To the Graduate Council:

I am submitting herewith a thesis written by Yuehui Fan entitled "Monte Carlo Simulation of Soil Particle Packing." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Agricultural Engineering.

Fred D. Tompkins, Major Professor

We have read this thesis
and recommend its acceptance:

Eric C. Drumm

Philip W. Schaefer

Accepted for the Council:

C W Minkel

Vice Provost
and Dean of The Graduate School

MONTE CARLO SIMULATION OF SOIL PARTICLE PACKING

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Yuehui Fan

August 1991

"If a man will begin with certainties, he will end with doubts; but if he will be content with doubts, he shall end in certainties."

Francis Bacon

"... no physical theory begins with formulae but rather with ideas and notions ..."

Einstein and Infeld

"... deeper physical insight combined with theoretical simplicity provides the short cuts leading immediately to the core of extremely complex problems ..."

Biot

To:

My Wife and My Mother,

for their devotion and support.

ACKNOWLEDGMENTS

The author is deeply grateful to the many people who directly or indirectly influenced this thesis and his education. Among these people, first of all, the author wishes to express sincere gratitude to his major professor and thesis supervisor, Dr. Fred D. Tompkins, who introduced the author to the interesting and challenging research area. The author wishes also to thank him for providing support from his research grant to present the results of some chapters of this thesis at the 1990 International Winter Meeting of ASAE, Hyatt Regency Chicago, Chicago, Illinois. His guidance and support were extremely helpful throughout the research project and graduate program.

The author thanks Dr. Houston Luttrell, head of the Agricultural Engineering Department, who generously made departmental resources available for the project.

Thanks are also extended to Dr. R. D. von Bernuth, a former committee member from the Agricultural Engineering Department at the University of Tennessee, Knoxville, who is now head of the Agricultural Engineering Department at Michigan State University, East Lansing. His course, "Computer Simulation of Agricultural Systems," stimulated many new ideas which later on were utilized in this manuscript.

The author would like to thank Dr. Eric C. Drumm, committee member from the Institute for Geotechnology, Department of Civil Engineering, for his unselfish help and endless support and encouragement throughout the graduate education. He provided invaluable insight for solving the soil related problems.

Thanks are also due to Dr. Philip W. Schaefer, committee member from the Mathematics Department. His support makes this thesis mathematically and theoretically stronger.

A special note of thanks is expressed to Mr. Paul W. Elliott and Mr. Craig Wagoner and their assistants in the Agricultural Engineering Research Shop for their outstanding work in constructing the experimental apparatus.

The author would also like to thank Dr. Bobby L. Bledsoe for his support and encouragement throughout the graduate studies. To all other members of the Agricultural Engineering Department who continuously supported the author throughout the writing of this thesis, the author wishes to express thanks.

Dr. Hamparsum Bozdogan, colleague and friend in the Statistics Department, was helpful in the selection of the computer simulation language.

Dr. Amir S. Mobasheran and Ms. Sue Smith, computer consultants and friends at the University of Tennessee Computing Center (UTCC), were most kind in spending many hours assisting in solving computer related problems. The author acknowledges the use of the University of Tennessee Computing Center, Knoxville, Tennessee, in obtaining the results shown here.

The author wishes to thank his wife, Xia Liu, for her patience, understanding and support during the writing of this thesis.

Last, but not least, the author wishes to express thanks and gratitude to the Agricultural Engineering Department at the University of Tennessee, Knoxville for continued financial support as a Research Assistant during his graduate studies, without which it would have been difficult to obtain the M.S. degree.

ABSTRACT

Monte Carlo simulation of soil particle packing, a fundamental problem encountered in modeling a soil system, and some immediate and potential applications of the model were examined. In the simulation, soil particles were dealt with as rigid spheres (or cylindric rods) with no overlap between them. The simulation is conceptually bifurcated: (i) simulation of particle size distribution, and (ii) simulation of particle location.

For the particle size simulation, a powerful Acceptance-Rejection (AR) method was adopted to deal with the continuous and irregular particle size distribution characteristic of soil. To enable the AR method to be used with a particle size distribution defined by <u>weight</u>, the relationship between such a distribution and one defined by <u>number</u> was established. A <u>cumulative</u> size distribution was simulated by conversion, based on numerical differentiation, to the associated <u>frequency density</u> size distribution. Thus, a <u>cumulative</u> size distribution defined by 'percent finer by weight vs. particle size,' a distribution commonly used in agricultural and civil engineering practice, can be approached in such a particle packing simulation.

For the simulation of particle location, an 'inserting particle' technique was proposed. The particles were inserted, one at a time, to a random location, forcing particles which were already resident to sequentially move to eliminate overlap between adjacent particles and between the container boundary and

individual particles. The process for generating and placing particles was continued until the overall porosity calculated for the simulated particle pack had reached the value specified by the user.

Since the simulation is founded entirely on realistic soil parameters (particle size distribution, overall porosity, etc.) and on the very nature of soil (discreteness, randomness, etc.), a wide range of applications is possible. A few of these applications, are presented herein. Based upon this model, the displacement field of soil particles under load was predicted, and predicted results agreed reasonably well with physical observations made in controlled laboratory experiments. Densest packing of soil particles can be approached, and the smallest porosity associated with this densest packing can be computed. Even very difficult problems, such as dynamic response of soil to a penetrating object, can be addressed using this modeling technique. The simulation will likely prove useful for research and engineering practice as it is applied to soil systems and other granular systems.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1. BACKGROUND AND STATEMENT OF THE PROBLEM

As an engineering material, soil is unquestionably very complex. Its granular construction along with other complicated properties makes problems related to prediction of soil response to loading extremely difficult. Considering only a few of the significant soil properties, things like non-homogeneity, non-isotropy, and time, moisture, and stress history dependence, can make various conventional approaches unwieldy or impractical. Because of the large deflections which typically occur and the more complicated surface layer involved, behavior of agricultural soils is even more difficult to predict than is soil behavior in most civil engineering applications (Raper and Erbach, 1990a).

The direct physical experiment method, either using samples from *in situ* soil or using artificial granular materials, addresses many practical problems occurring in soil engineering or illustrates certain behaviors of soils. However, the method faces not only the cost problem (which has been intensively pointed out by other authors) but also a problem of inability to isolate the variables. This last factor can prevent complete understanding of the mechanisms of various soil behaviors.

Powerful similitude principles play a major role in the development of aerodynamics and hydrodynamics, but have been less used in soil systems. A technical factor is the broad variability of soils as compared to air (Freitag et al., 1970). Where air has relatively well-defined and predicable physical characteristics, the characteristics of soil are not well identified, may not be measurable, and certainly are not adequately predictable.

Only highly simplified and idealized soil mechanics problems can be approached with analytical methods, such as the well known Boussinesq's solution (Jumikis, 1962, 1969; Wu, 1966; Lambe and Whitman, 1979; Koolen and Kuipers, 1983; Das, 1983).

The finite element method (FEM), on the other hand, provides a powerful tool which can be used with very complicated problems (see, e.g., Pollock et al., 1986; Eldin et al., 1990a, 1990b). However, the method has three substantial limitations as follows:

> First, the method requires a constitutive law which may be difficult to define for a very complicated material such as soil. In fact, a constitutive relationship which fully accounts for all intricacies of an agricultural soil has yet to be developed (Raper and Erbach, 1990b).
>
> Secondly, the empirical-type constitutive laws, widely used in FEM and other numerical techniques applied to soil problems, only provide an input-output "black box". Thus, the fundamental mechanisms of soil behavior are not clearly demonstrated.

Thirdly, the most critical limitation of the FEM lies in the fact that it cannot easily reflect the stochastic nature of a soil system. The ordinary finite element method is applicable only to a determined system; but soil, by its very nature, is a stochastic system. Soil property inputs to the FEM system are characterized by large variations. As a result, the outputs, almost always, are highly variable. Meaningful solutions, from a statistical viewpoint, should include the mean and the variance of the outputs. However, most FEM cannot produce such results.

To capture the randomness or discreteness properties of soil or other granular materials, many alternative approaches have been proposed (Bourdeau and Harr, 1989; Chang et al., 1989; Chang and Misra, 1989, 1991; Chang and Liao, 1990; Cundall and Strack, 1979a, 1979b; Dobry et al., 1991; Hardin and Blandford, 1989; Jong, 1971; Matsuoka, 1974; Nemat-Nasser, 1980; Rothenburg and Bathurst, 1989; Sadd et al., 1989, 1991; Satake, 1985; Strack and Cundall, 1978; Tobita, 1989; Ueng and Lee, 1990). For readers interested in a broader examination of various alternatives for granular materials in general, the author recommends Shahinpoor (1983) and Jenkins and Satake (1983). These alternatives have proved to be useful in the granular systems such as soil, but non reflects totally and simultaneously the randomness and discreteness properties of soil.

Monte Carlo simulation of a soil system, as another alternative, may allow better prediction of certain soil phenomena, resulting in a clearer understanding of the fundamentals of soil behavior, because the simulation can rather easily capture the discreteness and randomness properties simultaneously. To apply this method, the first and most fundamental step is simulating the packing of discrete soil particles to form the soil mass.

2. OBJECTIVES

The goal of the Monte Carlo simulation effort related to soil particle packing was to develop a new research tool applicable to problems in soils or other granular materials. Thus, a technique was sought which could simultaneously simulate a soil mass:

(1) characterized by any arbitrarily-chosen soil particle size distribution or cumulative size distribution,

(2) with any desired porosity including the smallest porosity (densest packing), and

(3) with a random location for each discrete particle.

CHAPTER II

REVIEW OF LITERATURE

While little work involving Monte Carlo simulation of soil particle packing is found in the literature, simulation efforts aimed at predicting the densest random packing of various granular materials have been reported (Tory et al., 1968, 1973; Visscher and Bolsterli, 1972; Gotoh et al., 1978; Jodrey and Tory, 1979, 1980, 1981; Powell, 1980; Vinogradov and Wierzba, 1986; Vinogradov, 1987; Vinogradov and Springer, 1990). Random packing of hard spheres (RPHS) has been extensively studied. These hard spheres are taken as models for particulate systems in a wide variety of fields (a rather extensive list from the most recent to the oldest ones: Cipra, 1991, 1990; Keen and McGreevy, 1990; Jullien and Meakin, 1990; Finney, 1977; Barker and Hoare, 1975; Gotoh and Finney, 1974; The Molecular Physics Correspondent of Nature, 1972; Beresford, 1969; Mason, 1968; Bernal and Finney, 1967; Susskind and Becker, 1966; Yerazunis et al., 1965; Scott and Mader, 1964; Scott, 1962, 1960; Rutgers, 1962; Klement et al., 1960; Mangelsdorf and Washington, 1960; Bernal and Mason, 1960; Bernal, 1960, 1959; Furukawa, 1959). Powell (1980) noted that equal-sized spheres are excellent models of the liquid and glassy states.

Most computer simulations of RPHS have been based on the Bennett-Matheson technique (for details, see Bennett, 1972; Adams and Matheson, 1972;

Matheson, 1974), wherein sphere packings are built sequentially by adding new spheres, one at a time, in contact with three other spheres already in place. The choice of sphere placement site can be varied. Bennett (1972) placed the new sphere at the site nearest to a fixed center, thereby generating a spherical structure with a decreasing density. Matheson (1974) produced homogeneous density by choosing the site nearest to a plane surface; while Tory et al. (1968, 1973), Jodrey and Tory (1980, 1981) and Visscher and Bolsterli (1972) chose the site by dropping spheres from random coordinate positions above an existing pile of spheres and allowing the dropped sphere to roll over the pile until it attained stable three-point contact.

All of the above schemes provide reasonable approaches to very dense random packing. However, each scheme has significant limitations in situations involving an arbitrarily chosen porosity. Therefore, none is wholly adequate for simulating an agricultural soil system because particle sizes vary within the soil mass and porosity may take on a range of values. Thus, a new, more appropriate scheme for particle specification and placement is required.

CHAPTER III

ASSUMPTIONS

Three assumptions were made to simplify the soil mass simulation. They are:

1. Rigid particle assumption

Individual soil particles are considered to be rigid bodies. Thus, soil deflection results from relative movement among soil particles within the soil mass. Soil particles themselves are assumed not to change size or shape (Lambe and Whitman, 1979).

2. No overlap assumption

No overlap exists between any two particles; that is, portions of no two particles can occupy the same space.

3. Spherical particle shape simplification

All soil particles are assumed to be sphere-shaped in the 3-dimensional case or circle-shaped in the 2-dimensional case. This assumption dramatically simplifies geometric considerations for describing the shapes and locations of individual soil particles and at least is appropriate for soil classified as sands or silts.

CHAPTER IV

METHOD AND PROCEDURE

Using the assumptions and simplifications noted above, the Monte Carlo simulation of soil particle packing can be achieved by following five major steps:

(1) generate particles of random sizes within a given size distribution or cumulative size distribution,

(2) place the particles in random locations,

(3) apply the no overlap rule to insure that portions of no two particles are assigned to the same space,

(4) confine all particles to the container boundaries, and

(5) determine whether the process for generating and placing particles needs to be continued based upon a comparison between calculated and desired porosity values.

Details for each step are described below:

Step 1. Use acceptance - rejection (A-R) methods to generate soil particles of random size within a given size distribution or cumulative size distribution

Generally speaking, there are three major methods used to generate random numbers for a given size distribution or cumulative size distribution (Johnson, 1987). The first is the inverse cumulative distribution function method. The second is the transformation method. The third includes the acceptance-rejection (A-R) methods. The first two can only be used with relatively simple probability distributions. However, the third method can be used with an arbitrarily-chosen size distribution or arbitrary cumulative size distribution.

The actual acceptance-rejection methods used in the soil packing simulation can be divided into four cases depending upon how the particle size distribution is specified:

Case 1: A specified particle size distribution defined by <u>number</u>

Suppose there is a given particle size distribution as illustrated in Figure 1. In simulating the distribution, a location associated with a particle size is randomly selected within the rectangular area ABCD. If the randomly-selected location lies on or below the curve defining the particle size distribution, the radius associated with the selection is accepted. Otherwise, the radius is rejected and another particle size is randomly selected. The process is detailed in the following:

First, a random number, R_{and1}, is generated from a uniform probability distribution ($0 \le R_{and1} \le 1$). That random number is linearly mapped to a random radius r within the interval [r_{min} , r_{max}], by using the formula:

$$r = r_{min} + (r_{max} - r_{min}) R_{and1} \qquad (r_{min} \le r \le r_{max}) \qquad (1)$$

where:

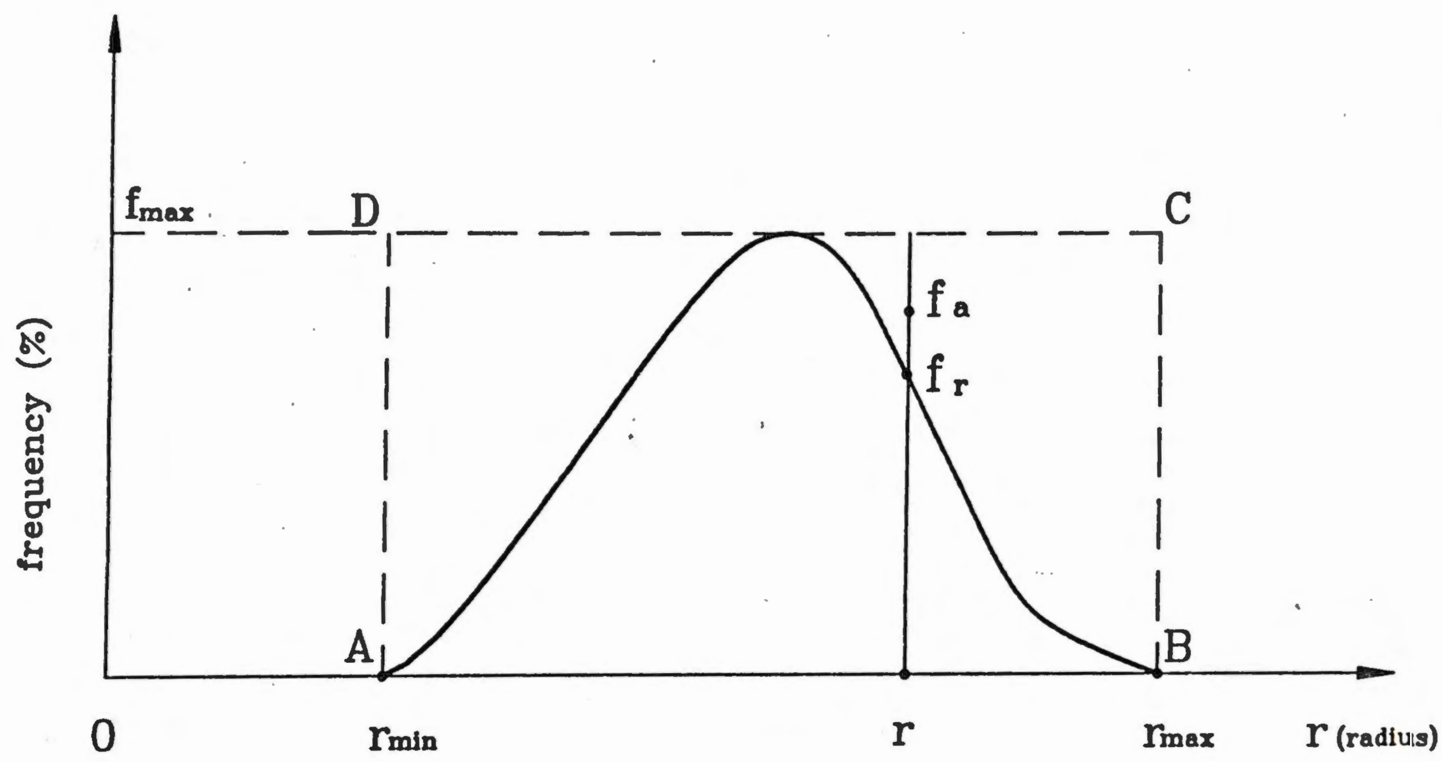

Figure 1. Generation of particles of random radius from a given size distribution using Acceptance-Rejection technique.

r --- random radius of a particle,

r_{min} --- minimum radius of particles in the given size distribution, and

r_{max} --- maximum radius of particles in the given size distribution.

Then, f_r , the size distribution frequency associated with the r, can be easily found from the given particle size distribution (Figure 1).

Secondly, R_{and2} , another independent random number, is generated from a uniform probability distribution ($0 \leq R_{and2} \leq 1$). This random number is linearly mapped to f_a which lies in the interval $[0, f_{max}]$, by using the equation:

$$f_a = f_{max} R_{and2} \qquad (0 \leq f_a \leq f_{max}) \tag{2}$$

where:

f_a --- an artificial frequency associated with R_{and2} , and

f_{max} --- maximum frequency occurring within the given size distribution.

When f_a is compared to f_r , a particle of radius r is:

(i) rejected if $f_a > f_r$, or

(ii) accepted if $f_a \leq f_r$.

If (i) is true and the particle is rejected, then Step 1 is repeated. If (ii) is true and the particle is accepted, then processing moves to Step 2, which is described below. Note that Case 1 is the basic case used in the simulation and that the other three cases described below eventually converge to Case 1.

Case 2: A given cumulative size distribution defined by 'percent finer by number'

This case can be easily translated into Case 1 by converting the given cumulative size distribution to a frequency distribution of sizes, based on numerical differentiation. The problem can then be approached as described in Case 1.

Case 3: A specified particle size distribution defined by weight

This case can be easily translated into Case 1 by converting the given size distribution by weight to a size distribution by number, based on the relationship:

$$f_n(r) = \left(\frac{\bar{r}}{r}\right)^2 f_w(r) \qquad \textit{(for 2-dimensional case)} \qquad (3a)$$

or,

$$f_n(r) = \left(\frac{\bar{r}}{r}\right)^3 f_w(r) \qquad \textit{(for 3-dimensional case)} \qquad (3b)$$

where:

r --- particle radius,

$f_n(r)$ --- the value of probability density function defined by particle number at particle radius r,

$f_w(r)$ --- the value of probability density function defined by particle weight at particle radius r, and

$\bar{r}$ --- overall mean radius of soil particles.

The problem can then be approached as described in Case 1. The reasoning, in detail, for the above equation is the following. By definition, $f_n(r)$ is the limit as $\Delta r \to 0$:

$$f_n(r) = \lim_{\Delta r \to 0} \frac{\Delta N/N}{\Delta r} \tag{4}$$

where:

Δr --- small interval of r,

ΔN --- number of particles within the size interval Δr, and

N --- the total number of particles.

Further, $f_w(r)$ is the limit as $\Delta r \to 0$:

$$f_w(r) = \lim_{\Delta r \to 0} \frac{\Delta W/W}{\Delta r} \tag{5}$$

where:

ΔW --- weight of particles within the size interval Δr, and

W --- the total weight of particles.

For any point r where $f_n(r) \neq 0$, combining equations (4) and (5) yields:

$$\frac{f_w(r)}{f_n(r)} = \frac{\lim\limits_{\Delta r \to 0} \dfrac{\Delta W/W}{\Delta r}}{\lim\limits_{\Delta r \to 0} \dfrac{\Delta N/N}{\Delta r}}$$

$$= \lim_{\Delta r \to 0} \frac{\dfrac{\Delta W / W}{\Delta r}}{\dfrac{\Delta N / N}{\Delta r}}$$

$$= \lim_{\Delta r \to 0} \frac{\Delta W / W}{\Delta N / N}$$

$$= \lim_{\Delta r \to 0} \left(\frac{\Delta W}{\Delta N}\frac{N}{W}\right)$$

$$= \frac{N}{W} \lim_{\Delta r \to 0} \frac{\Delta W}{\Delta N}$$

$$= \frac{1}{\bar{w}} \lim_{\Delta r \to 0} \frac{\Delta W}{\Delta N}$$

$$= \frac{w}{\bar{w}} \qquad (6)$$

where:

$\bar{w}$ --- overall mean weight of particles $\bar{w}=W/N$, and

w --- weight of a particle with a radius r, w = $\lim(\Delta W/\Delta N)$.

In the 2-dimensional case, $w=\pi r^2 l\gamma$ and $\bar{w}=\pi \bar{r}^2 l\gamma$, where l and γ are length and unit weight of soil particles, respectively. Therefore:

$$\frac{f_w(r)}{f_n(r)} = \frac{\pi r^2 l\gamma}{\pi \bar{r}^2 l\gamma}$$

$$= \left(\frac{r}{\bar{r}}\right)^2 \qquad (7a)$$

In the 3-dimensional case, $w=(4/3)\pi r^3\gamma$ and $\bar{w}=(4/3)\pi \bar{r}^3\gamma$, where γ is the unit weight of soil particles. Therefore:

$$\frac{f_w(r)}{f_n(r)} = \frac{\frac{4}{3}\pi r^3 \gamma}{\frac{4}{3}\pi \bar{r}^3 \gamma}$$

$$= \left(\frac{r}{\bar{r}}\right)^3 \tag{7b}$$

Manipulating certain terms on both sides of equations (7a) and (7b) immediately yields equations (3a) and (3b) respectively. For the points r where $f_n(r) = 0$, since $f_w(r) = 0$ either, equations (3a) and (3b) are also hold.

In fact, owing to the nature of the A-R method detailed in Case 1, if the probability density function is multiplied by any positive constant, the result of generated random size of particles will not change. In other words, what matters in the simulation is the <u>shape</u> of the distribution curve by <u>number</u>, not its actual values. Therefore, in the simulation, $\bar{r}$ can be simply set to unity (or any other convenient positive value), regardless what the true value of $\bar{r}$ is.

Case 4: A specified cumulative size distribution defined by 'percent finer by <u>weight</u>'

Due to technical difficulties related to counting the actual number of soil particles, most soil particle-size distribution curves are defined by 'percent finer by <u>weight</u> vs. effective diameter (or radius)'. This instance can be easily translated into Case 3 by converting the given cumulative size distribution by weight to a frequency density size distribution by weight, based upon numerical differentiation. The problem can then be approached as described in Case 3.

Step 2. Generate random locations for individual particles

The random locations for particles can be generated by using a combination of the uniform probability random generator and the linear mapping technique. For simplicity, a 2-dimensional description will be given. However, the scheme can be easily extended to the 3-dimensional case.

Suppose the control volume is a rectangular container with width, W_{idth}, and depth, D_{epth}, as illustrated in Figure 2. A random location for a particle within that container can be generated as follows:

(i) Generating horizontal locations for particles:

To determine a horizontal location, a random number, $R_{horizontal}$, is first generated from a uniform probability distribution ($0 \leq R_{horizontal} \leq 1$). The $R_{horizontal}$ is then linearly mapped to X which lies in the interval $[-W_{idth}/2 + r, W_{idth}/2 - r]$; that is,

$$\mathbf{X = (R_{horizontal} - 1/2)(W_{idth} - 2r) \qquad (-W_{idth}/2 + r \leq X \leq W_{idth}/2 - r)} \quad (8)$$

where:

X --- horizontal position of a particle, and

r --- the random radius generated in Step 1.

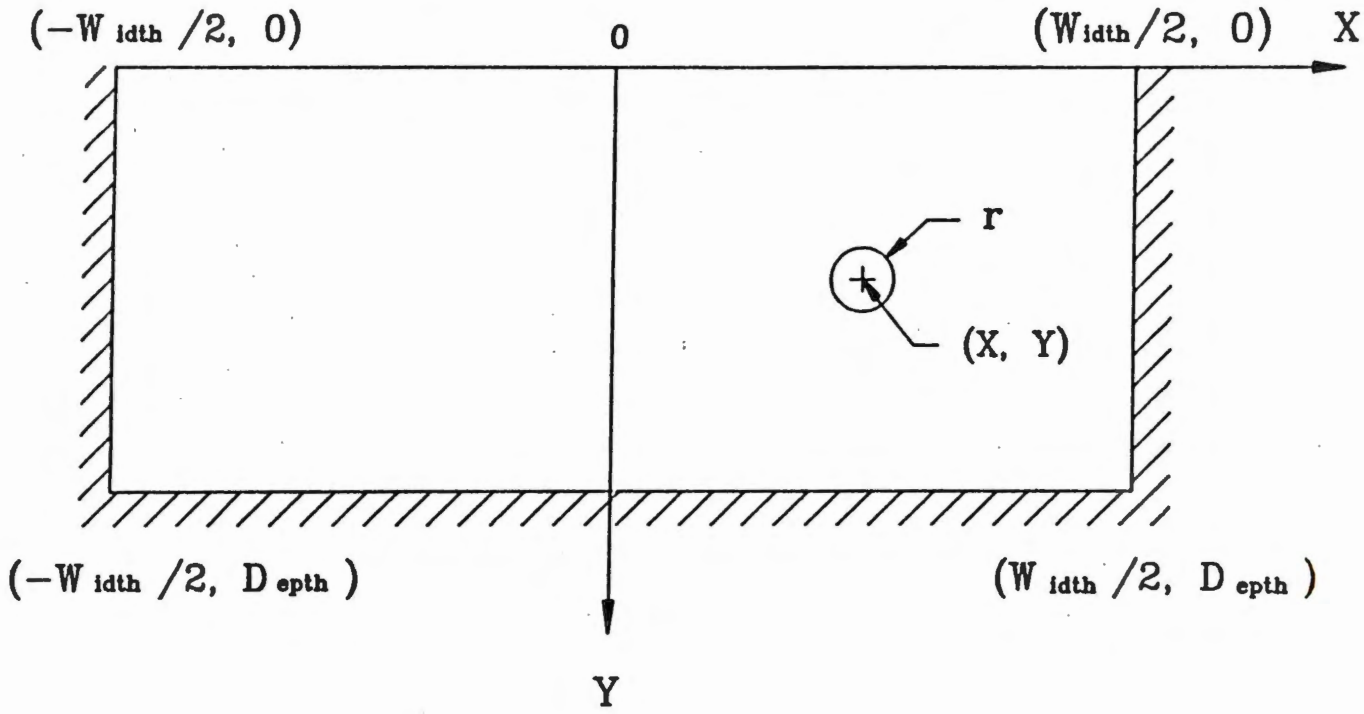

Figure 2. Generation of random locations for soil particles within the control volume.

(ii) Generating vertical locations for particles:

Another independent random number, $R_{vertical}$, is similarly generated from a uniform probability distribution ($0 \leq R_{vertical} \leq 1$). The $R_{vertical}$ is then linearly mapped to Y which lies in the interval [r, D_{epth} - r], that is:

$$\mathbf{Y = (D_{epth} - 2r) R_{vertical} + r \qquad (r \leq Y \leq D_{epth} - r)} \qquad (9)$$

where:

Y --- vertical location of particle.

Step 3. Use the no overlap rule to replace particles if necessary

If a particle location generated in Step 2 causes overlap with one or more particles already in place, the pre-existing particles are moved to provide space for the new particle. Particle movement is along a line connecting particle centroids to both the new particle and the previously placed particle (see Figure 3). This movement is continued, until no overlap exists. If this movement results in the relocated particles overlapping with some other particles, the replacement process is continued until no overlap exists for any particle. This process of sequentially moving particles to eliminate overlap, however, may eventually make portions of some particles lie outside the container boundary. To negate this, Step 4 is implemented as described below.

Step 4. Confine particles within the container boundaries

Movement of particles already in the container to accommodate a new particle placed at a random location, as described in Step 3, may result in some

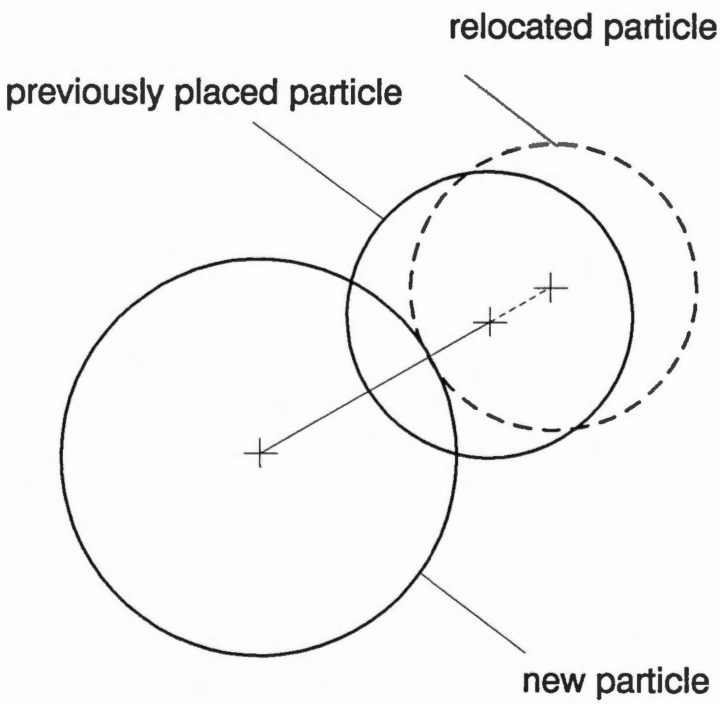

Figure 3. Elimination of overlapping of particles.

particles being pushed outside the container boundaries. If that should occur for a given particle, then the particle is moved along a line normal to the container boundary at the point of contact, until the particle lies completely inside the container (see Figure 4). If movement of this particle results in overlaps with adjacent particles, then Step 3 is again implemented. The iteration continues until all particles are contained within the boundaries of the container, and no particles are overlapping.

Step 5. Calculate porosity and compare with value required

By summing the volumes of the individual generated particles, the porosity of a soil mass (defined by a ratio between volume of voids and volume of container) can be calculated using the formula:

$$n = \frac{\textit{Volume of container} - \sum \textit{Volumes of the individual particles}}{\textit{Volume of container}} \qquad (10)$$

where:

n --- soil porosity.

If:

$$\mathbf{n > n_{given}} \qquad (11)$$

where:

n_{given} --- given or required soil porosity,

then the algorithm is repeated beginning at Step 1 to generate another particle.

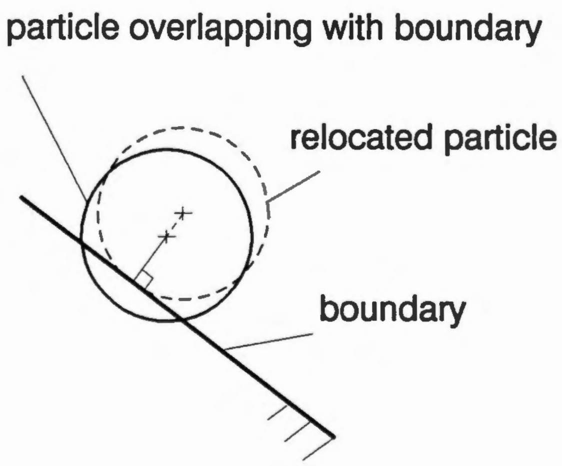

Figure 4. Elimination of overlapping between particle and boundary of the control volume.

