Computer Simulation of Random and Non-random Second-Phase Particle Distributions for Both Constant and Varying Particle Size

by

Mark S. Manfredi
Lieutenant, United States Navy
B.S., Massachusetts Maritime Academy

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September, 1992

UNCLASSIFIED

CURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

a SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution is unlimited

b. DECLASSIFICATION/DOWNGRADING SCHEDULE

PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

a. NAME OF PERFORMING ORGANIZATION
Iechanical Engineering Dept.
Naval Postgraduate School

6b. OFFICE SYMBOL *(if applicable)*
34

7a. NAME OF MONITORING ORGANIZATION
Naval Postgraduate School

c. ADDRESS *(City, State, and ZIP Code)*
Ionterey, CA 93943-5000

7b. ADDRESS *(City, State, and ZIP Code)*
Monterey, CA 93943-5000

a. NAME OF FUNDING/SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(if applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNI ACCESSIO

. TITLE *(Include Security Classification)* *COMPUTER SIMULATION OF RANDOM AND NON-RANDOM SECOND -PHASE PARTICLE ISTRIBUTIONS FOR BOTH CONSTANT AND VARYING PARTICLE SIZE.*

. PERSONAL AUTHOR(S)
Manfredi, Mark, S.

3a. TYPE OF REPORT
Iaster's Thesis

13b. TIME COVERED
FROM ______ TO:________

14. DATE OF REPORT *(Year, Month, Day)*
1992, September

15. PAGE COUNT
117

. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the offici policy or position of the Department of Defense or the United States Government.

. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Particle distribution, random distribution, non-random distribution, lognormal size distribution

. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Mechanical properties of two phase materials, such as strength, ductility and toughness, depend on the size istribution of the second phase. However, no methods are presently available to accurately quantify the homog y of the distribution of the second phase. Here, random and non-random second-phase particle distributions h een simulated by computer and analyzed for various area fractions. Distributions of particles with a lognormal istribution have been analyzed as well. A statistically sufficient number of particles for use in the model was de ined and used for all simulations. Average first nearest neighbor spacing values for dilute arrays of particles roach those of Poisson distributions of infinitesimal points. As the particle density increases, the average spa alues approach those of hexagonal arrays. For low area fractions there is little distinction between random and r andom distributions, both from statistical and visual perspectives. For higher area fractions there is a discern ifference between the statistical data for random and non-random distributions, but the visual differences are n bvious. These observations hold for both constant size particles and particles with a lognormal size distributio

. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

a. NAME OF RESPONSIBLE INDIVIDUAL
erry R. McNelley Prof., Mechanical Engineering

22b. TELEPHONE *(Include Area Code)*
(408) 646-2586

22c. OFFICE SYMBOL
ME/HY

83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

ABSTRACT

Mechanical properties of two phase materials, such as strength, ductility and toughness, depend on the size and distribution of the second phase. However, no methods are presently available to accurately quantify the homogeneity of the distribution of the second phase. Random and non-random second phase particle distributions have been simulated by computer and analyzed for various area fractions. Distributions of particles with a lognormal size distribution have been analyzed as well. A statistically sufficient number of particles for use in the model was determined and used for all simulations. Average first nearest neighbor spacing values for dilute arrays of particles approach those of Poisson distributions of infinitesimal points. As the particle density increases, the average spacing values approach those of hexagonal arrays. For low area fractions there is little distinction between random and non-random distributions, both from statistical and visual perspectives. For higher area fractions there is a discernable difference between the statistical data for random and non-random distributions, but the visual differences are more obvious. These observations hold for both constant size particles and particles with a lognormal size distribution.

TABLE OF CONTENTS

I. INTRODUCTION 1

II. BACKGROUND 6

III. EXPERIMENTAL PROCEDURE 11

A. OVERVIEW 11

B. RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES, RANPART1 11

C. NON-RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES, RANPART2 16

D. RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION, RANPART3 18

E. NON-RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION, RANPART4 21

IV. RESULTS AND DISCUSSION 23

A. OVERVIEW 23

B. DETERMINATION OF A SUFFICIENT NUMBER OF PARTICLES AND THE PROBLEM OF PARTICLES POSITIONED ALONG THE BORDER 23

C. RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES 32

D. NON-RANDOM DISTRIBUTION OF CONSTANT SIZE

PARTICLES 44

E. RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION 56

F. NON-RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION 67

V. CONCLUSIONS AND RECOMMENDATIONS 83

APPENDIX A. PROGRAM RANPART1.FOR 87

APPENDIX B. PROGRAM RANPART2.FOR 92

APPENDIX C. PROGRAM RANPART3.FOR 98

APPENDIX D. PROGRAM RANPART4.FOR 103

LIST OF REFERENCES 109

INITIAL DISTRIBUTION LIST 110

I. INTRODUCTION

Most engineering structural applications involve the use of materials containing second phases in the form of discrete precipitates or distributions of particulates. This latter category includes discontinuously reinforced metal-matrix composites. The behavior of these materials during processing and the final mechanical properties depend upon the nature, volume fraction, size and also the dispersion of the particles.

In both conventional alloys and in metal-matrix composites the second phase particles may be obstacles to the movement of dislocations. If a dislocation is prevented from completely traversing the crystal under an applied stress, then the crystal will become harder to deform. A higher stress is then required to move the dislocation past the obstacle and therefore the material must be stronger. Orowan [Ref. 1] found that the shear stress required for a dislocation to bypass a pair of small particles may be expressed as:

$$\tau = \frac{G_M b}{D} \qquad (1.1)$$

where G_M is the shear modulus of the matrix, b is the burgers vector and D is the space between particles in the slip plane. From this relationship it can be seen that a smaller inter-particle spacing will result in a larger stress to move the dislocation past the particles.

Larger second phase particles will also affect nucleation and grain growth during recrystallization of conventional alloys. In general it has been found that greater inter-particle spacing and larger particles will facilitate the process of recrystallization [Ref. 2]. Close particle spacing and small particles have the opposite effect. The second-phase particles increase the dislocation density when the material is plastically deformed, increasing the driving force for recrystallization. However, they also impede the rearrangement of dislocations to form mobil high angle grain boundaries and thereby hinder the migration of these grain boundaries, thus retarding recrystallization.

In research on Al alloys with an $AlCu_2$ particle size greater than 0.5 μm it was found that inter-particle spacing had the greatest effect on recrystallization [Ref. 2]. For wide particle spacings, subgrains formed by the rearrangement of dislocations and grew until they reached a critical size, at which point they would begin to migrate. The movement of the subgrains was impeded by the particles. This allowed for further nucleation of additional subgrains and resulted in accelerated recrystallization.

The deformation structure of a deformed alloy is also dependent on the particle size. Large particles cause local distortion of the lattice and increase the rate of nucleation. Small particles increase the homogeneity of the dislocation distribution which slows down the nucleation rate. Therefore, for an arrangement of coarse, widely dispersed particles the recrystallization is significantly increased. On the other hand, for a small particle diameter and small inter-particle spacing, the dislocation distribution is much more dense and homogeneous which reduces the rate of recrystallization.

Second phase particle distributions may be either random or non-random. Random or uniform distribution infers that local particle densities are the same throughout the material. Non-random distributions are just the opposite and may be the result of insufficient processing of the material. High and low density bands of particles may be the result of processing an MMC where the original material consisted of clustered groups of particles [Ref. 3].

Kocks [Ref. 4] showed that the average area *a* swept out by a single dislocation segment is dependent on the applied stress σ. This relationship is shown in Figure 1. It can be seen that there exist a stress in which a dislocation can sweep out an area and keep moving indefinitely. This asymptotic stress is the macroscopic flow stress σ. It is also apparent that the greatest flow stress for a given particle density is for a regular array of particles, is least for a clustered array, and the flow stress for a random array lies between the two. Thus the distribution of second phase particles has a significant role in determining the mechanical properties of two phase materials.

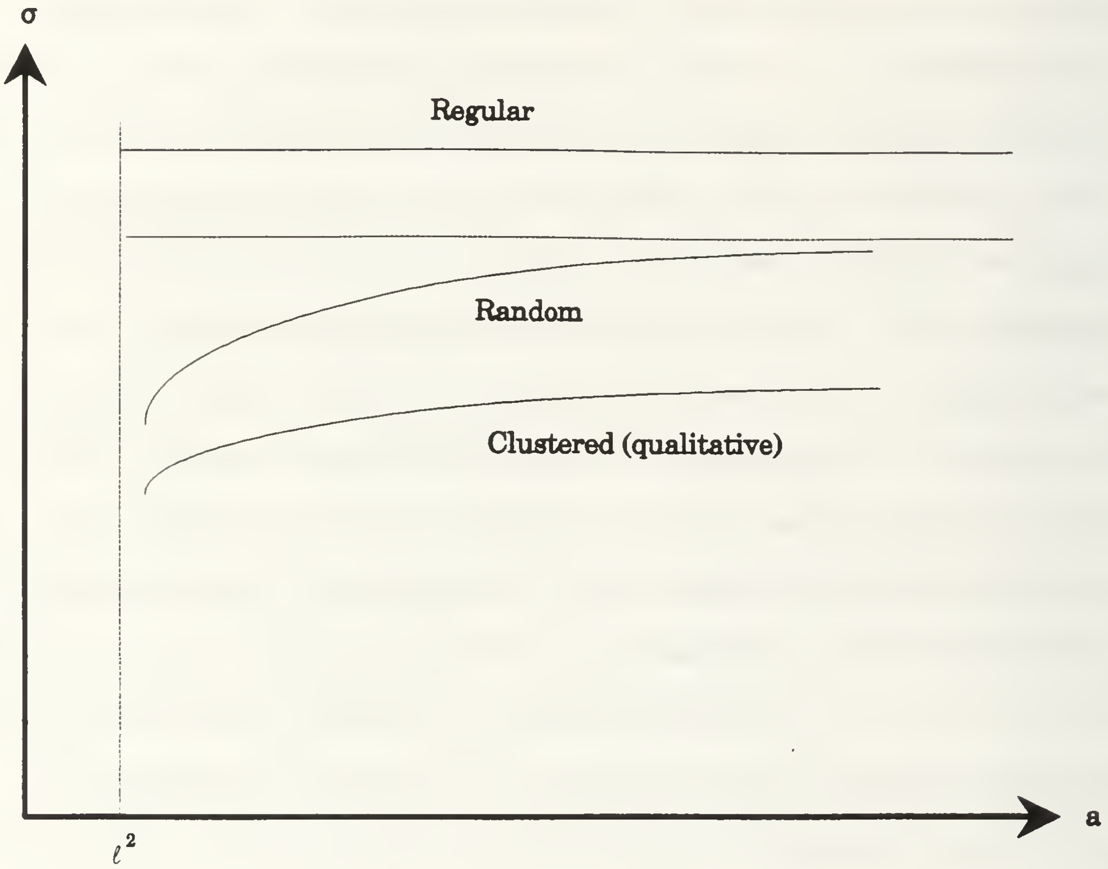


Figure 1. Applied stress σ vs mean free slip area a of one dislocation loop, for strong obstacles of area density $1/\ell^2$, in square, random and clustered arrays [Ref. 3].

Clearly, it would be desirable to have a means of accurately assessing the different types of microstructures resulting from various processing techniques. With this knowledge these techniques can be optimized to efficiently achieve the desired properties of the material. Current methods used to determine these values assume that the distribution is random and that the particles are infinitesimal points, which is certainly not the case for real materials. Methods do exist for finite sized particles but here the assumption is that they are distributed in a regular geometric array.

This study, which is essentially follows on from that of M. Pas [Ref. 5], was undertaken to quantify such values as average first nearest neighbor particle spacings (δ) for various area fractions, particle size distributions and for non-random cases as well. Any trends or patterns in the associated data was investigated as well. In order to facilitate mathematical and computer analysis computations were kept on a two dimensional basis.

II. BACKGROUND

The nearest neighbor separation distance between second-phase particles governs many properties of two phase materials. Properties such as strength, ductility, and toughness are sensitive to the mean separation distance and may also be affected by the range and uniformity of this distance throughout the material. Current methods for determining this spacing for particles of finite size is actually based on the assumption that the particles are infinitesimal points which do not occupy space. Throughout this thesis, unless otherwise specified, the word "point" shall be regarded as an infinitesimally small entity which does not occupy space. The word "particle" shall be defined as a small two or three dimensional object.

Underwood [Ref. 6] has described a method for determining the average first nearest neighbor distance, δ, for randomly distributed points. The method is based on the probability of finding a nearest neighbor to a given point within an annulus, of radius r and width dr, centered on the point (Figure. 2). This probability can be expressed as:

$$P(r)\,dr = \left[1 - \int_0^r P(r)\,dr\right] 2\pi N_A dr \qquad (2.1)$$

where N_A is the number of particles per unit area and r is the distance between a point and its first nearest neighbor. The

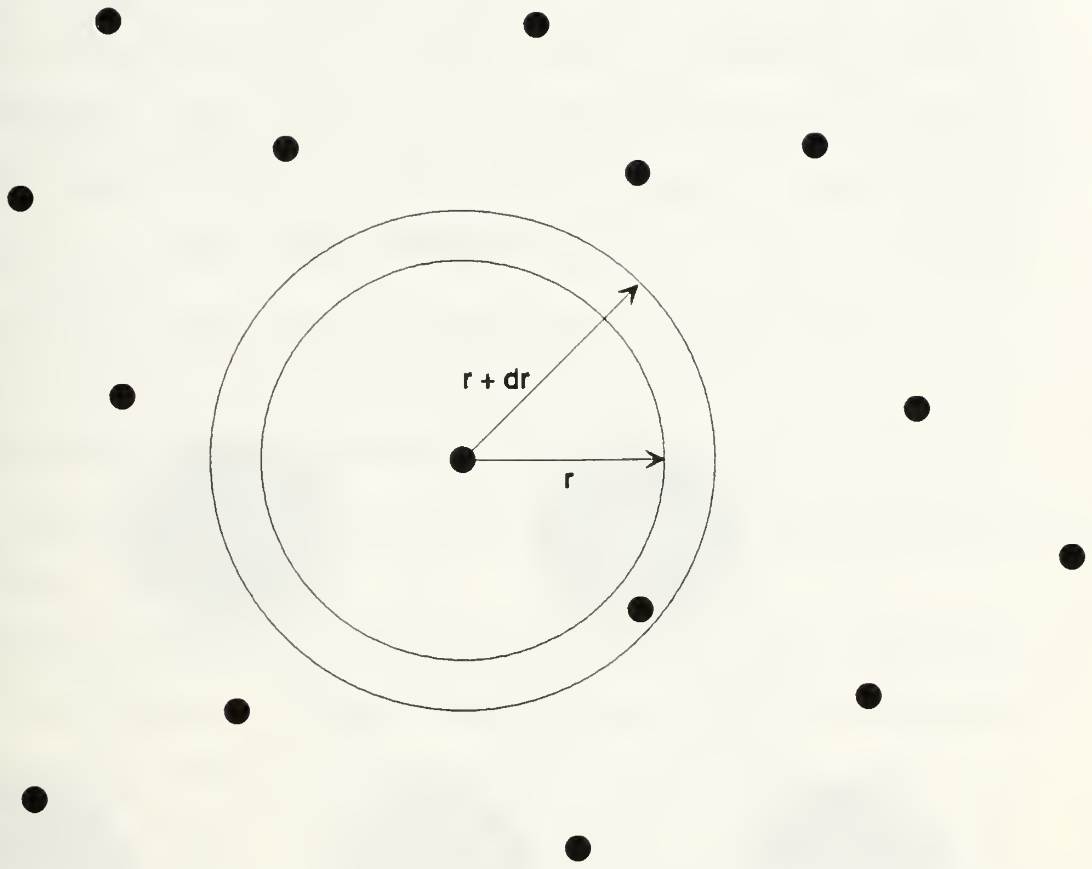


Figure 2. The probability of locating a nearest neighbor to a point may be found by integrating over an annulus centered on the point.

function *P(r)* which satisfies equation (2.1) was shown to be:

$$P(r) = 2\pi r N_A e^{-\pi r^2 N_A} \qquad (2.2)$$

Substituting equation (2.2) into the expression that defines the average first nearest neighbor distance results in the following expression for δ for a random distribution of infinitesimal points:

$$\delta = \int_0^\infty r P(r)\, dr = 0.500 N_A^{-1/2} \qquad (2.3)$$

On the other hand, if particles or points are arranged in a regular hexagonal array, Figure 3, the nearest neighbor distance for each particle will be the same everywhere and will be equal to $\delta_{\mathrm{Hex.}}$ for the array. A geometrical analysis results in the following expression for $\delta_{\mathrm{Hex.}}$:

$$\delta_{Hex.} = 1.075 N_A^{-1/2} \qquad (2.4)$$

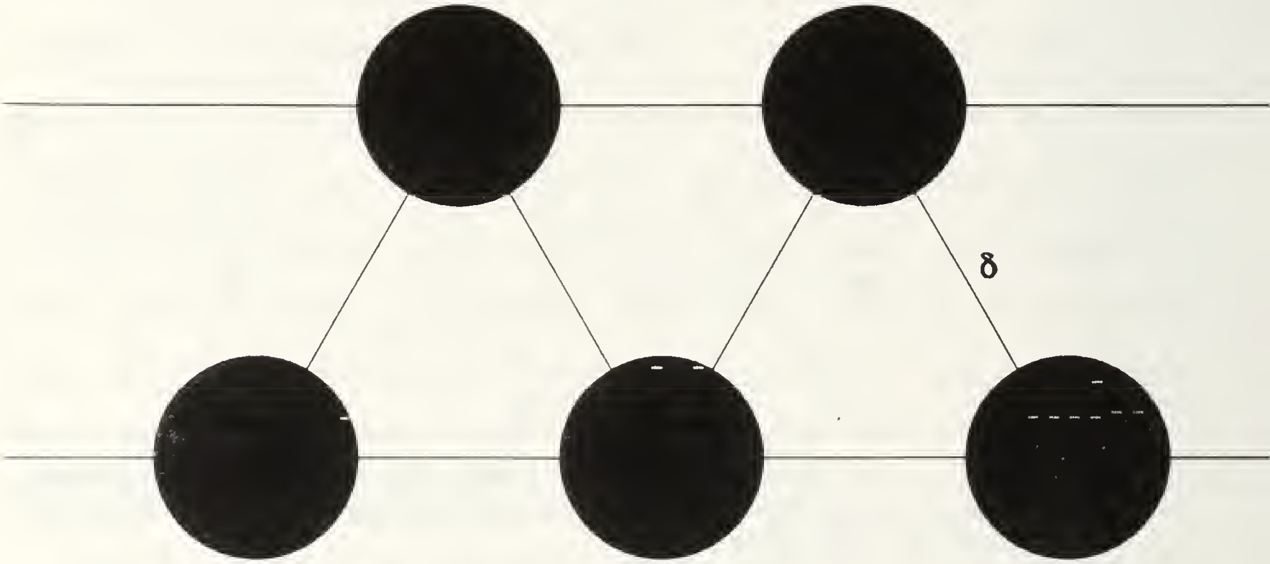


Figure 3. Particles arranged in a hexagonal array.

Based on equations (2.3) and (2.4) it can be seen that, for a given value of N_A, $\delta_{\mathrm{Hex.}}$ is greater than δ for a random distribution.

If particles (or points) are arranged into any regular array the first nearest neighbor distance (NND), *r*, would be the same for each particle and therefore would equal the resulting value of δ for the array. If any one of these particles (or points) were shifted out of its position in the array it would come closer to at least one other particle. Now, there would be at least two particles with a lower first NND, but none with a larger first NND since the remaining particles would still have other neighbors at a distance of r. Therefore, as the particles are shifted out of a perfect geometric array and become more random the value of δ will decrease towards a lower limit defined by the Poisson distribution.

Distributions of finite size particles can not truly be random. The area (or volume) occupied by a particle removes the possibility that another particle can be positioned in a portion of that area. This is the effect of non-overlapping particles. However, for dilute arrays (low area fractions) the possibility of two particles overlapping is small. Therefore, it is reasonable to expect that the value of δ for dilute arrays of finite size particles would in fact approach that of the Poisson distribution.

In previous work, performed by Mike Pas [Ref. 5], attempts to demonstrate that dilute arrays approached the Poisson distribution were not completely successful. Pas plotted N_A vs δ for various area fractions of constant size particles. The plots asymptotically approached values of δ greater than those of the Poisson distribution with decreasing area fraction. However, Pas did show that as the area fraction of particles increases, values for δ shift towards values approaching that of a hexagonal array, equation (2.4). Pas also found that in non-random (high and low density bands of particles) distributions, as the percentage of particles in the high density bands increases, the overall value for δ decreases.

The ultimate goal of this work would be an analytical solution to quantify the degree of randomness of the particle distributions in microstructures. Incorporating this solution with a computer based image analyzing system would make it possible to readily assess the mechanical properties dependent on the particle distributions.

III. EXPERIMENTAL PROCEDURE

A. OVERVIEW

The simulations of particle distributions were generated using a computer. Four independent programs were written, one for each of four cases analyzed: random distribution of constant-size particles (RANPART1); non-random distribution of constant size particles (RANPART2); random distribution of particles with a log-normal size distribution (RANPART3); and non-random distribution of particles with a log normal size distribution (RANPART4). RANPART1, 2, 3 and 4 are included in appendices A, B, C and D respectively.

For the first two cases the algorithms were based on programs originally developed by M. Pas [Ref. 5] and written in TURBO PASCAL. Here, Pas's programs were rewritten in FORTRAN 77 and incorporated several other modifications as deemed necessary. The programs were run either on an IBM-compatible PC or a Digital Equipment Corp. VAX 3100 work station.

B. RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES, RANPART1

The flow chart for RANPART1 is shown in Figure 4. In the input portion of the program the user enters the following parameters:

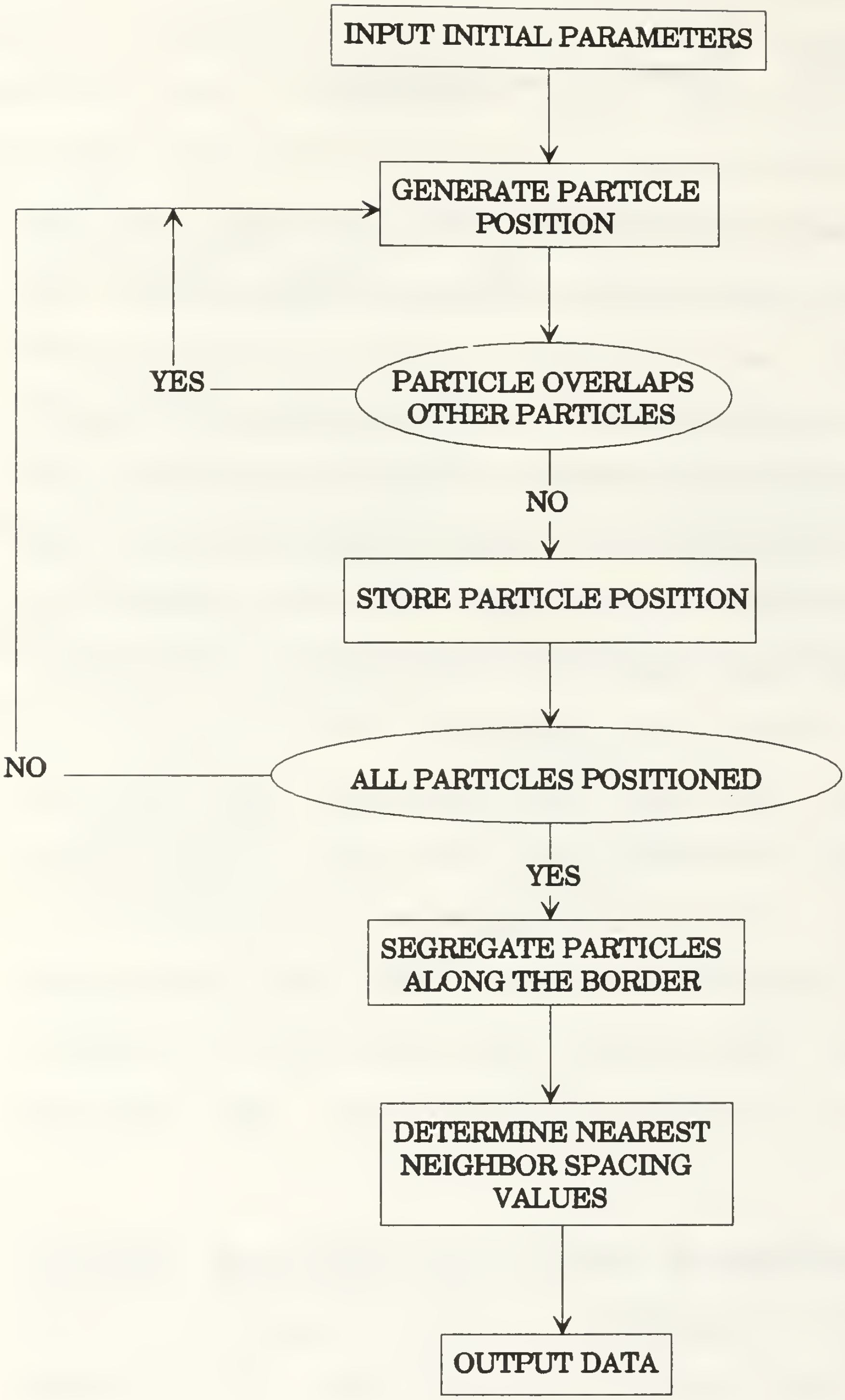

INPUT INITIAL PARAMETERS
GENERATE PARTICLE POSITION
YES
PARTICLE OVERLAPS OTHER PARTICLES
NO
STORE PARTICLE POSITION
NO
ALL PARTICLES POSITIONED
YES
SEGREGATE PARTICLES ALONG THE BORDER
DETERMINE NEAREST NEIGHBOR SPACING VALUES
OUTPUT DATA

Figure 4. Flow chart for the program RANPART1.FOR

- Area percentage of particles for the defined region;
- x dimension of the defined region;
- y dimension of the defined region;
- Minimum spacing between particles (if particles are allowed to come in contact, enter 0);
- Enter "n" if a specified number of particles is required or enter "r" if a specified particles radius is required;
- If entered "n", enter the total number of particles, if entered "r", enter the particles radius;
- Enter "A" if particles are of finite size or enter "p" if the particles are infinitesimal points.

From these parameters the total area of the defined region, total area occupied by the particles, the area per particle, and the total number of particles or particle radius (whichever value was not input) are calculated.

The particles are positioned in the next portion of the program. A pseudorandom number generator is called twice to provide x and y coordinates for a given particle. This particle is then checked to ensure that it does not overlap any other previously sited particle. This is accomplished by calculating the distance to every other particle. If this distance is less than twice the sum of the particle radius and the minimum spacing, the position is rejected and the program loops back to generate a new set of coordinates. This process is repeated until the total number of particles required to establish the desired area fraction has been met.

Once all the particle locations have been established, particles located along the border are segregated from those in the interior of the defined region. This is done so that errors in first nearest neighbor distance (NND) calculations associated with the border will be minimized. The underlying rationale for this step in the program will be discussed in the next chapter. All particle positions are checked and those having coordinates placing them within one particle radius of the border are moved to a separate array in the program memory.

First NND calculations are made by calculating for each <u>interior</u> particle the distance to every other particle, including those located along the border, and then finding and saving the minimum first NND value. This process is repeated for every other interior particle. The average first NND, δ, is calculated by summing the minimum first NND values and dividing by total number of interior particles .

In order to plot the distributions of first NND values an increment had to be determined. To do this, the Sturgis rule was applied as it is an established method for determining the number of class intervals from a set of values. The Sturgis rule is defined by the following expression:

$$N = 1 + 3.3\log(n) \qquad (3.1)$$

Where N is the number of class intervals and n is the number of values in the set. The value of n for this work is equal to the number of interior particles. The number of class intervals was divided into the difference between the maximum and minimum first NND values. This result (defined as "dpinc" in the programs) was used as the incremental range for sorting first NND values into equal sized intervals. The number of NND values in each interval was then divided by the total number of interior particles to get a relative quantity. The output of the program consists of all input parameters, the calculated values described above and a calculated value for N_A, the number of particles per unit area.

RANPART1 was run for area fractions ranging from 1% to 50% utilizing a particle radius of 0.5 units. The defined area was adjusted in each run to maintain a nominal value $N_p \approx 5000$ particles. The minimum spacing was kept at 0 for all simulations.

Each run was repeated using the same parameters, but now declaring the particles to be points. The results consisted of point distribution and a distribution of finite sized particles with the same value of N_A.

For each simulation equation (2.1) was used to plot a Poisson distribution corresponding the simulated distribution of particles and points. The value of N_A came from the simulated point distribution. The probability density function was plotted as a function of r in the range from the minimum and maximum values of first NND derived from the point distribution. The calculated increment Δr, based on the Sturgis rule, was substituted into the equation for dr.

C. NON-RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES, RANPART2

The following modifications were made to RANPART1 to produce RANPART2. The defined area in which the particles are positioned was divided into five sub-areas of equal size. In the input portion of the program the user declares what proportion of the total area occupied by the particles will be assigned to each subsection. The sum of the percentages must equal 100%.

After each particle (for finite sized particles) is checked for overlap, the program determines which sub-area the particle (or point) belongs. By maintaining a running sum on the area occupied by the particles in each sub-area, the computer determines whether the addition of another particle in a given sub-area would cause the area occupied by the particles to exceed the limit. The particle position is rejected if the limit will be exceeded. If this happens the program loops back to generate another pair of particle coordinates. When the process is complete the desired degree of banding has been achieved. Figure 5 is the flow chart for RANPART2.

All runs made for RANPART2 used 10%, 35%, 10%, 35%, and 10% sub-area particle densities. This was considered to provide a relatively large deviation between the high and low density bands. The simulations were made for area fractions ranging from 0.01 to 0.30.

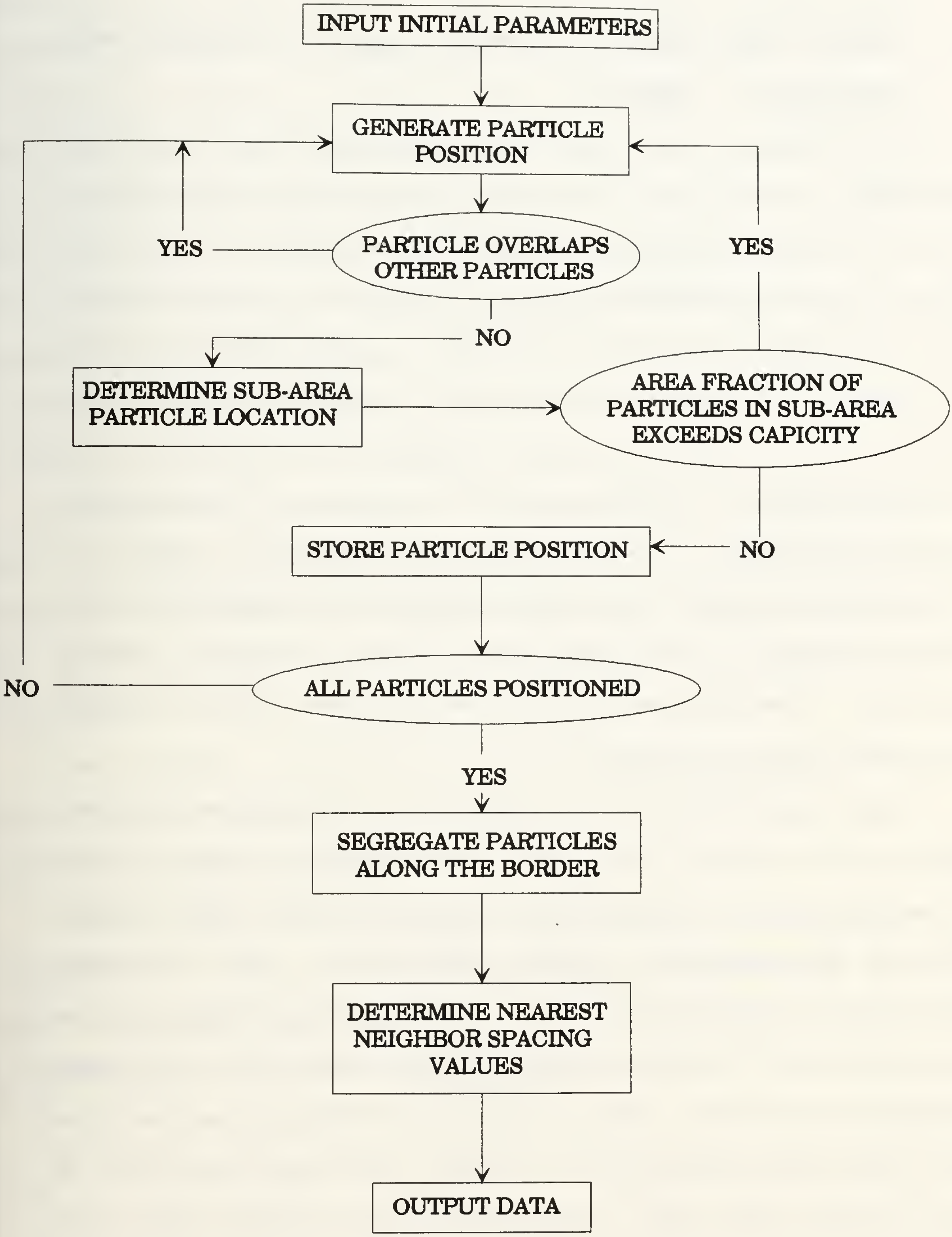

INPUT INITIAL PARAMETERS
GENERATE PARTICLE POSITION
PARTICLE OVERLAPS OTHER PARTICLES
YES
NO
DETERMINE SUB-AREA PARTICLE LOCATION
AREA FRACTION OF PARTICLES IN SUB-AREA EXCEEDS CAPICITY
YES
NO
STORE PARTICLE POSITION
ALL PARTICLES POSITIONED
NO
YES
SEGREGATE PARTICLES ALONG THE BORDER
DETERMINE NEAREST NEIGHBOR SPACING VALUES
OUTPUT DATA

Figure 5. Flow chart for the program RANPART2.FOR

D. RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION, RANPART3

In RANPART3 a second random number generator is used to determine a particle radius from a lognormal distribution. The call for the particle radius is made before the call for the particle coordinates so that, if particle overlap occurs, only a new position will be generated but the existing radius will still be used. Only when a particle is finally positioned or, if particle overlap occurs 600 times consecutively for the same particle radius, is a new particle radius generated. This process is required in order to maintain a lognormal size distribution and to prevent biasing towards smaller sized particles as the particle density increases. The program must now use individual particle radii in subsequent calculations.

The input portion of the program was modified as follows. The user must enter a value approximately equal to the natural logarithm of the mean particle radius desired, which is the mean of the underlying normal distribution, and a standard deviation. Several iterations of these two variables may be required to get the desired mean radius, especially for larger standard deviations. Since a point distribution derived from the same input parameters of constant size particles would be no different than that for particles of varying size, this option is not available for the programs with a lognormal particle size distribution. The total number of particles entered in the input portion of the program must be of sufficient quantity to ensure that the number of interior particles generated is enough to meet the desired area fraction.) Since now the particles are no longer of constant size there is no direct relationship between the number of particles and the total area occupied by the particles. A running sum of calculated particle areas is maintained. Particle generation continues until this sum is equal to the value corresponding to the desired area fraction.

Segregation of particles along the border is performed much the same; but here the largest particle radius in the distribution is used to establish the extent of the border. Any particle whose center lies outside this border is segregated.

The maximum, minimum and mean particle radius are determined and provided at the output. The flow chart for RANPART3 is shown in Figure 6.

RANPART3 was run for the same area fractions as RANPART1. Values of -3.07 and 0.4 were used as input for the mean of the underlying normal distribution and for the standard deviation, respectively.

