ABSTRACT

ANALYSIS OF RANDOM PACKING OF UNIFORM SPHERES USING THE MONTE-CARLO SIMULATION METHOD

by
Sai Tulluri

The study of random packing of spheres has had a long history. It particularly interests many researchers because randomly packed hard spheres exhibit some features of the properties of simple liquid, e.g. the packing density and the radial distribution. Many researchers have studied random packing both experimentally and also by using computer simulation.

Monte Carlo simulation method was used to generate random loose packing of hard spheres. Packing characteristics like packing density, radial distribution function, co –ordination, angular distribution and fabric tensor have been computed for this packing system.

The packing density for a system of 8000 particles with a diameter of 0.15 was computed to be 0.582. The packing density obtained for random packing of loose spheres is in good agreement with Owe Berg [2], Tory [18] and about 2% less than the experimental values of Scott [9].

The radial distribution function for the 8000-particle system was also computed for the packing bed. The peak occurrence in the radial distribution plot was found to be in good agreement with Scott [9], Nolan [11] and Powell [17].

An average co-ordination number for the same size system was computed to be 6.71291 ± 0.023433 which is in good agreement with Smith [12], Visscher [1], Nolan [11].

Angular Distribution of the particles was also found by computing the contact angles for all the spheres with their neighbors. The histogram of the contact angles shows that there is no preferred direction for the particles, which shows the packing is completely random and that there is no particular pattern in their arrangement.

The fabric tensor was computed. The analysis of fabric tensor shows that all the planes in the packing assembly, generated by the Monte Carlo simulation, are principal planes. Similar situation exists in ideal fluid. Hence it is shown that the Monte Carlo simulated assembly can serve as model for ideal fluid.

ANALYSIS OF RANDOM PACKING OF UNIFORM SPHERES USING THE MONTE-CARLO SIMULATION METHOD

by
Sai S. Tulluri

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirement for the Degree of
Master of Science in Mechanical Engineering

Department of Mechanical Engineering

May 2003

APPROVAL PAGE

ANALYSIS OF RANDOM PACKING OF HARD SPHERES GENERATED USING THE MONTE CARLO SIMULATION METHOD

Sai Tulluri

Dr. Anthony D. Rosato, Thesis Advisor — Date
Professor of Mechanical Engineering, NJIT

Dr. Chao Zhu, Committee Member — Date
Assistant Professor of Mechanical Engineering, NJIT

Dr. Kwabena A. Narh, Committee Member — Date
Associate Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Sai S. Tulluri

Degree: Master of Science

Date: May 2003

Undergraduate and Graduate Education:

- Master of Science in Mechanical Engineering
  New Jersey Institute of Technology, Newark, NJ, 2003

- Bachelor of Science in Mechanical Engineering
  M.R.V.R.G.R. College of Engineering, Vizianagaram, India, 2001

Major Mechanical Engineering

To My Beloved Parents

ACKNOWLEDGEMENT

I would like to thank my thesis advisor Dr. Anthony D. Rosato for the valuable input and support throughout the course of this thesis. I would also like to thank Dr. Chao Zhu and Dr. Narh for participating in my thesis committee. I would like to extend many thanks to the graduate students of Granular science laboratory.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

The implications of packing of spheres are diverse. They are visibly found in the fields of metallurgy, ceramics, physics, chemistry and various other engineering applications. Over the years packing of spheres, of equal, unequal sized and non-spherical particles, has interested many researchers and they have developed numerous methods to describe and simulate packing of spheres.

1.1 Published Material

In what follows, a discussion of some of the literature on the packing of spheres is presented. The selected papers are merely representative of the several of the more important and interesting investigations that have been carried out.

G.D. Scott and D. M. Kilgour [8] proposed that models of randomly packed spheres exhibit some properties of simple liquids in that their radial distribution function and packing density are similar. They conducted experiments for both random loose packing and random close packing with 1/8 in. plexiglass, nylon and steel balls in air and also with steel balls immersed in oil. A series of measurements for random loose packing and for random close packing, with help of a mechanical vibrator, were made. The packing density so obtained for random loose packing was 0.60 and for random close packing was 0.63.

T.G. Owe Berg, R.L. McDonald and R. J. Trainor Jr. [2] studied the packing of spheres experimentally with approximately 5,000 ball bearings of 1/8-inch diameter. The ball bearings were poured into a plexiglass cylindrical container, of 63 mm diameter.

Water was poured into the container and then frozen. The ice block was taken out of the container and structure was examined as the thawing of the ice progressed from the boundary inwards. By weighing the balls and measuring the height to which they filled the container determined the density, which varied from 0.586 to 0.592. The container was then placed on a shaker and treated for certain period of time. Three different types of shakers were used, which yielded densities of 0.615, 0.612 and 0.614. According to the author, packing of spheres depends very much on the mode of shaking; one-dimensional shaking gives irregular random packing. On the other hand, three-dimensional shaking gives an almost perfect hexagonal close packed structure, with packing density of 74%. All the experiments reported were one-dimensional shaking, which gave a random packing of 61%.

William M. Visscher and M. Bolsterli [1] approached the problem of random packing by means of Monte Carlo computer simulation of the physical process of dropping spheres into a bin. The spheres position themselves under the influence of a unidirectional (vertical) gravitational force, rather than toward a center of attraction. The computer in three-dimensional were set up as follows: Periodic horizontal boundaries were assumed i.e. the ball (x, y, z) reappears at ($x \pm L_x$, $y \pm L_y$, z). Balls were dropped sequentially from a random point above the L_x x L_y bin. When a ball was dropped it hits ball 'm' or the floor. When it comes in contact with ball 'm', then it rolls down in a vertical plane on 'm' until it is in contact with 'm' and 'n'. Then it rolls downward in contact with 'm' and 'n' until it makes contact with 'p'. If the contact 'm', 'n' and 'p' is stable, it stops. If not it rolls on the double contact that goes down most steeply and so on. Anytime a ball contacts the floor it stops. The density of random stacks of mono-size and

fuzzy mono-size discs was reported as 0.82. According to the author physical shaking increases the density by allowing the structure to seek a configuration of lower potential energy. Code for generating three-dimensional system was built and tested by generating h.c.p system. For this, the bottom layer was carefully placed in a hexagonal array. The subsequent balls were dropped from positions directly above these base balls, but displaced by a certain constant arbitrary amount to fix the direction of initial roll. This procedure generated h.c.p array, only when the base balls were chosen to have radii slightly larger than the radii of the subsequent balls. Keeping the same base ball field and dropping procedure, the distribution of subsequent balls was changed from mono-size to fuzzy mono-size with 10^{-6} spread. This distorted the regular h.c.p structure after 7 or 8 layers of balls, with sudden transition to a random structure with a density of 0.582.

Y.F Cheng, S.J. Guo and H.Y. Lai [16] performed a computer simulation for evolution of random structure of spherical particles in two-dimensional (2D) by distinct element method (DEM). The forces among the particles considered were gravity, contact force, friction force and Van der Waals interaction (VDWI). The motions of the particles considered were translation and rotation. Two hundred and forty mono-sized spherical particles were located without overlapping inside a rigid box by generating the co-ordinate centers of the particles with the help of a random number generator. The initial packing density was only 0.4712. The particles were allowed to settle down under gravity. When the co-efficient of friction among the particles and in between the particles and the walls of the box was 0.364, the packing density was 0.8236, when there was no friction the density was observed to be 0.8696.

M.J. Powell [14] generated a structure of randomly packed equal size spheres using computer simulation. The study was aimed at determining the nearest neighbors in the assembly. Powell says that the properties of particulate assemblies such as the structural properties of powders or electrical conductivity depend on the number and distribution of contacts of each particle, hence, the number of nearest neighbors is of particular interest in the range of r= 1.0-1.1 sphere diameter since this determines the number of actual touching contacts and the number of near contacts. Powell determined the density of random close packing of the computer-generated assembly to be 0.636.

G.T. Nolan and P.E. Kavanagh [11] proposed that the characteristics of random packing are determined by its interstices. They defined interstices to be a network of channels passing through the lattice, rather than isolated regions of space. It was stated that the lattice is transformed from random loose packing to random close packing as the mean interstices volume decreases from a maximum to minimum. They generated an algorithm to simulate any random packing between random loose packing to random close packing. The packing density was calculated by measuring every complete and fractional sphere positioned within a radius of 5-sphere diameter to the center of lattice. A contact was defined when there was an overlap between two spheres. The mean overlap was defined to be 0.002 of the sphere diameter. The packing density ranged from 0.509 to a maximum of 0.638. The random loose packing was 0.52. The mean co-ordination number varied from 4.4 to 5.9 as the structure changed from random loose packing to random close packing. The radial distribution function for random close packing has peaks at 1.0 and a split peak at 1.725 and 1.97.

J.S. Goodling and M.S. Khader [17] used a simple technique to find the number of neighboring spheres that are in contact with any given sphere. A single sample was prepared by casting randomly poured ¼ inch polystyrene spheres (269 spheres) with epoxy in a 3.11 in. I. D. plastic pipe. The solidified cylinder was mounted and shaped in a lathe chuck, yielding a total sample length of 2.5 inch. The r, θ and Z of each sphere were determined by photographic means. Having known r, θ and Z of each particle the co-ordination number was determined assuming that a contact point existed if two particles are within one ball diameter plus or minus 0.055 inch of each other. The co-ordination number distribution in the bed was determined. The lower and upper limits of the distribution were reported to be 3 and 12 respectively. The bulk of spheres made contact with 7,8,9,10 other spheres. The average number of contacts for the entire bed was 8.1.

D. J. Adams and A.J. Matheson [5] simulated random close packing of hard spheres by use of computer. The idea behind the method was to place a new sphere at the tetrahedral site nearest to the center of packing, thereby producing a spherical model. The packing fraction was reported to be 0.628.

1.2 Thesis Outline

This thesis is aimed at studying the random loose packing of hard spheres generated by a Monte Carlo simulation. The simulation generated a loose assembly of spheres under pouring that settle down under only gravitational force. Hard walls were implemented for this simulation.

The thesis is divided into five chapters, each of which presents an analysis of a computed quantity that characterizes some aspect of the structure. Results were compared with analogous quantities in the literature.

Chapter 3 explains the details of the Monte Carlo simulation method and explains the modifications made in the code for this study. Chapter 3 focuses on packing density calculations as a function of the system size. Chapter 4 discusses the co-ordination number distribution while the following chapter presents the radial distribution function. The angular contact distribution of the near neighbors is described in Chapter 6. Finally, a global characterization of the microstructure is given by the fabric tensor. The deviatoric component of the fabric tensor is also computed.

CHAPTER 2

MONTE-CARLO SIMULATION CODE

2.1 Discussion

A Monte Carlo code [22] was used in this work to generate the loose packing assemblies. In the simulation, the spheres are randomly placed in a cell with hard vertical walls. The particles were allowed to 'fall' down to a packed formation at the bottom of the cell. The system thus obtained is the initial configuration. In order to obtain a system with lowest possible potential, (a) particles were moved one at a time with in a δ-neighborhood to generate a trial configuration; (b) System Energies were computed for trial and original configurations. Trial state was accepted as the new state if the difference in the system energies between trial and original is less than or equal to zero, if it is greater than zero then the new state is accepted with a probability of

$$P(\Delta E) = e^{-\Delta E/KT}$$

Where ΔE: Difference in the potential of trial and original configurations

K: Boltzmann Constant

T: Absolute Temperature

The steps (a) and (b) were carried out until an attempt to move all the particles of the system, was made, which constitutes one pass. If there is an overlap of any two particles in the trial configuration, then the trial configuration is rejected. Many passes are required to collapse the system.

The cell in which the spheres were placed had square base 3.0 units on one side and height of 6.0 units. The diameter of the sphere was determined by dividing the base side of the cell by the cube root of number of particles.

The largest system used in the previous study by other students was 512-particle system for 500,000 passes. For twenty-five trials the average packing fraction for the systems was found to be 0.5684 with a standard deviation of 0.0026.

2.2 Procedure and Results

It was desired to run the Monte Carlo simulation code for 729-particle system. The diameters of the particles were determined using the above said procedure of dividing the base side of the cell by the cube root of number of particles. The diameter was found to be 0.333. The system was set to run for 500,000 passes. The packing fraction found for this system was 0.5707. The energy file which gives potential of the system was observed and it was found that there was no significant fall in the energy of the system after 250,000 passes. It was decided that the system reaches a steady state after approximately 250,000, and the code was to run for only 250,000 instead of 500,000 as did earlier. The plot in the Figure 2.1 shows the fall of potential of the system with the number of passes.

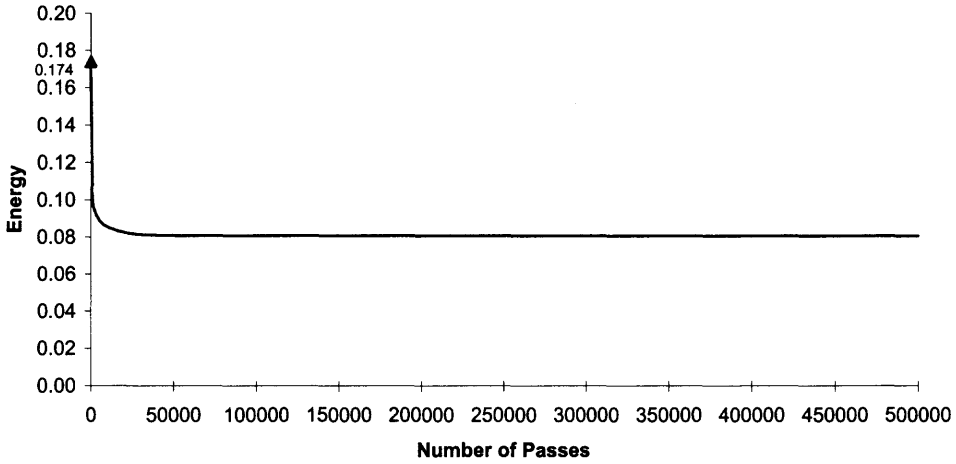


Figure 2.1 Fall of Potential with number of passes.

Twenty-five trials for the same size system were run, to generate twenty-five different configurations, when the plot file (case1.plo) was observed, it was found that all the configurations were the same. This led to the belief that the part of the code that generates random number wasn't working properly and on investigation it was observed that the same number was generated repeatedly, for different random number seed. To overcome this problem, that part of the code was rewritten and tested. This produced different configurations for the same size system.

The code was run for 25 times for system of 729 (9^3), 1000 (10^3), 1728 (12^3), 3375 (15^3), 8000(20^3) particles. The diameter of the particles was determined by dividing the base side of the cell by the cube root of number of particles. The plots exemplifying the drop of potential with the number of passes for each number of particles are shown below.

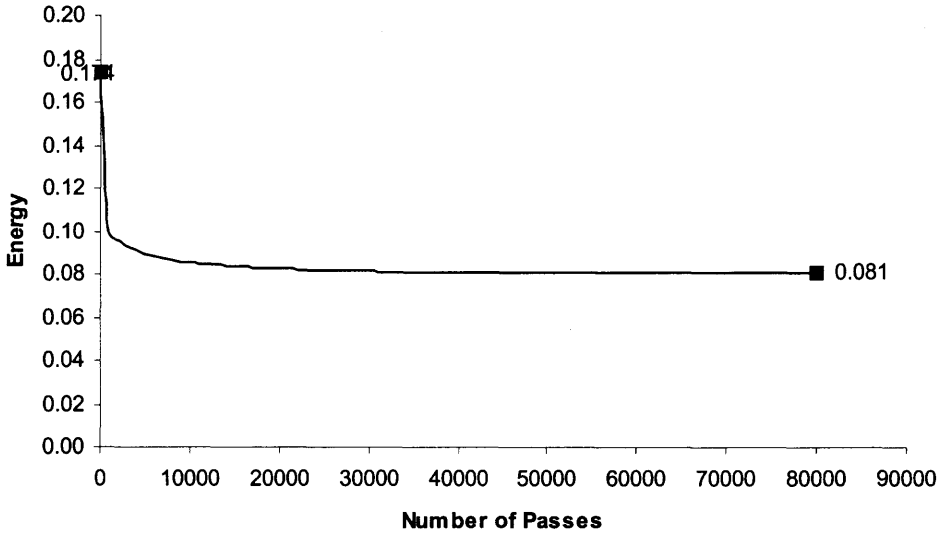


Figure 2.2 Potential Energy versus number of passes (N = 729).

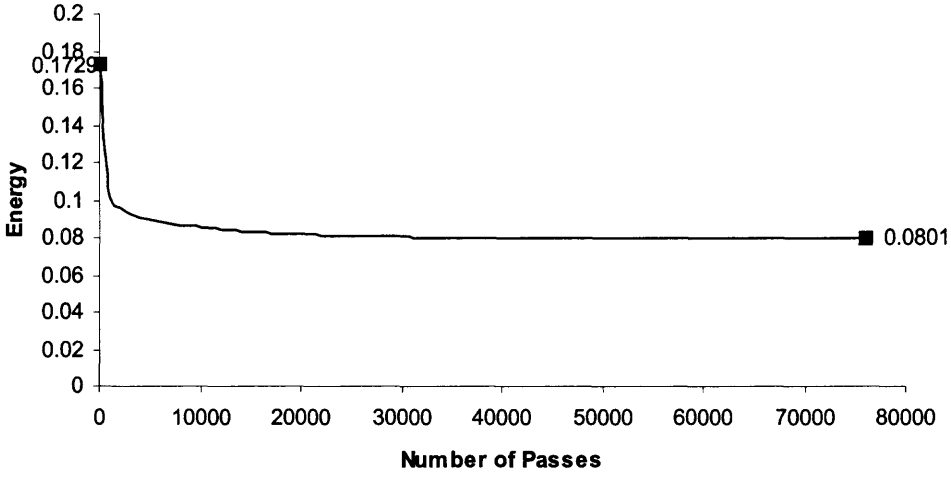


Figure 2.3 Potential Energy versus number of passes (N = 1000).

The potential energy of the systems was observed to remain constant after certain number of passes. The potential energies for 729, 1000, 1728 and 8000 particle system at the end of 250,000 passes are 0.081, 0.0801, 0.080 and .079 respectively. The drop in

potential energy was observed to be about 53 % of the original system energy. It was concluded that the system reached a steady state.

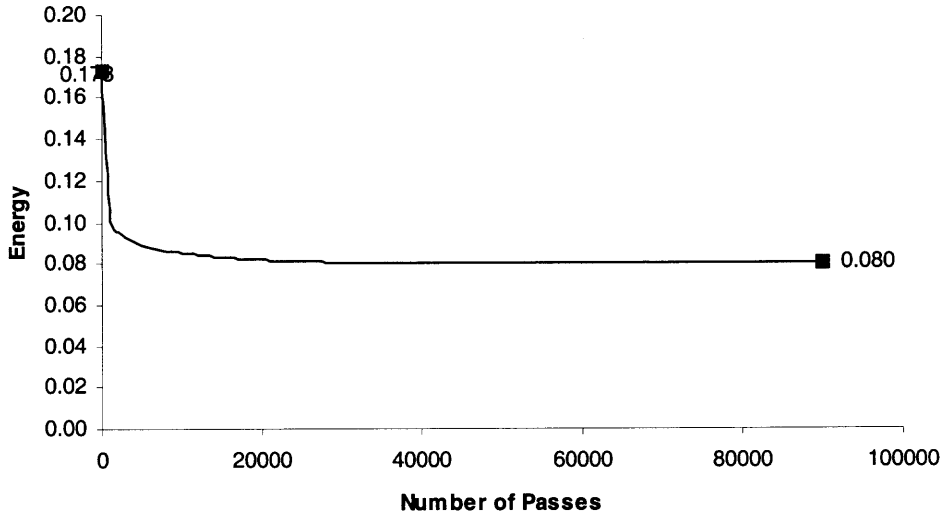


Figure 2.4 Potential Energy versus number of passes (N = 1728).

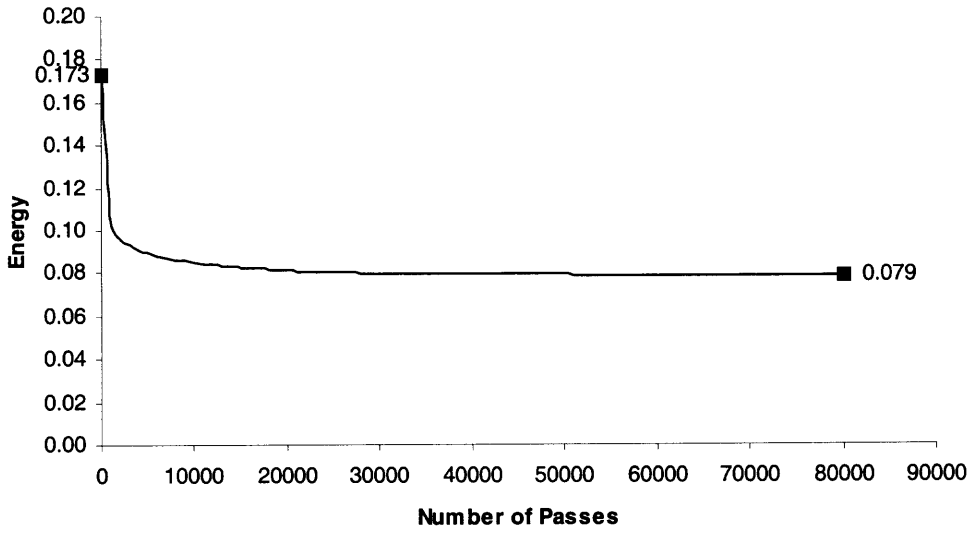


Figure 2.5 Potential Energy versus number of passes (N = 3375).