If:

$$\mathbf{n} \leq \mathbf{n}_{\mathbf{given}} \tag{12}$$

then the soil mass generation procedure is complete and particle addition is halted.

The computer language used in the simulation development was MATLAB (The Math Works, INC., 1989). The version 3.5 (current version) of PRO-MATLAB is written entirely in the C language, providing a completely integrated system, including graphics, programmable macros, IEEE arithmetic, a fast interpreter, and many analytical commands. The computer programs for the simulation can be found in the Appendices of this thesis.

To show the versatility and efficiency of the methodology, two examples are demonstrated below. The first example of simulated "soil" particle packing is illustrated in Figure 5. This "soil" mass has a porosity of $n=0.25$ and a uniform probability particle size distribution (by <u>number</u>) with $r_{min} = 1$ and $r_{max} = 3$.

The second shows a more sophisticated but realistic example. This example directly utilizes a real cumulative soil particle size distribution defined by <u>weight</u> (Figure 6). The essence of the distribution was adapted from Lambe (1951), but the particles with radii smaller than 10^{-2}mm (about 5 percent finer by weight) were ignored to save computer time and facilitate observation. Assuming the porosity $n=0.5$, a simulated 2-dimensional soil particle packing is illustrated in Figure 7.

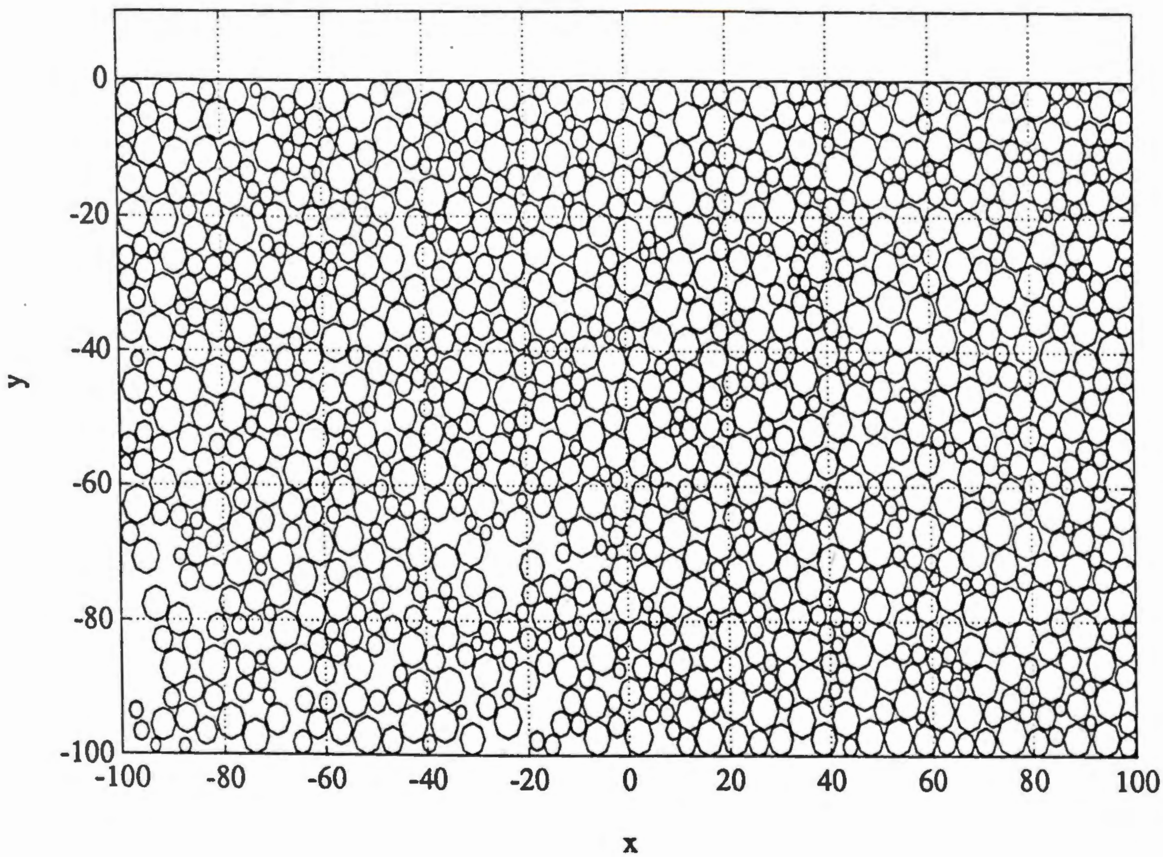

Figure 5. Simulated "soil" particle packing with a uniform probability particle size distribution by number (porosity =0.25, r_{min} = 1, r_{max} = 3).

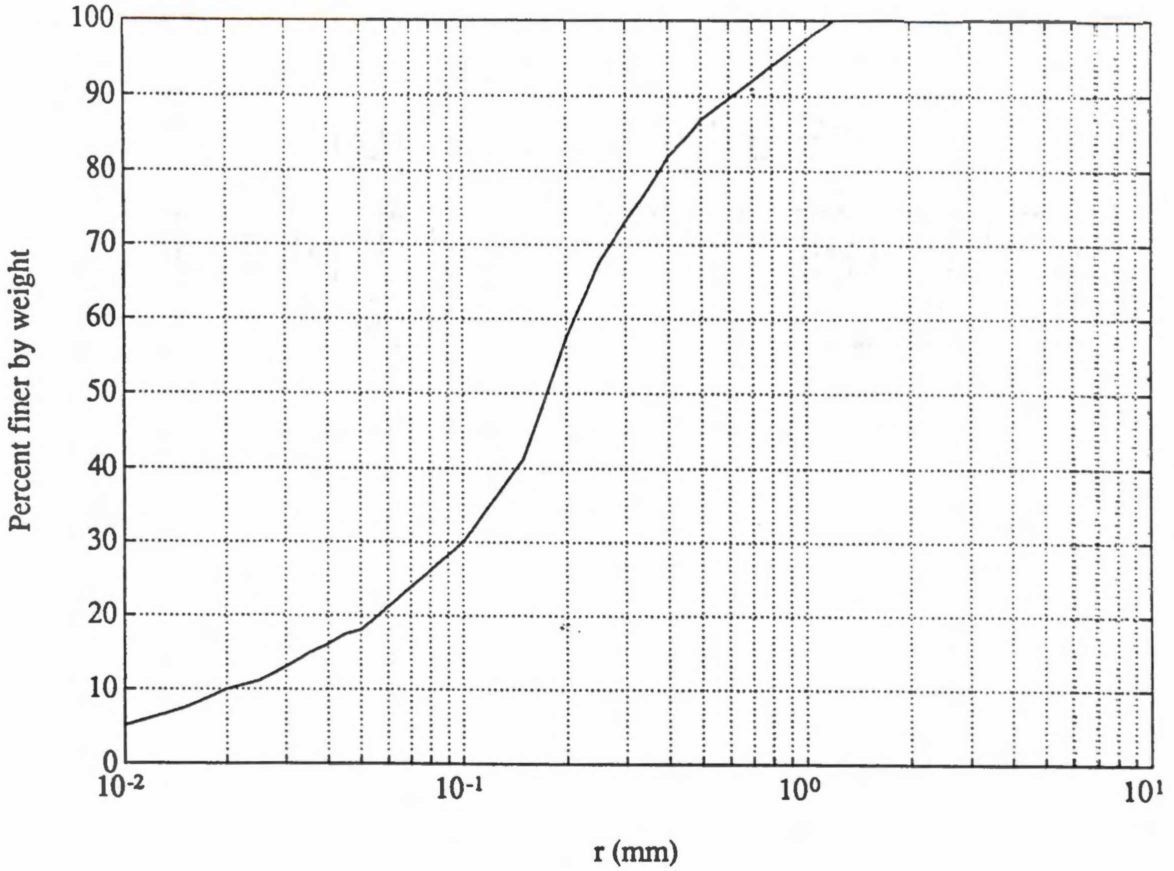

Figure 6. Cumulative soil particle size distribution defined by weight (adapted from Lambe, 1951).

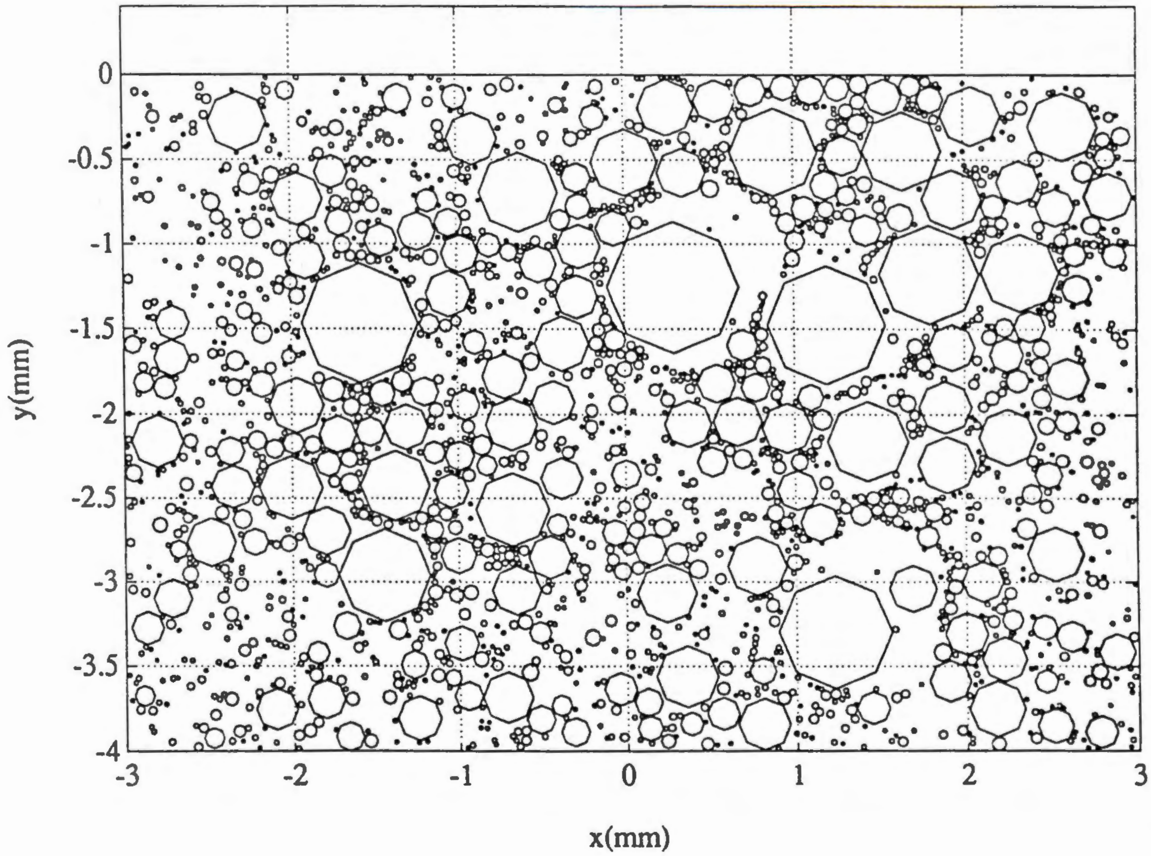

Figure 7. A simulated 2-dimensional soil particle packing based on porosity = 0.5 and the cumulative soil particle size distribution shown in Figure 6.

The above 2-dimensional examples, although less realistic than 3-dimensional cases, are convenient to visualize. It should be pointed out, however, that the corresponding 3-dimensional applications are not fundamentally different. The 2-d approach can be easily extended to 3-d; however, the computational effort may increase significantly.

CHAPTER V

APPLICATIONS

1. PREDICTION OF DISPLACEMENT FIELD

With the basic structure of the Monte Carlo simulation described above, many problems in soils or in other particulate systems previously considered very difficult can be approached in a more straightforward manner. Consider the classical problem of a soil mass indented or penetrated by a rigid punch, as illustrated in Figure 8. In the simulation, particles immediately beneath, and in actual contact with, the punch surface are assumed to move straight downward, creating overlaps. Particles adjacent to these particles are, in turn, assumed to move along lines normal to the tangent of contact points to eliminate the overlaps. Thus, the displacement field (Figure 9) can be generated based on iterations similar to those described in Steps 3 and 4 above --- the two steps used in generating the "soil" mass.

Results indicate that the particles within a triangular area immediately below the punch tend to move downward, while the particle displacements beyond this triangular area tend to be in a radial pattern. Soil immediately below the punch is compacted; that is, the particles are pushed closer together, eliminating much of the void space. Soil is then pushed outward, and the surface of the surrounding soil heaves. Magnitude of particle displacement

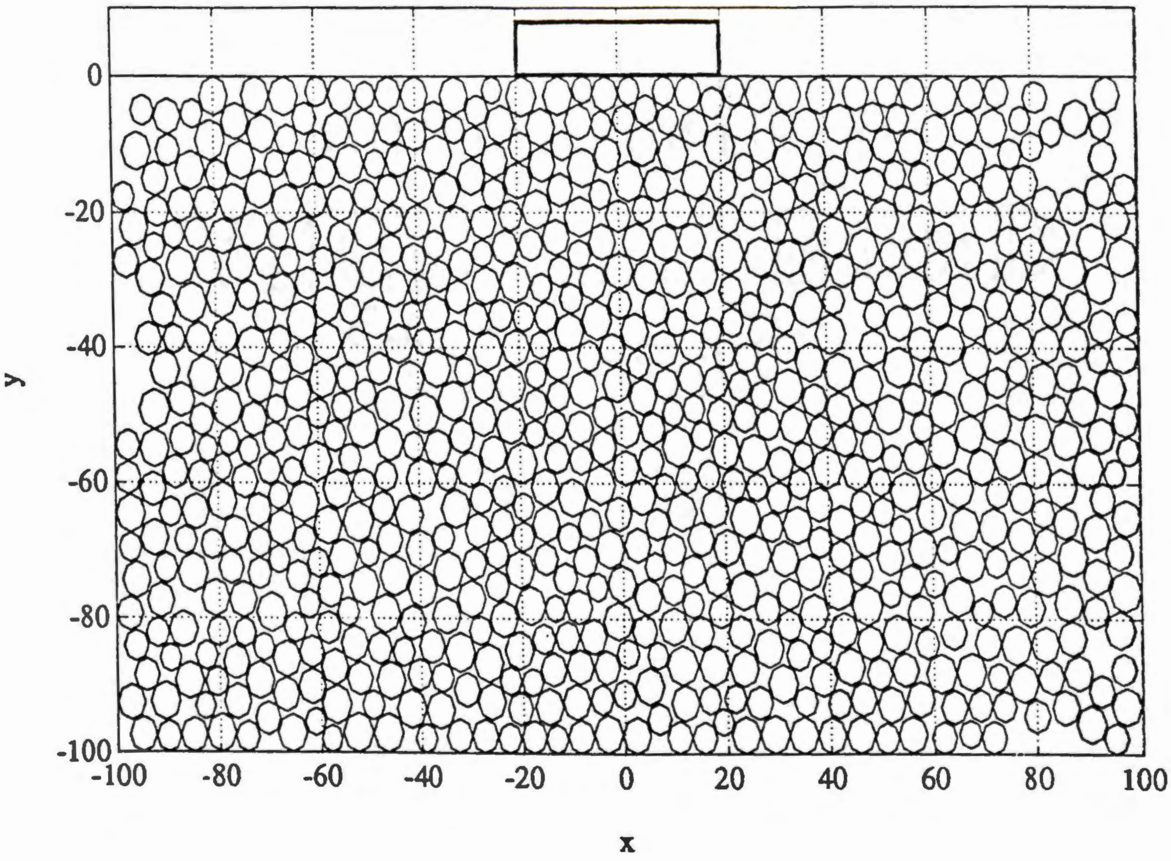

Figure 8. Simulated "soil" particle packing with a uniform probability particle size distribution by number (porosity = 0.22, r_{min} = 2, r_{max} = 3), prior to being indented by a rigid punch with width/2 = 20.

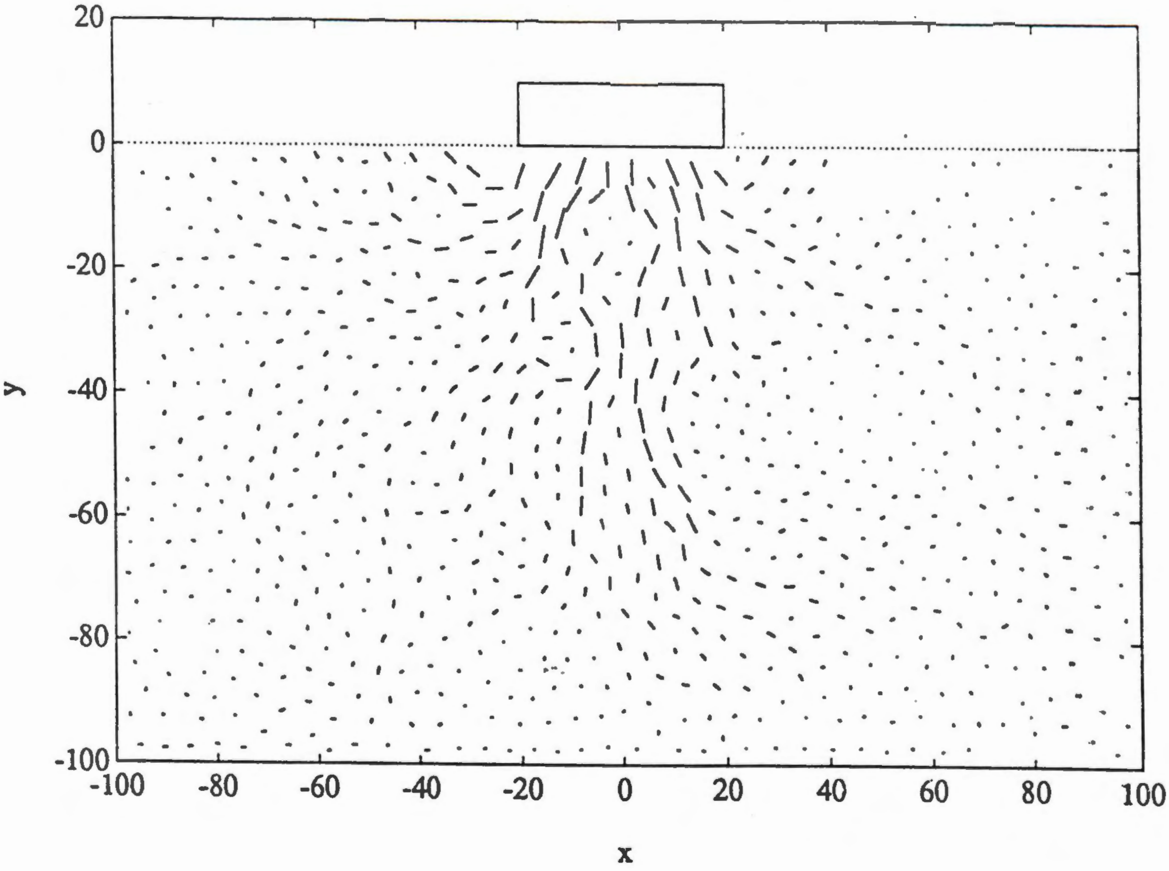

Figure 9. The displacement field for the "soil" mass shown in Figure 8. Vertical displacement of the punch = 5.

decreases with increased distance from the punch or with increased angle from the direction of punch travel. Surface particles adjacent to the punch are moved upward and laterally. Results of this simulation exercise agree closely with the physical observations made by Vesic (1963, 1967a, 1967b) using densely packed sand indented by a hard object. The results are also very similar to those achieved in an experiment using a stack of rods loaded by a rigid punch (Lambe and Whitman, 1979. p. 197). A significant difference is that the simulated displacement field of Figure 9 does not show an obvious boundary to the failure zone under the punch. A possible explanation is that the current simulation did not consider the influence of friction forces between adjacent particles. Further, the differences may also be due to the loose state of the soil packing assumed in the simulation. As friction and other factors are considered in the modeling process, more complicated soil problems could be approached and more reliable results could reasonably be expected.

In another example using the simulation (Figure 10), the range of the uniform probability particle size distribution is reduced with $r_{min} = 1$ and $r_{max} = 2$, while the porosity $n = 0.22$ is maintained the same as the one used in the previous example. The simulated soil particle displacement field with the same punch width and same punch vertical displacement as those in the example above is shown in Figure 11. This example shows that as particle sizes are reduced, the tendencies of displacement field described above become even more pronounced.

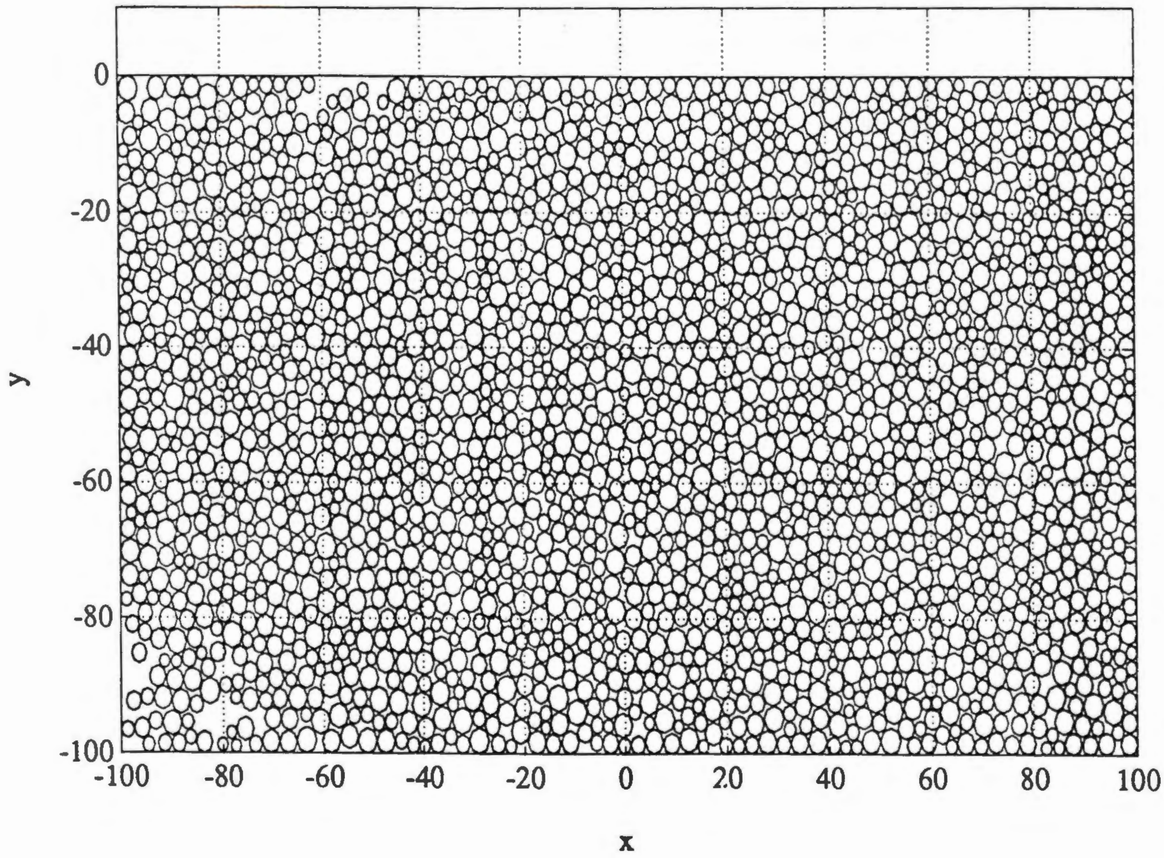

Figure 10. Simulated "soil" particle packing with a uniform probability particle size distribution by number (porosity = 0.22, r_{min} = 1, r_{max} = 2), prior to being indented by a rigid punch with width/2 = 20.

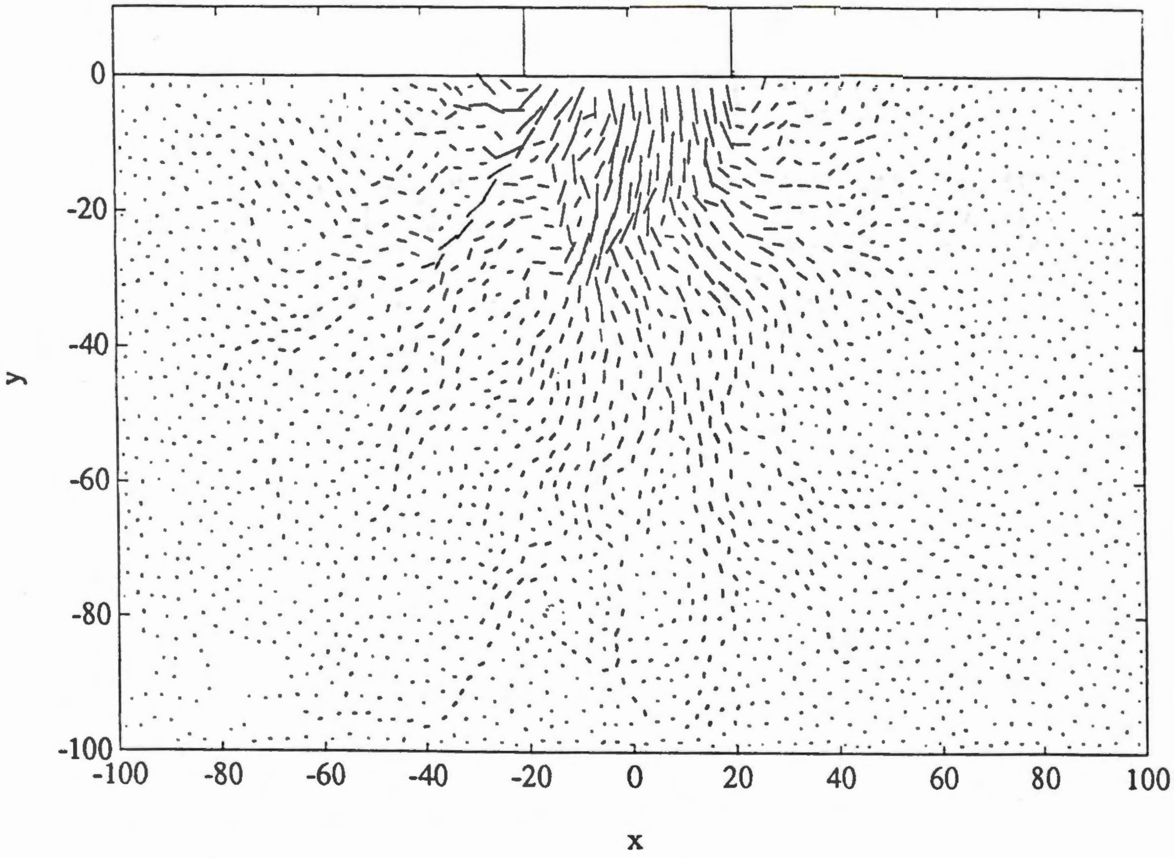

Figure 11. The displacement field for the "soil" mass shown in Figure 10.
Vertical displacement of the punch = 5.

The two examples above also illustrate a powerful characteristic of non-physical simulation; influence of an individual variable or factor (e.g., the influence of particle size in the above examples) is easily isolated from the influence of other variables or factors (e.g., the porosity, punch width, container width, or container depth). The 'isolating influence factor (or variable)' technique frequently offers a short cut leading directly to a deeper understanding of very sophisticated phenomena. However, such a technique may be very difficult, or perhaps even impossible, to use in real physical experiments.

The effect of porosity is illustrated in the third example. In this instance, all particles have uniform size $r=3$ and the porosity is reduced to 0.1744 (Figure 12). The simulated displacement path (Figure 13) shows a clear triangular area immediately below the punch. A "W"-shaped displacement path is clearly evident beyond this triangular area. This displacement field is characteristic of the behavior of very densely packed sand indented by a hard object (Vesic, 1963, 1967a, 1967b). The individual stages of the "soil" displacement process are shown in Figures 14 through 18. Figure 19 combines all segments of displacements occurring in the several stages to form the displacement curves for individual particles.

To more closely compare displacement fields associated with denser particle packing with those involving loose packing of particles, physical experiments were performed at the Department of Agricultural Engineering, The University

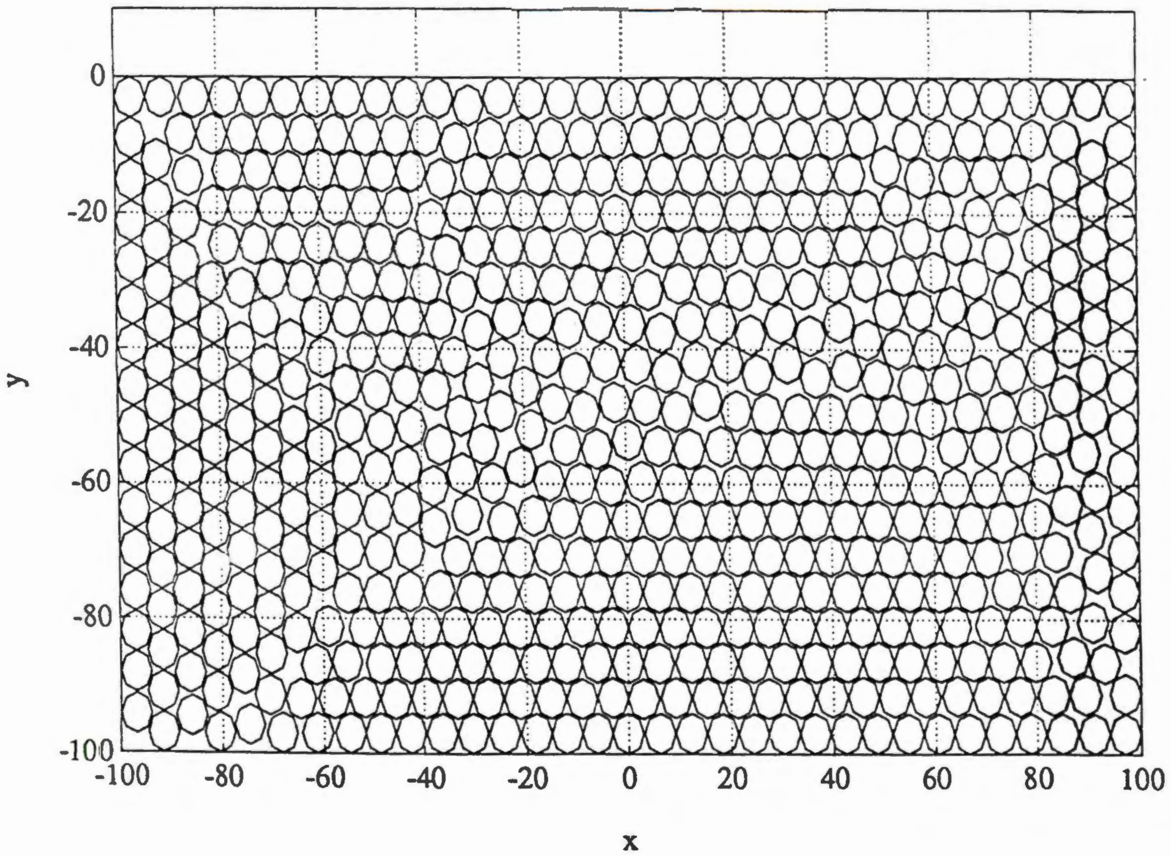

Figure 12. Simulated "soil" particle packing with a uniform size r = 3 and porosity = 0.1744, prior to being indented by a rigid punch with width/2 = 20.

