Computer Simulation of Granular Matter: A Study of An Industrial Grinding Mill

(Spine title: Computer Simulation of Granular Matter)
(Thesis format: Traditional)

by

John J. Drozd

Graduate Program in Department of Applied Mathematics

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario, Canada

© John J. Drozd 2004

THE UNIVERSITY OF WESTERN ONTARIO
FACULTY OF GRADUATE STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
______________________	______________________
Co-Supervisor	
______________________	______________________
Supervisory Committee	
______________________	______________________

The thesis by

John J. Drozd

entitled

Computer Simulation of Granular Matter: A Study of An Industrial Grinding Mill

is accepted in partial fulfillment of the
requirements for the degree of
Master of Science

Date ______________________ ______________________

Chair of the Thesis Examination Board

ABSTRACT

This thesis is a study of an industrial crushing process. Long steel rods are placed horizontally in a horizontal cylindrical drum supported on springs. A motor oscillates the springs and the drum is shaken vertically for the purpose of crushing a particular medium between the rods. The goal of this work is to ascertain the optimum oscillation that results in a force between the rods which achieves the ultimate stress of a particular medium that is to be crushed between the rods. By achieving "optimum oscillation" we mean to minimize the total energy required to grind the medium. This will depend on the fraction of collisions that produce a greater-than-threshold force, for a given power input. The smaller this fraction, the longer the machine will have to run to ensure that all the medium gets crushed. Mixing is also important. Modes that produce a lot of large forces may not mix very well, resulting in some part of the medium being well crushed, while another part of the medium is not. A computer simulation is used to answer these questions. The rods are simulated as two dimensional disks in a vertically, sinusoidally oscillating, circular container. Two dimensions are sufficient to obtain a first order approximation and understanding of the problem, assuming that the effects of the rods crossing over each other or hitting the end of the cylindrical drum are neglected. The motion of the disks is modelled using the equations of motion of particle dynamics under uniform gravitational acceleration, with inelastic collisions between the disks and inelastic collisions between the disks and the circular container. Collision energy and momentum transfer before and after a disk-disk or disk-wall collision are calculated to estimate the crushing force between two disks or between a disk and the circular wall. A distribution of these forces for various modes of oscillation is plotted to generate a map of the location in the container where the forces are large, to show where most of the grinding will occur. The flow of these disks is also studied and granular phenomena such as the onset of collective motion (convection), "force chains" and avalanche events are observed. A particularly efficient serial event-driven algorithm for calculating collisions is used.

ACKNOWLEDGEMENTS

I welcome this opportunity to thank all those who helped me in pursuing my thesis project with their comments and ideas. First and foremost, I would like to thank my supervisor Dr. Peter Poole and my co-supervisor Dr.-Ing. Robert Martinuzzi for their patience, helpful guidance and their careful, critical proofreading of my thesis. Both my supervisors provided me with a unique and valuable insight. I would like to thank Dr. Peter Poole for introducing me to this fascinating and active field of research. Also, I would like to thank both Dr. Peter Poole and Dr.-Ing. Robert Martinuzzi for their most valuable and critical comments during our long discussions. As well, I must thank Dr. Oleh Baran who was a postdoc with Drs. Poole and Martinuzzi. Dr. Baran gave me helpful suggestions in the programming aspects of my thesis. Furthermore, I would like to acknowledge Dr. Ivan Saika-Voivod who helped me with some of the theoretical derivations and to shellscript my code to get my C++ code running on the *SHARCNET* cluster of processors. I would also like to acknowledge two undergraduate students, Kirk Bevan and Brad Smith, who began the coding with me three years ago. Brad Smith is now a police officer and I promised to reward him with his favorite donut when he gives me a ride in his police cruiser. I would like to thank Dr. Sreeram Valluri for critically reading parts of my thesis and helping me with some of the theoretical derivations which involved some of the physics of the simulation. Most importantly, I would like to thank my parents and sisters for the patience, understanding and support they gave me throughout my entire life and for their continuing belief in me. Finally, I would like to acknowledge NSERC for their award of a Postgraduate Scholarship that has supported this study.

TABLE OF CONTENTS

CERTIFICATE OF EXAMINATION **ii**

ABSTRACT **iii**

ACKNOWLEDGEMENTS **iv**

TABLE OF CONTENTS **v**

LIST OF FIGURES **viii**

LIST OF SYMBOLS **xiv**

Chapter 1 INTRODUCTION **1**

Chapter 2 ALGORITHM AND FORMULAE **11**

2.1 Introduction 11
2.2 Event-Driven Lubachevsky Algorithm 12
2.3 Sectoring 14
2.4 Heap Tree Data Structure 15
2.5 Overlaps 17
2.6 Rod-Rod Collision Dynamics 20
2.7 Container Oscillations 24
2.8 Rod-Container Collision Dynamics 24
2.9 Circulation 26
2.10 Object-Oriented Structure of Program 36

Chapter 3 FORCES **40**

3.1 Introduction 40

3.2 Measuring Forces As Collision Energies 41

Chapter 4 RESULTS AND ANALYSIS **43**

4.1 Introduction 43

4.2 Verification of Results 43

4.3 Parameters for Simulation 48

4.4 Steady State 49

4.5 Mixing 54

4.6 Phase Diagrams 57

4.7 Circulation 60

4.8 Forces Analysis 75

4.9 Force Chains 85

4.10 Avalanche Movement and the Angle of Repose 86

Chapter 5 CONCLUSIONS AND RECOMMENDATIONS **87**

Appendix A Numerical Methods Used to Study Granular Materials **89**

A.1 Monte Carlo Method 89

A.2 Diffusing-Void Model 89

A.3 Method of Steepest Descent 90

A.4 Cellular Automata 90

A.5 Event-Driven Simulations 91

A.6 Time-Driven Simulations (Molecular Dynamics) 92

Appendix B Sample and Hold **94**

Appendix C Rotations **96**

Appendix D Direct Measures of Forces Based on Elasticity **101**

Appendix E Shepard Interpolation **109**

Appendix F Compact Disc **111**

Appendix G Permission to Print Figures **112**
G.1 Letter from American Society of Mechanical Engineers 113
G.2 Letter from Elsevier Publishers 114
G.3 Letter from American Physical Society 115
G.4 Letter from Dr. Harry L. Swinney 116

REFERENCES **117**

VITA **120**

LIST OF FIGURES

1.1 Small Amplitude Surface Waves: $\Gamma_a = 3.3$, $f = 20$ Hz, $d = 1.28$ mm, and $h_0/d = 17.5$ [2] (reprinted with written permission from publisher) 3

1.2 3-node Arching: $\Gamma_a = 8.5$, $f = 30$ Hz, $d = 1.28$ mm, and $h_0/d = 17.5$ [2] (reprinted with written permission from publisher) 4

1.3 Large Amplitude Surface Waves: The top photo shows waves superimposed on 0-node arching and the bottom photo shows waves superimposed on 1-node arching for $\Gamma_a = 6.2$, $f = 20$ Hz, $d = 1.28$ mm, and $h_0/d = 17.5$ [2] (reprinted with written permission from publisher) . . 5

1.4 Comparison of Experiment and Simulation: Numerical simulations and views from above of laboratory experiments conducted for vertically oscillated square pans filled with thin layers of lead spheres sieved between 0.5 and 0.6 mm. The simulations and experiments are a function of the frequency f, amplitude A, and depth h_o. In the figures, $\Gamma_a = 4\pi^2 f^2 A/g$ and $f^* = f\sqrt{h_o/g}$ where g is the acceleration due to gravity. The pictures show a top view of the standing wave patterns (squares, stripes, alternating phases of hexagons and flat layers) in the pans. The brightness indicates the height of the layer. [4] (reprinted with written permission from publisher and author) 7

1.5 Schematic Diagram of Typical Experimental Drum Filled with Rods . 9

1.6 Typical Experimental Drum [5] The balls inside the drum are replaced by rods. (permission has been requested from publisher to reproduce this figure) 10

2.1 Typical Event-Driven Simulation Without Gravity. This simulation could be realized by having billiard balls colliding on a circular bordered pool table. 12
2.2 Typical Event-Driven Simulation With Gravity 13
2.3 Heap Tree Data Structure [8] 17
2.4 Preempted Collisions. Figure(a) result of cycles 1 to 4: current time = 0; balls 1 and 4 have scheduled a collision for time 25; balls 2 and 3 have scheduled a collision for time 388. (b) Result of cycle 5: current time = 25; ball 1 has processed its collision with ball 4 for time 25; balls 1 and 2 have scheduled a collision for time 226; ball 2 cancelled an earlier collision with ball 3 for later time 388 and its collision is turned into an advancement for ball 3. (c) Result of cycle 6: current time = 25; ball 4 has processed its collision with ball 1 at time 25; balls 2 and 4 have scheduled a collision for time 87; ball 2 has cancelled an earlier scheduled collision with ball 1 for later time 226 and this collision is turned into an advancement for ball 1. [8] (reprinted with written permission from publisher) 19
2.5 A smooth disk collision [15] 21
2.6 Calculation of reflected velocity $\phi_r = \pi + \alpha - (\phi_i - \alpha) = \pi + 2\alpha - \phi_i$[3] 26
2.7 Typical Simulation Snapshot After 100,000 Collisions: frequency of oscillation $\omega_y = 126$ rad/s, amplitude of oscillation $A_y = 1.5$ cm, disk-disk coefficient of restitution $e_0 = 0.4$, disk-container coefficient of resitution $e_W = 1.0$. The container is the large white circle. The disks are the smaller blue circles except for the latest colliding disk which is the circle coloured in pink. 27
2.8 Typical Time Averaged Velocity Field After 100,000 Collisions. The large almost straight trajectories at the top are due to freely flowing particles near the beginning of the simulation. 28
2.9 Circulation Diagram [3] 30

2.10 Schematic Representation of the Formation of Bénard Cells [21] . . . 32
2.11 Typical Velocity Field for Symmetric Simulation 33
2.12 Net Circulation vs Time for Typical Asymmetric Simulation. The spikes in this plot are due to accelerated disks freely flowing at the top of the cluster. These freely flowing disks are mobilized by the transfer of energy from the container periodically hitting the bottom of the cluster. 34
2.13 Net Circulation vs Time for Typical Symmetric Simulation 35
2.14 Object Diagram 39

3.1 Modulus of Toughness [24] 42

4.1 FORTRAN Implementation: showing lifting of the granular bed . . . 45
4.2 C++ Implementation: showing lifting of the granular bed 45
4.3 FORTRAN Implementation: showing a solitary travelling surface wave and two-node arching 46
4.4 C++ Implementation: showing a solitary travelling surface wave and single-node arching 46
4.5 FORTRAN Implementation: showing evolution of a solitary travelling surface wave and two-node arching 47
4.6 C++ Implementation: showing evolution of a solitary travelling surface wave and single-node arching 47
4.7 Velocity-Dependent Restitution Coefficient [13] 49
4.8 Normalized Total Kinetic Energy (KE/M) versus time (t) for 100,000 collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 50
4.9 Normalized Total Kinetic Energy (KE/M) versus time (t) for 1 million collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 51
4.10 Power Spectrum of Normalized Total Kinetic Energy ($\mathcal{P}(\omega)$) for 1 million collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ . . . 53

4.11 Mean Square Displacement ($< r^2 >$) versus time (t) for 100,000 collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 55
4.12 Mean Square Displacement ($< r^2 >$) versus time (t) for 1 million collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 56
4.13 Phase Diagram: Amplitude (A) vs Frequency (ω_y): $e_0 = 0.4$, $e_W = 1.0$ 58
4.14 Time Averaged Net Circulation (Γ) vs Frequency of Container Oscillation (ω_y): $e_0 = 0.4$, $e_W = 1.0$ 61
4.15 Non-dimensional Parameter ($\Gamma/(A_y^2\omega_y)$) vs Frequency of Container Oscillation (ω_y): $e_0 = 0.4$, $e_W = 1.0$ 62
4.16 Time Averaged Velocity Field for Amplitude $A_y = 1.0$ cm and Frequency $\omega_y = 200$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 63
4.17 Time Averaged Velocity Field for Amplitude $A_y = 1.0$ cm and Frequency $\omega_y = 400$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 64
4.18 Time Averaged Velocity Field for Amplitude $A_y = 1.0$ cm and Frequency $\omega_y = 600$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 65
4.19 Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 126$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 66
4.20 Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 200$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 67
4.21 Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 400$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 68
4.22 Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 600$ rad/s: $e_0 = 0.4$, $e_W = 1.0$ 69
4.23 Snapshot Frame 1 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disk is coloured in pink. 70
4.24 Snapshot Frame 2 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink. 71

4.25 Snapshot Frame 3 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink. 72
4.26 Snapshot Frame 4 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink. 73
4.27 Net Circulation (Γ) vs Time (t): $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 74
4.28 Contour Plot of Time Averaged Disk Disk Collision Energies (CE): $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The contour levels are in units of 10^{-4} J/kg. 76
4.29 Unit Circle Orientation of θ Along Wall of Container 76
4.30 Bar Chart of Normalized Disk Container Collision Energies (CE/M) vs θ: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 77
4.31 Bar Chart of Number of Hits of Disks Against Container (n) vs θ: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 78
4.32 Bar Chart of Average Normalized Disk Container Collision Energies (<CE/M>) vs θ: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ . . 79
4.33 Velocity Field Snapshot for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 80
4.34 Number of Different Disks That Experience Mean Collision Energy (n_i) vs Time (t) for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. There are 150 disks in total. 81
4.35 Normalized Dissipated Energy (E/M) for Rod-Container Collisions vs Time (t) for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. 83
4.36 Power Spectrum of Normalized Dissipated Energy ($\mathcal{P}(\omega)$) for Rod-Container Collisions for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$ 84

4.37 Snapshot of Animation Showing Force Chain: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink. . 85
4.38 Snapshot of Animation Showing Avalanche Movement and Angle of Repose: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink. 86

B.1 (a) ideal case and (b) sample and hold case [31] 94

C.1 A rough disk collision [15] 97

D.1 Sketch of normal $\hat{n}$ and tangential $\hat{s}$ directions during disk-disk contacts [1] 102
D.2 Disk Overlap During Collision 102
D.3 Relating Spring Constant k to normal stress σ_{22} 105
D.4 Disk Compression Geometries 106
D.5 Figure for Calculating Displacement Field 107

LIST OF SYMBOLS

ω, ω_y	angular velocity, frequency of container oscillation
f	normal frequency of vibration
d	particle diameter
h_o	initial particle bed depth
h_o/d	dimensionless bed depth
a	acceleration of particle
g	acceleration due to gravity
Γ_a, Γ	dimensionless acceleration amplitude, net circulation
A, A_y	amplitude of vibration, amplitude of container oscillation
N	number of rods or disks
σ	diameter of disk
$\mathbf{r_i}$, $\mathbf{r_j}$	positions of disks i and j
$\mathbf{r_{ij}}$	relative distance between positions of disks i and j
$\mathbf{v_i}$, $\mathbf{v_j}$	velocities of disks i and j
$\mathbf{v_{ij}}$	difference between velocities of disks i and j
$\mathbf{v_i'}$, $\mathbf{v_j'}$	velocities of disks i and j after a collision
b_{ij}	dot product of difference in positions and difference in velocities
t_{ij}	collision time between disks i and j
t	current time
$\delta\mathbf{v_i}$	change in velocity due to a collision
$\mathbf{p_i}$, $\mathbf{p_j}$	momenta of disks i and j before a collision
$\mathbf{p_i'}$, $\mathbf{p_j'}$	momenta of disks i and j after a collision
m, $m_{\mathbf{disk}}$, m_a, m_b	mass of each disk
$m_{\mathbf{eff}}$	effective mass
$v_{\mathbf{n}}$	component of relative normal velocity

$e(v_n)$	velocity-dependent restitution coefficient
β, B	parameters used in calculating velocity-dependent restitution coefficient
ε, e_0, e	tunable parameter between 0 and 1 for restitution coefficient formula
$\bar{y}$	vertical displacement of the centre of the container
x, y	specify the location of the centre of a disk
v_x or u, v_y or v	disk velocity components
R	radius of the container
r	radius of a disk
x_c, y_c	location of the centre of the container
$\frac{d\bar{y}}{dt}$	velocity of the container
v_x'', v_y''	the velocity of a disk after colliding with the container
ϕ_i, ϕ_r	incident and reflected angles at disk-container collision
e_W	coefficient of restitution for a disk-container wall collision
α	angular position of disk at disk-container collision
Ω	vorticity of circulating disks
ϕ	scalar velocity potential, container diameter
V	velocity
∇	gradient operator
r_{max}	distance from farthest disk to centre of mass of the circulating system
r_o, h_o	specify centre of mass of the circulating system
$\mathbf{a_i}$	vector used to calculate angular velocity of a system of circulating disks
$\bar{\mathbf{v}}$	average velocity of all the disks
σ_x, σ_{22}	normal stress
ε_x	strain
ϵ_R	strain at rupture
U	modulus of toughness, strain energy density taken at strain at rupture
KE/M	normalized total kinetic energy
FFT, fft	Fast Fourier Transform
$\mathcal{P}(\omega)$	Power Spectrum
$< r^2 >$	mean square displacement
N_t	number of time steps

D	diffusion coefficient or diffusion constant
θ	angle along the circular container wall
n_i	number of disks that experience the mean collision energy i times
$\mathbf{L}$	angular momentum
$\mathbf{S}$	spin angular momentum
$\mathbf{J}$	total angular momentum
I	moment of inertia
ω_i, ω_j	rotational angular velocities of disks i and j
$\mathbf{v}_{\mathbf{ij}}^{\mathbf{imp}}$	relative velocity of rotating disks
$\delta\mathbf{v}_{\mathbf{ij}}^{\mathbf{imp}}$	change in relative velocity of rotating disks
$\delta\mathbf{p}_i$	impulse
κ	parameter including moment of inertia
$\hat{\mathbf{r}}_{ab}$	unit vector from disk a to disk b
$\mathbf{v}_{ab}$	relative velocity
$\mathbf{v}_n$	relative velocity in normal direction
$\mathbf{v}_t$	relative velocity in tangential direction
$\mathbf{v}_s$	relative surface velocity (tangential direction)
$\Delta\mathbf{v}_{na}$, $\mathbf{v}_{nb}$	changes in normal direction velocities
$\Delta\mathbf{v}_{ta}$, $\mathbf{v}_{tb}$	changes in tangential direction translational velocities
$\Delta\vec{\omega}_a$, $\Delta\vec{\omega}_b$	changes in rotational velocities
β^*, β_0	parameters determining rolling or sliding friction
$F_{\mathbf{el}}$	elastic restoration force
$F_{\mathbf{fric}}$	viscous friction force
γ	frictional damping coefficient
t_c	time it takes for a disk to deform
$r_{\mathbf{eff}}$	effective radius of two disks
k, k_1, k_2	spring constants
λ, μ	Lame constants
E	Young's modulus of elasticity
G	Shear modulus of rigidity
ϵ_{11}, ϵ_{22}	strains in x_1 and x_2 directions

u_1^i, u_2^i	displacements in x_1 and x_2 directions
$\frac{\partial u_1}{\partial x_1}$, $\frac{\partial u_2}{\partial x_2}$	displacement gradients

Chapter 1

INTRODUCTION

Jede Naturwissenschaft wäre wertlos, deren Behauptungen nicht in der Natur beobachtend nachegeprüft werden könnten; jede Kunst wäre wertlos, die die Menschen nicht mehr zu bewegen, ihnen den Sinn des Daseins nicht mehr zu erhellen vermöchte.[1]

Werner Heisenberg

Granular materials or *granular matter* is a general term describing a material that is made up of small particles, which might be grains, rocks, sand or pills. Consequently, granular matter is a system of interest to many disciplines such as biology, engineering, geology, material science and physics. For the purpose of this work, we are considering materials made up of discrete particles. Since researchers have just begun to theoretically describe granular materials, this field has attracted mathematicians and computer scientists, leaving many possibilities for the development of new theories and large-scale computer simulations to model granular phenomena and to compare these with the results of physical experiments [1].

In the study of granular flows, continuous energy input is needed in order to mobilize and maintain the particles in motion, due to the highly dissipative nature of particle collisions. Such matter in motion in many ways is fluid like but the flow

---

[1] Any natural science whose assumptions cannot be verified through observations in nature would be worthless; any art that isn't capable of moving people or of enlightening them as to the meaning of their existence would also be worthless. [1]

of such cannot be represented by a continuum. The matter is considered as discrete classical particles in which contact forces and dissipation are key ingredients. This combination of dissipation and excitation results in a variety of granular motion such as convection rolls and standing waves involving a homogeneous group of same-sized particles, or stratification of particles and segregation with a mixture of different-sized particles [1].