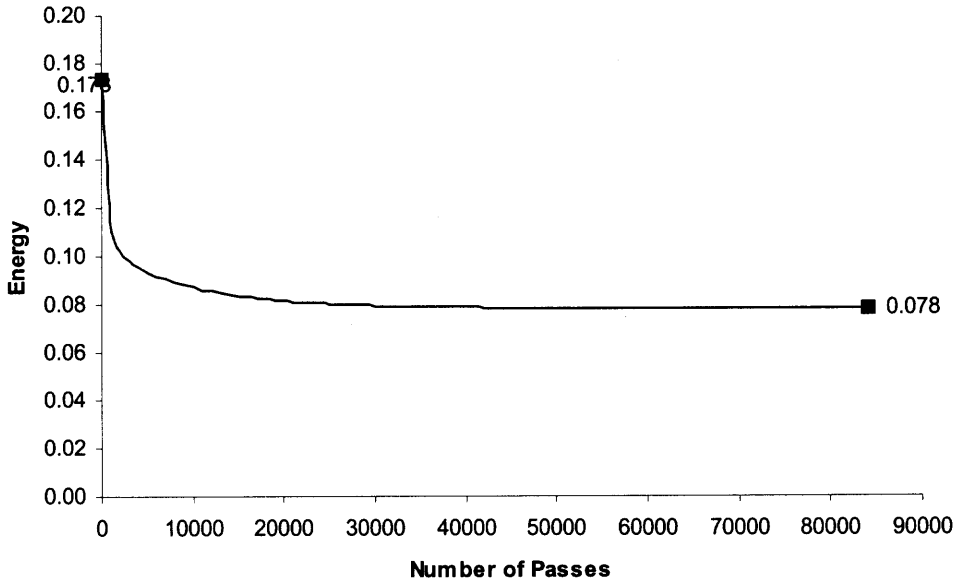


Figure 2.6 Potential Energy versus number of passes (N = 8000).

The solids fraction, co-ordination number, radial distribution function, angular distribution and fabric tensor for the set of runs at each number of particles were examined in the other chapters, to better understand the random loose packing of the simulated bed.

CHAPTER 3

SOLIDS FRACTION

3.1 Introduction

Spheres may be packed in a container either in regular fashion, or randomly as would result from pouring. Although the topic of this thesis is on random assemblies generated via a Monte Carlo method, a brief description of some of the features of regular packing is included. Complete details of this topic may be found in the literature (see for example [3]). In this chapter, solids fractions computed from the Monte Carlo simulated assemblies are presented and compared with published data.

In general, a regular packing may be envisioned to consist of ordered rows and layers with an angle of intersection between them. There are two types of layers that are generally considered depending on the value of this intersection angle, depicts in Fig. 3.1. When the angle between the rows is 90° it is called a *square* layer and when the angle of intersection is 60° the layer is called a *simple rhombic* or *triangular* layer. Various structures are possible giving rise to ordered systems having uniform solids fractions (i.e., the fraction of the volume occupied by solids) with characteristic lattice spacing. Such an assembly is in contrast to that which one obtains by simply pouring spheres into a containment vessel. Here, no global order is present and the solids fraction varies from point to point in the assembly.

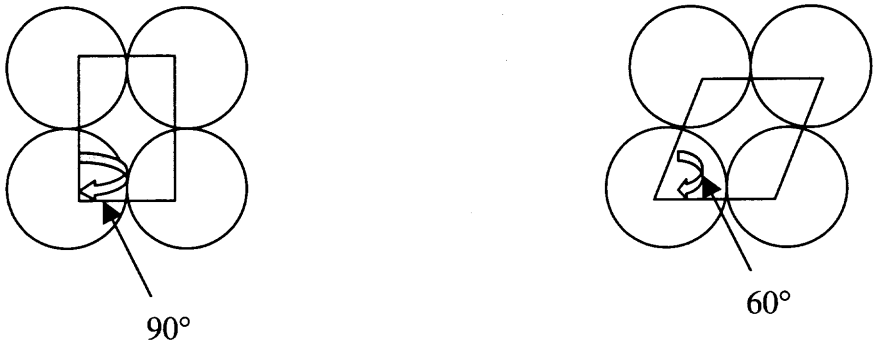


Figure 3.1 Square layer and Rectangular layer of packing arrangement

3.2 Random Packing

If the particles are arranged in an unsystematic or haphazard manner, then the arrangement is termed as random packing. Studies of random packings of hard spheres have been of interest since they can serve as models of liquid and glassy states. Random packing of spheres can be classified into two types.

a. Random Loose Packing

If the particles settle down randomly into a stable state under static gravitational force, such random packing is considered as loosely packed. The arrangement is stable under gravitational force but not stable when tapped or shaken. It has been reported that the solids fraction of random loose packing varies from 0.58 to 0.60.

b. Random Close Packing

A loosely packed random arrangement is stable under static gravitational force; but when such an arrangement is subjected to shaking or tapping it forms a denser arrangement, the density of which depends on the frequency of shaking. Values of the solids fraction for vibrated systems of spheres between 0.625 and 0.64 reported in the literature are generally denoted as being "random close".

3.3 Solids Fraction

Solids Fraction [3] ν can be defined as the ratio of volume of particles to the total volume or as one minus the porosity. That is, $\nu \equiv \frac{\text{Volume of solids}}{\text{Volume occupied}}$.

Many experimental and computational investigations focused on the solids fraction of random loose packing have appeared in the literature. The results of several of these are discussed here and the solids fractions are listed in Table 2.1

Scott and Kilgour [15] used a 1/8 in Plexiglas, nylon and steel balls in air and also in with steel in balls immersed in oil. The packing density obtained for steel balls in air for random loose packing was 0.60 and 0.63 for close packing.

Table 3.1 Comparison of Packing Fraction

Reference	Solids Fraction
Owe Berg [2]	0.586
Tory [18]	0.59
Visscher and Bosterli [1]	0.582
Scott [15]	0.60
Current	0.582

Nolan and Kavanagh [11] measured the packing density, for a simulated packing bed by measuring every complete and fractional sphere positioned within a radius of 5-sphere diameter from the center of lattice. They defined contact when there was an overlap between two spheres. The mean overlap was 0.002 of the sphere diameter. The packing density ranged from a minimum of 0.509 to 0.638 as the system moved from

random loose packing to random close packing. The value of random loose packing for this system was 0.52.

Owe Berg, McDonald and Trainor Jr. [2] experimentally studied packing of particles with 1/8-inch ball bearings. The packing fraction varied between 0.586 and 0.592 for loose packing.

Visscher and Bolsterli [1] approached by means of Monte Carlo computer simulation by the physical process of dropping spheres into a bin. The density of random structure was noted to be 0.582.

3.4 Monte Carlo Simulated Results

Packing density was calculated using the plane growth method. In this method, the entire length of packing bed was divided into a number of equal intervals, and packing fraction is calculated at the end of each interval. The packing fraction at 80% of the length of the packing bed was taken as the packing fraction of the entire system. This was done to avoid the variations in the packing fraction due to the irregularities on the top of the packing bed and also to account for the part of particles which were lost at the bottom of the cell since the cell was assumed to have hard walls.

The solids fraction was computed for 25 ensembles for different size packing assemblies that is (N= 729, 1000, 1728, 3375 and 8000 particles). The mean of the solids fraction for the 25 ensembles is taken as the solids fraction for that particular size of assembly. The sample deviation for the 25 ensembles was also computed. The mean solids fraction was found to be increasing as the system size increases. Table 3.2 shows the mean solids fraction along with sample deviation for each of the assemblies.

Table 3.2 Mean Solids Fraction for Different Packing Assemblies

System Size	Mean Solids Fraction	Sample Deviation
729	0.5707	0.00231
1000	0.5737	0.001919
1728	0.5750	0.00180
3375	0.578314	0.00132
8000	0.5820	0.00074

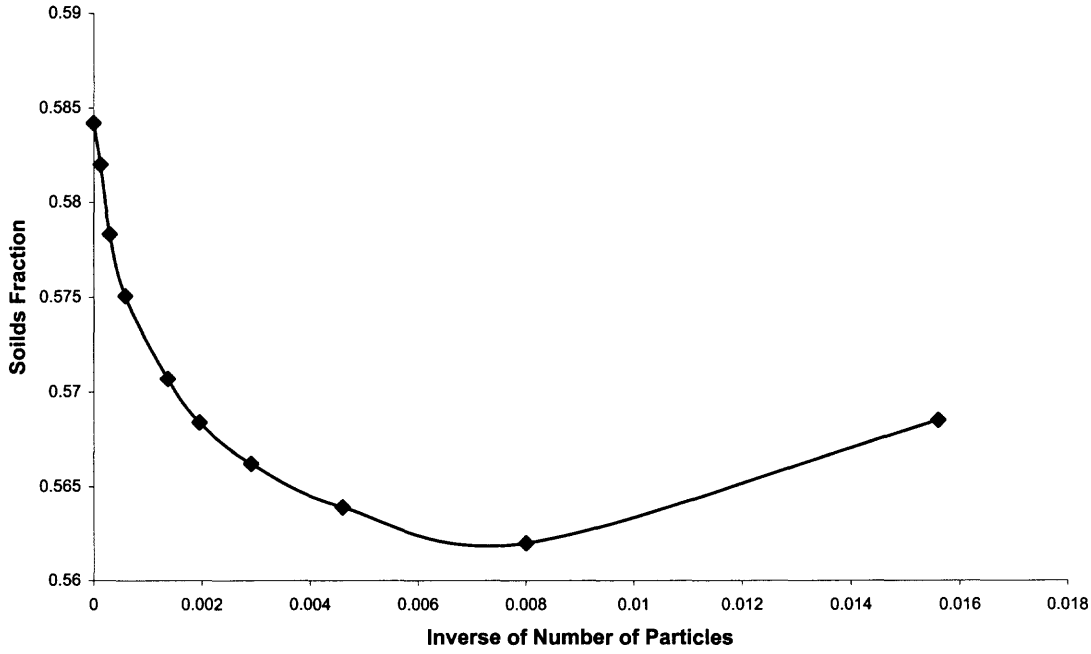


Figure 3.2 Solids fraction versus inverse of number of particles.

The solids fraction was plotted against the inverse of the number of particles and the curve was extrapolated to determine the solids fraction for infinitively large systems.

The packing fraction obtained for infinitively large systems (extrapolated) is 0.5842. The plot is shown in the Figure 3.1.

The solids fraction obtained for the system generated by Monte Carlo simulation was 0.582 for the 8000 particle system, which is in excellent agreement with Visscher and Bosterli [1] and is also in close agreement with Owe Berg [2], Tory[18].

CHAPTER 4

CO-ORDINATION NUMBER

4.1 Definitions and Overview

Co-ordination number n is defined as the number of spheres in contact with any given sphere. The value of n varies depending on packing arrangements. The table below shows the co-ordination number for various packing groups as well as the Packing volume fraction.

Table 4.1 Co-ordination Number and Solids Fraction

Packing Arrangement	Solids Fraction ν	*N*
Cubic	0.5236	6
Orthorhombic	0.6046	8
Tetragonal Sphenoidal	0.6981	10
Rhombohedral	0.7405	12

According to Smith, Foote and Busang [12], the regular closed packed hexagon array has the minimum porosity of 0.26 as compared to simple cubic array, which has the maximum porosity of 0.48. A random pack has porosity within the vicinity of 0.40.The loosest packing porosity was 0.447, obtained by pouring, and had the mean co-ordination number of 6.92. By shaking, the porosity was reported to be 0.440, 0.426 and 0.372 with corresponding mean co-ordination numbers of 7.34, 8.06 and 9.51. Table 4.2 below shows the results calculated from experimental data of Smith, Foote and Busang.

Table 4.2 Results from Smith, Foote and Busang [12]

Poured into a Glass Beaker	Porosity	**Mean Co-ordination Number**
Poured into a Glass Beaker	0.447	6.92
Poured and Shaken	0.440	7.34
Poured and Shaken	0.426	8.06
Shaken to Maximum Density	0.372	9.51
Added in small quantities and tapped intermittently	0.359	9.14

Wadsworth [13] studied the distribution of co-ordination number of smooth hard uniform spheres; the method of marking contacts was a modified version of Smith [12]. Wadsworth found that on and above the second layer there existed nearly perfect rhombohedral packing, which rapidly disappeared as the distance from the base increased. Wadsworth claimed that Wall effects are felt throughout the container and that there is no such thing as local wall effects. Wadsworth concluded that in cylindrical flat bottomed containers only one kind of close packing exists or tend to exists in random packing i.e. rhombohedral. The plots in Figure 4.1 below show the distribution of the number of contacts at the 2nd, 4th and 8th layers.

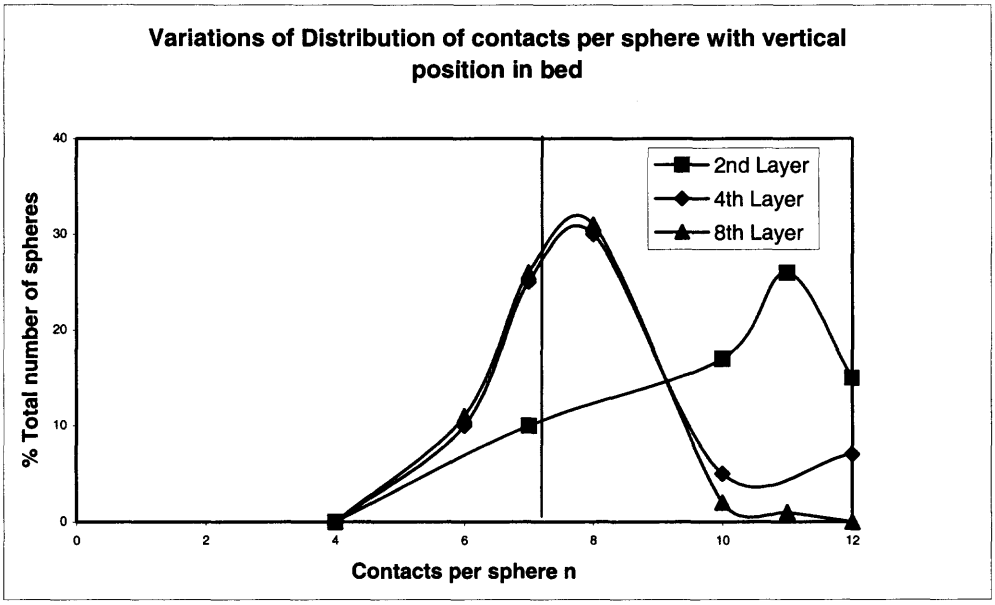


Figure 4.1 Co-ordination number distribution redrawn from Wadsworth [13].

Visscher [1] approached the problem of random packing by means of Monte Carlo simulation of the physical process of dropping spheres into a bin. A random structure with a density of 0.582 was generated, for which the average number of contacts per ball was computed to be 6.4.

Cheng [16] generated random structures of spherical particles in two dimensions by computer simulation as a result of using the distinct element method (DEM). It was observed that the co-ordination number varies from point to point. Some regions were found to have a co-ordination number of 6, while other regions had values as low as 2. The average co-ordination for 5 different regions was found to be 4.458, 3.665, 3.609, 3.397, and 4.31.

Goodling [17] used a simple technique to find the number of neighboring spheres that are in contact with any given sphere. The r, θ and Z location of each sphere was determined by photographic means. The co-ordination number was established under the

assumption that a contact point exists if two particles are within one ball diameter plus or minus 0.055 inch of one another. The distribution of co-ordination number is shown in Figure 4.2. It is observed that the upper and lower limits were 12 and 3 respectively. The bulk of spheres make contacts mostly with 7, 8, 9 and 10 other particles. The average number of contacts points for the entire bed was 8.1.

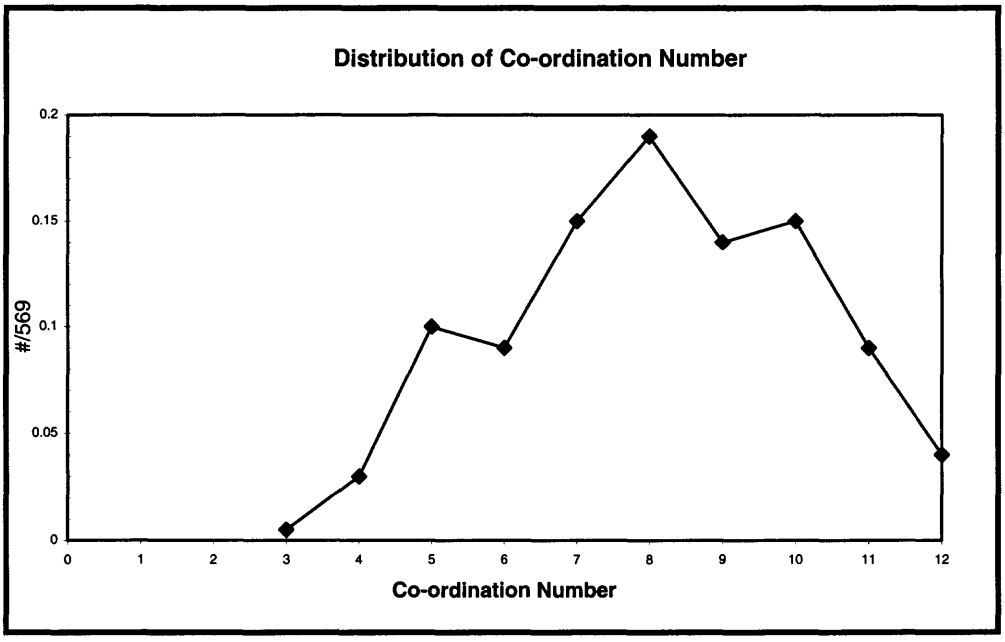


Figure 4.2 Co-ordination Number Distribution as redrawn from Goodling [17].

According to Nolan [11] the characteristics of random packing depend on interstices. The system is said to have loose packing if the mean interstice volume is at the maximum and if the mean interstice volume is at the minimum then the system is said to have close packing. Interstices were defined by Nolan, as a network of channels and not as isolated spaces. According to Nolan the interstice spaces are maximum when the packing is random in comparison to regular structures, which tend to have to less interstice spaces and hence maximum packing density. The average co-ordination

number varied from 4.4 to 5.9 as the system changed from loose packing to close packing.

4.2 Procedure and Results

A computer code was used to find the co-ordination number, i.e., number of contacts for each sphere. A contact is defined if two spheres, at 1.05*d* or less from each other (where *d* is the diameter of the particle). Using the code, contacts for all the particles were determined, and the average co-ordination number was also established. Values in the plots are averages taken over 25 ensembles that were run for each system size.

Figure 4.3a shows the co-ordination number distribution for a system with 1000 particles (d = 0.3"), where average was found to be 6.71291 ± 0.023433. Fig. 4.3b contains the sample deviation of the distribution over the 25 ensembles.

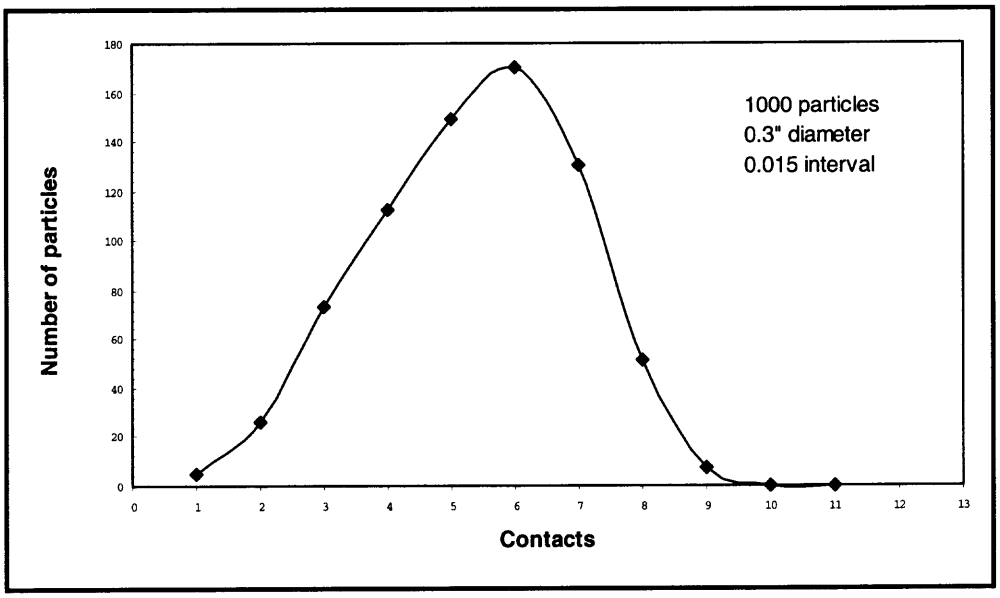


Figure 4.3a Co-ordination Number distributions for 1000 particle system.

