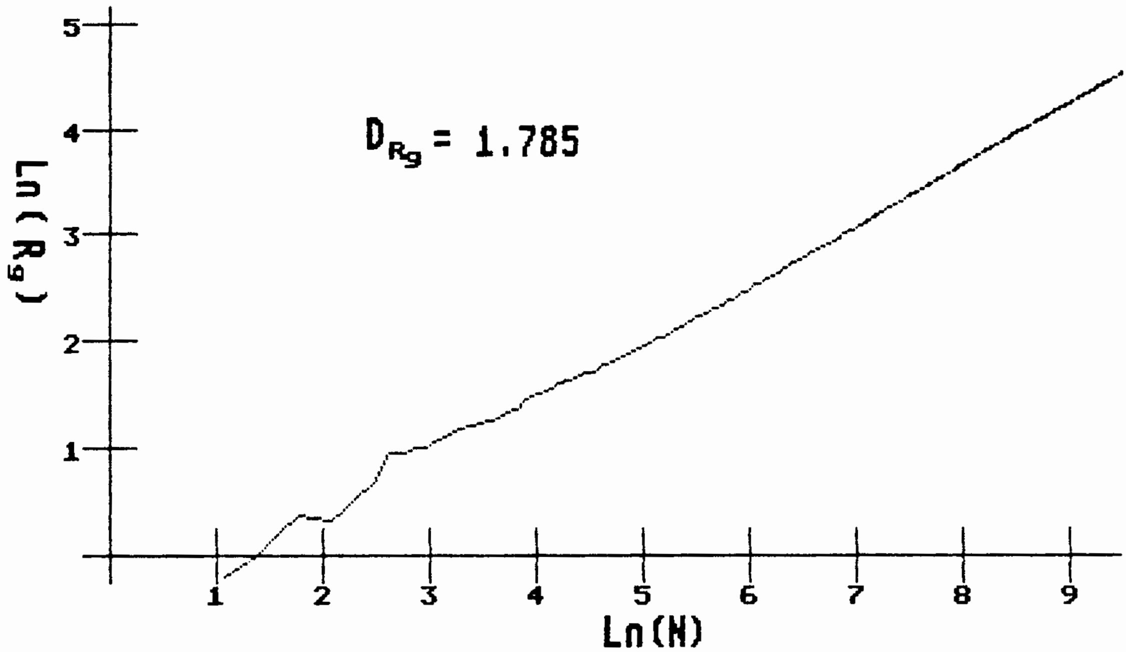


Figure 29. Ln(R_g) vs. Ln(N) for aggregate number 9.

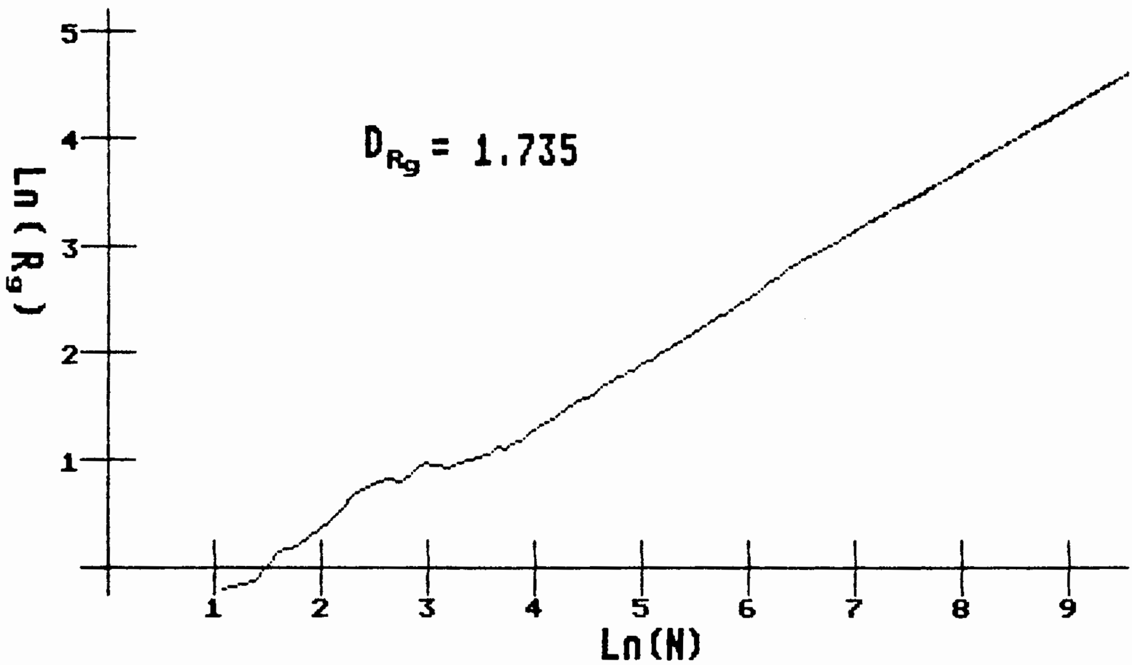


Figure 30. Ln(R_g) vs. Ln(N) for aggregate number 10.

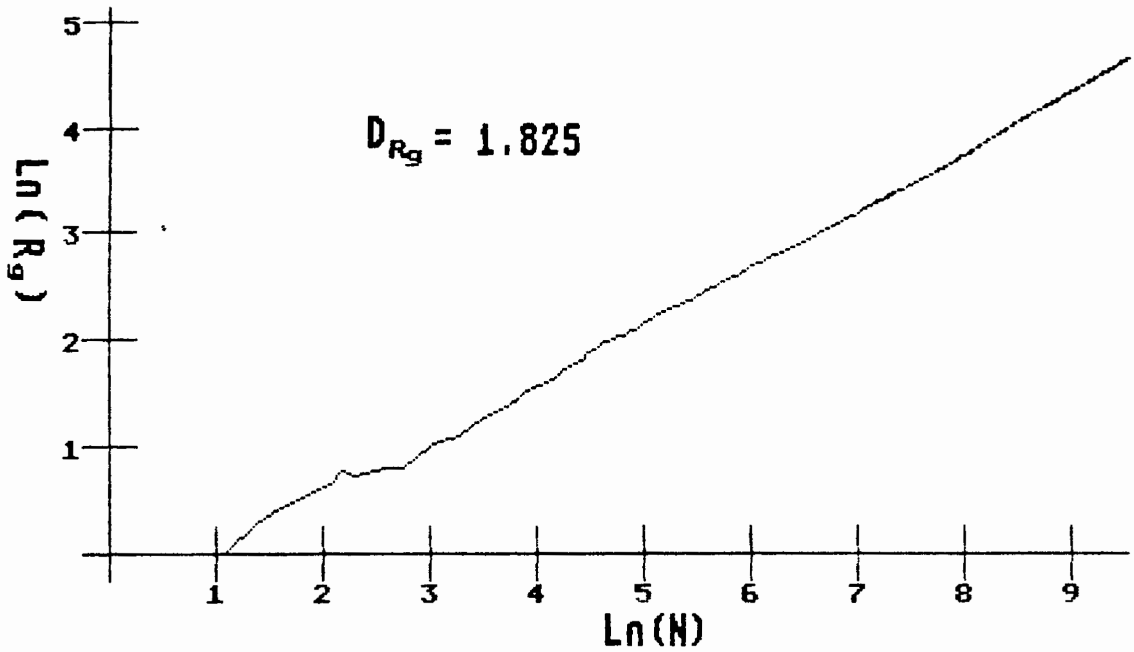


Figure 31. Ln(R_g) vs. Ln(N) for aggregate number 11.

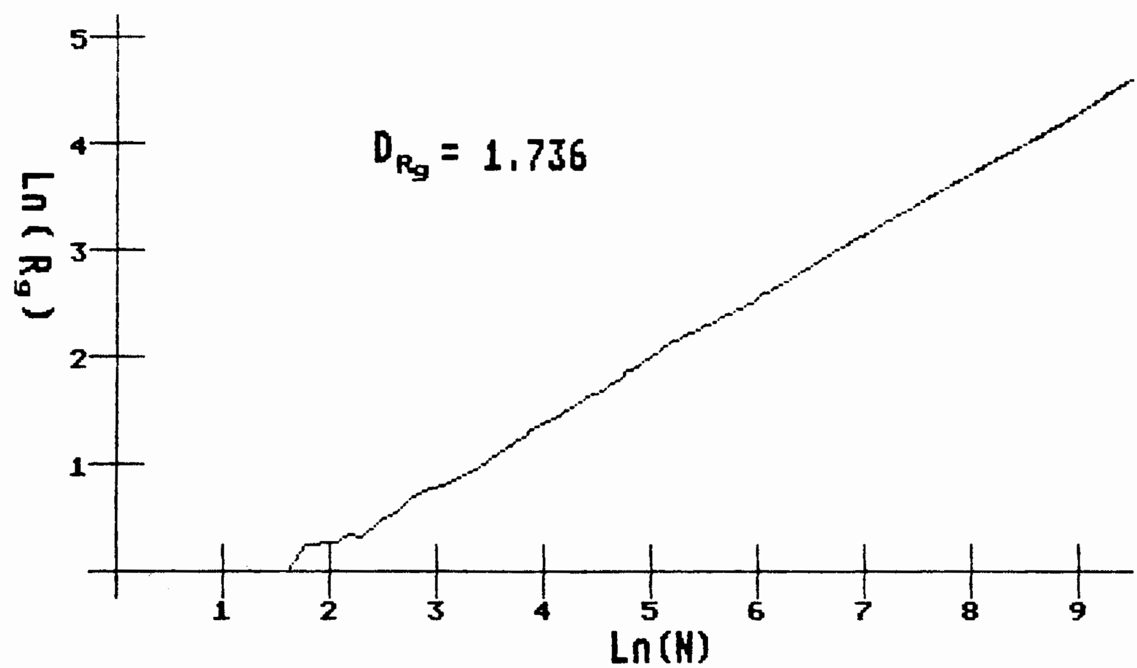


Figure 32. Ln(R_g) vs. Ln(N) for aggregate number 12.

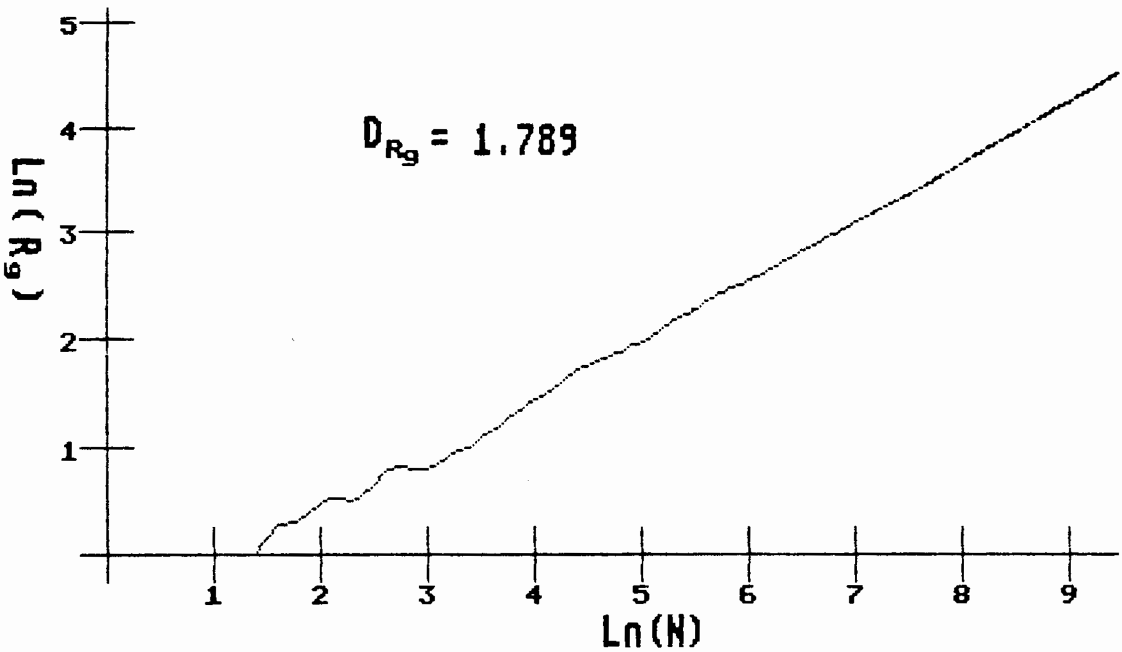


Figure 33. Ln(R_g) vs. Ln(N) for aggregate number 13.

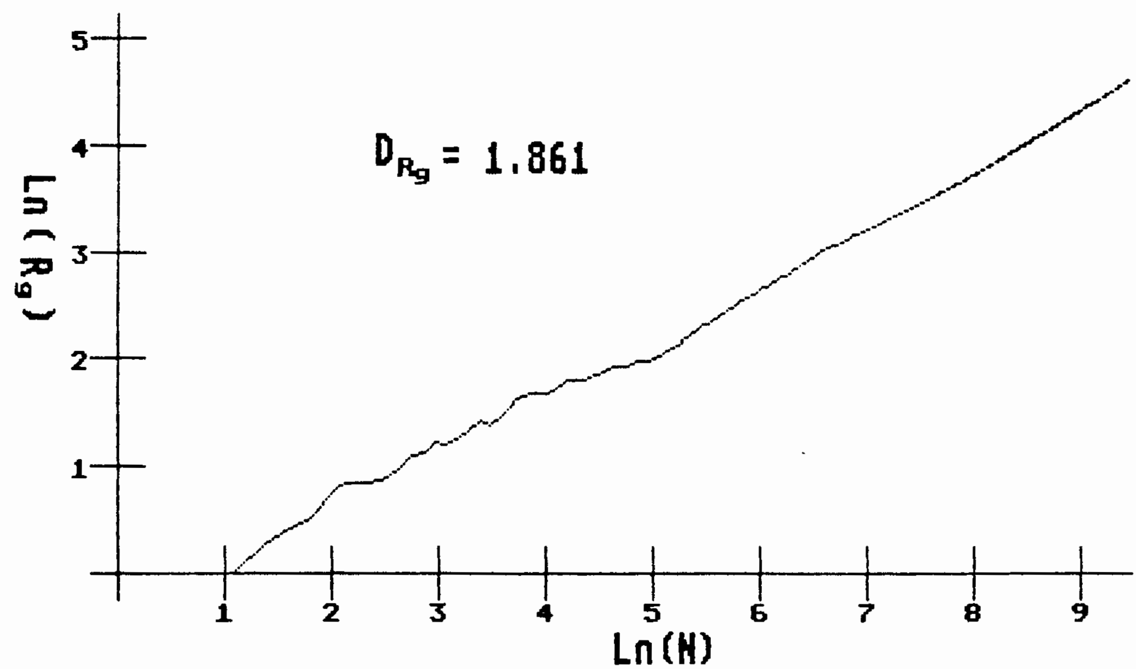


Figure 34. Ln(R_g) vs. Ln(N) for aggregate number 14.

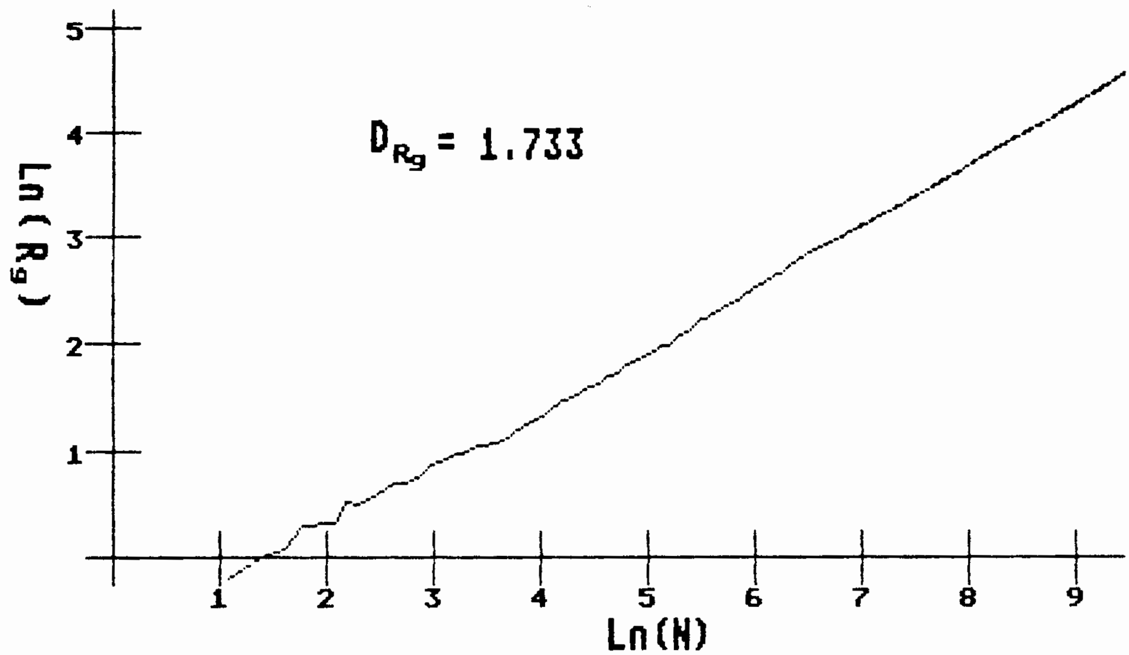


Figure 35. Ln(R_g) vs. Ln(N) for aggregate number 15.

