ANALYSIS OF HARD- AND SOFT–BODIED GRANULAR SYSTEMS IN ONE DIMENSION

Daniel Farnoly, Benjamin Golub, Hansun Hsiung, David Kelley, Tiffany Leung,
William Nicoll, Sonya Nikolaidis, Jeremy Pfund, Adam Shpigel, Julie Wu

Advisor: Dr. Paul V. Quinn
Teaching Assistant: Justin G. Hotchkiss

ABSTRACT

We examined the phase transition that occurs in various one dimensional granular systems [1]. We observe a phase transition in which the bottom layers begin to form a liquid to solid regime once a certain critical temperature is reached [2]. We used two types of computer simulations to test our theoretical model. We determine the first order nature of the phase transition and verify the functional form of the density of the system.

INTRODUCTION

There is growing interest in the field of granular systems. Simply put, a granular system consists of a number of small particles that undergo hard sphere collisions within an enclosed space, colliding with the walls of the container as well as with each other. [3]. The hard spherical particles display interesting behavior similar to that found in ordinary solid and liquid materials. In fact, granular materials are both solid-like and fluid-like in nature. The goal of physicists studying such systems is to accurately predict and measure the movements and properties of these particles using complex thermodynamic models. Although there has been an increase in the scientific research of granular systems, many unanswered questions and incomplete models still remain.

New developments in the field are needed because there are a large number of potential applications for accurate models explaining granular behavior [4]. For instance, many industrial processes involve granular materials like those involving sand, wheat, flour, and soil. Other applications might include more efficient methods of packing, greater efficiency in the drug industry, and a more complete understanding of avalanches and earthquakes. Realistic and efficient modeling of granular systems will make these and other applications reality.

Many issues must be resolved before a truly accurate theoretical model of a granular system can be developed. Some progress has already been made. Many researchers have proposed models for granular systems that will produce results for special cases, but a unifying theory must be established in order for these models to be used generally in the wide range of scientific or industrial needs. In this project we attempt to reduce the complex nature of granular systems to a well-known system such as the molecular liquids and solids.

Our goal was to confirm a proposed theoretical model for a granular system [1,2]. This required data with which we could check the accuracy of these predictions. However, even when simplified to a small number of particles in only one dimension, granular systems remain complicated and difficult to work with. Nevertheless, by using the power of computer simulations, it is possible to measure not only the position of each particle at any time, but also the center of mass, density, critical temperature, and overall kinetic energy of the system. It is also possible to alter any number of significant conditions, such as the forces of friction and gravity, which is impossible to do in a physical system. Computer simulations use simple Newtonian mechanics to simulate individual collisions, which produces fairly accurate data by which we test our theoretical model. In essence, we used a fairly simple simulation to test our elegant and compact theoretical model [3-6].

The modeling itself was performed with two major algorithms [4,5]. The first, termed Molecular Dynamics Code (MDCode), uses a “soft-sphere” model relying on a spring-interaction concept to simulate particle collisions. The other algorithm, Event Driven Code (EDCode), uses a variable time advancement and Newtonian momentum calculations to simulate hard-sphere collisions [3,6].

The large volume of data outputted by the simulations was analyzed and producing strong support for our proposed theoretical models. Furthermore, we were able to isolate a solid-liquid phase transition in a one-dimensional case. This raises an interesting question as to whether similar effects might be observed in two and three dimensional cases. If not, future research might focus on establishing reasons for the discrepancies between the models.

THEORY

Granular systems as an ideal gas

The goal of modeling granular systems is to be able to predict important physical properties when given some initial parameters. Since the system we are studying consists of small particles, it corresponds in some ways to the well-known ideal gas. However, it deviates in other important ways for which we must compensate.

Modeling an ideal gas is an attractive starting point, because the pressure of such a system is trivial to compute. The equation for an ideal gas,

$$P = \rho k T \tag{1}$$

expresses pressure in terms of density ρ, temperature T, and the Boltzman constant k = 1.3807x10^{-23} J/K. If we use the pressure difference equation,

$$\frac{dP}{dz} = -\rho g \tag{2}$$

and some calculus, Eq.(1) can be manipulated to express density in terms of height as follows:

$$\rho = \rho_0 e^{-gz/kt} \tag{3}$$

This density derived from the ideal gas equation, is inherently problematic. Indeed, that model makes certain assumptions which cannot be justified in terms of the macroscopic situation. The two most important ones are:

(a) The volumes of the particles are negligible (the solution is extremely dilute).
(b) The particles can overlap without any interparticle interactions (in other words, they are point particles).

In reality, the particles have definite volume that is significant to the system's physical properties; Furthermore, the collisions they undergo are important in determining the behavior of the system.

It is more sensible to express the pressure of a granular system as an ideal gas with a correction C [2,7]:

$$P = \rho kt + C(\rho) \ . \tag{4}$$

C will depend on the number of dimensions. For a two-dimensional system:

$$C = \frac{\rho^3 \pi D^2 \ \chi_2(\rho)}{2kt} . \tag{5}$$

In three dimensions the pressure correction term becomes,

$$C = \frac{2\rho^2 \pi D^3 \ \chi_3(\rho)}{3kt} \tag{6}$$

χ is the two-particle correlation function which tells you about the probability of two particles colliding based on their density and their location. This function also depends on whether the system is two-dimensional or three-dimensional as shown below.

$$\chi_2(\rho) = \frac{1 - \alpha_1 D^2 \rho + \alpha_2 D^4 \rho^2}{(1 - \alpha D^2 \rho)^2} , \tag{7}$$

$$\chi_3(\rho) = \frac{1 - \frac{\pi}{12} D^3 \rho}{(1 - \frac{\pi}{6} D^3 \rho)^3} . \tag{8}$$

Thus, we have succeeded in expressing the granular pressure in a more complex but significantly more useful form which is suitable for macroscopic granular particles when volume and collision cannot be neglected.

Relating the density and height

In a container where the particles are significantly influenced by gravity, the pressure, P varies with vertical height z. The following differential equation holds true:

$$\frac{dP}{dz} = -g\rho \, . \tag{9}$$

In most cases, people are interested in the form of the density profile. To obtain this, it is desirable to express particle density as a function of height [2,7]. To make this task easier, a substitution must first be made, to the differential in Eq.(9). The dimensionless quantities Φ and ζ are defined such that

$$\Phi = \frac{4}{\pi}\rho$$

represents an undimensionalized lattice density and

$$\zeta = \frac{z}{D}$$

represents the dimensionless number of solid layers. By solving Eq.(9) with these dimensionless variables and substituting in the ideal pressure with corrections in Eq.(4), The two-dimensional and three-dimensional density profiles can be found.
The two-dimensional equation is:

$$\zeta(\Phi) = -\frac{1}{\beta}[\ln(\phi) + c_1\phi + c_2\ln(1-\alpha\phi) + \frac{c_3}{1-\alpha\phi} + \frac{c_4}{(1-\alpha\phi)^2} - f(\phi_0)] \, , \tag{10}$$

where c_1, c_2, α, c_3, c_4, and $f(\Phi_o)$, are determined constants and

$$\beta = \frac{mgD}{T} \, .$$

The three-dimensional equation is:

$$\zeta(\Phi) = -\frac{1}{\beta}[\ln(\phi) - \frac{1}{(1-\alpha\phi)^2} + \frac{2}{(1-\alpha\phi)^2} - f(\phi_o)] \, , \tag{11}$$

where α and $f(\Phi_o)$ are determined constants and

$$\beta = \frac{mgD}{T}.$$

Ideally, one would want a function of the form $\Phi(\zeta)$, but solving the differential equation analytically for this form is impossible. Instead, the inverse function $\zeta(\Phi)$ is analytically obtained.