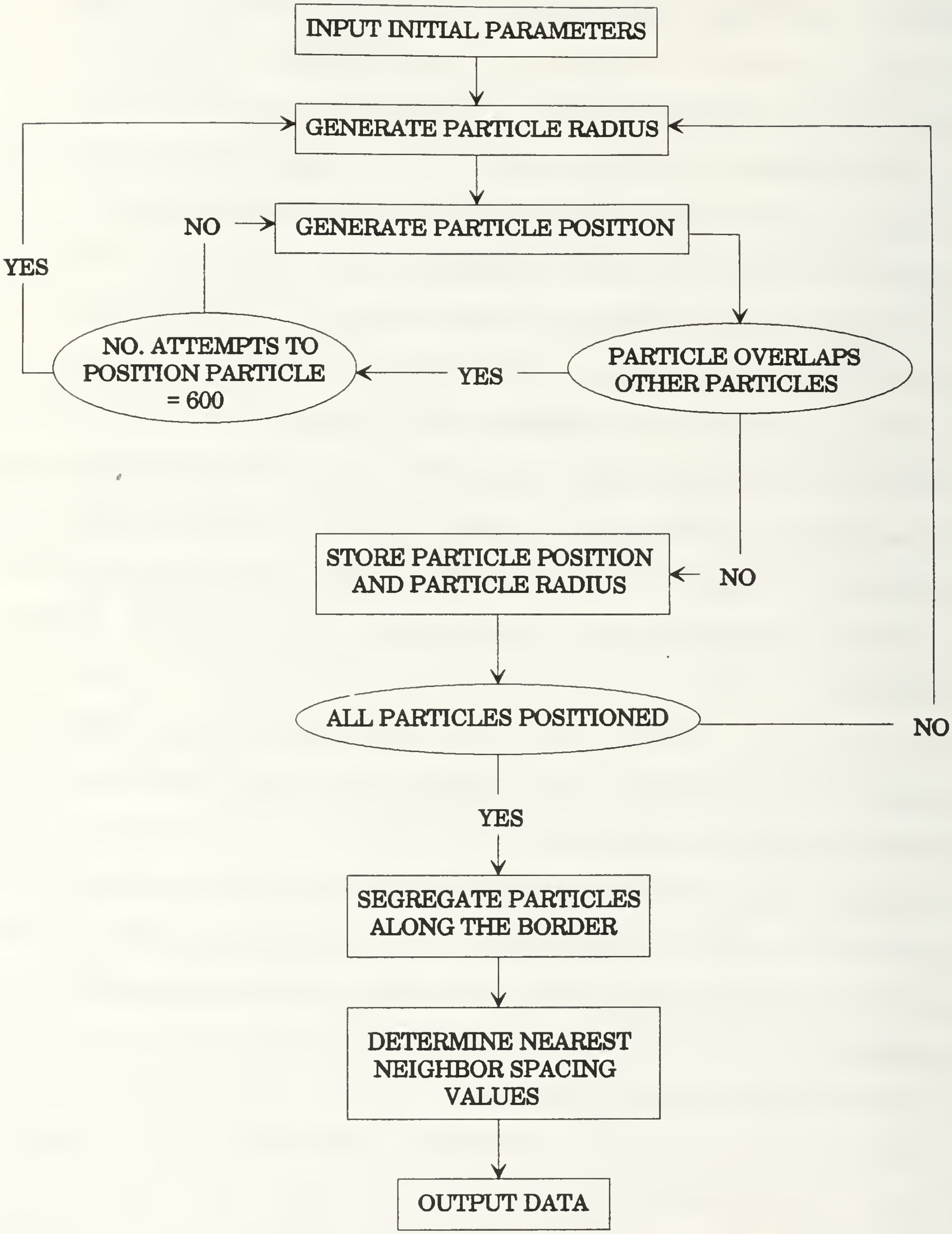

INPUT INITIAL PARAMETERS
GENERATE PARTICLE RADIUS
GENERATE PARTICLE POSITION
NO
YES
NO. ATTEMPTS TO POSITION PARTICLE = 600
YES
PARTICLE OVERLAPS OTHER PARTICLES
STORE PARTICLE POSITION AND PARTICLE RADIUS
NO
ALL PARTICLES POSITIONED
NO
YES
SEGREGATE PARTICLES ALONG THE BORDER
DETERMINE NEAREST NEIGHBOR SPACING VALUES
OUTPUT DATA

Figure 6. Flow chart for the program RANPART3.FOR

E. NON-RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION, RANPART4

RANPART4 was derived from RANPART3 in the same way RANPART2 was derived from RANPART1. Five sub-areas were incorporated into RANPART4 just as they were incorporated into RANPART2. To maintain a record of the area occupied by the particles in each sub-area, the area of each new particle needs to be calculated and added to the existing area in which it lies.

The flow chart for RANPART4 is shown in Figure 7. RANPART4 was run for the same area fractions as RANPART2 and for the same values of mean and standard deviation as RANPART3.

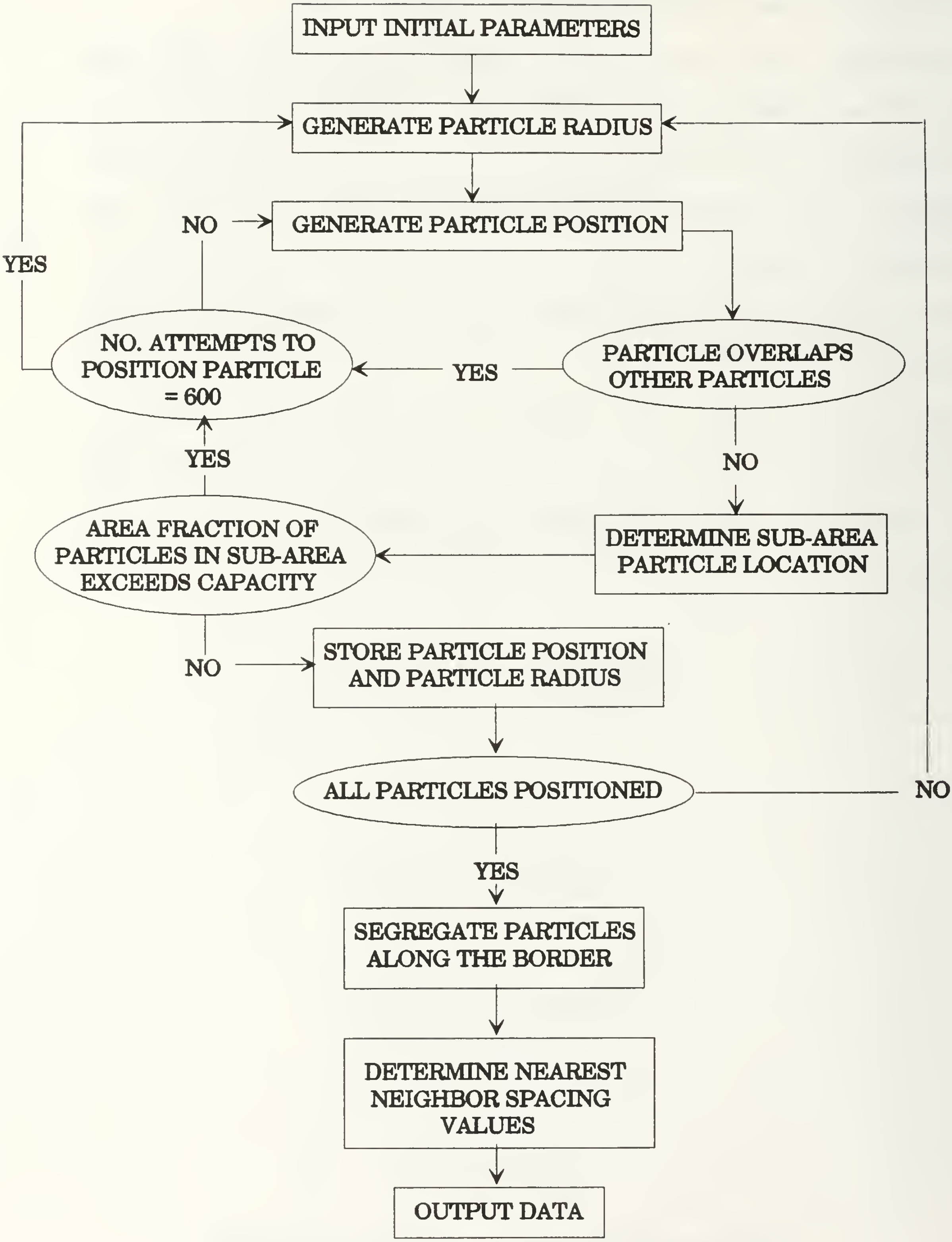

INPUT INITIAL PARAMETERS
GENERATE PARTICLE RADIUS
NO
GENERATE PARTICLE POSITION
YES
NO. ATTEMPTS TO POSITION PARTICLE = 600
YES
PARTICLE OVERLAPS OTHER PARTICLES
YES
NO
AREA FRACTION OF PARTICLES IN SUB-AREA EXCEEDS CAPACITY
DETERMINE SUB-AREA PARTICLE LOCATION
NO
STORE PARTICLE POSITION AND PARTICLE RADIUS
ALL PARTICLES POSITIONED
NO
YES
SEGREGATE PARTICLES ALONG THE BORDER
DETERMINE NEAREST NEIGHBOR SPACING VALUES
OUTPUT DATA

Figure 7. Flow chart for the program RANPART4.FOR

IV. RESULTS AND DISCUSSION

A. OVERVIEW

The first phase of this work consisted of a study to determine a statistically sufficient number of particles to be used in the simulations. The problem of particles positioned along the border of the array was also addressed in this phase. In the second phase, four general cases were studied: random distribution of points and finite sized particles of constant size; non-random distribution of points and particles of constant size; random distribution of particles with a log normal size distribution; and finally, non-random distribution of particles with a log normal size distribution.

B. DETERMINATION OF A SUFFICIENT NUMBER OF PARTICLES AND THE PROBLEM OF PARTICLES POSITIONED ALONG THE BORDER

In order to determine a statistically sufficient number of particles to be used in the simulations the program RANPART1 was modified. The modifications essentially consisted of enclosing RANPART1 inside two loops. This allowed repeated program running for the same parameters. Then, the average of the results of repeated runs, was calculated. This experiment was used to calculate an average value of δ, over a given number of runs, for various numbers of particles (N_p) , holding N_A constant. The test was conducted for two area fractions, 0.10 and 0.40 and for point distributions as well. N_A was arbitrarily set equal to 100. The results of the tests are shown in Table 1 and in Figure 8.

N_p	a_f = 0.1	a_f = 0.4	Points	No. Runs
10	0.06801	0.08902	0.05781	1000
20	0.06562	0.08633	0.05556	1000
30	0.06442	0.08524	0.05425	1000
40	0.06376	0.08455	0.05362	1000
50	0.06349	0.08434	0.05326	1000
60	0.06349	0.08400	0.05282	500
70	0.06282	0.08370	0.05271	500
60	0.06281	0.08352	0.05257	500
60	0.06281	0.06349	0.05240	500
800	0.06233	0.08326	0.05240	100
200	0.06173	0.08255	0.05146	100
400	0.06154	0.08223	0.05131	100
400	0.06127	0.08211	0.05106	100
500	0.06127	0.08198	0.05107	50
600	0.06107	0.08195	0.05083	50
700	0.06099	0.08198	0.05098	25
800	0.06091	0.08172	0.05046	25
900	0.06087	0.08172	0.05087	25
1000	0.06076	0.06173	0.05075	5
2000	0.06099	0.08198	0.05025	5
3000	0.06035	0.08434	0.05059	5
4000	0.06048	0.08434	0.05035	5
5000	0.06041	0.08127	0.05030	5

Table 1. Results from the modified RANPART1 for area fractions of 0.1 and 0.4 and for points, N_A = 100.

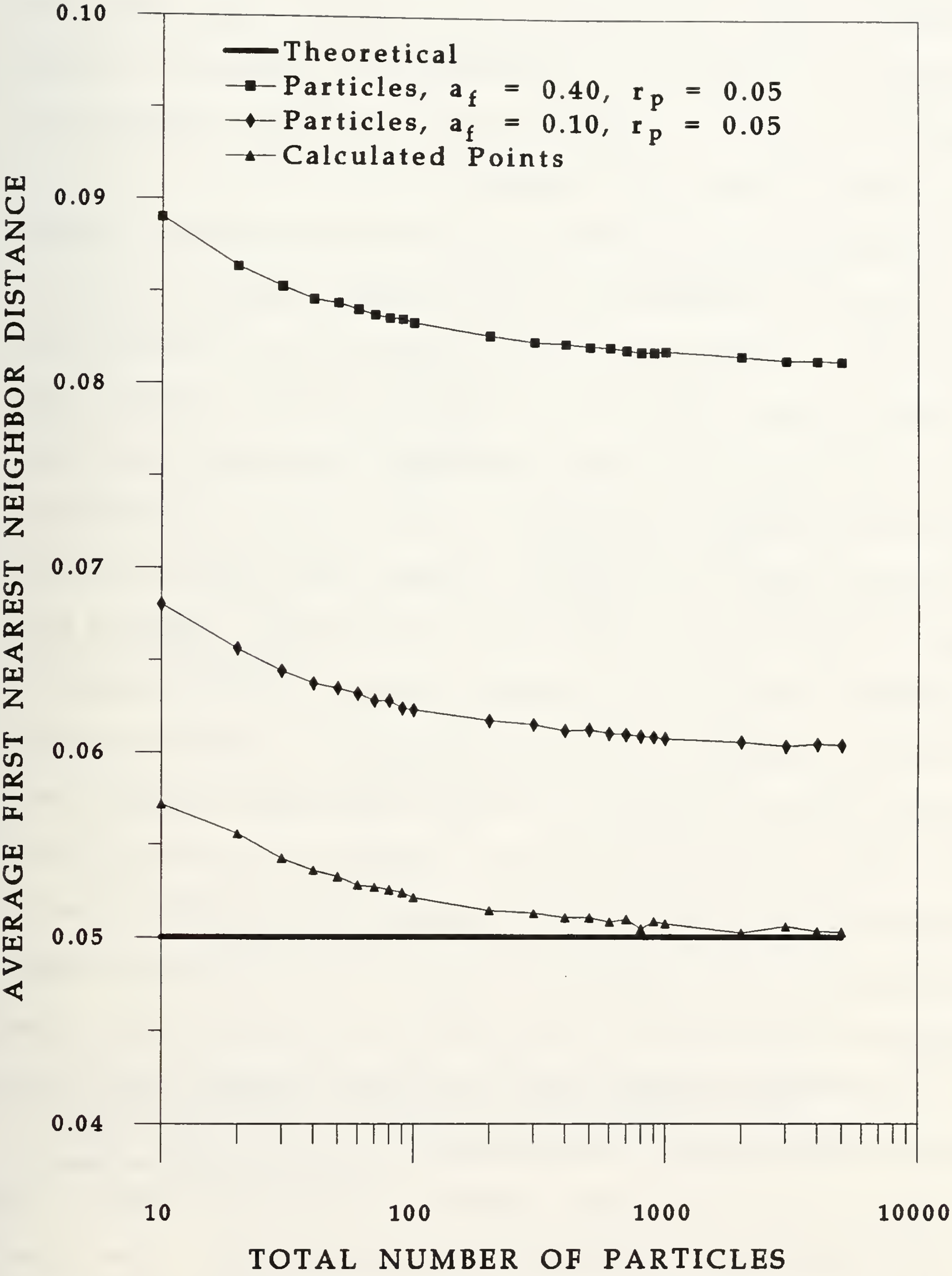


Figure 8. Total number of particles vs average first nearest neighbor distance for N_A = 100

For the random distribution of points, δ approaches the theoretical value of 0.05 for $N_P \geq 1000$. For both area fractions, 0.40 and 0.10, δ approaches and appears to remain constant at values of about 0.081 and 0.060 respectively, again at $Np \geq 1000$ in both cases.

In developing the method for positioning circles or points on a plane the dimensions of the planar surface had to be established. Within this finite sized area all the particles were positioned. From this situation it became evident that when calculating first nearest neighbor distances, a particle whose center lies on or near the border of the defined region could conceivably have a closer neighbor than that calculated. Figure 9 shows a defined area and a particle, *a*, located near the border. Particle *a*'s nearest neighbor within the area is particle *b*. However, it is possible that if the defined area had been larger, particle *a* may have a nearest neighbor even closer than *b* such as particle *c*. Clearly this would be the situation in any real material (imagine the field of view of a micrograph being expanded). Since the existence of these additional particles could only serve to lower the first NND, any calculations made for δ without taking this into considerations would be overestimated.

Calculations for δ were repeated, holding N_A constant and for the same area fractions, as described above. The model was modified such that first NND calculations were not made

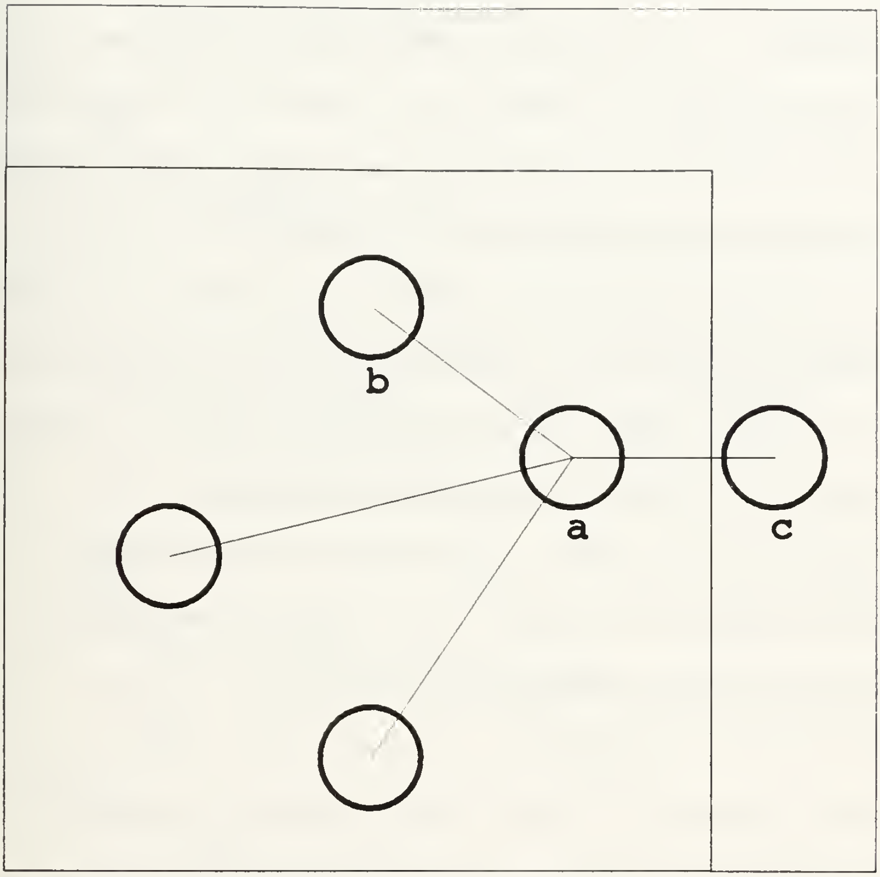


Figure 9. The border problem. Particles positioned along the border of a defined area (a) may have a potential nearest neighbor (c), if the defined region was expanded, which is closer than the nearest neighbor positioned inside the area (b).

for particles whose centers fell within one particle radius from the border. These border particles were used in determining first NND for particles located inside this exclusion range. The results are shown in Figure 10.

This attempt to eliminate the border problem was only partially successful as shown in Figure 10 and has the greatest effect for low numbers of particles. The values for δ were consistently lower for calculations involving both particles and points when the numbers of points or particles utilized was below 1000. For N_P greater than 1000 the effects of using a larger number of particles or points compensates for the border problem by simply increasing the relative number of particles located inside the defined area.

The data compiled in Table 1 was used to determine a more exact solution for the asymptotic value of δ (as Np → ∞) and show that it is a function of the $\delta_{Theo.}$ and N_p. Close examination of Figure 8 reveals the similar form of all three curves. That is, for any value of N_p the difference between $\delta_{Obs.}$ and $\delta_{Theo.}$ (for points) or $\delta_{Obs.}$ and $\delta_{Asym.}$ (for particles) appears to be the same. Based on the previous observation that this difference is due to the particles positioned along the border, the following relationship was defined:

$$\frac{N_{pB}}{N_p}\Delta\delta = \delta_{Obs.} - \delta_{Theo.} \qquad (4.1)$$

where N_{pB} is the number of particles along the border and $\Delta\delta$

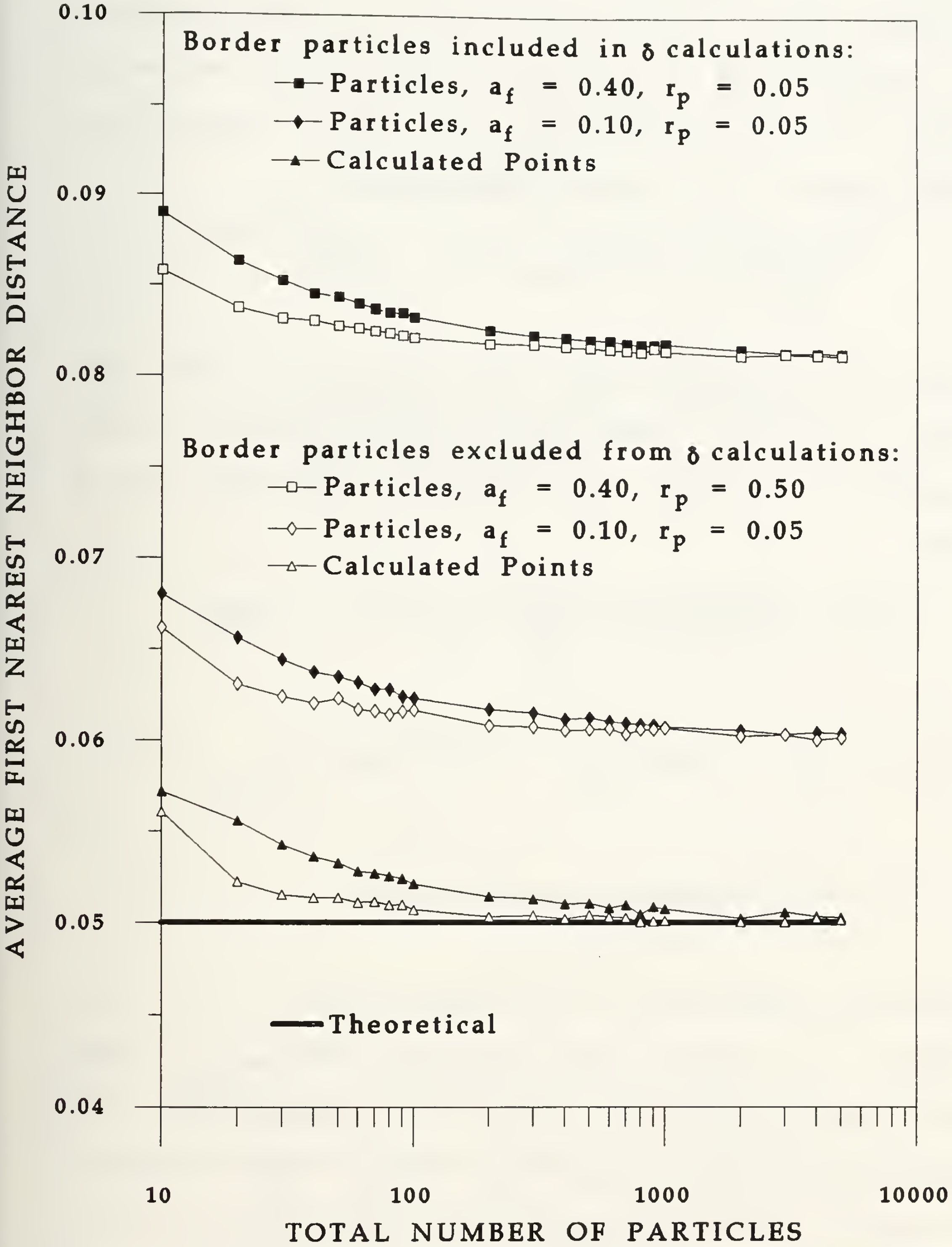


Figure 10. Total number of particles vs average first nearest neighbor distance for N_A = 100. The three pairs of curves demonstrate the affect of particles located on the border.

is the overestimate in δ associated with each particle along the border. For a random distribution of particles in a square region, N_p can be taken to be equal to the square of the number of particles along the border, N_{Pb}^2. Therefore $N_{Pb} = N_p^{1/2}$ and equation (4.1) can be expressed as:

$$\frac{1}{\sqrt{N_p}}\Delta\delta = \delta_{Obs.} - \delta_{Theo.} \qquad (4.2)$$

Figure 11 is a plot of $N_p^{-1/2}$ vs $\delta_{Obs.} - \delta_{Theo.}$ from the data compiled in Table 1. These curves clearly define a linear relationship from which the following equations can be derived:

$$Points: \quad \delta_{Obs.} = 0.0235\frac{1}{\sqrt{N_p}} + \delta_{Theo.} \qquad (4.3)$$

$$a_f = 0.1: \quad \delta_{Obs.} = 0.0248\frac{1}{\sqrt{N_p}} + 0.0100 + \delta_{Theo.} \qquad (4.4)$$

$$a_f = 0.4: \quad \delta_{Obs.} = 0.0248\frac{1}{\sqrt{N_p}} + 0.0309 + \delta_{Theo.} \qquad (4.5)$$

By taking the limit as $N_p \rightarrow \infty$ in each equation (4.3), (4.4) and (4.5), $\delta_{Obs.} \rightarrow \delta_{Asym.}$. The solutions for $\delta_{Asym.}$ in these equations are 0.0500, 0.0600 and 0.0809, respectively. In general, this value for $\delta_{Asym.}$ can be taken to be the average first nearest neighbor distance for any distribution of points or particles of finite size.

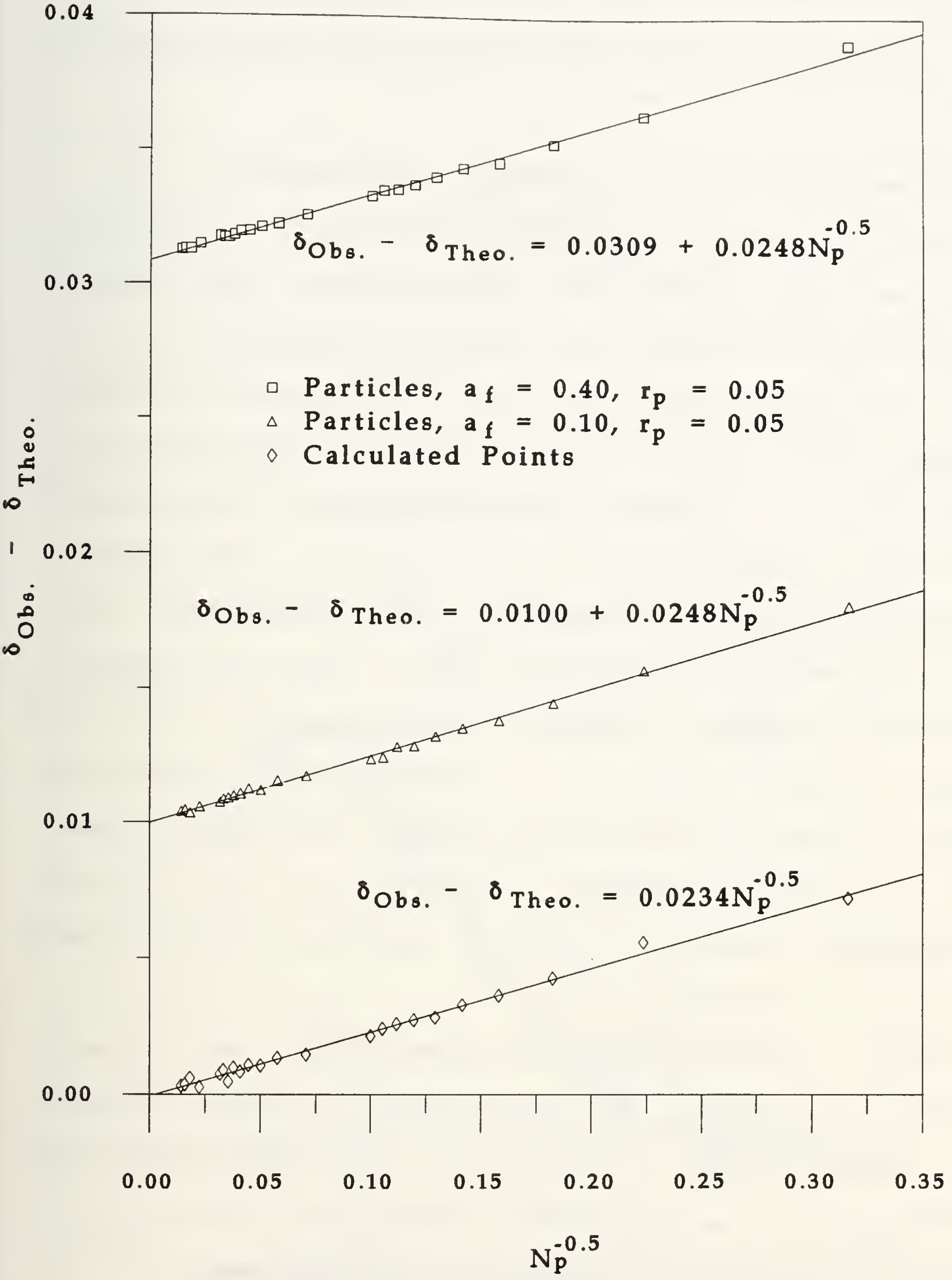


Figure 11. $N_p^{-0.5}$ vs $\delta_{Obs.} - \delta_{Theo.}$ for area fractions of 0.10 and 0.40.

Based on the results above all subsequent simulations were made using both the border particle exclusion algorithm and a nominal value of 5000 particles or points.

C. RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES

For all values of N_A evaluated, $1.275 \leq N_A \leq 63.82$, the calculated distribution for random points very closely approximated the corresponding Poisson distribution (Figure 12 -16). For area fractions less than 0.05, as shown in Figure 12, the particle distributions generated approximated the Poisson as well. In these dilute distributions the particles behave like points since the likelihood of two randomly generated particles overlapping is not significant. The difference between the distributions is that the particle distribution becomes truncated at a value equal to twice the particle radius. This was a result of the non-overlap criterion preventing the particle centers from getting closer than one particle diameter. Figures 17 - 21 are the corresponding plots of the actual particle distributions represented in Figures 12 - 16.

Figure 22 demonstrates the effect that, as the area fraction is increased, the distribution is shifted towards lower values of NND. A shift in δ from the theoretical Poisson value ($\delta_{Theo.} = 0.5N_A^{-1/2}$) towards the value for a hexagonal array ($\delta_{Hex.} = 1.075N_A^{-1/2}$) was also observed. At high area fractions the distributions for particles approach delta

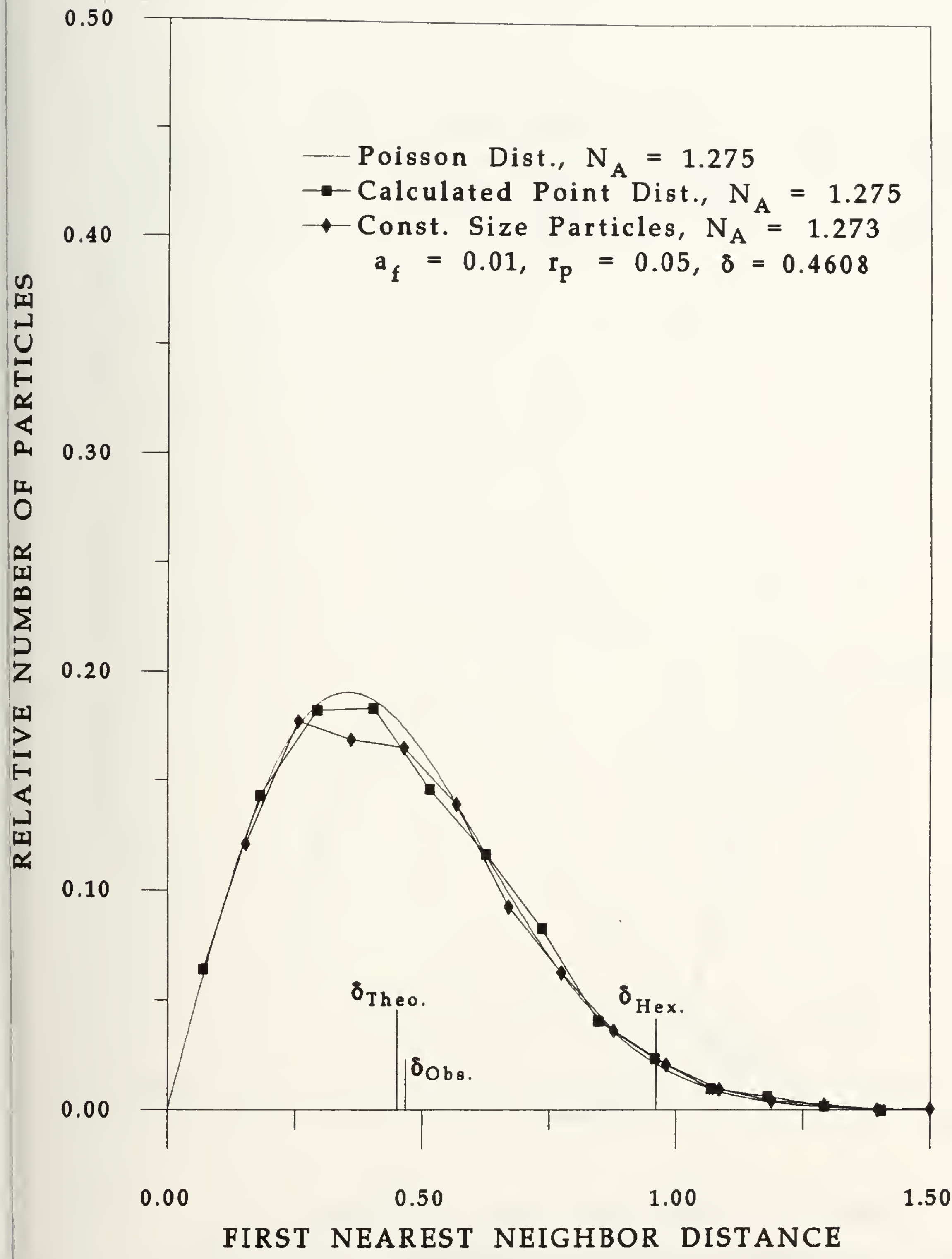


Figure 12. Random distribution of points and constant size particles.

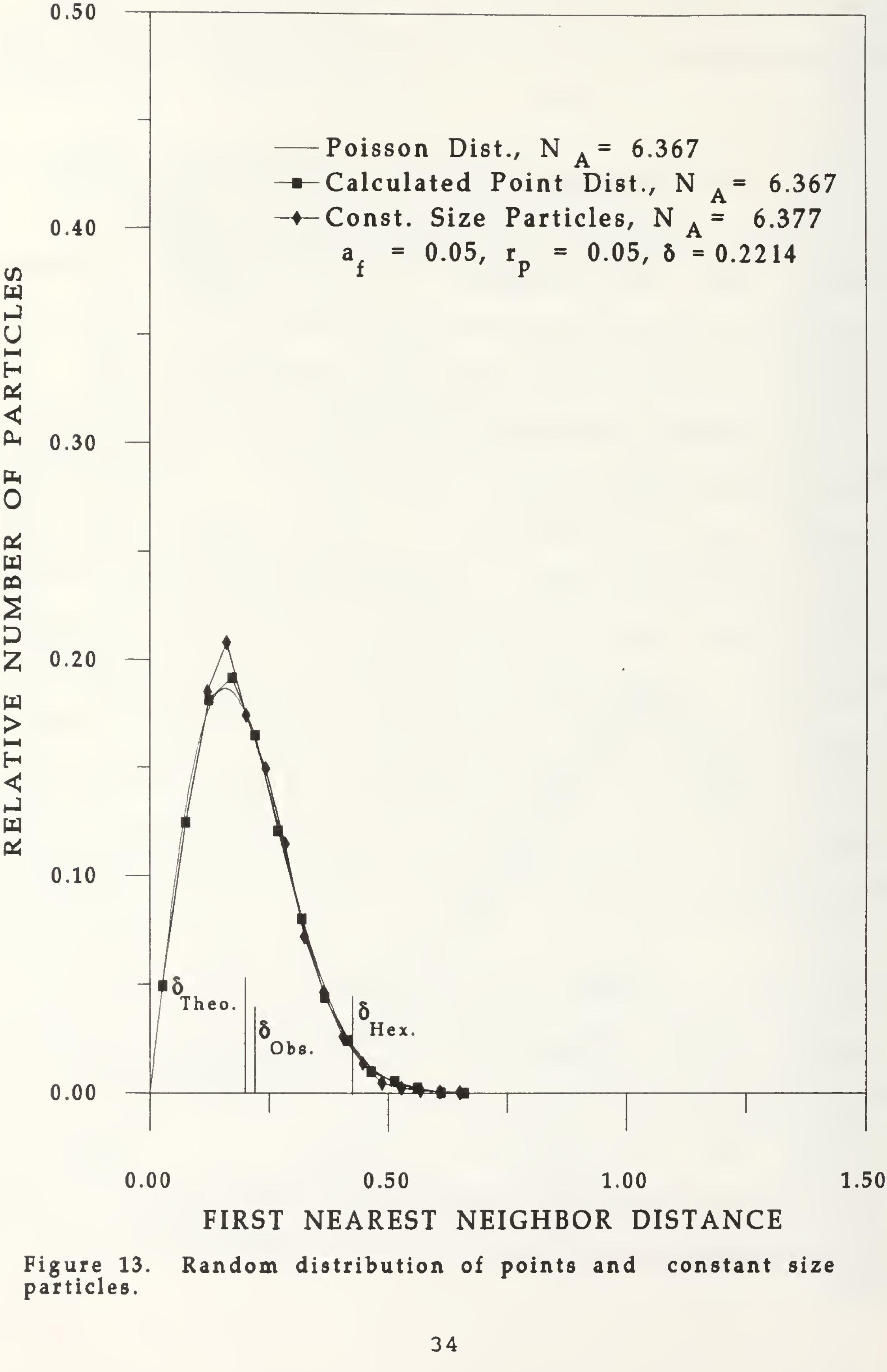


Figure 13. Random distribution of points and constant size particles.