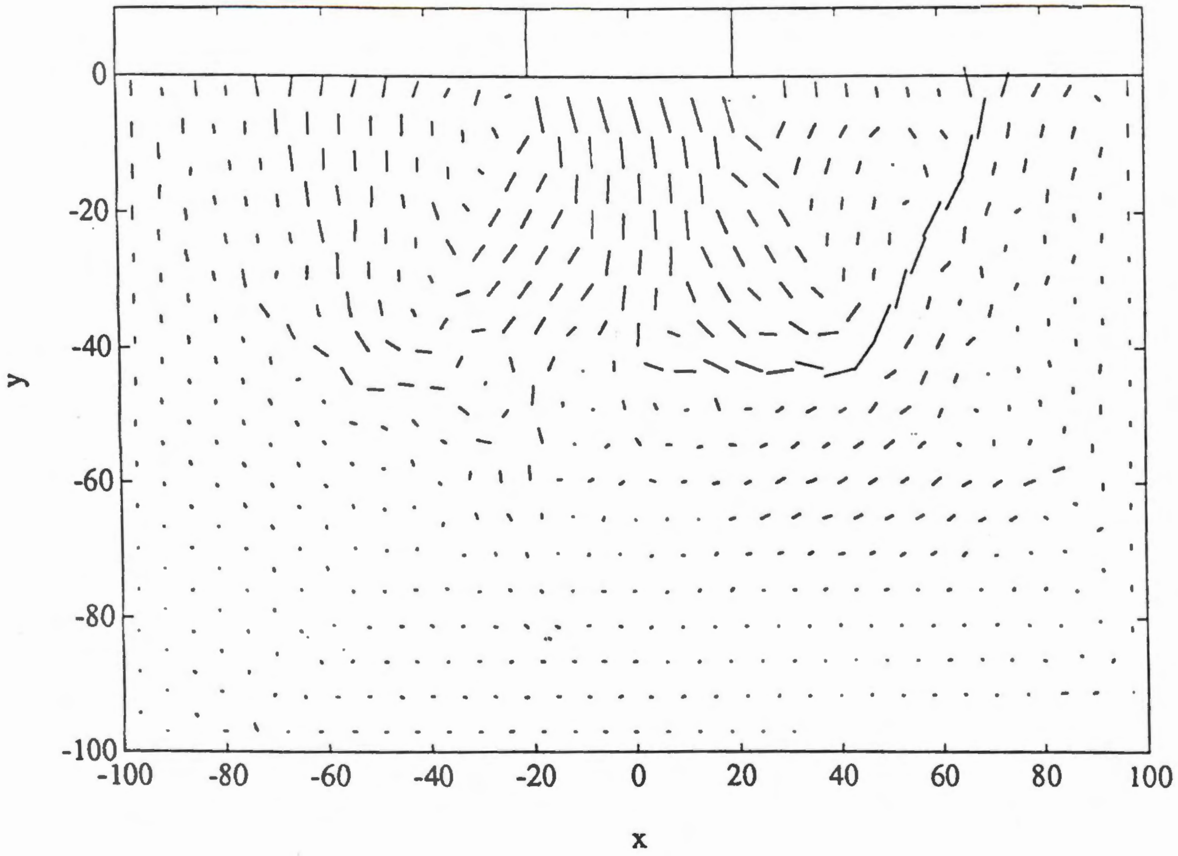

Figure 13. The displacement field for the "soil" mass shown in Figure 12. Total vertical displacement of the punch = 5.

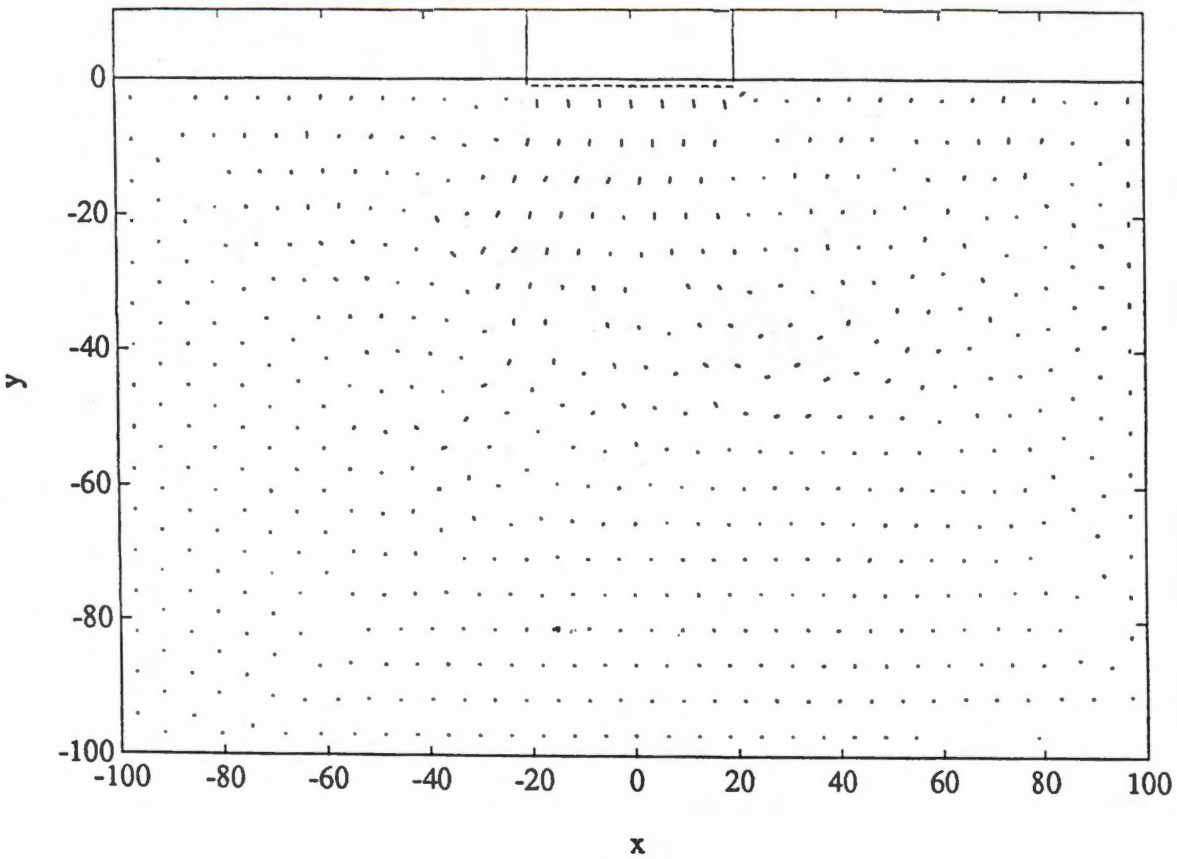

Figure 14. Incremental particle displacement for the punch-"soil" system shown in Figure 12 (punch depth = 0, and depth increment = 1).

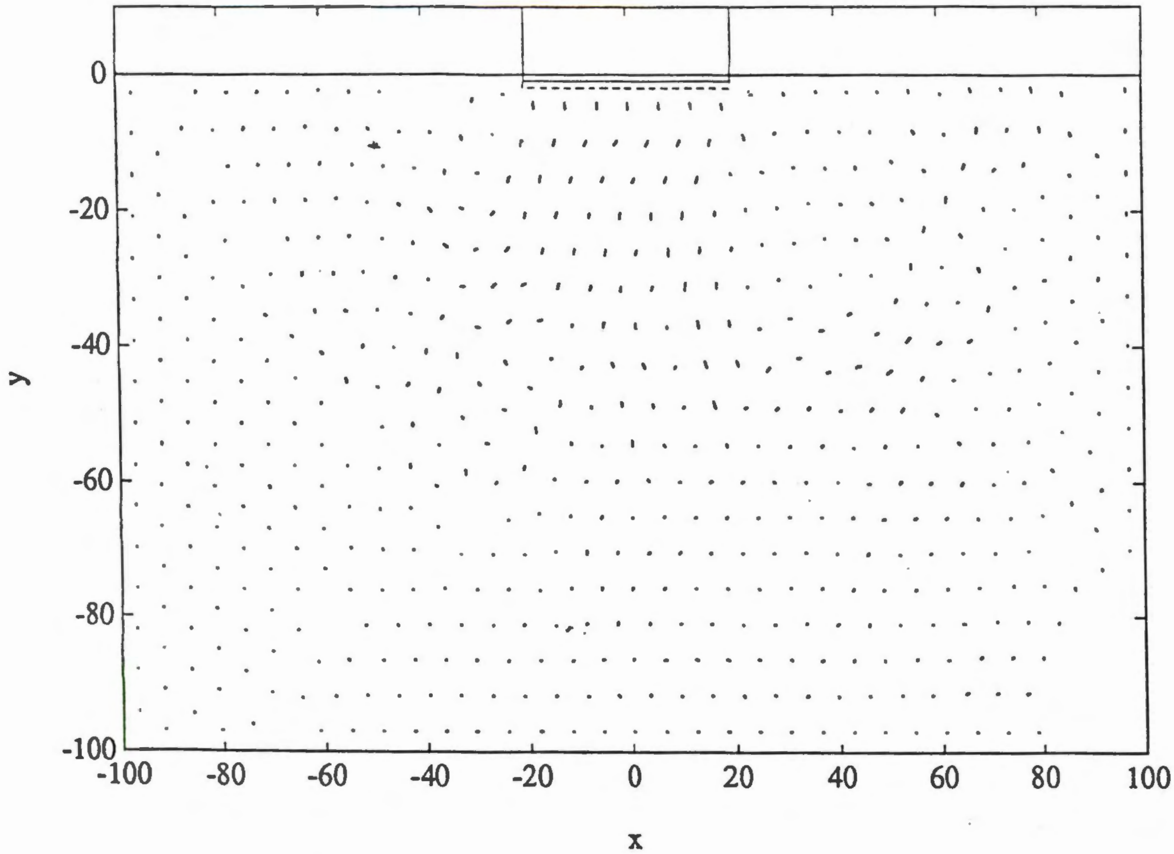

Figure 15. Incremental particle displacement for the punch-"soil" system shown in Figure 12 (punch depth = 1, and depth increment = 1).

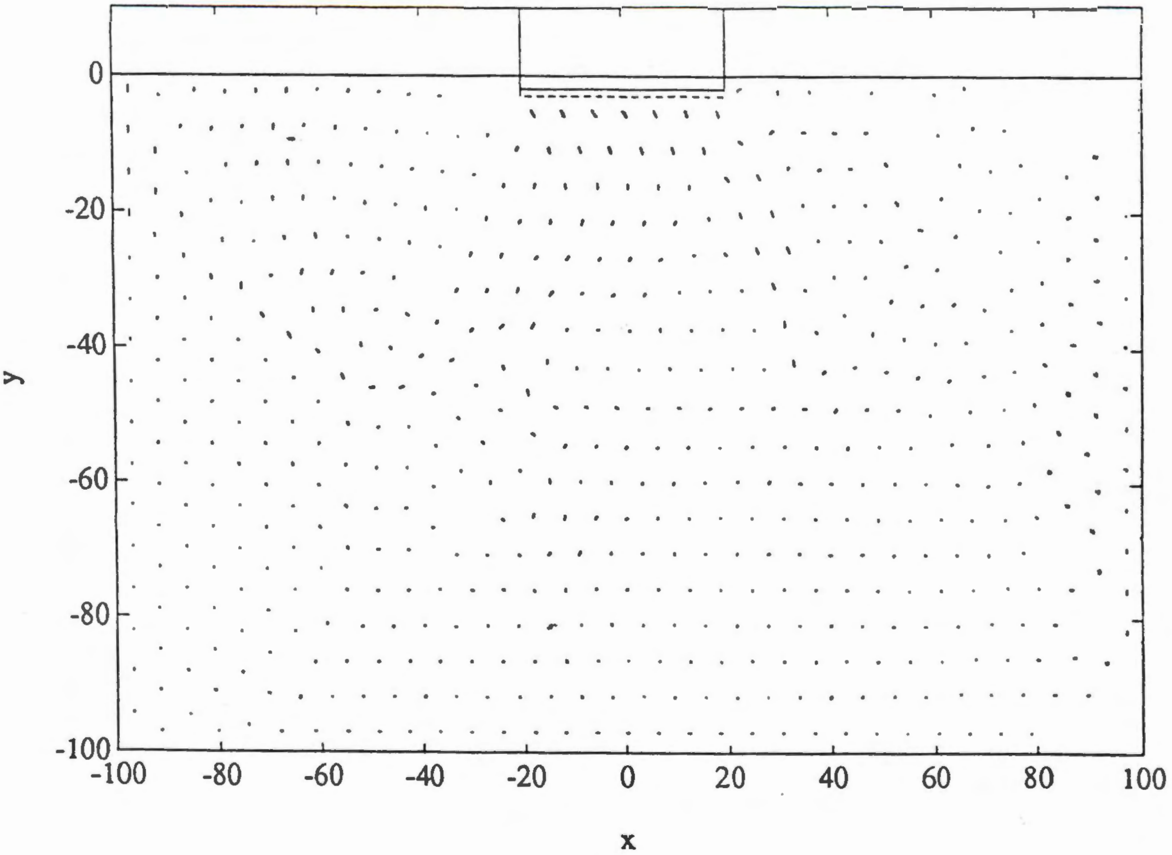

Figure 16. Incremental particle displacement for the punch-"soil" system shown in Figure 12 (punch depth = 2, and depth increment = 1).

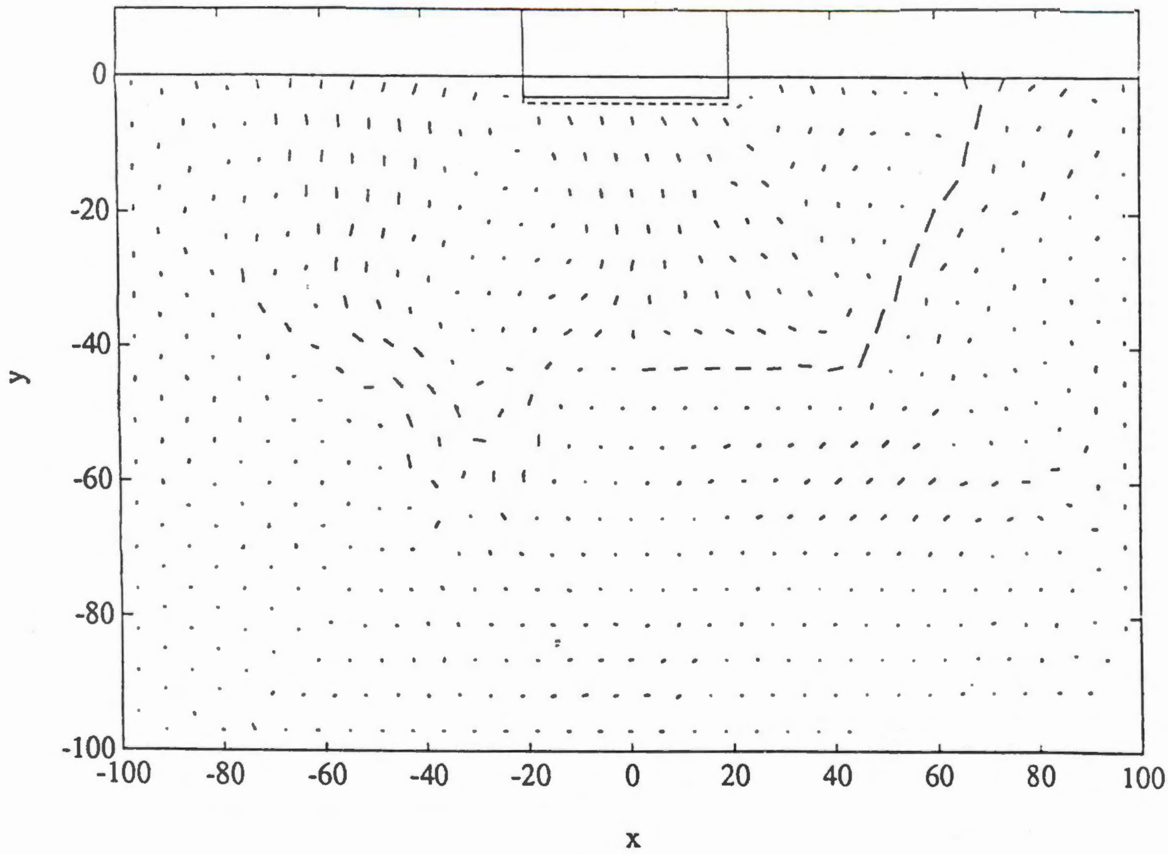

Figure 17. Incremental particle displacement for the punch-"soil" system shown in Figure 12 (punch depth = 3, and depth increment = 1).

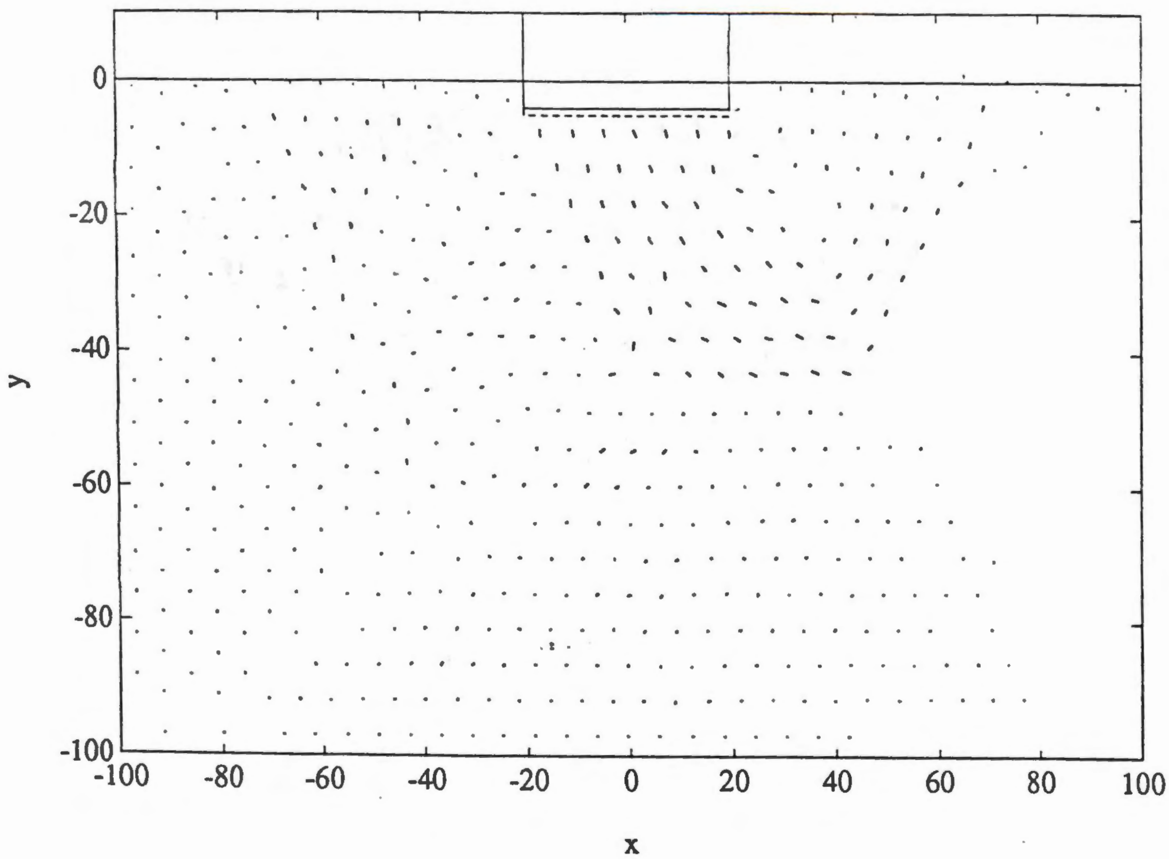

Figure 18. Incremental particle displacement for the punch-"soil" system shown in Figure 12 (punch depth = 4, and depth increment = 1).

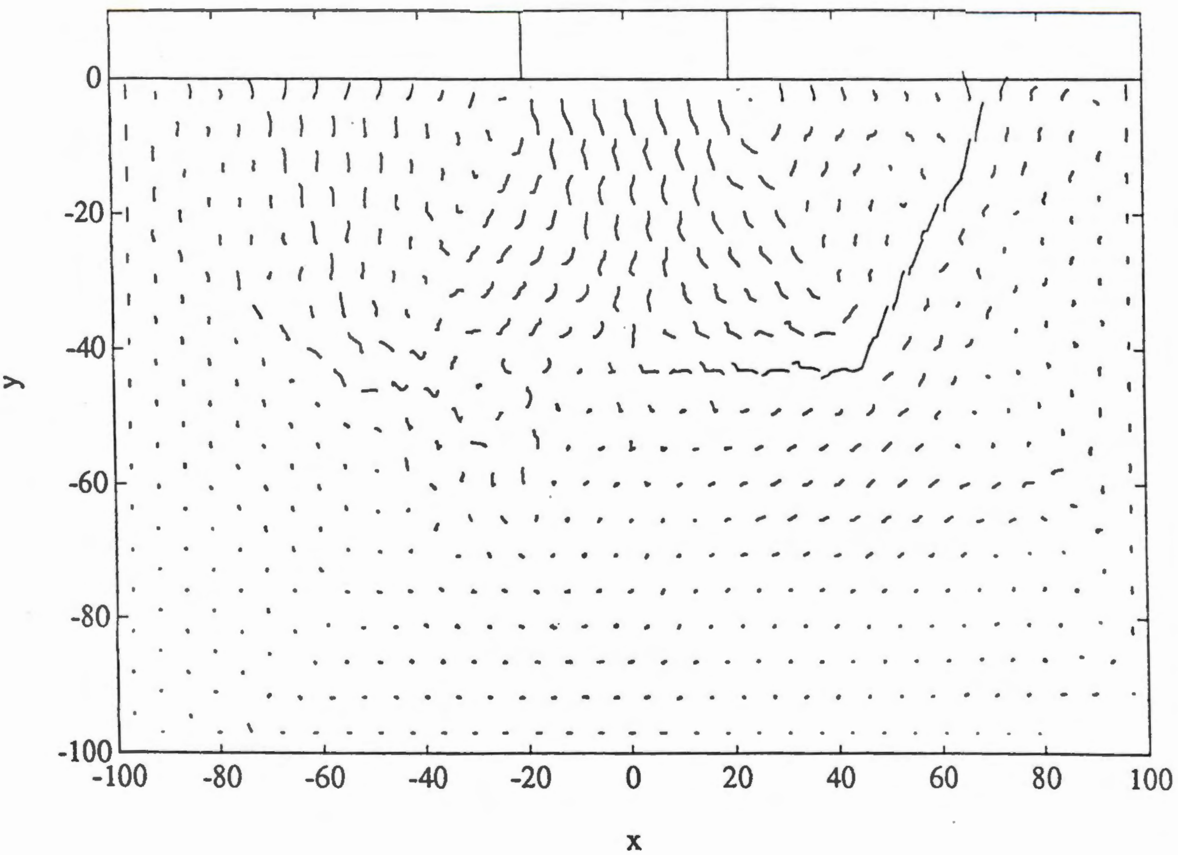

Figure 19. Total particle displacement for the "soil" mass show in Figure 12 under influence of a penetrating punch.

of Tennessee, Knoxville. Further, the experiments allowed comparison of behaviors of real systems to results based upon simulation. The experiments were similar to those performed by other investigators (Lambe and Whitman, 1979. p.197; Tokue, 1979; Scott, 1987; Hardin, 1989), but there were distinct differences.

As in previous experiments, steel rods instead of real soil particles were used; however, these displacement field investigations were performed not only using very dense packing but employing relatively loose packing. Steel rods, with a uniform length = 5 inches = 127 mm and three different radii, r_1 = 1/8 inch = 1.59 mm, r_2 = 3/16 inch = 2.38 mm, and r_3 = 1/4 inch = 3.18 mm, were used to physically simulate soil particles in the 2-dimensional case. A rigid rectangular box was used as a container. The rigid box was made of steel channel with a width = 6 inches = 152.4 mm and thickness = 3/16 inch = 2.38 mm. The inside dimensions of the container were: length x width x height = 200 mm x 300 mm x 200 mm. A rectangular-shaped punch, with width = 50 mm, length = 127 mm (note: punch length equal to the length of rods), and effective height = 20 mm, was used in the experiments. To facilitate operation and observation, the top side of the container was open and the front side was fitted with a transparent cover. To simplify analysis, the displacements of selected indicator rods (instead of all rods) were recorded. In both experiments, one with very dense packing and one with relatively loose packing, the rods initially filled the container to a depth of 180 mm. The designated locations of the indicator rods, the filling depth, and the cross sectional area of the container (from the frontal view) are shown in Figure 20. Since it was difficult to select particles which were precisely positioned at the designated locations of indicators, the actual indicators were selected from among those rods closest to the chosen locations. Selected numbers of rods of the various radii were randomly mixed before they were packed into the container.

In the experiment involving dense packing, 1527 rods with radius = 1.59 mm, 1104 with radius = 2.38 mm, and 410 with radius = 3.18 mm were randomly packed into the container. The porosity in this instance was 0.171. The total indenting depth (20 mm) of the punch was divided into four increments. For each indenting increment, coordinates indicating the positions of the indicator rods were recorded. The displacement field resulting from three typical runs of this experiment is shown in Figure 21.

In the experiment involving loose packing, rods of the same sizes in the same numerical quantity as used in the above experiment were employed in the initial packing. Then, rods within the packing were withdrawn from randomly chosen locations, one at a time, until the numbers of rods were reduced to 1477, 1084 and 390 for radii = 1.59 mm, 2.38 mm and 3.18 mm, respectively. Consequently, the porosity of the packing was increased to approximately 0.2. Experimental procedures similar to those used in the dense packing experiment were followed. The displacement field resulting from three typical runs of the second experiment is shown in Figure 22.

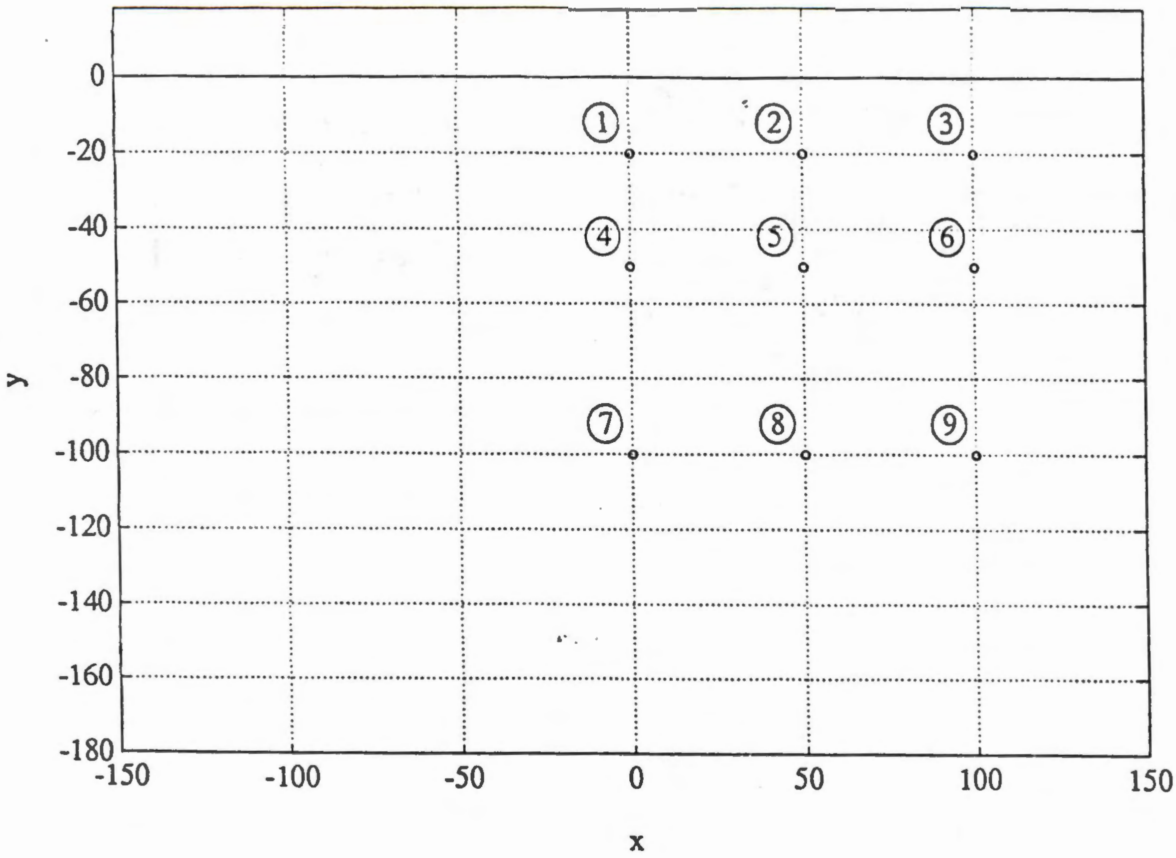

Figure 20. Indicator rods locations at the beginning of the physical experiment.

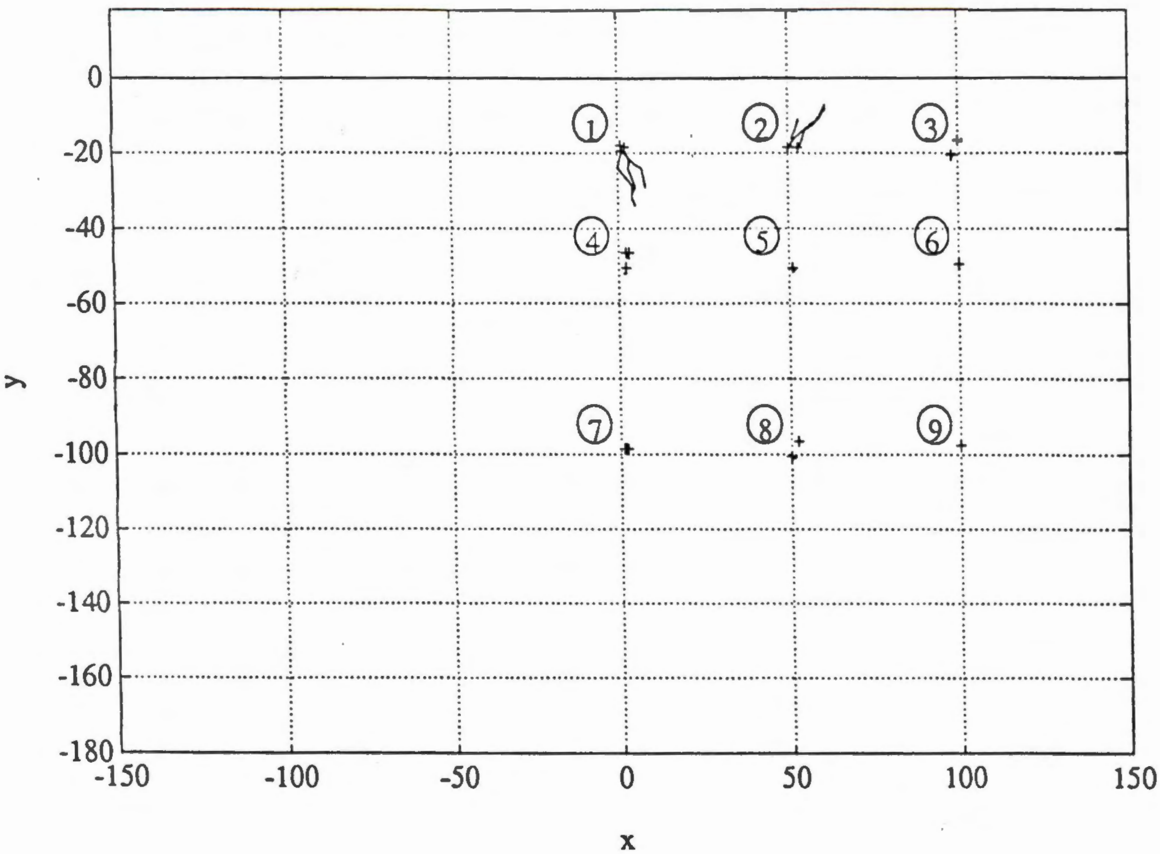

Figure 21. Displacements of indicator rods in three typical runs in a physical experiment with rod packing porosity = 0.171, punch width = 50, total indenting depth = 20, and indenting increment = 5.