CONDENSATION

The above equations Eq.(10) and Eq.(11) hold true as long as the temperature T of the granular system was below a predicted critical temperature T_c. Below T_c, a solid regime begins to form at the bottom of the container. [1,2,7] This occurs because a certain fraction of the particles lack the necessary kinetic energy to overcome gravity. The formation of a solid regime is called condensation. In out one-dimensional system with the "gravity-induced liquid-solid phase transition", the equation for the critical temperature is derived as a function of the control parameters of the system as shown below:

$$T_C = \frac{mgD \quad \mu}{\mu_0} \tag{12}$$

where m is particle mass, D is particle diameter, μ is the number of initial layers (or, in one dimension, the number of initial particles), and μ_0 is a constant which arises due to the packing of the solid region. The packing is determined by the close-packed density Φ_0. In our one dimensional system, the close-packed density Φ_0 can be found by analyzing the geometry of one particle layer. As the layers form, they pack together in an arrangement called square packing. Square packing is the densest packing possible in one dimension, as shown in Figure 1.

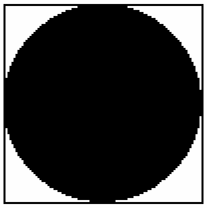

Figure 1: Square packing

The density of the particles that are square packed is π/4, or about 0.785. All solid layers will approximately have this density.

At T_c a single solid layer of particles forms on the bottom. If temperature is lowered further, more layers continue to form on top of the first. Particles in these solid-like layers are not properly included within Eq.(10) and Eq.(11). The solid layers of particles have little motion with respect to the bottom and therefore do not generate pressure against the bottom via collisions, as is assumed when using the above density functions. In essence, the density profile of the particles must be shifted up to include the layers of solid particles at the bottom. Otherwise, the integration of layers ζ with the lattice density Φ would count only the number of the moving particles, not the total number.

Phase transition during condensation

It is our hypothesis that the granular systems we are considering will undergo a first-order phase transition during condensation, similar to what is seen in water. The behavior of water and its thermodynamic properties are known and measurable using established equations. If, indeed, the granular particles in the container seemingly exhibit a similar transition, these same equations may be used to analyze what appears to be a complex and random system.

The total energy of the granular system is obtained with the equation shown below:

$$E = mg\langle z\rangle, \tag{13}$$

where $\langle z\rangle$ is the center of mass. The system is assumed to behave as if it were one giant particle, with a mass equal to the total mass of the system. Thus $\langle z\rangle$ is the height of the average center of mass. In order to analyze the phase transition, we start with the 1st Law of Thermodynamics,

$$Q = E + W,$$

the definition of entropy

$$S = \frac{Q}{T}$$

and Eq.(13) to get the following expression for entropy, S:

$$S = \frac{mg < z >}{T} + \frac{W}{T}. \tag{14}$$

We now have an expression for the entropy, S, with respect to temperature, T, and the measurable quantity $\langle z\rangle$. The order of a phase transition is determined by the derivative of the entropy. In first-order phase transitions, dS/dT is discontinuous. In other words, a first-order phase transition has an entropy function with an irremovable discontinuity (i.e. a jump, cusp, corner point), where the derivative is undefined. On the other hand, if a phase transition is second-order or higher, one will find an undefined higher-order derivative of $S(T)$. Taking the derivative of the Eq.(14) we get the following expression:

$$\frac{dS}{dT} = \frac{-mg < z >}{T^2} + \frac{mg}{T}\frac{d < z >}{dT} + \frac{d}{dT}\left[\frac{W}{T}\right]. \tag{15}$$

If $d<z>/dT$ is undefined at a certain temperature, then dS/dT will also be undefined, making the phase transition a first order transition. These two properties, temperature and average center of mass, are measurable quantities in our system of granular particles. By plotting $<z(T)>$, we might expect a discontinuity indicating a first-order phase change at temperature $T = T_c$, the critical temperature. According to ref.[1], this should appear as a kink adjoining a quadratic and linear function, are hypothesized to behave as

$$<z(T)> \propto \left(\frac{T}{T_c}\right)^2 \tag{16}$$

when $T<T_c$ and there is a status that, for a mixed liquid solid regime. For all values $T > T_c$, [1]

$$<z(T)> \propto \frac{T}{mg} \tag{17}$$

for an all liquid system. Therefore, a kink in the center of mass may appear as shown in Figure 2 below:

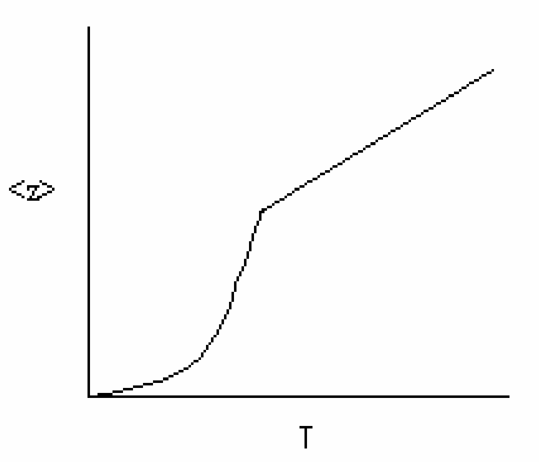

Figure 2: An illustration of <z> vs. T for a first order phase transition.

If such a kink is observed, our hypothesis stating that granular systems will undergo a first-order phase transition will be verified.

EVENT DRIVEN CODE

The event driven simulation code, otherwise known as an ED code, is designed to simulate the motion of hard-sphere particles in a closed system[3,6,7]. The ED code finds each collision, advances to the next, and uses momentum to find the velocities of each particle after the collision. Written in FORTRAN, this program was altered to emulate one-dimensional motion of particles. Contrary to the MD code, which will be discussed later, ED simulations are driven by a temperature reservoir, where each particle is excited due to heated walls.

As a result, the excitation occurs in a more classical sense, at least in terms of a standard Newtonian force being applied to each particle. These interactions take place with independent momentum calculations where each particle takes into account the vectors inside a "cell" where the probability of collision is likely. This feature is just a variable in the code that prevents unnecessary predictions from being made, thus saving computational resources. Another variable determines the magnitude of the speed given to each particle, similar to a temperature adjustment. In essence, each particle is kicked with a Gaussian velocity appropriate for a set value of the temperature reservoir being simulated. Velocities are selected randomly from a Gaussian distribution that provides likely velocities for particles in the system. Based on the movements of each particle, the code can then calculate the total kinetic energy of the system. Besides temperature, users can change variables in the simulation such as simulation time, number of particles, and sphere radius each time the program runs for a desired result. For example, one can lower the temperature variable to achieve a critical temperature.