Literature on granular materials often describes granular material phenomena by experiments. Materials that have been used for experiments include pharmaceutical pills, sugars, sands, seeds, and beads. A typical example is a granular experiment that fills a two-dimensional box with 3 mm diameter glass spheres and then vertically vibrates the bottom of the box. This experiment is described in the paper *Vertical Vibration of a Deep Bed of Granular Material in a Container* [2]. Photographs that capture the granular flow features characteristic of this experiment include *surface waves* and *arching* as shown in Figures 1.1, 1.2 and 1.3. In the figures, the parameters that influence the state of the particle bed include the amplitude, a, and the radian frequency of vibration, $\omega = 2\pi f$ where f is the normal frequency, the particle properties such as the particle diameter, d, and the initial particle bed depth, h_0 [2]. These variables are combined to give the dimensionless bed depth, h_0/d, and the dimensionless acceleration amplitude, $\Gamma_a = a\omega^2/g$, where g is the acceleration due to gravity [2].

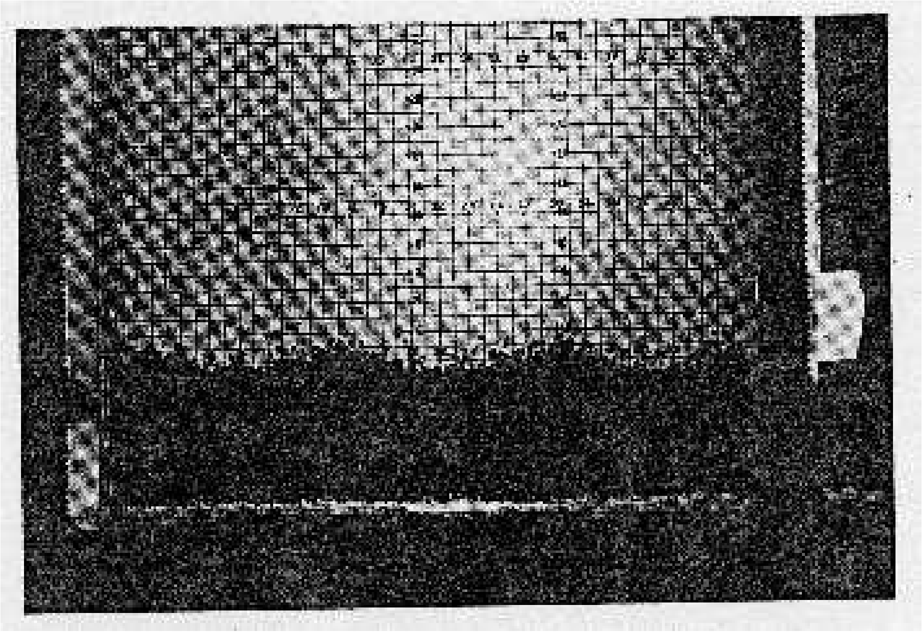


Figure 1.1: Small Amplitude Surface Waves: $\Gamma_a = 3.3$, $f = 20$ Hz, $d = 1.28$ mm, and $h_0/d = 17.5$ [2] (reprinted with written permission from publisher)

This is just one example of complex, collective phenomena observed experimentally in granular matter. Other phenomena found in granular materials include shear flow, dilatancy, solid-fluid transitions, convection rolls, free-surface flow, inclination angle, and density and stress fluctuations [1]. Vertical shaking experiments have investigated granular phenomena such as heap formation, convection motion, surface waves, stationary patterns, compactification, clustering and segregation [1]. Horizontal shaking experiments have studied the solid-fluid transition, critical-point exponents, crystallization, convection rolls, surface patterns, and inverted funnel flow in hoppers [1]. Stratification experiments between plates have been performed to study the avalanche movement and the angle of repose, segregation and dilatancy [1]. Conical hopper experiments have been used to study segregation during their filling, static wall stresses, outflow rate dependence on orifice and silo geometry, flow regions, segregation during outflow, density waves, dynamic wall stresses and silo design to decrease the stress fluctuations [1]. Rotating drum experiments have been used to extensively study different flow regimes (avalanches, continuous surface flow, and

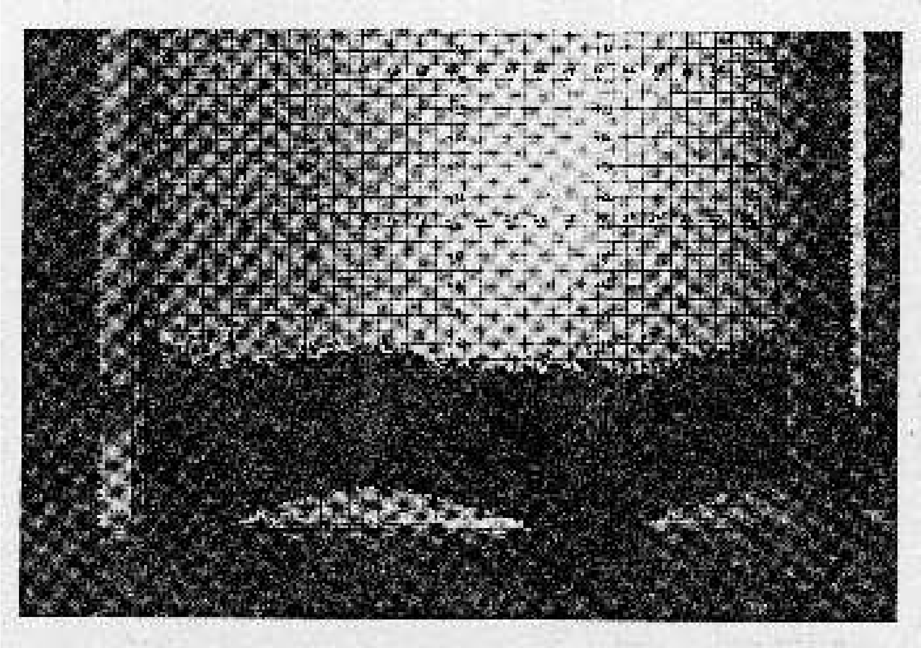


Figure 1.2: 3-node Arching: $\Gamma_a = 8.5$, $f = 30$ Hz, $d = 1.28$ mm, and $h_0/d = 17.5$ [2] (reprinted with written permission from publisher)

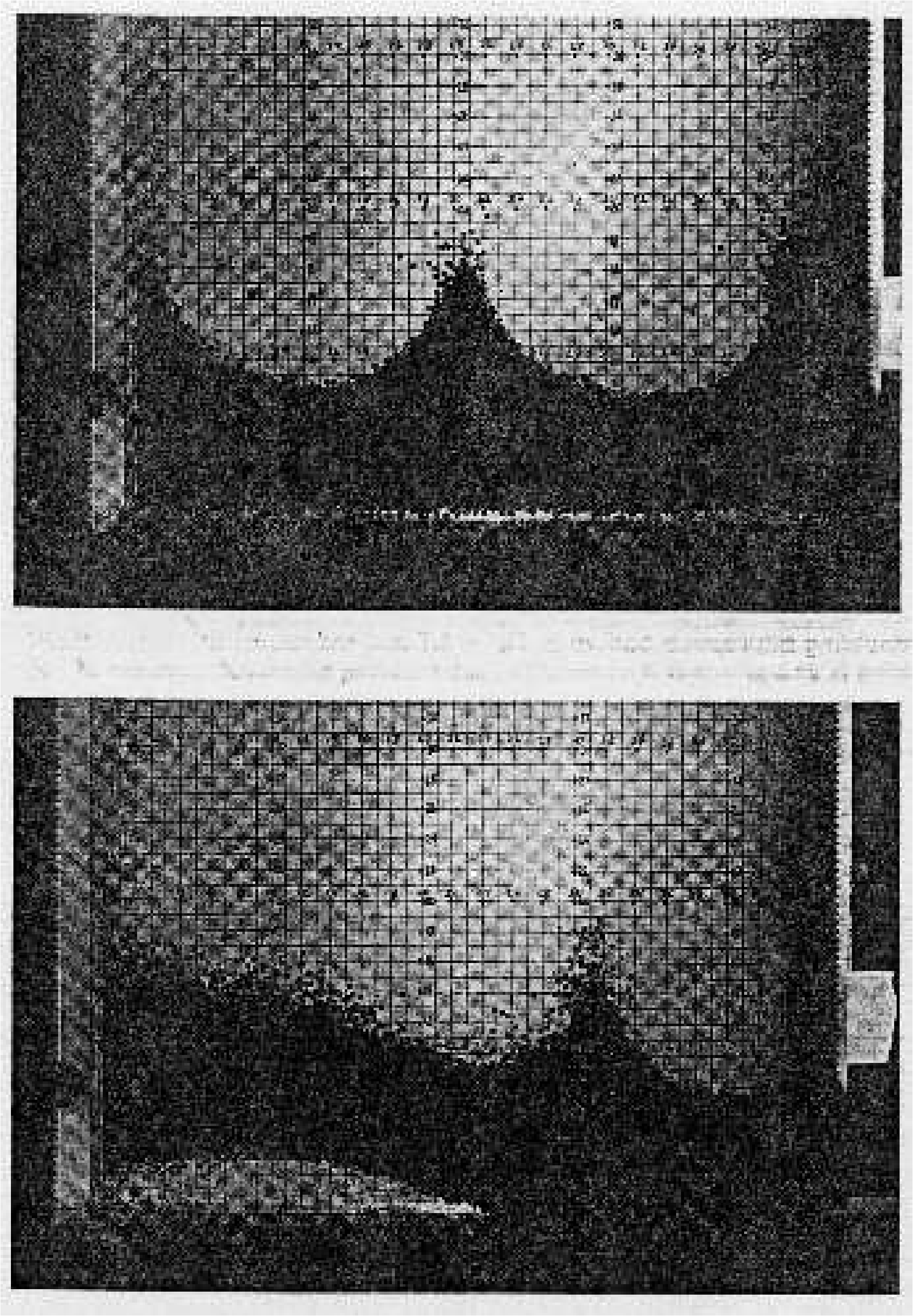


Figure 1.3: Large Amplitude Surface Waves: The top photo shows waves superimposed on 0-node arching and the bottom photo shows waves superimposed on 1-node arching for $\Gamma_a = 6.2$, $f = 20$ Hz, $d = 1.28$ mm, and $h_0/d = 17.5$ [2] (reprinted with written permission from publisher)

centrifugal regime), segregation (radial size segregation, radial density segregation, interplay of size and density segregation, friction-induced segregation, end-longitudinal segregation and axial segregation), axial band and wave dynamics, competition of mixing and radial segregation, front propagation and radial segregation [1]. More recently people have done experiments and simulations of oscillating, horizontally laid cylindrical pan geometries [3].

Furthermore, literature on granular materials often describes computer simulation of granular materials. Numerical simulation methods include Monte Carlo methods, diffusing-void models, method of steepest descent, cellular automata, event-driven simulations and time-driven molecular dynamics simulations [1]. These simulation methods are described in Appendix A. The latter two are the most commonly used.

Finally, literature on granular materials has also compared the results of computer simulations with experiments [4, 6]. The work of Bizon [4], for example, shows experimental and simulation results side by side. Laboratory experiments for vertically oscillated square pans filled with thin layers of lead spheres sieved between 0.5 and 0.6 mm were compared with computer simulations. As can be seen from Figure 1.4, the matching between the simulation and the experiment is remarkably close. This clearly demonstrates that computer simulations of granular matter are highly successful in quantifying granular phenomena.

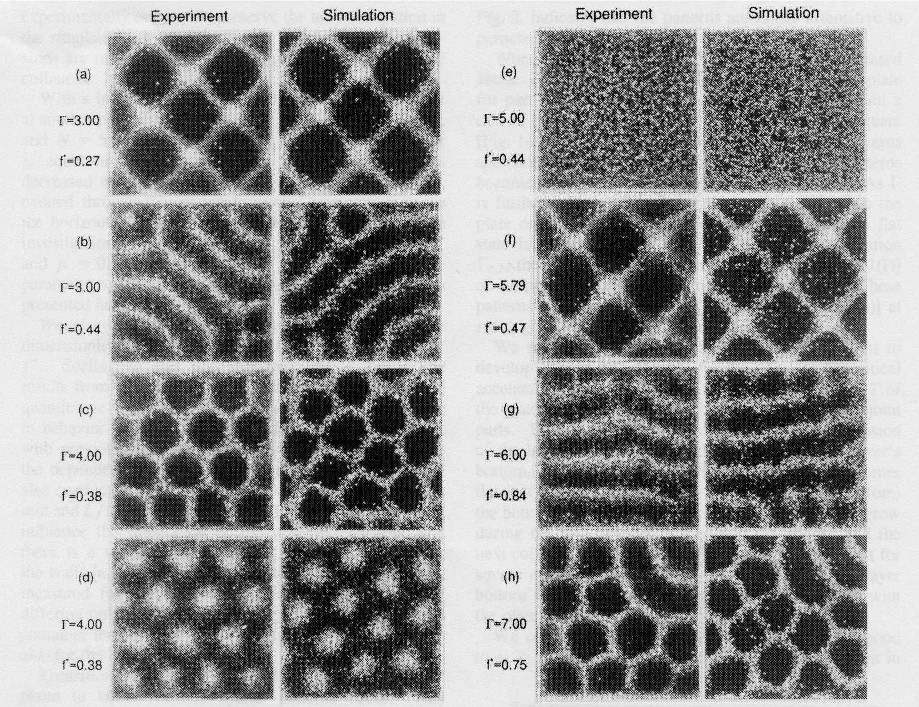


FIG. 1. Standing wave patterns: (a) squares, (b) stripes, (c) and (d) alternating phases of hexagons, (e) flat layer, (f) squares, (g) stripes, and (h) hexagons. Patterns (a)–(e) oscillate at f/2, (f)–(h) at $f/4$. The dimensionless layer depth N is 5.42. The brightness indicates the height of the layer. The experiments use lead spheres sieved between 0.5 and 0.6 mm.

Figure 1.4: Comparison of Experiment and Simulation: Numerical simulations and views from above of laboratory experiments conducted for vertically oscillated square pans filled with thin layers of lead spheres sieved between 0.5 and 0.6 mm. The simulations and experiments are a function of the frequency f, amplitude A, and depth h_o. In the figures, $\Gamma_a = 4\pi^2 f^2 A/g$ and $f^* = f\sqrt{h_o/g}$ where g is the acceleration due to gravity. The pictures show a top view of the standing wave patterns (squares, stripes, alternating phases of hexagons and flat layers) in the pans. The brightness indicates the height of the layer. [4] (reprinted with written permission from publisher and author)

This thesis describes an industrial granular crushing device (*Vibradrum* grinding machine) currently used in industry. See Figures 1.5 and 1.6. In the context of this thesis, the granular medium is composed of relatively large objects: three metre long, 2 centimetre diameter steel rods. Long rods are placed horizontally in a long, horizontally laid cylindrical drum that is supported on springs. The effect of the rods hitting the ends of the drum is neglected. The drum is shaken vertically with the goal of crushing a particular medium between the rods. Our goal is to ascertain the optimum oscillation that results in a force between the rods which achieves the ultimate stress of a particular medium that is to be crushed between the rods. By "optimum" we mean to minimize the total input of energy per unit time (power) required to grind the interstitial medium, which will depend on the fraction of collisions that produce a greater-than-threshold force, for a given power input. Here mixing will be important. Modes that produce large forces may not mix very well, resulting in some part of the medium being well crushed, while other parts are not. A literature search has revealed that simulations using vertical sinusoidal oscillation as a means of providing energy input, such as the one involving square pans as shown in Figure 1.4, and experiments and simulations of rotating cylindrical drums [1] exist, but (to our knowledge) no computer simulation study has been conducted of a vertically oscillating cylinder.

This thesis therefore attempts to simulate this device. The rods are simulated in two dimensions as disks in a vertically, sinusoidally oscillating, circular container. The motion of the disks is modelled through their classical equations of motion, with inelastic collisions between the disks and inelastic collisions between the disks and the circular container. Rotations of the disks are not incorporated in the model as it has been shown that the qualitative behavior of granular flow is not strongly affected by neglecting rotations [3]. For example, in the paper *Simulations of pattern formation in vibrated granular media*, Luding writes "This means that the rotational degree of freedom of the particles is not crucial for the instability to occur" [7]. By instability Luding means that he "find[s] surface patterns similar to surface waves obtained by parametric excitation in regular fluids, i.e. the Faraday instability [7]." Other papers by S. Luding and by the Bizon [4] group also have stated the same conclusion. Also effects of the rods crossing over one other or hitting the ends of the container have been neglected as these are believed to be minimal occurrences. Quantitative measures of collision forces are calculated, e.g., the change in normal velocity (change

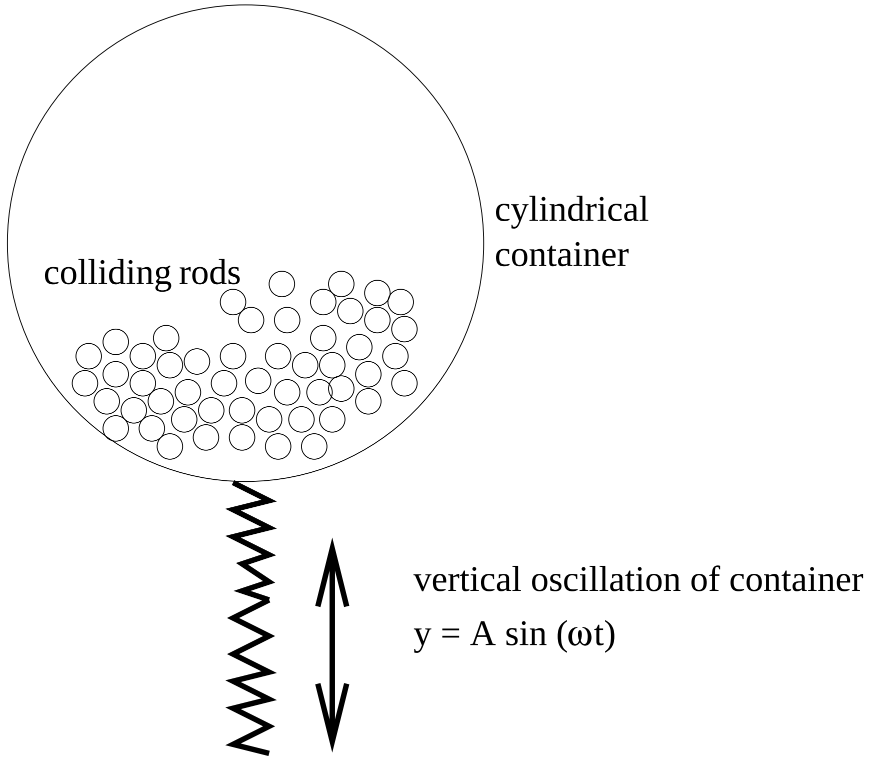


Figure 1.5: Schematic Diagram of Typical Experimental Drum Filled with Rods

in momentum) between the disks before and after a collision, and the collision energy (kinetic energy based on this change in normal velocity) . A distribution of these forces for various modes of oscillation is studied. A contour surface plot of these forces over the space of the container quantifies how the forces are spread across the container. A listing of the force that each disk experiences will indicate how the forces are spread across the population of disks. The flow of these disks is also studied and granular phenomena such as avalanche events which are related to the angle of repose, force chains, and the onset of collective motion (convection and circulation) are observed.

A particularly efficient serial event-driven algorithm for calculating collisions is used [8]. The programming language that is used is C++ and OpenGL libraries are used for the graphics.

The thesis is organized into the following chapters. The current chapter 1 has given a survey of granular literature and an introduction to the thesis topic. Chapter 2 describes the algorithm and formulae that are used in the computer simulation. Chapter 3 describes calculation of the crushing force. Chapter 4 discusses and analyzes the results of the simulation. Finally, chapter 5 presents the conclusions and recommends some possibilities for further investigation.

