A fractal analysis of diffusion limited aggregation

Cliff Myers
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

Part of the Physics Commons

Let us know how access to this document benefits you.

Recommended Citation
https://doi.org/10.15760/etd.5931

This Thesis is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

Title: A Fractal Analysis of Diffusion Limited Aggregation.

APPROVED BY MEMBERS OF THE THESIS COMMITTEE:

David I. Paul, Chair

Erik Bodegom

Jack S. Semura

Eugéne 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.

David I. Paul, Chair

Erik Bodegom

Jack S. Semura

Eugene A. Enneking

APPROVED:

David I. Paul, Chair, Department of Physics

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

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
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>REFERENCES CITED</td>
<td>52</td>
</tr>
<tr>
<td>APPENDIX</td>
<td></td>
</tr>
<tr>
<td>A    THE COMPUTER PROGRAMS</td>
<td>53</td>
</tr>
<tr>
<td>B    NUMERICAL DATA</td>
<td>120</td>
</tr>
<tr>
<td>C    GRAPHICAL DATA</td>
<td>137</td>
</tr>
<tr>
<td>D    ADDITIONAL RADIUS OF GYRATION ANALYSIS</td>
<td>153</td>
</tr>
<tr>
<td>E    CONSIDERATIONS FOR FURTHER WORK</td>
<td>160</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>DESCRIPTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Average Fractal Dimensions</td>
<td>22</td>
</tr>
<tr>
<td>II</td>
<td>Fractal Dimension Data for Individual Small Aggregates</td>
<td>120</td>
</tr>
<tr>
<td>III</td>
<td>Fractal Dimension Data for Individual Large Aggregates</td>
<td>121</td>
</tr>
<tr>
<td>IV</td>
<td>Corrected Radius of Gyration Results for Aggregate Number 20</td>
<td>156</td>
</tr>
<tr>
<td>V</td>
<td>Corrected Radius of Gyration Results for Aggregates Numbers 1 to 20, Inclusive</td>
<td>157</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Scale Invariance of a Fractal Aggregate</td>
<td>5</td>
</tr>
<tr>
<td>2. Sierpinski Gasket</td>
<td>6</td>
</tr>
<tr>
<td>3. Correlation Dependence on 'Window' Size</td>
<td>27</td>
</tr>
<tr>
<td>4. Fractal Dimension vs. Aggregate Number</td>
<td>28</td>
</tr>
<tr>
<td>5. Radius of Gyration Dependence on Number of Deposits for Small and Large Aggregates</td>
<td>32</td>
</tr>
<tr>
<td>6. Aggregate Number 20</td>
<td>34</td>
</tr>
<tr>
<td>7. Quartile Stages of Growth of Aggregate Number 20</td>
<td>35</td>
</tr>
<tr>
<td>8. Radius of Gyration Dependence on Number of Deposits for Aggregate Number 20</td>
<td>36</td>
</tr>
<tr>
<td>9. Radial Mass Distribution for Aggregate Number 20</td>
<td>36</td>
</tr>
<tr>
<td>10. Angular Mass Distribution for Aggregate Number 20 (Smoothed)</td>
<td>37</td>
</tr>
<tr>
<td>11. Mass Distribution in X for Aggregate Number 20</td>
<td>38</td>
</tr>
<tr>
<td>12. Mass Distribution in Y for Aggregate Number 20</td>
<td>39</td>
</tr>
<tr>
<td>13. Upper Percentiles of Deposition for Aggregate Number 20</td>
<td>40</td>
</tr>
</tbody>
</table>
14. Cumulative Probability Distribution in X and Y ........................................ 41
15. Cumulative Angular Mass Distribution of the 30 Large Aggregates (Smoothed) ........... 43
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_e) vs. Ln(N) for Aggregate Number 1 ...... 122
22. Ln(R_e) vs. Ln(N) for Aggregate Number 2 ...... 122
23. Ln(R_e) vs. Ln(N) for Aggregate Number 3 ...... 123
24. Ln(R_e) vs. Ln(N) for Aggregate Number 4 ...... 123
25. Ln(R_e) vs. Ln(N) for Aggregate Number 5 ...... 124
26. Ln(R_e) vs. Ln(N) for Aggregate Number 6 ...... 124
27. Ln(R_e) vs. Ln(N) for Aggregate Number 7 ...... 125
28. Ln(R_e) vs. Ln(N) for Aggregate Number 8 ...... 125
29. Ln(R_e) vs. Ln(N) for Aggregate Number 9 ...... 126
30. Ln(R_e) vs. Ln(N) for Aggregate Number 10 ...... 126
31. Ln(R_e) vs. Ln(N) for Aggregate Number 11 ...... 127
32. Ln(R_e) vs. Ln(N) for Aggregate Number 12 ...... 127
33. Ln(R_e) vs. Ln(N) for Aggregate Number 13 ...... 128
34. Ln(R_e) vs. Ln(N) for Aggregate Number 14 ...... 128
35. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 15 .... 129
36. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 16 .... 129
37. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 17 .... 130
38. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 18 .... 130
39. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 19 .... 131
40. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 20 .... 131
41. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 21 .... 132
42. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 22 .... 132
43. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 23 .... 133
44. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 24 .... 133
45. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 25 .... 134
46. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 26 .... 134
47. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 27 .... 135
48. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 28 .... 135
49. \( \ln(R_o) \) vs. \( \ln(N) \) for Aggregate Number 29 .... 136
50. \( \ln(R_o) \) 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.

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, \( \lambda \)). (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, \( \rho \), given by \( \rho = M/L^d \) for exact fractals is
\[
\rho = L^{D-d}. \tag{2}
\]

For the Sierpinski gasket of Figure 2, the mass scales according to \( M(2L) = 3M(L) = 2^D M(L) \) and \( D = \ln 3/\ln 2 \approx 1.585 \).

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:

\[ N(r) = (r/r_0)^D \]  

(3)

where \( N(r) \) is the quantity obtained by measuring a fractal medium with a gauge \( r_0 \). 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, \( C(r) = N^{-1} \sum \delta(r_i+r)\delta(r_i) \) 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:

\[ C(r) = r^D r^{-d} = r^{D-d} = r^\alpha. \]  

(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 $r \ll L_{\text{Max}}$. 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, \( R_{\text{inject}} = R_{\text{max}} + 5 \). The random walker was also 'killed' if \( R > R_{\text{kill}} = 3R_{\text{max}} \).

With an average aggregate size of 9700 pixels, Meakin obtained fractal dimensions, of \( 1.68 \pm 0.04 \) and \( 1.68 \pm 0.07 \) 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 \( r_{\text{MAX}} + 5 \) lattice units from the center seed, four units, if greater than \( r_{\text{MAX}} + 10 \) units, four, if greater than \( r_{\text{MAX}} + 20 \), eight if greater than \( r_{\text{MAX}} + 40 \), and sixteen if \( r_{\text{MAX}} + 80 \). 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 $x$ site is reached on the $k+1$ step is: (following Witten and Sander, 1983)

$$u(x, k+1) = \frac{1}{4} \sum_{\text{neighbors}} u(x+1, k),$$

where the summation over $\text{neighbors}$ runs over the 4 neighbors of $x$ 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 incoming particle (equivalent to the average concentration if many were simultaneously diffusing), with $\beta$ as the diffusion constant:
\[ \frac{\partial u}{\partial t} = B \nabla^2 u. \] (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 \( x \to \infty \). 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:

\[ \nabla^2 u = 0. \] (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, \( \nabla \Phi \), has zero divergence (\( \nabla \times \nabla \Phi = \nabla^2 \Phi = 0 \)). 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, \( E \), at the surface of the deposit (\( E \times -\nabla V = 0 \), and \( \nabla^2 V = 0 \)).
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 ± 0.03 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 ± 0.03.

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, $1.70 \pm .05$.

**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 SF₆ 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:

\[
\begin{align*}
\quad P(-v_{\text{normal}}) &= 1, \\
\quad P(+v_{\text{normal}}) &= P(\pm v_{\text{tangential}}) = 0.
\end{align*}
\]

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, \( R \leq R_{\text{max}} + 5 \), 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 $R_{\text{Inject}} = R_{\text{MAX}} + 5$ 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 $10^N \cdot 2^N < R - R_{\text{MAX}} < 10^N \cdot 2^{N-1}$ 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 $R_{\text{KILL}} = 2 \cdot R_{\text{MAX}} + 5$. 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, \( R > R_{\text{max}} = 32.5 \) 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 \( 2^n + 0.5 \) 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

Fractal Dimension from Average Correlation 'Window' Data

<table>
<thead>
<tr>
<th></th>
<th>Including Edge</th>
<th>Excluding Edge</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Squares</td>
<td>'Circles'</td>
</tr>
<tr>
<td>Small Aggregates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1.66410592</td>
<td>1.610013451</td>
</tr>
<tr>
<td>s.d.</td>
<td>.0082032213497</td>
<td>.0079478124734</td>
</tr>
<tr>
<td>Large Aggregates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1.6668462298</td>
<td>1.6107480877</td>
</tr>
<tr>
<td>s.d.</td>
<td>.0053549107253</td>
<td>.0050211512165</td>
</tr>
</tbody>
</table>

Fractal Dimension from Composite of all Aggregates based on Radius of Gyration

|                |                |
| Small Aggregates | 1.8452894007  |
| Large Aggregates | 1.8120055785  |

Average Aggregate Size

Small Aggregates $N = 4510 \pm 702$ pixels
Large Aggregates $N = 16298 \pm 2159$ 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 $D_{\text{c}} = 1.639 \pm 0.012$ and exact calculation yielded results of $D_{\text{e}} = 1.696 \pm 0.012$. For the large aggregates, results were, $D_{\text{c}} = 1.616 \pm 0.006$ and $D_{\text{e}} = 1.673 \pm 0.006$, 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'; \( D_c = 1.696 \pm 0.012 \) and \( D_s = 1.695 \pm 0.012 \), respectively, and for the large aggregates the dimensions were identical, \( D_c = D_s = 1.673 \pm 0.006 \). 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, $D_e = 1.664 \pm 0.008$, and for the large aggregates, $D_e = 1.667 \pm 0.005$. 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 $D = 1.67 \pm 0.01$. 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 $D = 1.67 \pm 0.01$. 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.

**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.

**Figure 3.** Correlation dependence on 'window' size.
Figure 4. Fractal dimension vs. aggregate number.
Radius of Gyration Results

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 $\leq 2\%$. 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, $D = 1.78 \pm 0.01$. This does not agree with the correlation function results. The relative discrepancy is $\approx 6.6\%$. 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, 

![Graph](figure5.png)

**Figure 5.** Radius of gyration dependence on number of deposits for small and large aggregates.
without cut off, are, for small aggregates, \( D = 1.84 \pm 0.07 \),
and for large aggregates, \( D = 1.80 \pm 0.05 \).

**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(R0) 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 \( N = 10, \ldots, 90 \). 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
30% and 40%, (which are projections of the deposition distribution onto the XY plane, for $P(X) \geq .30$ 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

Pixels displayed represent sites with deposition probability greater than or equal to the indicated percentage.

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

![Image: Cumulative mass distribution in Y for the 30 large aggregates.](image-url)

Figure 17. Cumulative mass distribution in Y for the 30 large aggregates.
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 formulation for the fractal dimension is, \( (\text{perimeter})^{1/p} \propto (\text{area})^{1/2} \), 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 \( D_c = 1.67 \pm 0.01 \).

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, $D_{R_g} = 1.75 \pm 0.08$.