We ran the simulation at least 10 times for different numbers of particles, $\mu = 10$, 20, 30, 40, 50 , and 60. After each run, the following sets of data were compiled: average density, particle coordinates throughout the simulation run, total kinetic energy, and center of mass. After collecting the data, the center of mass for each simulation was averaged over time with a simple program we made in C++, and a graph of $<z>$ vs. kinetic energy was tabulated. Additionally, we used the two dimensional density function to fit our existing density plots. This allows us to look for a correlation in the results of our experiments and those performed in two dimensions. A benefit of fitting the one dimensional density plots with the two dimensional density function, we can plot the center of mass $<z>$ as a function of the fit temperature $T_{2\rho}$ and find out if the phase transition is still observed. The results may be a similar graph of $<z>$ vs. kinetic energy. If this is the case, it further supports the original graphs.

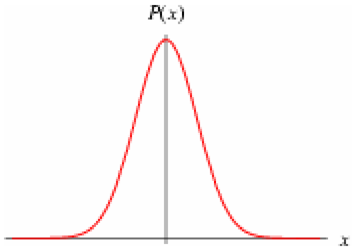

Figure 3: A Gaussian distribution function

MOLECULAR DYNAMICS CODE

In the real world, granular particles are not excited by temperature reservoirs, but rather by a vibrating plate [4,5]. A moving bottom plate is used to inject energy into the system. The vibrational strength of the plate, Γ, is controlled by the following equation

$$\Gamma = \frac{A\omega^2}{g}, \tag{18}$$

where $\omega = 2\pi f$, with f being the frequency.[1] By adjusting A, the amplitude of vibration, one can linearly change the vibrational strength. The resulting particle motion determines the value of T, average kinetic energy $(m\langle v\rangle^2) / 2$, which is used to get an effective temperature.

In the MD simulation, particle behavior is tracked through fixed time intervals Δt within the program that stay constant throughout a run. The locations of the particles, using the constant-acceleration and projectile motion equations of classical mechanics, are calculated for the next time interval and the system is advanced that far. When an overlap of particles is detected, the Leonard-Jones potential between these spheres is applied:

$$U(r) = 4\varepsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^{6}] \tag{19}$$

This makes the particles behave as if springs are quickly pushing them apart. Note that since the time interval of advance is constant, some brief particle overlap is inevitable. This overlap can be countered by decreasing Δt so that it is incredibly small, but then the run time is significantly increased.

Data for the MD program was gathered for values of particle number $\mu = 10, 15$, and 20. A single input file contains the variables A and μ, which can be altered before each run. Each run produces three data files, one containing the coordinates of the particles, another the average density over time, and the third, the center of mass, $\langle z\rangle$, over iteration number (proportional to time, it is a dimensionless quantity from the computer). Data collection is continued until the trials span from almost pure-solid to a pure liquid. Another program is then used to calculate $\langle z\rangle$, the average center of mass, for each run. Once again, the two-dimensional density profile is fit to the density data. The resultant temperatures from the fits are then used with $\langle z\rangle$ to obtain a graph of $\langle z\rangle$ v. T. This allows us to examine the vibrating wall system from a thermodynamic point of view.

---

[1] It was experimentally determined that $f = 40$ Hz works exceptionally well. This was the value used for all MDCode data runs.

VISUAL BASIC CODE

In addition to the Fortran-based simulation code, some supplementary programming was required to achieve our goals. Our team of programmers developed an application in Visual Basic which visually represents the data produced by the simulations. The data file input into the Visual Basic program consists of x and y coordinates for each particle grouped over a number of time intervals. The program also requires the number of particles as well as their radii. When it runs, the program first takes in the number of balls and their radii to properly scale the representation. Then, using the coordinates (mainly y, since x produces only a slight wobble) and a timer, the program displays each of the particles in the system and their movements as the time intervals progress. This allows the observer to more clearly visualize what is happening during the simulation, regardless of what model is being analyzed.

RESULTS

Graphs of center of mass $<z>$ versus average kinetic energy clearly showed two specific regions. One of them was parabolic, corresponding to the way particles are affected by an increase of injected energy in the solid state. The other part was linear, corresponding to the relationship between center of mass and temperature in the liquid state. The transition between these two regions of the graph was first-order in nature – similar to the melting of ice (see Theory). Furthermore, the data was well-correlated for both the linear and quadratic regions. We also observed a similar first-order transition for the MD Code data with a vibrating wall. Some of these data sets are shown below.

The simplest analyses are shown in Figures 4, 5, and 11, ($\mu = 60$, $\mu = 30$, and $\mu = 15$), where the average of the kinetic energy of the system is computed using every particle. Figures 5 and 8 are more sophisticated. Instead of computing the average kinetic energy using ½ $<mv^2>$, the temperatures are obtained by fitting density profiles according to the Enskog equation (10). Examples of fit density profiles are shown in Figures 6, 9, and 10. This is what we call the two-dimensional temperature because it is based on a two-dimensional density profile. In both cases, the discontinuity discussed above is apparent. It is interesting to note that each two-dimensional density profile has two or three parts: an optional gas phase, approximated by the Boltzman function; a liquid phase, approximated with the two-dimensional function; and a solid phase.

The particulars of the Boltzman and Enskog fits are given below.

$\mu = 60$ Boltzman: $T_1 = 2.5$ E -7; $x_0 = .138$ Enskog: $T = 2.5$ E -7; $\varphi_0 = .98$; shft = .065 (ED Code)

$\mu = 30$ Enskog: $T =$ E -7; $\varphi_0 = .95$; shft = .038 (ED Code)

$\mu = 15$ Boltzman: $T_1 = 8$ E -6; $x_0 = 3.35$ Enskog: $T = 2.1$ E -6; $\varphi_0 = .95$; shft = 2.4 (MD Code)