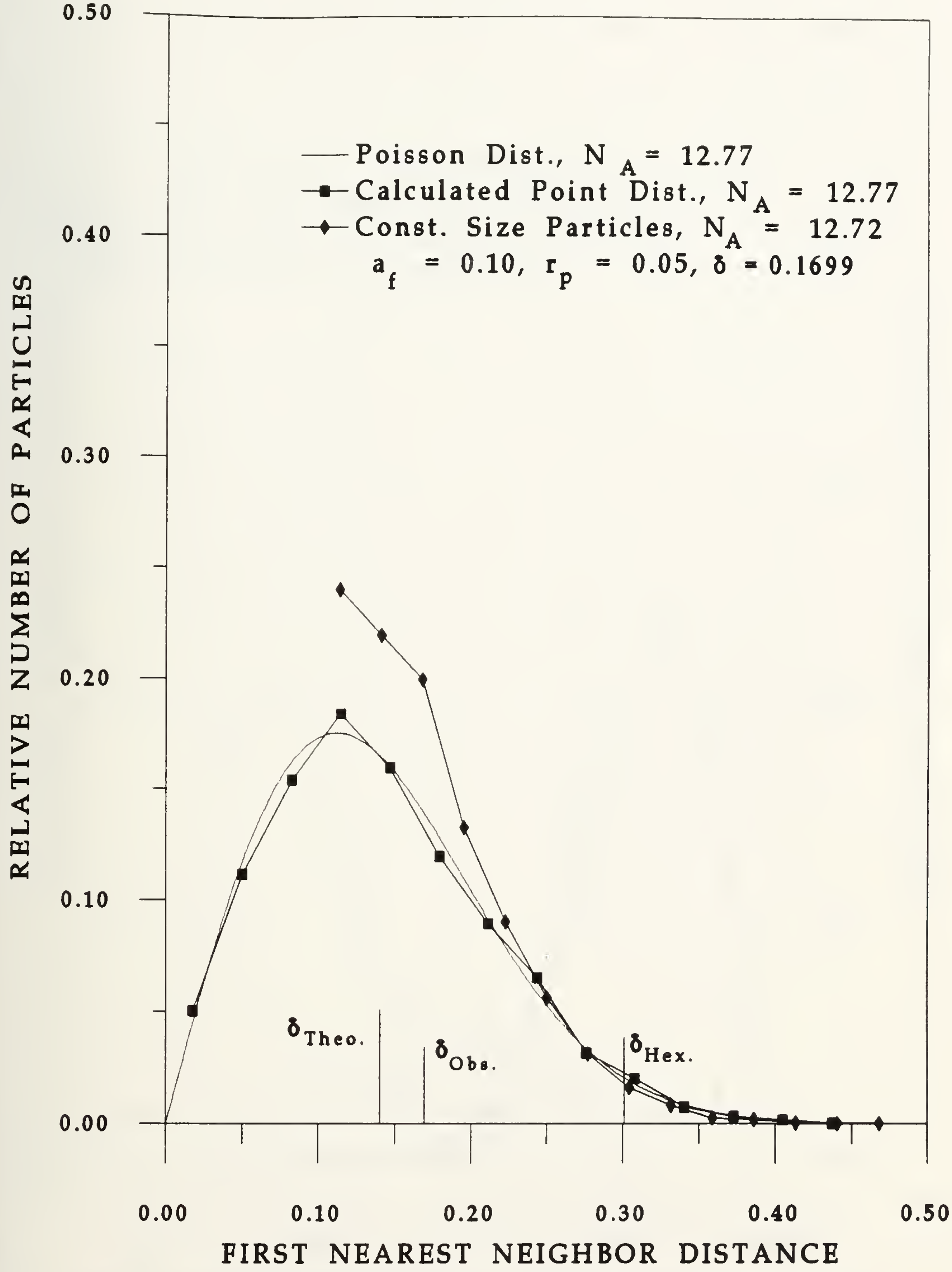


Figure 14. Random distribution of points and constant size particles.

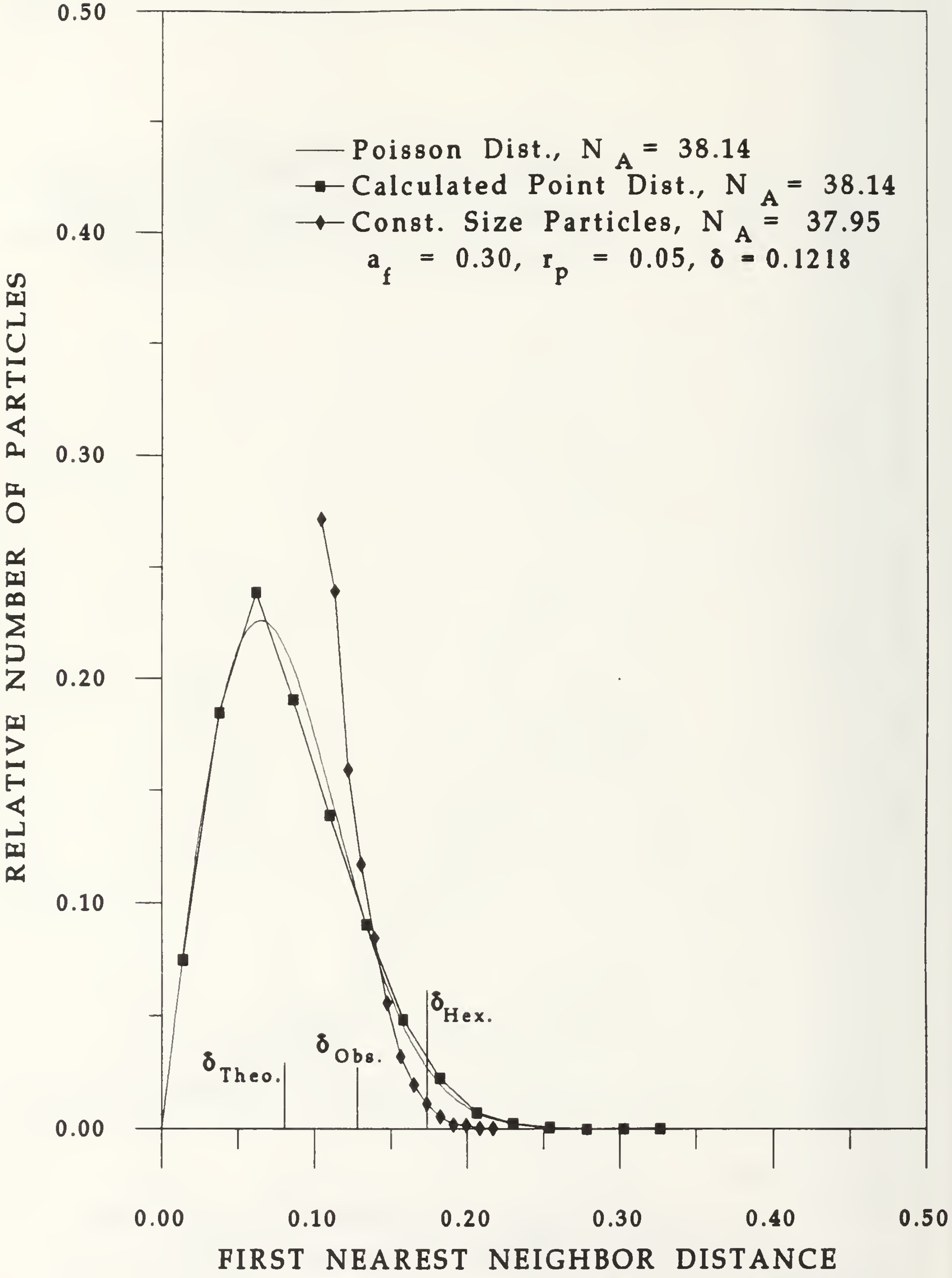


Figure 15. Random distribution of points and constant size particles.

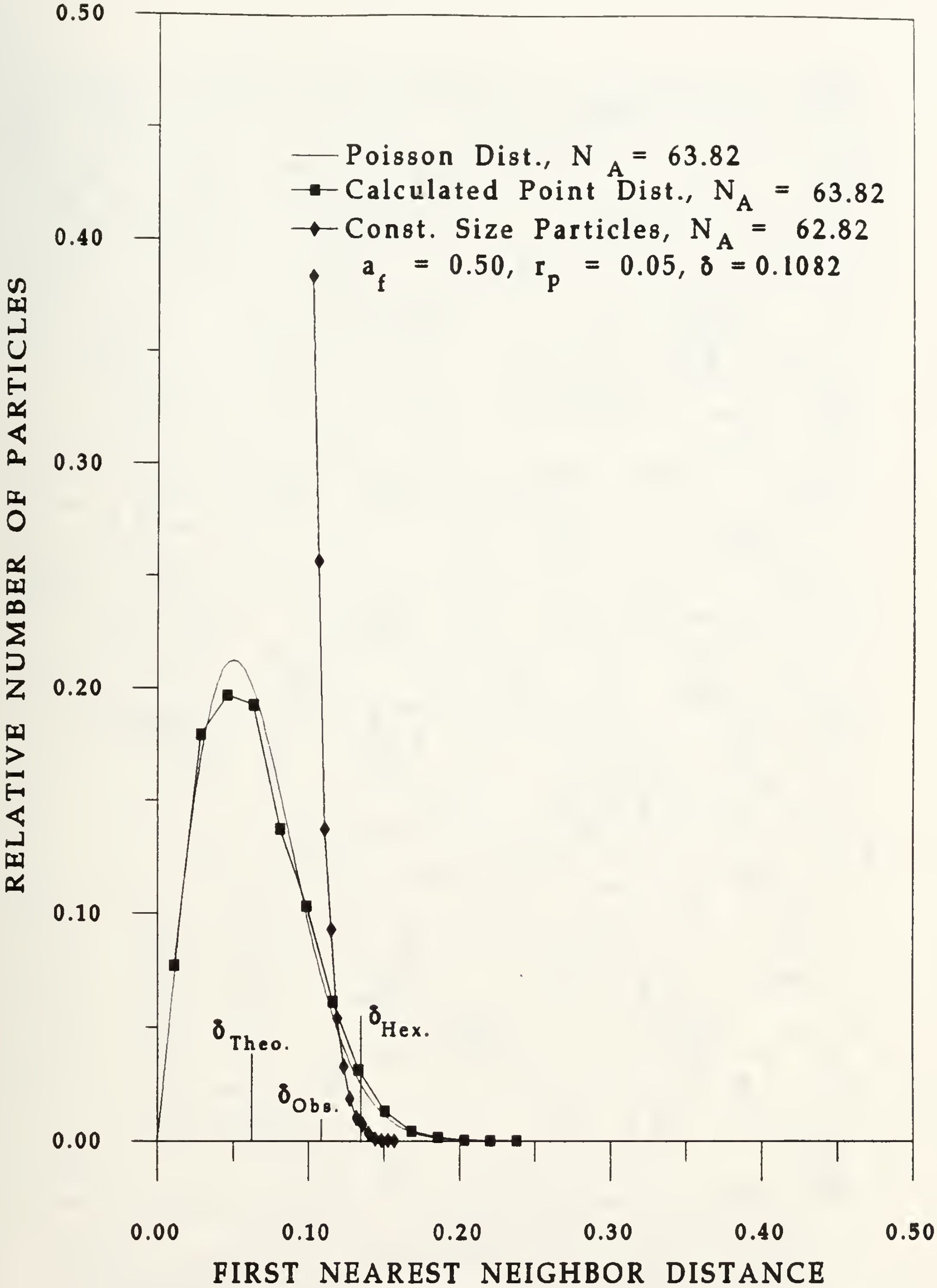
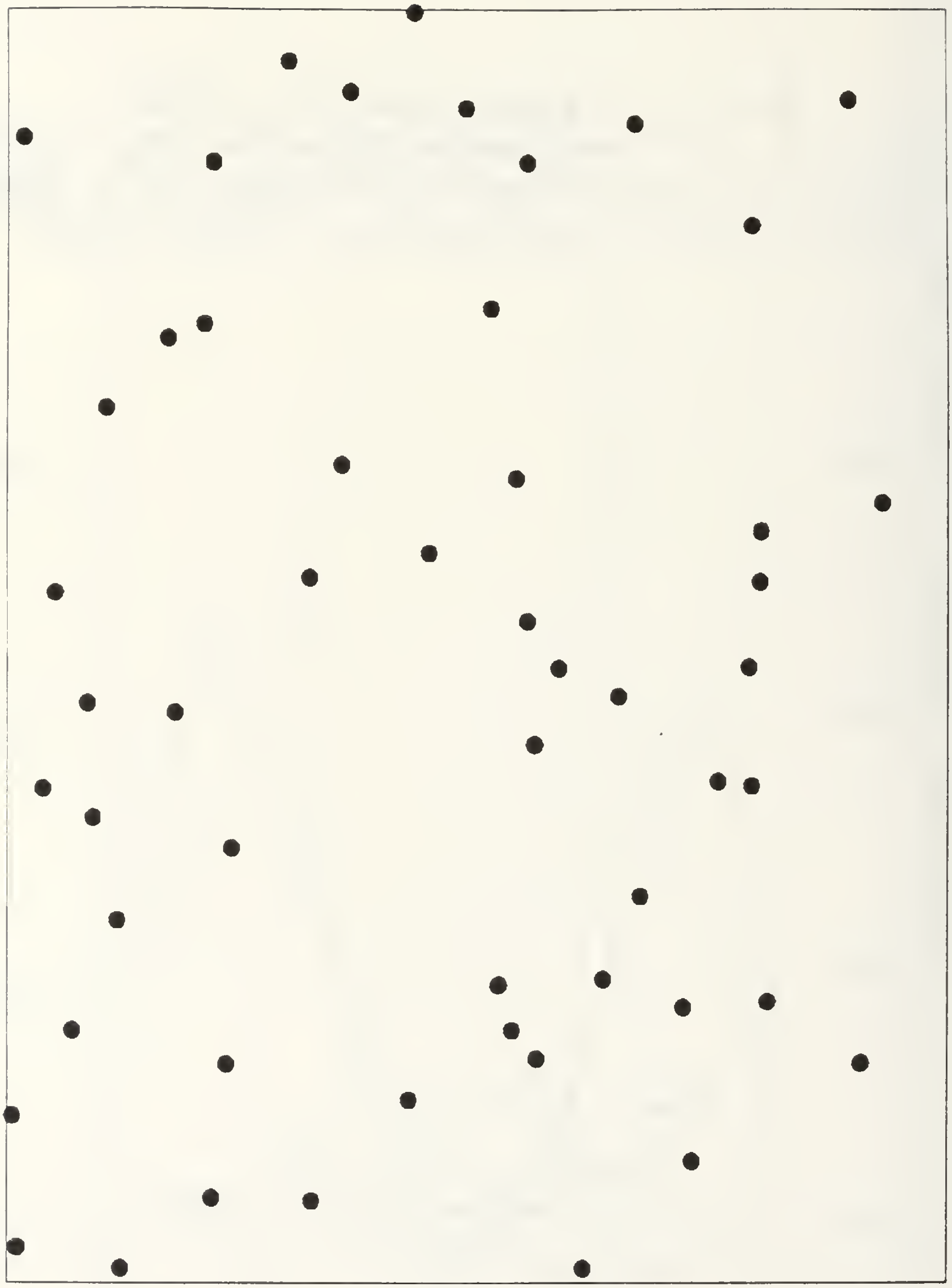
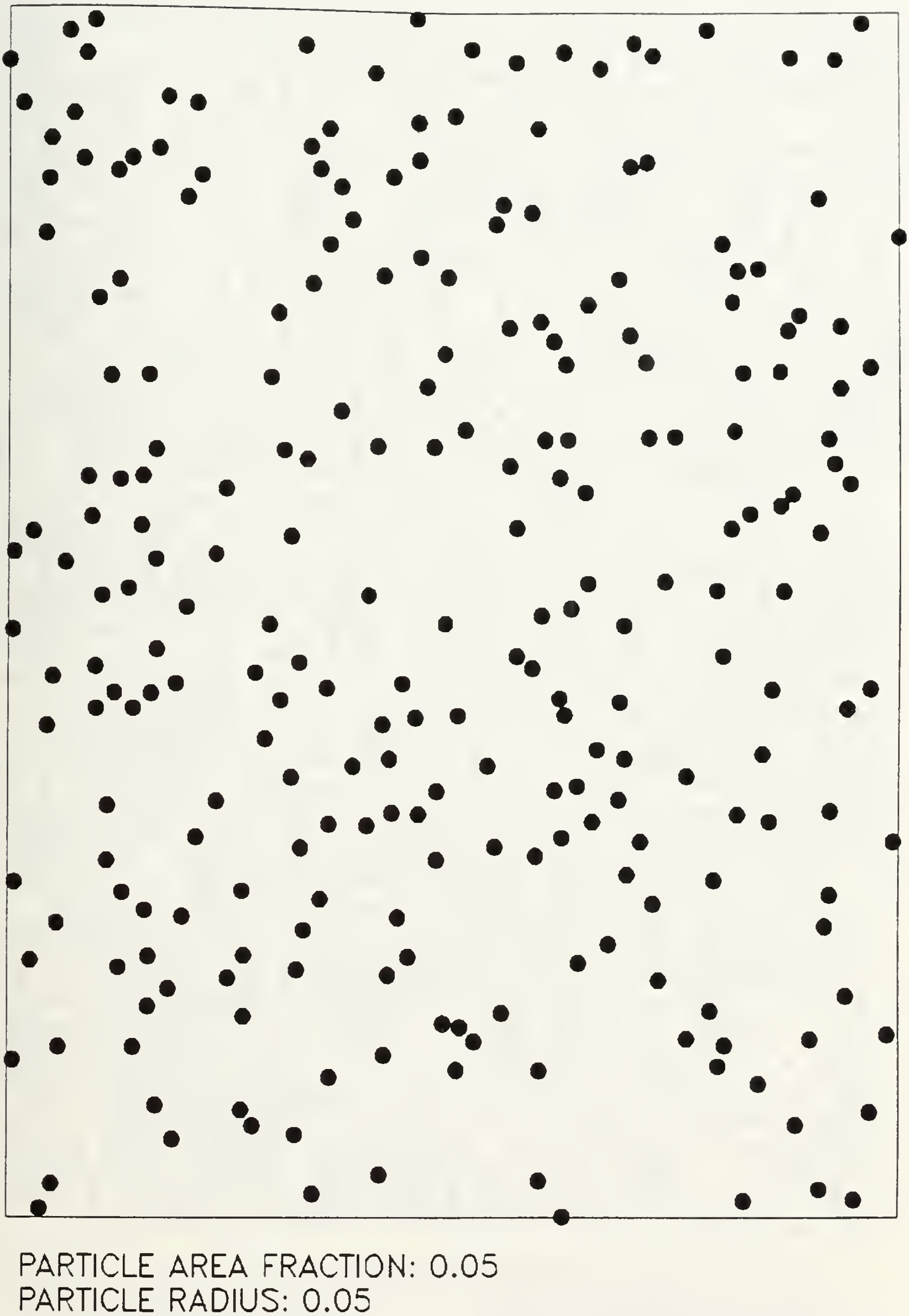


Figure 16. Random distribution of points and constant size particles.



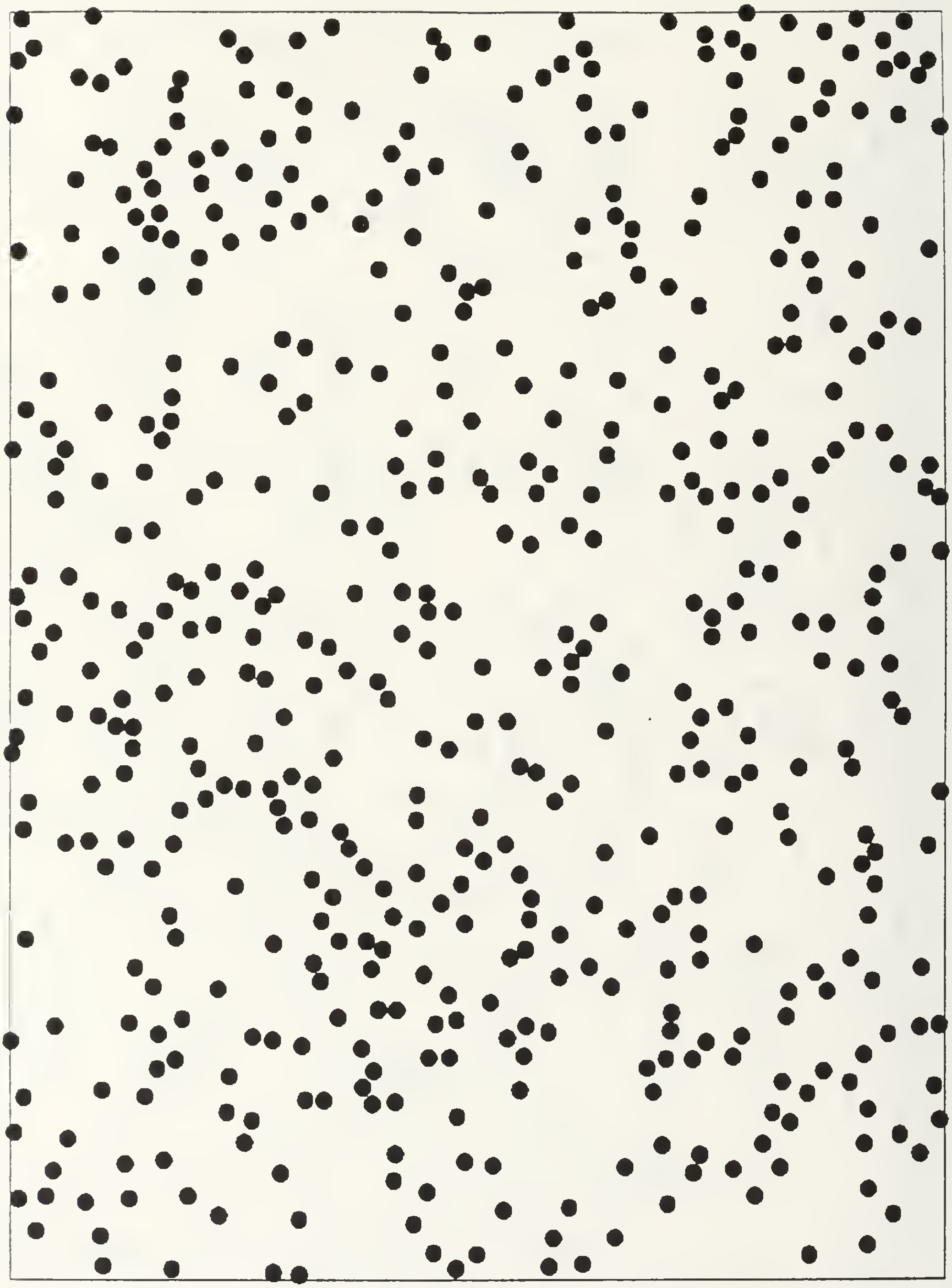
PARTICLE AREA FRACTION: 0.01
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.4608
NUMBER PARTICLES PER UNIT AREA: 1.273

Figure 17. Random distribution of constant size particles.



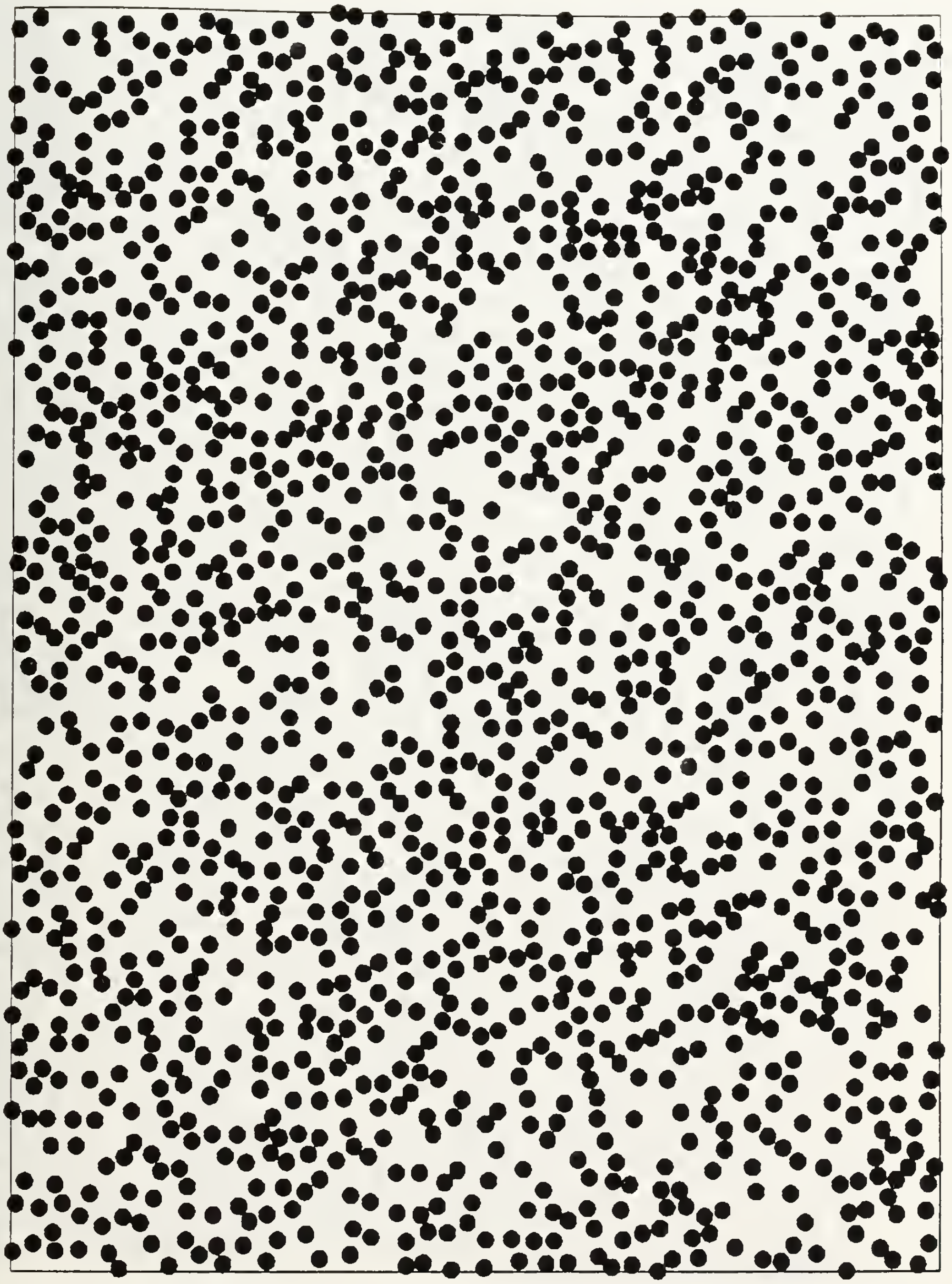
PARTICLE AREA FRACTION: 0.05
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.2214
NUMBER PARTICLES PER UNIT AREA: 6.377

Figure 18. Random distribution of constant size particles.



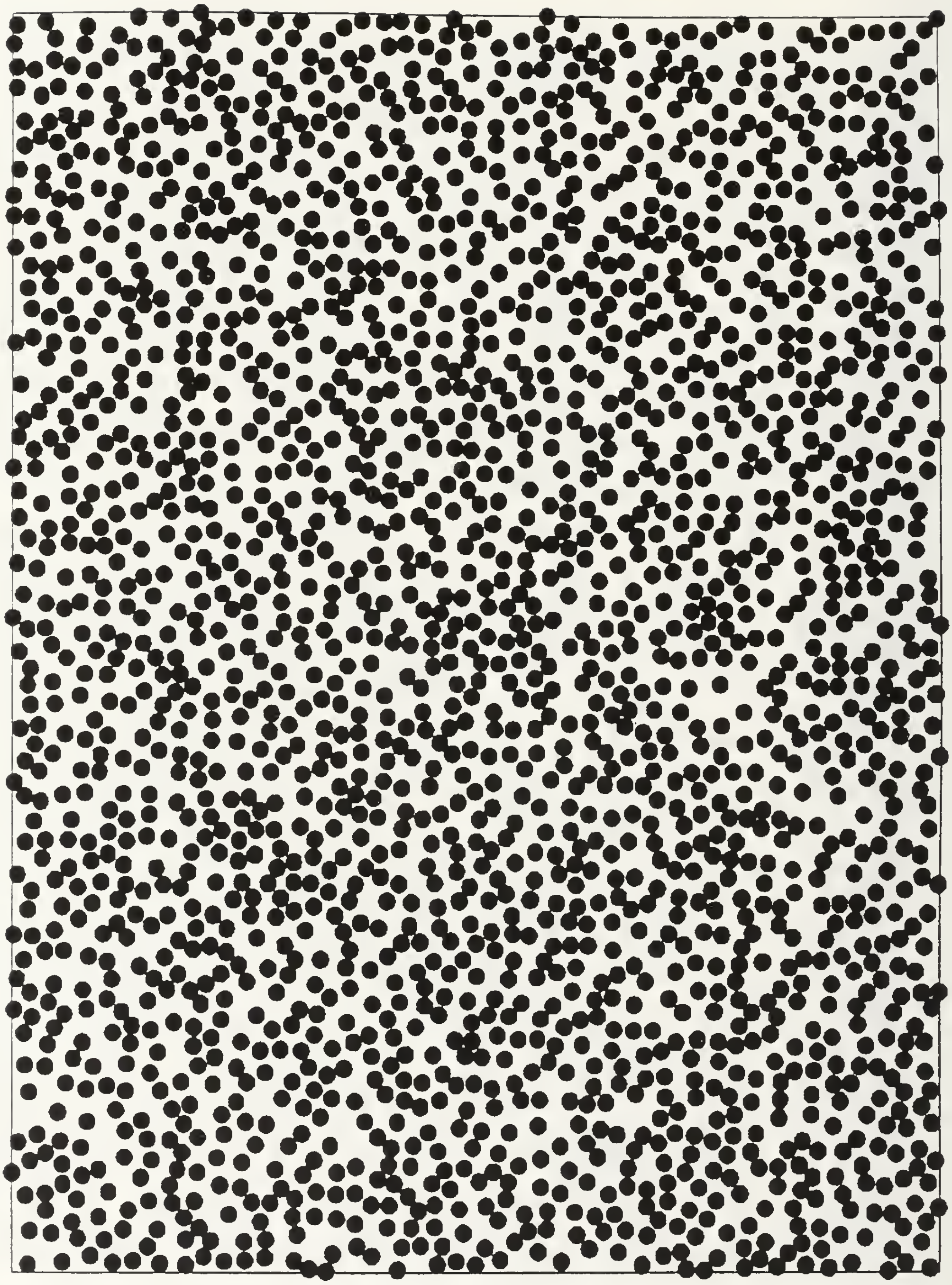
PARTICLE AREA FRACTION: 0.10
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.1699
NUMBER PARTICLES PER UNIT AREA: 12.72

Figure 19. Random distribution of constant size particles.



PARTICLE AREA FRACTION: 0.30
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.1218
NUMBER PARTICLES PER UNIT AREA: 37.95

Figure 20. Random distribution of constant size particles.



PARTICLE AREA FRACTION: 0.50
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.1082
NUMBER PARTICLES PER UNIT AREA: 62.82

Figure 21. Random distribution of constant size particles.

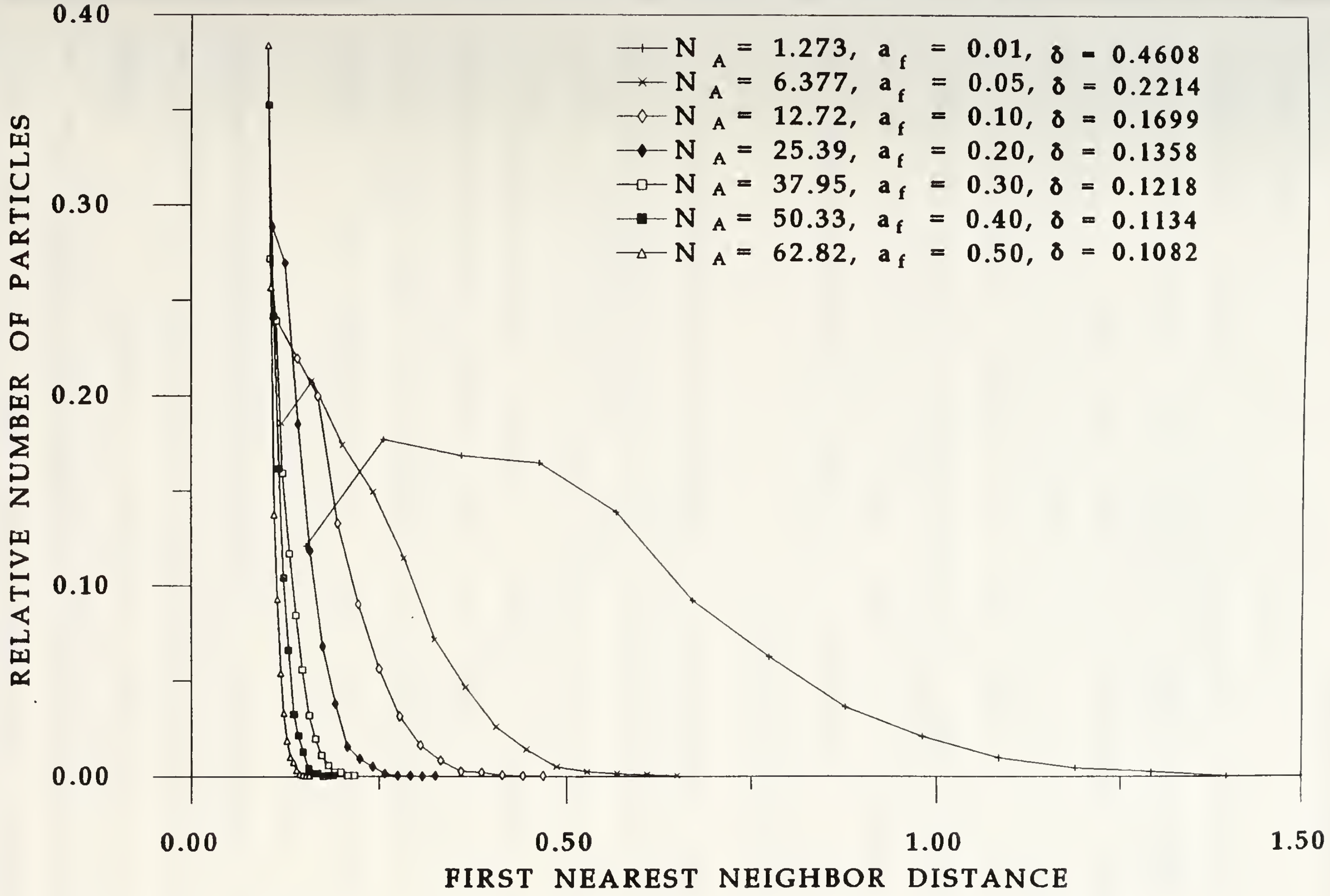


Figure 22. Random particle distributions for various area fractions; particle radius = 0.05.

functions with the minimum first NND having the highest frequency of occurrence.

The effects of increasing area fraction for finite sized particles is shown in Figure 23. As N_A increases from values of very diluted arrays to more dense distributions the corresponding values of δ shift from that of a random distribution of infinitesimal points to that of a hexagonal array. Figure 23 shows that this trend is consistent for various particle radii. The reader should note the asymptotic approach towards the random theoretical distribution of points. This is indicative of using a sufficient number of particles in the simulations.