The correlation function results using 'windows' of 1.5 to 32.5 lattice spacings of 1.67 ± 0.01 are in agreement with the accepted results of 1.68 ± 0.05, as reported by Meakin (1983b), where 'windows' of 5 to 50 lattice spacings were utilized. The radius of gyration results of 1.75 ± 0.08 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


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 valid 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 
Hide 
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 between title and data sections.
Text 408,16,16"Simulation of Diffusion--'
Text 408,32,16"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,"Radius growth radius:"
Text 408,348,"Angle of radius:"
Text 408,364,"Data on Last Dancer"
Text 408,380,"R: 0;"
Text 408,396,"Number of jumps:"

Else
' Else if a filename was selected for a seed, load the 
Load A$,Xbios(2)
' screen portion into the screen memory and the 
Load Left(A$,Instr(A$,"."))=ARP',Lpeek(Arrstr(Order%())
' array portion into the previously allocated array.
Endif

Jumps%=1
' Jumps% 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 deposition.
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 deposition 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 XX, YY  
' and draws the new one at the new coordinates.
End If
RdZ=int(Sqr((XZ-200)^2+(YZ-200)^2))
' Calculates the distance from the center of the deposit.
If RdZ>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 XX, YY  
' and erases it from the screen.
End If
If RdZ>Order%(1)+5 Then
' If outside deposition zone, scales the jumping distance: larger jumps will economize run time.
Jump%=2*int(1.442695*log((RdZ-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.
End If
Until Stuck
' Repeats dancing with this dancer until it's stuck (deposited or killed).
Exit If 20*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.
XZ=Random(720)
' Radial location in half degrees, 0 to 719.
YZ=200+int((Order%(1)+5)*cos(XZ*pi/360))
' Generating circle is Rmax+5, so y=Rcos(theta) and
XZ=200+int((Order%(1)+5)*sin(XZ*pi/360))
' theta=(halfdegrees x pi)/360.
Plot XZ, YZ
' Puts the new dancer on the screen.
Return
Procedure Up
Sub Y1, JumpX
' Jump up, so y coordinate is decremented by the distance to jump.
If Y1 < 0 Then
  ' If jump is off the screen, wraps around to the other edge,
  ' (never satisfied with killing circle present; dancer dies first).
  Add Y1, 400
Endif
Return
Procedure Down
Add Y1, JumpX
' Likewise, only jump is downward (increasing y coordinate).
If Y1 > 399 Then
  Sub Y1, 400
Endif
Return
Procedure Left
Sub X1, JumpX
' As above, only decrease x.
If X1 < 0 Then
  Add X1, 400
Endif
Return
Procedure Right
Add X1, JumpX
If X1 > 399 Then
  Sub X1, 400
Endif
Return
Procedure Check
' Checks to see if deposition conditions are satisfied. If they are then, stick, Stuck=True.
If Not Point(X1, Y1) Then
  ' If the point jumped to is already occupied, then collision is detected
  Stuck=True
  ' and stick at previous coordinates (Xold1, Yold1).
  Inc Order1(0)
  ' Records the number of deposits as being one greater.
  Order1(Order1(0))=1024*Yold1+Xold1
  ' Encodes and saves the coordinates of the deposited particle.
  Print At(12, 20): Using "####", Order1(0)-1;
  ' Displays the position
  Ral=Sqr((Xold1-200)^2+(Yold1-200)^2)
  Print At(55, 24): Using "####",Ral;
  ' and the radius of the deposit. Then calculates the angle from the center.
  Angle1=Atn((Yold1-200)/(Xold1-200)+0.011)*57.3
  Theta1=Angle1
  ' This calculates the true angle from the arctan function, which gives
  ' angles from -90 to +90 degrees, instead of 0 to 359 degrees.
  If Angle1<0 Then
    Theta1=360+Angle1
  Endif
  If Xold1<200 Then
    Theta1=180+Angle1
  Endif
Endif
If Rax>OrderX(1) Then
  ' If this is a maximum radius deposit, then
  OrderX(1)=Rax
  ' updates Rmax and
  MangleX=ThetaX
  ' reports the angle of the maximum radius of the deposit.
Endif
' Prints it all out...
Print At(75,21);Using "***",OrderX(1);
Print At(77,22);Using "***",MangleX;
Print At(63,24);Using "***",ThetaX;
Print At(69,25);Using "###%%",NjumpsX;
' Makes a beep to indicate deposition.
Sound 1,15,18,1
Sound 1,0
NjumpsX=0
' Resets Njumps for the new dancer which will be generated. It's here
' so NjumpsX 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",X1
  ' 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(XoldX,YoldX)=0 Then
    ' If there's a dancer on the screen at an old coordinate
    Plot XoldX,YoldX
    ' Erases it so that it doesn't appear in the SCR file.
  Endif
  If Point(X,Y)=0 Then

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 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, \((2R_{\text{max}} \text{ vs. } 3R_{\text{max}})\), 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:"\nXX=0
Repeat
If (XX And 7)=7
Print At(23,4);Using "X=##",XX
Endif
YY=0
Repeat
A=Sqr((50-XX)^2+(50-YY)^2)
Pythagoras(XX,YY)=A
Pythagoras(YY,XX)=A
Pythagoras(100-XX,YY)=A
Pythagoras(YY,100-XX)=A
Pythagoras(XX,100-YY)=A
Pythagoras(100-XX,100-YY)=A
Pythagoras(100-YY,XX)=A
Pythagoras(100-YY,100-XX)=A
Inc YY
Until YY>XX
Inc XX
Until XX>100
Print "Reading pixel integration table"
YY=0
Repeat
XX=YY
Repeat
Read Include(XX,YY)
Let Include(YY,XX)=Include(XX,YY)
Inc XX
Until XX>32
Inc YY
Until YY>32
Noxx=1
Radius%=2\*(Limit%-1)
Repeat
  Power(Ndx%,1)=Radius%+0.5
  Inc Ndx%
  Div Radius%,2
Until Ndx%>Limit%
Do
  Cls
  Shows
  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
  Defill 1,1
  Ptrvertpos%=mouse.
  If Frac(Ptrvertpos%/32)<0.5 Then
    Gosub Inbox(Ptrvertpos%)
  Else
    In%=0
  Endif
  Do
    Repeat
      Ptrvertpos%=mouse.
      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%=mouse.
      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$="" Or Switch%
      If Switch%=0 Then
        Key$=Str%(Switch%)
      Endif
      Switch%=0
  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$) GoTo Auto, Existingfile, Single, Help, Exit
InX = 0
SwitchZ = 0
Loop
End
Procedure Inbox(Ht%)
Ht% = 32%(Ht% \ div 32)
If Ht% > 16 And Ht% < 192 Then
  Pbox -1, Ht%, 500, Ht% + 16
  In% = Ptrvertpos% \ 32
EndIf
Return
Procedure Outbox(Ht%)
Ht% = 32% \ In%
Pbox -1, Ht%, 500, Ht% + 16
In% = 0
Return
Procedure Exit
Edit
Return
Procedure Help
Cls
Print "This program can run in automatic mode. The requirements are that"
Print "it must be given a disk with a series of .ARR files with their"
Print "associated .SCR files. There can be no other .ARR files on the disk."
Print "If there are no .ARR files in the current disk or directory, a bus"
Print "error (two bombs) will result."
Print "To use the pre-existing directory mode (eg. to do only some of"
Print "the .ARR files on a disk), create a text file named CORELATE.DIR,"
Print "containing the filenames of the .ARR files you wish to process."
Print "Each filename should appear on a single line in the file."
Print "In both these cases, the results go into a file called CORELATE.DAT"
Print "in a tabular form, with the filename at the top, followed by lines"
Print "with three numbers separated by commas. These represent R, Mdisk(R),"
Print "and Msquare(R) for each R processed (Mdisk is the average pixel"
Print "density in a disk of radius R). The slopes of the best-fit power"
Print "curves for each technique are printed on the next two lines. These"
Print "slopes are the fractal dimensions as determined by the two-point"
Print "correlation function over disks and squares respectively. The total"
Print "number of deposits and the number of pixel excluded to eliminate edge"
Print "effects are printed on the last two lines."
Print "The Single File mode allows you to process a single file on the"
Print "disk, which can be entered from a Fileselect box. The results do not"
Print "go into a file, but are just printed on the screen."
Print "Hit any key to continue"
Repeat
Until Inkey$<>**
Return
Procedure Single
Gosub Loader
If File$<>** Then
  Time=Timer
  Gosub Process(File$)
  Cls
  Gosub Secs_to_his((Timer-Time)/200)
  Print "Running time:";Has$;
  Ndx%=1
  Repeat
    Power(Ndx%,2)=Power(Ndx%,2)/(Power(0,1))
    Power(Ndx%,3)=Power(Ndx%,3)/(Power(0,1))
    Print Power(Ndx%,1);",";Power(Ndx%,2);",";Power(Ndx%,3)
    Inc Ndx%
  Until Ndx%>Limit%
  Power(0,0)=Limit%
  Gosub Power
  Print "Fractal Dimension(disk)=";Sloped
  Print "Fractal Dimension(square)=";Slopes
  Print "Total Number of Deposits=";Order%(0)-1
  Print "Number of excluded pixels=";Power(0,2)
  Print "Hit any key to continue"
  Repeat
    Until Inkey$<>**
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
  Endf
  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
    Endf
    Gosub Load(File$)
    Time=Timer
    Gosub Process(File$)
    Gosub Secs_to_his((Timer-Time)/200)
Print #1,File$
Print #1,"Running time:";Hms$
NdXI=1
Repeat
    Power(NdxI,2)=Power(NdxI,2)/(Power(0,1))
    Power(NdxI,3)=Power(NdxI,3)/(Power(0,1))
    Print #1,Power(NdxI,1);",";Power(NdxI,2);",";Power(NdxI,3)
    Inc NdXI
Until NdXI>LimitX
Power(0,1)=LimitX
Gosub Power
Print #1,"Fractal Dimension(disk)=";Sloped
Print #1,"Fractal dimension(square)=";Slopes
Print #1,"Total Number of Deposits=";OrderX(0)-1
Print #1,"Number of excluded pixels=";Power(0,2)
Close #1
Until Eof($8)
Escape:
Close
Return
Procedure Process(File$)
    Defill 0,1
    Pbox 401,0,639,399
    Defill 1,1
    RwindomaxX=Int(Power(1,1))
    RdepositchmaxX=OrderX(1)
    NdXI=1
    Power(0,1)=0
    Power(0,2)=0
    Repeat
        Power(NdxI,2)=0
        Power(NdxI,3)=0
        Inc NdXI
Until NdXI>LimitX
Print At(53,3);"File:":"File$
Print At(53,5);"N= 0"
Print At(53,7);"Using "Out of 0000 total deposits",OrderX(0)-1
Print At(53,9);"Excluded pixels= 0"
NX=2
Repeat
    XWl=OrderX(NX)\\1024
    YWL=OrderX(NX) And 1023
    If Abs(Sqr((XWL-200)^2+(YWL-200)^2))+RwindomaxX=RdepositchmaxX Then
        Inc Power(0,1)
    End If
    XWl=XWl-RwindomaxX
    Repeat
        Ywl=YWL-RwindomaxX
        Repeat
            If Point(XWl,YWl) Then
                Xpi=Pythagoras(XWl+50,YWl+50)
                WdxI=1
                Repeat
                    ...
Exit If Abs(XW-XZ)>Power(NdxZ,1) Or Abs(YW-YZ)>Power(NdxZ,1)
Inc Power(NdxZ,3)
Exit If Rpix>Power(NdxZ,1)-0.70710678119
If Rpix<Power(NdxZ,1)-0.70710678119 Then
  Inc Power(NdxZ,2)
Else
  Corner=Sign(Power(NdxZ,1)-Rpix)*0.5
  Rcnr=(Abs(Abs(XY)-Abs(XW))+Corner)^2
  Add Rcnr, (Abs(Abs(YZ)-Abs(YW))+Corner)^2
  Rcnr=Sign(Rcnr)
  If Power(NdxZ,1)<Min(Rpix,Rcnr) And Power(NdxZ,1)<Max(Rpix,Rcnr)
    Add Power(NdxZ,2), Include(Abs(XW-XZ),Abs(YW-YZ))
  Else
    If Power(NdxZ,1)<Rpix Then
      Inc Power(NdxZ,2)
    Endif
  Endif
End If
Inc NdxZ
Until NdxZ>LimitZ
End If
Inc YZ
Until YZYW=WindowmaxZ
Inc YZ
Until YZYW=WindowmaxZ
Else
  Inc Power(0,2)
  Print At(70,9):Using "###",Power(0.2)
Endif
Print At(53,5):Using "N=####",NZ-1
Inc NZ
Option "U1"
Until NZ<OrderZ
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$)
Hide
Arrayfill OrderZ(),0
load File$,Ueek(Arrptr(OrderZ())))
Gosub ParseFilename(File$)
load Pathname$:"\$Left$(File$,Instr(File$,"."))$"+SCR$1bios(2)
Return
Procedure ParseFilename(Fn$)
Local FirstZ,LastZ
Pathname$=Left$(Fn$,Instr(Fn$,"."))
FirstZ=Instr(Fn$,"\")
For IX=Len(Fn$) Downto 1
  If Mid$(Fn$,IX,1)="\n"
    Last%=IX
  End:
End If
Next IX
Pathname$=Pathname$+Mid$(Fn$,First%,Last%-First%)
File$=Mid$(Fn$,Last%+1)
Return
Procedure Open_file_for_output_or_append(File$,Chan$
  If Not Exist(File$)
    Open "O",#Chan$,File$
  Else
    Open "A",#Chan$,File$
  End:
Return
Procedure Power
Local IX,NX,Sumofx,Sumofy,Sumofz,Sumofproducts,Sumofprod2,Sumofsquares
NX=Power(0,0)
Sumofx=0
Sumofy=0
Sumofz=0
Sumofproducts=0
Sumofprod2=0
Sumofsquares=0
For IX=1 To NX
  Add Sumofx,Log(Power(IX,1))
  Add Sumofy,Log(Power(IX,2))
  Add Sumofz,Log(Power(IX,3))
  Add Sumofproducts,(Log(Power(IX,1)))*(Log(Power(IX,2)))
  Add Sumofprod2,(Log(Power(IX,1)))*(Log(Power(IX,3)))
  Add Sumofsquares,(Log(Power(IX,1)))^2
Next IX
Sloped=(NX*Sumofproducts-Sumofx*Sumofy)/(NX*Sumofsquares-Sumofx^2)
Slopes=(NX*Sumofprod2-Sumofx*Sumofz)/(NX*Sumofsquares-Sumofy^2)
Return
Procedure Secs_to_hms(Secs)
Local H,M,S
Has=""
H=(Secs\3600)
M=(Secs Mod 3600)\60
S=(Secs Mod 3600) Mod 60
If H>0 Then
  Has$=Str$(H)++ hours, "
End If
Has$=Has$+Str$(M)++ minutes, "+Str$(S)++ seconds"
Return
Data 1,.97173982736,.98323187634,.99072351790,0,0,0,.99509549182
Data 0,0,0,0,0,0,.99747439951,0,0,0,0,0,0,0,0,0,0,0,0,1,.99871790316
Data .54540604028,.76932502669,1,.87746746419,0,0,1,.93596316353
Data 0,0,0,0,0,0,.9919256448,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,.9983278216
Data .1365559153,1,.51818108335,0,0,1,.75601286272