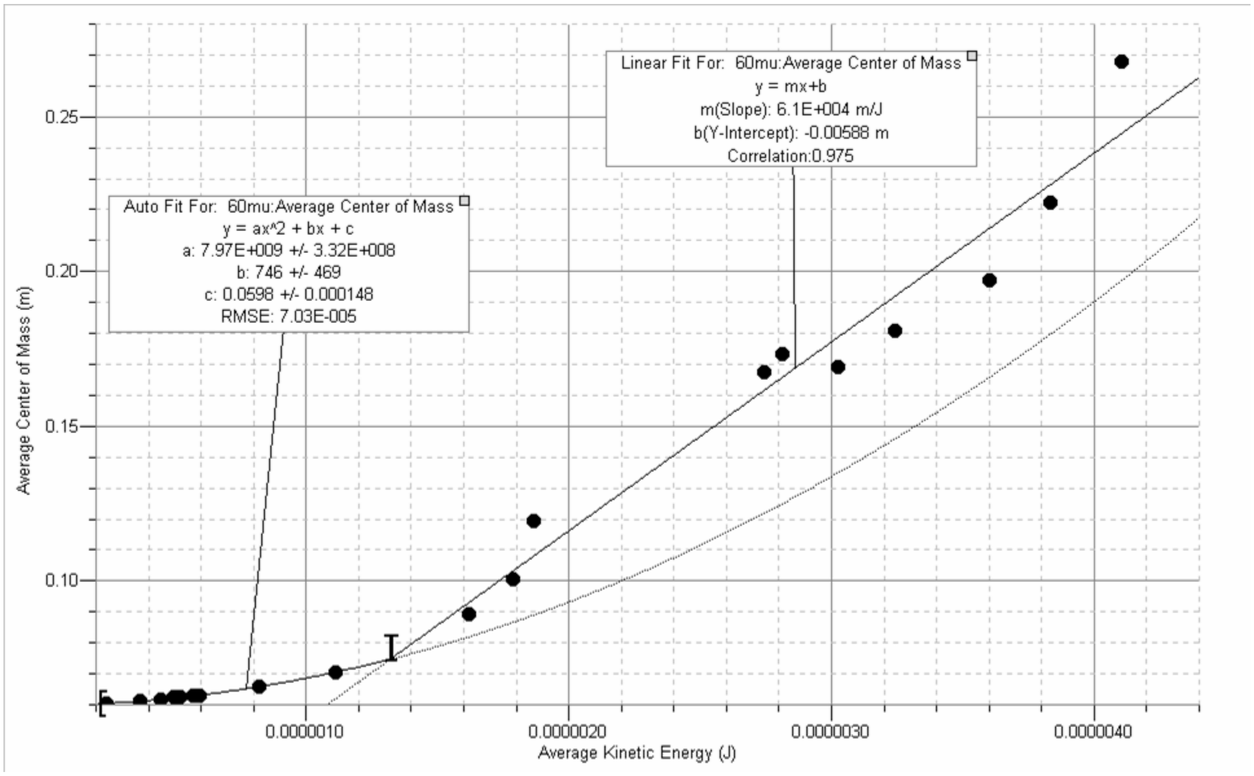

Figure 4: Average Center of Mass versus Average Kinetic Energy (EDcode) μ = 60

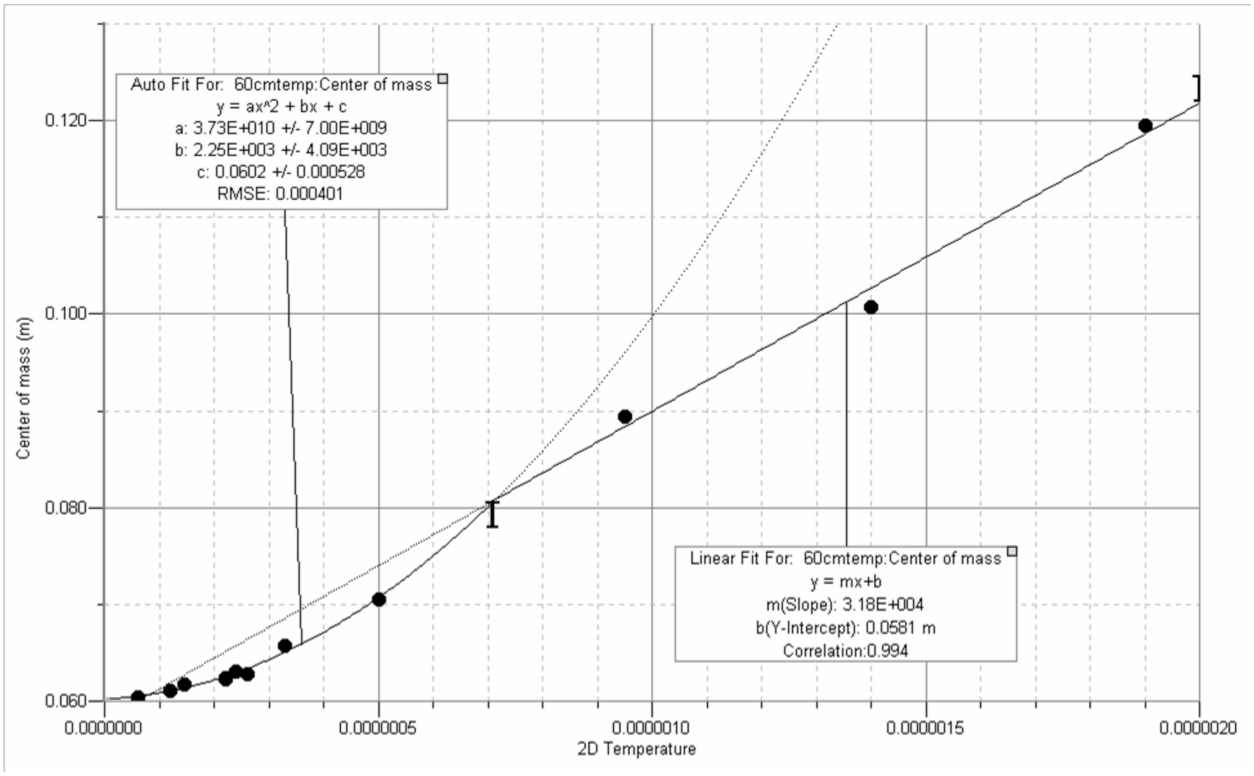

Figure 5: Average Center of Mass versus 2D Temperature (EDcode) μ = 60

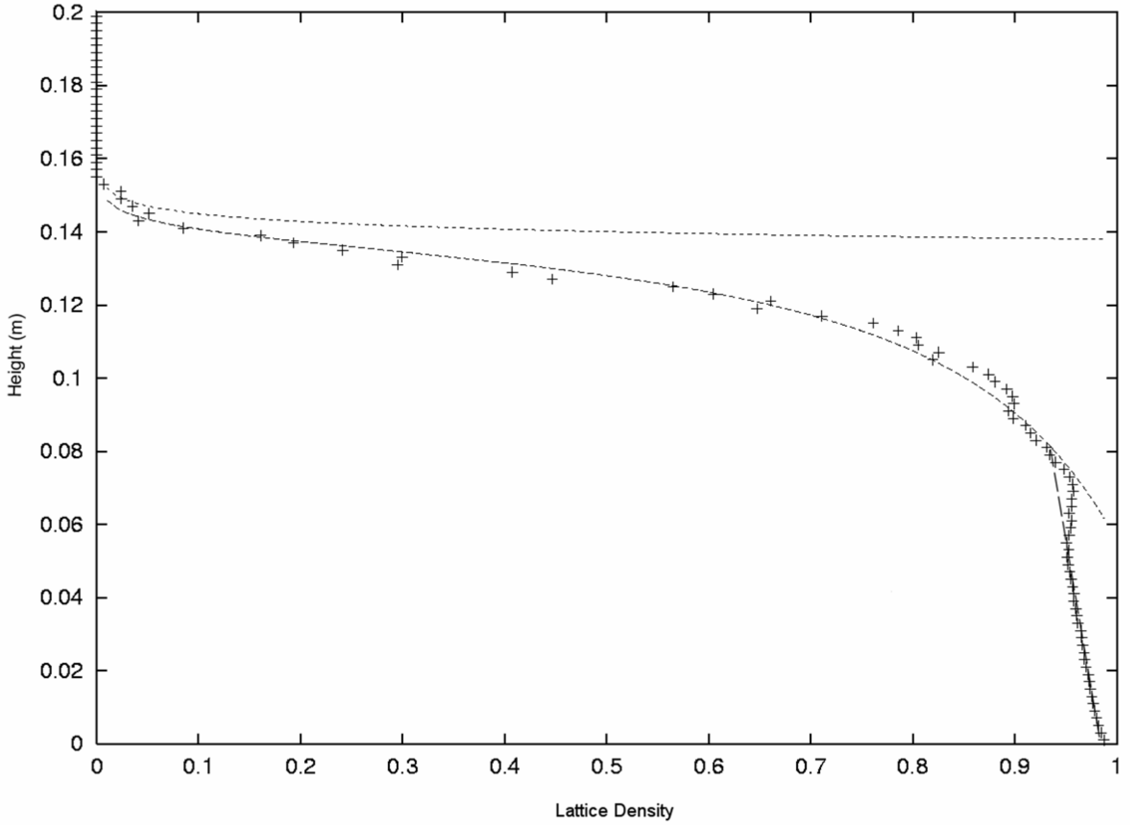

Figure 6: Example Density Profile (EDCode) *μ* = 60

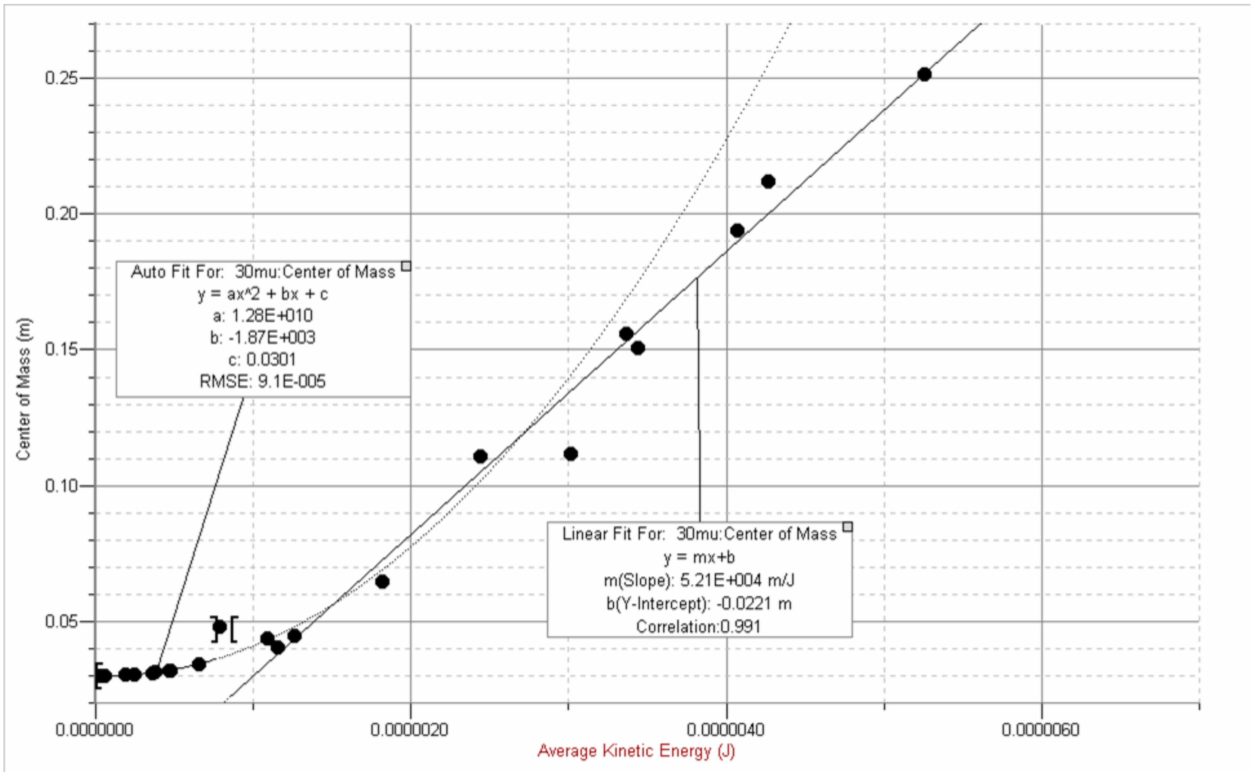

Figure 7: Average Center of Mass versus Average Kinetic Energy (EDcode) *μ* = 30

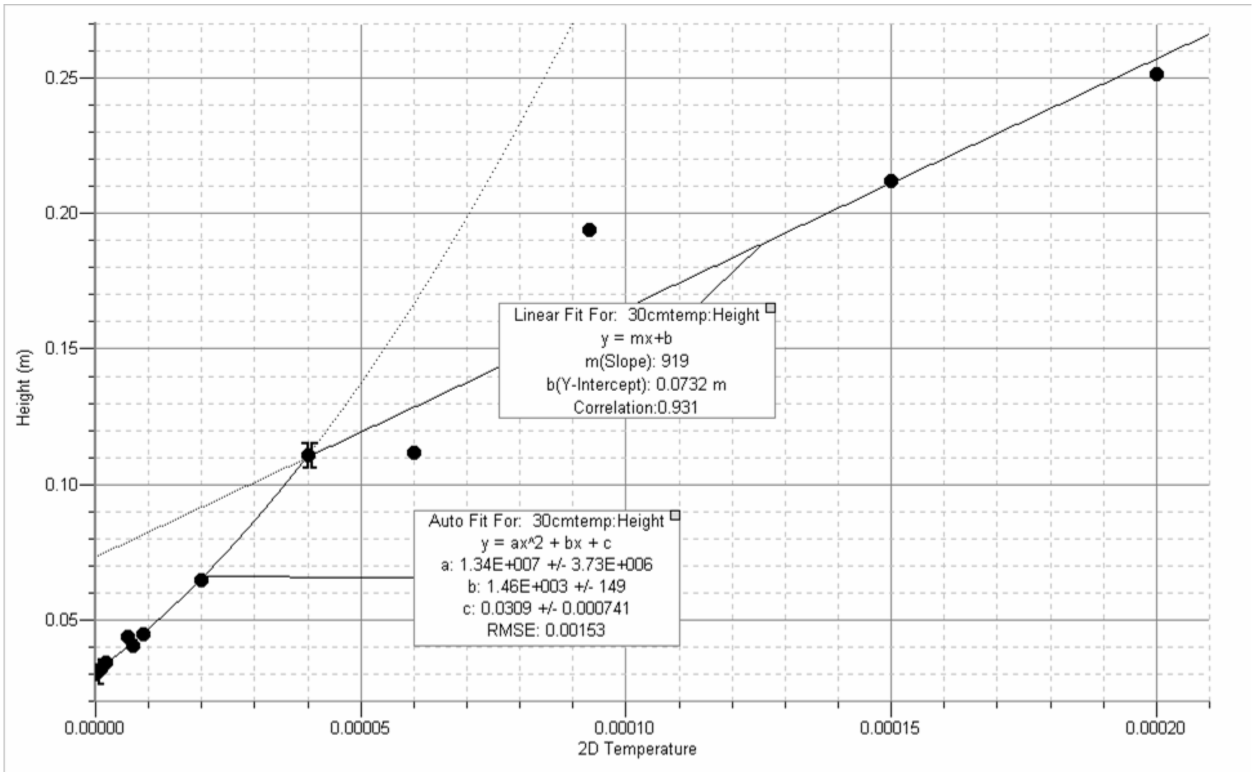

Figure 8: Average Center of Mass versus 2D Temperature (EDcode) μ = 30

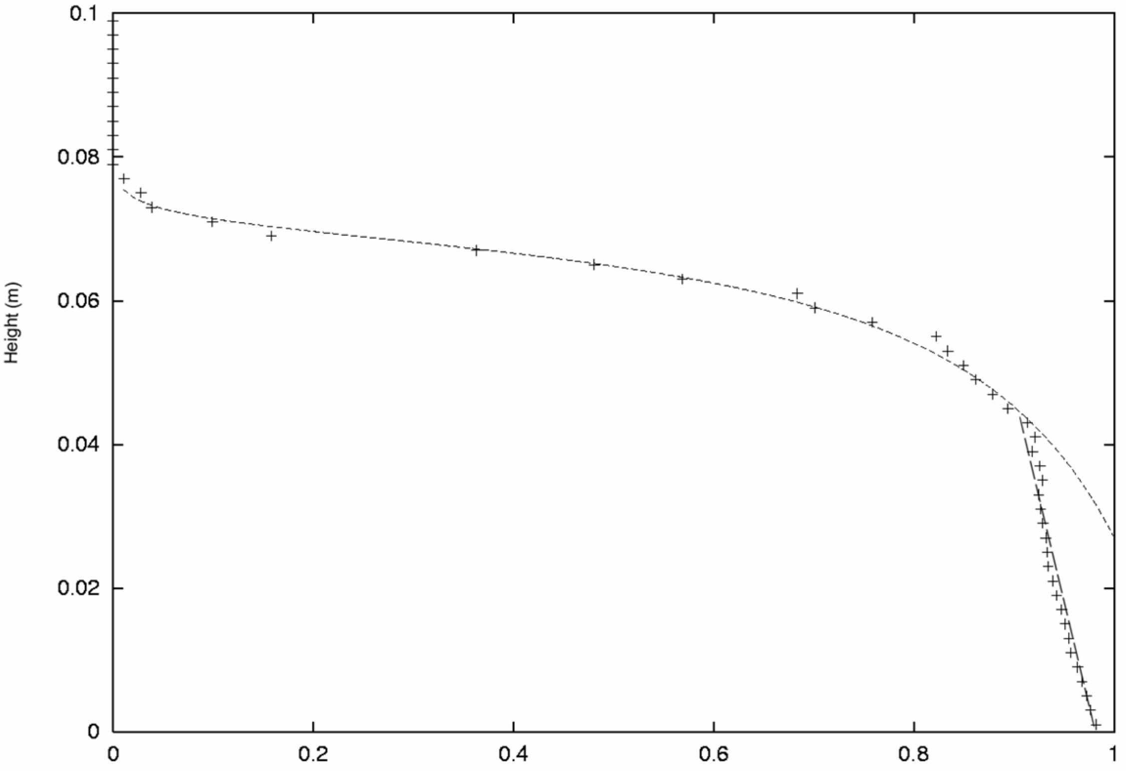

Figure 9: Example Density Profile (EDcode) μ = 30

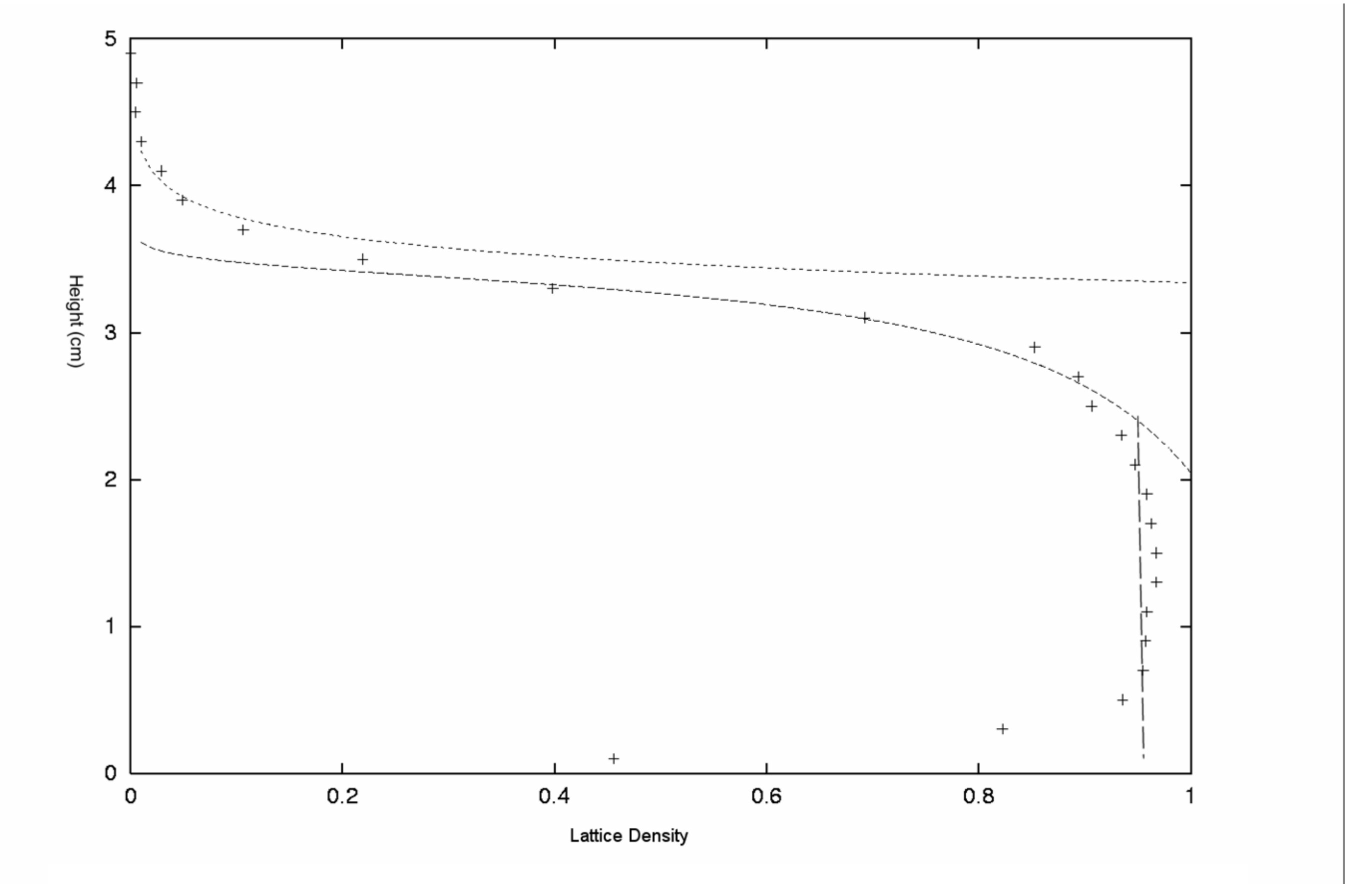

Figure 10: Example Density Profile (MDCode) *μ* = 15

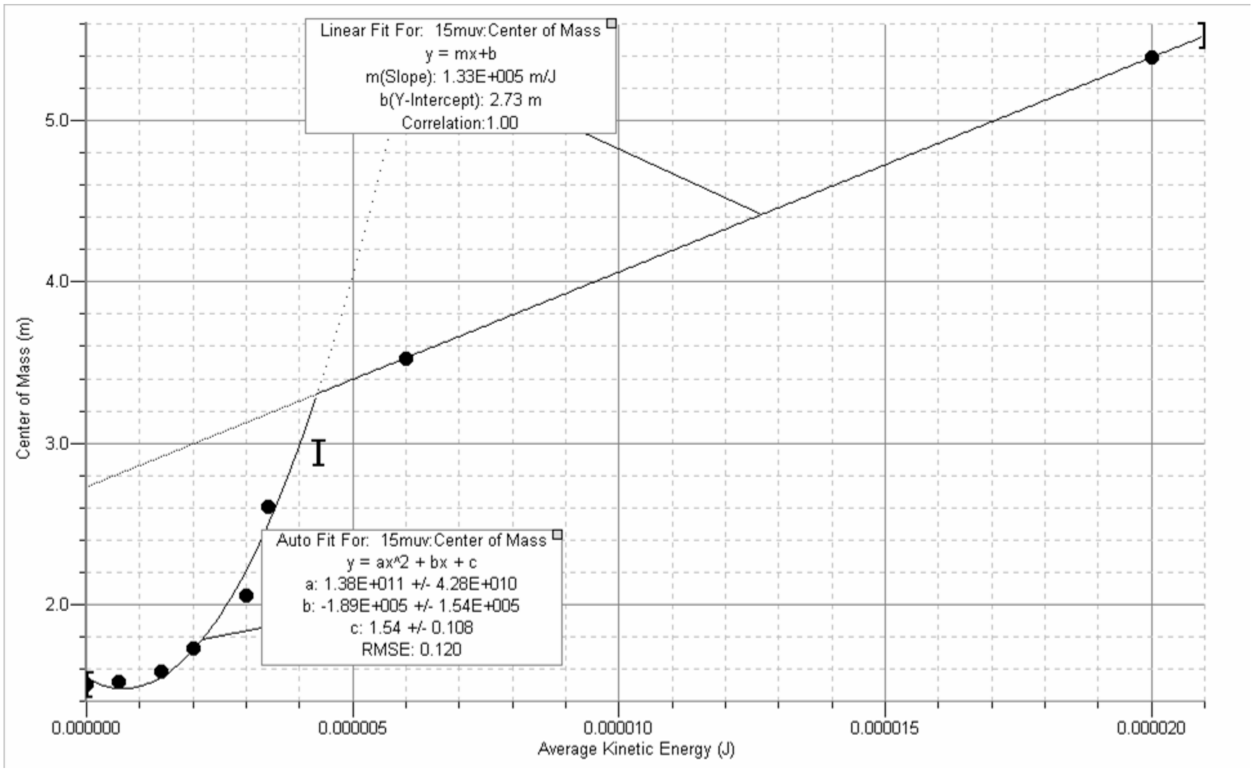

Figure 11: Average Center of Mass versus Average Kinetic Energy (MDcode) μ = 15

CONCLUSIONS

From these results, we conclude that both codes yield a first order phase transition. This is consistent with our theoretical model, which predicted a “kink”, or discontinuous differential, in the center of mass <*z*> versus temperature graphs. The correlations of actual data