D. NON-RANDOM DISTRIBUTION OF CONSTANT SIZE PARTICLES

A non-random distribution was simulated by dividing the total defined area in to five sub-areas of equal size. In all simulations there were three low density zones separated by two high density zones. The high density zones were each assigned 35% of the particles occupying the total area, while the low density zones were each assigned 10%. These parameters were used for all simulations.

The non-random distribution of constant size particles for various area fractions are shown in Figures 24 - 27. Plots of the particles themselves are shown in Figures 28 - 31. These figures were compared with the corresponding distributions and particle plots, with respect to area fraction, in the random

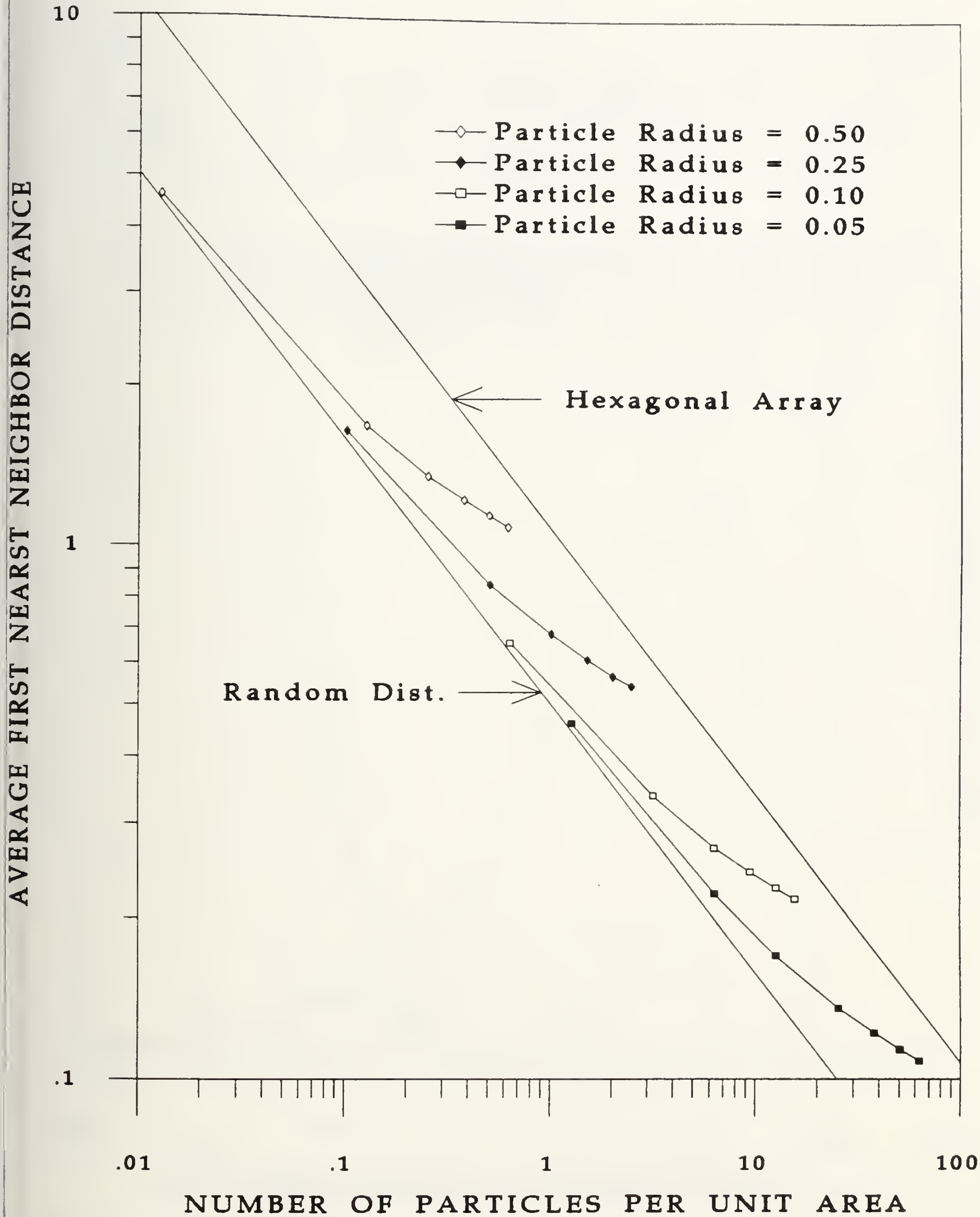


Figure 23. Number of particles per unit area vs average first nearest neighbor distance for particles of varying radius.

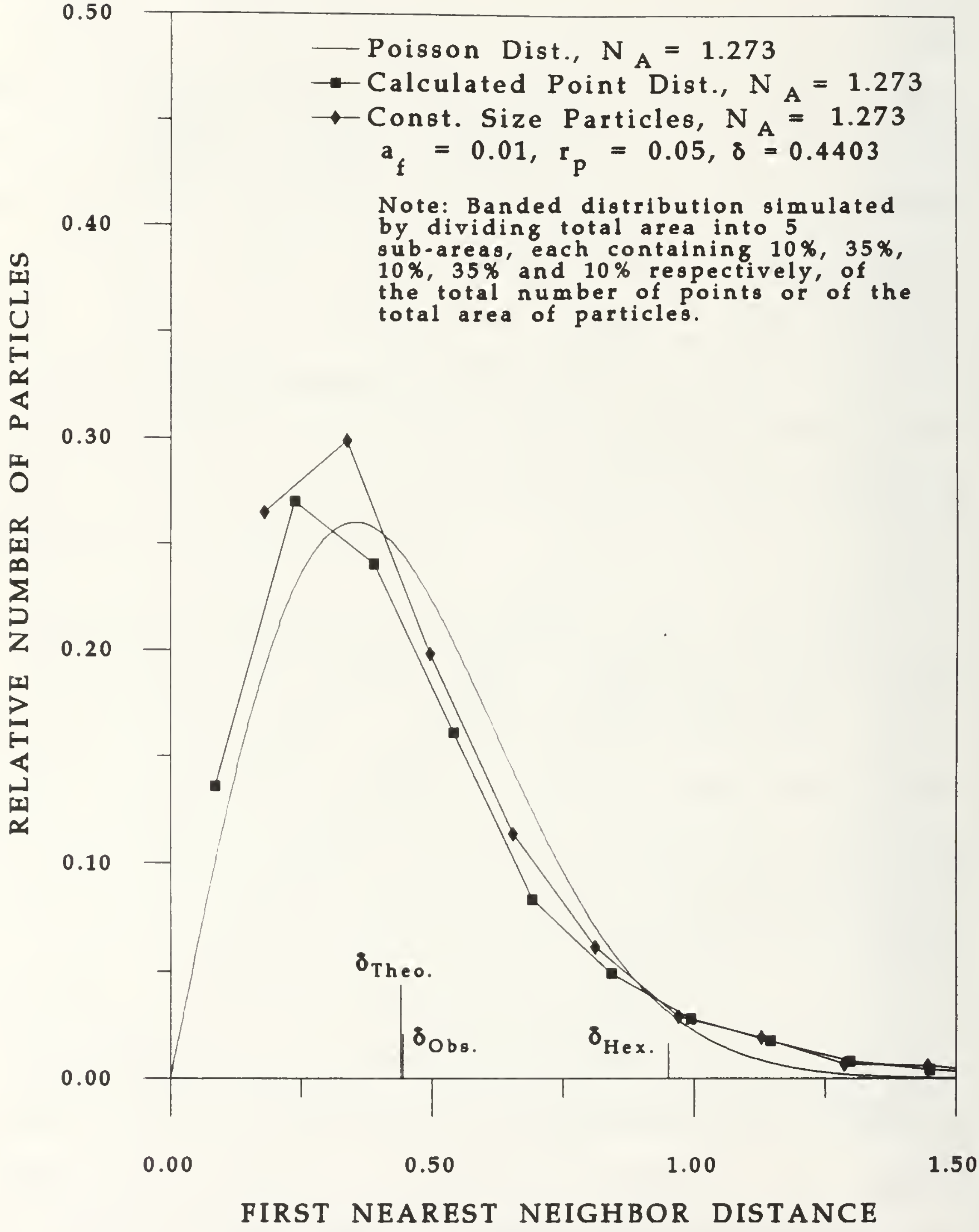


Figure 24. Non-random distribution of points and constant size particles.

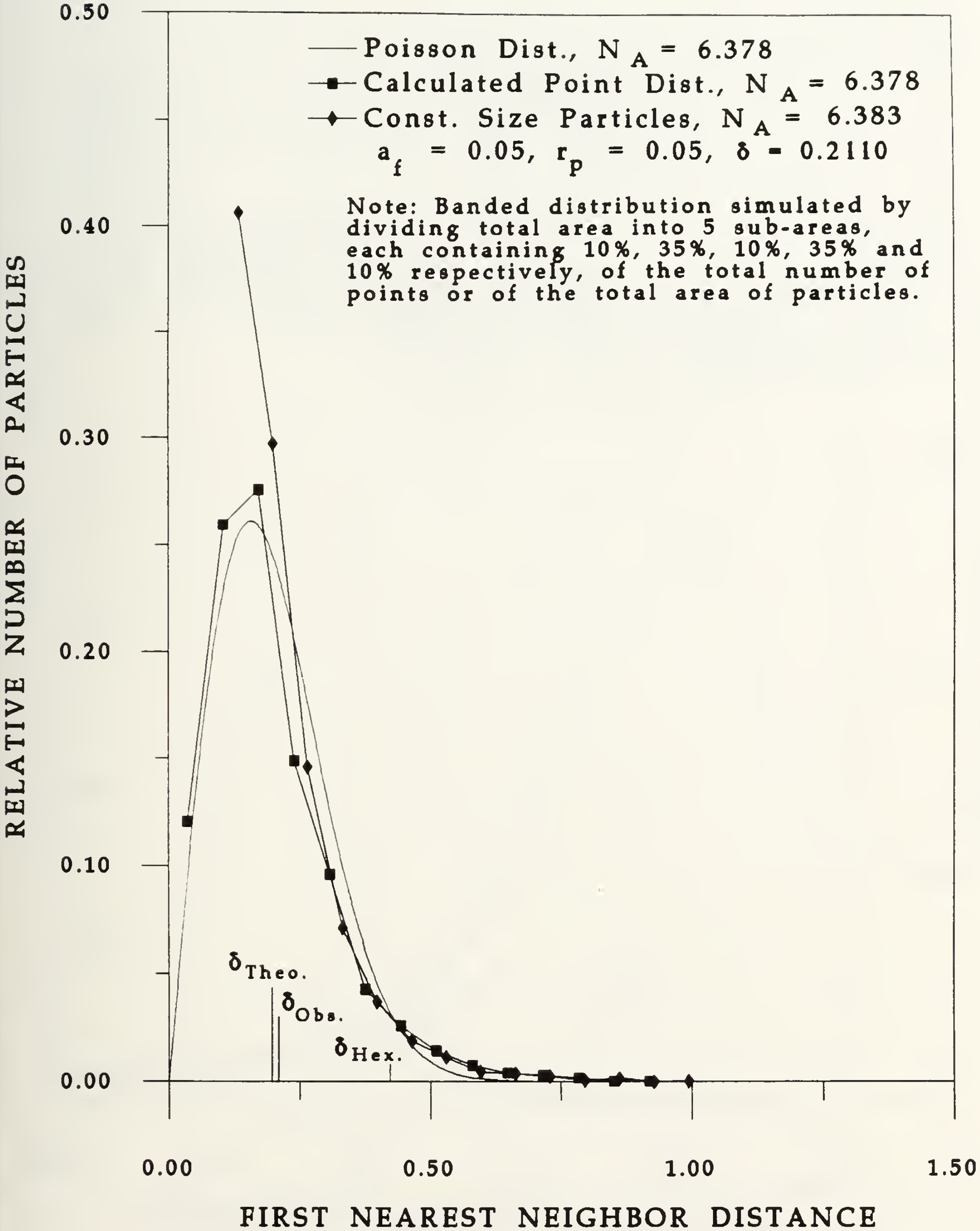


Figure 25. Non-random distribution of points and constant size particles.

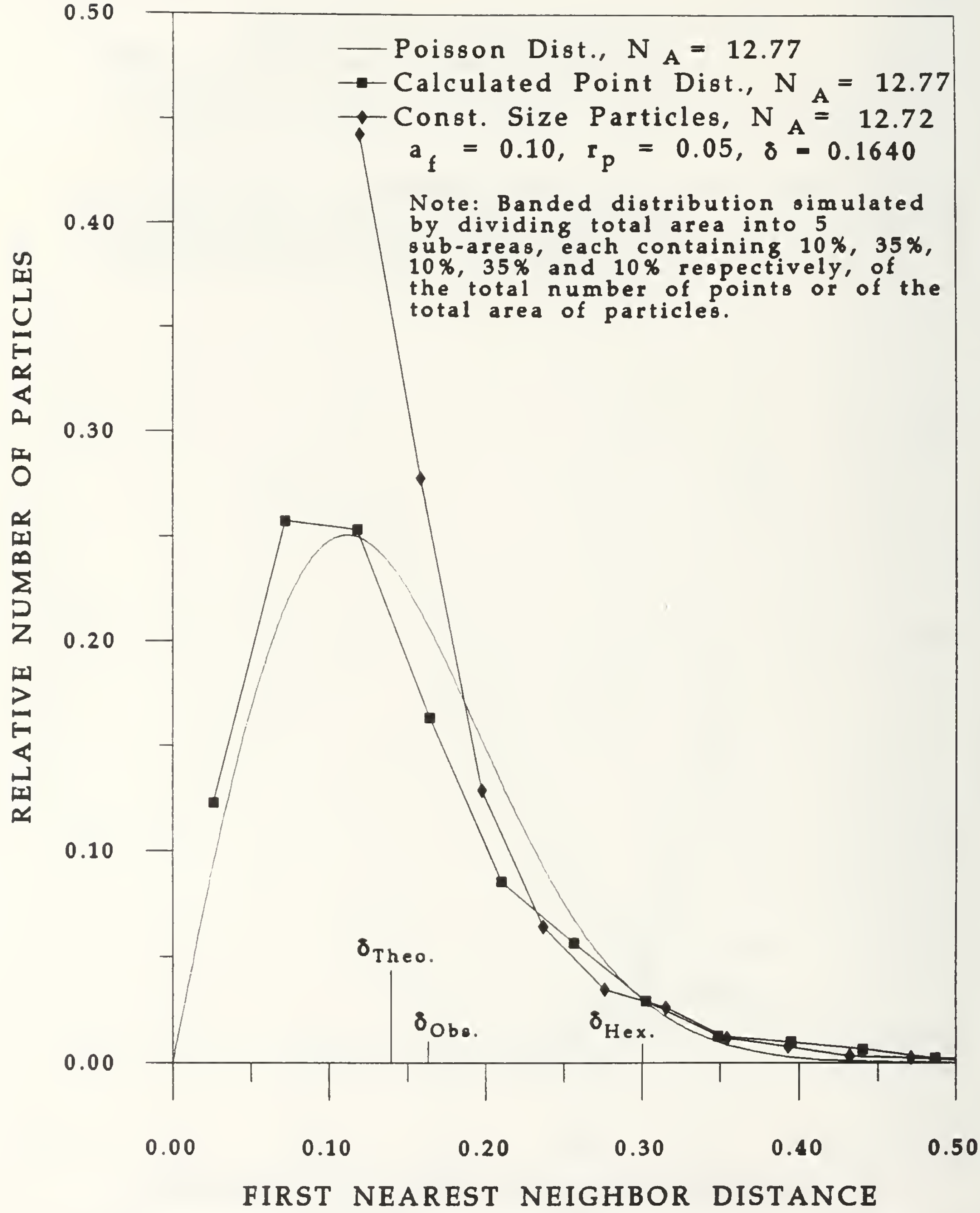


Figure 26. Non-random distribution of points and constant size particles.

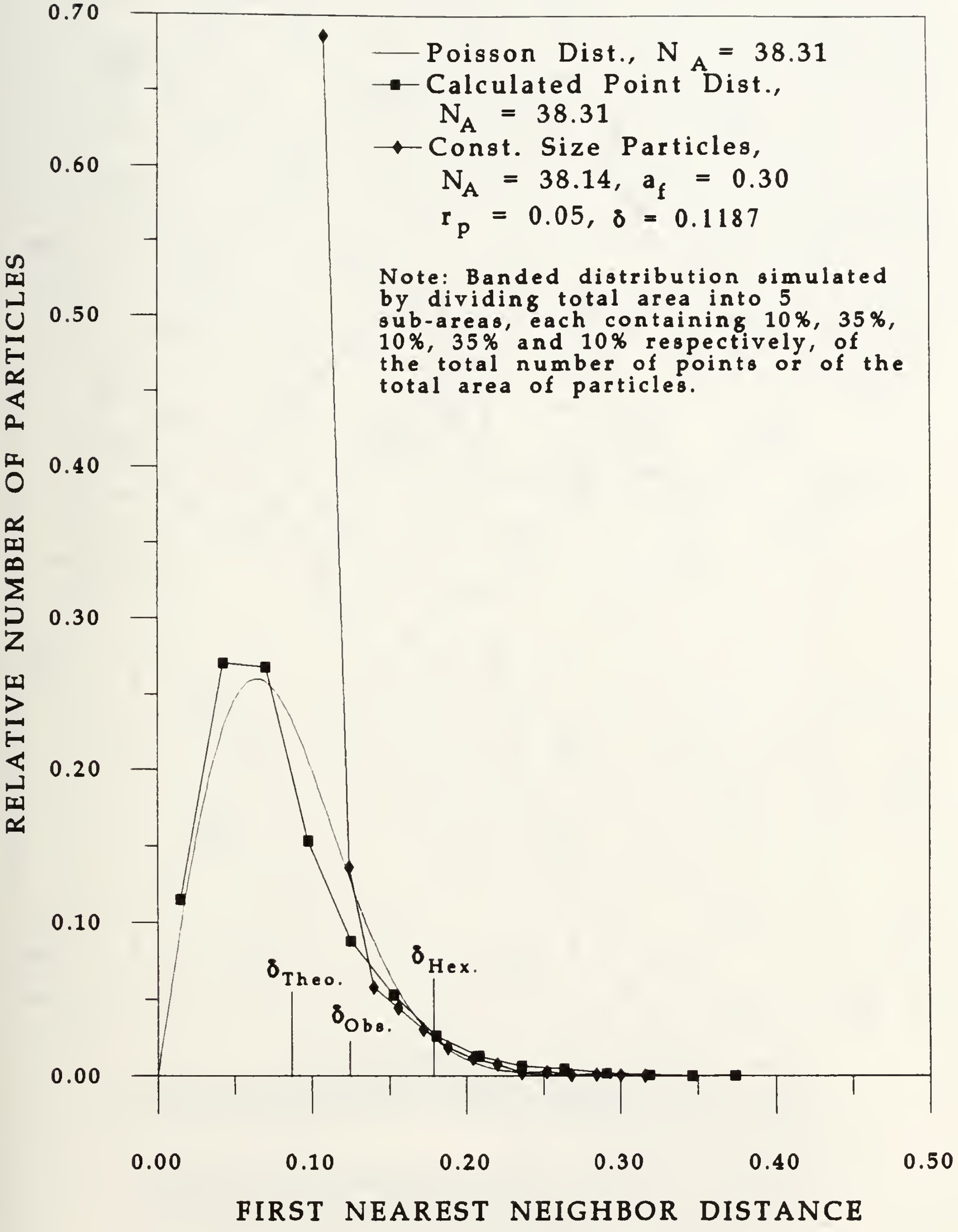
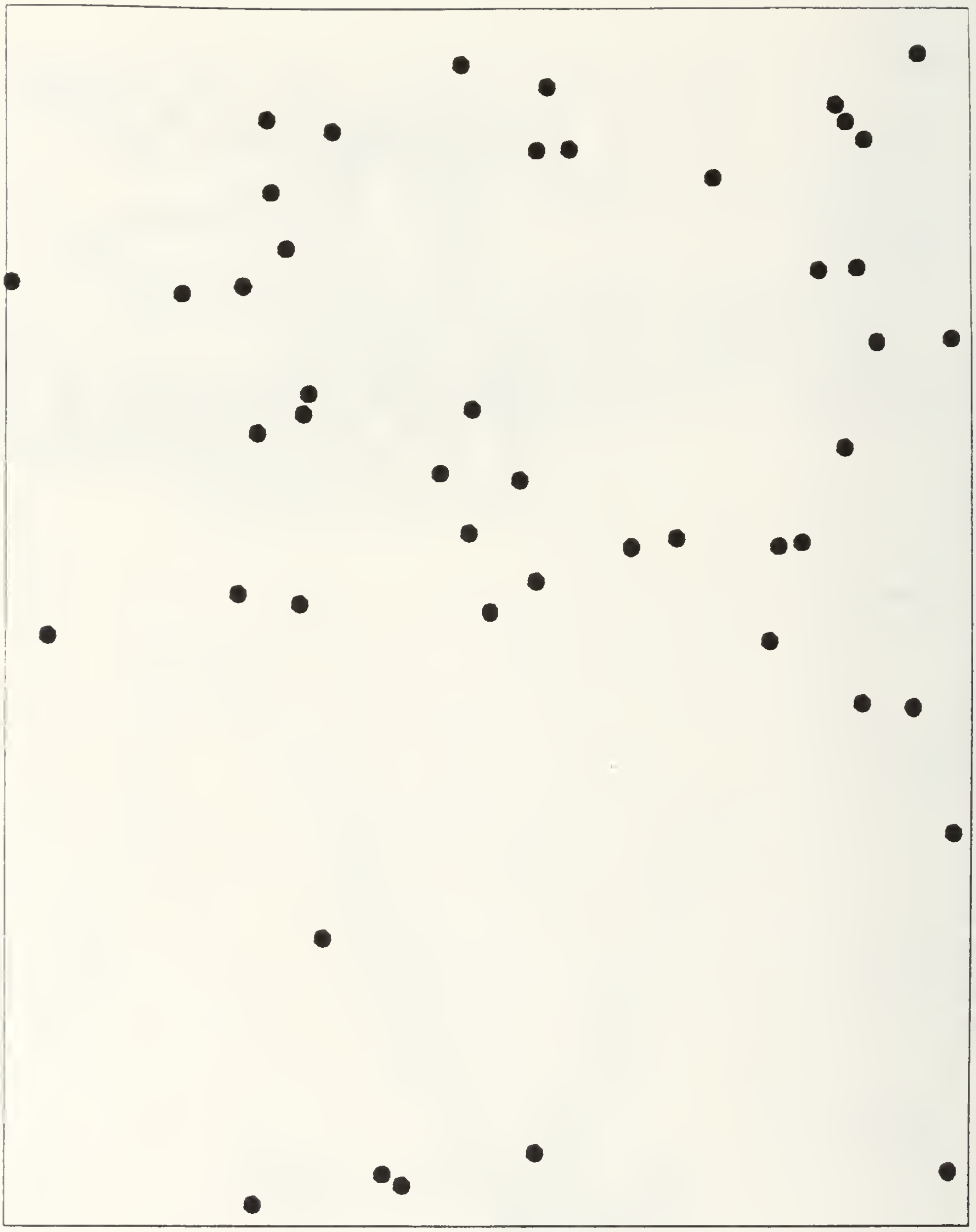
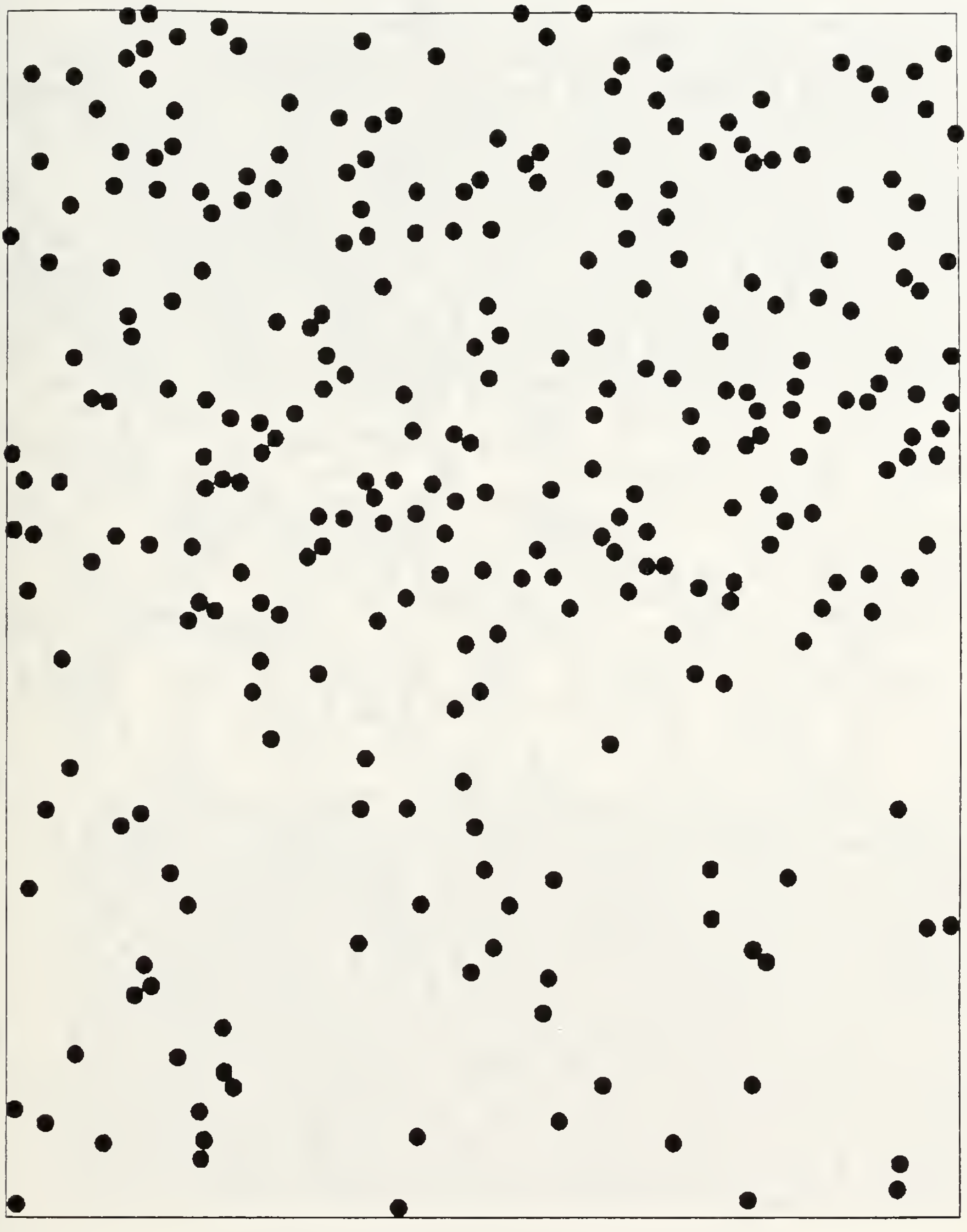


Figure 27. Non-random distribution of points and constant size particles.



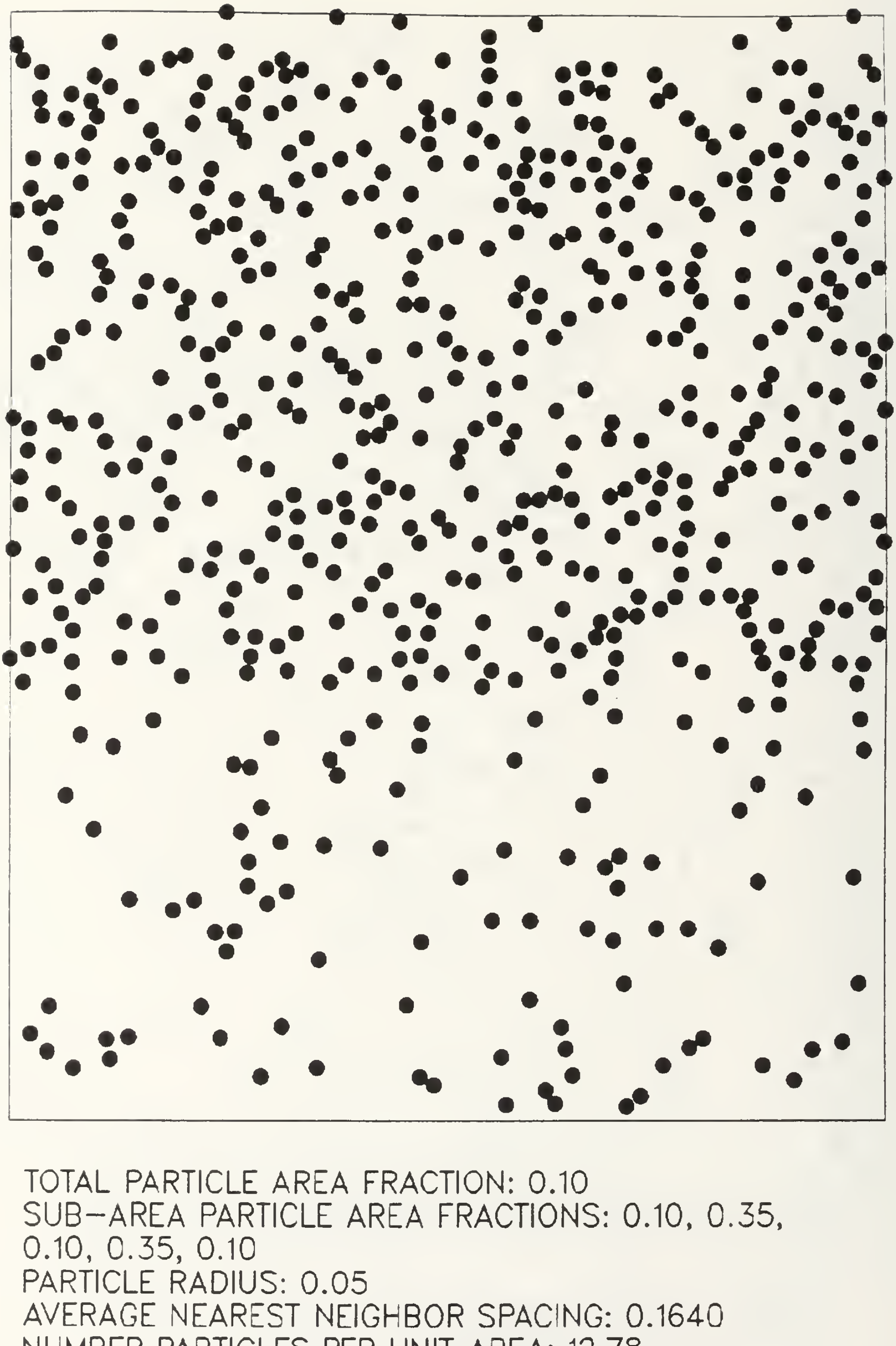
TOTAL PARTICLE AREA FRACTION: 0.01
SUB—AREA PARTICLE AREA FRACTIONS: 0.10, 0.35,
0.10, 0.35, 0.10
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.4403
NUMBER PARTICLES PER UNIT AREA: 1.273

Figure 28. Non—random distribution of constant size particles.



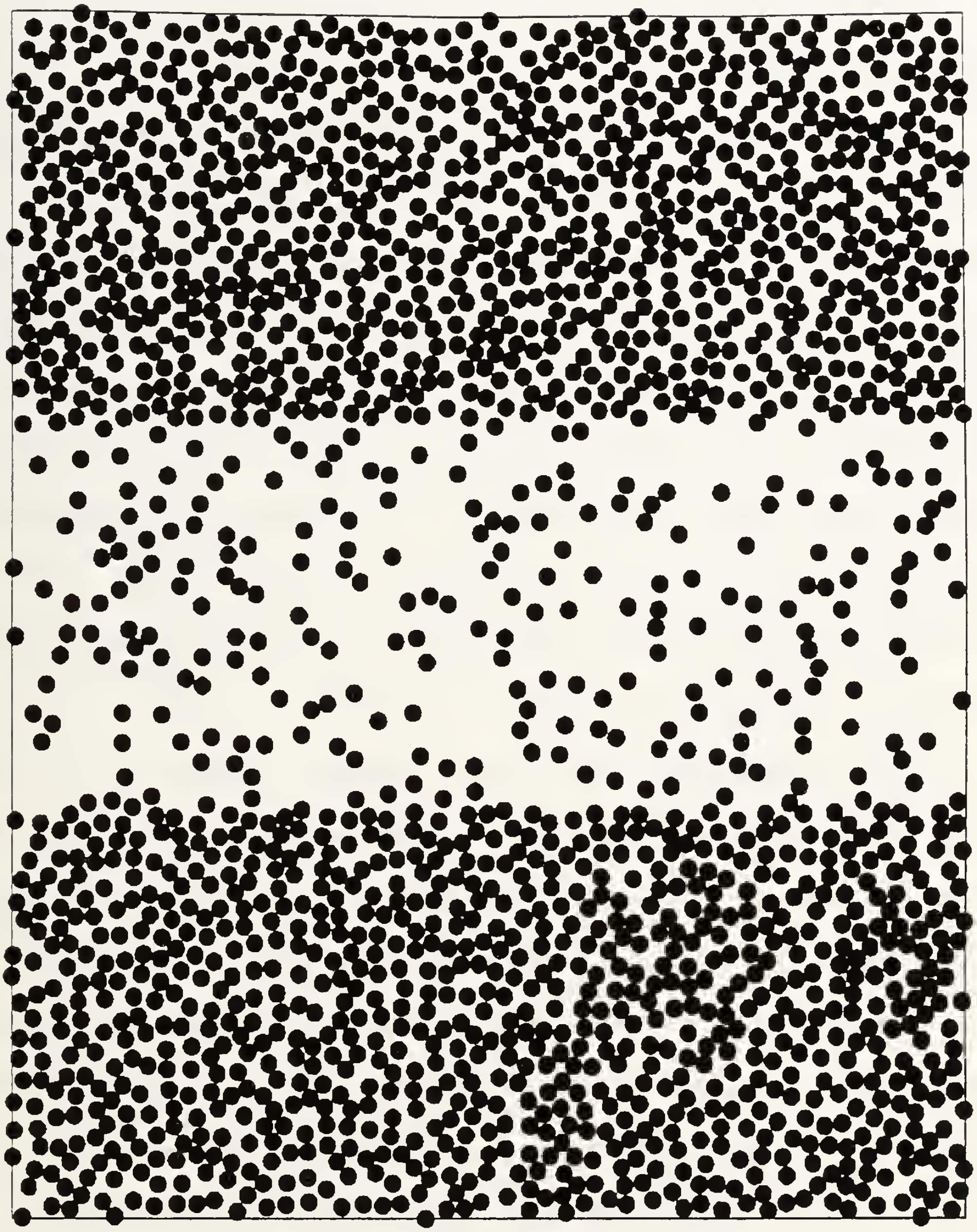
TOTAL PARTICLE AREA FRACTION: 0.05
SUB–AREA PARTICLE AREA FRACTIONS: 0.10, 0.35, 0.10, 0.35, 0.10
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.2110
NUMBER PARTICLES PER UNIT AREA: 6.383

Figure 29. Non–random distribution of constant size particles.



TOTAL PARTICLE AREA FRACTION: 0.10
SUB–AREA PARTICLE AREA FRACTIONS: 0.10, 0.35,
0.10, 0.35, 0.10
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.1640
NUMBER PARTICLES PER UNIT AREA: 12.78

Figure 30. Non–random distribution of constant size particles.



TOTAL PARTICLE AREA FRACTION: 0.30
SUB–AREA PARTICLE AREA FRACTIONS: 0.10, 0.35,
0.10, 0.35, 0.10
PARTICLE RADIUS: 0.05
AVERAGE NEAREST NEIGHBOR SPACING: 0.1187
NUMBER PARTICLES PER UNIT AREA: 38.14

Figure 31. Non–random distribution of constant size particles.

distribution case. It was observed that the fundamental form of the distributions, trend in values of the observed δ, and first NND were similar.

Additionally, in the non-random case, there was a small but consistent shift in the distribution towards lower values of first NND. This pattern was attributed to the presence of the high density bands. For a given N_A, the greater degree of packing of points or particles in the high density zones caused the first NND to shift toward lower values. This effect was also reflected in a greater frequency of the minimum first NND when compared to that of the random case. A summary of the observed trends in the non-random/constant particle size distributions are shown in Figure 32.

