ABSTRACT

DISCRETE ELEMENT SIMULATION AND NONLINEAR DYNAMIC ANALYSIS OF PARTICLES IN A SIMPLE LATTICE STRUCTURE

by
Liam E. Buckley

The study of particles interacting in lattice structures allows for insight into the complex interactions of granular flow, and the adaptation of such structures to mechanical apparatuses to handle the separation of bulk particulate matter. Applying methods of computer analysis to the interactions taking place within the particle-lattice system provides a particle level methodology to the study of the phenomenon taking place, as well as a stepping stone for future design of related devices.

A two prong approach is presented to the study of such particle-lattice systems. The first approach is composed of adapting an existing discrete element computer code to handle the geometry and peculiars of a particle traveling through a simple triangular lattice system. This discrete element code has been shown in previous research to accurately represent the interactions of such complex systems as a vibrating granular bed, and has been successful in predicting convective transport and other dynamical properties. The second approach, nonlinear dynamic analysis, applies the geometry of the lattice structure and attempts to wrap the physical particle-lattice interaction into a simple mapping function. Finally, a comparative analysis of the two previously mentioned methods of study is performed to physical experiments on an exact replica of the particle-lattice structure at hand.

DISCRETE ELEMENT SIMULATION AND NONLINEAR DYNAMIC ANALYSIS OF PARTICLES IN A SIMPLE LATTICE STRUCTURE

by
Liam E. Buckley

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Masters of Science in Mechanical Engineering

Department of Mechanical Engineering

May 2003

APPROVAL PAGE

DISCRETE ELEMENT SIMULATION AND NONLINEAR DYNAMIC ANALYSIS OF PARTICLES IN A SIMPLE LATTICE STRUCTURE

Liam E. Buckley

Dr. Anthony D. Rosato, Thesis Advisor — Date
Associate Chairperson for Graduate Studies and
Professor of Mechanical Engineering, NJIT

Dr. Nadine Aubry, Committee Member — Date
Chair and Jacobus Professor of Mechanical Engineering, NJIT

Dr. Denis Blackmore, Committee Member — Date
Professor of Mathematics, NJIT

BIOGRAPHICAL SKETCH

Author: Liam E. Buckley

Degree: Masters of Science

Date: May 2003

Undergraduate and Graduate Education:

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

- Bachelors of Science in Mechanical Engineering
  New Jersey Institute of Technology. Newark, NJ. 2001.

Major: Mechanical Engineering

Presentations and Publications:

Liam Buckley,
"Experimental, Simulation and Nonlinear Dynamic Analysis of Galton's Board" The 39th Annual Technical Meeting of the Society of Engineering Science (SES 2002), State College, Pennsylvania. October, 2002.

Liam Buckley,
"Computer Simulations of a Galton's Board"
The 13th Annual Saint Joseph's University Sigma Xi Student Research Symposium, Philadelphia, Pennsylvania. April, 2002.

To my parents.

ACKNOWLEDGMENT

I would like to thank my advisor Dr. Anthony Rosato, who through his own love of science, engineering and research instilled in me the confidence to look further at the puzzlements of the world. My gratitude and appreciation are also extended to Dr. Denis Blackmore and Dr. Nadine Aubry who provided their wisdom and efforts to the undertaking of this research.

And to my friends and lab partners of the Granular Science Laboratory at NJIT; Ninghua Zhang, Jian Liu, Christopher Oshman and Michael Sweetman, who provided their knowledge and friendship throughout my duration at the lab.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Introduction

The triangular lattice structure that is investigated in this work is an adaptation of the device first introduced by Sir. Francis Galton to investigate the Central Limit Theorem [1]. A planar surface is arranged with staggered pins normal to its surface, and small particles are dropped through the field of pins producing a normal distribution of particles as they exit the system, see Figures 1.1 and 1.2.

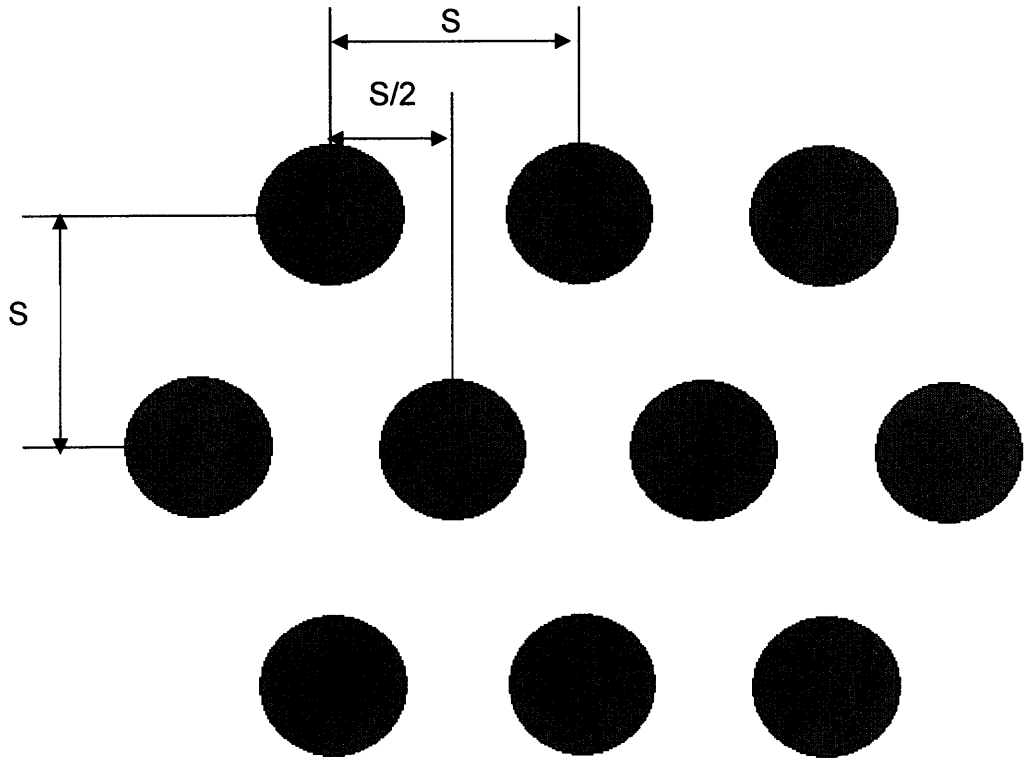

Figure 1.1 Example lattice structure.

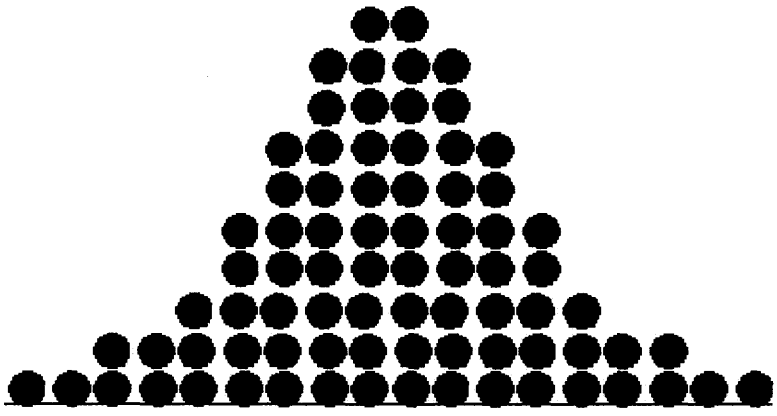

Figure 1.2 Example exit distribution of particles.

The triangular lattice, or Galton's, board has been used to investigate many problems that relate both to statistically determinable sequences and systems that appear to be chaotic in nature. In all of these subsequent investigations, the Law of Large Numbers comes into play. That is, the mean value of a measurable quantity tends to be bounded by a singular value which approaches its "true value" as the number of observations increases. In this work, the notion of the Law of Large Numbers is used to investigate the dynamic properties and diffusive nature of a particle traveling through a triangular lattice structure, i.e. a Galton's board, by using a series of models capable of both simulating the physical interactions within the lattice structure and mimicking the dynamic nature using a nonlinear approach which is thought to be a unique approach to this problem.

1.2 Objective

There are three main objectives accomplished in this work:

- Creation of models capable of accurately predicting the dynamic nature of a particle traveling through a lattice structure
- Validation of the methodology used through qualitative comparison to published work and experimental observation.
- To broaden the understanding of the dynamic nature and phenomenon taking place within the lattice structure.

1.3 Literature Survey

In an effort not to produce redundant research and to gain insight from the vast knowledge base available in the area of Galtons board, a survey of the available literature is appropriate.

One of the most influential and interesting sources of information pertaining to the diffusive nature observed in Galtons board is that of Bridgewater et al. [2]. In a paper appearing in 1969, Bridgewater et al. published results of experiments performed on a cylinder packed with glass spheres. Particles of varying size, but always smaller than those of the packing material, were allowed to percolate down from the center of the cylinder and through the packing medium which would then strike a plate covered in a layer of retarding grease. In this manner, the exit location of the percolating particles was noted within concentric circles about the center of the packed cylinder. Bridgewater et al. then modeled the diffusion process taking place within the packed cylinder to the cylindrical diffusion equation:

$$\frac{\partial n}{\partial t} = E_r\left(\frac{\partial^2 n}{\partial r^2} + \frac{1}{r}\frac{\partial n}{\partial r}\right) = E_r \nabla^2 n(r,t)\,, \tag{1.1}$$

where E_r represents axial diffusion and n the number of particles within the radius r at time t.

A solution of Equation 1.1 assuming that no particles are within the cylinder at time $t<0$ and that the number of particles within the cylinder is fixed for all time is given as:

$$4E_r t = \frac{r^2}{\ln\left(\frac{N_0}{N_0 - N}\right)}, \tag{1.2}$$

where N_0 is the total number of particles within the cylinder and N is the number of particles within a given radius r.

Oshman [3] published a thesis which included experimentally determined diffusion coefficients for varying particle materials traveling through a Galton Board in 2002. Oshmans work was largely an adaptation of Bridgewater et al.'s packed cylinder to that of a Cartesian coordinate system where he made the direct relation of r in (1.2) to displacement in the horizontal direction.

Sergeev et al. presented a paper in 1988 where he undertook a statistical mechanics approach to the diffusive nature of a particle traveling in a lattice structure [4]. A comparative Equation of (1.2) is derived where time in (1.2) is replaced with displacement in the vertical direction. Sergeev et al. further derive a theoretical value for the diffusion equation based upon the packing factor of the lattice structure and the diameters of the particle and pins:

$$D_{xx} = \frac{1}{2N(d_1 + d_2)} \tag{1.3}$$

where D_{xx} is the diffusion or trickle coefficient and d_1 and d_2 are the diameters of the pins and particles respectively.

In 1992, Hoover and Moran [5] modeled the particle pin interactions in a Galton board using an isokinetic method. They produced results which led to the discovery of strange attractors within the triangular lattice structure. Lue and Brenner in a separate paper investigated the notion of strange attractors of the triangular lattice structure in 1993 where they provided limiting cases based upon the collision parameters, namely the restitution coefficient, where the distribution of the particles exit position is not Gaussian [6]. Lue and Brenner further suggest that the notion of a random process taking place within the Galton Board is not true, but rather that the system is a deterministic one governed by Newton's Equations of Motion.

Similar papers have been published [7,8,9] to those of Bridgewater et al. where the size of the pins and particle are varied in order to gain insight into the driving mechanism behind the diffusive nature of Bridgewater et al.'s packed cylinder. In one of particular interests is that of Bruno et al. where the material and size of the percolating particle was determined not to have as great of an effect on the diffusion as that of the make up of the Galton board lattice.

1.4 Arrangement of Thesis

The arrangement of this thesis is as follows: Chapter 1 is an introduction to the work to be presented, Chapter 2 introduces the geometry and models used to study the dynamic nature of a particle traveling through a lattice structure, Chapter 3 introduces and reviews the dynamic and statistical properties that are of concern, Chapter 4 presents the results generated using both of the models presented in Chapter 3 and includes a discussion and conclusion. The following figure is a roadmap of this thesis.

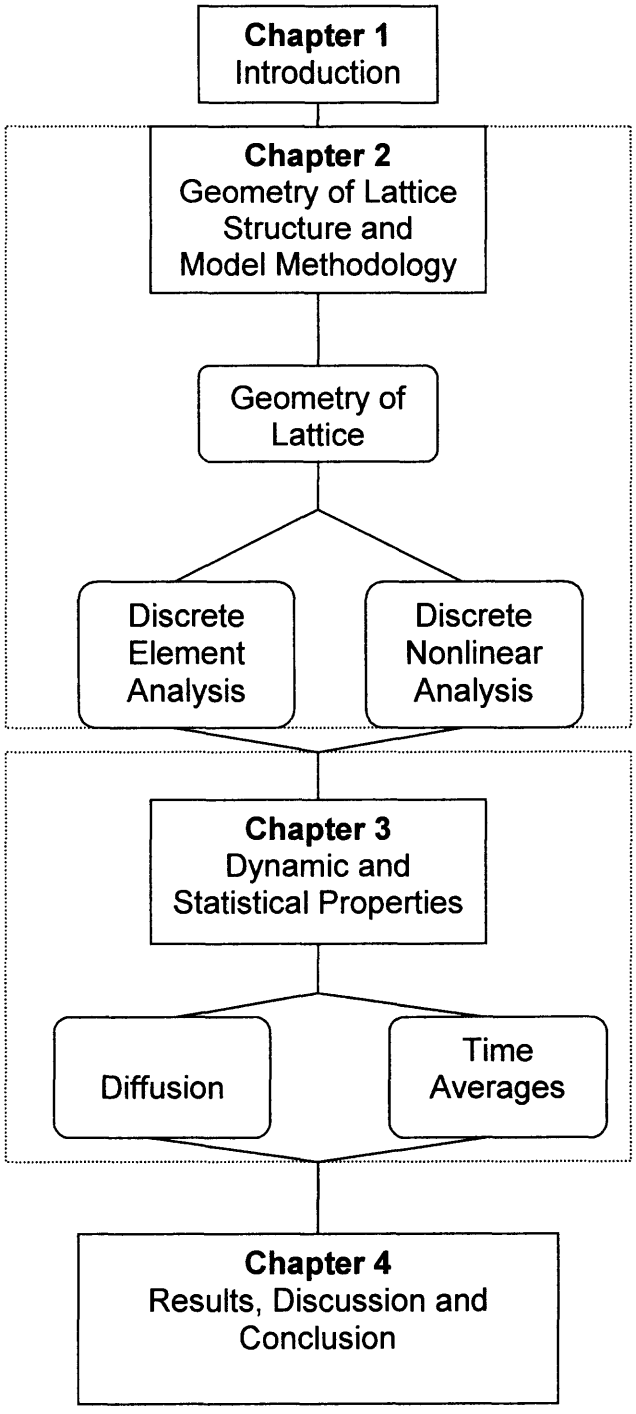

Figure 1.3 Flowchart of thesis.

CHAPTER 2

GEOMETRY OF LATTICE STRUCTURE AND MODEL METHODOLOGY

2.1 Introduction and Lattice Structure Geometry

2.1.1 Introduction

This chapter serves to introduce the reader to the geometry and the two methods used to study a particle traveling through a lattice structure. Section 2.2 introduces the discrete element analysis used and highlights the physical force model that was employed to simulate the pin particle collision. Section 2.3 presents the discrete nonlinear dynamic model that was developed to simulate the pin-particle collisions, and provides an example of the computations necessary to model the system.

2.1.2 Lattice Structure Geometry

The geometry of the lattice structure studied in this work is based upon the Galton board located in the Granular Science Laboratory at the New Jersey Institute of Technology. A schematic of the board is presented in Figure 2.1. The dimension s is defined by the step size of the columns and rows, $s/2$ is the staggered positioning or offset of the $(n+1)$ row of pins. The step size s used for the triangular lattice structure in this research was 0.396875 cm. The diameter of the pins was 0.15875 cm. The diameter of the particle falling through the lattice structure was 0.3175 cm. The width and height of the entire lattice structure was 39.29075 cm. Figure 2.2 is a 3-dimensional representation of a particle traveling through the lattice structure studied.

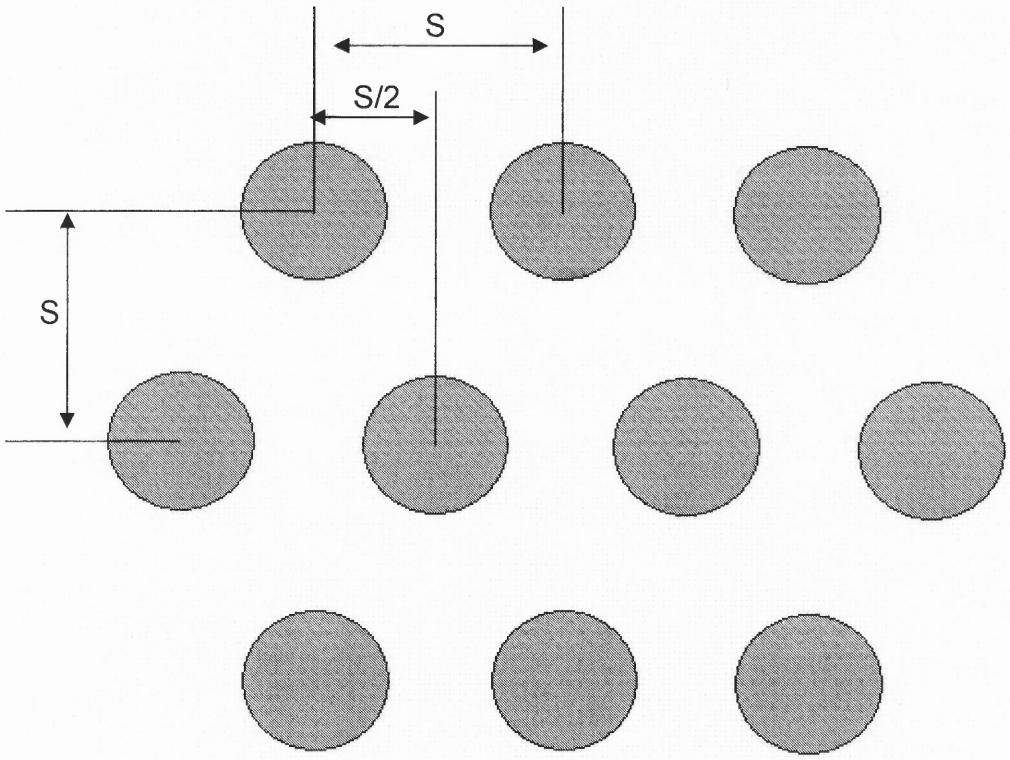

Figure 2.1 Section of lattice structure.