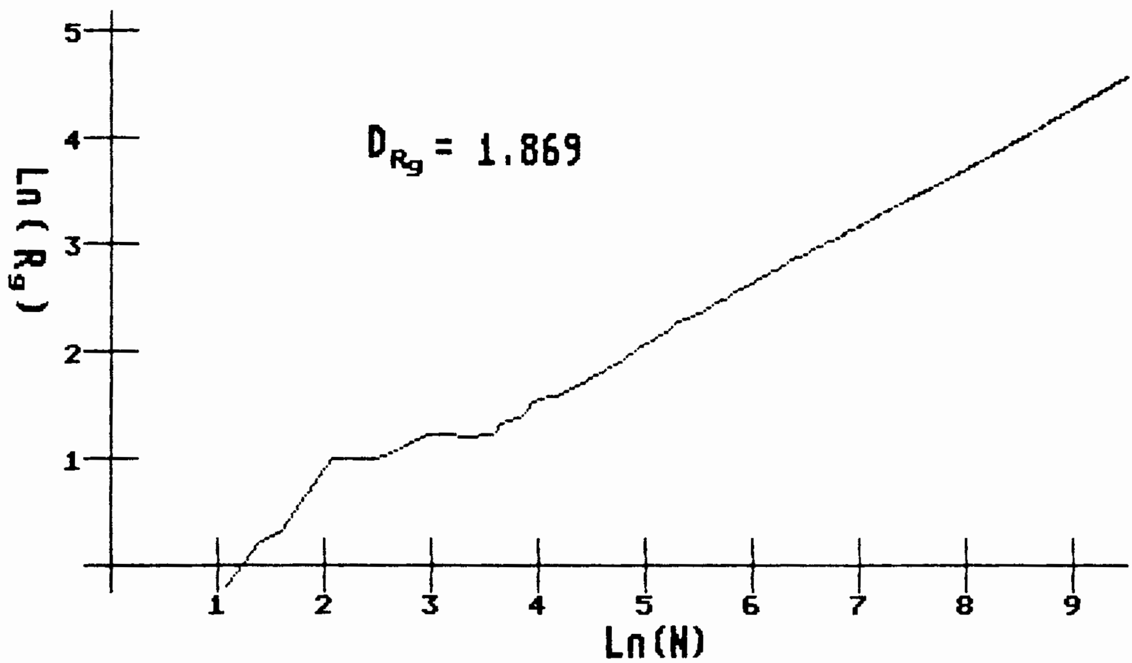


Figure 36. Ln(R_g) vs. Ln(N) for aggregate number 16.

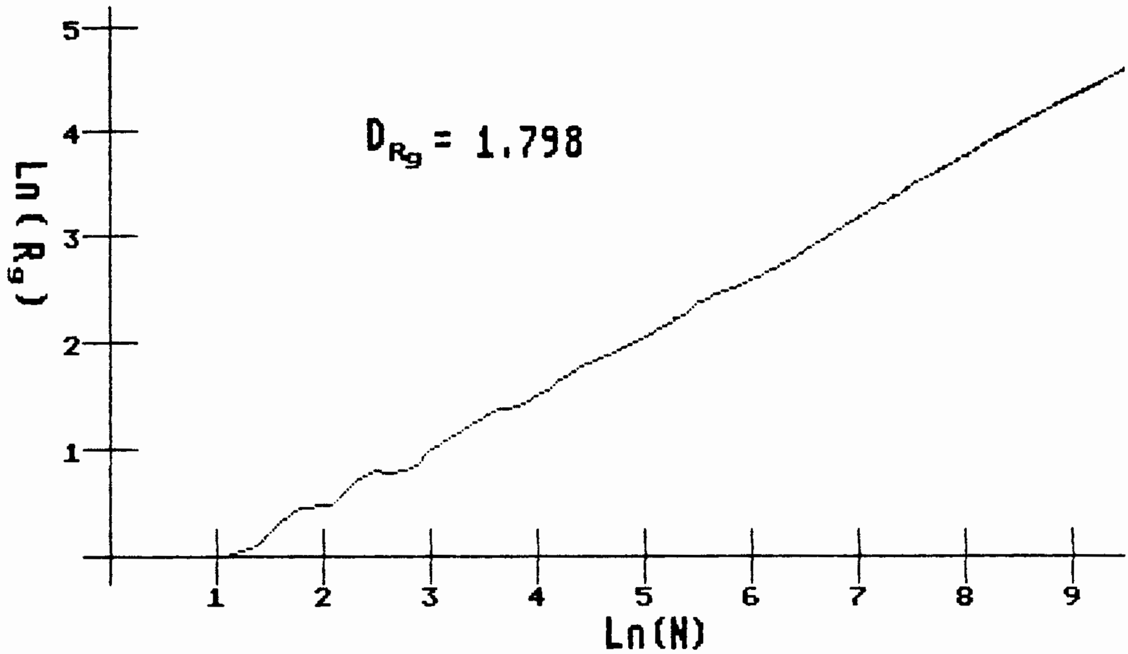


Figure 37. Ln(R_g) vs. Ln(N) for aggregate number 17.

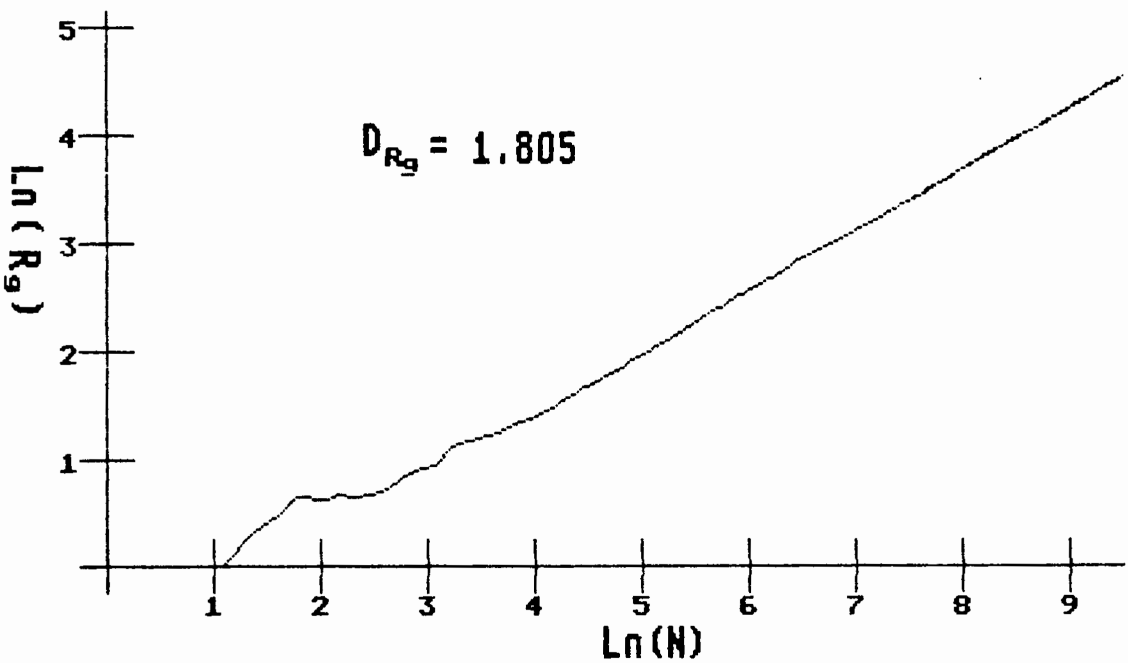


Figure 38. Ln(R_g) vs. Ln(N) for aggregate number 18.

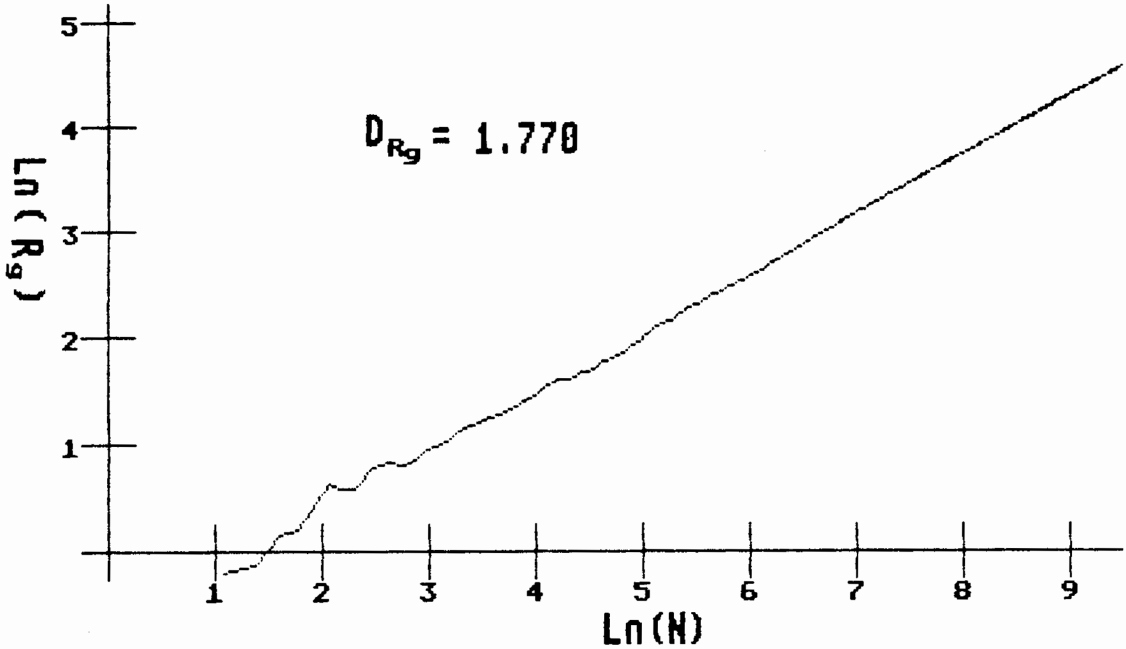


Figure 39. Ln(R_g) vs. Ln(N) for aggregate number 19.

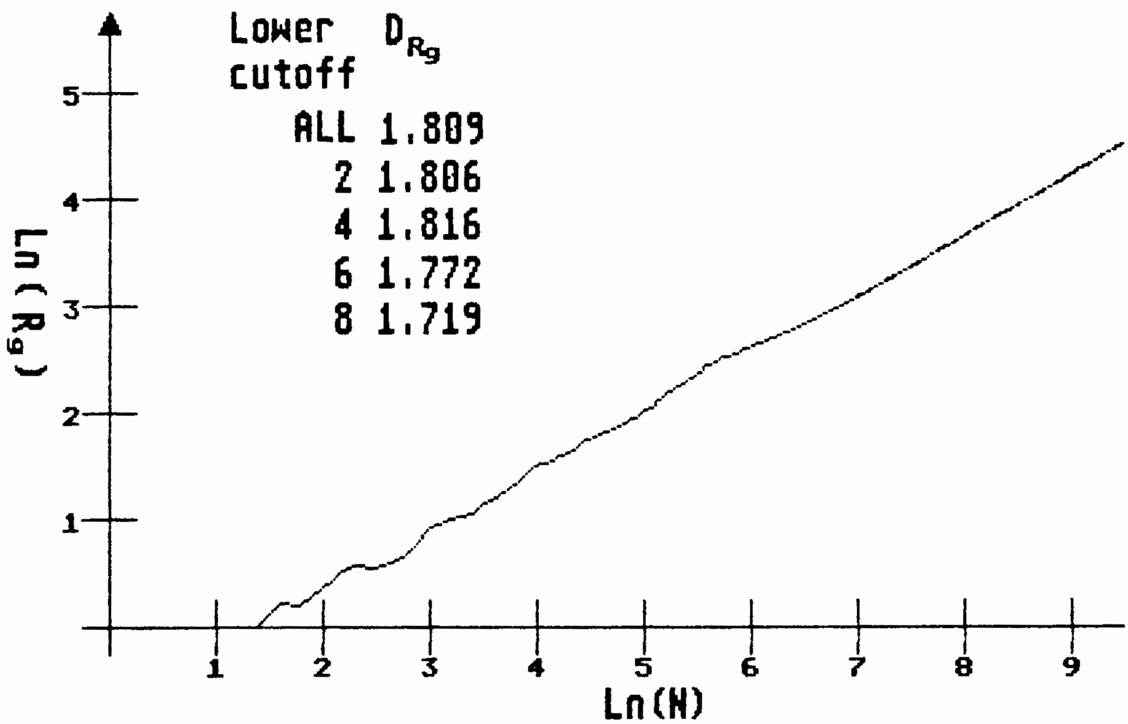


Figure 40. Ln(R_g) vs. Ln(N) for aggregate number 20.

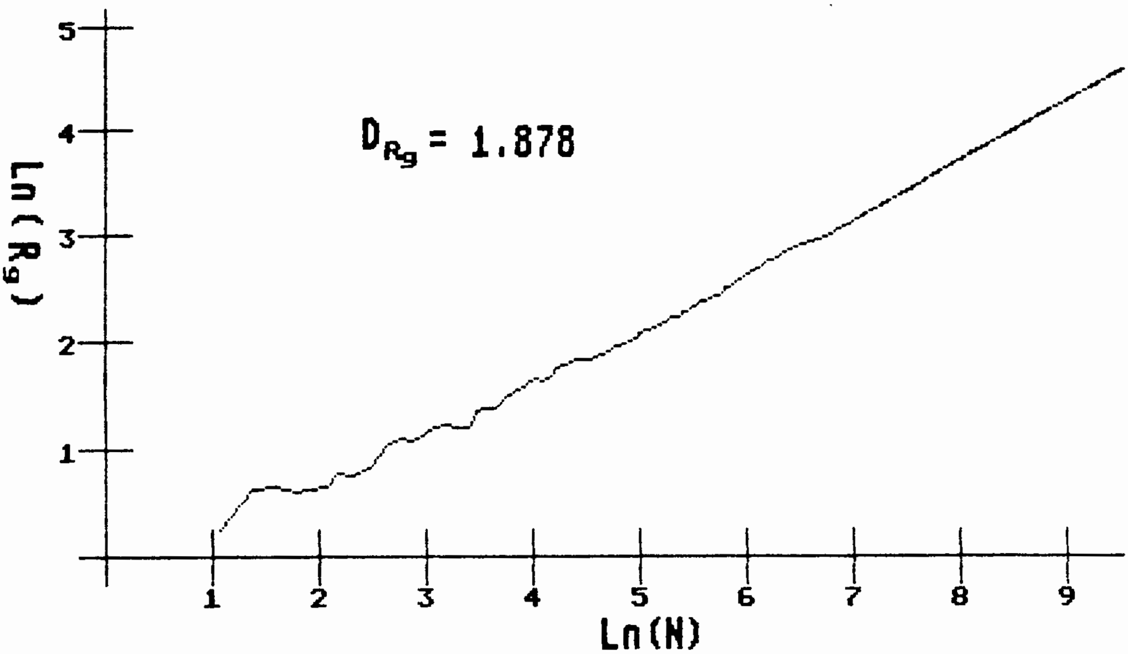


Figure 41. $Ln(R_g)$ vs. Ln(N) for aggregate number 21.