The corresponding particle plots, comparing the relative densities of the high and low density bands, are shown in Figures 28 - 31. Compare these plots with those of the random distributions for the same area fractions (Figures 17 - 21). For a_f = 0.30 the difference between the random (Figure 20) and non-random (Figure 31) distributions is completely obvious to the eye. There is a clear distinction between the high and low density bands in the non-random distribution. This distinction makes it easy for the observer to tell the difference between the random and non-random particle distributions based on these plots. Again, for a_f = 0.30, compare the distribution curves for the random (Figure. 15)

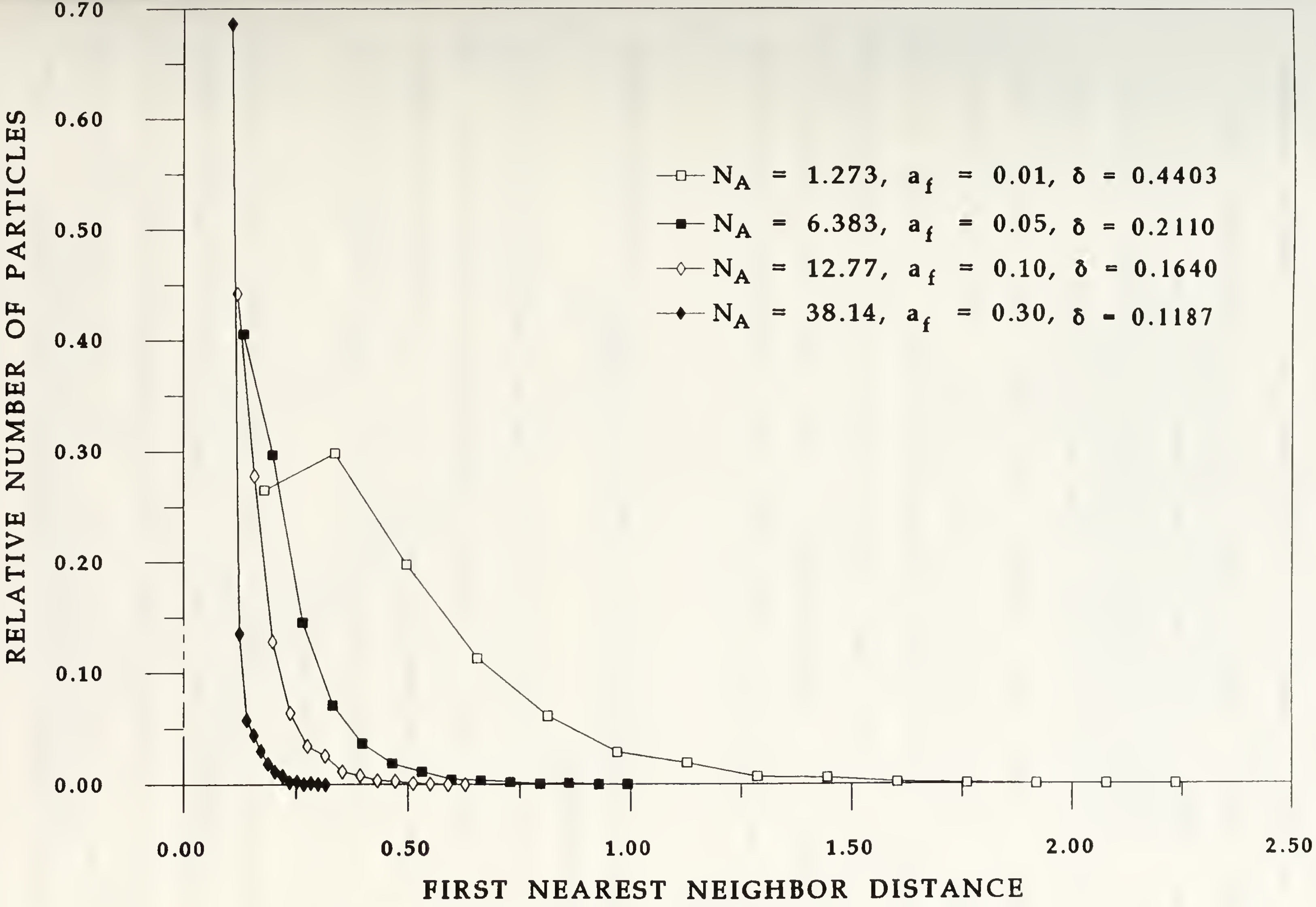


Figure 32. Non-random particle distributions for various area fractions; particle radius - 0.05

and non-random (Figure 27) cases. The distinction is not great and the discernable difference is a greater maximum frequency for the lowest first NND in the non-random case.

The visual difference in the particle plôts is much more subtle for low area fractions compared to high area fractions. Compare the random and non-random particle plots for $a_f = 0.01$ (Figures 17 and 28). Unlike the plots for the 30% area fraction, the distinction is not obvious. However, the same can be said for the NND distribution curves (Figures 12 and 24). For the low area fraction both curves approach the distribution of random points. This made any distinction between them even more subtle than the curves for the 30% area fraction.

E. RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION

In actual microstructures the second phase-particle size generally will not be constant throughout the material. In fact it has often been reported that second-phase particles follow a lognormal size distribution. Such was the basis for the following experiments.

The same area fractions were analyzed for particles with a lognormal size distribution (Figures 33 - 37) as with those for constant size particles. The corresponding particle plots are shown in Figures 38 - 42. For very dilute arrays of particles, the distributions approached very closely to those

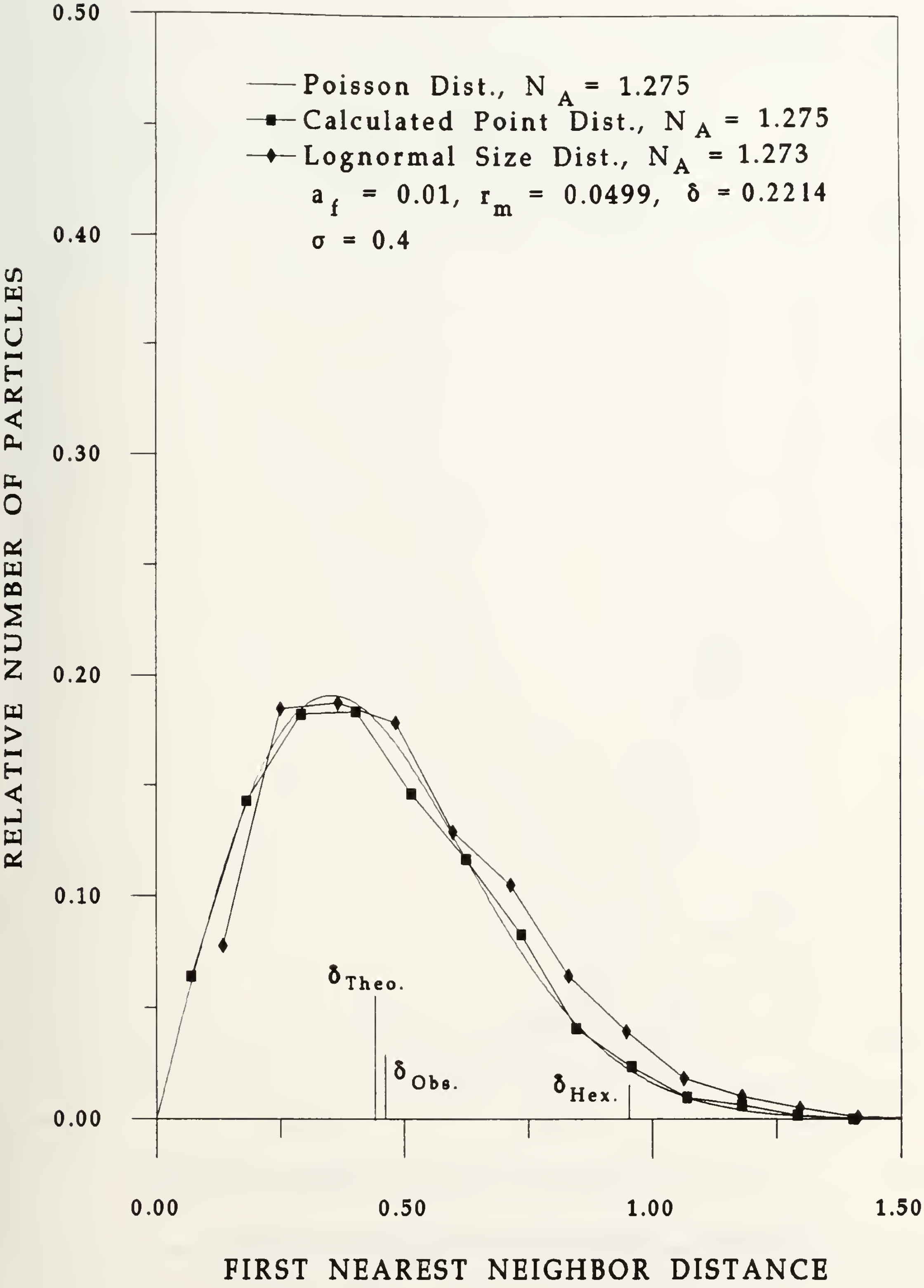


Figure 33. Random distribution of points and particles with a lognormal size distribution.

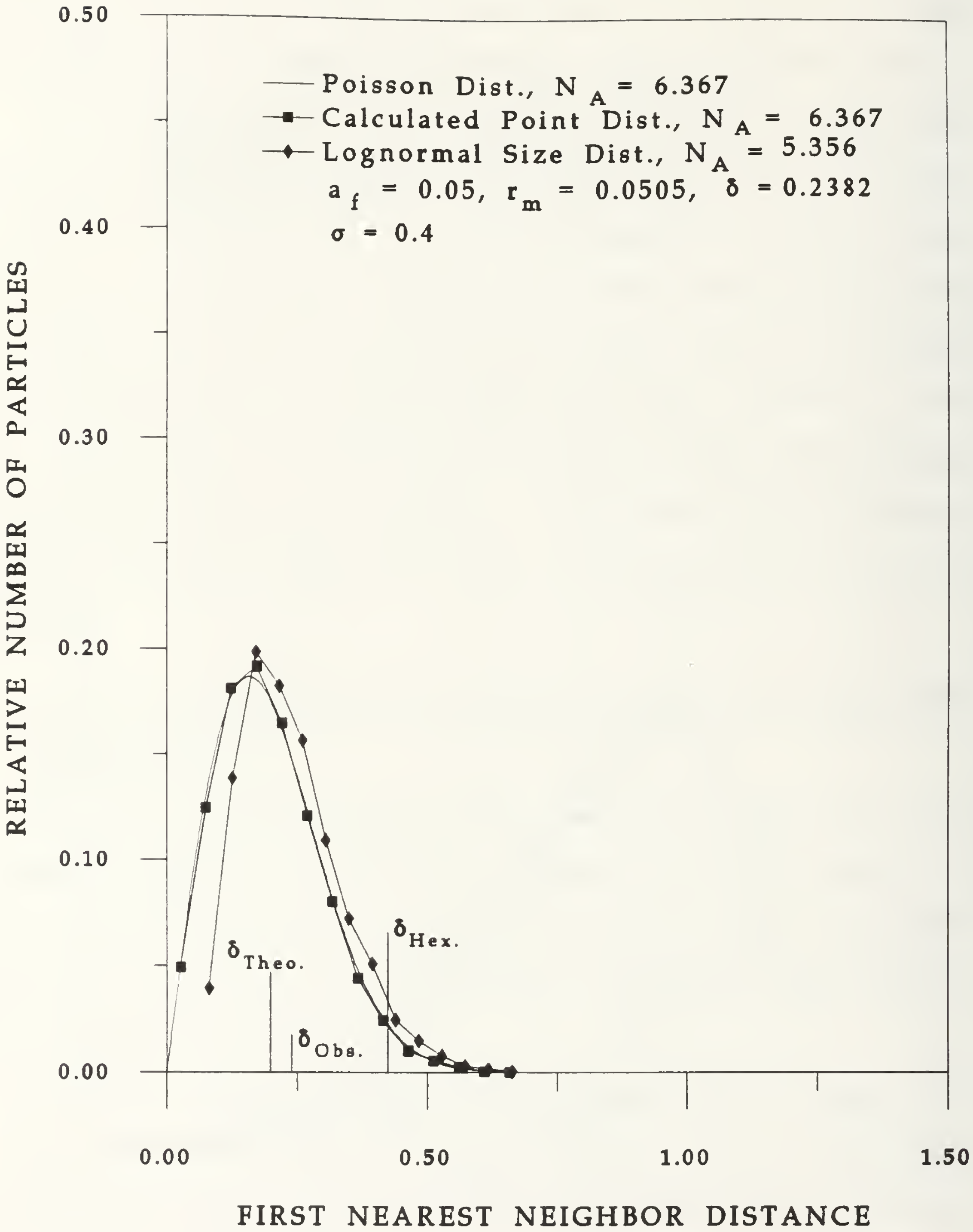


Figure 34. Random distribution of points and particles with a lognormal size distribution.

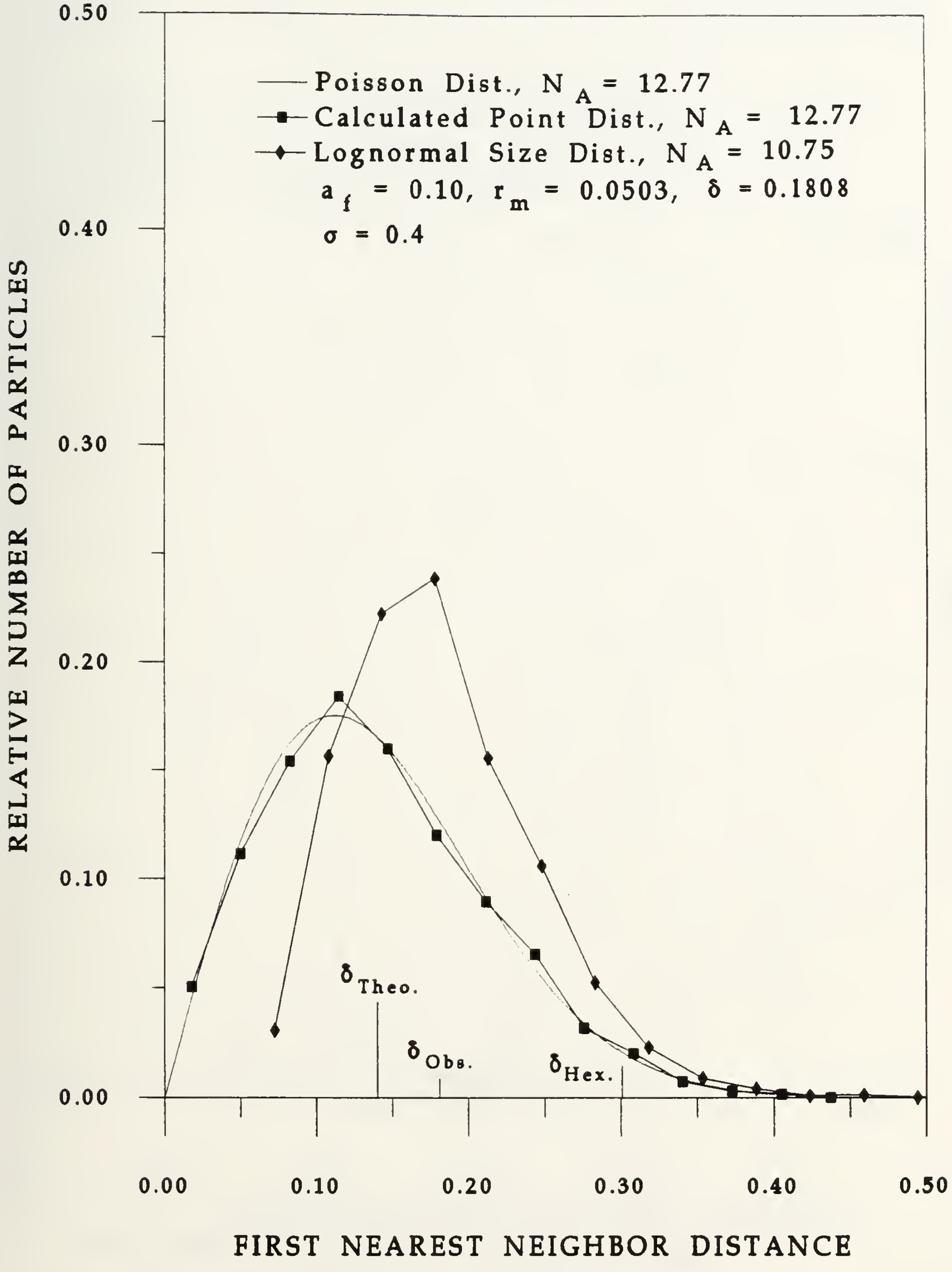


Figure 35. Random distribution of points and particles with a lognormal size distribution.

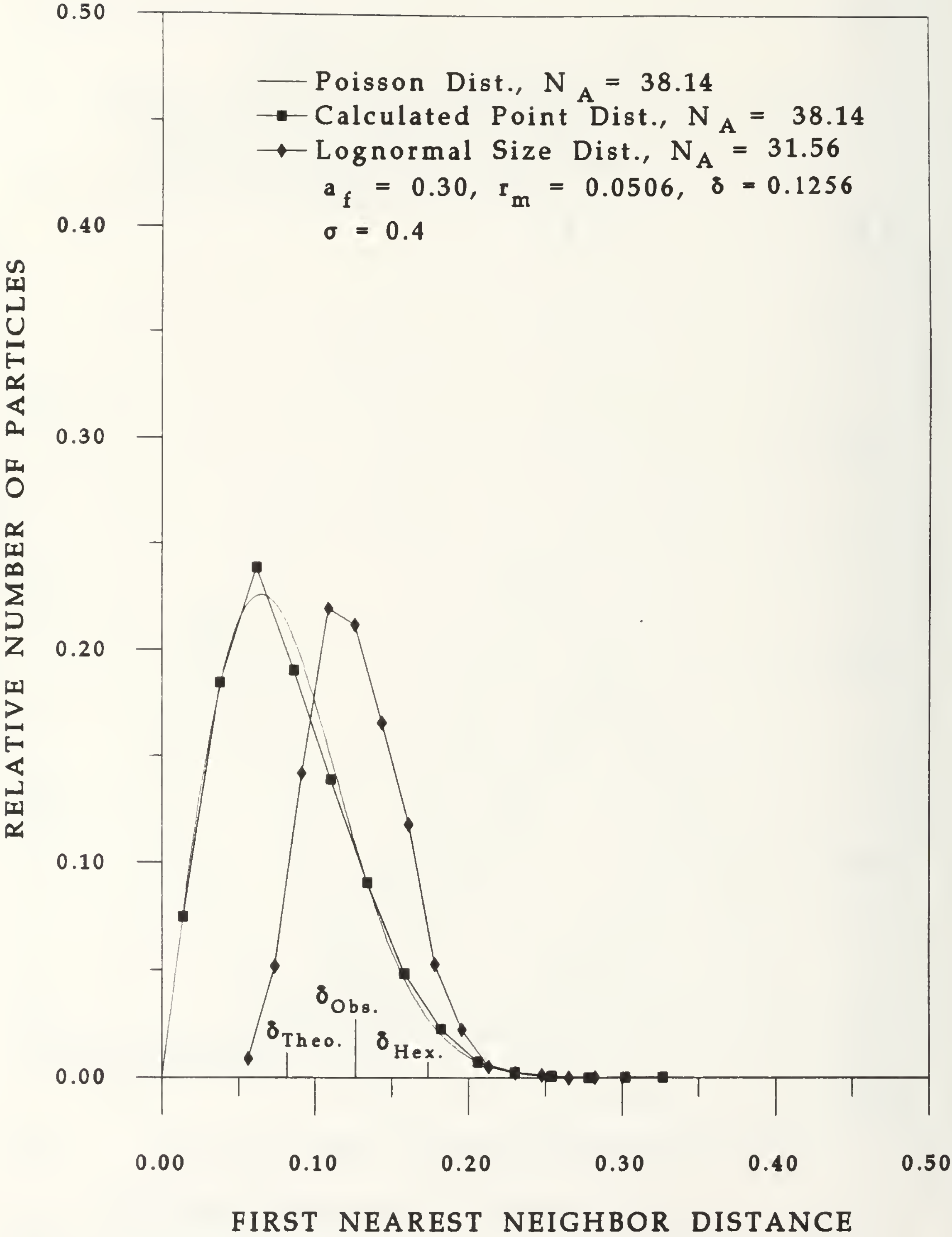


Figure 36. Random distribution of points and particles with a lognormal size distribution.

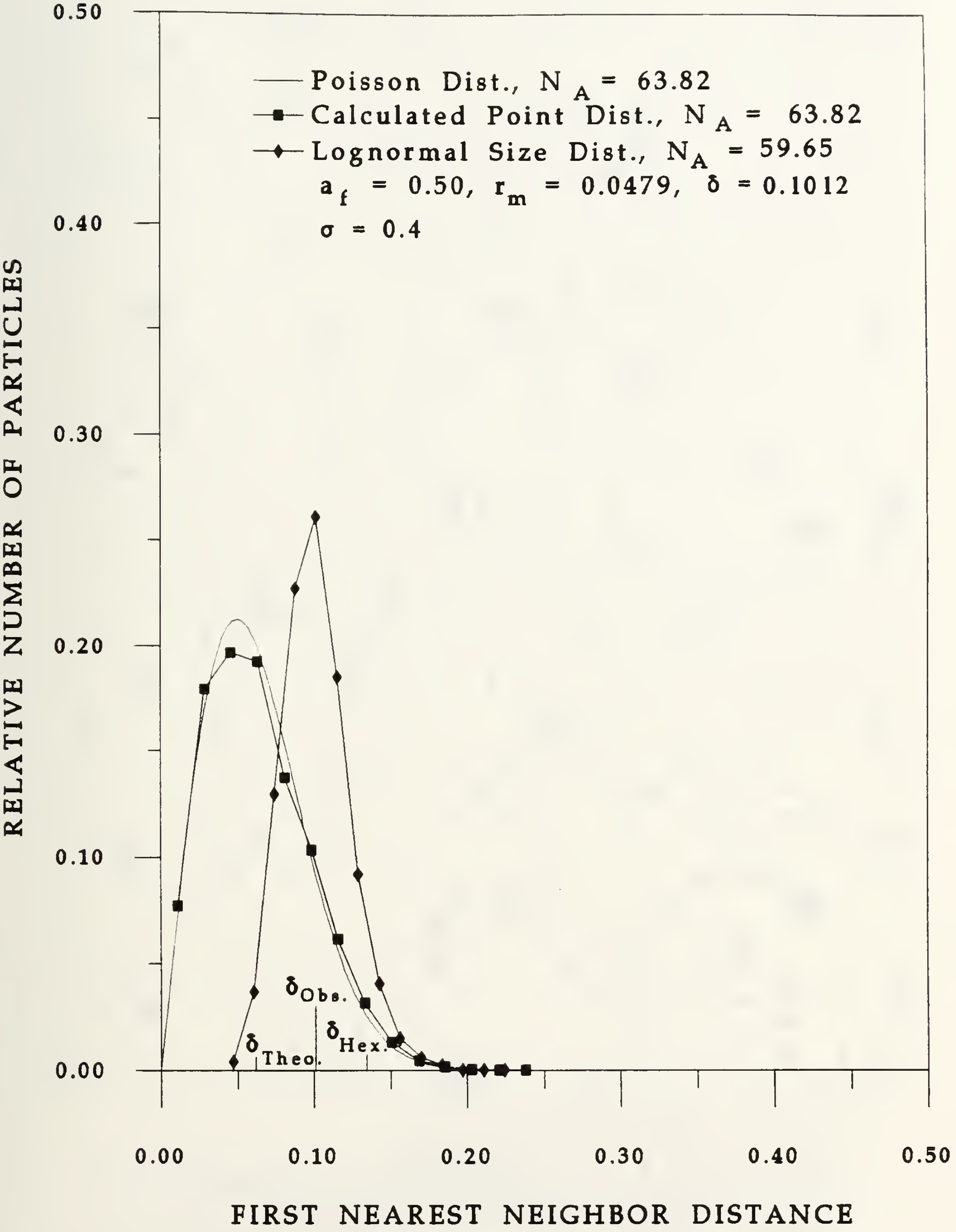
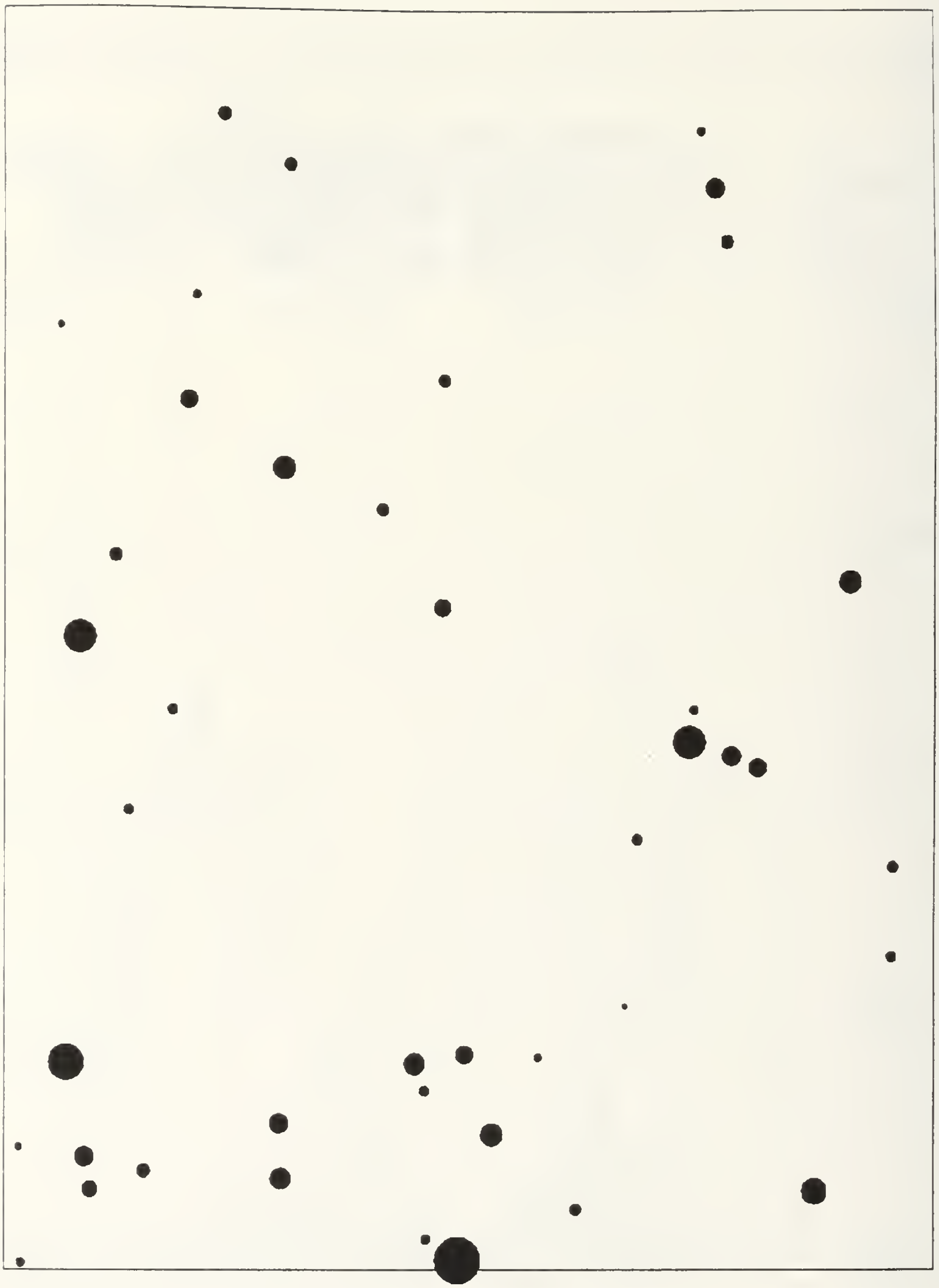
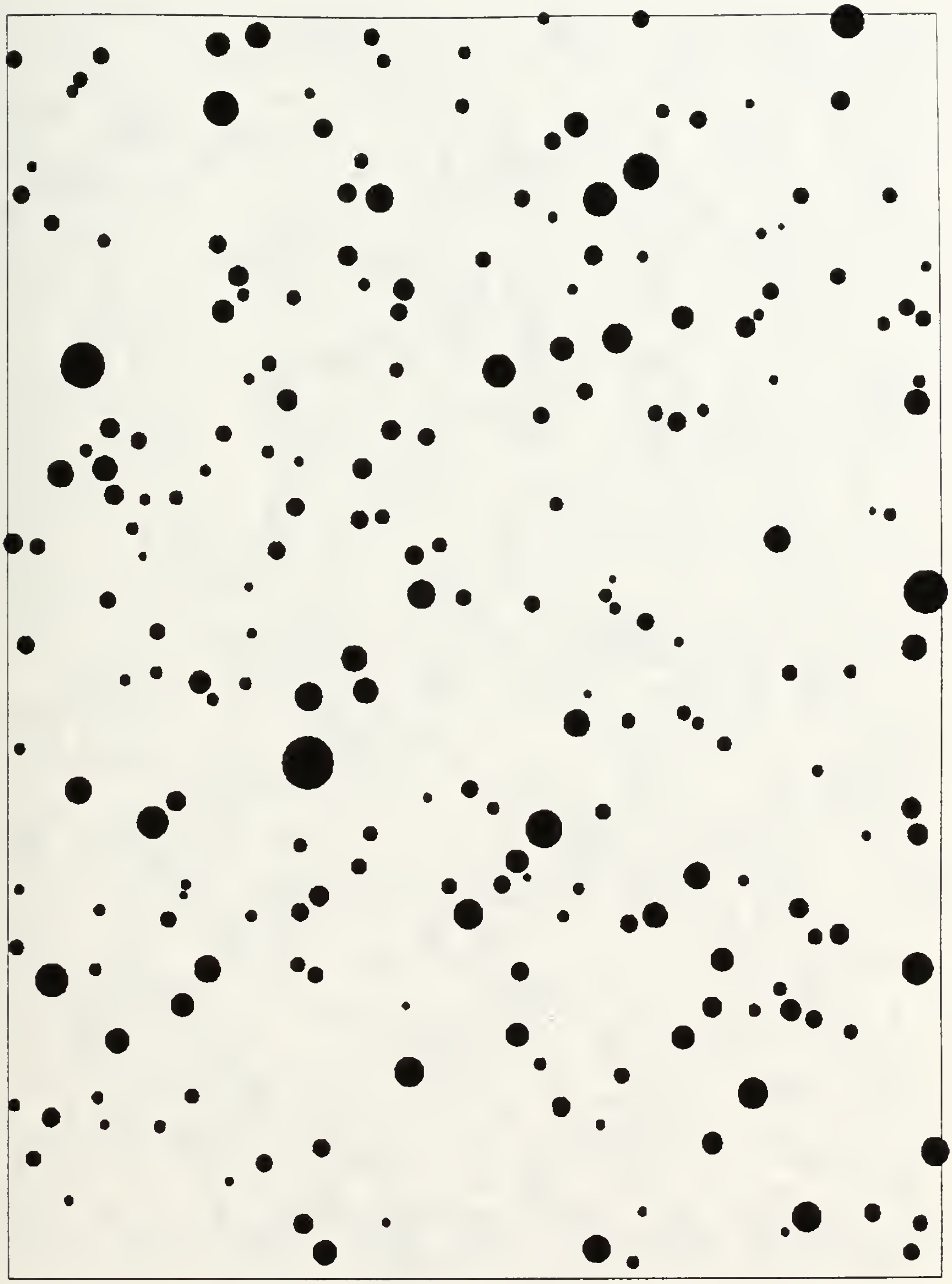


Figure 37. Random distribution of points and particles with a lognormal size distribution.



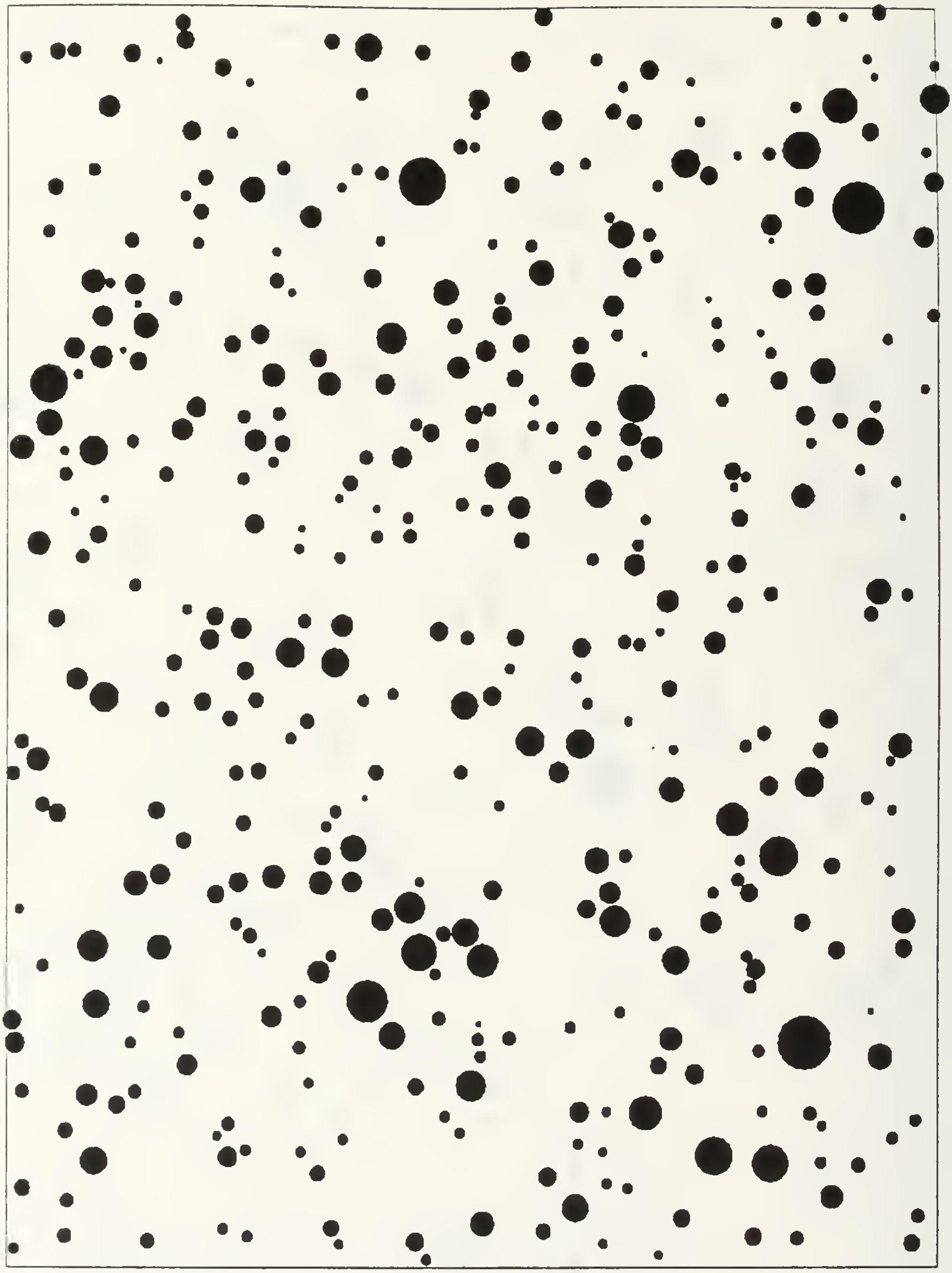
PARTICLE AREA FRACTION: 0.01
MEAN PARTICLE RADIUS: 0.0499
AVERAGE NEAREST NEIGHBOR SPACING: 0.4944
NUMBER PARTICLES PER UNIT AREA: 1.094

Figure 38. Random distribution of particles with a lognormal size distribution.