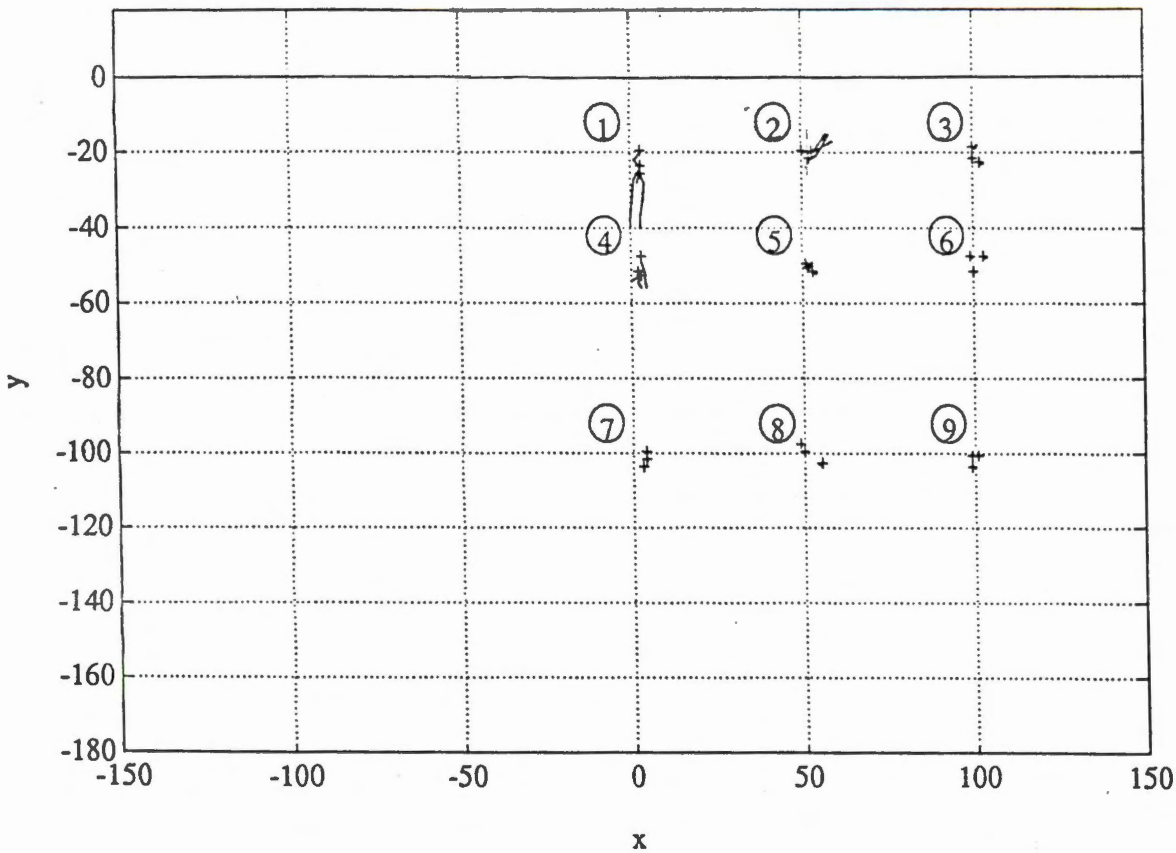

Figure 22. Displacements of indicator rods in three typical runs in a physical experiment with rod packing porosity = 0.2, punch width = 50, total indenting depth = 20, and indenting increment = 5.

Comparing Figure 22 with Figure 21 shows the following:

i) Along the direction of punch travel for some distance, particle displacements within the loosely packed mass were greater than those in the more densely packed mass (see indicators in Locations 1 and 4).

ii) Particle displacements in less dense packing had less tendency to deviate from the direction of punch travel (see indicators in Location 1 and 2).

Both these tendencies are consistent with behaviors predicted by computer simulation in the examples presented above.

A more direct comparison between the displacement field in a real particle packing and one simulated by the computer is given in the following. In this example, the computer simulation considered a porosity value n = 0.2, a value equal to the one used in the second physical experiment. The particle size distribution (a discrete distribution) used in the second physical experiment was approached by using a corresponding piecewise linear distribution. Recall that the proposed simulation technique was designed specifically for a real, continuous distribution such as occurs with soil particles. It was not specifically for a discrete distribution such as the rods represented. The same punch width and the same indenting depth used in the second physical experiment were again used in this computer simulation. The simulated rod packing before indention by the punch is shown in Figure 23. The simulated packing after indention is shown in Figure 24. The shape of the heaved surface in Figure 24 is very similar to the one observed in the corresponding physical experiment. The

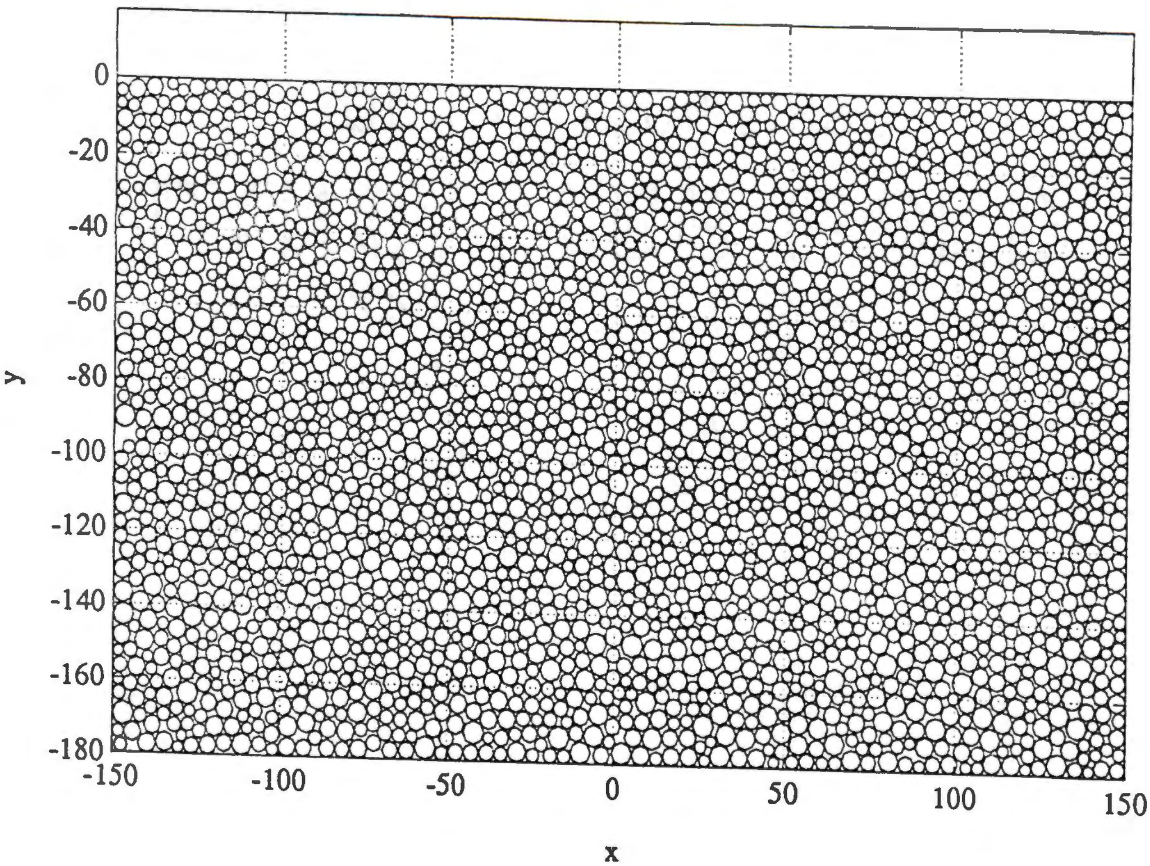

Figure 23. Simulated rod packing with a porosity = 0.2 and rod size distribution adopted from the second physical experiment.

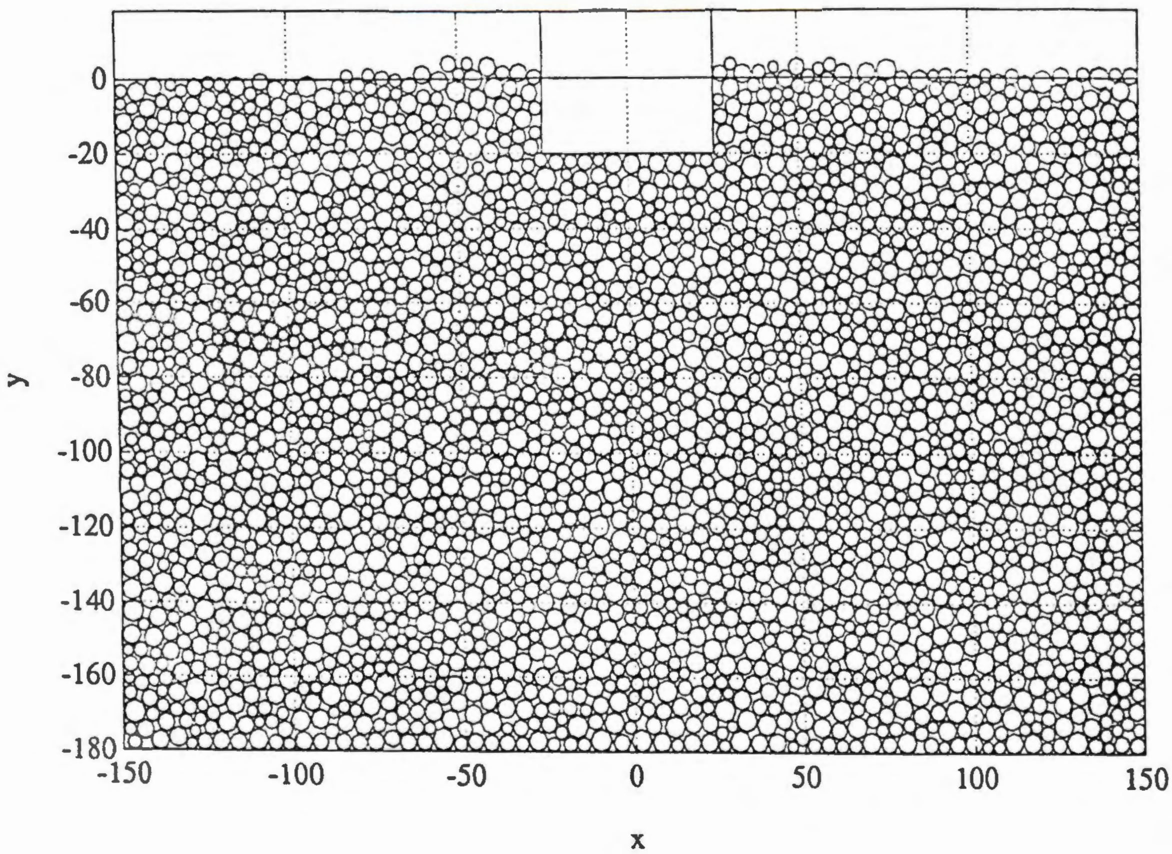

Figure 24. Simulated rod packing for the second physical experiment following indention by a punch with a width = 50 to a depth = 20.

simulated displacement field is shown in Figure 25. The simulated displacement history is shown in Figures 26 through 29. Figure 30 combines all segments of displacements occurring in the several stages to show the displacement curves for individual particles.

Comparing Figure 30 with Figure 22 indicates that:

i) In the area where displacements are observable (see indicators at Locations 1, 2, 4, and 5), the displacement curves from both the physical experiment and the computer simulation display the same tendencies of direction and magnitude. This means that the simulated results agree well with those from the physical experiment in the area of large particle movement.

ii) In areas well away from the indenting object (see indicators at Locations 7, 8, 9, 6, and 3), the simulation tended to over predict the magnitudes of particle displacements. The reason for the over prediction may be, as discussed earlier, due to the effect of the friction forces which were ignored in the present modelling effort. This information may be useful in calibrating the model results or in calibrating the model itself in the future.

It should be pointed out that the physical experiments above, with exactly the same parameters (such as the same particle size distribution, the same overall porosity, etc.) in three different runs, produced somewhat different displacement fields (see Figure 21 or 22). This phenomenon, which is common in agriculture soils, may simply be uncertainty characteristic of such a discrete and random system. This uncertainty characteristic embarrasses every

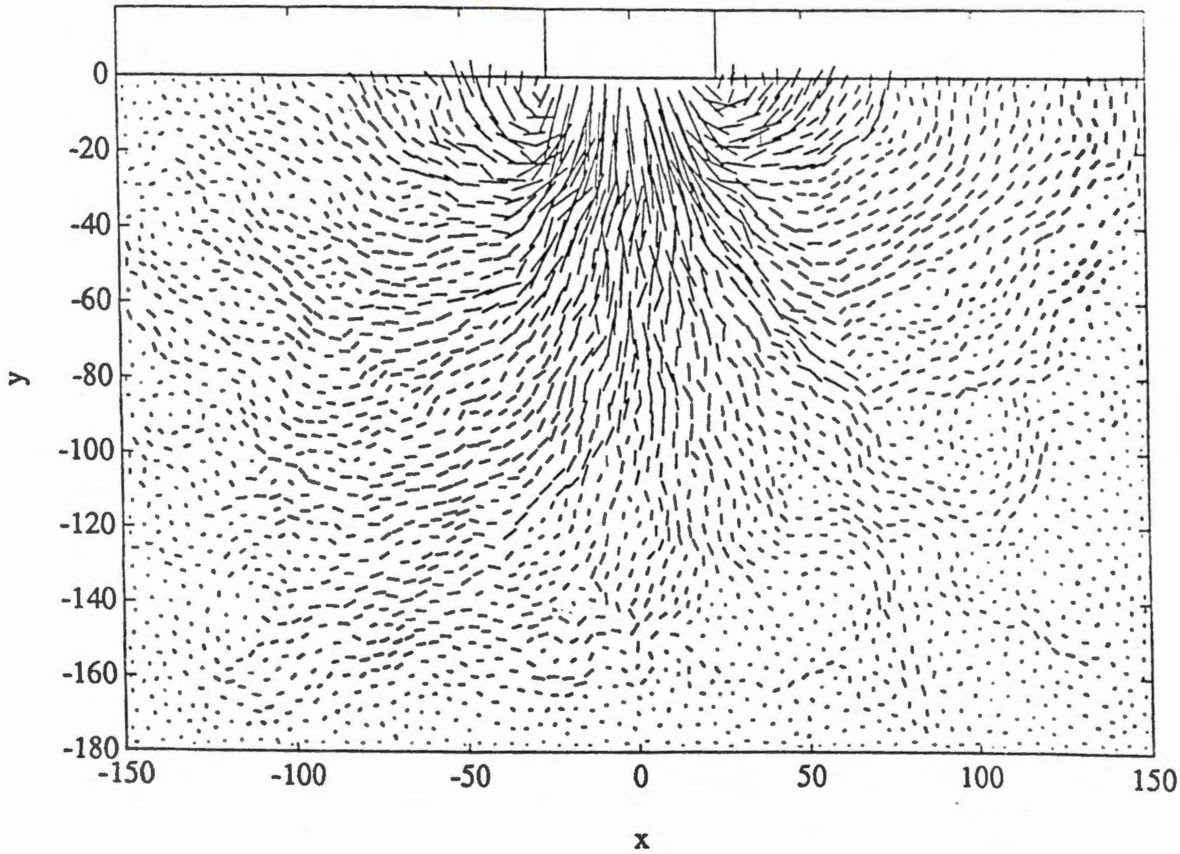

Figure 25. Simulated rod displacement field for the second physical experiment.

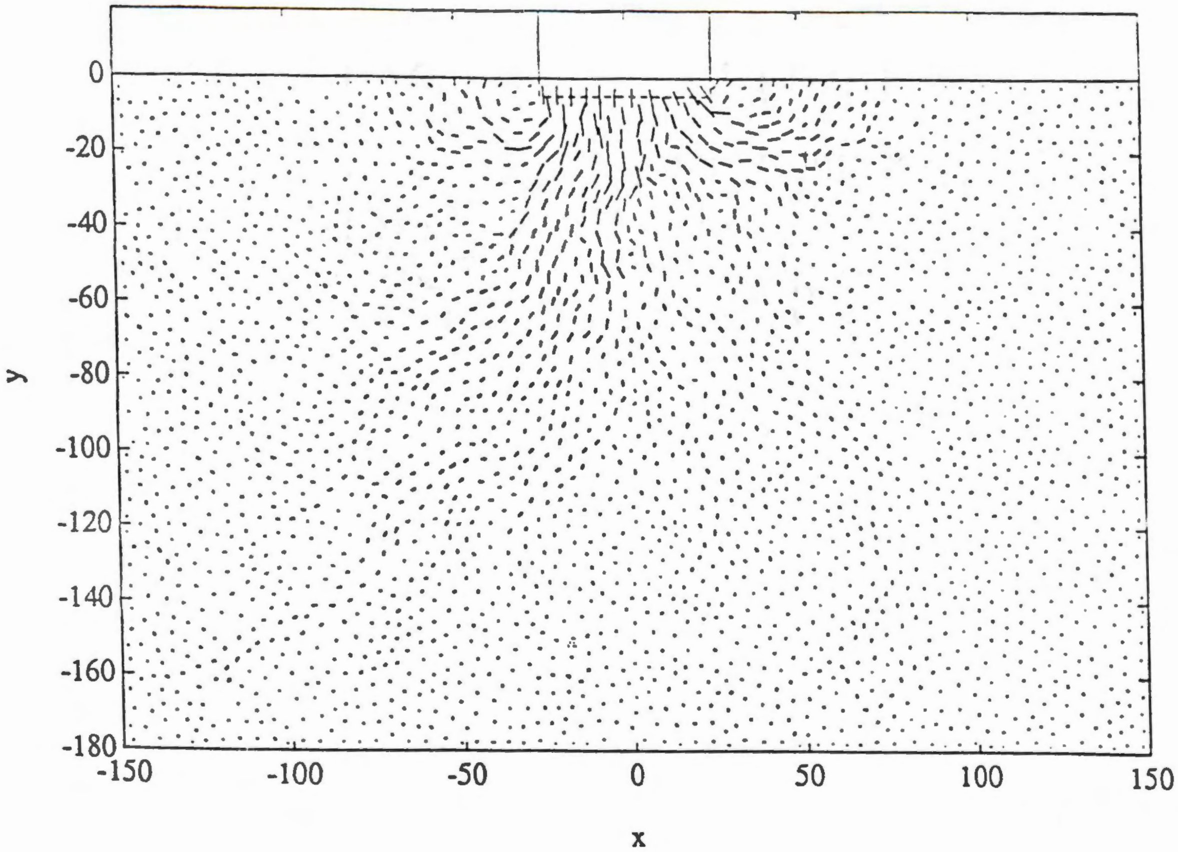

Figure 26. Simulated incremental rod displacement for the second physical experiment (punch depth = 0, and depth increment = 5).

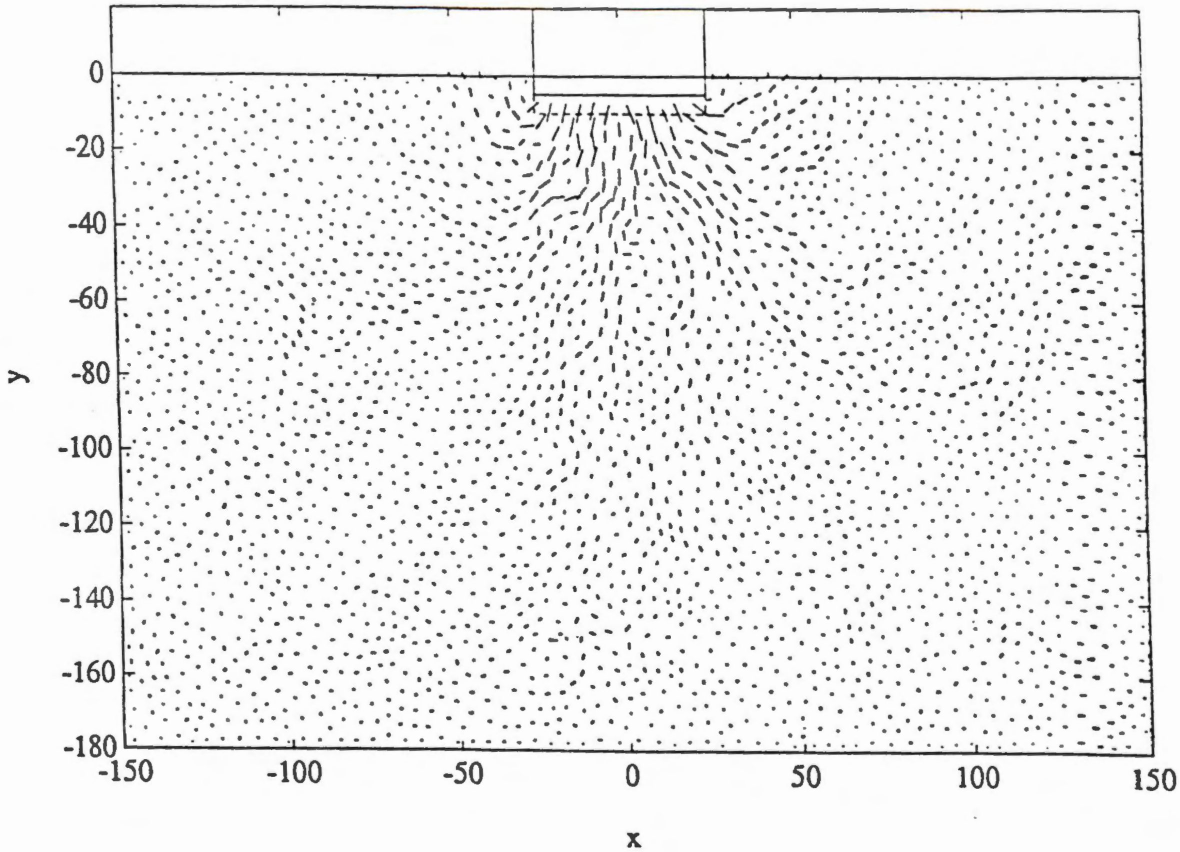

Figure 27. Simulated incremental rod displacement for the second physical experiment (punch depth = 5, and depth increment = 5).

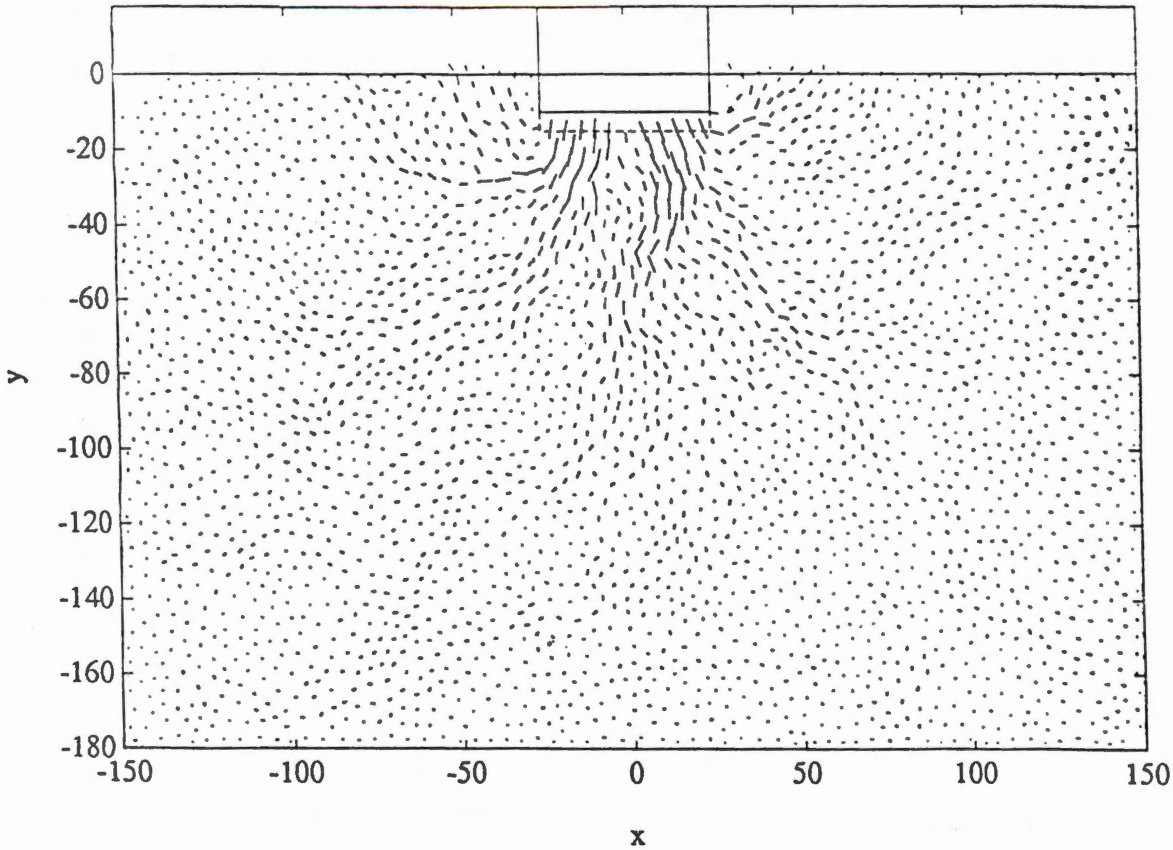

Figure 28. Simulated incremental rod displacement for the second physical experiment (punch depth = 10, and depth increment = 5).

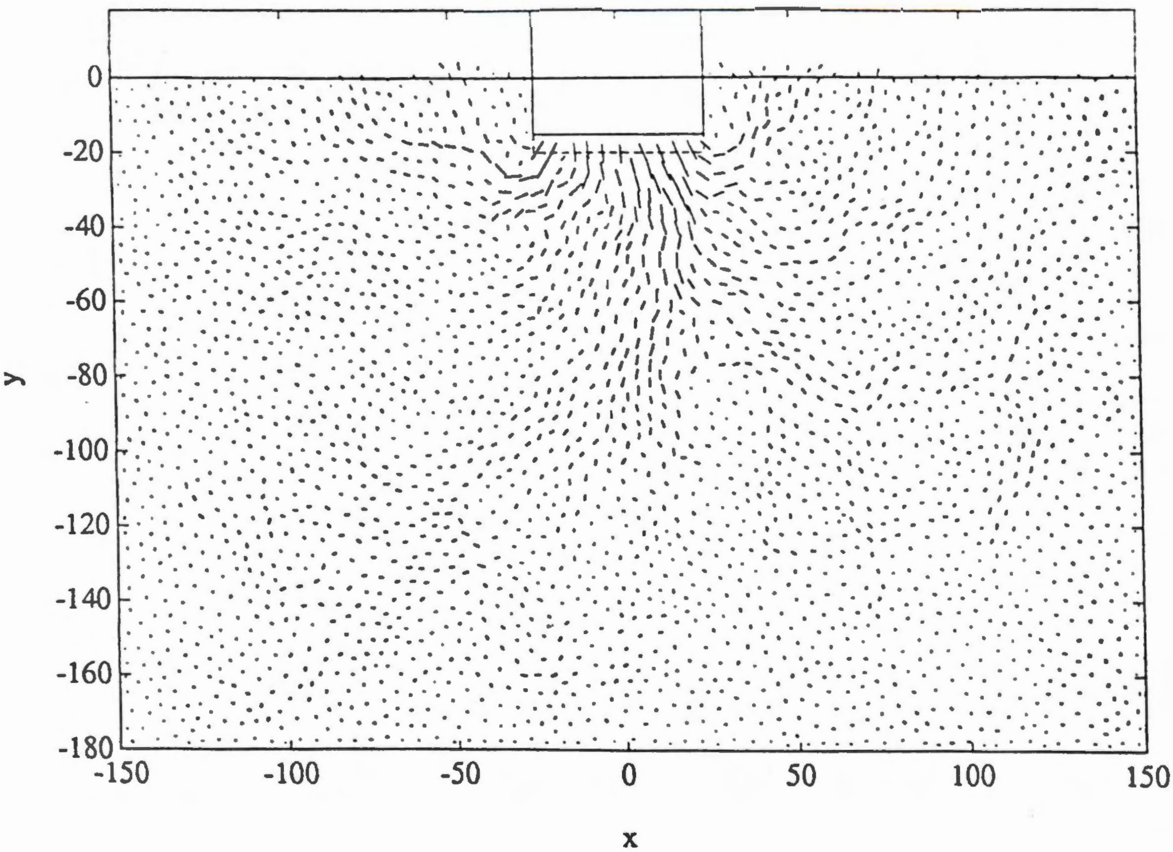

Figure 29. Simulated incremental rod displacement for the second physical experiment (punch depth = 15, and depth increment = 5).

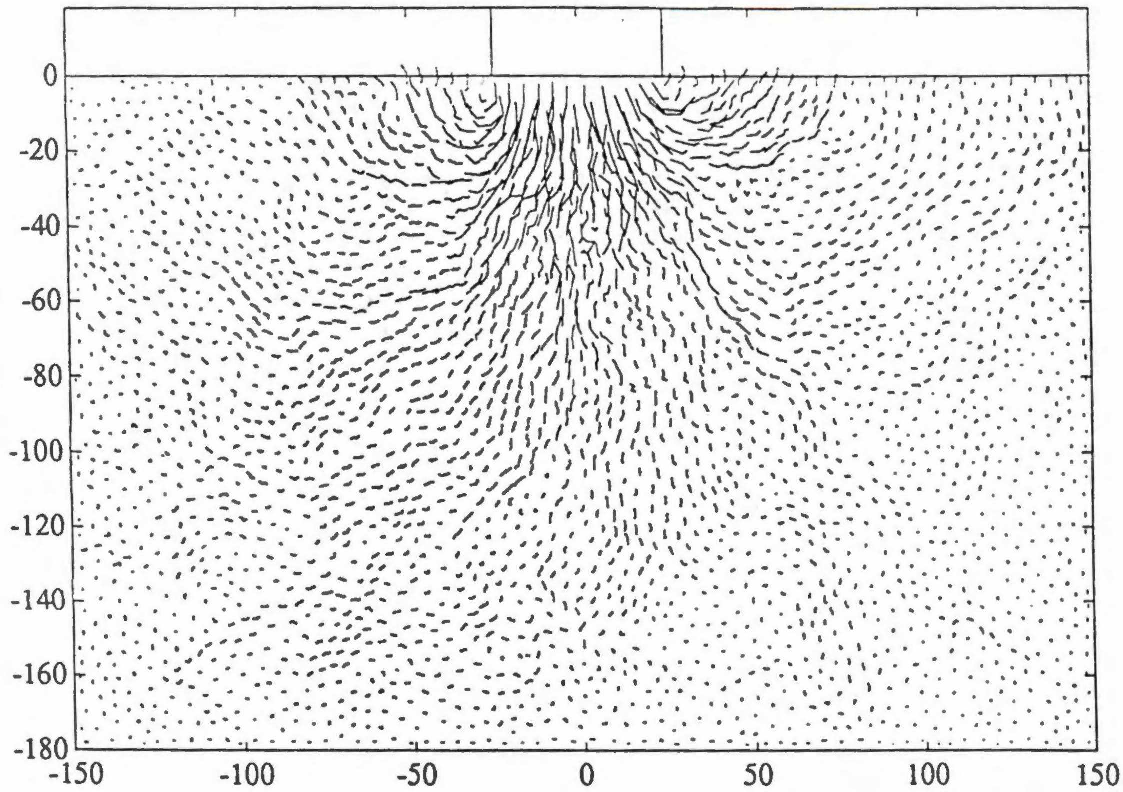

Figure 30. Simulated individual particle displacements for the second physical experiment.

determinate model, because a given determinate result will not necessarily agree with results from several different runs of a real physical experiment. Using average values of independent variables (or parameters) to predict the average value of a dependent variable is fundamentally wrong with such complicated systems such as soil, because of the non linearity of such systems. The large variations observed in the physical experiments described above come from the discrete and random construction. In different runs of a physical experiment, individual remixing and random placement will not be expected to produce exactly the same micro construction. Therefore, the different runs will, and should, give somewhat different displacement fields. Likewise, different runs of the Monte Carlo simulation will very likely give somewhat different results (Note: in the computer simulation example above, only one run is given). Therefore, one should not be surprised when a result from an individual run of the Monte Carlo simulation does not coincide exactly with a result from an individual run of the associated physical experiment. However, the average results over several simulations and repetitions of the physical experiment would be expected to agree, provided the Monte Carlo simulation realistically reflects the uncertain system.

2. PREDICTION OF THE DENSEST PACKING

Prediction of densest particle packing is another application of this simulation technique. If the required porosity is not specified in the procedure to generate and place soil particles, the number of soil particles in the fixed volume will continue to increase, resulting in smaller and smaller porosity values. As the porosity approaches the smallest possible value for a given particle size distribution, the iteration to produce and install another particle in the soil mass will require more and more processing time. This indicates that densest random packing of the particles is being approached. The porosity value associated with densest packing is very useful in analyses of soil compaction, bearing capacity, seepage, and similar problems.

One example of generated densest random packing has already been shown in Figure 12 above. The computer elapsed time for generating individual particles vs. particle number is shown in Figure 31. The computer cumulative elapsed time vs. the number of particles is shown in Figure 32. In Figure 32, the asymptotic value of the number of particles is simply the maximum number of particles which can be held in the specified control volume. Figures 33 and 34 show the relationship between porosity and the number of particles, and between porosity and the computer cumulative elapsed time, respectively. The asymptotic value for porosity shown in Figure 34 is the porosity associated with the densest random packing.

Another example used smaller uniform particles of radius $r=2$. The simulated very dense random packing is shown in Figure 35. The typical behaviors of this method for this specific set of particle parameters are shown in Figures 36 though 39.