Figure 2.2 3-Dimensional image of a particle traveling through a lattice structure.

2.2 Discrete Element Analysis

2.2.1 Introduction

Discrete Element Analysis is an adaptation of the more widely known analytical tool Molecular Dynamics. Molecular dynamics refers to the solution of an N-body problem by solving the equations of motion using non-energy-dissipation models. Consider the collision of two spheres, in a molecular dynamic simulation the collision of the two spheres would result in no energy loss. That is,

$$(E_{\text{Sphere 1}} + E_{\text{Sphere 2}})_{\text{Before Collision}} = (E_{\text{Sphere 1}} + E_{\text{Sphere 2}})_{\text{After Collision}} . \tag{2.1}$$

Molecular dynamic analysis has been used for many years to solve several interesting problems in both chemistry and biology, but more recently it has been adapted to handle the dissipative nature of physical contact in solids. The adapted version of the molecular dynamics model has come to be known as discrete element analysis. Discrete element analysis is similar to molecular dynamics by the fact that they are both concerned with the solution of an N-body problem, but discrete element analysis incorporates energy dispersion models in the solution. Again consider the collision of two spheres, in a discrete element simulation the collision of the spheres would result in an energy loss. That is,

$$(E_{\text{Sphere 1}} + E_{\text{Sphere 2}})_{\text{Before Collision}} \neq (E_{\text{Sphere 1}} + E_{\text{Sphere 2}})_{\text{After Collision}} , \tag{2.2}$$

where the energy lost would depend upon the nature of the spheres. In most mechanical models of particle collisions the energy loss due to a collision is proportional to a restitution coefficient, where the restitution coefficient is given as

$$e = \frac{v_1 - v_2}{u_1 - u_2} , \tag{2.3}$$

and *u* and *v* refer to the velocities of the two colliding spheres before and after contact, respectively.

2.2.2 Discrete Element Force Model

A major part of any discrete element simulation is the force model used. The force model depicts the nature of the phenomenon being studied, in this case collision or contact between two spheres. The energy-dissipative force model used in this research to simulate the collision of particles in a lattice structure is as follows.

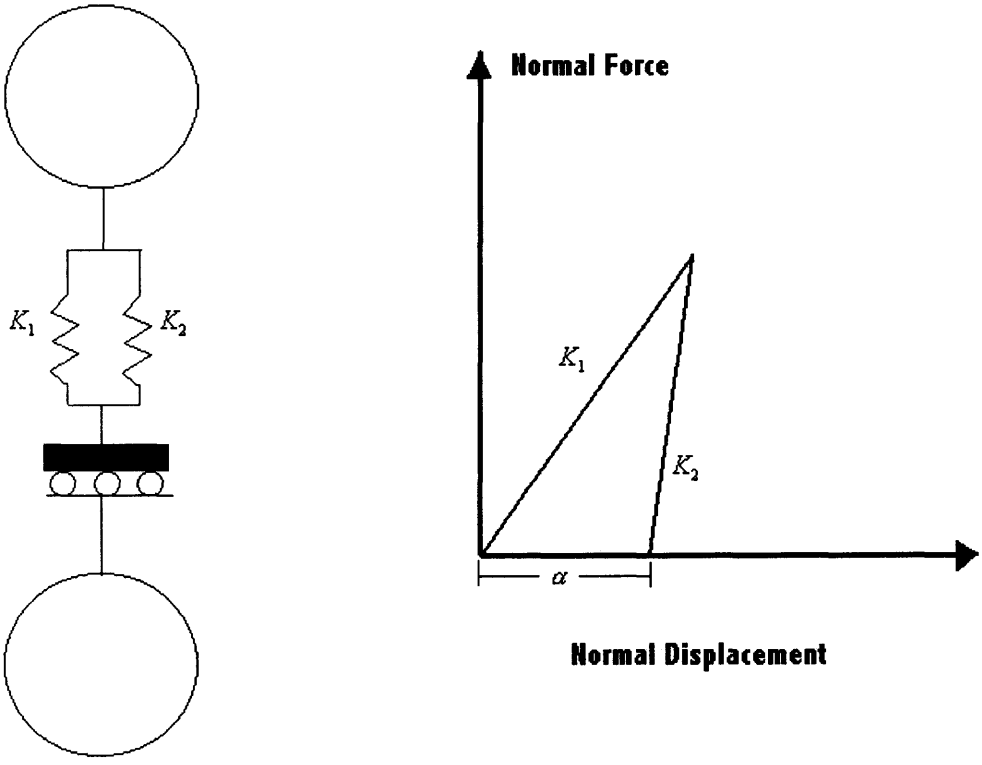

Figure 2.3 Force model for particle collisions.

Where the initial collision, or compression period of the colliding particles is modeled using the linear spring K_1 and the unloading or expansion period of the particles is modeled using the linear spring K_2. This model, which was originally termed the "partially latching spring model" is based on the work of Walton and Braun [10], where they showed that it can approximate the behavior observed in experiments and finite element calculations of colliding spheres. As mentioned previously, the amount of energy dissipated, or lost, during a collision is proportional to the restitution coefficient *e*. A brief derivation of *e* and the energy lost from the above model follows [11]. Consider the collision of two spheres of mass m_1 and m_2 with velocities v_1 and v_2. From the above force model (Figure 2.3), the equations of motion for the two spheres are

$$m_1 \frac{dv_1}{dt} = -F_1 \text{ and } m_2 \frac{dv_2}{dt} = -F . \tag{2.4}$$

The relative velocity of approach is defined as

$$\dot{\alpha} = v_1 + v_2 . \tag{2.5}$$

And the relative acceleration is given by

$$\ddot{\alpha} = -K_1 \alpha \frac{m_1 + m_2}{m_1 m_2} . \tag{2.6}$$

Simplifying (2.6) and making use of the reduced mass identity m_r yields the equation of motion for the normal displacement,

$$\frac{1}{2} d(\dot{\alpha})^2 = -\frac{K_1}{m_r} \alpha d\alpha , \tag{2.7}$$

where

$$m_r = \frac{m_1 m_2}{m_1 + m_2}. \tag{2.8}$$

Now let v_a be the approach velocity when α = 0 and α_m be the maximum deformation when $\dot{\alpha}$ = 0. Integrating (2.7) yields

$$\frac{1}{2}\dot{\alpha}\Big|_{\dot{\alpha}=v_a}^{\dot{\alpha}=0} = -\frac{K_1}{2m_r}\Big|_{\alpha=0}^{\alpha=\alpha_m} = v_a = \alpha_m\sqrt{\frac{K_1}{m_r}}. \tag{2.9}$$

In a similar fashion, considering the unloading period of the collision it is shown that v_s, the relative velocity of separation is

$$v_s = \alpha_m\sqrt{\frac{K_2}{m_r}}. \tag{2.10}$$

Now making use of the identity for the restitution coefficient (2.3), it is shown that *e* is

$$e = \sqrt{\frac{K_1}{K_2}}. \tag{2.11}$$

In terms of kinetic energy, *KE*, the energy lost is

$$KE_{lost} = \frac{1}{2}m_r(1-e^2)(v_a)^2. \tag{2.12}$$

2.2.3 Computer Simulations Using Discrete Element Analysis

The computer implementation of a discrete element simulation amounts to integrating the equations of motion for particles over many thousand time steps using a suitable numerical method. A brief flow chart of a computer program using discrete element analysis follows.

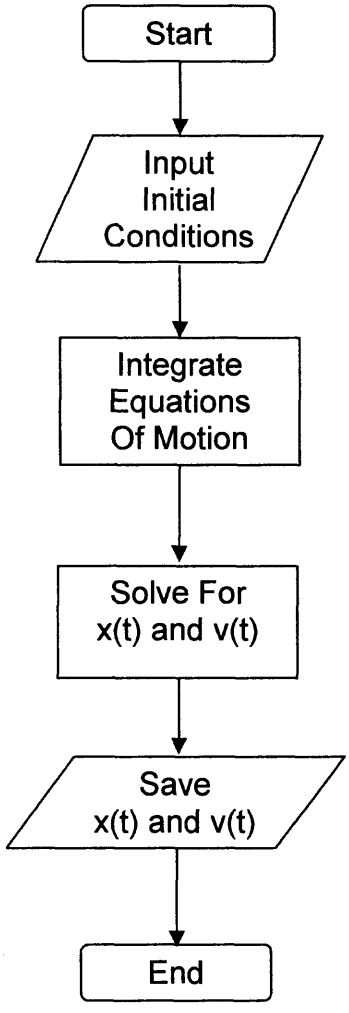

Figure 2.4 Example flowchart of a discrete element analysis program.

The discrete element code used in this study was adapted from previous codes supplied by Dr. Otis R. Walton [12]. The code named ***3dshear***, is a Fortran implementation of the above algorithm. Although not shown in the above flowchart, ***3dshear*** is a complex code containing several functions which search for collisions, calculate particle positions, calculate particle velocities, calculate particle forces, update link lists of near neighbors, determine boundary

conditions, and integrate the equations of motion of all particles in the system. Discrete element simulations are an important asset to the study of such systems as lattice structures because of their ability to provide a complete time-history of a particles dynamical nature while it interacts with the lattice system.

2.3 Nonlinear Dynamic Analysis

2.3.1 Introduction

In this section, an attempt to predict the dynamic nature of a particle traveling through a lattice structure using a nonlinear dynamic model is made. The model, developed by Dr. Denis Blackmore of the Mathematical Sciences Department at New Jersey Institute of Technology [13], wraps the physical phenomenon taking place during the pin-particle interactions into a mapping function capable of predicting relatively accurate representations of the particles trajectory. Further, the model provides reasonable measures of the diffusion coefficient and other dynamical properties. It is beyond the scope of this work to go into a full mathematical formulation and proof of the discrete nonlinear dynamic model due to the rather complicated and lengthy process so a brief description is presented instead.

2.3.2 Simple Rectangular Lattice Arrangment

In an attempt to develop a more complex model of a triangular lattice system, the fundamental approach of the model is laid out in a simpler rectangular arrangement. It is assumed in this rectangular model that all the rows and columns of pins have an equal spacing s creating a regular rectangular lattice, see Figure 2.5.

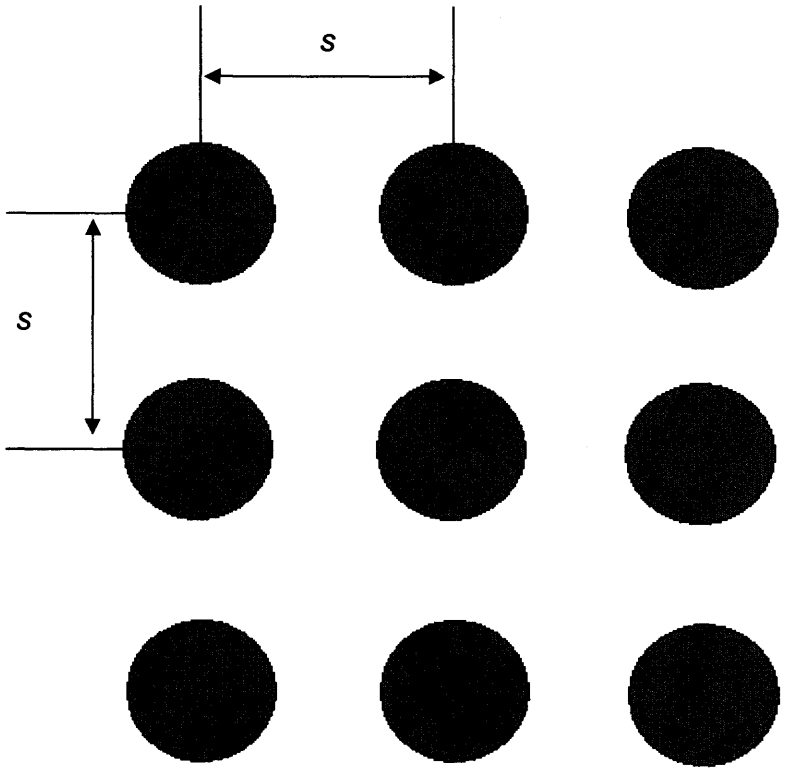

Figure 2.5 Rectangular lattice system.

By assuming an interest in only the distribution of the spheres as they exit the lattice structure, the rows and columns of pins may be extended infinitely, thus creating an iterative approach to calculating the position of the sphere traveling through the regular rectangular lattice system. The assumption of only being interested in the exit distribution allows one to create a mapping function $F:\boldsymbol{R}\rightarrow\boldsymbol{R}$, which has iterates F^n that represent the position of the sphere as it moves past the (n+1) row of pins. The horizontal, or x-direction of the sphere, is unchanged if the sphere does not come into contact with a pin, thus the mapping function F is the identity function at most points within the realm defined by $\boldsymbol{R}$.

In order to mimic, or model, the collision of the sphere and pin in an appropriate and reasonably accurate manner, an investigation of the contact and separation phenomenon taking place should be completed. It is known from previous studies [1,14] that there is approximately a 50% chance that the ball will bounce to the left or to the right of the pin after contact if the collision takes place relatively close to the center of the pin. The direction of the particles separation can to a certain degree of confidence be accurately construed as a function of the particles original contact point [14]. In this sense, it may be assumed that an appropriate means of mathematically modeling the direction and phenomena of separation is a sine function which incorporates a ratio of the relative position of the particle to the pin center and the sum of the particle's radius and the pin's radius. The separation function can be fined tuned further if a variable ($c > 0$) is included to control the amplitude of the sine function, in essence depicting the mechanical and dynamic proprieties of the two colliding materials, see Figure 2.6. The contact model can thus be summarized as

$$p_c(x) \equiv \begin{cases} 0, |x| \geq a \\ c\sin(\frac{\pi x}{a}), |x| < a \end{cases}, \tag{2.13}$$

where x is defined as the particles local position relative to a pin, and a as the difference between the particle radius and that of the pin radius. It is seen in the above formulation that if $|x|$ is greater than or equal to a then no contact occurs and the particle travels on to the next row of pins.

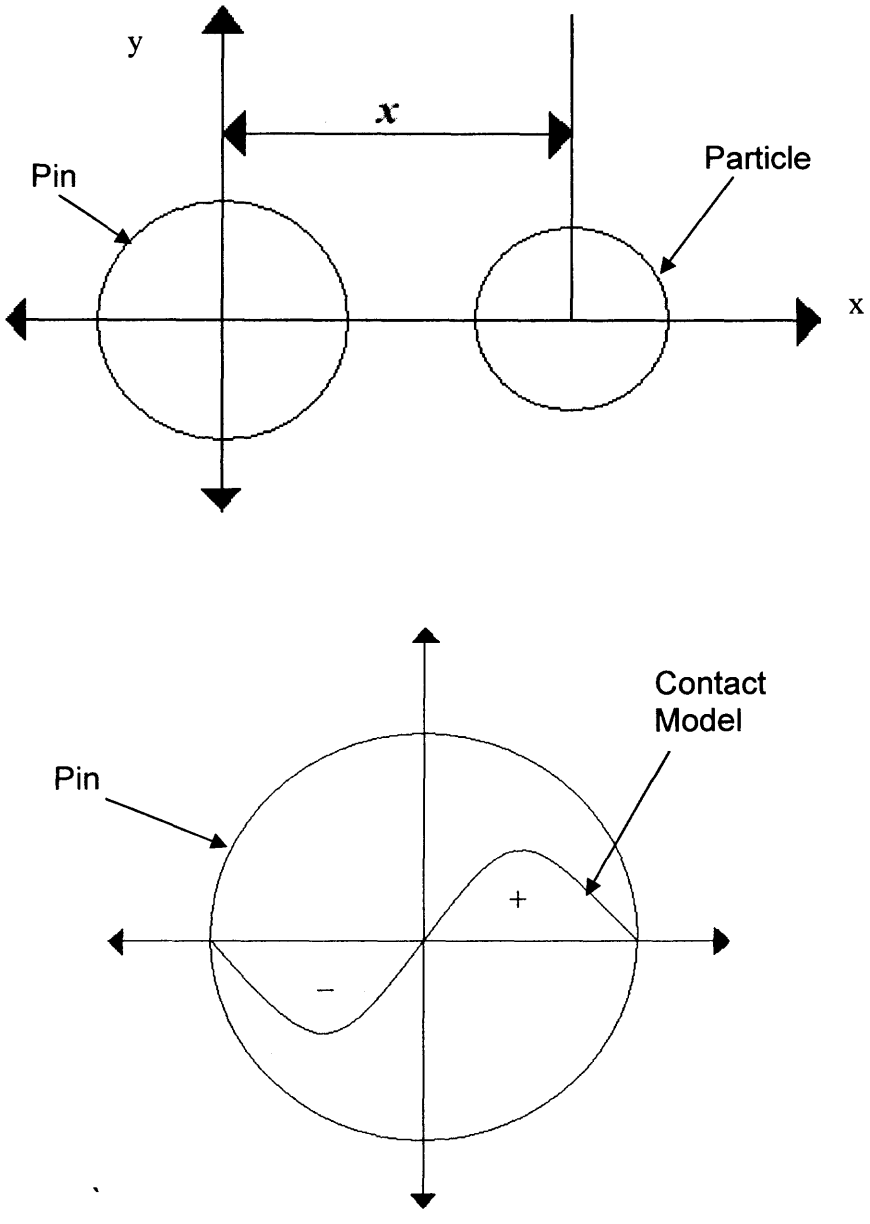

Figure 2.6 Depiction of contact model.

As can be seen from Figure 2.6, if the model bounces to the left of the pin center the contact model returns a negative value which implies a bounce or separation to the left, similarly if contact occurs on the right side of the pin center

the contact model returns a positive value implying a separation to the right. If contact occurs in the region of the pins center the model returns a null value implying no directional separation which under perfect conditions would be a reasonable assumption.