Radius of Gyration Program
Version=1.7
Revdate="29 Oct 88"
Dim OrderZ(30000)
Dim Radii(11,400)
Dim Pythagoras(100,100)
Do
Cls
Show
Print "Automatic Radius of Gyration Calculator, version="Version":"Revdate="Revdate
Print "Choose Mode of Operation: Type number or click on selection."
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"
Graph mode 3
Deffill 1.1
Ptrvertpos2=Mousey
If Frac(Ptrvertpos%/32)<0.5 Then
Gosub Inbox(Ptrvertpos%)
Else
lnZ=0
Endif
Do
Repeat
Ptrvertpos2=Mousey
If (lnZ>0) And (Frac(Ptrvertpos%/32)>0.5) Then
Gosub Outbox(Ptrvertpos%)
Endif
If (lnZ=0) And (Frac(Ptrvertpos%/32)<0.5) Then
Gosub Inbox(Ptrvertpos%)
Endif
SwitchZ=Mousek
If SwitchZ>0 Then
if lnZ>0 Then
SwitchZ=(Ptrvertpos%/32)
Else
SwitchZ=0
Sound 1,15,6,7,5
Sound 1,0
Endif
Endif
Key=inkey$
Until Key<"" Or SwitchZ
If SwitchZ Then
Key=Str$(SwitchZ)
Endif
Exit If Val(Key$)>0 And Val(Key$)<7
Sound 1,15,6,7,5
Sound 1,0
Loop
Cls
Graph mode 1
On Val(Key$) Gosub Auto,Existingfile,Single,Help,Exit,Special
InZ=0
SwitchZ=0
Loop
End

Procedure Inbox(Ht%)
    Ht%=32*(Ht%\32)
    If Ht%>16 And Ht%<24 Then
        Pbox -1,Ht%,500,Ht%+16
        InI=Ptrvertpos%\32
    Endif
    Return

Procedure Outbox(Ht%)
    Ht%=32*InI
    Pbox -1,Ht%,500,Ht%+16
    InI=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 File select 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_his((Time-Timer)/200)
    Print "Running time:" H1s$
    Print "Hit any key to continue";
    Repeat
        Inkey$=""
    Until Inkey$<>"
    Cls
    Gosub Drawaxes(100,300,0,450,250,0,40,30)
For II=1 To Radii(0,0)-1
    DeovarX=Log(Radii(0,II))*.40+100
    IndvarX=Log(Radii(1,II))*.30
    Dv2X=Log(Radii(0,II+1))*.40+100
    Iv2X=Log(Radii(1,II+1))*.30
    Draw DeovarX,300-IndvarX To Dv2X,300-Iv2X
Next X
Repeat
   Until Inkey$<*>"**
Endif
Return
Procedure Auto
   Dir *.*, ARR To * . YRATE. DIR*
   Gosub Existing file
Return
Procedure Existing file
   Open * .", 00 , * . YRATE. DIR*
   If Eof(00) Then
      Goto Escape
   Endif
   Repeat
      Input #0, File$
      If File$="" Then
         Print " Directory file is empty: either no .ARR files on current directory, or you forgot to fill the .DIR file."
         Print " Hit any key to continue."
         Repeat
            Until Inkey$<*>"
         Goto Escape
      Endif
      Gosub Load(File$)
      Gosub Process(File$)
   Until Eof(#0)
   Escape:
   Close
Return
Procedure Process (File$)
   Cis
   Line 400,0,400,399
   Print At(53,3): "File:" + File$
   Print At(53,5): "N= 0"
   Print At(53,7): "Using "Out of total deposits" , Order%=0"
   Sum=0
   Radii(0,0)=1
   AveX=0
   AveY=0
   Dest%=Int(((Radii(0,0)+10)^2.4)/82)
   NZ=2
   Repeat
      Xpixel%=Order%(NZ)\1024
      Ypixel%=Order%(NZ) And 1023
      AveX=(AveX%(NZ-2)+Xpixel%)/(NZ-1)
      AveY=(AveY%(NZ-2)+Ypixel%)/(NZ-1)
      Add Sum, (AveX-Xpixel%)^2+(AveY-Ypixel%)^2
      Plot Xpixel%, Ypixel%
      If (NZ-1)=Dest% Then
         Radii(1,Radii(0,0))=Sqr(Sum)/(NZ-1)
         Radii(0,Radii(0,0))=Dest%
Inc Radii(0,0)
Dest=RInt(((Radii(0,0)*10)^(2.4))/(82))
Endif
If (NZ-1) Mod 100=0 Then
Print At(53,5);Using "N=####".NZ-1
Endif
Inc NZ
Option "U1"
Until NZ>Order(0)
Dec Radii(0,0)
Sosub Parsefilename(File$)
File$=Pathname$+"\".Left$(File$.Instr(File$."."))"+GYA
Bsave File$.Lpeek(Arrptr(Radii())))..(Radii(0,0)+1)*12+B
Return
Procedure Loader
Print At(1,3);"Select array:"
Fileselect "\.".ARR","SEED.ARR",File$
If File"=*" Then
Sosub Load(File$)
Endif
Return
Procedure Load(File$)
Hiden
Arrayfill Order%(0,0)
Arrayfill Radii(0,0)
Bsave File$.Lpeek(Arrptr(Order%)))
Return
Procedure Parsefilename(Fn$)
Local First%,Last%,IX
Pathname=Left$(Fn$.Instr(Fn$."."))
First%=Instr(Fn$,"\"")
For IX=Len(Fn$) Downto 1
If Mid$(Fn$,IX,1)="\"
Last%=IX
Endif
Exit If Mid$(Fn$,IX,1)="\"
Next IX
Pathname=Pathname+Mid$(Fn$,First%,Last%-First%)
Files=Mid$(Fn$,Last%+1)
Return
Procedure Power
Local I%,NZ,Sumofx,Sumofy,Sumofz,Sumofproducts,Sumofprod2,Sumofsquares
NZ=Power(0,0)
Sumofx=0
Sumofy=0
Sumofz=0
Sumofproducts=0
Sumofprod2=0
Sumofsquares=0
For I%=1 To NZ
Add Sumofx,Log(Power(I%,1))
Add Sumofy,Log(Power(I%,2))
Add $\text{SumOfLog} = \log(\text{Power}(1, 2))$
Add $\text{SumOfLogProducts} = (\log(\text{Power}(1, 2))) \times (\log(\text{Power}(1, 3)))$
Add $\text{SumOfLog2} = (\log(\text{Power}(1, 1))) \times (\log(\text{Power}(1, 2)))$
Add $\text{SumOfSquares} = (\log(\text{Power}(1, 1)))^2$

Next N
Slopes = $\frac{\text{N} \times \text{SumOfLogProducts} - \text{SumOfLog} \times \text{SumOfLog2}}{\text{N} \times \text{SumOfSquares} - \text{SumOfLog2}^2}$
Slopes = $\frac{\text{N} \times \text{SumOfLog2} - \text{SumOfLog} \times \text{SumOfSquares}}{\text{N} \times \text{SumOfLog2}^2 - \text{SumOfLog2}^2}$

Return

Procedure Secs_to_hrds(Secs)
Local H, M, S
H = Secs \ 3600
M = (Secs \ Mod 3600) \ 60
S = (Secs \ Mod 3600) \ Mod 60
If H > 0 Then
   H =Str(S) + " hours, "
End if
H = H + Str(M) + " minutes, " + Str(S) + " seconds"
Return

Defline 1, 1, 1, 1
If LendX = 0 Then
   Defline 1, 1, 0, 1
End if
If LendY = 0 Then
   Defline 1, 1, 1, 0
End if
Draw OriginX - LendX, OriginY to OriginX + LendX, OriginY
Defline 1, 1, 1, 1
If UpenDy = 0 Then
   Defline 1, 1, 0, 1
End if
If LoendY = 0 Then
   Defline 1, 1, 1, 0
End if
Draw OriginX, OriginY + UpendY to OriginX, OriginY + LoendY
Local AX, LengthX
LengthX = 10
Defline 1, 1, 0, 0
If Hashx <> 0 Then
   For AX = OriginX to OriginX - LendX Step -Hashx
      Draw AX, OriginY - LengthX to AX, OriginY + LengthX
   Next AX
   For AX = OriginX to OriginX + LendX Step Hashx
      Draw AX, OriginY - LengthX to AX, OriginY + LengthX
   Next AX
End if
If Hashy <> 0 Then
   For AX = OriginX to OriginX - LendY Step -Hashy
      Draw OriginX + LengthX, AX to OriginX + LengthX, AX
   Next AX
   For AX = OriginX to OriginX + LoendY Step Hashy
      Draw OriginX + LengthX, AX to OriginX + LengthX, AX
   Next AX
End if
Draw Originx1+Length1,A1 To Originx1-Length1,A1
Next A1
Endif
Return
Procedure Centerofmass(P.array,LatX)
NZ=0
Swap $P.array, AvearrayZ()
Avex=0
Avey=0
Do
Inc NZ
Exit If NZ>LatX
Avex=(Avex*(NZ-1)+((AvearrayZ(NZ+1))
Avey=(Avey*(NZ-1)+(AvearrayZ(NZ+1) And 1023))/NZ
Loop
Swap $P.array, AvearrayZ()
Return
Procedure Special
Gosub Loader
If File$<>"** Then
Do
Print "Input number of deposits to include in Rg (up to ";OrderX(0)-1"," 0 to quit)";
Input Limit7.
Exit If Limit=0
Gosub Centerofmass(OrderZ().Limit7)
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$)
Print "Ln(# of deposits) ="Log(Limit7)
Print "Ln(Rg) ="Log(Sum/(Limit7))
Loop
Endif
Return
Procedure Specialprocess(File$)
Sum=0
NZ=2
Repeat
XpixelX=OrderX(NZ)
YpixelX=OrderX(NZ) And 1023
Add Sum, (Avex-XpixelX)^2+(Avey-YpixelX)^2
Inc NZ
Option "UI"
Until NZ>Limit7
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", "Involute", "Zonal growth",
Menu Menu$(I)
On Menu Gosub Handle_it_for_me
Print At(1, 3); "Menu Program Version"; "Version: ;"; "Revdate: 
Do
    On Menu
    Loop
End
Procedure Handle_it_for_me
Cls
If Menu(0) = 1 Then
    Gosub Give_info
Else
    On Menu(0) = 11 Gosub Quit, Dummy, Dummy, Invert, Disp, Prscreen, Strip, Viewarr
    If Menu(0) > 19 Then
        On Menu(0) = 19 Gosub Dummy, Dummy, Loader, Dummy, Animate, Involute, Zonal
    Endif
Endif
Menu Menu$(I)
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 $ then
    Hide
    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
    Show
Endif
Return

Procedure Disp
Print At(1,3);"File to display:"
Fileselect "\%.SCR","SEED.SCR",A$
If A$<>"" Then
  Hide
  Bload A$,Xbios(2)
  Repeat
  Until Inkey$<>"
  Show
Endif
Return

Procedure Prscreen
Print At(1,3);"File to print:"
Fileselect "\%.SCR","SEED.SCR",A$
If A$<>"" Then
  Hide
  Bload A$,Xbios(2)
  Sdpoke 1262,0
  Show
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$<>"" Then
    Hide
    Bload A$,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
    Show
  Endif
Endif
Return

Procedure Invert_window(XZ,YZ,XlZ,YlZ)
  Color 0
  Graphmode 3
  For AZ=YZ To YlZ
    For BZ=XZ To XlZ
      Plot BZ,AZ
    Next BZ
  Next AZ
Return