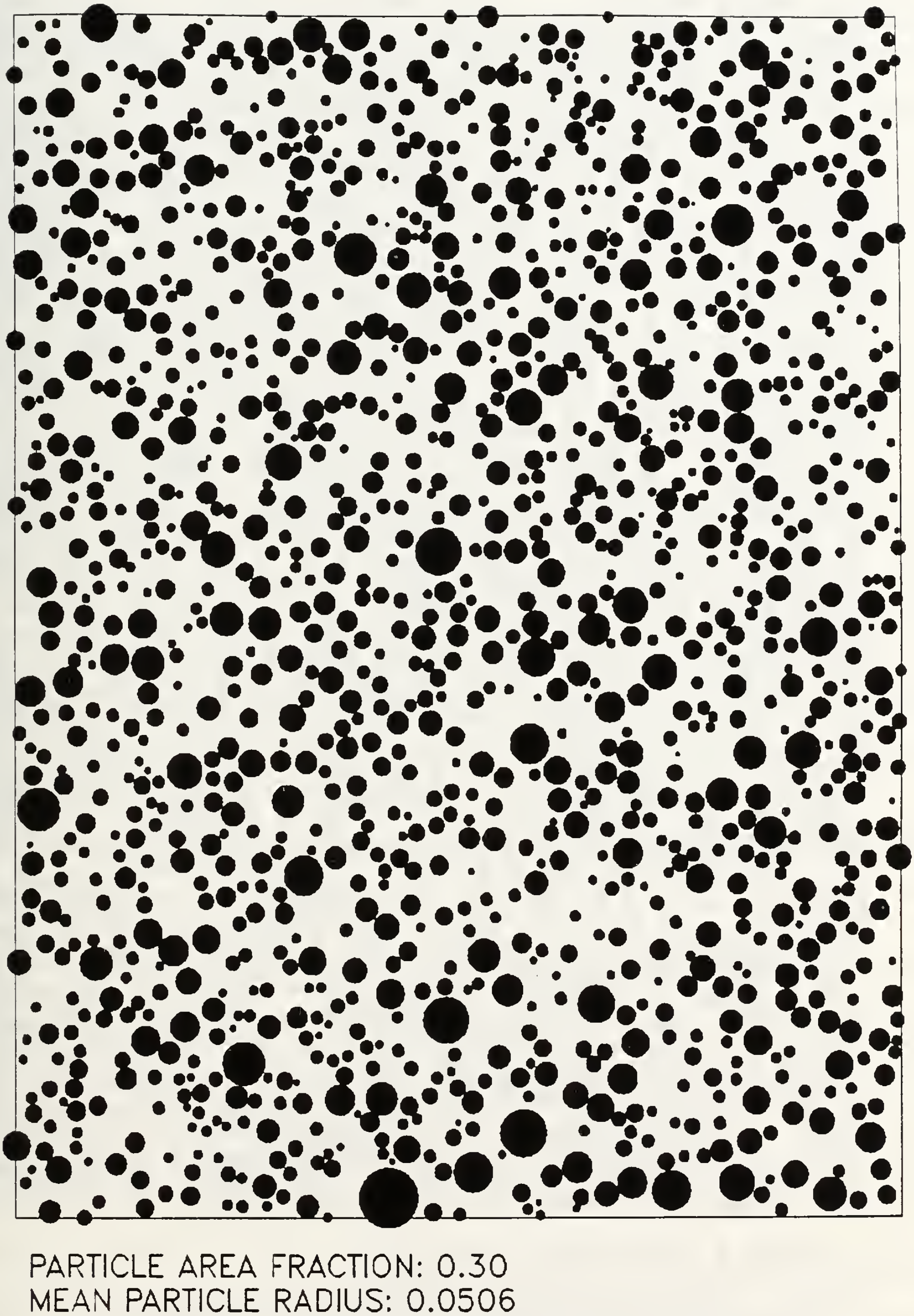
PARTICLE AREA FRACTION: 0.05
MEAN PARTICLE RADIUS: 0.0505
AVERAGE NEAREST NEIGHBOR SPACING: 0.2382
NUMBER PARTICLES PER UNIT AREA: 5.356

Figure 39. Random distribution of particles with a lognormal size distribution.



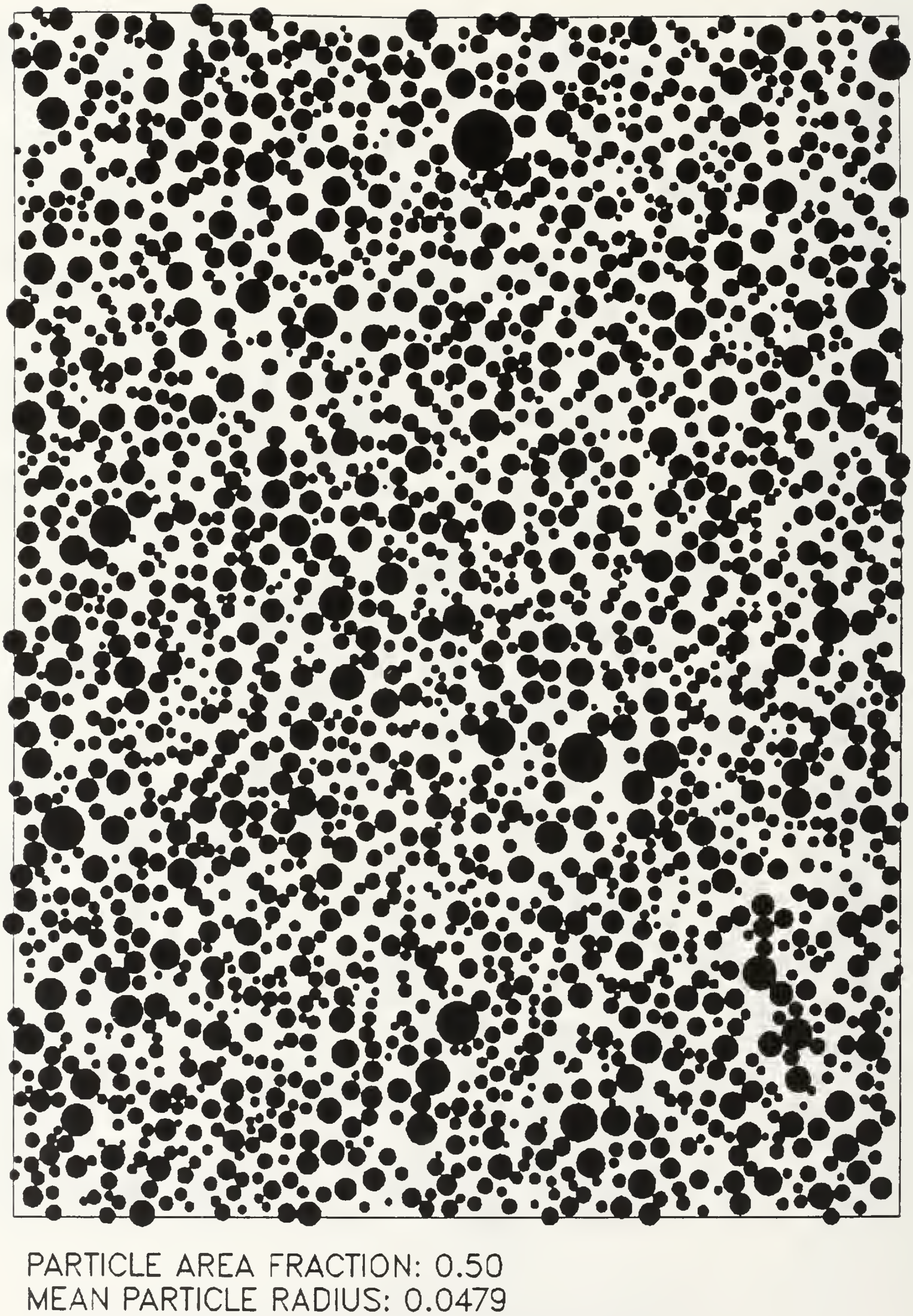
PARTICLE AREA FRACTION: 0.10
MEAN PARTICLE RADIUS: 0.0503
AVERAGE NEAREST NEIGHBOR SPACING: 0.1808
NUMBER PARTICLES PER UNIT AREA: 10.75

Figure 40. Random distribution of particles with a lognormal size distribution.



PARTICLE AREA FRACTION: 0.30
MEAN PARTICLE RADIUS: 0.0506
AVERAGE NEAREST NEIGHBOR SPACING: 0.1256
NUMBER PARTICLES PER UNIT AREA: 31.56

Figure 41. Random distribution of particles with a lognormal size distribution.



PARTICLE AREA FRACTION: 0.50
MEAN PARTICLE RADIUS: 0.0479
AVERAGE NEAREST NEIGHBOR SPACING: 0.1012
NUMBER PARTICLES PER UNIT AREA: 59.65

Figure 42. Random distribution of particles with a lognormal size distribution.

calculated for points and to the Poisson distribution (Figure 33). But now, as the area fraction is increased the particle distribution still maintains a shape similar to the Poisson distribution. Unlike the case for constant size particles, the minimum first NND can be smaller than the mean particle diameter. A smaller particle can always be fitted between two or more larger ones, thus reducing the first NND below the value of the mean particle diameter. This also resulted in a more nearly constant peak frequency of first NND for all ranges of area fractions examined.

The trends for δ of the particles and ranges of first NND (Figure 43) were the same as those in the constant size particle case for lower values of N_A. At larger values of N_A the value of δ for lognormally distributed particles becomes less than that for the corresponding constant particle sizes.

F. NON-RANDOM DISTRIBUTION OF PARTICLES WITH A LOGNORMAL SIZE DISTRIBUTION

The same parameters for banding used in the non-random, constant size particle case were used in the case of non-random distribution of particles with a lognormal size distribution. The results of this analysis are shown in Figures 44 - 47 for the distributions and Figures 48 - 51 for the particle plots. The features are similar in many respects to those reported earlier for non-random/constant particle size distributions. As the area fraction increases δ shifts

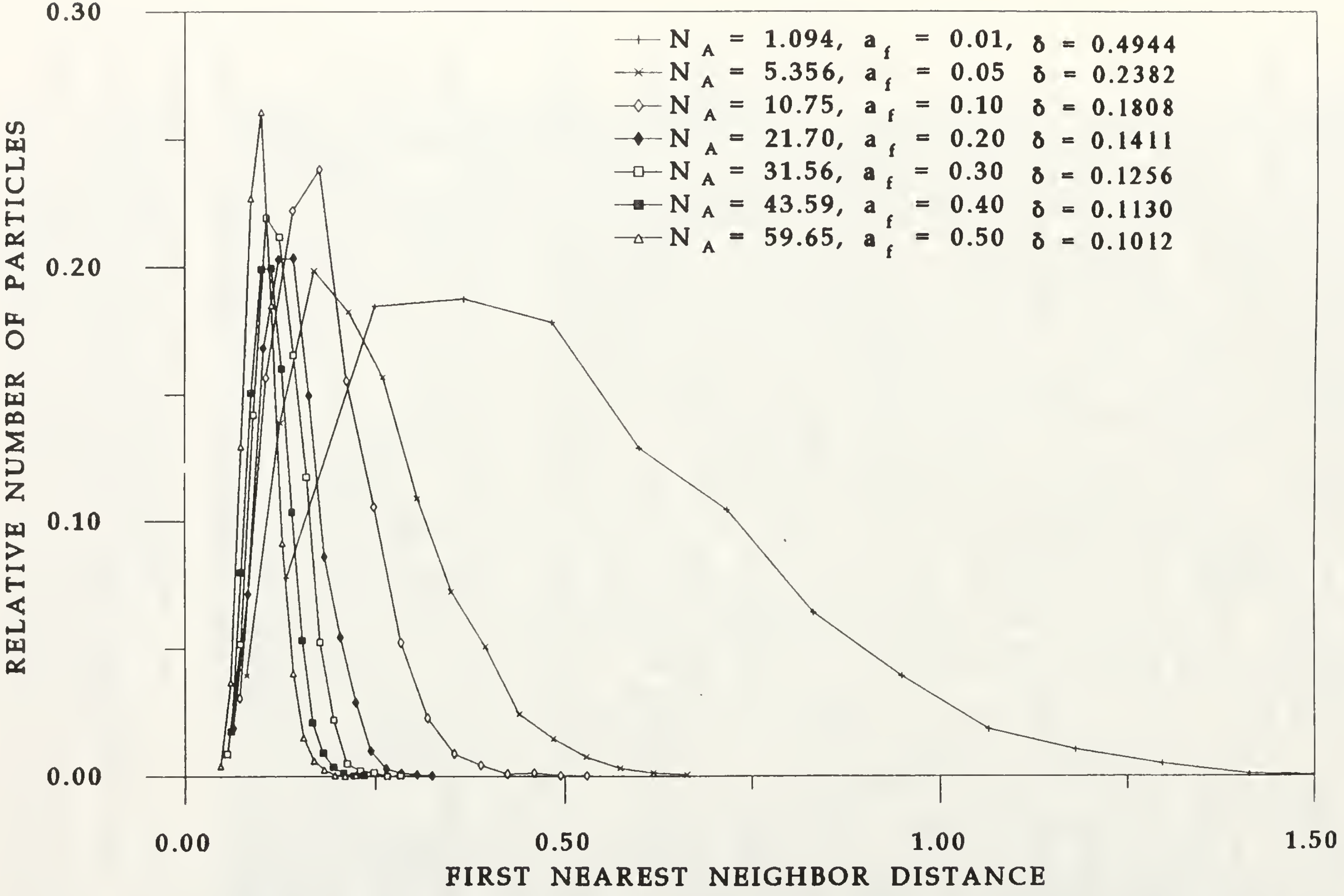


Figure 43. Random particle distributions for various area fractions; lognormal particle size distribution; mean particle radius = 0.05; standard deviation = 0.4.

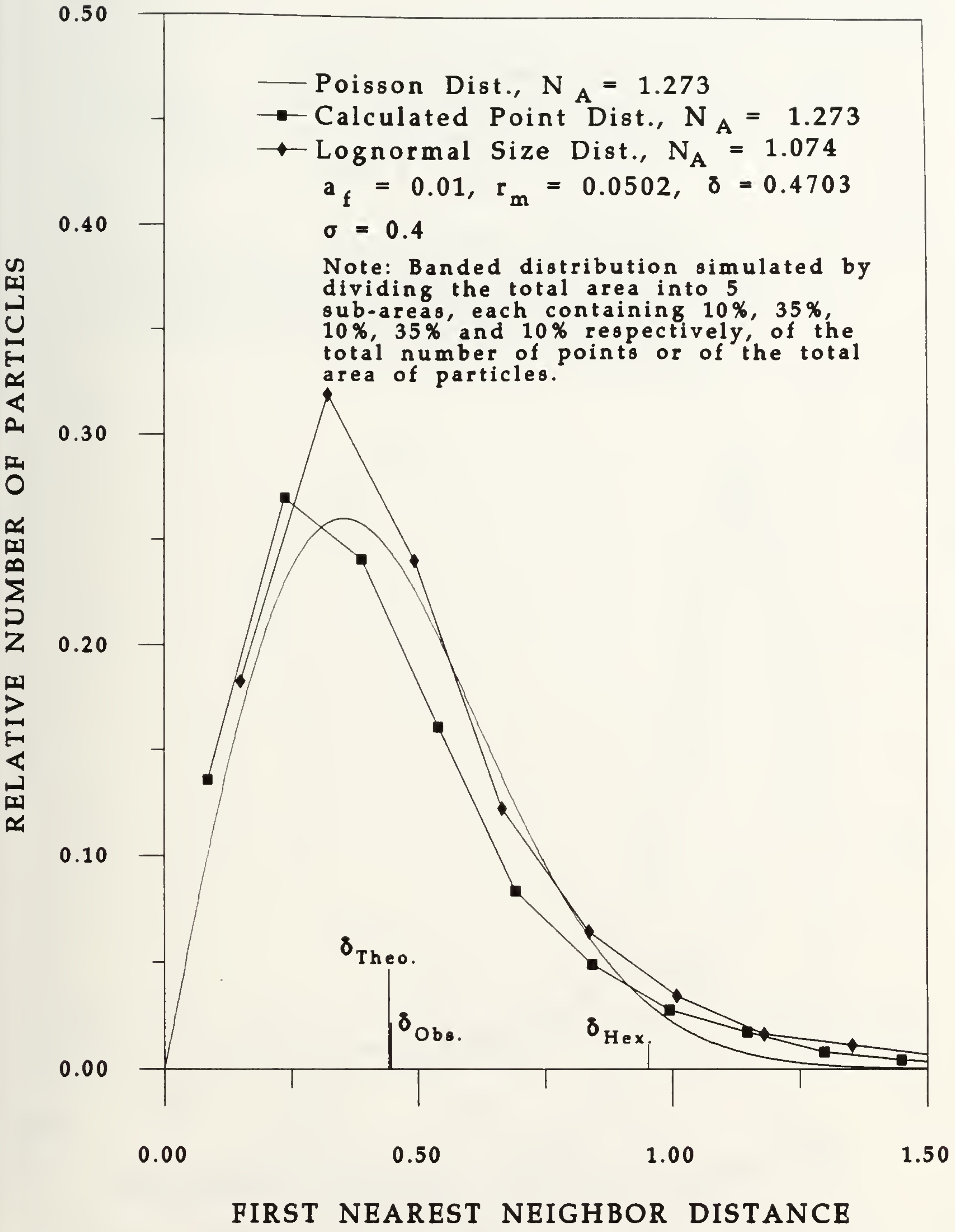


Figure 44. Non-random distribution of points and particles with a lognormal size distribution.

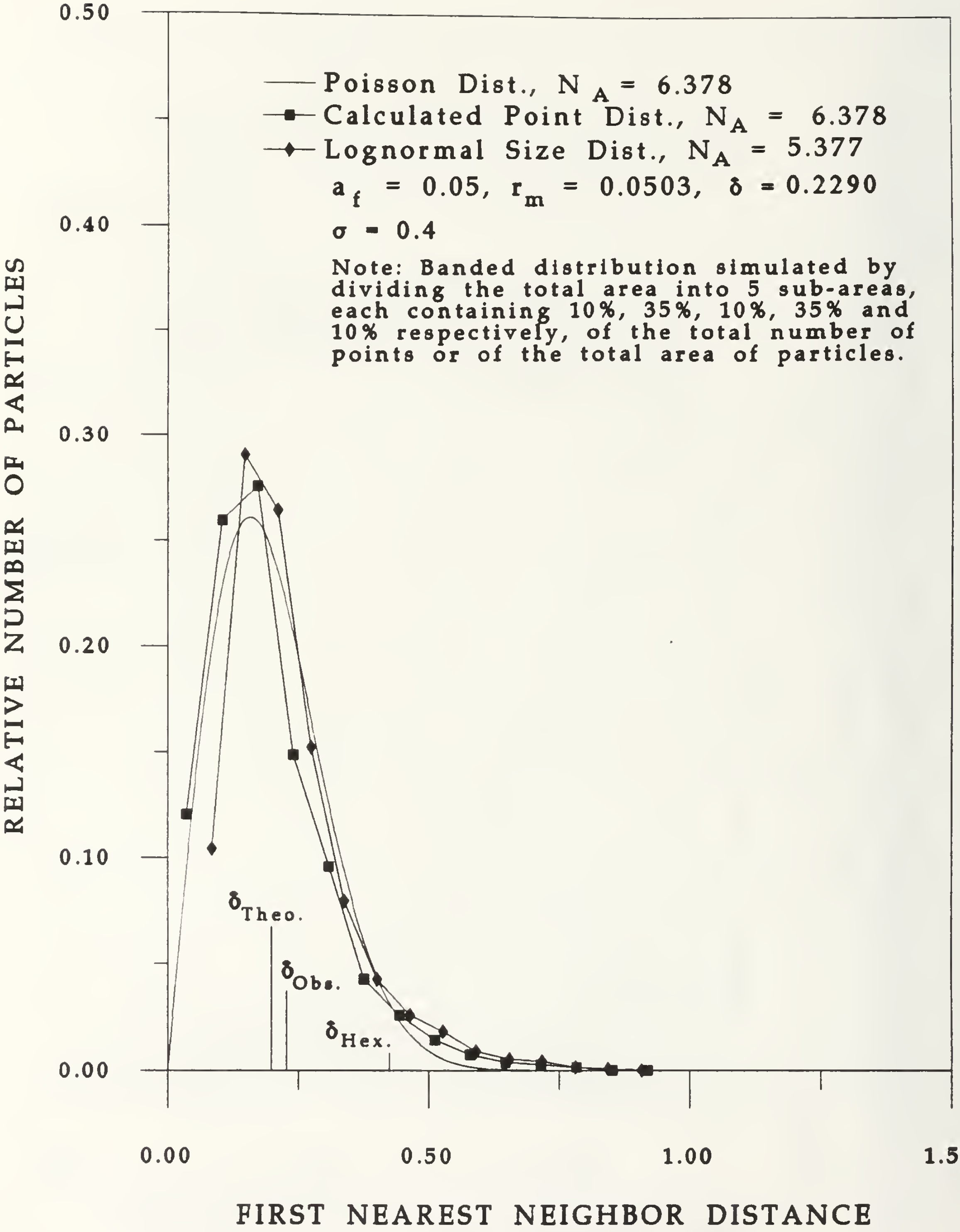


Figure 45. Non-random distribution of points and particles with a lognormal size distribution.

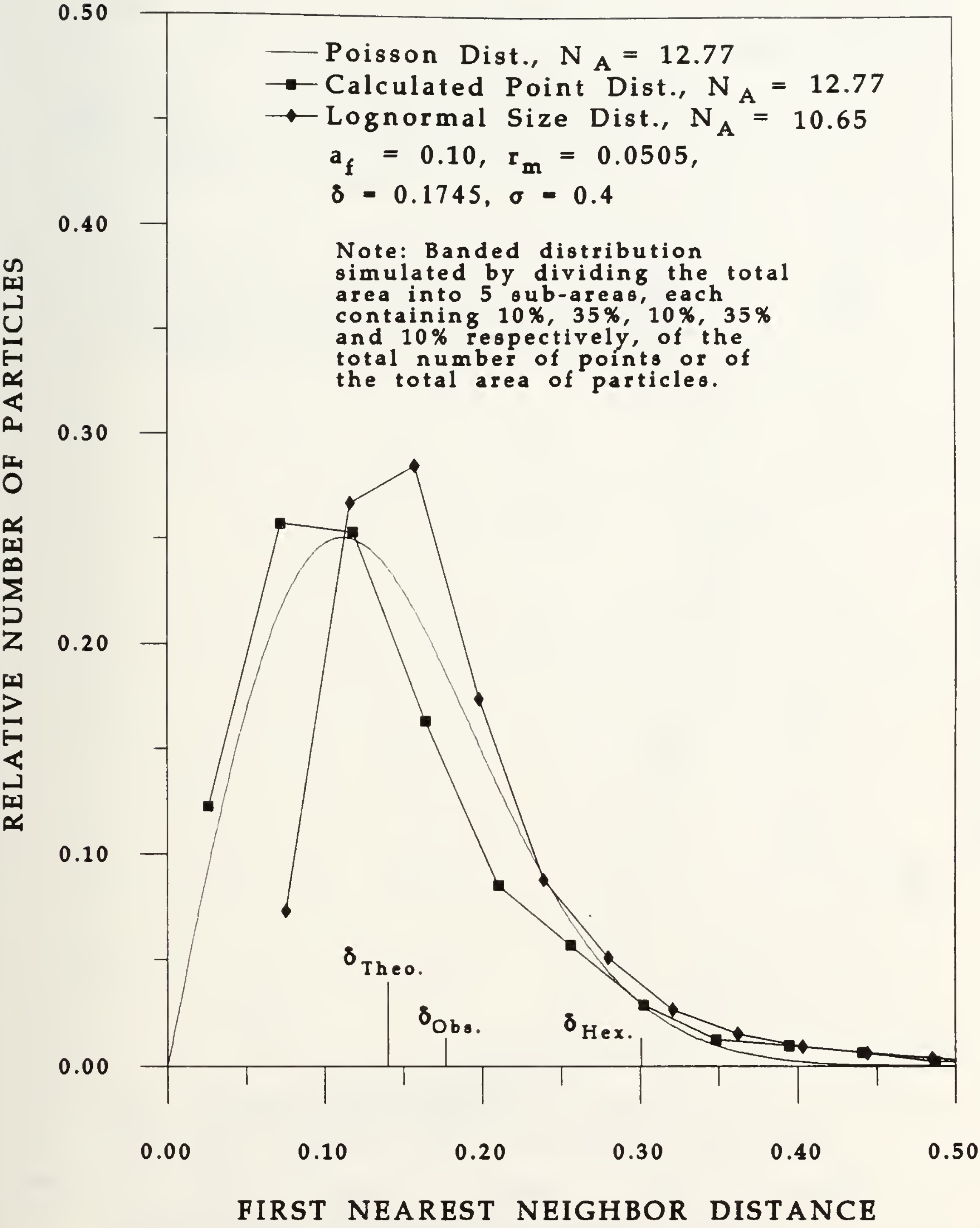


Figure 46. Non-random distribution of points and particles with a lognormal size distribution.

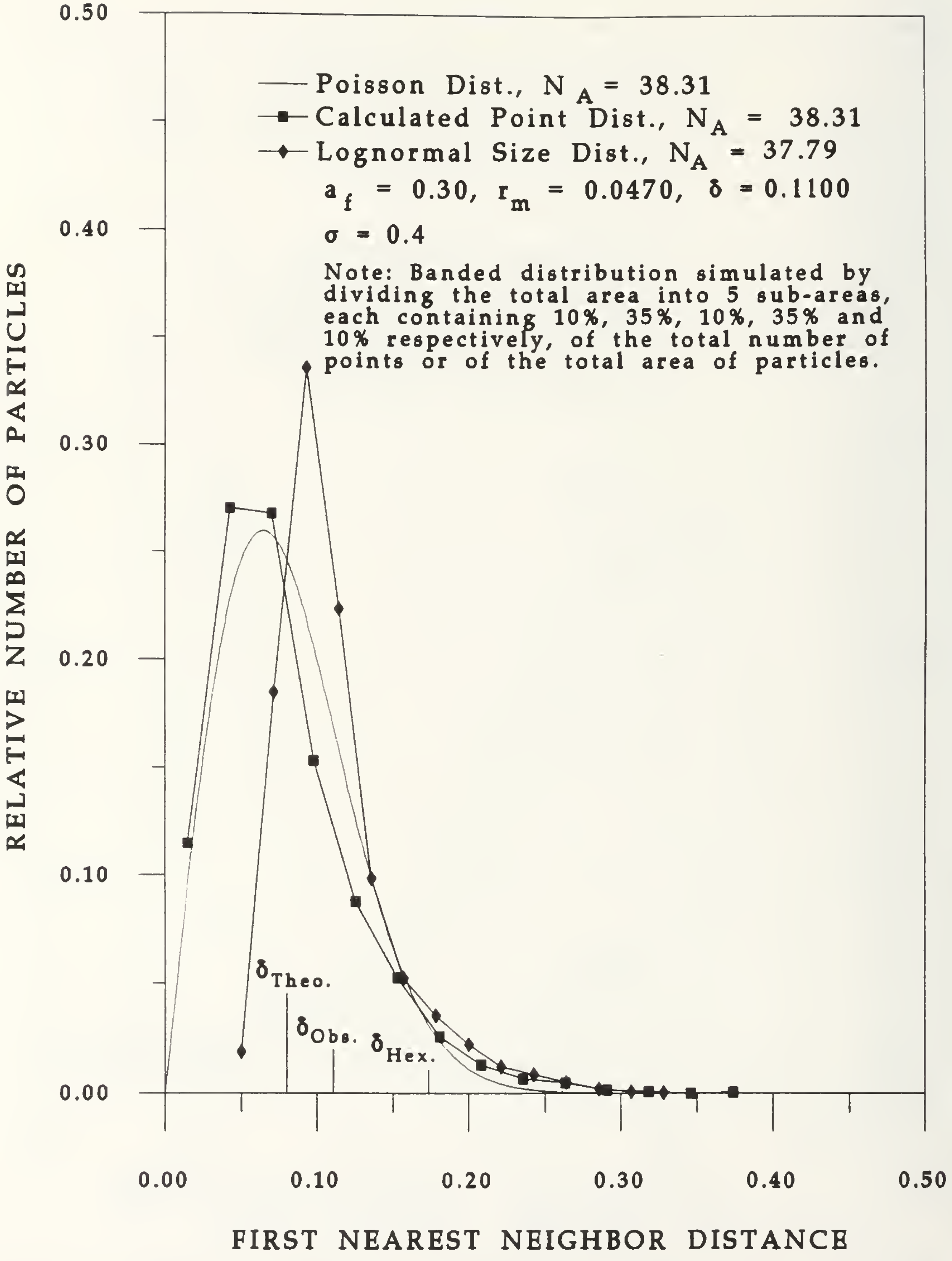
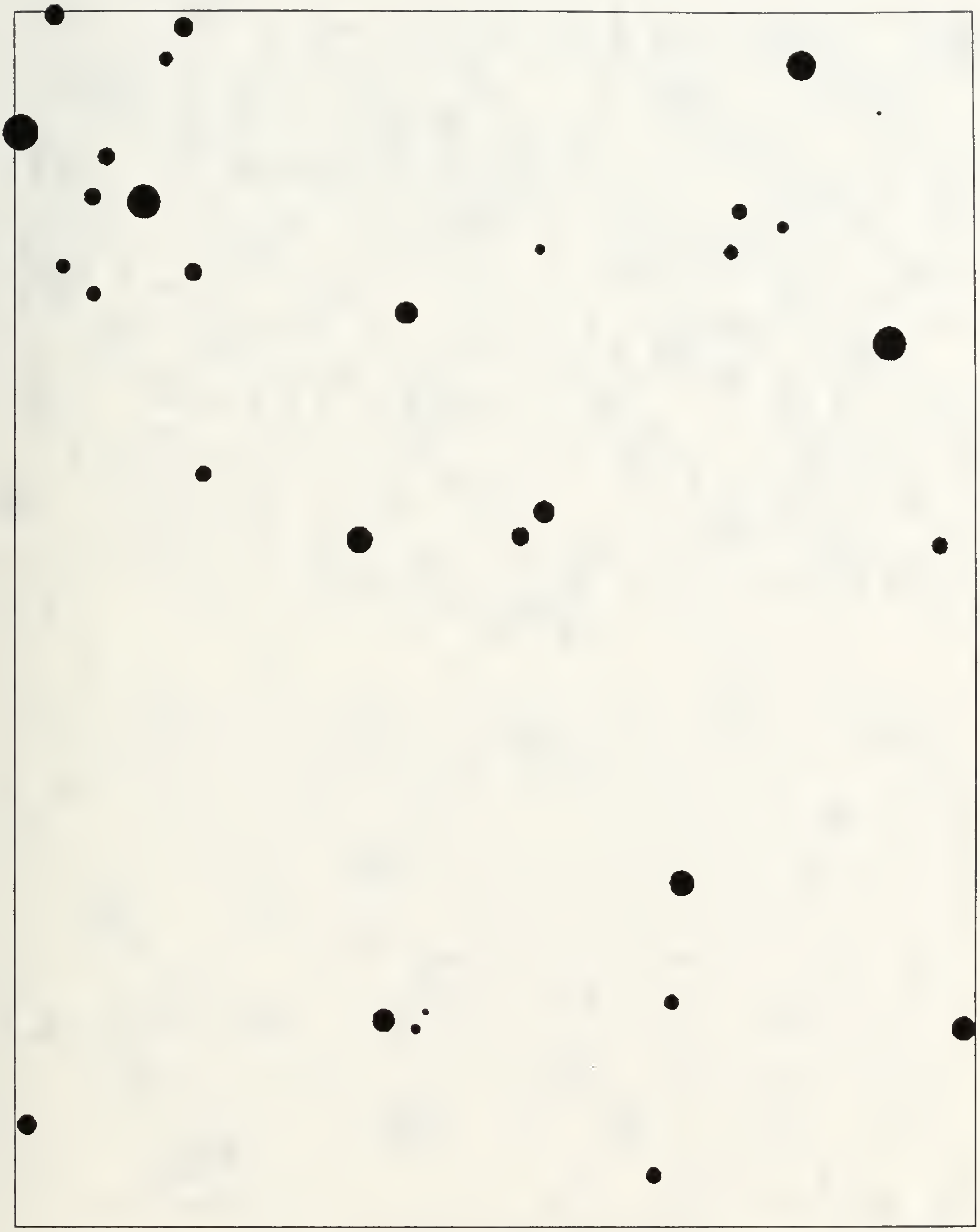
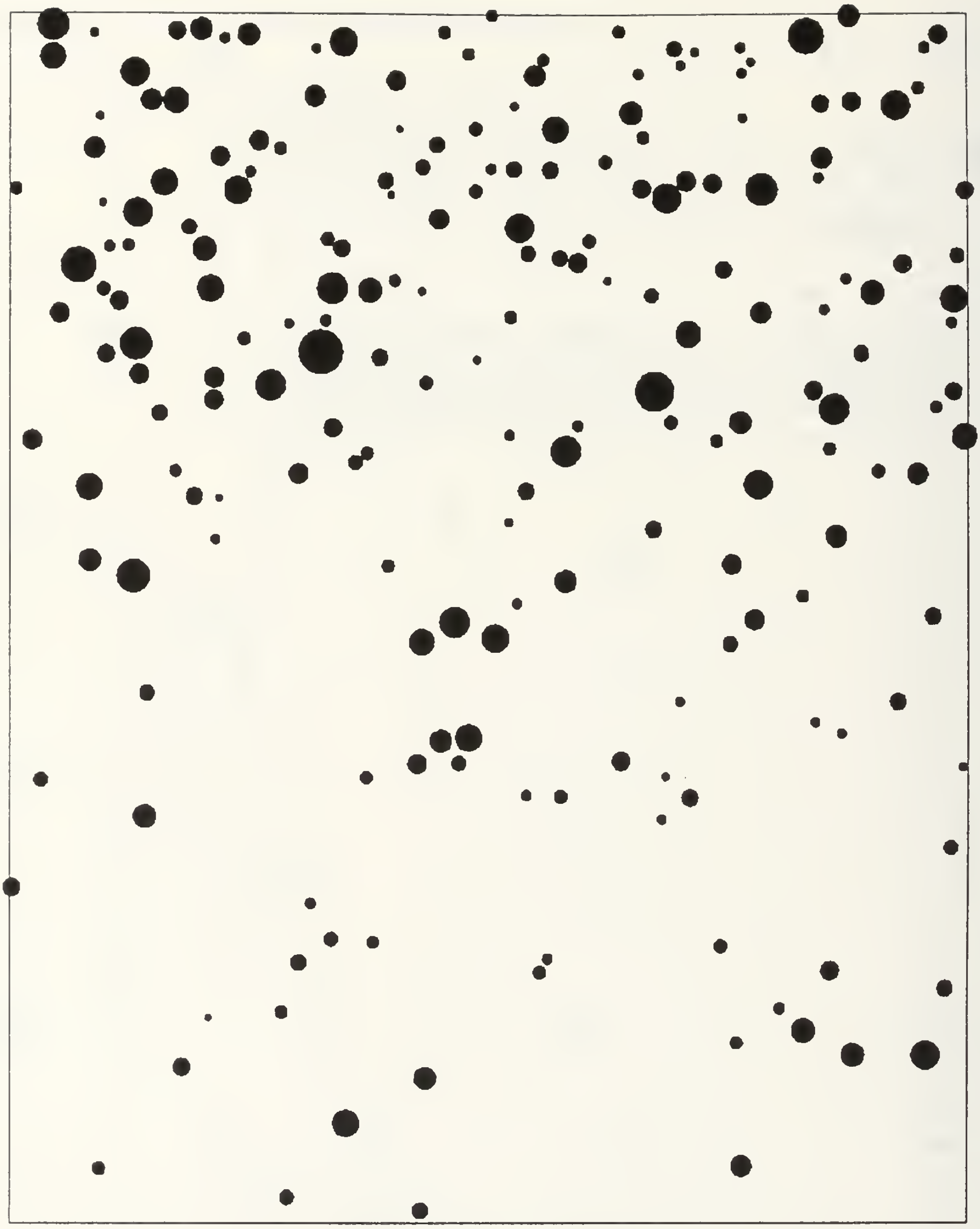


Figure 47. Non-random distribution of points and particles with a lognormal size distribution.