Incorporating the contact model into the mapping function *F* develops the following iterative relationship,

$$F(x) \equiv x + \sum_{n=-\infty}^{\infty} p_c(x - sn), \tag{2.14}$$

where *n* is bounded by the number of columns.

2.3.3 Triangular Lattice Structure

It is now possible to build upon the framework set forth in the rectangular lattice model, and formulize a more complex triangular system. The geometry of the triangular lattice is such that the rows and columns are separated from each other by a distance *s*, but in comparison to that of the rectangular model the next row of pins is offset in the x-direction by an interval of *s/2* from that of the previous row, see Figure 2.7.

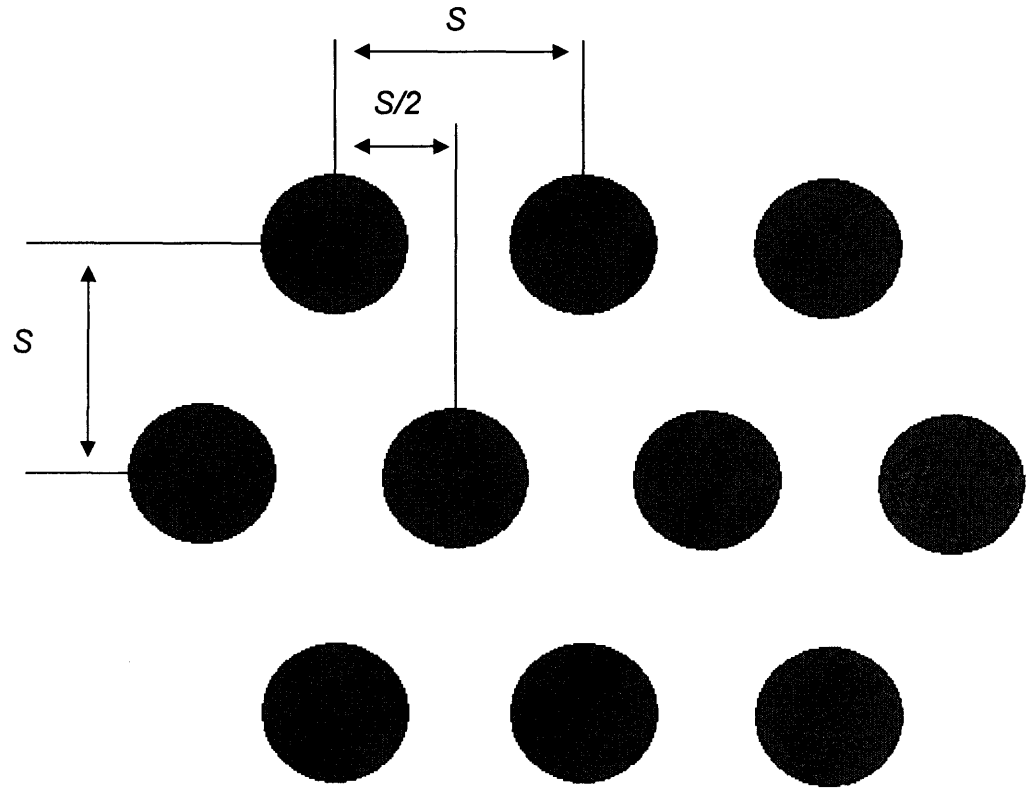

Figure 2.7 Triangular lattice system.

This configuration of pins matches that of the Galton Board previously described in section 2.1.2, see Figure 2.2. As with the rectangular discrete dynamical model presented in the previous section, interest lies solely in the position of the particle in the x-direction as it moves past the (n+1) row. The solution starts by defining a mapping function G having iterates G^n that define the trajectory of the particle. The mapping function can formally be defined as

$$G: \boldsymbol{R} \times \boldsymbol{N} \rightarrow \boldsymbol{R}, \tag{2.15}$$

where $\boldsymbol{N}:=\{0,1,2,3,\ldots\}$.

It can be seen that the mapping function *G* is actually just an adaptation of the rectangular mapping function *F*, and may be construed as merely a phase shift in the rectangular contact model. The formulation of *G* may be summarized as

$$G(x,n) := x + \sum_{m=-\infty}^{\infty} p_c(x - s(n + \frac{1+(-1)^n}{4})), \tag{2.16}$$

where m is bound by the number of rows and n by the number of columns in a row.

An illustrative example of how the discrete dynamical model works is presented. Assume a triangular lattice structure of the form defined by Figure 2.7. The problem begins by assuming a random starting position *P(x,y)* for the particle defined by an x-coordinate and a y-coordinate. The values for all y-coordinate positions in *P* are prescribed by the nature of the lattice structure. The model makes no attempt to define the particles trajectory in the vertical direction, but rather assumes a discrete step in the vertical direction defined by the height of each row in the lattice structure.

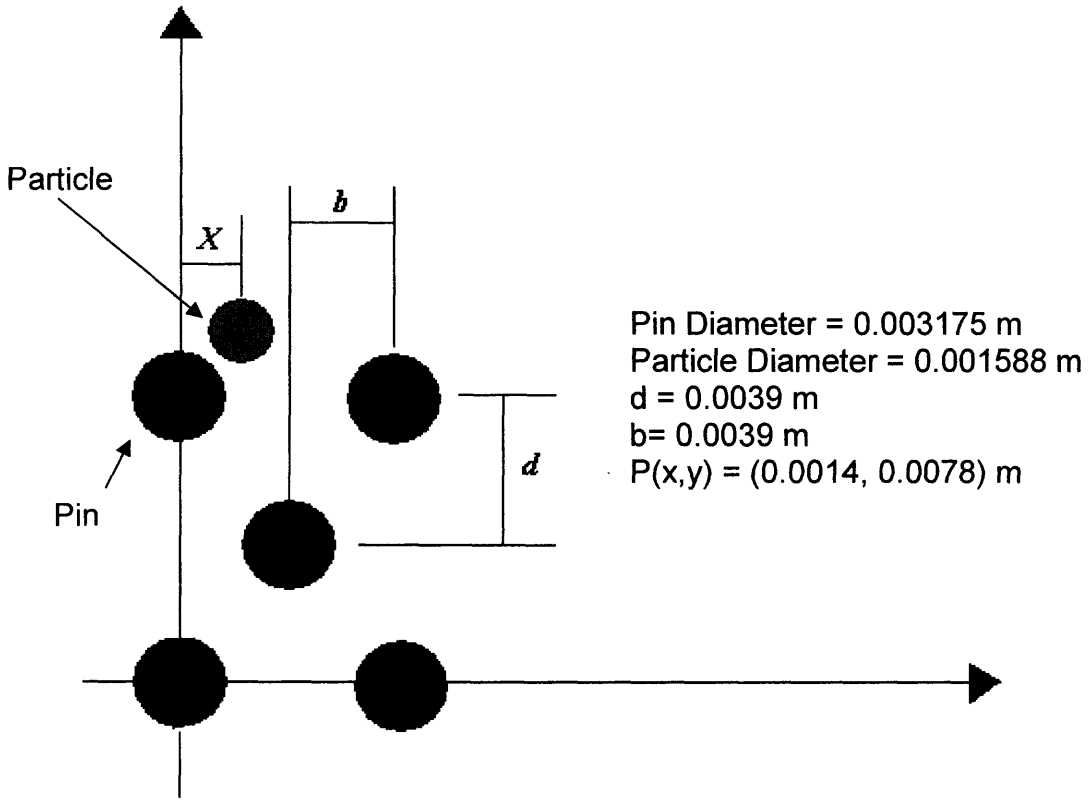

Figure 2.8 Example triangular lattice.

The problem starts by determining whether a collision will occur based upon the value of a, which was defined as the particle radius plus the pin radius.

$$a = \frac{0.001588}{2} + \frac{0.003175}{2} = 0.0023815 \tag{2.17}$$

Since the absolute value of the x-coordinate relative to the colliding pin is less than a it is determined a collision occurs.

$$x = 0.0014$$

$$a = 0.0023815 \tag{2.18}$$

$$x < a$$

Now a separation distance is calculated using the contact model.

$$p_c = c\sin(\frac{\pi x}{a})$$
$$p_c = 0.962144\,c \tag{2.19}$$

Where as stated before, c is a constant that attempts to mimic the physical and dynamical properties of the two colliding spheres. For this example's sake assume c = a, then

$$p_c = 0.002291. \tag{2.20}$$

And finally a new x position is calculated using the mapping function.

$$G(x,n) = x + p_c = 0.0014 + 0.002291 = 0.003691 \tag{2.21}$$

The process repeats now for the second row of pins, the new position of the particle is (0.0037,0.0039) m. The relative location of the particle in the x-direction to any possible pin that it can collide with is -0.0002 m, the only pin in the second row. It is seen again that the particle and pin collide, so the contact model is employed and yields

$$p_c = c\sin(\frac{\pi x}{a})$$
$$p_c = -0.000621. \tag{2.22}$$

And the mapping function yields the new x-coordinate,

$$G(x,n) = x + p_c = 0.0037 - 0.000621 = 0.003079\,. \tag{2.23}$$

2.3.4 Computer Implementation of the Discrete Dynamic Models

The implementation of the presented discrete dynamical models is a rather simple process. First a triangular or rectangular lattice system is defined using algorithms for the placement of pins in the discrete element simulations (see

Appendix A). After the geometry of the Galton Board is defined, a Monte-Carlo approach is taken to orient the particle near the center pin in the top row of rigid scatterers. Using a Monte-Carlo method of locating the particle drop position allows for a nearly infinite number of possible simulation runs. Next, a test is performed to determine if the particle collides with the center pin. If contact occurs, the appropriate separation equation based upon the lattice geometry is used to account for the phenomenon of the actual collision. If no contact occurs the horizontal position of the ball is unchanged. An incremental step equal to the spacing distance of rows is added to the particles vertical position to account for travel in the direction of gravity. The process is repeated until the user specified number of rows has been accounted for. The following flowchart (Figure 2.4) provides an illustration of the computer code. Computer code for the triangular nonlinear dynamic model is presented in Appendix A.

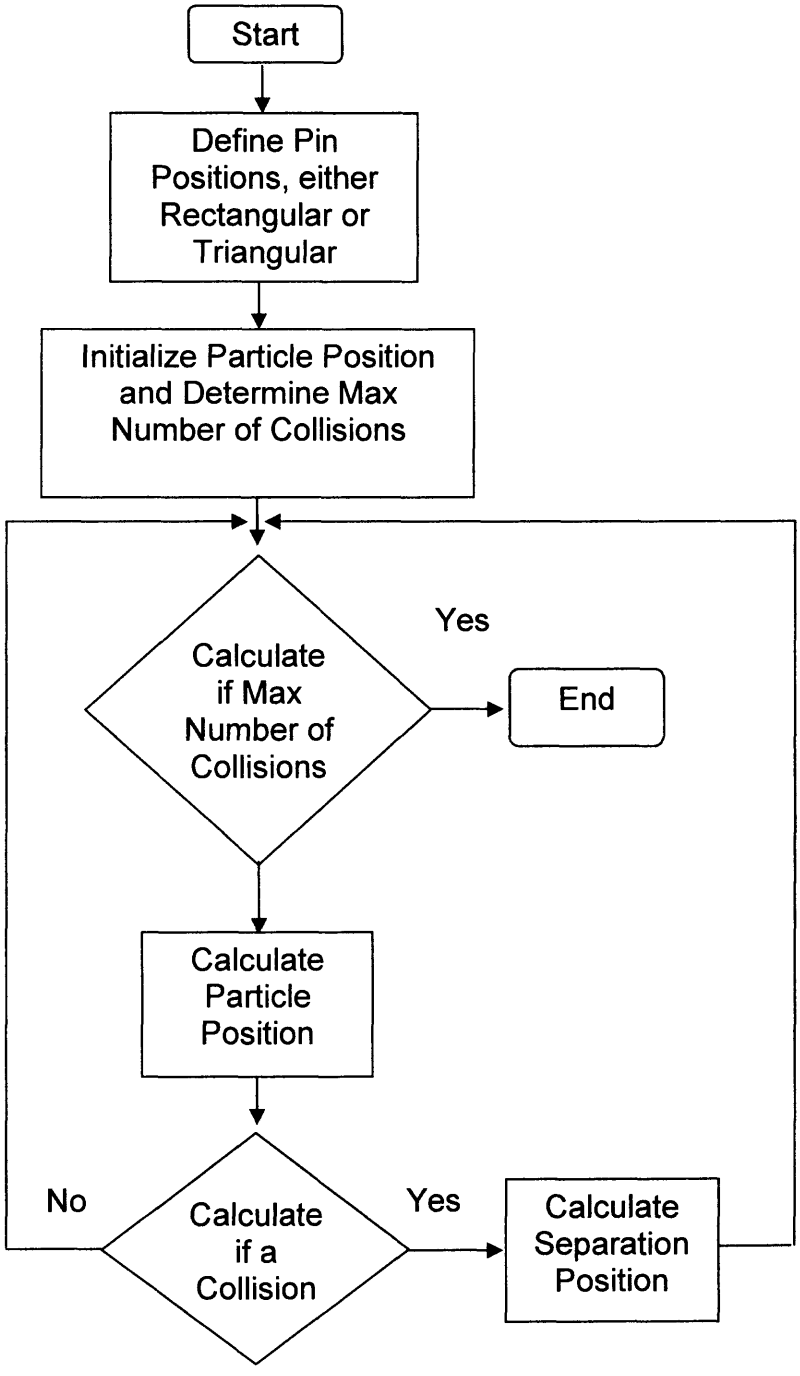

Figure 2.9 Flowchart of a nonlinear discrete dynamic model.

CHAPTER 3

DYNAMIC AND STATISTICAL PROPERTIES

3.1 Introduction

This chapter introduces the dynamic and statistical properties which were studied during this work. The primary dynamic quantities studied were the diffusion of the particle in the lattice structure and velocities of the particle in both the horizontal and vertical directions. Statistical properties studied included histograms of the particle positions as they exited the lattice structure, average residence time, and the correlation time of the particle.

3.2 Dynamic Properties

3.2.1 Diffusion Coefficient

The primary dynamic concern when analyzing the data produced by both the discrete element and the discrete nonlinear dynamic simulations was the particles diffusion within the lattice structure. Experimentally observing a particle or sphere traveling through a Galton Board one notes it's chaotic and random procession. Diffusion can be thought of as the spread of the particle through the lattice system and is an important transport phenomenon associated with the non-uniformity in the composition of the system being studied. Two methods were employed in order to determine the diffusion coefficient in the discrete element simulations, mean square displacement and velocity autocorrelation. The diffusion mechanism was calculated in the discrete nonlinear dynamic

simulations by making use of the exit position histogram (discussed later in section 3.3.2) and the mean square displacement method.

3.2.2 Mean Square Displacement

The mean square displacement method of determining the diffusion coefficient relies upon the Einstein expression for self-diffusion. A brief derivation of the Einstein relation is presented [11]. Suppose at time $t = 0$ a particle is located within a small volume centered at $\mathbf{r} = 0$. At time t the probability of finding the particle at $\mathbf{r}$ is denoted by $G_s(\mathbf{r},t)$, which obeys a diffusion equation by

$$\frac{d}{dt}G_s(\mathbf{r},t) = D\Delta^2 G_s(\mathbf{r},t)\,, \tag{3.1}$$

where D is the self-diffusion coefficient. Under isotropic assumptions, a solution of (3.1) subject to the initial condition that $G_s(\mathbf{r},0) = \delta\mathbf{r}$ is

$$G_s(\mathbf{r},t) = \frac{1}{(4\pi Dt)^{\frac{3}{2}}}\exp(-\frac{r^2}{4Dt})\,. \tag{3.2}$$

Since Equation (3.2) is the probability of finding the particle in a region $d\mathbf{r}$ at time t given that the particle was at the origin at time $t = 0$, the mean square displacement $\left\langle|\mathbf{r}|^2\right\rangle$ can be computed as the expected value of $\mathbf{r}^2$ and is given as

$$\left\langle|\mathbf{r}|^2\right\rangle = \int_0^{\infty}\mathbf{r}^2 G_s(\mathbf{r},t)4\pi\mathbf{r}^2 d\mathbf{r}\,. \tag{3.3}$$

Substituting (3.2) into (3.3), and integrating yields the Einstein relation,

$$2tD = \frac{1}{3}\left\langle\left[\mathbf{r}(t)-\mathbf{r}(0)\right]^2\right\rangle. \tag{3.4}$$

Thus the self-diffusion coefficient D can be computed from the mean square displacement as

$$D = \lim_{t\to\infty} \frac{1}{6t} \left\langle \left[\mathbf{r}(t) - \mathbf{r}(0)\right]^2 \right\rangle . \tag{3.5}$$

If the process being studied is ergodic the ensemble average in the mean square displacement can be replaced by a time average. Equation (3.5) can be further simplified by expanding it by its vector components yielding

$$\left\langle \left[\mathbf{r}\right]^2 \right\rangle = \sum_{t=1}^{t_{max}} \left\{ \left[x(t+\tau) - x(t)\right]^2 + \left[y(t+\tau) - y(t)\right]^2 + \left[z(t+\tau) - z(t)\right]^2 \right\} . \tag{3.6}$$

For the cases mentioned in this work, only the diffusion in the horizontal direction was studied. Reducing (3.6) to only contain diffusion along the x-axis yields

$$\left\langle \left[r_x\right]^2 \right\rangle = \sum_{t=1}^{t_{max}} \left\{ \left[x(t+\tau) - x(t)\right]^2 \right\} . \tag{3.7}$$

Thus the diffusion coefficient in the horizontal direction is given by

$$D_{xx} = \lim_{t\to\infty} \frac{1}{6t} \sum_{t=1}^{t_{max}} \left\{ \left[x(t+\tau) - x(t)\right]^2 \right\} . \tag{3.8}$$

As can be seen from Equation (3.8), the self-diffusion coefficient in the horizontal direction is proportional to the limiting slope of the mean square displacement as *t* approaches infinity. Since the discrete dynamic model makes no attempt to correlate time to the particle's trajectory, time is replaced in (3.8) a step in the height of the board.