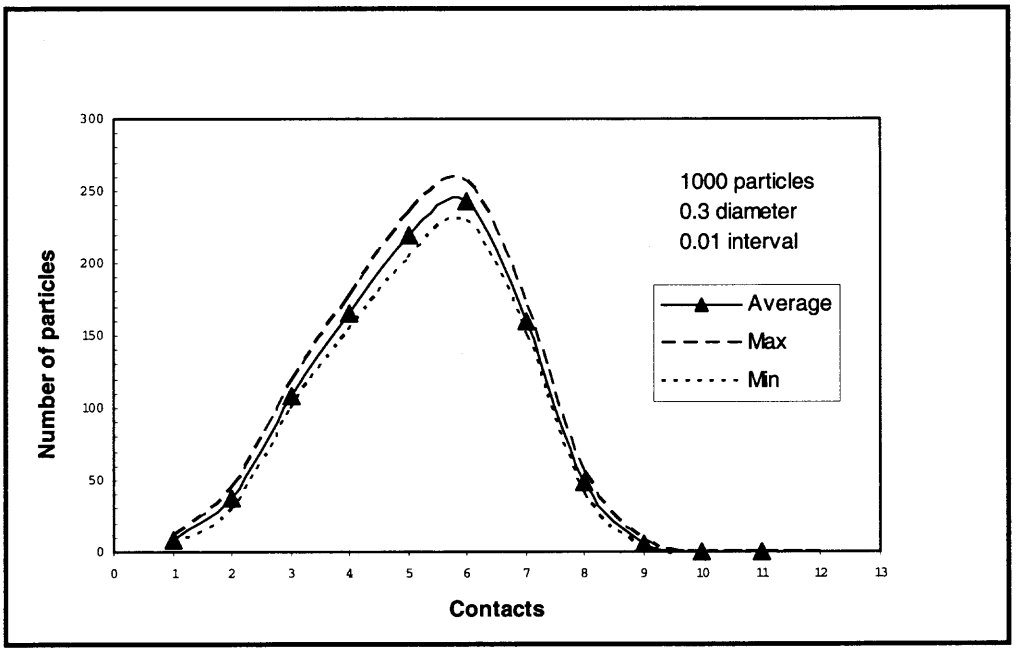


Figure 4.3b Sample deviation of co-ordination number distribution.

Results for the other systems (N = 729, 1728, 3375 and 8000) are shown in Figures 4.4, 4.5, and 4.6, respectively. The sample deviations for these systems were also computed as shown above in Figure 4.3b.

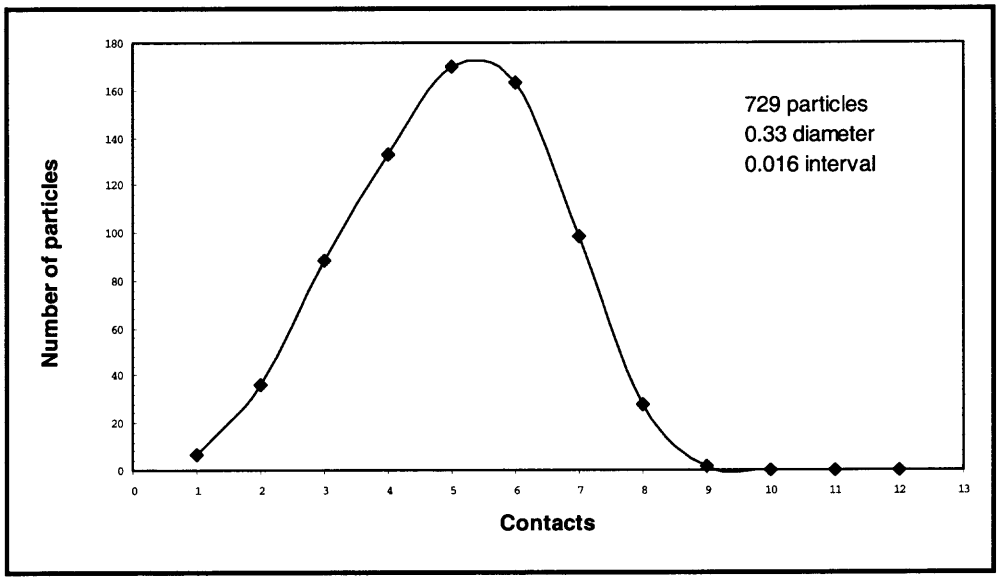


Figure 4.4 Co-Ordination Number distributions for 9^3 = 729-particle system.

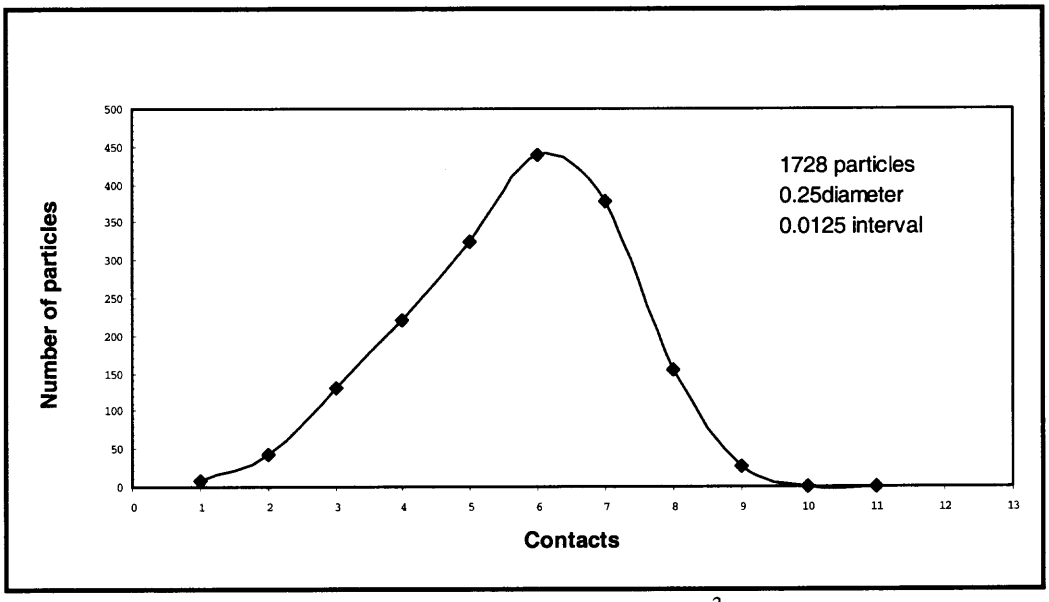


Figure 4.5 Co-Ordination Number distributions for $12^3 = 1728$ particle system.

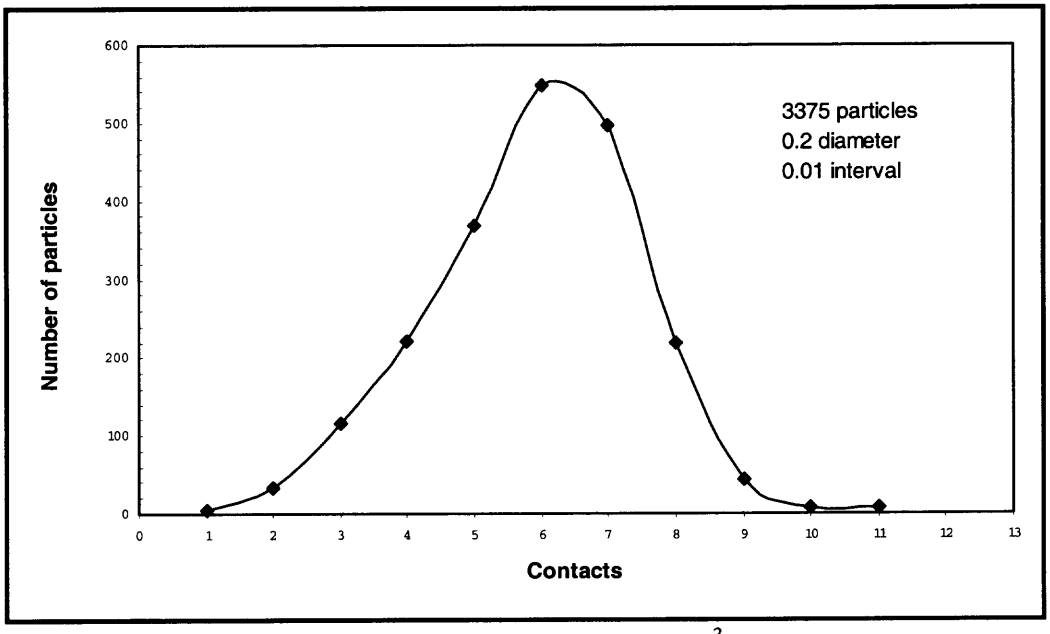


Figure 4.6 Co-Ordination Number distributions for $15^3 = 3375$ particle system.

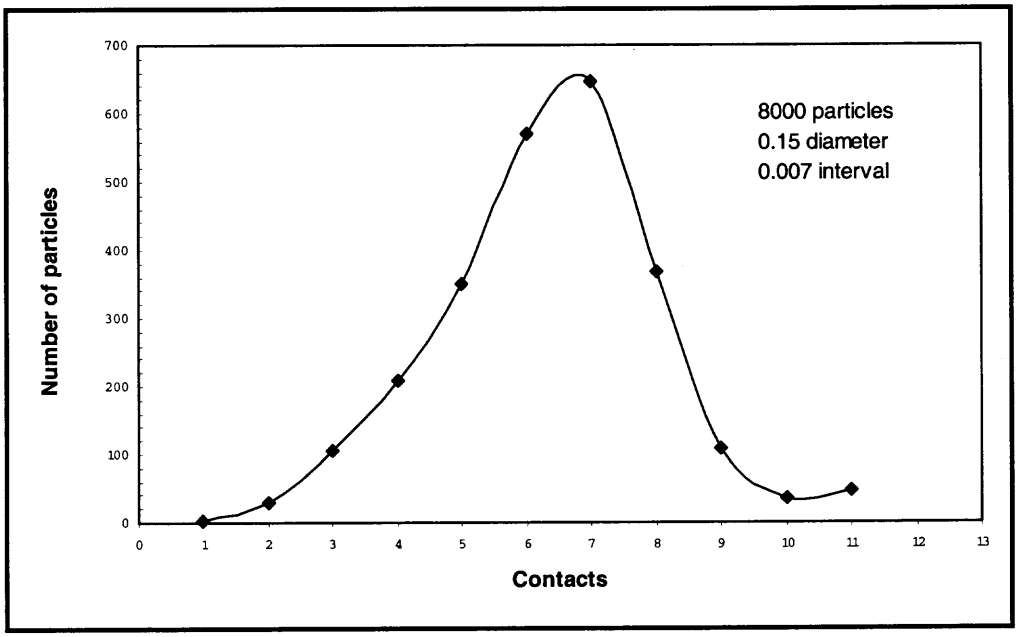


Figure 4.7 Co-Ordination Number distribution for $20^3 = 8000$ particle system.

The distribution of co-ordination number varies slightly from system to system. To understand how the size of the system affects its co-ordination number, all the distributions are plotted on a single plot, and as in Figure 4.9, it is observed that the peak, i.e., the highest number of particles with same co-ordination number, tends to shift slightly towards the right. This implies that as the system size increases there is an increase in the particles with higher co-ordination number, which in turn increases the average co-ordination number.

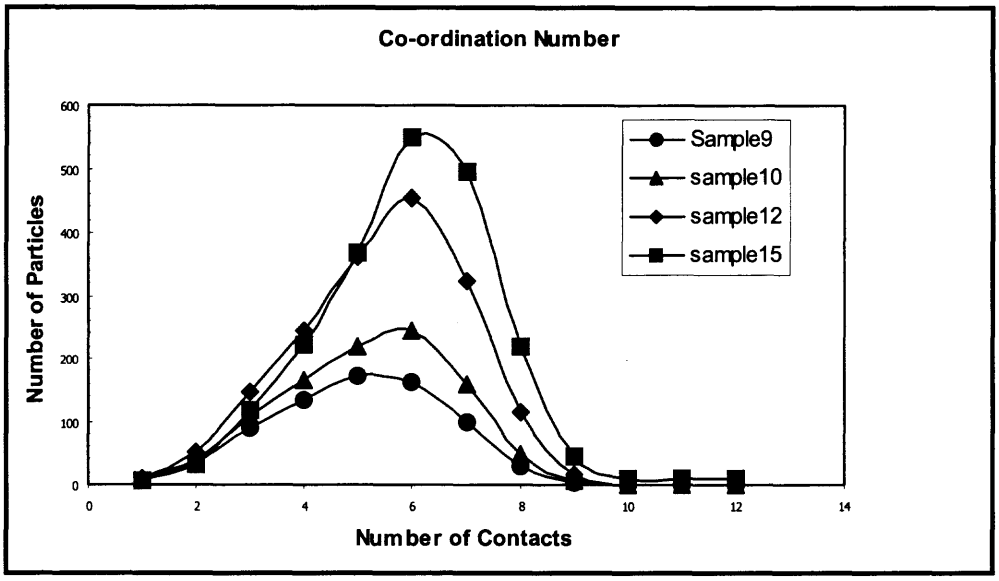


Figure 4.9 Co-ordination Number Distribution for Different Size systems.

The average co-ordination number for each system and is shown in the table below.

Table 4.3 Table Co-ordination Number for Different Size System

System Size	Average Co-ordination Number
729 Particles	5.34318 ± 0.0131
1000 Particles	5.52213 ± 0.00528
1728 Particles	5.66635 ± 0.01482
3375 Particles	5.97315 ± 0.01622
8000 Particles	6.71291 ± 0.023433

The Average co-ordination number of 6.71 for 8000-particle system is in good agreement with Smith [12], Visscher [1] and Nolan [11].

CHAPTER 5

RADIAL DISTRIBUTION FUNCTION

5.1 Definition and Overview

A better understanding of random packing of materials can be obtained from radial distribution function. *Radial Distribution function [3,4,5]* is defined as the number of particles as the function of distance from the origin. It is mathematically presented as

$$g(r) = \frac{Nav}{4\pi \times (r/D)^2 \times \Delta(r/D)}.$$

Where, Nav = Average number of sphere centers per interval

Δ (r/D): Interval

r : Radial distance

D: Sphere diameter

The radial distribution Function is normalized by dividing with $4\pi r^2$ to be constant over large 'r', where r is defined as the distance from the center of arbitrarily selected particle.

The importance of the study of the radial distribution function in the random packing of particles is that it serves as a model for liquids and helps to compare the experimental structure determination of real liquids [11]. Hence various researchers have studied the radial distribution of neighboring particles, using the radial distribution function.

A table showing the peak occurrence, which was found by various researchers, is shown below and comparison with the current values is also shown. The occurrence of the peaks in the current research is in close agreement with the values of [9, 11, and 14] as seen in the table below.

Table 5.1 Comparison with Published Data

	First Peak	Second	Third	Fourth	Fifth
Bennett [9]	1.00	1.73	2.68	3.53	4.38
Finney [6]	1.00	1.73	2.65	3.50	4.35
Matheson [5]	1.00	1.8	2.78	3.64	4.45
Scott [15]	1.00	1.83	2.64	3.45	-
Park [10]	1.00	1.9	2.7	3.5	4.5
Nolan [11]	1.00	1.975	2.65	-	-
Powell [14]	1.00	1.9			
Current	1.00	1.9	2.73	3.4	4.38

5.2 Procedure and Results

The distribution is plotted with respect to r/D; the plot is shown in the Figure 5.1.The radial distribution at intervals of 20% of the diameter of the particle was considered. The first peak occurs at the interval 1.0- 1.1, it is obvious that the first interval falls between these regions for the simple reason that no value of r/D can occur below 1.0, because the particles are rigid. That is, we start finding the centers of the particles only after the interval of 1.0-1.1 since there is no overlapping allowed in the simulated system.

The second peak occurs in the interval 1.9-2.0, which denotes that the second highest numbers of centers are found in this interval. Since it is random packing the values for second peak are less than 2.

In this analysis, for each system, 25 different configurations were simulated. The final value is the cumulative of all the 25 trials. Figure 5.2 shows the sample deviation for the system, showing the maximum and minimum value at that particular point.

A limiting value or critical value for the radial distribution function can be calculated from the previously determined packing fraction of the system, as in Scott [9]. The plot in figure 5.3 shows the limiting values for that system, which is .071. The limiting or critical value can be calculated by simple substitutions in radial distribution function.

The Figures 5.4, 5.5, 5.6 and 5.7 show the plot for 1000, 1728 and 3375 8000 particles respectively.

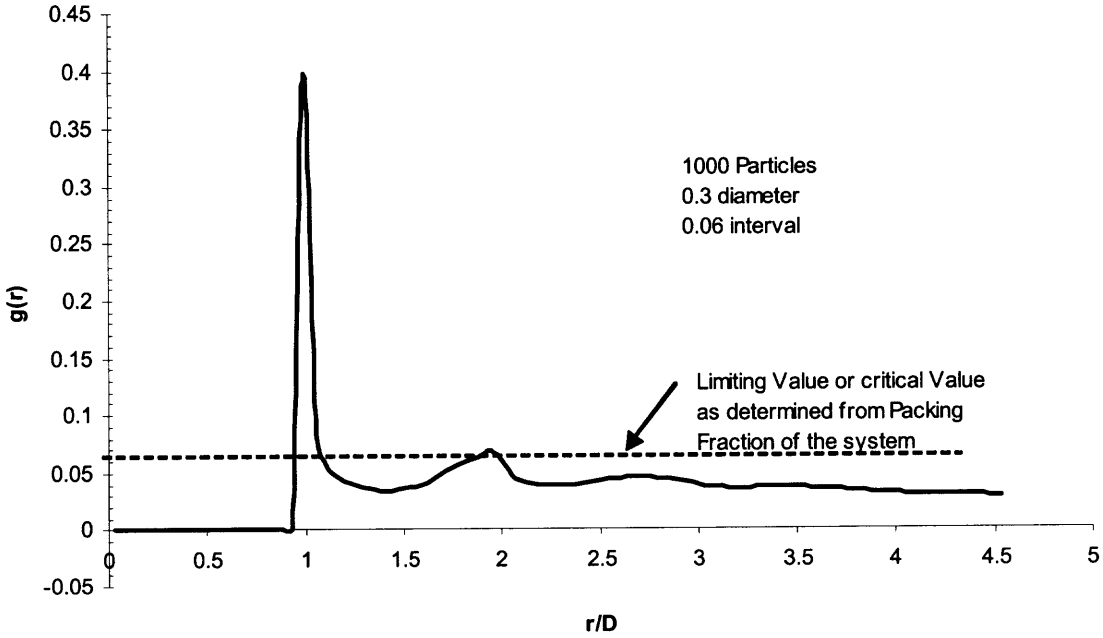


Figure 5.1 Radial distribution for $10^3 = 1000$ particles.

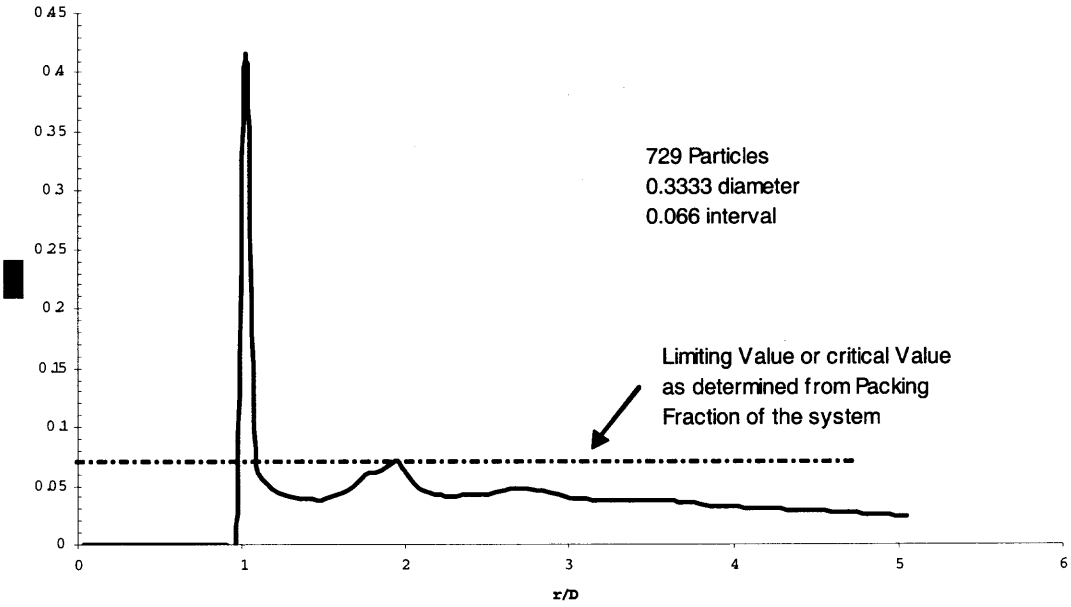


Figure 5.2 Radial distribution for $9^3 = 729$ particles.