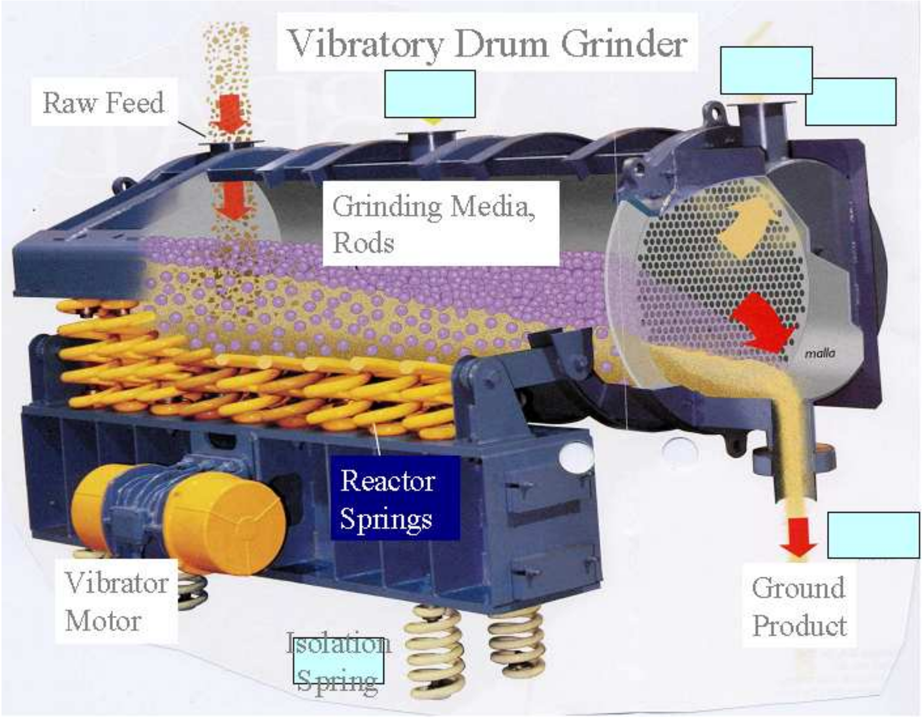


Figure 1.6: Typical Experimental Drum [5] The balls inside the drum are replaced by rods. (permission has been requested from publisher to reproduce this figure)

Chapter 2

ALGORITHM AND FORMULAE

2.1 Introduction

In this chapter we will develop the basic implementation of the algorithm and justify its use for modelling this particular application. The computer simulation's algorithm calculates Newtonian trajectories of hard, inelastic[1] bodies (rods simulated as two-dimensional smooth[2] disks) in a moving container (a moving drum simulated as a vertically oscillating, circular container). Two main types of algorithms were considered: an *event-driven* algorithm where calculations are performed and events are advanced discretely at each collision time step; and a *time-driven (molecular dynamics)* algorithm where events are advanced and calculations are performed continuously over a constant time step.

The event-driven algorithm was chosen because it is particularly advantageous for a simulation with relatively few particles where the collisions are instantaneous. In our simulation, there are only 150 particles that are all hard particles (steel rods) that have approximately instantaneous collisions (rod deformation is negligible for the motion). 150 particles were chosen to one-third volumetrically fill the container and match the experiments.

G. H. Ristow explains the advantages of the event-driven algorithm in his book entitled *Pattern Formation in Granular Materials* as follows:

> From a computational point of view it might be favorable in such a system to avoid using a constant time step, but rather to calculate all the collision times in the system and update all particle positions, velocities and accelerations to the shortest collision time calculated. This is especially

---

[1] In literature, all simulations have used inelastic particles to successfully simulate granular particle collisions [1].

[2] Simulating rough-surfaced disks is left as future work as described via rotations in Appendix C.

simple for hard spheres, where the collisions are instantaneous, i.e. the contact time is zero, and only binary collisions occur. After the colliding particles have been updated according to the collision model, the collision table is calculated and the circle starts anew.[1]

2.2 Event-Driven Lubachevsky Algorithm

An "event-driven" algorithm is used in the simulation. By "event-driven" we mean the shortest collision time is found over all possible collisions between pairs of disks and between any disk and the container, and once this shortest collision time is found, the disks' positions and velocities are advanced by this collision time. Then the shortest collision time is found for the next time step and the process is repeated. See Figure 2.1 for a typical arrangement of the event-driven simulation without gravity and Figure 2.2 for one with gravity.

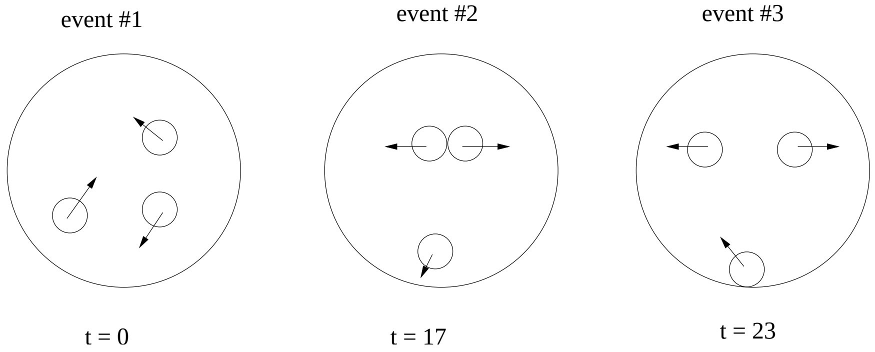


Figure 2.1: Typical Event-Driven Simulation Without Gravity. This simulation could be realized by having billiard balls colliding on a circular bordered pool table.

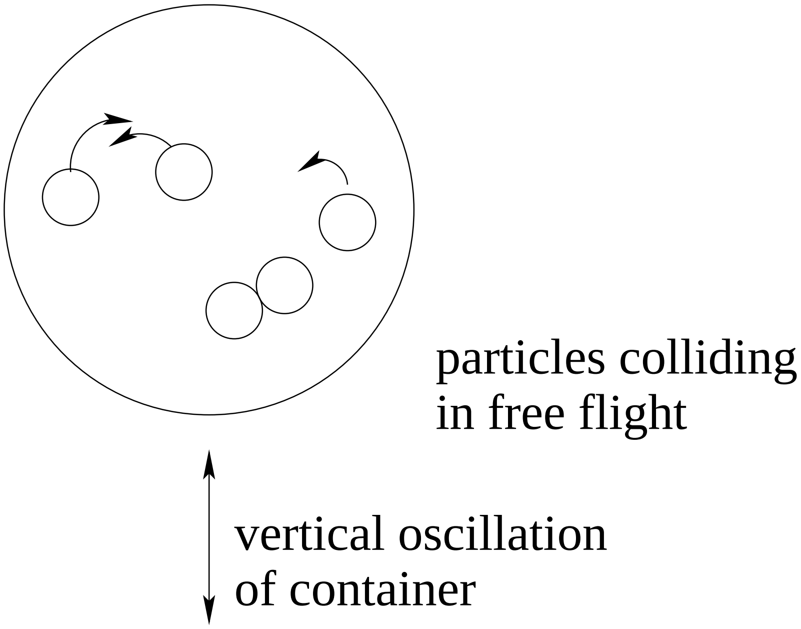


Figure 2.2: Typical Event-Driven Simulation With Gravity

An improvement of this algorithm was incorporated using an algorithm developed by Boris D. Lubachevsky that efficiently finds the shortest collision time using associated key “time values” [8]. The Lubachevsky algorithm works as follows. For simplicity, suppose there are only three disks in the container. Originally the “time value” (not the collision time) associated with each of the three disks is 0. Next suppose the first minimum collision time calculated is between disk 1 and disk 2 with a collision time of t1. The time value associated with disk 1 is set to t1 and the time value associated with disk 2 is also set to t1 and the positions and velocities of only the disks involved in the collision are updated.

By only updating the positions and velocities of the disks involved in the collision, rather than updating all the disks’ positions and velocities at each time step, the algorithm uses “delayed state updating” Not updating all the disks’ positions and velocities speeds up the algorithm significantly. Delayed state updating works as follows. The algorithm characterizes each collision as an *event*. The algorithm maintains an old and new event for each particle, essentially a double-buffering system as Lubachevsky explains:

> [An *event* is] ... the basic data unit of the algorithm and has the format *event = (time, state, partner)*, where *time* is the time to which *state* of a

> component responds. Note that *state* is the new state of the component *immediately after* the event, e.g., if a ... [disk] ... has experienced a collision at *time*, the velocity-coordinate of the *state* is the new velocity vector after the collision; *partner* identifies the other component ... [another disk, container, or sector border] ... involved in the event. At any stage of [the] simulation, the algorithm maintains two events for each component: an old, already processed in the past event and a new, next scheduled event. [8]

The time values of all the disks are stored in a heap tree data structure which has the property that the minimum time value is always at the top of the heap. Details of the heap data structure will be explained later. The algorithm then picks the disk associated with the minimum time value which is at the top of the heap which in our example we have chosen disk 3 with a time value of 0. The current time is set to this disk 3's time value. The algorithm next calculates the collision times between disk 3 and the other disks and between disk 3 and the container, namely between disk 3 and disk 1, between disk 3 and disk 2, and between disk 3 and the wall of the container. Suppose the minimum collision time is between disk 3 and disk 2 with a collision time of t2. The time value associated with disk 1 is unchanged, that is, it remains at t1. The time value associated with disk 2 is changed to its sum of collision times, that is, t1 + t2. The time value associated with disk 3 is changed to t2. Next, the positions and velocities of disks 2 and 3 are updated. Now suppose that t2 < t1. Then the algorithm chooses the disk with the minimum time t2, that is, the algorithm chooses disk 3 and calculates the collision times between disk 3 and the other two disks and the container wall. And the algorithm continues.

2.3 Sectoring

A further improvement that the program allows is dividing the physical domain into square sectors. (Although hexagons are the most efficient form of sectoring, squares are the simplest shapes to implement.) This way, collision times only between a disk and disks in its sector and its eight adjacent sectors can be calculated, eliminating the need to calculate collision times between a disk and all the other disks in the container. The only implementation change is that now a collision time is also calculated between a disk and a sector border as well. The Lubachevsky algorithm achieves a run time per particle collision which scales like $O(\log N)$ with sectoring and $O(N)$ without [8]. [Here N is the number of disks.] Since the simulation involved only 150 disks, sectoring was not used, although the option is available in the code should one want to simulate a larger system.

2.4 Heap Tree Data Structure

Since the event-driven algorithm always advances by the shortest collision time, a heap tree data structure was used to efficiently select the disk with the smallest "time value". The heap works as follows. For simplicity, consider a scenario with only four disks. Please refer to Figure 2.3 when reading the following description of the heap process. At the beginning, the program forms in a natural (starting at the top of the heap and moving down in levels, and going across at each level from left to right) order: disk 1 is at the top of the heap, then disks 2 and 3 at the second level (disk 1 is called the parent of disks 2 and 3 and disk 2 and 3 are called the children of disk 1), and then disk 4 is attached to disk 2 at the third level. See cycle 1 in Figure 2.3. In Figure 2.3, the time value of the disk is indicated in parentheses after the disk number. At each cycle the disk at the top of the heap known as the root of the tree is processed. Thus, at cycle 1 disk 1 is processed. As a result, the collision times between disk 1 and all the other disks and the collision time between disk 1 and the wall of the container are calculated. (Of course, if sectors are used, the collision times between disk 1 and its four surrounding sector borders are also calculated.) As before, suppose the minimum non-zero collision time calculated is between disk 1 and disk 2 with a collision time of t1(disk 1,disk 2) = 58. (The collision times will actually be floating point numbers but for ease of explanation, we will assume that they are integers.) As a result, the time values for disk 1 and disk 2 becomes 58. See cycle 1 in Figure 2.3. This time value of 58 together with disks 1 and 2 should move down the heap tree according to the heap tree rule: any parent must not have a time value key greater than its children. The time value is called a key because the tree is sorted by the key which is the time value. There are two possibilities to move disk 1 with time value 58 down: to the left or to the right branch. The program examines the branches from left to right. Disk 1 cannot move down left to the place of disk 2 since their time values are equal; thus disk 1 moves down right to the place of disk 3, disk 3 moves up to the root of the tree, and finally disk 2 moves to the place of disk 4, leaving disk 4 to move up to the spot vacated by disk 2. Disk 4 must not vmove to the top of the tree because disk 3 has the same time value of zero, and when time values are equal, the disks do not change positions. See the sorting phases of cycle 1 in Figure 2.3. Cycle 2 begins with disk 3 at the root. Disk 3 is processed and the shortest collision time for disk 3 is calculated to be t1(disk 2,disk 3) = 124. Thus the time value for disk 2 becomes 58 + 124 = 182, and the time value for disk 3 becomes 0 + 124 = 124. See cycle 2 of Figure 2.3. Now disk 3 must move down the tree. At the beginning of the cycle, because the disks are analyzed from left to right to follow the natural order, disk 4 on the left (not disk 1 on the right!) moves to the root and therefore gets processed. During this processing, the minimum collision time for disk 4 is calculated to be t1(disk 1,disk 4) = 25. Thus the time value for disk 4 becomes 0 + 25 = 25 and the time value for disk 1 becomes 58 + 25 = 83. See cycle 3 of Figure 2.3. During cycle 4, disk 4 gets processed. The minimum collision time for disk 4 is t1(disk 1,disk 4) = 150. Thus the time value for disk 4 becomes 25 + 150 = 175 and the time value for disk 1 becomes 83 + 150 = 233. See cycle 4 of Figure 2.3. Now disk 4 must move down to the place of disk 3 and disk 3 moves up to the root (top) of the tree. See the sorting phase of cycle 4 in Figure 2.3. The minimum collision time for disk 3 is then calculated and the processing of the heap tree continues. This explains the Lubachevsky algorithm that was used in the simulation.

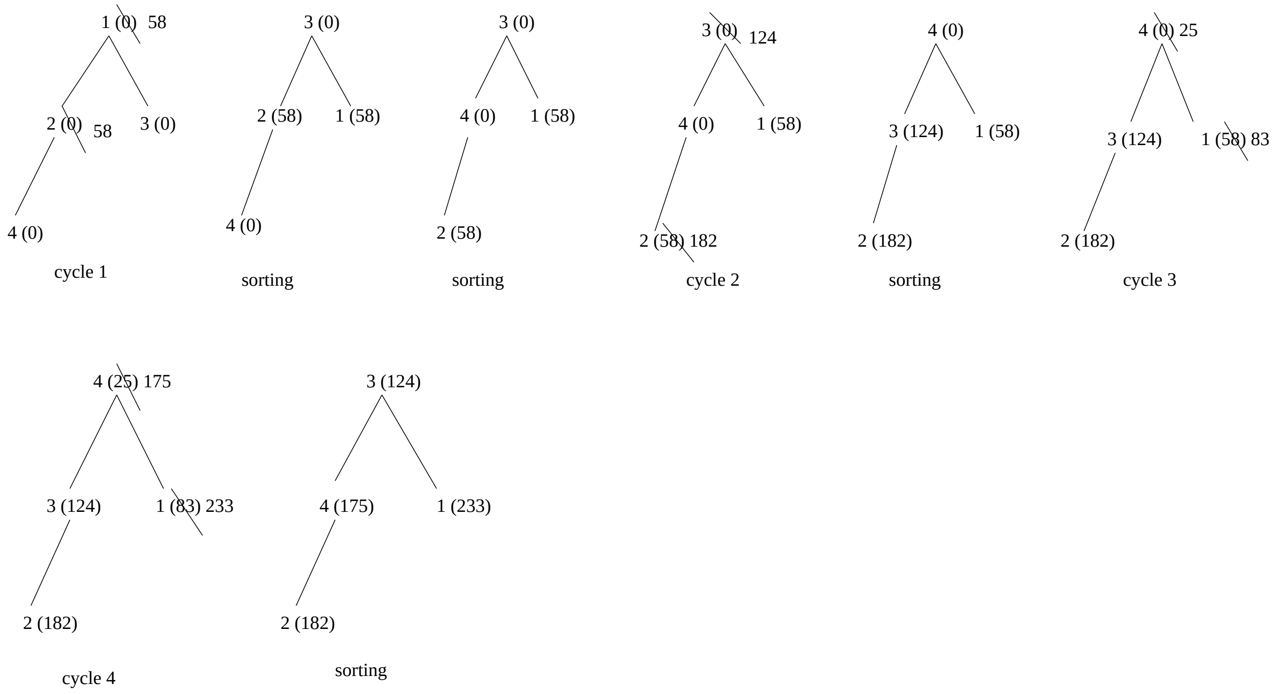


Figure 2.3: Heap Tree Data Structure [8]

2.5 Overlaps

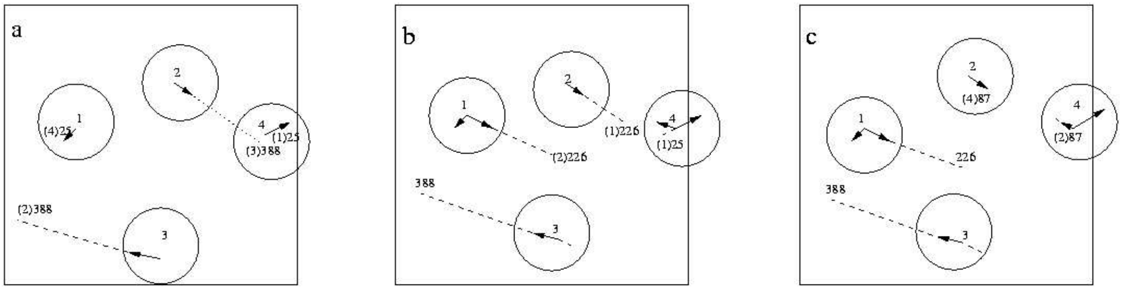
An inherent feature of event-driven algorithms is the overlapping of particles. This is a result of finite numerical precision. In addition, when low restitution coefficients are used, the system can undergo an infinite number of collisions in a finite time, leading to clustering and overlaps as well [1]. This is referred to as *inelastic collapse* [9, 10, 11, 12]. Thus the restitution coefficients between the disks, between the disks and the container, and an overlap tolerance parameter have to be tuned to avoid the simulation-ending inelastic collapse. We set the coefficient of restitution to be a function of velocity not only because it alleviates the problem of inelastic collapse (by avoiding the situation of having the disks stick together and collide an infinite number of times in a finite time interval), but more importantly because experiments show the restitution *is* a function of velocity in real materials [13, 14].

The exact formulation of this velocity dependence is detailed in section 2.6. Also, in particular, the previously described Lubachevsky algorithm has inherent overlaps of its own. Lubachevsky describes overlap occurrences in his simulation of colliding billiard balls as follows:

> The billiard simulation should be tolerant with respect to a small overlap of the balls. Figure 2.4 shows a "preemption of a preemptor" phenomenon when ball 1 has preempted a collision of balls 2 and 3 by scheduling an earlier collision with ball 2 ..., only to be later preempted by ball 4 which schedules an even earlier collision with ball 2 ... . In simulations with thousands of balls, more involved phenomena of this kind occur. While combined with the roundoff, they occasionally cause slight overlaps as shown in the following example. Suppose a scheduled collision of balls A and B for time(A,B) is later preempted by scheduling a collision of B and C for time(B,C) < time(A,B). As a result, the collision event for A becomes an advancement for time(A,B). Suppose that later in the computations, a collision of C and D scheduled for time(C,D) < time(B,C) preempts the collision of B and C. As a result, the collision event for B becomes an advancement for time(B,C). Now the originally scheduled collision of A and B for time(A,B) needs to be scheduled again. However, it will be done starting with different initial positions. If the ... [collision time calculation based on uniform gravity acceleration and conservation of momentum] is used in this scheduling, then ... [b]ecause of roundoff errors and different computational paths ... [the time] ... may be slightly negative as if balls A and B were slightly overlapping at time(A,B). [It is suggested that] ... the ... program handles this problem as follows: whenever [the] interaction time computes a negative but small by absolute value t, the value of t ... [be] ... replaced by zero. [8]

In our case, setting the time to zero does not work as the disks would not be advanced with a minimum zero collision time, so in the simulation, the time was set to 10^9 whenever t was negative to flag the collision in question to be dealt with later. Setting the time to 10^9 resulted in an overlap which was handled as described next.

The solution to handling overlaps is to allow an overlap tolerance between disks. Whenever an overlap greater than this tolerance occurred, the disks would be propagated backwards in time to within the overlap tolerance using the equations of motion in the presence of gravity, and the velocities of the overlapping disks would be adjusted as if they collided using conservation of momentum. If after propagating the



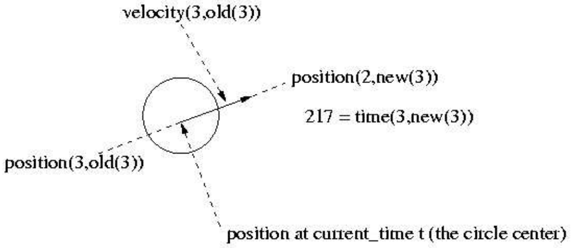


Figure 2.4: Preempted Collisions. Figure(a) result of cycles 1 to 4: current time = 0; balls 1 and 4 have scheduled a collision for time 25; balls 2 and 3 have scheduled a collision for time 388. (b) Result of cycle 5: current time = 25; ball 1 has processed its collision with ball 4 for time 25; balls 1 and 2 have scheduled a collision for time 226; ball 2 cancelled an earlier collision with ball 3 for later time 388 and its collision is turned into an advancement for ball 3. (c) Result of cycle 6: current time = 25; ball 4 has processed its collision with ball 1 at time 25; balls 2 and 4 have scheduled a collision for time 87; ball 2 has cancelled an earlier scheduled collision with ball 1 for later time 226 and this collision is turned into an advancement for ball 1. [8] (reprinted with written permission from publisher)

disks backwards in time, the disks overlapped with neighbouring disks, then a random “find spot” function was used to place one of the overlapping disks in an empty space within the container. For a million collisions, the very longest runs, the find spot function was used at most 0.02% of the time, and most of this occurred in the first 100,000 collisions of the simulation.