3.2.3 Velocity Autocorrelation

Autocorrelation is a method used in signal processing and time series analysis to determine the correlation of a signal with itself. Consider the function denoted by *z(t)*, the autocorrelation function of *z(t)* is given as

$$R(t) = \langle z(t)z(t+\tau)\rangle . \tag{3.9}$$

Equation (3.9) can be thought of as the product of the function *z(t)* with itself at time *t* and at time *(t+τ)* averaged over a certain number of experiments. As with the mean square displacement, (3.9) can be simplified if the process being studied is ergodic and is given as

$$R(t) = \frac{1}{t_{\max}} \sum_{\tau=1}^{\tau_{\max}} z(t)z(t+\tau) . \tag{3.10}$$

The diffusion coefficient is related to the autocorrelation function by the relation

$$D = \frac{1}{3}\int_0^{\infty} d\tau \langle \mathbf{v}(t) \bullet \mathbf{v}(o)\rangle , \tag{3.11}$$

where the vector **v** is the velocity vector of the particle. Again assuming an ergodic process, (3.11) can be simplified by expanding it by its components and substitution of (3.10).

$$D = \frac{1}{3}\sum_{\tau=1}^{\tau_{\max}} \left\{ v_x(t)v_x(t+\tau) + v_y(t)v_y(t+\tau) + v_z(t)v_z(t+\tau) \right\} \tag{3.12}$$

As with the mean square displacement, this research was only concerned with diffusion in the x direction, thus (3.12) reduces to

$$D_{xx} = \frac{1}{3}\sum_{\tau=1}^{\tau_{\max}} \left\{ v_x(t)v_x(t+\tau) \right\} . \tag{3.13}$$

As seen from (3.11), the diffusion coefficient is proportional to the integral of the velocity autocorrelation function.

3.2.4 Computer Implementation of the Mean Square Displacement and Velocity Autocorrelation

Both the mean square displacement and autocorrelation function were implemented using the same basic algorithm to carry out the calculations. For this reason only the algorithm for the velocity autocorrelation is shown below.

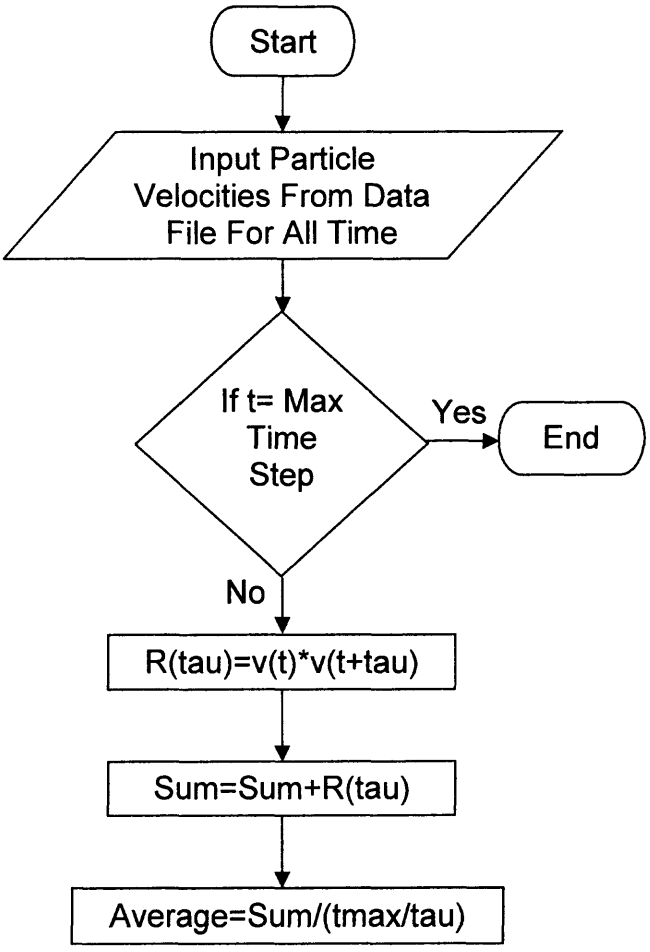

Figure 3.1 Flowchart of velocity autocorrelation function.

The integral of the velocity autocorrelation was completed using a basic trapezoidal method. The limiting slope of the mean square displacement was calculated using Microsoft Excel and a linear fit at the trailing end of the mean square displacement versus time shift τ curve. It is important to note that one must average the computed values of both MSD and VACF over all the simulations completed in order to accurately calculate a diffusion coefficient. Further, the number of simulations performed has a direct influence upon the accuracy of the calculated diffusion coefficient. For this reason the value calculated using the VACF method was approximated by determining a convergence time and ignoring the residual chaotic tail in the VACF vs. Tau curve. Although it has been suggested that this chaotic tail may have a contribution to the diffusion coefficient [15], it was determined by the author that integrating only up to the approximate convergence produces more realistic results. This dependence upon a convergence time and the number of experiments has little bearing upon the value calculated by means of the MSD method [15] so the value calculated should approximate the "true" diffusion coefficient with fewer simulations. Computer codes for both the mean square displacement and velocity autocorrelation methods are included in Appendix B

3.3 Statistical Quantities

3.3.1 Introduction

From a statistical standpoint two specific quantities come to mind when dealing with particles traveling through a lattice system, distributions and averages. The fundamental distributions of concern are the particles exit trajectory and residence time. From an average perspective, the primary concern lies with velocities and residence time and how they relate to theoretical and experimental values.

3.3.2 Exit Distribution

The exit distribution is a collection of several particle positions as they exit the lattice structure grouped into similar horizontal regions. The exit distribution of particles has been a topic of such systems as lattice structures for many years and provides a rather unique perspective into the mixing phenomenon that is taking place, it allows one to look at the process and determine if it follows a Gaussian distribution or some other form.

It has also been shown in previous work [2] that the distribution of particles as they exit such systems is directly related to the diffusion coefficient and can be calculated from such distributions using the following equation:

$$n = \frac{N_0}{4\pi D_{xx} t} \exp\left(\frac{-r^2}{4D_x t}\right), \tag{3.14}$$

where:

$$\begin{aligned} n &= \text{Number of Particles Per Unit Area,} \\ N_0 &= \text{Total Number of Particles,} \end{aligned} \tag{3.15}$$

t = time,

D_{xx} = Diffusion Coefficient In the X-Direction,

r = Horizontal Distance.

Equation (3.14) can be further simplified making use of the identity for the number of particles within a given radius,

$$N = \int_0^r 2\pi rndr, \tag{3.16}$$

and is given as

$$4D_{xx}t = \frac{r^2}{\ln(\frac{N_0}{N_0 - N})}. \tag{3.17}$$

This method of calculating the diffusion coefficient was used to correlate the constant c in the discrete nonlinear dynamic analysis presented in the previous section to the diffusion coefficient.

3.3.3 Averages

Three primary averages were considered in this work, x-velocity, y-velocity and residence time. The x-velocity and y-velocity for each simulation ran during the discrete element analysis were computed and compared to those experimentally determined by Oshman [3]. X-velocity was determined using the following formula:

$$V_x = r/t, \tag{3.18}$$

where V_x is the horizontal velocity component, r is the distance traveled in the x-direction from its starting point and t is the residence time, or the time the particle is resident in the lattice structure. The vertical component of velocity was calculated in the same manner,

$$V_y = H/t, \tag{3.19}$$

where V_y is the horizontal velocity component, H is the distance traveled in the y-direction and t is the residence time.

A theoretical maximum value of V_y can be calculated making use of the potential and kinetic energy equations and is given as

$$V_y = \sqrt{(2gH)}, \tag{3.20}$$

where g is the gravitational constant. In a similar manner a theoretical minimum value for the residence time can be calculated as

$$t = \sqrt{(\frac{H}{2g})}. \tag{3.21}$$

Residence times in the discrete element model were calculated by making use of the time-history available from the output of the simulations and then averaging over the total number of simulations.

CHAPTER 4

RESULTS, DISCUSSION AND CONCLUSION

4.1 Introduction

4.1.1 Introduction

This chapter reviews the results of the completed simulations for both the discrete element model and the nonlinear dynamic model. Due to the length of computer processing time it takes to run the discrete element model, only two case studies were performed. One case study for a 90° lattice angle, and one for a 70° lattice angle relative to the horizontal axis. All results for the discrete nonlinear model represent a 90° lattice angle. The following figure provides a road map to the presentation of the results.

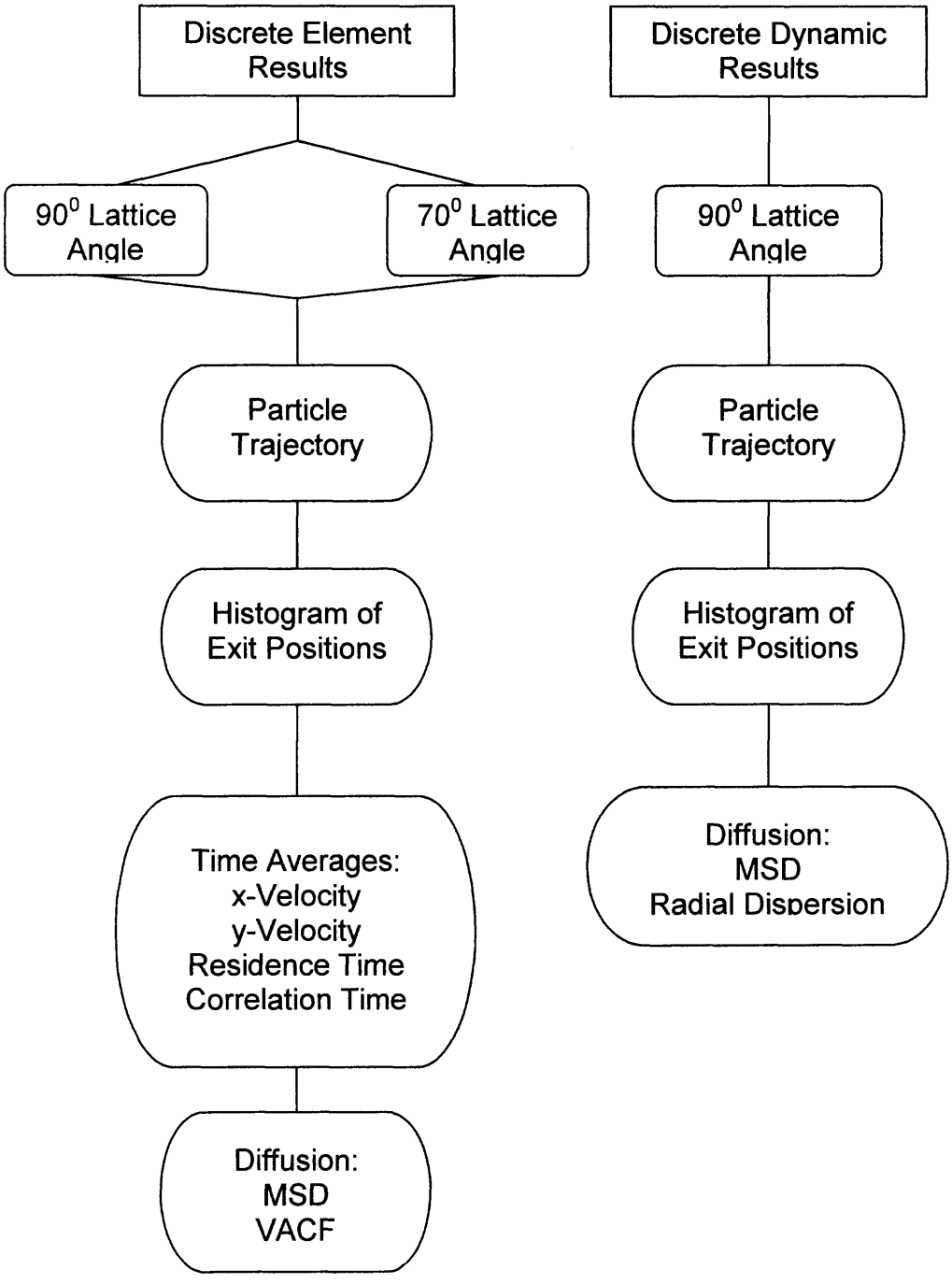

Figure 4.1 Roadmap of discrete element and discrete nonlinear dynamic results.

4.2 Discrete Element Results

4.2.1 Results for the 90° Lattice Angle

Table 4.1 lists the parameters used during this study for a lattice angle of 90°, the dimensions of the pins and particle represent actual dimensions of the Galton Board located in the Granular Science Laboratory at NJIT and are equivalent to the values used by Oshman [3].

Table 4.1 Parameters for a 90° Lattice Angle

Restitution Coefficient	0.60
Gravity in x, y, z	0, 9.81 m/s^2, 0
Number of Pins	5000
Particle Diameter	0.003175 m
Pin Diameter	0.001588 m
Number of Runs	1500

The easiest results to both generate and interpret are those of the particles trajectory in the lattice structure and the histogram of particle exit positions. Figure 4.2 is a sample trajectory of a particle falling through the lattice structure. The residence time for this particle was 6.84 seconds. Figure 4.3 is a histogram of 1500 particle exit positions.

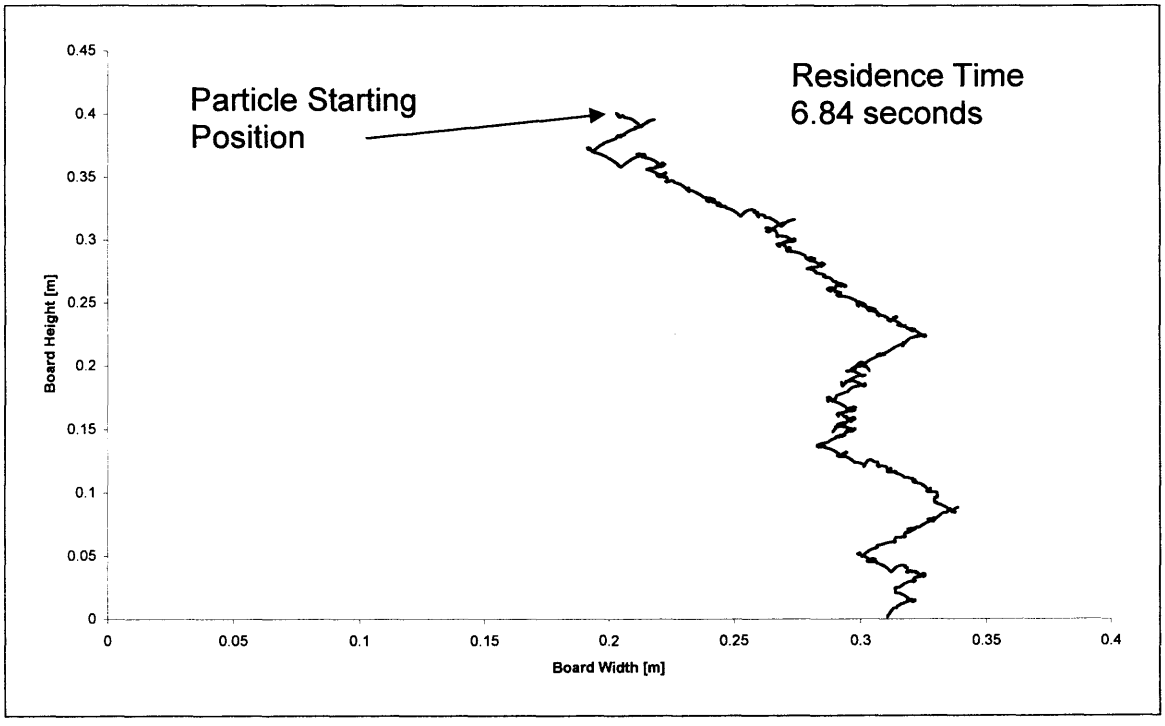

Figure 4.2 Sample particle trajectory at a 90^0 lattice angle.

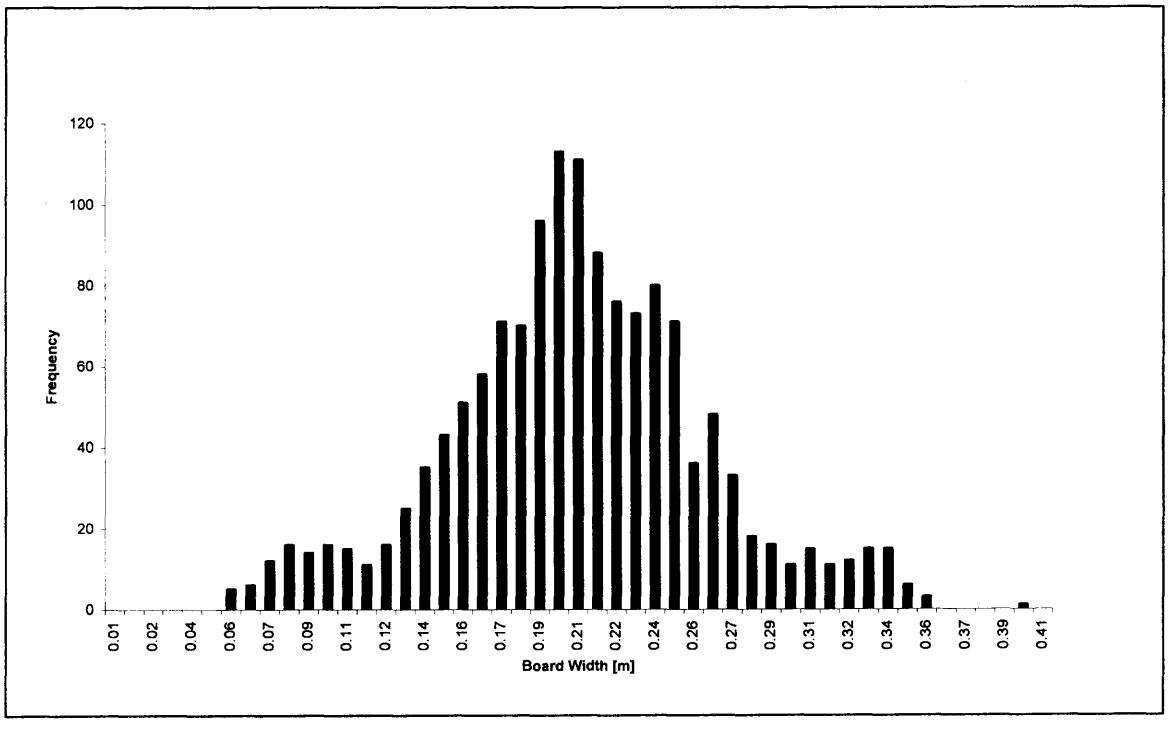

Figure 4.3 Histogram of 1500 particle exit positions for a 90^0 lattice angle.