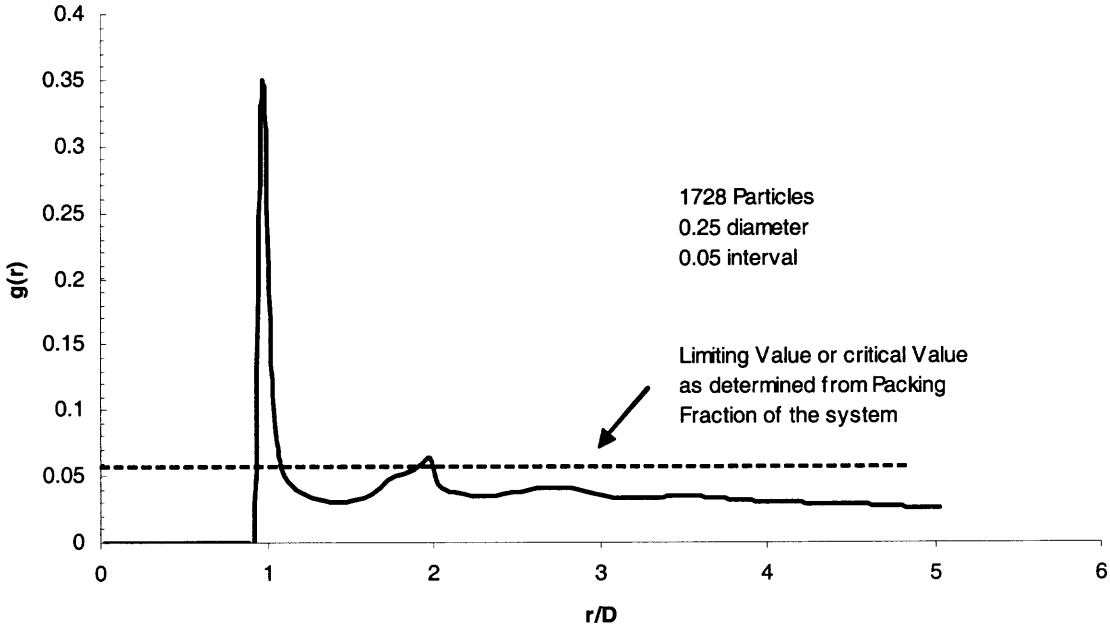


Figure 5.3 Radial distribution for $12^3 = 1728$ particles.

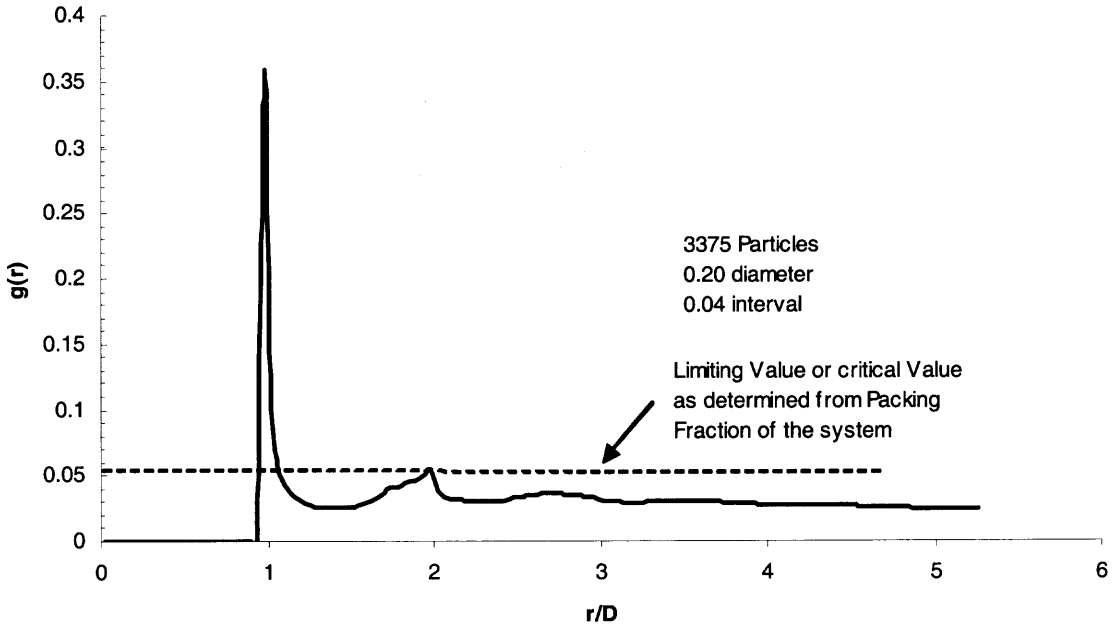


Figure 5.4 Radial distribution for $15^3 = 3375$ particles.

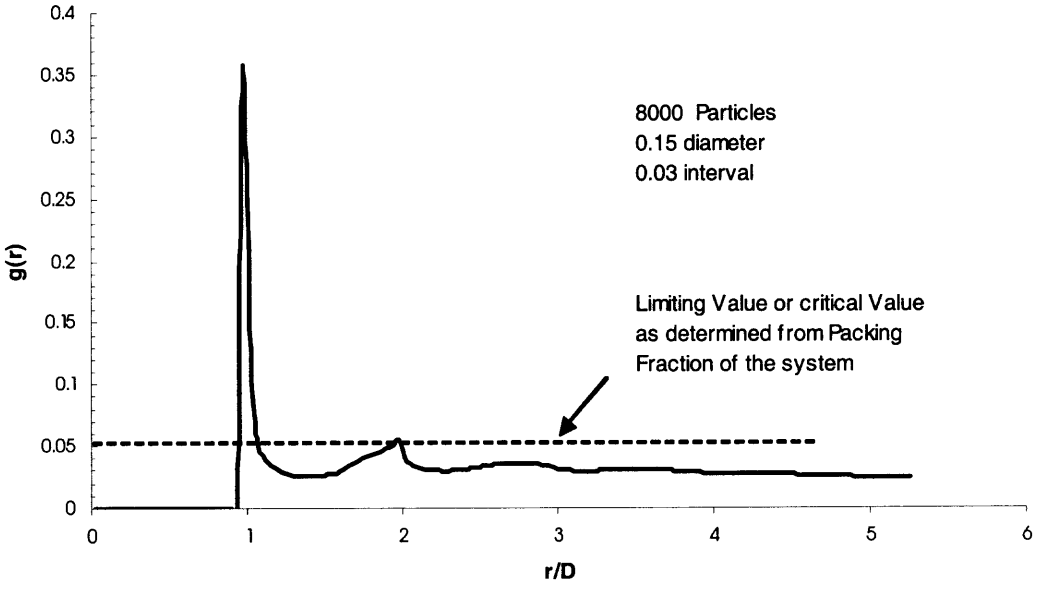


Figure 5.5 Radial distribution for $20^3 = 8000$ particles.

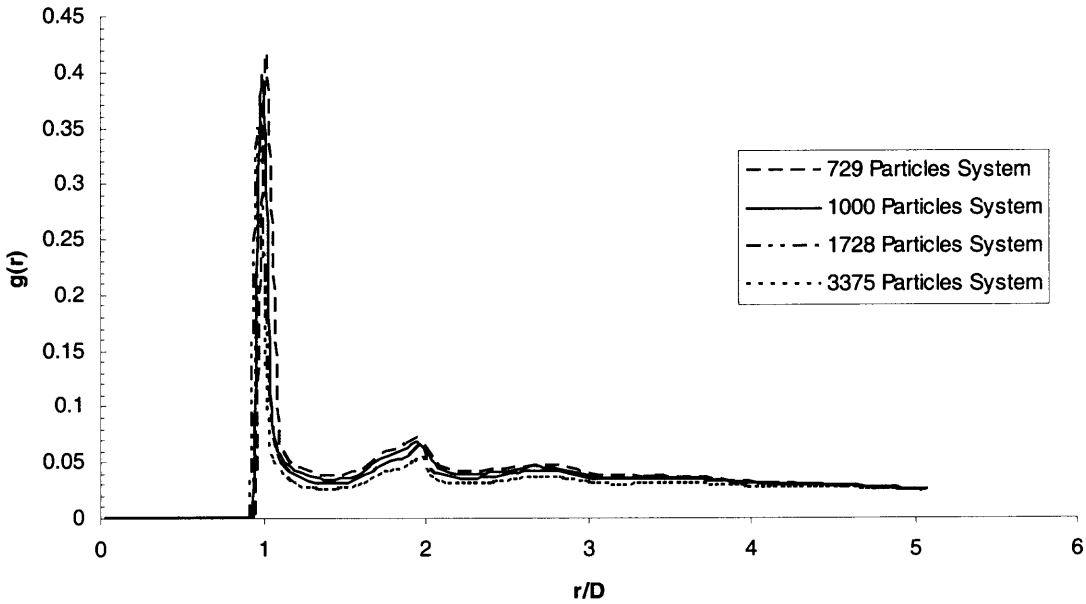


Figure 5.6 Radial distribution for different size systems.

The Figure 5.6 shows the radial distribution for different size systems on one plot. It is observed that there is not much difference in the location of the peaks as the system size increases.

The peaks in the radial distribution plots are in excellent agreement with the value of peaks obtained from previously published results of Scott [15], Nolan [11] and Powell [14].

CHAPTER 6

ANGULAR DITRIBUTION

6.1 Introduction

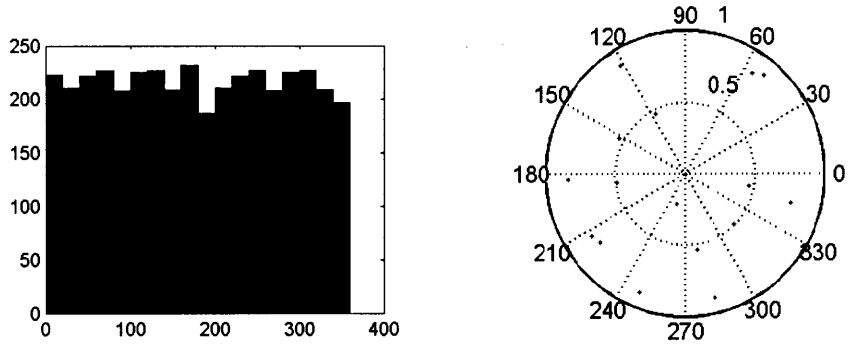
Angular contact distribution is the orientation of particles around its near neighbors. Angular contact distribution illustrates if any particular angle or direction is preferred by the particles in the arrangement under pouring conditions. In this chapter, we describe an attempt made to compute the contact angles of a particle with its near neighbors in the packing assembly generated by Monte Carlo simulation.

Contact angles of a particle are defined by a set of angles made by its near neighbors in horizontal and vertical plane. The angle made in the horizontal plane θ is defined from 0 to 2π. The angle made in the vertical plane Φ is defined from 0 to π in the vertical plane. Both θ and Φ are necessary to describe the position of near neighbors of a particle.

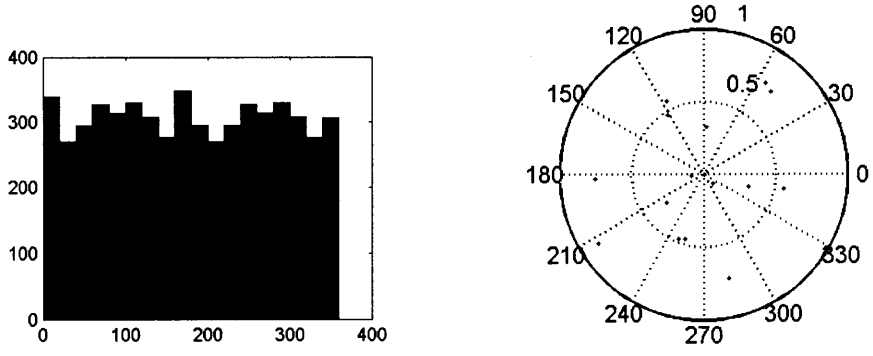
6.2 Procedure and Results

A computer code was used to compute the contact angles of the near neighbors. All the particles that are within a center distance of $1.05d$ (where'd' is the diameter of particles) from the center of particle in consideration are considered to be in contacts. The gap ($0.05d$) selected here is concurrent with the gap selected for computing co-ordination number. If the particles were determined to be contacting each other, then the angle made by the line joining centers of the particles with the horizontal plane, in which the center of the initial particle lies, 'θ' and the angle made with the vertical plane 'Φ', in clockwise direction are computed. The ordered pair (θ, Φ) gives the orientation of the contact

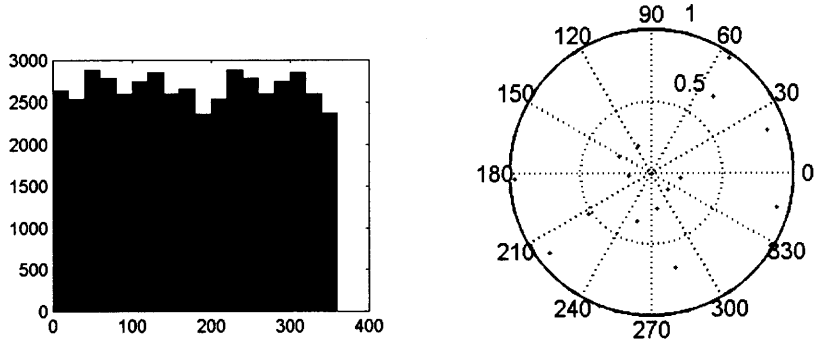
particle with respect to the initial particle. The contact angles for all the contacts were computed in a similar fashion. The plots below show the histograms for the contact angles for different size systems.



(a) $9^3 = 729$ particle system.

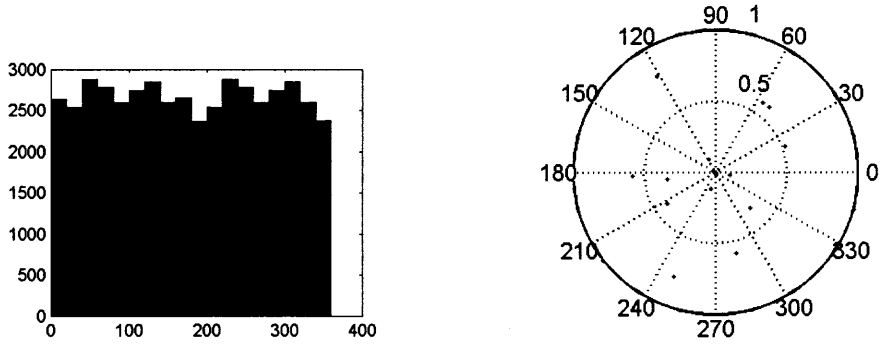


(b) $10^3 = 1000$ particle system

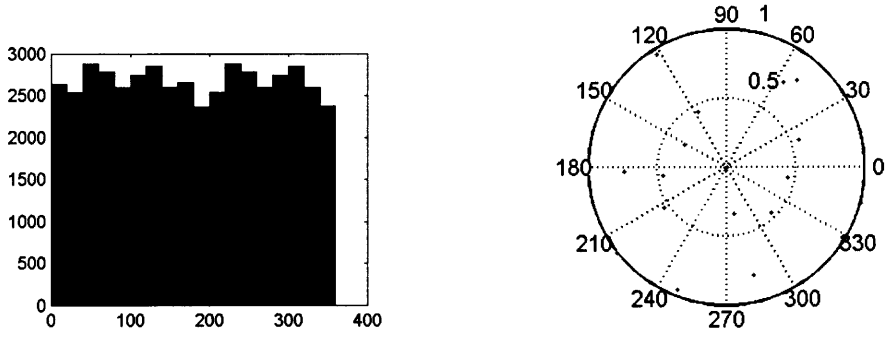


(c) $12^3 = 1728$ particle system.

Figure 6.2 Distributions of angle θ



(d) $5^3 = 3375$ particle system.

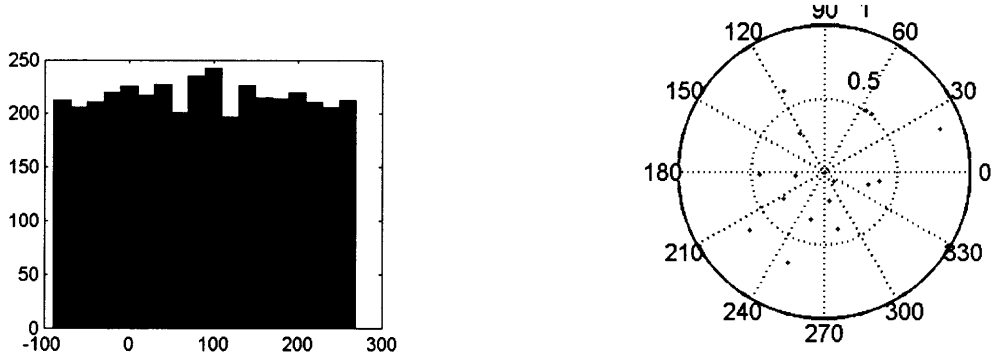


(e) $20^3 = 8000$ particle system

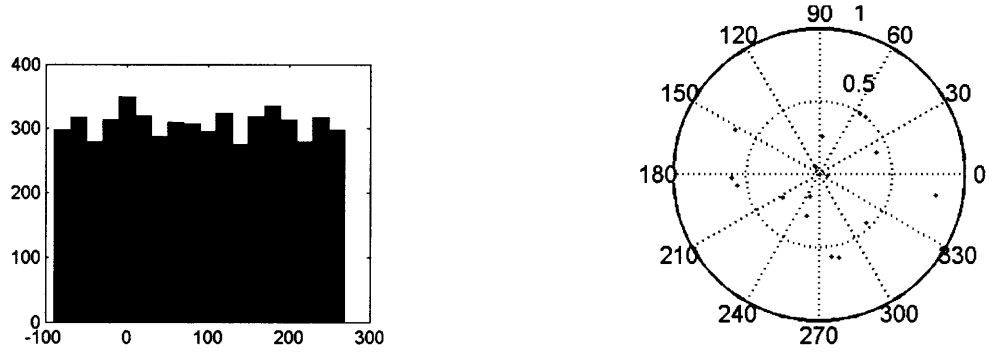
Figure 6.2 Distributions of angle θ (Continued).

The histograms and polar plots in the Figure 6.1(a-d) give the distribution of contact angle θ for different size systems. It is observed that there are no appreciable peaks or valleys in the histograms. This suggests that the particles are evenly distributed around their neighbors. The polar plots re-affirm the same conclusion. It was concluded from the histograms that the there is no preferred angle for near neighbors in the horizontal plane.

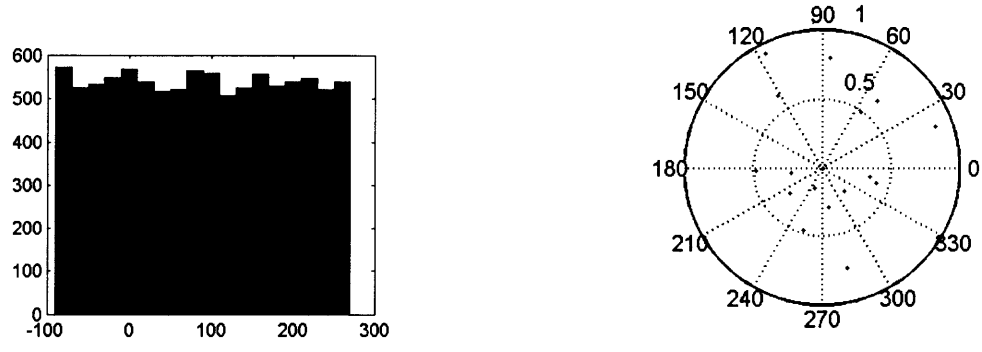
The distribution of angle made in the vertical plane Φ were also plotted in histograms and polar plots as shown in Figure 6.2(a-e).



(a) $9^3 = 729$ particle system.