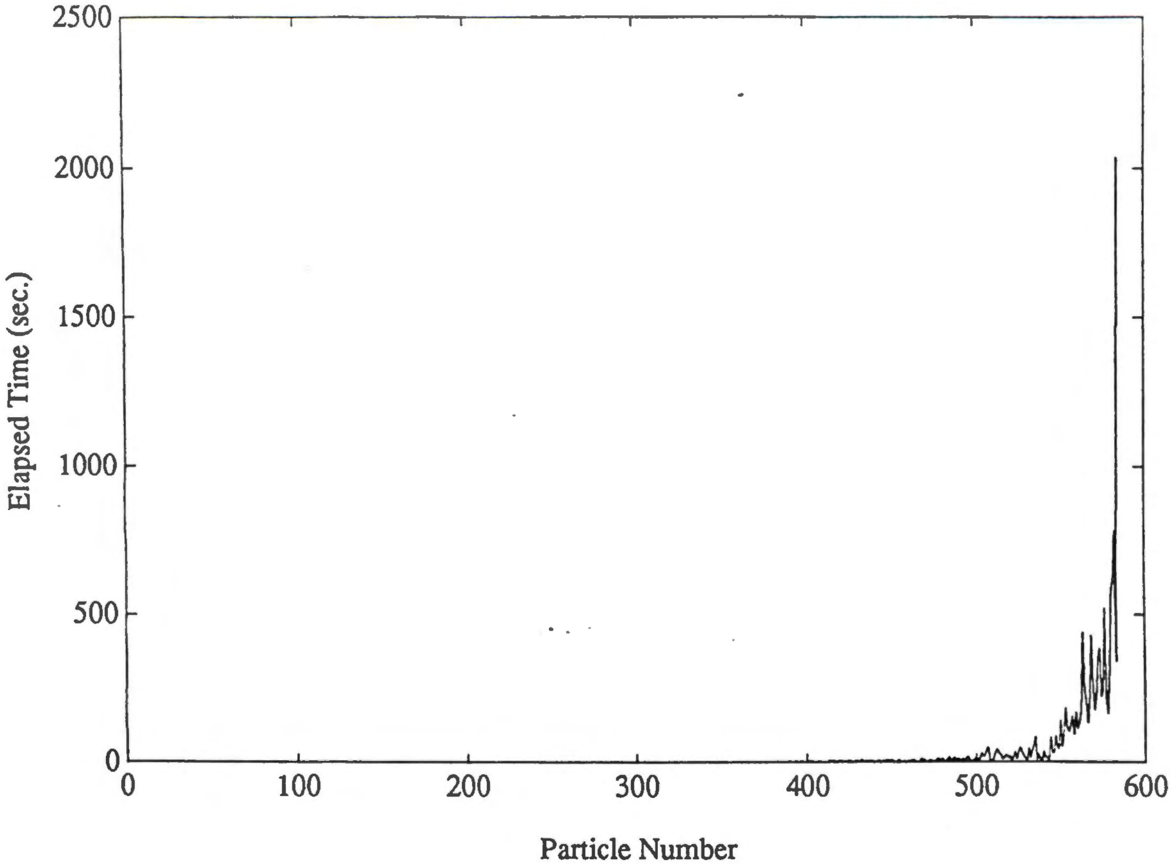

Figure 31. Elapsed time vs. particle number for the particle packing shown in Figure 12.

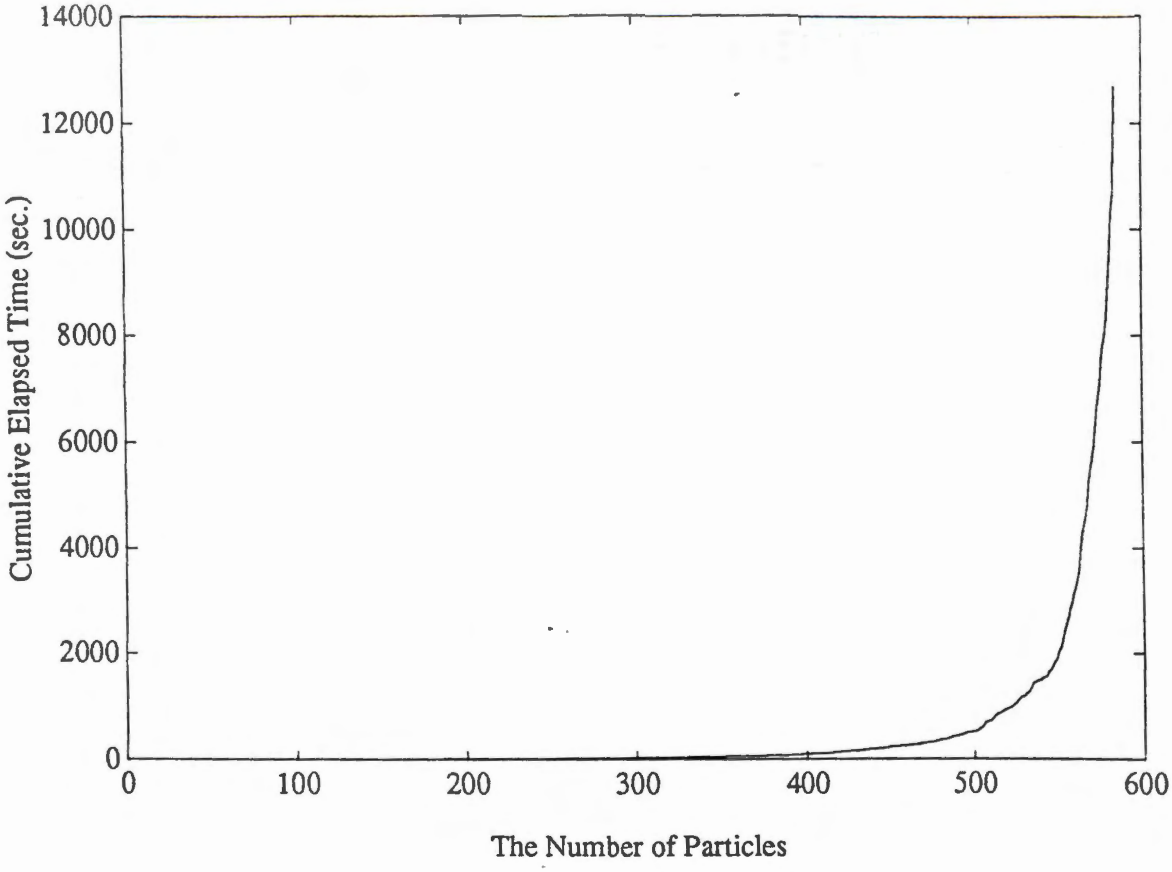

Figure 32. Cumulative elapsed time vs. the number of particles for the particle packing shown in Figure 12.

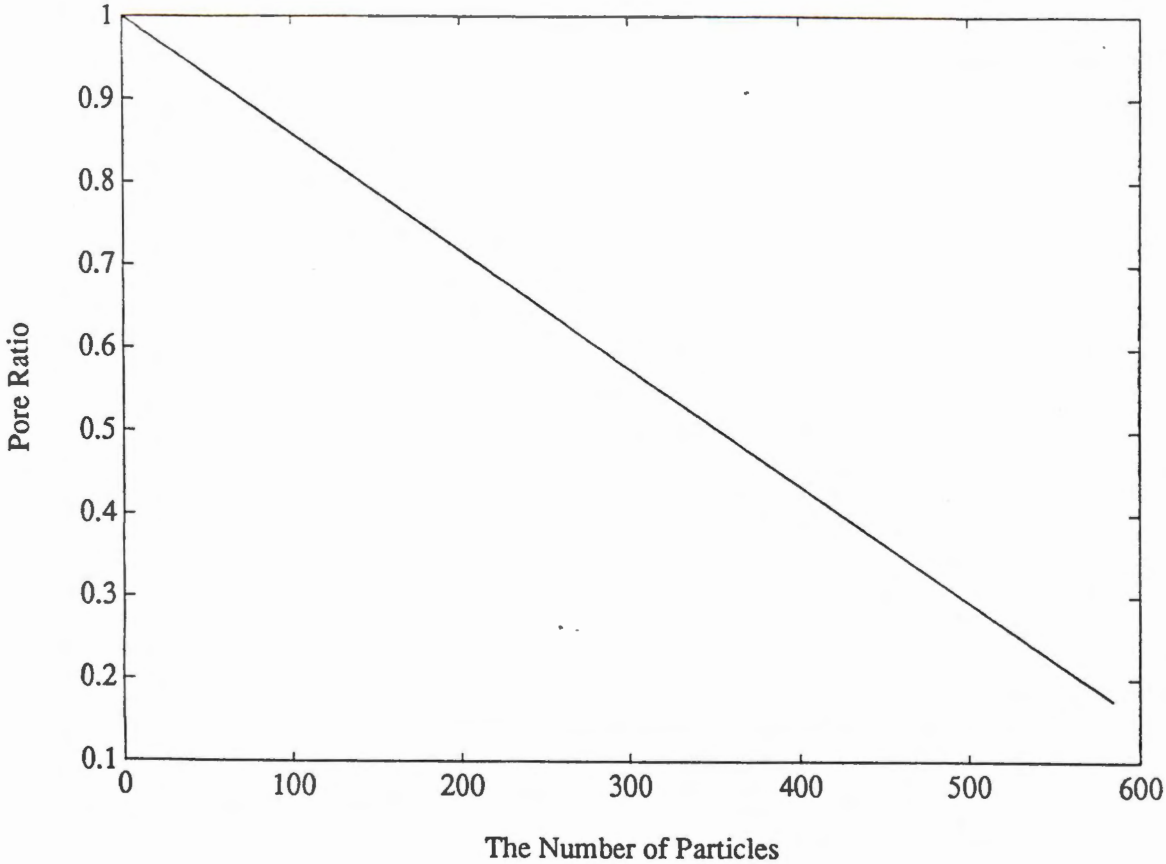

Figure 33. Porosity vs. the number of particles for the particle packing shown in Figure 12.

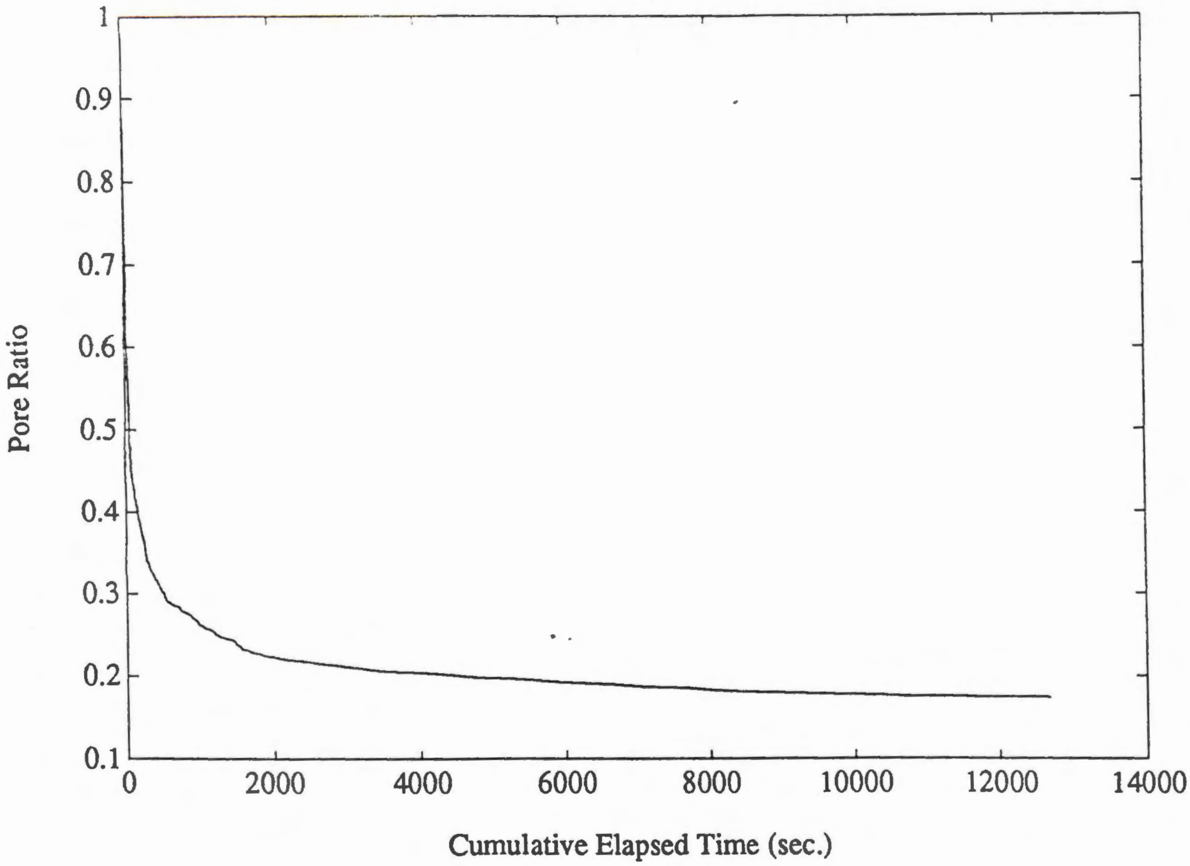

Figure 34. Porosity vs. cumulative elapsed time for the particle packing shown in Figure 12.

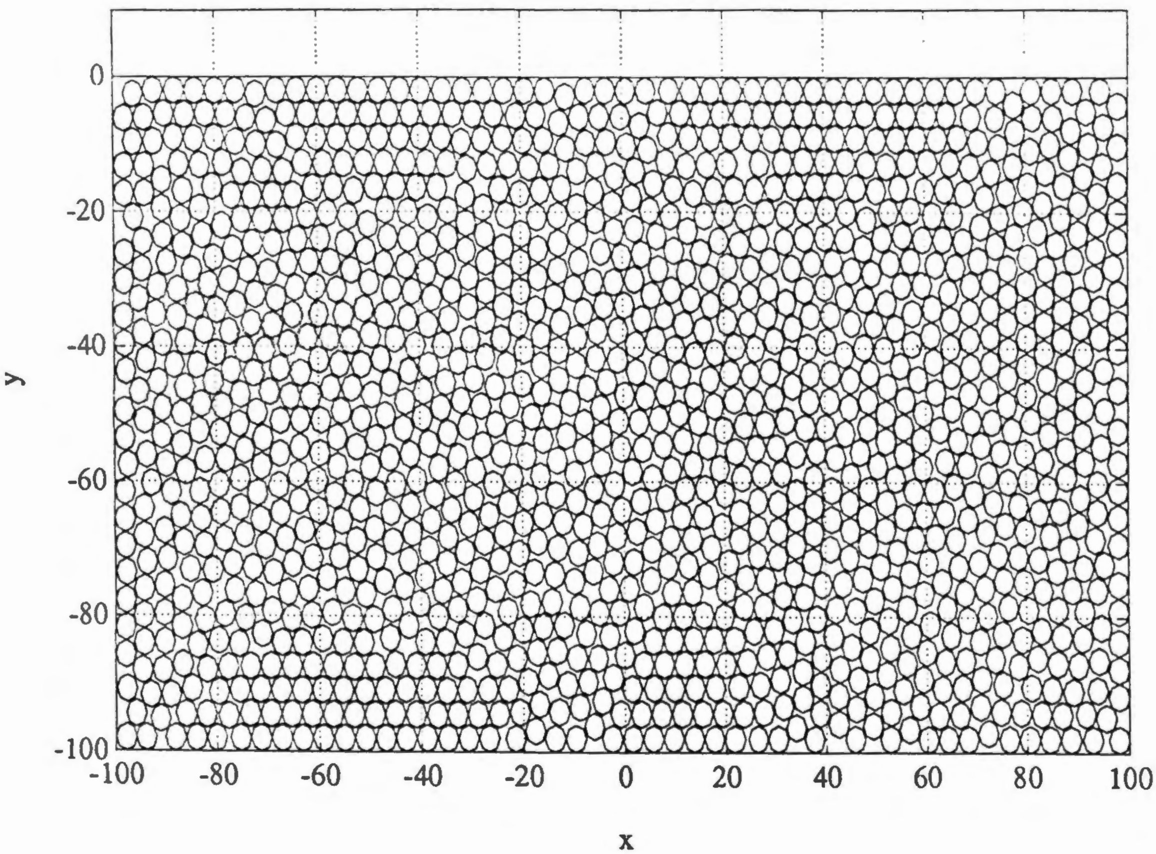

Figure 35. Simulated densest random particle packing with a uniform particle size r = 2, width of control volume = 200, and depth of control volume = 100.

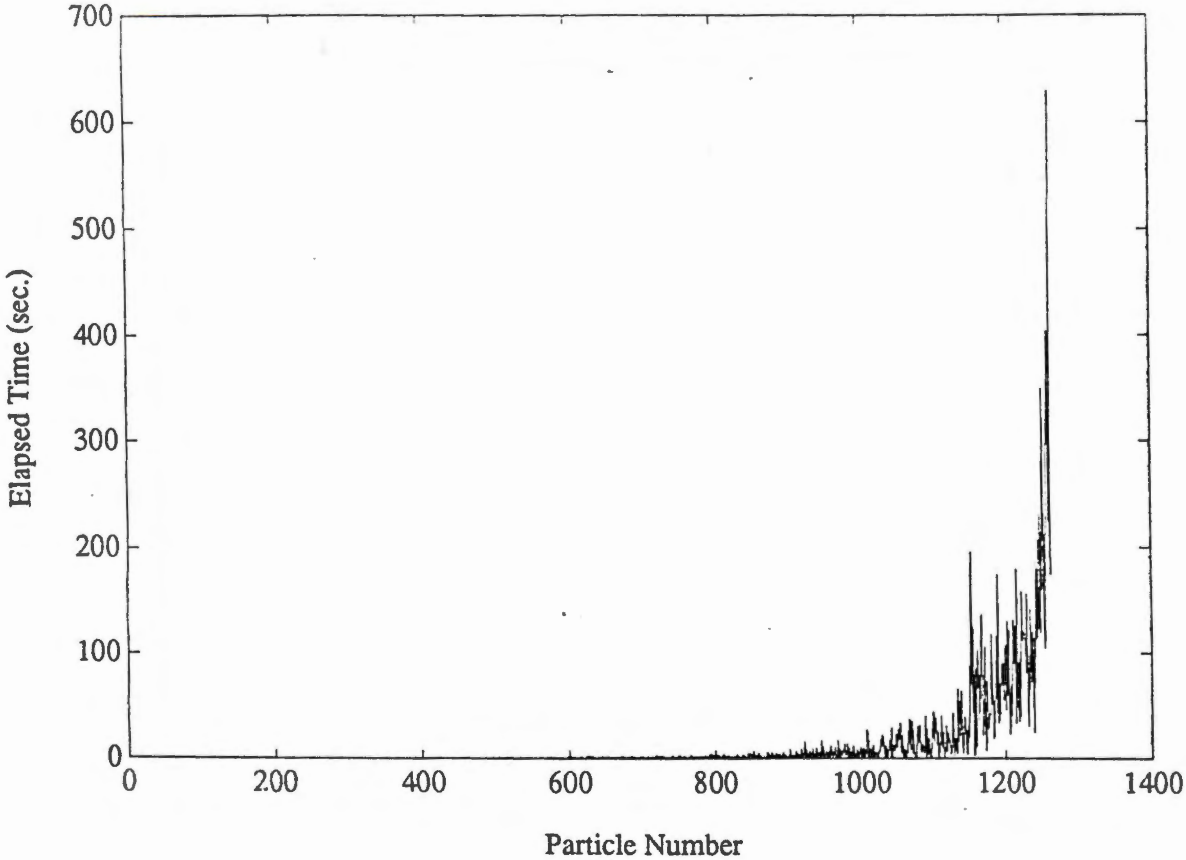

Figure 36. Elapsed time vs. particle number for the particle packing shown in Figure 35.

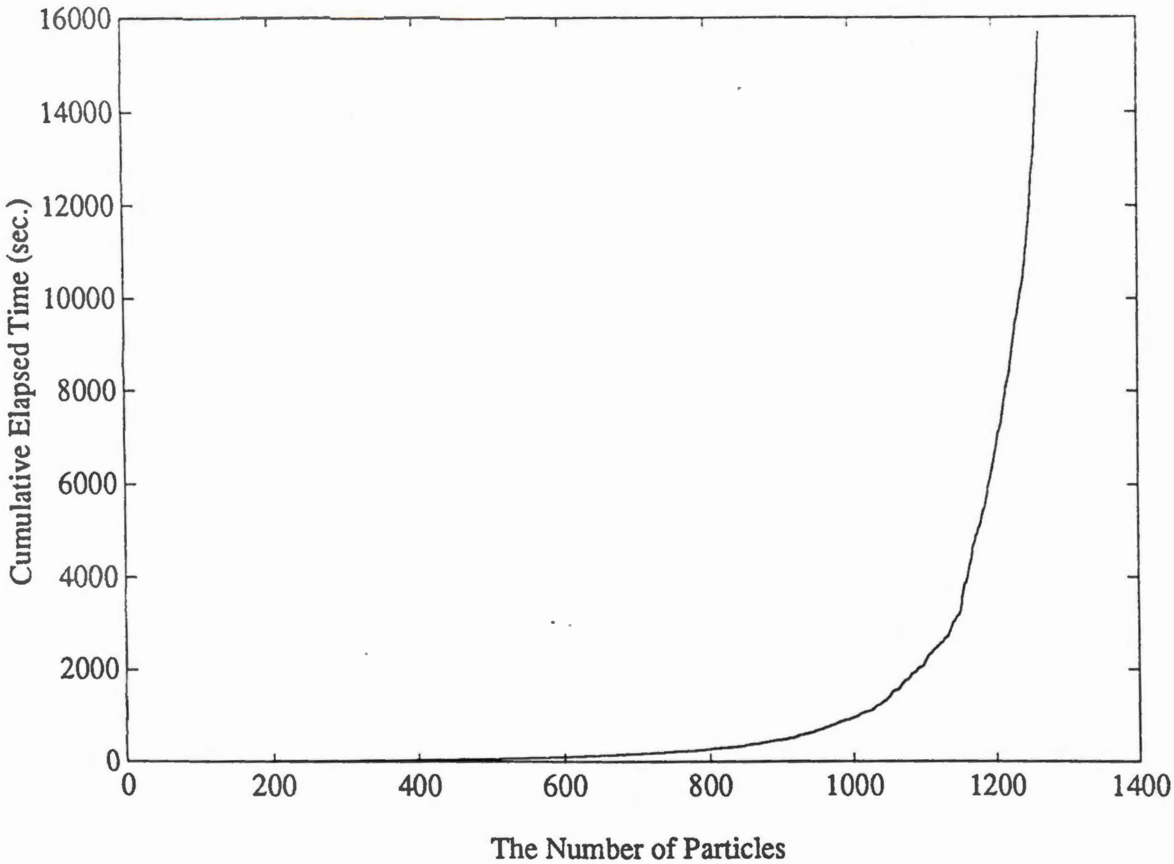

Figure 37. Cumulative elapsed time vs. the number of particle for the particle packing shown in Figure 35.

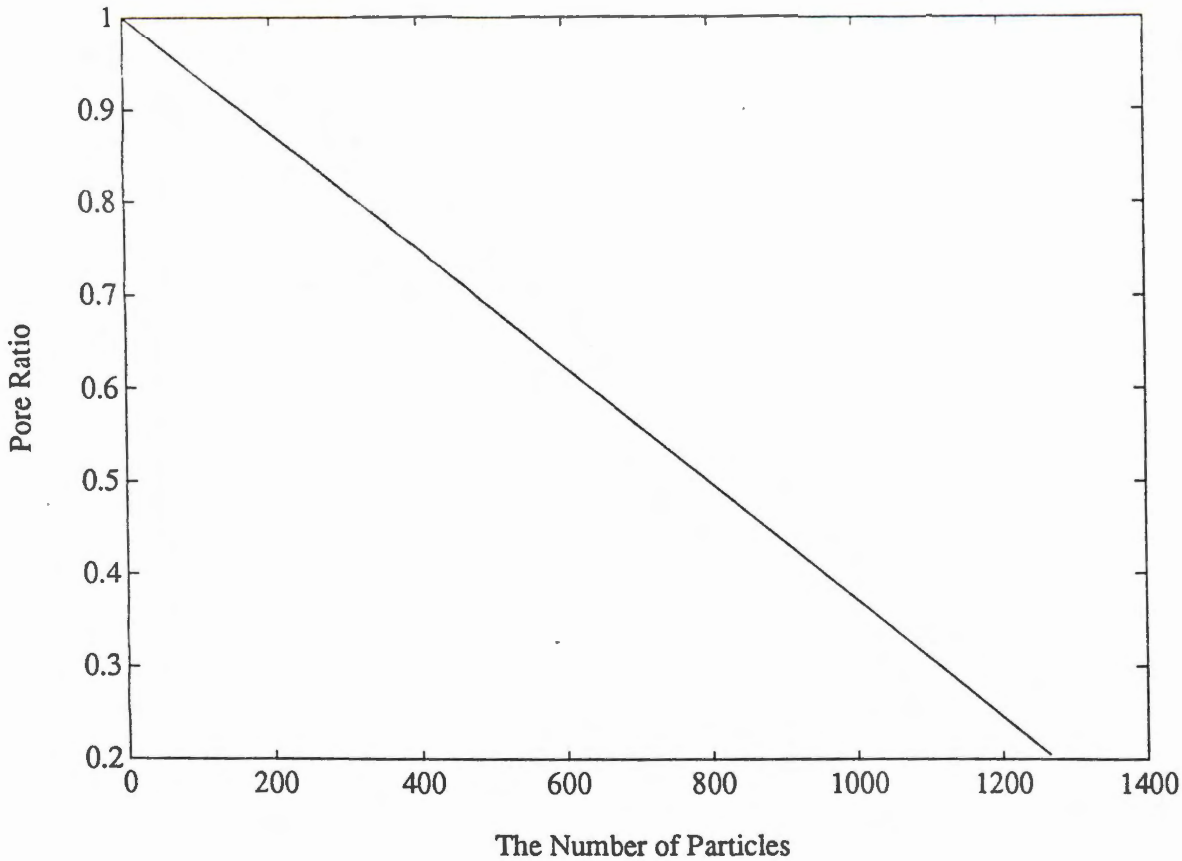

Figure 38. Porosity vs. the number of particle for the particle packing shown in Figure 35.

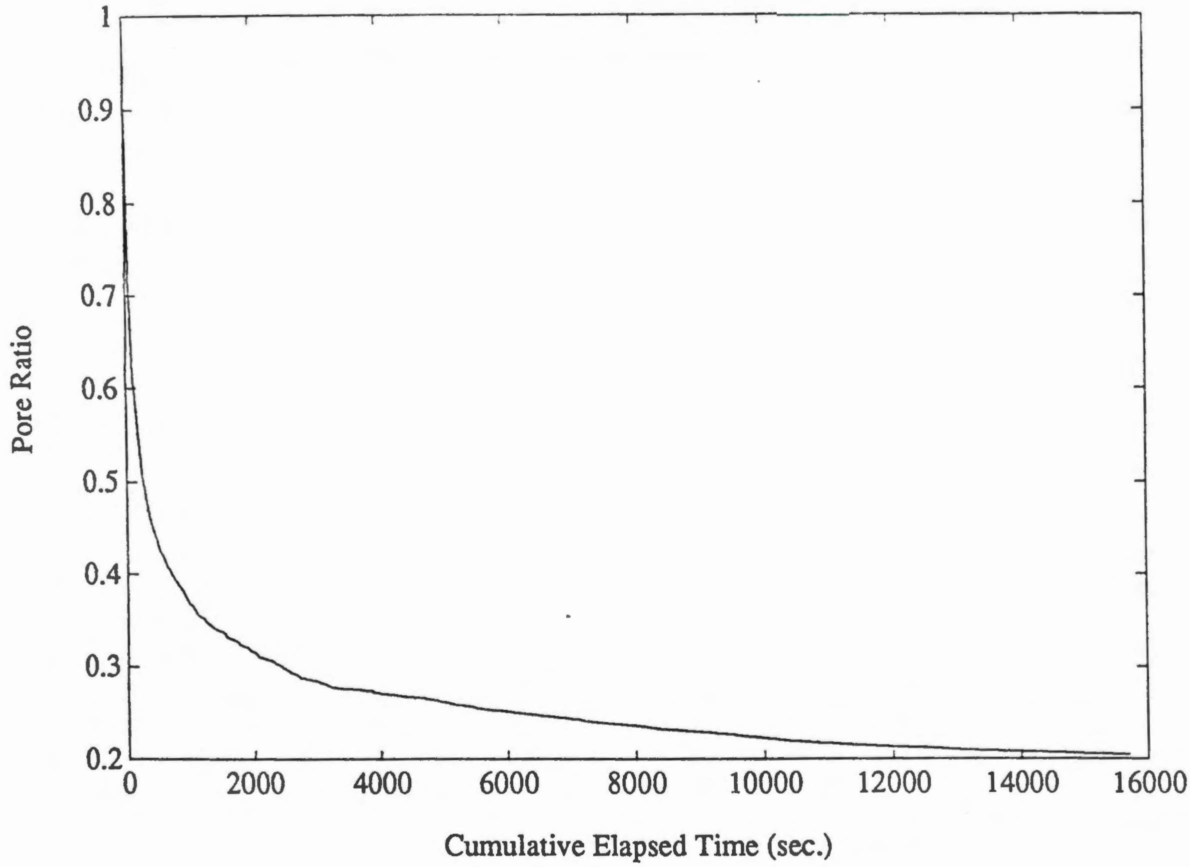

Figure 39. Porosity vs. cumulative elapsed time for the particle packing shown in Figure 35.

All the behaviors shown in the figures are self-explanatory. The elapsed time used for generating individual particles varies from one particle to another (Figures 36 and 31). Those particles generated early generally require much less computer time than those generated later. When the packing approaches the densest state, the time required to insert an individual particle will increase dramatically. This signal can be used to automatically terminate the packing procedure. Likewise, the computed cumulative elapsed time (summed over all pre-generated particles) can be used in a similar manner (Figures 37 and 32). Porosity decreases linearly with increases in the number of particles (Figures 38 and 33). The linearity relies on the fixed volume of the container and size uniformity of all particles in this special case. If the particle size is not uniform, the curve will randomly oscillate around a straight line similar to the line in Figures 38 and 33. The amplitude of oscillation will depend on the difference between r_{max} and r_{min}. The slope of the straight line will depend on the mean radius of particles. Porosity declines sharply at first and then more slowly as cumulative elapsed time increases (Figures 39 and 34). When packing approaches the densest state, the curve will become very flat. Using this property, the simulation process can be terminated at a very early stage without significantly reducing accuracy of the predicted results.

3. DYNAMIC RESPONSE OF SOIL TO LOADING CAUSED BY A PENETRATING OBJECT

The basic structure of the Monte Carlo simulation offers a bridge between localized and global phenomena. A potential use of the simulation is in the prediction of dynamic soil response to loading caused by a penetrating object. Consider the following scheme. First, divide the total displacement of a soil-contacting tool, such as the punch in Figures 8 and 9, into very small increments. The displacement field of the soil particles can be simulated for each of the increments by using the scheme of sequentially translating overlaps. Based upon the principle of virtual work, the sum of the kinetic energies of the individual particles should equal an unknown resistance force multiplied by the tool displacement increment. The magnitude of this resistance force caused by kinetic energy in the soil system can be determined from the relationship. If the kinetic energy of the punch is gradually reduced by the kinetic energy of the soil mass computed at each stage of punch movement, penetration depth vs. penetration time and many other useful relationships can be reasonably approached.