2.6 Rod-Rod Collision Dynamics

Disk-disk collision times were calculated as outlined in Allen and Tildesley’s book *Computer Simulation of Liquids* in their section on *Molecular Dynamics of Hard Systems* [15]. Their analysis is as follows. Consider two colliding disks, i and j, of diameter σ, whose positions at time t are $\mathbf{r}_i$ and $\mathbf{r}_j$, and whose velocities are $\mathbf{v}_i$ and $\mathbf{v}_j$. If these disks are to collide at time $t + t_{ij}$ then the following equation holds:

$$|\mathbf{r}_{ij}\left(t+t_{ij}\right)| = |\mathbf{r}_{ij} + \mathbf{v}_{ij}t_{ij}| = \sigma \tag{2.1}$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. It is important to note that the gravity term $-\frac{1}{2}gt_{ij}^2$ cancels out in equation (2.1) as shown below:

$$\begin{aligned} |\mathbf{r}_{ij} + \mathbf{v}_{ij}t_{ij}| &= |(\mathbf{r}_i + \mathbf{v}_i t_{ij} - \frac{1}{2}gt_{ij}^2) - (\mathbf{r}_j + \mathbf{v}_j t_{ij} - \frac{1}{2}gt_{ij}^2)| \\ &= |\left(\mathbf{r}_i + \mathbf{v}_i t_{ij}\right) - \left(\mathbf{r}_j + \mathbf{v}_j t_{ij}\right)| \\ &= |\mathbf{r}_{ij} + \mathbf{v}_{ij}t_{ij}| \end{aligned} \tag{2.2}$$

If we let $b_{ij} = \mathbf{r}_{ij} \cdot \mathbf{v}_{ij}$, then this equation becomes

$$v_{ij}^2 t_{ij}^2 + 2b_{ij}t_{ij} + r_{ij}^2 - \sigma^2 = 0. \tag{2.3}$$

This quadratic equation in t_{ij} can be solved using the quadratic formula. If $b_{ij} > 0$, then the disks are travelling away from each other and they will not collide. If $b_{ij} < 0$, it may still happen that $b_{ij}^2 - v_{ij}^2\left(r_{ij}^2 - \sigma^2\right) < 0$, and thus equation (2.3) will have complex roots and again the disks will not collide. Otherwise (assuming that the disks are not already overlapping) two positive roots arise, the smaller of which is the collision time

$$t_{ij} = \frac{-b_{ij} - \left(b_{ij}^2 - v_{ij}^2\left(r_{ij}^2 - \sigma^2\right)\right)^{1/2}}{v_{ij}^2}. \tag{2.4}$$

The velocities of the particles after impact are adjusted using conservation of total linear momentum and kinetic energy, and using equal masses of disks i and j, the velocity change $\delta \mathbf{v}_i$, such that

$$\mathbf{v}_i\left(after\right) = \mathbf{v}_i\left(before\right) + \delta \mathbf{v}_i \tag{2.5}$$

$$\mathbf{v}_j\left(after\right) = \mathbf{v}_j\left(before\right) - \delta \mathbf{v}_i . \tag{2.6}$$

The velocity change $\delta \mathbf{v}_i$ in equations (2.5) and (2.6) is given by

$$\delta \mathbf{v}_i = -\left(b_{ij}/\sigma^2\right)\mathbf{r}_{ij}\frac{1}{2}\left(1+e\right) = -\mathbf{v}_{ij}^{||} \tag{2.7}$$

with $b_{ij} = \mathbf{r}_{ij} \cdot \mathbf{v}_{ij}$ evaluated at the moment of impact and $\frac{1}{2}\left(1+e\right)$ being a coefficient of restitution factor. This restitution factor is explained in more detail later. As shown in Fig. 2.5, $\delta \mathbf{v}_i$ is the negative of the projection of $\mathbf{v}_{ij}$ along the $\mathbf{r}_{ij}$ direction, which is denoted as $-\mathbf{v}_{ij}^{||}$ in equation (2.7).

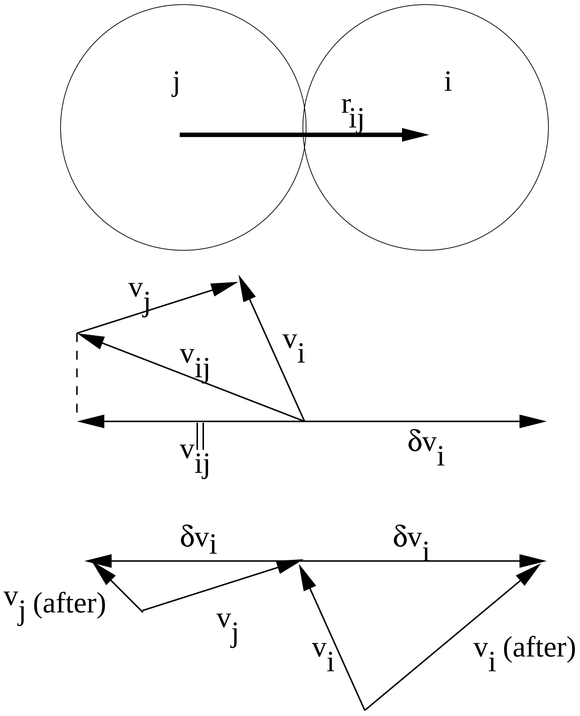


Figure 2.5: A smooth disk collision [15]

Equation 2.7 can be derived as follows. We know that at impact the change in momentum is proportional to the force. We also know that it will be a normal force along the line joining the centers of the disks in contact. Thus for disks i and j, with momenta $\mathbf{p_i}$ and $\mathbf{p_j}$:

$$\begin{aligned} \mathbf{p_i'} - \mathbf{p_i} &= C\mathbf{r_{ij}} && (2.8) \\ \mathbf{p_j'} - \mathbf{p_j} &= -C\mathbf{r_{ij}} && (2.9) \end{aligned}$$

where C is a constant. Here $\mathbf{p_i}$ and $\mathbf{p_j}$ are the momenta of disks i and j before a collision, and $\mathbf{p_i'}$ and $\mathbf{p_j'}$ are the momenta of disks i and j after a collision.

We also know from the conservation of kinetic energy and momentum:

$$\begin{aligned} \mathbf{p_i}^2 + \mathbf{p_j}^2 &= \mathbf{p_i}'^2 + \mathbf{p_j}'^2 && (2.10) \\ \mathbf{p_i} + \mathbf{p_j} &= \mathbf{p_i'} + \mathbf{p_j'}. && (2.11) \end{aligned}$$

Equations (2.10) and (2.11) can be combined to give

$$\mathbf{p_i} \cdot \mathbf{p_j} = \mathbf{p_i'} \cdot \mathbf{p_j'} \tag{2.12}$$

We also note that at impact $\mathbf{r_{ij}} \cdot \mathbf{r_{ij}} = \sigma^2$ where σ is the diameter of the disks. Combining this with Equations (2.8), (2.9) and (2.12), we have:

$$\begin{aligned} \mathbf{p_i} \cdot \mathbf{p_j} &= (C\mathbf{r_{ij}} + \mathbf{p_i}) \cdot (-C\mathbf{r_{ij}} + \mathbf{p_j}) \\ \mathbf{p_i} \cdot \mathbf{p_j} &= -C^2\sigma^2 + C\mathbf{r_{ij}} \cdot \mathbf{p_j} - C\mathbf{r_{ij}} \cdot \mathbf{p_i} + \mathbf{p_i} \cdot \mathbf{p_j} \\ 0 &= -C^2\sigma^2 + C\mathbf{r_{ij}} \cdot (\mathbf{p_j} - \mathbf{p_i}) \\ C\sigma^2 &= \mathbf{r_{ij}} \cdot (\mathbf{p_j} - \mathbf{p_i}) \\ C &= \frac{-m\mathbf{r_{ij}} \cdot \mathbf{v_{ij}}}{\sigma^2} \\ C &= \frac{-mb_{ij}}{\sigma^2} && (2.13) \end{aligned}$$

where m is the mass of a disk. Then upon substituting Equation (2.13) into Equations (2.8) and (2.9) we finally arrive at the desired Equations (2.5), (2.6) and (2.7) as shown below:

$$\begin{aligned} \mathbf{p_i'} - \mathbf{p_i} &= C\mathbf{r_{ij}} \\ m\mathbf{v_i'} - m\mathbf{v_i} &= \frac{-mb_{ij}}{\sigma^2}\mathbf{r_{ij}} \\ \mathbf{v_i'} - \mathbf{v_i} &= -\frac{b_{ij}}{\sigma^2}\mathbf{r_{ij}} && (2.14) \end{aligned}$$

and

$$\begin{aligned}\mathbf{p}'_{\mathbf{j}} - \mathbf{p}_{\mathbf{j}} &= -C\mathbf{r}_{\mathbf{ij}} \\ m\mathbf{v}'_{\mathbf{j}} - m\mathbf{v}_{\mathbf{j}} &= \frac{mb_{ij}}{\sigma^2}\mathbf{r}_{\mathbf{ij}} \\ \mathbf{v}'_{\mathbf{j}} - \mathbf{v}_{\mathbf{j}} &= \frac{b_{ij}}{\sigma^2}\mathbf{r}_{\mathbf{ij}}. \end{aligned} \tag{2.15}$$

The $\frac{1}{2}(1+e)$ in equation (2.7) is a coefficient of a restitution factor (with e varying between 0 and 1, i.e., $e = 1$ signifies a perfectly elastic collision) used by Walton [30] to account for dissipation of energy due to friction between the colliding disks. In fact the e is calculated as a velocity-dependent restitution coefficient developed by Bizon [4] and recently used by Baran [3] to reduce overlap occurrences as justified by experiments [9, 10, 11, 12] and defined below

$$e(v_n) = \begin{cases} 1 - Bv_n^{\beta} & , \quad v_n < v_0 \\ \varepsilon & , \quad v_n > v_0 \end{cases} \tag{2.16}$$

Here v_n is the component of relative velocity along the line joining the disk centers, $B = (1-\varepsilon)\, v_0^{-\beta}$, $\beta = 0.7$, $v_0 = \sqrt{g\sigma}$ and ε varying between 0 and 1 is a tunable parameter for the simulation. The advantages of using this velocity-dependent restitution coefficient are discussed in [4, 9, 10, 16, 17]. For example, in the paper entitled *Convection and Diffusion Patterns in Oscillated Granular Media*, it is stated that "Experimentally, e [the coefficient of resitution] is a function of collision velocity v_n. Further, simulations with a e that is independent of collision velocity are susceptible to divergences in the collision frequency" [16]. Similarly, in the paper entitled *Transport coefficients for granular media from molecular dynamics simulations*, it is stated that:

> When particles collide, new velocities are calculated by reversing the component of the relative particle velocity along the line joining particle centers and multiplying it by the coefficient of restitution e, which is between 0 and 1. If e is independent of collision velocity, a finite time singularity can occur in the collision frequency, a phenomenon known as inelastic collapse. In real materials, however, the coefficient of restitution is a function of collision velocity. [9]

Furthermore, rotations of the disks can be accounted for, although this *was not implemented* because it has been pointed out by Baran [3] and Luding [7] that the same qualitative granular phenomena can be obtained without rotations. Incorporation of rotations has been left for future work and details of implementing rotations can be found in the appendix.

2.7 Container Oscillations

The container was oscillated sinusoidally in a vertical direction using the following formula:

$$\bar{y} = A_y \sin\left(\omega_y t\right) \tag{2.17}$$

where $\bar{y}$ is the vertical movement of the centre of the container in cm, A_y is the amplitude in cm, ω_y is the angular frequency in radians/second and t is the current time in seconds. The horizontal location of the centre of the container remained stationary in the simulation. Horizontal oscillations can be implemented in future work.

2.8 Rod-Container Collision Dynamics

Since the container was moving, a numerical algorithm had to be used to calculate collision times between a disk and the wall of the container. The collision time was calculated in the reference frame of the moving container, and then translated back to the fixed frame at the end. The following equation was solved numerically:

$$\left(x + v_x t\right)^2 + \left(y + v_y t - \frac{1}{2} g t^2 - \bar{y}\right)^2 = \left(R - r\right)^2 . \tag{2.18}$$

Here x and y specify the location of the centre of the disk in cm, $\bar{y}$ is the vertical displacement of the centre of the container in cm (see equation (2.17)) v_x and v_y specify the disk's velocity in cm/s, $g = 981$ cm/s^2 is the acceleration due to gravity, t is the current time in seconds, R is the radius of the container in cm, and r is the radius of the disk in cm.

The *bisection method* was used to numerically solve equation (2.18). It is described in the book *Numerical Recipes in C*: "The idea is simple. Over some interval the function is known to pass through zero because it changes sign. Evaluate the function at the interval's midpoint and examine its sign. After each iteration the bounds containing the root decrease by a factor of two." [18]

In the simulation, the velocity of the disk after colliding with the wall of the container was adjusted using the following algorithm:

$$
\begin{aligned}
x_b &= x - x_C \\
y_b &= y - y_C - \bar{y} \\
v_y' &= v_y - \frac{d\bar{y}}{dt} - gt \\
\alpha &= \arctan\left(\frac{|y_b|}{|x_b|}\right) \\
\text{if } x_b &< 0 \text{ and } y_b > 0 \text{ then } \alpha = \pi - \alpha \\
\text{else if } x_b &> 0 \text{ and } y_b < 0 \text{ then } \alpha = \pi - \alpha \\
\text{else if } x_b &= 0 \text{ and } y_b > 0 \text{ then } \alpha = \pi/2 \\
\text{else if } x_b &= 0 \text{ and } y_b < 0 \text{ then } \alpha = 3\pi/2 \\
\text{else if } x_b &> 0 \text{ and } y_b = 0 \text{ then } \alpha = 0 \\
\text{else if } x_b &< 0 \text{ and } y_b = 0 \text{ then } \alpha = \pi \\
\text{if } |v_x| &\leq 1e - 15 \text{ then } \phi_i = \pi/2 \\
\text{else if } v_x &> 0 \text{ and } v_y' > 0 \\
\text{then } \phi_i &= \arctan\left(\frac{v_y'}{v_x}\right) \\
\text{else if } v_x &< 0 \text{ and } v_y' > 0 \\
\text{then } \phi_i &= \pi - \arctan\left(\frac{v_y'}{v_x}\right) \\
\text{else if } v_x &< 0 \text{ and } v_y' < 0 \\
\text{then } \phi_i &= \arctan\left(\frac{v_y'}{v_x}\right) + \pi \\
\text{else } \phi_i &= 2\pi - \arctan\left(\frac{v_y'}{v_x}\right) \\
\phi_r &= \pi + 2\alpha - \phi_i \\
v_A &= sqrt\left(v_x^2 + \left(v_y'\right)^2\right) \\
v_x'' &= \frac{1}{2}\left(1 + \varepsilon_W\right) v_A \cos\left(\phi_r\right)
\end{aligned}
$$

$$v_y'' = \frac{1}{2}(1+\varepsilon_W)\left(v_A \sin(\phi_r) + \frac{d\bar{y}}{dt}\right)$$

In the above algorithm, x and y specify the position of the centre of the disk, x_C and y_C specify the location of the centre of the container, $\bar{y}$ is the vertical displacement of the container, $\frac{d\bar{y}}{dt}$ is the velocity of the container (the time derivative of equation (2.17) $= A_y\omega_y \cos(\omega_y t)$), v_x and v_y specify the velocity of the disk before colliding with the wall of the container, t is the collision time between the disk and the container and v_x'' and v_y'' specify the velocity of the disk after colliding with the wall of the container. ϕ_i and ϕ_r are the incident and reflected angles of the velocity of the disk with respect to the surface normal to the container, and e_W is the coefficient of restitution for a disk-container collision varying between 0 and 1. $e_W = 1$ signifies a perfectly elastic collision. In the above algorithm, the formula $\phi_r = \pi + 2\alpha - \phi_i$ is derived from Figure 2.6.

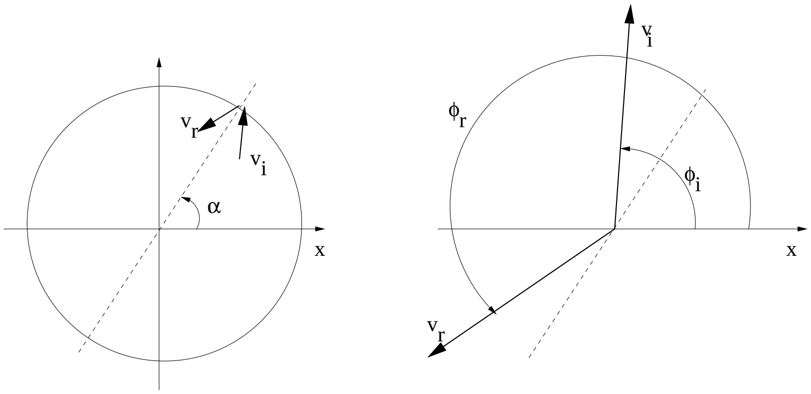


Figure 2.6: Calculation of reflected velocity $\phi_r = \pi + \alpha - (\phi_i - \alpha) = \pi + 2\alpha - \phi_i$[3]

2.9 Circulation

To understand the flow patterns of the colliding disks, the circulation was calculated. (The exact definition of circulation via formulae is described later in this section.) Visual observation and vector velocity flow field diagrams (which were produced from a tabulation of the velocities and positions at each time instant, time and space averaged over a 0.1 cm grid, then normalized using the average velocity and finally plotted using *gnuplot*) indicated that in all observed simulations there was a large steady state, elliptical vortex near the bottom of the container with subsidiary smaller vortices. This was because in all simulations the initial positions of the disks were skewed (such that the centre of mass of the disks was to the left or right of the vertical axis of symmetry of the container) resulting in an initial angular momentum, and thus inducing a net circulation. Inducing circulation obviously helps the mixing in the grinding and that is why the initial conditions were chosen to induce circulation. See Figures 2.7 and 2.8.

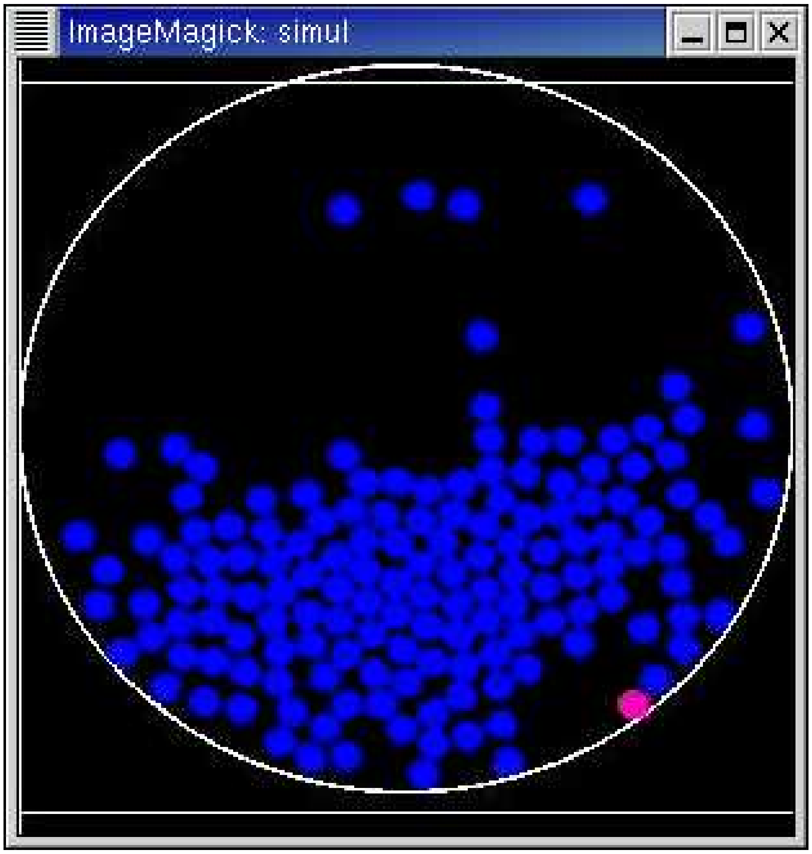


Figure 2.7: Typical Simulation Snapshot After 100,000 Collisions: frequency of oscillation $\omega_y = 126$ rad/s, amplitude of oscillation $A_y = 1.5$ cm, disk-disk coefficient of restitution $e_0 = 0.4$, disk-container coefficient of resitution $e_W = 1.0$. The container is the large white circle. The disks are the smaller blue circles except for the latest colliding disk which is the circle coloured in pink.