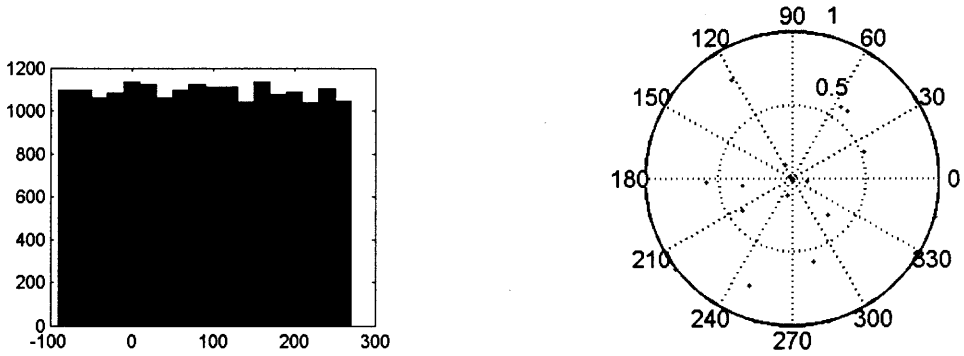


(b) $10^3 = 1000$ particle system

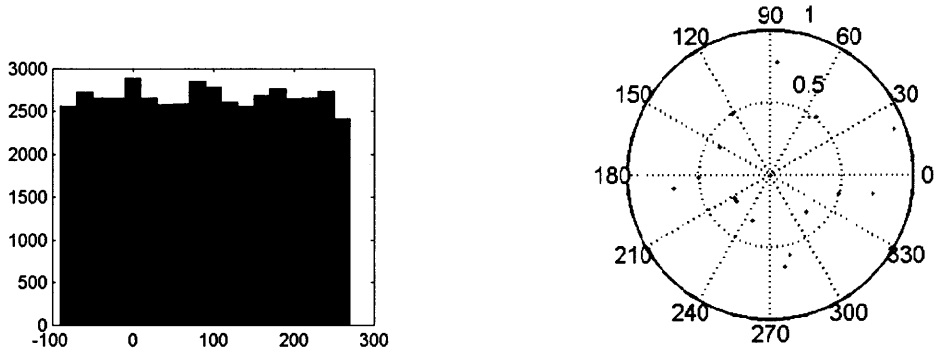


(c) $12^3 = 1728$ particle system.

Figure 6.2 Distributions of angle Φ



(d) $5^3 = 3375$ particle system.



(e) $20^3 = 8000$ particle system

Figure 6.2 Distributions of angle Φ (Continued).

The angular distributions of Φ show similar characteristics as seen in θ distributions. There were no appreciable peaks observed in the histograms as seen in the Figure 6.2 (a-d). This indicates to show that the Φ angles made by near neighbors were evenly distributed around the particle.

It was thus concluded that the particle assembly generated by Monte Carlo simulation was randomly arranged with no preferred angles in the arrangement of particles.

CHAPTER 7

FABRIC TENSOR

7.1 Introduction

The term 'fabric' [25] is generally used to denote the spatial arrangement of particles and associated voids. Local geometry and direction of each contact is important. The assembly of these unit vectors at N_c contact point's stands for the packing characteristic among neighboring particles. To determine the fabric characteristics for particular particle with N_c contact points, the unit vectors are introduced in the contact direction. The sum of the products of the directional cosines over the number of contacts gives the index measure representing the fabric characteristics for that particle. The average over many contacts over many particles gives the Fabric Tensor. This can be mathematically be represented as

$$\Phi_{ij} = \frac{1}{N_p} \sum_{k=1}^{N_p} \left\{ \frac{1}{N_c^{(k)}} \sum_{m=1}^{N_c^{(k)}} n_i^{(m)} n_j^{(m)} \right\}$$

Where	Φ_{ij}:	Fabric Tensor
	N_p:	Number of Particles
	N_c:	Number of contacts for a particle
	n_1, n_2 and n_3 :	The directional cosines of contacts.

The figure below shows the contacts normal for a particle 'P' with its neighbors.

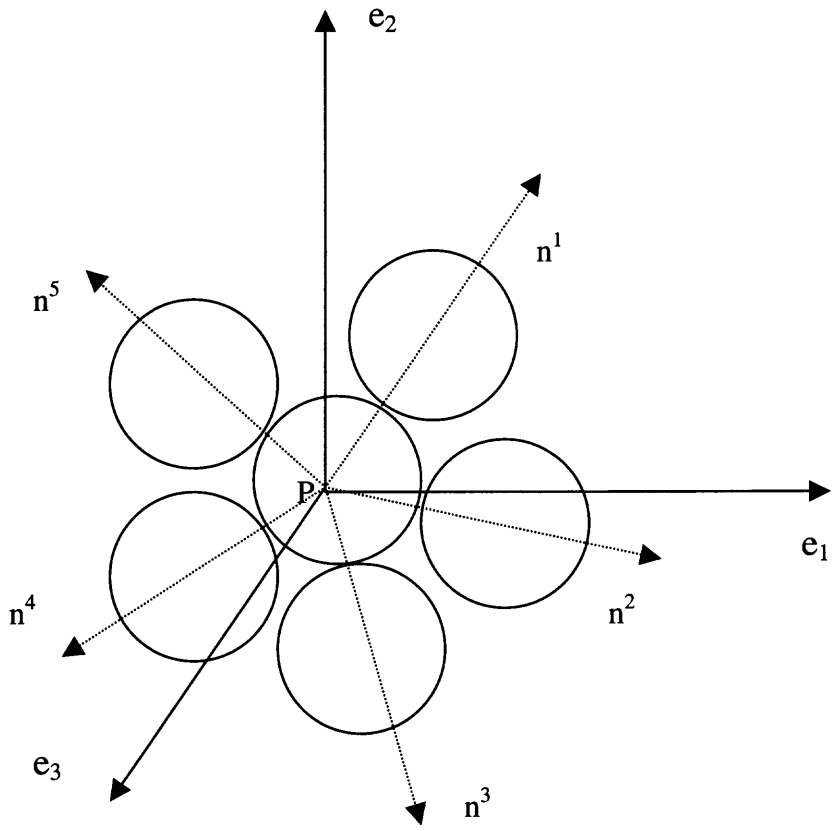


Figure 7.1 Orientation of contact normal for a particle P.

The co-ordinate axes are shown as e_1, e_2 and e_3. The contacts normal are n^1, n^2 and so on. Now the directional cosines are defined as the cosine of the angle between the contact normal and the co-ordinate axes. For the contact normal n^1 the directional cosines are

$$n_1 = \cos(n^1, e_1)$$

$$n_2 = \cos(n^1, e_2)$$

$$n_3 = \cos(n^1, e_3)$$

In similar fashion the directional cosines were determined for the other contacts. The summation of the product of directional cosines over the number of contacts gives the local fabric tensor. By summation of these local fabrics tensor over the number of particles gives the global fabric tensor.

7.2 Procedure and Results

Using the procedure described in the previous section to compute global fabric tensor for a packing assembly, a computer code was developed. The fabric tensor was computed for different size system of particles (N= 729, 1000, 1728, 3375 and 8000) generated by Monte Carlo simulation. A typical fabric tensor for 8000 particle assembly is shown below.

$$\Phi = \begin{pmatrix} 0.5414 & 0.000239 & -0.000178 \\ 0.000239 & 0.5390 & -0.000147 \\ -0.000178 & -0.000147 & 0.5406 \end{pmatrix}$$

It was observed that the fabric tensor for this system was a symmetric matrix with the diagonal elements being almost equal and the off diagonal elements being close to zero (almost negligible). The fact that the diagonal elements are equal shows that there is no preferred direction for the orientation of particles, and off diagonal elements being close to zero indicates the same. The contacts for the particles in Monte Carlo simulated bed have no preferred direction. The eigenvector [20] for the above fabric tensor was found to be

$$V = \begin{pmatrix} 0.6746 & -0.9795 & 0.3144 \\ -0.7382 & -0.0974 & -0.0415 \\ 0.0064 & 0.1746 & 0.9484 \end{pmatrix}$$

The eigenvalues [20] of the fabric tensor are the principal fabrics corresponding to which there are three sets of direction cosines, which establish the relationship of the principal planes to the origin of non principal axes. The eigenvalues are listed below

$$D = \begin{pmatrix} 0.5388 & 0 & 0 \\ 0 & 0.5417 & 0 \\ 0 & 0 & 0.5405 \end{pmatrix}$$

It is also observed that the sum of the eigenvlaues [20] is equal to the sum of the diagonal elements of the tensor. The principal fabrics obtained are independent of the orientation of the original co-ordinate system.

Eigenvalues obtained for the above fabric are almost equal; this shows that all the planes in the system of packing assembly are principal planes. This further means that the system of packing is isotropic.

A comparison of the computed fabric tensor for the current simulation is made with that calculated from an assembly of spheres generated via the "discrete element method" [27]. Basically, this method involves solving the equations of motion of a system of spheres that interact via idealized contact models. The integration time step for this method was of the order of 10^{-6} seconds. This is in contrast to the Monte Carlo method used in this work in which a time scale is absent. Two DEM data sets e were

analyzed. The first was obtained by pouring a system of 8000 spheres into a rectangular box having solid sidewalls, for which the aspect ratio L/d=10. The second set was generate by vibrating the poured assembly under sinusoidal displacement amplitude a/d =0.16 for a duration of three seconds and relaxing for one second. The fabric tensor after pouring for 1 second is represented by Φ_p and the fabric tensor for the system after vibration is represented by Φ_v. The component Φ_{22} for the system generated by DEM was observed to be 1/3 of the rest of the diagonal components, showing that there was a preferred direction for this system in y-direction.

$$\Phi_p = \begin{pmatrix} 0.4277 & 0.000123 & 0.00159 \\ 0.000123 & 0.14421 & 0.000692 \\ 0.00159 & 0.000692 & 0.42833 \end{pmatrix}$$

$$\Phi_v = \begin{pmatrix} 0.42799 & 0.0000137 & 0.001578 \\ 0.0000137 & 0.14367 & 0.000563 \\ 0.001578 & 0.000563 & 0.42833 \end{pmatrix}$$

The computer code was equipped with a sub-routine to calculate the fabric tensor when the original co-ordinate axis is rotated. The original co-ordinate axes were rotated by 30° and the fabric tensor was computed. As expected from the previous results, there was no change in the fabric tensor; the conclusion is that the system is isotropic. The fabric tensor for the new co-ordinate system obtained was

$$\Phi'' = \begin{pmatrix} 0.5404 & 0.000944 & -0.000248 \\ 0.000944 & 0.5390 & -0.000454 \\ -0.000248 & -0.000454 & 0.5409 \end{pmatrix}$$

The deviatoric part of the fabric tensor, denoted by Φ', is a measure of the degree of anisotropy. It is defined as,

$$\Phi' = \begin{pmatrix} \varphi_{11} - \varphi_m & \varphi_{12} & \varphi_{13} \\ \varphi_{21} & \varphi_{22} - \varphi_m & \varphi_{23} \\ \varphi_{31} & \varphi_{32} & \varphi_m \end{pmatrix}, \text{ where } \varphi_m \equiv \frac{1}{3}(\varphi_{11} + \varphi_{22} + \varphi_{33})$$

Here is the first fabric invariant. For a perfect isotropy, $\Phi' = [0]$. The computation of Φ' below shows relatively small diagonal components and even smaller off-diagonal values; and therefore, it is concluded that the degree of anisotropy is negligible, and that the structure of the Monte Carlo-generated assembly is nearly isotropic.

$$\Phi' = \begin{pmatrix} 0.001 & 0.000239 & -0.000178 \\ 0.000239 & -0.0013 & -0.000147 \\ -0.000178 & -0.000147 & 0.0003 \end{pmatrix}$$

The analysis of fabric tensor shows that all the planes in the packing assembly, generated by the Monte Carlo simulation, are principal planes. Similar situation exists in ideal fluid, in which there are no shearing stresses, and also hydrostatic pressure (such as submerged bodies). This emphasizes that the packing assembly generated by Monte Carlo simulation could serve as model for ideal fluids and hydrostatic stresses.

CHAPTER 8

SUMMARY AND CONCLUSIONS

The study of random packing of particles interest many researchers because of the scientific importance and practical applications. Many researchers have developed numerous methods to describe and simulate packing of spheres. This involved packing of spheres, of equal, unequal sized and non-spherical particles.

In this work the packing assemblies of mono-sized spheres generated by Monte Carlo simulation code were analyzed. Different size packing assemblies were simulated varying from 729-particle to 8000-particle system. These packing assemblies were analyzed for packing properties like solids fraction, co-ordination number, radial distribution function, angular distribution of contact angles and fabric tensor. The results and conclusions of the above properties are shown here.

Solids fraction is the ratio of volume particles to the number of particles. It is the property that shows how close the particles are packed in a packing assembly. The solids fraction in this study was computed using plane growth method, which involves dividing the entire length of packing bed into different intervals and computing the solids fraction at end of those intervals. The solids fraction at 80% of the length of the packing bed was considered to be the solids fraction of the entire assembly. The solids fraction obtained for the system generated by Monte Carlo simulation was 0.582 for the 8000-particle system, which is in excellent agreement with Visscher and Bosterli [1] and is also in close agreement with Owe Berg [2], Tory [18].

Co-ordination number is the number of spheres in contact with any given sphere. It is the property of packing assembly that shows the number of near neighbors for a particle. A computer code was used to find the co-ordination number, i.e., number of contacts for each sphere and also the average of the co-ordination number over the entire packing. The Average co-ordination number of 6.71 for 8000-particle system is in good agreement with Smith [12], Visscher [1] and Nolan [11].

A better understanding of random packing of materials can be obtained from radial distribution function. Radial distribution function is the number of particles as the function of distance from the origin. The importance of radial distribution function is that, it helps compare random packing with ideal liquids. The peaks in the radial distribution plots are in excellent agreement with the value of peaks obtained from previously published results of Scott [15], Nolan [11] and Powell [14].

Angular contact distribution is the orientation of particles around its near neighbors. Angular contact distribution illustrates if any particular angle or direction is preferred by the particles in the arrangement under pouring conditions. Contact angles of a particle are defined by a set of angles made by its near neighbors in horizontal and vertical plane. Analysis of these angles concluded that the particle assembly generated by Monte Carlo simulation was randomly arranged with no preferred angles in the arrangement of particles.

The other property of packing is the fabric tensor; it is generally used to denote the spatial arrangement of particles and associated voids. Local geometry and direction of each contact is important. It was observed that the fabric tensor for this system was a symmetric matrix with the diagonal elements being almost equal and the off diagonal

elements being close to zero (almost negligible). The eigenvalues for the fabric tensor were found to be equal. The deviatoric part of the fabric tensor, which measures the degree of anisotropy for packing system was found to be negligible for this packing assembly. The analysis of fabric tensor shows that all the planes in the packing assembly, generated by the Monte Carlo simulation, are principal planes. It was concluded that the packing assembly generated by Monte Carlo simulation could serve as model for ideal fluids and hydrostatic stresses.

It would be interesting to subject the packing assembly obtained from pouring to shaking under different amplitudes and study the packing properties and relating these properties to those of simple liquid. Study of vornoi polyhedra could be done as extension of this thesis. These could be considered for further study.

APPENDIX A

ALGORITHMS

The algorithm for the Monte Carlo simulation method along with the algorithms for the codes used to compute the various properties of packing like solids fraction, co-ordination number, radial distribution function, angular distribution and fabric tensor are listed here.

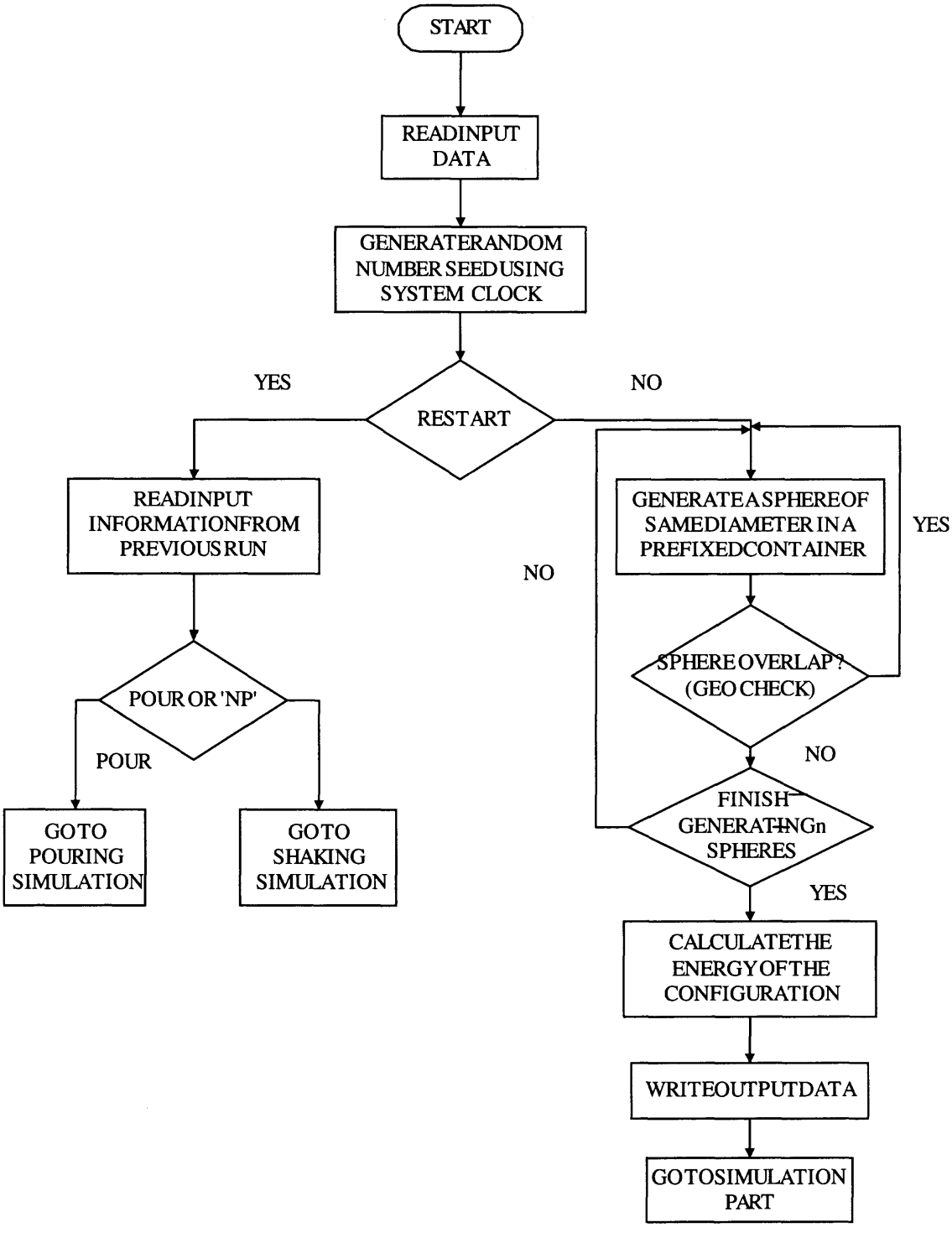


Figure A.1 Algorithm for the initial configuration for Monte Carlo simulation method.

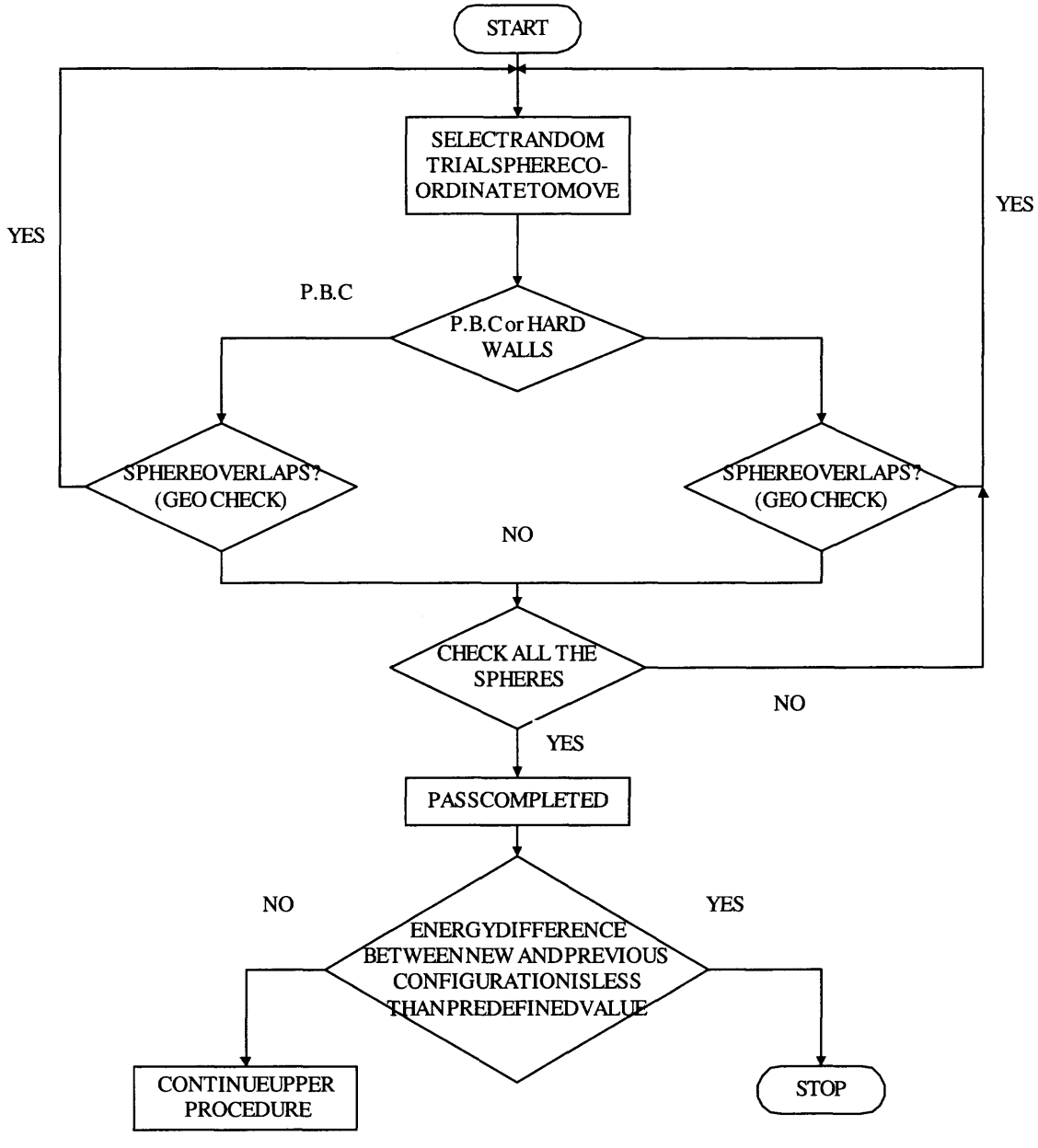


Figure A.2 Algorithm for Monte Carlo simulation of pouring.