Procedure Viewarr
Begin:
Local StartI, LineI, LenI, $ArrI, OldI, OptrI, NptrI, Changed$
Changed! = False
OptrI = Lpeek(Arrptr(OrderI))
NptrI = Lpeek(Arrptr(OrderI))
Lpoke OptrI, 30001
Lpoke NptrI, 30001
Arrayfill OrderI(), 0
Arrayfill OrderI(), 0
Print At(1, 3); "Array file to view:"
Fileselect "$\%.AR?", "SEED.ARR", Arr$
If Arr$<> "" Then
  Load Arr$, OptrI
  OldI = True
LenI = Order(0)
If Order(1)<> Int(Order(1)) Then
  Bmove OptrI, NptrI, B
If Order(0)/30001 Then
  Bmove OptrI, NptrI, Order(0)\#4+B
Else
  Bmove OptrI, NptrI, 120008
Endif
OldI = False
Endif
If Not OldI Then
  LenI = Order(0)
Endif
If Instr("23456709", Right$(Arr$, 1))=0 Then
  GlenI = LenI
  SegI = 1
  SegmentI = 0
Else
  Open "R", 1, Left$(Arr$, Len$(Arr$)-1)+"R", $
  Field 1, 4 As Buf$
  Get 1, 2
  GlenI = Cvl(Buf$)
  Close 1
  SegI = Val(Right$(Arr$, 1))
  SegmentI = 299994(SegI-1)
Endif
Sosub Viewarrscreen
Do
  For LineI = StartI To StartI+23
    If OldI Then
      If LineI = 0 Then
        Print At(1, 2); "$ = " Order(LineI)-1; "$
      Endif
      If LineI = 1 Then
        Print At(1, 3); "$max = " Order(LineI); "$
      Endif
      If LineI > 1 And LineI < 30001 Then
        Print At(1, LineI-StartI+2); Using "$\#####", LineI-1;
Print At(9,Line%)=Start%+2);Using *0000, Order(Line%)\262144;
Print At(10,Line%+Start%+2);Using *000, Order(Line%) Mod 262144\512;
Print At(12,Line%+Start%+2);Using *000, Order(Line%) Mod 512;
Endif
Else
If Line%=0 Then
  Print At(1,2); * N = "Order%Line%-%1";
Endif
If Line%=1 Then
  Print At(1,3); * Rmax = "Order\Line%";
Endif
If Line%>1 And Line%<3000 Then
  Print At(1,Line%+Start%+2);Using *0000, Line%+Segment%-1;
  Print At(10,Line%+Start%+2);Using *000, Order\Line%\1024;
  Print At(12,Line%+Start%+2);Using *000, Order\Line% And 1023;
Endif
Endif
If Line%>30000 Then
  Print At(1,Line%+Start%+2); *
Endif
Next Line%
Repeat
  A$=Inkey$
  Let Mouse%=Mousek
  If Mouse%<>0 Then
    A$="E"
    Ptx%=Mousex
    Pty%=Mousey
  Endif
  Until A$="**"
  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$(1) Then
    Gosub Changename
  Endif
  If A$="N" Then
   Cls
    Goto Begin
  Endif
If A$=Chr$(13) Or A$=Chr$(32) Or A$=Chr$(0)+Chr$(80) Then
  Add Start%, 24
  If Start%>99999 Then
Inc Seg%
Gosub Get_new_seg
Start%=0
Endif
Endif
If A$=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%-29999
If Seg%>0 Then Seg%=29999
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):"Y";
    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 SegI=1 Then
    Arr$=Left$(Arr$,Len(Arr$)-1)+"R"
Else
    Arr$=Left$(Arr$,Len(Arr$)-1)+Str$(SegI)
Endif
Arrayfill Order%,0
Segment%=29999#(SegI-1)
Print At(12,3):"Loading segment"SegI:"... Please wait."
Bload Arr$,Nptr%
Len%=Order%(0)
Gosub Viewarrayscreen
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$""=";Lpeek(A%)
Lpokes AI,30001
Changed! = True
Endif
If Old! Then
Seedlocation[]=$Seedlocation\1024;^512+{Seedlocation Mod 1024}
If Order(2)<Seedlocation Then
  Print "I'm seeding the array"
  Bmove AI+16, AI+22, 64*{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 AI+12, AI+16, 48*{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$="**"
  Gosub Viewwarrscreen
Return
Procedure Convert
Local A$
Cls
If Not Old! Then
  Print "This file appears to be converted already!"
  Print "New format N=;Order(0)-1"; "Max=;Order(1)
  Input "Should I convert it anyway (Y or N)? ", A$
Else
  A$="Y"
Endif
If Asc(A$) And 223=89 Then
  Lpokes Nptr1, 30001
  Arrayfill Order(1), 0
  Print "M x 1000: ****
  For X=0 To Order(0)
    If X>1
OrderX(X)=Order(X) Mod 262144
OrderX(X)=(OrderX(X) And 261632) xor (OrderX(X) And 511)
Else
   OrderX(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 DestI,Idx%
Ptrv%=Ptrv%+1
If PtrxI<=151 And Ptry%>1 Then
   IdxX=StartI+PtryI-3
   If IdxX>0 Then
      Print At(55,20);''D to delete''Idx%
      Print At(55,21);''<TAB> to insert blank''
      Print At(55,22);''<ESC> aborts.''
      If PtryX>96 And PtryX<151 Then
         Print At(55,22);''or type number'';
         If PtryX<119 Then
            DestI=1024
            Box 95,PtryX%+16-17,120,Ptry%16
            Print ''for X''
            Gosub Getnum(13)
         Else
            DestI=1
            Box 127,PtryX%+16-17,152,Ptry%16
            Print ''for Y''
            Gosub Getnum(17)
         Endif
      Else
         DestI=0
         Box -1,PtryX%+16-17,152,Ptry%16
         Gosub Getnum(0)
      Endif
   Else
      If IdxX=-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
      Endif
   Endif
Endif
Else
    DestX=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(DestcolX)
    Local Accept$,Nu$,Done!
    Let Done'=False
    Nu$=""
    Accept$=Chr$(4)+Chr$(9)+Chr$(27)
    If DestcolI<>0 Then
        Accept$=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 Nu`
    Endif
    Until Done!
Return

Procedure Delentry
    If Order%(0)-IdxX-1>0 Then
        Above MptrX+(IdxX+5)##4,MptrX+(IdxX+2)##4,(Order%(0)-IdxX-1)##4
    Endif
    If Order%(0)-IdxX-1>=0 Then
        OrderX(OrderX(0))=0
        Dec OrderX(0)
        Changed'=True
    Else
        Sound 1,15,6,7,2
        Sound 0,0
Endif
Let Done!=True
Return
Procedure Insspace
If Order%1-Idxl>O Then
  Baove NptrX+(IdxX+2)%4,NptrX+(IdxX+1)%4,(Order%1-IdxX)\$4
  Order%1(IdxX+1)=0
  Inc Order%0
  Changed!=True
  Let Done!=True
Endif
Return
Procedure Esc
Let Done!=True
Return
Procedure Endnum
Order%1(IdxX+1)=Order%1(IdxX+1) And (Not (1023#$DestX))
Add Order%1(IdxX+1),Val(Num$)#DestX
If IdxX+1>Order%0 Then
  Order%0=IdxX+1
  Len%=Order%0
Endif
Changed!=True
Let Done!=True
Return
Procedure Delchar
If Num$<>(
  Num$=Left$(Num$,Len$(Num$)-1)
  Print At(Destcol%,IdxX-StartX+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%,IdxX-StartX+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%,XX
Pathname=Left$(Fn$,Instr$(Fn$,"\%")
First%=Instr$(Fn$,"\%")
For XX=Len$(Fn$) Downto 1
  If Mid$(Fn$,XX,1)=="\%"
    Last%=XX
Endif
Exit If Mid$(Fn$, X2, 1) = "\n"
Next X2
Pathname$=Pathname$+Mid$(Fn$, First2, Last2-First2)
File$=Mid$(Fn$, Last2+1)
Return
Procedure Loader
  Print At(1, 3); "Select array:"
  Fileselect \ARR", "SEED. ARR", File$
  If File$<> Then
    Load File$, Lpeek(Arrptr|Order|))
  Endif
Return
Procedure Save
  Local X1, F1, F2, Tlen
  Tlen=Min(LenX, 30000)
  A2=Notr
  F1=$4
  F2=8
  If Old! Then
    A2=Optr
    F1=6
    F2=10
  Endif
  Print At(32, 2); "Saving array to" Arr$
  Bsave Arr$, A2, Tlen$X F1+F2%
  Changed1=False
  Gosub Viewarrscreen
  Return
Procedure Changename
  Local Taps$
  Taps=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$=Taps$
  Gosub Save
  Return
Procedure Drawscrean
  If File$<> Then
    Gosub Loader
  Endif
 Cls
  Hidem
  Graphmode 3
  Color 1
  Line 400, 0, 400, 399
  Defext 0.16, 0.32
  Text 455, 45, " Animator"
Procedure Plot(Start%, Finish%, Direction%, Width%)
Local Wait!$, SS
XX = Start%
Repeat
If (Direction%>0 And XX<Finish%) Or (Direction%<0 And XX>Finish%) Then
  Plot Order(XX)\1024, Order(XX) And 1023
  If Width%>0 And XX>Width%+1 Then
    Plot Order(XX-Width%)\1024, Order(XX-Width%) And 1023
  Endif
  Add XX, Direction%
Endif
A$=Inkey$
If Wait! Then
  Repeat
    A$=Inkey$
  Until A$<>"
Endif
If A$<>"" Then
  If A$="# Then
    Plot Order(XX)\1024, Order(XX) And 1023
    If Width%>0 And XX>Width%+1 Then
      Plot Order(XX-Width%)\1024, Order(XX-Width%) And 1023
    Endif
    Swap Start%, Finish%
    Mul Direction%, -1
  Endif
  If A$="." 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$="/" Then
  Sget S$
  AI=1X
  Gosub Plot(2,XX,1,0)
  Sput S$
  XX=AI
  A$=""
Endif
Endif
Until A$=Chr$(0)+Chr$(97)
Return
Procedure Animate
  Gosub Drawscren
  Gosub Plot(2,Order%0,1,0)
  Setcolor 0,1
  Show
Return
Procedure Involute
  Gosub Drawscren
  Gosub Plot(Order%0,2,-1,0)
  Setcolor 0,1
  Show
Return
Procedure Zonal
  Cls
  Print At(10,12);
  Input "Enter number of pixels to display in deposition zone";Width%
  Gosub Drawscren
  Gosub Plot(2,Order%0,1,Width%)
  Setcolor 0,1
  Show
Return
' Core menu 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 ","FNB 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$()
On Menu Gosub Handle_it_for_me
Print At(1,3);"Correlation functions, Version"Version;","Revdate#
Do
On Menu
Loop
Procedure Handle_it_for_me
Cls
If Menu(0)=1 Then
Gosub Give_info
Else
On Menu(0)=11 Gosub Quit,D,D,Auto,Massxy,Massrt,D,D,Massxy,Massrt
If Menu(0)=21 Then
On Menu(0)=21 Gosub Dummy,D,Prep_for_new,Dummy,Viewdat,Plotya,Regression
Endif
Endif
Menu Menu$()
Print At(1,3);"Select function:" Return
Procedure Give_info
Return
Procedure Quit
Menu Kill
Edit
Return
Procedure Auto
Local A2,B%,C%,I%,J%
Gosub Loader
Cls
If File#<>0"" Then
Input "Input n:";N%
Print "Calculating Autocorrelation vectors"
For A%=2 To OrderX(O)-N%
If A% Mod 100=0 Then
Print At(1,6);"N=";A%
Endif
J%=OrderX(A%+N%) I%=OrderX(A%)
X%=I%+1024 Y%=I%+1024
Y%=I% And 1023 Y%=I% And 1023
OrderX(A%+1)=Int(Sqr((Y%-Y%)^2+(X%-X%)^2))
Next A% 
Cls
Order%[Order%(0)]=0
Order%[Order%(0)-1]=0
For AX=Order%(0)-1 To Order%(0)
    Order%(AX)=0
Next AX
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 AX=1 To 600
    Color 1
    Draw AX+20,380 To AX+20,380-Order%(AX+Begin%-1)
    Color 0
    Draw AX+20,379-Order%(AX+Begin%-1) To AX+20,0
Next AX
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(0Order%(0))
    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 XI=2 To Order%(0)
    If XI Mod 100=0 Then
        Print At(11,6);XI
    Endif
    Inc Results(0,Order%(XI)
    Inc Results(1,Order%(XI) And 1023)
Next XI
Gosub Set_up(320,240,220,220,200,0,100,180,1,1)
Gosub Dispxy
Endif
Return
Procedure Massxy
Local Coil%,Exp%,Freq%,Iter%,Sixbit%,Uncoil%,Xpixel%,Ypixel%
Gosub Loader
Cls
If File$<>"** Then
    Print At(1,5);"Calculating Center of Mass ... Please wait"
    Gosub Fcenterofmass(0Order%(0))
    Print "Center of mass at X=":AveX","Y=":AveY
Print At(1,5); "Calculating X and Y density functions"
Print "Processed 0 of 160000 Points"
For Iter%=1 To 32000
  Coil%= (Iter%-1) * 5
  Freq%= OrderZ(Iter%)
  If Coil% Mod 100=0 Then
    Print At(11,6); Coil%
  Endif
  If Freq%<>0 Then
    For SixbitX=0 To 4
      Exp%= 64 * SixbitX
      Freq%= OrderZ(Iter%) And (63*Exp%)
      If Freq%<>0 Then
        Div Freq%, Exp%
        Uncoil%= Coil% + SixbitX
        Xpixel%= Uncoil% \ 400
        Ypixel%= Uncoil% Mod 400
        Add Results(0, Xpixel%), Freq%
        Add Results(1, Ypixel%), Freq%
        Option "UI"
      Endif
    Next SixbitX
  Endif
  Next Iter%
Gosub Set_up(320, 240, 220, 220, 200, 0, 100, 180, 1, -20)
Gosub Dispxy
Endif
Return
Procedure Dispxy
Tester:
Cls
Graphmode 1
If Loaded! Then
  File$= 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 Cad_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%4)
Lbl$=Left$(Lbl$,Len(Lbl$)-1)+'Y'
Print At (AI,22):Lbl$
Print At (52,4); "Center Of Mass:"
Print At (52,5); Avey; ";" Avey
Gosub Cad_driver("")
Return
Procedure Massrt
Local X%,B%,R%,Th%,Rav,Rho%
Graphmode 1
Gosub Loader
Cls
If File$<>"" Then
Print At (1,5); "Calculating Center of Mass .... Please wait"
Gosub Centerofmass(#Order%)
Print "Center of mass at X=": Avex; "Y=": Avey
Print At (1,5); "Calculating R and Theta density functions"
Print "Processed 0 of "#Order%" Points"
For X%=2 To #Order%
If X% Mod 100=0 Then
Print At (11,6); X%
Endif
A%=Order%(X%)\1024
B%=Order%(X%) And 1023
R%=Int(Sqr((A%-Avex-200)^2+(B%-Avey-200)^2))
Th%=Trunc(Atn((B%+Avey-200)/(A%-Avex-200+0.00001))*57.3)
Add Th%,180
If A%-Avex-200<0 Then
Add Th%,180
Endif
If Th%>360 Then
Sub Th%,360
Endif
Inc Results(0,R%)
Inc Results(1,Th%)
Next X%
Gosub Set_up(100,250,0,220,220,0,50,100,1,1)
Gosub Disprt
Endif
Return
Procedure Fmassrt
Local Coil%,Exp%,Freq%,Iter%,Sixbit%,Uncoil%,Xpixel%,Ypixel%
Gosub Floader
Cls
If File$<>"" 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%=6*Sixbit%  
         Freq%=Order%(Iter%) And (63*Exp%) 
         If Freq%<>0 Then  
            Div Freq%,Exp%  
            Uncoil%=Coil%+Sixbit%  
            Xpixel%=Uncoil%*400  
            Ypixel%=Uncoil% Mod 400  
            Rx%=Int(Sqr((Xpixel%-Avex-200)^2+(Ypixel%-Avey-200)^2))  
            Thz=Trunc(Min((Ypixel%-Avey-200)/(Xpixel%-Avex-200+0.00001))*57.3)  
            Add Thz,180  
            If Xpixel%-Avex-200<0 Then  
               Add Thz,180  
            Endif  
            If Thz>=360 Then  
               Sub Thz,360  
            Endif  
            Add Results(0,Rx%),Freq%  
            Add Results(1,Thz),Freq%  
            Option "U1"  
         Endif  
      Next Sixbit%  
   Endif  
   Next Iter%  
Gosub Set_up(100,250,0,220,220,0,50,100,-20,-20)  
Gosub Dis Plt  
Endif  
Return  
Procedure DisPlt  
Cls  
If Loaded' Then  
   File$=Dat$  
Endif  
Gosub Parsefilename(File$)  
Testing:  
Sclx%=Std_graphZ(9)  
Std_graphZ(9)=1  
Gosub Axes(Std_graphZ())  
Gosub Label_hashes(Std_graphZ())  
Std_graphZ(0)=1+8  
Std_graphZ(3)=0  
Std_graphZ(4)=201  
Std_graphZ(5)=0
Gosub Plot(#Std_graph%1())
  Lbl1$="Deposit "+File$" Mass Distribution Function in R"
  AZ=#40-Len(Lbl1$)/2
  Print At(AZ,22);Lb1$
  Print At(52,4);"Center Of Mass:"
  Print At(52,5);Avex="Ave
  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 Cad_driver("r")
  Cls
  Gosub Set_up(220,200,180,180,180,180,150,150,Sc!,X!,Std_graph1.)
  Gosub Axes(#Std_graph%1())
  Gosub Label_hashes(#Std_graph%1())
  Std_graph1(0)=1+4+32 !1 color, 4 upper, 32 polar
  Std_graph1(4)=360
  Std_graph1(5)=0
  Gosub Plot(#Std_graph%1())
  Lbl1$=Left$(Lbl1$,Len(Lbl1$)-1)+"Theta"
  AZ=Len(Lbl1$)
  Print At(52,2);Left$(Lbl1$,AZ-35);
  Print At(52,3);Mid$(Lbl1$,AZ-34,17)
  Print At(52,4);Right$(Lbl1$,17)
  Print At(52,6);"Center Of Mass:"
  Print At(52,7);Avex="Ave
  Gosub Cad_driver("r")
  Let Loaded=True
  Len%=360
  Return