Table 4.2 summarizes the time averages for the 90^0 lattice angle.

Table 4.2 Time Averages for a 90^0 Lattice Angle

x-Velocity	-0.041 cm/s
y-Velocity	5.92 cm/s
Residence Time	6.7 seconds
Correlation Time	0.0038 seconds

The plot of MSD vs. Time Shift τ is shown in Figure 4.4 for all 1500 simulation runs, The graph of VACF vs. Time Shift τ is Shown in Figure 4.5. The respective Diffusion coefficients for each method are summarized in Table 4.3.

Table 4.3 Summary of Calculated Diffusion Coefficients for a 90^0 Lattice Angle

MSD	1.48 cm^2/s
VACF	1.3 cm^2/s

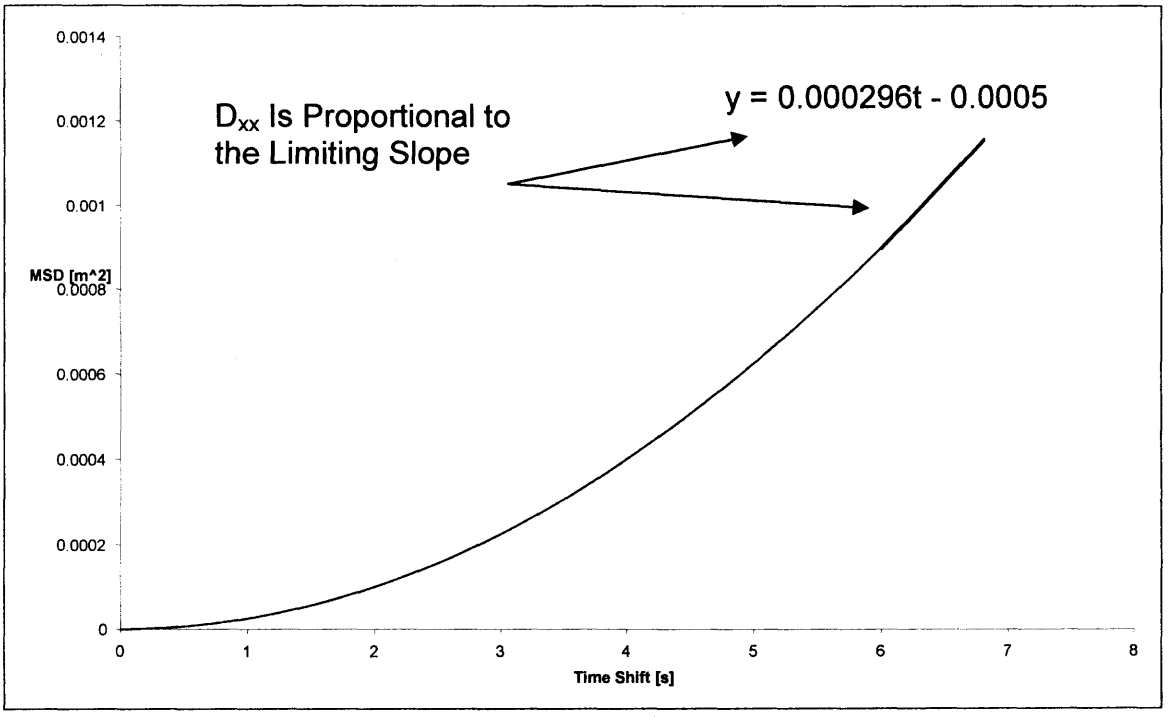

Figure 4.4 MSD vs. Time Shift τ for a 90^0 lattice angle.

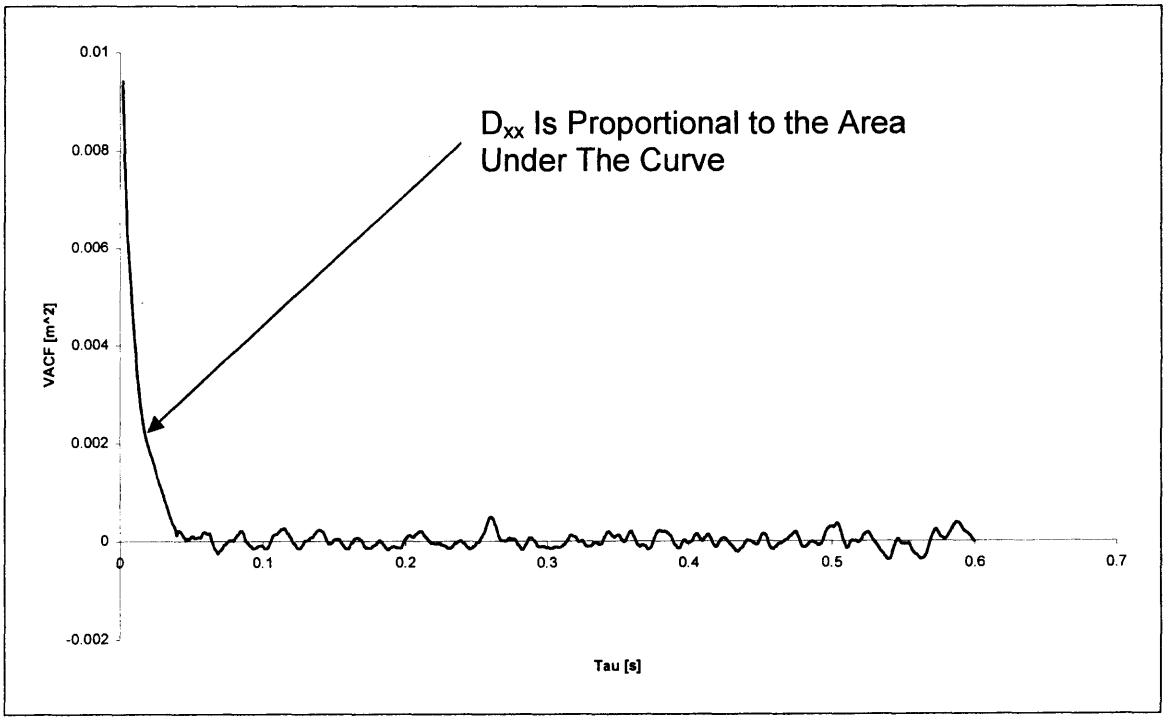

Figure 4.5 VACF vs. Time Shift τ for a 90^0 lattice angle.

4.2.2 Results for the 70° Lattice Angle

Table 4.4 lists the parameters used during this study for a lattice angle of 70°.

Table 4.4 Parameters for a 70° Lattice Angle

Restitution Coefficient	0.60
Gravity in x, y, z	0, 9.22 m/s^2, 3.36 m/s^2
Number of Pins	5000
Particle Diameter	0.003175 m
Pin Diameter	0.001588 m
Number of Runs	1500

Figure 4.5 is a sample trajectory of a particle falling through the lattice structure. The residence time for this particle was 7.52 seconds. Figure 4.6 is a histogram of 1500 particle exit positions.

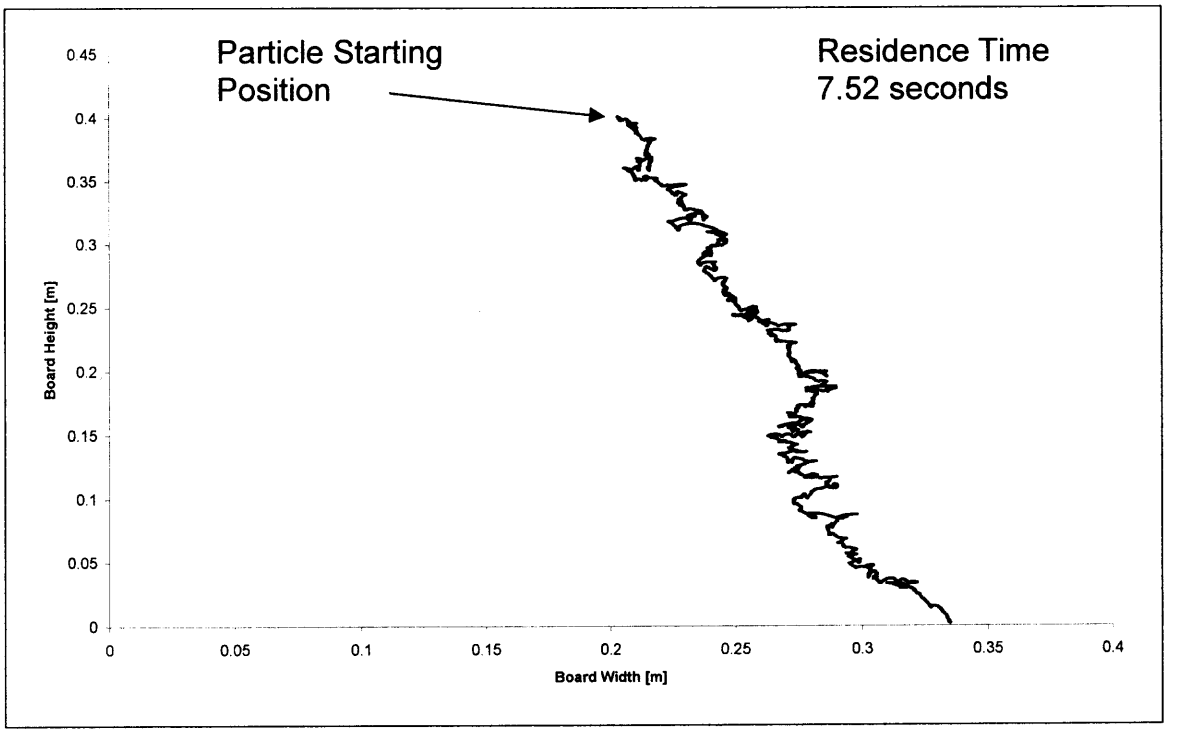

Figure 4.6 Sample particle trajectory at a 70^0 lattice angle.

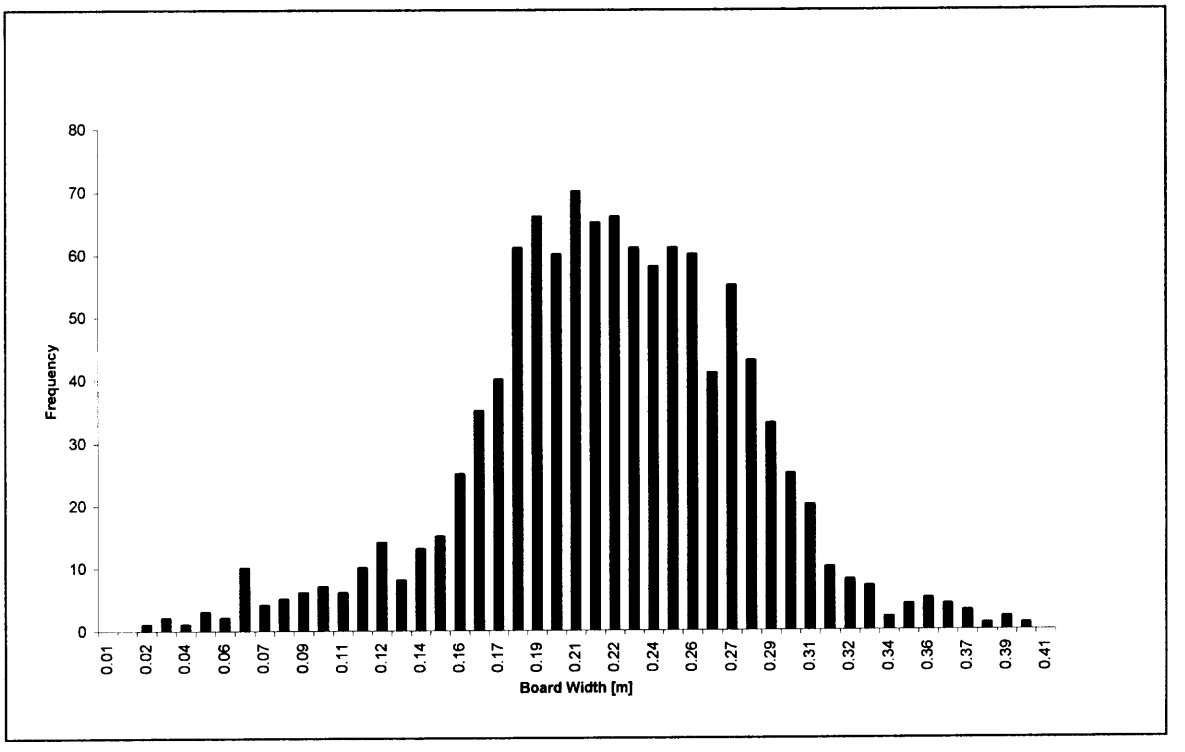

Figure 4.7 Histogram of 1500 particle exit positions for a 70^0 lattice angle.

Table 4.5 summarizes the time averages for the 70^0 board angle.

Table 4.5 Time Averages for a 70^0 Lattice Angle

x-Velocity	0.025 cm/s
y-Velocity	5.6 cm/s
Residence Time	7.12 seconds
Correlation Time	0.0042 seconds

The plot of MSD vs. Time Shift τ is shown in Figure 4.7 for all 1500 simulation runs, the graph of VACF vs. Time Shift τ is Shown in Figure 4.8. The respective Diffusion coefficients for each method are summarized in Table 4.6.

Table 4.6 Summary of Calculated Diffusion Coefficients for a 70^0 Lattice Angle

MSD	2.43 cm^2/s
VACF	2.19 cm^2/s

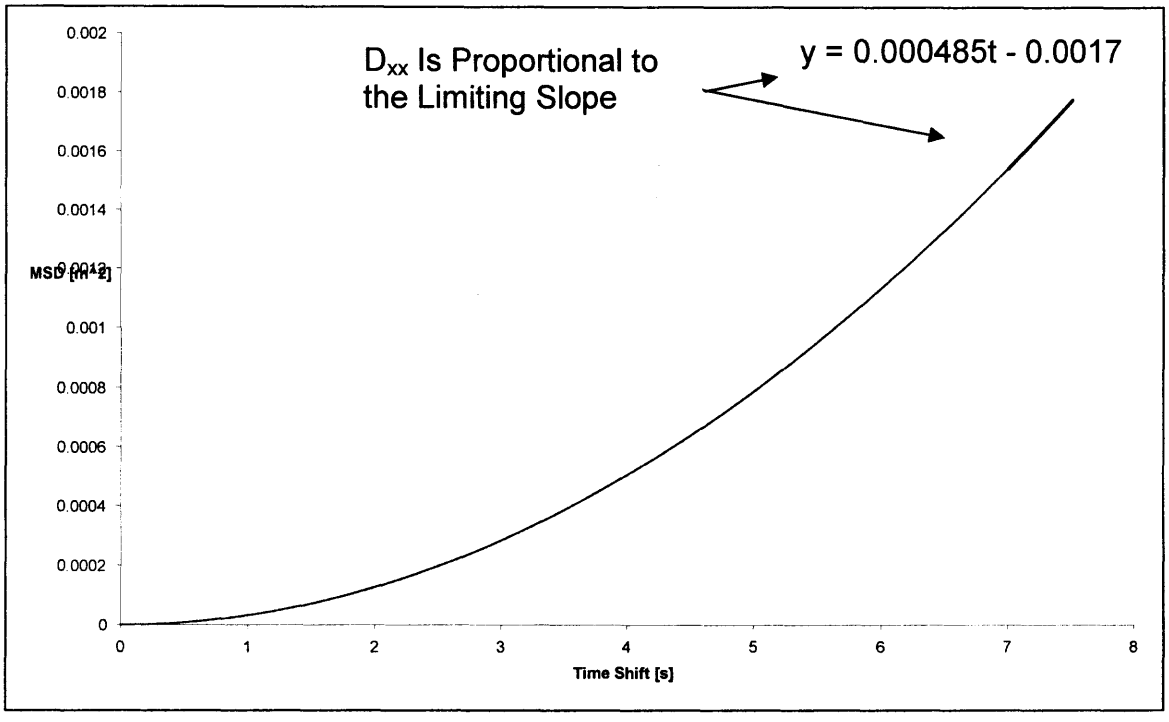

Figure 4.8 MSD vs. Time Shift τ for a 70^0 lattice angle.

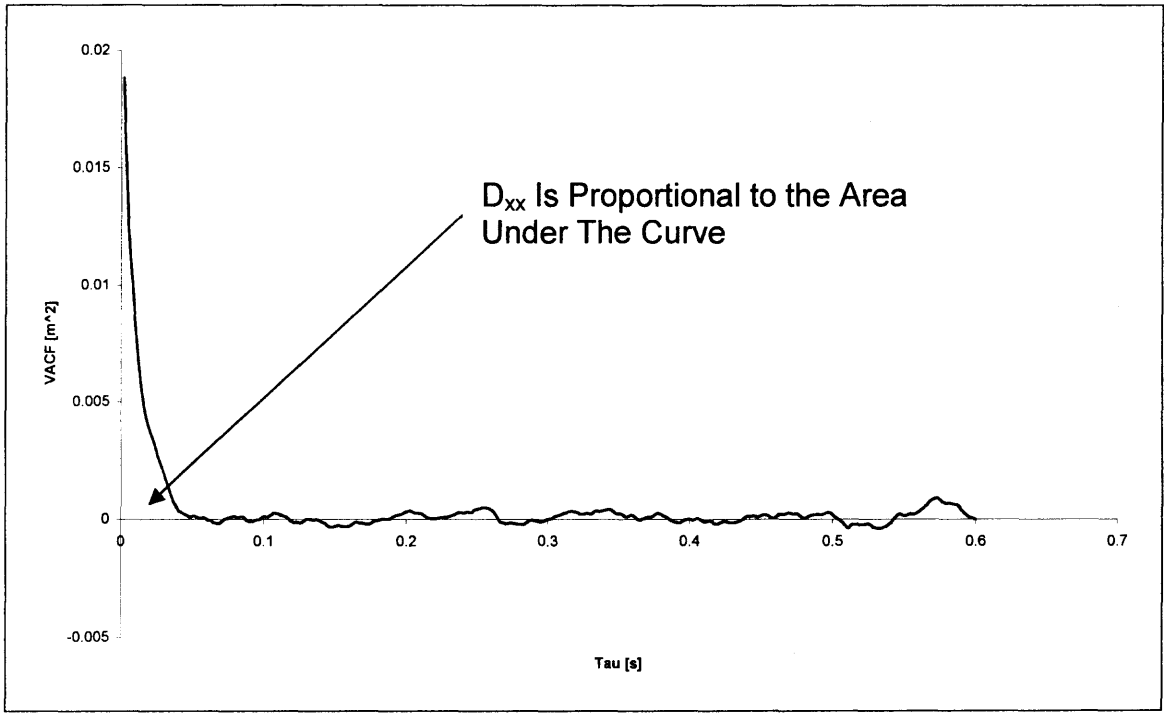

Figure 4.9 VACF vs. Time Shift τ for a 70^0 lattice angle.