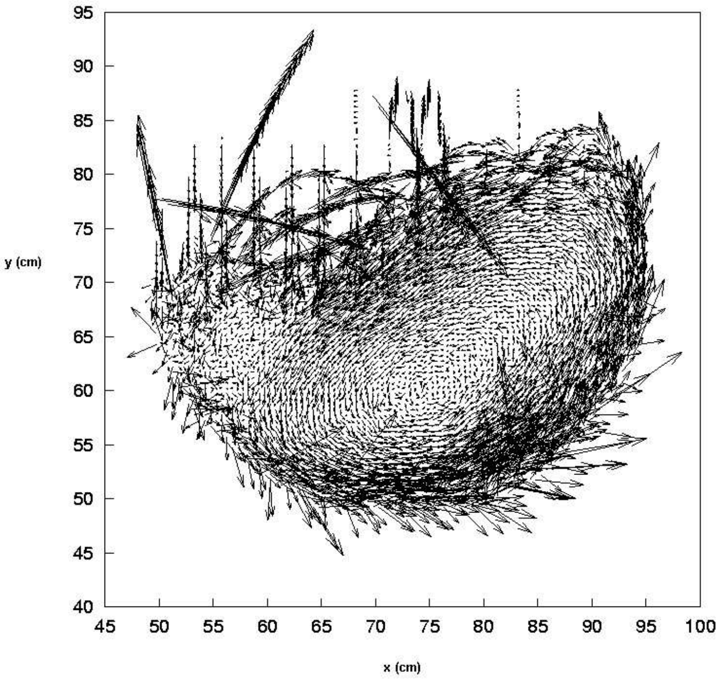


Figure 2.8: Typical Time Averaged Velocity Field After 100,000 Collisions. The large almost straight trajectories at the top are due to freely flowing particles near the beginning of the simulation.

The net circulation Γ of the whole system was calculated by first calculating the overall angular velocity ω of the vortex about the centre of mass of the system and then using the formula

$$\Gamma = \int \Omega dA = \int 2\omega dA = 2\omega\pi r_{max}^2, \tag{2.19}$$

where $\Omega = 2\omega$ is the angular velocity of rotation or *vorticity* of the system and r_{max} is the distance from the centre of the farthest disk to the centre of mass of the system. Equation (2.19) was developed by Hughes [19] who defines the rotation or vorticity in a fluid as

$$\Omega = \nabla \times V \tag{2.20}$$

where $V = -\nabla\phi$ is the velocity that is derivable from a scalar velocity potential ϕ. Hughes also relates the angular velocity of an infinitesimal element of fluid ω to the vorticity Ω as

$$\Omega = 2\omega. \tag{2.21}$$

According to Kelvin's theorem [19]:

$$\frac{d}{dt}\int_s \Omega dA = 0. \tag{2.22}$$

Thus any area that encloses the velocity field is valid. The simplest area of a circle was chosen whose area is πr_{max}^2.

The angular velocity of the vortex was calculated using a modification of a formula developed by Baran [3] (see Figure 2.9):

$$\omega = \frac{1}{N}\sum_i \frac{1}{a_i}(\mathbf{v}_i - \bar{\mathbf{v}}) \cdot (\hat{\mathbf{k}} \times \hat{\mathbf{a}}_i). \tag{2.23}$$

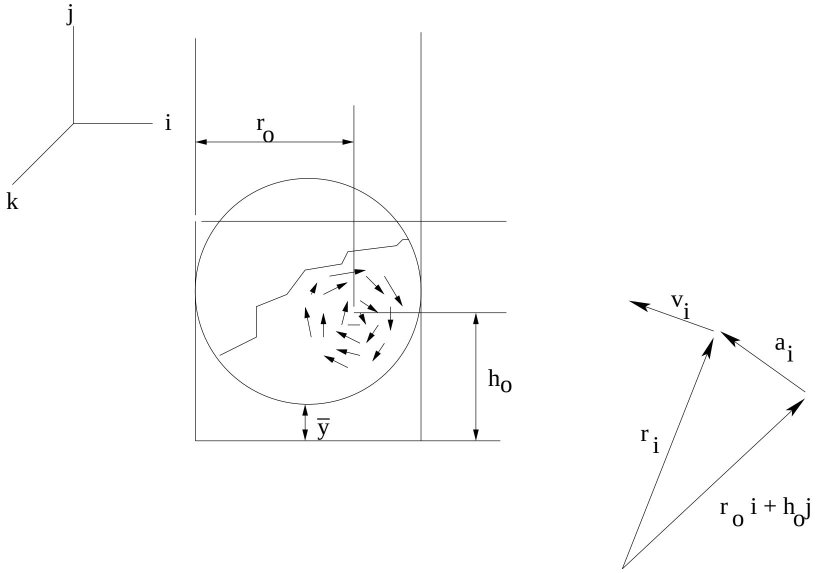


Figure 2.9: Circulation Diagram [3]

Here, $\hat{\mathbf{a}}_i$ and a_i are the direction and length of vector $\mathbf{a}_i = \mathbf{r}_i - (r_o\hat{\mathbf{i}} + h_o\hat{\mathbf{j}})$ where r_o is the horizontal distance to the centre of the vortex, h_o is the vertical distance to the centre of the vortex. r_o and h_o are shown in Figure 2.9. N is the number of disks in the container, v_i is the velocity of the *ith* disk and $\bar{\mathbf{v}}$ is the average velocity of all the disks. Subtracting the average velocity was required to make the circulation invariant under a Galilean transformation. This frame invariance was checked for a circular vortex using potential flow theory. [19]

According to the potential flow theory, the vorticity Ω and its derivative can be expressed as a function of a complex variable $z = x + iy$ as [19]:

$$\Omega(z) = U_0 z + \frac{\Gamma}{2\pi i}\ln(z - z_0) \tag{2.24}$$

$$\frac{d\Omega}{dz} = U_0 + \frac{\Gamma}{2\pi i}\frac{1}{z - z_0} = u - iv \tag{2.25}$$

where u and v are the velocities in the x and y direction, respectively. U_0 is a Galilean transformation.

Equation (2.25) can be rearranged to give

$$u = U_0 - \frac{\Gamma}{2\pi}\frac{(y - y_0)}{(x - x_0)^2 + (y - y_0)^2} \tag{2.26}$$

$$v = \frac{\Gamma}{2\pi}\frac{(x - x_0)}{(x - x_0)^2 + (y - y_0)^2} \tag{2.27}$$

where x_0 and y_0 specify the centre of the vortex, $\Gamma = \Gamma_{max}\left(\frac{r}{r_{max}}\right)^2$ is the circulation where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the radial distance of a particle to the centre of the vortex and r_{max} is the distance to the farthest particle.

The following parameters were chosen as a test case for verification: x_0 and y_0 as the centre of the vortex, $\Gamma_{max} = 100\pi$, $U_0 = 1000$, and random x and y positions of 150 particles. The velocities, u and v were calculated using equations (2.26) and (2.27) to generate the velocity flow field, and then using these velocities, the overall angular velocity and circulation were calculated using equations (2.19) and (2.23). The maximum circulation of $\Gamma_{max} = 100\pi$ was successfully recovered. In the simulated data, u and v are not calculated but rather are values obtained from conservation of energy and momentum. In the observed simulations' time-averaged velocity field diagrams (see Figure 2.8), there is mainly one large elliptical shaped vortex. This can be explained as follows. The system will begin with no predominant circulation during the transient phase but then as the steady state phase is approached there will be a fluctuation-driven symmetry breaking and the system will pick either a clockwise or counter-clockwise circulation. This will result in a sloped angle of repose: negatively sloped down to the right for a clockwise circulation or positively sloped up to the right for a counter-clockwise circulation. The system basically has a mirror of symmetry about a vertical line through the centre of the container. We can break that symmetry early and control which of these two symmetric motions the system adopts by skewing the initial positions of the disks to one side or the other. This skewness creates an initial angular momentum, and thus induces a net circulation which aids the mixing in the grinding. It was found that when the disks' initial positions were skewed to the left it induced a net counter-clockwise circulation, and when the disks' positions were skewed to the right it induced a net clockwise circulation. When the disks' initial positions were symmetric there was no predominant circulation at the beginning. For the analysis, a counter-clockwise circulation was chosen in order to study and compare the results consistently for the same counter-clockwise symmetry. A similar analysis could be done for a clockwise symmetry. It is to be noted that a *Marangoni* or *Bénard* effect could create a double vortex if the excitation is increased enough in the symmetric situation as shown in Figure 2.11 [20, 21]. Mills describes the *Marangoni effect* in a fluid in the article *The Effect of Interfacial Phenomena of*

Materials Processing as follows:

> When a thin layer of fluid (Figure 2.10) is heated from below, any instability which occurs will result in the transfer of hot liquid to the surface. For most liquids surface tension decreases with increasing temperature, thus this transport of hot liquid will result in a lower surface tension at the point of emergence. Consequently, there will be a radially outward flow of liquid along the surface. Such behaviour results in the formation of cells (known as Bénard cells) with a hexagonal or polygonal geometry. [21]

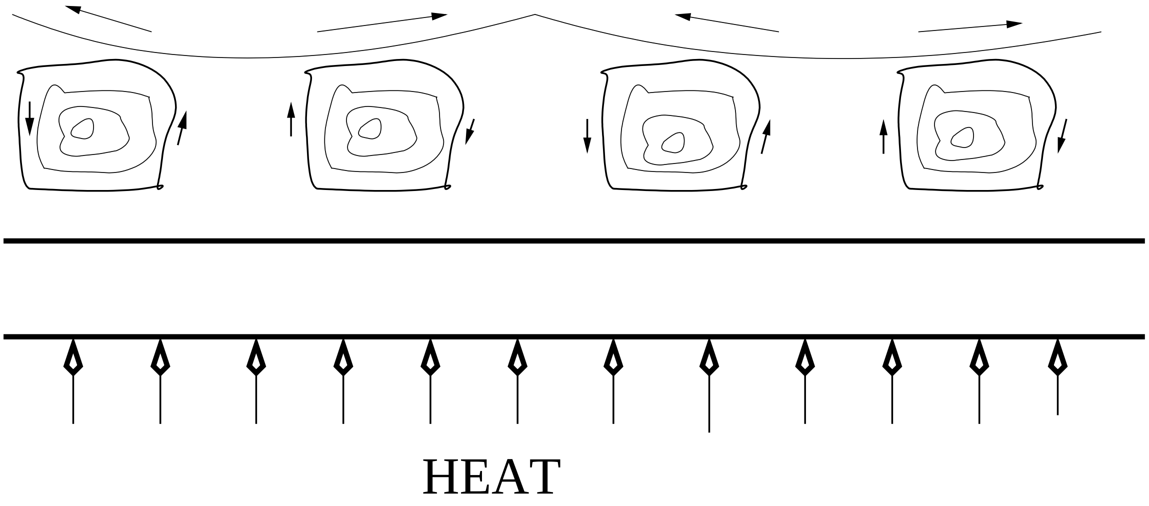


Figure 2.10: Schematic Representation of the Formation of Bénard Cells [21]

In calculating the circulation from the simulated data, the centre of mass of the system was chosen for the purpose of calculating the angular velocity as shown in Equation (2.23) See Figure 2.12 for a typical plot of net circulation vs time for an asymmetric case and Figure 2.13 for a symmetric case.

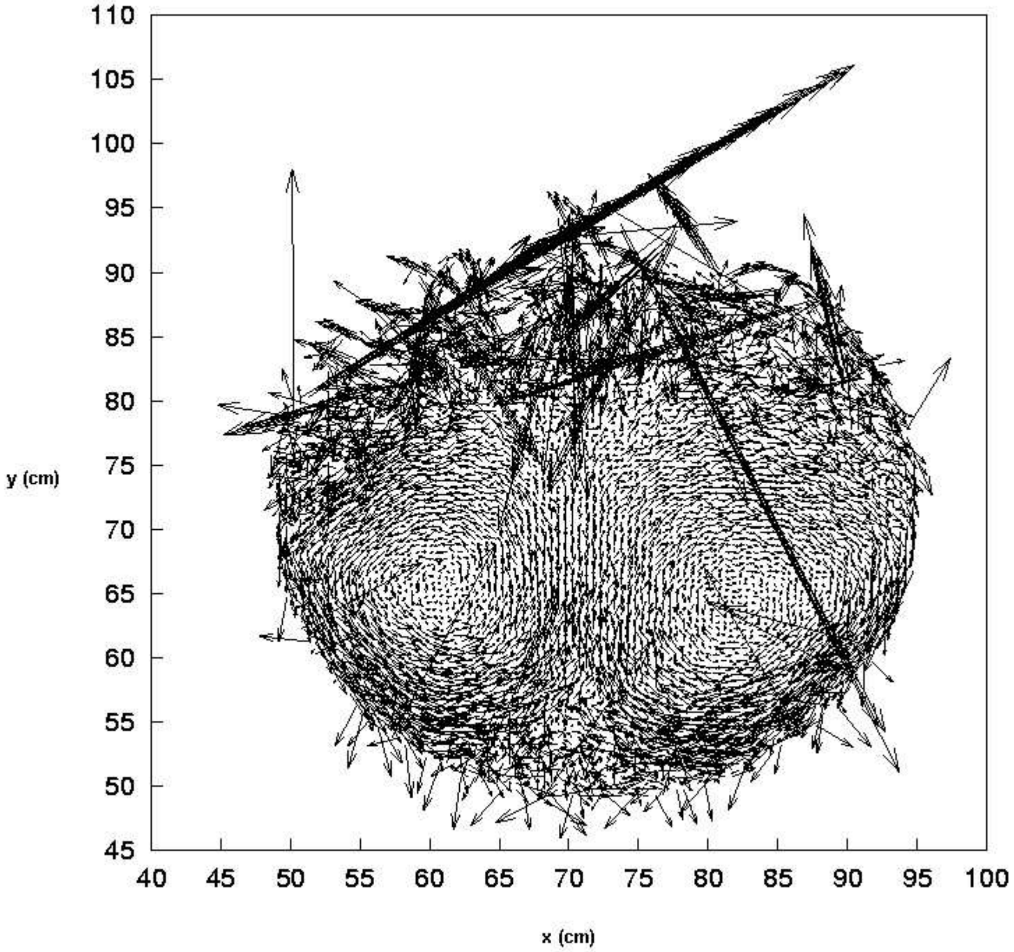


Figure 2.11: Typical Velocity Field for Symmetric Simulation

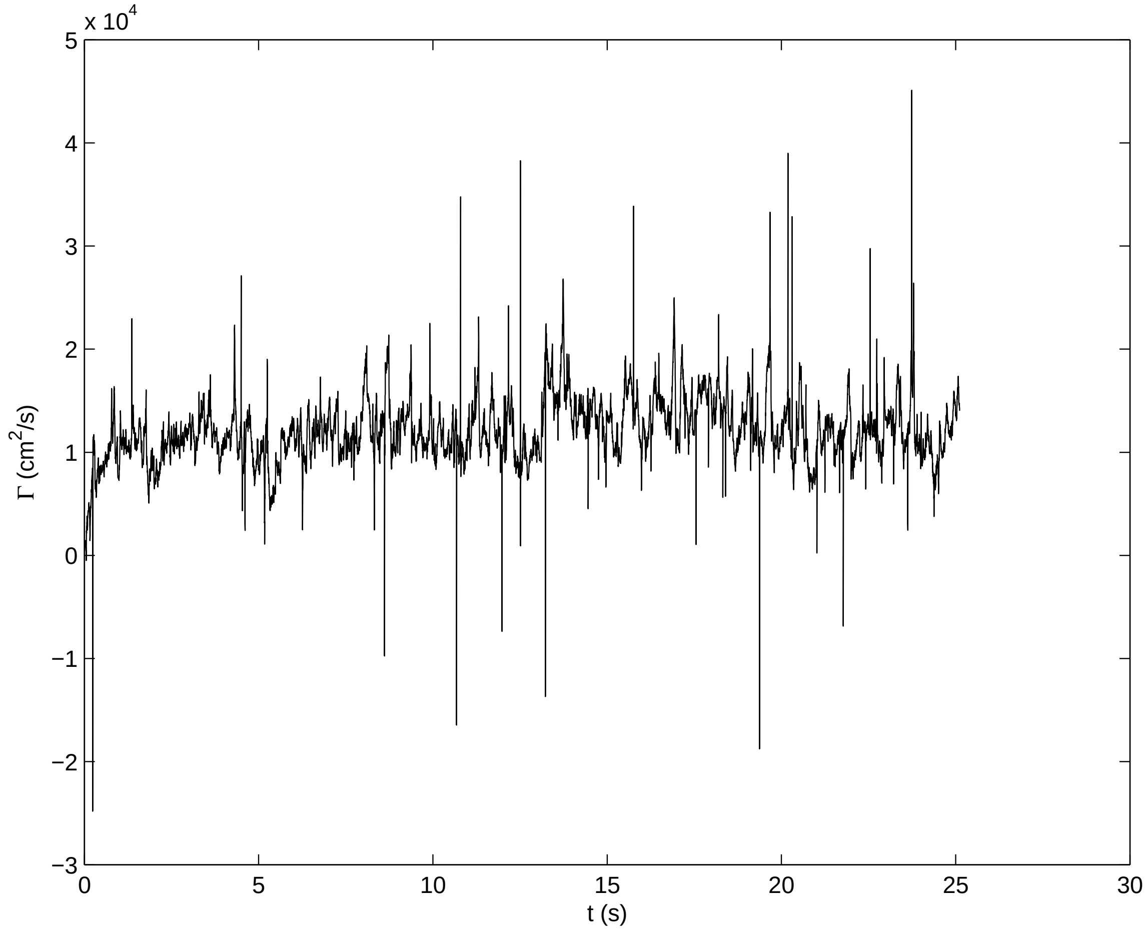


Figure 2.12: Net Circulation vs Time for Typical Asymmetric Simulation. The spikes in this plot are due to accelerated disks freely flowing at the top of the cluster. These freely flowing disks are mobilized by the transfer of energy from the container periodically hitting the bottom of the cluster.

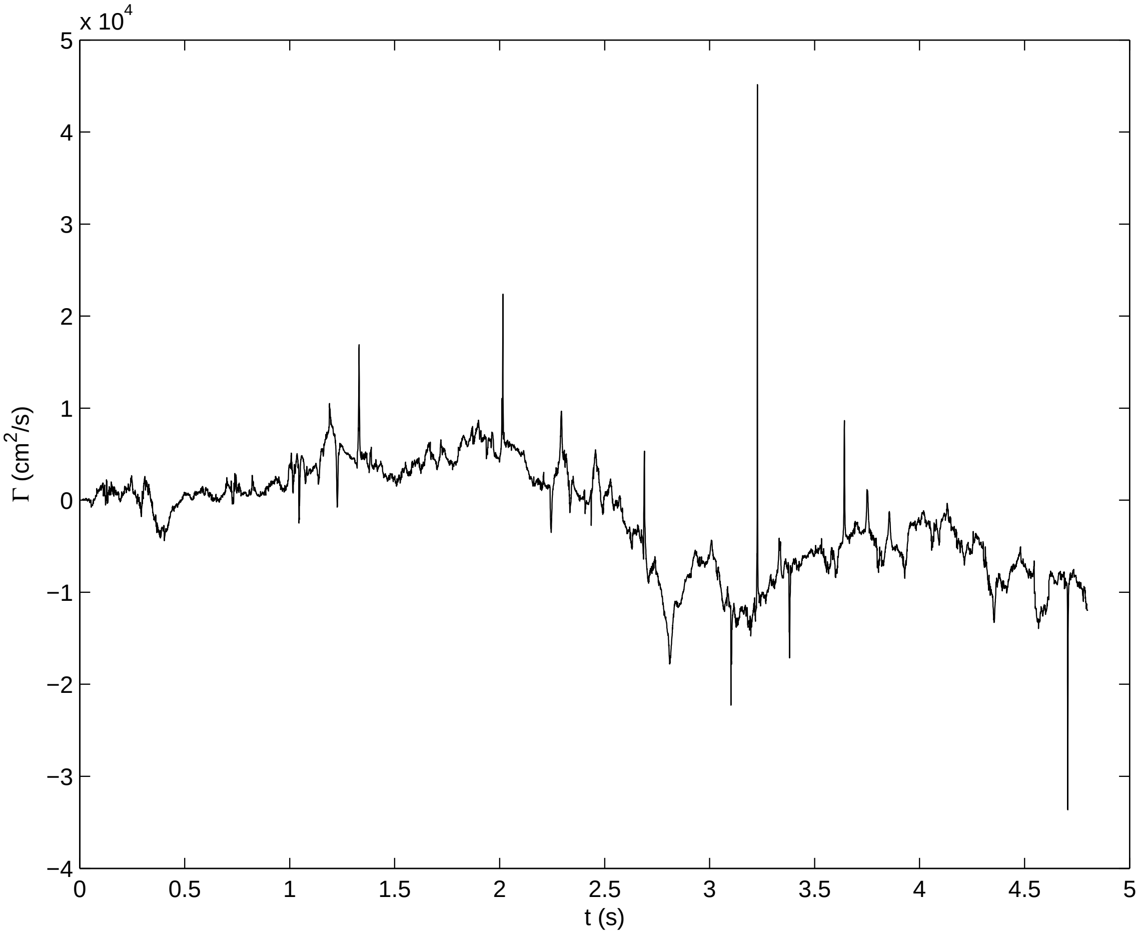


Figure 2.13: Net Circulation vs Time for Typical Symmetric Simulation

2.10 Object-Oriented Structure of Program

The program was written using the C++ programming language. OpenGL libraries were used for the graphics. There are 16,700 lines of simulation code and 6,247 lines of post processing code in total. It was natural to break the program up into object modules called *classes*. A class contains data and functions that operate on its data. Classes can be abstract base classes, and concrete classes which can inherit these base classes and other concrete classes. Template classes that can take data of any type can also be used in C++. All these C++ features were used in developing the program. The classes are listed in alphabetical order below. A description of the class design follows the listing which is finally followed by an object diagram showing interdependencies between the classes.