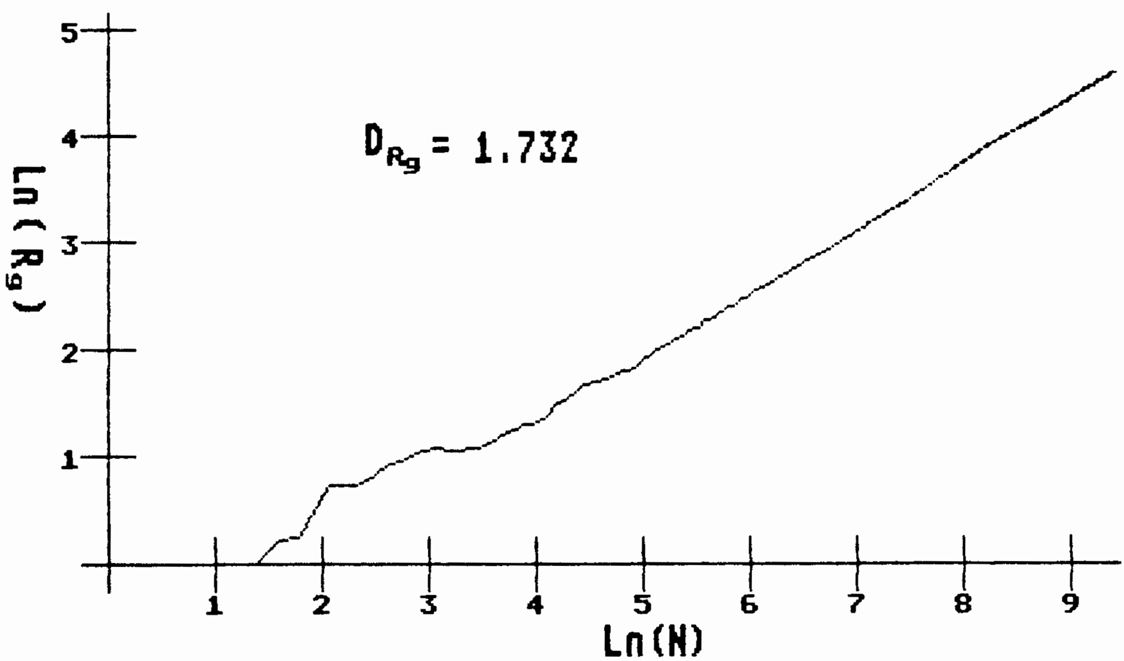


Figure 42. $Ln(R_g)$ vs. Ln(N) for aggregate number 22.

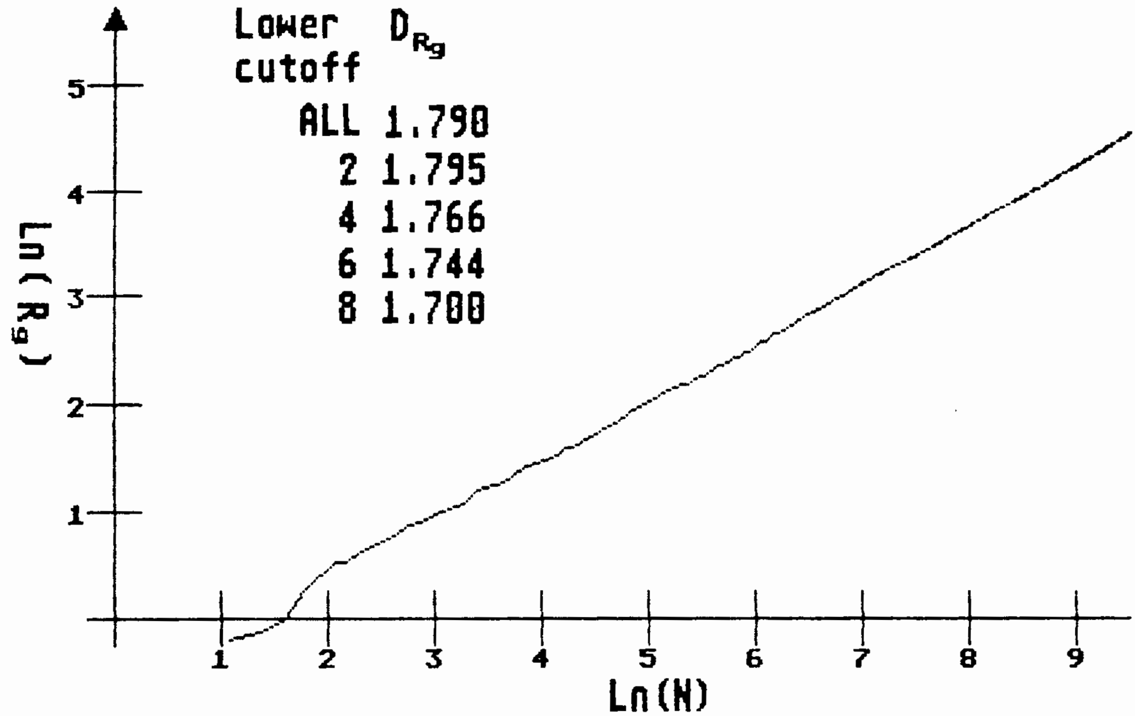


Figure 43. Ln(R_g) vs. Ln(N) for aggregate number 23.

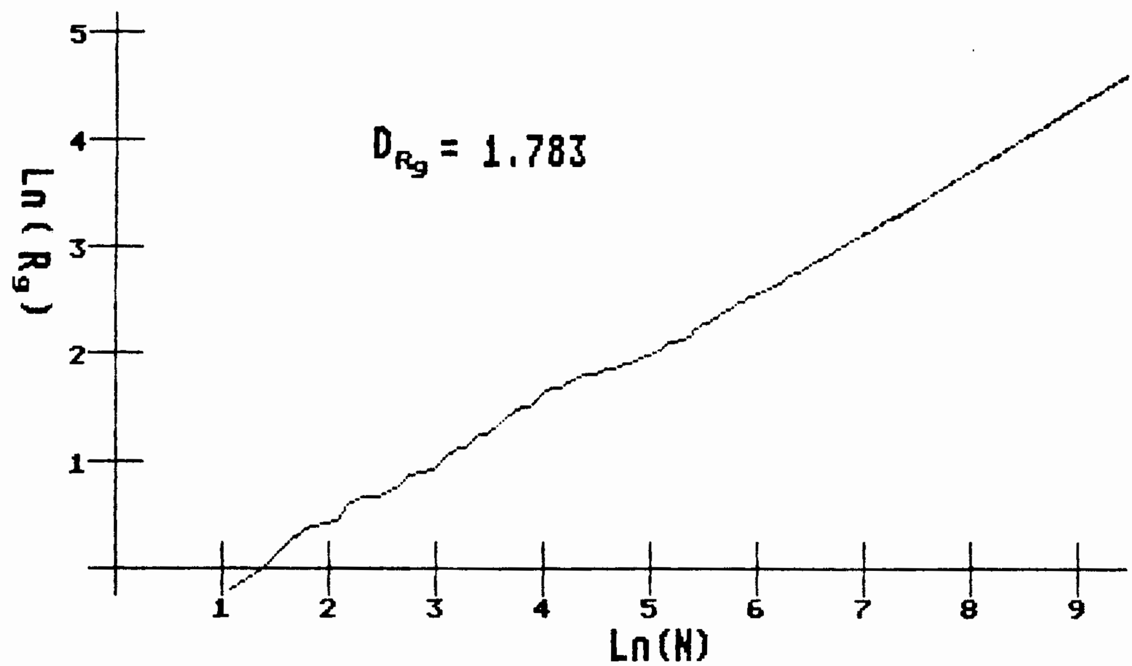


Figure 44. Ln(R_g) vs. Ln(N) for aggregate number 24.

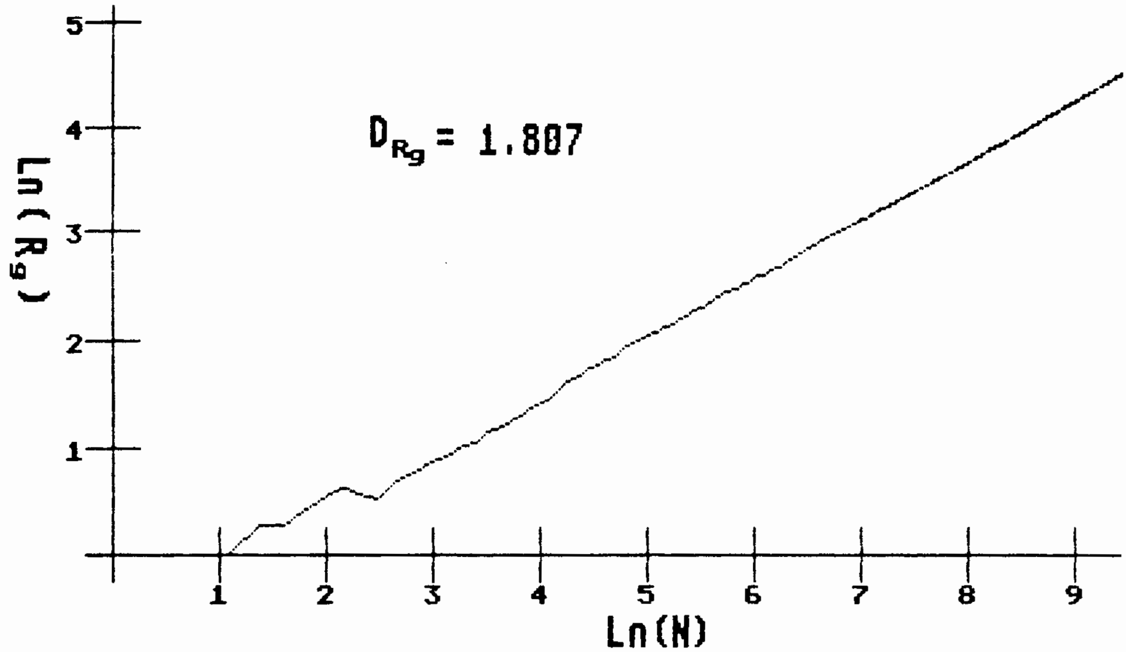


Figure 45. Ln(R_g) vs. Ln(N) for aggregate number 25.

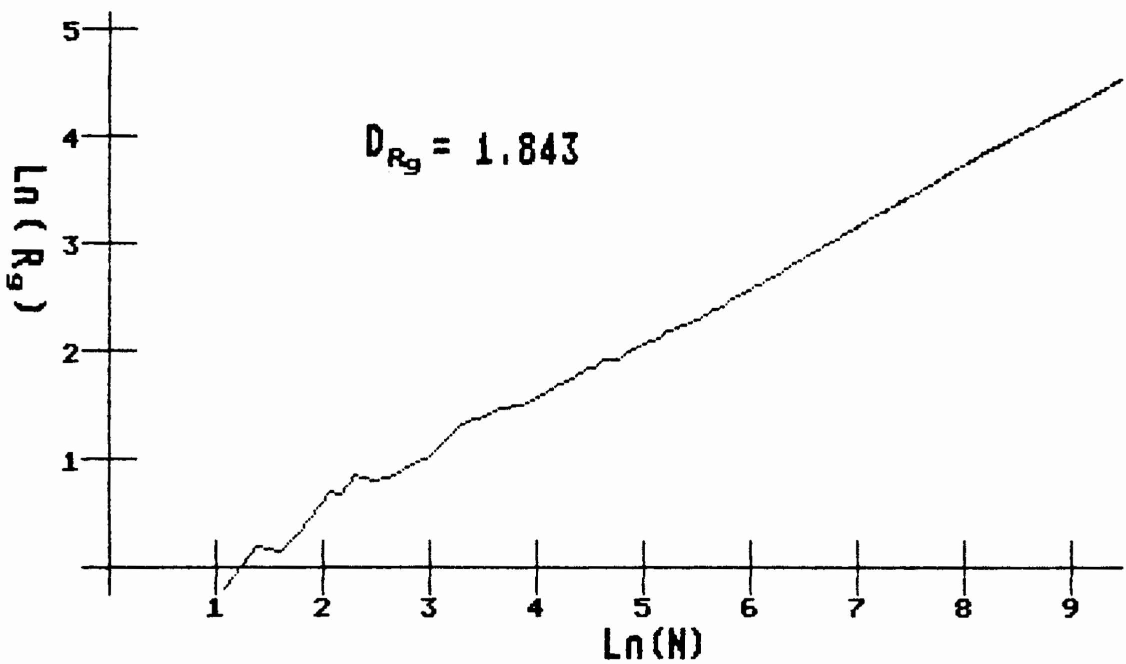


Figure 46. Ln(R_g) vs. Ln(N) for aggregate number 26.

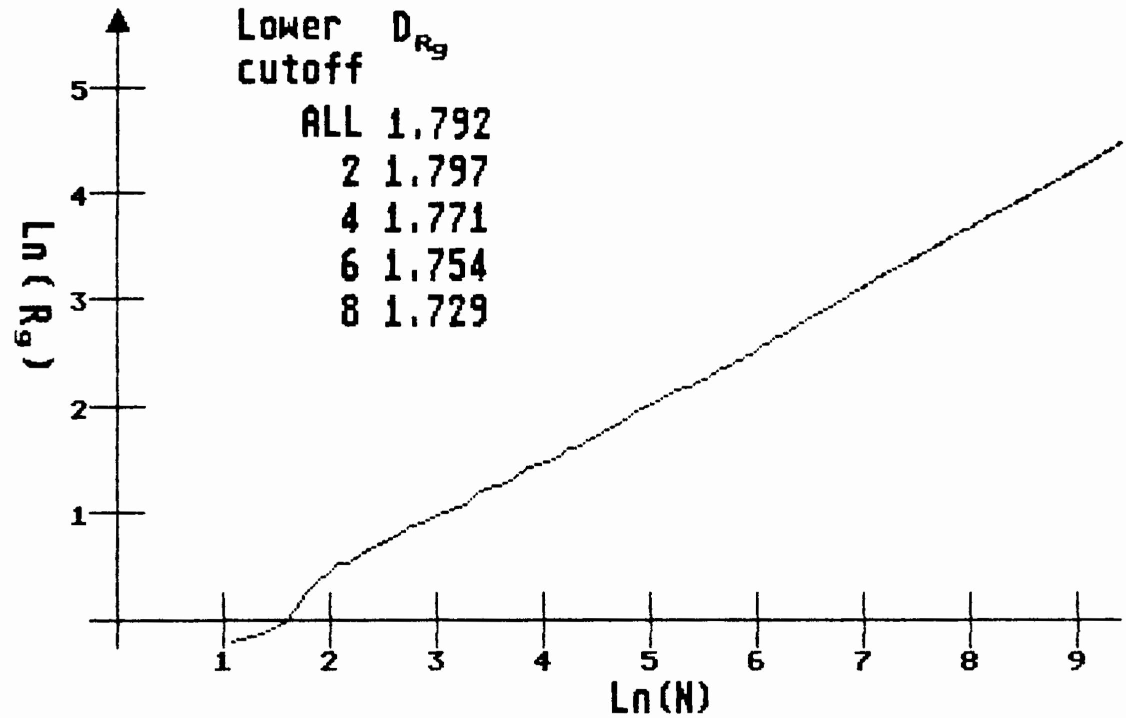


Figure 47. $Ln(R_g)$ vs. $Ln(N)$ for aggregate number 27.