4.3 Discrete Nonlinear Dynamic Results

4.3.1 Results for the 90° Lattice Angle

Figure 4.7 is an example of a normal trajectory from the Discrete Nonlinear Dynamic simulations (C=0.4).

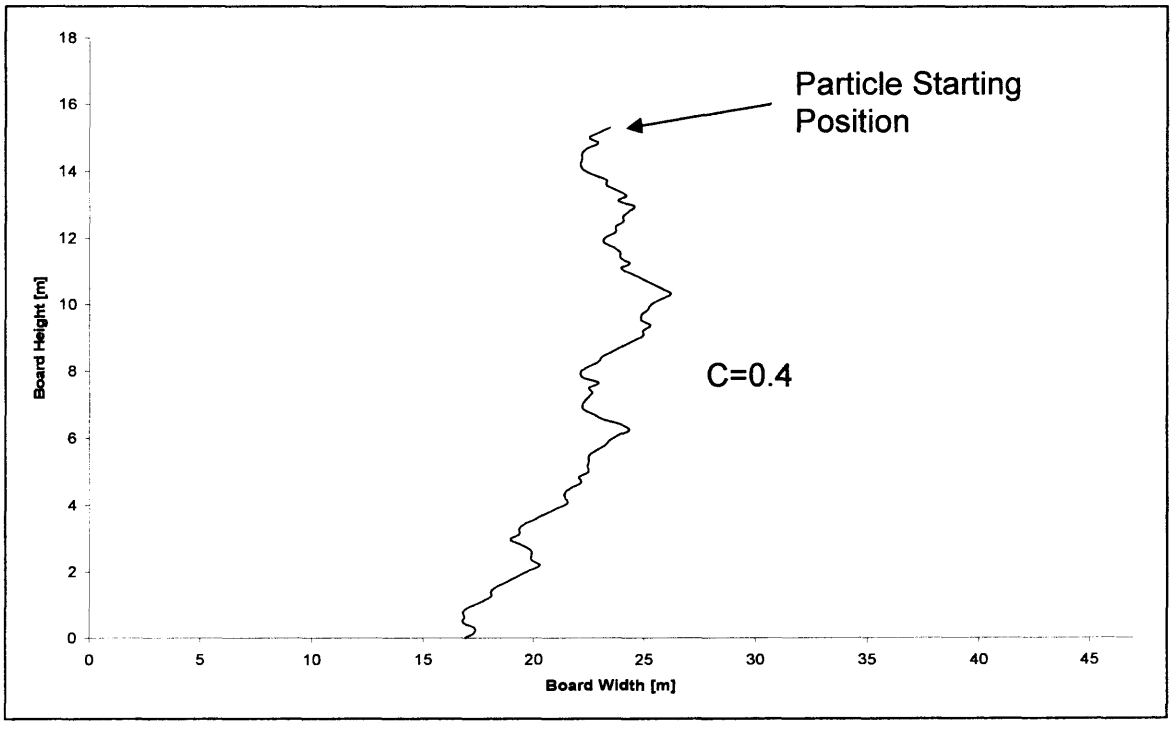

Figure 4.10 Example trajectory from the discrete nonlinear dynamic model.

Figure 4.8 represents two trajectories for which the constant C produces a periodic trajectory (C=0.08 and C=0.015)

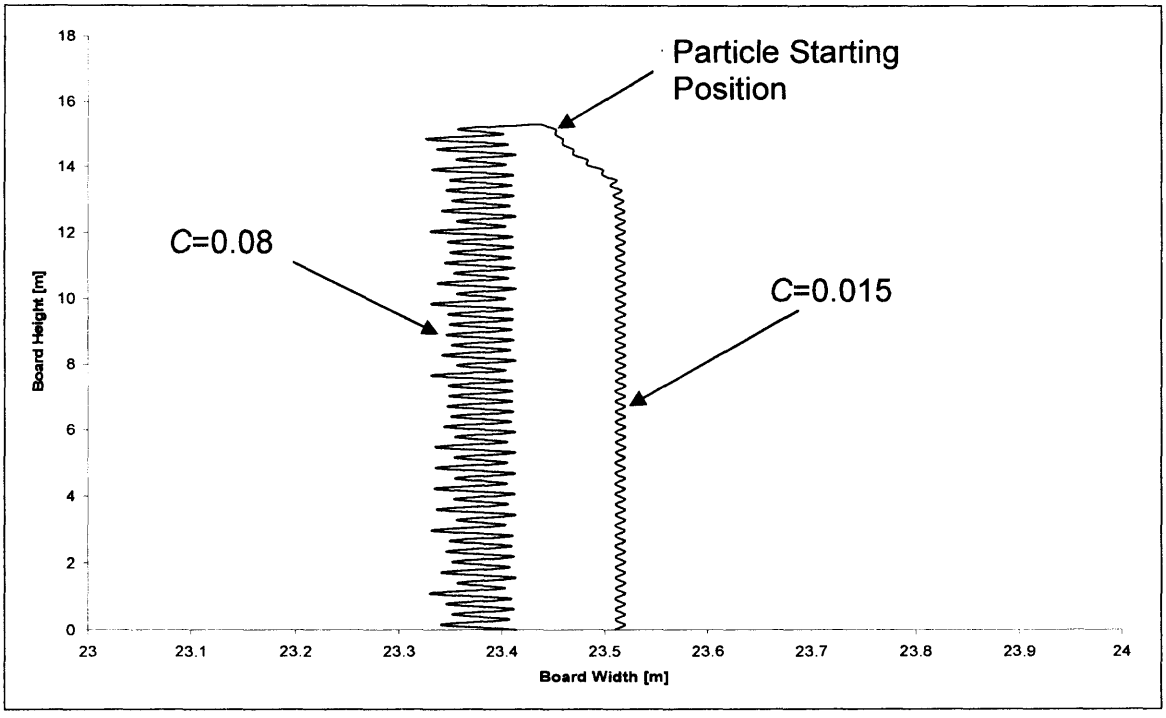

Figure 4.11 Examples of periodic trajectories.

Figure 4.9 is a histogram of 10000 particle exit positions where the constant *C=0.42* produces nearly normal distributions. Figure 4.10 is a histogram of 1000 particle exit positions where the constant *C*=0.015 produces non-normal results.

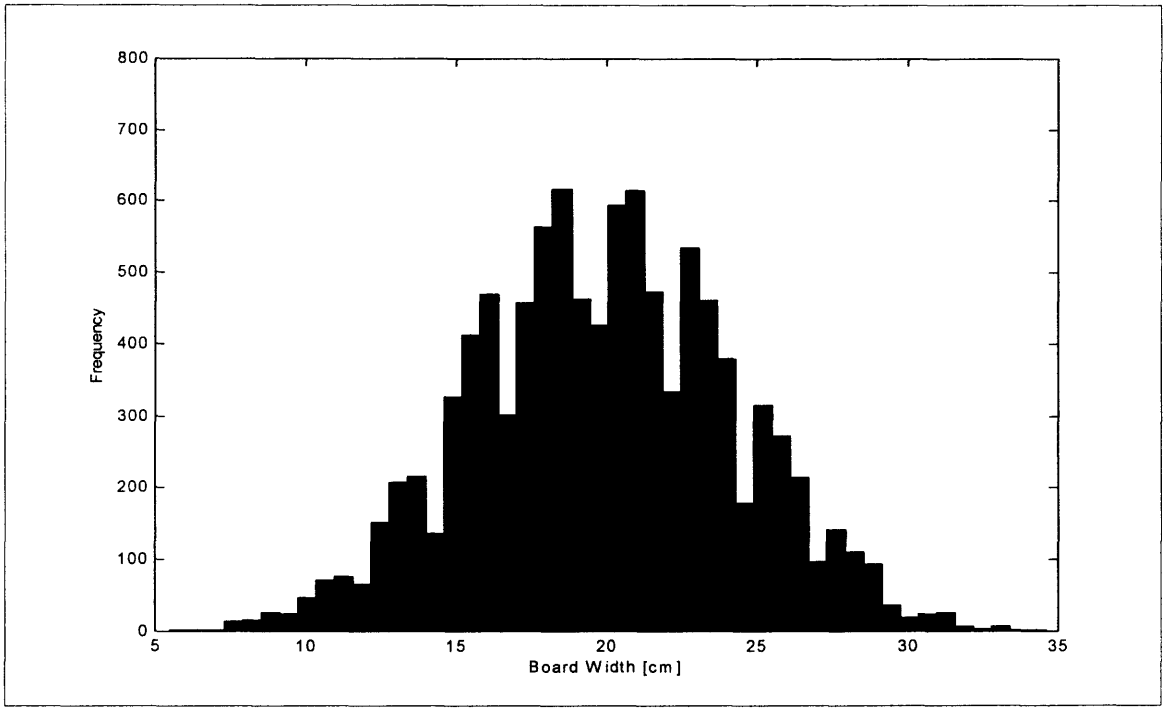

Figure 4.12 10000 particle exit position histogram (board width 39.69 cm).

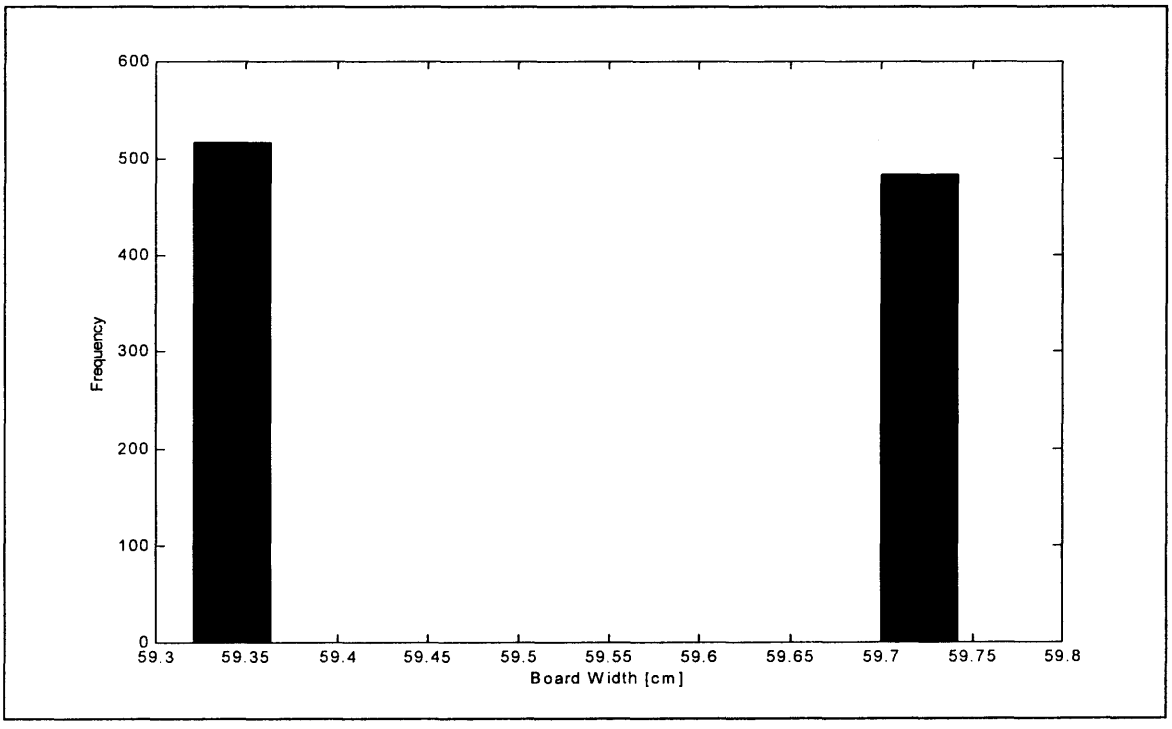

Figure 4.13 1000 particle exit position histogram (board width 119.06 cm).

Figure 4.11 Is an Example MSD vs. Vertical Shift plot for the Discrete Nonlinear Dynamic Simulations (*C*=0.3).

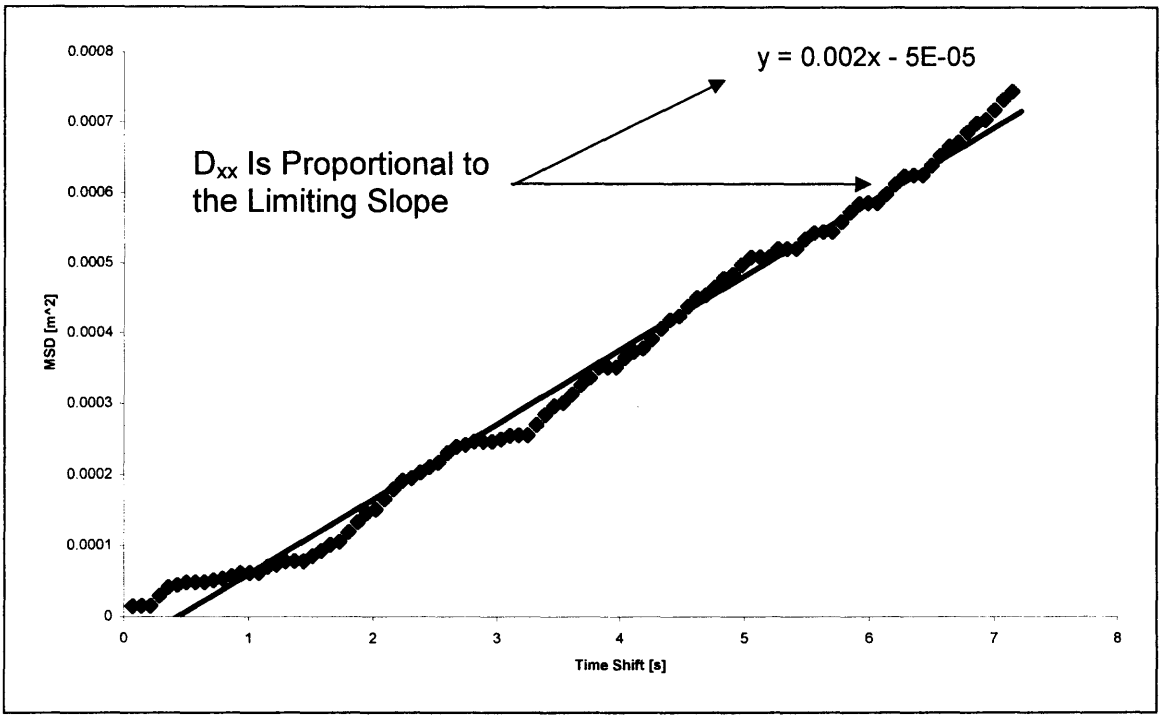

Figure 4.14 Example MSD vs. Time Shift plot (C=0.3).

Figure 4.12 is an example r^2 vs. $\ln(N_0/(N_0-N))$ used in the radial method to calculate diffusion. Figure 4.13 is a plot of the constant *C* vs. the slope of r^2 vs. $\ln(N_0/(N_0-N))$ which is proportional to the diffusion coefficient.

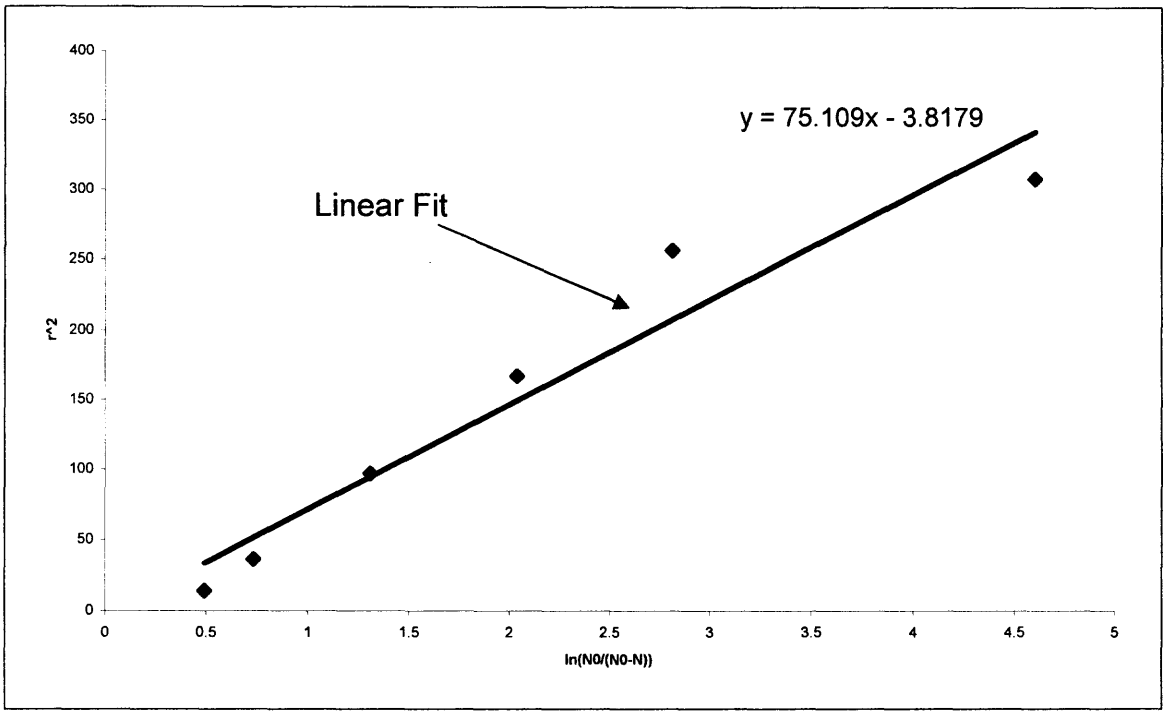

Figure 4.15 Example r^2 vs. $\ln(N_0/(N_0-N))$ (C=0.6).