- Buffer (template class used to create heap for sorting collision times)
- Circle (concrete class)
- Circle_Buffer (concrete class)
- Container (abstract class)
- Container_2DSSC_Box (concrete class)
- Event_2D (concrete class)
- Event_Buffer (concrete class)
- List (template class used to create a sector list for each event and an event list for each sector)
- Long (used to create a composite data type with an integer and a floating point number)
- Sector (abstract class)
- Sectorization (abstract class)
- Square_Sector (concrete class)
- Square_Sectorization (concrete class)

- Walls (abstract class)
- Walls_2D_Box (concrete class)
- Walls_2D_Box_Circles (concrete class)

Following is a brief description of the object-oriented class design.

The purpose of the simulation is to take a container of N particles, in this case a circular container (inscribed in a 2D square box), fill it with disks represented by circles, and divide the container into sectors. When the box is divided into many sectors, collisions can be calculated sector by sector. A particle need only calculate collisions with particles in adjacent sectors and interactions with nearby sector borders.

The Event_2D class consists of x and y position vectors, a horizontal velocity vector component, v_x, and a vertical velocity vector component, v_y, a time variable, and an identification (id) field assigning the Event_2D a number equal to that of the particle it is simulating, and a partner field which is used abstractly to hold the identification numbers of particles with which the particle has collided. The two classes Event_2D and Sector are co-dependent with two many-to-many associations between them. The *OLD* association is implemented in both classes using the List class to create a linked list of pointers. Thus the Sector class is abstract and contains a unique id field and a linked list of Event_2D pointers instantiating the *OLD* association. The *NEW* association is implemented in only Event_2D. An Event_2D object cannot, by design, have a *NEW* and *OLD* association at the same time, but must have either an *OLD* association or a *NEW* association at all times. An Event_2D object with an *OLD* association is said to be in an old state and one with a *NEW* association is said to be in a new state. Thus, included with all the other attributes are two SectorList pointers, Old and New, created using the List class. The lists pointed to contain links (more pointers) to the sectors which the given Event_2D object resides in. One of the list pointers is empty and one is full at all times. It is this feature which assigns the Event_2D object a state of old or new. These old and new states correspond to the buffer implementation featured in the Lubachevsky algorithm as described in section 2.2. The Lubachevsky algorithm details a new and old state for each particle, so the id field of Event_2D is used to correspond to the new and old states of a particle.

The Circle class inherits all of the attributes of the Event_2D class. It adds only a radius attribute to the list of attributes acquired from Event_2D. The Square_Sector class inherits all the attributes of the Sector class. It adds a width attribute, which specifies the width of a square shaped sector, and four Square_Sector pointer attributes that hold addresses of adjacent Square_Sector objects – in a grid a square will have a maximum of four other squares adjacent to it. The Sectorization class holds and thus inherits many Sectors, and the Square_Sectorization class holds and thus inherits many Square_Sectors.

The Event_Buffer class is a heap of Event_2D classes and the Circle_Buffer class is a heap of Circle classes. The Event_Buffer class and the Circle_Buffer class use the template Buffer class which contains the abstract heap. (See section 2.4 for a description of the heap.) The Container_2DSSC_Box class uses the Circle_Buffer class (for particle-particle interactions) and Square_Sectorization class (for particle-sector border interactions) and Walls_2D_Box_Circles class (for particle-wall interactions) in processing the Lubachevsky algorithm. The Walls_2D_Box_Circles class inherits the Walls_2D_Box class and its abstract Walls class.

Following is an *object diagram* showing the interdependencies of the object classes.

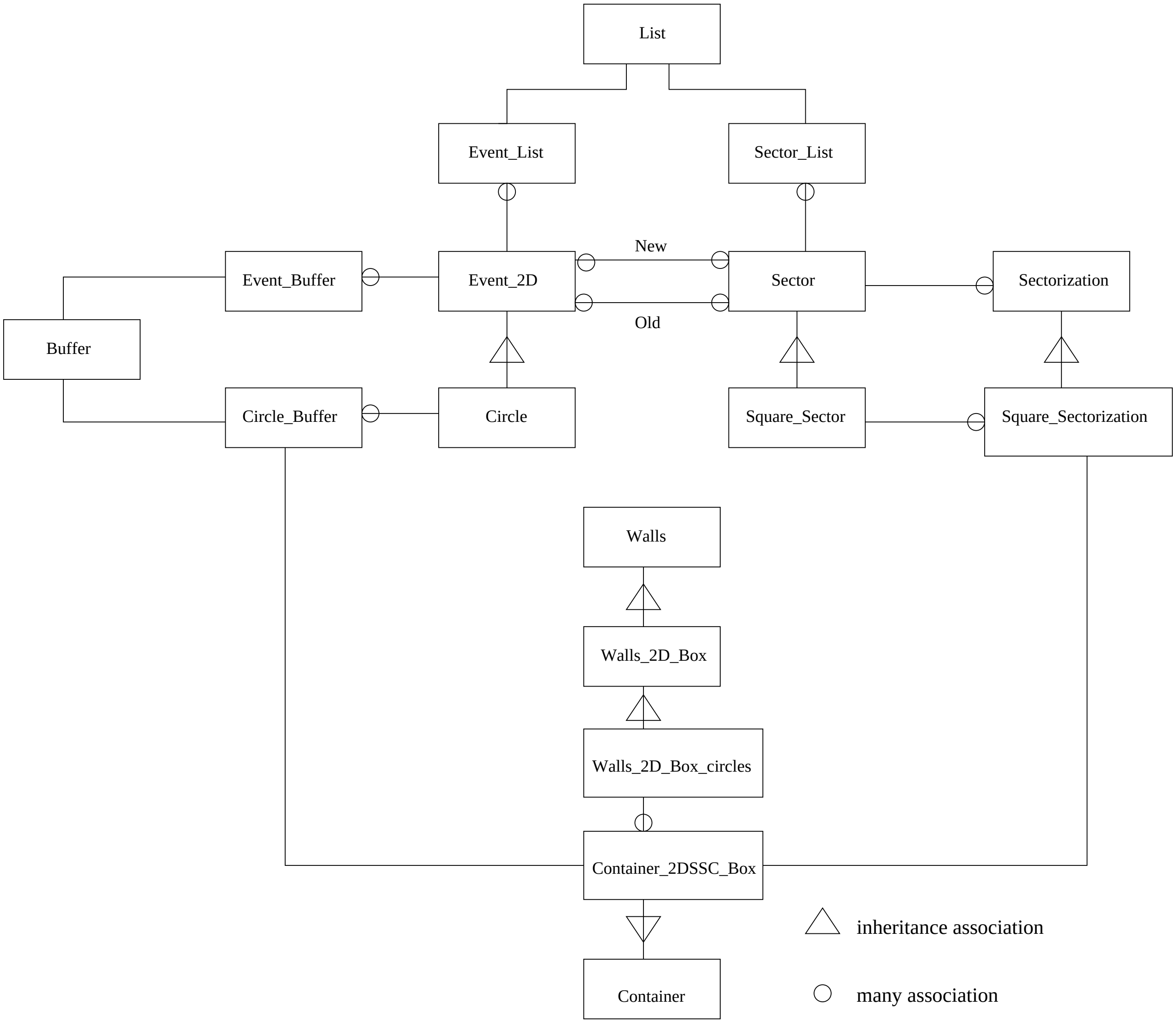


Figure 2.14: Object Diagram

Chapter 3

FORCES

3.1 Introduction

The purpose of the simulation is to optimize[1] the container oscillation to achieve a desired crushing force (the force that a particle would experience if it were between two colliding disks). Thus devising a method of measuring the normal crushing force between the disks is an integral part of this thesis. In Appendix D is a description of the methods that have been previously used to measure forces *directly* between particles in granular material simulations based on elasticity theory, and an explanation why these methods are inappropriate for an industrial crushing application. Basically, in order to calculate forces directly, one has to make assumptions that the disks behave elastically, and it has been shown from experiments that the values obtained for forces when assuming elastic behaviour are off by many orders of magnitude [22]. Besides, the forces on the disks is not what is desired, but rather the forces on the particle to be crushed between the disks and between a disk and the container. Thus, rather than measuring forces directly, we adopt a method that relates a *collision energy* between the disks to the *modulus of toughness* of a material that is to be crushed. The modulus of toughness is a physical parameter that can be measured by experiment. There exist various testing machines such as the *Satec Systems Universal machine* [23] that uses its microprocessor controlled operations to calculate the modulus of toughness.

---

[1] We are optimizing the container oscillation that is modelled by a sinusoidal osciallation. Since the container is oscillated on springs and we know that springs oscillate with simple harmonic behaviour, a sinusoidal oscillation model is appropriate.

3.2 Measuring Forces As Collision Energies

For a disk-disk collision, the collision energy can be calculated as

$$E_c = \frac{1}{2} m_{\mathrm{eff}} v_{\mathrm{n}}^2, \tag{3.1}$$

where $m_{\mathrm{eff}} = \frac{m_{disk} \cdot m_{disk}}{m_{disk} + m_{disk}} = \frac{1}{2} m_{disk}$ is the effective mass of two disks, and v_{n} is the relative normal velocity between the disks *before* the collision.

Similarly, for a disk-container collision, the collision energy can be calculated as

$$E_c = \frac{1}{2} m_{\mathrm{eff}} v_{\mathrm{n}}^2, \tag{3.2}$$

where $m_{\mathrm{eff}} = \lim_{m_{cont} \to \infty} \frac{m_{disk} \cdot m_{cont}}{m_{disk} + m_{cont}} = m_{disk}$ is the effective mass of a disk and the container (taking the mass of the container to be infinite as compared to the mass of a disk) and v_{n} is the dot product of the velocity of the disk before the collision and a unit vector of the container surface normal $\hat{n}$, that is calculated as

$$v_{\mathrm{n}} = \frac{\left(-x v_x - y \left(v_y - \frac{d\bar{y}}{dt}\right)\right)}{\sqrt{x^2 + y^2}}, \tag{3.3}$$

where x, y, v_x, and v_y specify the position and velocity of the disk when it is in contact with the container just before collision, and $\frac{d\bar{y}}{dt}$ is the velocity of the container.

Independent of how the disks deform in a collision or what the forces are between the disks, the collision energy will be the maximum energy available to crush the medium between the disks. When the disks crush the medium between them, we can calculate this total amount of energy consumed and call this the collision energy. This collision energy is not necessarily equal to but is proportional to the modulus of toughness of the medium that is to be crushed. That is why the collision energy can be compared to the *modulus of toughness* of the material that is to be crushed between the disks. Ferdinand P. Beer and E. Russell Johnston Jr. in their engineering textbook *Mechanics of Materials* [24] define the *modulus of toughness* as a *strain-energy density*, u, taken to the strain at rupture, ε_R using the formula:

$$u = \int_0^{\varepsilon_R} \sigma_x d\varepsilon_x. \tag{3.4}$$

It is basically the area under the stress-strain curve taken up to the point at rupture as shown in Figure 3.1.

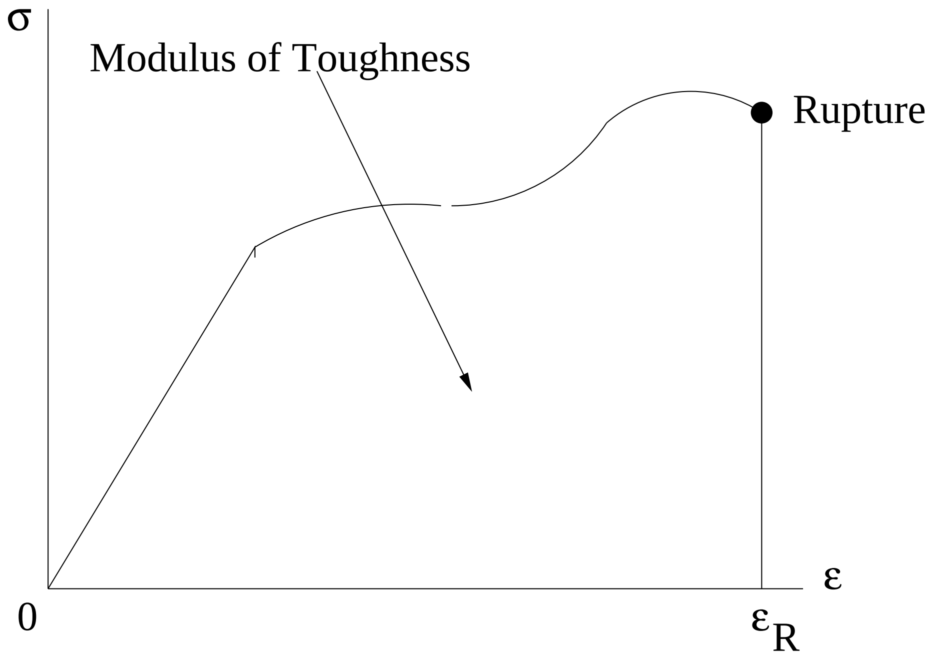


Figure 3.1: Modulus of Toughness [24]

An interesting question to ask is what would be the difference for a material having a greater strain at rupture but with the same area under the stress-strain curve, i.e. with the same modulus of toughness as a material with a smaller strain at rupture? Evidently, the material with the greater strain at rupture would be more elastic and *squashier* than the material with the smaller strain at rupture, but for the purposes of this model one is only concerned with the modulus of toughness, so this would not make a difference. All one cares about is that the collision energy between disks or between a disk and the container wall meets the modulus of toughness value of the material that is to be crushed.

Finally, one can say that by comparing the collision energy between two disks or between a disk and the container wall with the modulus of toughness of the material to be crushed, one can directly model the disks achieving the ultimate stress of the material to be crushed. That is, when the collision energy between two disks or between a disk and the container wall is greater than or equal to the modulus of toughness, one can stipulate that the force between the colliding disks or a disk and the container wall has achieved the ultimate stress of the material that is to be crushed, and thus result in successful grinding.

Chapter 4

RESULTS AND ANALYSIS

4.1 Introduction

This chapter discusses the results of the simulation, namely, the generation of circulation, and macroscopic features such as mixing times, diffusion, energy balances between rods, and between rods and the moving container of the grinding mill. Plots are used to present these features. This provides an analytical tool to study the physical features of the grinding mill. At the beginning of this chapter, it is also shown that the simulation code has been verified against published works and independently written code. The code was also verified for consistency by changing the initial conditions (velocities and positions of the disks) to confirm that the results are obtained from systems for which transients have been removed.

4.2 Verification of Results

According to my literature search, a vertically oscillating circular container has never before been simulated. Thus, in order to verify the correctness of the code, the code was adapted to simulate a geometry that has been extensively studied: a two-dimensional box geometry with a vertically vibrating bottom. Granular flow features characteristic of this geometry include *surface waves and arching* as described in the paper *Vertical Vibration of a Deep Bed of Granular Material in a Container*. [2] The paper describes the surface wave phenomenon as follows:

> Another phenomenon that is observed for deep beds is the formation of surface waves...The waves travel from the lowest point of the heap up the slope to the peak but do not interfere with the continuous avalanche of particles associated with the convection pattern. The waves increase in length and decrease in height as they travel up the slope and eventually disappear at the peak. [2]

The paper describes the arching phenomenon as follows:

> Another phenomenon that has been observed in deep beds is known as “arching.” This behavior consists of sections of the particle bed oscillating out of phase with one another. The boundaries between these regions are known as “nodes”... [2]

See Figures 1.1, 1.2 and 1.3 in Chapter 1. Simulation results were compared between this C++ implementation and an independently written FORTRAN implementation by Dr. Oleh Baran [3]. Computer animations produced from both the C++ and FORTRAN simulations showed surface waves and arching. The FORTRAN simulation was slightly faster than the C++ simulation.

Following are three consecutive frames (snapshots of the animation) of both the C++ and Fortran simulations showing arching and surface waves. See Figures 4.1, 4.3 and 4.5 for three consecutive frames from the FORTRAN implementation, and see Figures 4.2, 4.4 and 4.6 for three consecutive frames from the C++ implementation. The figures from the FORTRAN implementation have surface waves that are spiky as compared to the figures from the C++ implementation. This is because larger initial velocities were chosen for the FORTRAN implementation and there is also a difference due to randomness of the particle movement. Movies of the these FORTRAN and C++ simulations are on the compact disc that accompanies this thesis as described in Appendix F.

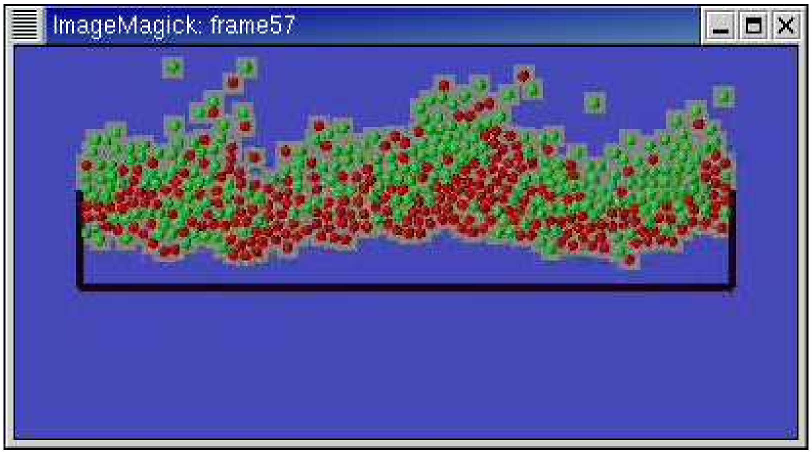


Figure 4.1: FORTRAN Implementation: showing lifting of the granular bed

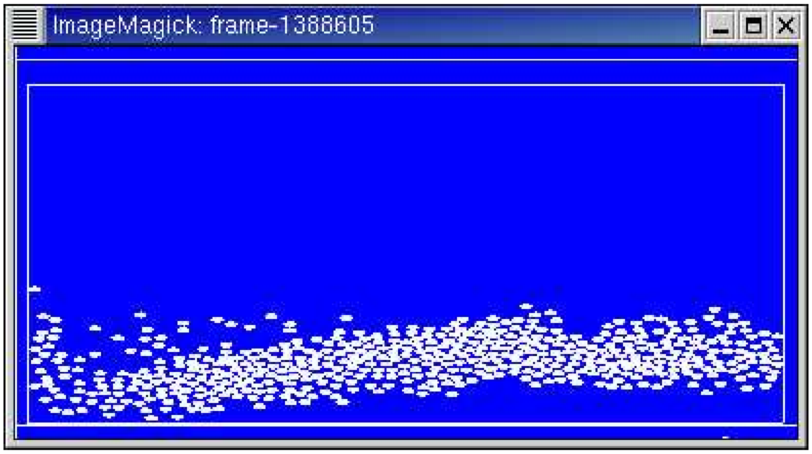


Figure 4.2: C++ Implementation: showing lifting of the granular bed

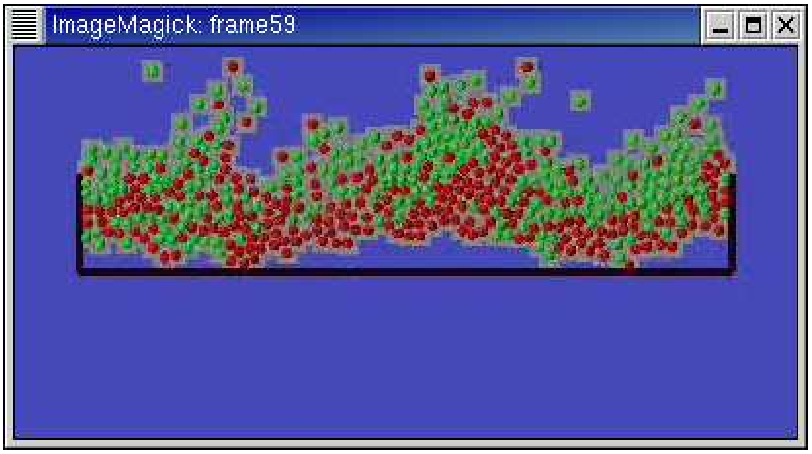


Figure 4.3: FORTRAN Implementation: showing a solitary travelling surface wave and two-node arching