The following example considers simulated particle packing with a uniform size distribution of $r_{min} = 1$, $r_{max} = 3$, porosity n = 0.22 and mass density of individual particles = 1 (see Figure 40). Suppose a punch with width/2 = 20, mass = 200, and initial speed = 2.0×10^4 (1/sec) penetrates the mass of

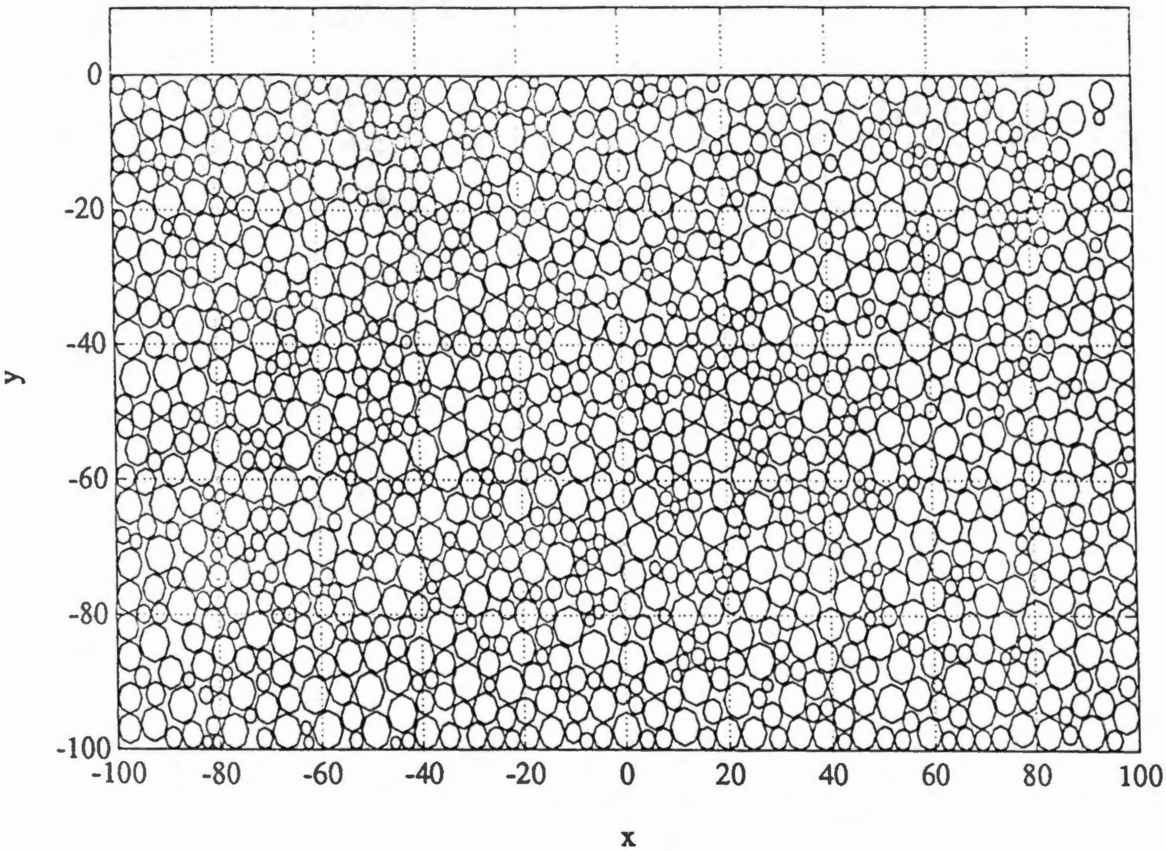

Figure 40. Simulated "soil" particle packing with a uniform probability particle size distribution by number (porosity = 0.22, r_{min} = 1, r_{max} = 3) prior to dynamical penetration by a rigid punch with width/2 = 20.

simulated particles. If a time increment = 1.0×10^{-4} sec. is used, then the total displacement field of particles can be shown in Figure 41. Particle movement for each time increment is shown in Figures 42 through 52.

One point worthy of note is the fact that some displacement vectors of individual particles cross each other (Figure 41). A physical interpretation suggests that different particles pass a given position; such events are not allowed in a continuous model. However, this phenomenon can be actually observed in real granular materials, because particles can pass the same position at <u>different times</u>, or in <u>different stages of displacement</u>. Thus, the simulation results showing crossing of displacement vectors is a realistic reflection of the physical world. Further, for the exact same increments of time, the increments of punch depth computed are gradually reduced (Figures 42 through 52), due to the fact that the speed of punch is gradually reduced as is the kinetic energy saved in the punch. Displacements of soil particles, likewise, tend to be correspondingly reduced. Particle behavior becomes increasingly sophisticated, because the displacements of particles will continuously influence the construction of the particle packing. This construction change, in turn, will influence the further displacement of particles. Such phenomena may be very difficult, if not impossible, to describe in a continues model.

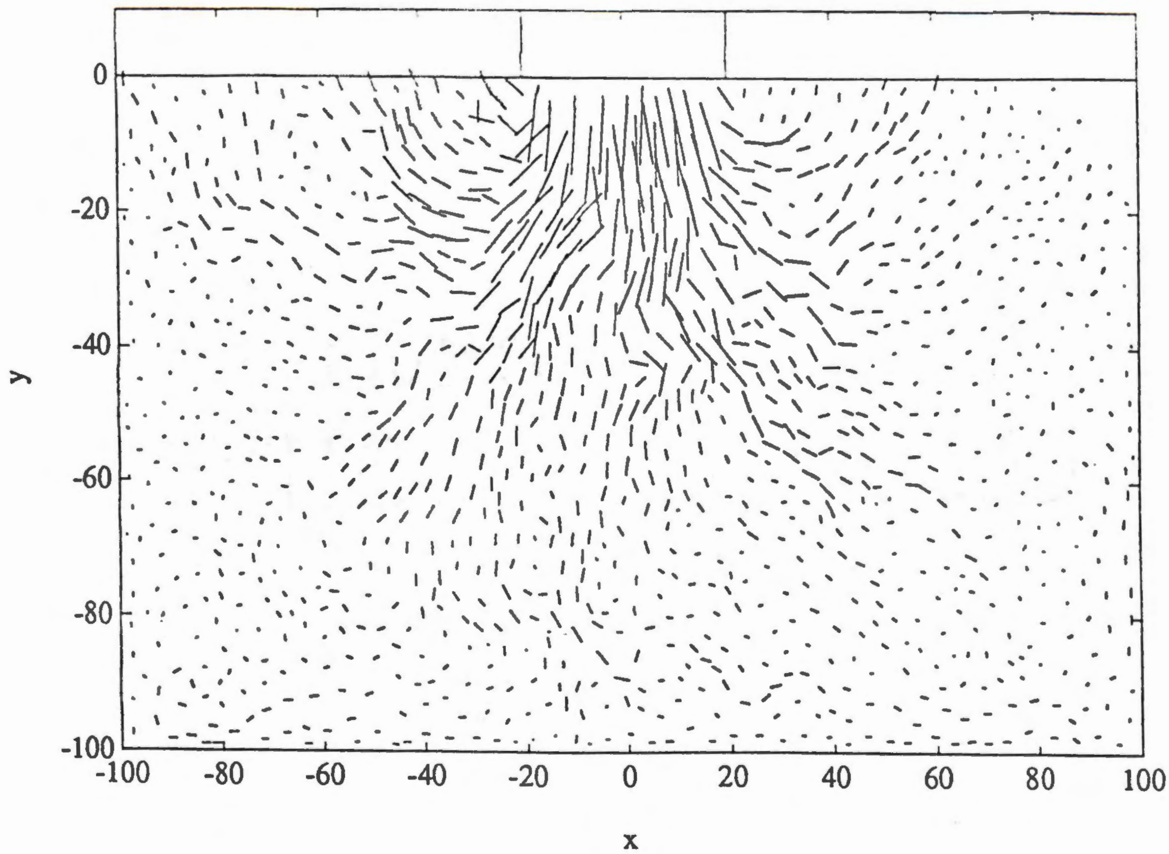

Figure 41. The simulated displacement field for the punch-"soil" system shown in Figure 40. Initial punch speed = 2.0×10^4 (1/sec), punch mass = 2.0×10^3, and total vertical displacement of the punch = 11.57.

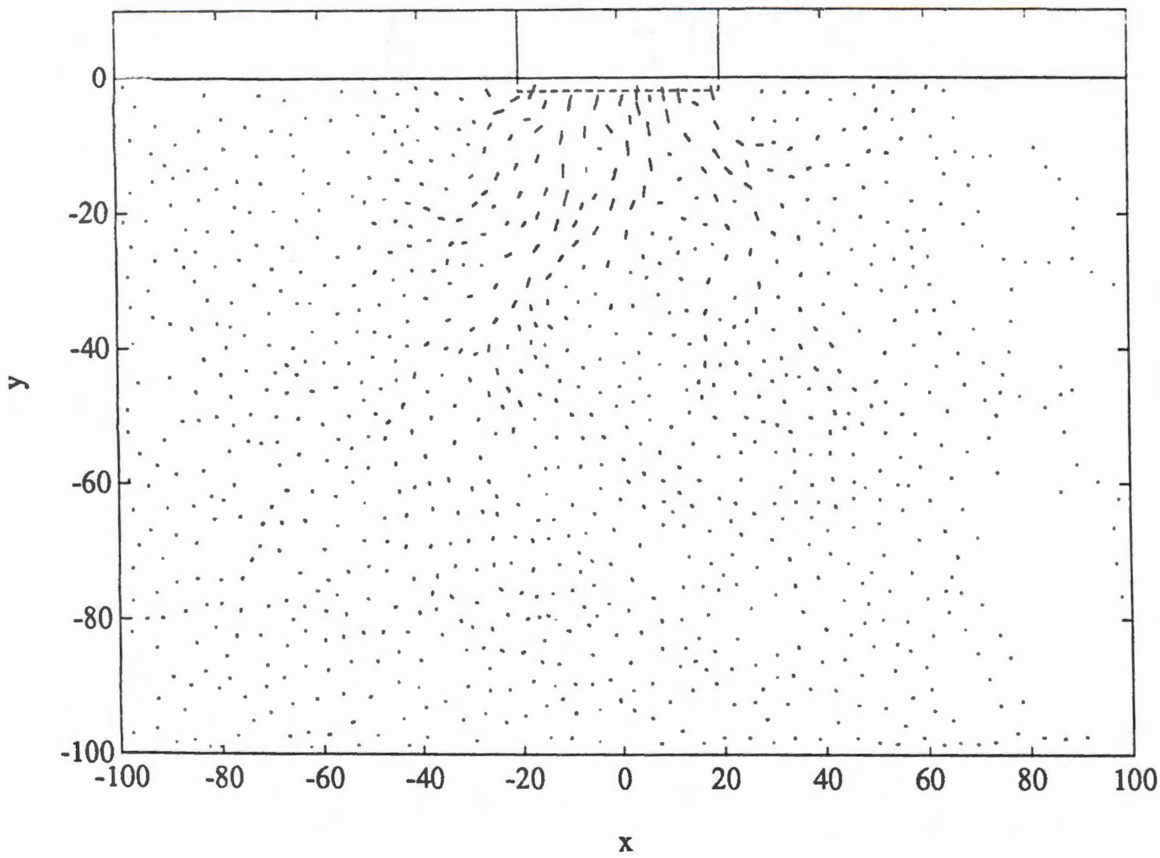

Figure 42. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 0, punch depth = 0, and depth increment = 2).

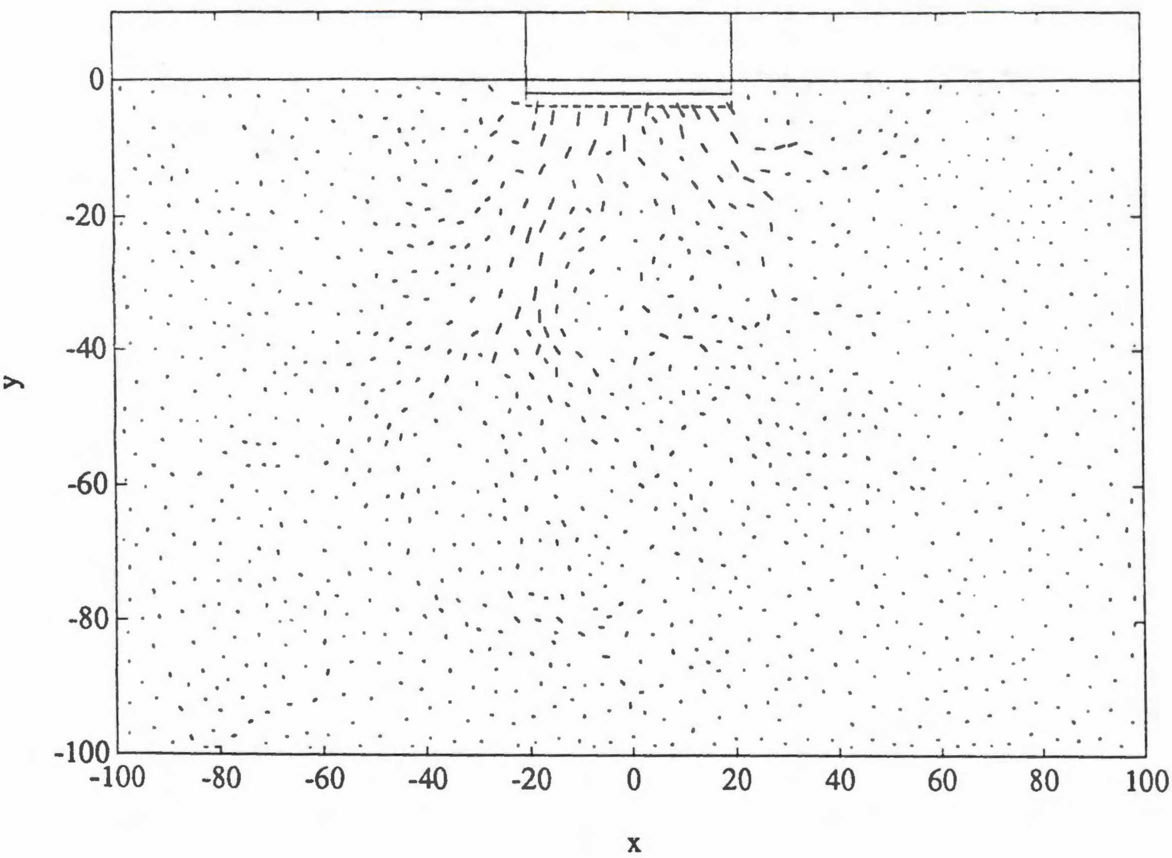

Figure 43. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 1.0×10^{-4} sec, punch depth = 2, and depth increment = 1.864).

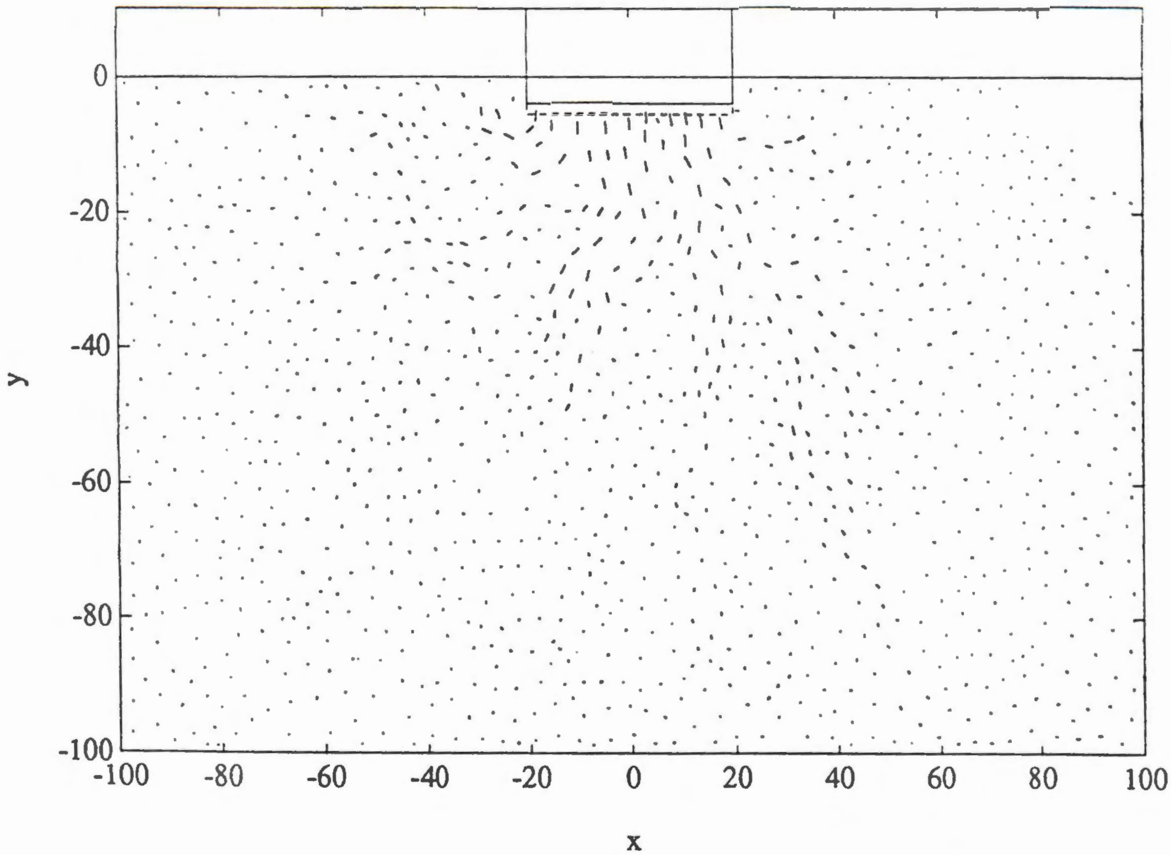

Figure 44. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 2.0 x 10^{-4} sec, punch depth = 3.864 and depth increment = 1.621).

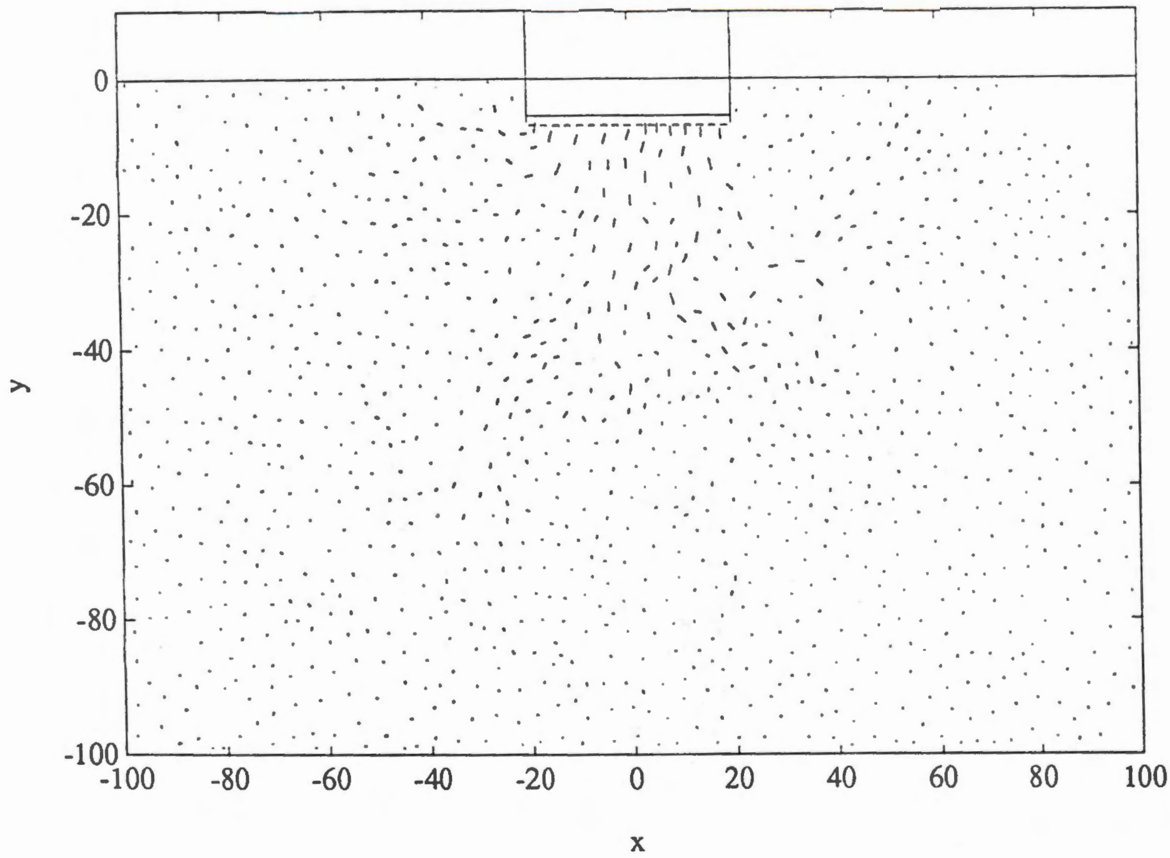

Figure 45. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 3.0×10^{-4} sec, punch depth = 5.485, and depth increment = 1.427).

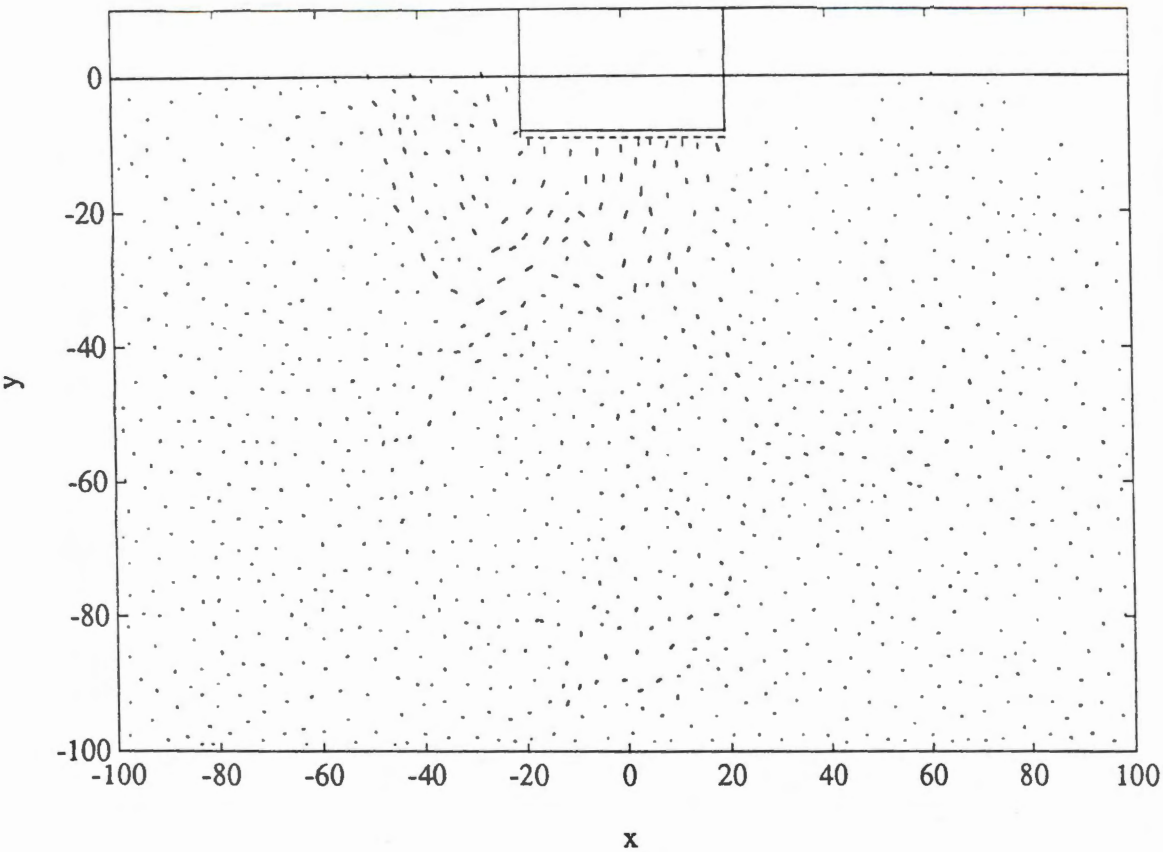

Figure 46. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 5.0 x 10^{-4} sec, punch depth = 8.132, and depth increment = 0.951).

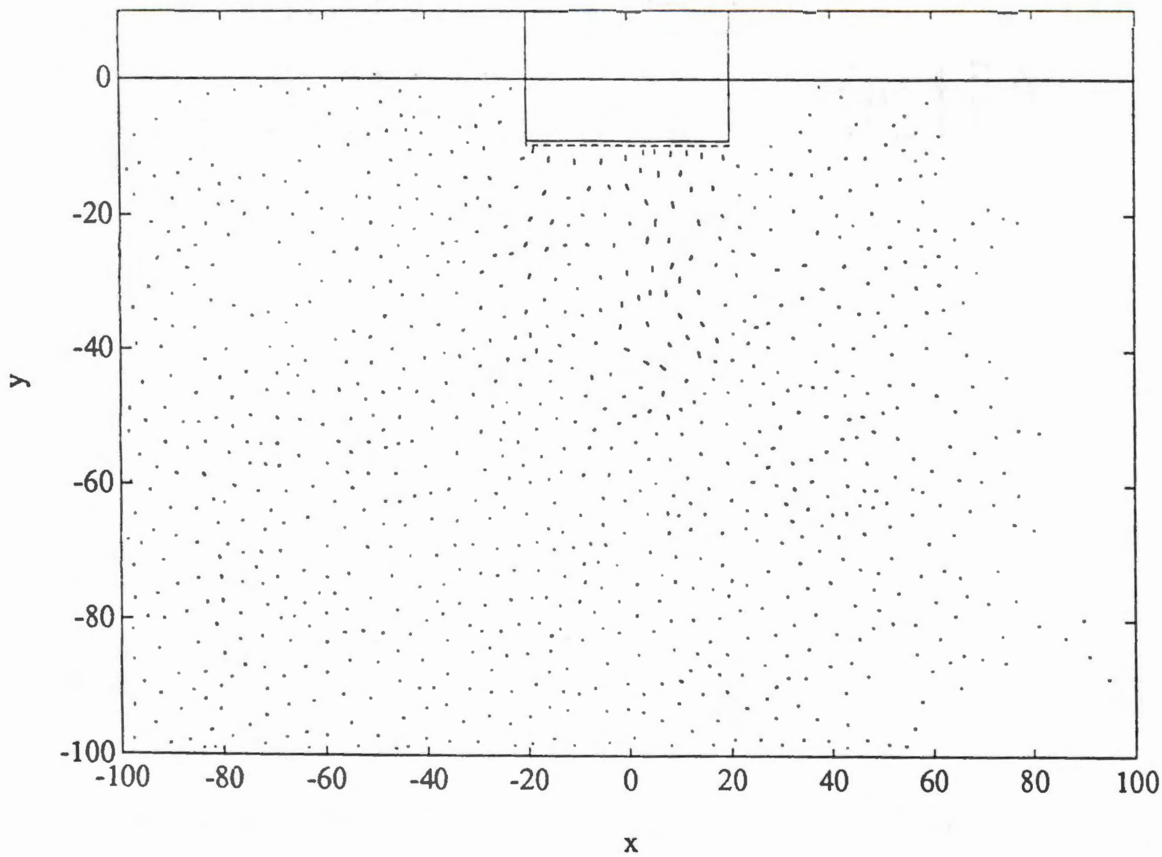

Figure 47. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 6.0 x 10^{-4} sec, punch depth = 9.083, and depth increment = 0.7421).

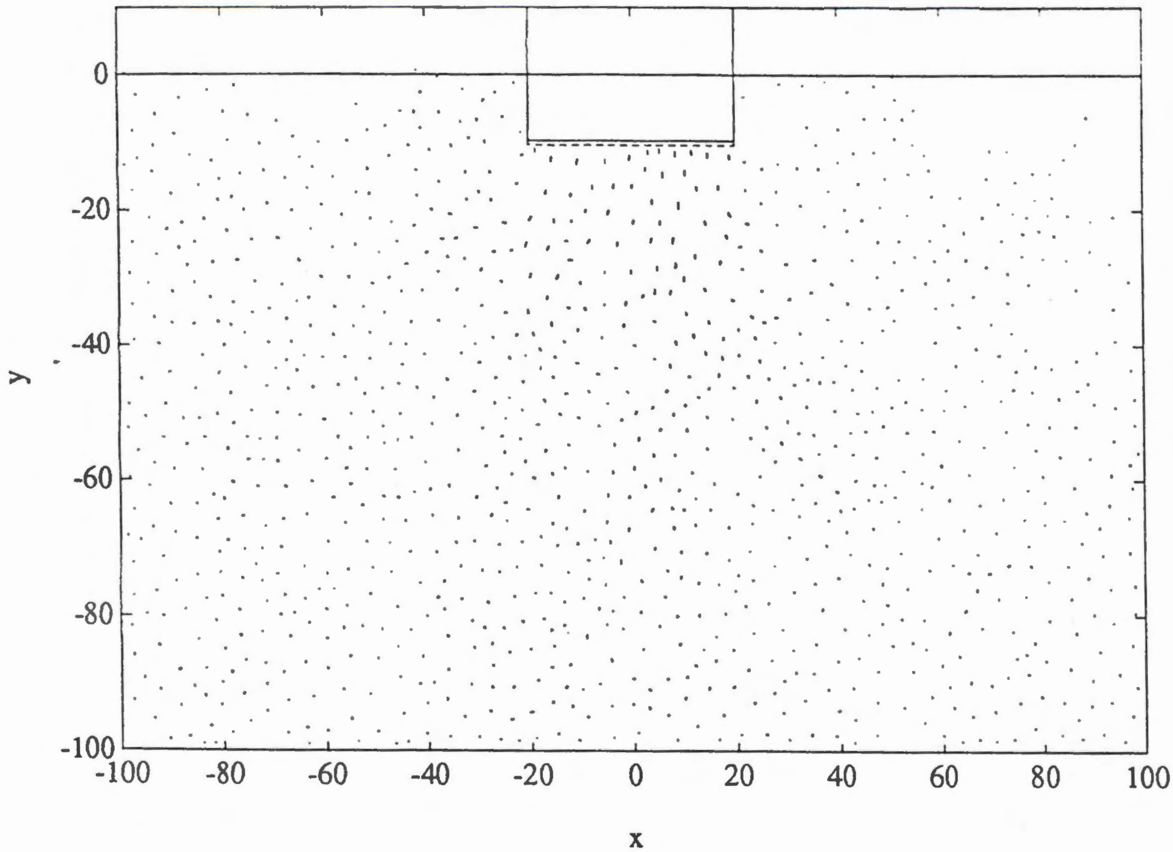

Figure 48. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 7.0 x 10^{-4} sec, punch depth = 9.825, and depth increment = 0.621).

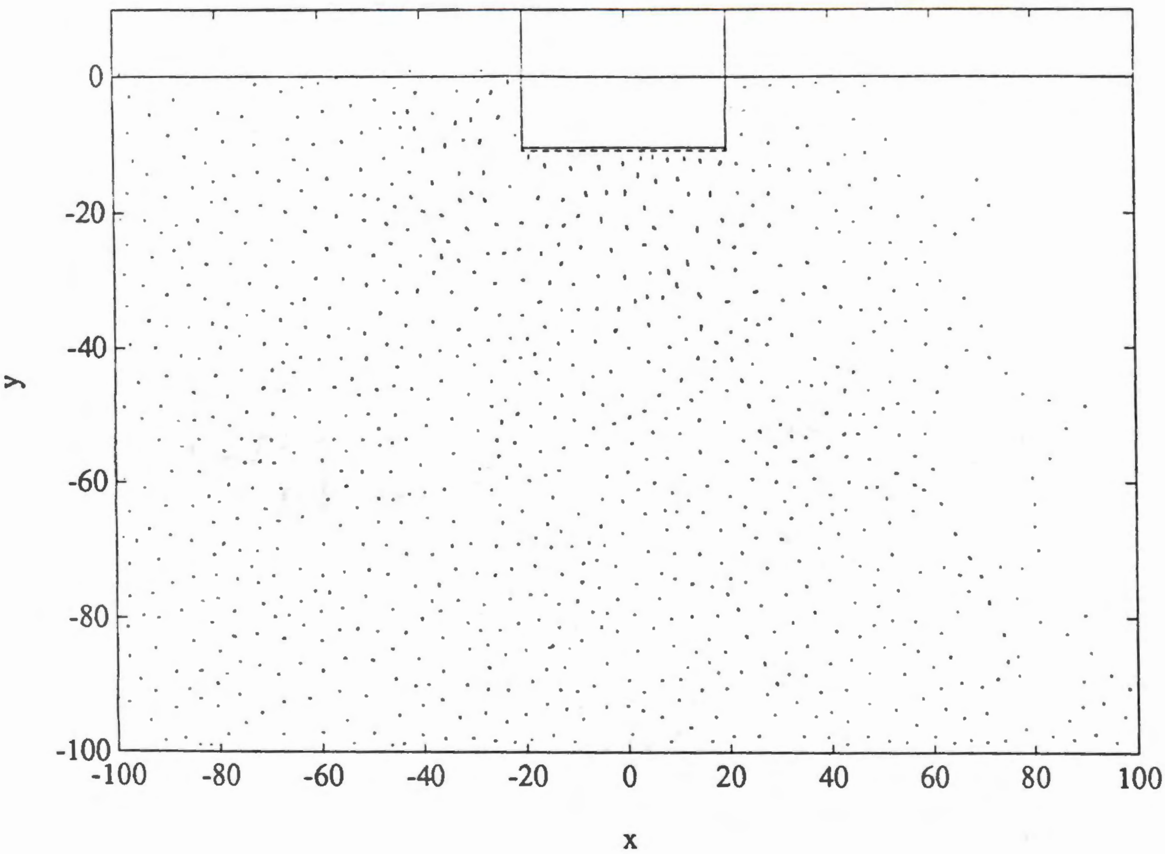

Figure 49. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 8.0×10^{-4} sec, punch depth = 10.45, and depth increment = 0.4935).