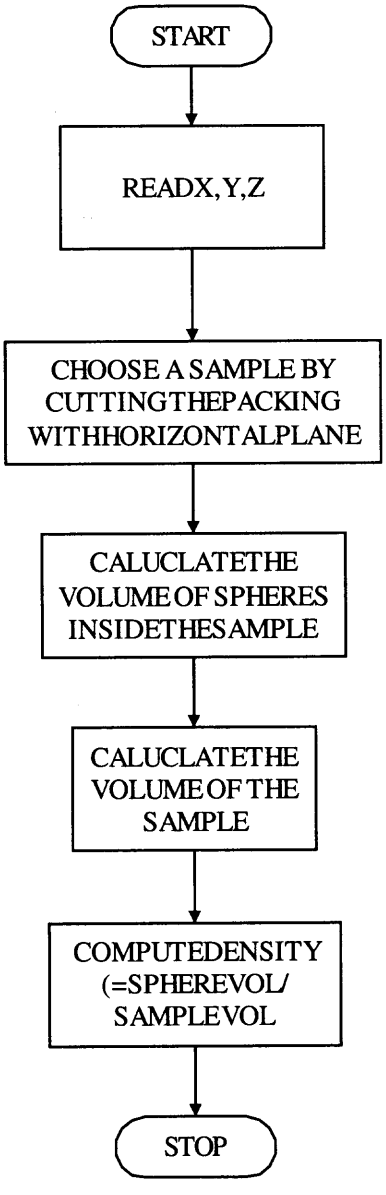


Figure A.3 Algorithm for plane growth method for solids fraction.

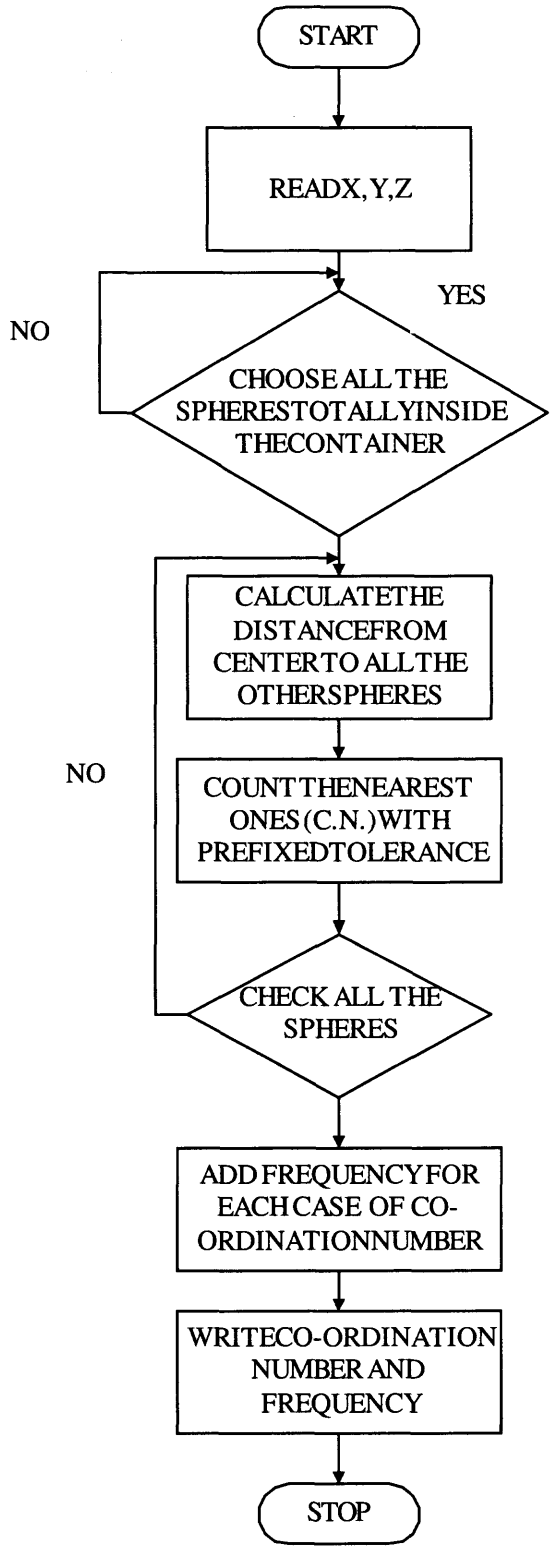


Figure A.4 Algorithm for co-ordination number.

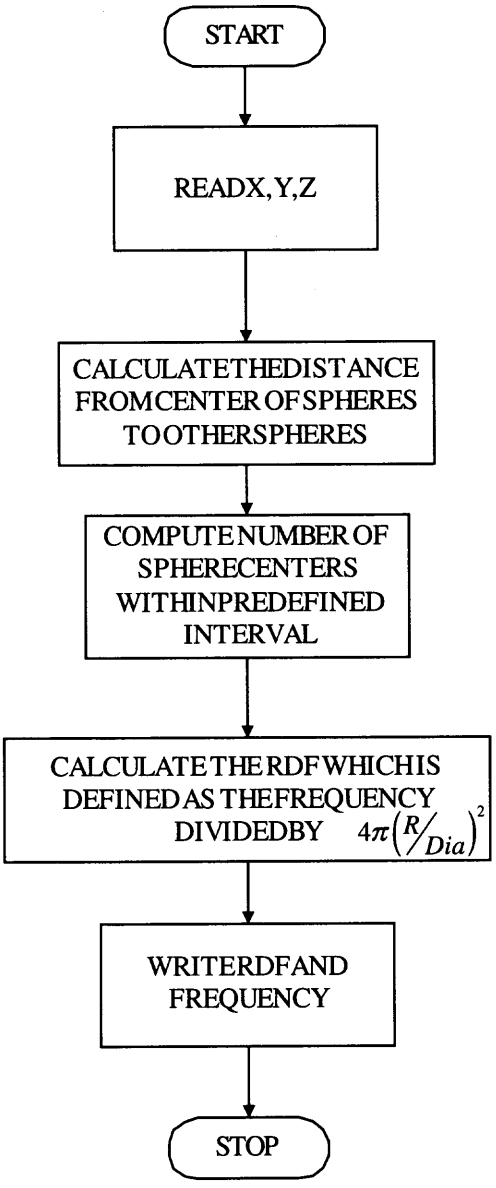


Figure A.5 Algorithm for radial distribution function.

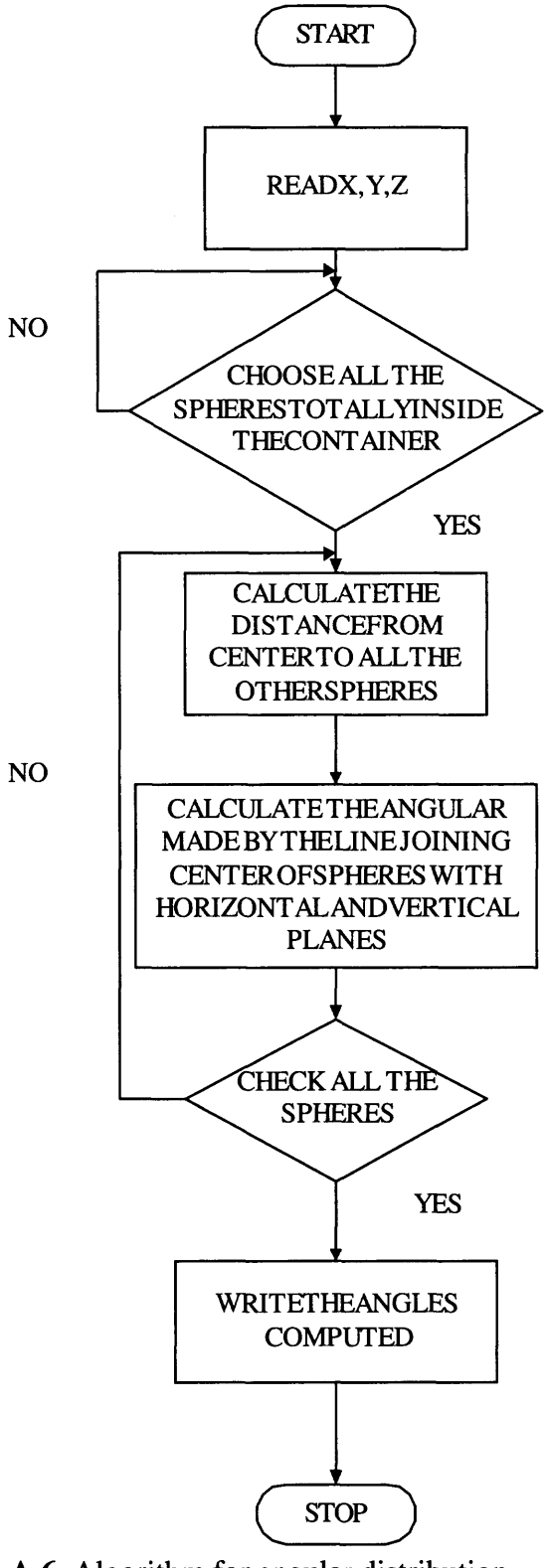


Figure A.6 Algorithm for angular distribution.

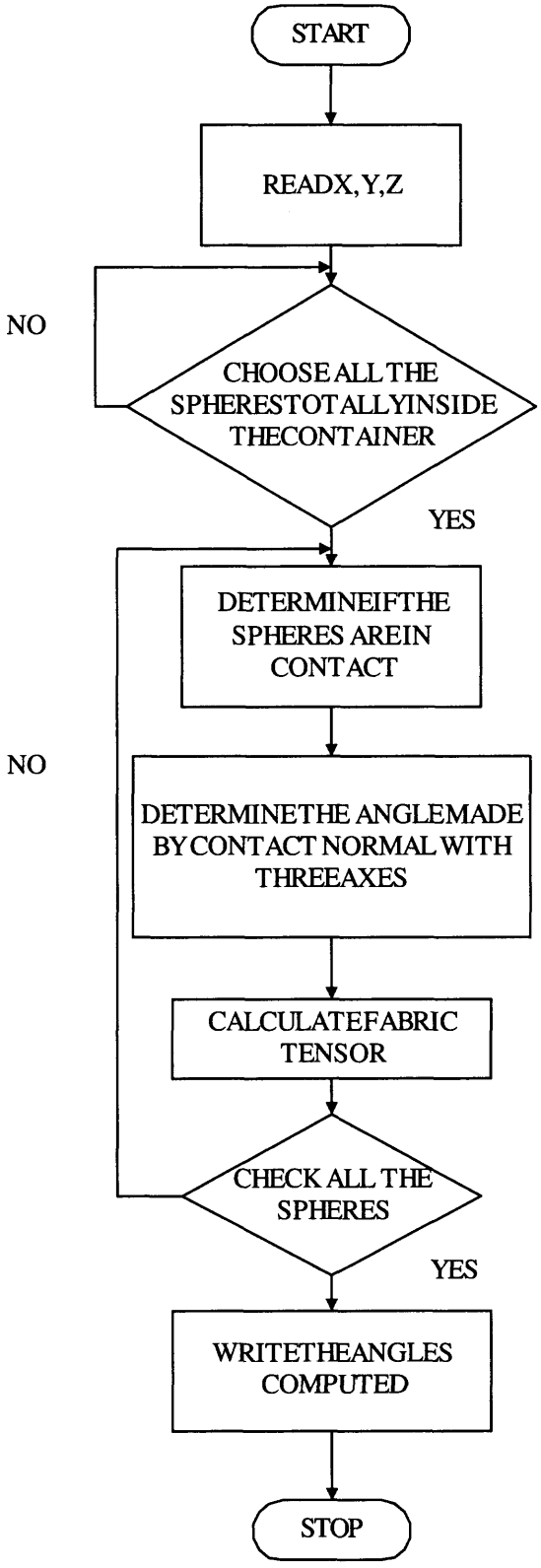


Figure A.7 Algorithm for fabric tensor.

APPENDIX B

B.1 Plane growth method

The code below computes the solids fraction for a packing assembly using plane growth method. The output file 'Solids-Fraction' contains the solids fraction at each interval.

```
c=================================================================
c       PROGRAM TO CALCULATE THE CUMULATIVE PACKING DENSITY OF
c       PARTICLE BED
c
=================================================================
c       case1.plo           The file containing the co-ordinates of the particles in the bed
c       Solids Fraction     Output file containing the solids fraction
c       zones           Number of intervals into which the height of bed is divided   into
c       dia                     Diameter of the particles
c
=================================================================
c                       BEGINNING OF PROGRAM
c ================================================================

      INTEGER I,J,K,P,Q,J,N1(100,30),N2(100,30),N3(100,30)
      INTEGER zones,particle_number,ifile

      REAL YMAX(100),rad,dia,PI,delta,h
      REAL YMIN(100),XMAX(100),XMIN(100),ZMAX(100),ZMIN(100)
      REAL X(100,16000),Y(100,16000),Z(100,16000),VOL(100,30)
      REAL VOL1(100,30),VOL2(100,30),VOL3(100,30),PF(100,30)
      REAL SUM(100),AVERAGE(100),no_files

       half = 0.50d0
c
      WRITE (6,100)
      READ (5,*) particle_number
      WRITE (6,120)
      READ (5,*) no_files
      WRITE (6,135)
      READ (5,*) dia
      WRITE (6,137)
      READ (5,*) zones
c
      OPEN (13,FILE='case1.plo',STATUS='OLD')
      OPEN (7,FILE='Solids-Fraction',STATUS='NEW')
c
```

```
      OPEN (13,FILE='case1.plo',STATUS='OLD')
      OPEN (7,FILE='Solids-Fraction',STATUS='NEW')
c
c     ********* Loop over the number of data files *********
      ifile = 1
c
 576  format(3x,'ifile = ', I6, 5x,'no_files = ',I6)
c
 37   If (ifile .gt. no_files) Goto 63
      q = ifile
c
      DO 50 i=1,particle_number
        READ  (13,*) X(Q,i),Y(Q,i),Z(Q,i)
50    CONTINUE
c
c
      XCELL = 3.0d0
      ZCELL = 3.0d0
      rad= dia * half
      PI=3.1415926

C OBTAINING THE MAXIMUM Y VALUE
      XMAX(Q)=X(Q,1)
      YMAX(Q)=Y(Q,1)
      ZMAX(Q)=Z(Q,1)
      XMIN(Q)=X(Q,1)
      YMIN(Q)=Y(Q,1)
      ZMIN(Q)=Z(Q,1)
      SUM(Q)=Y(Q,1)
c
      DO 55 J=2,particle_number-5
        SUM(Q)=SUM(Q)+Y(Q,J)
        IF (Y(Q,J).GT.YMAX(Q)) YMAX(Q)=Y(Q,J)
        IF (Y(Q,J).LT.YMIN(Q)) YMIN(Q)=Y(Q,J)
        IF (X(Q,J).GT.XMAX(Q)) XMAX(Q)=X(Q,J)
        IF (X(Q,J).LT.XMIN(Q)) XMIN(Q)=X(Q,J)
        IF (Z(Q,J).GT.ZMAX(Q)) ZMAX(Q)=Z(Q,J)
        IF (Z(Q,J).LT.ZMIN(Q)) ZMIN(Q)=Z(Q,J)
55    CONTINUE
c
      YMAX(Q)=YMAX(Q)+rad
      XMAX(Q)=XMAX(Q)+rad
      ZMAX(Q)=ZMAX(Q)+rad
      AVERAGE(Q)=SUM(Q)/(particle_number-5)
c
c     Output to File "Solids-Fraction"
```

```
c
      WRITE (7,400) YMAX(Q)
      WRITE (7,401) XMAX(Q)
      WRITE (7,402) ZMAX(Q)
      WRITE (7,403) YMIN(Q)
      WRITE (7,404) XMIN(Q)
      WRITE (7,405) ZMIN(Q)
      WRITE (7,410) AVERAGE(Q)
c
      delta=YMAX(Q)/ZONES
      P=1
      VOL(Q,P-1)=0.0d0
c
58    IF (P .GT. zones) GOTO 70
c
      N1(Q,P)=0
      N2(Q,P)=0
      N3(Q,P)=0
      VOL1(Q,P)=0.0d0
      VOL2(Q,P)=0.0d0
      VOL3(Q,P)=0.0d0
c
      DO 60 K=1,particle_number-1
        IF (Y(Q,K).LE.(P*delta-rad)) THEN
          N1(Q,P)=N1(Q,P)+1
          VOL1(Q,P)= VOL1(Q,P)+4*PI*rad*rad*rad/3.0d0
        ELSEIF ((Y(Q,K).GT.(P*delta-rad)).AND.
     +        (Y(Q,K).LE.P*delta)) THEN
          h=Y(Q,K)+rad-P*delta
          N2(Q,P)= N2(Q,P)+1
          VOL2(Q,P)= VOL2(Q,P)+4*PI*rad*rad*rad/3.0d0
     +              -PI*h*h*(3*rad-h)/3.0d0
        ELSEIF ((Y(Q,K).GT.(P*delta)).AND.
     +        (Y(Q,K).LE.(P*delta+rad))) THEN
          h=rad+delta*P-Y(Q,K)
          N3(Q,P)=N3(Q,P)+1
          VOL3(Q,P)=VOL3(Q,P)+PI*h*h*(3*rad-h)/3.0d0
        ENDIF
60    CONTINUE
c
      VOL(Q,P)=VOL1(Q,P)+VOL2(Q,P)+VOL3(Q,P)
      temp = xcell*zcell*delta*P
      PF(Q,P)=VOL(Q,P)/(XCELL*ZCELL*delta*P)
      WRITE (7,500) P,PF(Q,P)
      P=P+1
      GOTO 58
```

```
c
 70  ifile = ifile + 1
     Goto 37
c
c
 63  Continue
c
     CLOSE (UNIT=13)
     CLOSE (UNIT=7)
     STOP
c
c    **********Format Statements***************
c
100  FORMAT (1x,"Enter the number of particles")
120  FORMAT (1x,"Enter the number of files")
135  FORMAT (1x,"Enter the Diameter of the particles")
137  FORMAT (1x,"Enter the number of zones in Y direction")
200  FORMAT (/,1X,"**time= ",1p,e12.4,/,
   1       1X,"cumulative computer time (s)= ",e12.4)
300  FORMAT (/,3x,A1,2x,A4,8x,A4,8x,A4)
350  FORMAT (3e12.4 )
400  FORMAT (1x,"The maximum Y value is",1X,F8.6)
401  FORMAT (1x,"The maximum X value is",1X,F8.6)
402  FORMAT (1x,"The maximum Z value is",1X,F8.6)
403  FORMAT (1x,"The minimum Y value is",1X,F8.6)
404  FORMAT (1x,"The minimum X value is",1X,F8.6)
405  FORMAT (1x,"The minimum Z value is",1X,F8.6)
410  FORMAT (1x,"The average Y value is",1X,F8.6)
500  FORMAT ("The cumulative packing density of zone",i2," is",E12.5)
     END


c
=================================================================
c                        END OF PROGRAM
c ================================================================
```