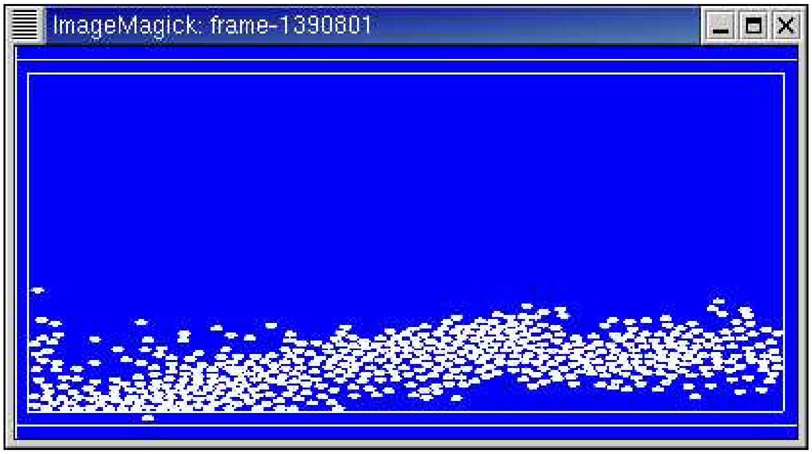


Figure 4.4: C++ Implementation: showing a solitary travelling surface wave and single-node arching

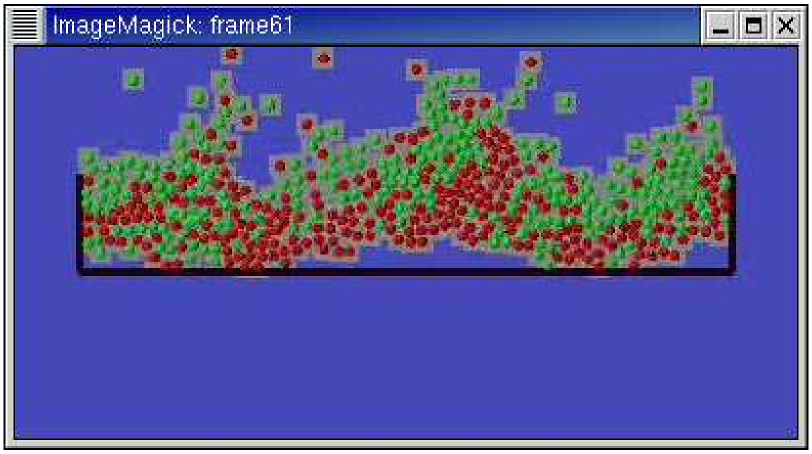


Figure 4.5: FORTRAN Implementation: showing evolution of a solitary travelling surface wave and two-node arching

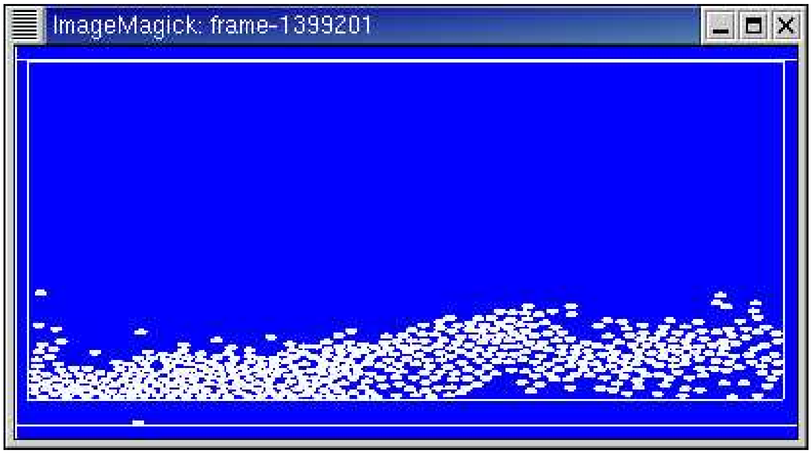


Figure 4.6: C++ Implementation: showing evolution of a solitary travelling surface wave and single-node arching

4.3 Parameters for Simulation

The computer simulation program has various parameters that can be set to control the physics of the simulation. These parameters control the movement of disks before and after a collision which affect circulation patterns and mixing of the disks, clustering and the avalanche movement of the disks, and the occurrence of force chains developing along a line of colliding disks. These all affect the distribution of collision energies between two colliding disks and between a disk colliding with the moving container.

Certain fixed parameters were used in all the simulation runs:
g (acceleration due to gravity): 981 cm/s^2
σ (disk diameter): 2 cm
ϕ (container diameter): 48 cm

The above disk diameter and container diameter are based on actual dimensions that are used in current physical experiments.

Using the above fixed parameters the code was first run through the following coarse test matrix of varying parameters:
ω_y (frequency of vertical oscillation in radians/second): 10, 126, 150, 500
A_y (amplitude of vertical oscillation in cm): 0.5, 0.75, 1, 1.5, 2
e_0 (restitution coefficient for a disk-disk collision): 0.1, 0.4, 0.5, 0.6
e_W (restitution coefficient for a disk-container collision): 0.4, 0.7, 0.9, 1.0

The frequency, ω_y, and amplitude of the container oscillation, A_y, were based on an actual frequency and amplitude that was used in current physical experiments. The coefficient of restitution between the steel disks was based on physical experiments of steel balls hitting a steel plate as shown in Figure 4.7.

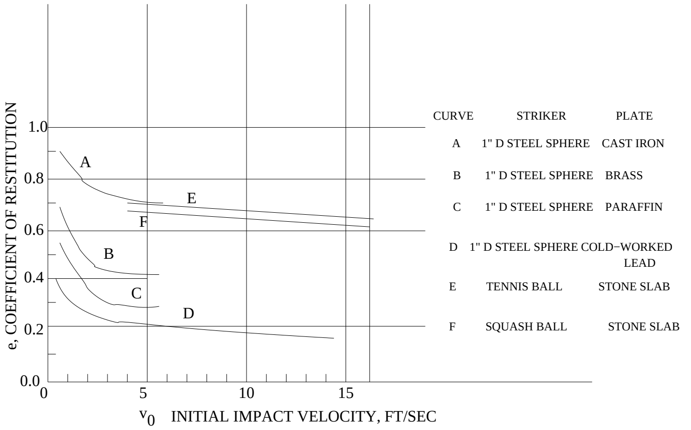


Figure 4.7: Velocity-Dependent Restitution Coefficient [13]

The program was run using the above parameters and a simulation that produced a realistic motion was selected (ω_y = 126 rad/s = 20 Hz, A_y = 1.5 cm, e_0 = 0.4, e_W = 1.0). By a realistic motion, we mean that the disks would cluster together at the bottom of the container within a relatively short period of time with few overlaps.

4.4 Steady State

Using data from this simulation, the first task was to determine the time when the simulation reached a condition of *steady state*, that is, after the transient portion of the simulation. This was important as physical data was only meaningful after reaching a steady state. A steady state was visually observed in animations of the simulation when the disks would cluster near the bottom of the circular container, but in order to get a number for the time when steady state was reached, the total kinetic energy of the system of disks was studied over time. First plots were made of the *normalized* total kinetic energy (KE/M) versus time (t). In order to achieve results that are independent of mass, a mass of 1 kg was used for the mass of each disk in the calculations so in effect the kinetic energy was normalized by the mass. This way the results are general and can be used with the disks of any mass. All one has to do is multiply the normalized kinetic energy by the mass of the disk to get the actual kinetic energy. Figures 4.8 and 4.9 show typical plots of the total kinetic energy versus time for 100,000 collisions and 1 million collisions, respectively.

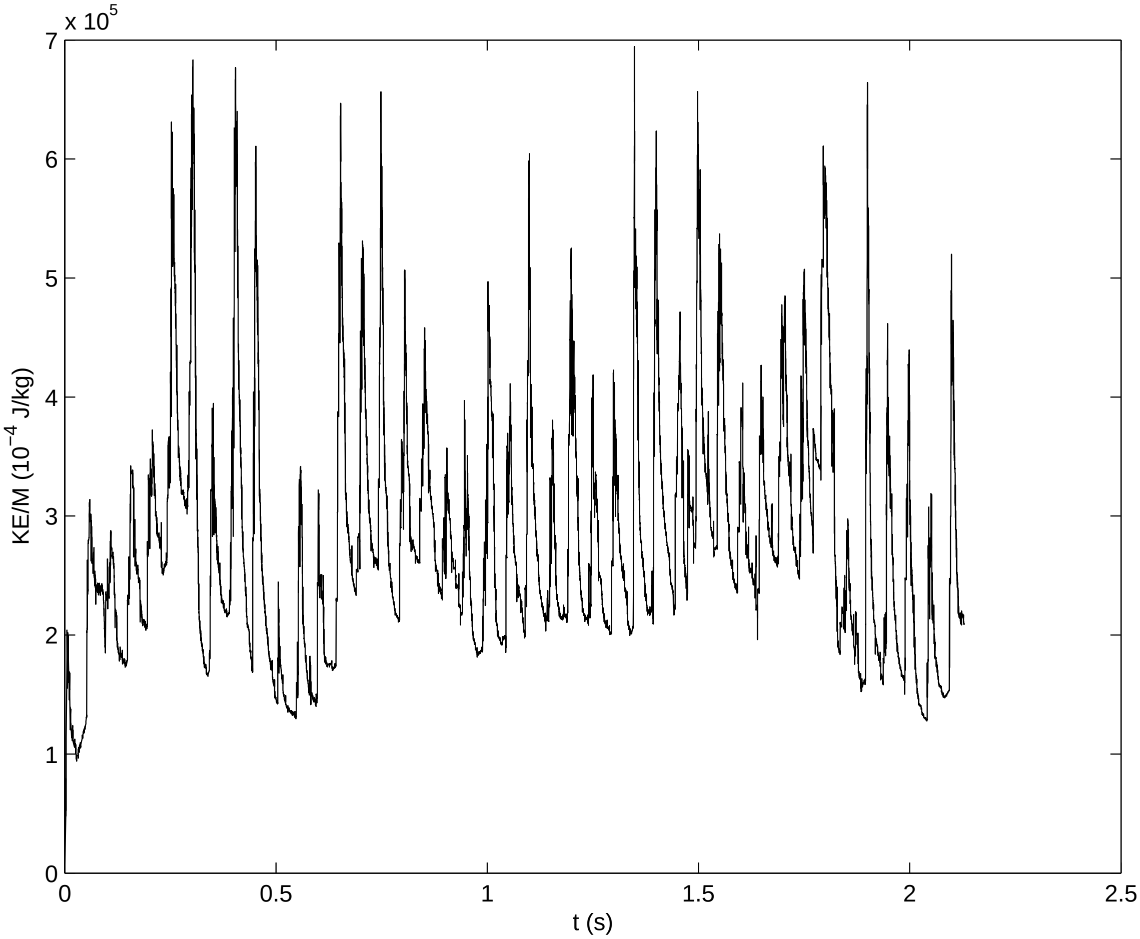


Figure 4.8: Normalized Total Kinetic Energy (KE/M) versus time (t) for 100,000 collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

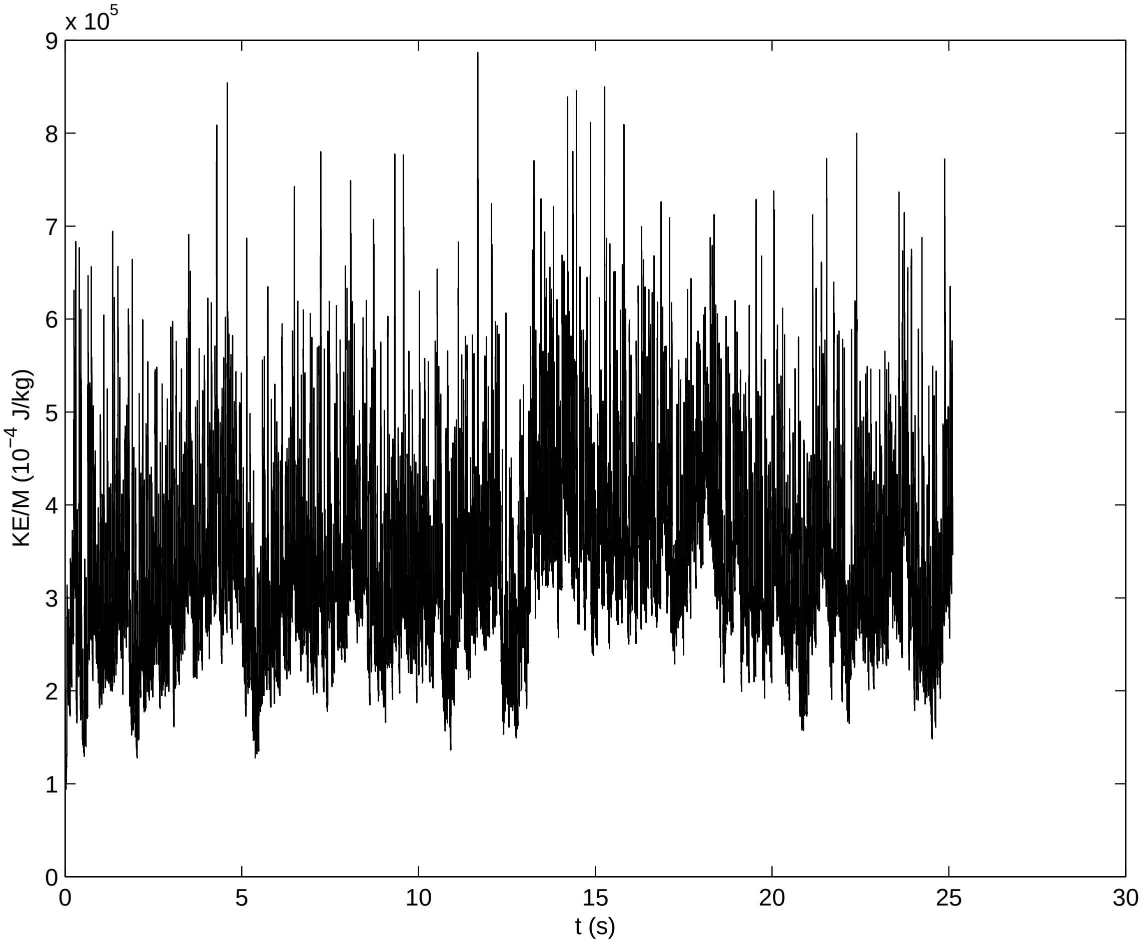


Figure 4.9: Normalized Total Kinetic Energy (KE/M) versus time (t) for 1 million collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

Since the container oscillation was pumping energy into the system and the disks were dissipating energy at each collision, from Figures 4.8 and 4.9, one can see that the steady-state was reached when the total energy would start to oscillate about a constant (greater than 0) value. This is at about 1 second. This steady state time of 1 second was typical of all the simulations. As can be seen from these plots, there is an underlying long wavelength which is due to the wavelength of the container oscillation and the wavelength of the oscillation of the cluster of disks periodically colliding with the bottom of the container. Thus an FFT power spectrum[1] was produced to filter out the random fluctuations about a constant mean and determine the characteristic frequencies. Since the data was spread over very tiny time intervals (of the order of 10^{-5} which was 100 times smaller than the time interval of 10^{-3} for the fourier transform), the error in the spectrum resulting from this sample and hold technique should be minimal. An analysis of the effect of the sample and hold technique on a signal spectrum is given in Appendix B. See Figure 4.10.

---

[1] Uniformly spaced data is required as input for the fft algorithm. So, since the data was spaced over unequal collision times and thus was not uniformly spaced over time, a *sample and hold* technique was used where the data closest to the left of a uniformly spaced time point was taken as the data at that time point.

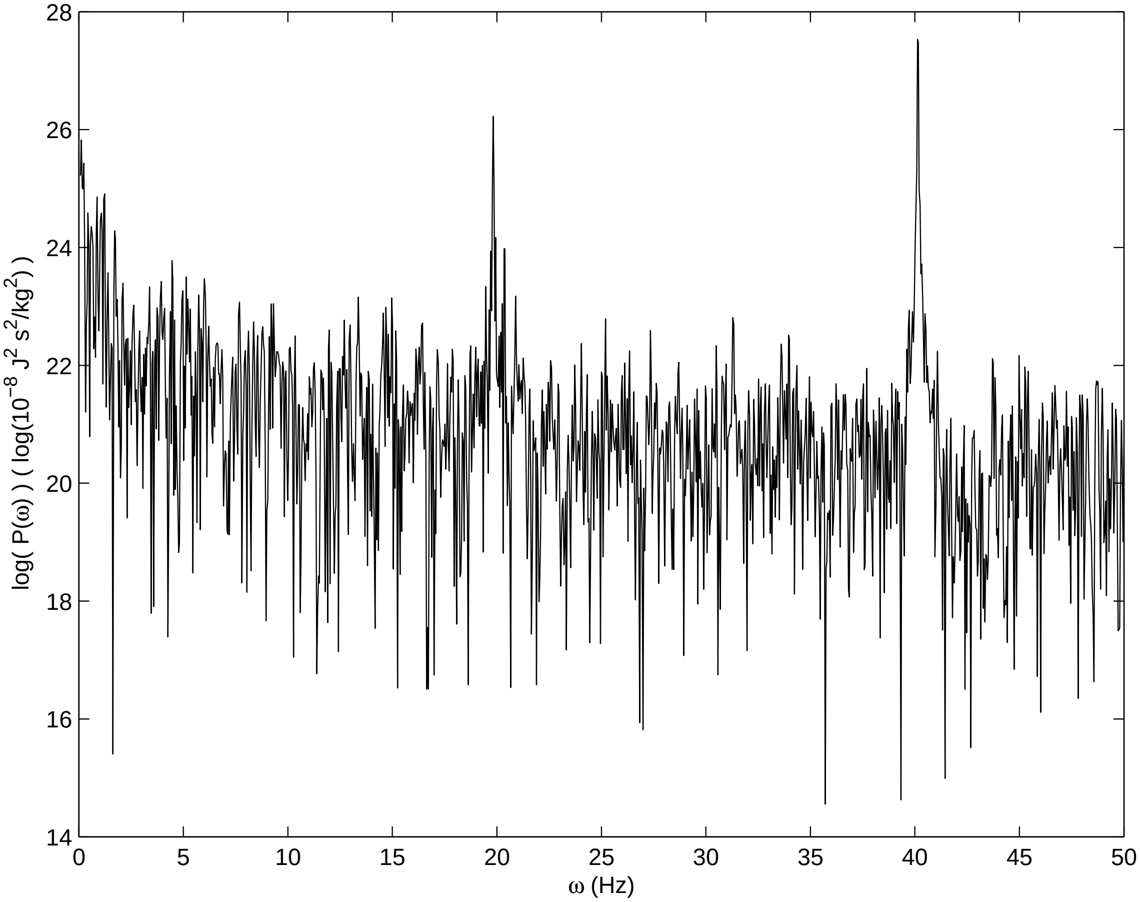


Figure 4.10: Power Spectrum of Normalized Total Kinetic Energy ($\mathcal{P}(\omega)$) for 1 million collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

As can be seen from Figure 4.10, there are two characteristic frequencies, one at 20 Hz which is the container oscillation frequency ($\omega = 20$ Hz $= 126$ rad/s) and another at 40 Hz which is twice the container oscillation frequency. The reason that twice the oscillation frequency is present is that the disks colliding with the oscillating container acquire a velocity of the form $v = v_o \cos(\omega t)$ and therefore the kinetic energy is

$$\begin{aligned} KE &= \frac{1}{2}mv^2 \\ &= \frac{1}{2}v_o^2\cos^2(\omega t) \\ &= \frac{1}{2}\left[v_o^2\left(\frac{1}{2}\left(1+\cos(2\omega t)\right)\right)\right] \end{aligned} \tag{4.1}$$

Hence, there is a term with twice the frequency (2ω) in the kinetic energy.

4.5 Mixing

Mean square displacement plots (from which mixing times to see how well the disks mix, and diffusion coefficients to see how well the disks diffuse, can be obtained) were made based on a constant time origin (the steady state time) and moving time origins as shown in Figures 4.11 and 4.12. Both the plot for 100,000 collisions and the plot for 1 million collisions are from a simulation using the same values for the parameters ($\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$ and $e_W = 1.0$). Mean square displacements based on a fixed time origin, t_0, were calculated using Equation (4.2). Mean square square displacements based on a moving time origin, t', were calculated using Equation (4.3)

$$< r^2 >_{t-t_0} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(\vec{r}_i(t) - \bar{y}(t)\right) - \left(\vec{r}_i(t_0) - \bar{y}(t_0)\right)\right] \tag{4.2}$$

$$< r^2 >_{t'-t} = \frac{1}{N_t}\sum_{t}\frac{1}{N}\sum_{i=1}^{N}\left[\left(\vec{r}_i(t') - \bar{y}(t')\right) - \left(\vec{r}_i(t) - \bar{y}(t)\right)\right] \tag{4.3}$$

where N is the number of disks, N_t is the number of time steps, r_i is the position of the ith disk, and $\bar{y}$ is the vertical displacement of the container.