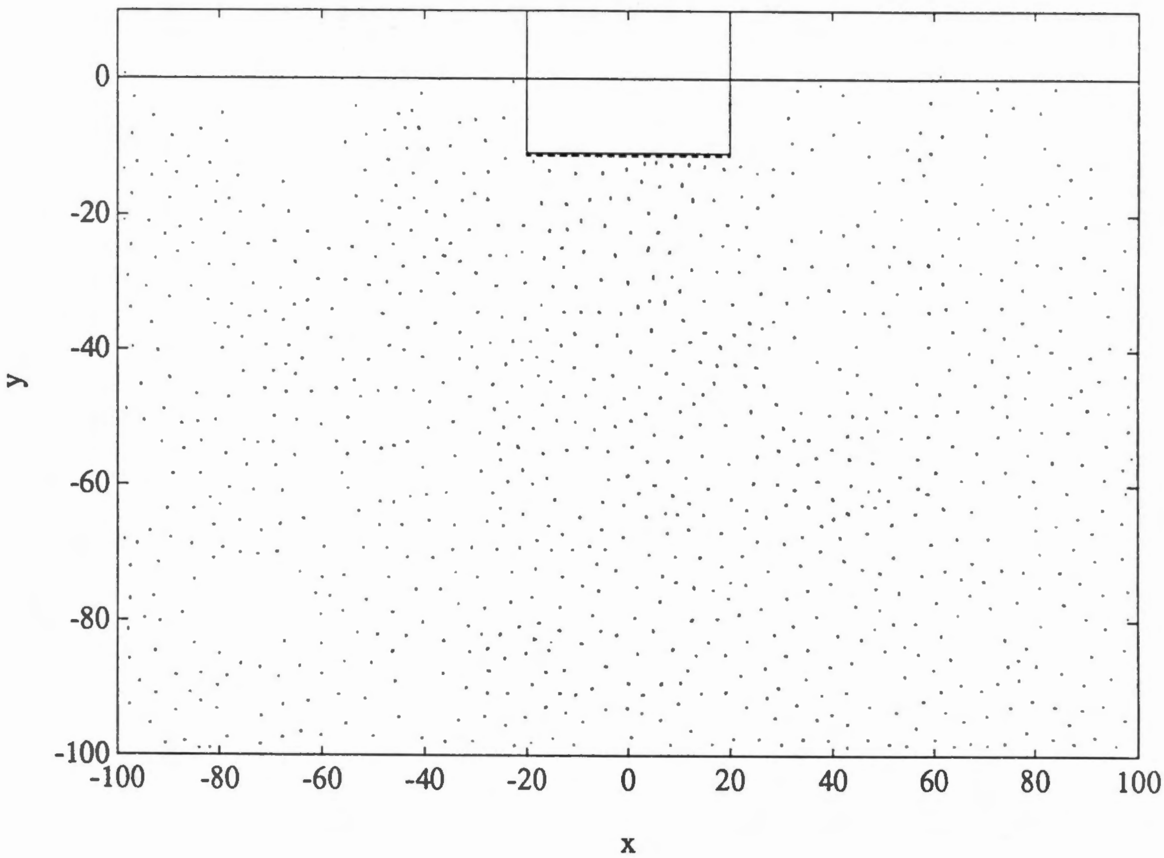

Figure 50. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 9.0×10^{-4} sec, punch depth = 10.94, and depth increment = 0.328).

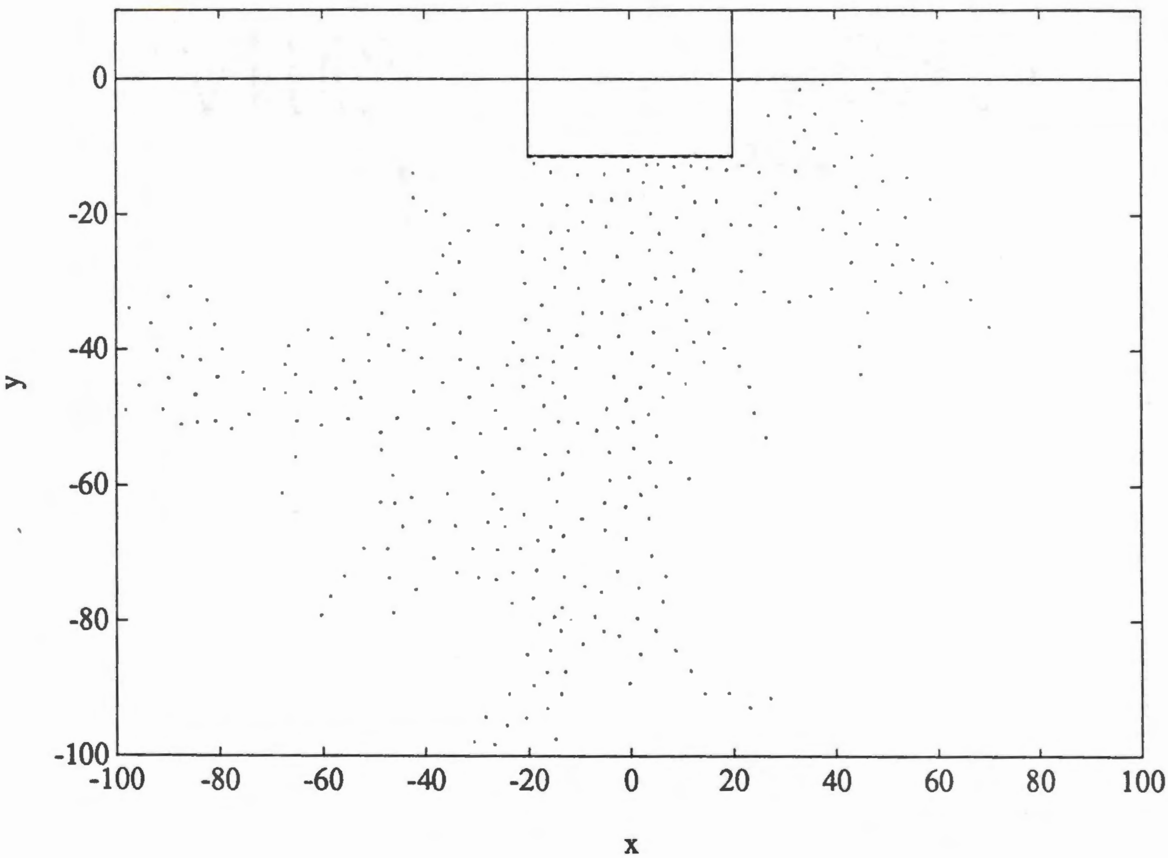

Figure 51. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 1.0 x 10^{-3} sec, punch depth = 11.27, and depth increment = 0.1839).

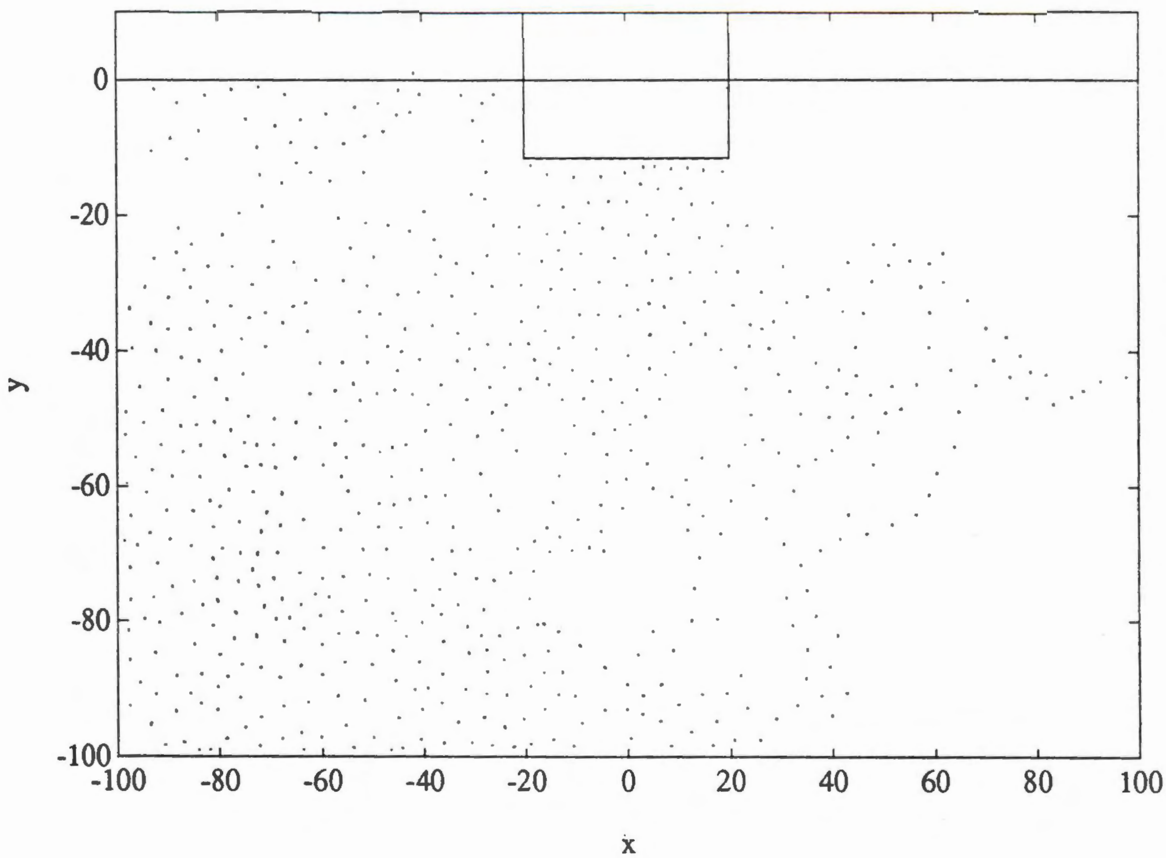

Figure 52. Incremental particle displacement for the punch-"soil" system shown in Figure 40 (time = 1.1×10^{-3} sec, punch depth = 11.45, and depth increment = 0.1227).

CHAPTER VI

CONCLUSIONS

The Monte Carlo simulation is an attractive method of approaching problems involving materials with granular and random structures. A precise knowledge of the positions of all individual particles is invaluable for subsequent analyses and applications (Tory et al., 1973). The simulation has many potential applications, not only in soil engineering, but in crystallography, biology, nuclear engineering, and petroleum engineering (Jodrey and Tory, 1979).

Limitations to use of the Monte Carlo simulation are apparent. Data describing the particle size frequency distributions or particle cumulative size distributions for various soils may not be readily available or easily attainable. Further, particle movement may not necessarily be along lines connecting centroids of adjacent particles or normal to penetrating tools at contact points, especially when the effects of friction cannot be ignored. Experimental verification of the simulation will be necessary before use of the procedure can be broadened. However, in simple laboratory experiments using rods to simulate loosely- and densely-packed soil particles, particle displacement fields observed when the soil mass was penetrated by a punch were similar to results predicted by the Monte Carlo simulation.

BIBLIOGRAPHY

BIBLIOGRAPHY

Adams, D. J. and A. J. Matheson. 1972. Computation of dense random packing of hard spheres. The Journal of Chemical Physics 56:1989-1994.

Barker, J. A. and M. R. Hoare. 1975. Relaxation of the Bernal model. Nature 257:120-122.

Bennett, C. H. 1972. Serially deposited amorphous aggregates of hard spheres. The Journal of Applied Physics 43:2727-2734.

Beresford, R. H. 1969. Statistical geometry of random heaps of equal hard spheres. Nature 224:550-553.

Bernal, J. D. 1959. A geometrical approach to the structure of liquids. Nature 183:141-147.

Bernal, J. D. 1960. Geometry of the structure of monatomic liquids. Nature 185:68-70.

Bernal, J. D. and J. L. Finney. 1967. Random packing of spheres in non-rigid containers. Nature 214:265-266.

Bernal, J. D. and J. Mason. 1960. Co-ordination of randomly packed spheres. Nature 188:910-911.

Bourdeau, P. L. and M. E. Harr. 1989. Stochastic theory of settlement of loose cohesionless soils. Geotechnique 39(4):641-654.

Chang, C. S., A. Misra and J. H. Xue. 1989. Incremental stress strain relationships for regular packings made of multi-sized particles. International Journal of Solids Structures 25(6):665-681.

Chang, C. S. and A. Misra. 1989. Computer simulation and modelling of mechanical properties of particulates. Computers and Geotechnics 7:269-287.

Chang, C. S. and A. Misra. 1991. Stress-strain modeling of heterogeneous granular solids based on micro-mechanics. In (C. S. Desai et al. ed.): Constitutive Laws for Engineering Materials. pp501-504. Proceedings of the Third International Conference on Constitutive Laws for Engineering

Materials: Theory and Applications, Tucson, Arizona, USA, January 7-12, 1991. ASME press, New York.

Chang, C. S. and C. L. Liao. 1990. Constitutive relation for a particulate medium with the effect of particle rotation. International Journal of Solids Structures 26(4):437-453.

Cipra, B. 1990. Packing your n-dimensional marbles. Science 247:1035.

Cipra, B. 1991. Music of the spheres. Science 251:1028.

Cundall, P. A. and O. D. L. Strack. 1979(a). The distinct element method as a tool for research in granular media, Part II. Report to the national science foundation concerning NSF Grant Eng76-20711.

Cundall, P. A. and O. D. L. Strack. 1979(b). A discrete numerical model for granular assemblies. Geotechnique 29(1):47-65.

Das, B. M. 1983. Advanced Soil Mechanics. Hermisphere Publishing Corporation, Washington. McGraw-Hill Book Company, New York.

Dobry, R., T-T. Ng., E. Petrakis and A. Seridi. 1991. General model for contact law between two rough spheres. Journal of Engineering Mechanics 117(6):1365-1381.

Eldin, A. M. Z., K. C. Watts, S. M. Younis and M. Shaibon. 1990(a). Development of a two dimensional finite element model for soil-tool interaction. Paper No. 90-1544, American Society of Agricultural Engineers, St. Joseph, Michigan.

Eldin, A. M. Z., K. C. Watts, A. P. Onwualu, S. M. Younis and M. Shaibon. 1990(b). Experimental evaluation of a two dimensional finite element, soil-tool interaction model. Paper No. 90-1545, American Society of Agricultural Engineers, St. Joseph, Michigan.

Finney, J. L. 1977. Modelling the structures of amorphous metals and alloys. Nature 266:309-314.

Freitag, D. R., R. L. Schafer and R. D. Wismer. 1970. Similitude of soil-machine systems. Journal of Terramechanics. 7(2):25-59.

Furukawa, K. 1959. A structural model for monatomic liquids including metallic liquids. Nature 184:1209-1210.

Gotoh, K. and J. L. Finney. 1974. Statistical geometrical approach to random packing density of equal spheres. Nature 252:202-205.

Gotoh, K., W. S. Jodrey and E. M. Tory. 1978. A random packing structure of equal spheres - statistical geometrical analysis of tetrahedral configurations. Powder Technology 20:233-242.

Hardin, B. O. 1989. Low-stress dilation test. Journal of Geotechnical Engineering 115(6):769-787.

Hardin, B. O. and G. E. Blandford. 1989. Elasticity of particulate materials. Journal of Geotechnical Engineering 115(6):788-805.

Jenkins, J. T. and M. Satake. 1983. Mechanics of Granular Materials: New Models and Constitutive Relations. Proceedings of the U.S. / Japan Seminar on New Models and Constitutive Relations in the Mechanics of Granular Materials, Ithaca, New York, August 23-27, 1982. Elsevier Science Publishing Company Inc., Amsterdam.

Jodrey, W. S. and E. M. Tory. 1979. Simulation of random packing of spheres. Simulation 32:1-12.

Jodrey, W. S. and E. M. Tory. 1980. Random sequential packing in R^n. Journal of Statistical Computation and Simulation 10:87-93.

Jodrey, W. S. and E. M. Tory. 1981. Computer simulation of isotropic, homogeneous, dense random packing of equal spheres. Powder Technology 30:111-118.

Johnson, M. E. 1987. Multivariate Statistical Simulation. John Wiley & Sons, New York. pp. 18-29.

Jong, G. D. J. D. 1971. The double sliding, free rotating model for granular assemblies. Geotechnique 21:155-163.

Jullien, R. and P. Meakin. 1990. A mechanism for particle size segregation in three dimensions. Nature 344:425.

Jumikis, A. R. 1962. Soil Mechanics. Van Nostrand Reinhold Company, New York.

Jumikis, A. R. 1969. Theoretical Soil Mechanics. Van Nostrand Reinhold Company, New York.

Keen, D. A. and R. L. McGreevy. 1990. Structural modelling of glasses using reverse Monte Carlo simulation. Nature 344:423-425.

Klement, W. J., R. H. Willens and P. Duwez. 1960. Non-crystalline structure in solidified gold-silicon alloys. Nature 187:869-870.

Koolen, A. J. and H. Kuipers. 1983. Agricultural Soil Mechanics. Springer-Verlag, Berlin.

Lambe, T. W. and R. V. Whitman. 1979. Soil Mechanics, SI Version. John Wiley & Sons, New York.

Lambe, T. W. 1951. Soil Testing for Engineering. John Wiley & Sons, New York.

Mangelsdorf, P. C. J. and E. L. Washington. 1960. Packing of mixtures of hard spheres. Nature 187:930-931.

Mason, G. 1968. Radial distribution functions from small packings of spheres. Nature 217:733-735.

Matheson, A. J. 1974. Computation of a random packing of hard spheres. The Journal of Physics C: Solid State Physics 7:2569-2576.

Matsuoka, H. 1974. A microscopic study on shear mechanism of granular materials. Soils and Foundations 14(1):29-43.

Nemat-Nasser, S. 1980. The behavior of granular materials in simple shear. Soils and Foundations 20(3):59-73.

Pollock, J. D., J. V. Perumpral and T. Kuppusamy. 1986. Finite element analysis of multipass effects of vehicles on soil compaction. Transactions of the ASAE 29(1):45-50.

Powell, M. J. 1980. Computer-simulated random packing of spheres. Powder Technology 25:45-52.

Raper, R. L. and D. C. Erbach. 1990(a). Prediction of soil stresses using the finite element method. Transactions of the ASAE 33(3):725-730.

Raper, R. L. and D. C. Erbach. 1990(b). Effect of variable linear elastic parameters on finite element prediction of soil compaction. Transactions of the ASAE 33(3):731-736.

Rothenburg, L. and R. J. Bathurst. 1989. Analytical study of induced anisotropy in idealized granular materials. Geotechnique 39(4):601-614.

Rutgers, R. 1962. Packing of spheres. Nature 193:465-466.

Sadd, M. H., A. Shukla and H. Mei. 1989. Computational and experimental modeling of wave propagation in granular materials. Proc. of the 4th International Conference on Computational Methods and Experimental Measurements, pp.325-334, Capri, Italy.

Sadd, M. H., A. Shukla, Q. M. Tai and Y. Xu. 1991. Micromechanical constitutive behavior of granular media under dynamic loading conditions. In (C. S. Desai et al. ed.): Constitutive Laws for Engineering Materials. pp.539-542. Proceedings of the Third International Conference on Constitutive Laws for Engineering Materials: Theory and Applications, Tusson, Arizona, USA, January 7-12, 1991. ASME press, New York.

Satake, M. 1985. Graph-theoretical approach to the mechanics of granular materials. In (A. J. M. Spencer ed.): Continuum Models of Discrete Systems: Proceedings of fifth international symposium on continuum models of discrete systems /Nottingham/ 14-20 July 1985. AA. Balkemo Publishers, Netherlands.

Scott, G. D. 1960. Packing of spheres. Nature 188:908-909.

Scott, G. D. 1962. Radial distribution of the random close packing of equal spheres. Nature 194:956-957.

Scott, G. D. and D. L. Mader. 1964. Angular distribution of random close-packed equal spheres. Nature 201:382-383.

Scott, R. F. 1987. Failure. Geotechnique 37(4):423-466.

Shahinpoor, M. 1983. Advances in the Mechanics and the Flow of Granular Materials. Volume I. Gulf Publishing Company, Houston.

Strack, O. D. L. and P. A. Cundall. 1978. The distinct element method as a tool for research in granular media, Part I. Report to the national science foundation concerning NSF Grant Eng76-20711.

Susskind, H. and W. Becker. 1966. Random packing of spheres in non-rigid containers. Nature 211:1564-1565.

The Math Works, Inc. 1989. PRO-MATLAB™ User's Guide.

The Molecular Physics Correspondent of Nature. 1972. What is random packing? Nature 239:488-489.

Tobita, Y. 1989. Fabric tensors in constitutive equations for granular materials. Soils and Foundations 29(4):91-104.

Tokue, T. 1979. Deformation behaviours of dry sand under cyclic loading and a stress-dilatancy model. Soils and Foundations 19(2):63-78.

Tory, E. M., B. H. Church, M. K. Tam, and M. Ratner. 1973. Simulated random packing of equal spheres. The Canadian Journal of Chemical Engineering 51:484-493.

Tory, E. M., N. A. Cochrane, and S. R. Waddell. 1968. Anisotropy in simulated random packing of equal spheres. Nature 220:1023-1024.

Ueng, T. S. and C. J. Lee. 1990. Deformation behavior of sand under shear-particulate approach. Journal of Geotechnical Engineering 116(11):1625-1640.

Vesic, A. S. 1963. Bearing capacity of deep foundations in sand. Highway Research Board Record, No. 39.

Vesic, A. S. 1967(a). A study of bearing capacity of deep foundations. Georgia Inst. of Technology, Atlanta (March).

Vesic, A. S. 1967(b). Ultimate loads and settlements of deep foundations in sand. Proc. Symposium on Bearing Capacity and Settlement of Foundations, Duke Univ., Durham. N.C.

Vinogradov, O. G., and P. Wierzba. 1986. Simulation of motion of cluster of ice floes. In: D. O. Koval (ed.). Proceedings of the IASTED International Conference: Applied Simulation and Modelling, ASM'86, British Columbia, Canada. pp. 408-411.

Vinogradov, O. G. 1987. Simulation methodology for a flow of interactive ice floes around an obstacle. International Journal of Modelling & Simulation 7:28-31.

Vinogradov, O. G. and A. Springer. 1990. Simulation of motion of multi-body system with interactions. In: The Proceedings of the 1990 summer computer simulation conference. pp. 51-56.

Visscher, W. M. and M. Bolsterli. 1972. Random packing of equal and unequal spheres in two and three dimensions. Nature 27:504-507.

Wu, T. S. 1966. Soil Mechanics. Allyn and Bacon, Inc., Boston.

Yerazunis, S., S. W. Cornell and B. Wintner. 1965. Dense random packing of binary mixtures of spheres. Nature 207:835-837.