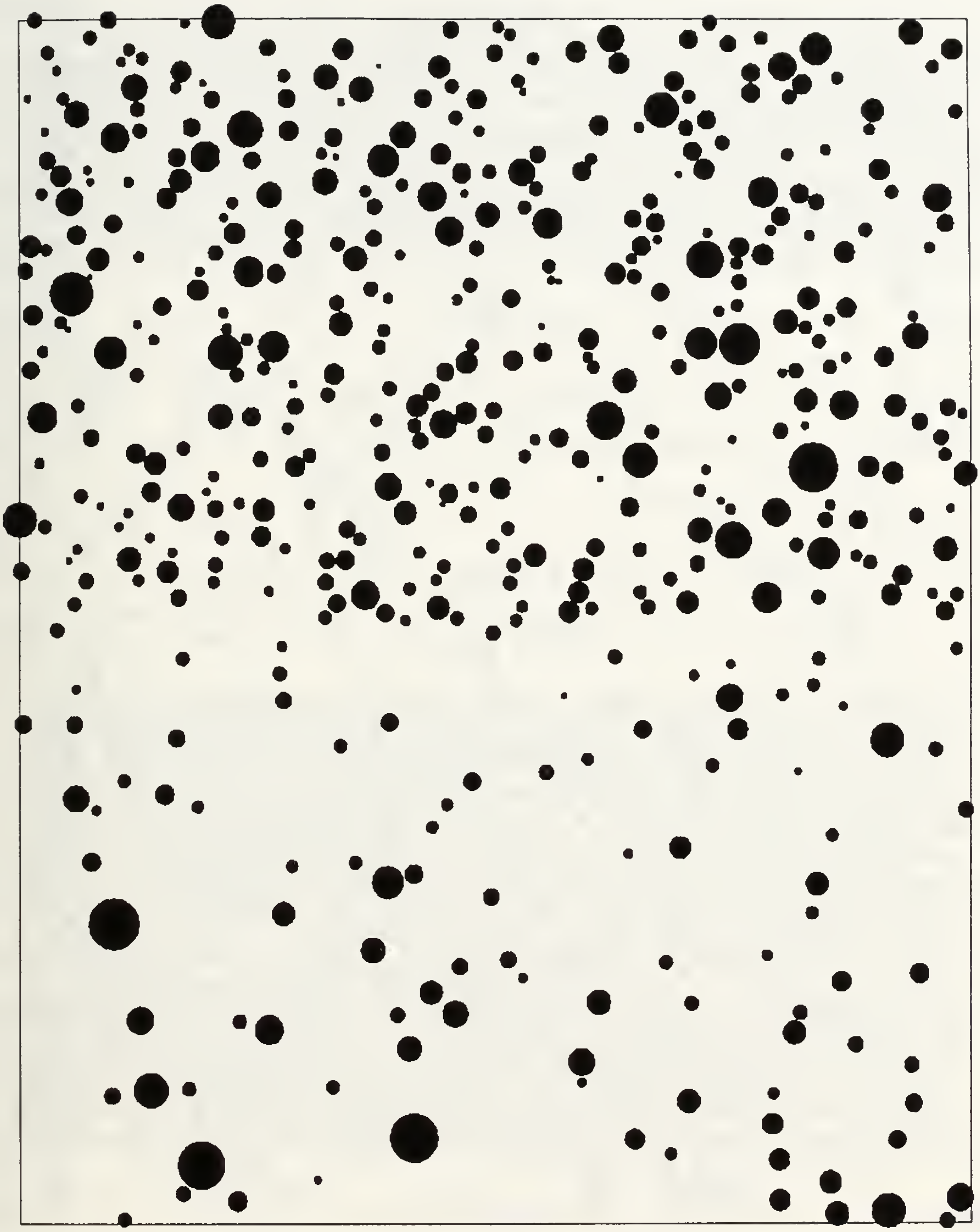
TOTAL PARTICLE AREA FRACTION: 0.01
SUB–AREA PARTICLE AREA FRACTIONS: 0.10, 0.35, 0.10, 0.35, 0.10
MEAN PARTICLE RADIUS: 0.0502
AVERAGE NEAREST NEIGHBOR SPACING: 0.4703
NUMBER PARTICLES PER UNIT AREA: 1.074

Figure 48. Non–random distribution of particles with a log–normal size distribution.



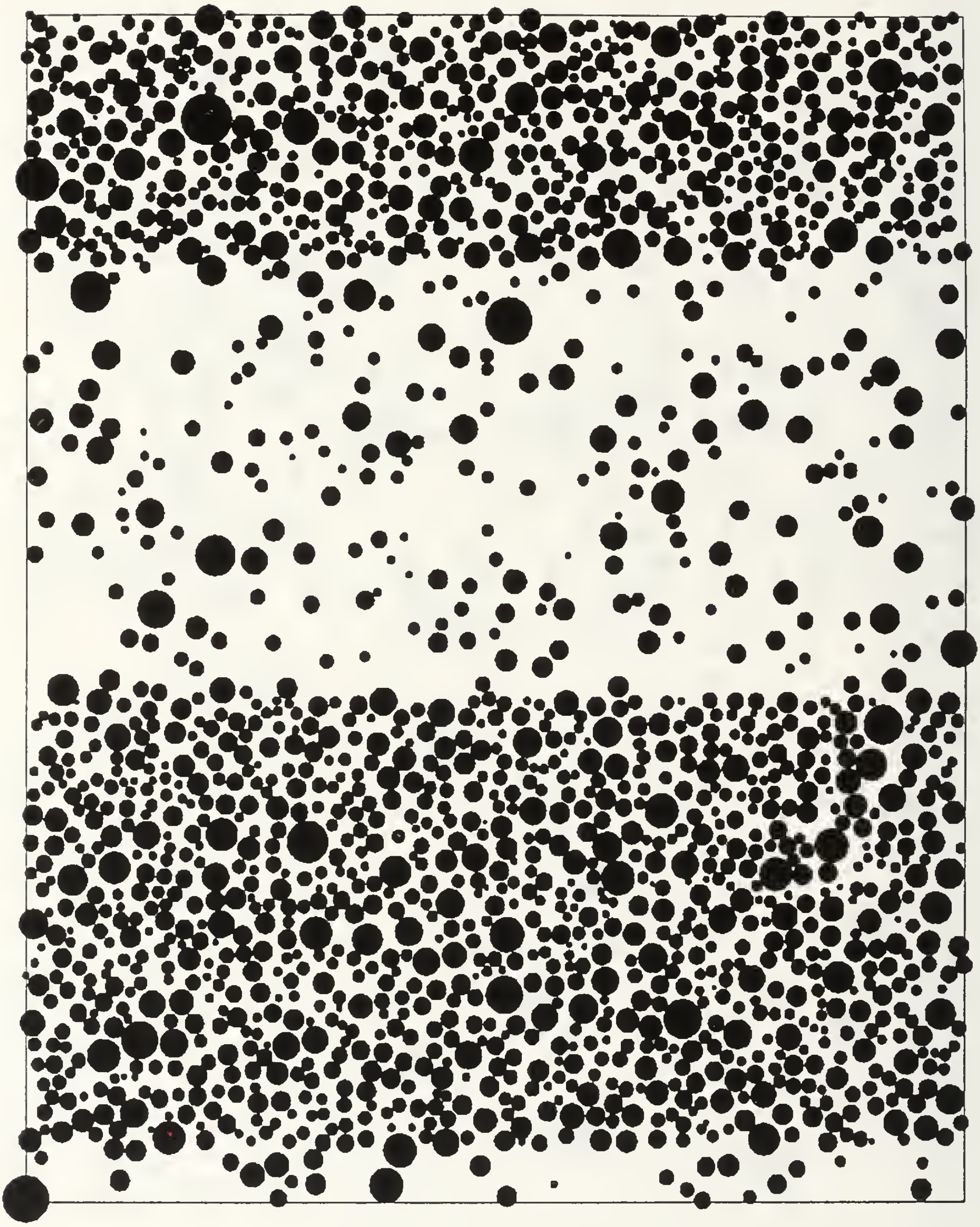
TOTAL PARTICLE AREA FRACTION: 0.05
SUB—AREA PARTICLE AREA FRACTIONS: 0.10, 0.35,
0.10, 0.35, 0.10
MEAN PARTICLE RADIUS: 0.0503
AVERAGE NEAREST NEIGHBOR SPACING: 0.2290
NUMBER PARTICLES PER UNIT AREA: 5.377

Figure 49. Non—random distribution of particles with a log-normal size distribution.



TOTAL PARTICLE AREA FRACTION: 0.10
SUB–AREA PARTICLE AREA FRACTIONS: 0.10, 0.35,
0.10, 0.35, 0.10
MEAN PARTICLE RADIUS: 0.0508
AVERAGE NEAREST NEIGHBOR SPACING: 0.1762
NUMBER PARTICLES PER UNIT AREA: 10.53

Figure 50. Non–random distribution of particles with a log–normal size distribution.



TOTAL PARTICLE AREA FRACTION: 0.30
SUB–AREA PARTICLE AREA FRACTIONS: 0.10, 0.35, 0.10, 0.35, 0.10
MEAN PARTICLE RADIUS: 0.0470
AVERAGE NEAREST NEIGHBOR SPACING: 0.1100
NUMBER PARTICLES PER UNIT AREA: 37.79

Figure 51. Non–random distribution of particles with a log–normal size distribution.

from a very close approximation to the value for a Poisson distribution towards that of a hexagonal array. Unlike the cases for constant size particles, the distribution curve maintains a shape similar to the Poisson distribution and does not reach a peak frequency of first NND corresponding to the minimum first NND. This, again, is due to the possibility of positioning a smaller particle between two or more larger ones. There is a consistent increase in the maximum occurrence of any first NND value over that of the random case. This is also a result of the greater density of particles in the high density zones. Figure 52 summarizes the shift in the distributions as the area fraction increases.

In comparing the particle plots for non-random distributions of particles with a lognormal size distribution, Figures 48 - 51, to those of the random case, the same observations are made as for the constant particle size distributions. For relatively high area fractions the difference between random and non-random particle dispersions is obvious to the eye. But for very low area fractions it becomes more difficult to differentiate between the two. This observation is summarized graphically in Figures 53, 54 and 55. The first two figures represent random and non-random distributions, respectively, for particles of constant and varying size. Both figures are indeed very similar, each showing the effect of smaller particles positioned between

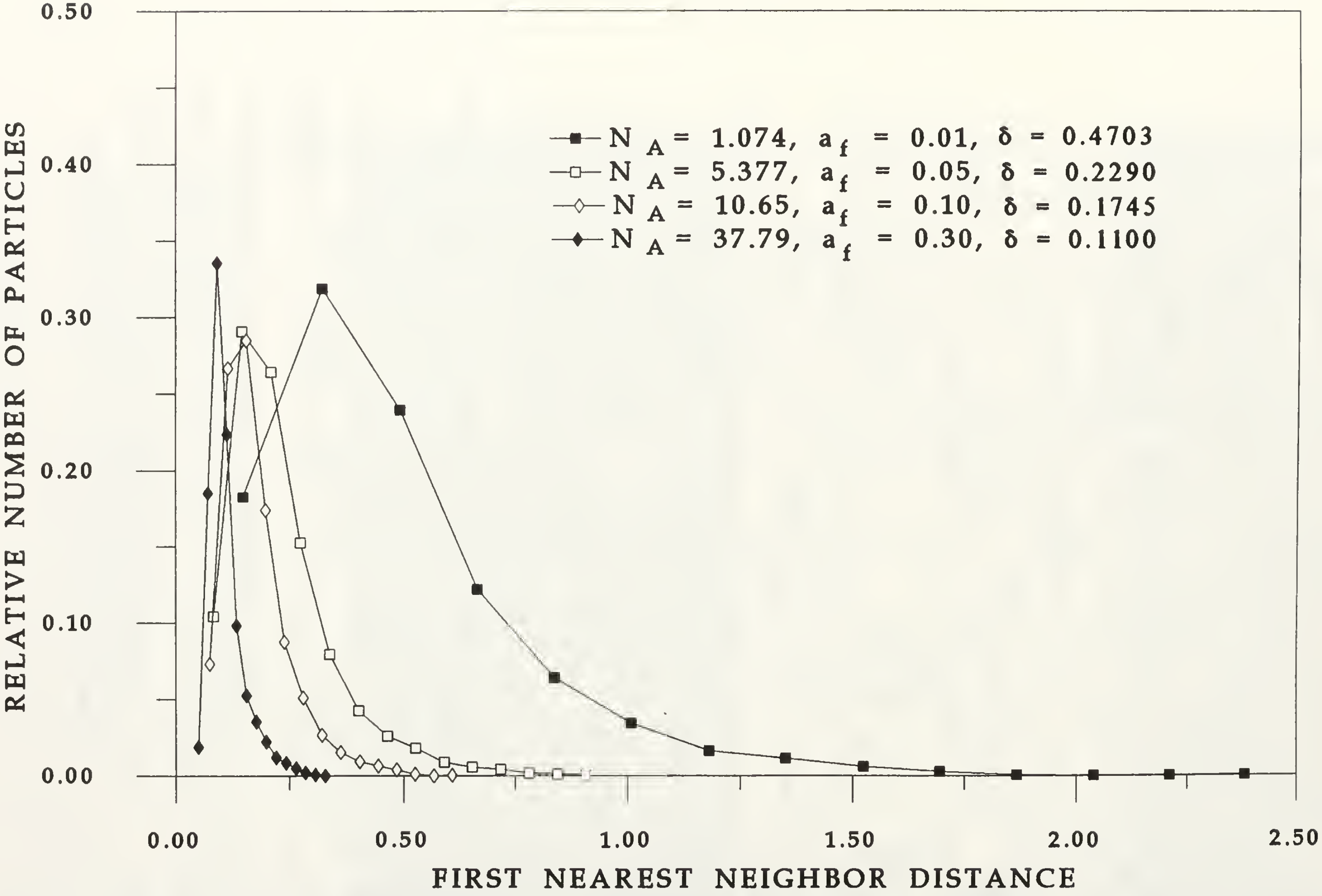


Figure 52. Non-random particle distributions for various area fractions; lognormal particle size distribution; mean particle radius = 0.05; standard deviation = 0.4.

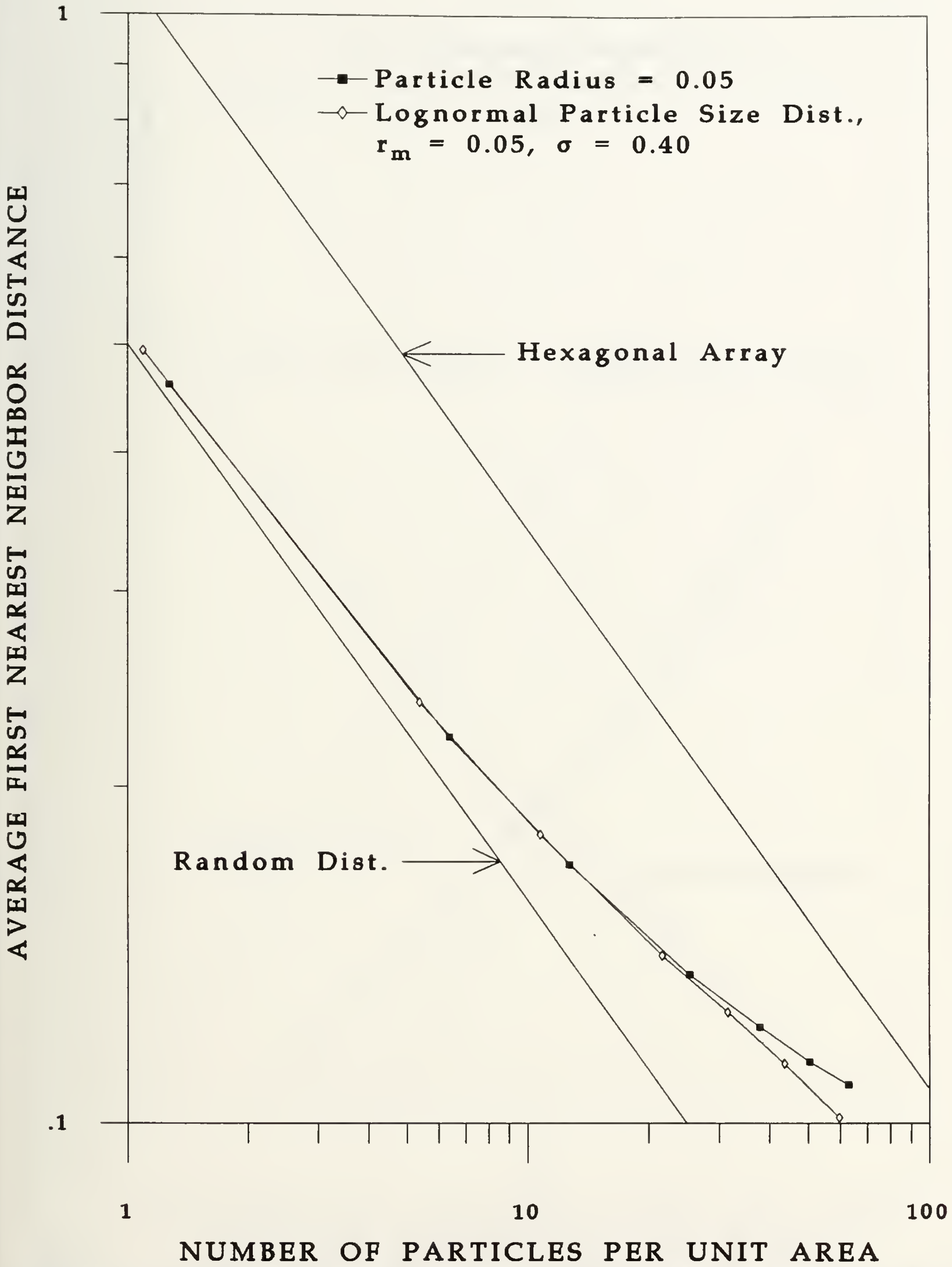


Figure 53. Number of particles per unit area vs average first nearest neighbor distance; for random distributions of constant size particles and for particles with a lognormal size distribution.

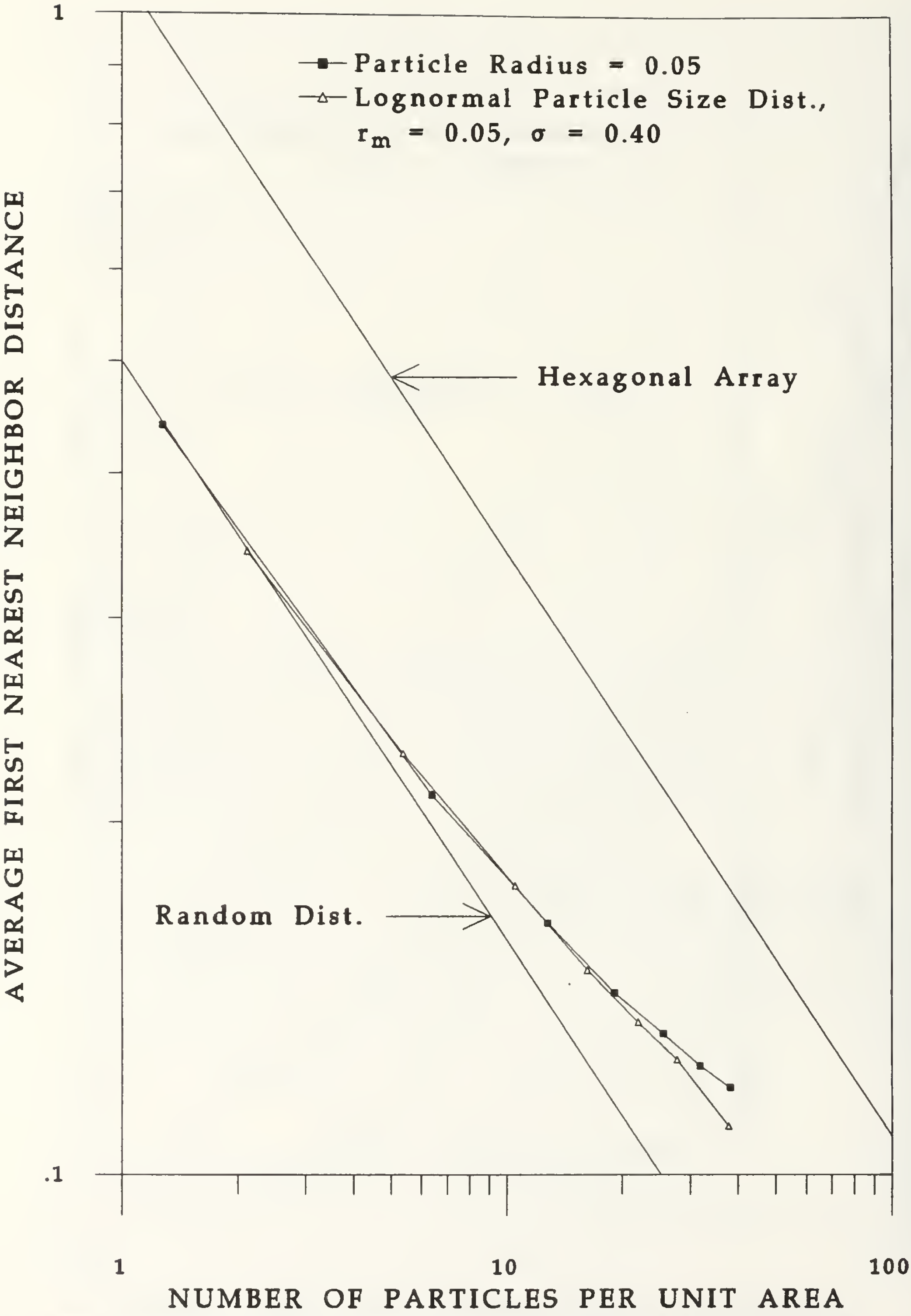


Figure 54. Number of particles per unit area vs average first nearest neighbor distance; for non-random distributions of constant size particles and for particles with a lognormal size distribution.

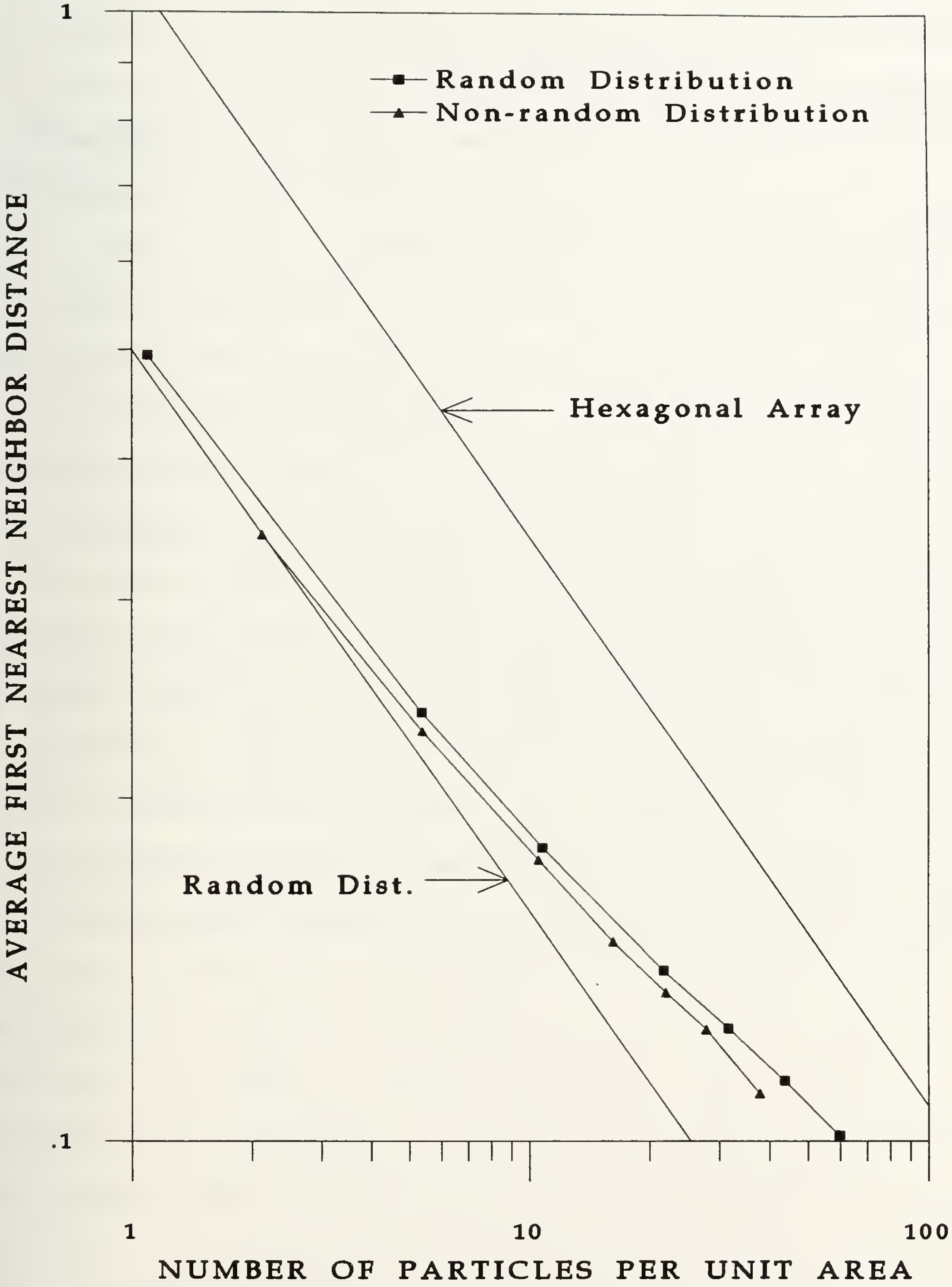


Figure 55. Number of particles per unit area vs average first nearest neighbor distance; for random and non-random distributions of particles with a log normal size distribution, $r_m = 0.05$, $\sigma = 0.4$.

larger ones reducing the value of δ for a given N_A. Further, in comparing the two figures the minimal deviation between random and non-random distributions was reflected in the close similarities of the curves. This observation is more apparent in figure 55 for the case of random and non-random distributions of particles with a lognormal size variance.

The analysis conducted shows that the human eye is very capable of detecting non-randomness of particle distributions for sufficiently dense arrays. However, when presented with the statistical data, the discernment of random or non-random is not so obvious. The degree of non-randomness imposed on the model was judged to be relatively high, 35% and 10% for high and low density bands respectively. Had the difference between the two bands been less, there would have been even greater similarities between the random and non-random distributions. The question then becomes; is it possible to develop a system which can differentiate between random and non-random particle distributions and quantify the results?

V. CONCLUSIONS AND RECOMMENDATIONS

The following conclusions may be drawn from the work described in the preceding sections.

1. The simulations performed for large values of N_A reproduce the work performed by Pas [Ref. 5] and therefore substantiate those results.

2. For more dilute arrays, because the problem of particles along the border was not recognized, Pas's calculations for first NND was overestimated.

3. As the number of particles utilized in any simulation increases, the relative number of particles appearing along the border decreases. For this reason the errors resulting in calculations of first NND can be reduced by using a sufficient number of particles (at least 1000). Calculations of first NND for points approached the theoretical Poisson distribution as the total number of points utilized was increased. The same can be said for the distribution of finite sized particles. However in this case, as the total number of particles increases (approaches 1000 and beyond), δ asymptotically approaches a constant value which lies between the minimum possible spacing ($2 \times r_p$) and the spacing for a hexagonal array. Based on these observations it is possible to compensate for the statistical inaccuracies by utilizing at least 1000 particles in the simulations.

4. Attempts to compensate for errors arising in the first NND calculations when using a small number of particles by eliminating particles along the border was only partially successful. Any gains made using the procedure to reduce the border particle problem for small number of particles was overtaken by simply increasing the total number of particles.

5. In all four cases analyzed (random distribution/constant size particles, non-random distribution/constant size particles, random distribution/lognormal particle size distribution, and non-random distribution/lognormal particle size distribution) the non-overlap criterion caused the distribution to deviate from the Poisson to that of a hexagonal array as the density of the particles increased. Using a nominal value of 5000 particles the model very closely simulated the Poisson distribution in all cases for dilute arrays.

6. For constant size particles in high area fractions ($a_f \geq 0.10$), for both random and non-random distributions, the most frequently occurring first NND was the minimum first NND. This was due to the non-overlap criterion and that the minimum first NND can not be less than the particle diameter.

7. The non-random distribution of points and constant size particles was shifted to the left of the corresponding Poisson distribution. This was due to the presence of the high density zones. For a given N_A, the closer packing of the

points or particles in the high density zones causes NND to shift towards lower values. This was also reflected in a greater frequency of the minimum first NND when compared to that of the random case.

8. The random distribution of particles with a lognormal size distribution maintained the shape of a Poisson field for all area fractions. Unlike the case for constant size particles, the minimum first NND can be smaller than the mean particle diameter. A smaller particle can always fit between two or more larger ones thus lowering the minimum first NND. This also resulted in a more constant peak frequency of first NND for all ranges of area fractions examined. These observations held true as well for the case of non-random/lognormal particle size distribution.

After reviewing all the accumulated data there did not appear to be as significant a difference between the plotted distributions of the random and non-random cases as there was between the plots of the particles themselves. Based then on the statistical nature of this work it appears that the best differentiater between random and non-random particle distributions is the human eye.

To follow up on this work the following studies are recommended. A similar analysis should be made for the three dimensional case in order to parallel the actual particle distributions. Other types of non-randomness, such as clusters, should be explored to determine their effects on distributions. Area fractions greater than 0.50 should be examined as well. Finally, an analysis should be made, based on results of this work, to determine the feasibility of developing a computer program which could determine the degree of randomness in the distributions of second phase particles.

APPENDIX A. PROGRAM RANPART1.FOR

```
      PROGRAM RANPART1
*     Glossary of variables:
*     Pi           3.14159
*     Xpos         X coordinate of an interior particle
*     Ypos         Y coordinate of an interior particle
*     XposB        X coordinate of a particle along the
*                  border
*     YposB        Y coordinate of a particle along the
*                  border
*     AP           Area percentage of the particles
*     TPA          Total area fraction of the particles
*     Pspace       Min. spacing between particles
*     Pradius      Particle radius
*    TOTarea      Defined area where particles are positioned
*     Parea        Area occupied by each particle
*     dp           Distance between interior particles
*     dp2          Distance between interior particles
*    dp2B         Distance between an interior particle and a
*                  border particle
*     dp3          Nearest neighbor spacing
*     dp4          Sum of nearest neighbor spacings
*     dist         Particle diameter plus min spacing
*     dpavg        Average nearest neighbor spacing
*     X            Horizontal dimension of defined area
*     Y            Vertical dimension of defined area
*     dpmin        Minimum nearest neighbor spacing
*      dp2Bmin      Distance between an interior particle and
*                  its nearest border particle
*     dpmax        Maximum nearest neighbor spacing
*     dpinc        Size of each range between dpmin and dpmax
*                  based on the Sturgis Rule
*     Delta        Minimum value of each range in dpinc
*      Np           Number of particles corrosponding to each
*                  Delta
*     NPA          Number of particles per unit area
*     Pnum         Total number of particles generated
*     Ans          Either "A" for area or "p" for points,
*                  depending on
*                  type of particles
*    Part         Either "n" for a preset number of particles
*                  or "r" for a preset particle radius
*
      REAL Pi, Xpos(5200), Ypos(5200), XposB(250), YposB(250),
     :TPA, AP, Pspace, Pradius, TOTarea, Parea, dp, dp2, dp2B,
     :dp2Bmin, dp3(5200), dp4, dist, dpavg, X, Y, dpmin, NPA,
     :dpmax, dpinc, g, Delta(30), Np(30)
```

```
*
      INTEGER a, b, c, d, i, j, k, l, m, n, Pnum, f, h, e
*
      CHARACTER Ans*1, Part*1
*
      PARAMETER (Pi=3.14159)
*
      DATA n, dp4, e, dpmin, dpmax, Np(1), a, b /
     :1, 0.0, 0, 10.0, 0.001, 0, 1, 1 /
*
      OPEN (10,file='rp1x.dat',status='new')
      OPEN (20,file='rp1y.dat',status='new')
      OPEN (30,file='Del1.dat',status='new')
      OPEN (40,file='Np1.dat',status='new')
      OPEN (50,file='rp1.dat',status='new')
*
      PRINT*,' Enter the area percentage of particles.'
      READ*, AP
      PRINT*,' Enter the length of the X - scale.'
      READ*, X
      PRINT*,' Enter the length of the Y - scale.'
      READ*, Y
      TOTarea = X*Y
      TPA = AP/100*TOTarea
      PRINT*,' Enter the minimum spacing between particles.'
      READ*, Pspace
      PRINT*,' If particle generation is to based on a preset
     :number of'
      PRINT*,' particles enter "n".  OR  If particle
     :generation is to'
      PRINT*,' be based on a preset particle radius enter
     :"r".'
      READ*,Part
      IF (Part .EQ. 'n') THEN
      PRINT*,'Enter total number of particles.'
      READ*,Pnum
      Parea = TPA/Pnum
      Pradius = SQRT(Parea/Pi)
      ELSEIF (Part .EQ. 'r') THEN
      PRINT*,' Enter the radius of the particles.'
      READ*, Pradius
      Parea = Pi*(Pradius**2)
      Pnum = NINT(TPA/Parea)
      ENDIF
      PRINT*,' Are the particles finite areas or points?
     :Enter "A" or "p".'
      READ*, Ans
*
*     Determine random position of particles.
*
      PRINT*,' Determining random position of particles.'
      dist = 2*Pradius + Pspace
      Xpos(1) = RRAND()
      DO 100 i = 1, Pnum
150   CONTINUE
        Xpos(i) = RND()*X
        Ypos(i) = RND()*Y
*
*     Check for overlap of particles with a finite area
*
        IF (Ans .EQ. 'p') GOTO 250
        DO 200 k = 1, i - 1
          dp = SQRT((Xpos(i) - Xpos(k))**2 +
     :               (Ypos(i) - Ypos(k))**2)
          IF (dp .LT. dist) GOTO 150
200       CONTINUE
250     CONTINUE
100   CONTINUE
*
*     Segregate particles along the perimeter from those in
*     the interior of the specified area.
*
      PRINT*,' Segregating particles on the border from the
     :interior.'
      DO 275 d = 1, Pnum
        IF ((Xpos(d) .LE. Pradius + Pspace) .OR.
     :       (Xpos(d) .GE. X - Pradius - Pspace) .OR.
     :       (Ypos(d) .LE. Pradius + Pspace) .OR.
     :       (Ypos(d) .GE. Y - Pradius - Pspace)) THEN
          XposB(b) = Xpos(d)
          YposB(b) = Ypos(d)
          b = b + 1
        ELSE
          Xpos(a) = Xpos(d)
          Ypos(a) = Ypos(d)
          WRITE(10,*) Xpos(a)
          WRITE(20,*) Ypos(a)
          a = a + 1
        ENDIF
275   CONTINUE
*
*     Determine the average distance between nearest
*     neighbors.
*
      PRINT*,' Determining nearest neighbor distances.'
      DO 300 j = 1, Pnum - (b - 1)
        dp3(n) = SQRT(X**2 + Y**2)
        dp2Bmin = SQRT(X**2 + Y**2)
        DO 350 c = 1, (b - 1)
          dp2B = SQRT((Xpos(j) - XposB(c))**2 +
     :                 (Ypos(j) - YposB(c))**2)
          dp2Bmin = MIN(dp2Bmin, dp2B)
```

```
350     CONTINUE
        DO 400 l = 1, Pnum - (b - 1)
          dp2 = SQRT((Xpos(l) - Xpos(j))**2 +
     :                (Ypos(l) - Ypos(j))**2)
          IF (dp2 .NE. 0.0) THEN
            dp3(n) = MIN(dp2, dp3(n), dp2Bmin)
          ENDIF
400       CONTINUE
        n = n + 1
300   CONTINUE
      DO 500 m = 1, Pnum - (b - 1)
        dp4 = dp4 + dp3(m)
500   CONTINUE
      dpavg = dp4/(Pnum - (b - 1))
*
*    Determine the number of particles falling in each range
*     of nearest neighbor spacings (based on the Sturgis
*     rule).
*
      PRINT*,' Grouping nearest neighbor values into ranges.'
      DO 600 f = 1, Pnum - 1 - (b - 1)
        dpmin = MIN(dp3(f), dp3(f+1), dpmin)
        dpmax = MAX(dp3(f), dp3(f+1), dpmax)
600   CONTINUE
      dpinc = (dpmax - dpmin)/(1 + 3.3*LOG10(Pnum - (b - 1)))
      DO 700 g = dpmin, dpmax, dpinc
        e = e + 1
        DO 800 h = 1, Pnum - (b - 1)
          IF ((dp3(h) .GE. g) .AND. (dp3(h) .LT. g + dpinc))
     :    THEN
           Np(e) = Np(e) + 1
          ENDIF
800     CONTINUE
        Delta(e) = (2*g + dpinc)/2
        WRITE(30,*)Delta(e)
        WRITE(40,*)Np(e)/(Pnum - (b - 1))
700   CONTINUE
      NPA = (Pnum - (b - 1))/(X - 2*Pradius - 2*Pspace)/
     :      (Y - 2*Pradius - 2*Pspace)
      WRITE(50,10)'The area percentage of particles is', AP
      WRITE(50,20)'X dimension is',X
      WRITE(50,20)'Y dimension is',Y
      WRITE(50,20)'Min spacing between particles is',Pspace
      WRITE(50,20)'Particle radius is', Pradius
      WRITE(50,20)'Particle area is', Parea
      WRITE(50,20)'Total number of particles is', Pnum
      WRITE(50,20)'Number of particles along the perimeter
     :is',b - 1
      WRITE(50,20)'Number of particles in area interior is',a
     :- 1
     WRITE(50,20)'Number particles per unit area', NPA
```

```
       WRITE(50,20)'Average spacing between nearest neighbors
      :is', dpavg
       WRITE(50,20)'Min nearest neighbor spacing is',dpmin
       WRITE(50,20)'Max nearest neighbor spacing is',dpmax
       PRINT*,' The area percentage of particles is', AP
       PRINT*,' X dimension is',X
       PRINT*,' Y dimension is',Y
       PRINT*,' Min spacing between particles is',Pspace
       PRINT*,' Particle radius is', Pradius
       PRINT*,' Particle area is', Parea
       PRINT*,' Total number of particles is', Pnum
       PRINT*,' Number of particles along the perimeter', b - 1
       PRINT*,' Number of particles in area interior', a - 1
       PRINT*,' Number particles per unit area is', NPA
       PRINT*,' Average spacing between nearest neighbors is',
      :dpavg
       PRINT*,' Min nearest neighbor spacing is',dpmin
       PRINT*,' Max nearest neighbor spacing is',dpmax
10     FORMAT ( //////,T8,A,F10.4)
20     FORMAT (T8,A,F10.4)
       END
```