Procedure Cad_driver(Char$)
  Local Char$
  Char$=-''<*>""
  Do
    Gosub Fline(*S = Smoothing  M = Modes  E = Edit Screen  <ESC> aborts <CR> stores screen*)
    Repeat
      AS=Inkey$
      Until AS"**
    Exit If AS$=Chr$(27)
    If (Asc(AS$) And 95)=83 Then
      On Char% Gosub Label_ave
      Gosub Smoothing
      On Char% Gosub Label_ave
Endif
On -((Asc(A$) And $5)=77) Gosub Modes
On -((Asc(A$) And $5)=69) Gosub Ed
If $A$=Chr$(13) Then
  Gosub Paline(**)
  Sget Scr$
  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$(13)
Loop
Return
Procedure Label_ave
  Graphmode 3
  Deftext 1,0,0,6
  Text 100+Ave-3,250+20+10,Chr$
  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 $X1$,N1,Cum1,Nr1,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%0)
  If Not Split! Then
    For $X1=$(Std_graph%3) To (Std_graph%4)
      Cum1=0
      Nr1=0
      For N1=-Interval% To Interval%
        If $X1+N1>=0 And $X1+N1<=400 Then
          Add Cum1,Results(Abs(Upper!),$X1+N1)
          Inc Nr1
        Endif
        Next N1
      Next $X1
      Results(Abs(Upper!),$X1)=Int(Cum1/Nr1)
    Next $X1
  Endif

Else
End if
Add Std_graph(0.1
Gosub Plot(Std_graph())
Return

Procedure Ed
Local X,Y,K,A$
Gosub P1line("(Left Button) - Add Text  (Right Button) - Move Area  (ESC) exits")
Repeat
Mouse X,Y,K
On K Gosub Text,Move
If K Then
    Gosub P1line("(Left Button) - Add Text  (Right Button) - Move Area  (ESC) exits")
    K=0
End if
A$=Inkey$
Until (A$=Chr$(13)) Or (A$=Chr$(127))
Return

Procedure Prev_for_new
Let Loaded!:=False
Dat$=""
Return

Procedure Viewdat
Local Start%,Line%,A$,Old!,Optr%,Nptr%
Optr%=Lpeek(Arrptr(Results))
If Not Loaded! Then
    Gosub Dloader
End if
If Dat$<>"" Then
    Gosub Viewdat-screen
    Do
        For Line%=Start% To Start%+23
            If Line%>401 Then
                Print At(1,Line%-Start%+2);Using ",Line%+Base1;
                Print At(12,Line%-Start%+2);Using *,Results(0,Line%+Base1);
                Print At(128,Line%-Start%+2);Using *,Results(1,Line%+Base1);
            End if
            If Line%>400 Then
                Print At(1,Line%-Start%+1);"
            End if
        Next Line%
    End do
    Repeat
        A$=Inkey$
    Until A$=""
    If A$=Chr$(13) Or A$=Chr$(12) Or A$=Chr$(0)+Chr$(80) Then
        Add Start%,25
        If Start%>375 Then
            Start%=375
        End if
    End if
    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$(77) Then
  Start%=(Len%)/25
Endif
Exit If A$=Chr$(27)
Loop
Endif
Cls
Return
Procedure ViewDateScreen
Cls
Box 384,75,600,185
Text 392,93,"Up arrow - Page up"
Text 392,109,"Down arrow - Page down"
Text 392,125,"(Space), <CR> - same as down arrow"
Text 392,141,"(Home) - Top of array"
Text 392,157,"Left arrow - Last page of array"
Text 392,173,"(Esc) - Main menu"
Start%=0
Print At(1,1);"N of";Len%
Print At(12,1);Zero$;
Print At(28,1);One$;
Print At(49,1);Dat$;
Return
Procedure Plotya
If Not Loaded! Then
  Gosub Dloader
Endif
If Dat$<>** Then
  If Typ$="XY" Then
    Gosub Dispxy
  Else
    If Typ$="RY" Then
      Gosub DispRy
    Else
      Cls
      Gosub Axes(#Std_graphX())
      Gosub Label_hashes(#Std_graphX())
      Gosub Plot(#Std_graphX())
      Gosub Regression
      Gosub Cad_driver(**)
    Endif
  Endif
Endif
Endif
Return
Procedure Regression
Local Cutoff, Cutoff%
If Not Loaded! Then
Gosub Dloader
Endif
If Dat$<>"** 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((0.41566566667)^(0.4 Cutoff))
Endif
For NX=Cutoff% To Len%
Inc Power(0,0)
Power(0,Power(0,0))=Results(0,NX)
Power(1,Power(0,0))=Results(1,NX)
Next NX
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$<>"** Then
Arrayfill Results(),0
Arrayfill Order%(1,0)
Bload File$,Lpeek(Arrptr!Order%(1))
Endif
Return
Procedure Floader
Print At(1,3);"Select array: "
Fileselect "*.*.FHS","LONGLIST.FHS",File$
If File$<>"** Then
Arrayfill Results(),0
Arrayfill Order%(1,0)
Bload File$,Lpeek(Arrptr(Order%(1)))
Endif
Return
Procedure Dloader
Local Tp$,Tp%
Do
Print At(1,3);"Select array: "
Fileselect "*.*.YA","",Dat$
If Dat$<>"** Then
Tp$=File$
Tp$=Pathname$
Gosub Parsefilename(Dat$)
Typ$=Mid$(File$,Instr(File$,".")+1)
Datpath$=Pathname$
Pathname$=Tp$
End
File# = Tf$

Endif

Exit If Instr("PYA6YAXYARYA", Typ$) = 0 Or (Dat$ ="")
Print At(1, 11);Dat$" is an unknown type of data file. Please "
Print "enter file with .PYA, .LYA, .XYA, or .RYA extension."