APPENDICES

APPENDIX A

The Main Program of MCSSP (Monte Carlo Simulation of Soil Particle Packing)

```
%                                    --- MAIN PROGRAM ---


%                         ****** PREPARE COMPUTER AND SCREEN ******

echo off
clear
clc
clg
axis;axis;
rand('seed',sum(clock*100000))
rand('uniform')
casesen on

disp('                         M   M    CCCC    SSSS    SSSS   PPPP      ')
disp('                         MM MM   C       S       S       P   P     ')
disp('                         M M M   C       S       S       P   P     ')
disp('                         M M M   C        SSS     SSS    PPPP      ')
disp('                         M   M   C           S       S   P         ')
disp('                         M   M   C           S       S   P         ')
disp('                         M   M    CCCC   SSSS    SSSS    P         ')
disp('                                                                   ')
disp('                   Monte Carlo Simulation of Soil Particle Packing ')
disp('                                                                   ')
disp('                                                                   ')
disp('                                    Developed By                   ')
disp('                                                                   ')
disp('                                     Yuehui Fan                    ')
disp('                                                                   ')
disp('                         Agriculture Engineering Department        ')
disp('                       The University of Tennessee, Knoxville      ')
disp('                                                                   ')
disp('                                      Oct. 1990                    ')
disp('                             (strike any key to continue)'),pause

clc
flagsave=menu('Save input paramaters','yes','no')
if flagsave==1
   diary mcssp.in
else
   diary off
end
clc


disp('                             ****** INPUT PARAMATERS ****** ')

if flagsave==1
   flagprip=menu('Print input paramaters','yes','no')
```

```
end
flagprig=menu('Print the graphs generated by MCSSP','yes','no')
if flagprig==1
   ncopy=input('Enter the number of graghics copy required:     ')
end

flagdis=menu('Choose distribution type of soil particles',...
   'only one size',...
   'uniform size distribution',...
   'normal size distribution',...
   'arbitrary size distribution',...
   'arbitrary cumulative size distribution')
if flagdis==1
   rmean=input('Enter radius of soil particles:     ')
   rvar=0;
   rstd=0;
   rmax=0;
   rmin=0;
   fm=0;
   Rvec=0;
   Fvec=0;
   ndis=0;
   flagfit=0;
elseif flagdis==2
   rmax=input('Enter max radius of soil particles:     ')
   rmin=input('Enter min radius of soil particles:     ')
   rmean=(rmax+rmin)/2;
   rvar=(rmax-rmin)/2;
   rstd=0;
   fm=0;
   Rvec=0;
   Fvec=0;
   ndis=0;
   flagfit=0;
elseif flagdis==3
   rmean=input('Enter mean radius of soil particles:     ')
   rstd=input('Enter standard deviation of soil particles:     ')
   rvar=rstd;
   rmax=0;
   rmin=0;
   fm=0;
   Rvec=0;
   Fvec=0;
   ndis=0;
   flagfit=0;
elseif flagdis==4
   ndis=input('Enter number of radius-frequency pairs:     ')
   RF=zeros(ndis,2);
   for ind=1:ndis
       fprintf('\n\n  Enter %g#',ind)
       disp(' radius-frequency pair:')
```

```
      RF(ind,:)=input(' ==>     ');
   end
   RF
   Rvec=RF(:,1);
   Fvec=RF(:,2);
   fm=max(Fvec);
   rmean=(Rvec(ndis)+Rvec(1))/2;
   rmax=max(Rvec);
   rmin=min(Rvec);
   Rvar=0;
   rstd=0;
elseif flagdis==5
   ndis=input('Number of cumulative radius-frequency pairs:     ')
   RF=zeros(ndis,2);
   for ind=1:ndis
      fprintf('\n\n  Enter %g#',ind)
      disp(' cumulative radius-frequency pair:')
      RF(ind,:)=input(' ==>     ');
   end
   RF
   Rvec=RF(:,1);
   Fvec=diff(RF(:,2))./diff(Rvec);
   Fvec(ndis)=Fvec(ndis-1);
   fm=max(Fvec);
   rmean=(Rvec(ndis)+Rvec(1))/2;
   rmax=max(Rvec);
   rmin=min(Rvec);
   rvar=0;
   rstd=0;
   flagfp=menu...
      ('Plot cumulative soil size distribution',...
      'yes','no');
   if flagfp==1
      flagfit=menu...
      ('Type of curve fitting for cumulative soil size
distribution',...
         'piecewise linear interpolation',...
         'cubic spline fitting',...
         'polynomial fitting')
      flagpaper=menu('Choose types of graph paper',...
         'linear X-Y plot',...
         'semi-log X-Y plot');
      Rfit=rmin:(rmax-rmin)/100:rmax;
      if flagfit==1
         if flagpaper==1
            plot(Rvec,RF(:,2),'-')
         else
            semilogx(Rvec,RF(:,2),'-')
         end
         title(...
         'PICEWISE LINEAR INTERPOLATION FOR CUMUL SIZE DISTRIBUTION')
```

```
      elseif flagfit==2
         Ffit=spline(Rvec,RF(:,2),Rfit);
         if flagpaper==1
            plot(Rfit,Ffit,'-')
         else
            semilogx(Rfit,Ffit,'-')
         end
         title('CUBIC SPLINE FITTING FOR CUMUL SIZE DISTRIBUTION')
      elseif flagfit==3
         Cp=polyfit(Rvec,RF(:,2),ndis-1);
         Ffit=polyval(Cp,Rfit);
         if flagpaper==1
            plot(Rfit,Ffit,'-')
         else
            semilogx(Rfit,Ffit,'-')
         end
         title('POLYNOMIAL FITTING FOR CUMULATIVE SIZE DISTRIBUTION')
      end
      xlabel('r')
      ylabel('cumulative frequency')
      grid
      if flagprig==1
         meta cfd
         !gpp cfd /dps
         for j=1:ncopy
             !imprint/ultrascript cfd.ps
         end
%         !del cfd.met;*
%         !del cfd.ps;*
      end
      clg
   end
end
if (flagdis==4)|(flagdis==5)
   flagfit=menu...
      ('Choose type of curve fitting for soil size distribution',...
      'piecewise linear interpolation',...
      'cubic spline fitting',...
      'polynomial fitting')
   flagfp=menu...
      ('Check the curve fitting for soil size distribution',...
      'yes','no');
   if flagfp==1
      flagpaper=menu('Choose types of graph paper',...
         'linear X-Y plot',...
         'semi-log X-Y plot');
     Rfit=rmin:(rmax-rmin)/100:rmax;
      if flagfit==1
         if flagpaper==1
            plot(Rvec,Fvec,'-')
         else
```

```
            semilogx(Rvec,Fvec,'-')
         end
         title('PICEWISE LINEAR INTERPOLATION FOR SIZE DISTRIBUTION')
      elseif flagfit==2
         Ffit=spline(Rvec,Fvec,Rfit);
         if flagpaper==1
            plot(Rfit,Ffit,'-')
         else
            semilogx(Rfit,Ffit,'-')
         end
         title('CUBIC SPLINE FITTING FOR SIZE DISTRIBUTION')
      elseif flagfit==3
         Cp=polyfit(Rvec,Fvec,ndis-1);
         Ffit=polyval(Cp,Rfit);
         if flagpaper==1
            plot(Rfit,Ffit,'-')
         else
            semilogx(Rfit,Ffit,'-')
         end
         title('POLYNOMIAL FITTING FOR SIZE DISTRIBUTION')
      end
      xlabel('r')
      ylabel('frequency')
      grid
      if flagprig==1
         meta fsd
         !gpp fsd /dps
         for j=1:ncopy
             !imprint/ultrascript fsd.ps
         end
%         !del fsd.met;*
%         !del fsd.ps;*
      end
      clg
   end
end
flagpackdens=menu('Choose soil packing type',...
   'densest packing',...
   'packing based on required pore ratio')
if flagpackdens==1
   flagtermi=menu('Criterion to terminate packing process',...
      'specify max elapsed time to generate the last partical',...
      'specify totle elapsed time to generate the whol packing')
   if flagtermi==1
      temax=input(...
        'Max elapsed time (sec.) to generate the last partical:  ')
   else
      tcum=input(...
        'Totle elapsed time (sec.) to generate the whol packing:  ')
   end
end
```

```
if flagpackdens==2
   ratioreq=input('Enter required soil pore ratio:     ')
end
width=input('Enter width of box confining soil particles:      ')
depth=input('Enter depth of box confining soil particles:      ')
flagimpact=menu('Choose type of penetration',...
   'static one',...
   'dynamic one',...
   'no penetration')
if flagimpact==1 | flagimpact==2
   if flagimpact==1
      movet=input('Enter total movement of punch:     ')
      k=input('The number of increments for punch movement:      ')
   else
      speedpunch=input('Enter initial speed of punch:     ')
      dt=input('Enter increment of time:      ')
      masspunch=input('Enter mass of punch:    ')
      gammar=input('Enter density of soil particle:     ')
   end
   rlaud=input('Enter radius of punch:      ')
   flagprofile=menu('Show profile of soil surface','yes','no')
   if flagprofile==1
      nsub=input('Enter number of subwidth for soil profile:     ')
   end
   flagpmove=menu('Show individual soil particle movement','yes','no')
   flaghistdis=menu('Show displacement history','yes','no')
   fd=input('Enter factor to multiply displacements:      ')
end
axisar=menu('Choose aspect ratio of axis','normal','square')
if axisar==1
   axis('normal')
elseif axisar==2
   axis('square')
end
if flagprip==1
   !imprint mcssp.in
end
clc


disp('                 ****** SIMULATE SOIL PARTICLE PACKING ******')

diary off
disp('                        (Wait,computer is in process.)       ')
t1=clock;
flops(0);
T=[-pi:pi/4:pi]';
A=sin(T);
B=cos(T);
Unit=ones(A);
axis([-width/2 width/2 -depth depth/10]);
```

```
Xbox=[-width/2 -width/2 width/2 width/2 -width/2];
Ybox=[-depth 0 0 -depth -depth];
delt=rmean/50;
if flagpackdens==1
   if flagtermi==1
      timetole=temax;
   else
      timetole=tcum;
   end
   [PP,n,ratiodensest,Et,Ratiopo]=rpd(flagdis,rmean,rvar,rstd,...
      rmax,rmin,flagfit,fm,Rvec,Fvec,ndis,width,depth,delt,...
      flagtermi,timetole);
   if flagsave==1
      diary on
      n
      ratiodensest
      diary off
   end
else
   [PP,n]=rpm(ratioreq,flagdis,rmean,rvar,rstd,rmax,rmin,...
      flagfit,fm,Rvec,Fvec,ndis,width,depth,delt);
end
PPo=PP;
plot(Xbox,Ybox,'-')
hold on
plot(A*PP(:,2)'+Unit*PP(:,3)',-B*PP(:,2)'-Unit*PP(:,4)','-')
title('SIMULATION OF SOIL PARTICLES PACKING')
xlabel('x')
ylabel('y')
grid
te=etime(clock,t1);
if flagprig==1
   meta spp
   !gpp spp /dps
   for j=1:ncopy
       !imprint/ultrascript spp.ps
   end
%   !del spp.met;*
%   !del spp.ps;*
end
%!show day
%timeelapsed=te
%flop=flops
hold off
clg

if flagpackdens==1
   plot([1:n],Et)
   hold on
   title('Elapsed Time vs. Particle Number')
   xlabel('Particle Number')
```

```
ylabel('Elapsed Time (sec.)')
if flagprig==1
   meta etn
   !gpp etn /dps
   for j=1:ncopy
       !imprint/ultrascript etn.ps
   end
end
hold off

plot([1:n],cumsum(Et))
hold on
title('Cumulative Elapsed Time vs. The Number of Particles')
xlabel('The Number of Particles')
ylabel('Cumulative Elapsed Time (sec.)')
if flagprig==1
   meta ctn
   !gpp ctn /dps
   for j=1:ncopy
       !imprint/ultrascript ctn.ps
   end
end
hold off

plot([0:n],[1;Ratiopo])
hold on
title('Pore Ratio vs. The Number of Particles')
xlabel('The Number of Particles')
ylabel('Pore Ratio')
if flagprig==1
   meta prn
   !gpp prn /dps
   for j=1:ncopy
       !imprint/ultrascript prn.ps
   end
end
hold off

plot([0;cumsum(Et)],[1;Ratiopo])
hold on
title('Pore Ratio vs. Cumulative Elapsed Time')
xlabel('Cumulative Elapsed Time (sec.)')
ylabel('Pore Ratio')
if flagprig==1
   meta ctpr
   !gpp ctpr /dps
   for j=1:ncopy
       !imprint/ultrascript ctpr.ps
   end
end
hold off
```

```
   if flagimpact==3
      clc
      disp('                                    *** END *** ')
      break
      save mcssp
   end
end
clc


disp('                    ****** SIMULATE SOIL PARTICLE MOVEMENT ******')

%disp('                          (Pause,strike any key to continue.)')
%pause
clg
if flagpmove~=1
   disp('                          (Wait, computer is in process.)')
end
Xsuf=[-width width];
Ysuf=[0 0];
axis([-width/2 width/2 -depth depth/10]);
if flagpmove==1
   clc
   plot(Xbox,Ybox,'-')
   hold on
   plot(A*PP(:,2)'+Unit*PP(:,3)',-B*PP(:,2)'-Unit*PP(:,4)','-')
   title('SIMULATION OF SOIL PARTICLE MOVEMENT')
   xlabel('x')
   ylabel('y')
   grid
   plot(Xsuf,Ysuf,'i')
   plot(Xsuf,Ysuf,'--')
end
flag=1;
if flaghistdis==1
   XP=PP(:,3)';
   YP=PP(:,4)';
end
if flagimpact==1
   dmove=movet/k;
   move=dmove;
   Move=0;
   while move<=movet
      if flagpmove==1
         PP=fmvm(PP,n,rlaud,move,A,B,delt,flag,depth,width);
      else
         PP=fmv(PP,n,rlaud,move,delt,flag,depth,width);
      end
      if flaghistdis==1
         XP=[XP;PP(:,3)'];
         YP=[YP;PP(:,4)'];
```

```
      end
      Move=[Move,move];
      move=move+dmove;
   end
   move=movet;
elseif flagimpact==2
   energypunch=.5*masspunch*speedpunch*speedpunch;
   energyloss=0;
   Energyloss=0;
   Forceimpact=0;
   move=0;
   Move=0;
   Speedpunch=speedpunch;
   timeimpact=0;
   Timeimpact=timeimpact;
   energyrem=energypunch-energyloss;
   while energyrem/energypunch>.001
      PPt=PP;
      move=move+speedpunch*dt;
      if flagpmove==1
         PP=fmvm(PP,n,rlaud,move,A,B,delt,flag,depth,width);
      else
         PP=fmv(PP,n,rlaud,move,delt,flag,depth,width);
      end
      if flaghistdis==1
         XP=[XP;PP(:,3)'];
         YP=[YP;PP(:,4)'];
      end
      Rsq=PP(:,2).*PP(:,2);
      Vsq=((PPt(:,3)-PP(:,3)).^2+(PPt(:,4)-PP(:,4)).^2)/dt^2;
      Energyparticles=.5*gammar*pi*Rsq.*Vsq;
      energyloss=energyloss+sum(Energyparticles);
      Energyloss=[Energyloss,energyloss];
      Move=[Move,move];
      energyrem=energypunch-energyloss;
      if energyrem>0
         Forceimpact=[Forceimpact,sum(Energyparticles)/move];
         speedpunch=sqrt(2*energyrem/masspunch);
      else
         Forceimpact=[Forceimpact,0];
         speedpunch=0;
      end
      Speedpunch=[Speedpunch,speedpunch];
      Timeimpact=[Timeimpact,timeimpact+dt];
   end
end
Xlaud=[rlaud rlaud -rlaud -rlaud];
Ylaud=[10 -move -move 10];
axis;
if flagpmove==1
   plot(Xlaud,Ylaud,'-')
```

```
end
if flagprofile==1
   subwidth=(width/2-rlaud)/nsub;
   indi=0;
   for ind=1:nsub
       Fs=find(((rlaud+(ind-1)*subwidth)<=PP(:,3))...
          &(PP(:,3)<(rlaud+ind*subwidth)));
       if ~isempty(Fs)
          indi=indi+1;
          [Ys(indi),kk]=min(PP(Fs(:),4)-PP(Fs(:),2));
          Xs(indi)=PP(Fs(kk),3);
       end
   end
   Xi=rlaud:subwidth/40:width/2;
   Yi=spline(Xs,Ys,Xi);
   if flagpmove~=1
      axis;
      plot(Xbox,Ybox,'-')
      hold on
      plot(Xsuf,Ysuf,'i')
      plot(Xsuf,Ysuf,'--')
      plot(A*PP(:,2)'+Unit*PP(:,3)',-B*PP(:,2)'-Unit*PP(:,4)','-')
      plot(Xlaud,Ylaud,'-')
      title('PARTICLE PACKING AND SURFACE PROFILE AFTER DISPLACEMENT')
      xlabel('x')
      ylabel('y')
      grid
   end
   plot(Xi,-Yi)
end
if flagprig==1
   meta spm
   !gpp spm /dps
   for j=1:ncopy
       !imprint/ultrascript spm.ps
   end
%   !del spm.met;*
%   !del spm.ps;*
end
hold off
clc


disp('                         ****** SHOW DISPLACEMENT FIELD ******')

clg
axis;
plot(Xbox,Ybox,'-')
hold on
plot(Xsuf,Ysuf,'i')
plot(Xsuf,Ysuf,':')
```

```
Xlaud=[rlaud rlaud -rlaud -rlaud];
Ylaud=[10 0 0 10];
plot(Xlaud,Ylaud,'-')
for j=1:n
    if PP(j,3)~=PPo(j,3) | PP(j,4)~=PPo(j,4)
       plot([PPo(j,3),PPo(j,3)+fd*(PP(j,3)-PPo(j,3))],...
       -[PPo(j,4),PPo(j,4)+fd*(PP(j,4)-PPo(j,4))],'-')
    end
end
title('SHOW DISPLACEMENT FIELD')
xlabel('x')
ylabel('y')
if flagprig==1
   meta disf
   !gpp disf /dps
   for j=1:ncopy
       ! imprint/ultrascript disf.ps
   end
%   !del disf.met;*
%   !del disf.ps;*
end
hold off
clg

if flaghistdis==1
   if flagimpact==2
      [nr,nc]=size(XP);
      k=nr-1;
   end
   for i=1:k
       axis;
       plot(Xbox,Ybox,'-')
       hold on
       plot(Xsuf,Ysuf,'i')
       plot(Xsuf,Ysuf,':')
       Ylaud=[10 -Move(i) -Move(i) 10];
       plot(Xlaud,Ylaud,'-')
       Ylaud=[-Move(i) -Move(i+1) -Move(i+1) -Move(i)];
       plot(Xlaud,Ylaud,'--')
       for j=1:n
           if XP(i,j)~=XP(i+1,j) | YP(i,j)~=YP(i+1,j)
              plot([XP(i,j),XP(i+1,j)],...
              -[YP(i,j),YP(i+1,j)],'-')
           end
       end
       title(['DISPLACEMENT HISTORY (PROBE DEPTH=',...
          num2str(Move(i)),' AND DEPTH INCREMENT=',...
          num2str(Move(i+1)-Move(i)),')'])
       xlabel('x')
       ylabel('y')
       if flagprig==1
```

```
          meta idh
          !gpp idh /dps
          for j=1:ncopy
              !imprint/ultrascript idh.ps
          end
%          !del idh.met;*
%          !del idh.ps;*
       end
       keyboard
       hold off
       clg
   end

   axis;
   plot(Xbox,Ybox,'-')
   hold on
   plot(Xsuf,Ysuf,'i')
   plot(Xsuf,Ysuf,':')
   Ylaud=[10 0 0 10];
   plot(Xlaud,Ylaud,'-')
   for i=1:k
       for j=1:n
           if XP(i,j)~=XP(i+1,j) | YP(i,j)~=YP(i+1,j)
              plot([XP(i,j),XP(i+1,j)],...
              -[YP(i,j),YP(i+1,j)],'-')
           end
       end
   end
   title('SHOW DISPLACEMENT HISTORY')
   xlabel('x')
   ylabel('y')
   if flagprig==1
      meta dph
      !gpp dph /dps
      for j=1:ncopy
          !imprint/ultrascript dph.ps
      end
%      !del dph.met;*
%      !del dph.ps;*
   end
   keyboard
   hold off
   clg
end
clc


disp('                       ****** ZOOM ON PROFILE OF SOIL SURFACE ******')

disp('                           (For soil profile show required only)')
% disp('                               (strike any key to continue)'),pause
```

```
 if flagprofile==1
    clc
    plot(Xi,-Yi)
    title('PROFILE OF SOIL SURFACE')
    xlabel('x')
    ylabel('y')
    grid
    if flagprig==1
       meta zprof
       !gpp zprof /dps
       for j=1:ncopy
           !imprint/ultrascript zprof.ps
       end
%       !del zprof.met;*
%       !del zprof.ps;*
    end
end
clc


disp('      ***** SHOW RESPONSE OF SOIL TO DYNAMIC PENETRATION *****')

disp('                         (For dynamic penetration only)')
if flagimpact~=2
   clc
   disp('                                       *** END *** ')
   break
   save mcssp
end
disp('Plot Impact Force vs. Elapsed Time')
plot(Timeimpact,Forceimpact)
hold on
title('Impact Force vs. Elapsed Time')
xlabel('Elapsed Time')
ylabel('Impact Force')
keyboard
if flagprig==1
   meta ft
   !gpp ft /dps
   for j=1:ncopy
       !imprint/ultrascript ft.ps
   end
%   !del ft.met;*
%   !del zprof.ps;*
end
hold off

disp('Plot Probe Depth vs. Elapsed Time')
plot(Timeimpact,Move)
hold on
title('Probe Depth vs. Elapsed Time')
```

```
xlabel('Elapsed Time')
ylabel('Probe Depth')
keyboard
if flagprig==1
   meta pdt
   !gpp pdt /dps
   for j=1:ncopy
       !imprint/ultrascript pdt.ps
   end
end
hold off

disp('Plot Energy lost vs. Elapsed Time')
plot(Timeimpact,Energyloss)
hold on
title('Energy lost vs. Elapsed Time')
xlabel('Elapsed Time')
ylabel('Energy lost')
keyboard
if flagprig==1
   meta et
   !gpp et /dps
   for j=1:ncopy
       !imprint/ultrascript et.ps
   end
end
hold off

disp('Plot Impact Speed vs. Elapsed Time')
plot(Timeimpact,Speedpunch)
hold on
title('Impact Speed vs. Elapsed Time')
xlabel('Elapsed Time')
ylabel('Impact Speed')
keyboard
if flagprig==1
   meta st
   !gpp st /dps
   for j=1:ncopy
       !imprint/ultrascript st.ps
   end
end
hold off

disp('Plot Impact Speed vs. Probe Depth')
plot(Move,Speedpunch)
hold on
title('Impact Speed vs. Probe Depth')
xlabel('Probe Depth')
ylabel('Impact Speed')
keyboard
```

```
if flagprig==1
   meta sd
   !gpp sd /dps
   for j=1:ncopy
       !imprint/ultrascript sd.ps
   end
end
hold off

disp('Plot Impact Force vs. Probe Depth')
plot(Move,Forceimpact)
hold on
title('Impact Force vs. Probe Depth')
xlabel('Probe Depth')
ylabel('Impact Force')
keyboard
if flagprig==1
   meta fd
   !gpp fd /dps
   for j=1:ncopy
       !imprint/ultrascript fd.ps
   end
end
hold off

disp('Plot Energy Lost vs. Probe Depth')
plot(Move,Energyloss)
hold on
title('Energy lost vs. Probe Depth')
xlabel('Probe Depth')
ylabel('Energy lost from Punch')
keyboard
if flagprig==1
   meta ed
   !gpp ed /dps
   for j=1:ncopy
       !imprint/ultrascript ed.ps
   end
end
hold off
clc
save mcssp


disp('                                    *** END *** ')
```

APPENDIX B

M-function to Generate Random Packing

```
function [PP,n]=rpm(ratioreq,flagdis,rmean,rvar,rstd,rmax,rmin,...
                flagfit,fm,Rvec,Fvec,ndis,width,depth,delt)

%***********************************************************%
% An M-file to generate random packing based on given size   %
% distribution of particles                                  %
%                                                            %
% AUTHOR: Yuehui Fan  3-4-91                                 %
%***********************************************************%

flag=0;
move=0;
rlaud=0;
n=0;
PP=[];
areapar=0;
areatal=width*depth;
ratiopo=1;
while ratiopo>=ratioreq
   n=n+1;
   PP=fslm(PP,n,flagdis,rmean,rvar,rstd,rmax,rmin,flagfit,width,...
      depth,flag,delt,move,rlaud,fm,Rvec,Fvec,ndis);
   areapar=pi*PP(n,2)*PP(n,2)+areapar;
   ratiopo=(areatal-areapar)/areatal;
end
```

APPENDIX C

M-function to Generate Densest Random Packing

```
function
[PP,n,ratiodensest,Et,Ratiopo]=rpd(flagdis,rmean,rvar,rstd,...
         rmax,rmin,flagfit,fm,Rvec,Fvec,ndis,width,depth,delt,...
         flagtermi,timetole)

%****************************************************************%
% An M-function to generate the densest random packing based on  %
% given size distribution of particles.                          %
%                                                                %
% AUTHOR: Yuehui Fan  3-10-91                                    %
%****************************************************************%


flag=0;
move=0;
rlaud=0;
n=0;
PP=[];
areapar=0;
areatal=width*depth;
Et=[];
et=0;
Ratiopo=[];
while et<timetole
   n=n+1;
   t0=clock;
   PP=fslm(PP,n,flagdis,rmean,rvar,rstd,rmax,rmin,flagfit,width,...
      depth,flag,delt,move,rlaud,fm,Rvec,Fvec,ndis);
   et=etime(clock,t0)
   Et=[Et;et];
   if flagtermi==2
      et=sum(Et);
   end
   areapar=pi*PP(n,2)*PP(n,2)+areapar;
   ratiopo=(areatal-areapar)/areatal;
   Ratiopo=[Ratiopo;ratiopo];
end
ratiodensest=ratiopo;
```

APPENDICES D

M-function to Generate Random Size and Location

```
function PP=fslm(PP,n,flagdis,rmean,rvar,rstd,rmax,rmin,flagfit,...
            width,depth,flag,delt,move,rlaud,fm,Rvec,Fvec,ndis)

% An M-file called by 'frpm.m'

if (flagdis==1)|(flagdis==2)
   r=rmean+2*rvar*(rand-0.5);
elseif flagdis==3
   rand('normal')
   r=rmean+rstd*rand;
   rand('uniform')
elseif (flagdis==4)|(flagdis==5)
   rflag=0;
   while ~rflag
      r=rmax-(1-rand)*(rmax-rmin);
      if flagfit==1
         fc=table1([Rvec,Fvec],r);
      elseif flagfit==2
         fc=spline(Rvec,Fvec,r);
      elseif flagfit==3
         Cp=polyfit(Rvec,Fvec,ndis-1);
         fc=polyval(Cp,r)
      end
      if fm*rand<=fc
         rflag=1;
      end
   end
end
xcentre=(rand-0.5)*(width-2*r);
ycentre=(depth-2*r)*rand+r;
PP=[PP;n r xcentre ycentre];
new=n;
[PP,G]=mol(PP,n,new,delt);
Fo=find(G);
while ~isempty(Fo)
   Go=zeros(n,1);
   lengthFo=length(Fo);
   for ind=1:lengthFo
       new=PP(Fo(ind),1);
       PP(new,:)=mdc(PP(new,:),flag,delt,depth,width,move,rlaud);
       [PP,G]=mol(PP,n,new,delt);
       Go=Go|G;
   end
   Fo=find(Go);
end
```

APPENDIX E

M-function to Generate Particle Movement

```
function PP=fmv(PP,n,rlaud,move,delt,flag,depth,width)

% An M-function to generate particle movement

Pmd=find(PP(:,4)-PP(:,2)<move & abs(PP(:,3))<rlaud);
if ~isempty(Pmd)
   PP(Pmd,4)=move+PP(Pmd,2)+delt;
end
Pmr=find(PP(:,3)>0 & PP(:,4)<move & PP(:,3)-PP(:,2)<rlaud);
if ~isempty(Pmr)
   PP(Pmr,3)=rlaud+PP(Pmr,2)+delt;
end
Pml=find(PP(:,3)<0 & PP(:,4)<move & PP(:,3)+PP(:,2)>-rlaud);
if ~isempty(Pml)
   PP(Pml,3)=-rlaud-PP(Pml,2)-delt;
end
Pmove=[Pmd;Pml;Pmr];
while ~isempty(Pmove)
      [PP,G]=fmt(PP,n,Pmove,flag,delt,depth,width,move,rlaud);
      Pmove=find(G);
end
```

APPENDIX F

M-function to Generate and Show Particle Movement

```
function PP=fmvm(PP,n,rlaud,move,A,B,delt,flag,depth,width)

% An M-function to generate and show particle movement

Unit=ones(A);
Xlaud=[rlaud rlaud -rlaud -rlaud];
Ylaud=[10 -move -move 10];
plot(Xlaud,Ylaud,'-')
Pmd=find(PP(:,4)-PP(:,2)<move & abs(PP(:,3))<rlaud);
if ~isempty(Pmd)
   for ind=1:length(Pmd)
       plot (A*PP(Pmd(ind),2)+PP(Pmd(ind),3),...
            -B*PP(Pmd(ind),2)-PP(Pmd(ind),4),'i');
   end
   PP(Pmd,4)=move+PP(Pmd,2)+delt;
   plot (A*PP(Pmd,2)'+Unit*PP(Pmd,3)',...
        -B*PP(Pmd,2)'-Unit*PP(Pmd,4)','-');
end
Pmr=find(PP(:,3)>0 & PP(:,4)<move & PP(:,3)-PP(:,2)<rlaud);
if ~isempty(Pmr)
   for ind=1:length(Pmr)
       plot (A*PP(Pmr(ind),2)+PP(Pmr(ind),3),...
            -B*PP(Pmr(ind),2)-PP(Pmr(ind),4),'i');
   end
   PP(Pmr,3)=rlaud+PP(Pmr,2)+delt;
   plot (A*PP(Pmr,2)'+Unit*PP(Pmr,3)',...
        -B*PP(Pmr,2)'-Unit*PP(Pmr,4)','-');
end
Pml=find(PP(:,3)<0 & PP(:,4)<move & PP(:,3)+PP(:,2)>-rlaud);
if ~isempty(Pml)
   for ind=1:length(Pml)
       plot (A*PP(Pml(ind),2)+PP(Pml(ind),3),...
            -B*PP(Pml(ind),2)-PP(Pml(ind),4),'i');
   end
   PP(Pml,3)=-rlaud-PP(Pml,2)-delt;
   plot (A*PP(Pml,2)'+Unit*PP(Pml,3)',...
        -B*PP(Pml,2)'-Unit*PP(Pml,4)','-');
end
Pmove=[Pmd;Pml;Pmr];
while ~isempty(Pmove)
      [PP,G]=fmtm(PP,n,Pmove,flag,delt,depth,width,A,B,move,rlaud);
      Pmove=find(G);
end
plot(Xlaud,Ylaud,'i')
```

APPENDIX G

M-function to Iterate Particle Movement

```
function [PP,G]=fmt(PP,n,Pmove,flag,delt,depth,width,move,rlaud)

% An M-file to iterate particle movement

lpm=length(Pmove);
Go=zeros(n,1);
for ind=1:lpm
    new=PP(Pmove(ind),1);
    Pin=PP(new,:);
    PP(new,:)=mdc(Pin,flag,delt,depth,width,move,rlaud);
    PPi=PP;
    [PP,G]=mol(PPi,n,new,delt);
    Go=Go|G;
end
G=Go;
```

APPENDIX H

M-function to Iterate and Show Particle Movement

```
function [PP,G]=fmtm(PP,n,Pmove,flag,delt,depth,width,A,B,...
         move,rlaud)

% An M-file to iterate and show particle movement

lpm=length(Pmove);
Go=zeros(n,1);
for ind=1:lpm
    new=PP(Pmove(ind),1);
    Pin=PP(new,:);
    PP(new,:)=mdc(Pin,flag,delt,depth,width,move,rlaud);
    if PP(new,:)~=Pin
       plot (Pin(2)*A+Pin(3),-Pin(2)*B-Pin(4),'i')
       plot (PP(new,2)*A+PP(new,3),-PP(new,2)*B-PP(new,4),'-')
    end
    PPi=PP;
    [PP,G]=mol(PPi,n,new,delt);
    F=find(G);
    if ~isempty(F)
       for i=1:length(F)
           plot (PPi(F(i),2)*A+PPi(F(i),3),...
                -PPi(F(i),2)*B-PPi(F(i),4),'i')
           plot (PP(F(i),2)*A+PP(F(i),3),-PP(F(i),2)*B-PP(F(i),4),'-')
       end
    end
    Go=Go|G;
end
G=Go;
```

APPENDIX I

MEX-file to Confine Particles in Domain

```
      SUBROUTINE mdc(Po,Pin,flag,delt,depth,width,move,rlaud)
C
C     An MEX-file to confine particles in domain
C
      INTEGER i
      REAL*8 flag
      REAL*8 Po(1,4),Pin(1,4),delt,depth,width,move,rlaud
C
      DO i=1,4
         Po(1,i)=Pin(1,i)
      ENDDO
      IF (flag .EQ. 0) THEN
         IF (Pin(1,4)-Pin(1,2) .LT. 0) THEN
            Po(1,4)=Pin(1,2)+delt
         ENDIF
      ENDIF
      IF (Pin(1,4)+Pin(1,2) .GT. depth) THEN
         Po(1,4)=depth-Pin(1,2)-delt
      ENDIF
      IF (Pin(1,3)+Pin(1,2) .GT. width/2.) THEN
         Po(1,3)=width/2.-Pin(1,2)-delt
      ENDIF
      IF (Pin(1,3)-Pin(1,2) .LT. -width/2.) THEN
         Po(1,3)=-width/2.+Pin(1,2)+delt
      ENDIF
      IF (flag .EQ. 1) THEN
         IF (Pin(1,4)-Pin(1,2) .LT. move .AND.
     &      ABS(Pin(1,3)) .LT. rlaud) THEN
            Po(1,4)=move+Pin(1,2)+delt
         ENDIF
         IF ((Pin(1,3) .GT. 0) .AND. (Pin(1,4) .LT. move) .AND.
     &      (Pin(1,3)-Pin(1,2) .LT. rlaud)) THEN
            Po(1,3)=rlaud+Pin(1,2)+delt
         ENDIF
         IF ((Pin(1,3) .LT. 0) .AND. (Pin(1,4) .LT. move) .AND.
     &      (Pin(1,3)+Pin(1,2) .GT. -rlaud)) THEN
            Po(1,3)=-rlaud-Pin(1,2)-delt
         ENDIF
      ENDIF
C
      RETURN
      END
```

APPENDIX J

Gateway MEX-file to Confine Particles in Domain

```
C MDCG.FOR - Gateway function for MDC.FOR
C
C This is a FORTRAN code required for interfacing a .MEX file to
C MATLAB.
C
C This subroutine is the main gateway to MATLAB.  When a MEX function
C  is executed MATLAB calls the USRFCN subroutine in the corresponding
C  MEX file.
C
      SUBROUTINE USRFCN(NLHS, PLHS, NRHS, PRHS)
      INTEGER*4 PLHS(*), PRHS(*)
      INTEGER NLHS, NRHS
C
      INTEGER*4 CRTMAT, REALP, IMAGP, GETGLO, ALREAL, ALINT, GETSTR
      DOUBLE PRECISION GETSCA
C
C KEEP THE ABOVE SUBROUTINE, ARGUMENT, AND FUNCTION DECLARATIONS FOR
C USE IN ALL FORTRAN MEX FILES.
C-----------------------------------------------------------------------
C
      INTEGER*4 PoP, PinP, flagP, deltP, depthP, widthP, moveP, rlaudP
      INTEGER*4 Nr, Nc
C
C CHECK FOR PROPER NUMBER OF ARGUMENTS
C
      IF (NRHS .NE. 7) THEN
        CALL MEXERR('MDC requires 7 input arguments')
      ELSEIF (NLHS .NE. 1) THEN
        CALL MEXERR('MDC requires 1 output argument')
      ENDIF
      CALL GETSIZ(PRHS(1),Nr,Nc)
C
C CREATE MATRIXES FOR RETURN ARGUMENT
C
      PLHS(1) = CRTMAT(Nr,Nc,0)
C
C ASSIGN POINTERS TO THE VARIOUS PARAMETERS
C
      PoP = REALP(PLHS(1))
C
      PinP = REALP(PRHS(1))
      flagP = REALP(PRHS(2))
      deltP = REALP(PRHS(3))
      depthP = REALP(PRHS(4))
      widthP = REALP(PRHS(5))
      moveP = REALP(PRHS(6))
      rlaudP = REALP(PRHS(7))
C
C DO THE ACTUAL COMPUTATIONS IN A SUBROUTINE
C NOTE: THE VARIABLES:
C             PoP, PinP, flagP, deltP, depthP, widthP, moveP, and rlaudP
```

```
C        CONTAIN THE ADDRESSES OF THE NEWELY CREATED ARRAYS.  THE CALL
C        TO MOL USES THE %VAL TO CAUSE THE ADDRESSES STORED IN THE
C        VARIABLES TO BE PASSED TO MOL RATHER THAN THE ADDRESSES OF THE
C        VARIABLES THEMSELVES WHICH IS HOW FORTRAN NORMALLY WORKS.
C
      CALL mdc(%VAL(PoP),%VAL(PinP),%VAL(flagP),%VAL(deltP),
     & %VAL(depthP),%VAL(widthP),%VAL(moveP),%VAL(rlaudP))
C
      RETURN
      END
```

APPENDIX K

MEX-file to Solve Overlap

```
      SUBROUTINE mol(PPo,G,PPi,n,new,delt)
c
c     An MEX-file to solve overlap
c
      INTEGER i,j,ni,newi
      REAL*8 n,new
      REAL*8 PPo(int(n),4),PPi(int(n),4),delt,G(int(n))
      REAL*8 X, Y, rolap
      REAL*8 theta,R,Overlap
c
      ni=int(n)
      newi=int(new)
      DO i=1,ni
         G(i)=0
         DO j=1,4
            PPo(i,j)=PPi(i,j)
         ENDDO
      ENDDO
      IF ((newi .EQ. 1) .AND. (ni .EQ. 1)) THEN
         GOTO 1000
      ENDIF
      DO i=1,ni
         X=PPi(newi,3)-PPi(i,3)
         Y=PPi(newi,4)-PPi(i,4)
         R=sqrt(X*X+Y*Y)
         Overlap=R-PPi(newi,2)-PPi(i,2)
         IF ((R .EQ. 0.) .AND. (i .NE. newi)) THEN
            rolap=PPi(i,2)+PPi(newi,2)+delt
            theta=3.1415926*(1/2+ran(newi))
            PPo(i,3)=PPi(i,3)+rolap*sin(theta)
            PPo(i,4)=PPi(i,4)+rolap*cos(theta)
            G(i)=1
         ELSEIF ((Overlap .LT. -delt) .AND. (i .NE. newi)) THEN
            PPo(i,3)=PPi(i,3)-(Overlap-delt)*(PPi(i,3)-PPi(newi,3))/R
            PPo(i,4)=PPi(i,4)-(Overlap-delt)*(PPi(i,4)-PPi(newi,4))/R
            G(i)=1
         ENDIF
      ENDDO
1000  RETURN
      END
```

APPENDIX L

Gateway MEX-file to Solve Overlap

```
C MOLG.FOR - Gateway function for MOL.FOR
C
C This is a FORTRAN code required for interfacing a .MEX file to
C MATLAB.
C
C This subroutine is the main gateway to MATLAB.  When a MEX function
C  is executed MATLAB calls the USRFCN subroutine in the corresponding
C  MEX file.
C
      SUBROUTINE USRFCN(NLHS, PLHS, NRHS, PRHS)
      INTEGER*4 PLHS(*), PRHS(*)
      INTEGER NLHS, NRHS
C
      INTEGER*4 CRTMAT, REALP, IMAGP, GETGLO, ALREAL, ALINT, GETSTR
      DOUBLE PRECISION GETSCA
C
C KEEP THE ABOVE SUBROUTINE, ARGUMENT, AND FUNCTION DECLARATIONS FOR
C USE IN ALL FORTRAN MEX FILES.
C-----------------------------------------------------------------------
C
      INTEGER*4 PPoP, GP, PPiP, nP, newP, deltP
      INTEGER*4 Nr, Nc
C
C CHECK FOR PROPER NUMBER OF ARGUMENTS
C
      IF (NRHS .NE. 4) THEN
        CALL MEXERR('MOL requires 4 input arguments')
      ELSEIF (NLHS .NE. 2) THEN
        CALL MEXERR('MOL requires 2 output argument')
      ENDIF
      CALL GETSIZ(PRHS(1),Nr,Nc)
C
C CREATE MATRIXES FOR RETURN ARGUMENT
C
      PLHS(1) = CRTMAT(Nr,Nc,0)
      PLHS(2) = CRTMAT(Nr,1,0)
C
C ASSIGN POINTERS TO THE VARIOUS PARAMETERS
C
      PPoP = REALP(PLHS(1))
      GP = REALP(PLHS(2))
C
      PPiP = REALP(PRHS(1))
      nP = REALP(PRHS(2))
      newP = REALP(PRHS(3))
      deltP = REALP(PRHS(4))
C
C DO THE ACTUAL COMPUTATIONS IN A SUBROUTINE
C NOTE: THE VARIABLES:
C                       PPoP, GP, PPiP, nP, newP, AND deltP
C         CONTAIN THE ADDRESSES OF THE NEWELY CREATED ARRAYS.  THE CALL
```

```
C        TO MOL USES THE %VAL TO CAUSE THE ADDRESSES STORED IN THE
C        VARIABLES TO BE PASSED TO MOL RATHER THAN THE ADDRESSES OF THE
C        VARIABLES THEMSELVES WHICH IS HOW FORTRAN NORMALLY WORKS.
C
      CALL mol(%VAL(PPoP),%VAL(GP),%VAL(PPiP),%VAL(nP),%VAL(newP),
     & %VAL(deltP))
C
      RETURN
      END
```

VITA

Yuehui Fan was born in Zigong, Sichuan, People's Republic of China on November 3, 1958. He attended elementary schools and high school in that city. In March 1978, he entered Southwestern Agricultural University, Chongqing, People's Republic of China, and in February 1982 he received the Bachelor of Science degree in Agricultural Engineering. In the ensuing 5 years he worked as a teaching assistant and a lecturer in that university.

In the summer of 1988 he accepted a research assistantship in the Department of Agricultural Engineering, The University of Tennessee, Knoxville and began study toward a Master's degree. He received the degree of Master of Science with a major in Agricultural Engineering and a minor in Mathematics in August 1991.

The author was awarded memberships in both Phi Kappa Phi and Gamma Sigma Delta, the Honor Society of Agriculture.