points with the proposed curves varied, but were generally very good. These discrepancies can be explained using a number of factors, one of which is a failure to account for the subtleties of particle interactions at high levels in the box.

Our one-dimensional density graphs are fit remarkably well with the proposed two-dimensional density profile, in all cases where the system was either extremely excited or static. We determine from this that our simulations in one dimension can be modeled as a two-dimensional case. The only significant difference is that one-dimensional models limit particle movement in a horizontal direction.

In addition, a graph plotting the critical temperature against the initial number of particles, taken from the ED Code data was linear, supporting the theoretical formula for the critical temperature. We were unable to perform a similar analysis of the MD Code data due to time constraints. However, we are fairly confident that when the full analysis is carried out with the MD Code, the results will be the same for the vibrating wall system. This further supports our idea that a one-dimensional system is comparable to a limited two-dimensional system.

Finally, our results were largely consistent with the theories presented by D. C. Hong et al[1 - 3] discussing the condensation of spheres under gravity. Overall, we showed that a granular system involving macro-particles exhibits thermodynamic properties of other, well-known systems such as the solid-to-liquid transition of water. This is a major step in the process of developing general and successful theories for the modeling of granular systems.

FUTURE WORK

Though a great deal of progress has been made in examining the dynamics of granular systems, much work remains. The next logical step is to attempt to observe the existence of a phase transition in two and three dimensions.

There are seemingly endless possibilities inherent in the study of the mechanics of granular systems. Earthquake dynamics immediately come to mind as a likely forum in which the study of granular physics could be applied. Seismic activity is simply a more complex form of the vibrational simulations we conducted. The simulated particles of the model correspond to the tiny dirt particles lying over a moving fault. Furthermore, models of granular motion could be put to use in the study of the physical mixing of substances. These mixing studies could be put to use in such fields as pharmaceuticals, where large amounts of granular substances are processed. In this way, granular systems, a field which can be classified as completely abstract, does in fact have applications in the real world. This field of study promises to remain a fruitful area of research for many years in the future. This is due to its application to practical matters as well as the insights it provides into the basic functioning of the world in which we live.

REFERENCES

[1] D.C. Hong, *Condensation of Hard Spheres Under Gravity: Exact Results in One Dimension*, http://xxx.lanl.gov, cond-mat/000749, (2000).

[2] D.C. Hong, *Physica A* **271**, 192 (1999).

[3] S. Luding, *Models and Simulations of Granular Materials*, Ph.D. thesis, Albert Ludwigs University, Germany (1994).

[4] G.A.C. Galls, H. Herrmann, and S. Sokolowski, *Physica* (Amsterdam), **189A**, 437 (1992).

[5] G.H. Ristow, G. Strassburger, I. Rehberg, *Phys. Rev. Lett.* **79**, 833 (1997).

[6] S. Luding, H.J. Herrmann, and A. Blumen, *Phys. Rev. E* **50**, 3100 (1994).