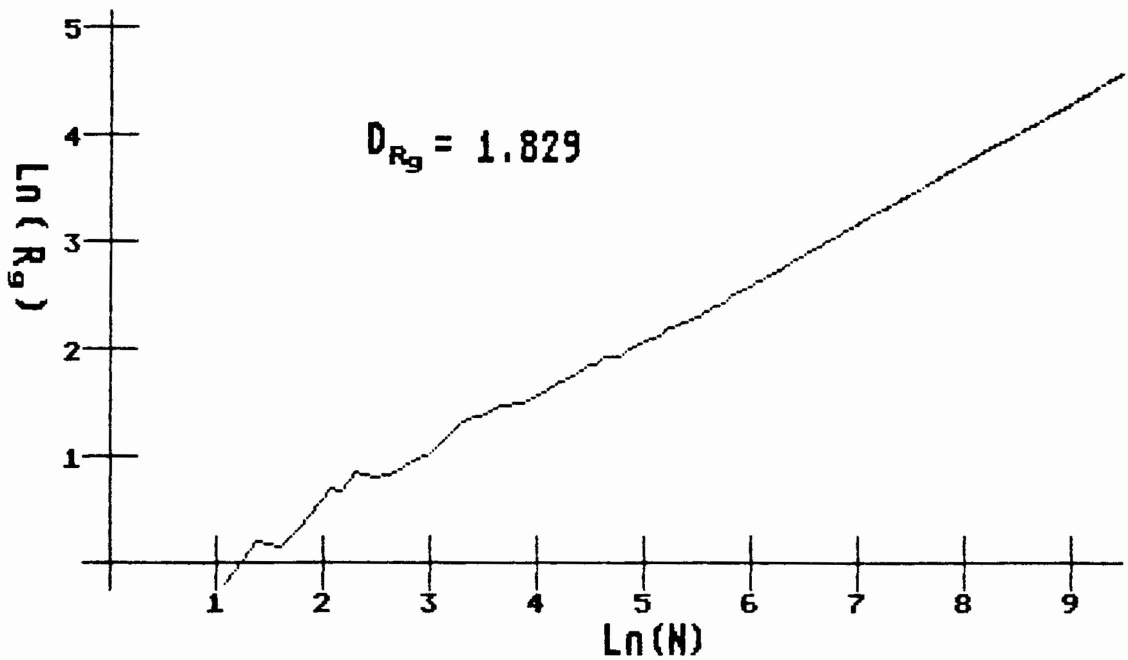


Figure 48. $Ln(R_g)$ vs. $Ln(N)$ for aggregate number 28.

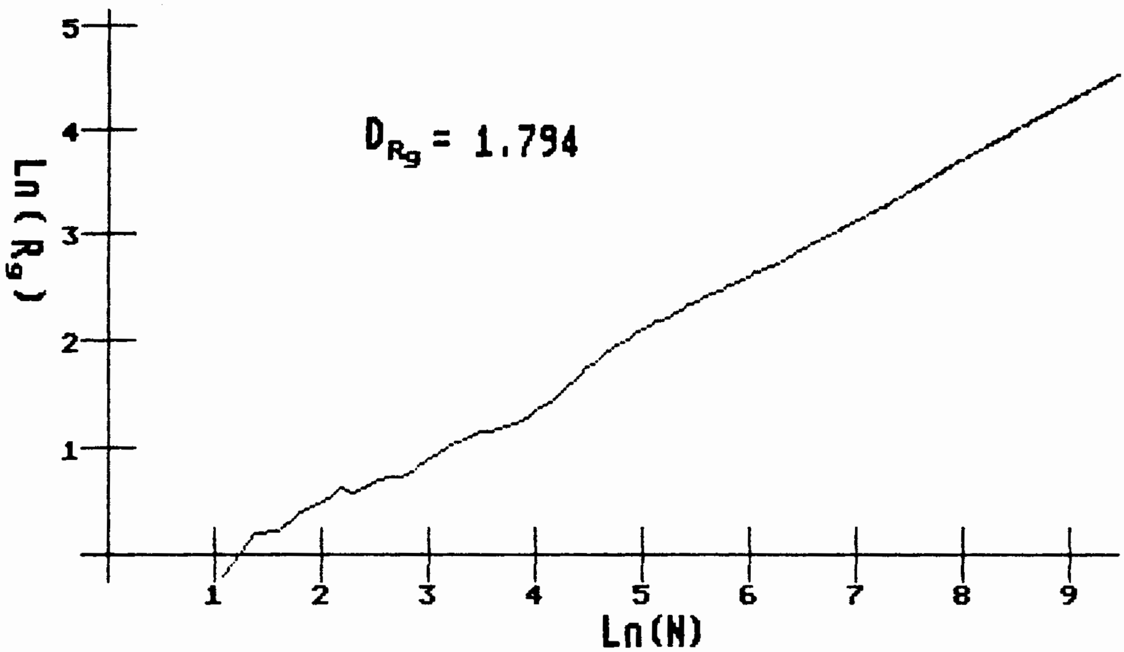


Figure 49. $Ln(R_g)$ vs. $Ln(N)$ for aggregate number 29.

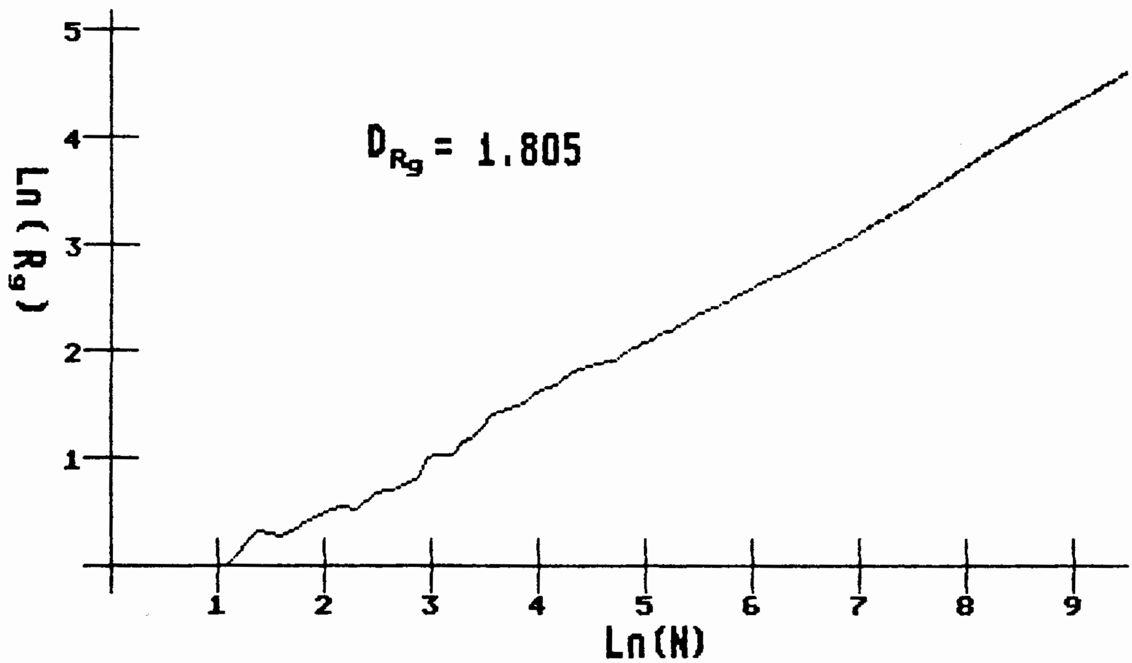
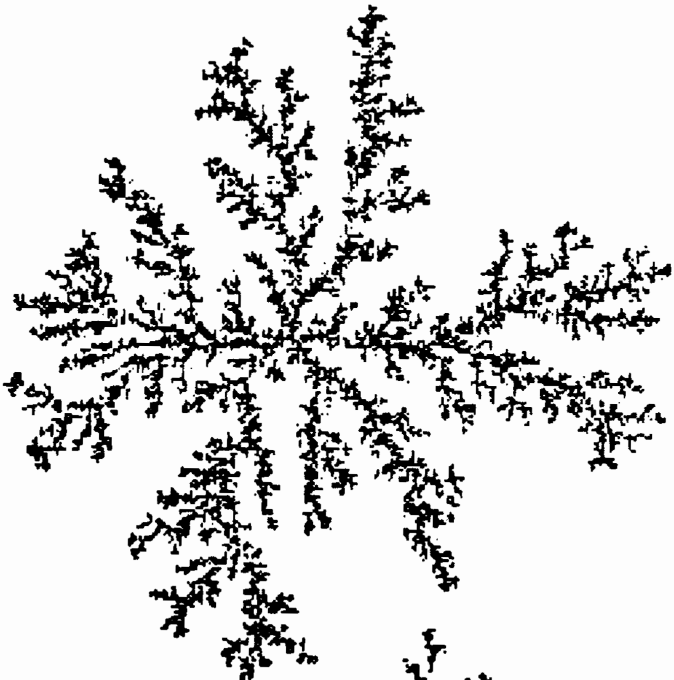


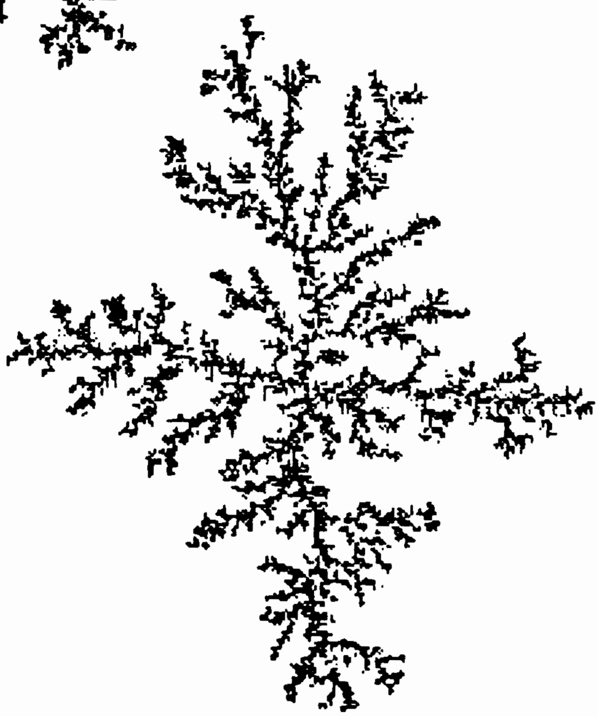
Figure 50. $Ln(R_g)$ vs. $Ln(N)$ for aggregate number 30.