Loop

If Dat$ <>"" Then
  Arrayfill Results(1, 0)
  Load Dat$, Lpeek(Arrptr(Results(1)))
  Let Loaded = True
  If Instr("FG", Left$(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, 420, 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 M"
      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
      LenX = 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 Tya<"RYA" Then
Len1=360
Len0=201
Len2=360
Zeros="Density in R"
Let Ones="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$)
Local First%, Last%, XX
Pathname=Left$(Fn$, Instr(Fn$, ":"))
First%=Instr(Fn$, ":")
For XX=Len(Fn$) Downto 1
If Mid$(Fn$, XX, 1)
Last%=XX
Endif
Exit If Mid$(Fn$, XX, 1)
Next XX
Pathname=Pathname+Mid$(Fn$, First%, Last%-First%)
File$=Mid$(Fn$, Last%+1)
Return
Procedure Set_up(Orx%, Ory%, Lenx%, Rendx%, Tendy%, Bendy%, Hashx%, Hashy%, Scx%, Scy%)
Std_graph(1)=Orx%
Std_graph(2)=Ory%
Std_graph(6)=Lenx%+Rendx%
Std_graph(7)=Tendy%+Bendy%
Std_graph(8)=Hashx%+Hashy%
Std_graph(9)=Scx%
Std_graph(10)=Scy%
Return
Procedure Axes(P.array)
Local Orx%, Ory%, Lenx%, Rendx%, Tendy%, Bendy%, Hashx%, Hashy%
Local AX, Length%
Swap P.array, Array%(1)
Orx%=Array%(1)
Ory%=Array%(2)
Lenx%=Array%(6)+65536
Rendx%=Array%(6) And 65535
Tendy%=Array%(7)+65536
Bendy%=Array%(7) And 65535
Hashx%=Array%(8)+65536
Hashy%=Array%(8) And 65535
Gosub Drawaxes(*Array%(1))
Length%=10
Defline 1,1,0,0
If Hashxl<>0 Then
   For A7=Orxl To Orx%-Lendxl Step -Hashxl
      Draw A7,Oryl-Lengthl To A7,Oryl+Lengthl
   Next A7
   For A7=Orx% To Orx%+Rendx% Step Hashx%
      Draw A7,Oryl-Lengthl To A7,Oryl+Lengthl
   Next A7
Endif
If Hashyl<>0 Then
   For A7=Oryl To Oryl-Tendyl Step -Hashyl
      Draw Orx%-Lengthl,A7 To Orx%-Lengthl,A7
   Next A7
   For A7=Oryl To Oryl+Bendyl Step Hashyl
      Draw Orx%-Lengthl,A7 To Orx%-Lengthl,A7
   Next A7
Endif
Swap #P.array,Arrayl()
Return
Procedure Drawaxes(P.array)
   Local Orxl,Oryl,Lendxl,Rendxl,Tendyl,Bendyl,Hashxl,Hashyl
   Swap P.array,Arrayl()
   Orxl=Arrayl(1)
   Oryl=Arrayl(2)
   Lendxl=Arrayl(6) \& 65536
   Rendxl=Arrayl(6) And 65535
   Tendyl=Arrayl(7) \& 65536
   Bendyl=Arrayl(7) And 65535
   Defline 1,1,1,1
   If Lendxl=0 Then
      Defline 1,1,0,1
   Endif
   If Rendxl=0 Then
      Defline 1,1,1,0
   Endif
   Draw Orx%-Lendxl,Oryl To Orx%+Rendx%,Oryl
   Defline 1,1,1,1
   If Tendyl=0 Then
      Defline 1,1,0,1
   Endif
   If Bendyl=0 Then
      Defline 1,1,1,0
   Endif
   Draw Orxl,Oryl-Tendyl To Orxl,Oryl+Bendyl
   Swap #P.array,Arrayl()
   Defline 1,1,0,0
Return
Procedure Label_hashes(P.array)
   Local Hashl,Loendl,Hiendl,Scale,A7,Lbl$
   Swap #P.array,Arrayl()
   Defext 1,0,0,6
   Hashl=Arrayl(8) \& 65536
   Loendl=Arrayl(6) \& 65536
Hiend% = Array% (6) And 65535
Scale = Array% (9)
If Scale < 0 Then
  Scale = -1/Scale
Endif
Ax = Hash X To Hiend X Step Hash%
While Ax < Hiend X
  Lbl$ = Str $(Ax/Scale)
  Text Array% (1) + Ax - Len (Lb$) #4, Array% (2) + 18, Lb$
  If Ax < Loend X Then
    Text Array% (1) - Ax - (Len (Lb$) + 1) #4, Array% (2) + 18, "*" + Lb$
    Endif
  Add Ax, Hash %
  Ax = Hash % To Hiend X Step Hash%
Endwhile
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
Ax = Hash % To Hiend X Step Hash%
While Ax < Hiend X
  Lbl$ = Str $(Ax/Scale)
  Text Array% (1) - 10 - Len (Lb$) #8, Array% (2) - Ax + 4, Lb$
  If Ax < Loend X Then
    Text Array% (1) - 10 - (Len (Lb$) + 1) #8, Array% (2) + Ax + 4, "*" + Lb$
    Endif
  Add Ax, Hash %
  Ax = Hash % To Hiend X Step Hash%
Endwhile
Swap Arm array, Array %()
Deertext 1, 0, 0, 13
Return
Procedure Plot (P. array)
  Local Flg %, Upp er, Collect %, Xpoint %, Pol ar %, Lin e %, Spl it %, Log s %, Count %
  Local Dat u s, Ind ep, Xbeg X, Xend X, Ybeg X, Yend X, Max X, Sus X, Nave X
  Local Cl x, Cl y
  Swap P. array, Array %()
  Flags% = Array% (0)
  Upp er% = Flags% And 4
  Collect% = Flags% And 8
  Xpoint% = Flags% And 16
  Polar% = Flags% And 32
  Let Line% = Flags% And 64
  Split% = Flags% And 128
  Log s% = Flags% And 256
  Cl x% = Array% (9)
  Cl y% = Array% (10)
  If Cl x% < 0 Then
    Cl x% = -1/Cl x%
    Endif
  If Cl y% < 0 Then
    Cl y% = -1/Cl y%
    Endif
Scl y=-1/Scl
Endif
If Collect! Then
  SumZ=0
  NaveZ=0
Endif
Color Flags% And 1
Graphmode (Flags% And 2)+1
For CountZ=ArrayZ(3) To ArrayZ(4)
  If Split! Then
    Indep=Results(0,CountZ)
    Datu=Results(1,CountZ)
  Else
    Indep=CountZ
    Datu=Results(Abs(Upper!),CountZ)
  Endif
  If Logs! Then
    Indep=Log(Indep)
    Datu=Log(Datu)
  Endif
  If Not Polar! Then
    XbegZ=ArrayZ(1)+Abs(Xplot)+1 [(Indep*Sclx+ArrayZ(5))
    YbegZ=ArrayZ(2)+Abs(Xplot)+1 [(Indep*Scly+ArrayZ(5))
    XendZ=Datu*Sclx*Aabs(Xplot)
    YendZ=Datu*Scly*Aabs(Xplot)+1
  Else
    XbegZ=ArrayZ(1)
    YbegZ=ArrayZ(2)
    XendZ=Datu*Scly*Coss(Indep*Sclx-180)*fi/180)
    YendZ=-Datu*Scly*Sin(Indep*Sclx-180)*fi/180
  Endif
  If Line! Then
    If CountZ=ArrayZ(3) Then
      Draw XbegZ+XendZ,YbegZ-YendZ
    Else
      Draw To XbegZ+XendZ,YbegZ-YendZ
    Endif
  Else
    Draw XbegZ,YbegZ To XbegZ+XendZ,YbegZ-YendZ
  Endif
  If Collect! Then
    If DatuX>MaxZ Then
      MaxZ=DatuX
      ModeX=CountZ
    Endif
    Add SumX,Results(Abs(Upper!),CountX)*CountZ
    Add NaveX,Results(Abs(Upper!),CountX)
  Endif
Next CountZ
If Collect! Then
  Ave=SumZ/NaveZ
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 P1line(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$))
Endif
  Graphmode 1
  Shows
  Return
Procedure PText
  Local X, Y, K, In$, Title$, Mse$, Bigase$, Scr$, Size%, A$
  Size%=6
  Gosub P1line("<Left> - Locate text line  T - Toggle print Size  <ESC> aborts")
  Deftext 1,0,0,Size%
  Graphmode 1
  Mse$=Mk$(393224)+Mk$(65536)+Mk$(1)+Mk$(12080412160)+Mk$(1811949568)
  Mse$=Mk$(67109880)+Mk$(671116228)+Mk$(1843462016)+Mk$(1)+Mk$(1)+Mk$(1)
  Mse$=Mk$(27648)+Mk$(268439552)+Mk$(268439552)+Mk$(268439552)
  Mse$=Mk$(1811939328)+Mk$(1)+Mk$(1)+Mk$(1)
  Bigase$=Mk$(65537)+Mk$(1)+Mk$(1) ! Ref at 1,1; filler; standard colors 0,1
  Bigase$=Bigase$+Mk$(106536536)+Mk$(545267840)+Mk$(1530985992)
  Bigase$=Bigase$+Mk$(1147108448)+Mk$(1079984224)+Mk$(1590038144)
  Bigase$=Bigase$+Mk$(1065353216)
  Bigase$=Bigase$+Mk$(7936)+Mk$(520101632)+Mk$(6135856)+Mk$(1145076800)
  Bigase$=Bigase$+Mk$(1077952576)+Mk$(545267456)+Mk$(520101632)+Mk$(0)
  Deflouse Mse$
  Repeat
  Until Mousek=0
  Repeat
  Mouse X,Y,K
  A$=Inkey$
  If (Asc(A$) And 95)=84 Then
    SizeX=(SizeX-9.5)+9.5
  If SizeX=13 Then
    Deflouse Bigase$
  Else
    Deflouse Mse$
  Endif
  Deftext 1,0,0,SizeX
Endif
Exit If ASC=Chr$(27)
If X=1 Then
Get Scr$,
Gosub Pline("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$=Chr$(13) 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%,YO%
Mouse X,Y,K
Gosub P4line("<Right> - opens box, release records area.")

Graphmode 3
X0%=X
YO%=Y
While K=2
  Mouse X,Y,K
  Box X0%,YO%,X,Y
  Box X0%,YO%,X,Y
Wend
Get X0%,YO%,X,Y,Mse$
X0%=Min(X0%,X)
YO%=Min(Y0%,Y)
Put X0%,YO%,Mse$,6
Gosub P4line("<Left> - Places area C - Copies area D - Deletes area <ESC> aborts")

Hide

Graphmode 1
Do
  Mouse X,Y,K
  Put X,Y,Mse$,6
  A$=Intey$
  If (Asc(A$) And 95)=67 Then
    Put X0%,YO%,Mse$,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$,6
    Put X0%,YO%,Mse$,7
    K=1
  Endif
  Exit If K=1 Or MouseK=1
  Put X,Y,Mse$,6
Loop

Show

Repeat
  Until MouseK=0

Return

Procedure Centerofmass(P.array)
N%=0
Swap P.array,Avearray%(0)
Print "Processed 0 of" Avearray%(0)" Points"
Avex=Avearray%(2)
Avey=Avearray%(2) And 1023
Do
  Inc N%
  If N% Mod 100=0 Then
Print At(11,6):NZ
Endif
Exit If Avearray%(NX+1)=0 And Avearray%(NX+2)=0
AveX=(AveX%(NX-1)+(Avearray%(NX+1))\1024)/NX
AveY=(AveY%(NX-1)+(Avearray%(NX+1) And 1023))\NX
Loop
Sub AveX,200
AveY=200-AveY
Swap #P.array,Avearray%(l
Return
Procedure Fcenterofmass(P.array)
Swap #P.array,Avearray%(l
Print "Processed 0 of 160000 Points"
Countl=0
NX=0
Repeat
Inc NX
Until Avearray%(NX)<>0
Sixbitl=0
While (Avearray%(NX) And (63\&64^Sixbitl))=0
Inc Sixbitl
Wend
Uncoill=(NX-1)\5\Sixbitl
Exp%=64\Sixbitl
Freq%(Avearray%(NX) And (63\&Exp%=)/Exp%
AveX=(Uncoill\400)\Freq%
AveY=(Uncoill Mod 400)\Freq%
Add Countl,Freq%
Inc Sixbitl
Repeat
Exp%=64\Sixbitl
Freq%(Avearray%(NX) And (63\&Exp%=)/Exp%
If Freq%<>0 Then
AveX=(AveX%Countl+(Uncoill\400)\Freq%)/(Countl+Freq%)
AveY=(AveY%Countl+(Uncoill Mod 400)\Freq%)/(Countl+Freq%)
Add Countl,Freq%
Endif
Inc Sixbitl
Until Sixbitl>4
Inc NX
Repeat
Coil%(NX-1)\5
Freq%(Avearray%(NX)
Uncoill=Coill+Sixbitl
If Uncoill Mod 100=0 Then
Print At(11,6):Uncoill
Endif
If Freq%<>0 Then
For Sixbitl=0 To 4
Exp%=64\Sixbitl
Freq%(Avearray%(NX) And (63\&Exp%=)
If Freq%<>0 Then
Div Freq%, Exp7.
Avex=(Avex*Count%+(Uncoll%\400)*Freq%)/(Count%+Freq%)
Avey=(Avey*Count%+(Uncoll% Mod 400)*Freq%)/(Count%+Freq%)
Add Count%, Freq%
Endif
Next Sixbit%
Endif
Inc N%
Until N%>32000
Sub Avey,200
Avey=200-Avey
Swap 1P.array,Avearray%()
Return

Procedure Power
Local IX,NX,Sumofx,Sumofy,Sumofproducts,Sumofsquares
NX=Power(0,0)
Sumofx=0
Sumofy=0
Sumofproducts=0
Sumofsquares=0
For IX=1 To NX
Add Sumofx,Log(Power(0,IX))
Add Sumofy,Log(Power(1,IX))
Add Sumofproducts,Log(Power(0,IX))\Log(Power(1,IX))
Add Sumofsquares,Log(Power(0,IX))^2
Next IX
Slope=(NX*Sumofproducts-Sumofx*Sumofy)/(NX*Sumofsquares-Sumofx^2)
Intercept=(Sumofsquares*Sumofy-Sumofx*Sumofproducts)/(NX*Sumofsquares-Sumofx^2)
Return