[7] P.V. Quinn and D.C. Hong, *Phys. Rev. E* **62**, 8295 (2000).

APPENDIX

```
                        MD/EDcode visual extrapolation code VERSION 5.00
Begin VB.Form frmMain
   AutoRedraw      =  -1  'True
   Caption         =   "SUPER DISPLAY V. 2.784"
   ClientHeight    =   8385
   ClientLeft      =   2715
   ClientTop       =   2520
   ClientWidth     =   5955
   ControlBox      =   0   'False
   DrawStyle       =   6  'Inside Solid
   LinkTopic       =   "Form1"
   ScaleHeight     =   8385
   ScaleWidth      =   5955
   Begin VB.CommandButton CGS
      Caption         =   "CGS(MD)"
      Height          =   3735
      Left            =   480
      TabIndex        =   2
      Top             =   4440
      Width           =   495
   End
   Begin VB.Timer timer
      Enabled         =   0   'False
      Interval        =   10
```

```
    Left            =   3360
    Top             =   840
  End
  Begin VB.CommandButton MKS
    Caption         =   "MKS(ED)"
    Height          =   3975
    Left            =   480
    TabIndex        =   1
    Top             =   360
    Width           =   495
  End
  Begin VB.PictureBox display
    Appearance      =   0  'Flat
    AutoRedraw      =   -1  'True
    BackColor       =   &H80000005&
    CausesValidation=   0   'False
    FillStyle       =   0  'Solid
    ForeColor       =   &H80000008&
    Height          =   8175
    Left            =   1800
    ScaleHeight     =   8145
    ScaleWidth      =   2505
    TabIndex        =   0
    Top             =   120
    Width           =   2535
  End
  Begin VB.Menu mnuFile
    Caption         =   "F&ile"
    Begin VB.Menu mnuFileAbout
      Caption         =   "A&bout"
    End
    Begin VB.Menu mnuFileExit
      Caption         =   "E&xit"
    End
  End
End
Attribute VB_Name = "frmMain"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
   Dim coordinates() As Double
   Dim radius As Double, numIntervals As Long, numBalls As Integer
   Dim curInterval As Long
   Dim scaleFactor As Double


Private Sub MKS_Click()
 ‘Specification of scaling for ED code
   timer.Interval = 50
   scaleFactor = 100000
 ‘Call to read in the ED code file
   readFileED
 ‘Initialization of the current time interval
   curInterval = 0
```

```
 ‘Initialization of timer, effectively triggering the drawing mechanism
   timer.Enabled = True

End Sub

Private Sub CGS_Click()
‘Specification of scaling for MD code
   timer.Interval = 1
   scaleFactor = 1000
‘Call to read in the MD code file
   readFileMD
‘Initialization of the current time interval
   curInterval = 0
‘Initialization of timer, effectively triggering the drawing mechanism
   timer.Enabled = True

End Sub

Private Sub readFileMD()
   Dim dummy As String
   Dim i As Integer, j As Integer
   Dim firstSpace As Integer

 ‘Open MD file
   Open App.Path & "\MD.d" For Input As #1

 ‘Read in the number of balls, radius
   Input #1, numBalls
   Input #1, radius

 ‘Set radius to appropriate value (utilizing actual radius and scaling factor)
   radius = radius * scaleFactor
 ‘Re-dimensionalize the coordinates
   ReDim coordinates(1, numBalls, 0)
 ‘Initialize the number of intervals
   numIntervals = 0

 ‘Main loop for data input
   Do
    ‘Read in blank line spacings
      Line Input #1, dummy

    ‘Re-dimensionalize the coordinates to include each successive interval
      ReDim Preserve coordinates(1, numBalls, numIntervals)
    ‘Sub-loop to read in ball coordinates for current time interval
      For j = 0 To numBalls - 1
       ‘Read in entire line
         Line Input #1, dummy
       ‘Locate the spacing between x & y variables
         firstSpace = InStr(4, dummy, " ")
       ‘Read in x coordinate
         coordinates(0, j, numIntervals) = CDbl(Mid(dummy, 3, firstSpace - 1)) * scaleFactor '* 5 - 5
* radius + radius
       ‘Read in y coordinate
         coordinates(1, j, numIntervals) = CDbl(Mid(dummy, firstSpace + 2)) * scaleFactor
      Next j
```

```
      ‘Increment the number of intervals
        numIntervals = numIntervals + 1

    Loop While Not EOF(1)
 ‘Close MD file
    Close #1
 ‘Set the display control width according to ballsize
    display.Width = 3 * radius + 20
End Sub


Private Sub readFileED()
    Dim dummy As String
    Dim i As Integer, j As Integer
    Dim firstSpace As Integer

‘Open ED file
    Open App.Path & "\ED.d" For Input As #1

‘Read in the number of balls, radius
    Input #1, numBalls
    Input #1, radius
‘Set radius to appropriate value (utilizing actual radius and scaling factor)
    radius = radius * scaleFactor
‘Re-dimensionalize the coordinates
    ReDim coordinates(1, numBalls, 0)
‘Initialize the number of intervals
    numIntervals = 0

‘Main loop for data input
    Do
     ‘Read in blank line spacings
        Line Input #1, dummy
     ‘Re-dimensionalize the coordinates to include each successive interval
        ReDim Preserve coordinates(1, numBalls, numIntervals)
     ‘Sub-loop to read in ball coordinates for current time interval
        For j = 0 To numBalls - 1
         ‘Read in entire line
            Line Input #1, dummy
         ‘Locate the spacing between x & y variables
            firstSpace = InStr(4, dummy, " ")
         ‘Read in x coordinate
            coordinates(0, j, numIntervals) = CDbl(Mid(dummy, 3, firstSpace - 1)) * scaleFactor * 3 - 3
* radius + radius
         ‘Read in y coordinate
            coordinates(1, j, numIntervals) = CDbl(Mid(dummy, firstSpace + 2)) * scaleFactor
        Next j
      ‘Increment the number of intervals
        numIntervals = numIntervals + 1

    Loop While Not EOF(1)

‘Close MD file
    Close #1
 ‘Set the display control width according to ballsize
    display.Width = 3 * radius + 20
```

```
End Sub


Private Sub draw()
    Dim curBall As Integer
 ‘Clear the current display screen
    display.Cls
 ‘Check for termination of data, if done, stop timer
    If curInterval = numIntervals - 1 Then timer.Enabled = False
 ‘Main loop, draws to display every ball in its new location for the current time interval
    For curBall = 0 To numBalls - 1
        display.Circle (coordinates(0, curBall, curInterval), display.Height - coordinates(1, curBall, curInterval)), radius
    Next curBall
 ‘Increment time interval
    curInterval = curInterval + 1

End Sub

Private Sub mnuFileAbout_Click()
    MsgBox "Written by GSS2002 Granular Systems Project.", vbOKOnly, "About"
End Sub

Private Sub mnuFileExit_Click()
    End
End Sub

Private Sub timer_Timer()
 ‘Call to draw function each time iteration
   draw

End Sub
```