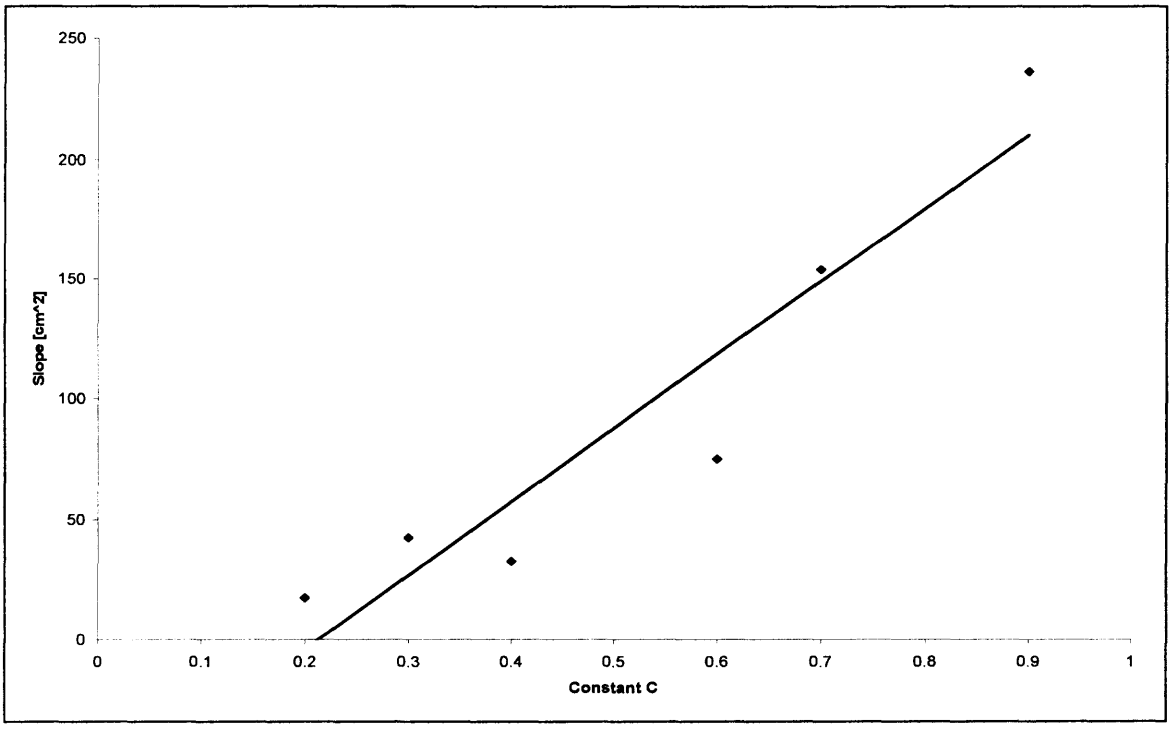

Figure 4.16 Slope of [r^2 vs. $\ln(N_0/(N_0-N))$] vs. Constant C.

Table 4.7 Summarizes' the values of the slope over various constant *C* values.

Table 4.7 Values of Slope of [r^2 vs. $\ln(N_0/(N_0-N))$] for Various Values of *C*

Constant *C*	**Slope [cm^2]**
0.2	17.39
0.3	42.489
0.4	32.61
0.5	151.82
0.6	75.109
0.7	153.99
0.8	453.63
0.9	236.36

4.4 Discussion and Conclusion

4.4.1 Discussion

The discussion follows the same format as the presentation of the results, essentially following the same roadmap laid out in Figure 4.1. As can be seen from Figures 4.2 and 4.5, the trajectories of the particle traveling through the lattice structure for 90^0 and 70^0 degrees, respectively, for the discrete element model appear to be chaotic in nature and thus the model represents actual observations quite well. This result, although trivial in nature, has serious consequences when the nonlinear model results are discussed. Figure 4.8, a sample trajectory from the discrete nonlinear model, represents a trajectory for which the dynamics failed to produce random walks. The resulting exit distribution of particles in cases such as this produce U-shaped curves (see Figure 4.10) instead of the desired bell shaped curve usually associated with a normal distribution for which the Galton board is known to produce [1,4,7,8,9,14]. The periodic trajectories, which are evident in Figure 4.8, were predicted in [13] and are also prevalent in the work of Lue and Brenner [6] for which they associate particular elasticity and pin density envelopes with periodic regimes. It is possible to avoid these periodic regimes following the scheme laid out in [13] for bounding the constant *C*. Equation 4.1 predicts at which values of *C* normal distributions become prevalent,

$$\frac{2a}{\pi}\cos^{-1}\left(\frac{-a}{c\pi}\right)+2c\sqrt{1-\left(\frac{a}{c\pi}\right)^2}=s+2a. \tag{4.1}$$

Where as with the equations presented in Chapter 2, *a* is equivalent to the sum of the pin and particle radii. Table 4.8 is a parametric tabulation of (4.1), note that $s+2a=1.27$ cm.

Table 4.8 Tabulation of Equation (4.1)

C	*S+2a*
0.1	0.86
0.2	1.25
0.3	1.64
0.4	2.04
0.5	2.43
0.6	2.83
0.7	3.23
0.8	3.63
0.9	4.03
1.0	4.43

As can be seen from Table 4.8 the first appropriate value of *C* is greater than 0.2. In the sense of elasticity, the constant *C* seems to correspond to the dimensionless restitution coefficient. Further proof of the correlation between *C* and the restitution coefficient *e* can be drawn from the functionality of the constant *C*. Looking back at section 2.3.2 the primary role of *C* was to restrict the amplitude of the nonlinear contact model in order to approximate different

material types, although a direct relation is cautioned due to the nonlinearity and lack of a time equivalent in the model. Further discussion on selection of an appropriate value for *C* follows when diffusion is discussed.

The histograms of the two trial cases performed in the discrete dynamic model appear to be nearly normal in shape, although Figure 4.3, the histogram for the 90^0 board angle, raises questions in the tails of the distribution. The lack of particles in the further regions of the Galton board may be due to the obliquity of the board, i.e. the driving gravity field may have restricted significant development of the expected random walk thus bounding the distribution. Unfortunately experimental data is lacking at this board angle to draw any conclusion on the accuracy of the distribution.

Calculated time averages for the discrete element model appear to be in line with those of the values experimentally observed by Oshman [3], further they fall within the theoretical approximations presented in Chapter 3 (maximum y-velocity=2.79 m/s, minimum residence time=0.14 seconds). The correlation times, which are a function of the particles velocity history, also seem to be plausible based upon the chaotic trajectories and the similarities in the two values at both board angles.

The calculated values of the diffusion coefficient for the discrete element model have two implications. The first being that they both are relatively close to the values experimentally observed by Oshman [3] (1.68 cm/s^2 for steel particles at a 70^0 board angle), and the second is that they are within a close approximation of each other. The first implication reveals that the discrete

element model is capable of accurately predicting the diffusive nature of the lattice structure, the second gives credence to the two separate methods used to calculate diffusion and the theory behind them. The values of the diffusion coefficient calculated using the discrete nonlinear model are slightly harder to interpret. The lack of a time basis in the nonlinear model results in a non-time based diffusion coefficient, although the slopes presented in Table 4.7 fall within range of those predicted by Oshman [3]. Further, the slope of the mean square displacement curve in 4.11 is proportional to that of the curves in Figures 4.4 and 4.7. This proportionality is an important point due to the linear relationship between velocity, displacement and time, it can be shown that the slope of the MSD curve in Figure 4.11 would be equivalent to that of a time based curve and thus diffusion can be calculated by assuming a proper time value for the residence time (t).

$$D_{xx} = \frac{Slope}{2t} \tag{4.2}$$

In a similar fashion a diffusion coefficient for the values reported in Table 4.7 can be calculated using (4.3),

$$D_{xx} = \frac{Slope}{4t}. \tag{4.3}$$

The problem now lies in the selection of a constant C value to properly mimic the system.

Before selecting a C value, a discussion on the effect of the width and height of the board is relevant. A reasonable question to ask is what effect the

dimensions of the board have upon the values calculated, particularly that of the diffusion coefficient. A parametric approach was taken to this problem where the height of the board was varied as 1*h*, 2*h*, 3*h*, 4*h*, 5*h* and 6*h*, where *h* is equal to the height of the Galton board at NJIT (*h*=0.3929075 m). It was found that the normalized standard deviation (normalized by the height of the board i.e. 1,2,3...N), which is proportional to the diffusion coefficient, remained constant over all board heights. The width of the board was increased accordingly as to avoid bounding the resulting distribution. Results from this study are plotted in Figure 4.14. The same train of thought applies to the number of particle drops that make up the histograms. A similar study to that of the varying board height was performed using successive number of particle drops to test if the system was truly ergodic. The results of the ergodicity test yielded similar results, which are plotted on Figure 4.15. An analogous study was performed with the discrete element model which resulted in the same conclusion.

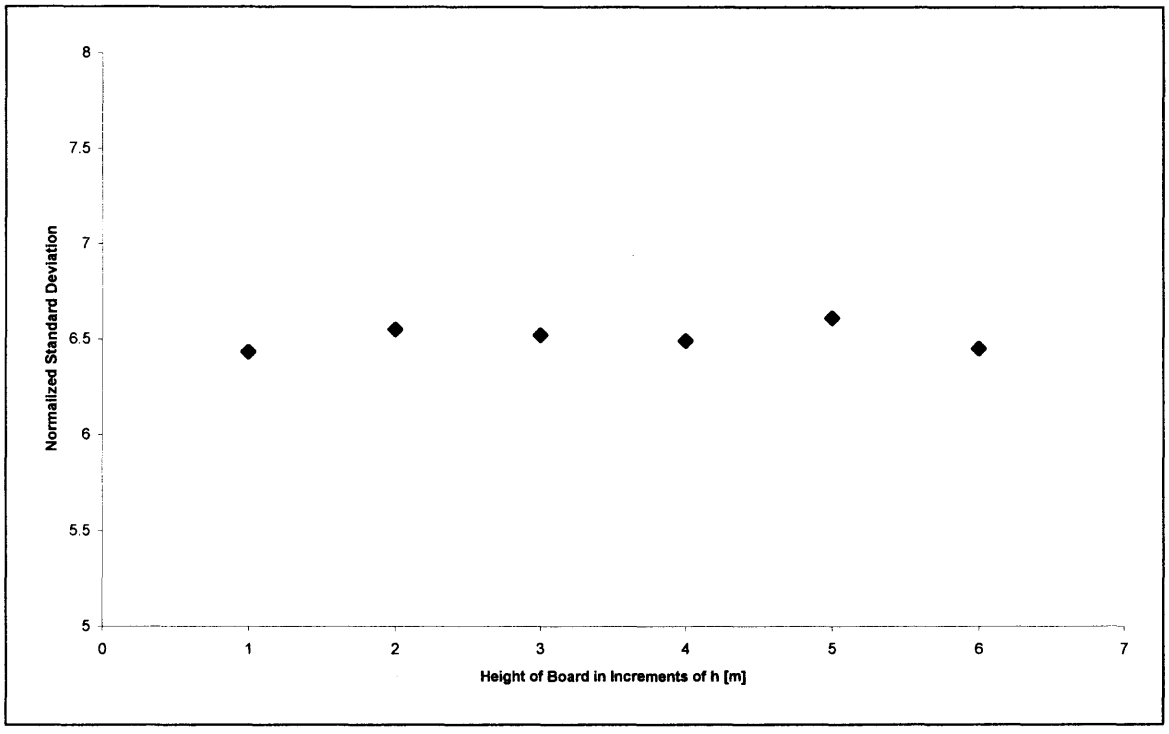

Figure 4.17 Normalized standard deviation as a function of board height (C=0.6).

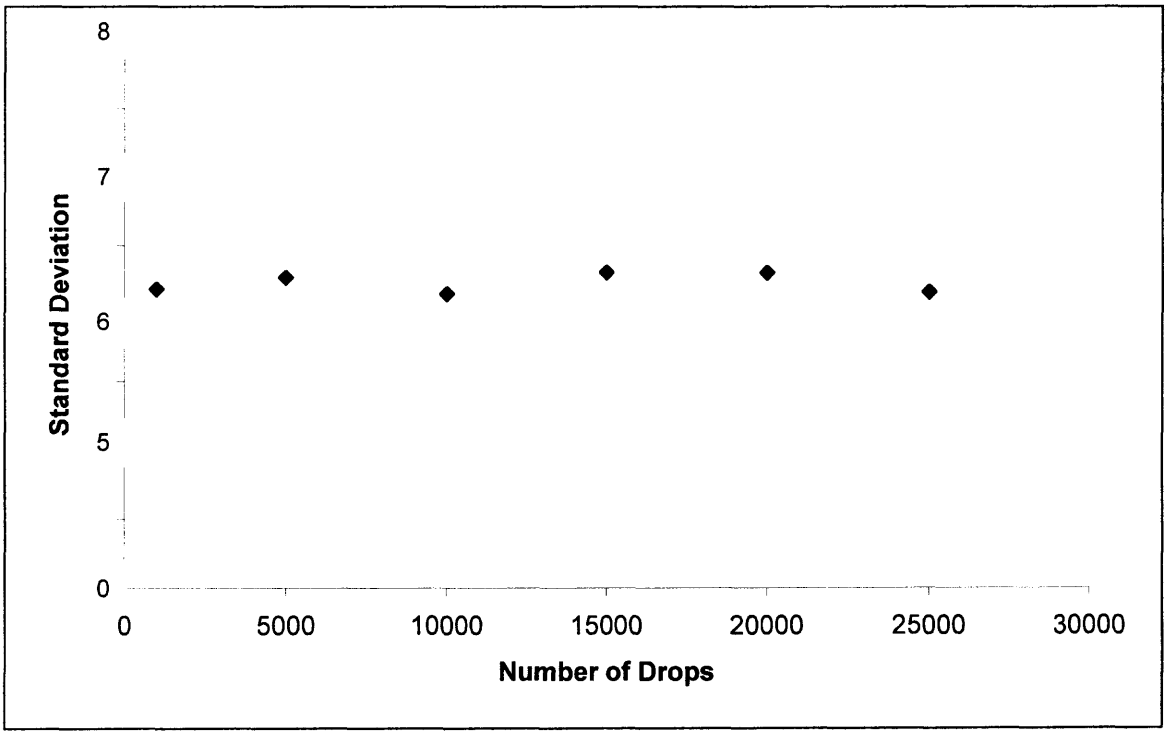

Figure 4.18 Standard deviation as a function of the number of particle drops.

Now that it has been shown that the diffusion coefficient remains constant over varying board heights an appropriate value for the constant *C* can be chosen. One method of choosing *C* is to compare the slopes of the r^2 vs. $\ln(N_0/(N_0-N))$ plot to those experimentally derived for various material types by Oshman [3]. An example of a selection of *C* is to compare Oshman's [3] slope value for stainless steel to that of the varying slopes based on *C*. For stainless steel at an angle of 70^0 Oshman [3] reports a slope of 48.61 cm^2. A best guess then at the value of *C* for steel particles and steel pins (if *C* is proportional to the restitution coefficient then *C* will be dependant upon the two materials in contact) based upon the values in Table 4.7 is approximately 0.3. It should be noted here that the distributions of particle exit positions and thus the slopes of r^2 vs. $\ln(N_0/(N_0-N))$ are somewhat skewed due to the poor modeling capabilities for collision in the neighborhood of $x = \pm a$ and $x = 0$, i.e. near the ends and center of the pins. If Oshmans [3] reported values for residence time are used in this example case (t=7.218 seconds) diffusion can be calculated using (4.3),

$$D_{xx} = \frac{Slope}{4t} = \frac{42.489cm^2}{4*7.218s} = 1.47\ cm^2/s. \tag{4.3}$$

In a similar fashion, Equation (4.2) and Figure 4.11 the MSD vs. Vertical Shift plot for C=0.3 can be used to calculate the diffusion coefficient,

$$D_{xx} = \frac{Slope}{2t} = \frac{20.0cm^2}{2*7.218s} = 1.39\,cm^2/s. \tag{4.3}$$

Table 4.9 is a summary of the results for both models with a comparison to experimental values [3].

Table 4.9 Summary of Results

Quantity	Discrete Element Model $70^0/90^0$	Discrete Nonlinear Model	Experimental 70^0
x-Velocity	-0.041 / 0.025 cm/s	N/A	-0.077 cm/s
y-Velocity	5.92 / 5.6cm/s	N/A	5.490 cm/s
Residence Time	6.7 / 7.12 seconds	N/A	7.218 seconds
Diffusion Coefficient			
MSD	2.43 / 1.48 cm^2/s	1.00 cm^2/s	N/A
VACF	2.19 / 1.30 cm^2/s	N/A	N/A
Radial Dispersion	N/A	1.39 cm^2/s	1.68 cm^2/s

As a final note for this section, an explanation for the deviation of the values in Table 4.9 is given. Looking at the two different models a vast difference in approach to calculating the trajectory of a particle traveling through a triangular lattice structure is seen, but there is a common link between the models. The common link is that of the idea of "Small Causes, Large Effects" [14]. In the discrete element model the small causes for the final position of the particle as it exited the lattice structure are due to theoretical calculations of particle collisions and the particles initial starting point. In the same manner the

small causes in the discrete nonlinear model are due to the initial starting position and the nonlinear contact model. In the physical experiments performed by Oshman [3],which were heavily referenced in this work, there could of been nearly an infinite number of small causes due to such things as vibration, material surface roughness and deviations in the nominal dimensions of the pins. The sheer fact that the calculated quantities previously presented were within the same order of magnitude, let alone such close approximation is a credit to both of the models.

4.4.2 Conclusion

The main objectives listed in Chapter 1 have been completed. The framework for future study of particles traveling through a lattice structure has been created using two independent modeling methodologies which span the spectrum of modeling capabilities from high resolution (discrete element model) to time effective (discrete nonlinear). A quantitative comparison of the results generated using both models was completed using both experimental and published works pertaining to the Galton board concluding in a relative measure of confidence for both models. Finally, the knowledge base of information pertaining to Galton's board was expanded, although by a small measure.