APPENDIX B. PROGRAM RANPART2.FOR

```
      PROGRAM RANPART2
*
*     Glossary of variables:
*     Pi            3.14159
*     Xpos          X coordinate of an interior particle
*     Ypos          Y coordinate of an interior particle
*    XposB         X coordinate of a particle along the border
*    YposB         Y coordinate of a particle along the border
*     AP            Area percentage of the particles
*     TPA           Total area fraction of the particles
*     Pspace        Min. spacing between particles
*     Pradius       Particle radius
*     TOTarea       Defined area of particles
*     Parea         Area occupied by each particle
*     dp            Distance between interior particles
*     dp2           Distance between interior particles
*    dp2B          Distance between an interior particle and a
*                   border particle
*     dp3           Nearest neighbor spacing
*     dp4           Sum of nearest neighbor spacings
*     dist          Particle diameter plus min spacing
*     dpavg         Average nearest neighbor spacing
*     X             Horizontal dimension of total area
*     Y             Vertical dimension of total area
*     dpmin         Minimum nearest neighbor spacing
*     dp2Bmin       Distance between an interior particle and
*                   its nearest border particle
*     dpmax         Maximum nearest neighbor spacing
*     dpinc         Size of each range between dpmin and dpmax
*                   based on the Sturgis Rule
*     Delta         Minimum value of each range in dpinc
*     Np            Number of particles corrosponding to each
*                   Delta
*     NPA           Number of particles per unit area
*     Pnum          Total number of particles generated
*     Ans           Either "A" for area or "p" for points,
*                   depending on type of particles
*    Part          Either "n" for a preset number of particles
*                   or "r" for a preset particle radius
*     AF1-5         Area percentage of particles in each of the
*                   five sub-areas
*     A1-5          Subtotal of particles in each sub-area
*
      REAL Pi, Xpos(5200), Ypos(5200), XposB(250), YposB(250),
     :TPA, AP, Pspace, Pradius, TOTarea, Parea, dp, dp2, dp2B,
     :dp2Bmin, dp3(5200), dp4, dist, dpavg, X, Y, dpmin, NPA,
     :dpmax, dpinc, g, Delta(30), Np(30), A1, AF1, A2, AF2,
     :A3, AF3, A4, AF4, A5, AF5
*
      INTEGER a, b, c, d, i, j, k, l, m, n, Pnum, f, h, e
*
      CHARACTER Ans*1, Part*1
*
      PARAMETER (Pi=3.14159)
*
      DATA n, dp4, e, dpmin, dpmax, Np(1), a, b, A1, A2, A3,
     :A4, A5 /
     :1, 0.0, 0, 10.0, 0.001, 0, 1, 1, 0.0, 0.0, 0.0, 0.0,
     :0.0/
*
      OPEN (10,file='rp2x.dat',status='new')
      OPEN (20,file='rp2y.dat',status='new')
      OPEN (30,file='Del2.dat',status='new')
      OPEN (40,file='Np2.dat',status='new')
      OPEN (50,file='rp2.dat',status='new')
*
      PRINT*,' Enter the area percentage of particles.'
      READ*, AP
      PRINT*,' Enter 5 sub-area percentages of particles.'
      READ*, AF1, AF2, AF3, AF4, AF5
      AF1 = AF1/100
      AF2 = AF2/100
      AF3 = AF3/100
      AF4 = AF4/100
      AF5 = AF5/100
      PRINT*,' Enter the length of the X - scale.'
      READ*, X
      PRINT*,' Enter the length of the Y - scale.'
      READ*, Y
      TOTarea = X*Y
      TPA = AP/100*TOTarea
      PRINT*,' Enter the minimum spacing between particles.'
      READ*, Pspace
      PRINT*,' If particle generation is to based on a preset
     :number of'
      PRINT*,' particles enter "n".  OR  If particle
     :generation is to'
      PRINT*,' be based on a preset particle radius enter
     :"r".'
      READ*,Part
      IF (Part .EQ. 'n') THEN
        PRINT*,'Enter total number of particles.'
        READ*,Pnum
        Parea = TPA/Pnum
        Pradius = SQRT(Parea/Pi)
      ELSEIF (Part .EQ. 'r') THEN
        PRINT*,' Enter the radius of the particles.'
```

```
          READ*, Pradius
          Parea = Pi*(Pradius**2)
          Pnum = NINT(TPA/Parea)
        ENDIF
        PRINT*,' Are the particles finite areas or points?
       :Enter "A" or "p".'
        READ*, Ans
*
*       Determine random position of particles.
*
        PRINT*,' Determining random-banded position of
       :particles.'
        dist = 2*Pradius + Pspace
        Xpos(1) = RRAND()
        DO 100 i = 1, Pnum
150         CONTINUE
          Xpos(i) = RND()*X
          Ypos(i) = RND()*Y
*
*           Check for overlap of particles with a finite area
*
          IF (Ans .EQ. 'p') GOTO 250
          DO 200 k = 1, i - 1
            dp = SQRT((Xpos(i) - Xpos(k))**2 +
       :               (Ypos(i) - Ypos(k))**2)
            IF (dp .LT. dist) GOTO 150
200         CONTINUE
250     CONTINUE
        IF ((Ypos(i) .GE. 0.0) .AND. (Ypos(i) .LT. 0.2*Y)) THEN
          IF (A1 .GT. TPA*AF1) GOTO 150
          A1 = A1 + Parea
        ELSEIF ((Ypos(i) .GE. 0.2*Y) .AND. (Ypos(i) .LT. 0.4*Y))
       :THEN
          IF (A2 .GT. TPA*AF2) GOTO 150
          A2 = A2 + Parea
        ELSEIF ((Ypos(i) .GE. 0.4*Y) .AND. (Ypos(i) .LT. 0.6*Y))
       :THEN
          IF (A3 .GT. TPA*AF3) GOTO 150
          A3 = A3 + Parea
        ELSEIF ((Ypos(i) .GE. 0.6*Y) .AND. (Ypos(i) .LT. 0.8*Y))
       :THEN
          IF (A4 .GT. TPA*AF4) GOTO 150
          A4 = A4 + Parea
        ELSEIF ((Ypos(i) .GE. 0.8) .AND. (Ypos(i) .LE. Y)) THEN
          IF (A5 .GT. TPA*AF5) GOTO 150
          A5 = A5 + Parea
        ENDIF
100     CONTINUE
*
*       Segregate particles along the perimeter from those in
*       the interior of the specified area.
```

```
*
      PRINT*,' Segregating particles on the border from the
     :interior.'
      DO 275 d = 1, Pnum
        IF ((Xpos(d) .LE. Pradius + Pspace) .OR.
     :      (Xpos(d) .GE. X - Pradius - Pspace) .OR.
     :      (Ypos(d) .LE. Pradius + Pspace) .OR.
     :      (Ypos(d) .GE. Y - Pradius - Pspace)) THEN
          XposB(b) = Xpos(d)
          YposB(b) = Ypos(d)
          b = b + 1
        ELSE
          Xpos(a) = Xpos(d)
          Ypos(a) = Ypos(d)
          WRITE(10,*) Xpos(a)
          WRITE(20,*) Ypos(a)
          a = a + 1
        ENDIF
275   CONTINUE
*
*     Determine the average distance between nearest
*     neighbors.
*
      PRINT*,' Determining nearest neighbor distances.'
      DO 300 j = 1, Pnum - (b - 1)
        dp3(n) = SQRT(X**2 + Y**2)
        dp2Bmin = SQRT(X**2 + Y**2)
        DO 350 c = 1, (b - 1)
          dp2B = SQRT((Xpos(j) - XposB(c))**2 +
     :                    (Ypos(j) - YposB(c))**2)
          dp2Bmin = MIN(dp2Bmin, dp2B)
350     CONTINUE
        DO 400 l = 1, Pnum - (b - 1)
          dp2 = SQRT((Xpos(l) - Xpos(j))**2 +
     :                   (Ypos(l) - Ypos(j))**2)
          IF (dp2 .NE. 0.0) THEN
           dp3(n) = MIN(dp2, dp3(n), dp2Bmin)
          ENDIF
400       CONTINUE
        n = n + 1
300   CONTINUE
      DO 500 m = 1, Pnum - (b - 1)
        dp4 = dp4 + dp3(m)
500   CONTINUE
      dpavg = dp4/(Pnum - (b - 1))
*
*     Determine the number of particles falling in each range
*      of nearest neighbor spacings (based on the Sturgis
*     rule).
*
      DO 600 f = 1, Pnum - 1 - (b - 1)
```

```
          dpmin = MIN(dp3(f), dp3(f+1), dpmin)
          dpmax = MAX(dp3(f), dp3(f+1), dpmax)
600     CONTINUE
        dpinc = (dpmax - dpmin)/(1 + 3.3*LOG10(Pnum - (b - 1)))
        DO 700 g = dpmin, dpmax, dpinc
          e = e + 1
          DO 800 h = 1, Pnum - (b - 1)
            IF ((dp3(h) .GE. g) .AND. (dp3(h) .LT. g + dpinc))
     :      THEN
             Np(e) = Np(e) + 1
            ENDIF
800       CONTINUE
          Delta(e) = (2*g + dpinc)/2
          WRITE(30,*)Delta(e)
          WRITE(40,*)Np(e)/(Pnum - (b - 1))
700     CONTINUE
        NPA = (Pnum - (b - 1))/(X - 2*Pradius - 2*Pspace)/
     :        (Y - 2*Pradius - 2*Pspace)
        WRITE(50,10)'The area percentage of particles is', AP
        WRITE(50,20)'X dimension is',X
        WRITE(50,20)'Y dimension is',Y
        WRITE(50,20)'Min spacing between particles is',Pspace
        WRITE(50,20)'Particle radius is', Pradius
        WRITE(50,20)'Particle area is', Parea
        WRITE(50,20)'Total number of particles is', Pnum
        WRITE(50,20)'Number of particles along the perimeter
     :is',b - 1
        WRITE(50,20)'Number of particles in area interior is',a
     :- 1
        WRITE(50,20)'Number particles per unit area', NPA
        WRITE(50,20)'Average spacing between nearest neighbors
     :is', dpavg
        WRITE(50,20)'Min nearest neighbor spacing is',dpmin
        WRITE(50,20)'Max nearest neighbor spacing is',dpmax
        PRINT*,' The area percentage of particles is', AP
        PRINT*,' X dimension is',X
        PRINT*,' Y dimension is',Y
        PRINT*,' Min spacing between particles is',Pspace
        PRINT*,' Particle radius is', Pradius
        PRINT*,' Particle area is', Parea
        PRINT*,' Total number of particles is', Pnum
        PRINT*,' Number of particles along the perimeter', b -
     :1
        PRINT*,' Number of particles in area interior', a - 1
        PRINT*,' Number particles per unit area is', NPA
        PRINT*,' Average spacing between nearest neighbors is',
     :dpavg
        PRINT*,' Min nearest neighbor spacing is',dpmin
        PRINT*,' Max nearest neighbor spacing is',dpmax
```

```
10    FORMAT ( //////,T8,A,F10.4)
20    FORMAT (T8,A,F10.4)
      END
```

APPENDIX C. PROGRAM RANPART3.FOR

```
      PROGRAM RANPART3

*     Glossary of variables:
*     Pi          3.14159
*     Xpos        X coordinate of an interior particle
*     Ypos        Y coordinate of an interior particle
*     XposB      X coordinate of a particle along the border
*     YposB      Y coordinate of a particle along the border
*     AP          Area percentage of the particles
*     TPA         Total area occupied by the particles
*     Pspace      Min. spacing between particles
*     TOTarea     Defined area of particles
*     Parea       Area occupied by each particle
*     dp          Distance between interior particles
*     dp2         Distance between interior particles
*     dp2B       Distance between an interior particle and a
*                 border particle
*     dp3         Nearest neighbor spacing
*     dp4         Sum of nearest neighbor spacings
*     dist        Particle diameter plus min spacing
*     dpavg       Average nearest neighbor spacing
*     X           Horizontal dimension of total area
*     Y           Vertical dimension of total area
*     dpmin       Minimum nearest neighbor spacing
*     dp2Bmin     Distance between an interior particle and
*                 its nearest border particle
*     dpmax       Maximum nearest neighbor spacing
*     dpinc       Size of each range between dpmin and dpmax
*                 based on the Sturgis Rule
*     Delta       Minimum value of each range in dpinc
*     Np          Number of particles corrosponding to each
*                 Delta
*     NPA         Number of particles per unit area
*     Pnum        Total number of particles generated
*     R           Generated particle radius from lonormal
*                 distribution
*     Rad(i)      Stored particle radius
*     RadB(i)     Stored border particle radius
*     S           Standard deviation
*     u           Log mean particle radius
*     Radmax      Maximum particle radius
*     Radmin      Minimum particle radius
*     RadBmax     Largest border particle
*
      REAL Pi, Xpos(10000), Ypos(10000), Rad(10000),
     :XposB(300), YposB(300), RadB(300), R, TPA, AP, Pspace,
     :TOTarea, Parea, dp, TRad, dp2, dp2B, dp2Bmin, u, S,
     :dp3(10000), dp4, dist, dpavg, X, Y, dpmin, NPA,
     :dpmax, dpinc, g, Delta(50), Np(50), Pnum, RadBmax,
     :Radmax, Radmin
*
      INTEGER a, b, c, d, e, f, h, i, j, k, l, m, n, p, q, nq
*
      PARAMETER (Pi=3.14159)
*
      DATA n, dp4, e, dpmin, dpmax, Np(1), a, b, Parea, TRad,
     :RadBmax, Radmax, Radmin, nr, nq /
     :1, 0.0, 0, 10.0, 0.001, 0, 1, 1, 0.0, 0.0, .00001,
     :.00001, 1000, 0, 0 /
*
      OPEN (10,file='rp3x.dat',status='NEW')
      OPEN (20,file='rp3y.dat',status='NEW')
      OPEN (30,file='Del3.dat',status='NEW')
      OPEN (40,file='Np3.dat',status='NEW')
      OPEN (50,file='rp3.dat',status='NEW')
      OPEN (60,file='Rad3.dat',status='NEW')
*
      PRINT*,' Enter the area percentage of particles.'
      READ*, AP
      PRINT*,' Enter the length of the X - scale.'
      READ*, X
      PRINT*,' Enter the length of the Y - scale.'
      READ*, Y
      TOTarea = X*Y
      TPA = AP/100*TOTarea
      PRINT*,' Enter the mean particle radius and standard
     :deviation...'
      READ*, u, S
      PRINT*,' Enter the minimum spacing between particles.'
      READ*, Pspace
      PRINT*,' Enter the total number of particles...'
      READ*,Pnum
*
*     Determine random position and radius of each particle.
*
      ISEED = 0
      CALL RNSET (ISEED)
      DO 100 i = 1, Pnum
175   CONTINUE
        q = 0
        CALL RNLNL (1, u, S, R)
150     CONTINUE
        q = q + 1
        IF (q .GE. 600) THEN
        nq = nq + 1
        GOTO 175
        ENDIF
        Xpos(i) = RNUNF()*X
        Ypos(i) = RNUNF()*Y
        Rad(i) = R
*
*         Check for overlap of particles
*
      DO 200 k = 1, i - 1
          dp = SQRT((Xpos(i) - Xpos(k))**2 +
     :                (Ypos(i) - Ypos(k))**2)
          dist = Rad(i) + Rad(k) + Pspace
          IF (dp .LT. dist) GOTO 150
200        CONTINUE
      Parea = Parea + Pi*Rad(i)**2
      IF (Parea .GE. TPA) THEN
          Pnum = i
          GOTO 250
      ENDIF
100   CONTINUE
250   CONTINUE
*
*     Segregate particles along the perimeter from those in
*     the interior of the specified area.
*
      DO 275 d = 1, Pnum
        IF ((Xpos(d) .LE. Rad(d) + Pspace) .OR.
     :         (Xpos(d) .GE. X - Rad(d) - Pspace) .OR.
     :         (Ypos(d) .LE. Rad(d) + Pspace) .OR.
     :         (Ypos(d) .GE. Y - Rad(d) - Pspace)) THEN
          RadBmax = MAX(Rad(d), RadBmax)
        ENDIF
275   CONTINUE
      DO 280 p = 1, Pnum
        IF ((Xpos(p) .LE. RadBmax + Pspace) .OR.
     :       (Xpos(p) .GE. X - RadBmax - Pspace) .OR.
     :       (Ypos(p) .LE. RadBmax + Pspace) .OR.
     :       (Ypos(p) .GE. Y - RadBmax - Pspace)) THEN
          XposB(b) = Xpos(p)
          YposB(b) = Ypos(p)
          RadB(b) = Rad(p)
          b = b + 1
        ELSE
          Xpos(a) = Xpos(p)
          Ypos(a) = Ypos(p)
          Rad(a) = Rad(p)
          TRad = TRad + Rad(a)
          WRITE(10,*) Xpos(a)
          WRITE(20,*) Ypos(a)
          WRITE(60,*) Rad(a)
          a = a + 1
        ENDIF
280   CONTINUE
```

```
*
*     Determine the average distance between nearest
*     neighbors.
*
      DO 300 j = 1, Pnum - (b - 1)
        dp3(n) = SQRT(X**2 + Y**2)
        dp2Bmin = SQRT(X**2 + Y**2)
        DO 350 c = 1, (b - 1)
          dp2B = SQRT((Xpos(j) - XposB(c))**2 +
     :                 (Ypos(j) - YposB(c))**2)
          dp2Bmin = MIN(dp2Bmin, dp2B)
350     CONTINUE
        DO 400 l = 1, Pnum - (b - 1)
          dp2 = SQRT((Xpos(l) - Xpos(j))**2 +
     :                (Ypos(l) - Ypos(j))**2)
          IF (dp2 .NE. 0.0) THEN
           dp3(n) = MIN(dp2, dp3(n), dp2Bmin)
          ENDIF
400       CONTINUE
        n = n + 1
300   CONTINUE
      DO 500 m = 1, Pnum - (b - 1)
        dp4 = dp4 + dp3(m)
500   CONTINUE
      dpavg = dp4/(Pnum - (b - 1))
*
*     Determine the number of particles falling in each
*    range of nearest neighbor spacings (based on the Sturgis
*     rule).
*
      DO 600 f = 1, Pnum - (b - 1)
        dpmin = MIN(dp3(f), dpmin)
        dpmax = MAX(dp3(f), dpmax)
        Radmin = MIN(Rad(f), Radmin)
        Radmax = MAX(Rad(f), Radmax)
600   CONTINUE
      dpinc = (dpmax - dpmin)/(1 + 3.3*LOG10(Pnum - (b - 1)))
      DO 700 g = dpmin, dpmax, dpinc
        e = e + 1
        DO 800 h = 1, Pnum - (b - 1)
          IF ((dp3(h) .GE. g) .AND. (dp3(h) .LT. g + dpinc))
     :    THEN
           Np(e) = Np(e) + 1
          ENDIF
800       CONTINUE
          Delta(e) = (2*g + dpinc)/2
          WRITE(30,*)Delta(e)
          WRITE(40,*)Np(e)/(Pnum - (b - 1))
700   CONTINUE
      NPA = (Pnum - (b - 1))/(X - 2*RadBmax - 2*Pspace)/
     :      (Y - 2*RadBmax - 2*Pspace)
```

```
      WRITE(50,10)’The area percentage of particles is’, AP
      WRITE(50,20)’X dimension is’,X
      WRITE(50,20)’Y dimension is’,Y
      WRITE(50,20)’Min spacing between particles is’,Pspace
      WRITE(50,20)’Mean particle radius is’, TRad/(Pnum - (b
     :- 1))
      WRITE(50,20)’Total area occupied by particles’, Parea
      WRITE(50,20)’Total number of particles is’, Pnum
      WRITE(50,30)’Number of particles along the border is’,b
     :- 1
      WRITE(50,30)’Number of particles in area interior is’,a
     :- 1
      WRITE(50,20)’Number particles per unit area’, NPA
      WRITE(50,20)’Average nearest neighbor spacing is’, dpavg
      WRITE(50,20)’Min nearest neighbor spacing is’,dpmin
      WRITE(50,20)’Max nearest neighbor spacing is’,dpmax
      WRITE(50,20)’Max particle radius is’,Radmax
      WRITE(50,20)’Min particle radius is’,Radmin
      PRINT*,’ The area percentage of particles is’, AP
      PRINT*,’ X dimension is’,X
      PRINT*,’ Y dimension is’,Y
      PRINT*,’ Min spacing between particles is’,Pspace
      PRINT*,’ Mean particle radius is’, TRad/(Pnum - (b - 1))
      PRINT*,’ Total area occupied by particles is’, Parea
      PRINT*,’ Total number of particles is’, Pnum
      PRINT*,’ Number of particles along the border’, b - 1
      PRINT*,’ Number of particles in area interior’, a - 1
      PRINT*,’ Number particles per unit area is’, NPA
      PRINT*,’ Average nearest neighbor spacing is’, dpavg
      PRINT*,’ Min nearest neighbor spacing is’,dpmin
      PRINT*,’ Max nearest neighbor spacing is’,dpmax
      PRINT*,’ Max particle radius is’,Radmax
      PRINT*,’ Min particle radius is’,Radmin
10    FORMAT(T6,A,T50,F10.4)
20    FORMAT(T6,A,T50,F10.4)
30    FORMAT(T6,A,T46,I9)
      END
```

APPENDIX D. PROGRAM RANPART4.FOR

```
      PROGRAM RANPART4

*     Glossary of variables:
*     Pi          3.14159
*     Xpos        X coordinate of an interior particle
*     Ypos        Y coordinate of an interior particle
*     XposB       X coordinate of a particle along the border
*     YposB       Y coordinate of a particle along the border
*     AP          Area percentage of the particles
*     TPA         Total area occupied by the particles
*     Pspace      Min. spacing between particles
*     TOTarea     Defined area of particles
*     Parea       Area occupied by each particle
*     dp          Distance between interior particles
*     dp2         Distance between interior particles
*     dp2B        Distance between an interior particle and a
*                 border particle
*     dp3         Nearest neighbor spacing
*     dp4         Sum of nearest neighbor spacings
*     dist        Particle diameter plus min spacing
*     dpavg       Average nearest neighbor spacing
*     X           Horizontal dimension of total area
*     Y           Vertical dimension of total area
*     dpmin       Minimum nearest neighbor spacing
*     dp2Bmin     Distance between an interior particle and
*                 its nearest border particle
*     dpmax       Maximum nearest neighbor spacing
*     dpinc       Size of each range between dpmin and dpmax
*                 based on the Sturgis Rule
*     Delta       Minimum value of each range in dpinc
*     Np          Number of particles corrosponding to each
*                 Delta
*     NPA         Number of particles per unit area
*     Pnum        Total number of particles generated
*     R           Generated particle radius from lonormal
*                 distribution
*     Rad(i)      Stored particle radius
*     RadB(i)     Stored border particle radius
*     S           Standard deviation
*     u           Log mean particle radius
*     Radmax      Maximum particle radius
*     Radmin      Minimum particle radius
*     RadBmax     Largest border particle
*     AF1-5       Area percentage of particles in each of the
*                 five sub-areas
*     A1-5        Subtotal of particles in each sub-area
```

```
**
      REAL Pi, Xpos(10000), Ypos(10000), Rad(10000),
     :XposB(300), YposB(300), RadB(300), R, TPA, AP, Pspace,
     :u, TOTarea, Parea, dp, TRad, dp2, dp2B, dp2Bmin, S,
     :dp3(10000), dp4, dist, dpavg, X, Y, dpmin, NPA, dpmax,
     :dpinc, g, Delta(50), Np(50), Pnum, RadBmax, Radmax,
     :Radmin, A1, AF1, A2, AF2, A3, AF3, A4, AF4, A5, AF5
*
      INTEGER a, b, c, d, e, f, h, i, j, k, l, m, n, p, q, nq
*
      PARAMETER (Pi=3.14159)
*
      DATA n, dp4, e, dpmin, dpmax, Np(1), a, b, Parea, TRad,
     :RadBmax, Radmax, Radmin, A1, A2, A3, A4, A5, nr, nq /
     :1, 0.0, 0, 10.0, 0.001, 0, 1, 1, 0.0, 0.0, .00001,
     :.00001, 1000, 0.0, 0.0, 0.0, 0.0, 0.0, 0, 0 /
*
      OPEN (10,file=’rp4x.dat’,status=’NEW’)
      OPEN (20,file=’rp4y.dat’,status=’NEW’)
      OPEN (30,file=’Del4.dat’,status=’NEW’)
      OPEN (40,file=’Np4.dat’,status=’NEW’)
      OPEN (50,file=’rp4.dat’,status=’NEW’)
      OPEN (60,file=’Rad4.dat’,status=’NEW’)
*
      PRINT*,’ Enter the area percentage of particles.’
      READ*, AP
      PRINT*,’ Enter 5 sub-area percentages of particles’
      READ*, AF1, AF2, AF3, AF4, AF5
      AF1 = AF1/100
      AF2 = AF2/100
      AF3 = AF3/100
      AF4 = AF4/100
      AF5 = AF5/100
      PRINT*,’ Enter the length of the X - scale.’
      READ*, X
      PRINT*,’ Enter the length of the Y - scale.’
      READ*, Y
      TOTarea = X*Y
      TPA = AP/100*TOTarea
      PRINT*,’ Enter the mean particle radius and standard
     :deviation...’
      READ*, u, S
      PRINT*,’ Enter the minimum spacing between particles.’
      READ*, Pspace
      PRINT*,’ Enter the total number of particles...’
      READ*,Pnum
*
*     Determine random position and radius of each particle.
*
      ISEED = 0
      CALL RNSET (ISEED)
      DO 100 i = 1, Pnum
175   CONTINUE
        q = 0
        CALL RNLNL (1, u, S, R)
150     CONTINUE
        q = q + 1
        IF (q .GE. 600) THEN
        nq = nq + 1
        GOTO 175
        ENDIF
        Xpos(i) = RNUNF()*X
        Ypos(i) = RNUNF()*Y
        Rad(i) = LOG(R)
*
*         Check for overlap of particles
*
        DO 200 k = 1, i - 1
          dp = SQRT((Xpos(i) - Xpos(k))**2 +
     :                (Ypos(i) - Ypos(k))**2)
          dist = Rad(i) + Rad(k) + Pspace
          IF (dp .LT. dist) GOTO 150
200     CONTINUE
      IF ((Ypos(i) .GE. 0.0) .AND. (Ypos(i) .LT. 0.2*Y)) THEN
        IF (A1 .GT. TPA*AF1) GOTO 150
        A1 = A1 + Pi*Rad(i)**2
      ELSEIF ((Ypos(i) .GE. 0.2*Y) .AND. (Ypos(i) .LT. 0.4*Y))
     :THEN
        IF (A2 .GT. TPA*AF2) GOTO 150
        A2 = A2 + Pi*Rad(i)**2
      ELSEIF ((Ypos(i) .GE. 0.4*Y) .AND. (Ypos(i) .LT. 0.6*Y))
     :THEN
        IF (A3 .GT. TPA*AF3) GOTO 150
        A3 = A3 + Pi*Rad(i)**2

      ELSEIF ((Ypos(i) .GE. 0.6*Y) .AND. (Ypos(i) .LT. 0.8*Y))
     :THEN
        IF (A4 .GT. TPA*AF4) GOTO 150
        A4 = A4 + Pi*Rad(i)**2
      ELSEIF ((Ypos(i) .GE. 0.8*Y) .AND. (Ypos(i) .LE. Y))
     :THEN
        IF (A5 .GT. TPA*AF5) GOTO 150
        A5 = A5 + Pi*Rad(i)**2
      ENDIF
        Parea = Parea + Pi*Rad(i)**2
        IF (Parea .GE. TPA) THEN
                Pnum = i
                GOTO 250
        ENDIF

100   CONTINUE
```

```
250   CONTINUE
*
*     Segregate particles along the border from those in
*     the in of the specified area.
*
      DO 275 d = 1, Pnum
        IF ((Xpos(d) .LE. Rad(d) + Pspace) .OR.
     :       (Xpos(d) .GE. X - Rad(d) - Pspace) .OR.
     :       (Ypos(d) .LE. Rad(d) + Pspace) .OR.
     :       (Ypos(d) .GE. Y - Rad(d) - Pspace)) THEN
          RadBmax = MAX(Rad(d), RadBmax)
        ENDIF
275   CONTINUE
      DO 280 p = 1, Pnum
        IF ((Xpos(p) .LE. RadBmax + Pspace) .OR.
     :     (Xpos(p) .GE. X - RadBmax - Pspace) .OR.
     :     (Ypos(p) .LE. RadBmax + Pspace) .OR.
     :     (Ypos(p) .GE. Y - RadBmax - Pspace)) THEN
          XposB(b) = Xpos(p)
          YposB(b) = Ypos(p)
          RadB(b) = Rad(p)
          b = b + 1
        ELSE
          Xpos(a) = Xpos(p)
          Ypos(a) = Ypos(p)
          Rad(a) = Rad(p)
          TRad = TRad + Rad(a)
          WRITE(10,*) Xpos(a)
          WRITE(20,*) Ypos(a)
          WRITE(60,*) Rad(a)
          a = a + 1
        ENDIF
280   CONTINUE
*
*     Determine the average distance between nearest

      DO 300 j = 1, Pnum - (b - 1)
        dp3(n) = SQRT(X**2 + Y**2)
        dp2Bmin = SQRT(X**2 + Y**2)
        DO 350 c = 1, (b - 1)
          dp2B = SQRT((Xpos(j) - XposB(c))**2 +
     :                    (Ypos(j) - YposB(c))**2)
          dp2Bmin = MIN(dp2Bmin, dp2B)
350       CONTINUE
        DO 400 l = 1, Pnum - (b - 1)
          dp2 = SQRT((Xpos(l) - Xpos(j))**2 +
     :                   (Ypos(l) - Ypos(j))**2)
          IF (dp2 .NE. 0.0) THEN
           dp3(n) = MIN(dp2, dp3(n), dp2Bmin)
          ENDIF
400       CONTINUE
```

```
        n = n + 1
300   CONTINUE
      DO 500 m = 1, Pnum - (b - 1)
        dp4 = dp4 + dp3(m)
500   CONTINUE
      dpavg = dp4/(Pnum - (b - 1))
*
*    Determine the number of particles falling in each range
*     of nearest neighbor spacings (based on the Sturgis
*     rule).
*
      DO 600 f = 1, Pnum - (b - 1)
        dpmin = MIN(dp3(f), dpmin)
        dpmax = MAX(dp3(f), dpmax)
        Radmin = MIN(Rad(f), Radmin)
        Radmax = MAX(Rad(f), Radmax)
600   CONTINUE
      dpinc = (dpmax - dpmin)/(1 + 3.3*LOG10(Pnum - (b - 1)))
      DO 700 g = dpmin, dpmax, dpinc
        e = e + 1
        DO 800 h = 1, Pnum - (b - 1)
          IF ((dp3(h) .GE. g) .AND. (dp3(h) .LT. g + dpinc))
     :    THEN
            Np(e) = Np(e) + 1
          ENDIF
800     CONTINUE
        Delta(e) = (2*g + dpinc)/2
        WRITE(30,*)Delta(e)
        WRITE(40,*)Np(e)/(Pnum - (b - 1))
700   CONTINUE
      NPA = (Pnum - (b - 1))/(X - 2*RadBmax - 2*Pspace)/
     :       (Y - 2*RadBmax - 2*Pspace)
      WRITE(50,10)'The area percentage of particles is', AP
      WRITE(50,50) 'The sub-area percentages of particles
     :are:'
      WRITE(50,40) AF1,AF2,AF3,AF4,AF5
      WRITE(50,20)'X dimension is',X
      WRITE(50,20)'Y dimension is',Y
      WRITE(50,20)'Min spacing between particles is',Pspace
      WRITE(50,20)'Mean particle radius is', TRad/(Pnum - (b
     :- 1))
      WRITE(50,20)'Total area occupied by particles', Parea
      WRITE(50,20)'Total number of particles is', Pnum
      WRITE(50,30)'Number of particles along the border is',b
     :- 1
      WRITE(50,30)'Number of particles in area interior is',a
     :- 1
      WRITE(50,20)'Number particles per unit area', NPA
      WRITE(50,20)'Average nearest neighbor spacing is', dpavg
      WRITE(50,20)'Min nearest neighbor spacing is',dpmin
      WRITE(50,20)'Max nearest neighbor spacing is',dpmax
      WRITE(50,20)’Max particle radius is’,Radmax
      WRITE(50,20)’Min particle radius is’,Radmin
      PRINT*,’ The area percentage of particles is’, AP
      PRINT’(T6,A,T45,5F4.1)’,’The sub-area percentages of
     :particles are’,AF1,AF2,AF3,AF4,AF5
      PRINT’(T6,A,T45,5F6.2)’,’The area occupied by particles
     :in each sub-area is...’,A1, A2, A3, A4, A5
      PRINT*,’ X dimension is’,X
      PRINT*,’ Y dimension is’,Y
      PRINT*,’ Min spacing between particles is’,Pspace
      PRINT*,’ Mean particle radius is’, TRad/(Pnum - (b - 1))
      PRINT*,’ Total area occupied by particles is’, Parea
      PRINT*,’ Total number of particles is’, Pnum
      PRINT*,’ Number of particles along the border’, b - 1
      PRINT*,’ Number of particles in area interior’, a - 1
      PRINT*,’ Number particles per unit area is’, NPA
      PRINT*,’ Average nearest neighbor spacing is’, dpavg
      PRINT*,’ Min nearest neighbor spacing is’,dpmin
      PRINT*,’ Max nearest neighbor spacing is’,dpmax
      PRINT*,’ Max particle radius is’,Radmax
      PRINT*,’ Min particle radius is’,Radmin
10    FORMAT(T6,A,T50,F10.4)
20    FORMAT(T6,A,T50,F10.4)
30    FORMAT(T6,A,T45,I10)
40    FORMAT(5F6.2)
50    FORMAT(A)
      END
```

LIST OF REFERENCES

1. Ashby, M. F., "*On the Orowan Stress*", Physics of Strength and Plasticity, pp. 117 - 126, M.I.T. Press, 1969.

2. Köster, U., "*Recrystallization Involving a Second Phase*", Metals Science Journal, Vol. 8, pp. 151, 1974.

3. Macri, P. D.,"*Processing, Microstructure and Elevated Temperature Mechanical Properties of a 6061 Aluminum-Alumina Metal Matrix Composite*", Master's Thesis, Naval Postgraduate School, Monterey, California, December 1990.

4. Kocks, U. F., "*A Statistical Theory of Alloy Hardening*", Physics of strength and Plasticity, pp. 143, M.I.T. Press, 1969.

5. Pas, M., "*Computer Simulation of Random and Non-Random Second Phase Distributions in Two Phase Materials*", Master's Thesis, Naval Postgraduate School, Monterey, California, December 1990.

6. Underwood, E. E., *Quantitative Stereology*, pp. 84 - 85, Addison-Wesley Publishing Company Inc., 1970.

INITIAL DISTRIBUTION LIST

No.Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, VA 22304-6145

2. Library, Code 52 2
Naval Postgraduate School
Monterey, CA 93943-5000

3. Department Chairman, Code ME/Hy 1
Mechanical Engineering Department
Naval Postgraduate School
Monterey, CA 93943-5000

4. Professor T. McNelley, Code ME/MC 5
Mechanical Engineering Department
Naval Postgraduate School
Monterey, CA 93943-5000

5. LT Mark S. Manfredi 2
603 Dupont St.
Portsmouth, RI 02871