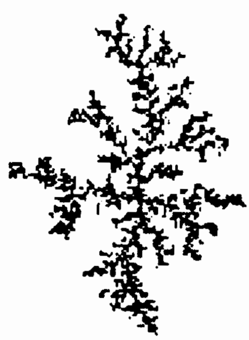
APPENDIX C

GRAPHICAL DATA

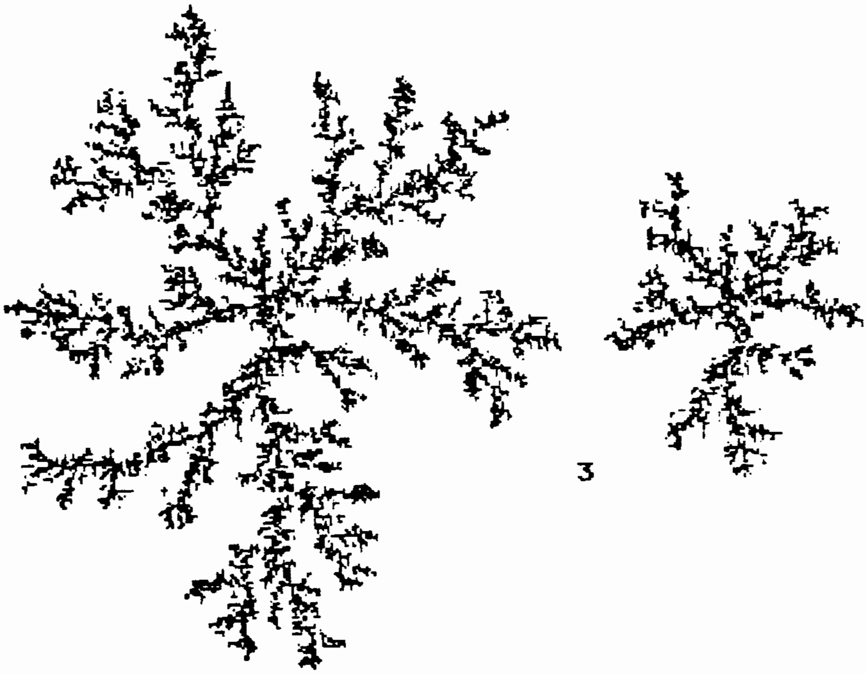


1

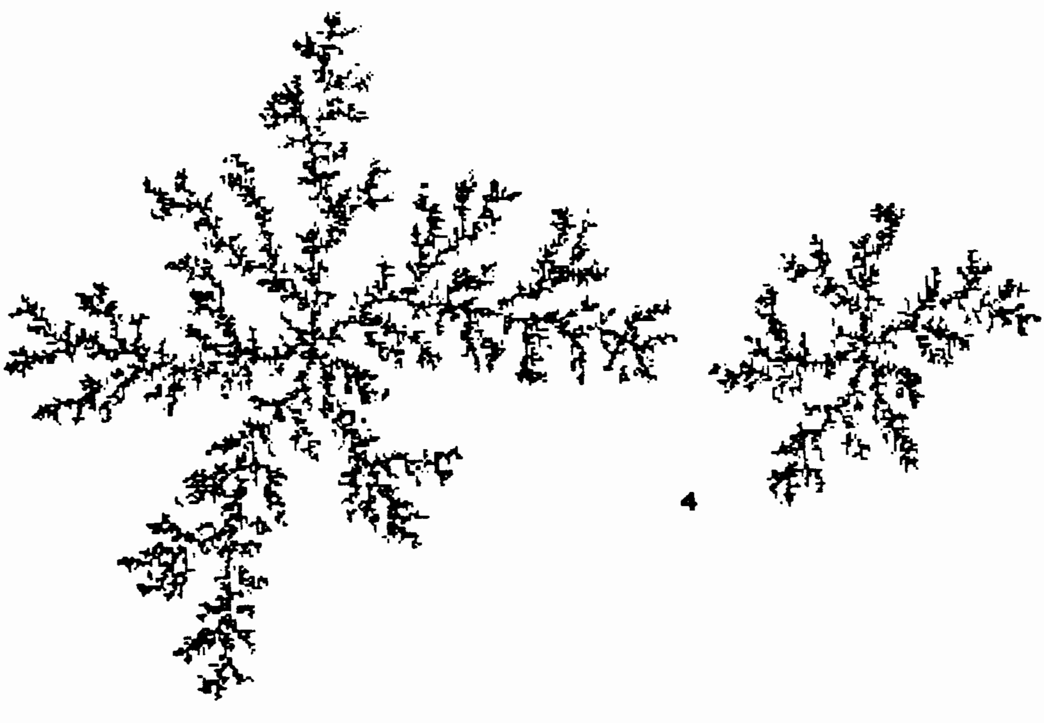




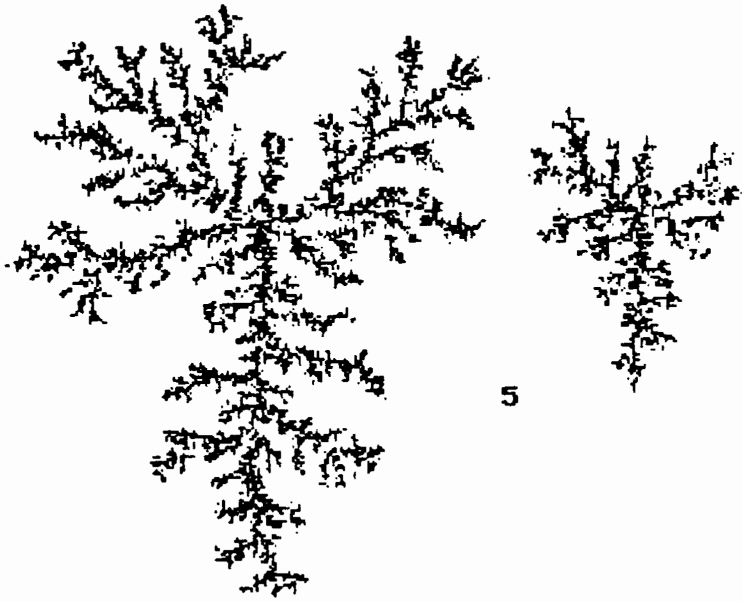
2



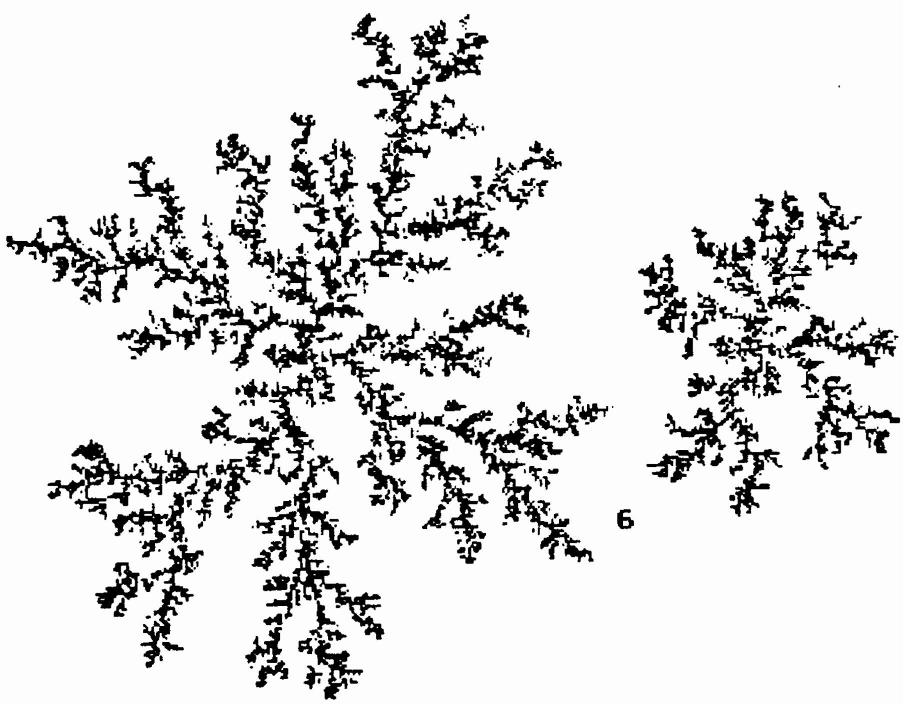
3



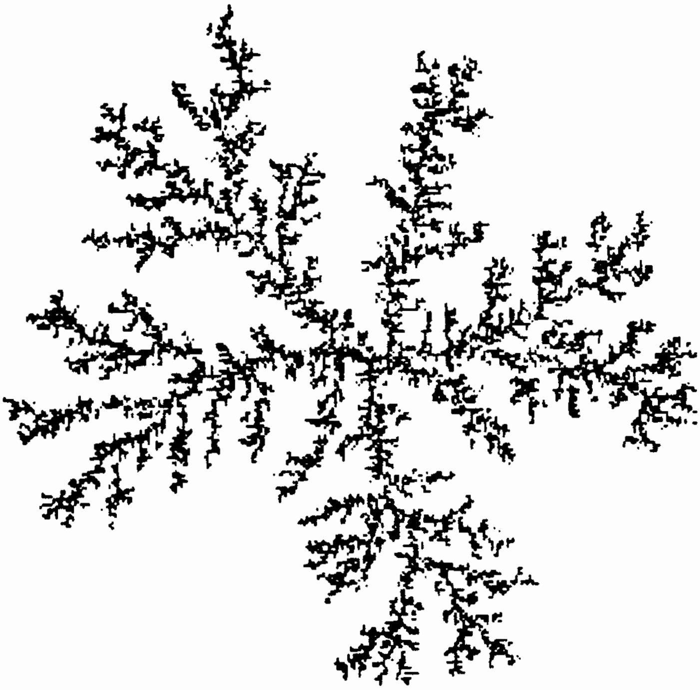
4

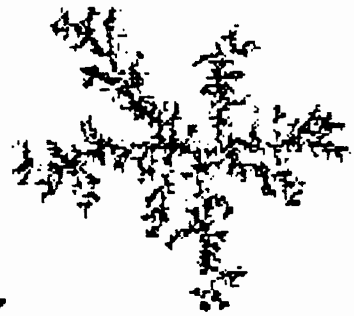


5

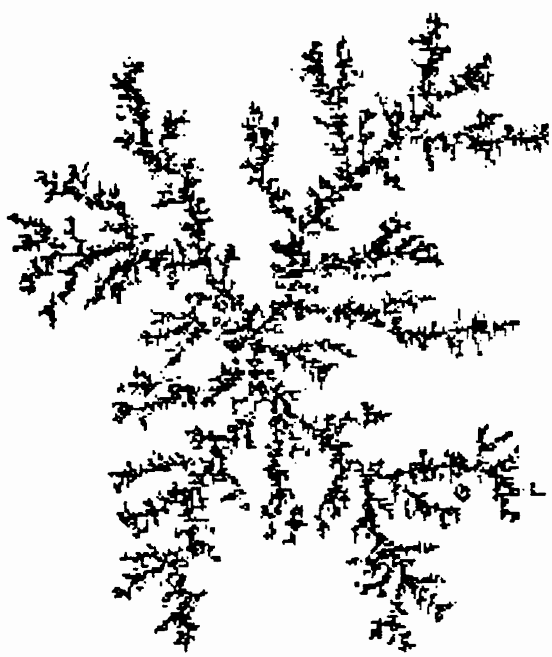


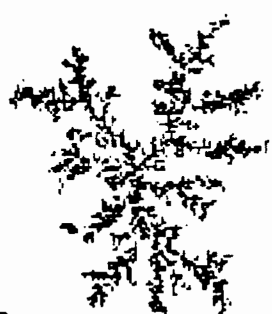
6



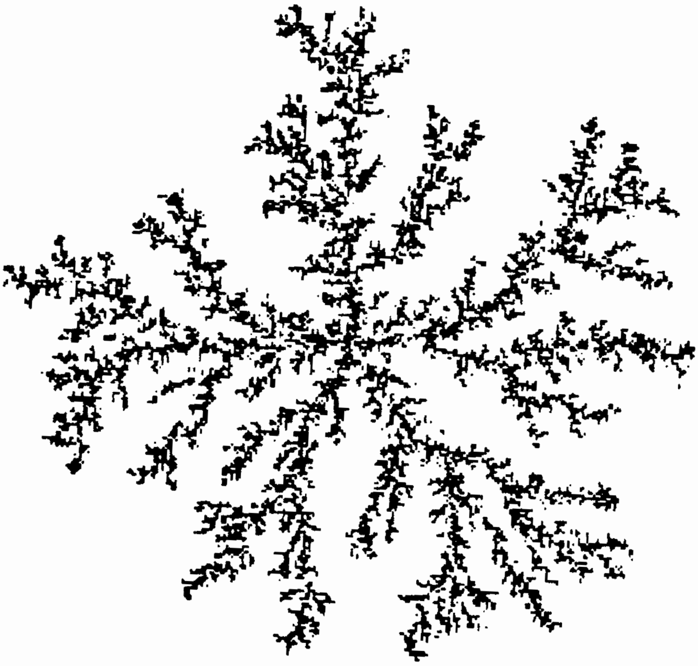
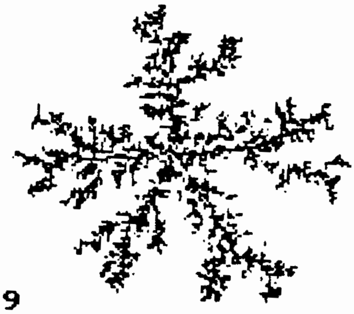