The coupling of the presented models allows for a two pronged approach to the study of lattice-particle systems by providing a model that can predict long term averages in a relatively short time period. One example is diffusion, which requires several thousand observations and a model capable of accurately depicting the physical phenomena taking place. The need for the two models is

highlighted in each of their individual advantages and shortcomings. The discrete element model provides a time-history of the particles dynamic nature, but its downfall is that it takes approximately 24 hours to generate a single particle trajectory. This limits its ability to generate enough results to predict quantities such as diffusion, which are dependant upon the number of observations. The discrete nonlinear model provides a means of generating several thousand particle trajectories in a relatively short period of time but requires a measure of comparison to determine the appropriate constant C. Figure 4.16 provides a hierarchy of both of the models capabilities.

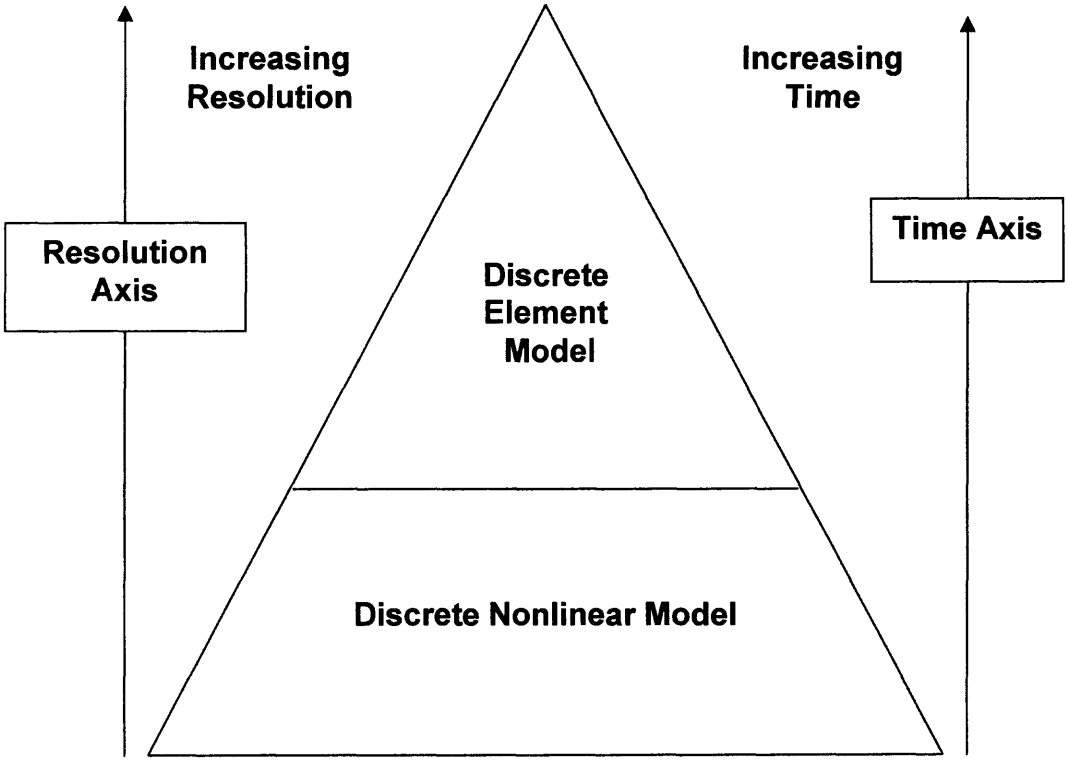

Figure 4.19 Model hierarchies.

Future work based upon this study should include several facets, enveloping almost all of the work presented. For the discrete element method most of the work to be performed should be done on the effectiveness of the computer code. For the discrete nonlinear model, most of the work should pertain to accurately depicting the contact and separation of the pin-particle collision in the regions of the pin center and ends. Further, the knowledge of the discrete nonlinear dynamic model should be expanded to understand its dependence on the lattice angle, particle size, and lattice arrangement. In all future work, the methodology undertaken should match that of the dual model method outlined here and may take the following form;

- First, a system will be defined with the geometry and material parameters clearly defined.
- Both the discrete element model and the discrete nonlinear model will be adapted to reflect the new geometry and material properties.
- Both models will be run for a small number of particle drops and the results will be compared to each other to show agreement in the model results.
- The discrete nonlinear model once shown to predict long term behavior based on the above comparison will provide any future investigation of the long term behavior in the lattice structure, while the discrete element model will provide insight into the dynamic nature.

APPENDIX A

DISCRETE NONLINEAR MODEL FOR A TRIANGULAR LATTICE

The following code is a C++ implementation of the discrete nonlinear model presented in Chapter 2 for a triangular lattice. The random number generator was adapted from [16].

```
//Discrete Non-Linear Dynamic Model

//Triangular Version

//Written by: Liam Buckley
//Granular Science Laboratory
//Department of Mechanical Engineering
//New Jersey Institute of Technology
//Last Modified 4/16/03



#include <iostream.h>
#include <math.h>
#include <sstream>
#include <fstream.h>
#include <cstdlib>

//**********ALL UNITS ARE IN
INCHES****************************************

const long double a=0.09375;                    //a=pin radius+ball radius
long double ydis=0.0;                           //initial pin y spacing
long double xdis=0.0;                           //initial pin x spacing
const long double pindis=0.3125;                //pin spacing
const long double rowdis=0.15625;               //row and column spacing
const long int b=1000000;                       //spacing variable for structures
long double xball=23.4375;                      //initial ball position in x
long double yball=15.4688;                      //initial ball position in y
long double theta;                              //theta=p from the mapping function
long double pi=3.14159;                         //pi
long double c=0.33;          //constant c from the mapping function
long double x=0.0;           //x from paper
long double absx=0.0;        //absolute value of x
const int max = 128;         //maximum file name size

//pin position file
char data1[max] = "c:/research/blackmore_triag/pinpos.csv";

 //exit position file
char data2[max] = "c:/research/blackmore_triag/e";

double s;       //Variable for passing a random number
int p;          //Integer for ball position name buffer
int seed;       //seed for random number generator
```

```
//THE FOLLOWING ARE ALL FOR THE RANDOM NUMBER
//GENERATOR TAKEN FROM NUMERICAL RECIPIES IN C
#define im1 2147483563
#define im2 2147483399
#define am (1.0/im1)
#define imm1 (im1-1)
#define ia1 40014
#define ia2 40692
#define iq1 53668
#define iq2 52774
#define ir1 12211
#define ir2 3791
#define ntab 32
#define ndiv (1+imm1/ntab)
#define eps 1.2e-7
#define rnmx (1.0-eps)

long idum;

//Pre calling the function balldataout
//so that it may be passed prior to its definition

void balldataout();

//Structure for pin positions

struct pinpos
        {
                long double x;
                long double y;
        };
pinpos pos[b];

//Structure for ball positions

struct ballpos
        {
                long double x;
                long double y;
        };
ballpos ball[b];
```

```
//Random number generator taken from "Numerical Recipies in C"

float ran2(long pass)
{
        pass=idum;
        int j;
        long k;
        static long idum2=123456789;
        static long iy=0;
        static long iv[ntab];
        float temp;

        if (idum <=0)
        {
                if (-(idum) < 1) idum=1;
                else idum = -(idum);
                idum2=(idum);
                for(j=ntab+7;j>=0;j--)
                {
                        k=(idum)/iq1;
                        idum=ia1*(idum-k*iq1)-k*ir1;
                        if (idum < 0) idum += im1;
                        if (j < ntab) iv[j] = idum;
                }

                iy=iv[0];
        }

        k=(idum)/iq1;
        idum=ia1*(idum-k*iq1)-k*ir1;
        if (idum <0) idum += im1;
        k=idum2/iq2;
        idum2=ia2*(idum2-k*iq2)-k*ir2;
        if (idum2 <0) idum2 += im2;
        j=iy/ndiv;
        iy=iv[j]-idum2;
        iv[j]=idum;
        if (iy <1) iy += imm1;
        if ((temp=am*iy) > rnmx) return rnmx;
        else return temp;
}
```

```
//Pin Position Function

void pin()
{

        int cylno=0;
        ydis=0.0;

        int m=1;
        for (int i=0;i<100;i++)             //NUMBER OF ROWS
                {
                        xdis=0.0;

                        for (int j=0;j<300;j++)  //NUMBER OF COLUMNS 2x
                                {
                                if (m==1)
                                {
                                pos[cylno].x=xdis;
                                pos[cylno].y=ydis;

                                cylno=cylno+1;
                                xdis=xdis+rowdis;
                                }
                                else
                                xdis=xdis+rowdis;
                                m=abs(m-1);
                                }
                        m=abs(m-1);
                        ydis=ydis+(rowdis);
                }

}


//Ball position function

void ballpos()
{

        ofstream out(data2);
        if (!out)
        {
                cout << "Cannot Open Position Output File.\n";
                cout << "File May Be In Use By Another Program.\n";
        }
```

```
else
{
for(p=0;p<1000;p++)        //P=NUMBER OF DROPS
{
xball=23.4375;      //ORIGINAL BALL LOCATION AT DROP X AND Y
yball=15.4688;


//ONLY CALL THE RANDOM NUMBER GENERATOR ONCE WITH THE
//SEED
if(p==0)
{
seed=-56897456235648;

}
else seed=-idum;


s=ran2(seed);
s=(s-0.5)/1000.0;
xball=xball+s;
long double z=0.0;
for(int k=0;k<100;k++)  //k=NUMBER OF ROWS...ie HEIGHT
        {


        for(int t=0;t<15000;t++)
        {
        long double test=pow(pow(pos[t].y-yball,2),0.5);
        if (test<=a)
        {
        x=xball-pos[t].x;
        absx=pow(pow((x),2),0.5);
        if (absx<a)
        {


        theta=c*sin(pi*(x/a));
        xball=xball+theta;
        }
        }
        }

        ball[k].x=xball;
        ball[k].y=yball-rowdis;
        yball=yball-(rowdis);
```

```
                z=z+1.0;
                }
        out << ball[99].x <<"\n";

        balldataout();

        }

}
}


//Pin position output function

void pindataout()
{

        ofstream out(data1);
        if (!out)
        {
                cout << "Cannot Open Position Output File.\n";
                cout << "File May Be In Use By Another Program.\n";
        }

        else
        {
                for(int d=0;d<15000;d++)   //d=TOTAL NUMBER OF PINS
                {
                out << pos[d].x <<","<<pos[d].y<<"\n";
                }
        }
}
```

```
//Ball position output function

void balldataout()
{

        char fnamebuf[max];
        sprintf(fnamebuf,"pos%02i.csv",p);
        ofstream out(fnamebuf);
        if (!out)

                {
                cout << "Cannot Open Position Output File.\n";
                cout << "File May Be In Use By Another Program.\n";
                }

        else

                {
                for(int r=0;r<100;r++)
//r=TOTAL NUMBER OF BALL POSITIONS DURING A DROP=NUMBER OF
//ROWS
                {

                out <<ball[r].x <<"," <<ball[r].y <<"\n";

                }

                }


}




//Main function
int main()
{
pin();
ballpos();
//pindataout();

return 0;
}
```

APPENDIX B

MEAN SQUARE DISPLACEMENT AND VELOCITY AUTO CORRELATION COMPUTER PROGRAMS

The following codes are C++ implementations of the mean square displacement and velocity autocorrelation algorithms presented in Chapter 3.

```
//        Diffusion Coefficient (Dxx) Calculator
//        Mean Square Displacement

// Include The Necessary Header Files
#include <math.h>
#include <iostream.h>
#include <sstream>
#include <fstream.h>
#include <cstdlib>


// Global Variables

const int b = 10000;
const int max = 128;          // Maximum Data File Name Size

// ***String For Position Data File Name
char data3[max] = "c:/research/blackmore_triag/pos1";

// ***String For Position Output Data
char data4[max] = "c:/research/blackmore_triag/posout1";

double dt = 0.001;   // ***Variable For Time Step
double dtsum;        // Variable For The Sum Of The Time Steps
int nsteps = 100;    // Variable For The Number of Time Steps In A File
double sum;
double f;
double a;
double dummy;

// Structure For Position Input File
struct position
        {
                double x;
                double y;


        };
position pos[b];
```

```
struct meansquare
	{
		double x;

	};
meansquare msd[b];



// Input Position Data File
void PosDataIn()
{
	ifstream infile(data3);
	if (!infile)
		cout << "Error: File did not open\n";
	else
	{


		for (int j=0; j < nsteps; j++)
		{


			infile >> pos[j].x;
			infile >> pos[j].y;


		}


	}
	infile.close();


}
```

```
void MSD()
{
        ofstream out(data4);
        if (!out)
        {
                cout << "Cannot Open Position Output File.\n";
                cout << "File May Be In Use By Another Program.\n";
        }

        else
        {

                for (int t=0; t< 100  ; t++)  /// DELAY TIME LOOP
                {


                        sum=0;
                        dummy=0;

                        for (int m =0;m < (nsteps-1)-t; m++)  /// TIME ORIGIN LOOP

                        {

                                dummy=dummy+1;
                                a=pow((pos[m+t].x - pos[m].x),2);

                                sum=(sum+a);


                        }


                        out << sum/(dummy);
                        out <<"\n";


                }

        }



}
```

```
// Main Function

int main()
{

        PosDataIn();
        MSD();
        return 0;
}
```

```
//      Diffusion Coefficient (Dxx) Calculator
//      VACF

// Include The Necessary Header Files


#include <math.h>
#include <iostream.h>
#include <sstream>
#include <fstream.h>
#include <cstdlib>

// Global Variables

const int b = 10000; // ***Variable For Creating An Initial Array Size. Depends On
                     //    Total Number Of Data Points In Both The
                     //     Data Files.  You Can Also Use nsteps For This Value.

const int max = 128;    // Maximum Data File Name Size

// ***String For  VELOCITY Data File Name
char data3[max] = "d:/velocity_files/Gvel01";

// ***String For VELOCITY Output Data
char data4[max] = "c:/research/VACF/vel_out1";

double dt = 0.001;      // ***Variable For Time Step
double dtsum;           // Variable For The Sum Of The Time Steps
int nsteps = 7000;      // Variable For The Number of Time Steps In A File
double sum;
double f;
double a;
double dummy;

// Structure For Velocity Input File
struct velocity
        {
                double x;
                double y;
                double z;

        };
velocity vel[b];
```

```
// Input Velocity Data File
void VelDataIn()
{
        ifstream infile(data3);
        if (!infile)
                cout << "Error: File did not open\n";
        else
        {
            for (int j=0; j < nsteps; j++)
                {


                        infile >> vel[j].x;
                        infile >> vel[j].y;
                        infile >> vel[j].z;

                }


        }
        infile.close();


}


void VACF()
{

        ofstream out(data4);
        if (!out)
        {
                cout << "Cannot Open Position Output File.\n";
                cout << "File May Be In Use By Another Program.\n";
        }

        else
        {

                for (int t=0; t< 400  ; t++)  /// DELAY TIME LOOP
                {



                        for (int m =0;m < (nsteps-1)-t; m++)  /// TIME ORIGIN LOOP
```

```
                {

                        dummy=dummy+1;
                        a=vel[m+t].x*vel[m].x;

                        sum=(sum+a);


                }


                out << sum/(dummy);
                out <<"\n";

                sum=0;
                dummy=0;
        }

    }



}


// Main Function

int main()
{

    VelDataIn();
    VACF();
    return 0;
}
```

REFERENCES

[1] Galton, F. "Typical laws of heredity", Nature. 15 (1877), 492-495, 512-514 and 532-533.

[2] Bridgwater, J., N.W. Sharpe, and D.C. Stocker. "Particle Mixing by Percolation." Trans. Institute of Chemical Engineers. 49 (1971): 163-169.

[3] Oshman, C. "Experiments with a Galton Board". Thesis. NJIT, 2002.

[4] Sergeev, S.P., T.V. Baronenkova, V.V. Dil'man, and I.Y. Kreindel. "Flow of Dispersed Medium in a Model of a Granular Bed." Theoretical Foundations of Chemical Engineering (English Translation of Teoreticheskie Osnovy Khimicheskoi Tekhnologii). 22 (1988): 134-140.

[5] Hoover, W.G., and B Moran. "Viscous Attractors for the Galton Board." Chaos. 2 (1992): 599-602.

[6] Lue, A., and H. Brenner. "Phase Flow and Statistical Structure of Galton-Board Systems." Physical Review E. 47 (1993): 3128-3144.

[7] Bruno, L., A. Calvo, I. Ippolito, S. Bourles, A. Valance, and D. Bideau. "Study on the Mixing of Discs in a Galton's Device." IUTAM Symposium: Segregation in Granular Flows, USA (2000): 73-80.

[8] Ippolito, I., Sampson, S. Bourles, and D. Bideau. "Particle Diffusion in a Porous Medium." Powders and Grains. 97 (1997): 551-554.

[9] Bridgwater, J., and N.D. Ingram. "Rate of Spontaneous Inter-Particle Percolation." Trans. Institute of Chemical Engineers. 49 (1971): 163-169.

[10] Walton, O. "Numerical Simulation of Inelastic, Frictional Particle-Particle Interactions." Particulate Two Phase Flow. Ed. M.C. Roco. Boston: Butterworth-Heinemann, 1992. 250.

[11] Yidan, L. "Particle Dynamics Modeling of Vibrating Granular Beds". Diss. NJIT, 1994.

[12] Otis R. Walton. Lawrence Livermore National Laboratory. Livermore, CA. 94550.

[13] Rosato, A.D., Blackmore, D., Buckley, L., Oshman, C., and Johnson, M. "Experimental, Simulation And Nonlinear Dynamic Analysis Of Galton's Board." Unpublished Manuscript. Dept of Mechanical Engineering, NJIT.

[14] von Mises, R. *Probability, Statistics and Truth*. Dover. New York. 1981.

[15] Haile, J.M. *Molecular Dynamic Simulations*. John Wiley and Sons. New York. 1992.

[16] Press, W.H., Vetterling, W.T., Teukolsky, S.A., and B.P. Flannery. *Numerical Recipes in C. Cambridge University Press*. New York. 1992.