B.2 Co-ordination number

(a) Computing the co-ordination number

Co-ordination number was computed using this code; the input for the code is the co-ordinates of particles, number of particles and the diameter of particles. The output files from the code are Cord[i], array containing the co-ordination number of frequency of occurrence and sum[i], nx1 array containing the number of particles in touch with each particle in the system.

```
c=================================================================
c                          CO-ORDINATION NUMBER
c
=================================================================
c
c       center          The file containing the co-ordinates of the particles
c
c       avgcn           File containing the average co-ordination number
c
c      co-ord.dat       File containing all the average of number of particles in
c                       touch with each particle in the different configurations
c
c       cord[i]         12x1 array containing the co-ordination number frequency of
c                       occurrence.
c       sum[i]          nx1 array containing the number of particles in touch with
c                       each particle in the system
c
c      x[i], y[i],z[i]  The co-ordinates of the particles
c      gap              Search diameter
c      dia              Diameter of particles
c
c ================================================================
c                          BEGINNING OF PROGRAM
c
c=================================================================
c
      REAL*8  dx, dy, dz, dia, gap, dst, search
      REAL*8  dummy(8000),avecn,avcn
      INTEGER         cord(12),coord(12),sum(8000), i, j, k, n, total,nc
      REAL*8  x(8000), y(8000), z(8000),navg
```

```
c
c
      OPEN(1,FILE='center',STATUS='old')
      OPEN(2,FILE='avgcn',STATUS='unknown')
      OPEN(3,FILE='co-ord.dat',STATUS='unknown')
      OPEN(4,FILE='sum',STATUS='unknown')
      OPEN(7,FILE='cnumber',STATUS='NEW')
c
c
      WRITE(*,*) 'Enter number of particles'
      READ(*,*)  n
      WRITE(*,*) 'Enter the diameter'
      READ(*,*) dia
      WRITE(*,*) 'Enter the gap'
      READ(*,*) gap
      write(*,*) 'enter number of configurations'
      read(*,*) nc

      search = dia + gap


      DO 51 p= 1,nc
c
      DO 1 i = 1, 2000
         sum(i) = 0.0
         x(i) = 0.0
         y(i) = 0.0
         z(i) = 0.0
1      CONTINUE
      do 98 i=1, 12
      coord(i) = 0
98       continue

      DO 3 i= 1, n
            READ(1,*)  x(i), y(i), z(i)
3      CONTINUE


      k = 1
5      j = k + 1
6      IF (j .GT. n) THEN
         k = k + 1
         IF (k .GT. n) THEN
           GOTO 100
         ENDIF
         GOTO 5
```

```
      ENDIF
c
      dx = ABS(x(k) - x(j))
      dy = ABS(y(k) - y(j))
      dz = ABS(z(k) - z(j))

      dst = SQRT(dx*dx + dy*dy + dz*dz)
      IF (dst.LE.search) THEN
        sum(k) = sum(k) + 1
        sum(j) = sum(j) + 1
      ENDIF
      j = j + 1
      GOTO 6
c
c ..............................................................................
c        Calculation of the co-ordination number frequency
c ..............................................................................
c
100   WRITE(3,*)

      total = 0.0
      avecn = 0.0

      DO 10 i = 1, 12

        DO 20 j = 1, n

          IF (sum(j).eq.i) THEN
            cord(i) = cord(i) + 1
            coord(i) = coord(i) +1
          ENDIF
20        CONTINUE

        total = cord(i) + total
10    CONTINUE
c.............................................................................
c        Calculation of the average co-ordination number
c.............................................................................

        DO 15 i = 1, 12
          j = i
          avecn = j*cord(i) + avecn
15    CONTINUE

      avcn = avecn/total
      write(7,*) avcn
```

```
      navg= navg+avcn
c
      do 99 i= 1,12
      write(4,*) i, coord(i)
99     continue
51    continue
      DO 30 i = 1, 12
        WRITE(3,*) i, (cord(i)/nc)
30    CONTINUE


      write(2,*) total/nc, (navg/nc)
      write(7,*) navg/nc
     CLOSE(unit=1)
c      CLOSE(unit=2)
      CLOSE(unit=3)
      CLOSE(unit=4)
      CLOSE(unit=7)
      STOP
      END
c
c=================================================================
c                        End of Program
c=================================================================
```

(b) Computing the sample deviation for co-ordination number

The program is used to compute the sample deviation of co-ordination number distribution frequency and the average co-ordination number for the different configurations. The files 'sum', 'co-ord.dat' and 'cnumber' from the 'co-ord.f' are used as input files.

```
c=================================================================
c                              SAMPLE DEVIATION
c
=================================================================
c       Max   File containing the maximum values of co-ordination number   frequency
c
c       Min   File containing the minimum values of co-ordination number frequency
c
c      Cndev  File containing the average co-ordination number for different
c             configurations
c
=================================================================
c                              BEGINNING OF PROGRAM
c
=================================================================
      INTEGER   i,a,j,k,p, numbin, isave,nc
      REAL*8  sg(20000), sum(8000), total(8000), samdev(8000)
      REAL*8  ng(8000) , rsd1(8000),freq(8000),diff(8000)
      REAL*8  max(8000),min(8000), sam(8000),dum(8000)
      Character*5    dummy
      Real*8  avgcn,cn(25),cnsum,cndiff,cntotal,cnvar,cnssd


      OPEN(1,FILE='sum',status='unknown')
      OPEN(2,FILE='co-ord.dat',status='unknown')
      OPEN(3,FILE='Max',status='unknown')
      OPEN(4,FILE='Min',status='unknown')
      OPEN(70,FILE='cnumber',status='OLD')
      OPEN(80,FILE='CNdev',status='NEW')

c.................................................................
c       Calculation of the Sample Deviation
c .................................................................
```

```
      write(*,*) 'Enter Numbin'
      read(*,*) numbin
      write(*,*) 'enter nc'
      read(*,*) nc
      a = nc-1

      do 6 k=1,numbin
      read(2,*) rsd1(k), ng(k)
      total(k) =0.0
 6     continue

      do 9 k=1,nc
       read(70,*) cn(k)
 9     continue
      read(70,*) avgcn

      cntotal=0.0


      do 91 i=1,nc

     do 32 p=1,numbin
     sg(p) =0.0
32    continue

      do 22 p=1,numbin
      read(1,*) dum(p), sg(p)
22      continue

       do 51 j= 1, numbin
c
      sum(j) =0.0
      diff(j)=0.0
      sum(j) = (sg(j)-ng(j))
c         write(*,*)  'sum=',sum(j)
      diff(j) = sum(j) * sum(j)
c           write(*,*) 'diff(j)=',diff(j)
         total(j) = total(j) + diff(j)
c        write(*,*) 'total=',total(j)
51      continue

91      continue

      do 62 j= 1, numbin
       sam(j)= total(j)/a
      samdev(j) = sqrt (sam(j))
```

```
62      continue
      do 63 j=1,numbin
      max(j) = (ng(j) + samdev(j))
      min(j) = (ng(j) - samdev(j))
     write(3,*) max(j)
     write(4,*) min(j)
63    continue

      Do 110 i=1,nc

      cndiff=cn(i)-avgcn
      cnsq=cndiff*cndiff
      cntotal=cntotal+cnsq

110     continue

      cnvar= (cntotal)/(nc-1)
      cnssd= sqrt(cnvar)
     write(80,*) 'Sample devition of Co-ordination no.'
     Write(80,*)  avgcn,'(+-)',cnssd



    CLOSE(unit=1)
    CLOSE(unit=2)
    CLOSE(unit=3)
    close(unit=4)
    CLOSE(unit=70)
    close(unit=80)
    STOP
    END
c
c=================================================================
c                        End of Program
c=================================================================
```

B.3 Radial distribution function

(a) To compute the radial distribution function

The code below is used to compute the radial distribution function. The input file required is the file containing the co-ordinates of the particle. The radial distribution is written to 'Cumal.dat', 'rad.dat'

```
c=================================================================
c       Radial Distribution Function
c=================================================================
c
c       Variables
c
c       freq            Integer Array containing histogram of
c                       intersphere distances
c       g               Array containing the "distribution function"
c                       The distribution function is computed at the mid point
c                       of each bin and is defined as the number of sphere
c                       centers in (r, r+dr) divided by 4*pi*(r*dia)**2.
c                       "dr" is not included here.
c       (x,y,z)         Arrays of sphere center co-ordinates
c        n              number of particles
c       binwid          bin width in units of sphere diameter
c                       redefined as (binwid*dia)
c       x,dy,dz         difference of x,y, z co-ordinates of 2 spheres
c       dia             sphere diameter
c       rad             sphere radius
c       dmax            maximum distance for which "g"  is to be calculated
c       eps             error parameter to account for m/c accuracy (1.0e-07)
c       dst             intersphere distance
c        numbin         Total number of bins =dmax/binwid (integer)
c       low             Integer used for "Binary Search" to locate correct bin
c       mid             Integer used for "Binary Search" to locate correct bin
c        high           Integer used for "Binary Search" to locate correct bin

c ..................................................................................
c
c       Input and Output files
c
c       Unit 1          File 'center' from which date is read
c       Unit 2          File 'rad' to which date is written
```

```
c
c ..............................................................................
c==============================================================
c                    BEGINNING OF PROGRAM
c==============================================================
c
c
      REAL*8  binwid, dx, dy, dz, dia, rad, eps, dst, bmid, nc
      REAL*8  lcube, ymax, dmax, dummy(8000),binwid1
      INTEGER         c,freq(8000), low, high, mid, i, j,k,p, numbin, isave
      REAL*8   nfreq(8000), ng(8000),max(8000),min(8000)
      REAL*8  x(8000), y(8000), z(8000), g(8000), pi, rs, rsd1
c
      OPEN(1,FILE='center',STATUS='unknown')
      OPEN(2,FILE='rad',STATUS='UNKNOWN')
      OPEN(3,FILE='SampleDev', STATUS='UNKNOWN')
c
      WRITE(*,*) 'Enter the number of Particles'
      READ(*,*) n
       Write(*,*) 'Enter the diameter'
      READ(*,*) dia
       Write(*,*)'Enter the bandwidth'
      READ(*,*) binwid1
       Write(*,*) 'Enter the lenght of the cube'
      READ(*,*) lcube
       Write(*,*) 'Enter the  number of configurations'
       Read(*,*) nc
      PRINT*, n, dia, binwid, lcube
c
      pi = 3.1415926536
      eps = 1.0E-07
c
      DO 25 i = 1, n
      nfreq(i) = 0.0
      ng(i) = 0.0
      max(i)=0.0
      min(i)=0.0
25     CONTINUE
c
c
      c=0
      DO 15  p= 1, nc
     c= c+1
      DO 1 i= 1, n
         x(i) = 0.0
         y(i) = 0.0
```

```
      z(i) = 0.0
      g(i) = 0.0
1    CONTINUE

     DO 2 i= 1, n
          READ(1,*)  x(i), y(i), z(i)
2    CONTINUE
     binwid=binwid1
c
     ymax = y(1)
     DO 3 i = 2, n
       IF (y(i) .GT. ymax) ymax = y(i)
3    CONTINUE

     IF (ymax .GT. lcube) THEN
       dmax = ymax
     ELSE
       dmax = lcube
     ENDIF
c
     DO 4 i=1, n
       freq(i)=0.0
4    CONTINUE
c
     rad = dia*0.5
     binwid = binwid*dia
     numbin = INT(dmax/binwid)
     write(*,*)'Numbin=', numbin
c
c
     k = 1
5    j = k + 1
6    IF (j .GT. n) THEN
       k = k + 1
       IF (k .GT. n) THEN
         GOTO 100
       ENDIF
       GOTO 5
     ENDIF
c

     dx = ABS(x(k) - x(j))
     dy = ABS(y(k) - y(j))
     dz = ABS(z(k) - z(j))

     dst = SQRT(dx*dx + dy*dy + dz*dz)
```

```
c
c ..................................................................
c                    Start of Binary Search
c ..................................................................
c
      low = 1
      high = numbin

50     mid = (low + high)/2
      bmid = mid*binwid
c
      IF (dst .LT. bmid) high = mid - 1
      IF (dst .GT. bmid) low = mid + 1
      IF ((dst .LE. bmid+eps).and.(dst.GE.bmid-eps)) THEN
        freq(mid) = freq(mid) + 1
        j = j + 1
        GOTO 6
      ENDIF
c

      IF (low .LE. high) THEN
        GOTO 50
      ELSE
        isave = low
        freq(isave) = freq(isave) + 1
        j = j + 1
        GOTO 6
      ENDIF
c
100    WRITE(2,*)

c
c ..................................................................
c          Calculation of the Radial Distribution Function
c ..................................................................
c
      rs = binwid
c
       DO 110 j = 1, numbin
        rsd1 = (rs - (0.5*binwid))/dia
        g(j) = 2*freq(j)/(4.0*pi*rsd1*rsd1*n)
        if( p .eq. 1) max(j)=g(j)
       if(p .eq. 1) min(j)=g(j)
c
       if(g(j) .ge. max(j)) max(j)=g(j)
       if(g(j) .le. min(j)) min(j)=g(j)
```

```
c
        nfreq(j) = nfreq(j)+ freq(j)
       ng(j) = g(j)+  ng(j)
c        write(*,*) ng(j)
       rs = rs + binwid
110   CONTINUE
15    CONTINUE
c.....................................................................
c     Calculating the cumilative of all the configurations
c.....................................................................
      rs = binwid
      DO 18 j = 1,numbin
      rsd1 = (rs - (0.5*binwid))/dia
      nfreq(j) = nfreq(j)/nc
      ng(j) = ng(j)/nc
      write(2,*) rsd1 ,nfreq(j), ng(j)
      write(3,*) rsd1,max(j),min(j)
      rs = rs + binwid
18     CONTINUE
c.....................................................................
c        Calculation of the Co-ordination number
c        The number of spheres in contact with a given sphere
c ....................................................................
c


120   CLOSE(unit=1)
     CLOSE(unit=2)
     CLOSE(unit=3)
     STOP
     END
c
c=====================================================================
c                        End of Program
c=====================================================================
```

(b) To compute the sample deviation for radial distribution function

The code is used to compute the sample deviation for the radial distribution over number of different configurations. The input files are 'Cumal.dat' and 'rad.dat' from radial distribution code. The outputs from the code are 'Max' and 'Min' which has the maximum and minimum values of radial distribution.

```
c------------------------------------------------------------ ------------------
C          BEGNING OF PROGRAM
C-----------------------------------------------------------------------------
      INTEGER   i,a,j,k,p, numbin, isave,nc
      REAL*8   sg(6000), sum(2000), total(2000), samdev(2000)
      REAL*8   ng(2000) , rsd1(2000),freq(2000),diff(2000)
      REAL*8   max(2000),min(2000), sam(2000)
      OPEN(1,FILE='Cumlat.dat',status='unknown')
      OPEN(2,FILE='rad.dat',status='unknown')
      OPEN(3,FILE='Max',status='unknown')
      OPEN(4,FILE='Min',status='unknown')
c.............................................................
c        Calculation of the Sample Deviation
c ............................................................
        write(*,*) 'Enter Numbin'
        read(*,*) numbin
        write(*,*) 'enter nc'
        read(*,*) nc
        a = nc-1

        do 6 k=1,numbin
        read(2,*) rsd1(k),freq(k),ng(k)
        total(k) =0.0
 6       continue

        do 91 i=1,nc

       do 32 p=1,numbin
       sg(p) =0.0
 32     continue

        do 22 p=1,numbin
        read(1,*) sg(p)
```

```
22      continue

        do 51 j= 1, numbin
c
       sum(j) =0.0
       diff(j)=0.0
       sum(j) = (sg(j)-ng(j))
       diff(j) = sum(j) * sum(j)
          total(j) = total(j) + diff(j)
c         write(*,*) 'total=',total(j)
51      continue
91      continue

       do 62 j= 1, numbin
        sam(j)= total(j)/a
        samdev(j) = sqrt (sam(j))
62      continue
      do 63 j=1,numbin
      max(j) = (ng(j) + samdev(j))
      min(j) = (ng(j) - samdev(j))
     write(3,*) max(j)
     write(4,*) min(j)
63     continue
c
    CLOSE(unit=1)
    CLOSE(unit=2)
    CLOSE(unit=3)
    close(unit=4)
    STOP
    END
c
c=================================================================
c                        End of Program
c=================================================================
```

B.4. Angular distribution

The angular distribution code is used to compute the angles made by the contacting spheres. The file containing the co-ordinates of the particles is the input file. The angles made with horizontal plane are written in theta[1-12] and angles made in vertical direction are written phi[1-12].

```
c=================================================================
c     ANGULAR CONTACT AND COORDINATION NUMBER DISTRIBUTIONS
c=================================================================
c       phi1-12[i]      arrays containing phi angles that any particle
c                       makes with the neighboring particles
c       theta1-12[i]    arrays of theta angles that any particle
c                       makes with the neighboring particle
c Note: ph5 contains particles with 5 neighbors...that phi contains
c         particles with i = 1 to 12 neighbors. Same applies to theta[i]
c
=================================================================
c                       BEGINING OF PROGRAM
c=================================================================
c
      REAL*8  dx, dy, dz, dia, gap, dst,dst1, search
      REAL*8  dummy(2000), dummy1, dummy2
      REAL*4  phi(12), theta(12), x1, y1, z1
      INTEGER  sum, i, j, k, n
      REAL*8  x(8000), y(8000), z(8000)
c
c
c Files (1 - 2): input files

      OPEN(1,FILE='center',STATUS='old')
c      OPEN(2,FILE='gap',STATUS='old')
      Write(*,*) 'enter the number of particles'
      READ(*,*) n
      Write(*,*) 'enter diameter'
      READ(*,*) dia
      write(*,*) 'Enter the gap'
      READ(*,*) gap
      write(*,*) 'all parameters read'

c      write(*,*) ' 2 files'

c Files (3 - 14): output files for values of phi for i = 1 to 12
```

```
c  i = 1 means one neighbhor, i = 2 means two neighbhors...etc

      OPEN(3,FILE='ph1',STATUS='unknown')
      OPEN(4,FILE='ph2',STATUS='unknown')
      OPEN(50,FILE='ph3',STATUS='unknown')
      OPEN(60,FILE='ph4',STATUS='unknown')
      OPEN(7,FILE='ph5',STATUS='unknown')
      OPEN(8,FILE='ph6',STATUS='unknown')
      OPEN(9,FILE='ph7',STATUS='unknown')
      OPEN(10,FILE='ph8',STATUS='unknown')
      OPEN(11,FILE='ph9',STATUS='unknown')
      OPEN(12,FILE='ph10',STATUS='unknown')
      OPEN(13,FILE='ph11',STATUS='unknown')
      OPEN(14,FILE='ph12',STATUS='unknown')


c Files (15 - 26): ouput files for thetai, i = 1 to 12

      OPEN(15,FILE='theta1',STATUS='unknown')
      OPEN(16,FILE='theta2',STATUS='unknown')
      OPEN(17,FILE='theta3',STATUS='unknown')
      OPEN(18,FILE='theta4',STATUS='unknown')
      OPEN(19,FILE='theta5',STATUS='unknown')
      OPEN(20,FILE='theta6',STATUS='unknown')
      OPEN(21,FILE='theta7',STATUS='unknown')
      OPEN(22,FILE='theta8',STATUS='unknown')
      OPEN(23,FILE='theta9',STATUS='unknown')
      OPEN(24,FILE='theta10',STATUS='unknown')
      OPEN(25,FILE='theta11',STATUS='unknown')
      OPEN(26,FILE='theta12',STATUS='unknown')

      OPEN(27,FILE='dump',STATUS= 'unknown')
      OPEN(28, FILE='phidump', STATUS= 'unknown')
       search = dia + gap

c      dummy1 = 0
c      dummy2 = 0
c
c
      DO 1 i = 1, 2000
         x(i) = 0.0
         y(i) = 0.0
         z(i) = 0.0
1     CONTINUE

c
```

```
      DO 3 i= 1, n
           READ(1,*)  x(i), y(i), z(i)
3     CONTINUE

c 102

      DO 2  i = 1, 12
        phi(i) = 0.0
        theta(i) = 0.0
2     CONTINUE

      sum = 1
      j = 1

      k = 1
c5     j = 1
c      sum = 0

c------- Loop for incrementing k

6     IF (j .GT. n) THEN

      IF (sum.GT.1) THEN
       DO 9 m = 1, sum-1
        DO 8 i = m+1, sum

            IF (theta(m).GT.theta(i)) THEN
                dummy2 = theta(m)
                dummy1 = phi(m)
                theta(m) = theta(i)
                phi(m) = phi(i)
                theta(i) = dummy2
                phi(i) = dummy1
            ENDIF
8       CONTINUE
9       CONTINUE
      ENDIF

        IF (sum.EQ.1) THEN
           WRITE(3,*) k, phi(1)
           WRITE(15,*) k, theta(1)
           write(27,*)  theta(1)
           write(28,*) phi(1)
        ENDIF

        IF (sum.EQ.2) THEN
```

```
   WRITE(4,*) k,  phi(1), phi(2)
   WRITE(16,*) k, theta(1), theta(2)
   write(28,*) phi(1)
   write(28,*) phi(2)
ENDIF

IF (sum.EQ.3) THEN
   WRITE(50,*) k,  (phi(i), i=1,3)
   WRITE(17,*) k, theta(1), theta(2), theta(3)
ENDIF

IF (sum.EQ.4) THEN
WRITE(60,21) k,phi(1), phi(2), phi(3), phi(4)
 WRITE(18,21) k, (theta(i), i = 1, 4)

ENDIF

IF (sum.EQ.5) THEN
   WRITE(7,*) k,  (phi(i), i = 1, 5)
   WRITE(19,*) k, (theta(i), i = 1, 5)

 ENDIF

IF (sum.EQ.6) THEN
   WRITE(8,41) k,  (phi(i), i = 1, 6)
   WRITE(20,41) k, (theta(i), i = 1, 6)
ENDIF

IF (sum.EQ.7) THEN
   WRITE(9,51) k,  (phi(i), i = 1, 7)
   WRITE(21,51) k, (theta(i), i = 1, 7)

ENDIF

IF (sum.EQ.8) THEN
   WRITE(10,61) k, (phi(i), i = 1, 8)
   WRITE(22,61) k, (theta(i), i = 1, 8)
ENDIF

IF (sum.EQ.9) THEN
   WRITE(11,71) k, (phi(i), i = 1, 9)
   WRITE(23,71) k, (theta(i), i = 1, 9)

ENDIF

IF (sum.EQ.10) THEN
```

```
          WRITE(12,81) k, (phi(i), i = 1, 10)
          WRITE(24,81) k, (theta(i), i = 1, 10)

        ENDIF

        IF (sum.EQ.11) THEN
          WRITE(13,91) k, (phi(i), i = 1, 11)
          WRITE(25,91) k, (theta(i), i = 1, 11)

        ENDIF

        IF (sum.EQ.12) THEN
          WRITE(14,*) k, (phi(i), i = 1, 12)
          WRITE(26,101) k, (theta(i), i = 1, 12)

        ENDIF

        k = k + 1
        sum = 0

        DO 7  i = 1, 12
        phi(i) = 0.0
        theta(i) = 0.0
7        CONTINUE

      IF (k .GT. n) THEN
          GOTO 100
        ENDIF

        j=1
        sum=0
        ENDIF

c------ Loop ends here

      IF (j.EQ.k) j = j + 1


      dx = x(k) - x(j)
      dy = y(k) - y(j)
      dz = z(k) - z(j)
      x1 = (x(j) - x(k))
      y1 = (y(j) - y(k))
      z1 = (z(j) - z(k))
c      WRITE(27,*) 'x',x1, 'y' , y1,'z', z1
      dst1= dx*dx+dy*dy+dz*dz
```

```
c     write(*,*) dst1
      dst = SQRT(dx*dx + dy*dy + dz*dz)
c     WRITE(27,*) dst

      IF (dst.LE.search) THEN

        sum = sum + 1
        r = SQRT(dx*dx + dy*dy)

        phi(sum) = atan(dx/dz)
c         IF(sum.eq.4) write(*,*) k, r, dz
          IF (z1.LT.0) phi(sum) = 3.14 -phi(sum)
          phi(sum) = INT(180*phi(sum)/3.14)
c         IF(sum.EQ.4) write(60,*) (phi(i), i=1,4)

          IF ((dy.EQ.0).AND.(dx.EQ.0)) THEN
            theta(sum) = 0
          ELSE
            theta(sum) = atan(dy/dx)
          ENDIF

          IF ((x1.LT.0) .AND. (y1.GT.0)) THEN
            theta(sum) =  3.14 - abs(theta(sum))
          ENDIF
          IF ((x1.LT.0) .AND. (y1.LT.0)) THEN
            theta(sum) =  abs(theta(sum)) + 3.14
        ENDIF
        IF ((x1.GT.0) .AND. (y1.LT.0)) THEN
        theta(sum) = 2*3.14 - abs(theta(sum))
       ENDIF
        IF ((x1.EQ.0).AND.(y1.LT.0)) theta(sum) = 4.71
        IF ((x1.LT.0).AND.(y1.EQ.0)) theta(sum) = 3.14


        theta(sum) = INT(180*theta(sum)/3.14)
c         IF(sum.EQ.4) WRITE(18,*) k, (theta(i), i=1,3)
        WRITE(*,*) atan(0.0/1.0)

      ENDIF

      j = j + 1
      GOTO 6

100   CLOSE(unit=1)
      CLOSE(unit=2)
      CLOSE(unit=3)
```

```
      CLOSE(unit=4)
      CLOSE(unit=50)
      CLOSE(unit=60)
      CLOSE(unit=7)
      CLOSE(unit=8)
      CLOSE(unit=9)
      CLOSE(unit=10)
      CLOSE(unit=11)
      CLOSE(unit=12)
      CLOSE(unit=13)
      CLOSE(unit=14)
      CLOSE(unit=15)
      CLOSE(unit=16)
      CLOSE(unit=17)
      CLOSE(unit=18)
      CLOSE(unit=19)
      CLOSE(unit=20)
      CLOSE(unit=21)
      CLOSE(unit=22)
      CLOSE(unit=23)
      CLOSE(unit=24)
      CLOSE(unit=25)
      CLOSE(unit=26)
      close(unit=27)
      close(unit=28)
21    FORMAT (5(X, F7.3))
31    FORMAT (6(X, F7.3))
41    FORMAT (7(X, F7.3))
51    FORMAT (8(X, F7.3))
61    FORMAT (9(X, F7.3))
71    FORMAT (10(X, F7.3))
81    FORMAT (11(X, F7.3))
91    FORMAT (12(X, F7.3))
101   FORMAT (13(X, F7.3))
      STOP
      END
c=================================================================
c                        End of Program
c=================================================================
```