7

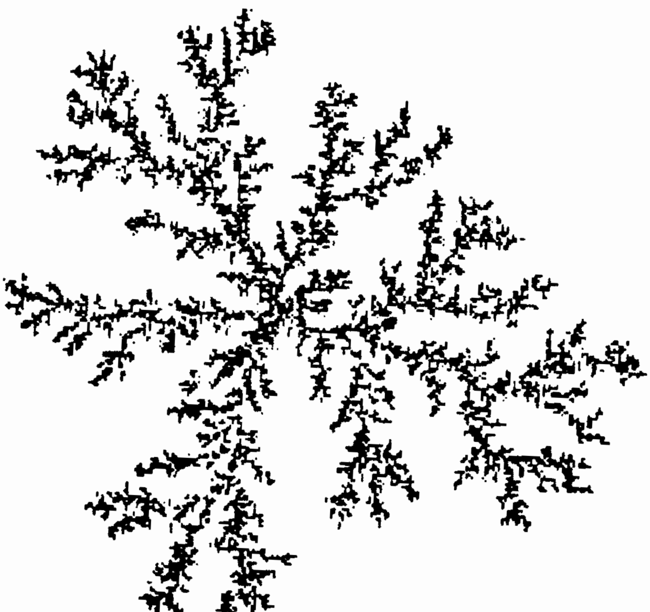




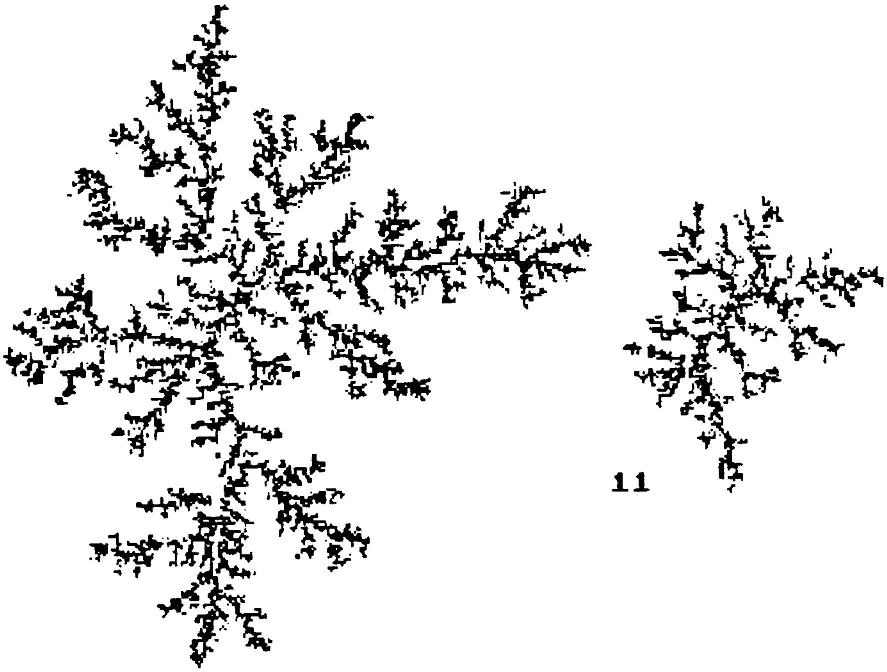
8

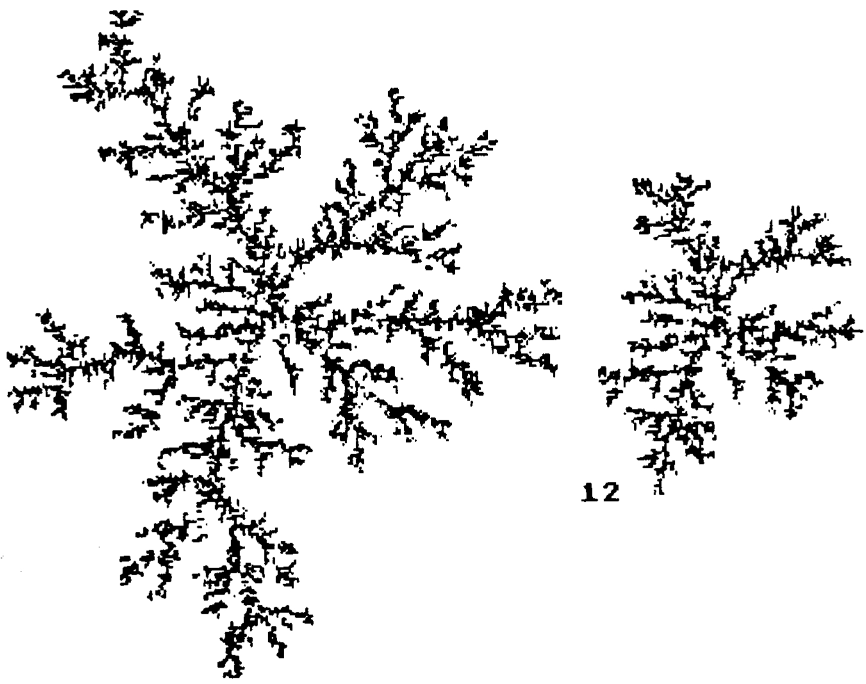
9



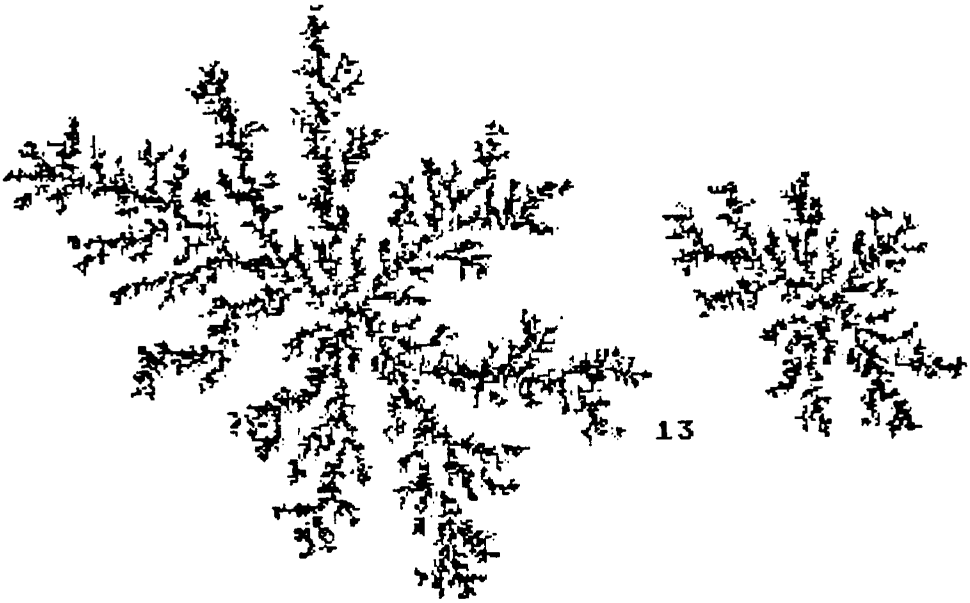

10



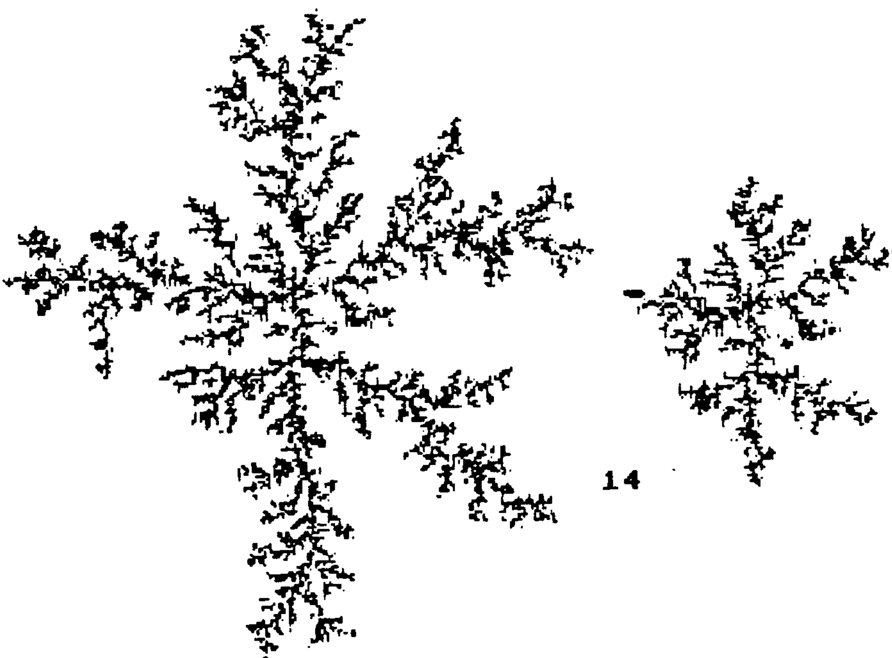
11



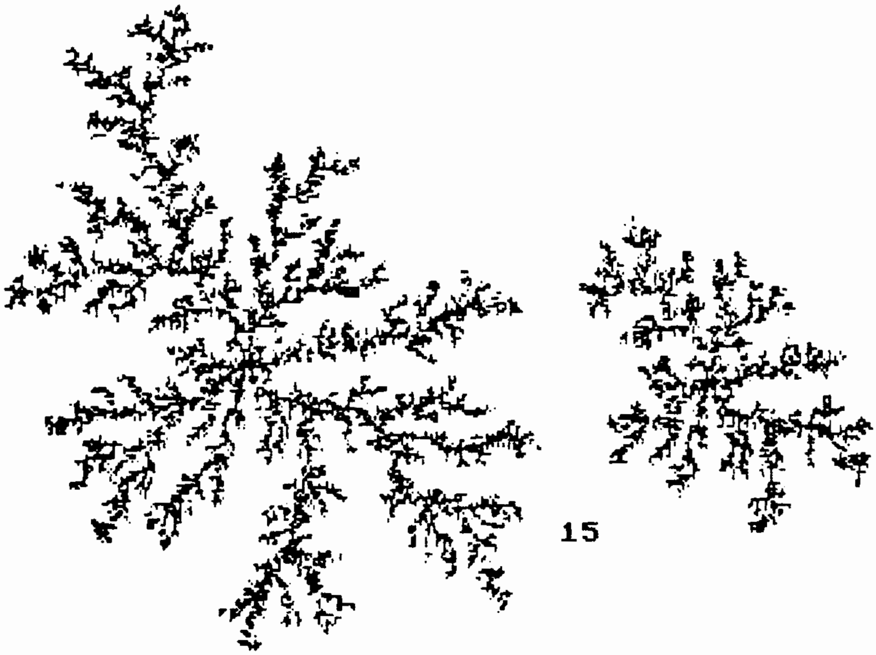
12



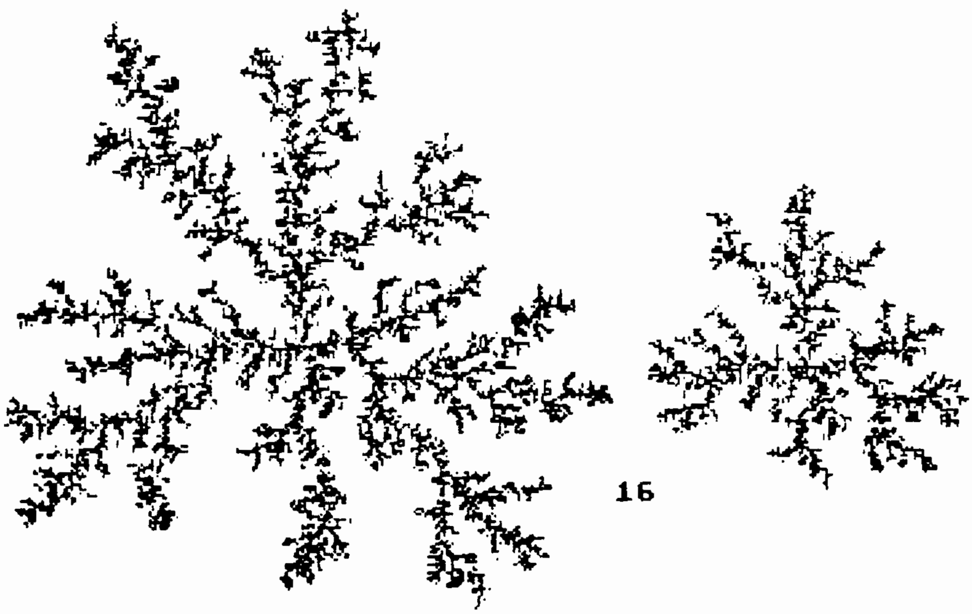
13



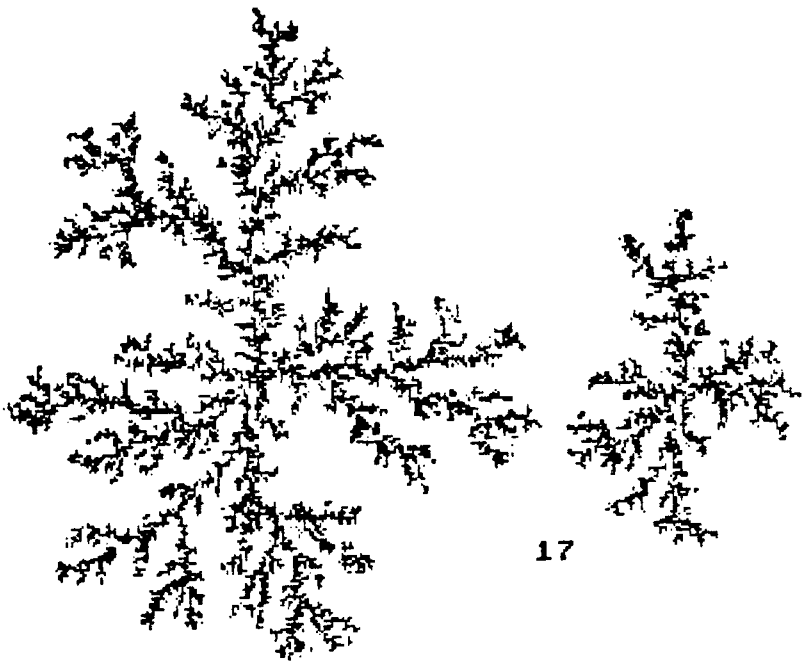
14



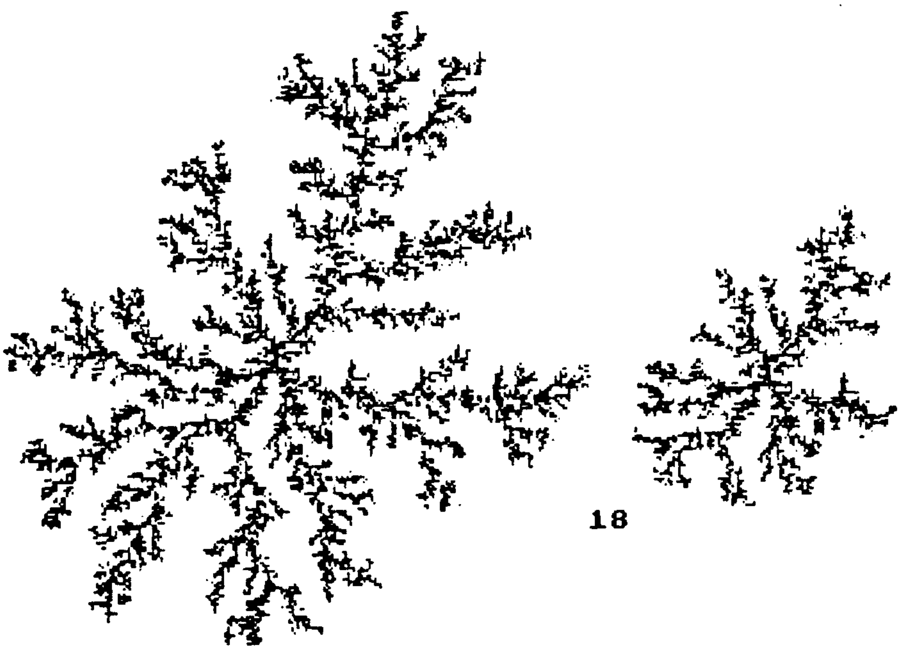
15



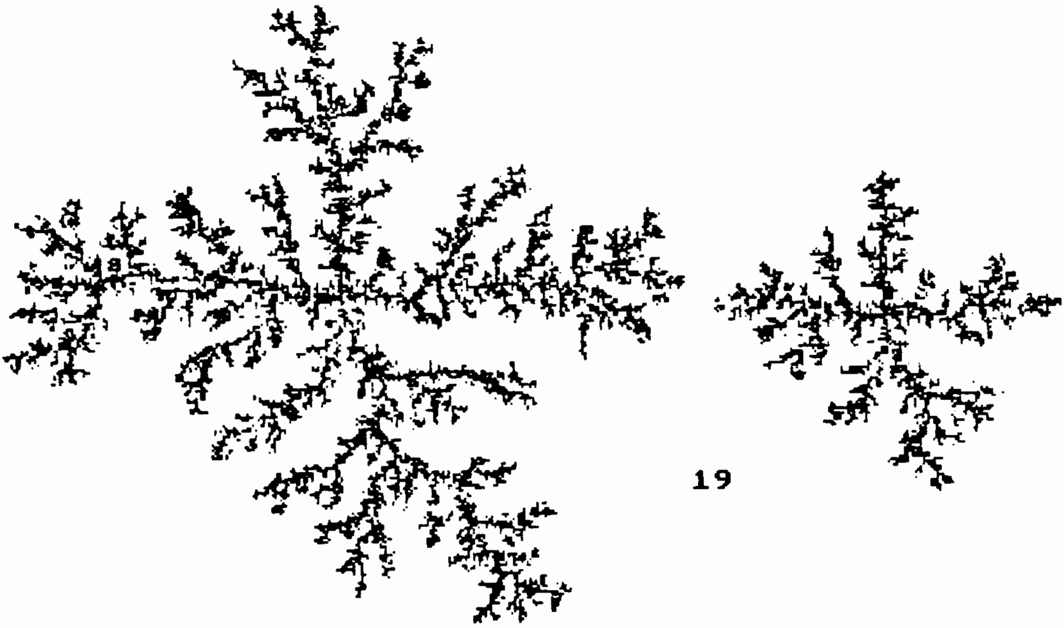
16



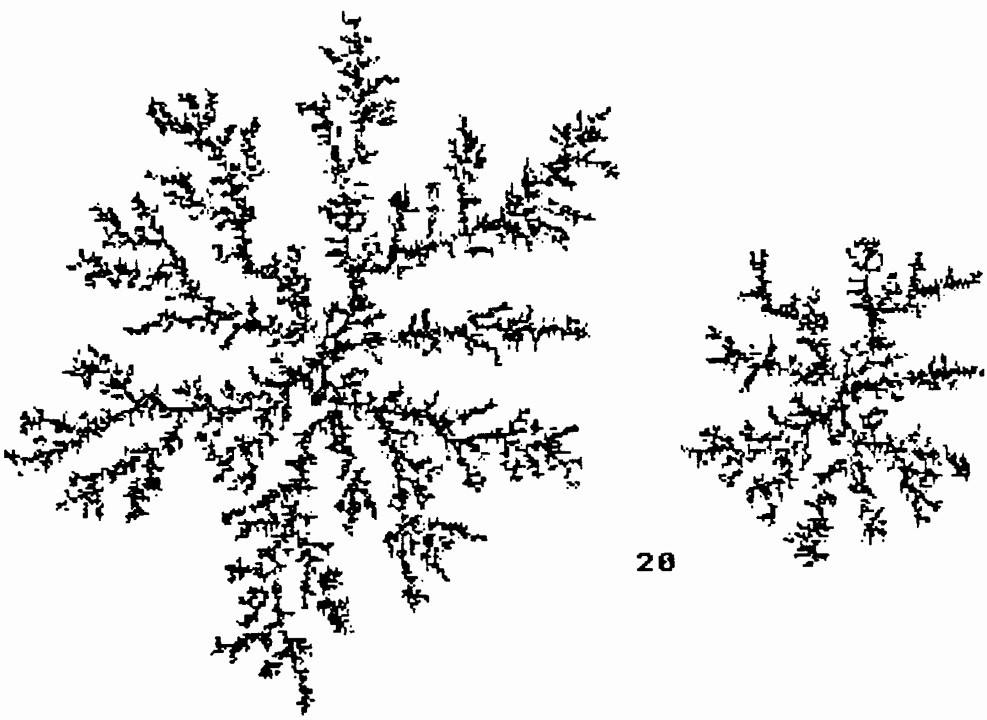
17



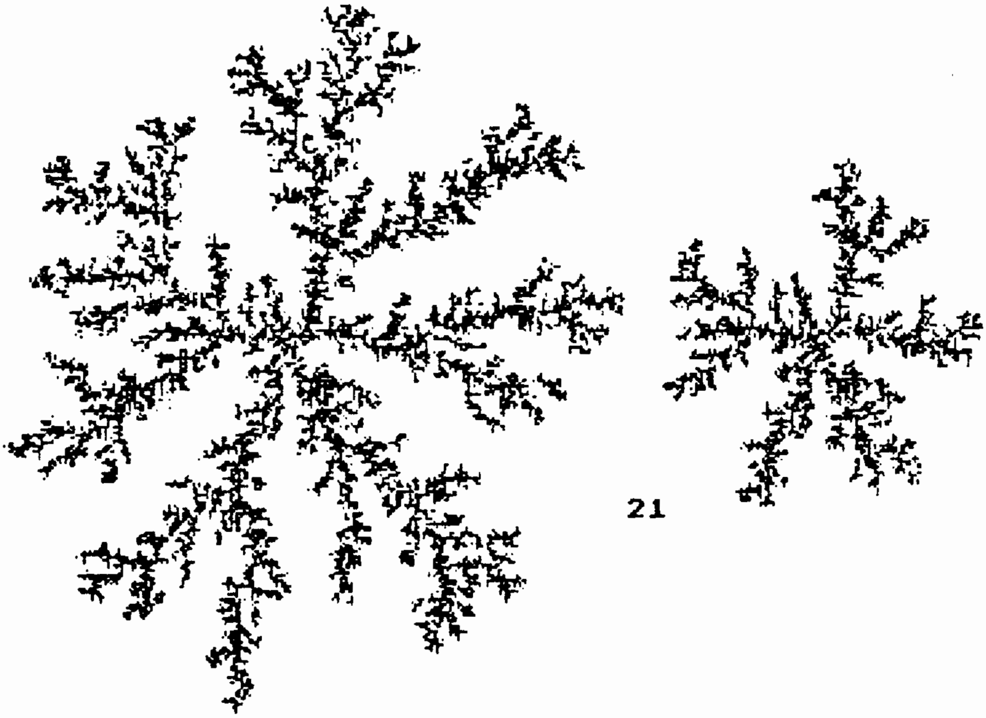
18



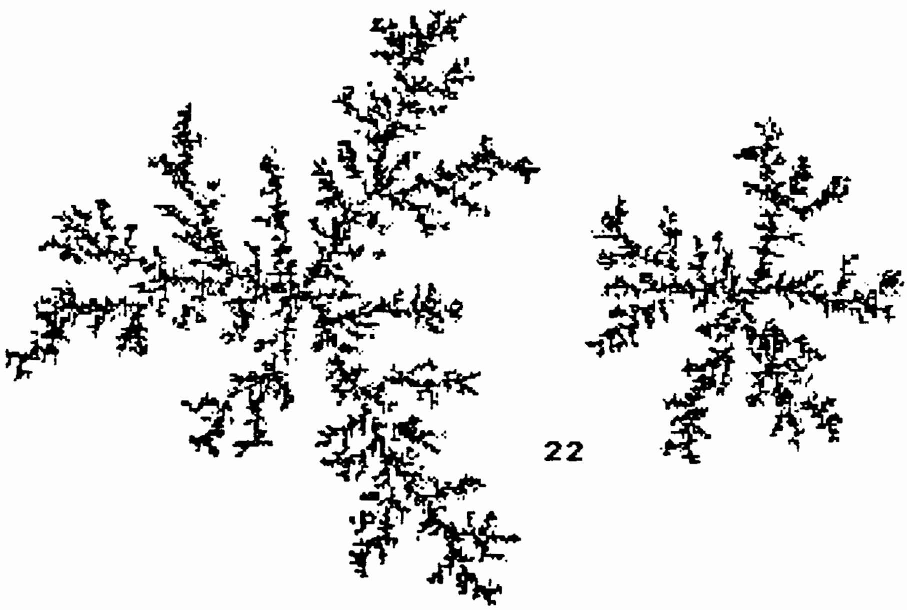
19



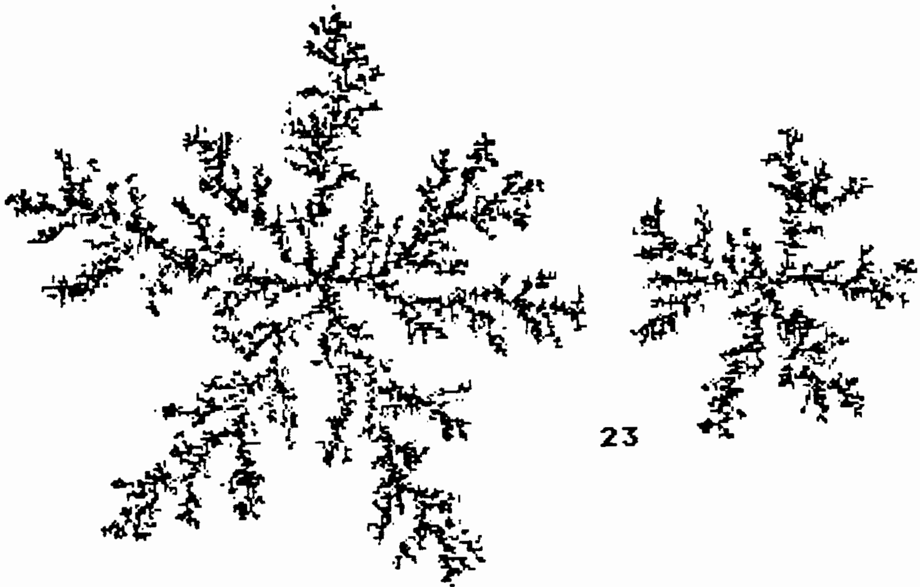
20



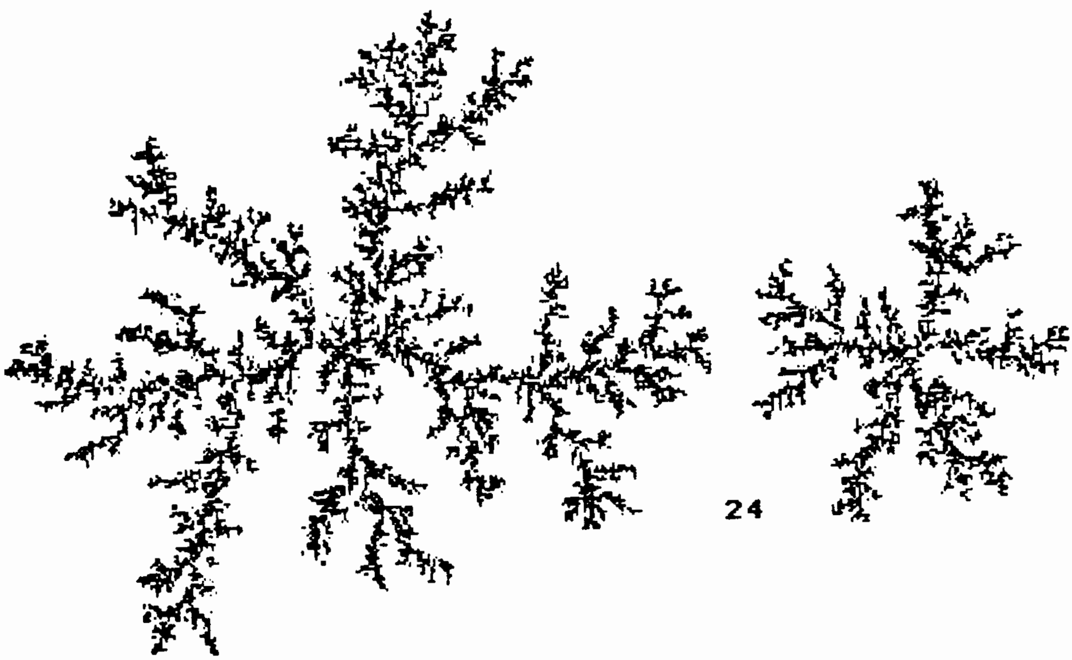
21



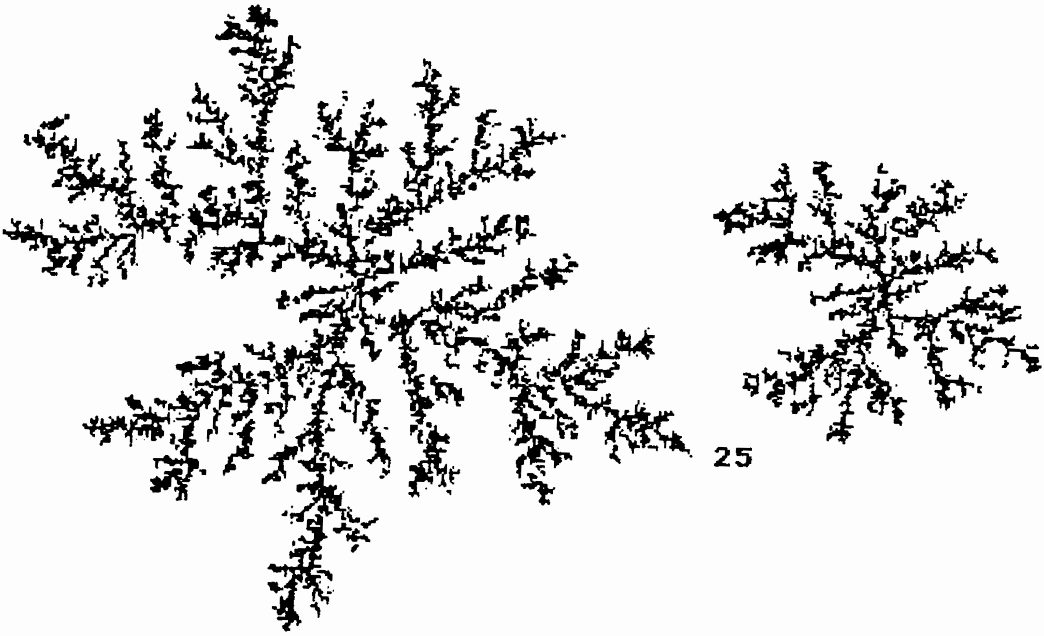
22



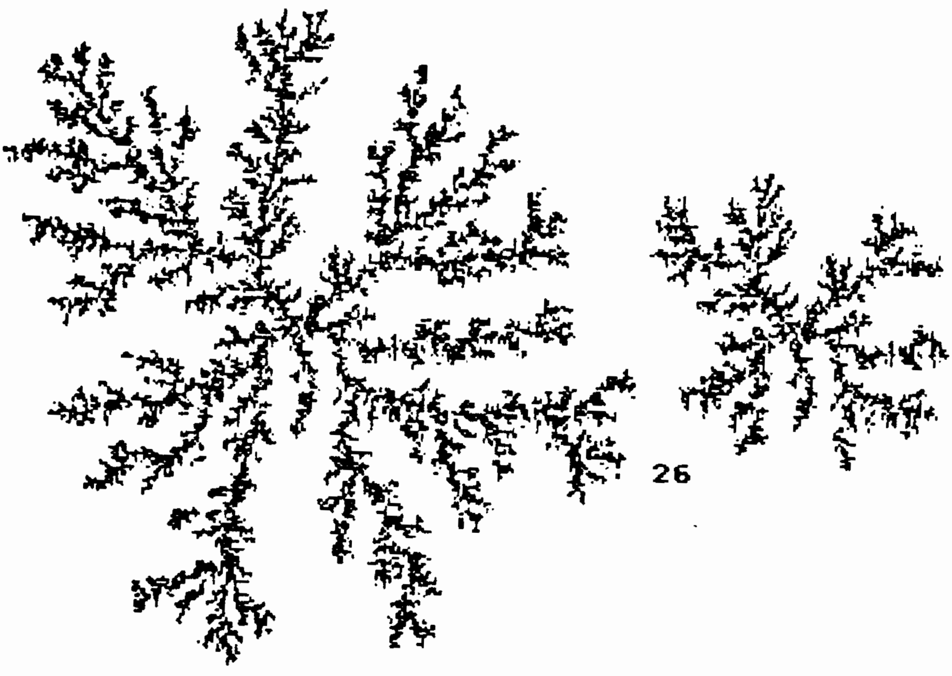
23



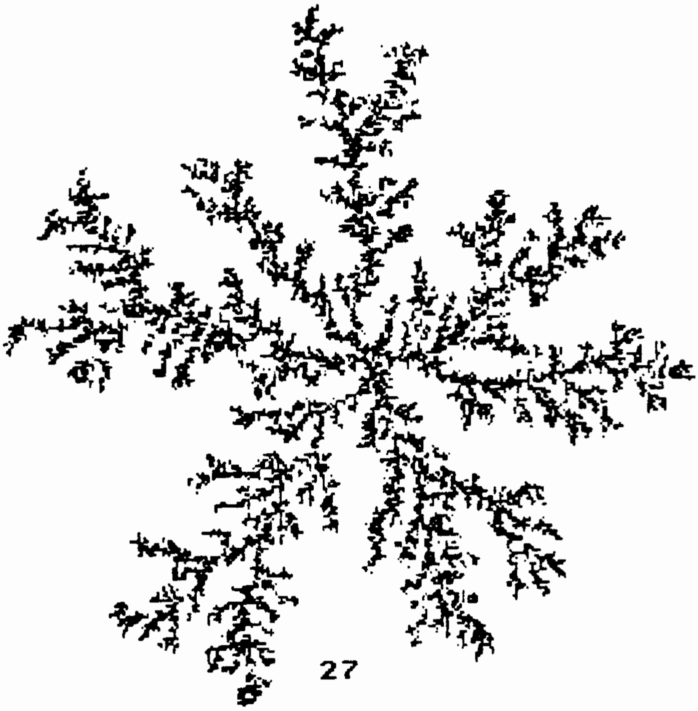
24



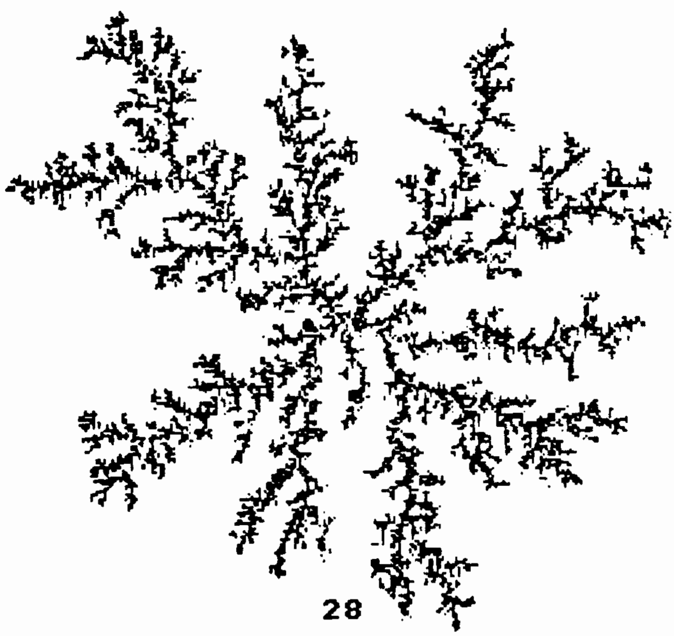
25



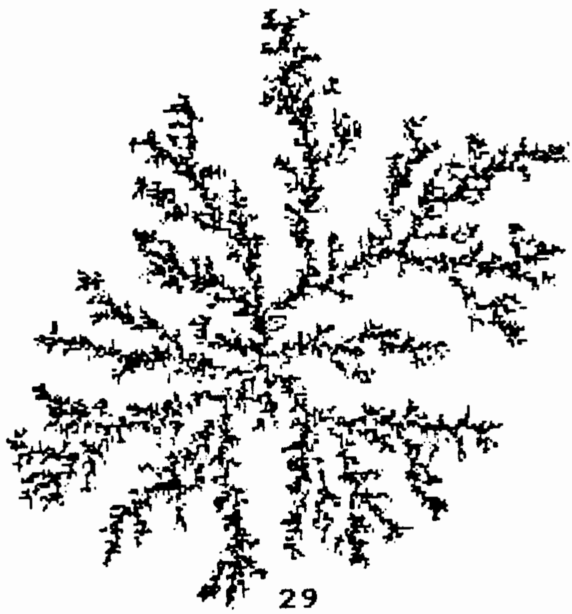
26



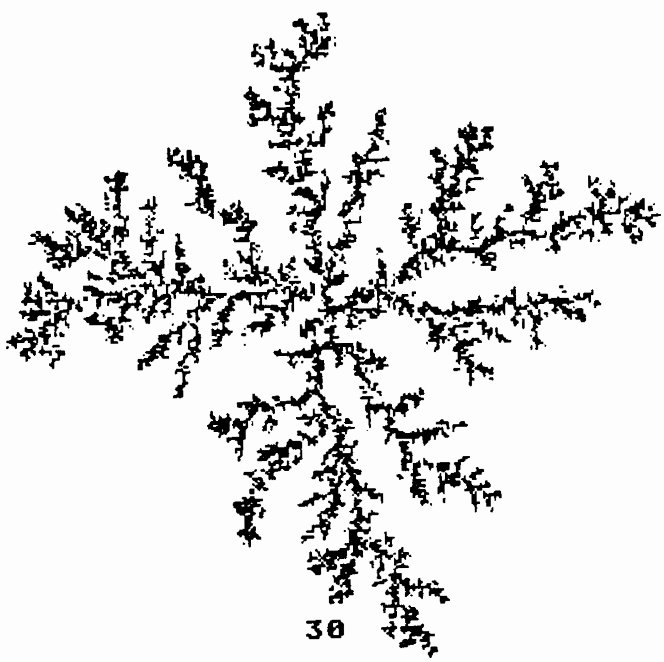
27



28

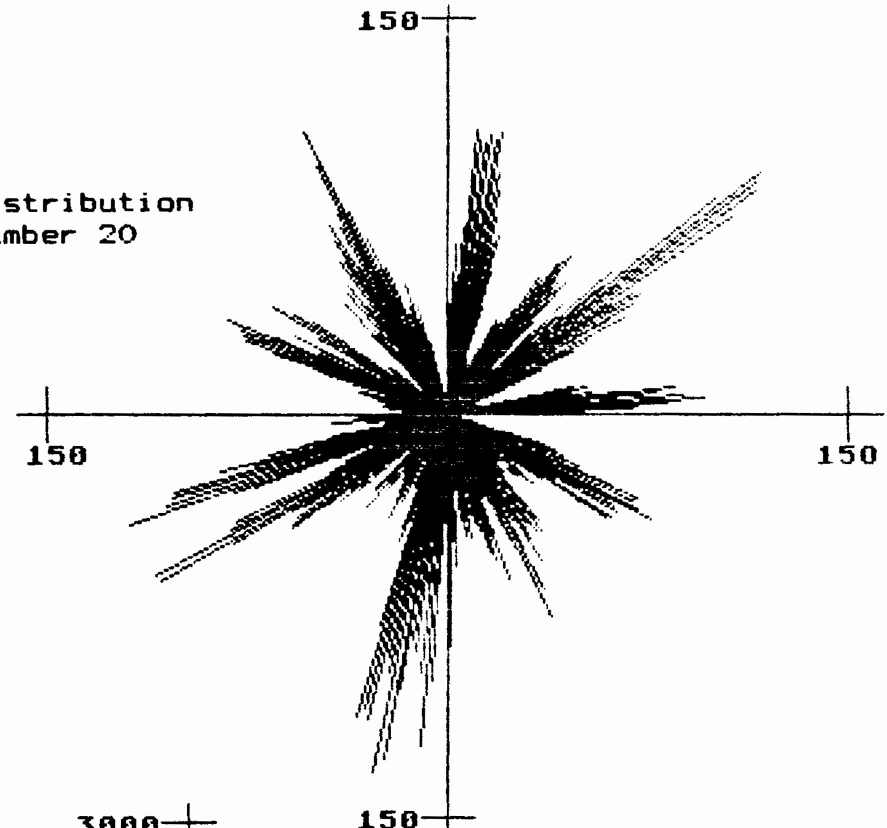


29

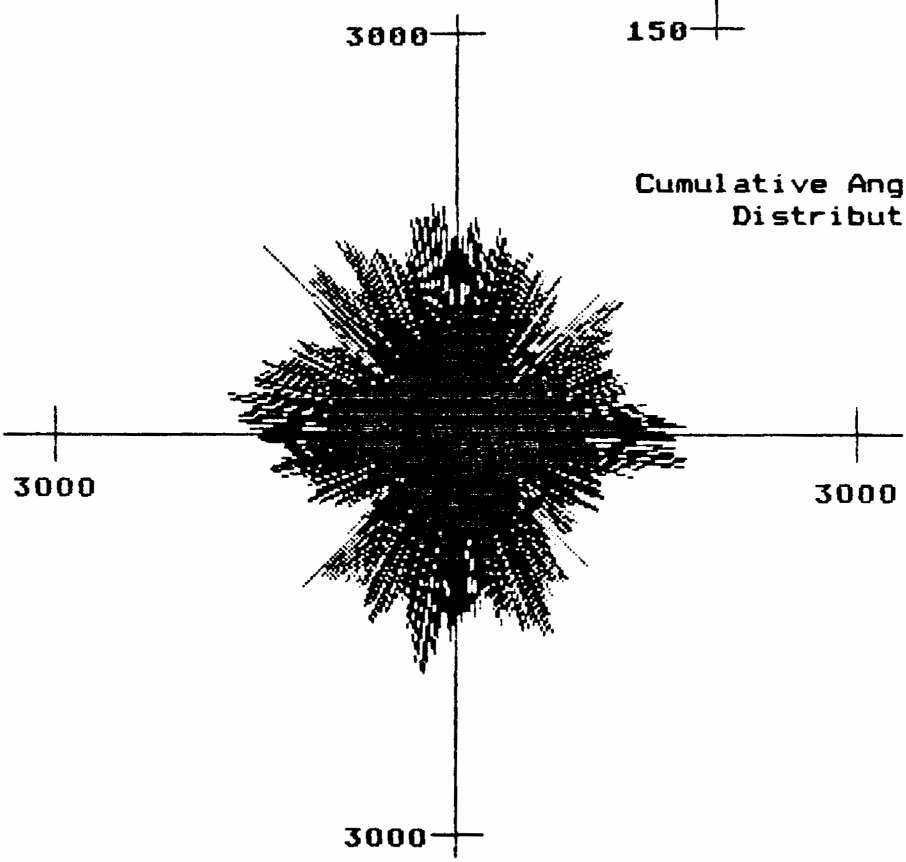


30

Angular Mass Distribution
Aggregate Number 20



Cumulative Angular Mass
Distribution



APPENDIX D

ADDITIONAL RADIUS OF GYRATION ANALYSIS

The radius of gyration is defined as the average sum of squares of the distances from the center of deposition to each deposit. The exact calculation of the radius of gyration dependence on the number of deposits would have necessitated N recalculations for the center of deposition and consequently a much longer process time. The assumption was made that the average center of deposition, for a large sample of aggregates, would be near the lattice origin. However, as discussed above, the average center of deposition was appreciably displaced from the origin. Moreover, the discrepancy in the fractal dimension, as based on this approximate radius of gyration, was unacceptable. In order to obtain a reasonable bound on this error it would be necessary to be able to estimate the dependence that this displacement had on the number of deposits. Analysis of the composite of all the aggregates and also of aggregate number 20, indicated that this displacement was not even monotonic. Instead of analyzing this distribution further, and estimating the fractal dimension using data that was known to be in error, it became obvious that it would be most prudent to recalculate the exact radius of gyration for a selected number of deposits and to obtain an approximate fractal dimension based on exact data. The following provides the details of the above argument and the resulting analysis.

The parallel axis theorem for the moment of inertia, $I = I_{c.m.} + N \cdot L^2$, where L is the displacement from the center of mass, c.m., can be utilized to modify the radius of gyration, $R_g = (I/N)^{1/2}$. The dependence, $L = L(N)$, was not obtainable, only $L(N_{max.})$ was known. Although, regression over all N of the deposits would have been the preferred method, however, without the corrections based on L(N), the results would have been systematically in error. A two-point approximation for the slope of $Ln(R_g)$ vs. Ln(N) could have been obtained (utilizing the parallel axis theorem with the final displacements of the centers of deposition) by using the final deposits of the small and large forms of the same aggregate (Slope = Slope($N_{max.}$)). However, recalculation of the radius of gyration based on the center of mass for a limited number of points would not have required an excessive amount of time. Thus, the radius of gyration program was modified and these data points were calculated directly. A more thorough analysis of aggregate number 20 was also performed in order to provide an additional comparison. These slopes, of 26 independent aggregates, were averaged. The result was compared with the slope of the least squares regression line based on the plot of the 52 data points. Any discrepancy here would indicate correlations between those data points associated with the large and small forms of the same aggregate.