' Deposition Frequency Histogram Program
Version=1.5
Revdate$="28 Jun 88"
Print "Deposition Frequency Histogramaker, version":Version;","Revdate$
Print "This program requires maximum 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 a; all pixels(x,y) that have a" Print "P(deposit) greater, lower, or equal to a will be displayed."
Print "The synthesized deposit will then be stored as a standard .SCR file with" Print "the exception that a (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"
Di• Order%(30000)
Di• Longlist%(32000)
D=0
Grand%=O
Segment%=0
Arrayfill Longlist(1,0)
Cls
Do
Cls
Showa
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 a for upper slice" 
Print
Print "3 Process field array with input of a for lower slice"
Print
Print "4 Process field array for frequency contours"
Print
Print "5 Helpful hints and instructions"
Print
Print "6 Exit"
Grasmode 3
Deffill 1.1
Ptrvertpos=Mousey
If Frac(Ptrvertpos/32)<0.5 Then
   Gosub Inbox(Ptrvertpos)
Else
   InX=0
Endif
Do
Repeat
   Ptrvertpos=Mousey
   If (InX>0) And (Frac(Ptrvertpos/32)>0.5) Then
      Gosub Outbox(Ptrvertpos)
   Endif
   If (InX=0) And (Frac(Ptrvertpos/32)<0.5) Then
      Gosub Inbox(Ptrvertpos)
   Endif
Switch%=Mousek
If Switch%>0 Then
   If InX>0 Then
      Switch%=(Ptrvertpos/32)
   Else
      Switch%=0
      Sound 1,15,6,7,5
      Sound 1,0
   Endif
Endif
Endif
Key$=Inkey$
Until Key$<>"" Or Switch%
If Switch% Then
  KeyS=Str$(Switch%)
End if
Exit if Val(Key$)$>0 And Val(Key$)$<7
  Sound 1,15,6,7,5
  Sound 1,0
Loop
Cls
Granhaade 1
On Val(Key$)$ Gosub Array,Upper,Lower,Contour,Help,Exit
In%=0
Switch%=0
Loop
End
Procedure Inbox(Ht%)
  Ht%=321(Ht%+32)
  If Ht%>16 And Ht%<224 Then
    Pbox -1,Ht%,500,Ht%+16
    In%=Ptrvertpos%\32
  End if
Return
Procedure Outbox(Ht%)
  Ht%=321In%
  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 e"n
Print "which is input at the prompt. .SCR, .FH6, 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 SixbitX,ExpX,FreqX,MainX,UncoilX,IterX
On Error GoSub Seg_array
GoSub Checkandload
Tmax=Longlist1(1041) And 63 ! The center pixel (200,200) always on.
GoSub Set_freq
File$="UPPER"+Str$(Int(FminX/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 a >="FminX/Tmax"
Print At(52,7);"based on"Tmax"deposits"
Print At(52,20);"% of screen painted"
OrderX(0)=1
OrderX(1)=201
For IterX=1 To 32000
    CoilX=IterX-1/5
    FreqX=Longlist1(IterX)
    If IterX Mod 520=0 Then
        Print At(52,20);Using "####",IterX/32006
    Endif
    If FreqX<>0 Then
        For SixbitX=0 To 4
            UncoilX=CoilX+SixbitX
            ExpX=64*SixbitX
            FreqX=Longlist1(IterX) And (63*ExpX)
            If FreqX<>0 Then
                Div FreqX,ExpX
                If FreqX=FminX Then
                    XpixelX=UncoilX/400
                    YpixelX=UncoilX Mod 400
                    Plot XpixelX,YpixelX
                    Inc OrderX(0)
                    Option "U1"
                    OrderX(OrderX(0))=XpixelX*1024+YpixelX
                Endif
                Endif
            Next SixbitX
        Endif
    Next IterX
Hide
Print At(52,20);"Save file names:"
Print At(52,21);File$
Print At(52,22);"and"
Print At(52,23);"*ARR"
GoSub Save_segs
Return
Procedure Lower
Local Sixbit%, Exp%, Freq%, Fmain%, Coil%, Uncoll%, Iter%
On Error GOSUB Seg_array
GOSUB Checkandload
Tmax%=Longlist%(16041) And 63 ! The center pixel (200, 200) always on.
GOSUB Get_freq
Files="LOWEr"+Str$(Int(Fmain%/Tmax%*100))+.SCR"
Cls
Print At(52,1); "lower slice of"
Print At(52,2); "frequency histogram"
Print At(52,4); "pixels displayed"
Print At(52,5); "represent sites with"
Print At(52,6); "frequency <=" Fmain%/Tmax%
Print At(52,7); "based on " Tmax% "deposits"
Print At(52,20); " 0% of screen painted"
Order%=1
Order%(1)=101
For Iter%=1 To 32000
Coil%=Iter%*1+5
Freq%=Longlist%(Iter%)
If Iter% Mod 320=0 Then
Print At(52,20); Using "####", Iter%/32000
Endif
If Freq<>0 Then
For Sixbit%=0 To 4
Uncoil%=Coil%+Sixbit%
Exp%=64*Sixbit%
Freq%=Longlist%(Iter%) And /63*Exp%
If Freq<>0 Then
Div Freq%, Exp%
If Freq<=Fmain% And Freq<>0 Then
Xpixel%=Uncoil%*400
Ypixel%=Uncoil% Mod 400
Plot Xpixel%, Ypixel%
Inc Order%(0)
Option "U1"
Order%(Order%(0))=Xpixel%*1024+Ypixel%
Endif
Endif
Next Sixbit%
Endif
Next Iter%
Hide
Print At(52,20); Space$(28)
Bsaves Files, xbios(2), 32000
Print At(52,20); "save filenames:"
Print At(52,21); Files
Print At(52,22); "and"
Files=Left$(Files, Instr$(Files, "."))+.ARR"
Print At(52,23); Files
GOSUB Save_segs
Return
Procedure Contour
Local Sixbit%,Exp%,Freq%,Fmin%,Coil%,Uncoil%,Iter%
On Error GoTo Seg_array
Gosub Checkload
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(52,3);"Pixels displayed"
Print At(52,4);"represent sites with"
Print At(52,5);"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)=291
For Iter%=1 To 32000
Coil!=(Iter%-1)4
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$(0)
Option "UI"
Order$(Order$(0)+Xpixel%1024+Ypixel%
Endif
Endif
Next Sixbit%
Endif
Next Iter%
Hide
Print At(52,20);Space$(28)
Bsave File$,Xbios$(2),32000
Print At(52,20);"Save filenames:"n
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.FSH"
            Print "Checking device""Left$(File$+2)" for LONGLIST.FSH"
            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$(27) Then Edit Endif
        Loop
        Print "Loading""File$"
        Load File$, Peek(Arrptr(Longlist$(1)))
        Arrayfill Order$(1), 0
        Endif
    Return
End

**Procedure Get_freq**

Local Fain$

Print "Cutoff frequencies must be integer multiples of 1/"; T1ax%
Print "Frequency will automatically be rounded to nearest 1/"; T1ax%; "th."

Do
    Input "Cutoff frequency (absolute n, or all?);Fain$
    If Instr$(Fain$, ",")<>0 Then
        Fain%=T1ax%#0.01#Val$(Fain$)+0.5
    Else
        Fain%=Val$(Fain$)+0.5
    Endif
    Exit If Fain%<=T1ax%
    Print "Frequency can't exceed 100% or""T1ax%""deposits. Please reenter."
Loop

**Procedure Seg_array**

Local Ecode$, Seg$, Seg$

Ecode$=Err

On Error Gosub Seg_array
    On Error=16 Then
        On Error=
        Error Ecode$
        Endif
    Seg$=(Seg%+29999)+1

End
Print At(S1,21);"Segmenting .ARR file"  
Print At(S1,22);"Segment"' Seg%  
Print At(S1,23);"Please wait..."  
If Seg%>1 Then  
  Seg%=Left$(File$,Instr$(File$,"."))+#ARR+#Str$(Seg%)  
Else  
  Seg%=Left$(File$,Instr$(File$,"."))+#ARR*  
Endif  
Dec Order(0)  
Bsave Seg%,Lpeek(Arrptr(Order(1))),Order(0)##4+8  
Arrayfill Order(1),0  
Order(0)=2  
Order(1)=201  
Add Segment,29999  
Print At(S2,21);Spc(29)  
Print At(S2,22);Spc(29)  
Print At(S2,23);Spc(29)  
Resume  
Return  
Procedure Save_segs  
Local Base$  
Base$=Left$(File$,Instr$(File$,"."))  
If Segment=0 Then  
  Bsave File$,Lpeek(Arrptr(Order(1))),Order(0)##4+8  
Else  
  File$=Base$+#ARR+Str$(Int(Segment##29999)+1)  
  Bsave File$,Lpeek(Arrptr(Order(1))),Order(0)##4+8  
  Open "R",0,Base$+#ARR*,4  
  Field 1,4 As Buf$  
  Lset Buf$=MK$(Segment*Order(0))  
  Put 1,2  
  Close 1  
Endif  
Return  
Procedure Array  
Print At(S2,1);"Deposit Grand Total= 0"  
Print At(S2,2);"File:"  
Print At(S2,5);"File number=",'0  
Print At(S2,7);"N= 0"  
Print At(S2,9);"Out of 0 total deposits"  
Repeat  
  Dir ".".ARR To "FREQHIST.DIR"  
  Open ".",0,"FREQHIST.DIR"  
Repeat  
    Gosub Loader  
    Gosub Process  
Until Eof(0)  
Repeat  
  Print At(S2,22);"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"
  Repeat
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.FH6",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",OrderI(0);1
  MZ=2
  Repeat
    Xpixel%=Order%/(MZ)\1024
    Ypixel%=Order%/(MZ) And 1023
    Plot Xpixel%,Ypixel%
    Coil%=400*Xpixel%+Ypixel%
    Disp%=Coil% Mod 5
    Coil%=Coil%\5
    Add Longlist%(Coil%+1),64*Disp1
Inc GrandZ
If (NZ-1) Mod 100=0 Then
  Print At(72,1):Using "$$$$$$",GrandZ
  Print At(54,7):Using "$$$$$$",NZ-1
Endif
Inc NZ
Until NZ>OrderZ(O)
Print At(52,1):Using "Deposit Grand Total=$$$$$$",GrandZ
LonglistZ(O)=GrandZ
Return
Procedure Loader
  Arrayfill OrderZ(),0
  Input 0,Files
  Load Files,peek(Arrptr/OrderZ())
Return
Procedure Blank
  Defill 0,1
  Pbox 401,124,639,399
Return
# APPENDIX B

## NUMERICAL DATA

### TABLE II

**FRACTAL DIMENSION DATA FOR INDIVIDUAL SMALL AGGREGATES**

<table>
<thead>
<tr>
<th>Deposit</th>
<th>N</th>
<th>Dm</th>
<th>'Circles'</th>
<th>Dc</th>
<th>'Circles'</th>
<th>Circles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4767</td>
<td>1.8355364401</td>
<td>1.6573104232</td>
<td>1.6020511402</td>
<td>1.680587015</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3825</td>
<td>1.9224613245</td>
<td>1.6609789001</td>
<td>1.6064645531</td>
<td>1.6810192676</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3999</td>
<td>1.8374010038</td>
<td>1.6599607926</td>
<td>1.6062149391</td>
<td>1.6875667926</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4621</td>
<td>1.7910889197</td>
<td>1.652560963</td>
<td>1.598715229</td>
<td>1.6831302681</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2972</td>
<td>1.8931199782</td>
<td>1.6661022793</td>
<td>1.6020511402</td>
<td>1.6880587015</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>4969</td>
<td>1.8568094251</td>
<td>1.658277456</td>
<td>1.6020197103</td>
<td>1.6880587015</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4639</td>
<td>2.0458284656</td>
<td>1.6453806026</td>
<td>1.5943751484</td>
<td>1.6857970137</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>4354</td>
<td>1.807984199</td>
<td>1.6621301618</td>
<td>1.6069461333</td>
<td>1.6968237371</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>5335</td>
<td>1.8242807075</td>
<td>1.671625386</td>
<td>1.6187794988</td>
<td>1.7187509464</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>4512</td>
<td>1.8398362972</td>
<td>1.6740125454</td>
<td>1.6162811829</td>
<td>1.6974291693</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>3314</td>
<td>1.871951025</td>
<td>1.6601788582</td>
<td>1.6039026676</td>
<td>1.6749274673</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>4622</td>
<td>1.7301422922</td>
<td>1.6670967371</td>
<td>1.6151354645</td>
<td>1.6974291693</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>4529</td>
<td>1.816270042</td>
<td>1.6852599808</td>
<td>1.629300178</td>
<td>1.7195551784</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>3793</td>
<td>1.9113907128</td>
<td>1.6597513458</td>
<td>1.604103153</td>
<td>1.6807282872</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>5000</td>
<td>1.744598435</td>
<td>1.674626863</td>
<td>1.6185214249</td>
<td>1.7054809105</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>5042</td>
<td>1.908052491</td>
<td>1.668358268</td>
<td>1.6143191837</td>
<td>1.7050727751</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>3795</td>
<td>1.8193060175</td>
<td>1.6594866335</td>
<td>1.6081433871</td>
<td>1.688575014</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>4420</td>
<td>1.8347762916</td>
<td>1.6675455251</td>
<td>1.6127097971</td>
<td>1.6962384505</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>4411</td>
<td>1.7749146693</td>
<td>1.6571104783</td>
<td>1.6045614382</td>
<td>1.6832531101</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>5764</td>
<td>1.839819071</td>
<td>1.667699506</td>
<td>1.6104902431</td>
<td>1.717788699</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>5118</td>
<td>1.9340069265</td>
<td>1.663276023</td>
<td>1.6127396026</td>
<td>1.7101350582</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>3506</td>
<td>1.8022675617</td>
<td>1.6635810306</td>
<td>1.6119526325</td>
<td>1.6881568945</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>5238</td>
<td>1.8128932524</td>
<td>1.663527148</td>
<td>1.6087703385</td>
<td>1.7004290546</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>4132</td>
<td>1.8449878424</td>
<td>1.662567821</td>
<td>1.6062567044</td>
<td>1.6925987278</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>5212</td>
<td>1.8311292049</td>
<td>1.6781571799</td>
<td>1.622053334</td>
<td>1.713615252</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>5080</td>
<td>1.8549047671</td>
<td>1.6544908435</td>
<td>1.6005912453</td>
<td>1.6962772363</td>
</tr>
</tbody>
</table>
# Table III