B.5 Fabric tensor

(a) Computing the fabric tensor

This code is used to compute the fabric tensor of a packing assembly. The file containing the co-ordinates of the particles is the input file and fabric tensor is written in the output file 'Tensor'. Number of parti cles and the diameter of particles should be entered as input data.

```
c=================================================================
c                    Program To compute the Fabric Tensor C
c=================================================================
c       rad3.dat     Input file
c       Tensor       Output file containing the Fabric Tensor
c       Q3(3,3)      Fabric Tensor
c       d            Diameter of particles
c       N            Number of particles
c       noc(i)       Number of contacts
c=================================================================
c                    BEGINNING OF PROGRAM
c=================================================================

      Real*8 x(8000), y(8000), z(8000), Q(3,3)
      Real*8 x1(8000), y1(8000), z1(8000) , e, d
      Real*8 a,b,c,dc,mod,n1,n2,p,p1,p2,p3,Q1(3,3)
      Real*8 Q2(3,3),Q3(3,3)
      Integer i,j,k,l,m,N,noc(125000)
C
      open(unit=12,file='rad3.dat',status='Unknown')
      Open(unit=15,file='Tensor',status='New')
C
      write(*,*) 'Enter the Number of particles'
      read(*,*) N
      write(*,*) 'Enter the Diameter of the particle'
      read(*,*) d
c
C*****Intializing the varibales******
      do 55 k= 1 , N
       x(k)=0.0
       y(k)=0.0
       z(k)=0.0
       x1(k)=0.0
       y1(k)=0.0
```

```
      z1(k)=0.0
 55     continue
        do 71 i= 1, 3
          do 72 j=1,3
          Q(i,j) =0.0
 72       continue
 71       continue
C
      e =(d*10)/100
C
C******  To Read The Input File*******
C
      do 50 k = 1, N
      read(12,*) x(k), y(k), z(k)

50      continue
c**The loop for determining the number of contacts***
c
          do 62 i=1, N
c
             do 63 j=1, N
c      write(*,*) 'Start'
      a=0.0
      b=0.0
      c=0.0

      If (i .ne. j) then
      a=x(i)-x(j)
      b=y(i)-y(j)
      c=z(i)-z(j)
      dc=(a*a+b*b+c*c)*(a*a+b*b+c*c)
C
       If (dc .le. (d+e)*(d+e)) THEN

C
C*** Co-ordinates of Jth element w.r.t Ith Element******
c
         x1(j) =x(j)-x(i)
         y1(j) =y(j)-y(i)
         z1(j) =z(j)-z(i)
          noc(i)=noc(i)+1
          p1= x1(j)*x1(j)
          p2= y1(j)*y1(j)
          p3= z1(j)*z1(j)
          p= p1+p2+p3
          mod = SQRT(p)
```

```
      do 60 l=1, 3
         do 61 m= 1, 3
      if((l .eq. 1) .and. (m .eq. 1)) then
       n1=x1(j)/mod
       n2=x1(j)/mod

c
     else if((l.eq.1) .and. (m .eq. 2)) then
      n1=x1(j)/mod
      n2=y1(j)/mod
c
     else if((l.eq.1) .and. (m.eq.3)) then
      n1=x1(j)/mod
      n2=z1(j)/mod

c

     else if((l.eq.2) .and. (m.eq.1)) then
      n1=y1(j)/mod
      n2=x1(j)/mod

c
    else if((l.eq.2) .and. (m.eq.2)) then
      n1=y1(j)/mod
      n2=y1(j)/mod

c
    else if((l.eq.2) .and. (m.eq.3)) then
      n1=y1(j)/mod
      n2=z1(j)/mod

c
    else if((l.eq.3) .and. (m.eq.1)) then
      n1=z1(j)/mod
      n2=x1(j)/mod

c
    else if((l.eq.3) .and. (m.eq.2)) then
      n1=z1(j)/mod
      n2=y1(j)/mod
c
    Else if((l.eq.3) .and. (m.eq.3)) then
      n1=z1(j)/mod
      n2=z1(j)/mod
    Else
```

```
      write(*,*) ' Not Valid Range'

      END IF
      Q(l,m) = Q(l,m) + (n1*n2)

 61         continue
c
 60        continue

       END IF
      Else
       END IF

63      continue

      do 74 l=1,3
        do 75 m=1,3
       write(14,*) Q(l,m)
       Q2(l,m)=Q(l,m)/noc(i)
       Q3(l,m)=Q3(l,m)+Q2(l,m)
75       continue
74       continue

62    continue
      do 9 l= 1, 3
         do 10 m= 1, 3
     Q1(l,m)= Q2(l,m)/N
    write(15,*) 'Q(',l,m,')=', Q1(l,m)
10      continue
9       continue

    close(unit=12)
    close(unit=15)
    close(unit=14)
     End

c=================================================================
c                        End of Program
c=================================================================
```

(b) To compute rotation of fabric tensor

The code is used to compute the fabric tensor for rotated co-ordinate axes. The tensor computed in fabric tensor code is the input file. The angles of rotation for each co-ordinate axes should also be provided.

```
c=================================================================
c                    SUBROUTINE TO CALCULATE THE FABRIC TENSOR
c                          FOR DIFFERENT ORIENTATION
c=================================================================
c       alpha          Angle of rotation of X-axis
c       beta           Angle of rotation of Y-axis
c       gamma          Angle of rotation of Z-axis
c       RotatedMatrix Output file
c=================================================================
c                    BEGINNING OF PROGRAM
c=================================================================


      REAL*8  T(3,3), Q(3,3), R(3,3)
      INTEGER  i,j,k
      REAL    alpha,beta,gamma
      REAL  l1,l2,l3,m1,m2,m3,n1,n2,n3


      Open(unit=1,File='RotatedMatrix',Status='New')
      DO 21 i=1,3
       DO 22 j=1,3
      Q(i,j)=0.0
      WRITE(*,*)'Enter the values of Q','Q(',i,j,')'
      READ (*,*) Q(i,j)
22     CONTINUE
21     CONTINUE

      WRITE(*,*) 'Enter the Values of a,b,g'
      read(*,*)  a,b,g
      alpha= (a*3.1415926535897932)/180
      beta = (b*3.1415926535897932)/180
      gamma =(c*3.1415926535897932)/180
C
```

```
C     TRANSFORMATION MATRIX
C
      T(1,1) = cos(alpha)*cos(beta)
      T(1,2) = cos(alpha)*sin(beta)*sin(gamma)
     +          -sin(alpha)*cos(gamma)
      T(1,3) = cos(alpha)*sin(beta)*cos(gamma)
     +          +sin(alpha)*sin(gamma)
      T(2,1) = sin(alpha)* cos(beta)
      T(2,2) = sin(alpha)*sin(beta)*sin(gamma)
     +       + cos(alpha)*cos(gamma)
      T(2,3) = sin(alpha)*sin(beta)*cos(gamma)
     +       - cos(alpha)*sin(gamma)
      T(3,1) = -sin(beta)
      T(3,2) = cos(beta)*sin(gamma)
      T(3,3) = cos(beta)*cos(gamma)
C
C
C
      l1=T(1,1)
      l2=T(2,1)
      l3=T(3,1)

      m1=T(1,2)
      m2=T(2,2)
      m3=T(3,2)

      n1=T(1,3)
      n2=T(2,3)
      n3=T(3,3)

      R(1,1)= Q(1,1)*l1*l1+Q(2,2)*m1*m1+Q(3,3)*n1*n1+
     +      2*(Q(1,2)*l1*m1+Q(2,3)*m1*n1+Q(1,3)*l1*n1)

      R(1,2)= Q(1,1)*l1* l2+Q(2,2)*m1*m2+Q(3,3)*n1*n2
     +      +Q(1,2)*(l1*m2+m1*l2)+Q(2,3)*(m1*n2+n1*m2)
     +      +Q(1,3)*(n1*l2+l1*n2)

      R(1,3)=Q(1,1)*l1*l3+Q(2,2)*m1*m3+Q(3,3)*n1*n3
     +      +Q(1,2)*(l1*m3+m1*l3)+Q(2,3)*(m1*n3+n1*m3)
     +      +Q(1,3)*(n1*l3+l1*n3)

      R(2,2)=Q(1,1)*l2*l2+Q(2,2)*m2*m2+Q(3,3)*n2*n2+
     +      2*(Q(1,2)*l2*m2+Q(2,3)*m2*n2+Q(1,3)*l2*n2)

      R(2,3)=Q(1,1)*l2*l3+Q(2,2)*m2*m3+Q(3,3)*n2*n3
     +      +Q(1,2)*(l2*m3+m2*l3)+Q(2,3)*(m2*n3+n2*m3)
```

```
+      +Q(1,3)*(n2*l3+l2*n3)

 R(3,3)=Q(1,1)*l3*l3+Q(2,2)*m3*m3+Q(3,3)*n3*n3+
+       2*(Q(1,2)*l3*m3+Q(2,3)*m3*n3+Q(1,3)*l3*n3)

 R(2,1)=R(1,2)
 R(3,2)=R(2,3)
 R(3,1)=R(1,3)
    Write(1,*) ' Directional Cosines'

      write(1,*) T(1,1), T(1,2), T(1,3)
      write(1,*) T(2,1), T(2,2), T(2,3)
      write(1,*) T(3,1), T(3,2), T(3,3)

    write(1,*) '*************************'
    write(1,*) ' Fabric Tensor in the New Orientation'
    DO 15 i= 1, 3
     DO 18 j=1, 3
    write(1,*) 'R(',i,j,')',R(i,j)
18     continue
15     continue

   close(unit=1)
   END
```

APPENDIX C

DATA FROM THE ANALYSIS

Data obtained from the solids fraction and co-ordination number codes are shown below.

Table C.1 Solids fraction for different size systems

729 Particles	1000 Particles	1728 Particles	3375 Particles	8000 Particles
Solids-Fraction	Solids-Fraction	Solids-Fraction	Solids-Fraction	Solids-Fraction
0.5702	0.5727	0.57451	0.57607	0.58
0.571	0.5722	0.57843	0.578999	0.5813
0.5672	0.57418	0.57411	0.57965	0.5809
0.5685	0.5755	0.57328	0.57874	0.5825
0.56763	0.57404	0.57676	0.57963	0.5821
0.56871	0.57268	0.57542	0.579	0.5821
0.57191	0.57644	0.5761	0.57684	0.5816
0.56863	0.57396	0.57095	0.57869	0.5812
0.57192	0.57327	0.57679	0.578	0.5806
0.57218	0.57298	0.57466	0.57817	0.5802
0.57102	0.57179	0.57617	0.57871	0.5812
0.57102	0.5723	0.5781	0.57723	0.5808
0.57107	0.57317	0.5741	0.57507	0.5814
0.57021	0.5777	0.57455	0.57782	0.5799
0.56743	0.5705	0.57402	0.57874	0.5815
0.57327	0.57235	0.57795	0.58045	0.5808
0.57268	0.57612	0.5749	0.57938	0.582
0.57131	0.57484	0.57562	0.57801	0.5813
0.56658	0.5735	0.57473	0.57787	0.5818
0.57109	0.57445	0.57488	0.57676	0.5816
0.57685	0.57335	0.57643	0.57757	0.5817
0.57152	0.57511	0.57557	0.57845	0.5821
0.57208	0.57153	0.57353	0.58086	0.5815
0.57244	0.57151	0.57284	0.57857	0.5819
0.57137	0.57848	0.57236	0.57926	0.5821
Avg. 0.56658	0.5705	0.57236	0.57507	0.582
Max. 0.57685	0.57848	0.57843	0.58086	0.5825
Min. 0.5707	0.57378	0.5750704	0.578314	0.5799
Sdev 0.00231	0.001919	0.001804389	0.001324275	0.0006861

Table C.2 Co-ordination number distribution for 729 particle system

Co-ordination Number	Number of particles
1	5
2	23
3	76
4	112
5	149
6	170
7	130
8	51
9	7
10	0
11	0
12	0

Table C.3 Co-ordination number distribution for 1000 particle system

Co-ordination Number	Number of particles
1	5
2	30
3	89
4	141
5	190
6	240
7	200
8	84
9	14
10	1
11	0
12	0

Table C.4 Co-ordination number distribution for 1728 particle system

Co-ordination Number	Number of particles
1	7
2	42
3	129
4	222
5	324
6	439
7	377
8	154
9	27
10	1
11	0
12	0

Table C.5 Co-ordination number distribution for 3375 particle system

Co-ordination Number	Number of particles
1	7
2	42
3	129
4	222
5	324
6	439
7	377
8	154
9	27
10	1
11	0
12	0

Table C.6 Co-ordination number distribution for 8000 particle system

Co-ordination Number	Number of particles
1	4
2	29
3	106
4	208
5	349
6	571
7	647
8	369
9	110
10	36
11	47
12	65

REFERENCES

1. W. M. Visscher & M. Bolsterli; Random packing of Equal and unequal spheres in two and three dimensions, Nature Vol.239 (1972) 504-509.

2. T. G. Owe Berg, R.L. McDonald and R.J. Trainor, Jr; The Packing of Spheres; Powder Technol.,3(1969/70) 183-188.

3. D. J. Cumberland and R. J. Crawford; Elsevier 1987.

4. J. D. Bernal, "A geometrical approach to the structure", "Geometry of the structure of monotomic liquids," Nature, 1985, 68-70(1960).

5. D. J. Adams and A. J. Matheson, "Computation of dense random packing of hard spheres", J. Chem. Phys. 56 1989-1994(1972).

6. J. L. Finney, "Random packing and structure of simple liquids", Pro. Roy. Soc. Lond. 319, 478-493.

7. A. J. Matheson, "Computation of a random packing of hard spheres", J. Phys. 72569-2576(1974).

8. G. D. Scott, " Packing of spheres", Nature 188 908-909(1960).

9. C. H. Bennett, "Serially deposited amorphous aggregates of hard spheres", J. Appl. Phsys. , 6 2727-2750 (1972).

10. S. Park, " The simulation of random packing of hard spheres using Monte Carlo simulation method", Masters' Thesis, New Jersey Institute of Technology(1990).

11. G. T. Nolan and P. E. Kavanagh; Computer Simulation of Random Packing Of Hard Spheres; Powder Technology, 72(1992) 149-155.

12. W. O. Smith, P.D.Foote , P. F. Bussang ;" Packing of Homogeneous Spheres", Phys. Rev. 34[2] (1929) 1271-1274.

13. J. Wadsworth, T. G. Neih, Scripta. metallurgical materials 25(1991) 955.

14. M. J. Powell, " Computer simulated random packing of spheres", Powder Technology 25, 45-52 (1980).

15. G. D. Scott and D. M. Kilgour; "The density of random close packing of spheres", Brit. J. Applied. Phys. (J. Phys. D), 1969, SER. 2, Vol. 2.

16. Y. F. Cheng, S. J. Guo and H.Y. Lai; "Dynamic simulation of random packing of spherical particles", Powder Technology, 107(2000) 123-130.

17. A. B. Yu and N. Standish, "An analytical-parametric theory of the random packing of particles", Powder Technology, 55 (1988) 171-186.

18. E. M. Tory, N. A. Cochran and S. R. Waddell; "Anisotropy in simulated Random Packing of Equal Spheres', Nature 220, 1023-1024 (1968).

19. J. S. Goodling and M. S. Khader; " Co-ordination Number Distribution Of Spherical Particles in a Packed Cylindrical Bed",

20. A. C. Ugural and S. K. Fenster; "Advanced Strength and Applied Elasticity", Elsevier 1975.

21. A. Rosato, F. Prinz, K. J. Standburg and R. Swendsen; Monte Carlo simulation of particulate matter segregation", Powder Technology, 49(1986) 59-69.

22. G. T. Nolan and P. E. Kavanagh; "Computer Simulation of random packing of spheres with log-normal distributions", Powder Technology, 76(1993) 309-316.

23. J. A. Dodds; "Simplest statistical geometric model of the simplest version of the multi-component random packing problem", Nature Vol. 256 July 17 1975.

24. M. Oda and K. Iwashita, "Introduction to mechanics of granular materials", Balkema, Rotterdam, 1999.

25. A. J. Sederman, P. Alexander, L.F. Gladden, " Structure of packed beds probed by magnetic resonance imaging", Powder Technology 117 (2001) 255-269.

26. N. Zhang, " Vibration-induced densification of granular materials", PhD. Dissertation, New Jersey Institute of Technology, 2003.