The result of the two-point slope calculation for aggregate number 20 is, slope = 0.592, which gives a fractal dimension of D_{Rg} = 1.69. The results based on the approximate radius of gyration for aggregate number 20 from Appendix B, are, for the small aggregate, D_{Rg} = 1.83, and for the large aggregate, D_{Rg} = 1.81, their average is 1.82. Even though there is considerable variation among any of the individual deposits, this discrepancy is substantial. Aggregate number 20 was sampled at 20 increments of 5% of N_{MAX}. and this data was analyzed using least squares. The resulting fractal dimension based on the slope of the regression line is D_{Rg} = 1.67. The coefficient of determination, R^2, for the regression is 0.95. This is in close agreement with the more approximate result based on the two-point slope calculation. Thus, the two-point slope method yields credible results. The data obtained for aggregate number 20 is listed below in Table IV and the graph is in Figure 51.

The average of the two-point slope calculations of aggregates numbers 1 to 26, inclusive, using the final deposits of the small and large forms of each aggregate is, slope = .58 ± .02. This result yields a fractal dimension of 1.73 ± .06. The raw data for this calculation is listed

TABLE IV

CORRECTED RADIUS OF GYRATION RESULTS FOR AGGREGATE NUMBER 20

% N	DEPOSITS	Ln(N)	Ln(R_g)	$R_{c.m.}$
100	17715	9.782170	4.690065	4.017097
95	16829	9.730858	4.658574	4.096033
90	15944	9.676838	4.626698	3.901621
85	15058	9.619665	4.593242	4.014032
80	14172	9.559023	4.558203	4.011677
75	13286	9.494467	4.520182	4.383139
70	12401	9.425532	4.479921	4.448924
65	11515	9.351406	4.436583	4.809595
60	10629	9.271342	4.390161	5.046088
55	9743	9.184303	4.338853	5.412689
50	8858	9.089076	4.281730	6.012647
45	7972	8.983691	4.218701	5.740685
40	7086	8.865877	4.148299	5.921705
35	6200	8.732305	4.069269	6.429103
30	5315	8.578288	3.977641	7.069431
25	4429	8.395929	3.868652	7.602726
20	3543	8.172728	3.739125	7.767349
15	2657	7.884954	3.571964	6.718258
10	1772	7.479864	3.332235	5.915089
5	886	6.786717	2.950274	4.181003

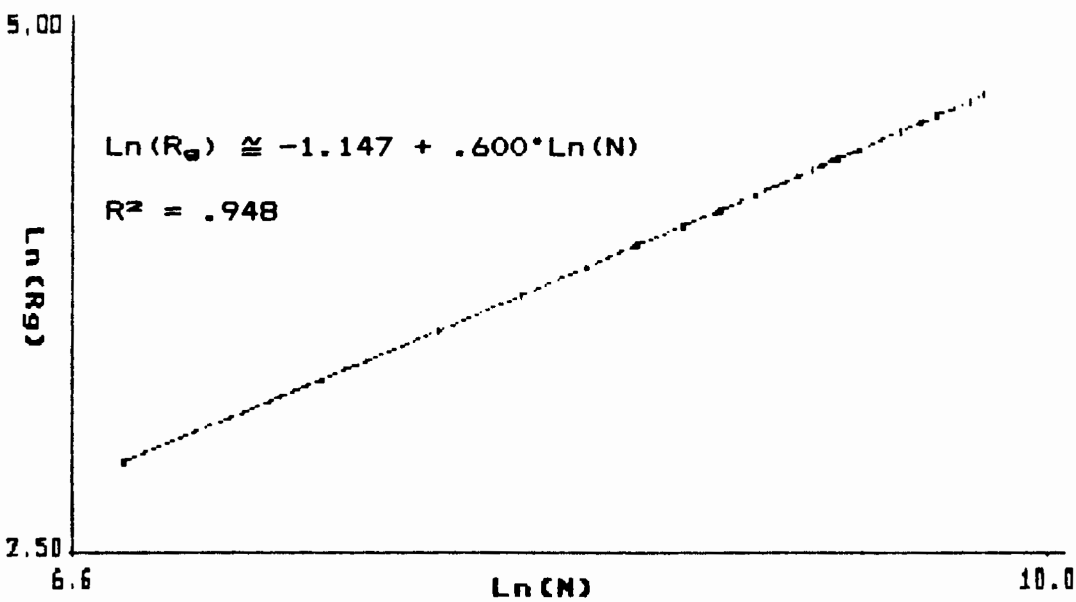


Figure 51. Corrected radius of gyration dependence on number of deposits for aggregate number 20.

in Table V and the coordinates are plotted in Figure 52. The graph was analyzed using linear regression and the slope of the regression line is, slope = .571. The correlation coefficient for the regression is, R = .99 and the residual variance is .028. These results yield a fractal dimension, $D_{Rg} = 1.75 \pm .08$. Additional analysis of the covariance of the paired points associated with the small and large forms of the aggregates was not performed because the results of the two methods of calculation were in agreement.

TABLE V

CORRECTED RADIUS OF GYRATION RESULTS FOR AGGREGATES NUMBERS 1 TO 20, INCLUSIVE

AGGREGATE	SMALL		LARGE	
NUMBER	L(N)	Ln(R_g)	L(N)	Ln(R_g)
1	8.469472	3.963612	9.765833	4.693800
2	8.249315	3.853862	9.607237	4.646392
3	8.267962	3.850265	9.642836	4.639245
4	8.438366	3.962984	9.631877	4.669662
5	7.996991	3.731075	9.409845	4.545592
6	8.510973	3.994435	9.801012	4.722027
7	8.442254	4.004926	9.874521	4.784022
8	8.378850	3.920564	9.712629	4.706285
9	8.582045	4.004637	9.964959	4.782405
10	8.414496	3.925898	9.752024	4.699201
11	8.105911	3.785593	9.525589	4.614721
12	8.438581	3.952668	9.574636	4.629936
13	8.418256	3.906203	9.636914	4.618569
14	8.240913	3.833317	9.461409	4.592153
15	8.517194	3.979296	9.608915	4.633099
16	8.525558	3.983830	9.700514	4.667667
17	8.241439	3.902611	9.612064	4.631654
18	8.393839	3.920980	9.701248	4.659043
19	8.391857	3.934570	9.664720	4.672567
20	8.659385	4.025738	9.782168	4.690065
21	8.615770	4.043975	9.865526	4.744267
22	8.162231	3.839269	9.442482	4.597081
23	8.563695	3.986628	9.708932	4.676501
24	8.326517	3.892380	9.750394	4.742227
25	8.558719	3.985567	9.735482	4.672455
26	8.533068	4.018763	9.884102	4.752329

Although time did not allow for additional analysis, an examination of the dependence that the displacement of the center of deposition has on the number of deposits could explain the concavity which was previously noticed in the graphs of $Ln(R_g)$ vs. $Ln(N)$. The previously mentioned cut offs in the regression analysis of, 0 to 6, only excluded a relatively small number of pixels (<2.5% of the average number of pixels, 16298). Furthermore, the displacement of the center of deposition appears to quickly attain a value comparable with the final displacement after only 5% of the total deposits. The sequence of regressions which indicated a convexity in the graphs of $Ln(R_g)$ vs. $Ln(N)$ (concavity in the fractal dimension) occurred over the same range of