## Fractal Dimension Data for Individual Large Aggregates

<table>
<thead>
<tr>
<th>Deposit</th>
<th>$N$</th>
<th>$D_m$</th>
<th>$D_c$</th>
<th>$D_s$</th>
<th>$D_k$</th>
<th>$D_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>17428</td>
<td>1.82979542</td>
<td>1.669483252</td>
<td>1.6126899269</td>
<td>1.6747005497</td>
<td>1.670355607</td>
</tr>
<tr>
<td>2</td>
<td>14972</td>
<td>1.8571485274</td>
<td>1.665933746</td>
<td>1.611384213</td>
<td>1.675095405</td>
<td>1.646791979</td>
</tr>
<tr>
<td>3</td>
<td>15141</td>
<td>1.777909866</td>
<td>1.6695384163</td>
<td>1.61414633</td>
<td>1.671104925</td>
<td>1.6163791096</td>
</tr>
<tr>
<td>4</td>
<td>15243</td>
<td>1.774517225</td>
<td>1.6682995941</td>
<td>1.6118757841</td>
<td>1.675022653</td>
<td>1.614864462</td>
</tr>
<tr>
<td>5</td>
<td>12208</td>
<td>1.8329317474</td>
<td>1.6688731604</td>
<td>1.6120068888</td>
<td>1.670575911</td>
<td>1.638014772</td>
</tr>
<tr>
<td>6</td>
<td>18052</td>
<td>1.84376623</td>
<td>1.6653686705</td>
<td>1.6073511179</td>
<td>1.673057665</td>
<td>1.610872685</td>
</tr>
<tr>
<td>7</td>
<td>19429</td>
<td>1.9543270665</td>
<td>1.6670432838</td>
<td>1.606346917</td>
<td>1.678194158</td>
<td>1.6212366642</td>
</tr>
<tr>
<td>8</td>
<td>16525</td>
<td>1.775629843</td>
<td>1.659262413</td>
<td>1.6031247951</td>
<td>1.668189111</td>
<td>1.615453573</td>
</tr>
<tr>
<td>9</td>
<td>21268</td>
<td>1.785359122</td>
<td>1.6672958319</td>
<td>1.6113788372</td>
<td>1.679362124</td>
<td>1.620381864</td>
</tr>
<tr>
<td>10</td>
<td>17189</td>
<td>1.7354014398</td>
<td>1.663570367</td>
<td>1.6098319099</td>
<td>1.669308723</td>
<td>1.644074109</td>
</tr>
<tr>
<td>11</td>
<td>13706</td>
<td>1.8248885544</td>
<td>1.6666992685</td>
<td>1.6092906897</td>
<td>1.6700949655</td>
<td>1.612800032</td>
</tr>
<tr>
<td>12</td>
<td>14395</td>
<td>1.7337458246</td>
<td>1.668016443</td>
<td>1.610045142</td>
<td>1.671784013</td>
<td>1.631569847</td>
</tr>
<tr>
<td>13</td>
<td>13520</td>
<td>1.78879105</td>
<td>1.676510535</td>
<td>1.63201931</td>
<td>1.6799353916</td>
<td>1.62090424</td>
</tr>
<tr>
<td>14</td>
<td>12854</td>
<td>1.8613357844</td>
<td>1.667974528</td>
<td>1.6125763577</td>
<td>1.6715098094</td>
<td>1.6167550434</td>
</tr>
<tr>
<td>15</td>
<td>14897</td>
<td>1.73335049</td>
<td>1.64982349</td>
<td>1.6076303434</td>
<td>1.6718440979</td>
<td>1.6138223356</td>
</tr>
<tr>
<td>16</td>
<td>16326</td>
<td>1.8493838524</td>
<td>1.6607155269</td>
<td>1.6052765298</td>
<td>1.664775932</td>
<td>1.608647983</td>
</tr>
<tr>
<td>17</td>
<td>14944</td>
<td>1.797824281</td>
<td>1.6689430768</td>
<td>1.6111278969</td>
<td>1.6716377209</td>
<td>1.613429827</td>
</tr>
<tr>
<td>18</td>
<td>16339</td>
<td>1.8051444432</td>
<td>1.6637643543</td>
<td>1.63917652</td>
<td>1.6667147955</td>
<td>1.641029009</td>
</tr>
<tr>
<td>19</td>
<td>15752</td>
<td>1.769612464</td>
<td>1.6720235857</td>
<td>1.614437524</td>
<td>1.675944255</td>
<td>1.617854626</td>
</tr>
<tr>
<td>20</td>
<td>17715</td>
<td>1.802851386</td>
<td>1.663071914</td>
<td>1.603771854</td>
<td>1.6662993666</td>
<td>1.6105453471</td>
</tr>
<tr>
<td>21</td>
<td>14725</td>
<td>1.877910182</td>
<td>1.667026357</td>
<td>1.6142472474</td>
<td>1.675269272</td>
<td>1.621859511</td>
</tr>
<tr>
<td>22</td>
<td>12613</td>
<td>1.732079325</td>
<td>1.670146412</td>
<td>1.615916395</td>
<td>1.6745998534</td>
<td>1.619571032</td>
</tr>
<tr>
<td>23</td>
<td>1644</td>
<td>1.789308771</td>
<td>1.6744682685</td>
<td>1.6178778081</td>
<td>1.6815031254</td>
<td>1.6243320708</td>
</tr>
<tr>
<td>24</td>
<td>1761</td>
<td>1.782920103</td>
<td>1.6688924955</td>
<td>1.6152812764</td>
<td>1.677442037</td>
<td>1.623302901</td>
</tr>
<tr>
<td>25</td>
<td>16907</td>
<td>1.806531114</td>
<td>1.6733568534</td>
<td>1.6161727463</td>
<td>1.676282729</td>
<td>1.6220518204</td>
</tr>
<tr>
<td>26</td>
<td>19615</td>
<td>1.8474985009</td>
<td>1.661437214</td>
<td>1.607204545</td>
<td>1.669570227</td>
<td>1.61454239</td>
</tr>
<tr>
<td>27</td>
<td>19056</td>
<td>1.7922802635</td>
<td>1.6716166637</td>
<td>1.619880929</td>
<td>1.6857029235</td>
<td>1.6287472478</td>
</tr>
<tr>
<td>28</td>
<td>17401</td>
<td>1.8293229347</td>
<td>1.6573971729</td>
<td>1.604228967</td>
<td>1.653875793</td>
<td>1.6101776992</td>
</tr>
<tr>
<td>29</td>
<td>15680</td>
<td>1.7937369861</td>
<td>1.6629574201</td>
<td>1.6036376033</td>
<td>1.662379819</td>
<td>1.6100401207</td>
</tr>
<tr>
<td>30</td>
<td>14909</td>
<td>1.8046050822</td>
<td>1.6536850346</td>
<td>1.5971666179</td>
<td>1.657305395</td>
<td>1.600491638</td>
</tr>
</tbody>
</table>

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.
\[ D_{Rg} = 1.830 \]

**Figure 21.** \( \ln(R_g) \) vs. \( \ln(N) \) for aggregate number 1.

\[ D_{Rg} = 1.857 \]

**Figure 22.** \( \ln(R_g) \) vs. \( \ln(N) \) for aggregate number 2.
Figure 23. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 3.

$D_{R_g} = 1.778$

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

$D_{R_g} = 1.775$
Figure 25. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 5.

$D_{R_g} = 1.833$

Figure 26. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 6.

$D_{R_g} = 1.844$
Figure 27. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 7.

$D_{R_g} = 1.954$

Figure 28. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 8.

$D_{R_g} = 1.776$
Figure 29. \( \ln(R_g) \) vs. \( \ln(N) \) for aggregate number 9.

\[ D_{R_g} = 1.785 \]

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

\[ D_{R_g} = 1.735 \]
$D_{Rg} = 1.825$

Figure 31. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 11.

$D_{Rg} = 1.736$

Figure 32. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 12.
\[ D_{Rg} = 1.789 \]

*Figure 33.* \( \ln(R_g) \) vs. \( \ln(N) \) for aggregate number 13.

\[ D_{Rg} = 1.861 \]

*Figure 34.* \( \ln(R_g) \) vs. \( \ln(N) \) for aggregate number 14.
Figure 35. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 15.

$D_{R_g} = 1.733$

Figure 36. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 16.

$D_{R_g} = 1.869$
$D_{R_g} = 1.798$

Figure 37. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 17.

$D_{R_g} = 1.805$

Figure 38. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 18.
\[ D_{Rg} = 1.770 \]

**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.

$D_{R_g} = 1.878$

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

$D_{R_g} = 1.732$
Lower $D_{R_g}$ cutoff

ALL 1.798
2 1.795
4 1.766
6 1.744
8 1.700

Figure 43. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 23.

$D_{R_g} = 1.783$

Figure 44. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 24.
Figure 45. Ln($R_g$) vs. Ln($N$) for aggregate number 25.

$D_{R_g} = 1.807$

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

$D_{R_g} = 1.843$
Lower $D_{Rg}$ cutoff

<table>
<thead>
<tr>
<th>ALL</th>
<th>1.792</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.797</td>
</tr>
<tr>
<td>4</td>
<td>1.771</td>
</tr>
<tr>
<td>6</td>
<td>1.754</td>
</tr>
<tr>
<td>8</td>
<td>1.729</td>
</tr>
</tbody>
</table>

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

$D_{Rg} = 1.829$

Figure 48. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 28.
Figure 49. $\ln(R_g)$ vs. $\ln(N)$ for aggregate number 29.

$D_{R_g} = 1.794$

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

$D_{R_g} = 1.805$
APPENDIX C

GRAPHICAL DATA

1

2
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, 

\[ I = I_{c.m.} + N L^2, \]

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 \( L(N_{\text{max}}) \) 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\( (N_{\text{max}}) \)). 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 \( D_{fractal} = 1.69 \). The results based on the approximate radius of gyration for aggregate number 20 from Appendix B, are, for the small aggregate, \( D_{fractal} = 1.83 \), and for the large aggregate, \( D_{fractal} = 1.81 \), 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 \( D_{fractal} = 1.67 \). 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 = \( 0.58 \pm 0.02 \). This result yields a fractal dimension of \( 1.73 \pm 0.06 \). The raw data for this calculation is listed
TABLE IV
CORRECTED RADIUS OF GYRATION RESULTS FOR AGGREGATE NUMBER 20

<table>
<thead>
<tr>
<th>% N</th>
<th>DEPOSITS</th>
<th>Ln(N)</th>
<th>Ln(R_g)</th>
<th>R_g</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>17715</td>
<td>9.78210</td>
<td>4.690065</td>
<td>4.017097</td>
</tr>
<tr>
<td>95</td>
<td>16829</td>
<td>9.730858</td>
<td>4.658574</td>
<td>4.096033</td>
</tr>
<tr>
<td>90</td>
<td>15944</td>
<td>9.676838</td>
<td>4.626698</td>
<td>3.901621</td>
</tr>
<tr>
<td>85</td>
<td>15058</td>
<td>9.619665</td>
<td>4.593242</td>
<td>4.014032</td>
</tr>
<tr>
<td>80</td>
<td>14172</td>
<td>9.559023</td>
<td>4.558203</td>
<td>4.011677</td>
</tr>
<tr>
<td>75</td>
<td>13286</td>
<td>9.494467</td>
<td>4.520182</td>
<td>4.383139</td>
</tr>
<tr>
<td>70</td>
<td>12401</td>
<td>9.425532</td>
<td>4.479921</td>
<td>4.448924</td>
</tr>
<tr>
<td>65</td>
<td>11515</td>
<td>9.351406</td>
<td>4.436583</td>
<td>4.809595</td>
</tr>
<tr>
<td>60</td>
<td>10629</td>
<td>9.271342</td>
<td>4.390161</td>
<td>5.046088</td>
</tr>
<tr>
<td>55</td>
<td>9743</td>
<td>9.184303</td>
<td>4.338853</td>
<td>5.412689</td>
</tr>
<tr>
<td>50</td>
<td>8858</td>
<td>9.089076</td>
<td>4.281730</td>
<td>6.012647</td>
</tr>
<tr>
<td>45</td>
<td>7972</td>
<td>8.983691</td>
<td>4.218701</td>
<td>5.740685</td>
</tr>
<tr>
<td>40</td>
<td>7086</td>
<td>8.865877</td>
<td>4.148299</td>
<td>5.921705</td>
</tr>
<tr>
<td>35</td>
<td>6200</td>
<td>8.732305</td>
<td>4.069269</td>
<td>6.429103</td>
</tr>
<tr>
<td>30</td>
<td>5315</td>
<td>8.578288</td>
<td>3.977641</td>
<td>7.069431</td>
</tr>
<tr>
<td>25</td>
<td>4429</td>
<td>8.395929</td>
<td>3.868652</td>
<td>7.602726</td>
</tr>
<tr>
<td>20</td>
<td>3543</td>
<td>8.172728</td>
<td>3.739126</td>
<td>7.767349</td>
</tr>
<tr>
<td>15</td>
<td>2657</td>
<td>7.884954</td>
<td>3.571964</td>
<td>6.718258</td>
</tr>
<tr>
<td>10</td>
<td>1772</td>
<td>7.479864</td>
<td>3.332235</td>
<td>5.915089</td>
</tr>
<tr>
<td>5</td>
<td>886</td>
<td>6.786717</td>
<td>2.950274</td>
<td>4.181003</td>
</tr>
</tbody>
</table>

\[
\text{Ln}(R_g) \approx -1.147 + 0.600 \times \text{Ln}(N)
\]
\[
R^2 = 0.948
\]

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, \( \text{slope} = .571 \). The correlation coefficient for the regression is, \( R = .99 \) and the residual variance is \( .028 \). These results yield a fractal dimension, \( D_{\text{fractal}} = 1.75 \pm .08 \). 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

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

$$\ln(R_g) \approx -0.859 + 0.571 \ln(N)$$

$R = 0.997$

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 Rc.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, $D_{fr} = 1.73 \pm 0.06$, 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 pseudo-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 probability. 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 toroidal, 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.