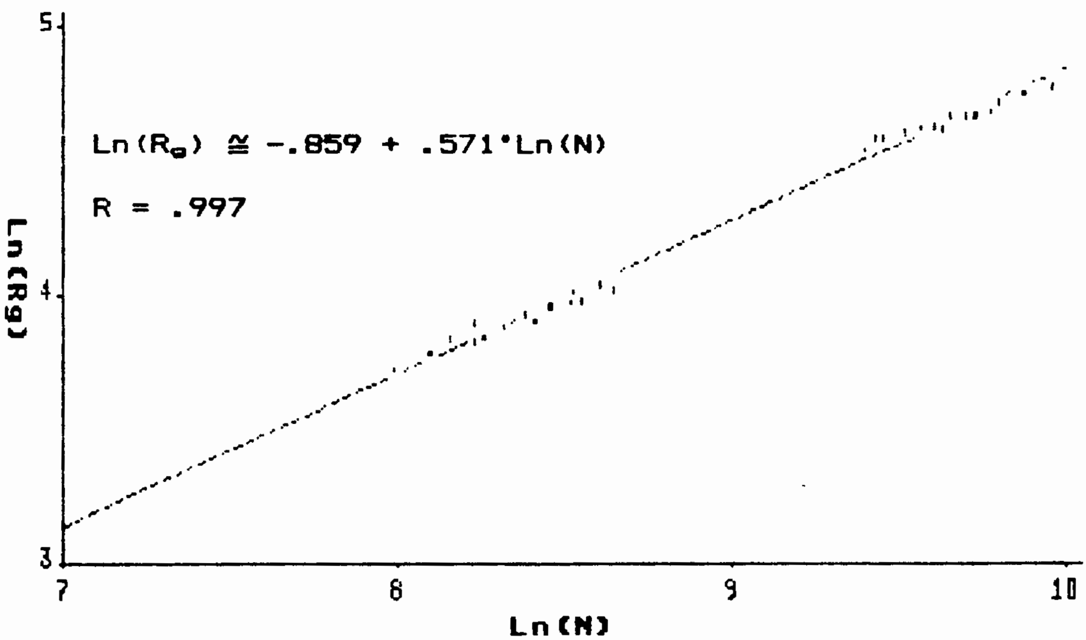


Figure 52. Corrected radius of gyration dependence on total number of deposits for 26 small and large aggregates.

deposition in which the displacement was convex, evident in the data shown in Table IV for $R_{C.M.}$ and N. This suggests that they are correlated just as the corrections to the formula for the radius of gyration would require and that the concavity may be related to the systematic error.

The estimate for the fractal dimension which is based on the average of the slopes is regarded as the most accurate. This result, $D_{RG} = 1.73 \pm .06$, reflecting the corrections in the radius of gyration, is approximately 3% less than the result which utilized the uncorrected radius of gyration.

APPENDIX E

CONSIDERATIONS FOR FURTHER WORK

In addition to those items already presented as subjects for further study, the following ideas could also provide more insight into the model.

Analysis of the effect of varying the width of the exclusion zone, or of making it more closely conform to the mean perimeter, instead of merely being concentric with the lattice origin, could provide insight into the active zone. The correlation function could also be separately evaluated over the excluded edge and the results compared to the results from the interior.

The average coordination number could be used to measure the local density and then be compared to the results of the correlation function. The sizes of the correlation windows could also be varied, although, no effect was noticed between the sizes used in this thesis to those used by Meakin.

The random walk routine could be altered with a deterministic component to simulate motion in an imposed field (Langevin equation).

The 'sticking' probability could be made to be a function of the local curvature, (Gibbs-Thompson relation) to realistically model solidification processes. Diffusion within the aggregate and 'slumping' of the perimeter could also be investigated .

The number of jumps a random walker takes prior to deposition could be used as a psuedo-time in order to study the dynamics of growth. However, it would be necessary to adjust its values so that the velocities would not be greater for the longer jump distances in the diffusion zone.

The axial center of mass could be defined along the arms of the aggregate to study the motion of the arms. Patterns and cycles of movement, independent of and also in coordination with neighboring arms could possibly be detected.

Dimensionless ratios of the step-size in the deposition zone, the size of the random walkers, and the distance of interaction with the aggregate could be formed, analogous to the Peclet number, and could be related to the fractal dimension.

The deposition probability could be found using relaxation methods, similarly, a large deposit could be bombarded many times and the number of attempted depositions could be recorded for the perimeter sites also giving the probability distribution. It is expected that the tips of the arms would have the greatest probablity. The average penetration depth could also be found.

If a color monitor were used, the age of the deposits could be color coded, and each color could have different diffusion and deposition properties.

The geometry of the arms could be analyzed to determine what factors might affect the ratios of the length and spacing and lengths of the side branches.

Various boundary conditions could be utilized in place of a the 'killing' circle such as reflecting or toroidial, and the geometry of the boundary could be changed to model diffusion along a channel or at a planar surface.

Finally, seeds of different geometries could be utilized, in order to investigate how persistent a sharp corner might grow, or how a cavity might be filled in.