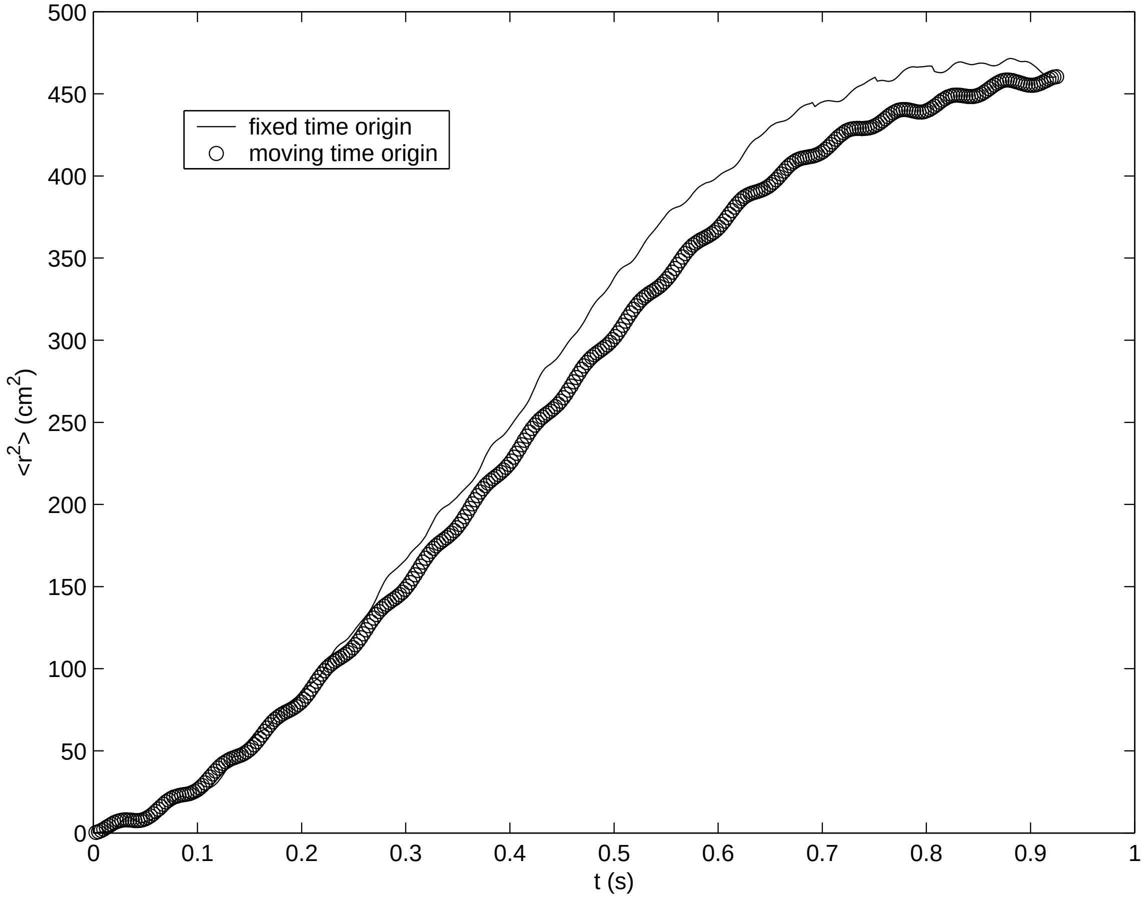


Figure 4.11: Mean Square Displacement ($< r^2 >$) versus time (t) for 100,000 collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

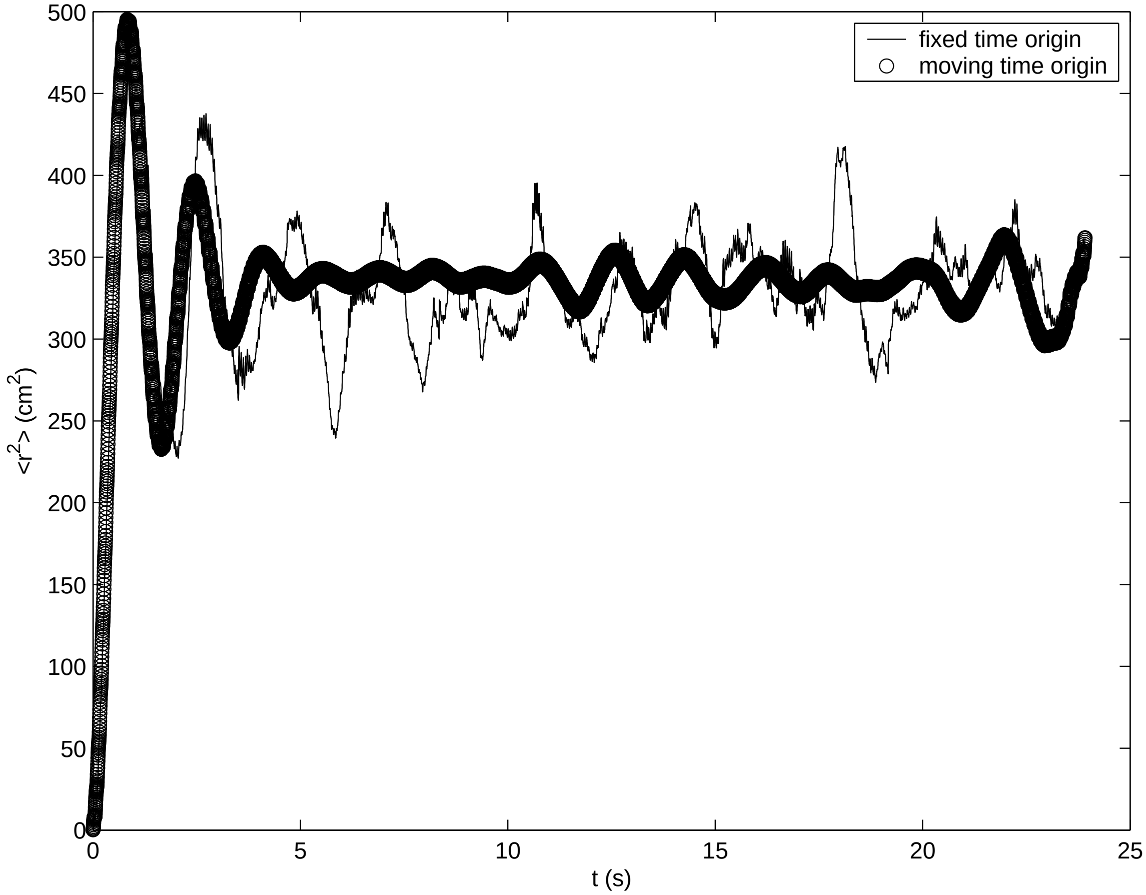


Figure 4.12: Mean Square Displacement ($< r^2 >$) versus time (t) for 1 million collisions: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

The more informative mean square displacement (MSD) plot is the one for 1 million collisions. The average disk moves across the bed fairly quickly, and after that the MSD saturates at the value of r^2 characteristic of the size of the bed. As can be seen from Figure 4.12, a reasonable definition of the mixing time is the time it takes for the MSD to reach the *saturated* MSD (at about $r^2 = 330$) for the first time. This is more direct, easier, and doesn't make any assumptions that the diffusive behavior does or does not exist. If however, one wanted to calculate the diffusion constant to quantify how the disks diffuse in the container, one would use Equation (4.4) as a definition.

$$D = \frac{1}{2d} \lim_{t \to \infty} \frac{< r^2 >}{t} = \frac{1}{2d} \times \text{ slope of straight line portion} \tag{4.4}$$

where $d = 2$ is the dimension of the simulation. Because the disks are constrained by the container, the mixing time is a more useful parameter than the diffusion coefficient.

4.6 Phase Diagrams

Using the coarse test matrix, simulation runs were characterized as those that didn't suffer from *inelastic collapse* before 5 seconds (which is well past the 1 second steady state time) and those that gave circulation (a reasonable circulation value was taken as $\Gamma = 12,000$ cm^2/s which was characteristic of the animation that gave realistic results for $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$ and $e_W = 1.0$). By realistic results, we mean that the disks would cluster together at the bottom of the container within a relatively short period of time with few overlaps. In order to optimize the grinding, realistic looking simulations are required that have a reasonable value for circulation. A reasonable circulation value such as $\Gamma = 12,000$ cm^2/s aids in the mixing and thus improves the efficiency of the grinding process. Finding inelastic collapse before 5 seconds is a signature of a non-circulating system. See Section 2.5 for a definition of inelastic collapse.

Phase diagrams of A_y versus ω_y showing phases where the simulation lasted at least 5 seconds before suffering from inelastic collapse and for which the net circulation Γ was above 12,000 cm^2/s were produced for varying restitution coefficients, e_0 and e_W. Figure 4.13 shows a phase diagram for the case of $A_y = 1.5$ cm, $e_0 = 0.4$ and $e_W = 1.0$.

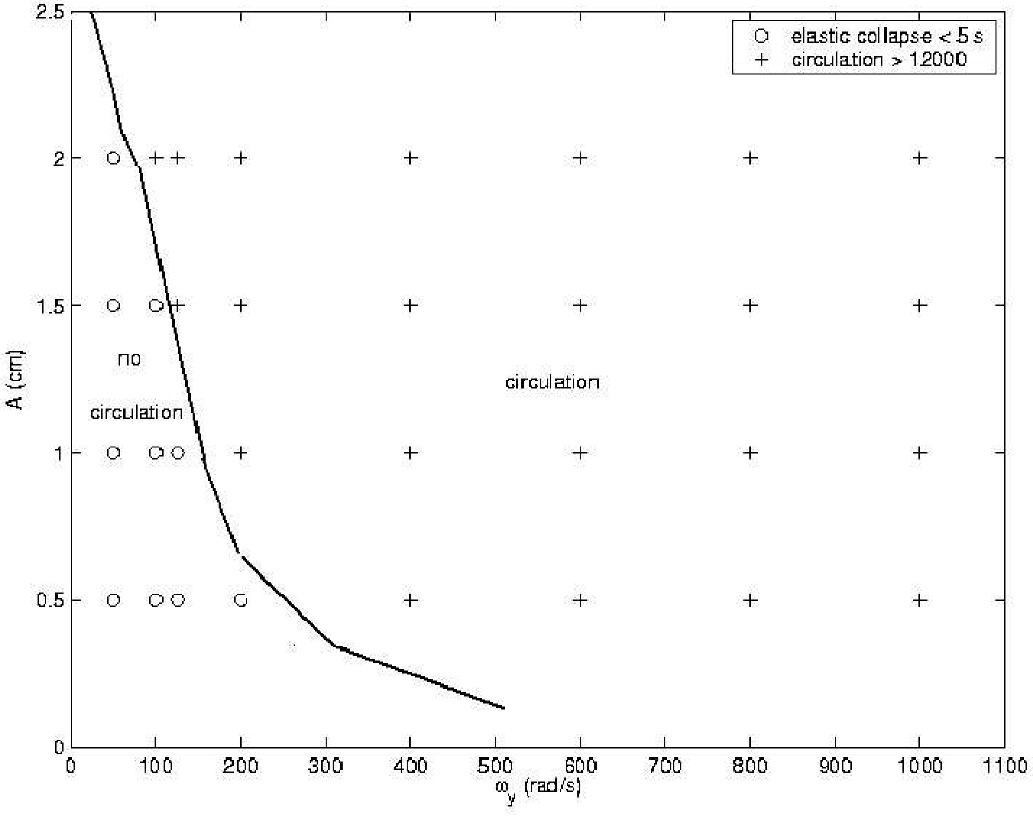


Figure 4.13: Phase Diagram: Amplitude (A) vs Frequency (ω_y): $e_0 = 0.4$, $e_W = 1.0$

In all observed simulations there was mainly one large vortex (see Figure 2.8) , but if future simulations result in more than one vortex, then phase diagrams could also be based on varying modes of circulation. The snapshot of the simulation shown in Figure 2.7 and the velocity field shown in Figure 2.8 reveal that the largest velocities are where the disks come in contact with the container and also where they are free to move at the top layer of the cluster of disks. It is to be noted that for the course test matrix, all the simulations for $\omega_y = 10$ rad/s suffered from inelastic collapse too soon (before 1 second). (The data for $\omega_y = 10$ rad/s is not plotted on the phase diagram because the scale for ω_y is in units of 100 rad/s.)

4.7 Circulation

To study the circulation dependence which characterizes the movement of the disks, a plot of the average net circulation Γ vs oscillation frequency ω_y was made (see Figure 4.14). This plot shows a linear dependence which is expected from the defining equation of Γ given in equation 2.19. A plot of the nondimensional parameter $\Gamma/(A_y^2\omega_y)$ vs ω_y was also produced (see Figure 4.15). Here, A_y is the amplitude of the container oscillation. Figure 4.15 shows a levelling off after a certain frequency, which indicates a change in circulatory movement of the disks at this levelling off frequency (more freely uniform circulation). See the corresponding velocity field figures 4.16, 4.17 and 4.18 for simulations with an amplitude of 1 cm, and figures 4.19, 4.20, 4.21 and 4.22 for simulations with an amplitude of 1.5 cm. The same circulation behaviour was found for the points on Figure 4.15 for amplitudes of container oscillation of 0.5 cm and 2 cm. Snapshots of the simulations that had these time averaged freely uniformly circulating velocity vector field diagrams (what we term the circulating *granular gas* phase) are given in Figures 4.23, 4.24, 4.25 and 4.26. By a *granular gas phase* we mean that the disks are sparsely spaced throughout the interior space of the circular container. From these snapshots, we see that the disks first do cluster but the amplitude (A_y, ω_y) combination is large enough to cause the cluster of disks to bounce against the container so much that the cluster begins to circulate and later dissipate into a circulating gas of disks. This situation is for points along the horizontal portions of Figure 4.15. For points along the sloped portion, the (A_y, ω_y) combination is small enough so that the cluster does not dissipate into a gas but remains a cluster for the duration of the simulation. From these figures, one can conclude that the desired clustering phenomenon is observed for points in Figure 4.15 that are along the slope and not at the horizontally located points where the cluster dissipates into a free uniform circulating granular gas. More information on granular gases can be found in the book entitled *Granular Gases* edited by Thorsten Pöschel and Stefan Luding [25]. See Figure 4.27 for a typical plot of the net circulation vs time which shows a clear non-zero average net circulation Γ of 12,000 cm^2/s.

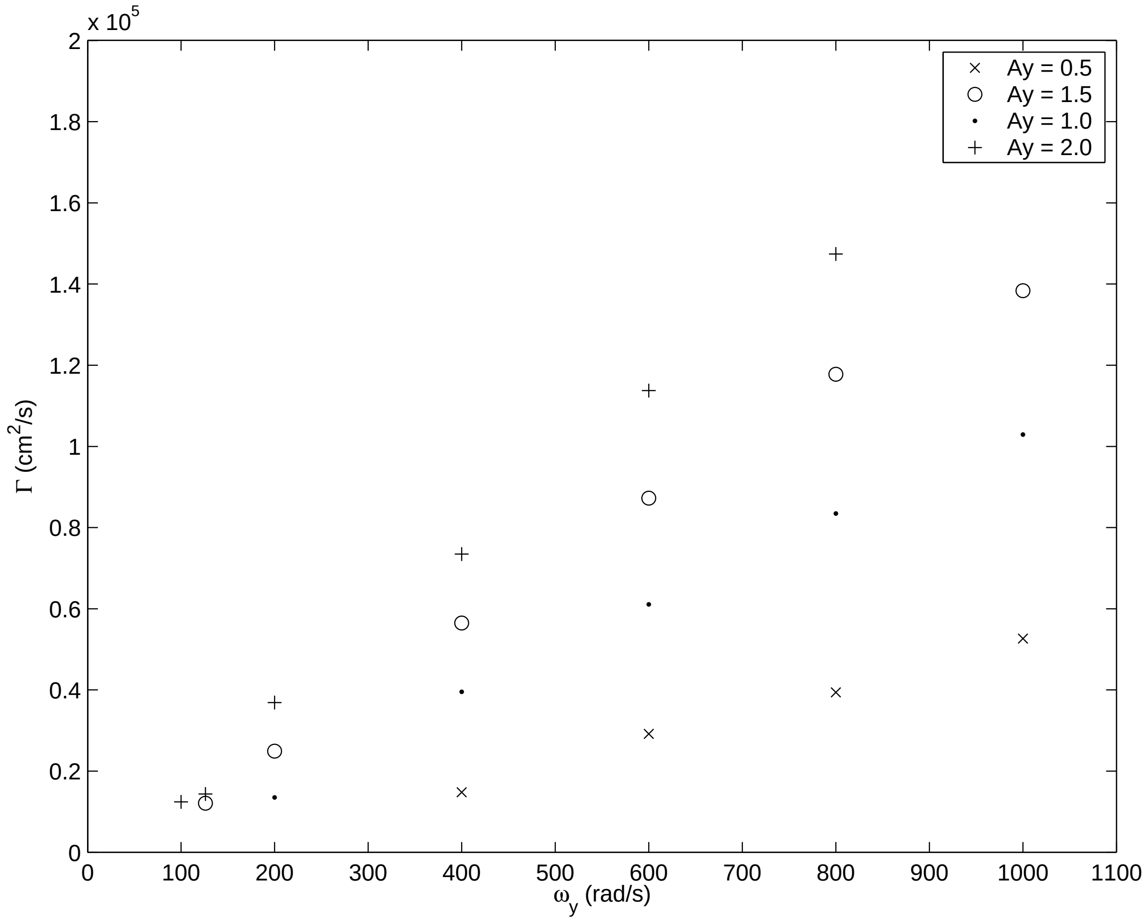


Figure 4.14: Time Averaged Net Circulation (Γ) vs Frequency of Container Oscillation (ω_y): $e_0 = 0.4$, $e_W = 1.0$

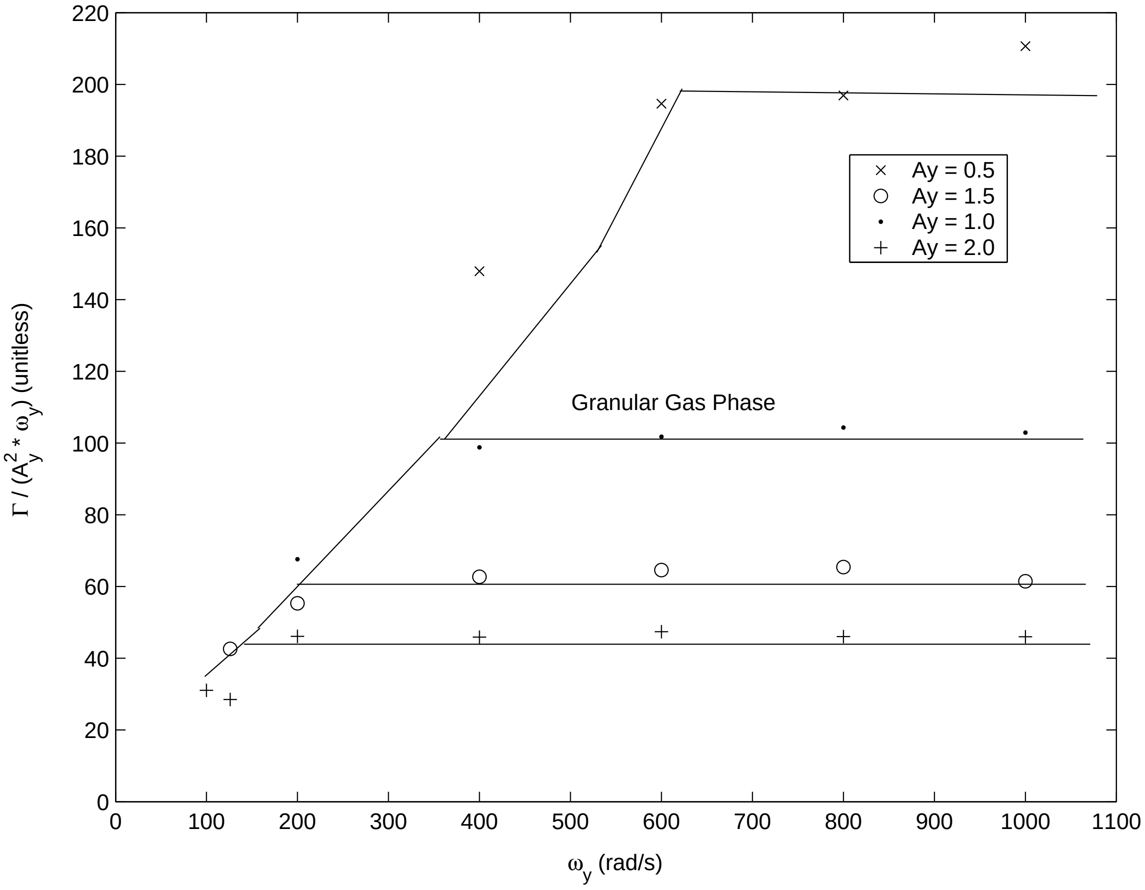


Figure 4.15: Non-dimensional Parameter ($\Gamma/(A_y^2\omega_y)$) vs Frequency of Container Oscillation (ω_y): $e_0 = 0.4$, $e_W = 1.0$

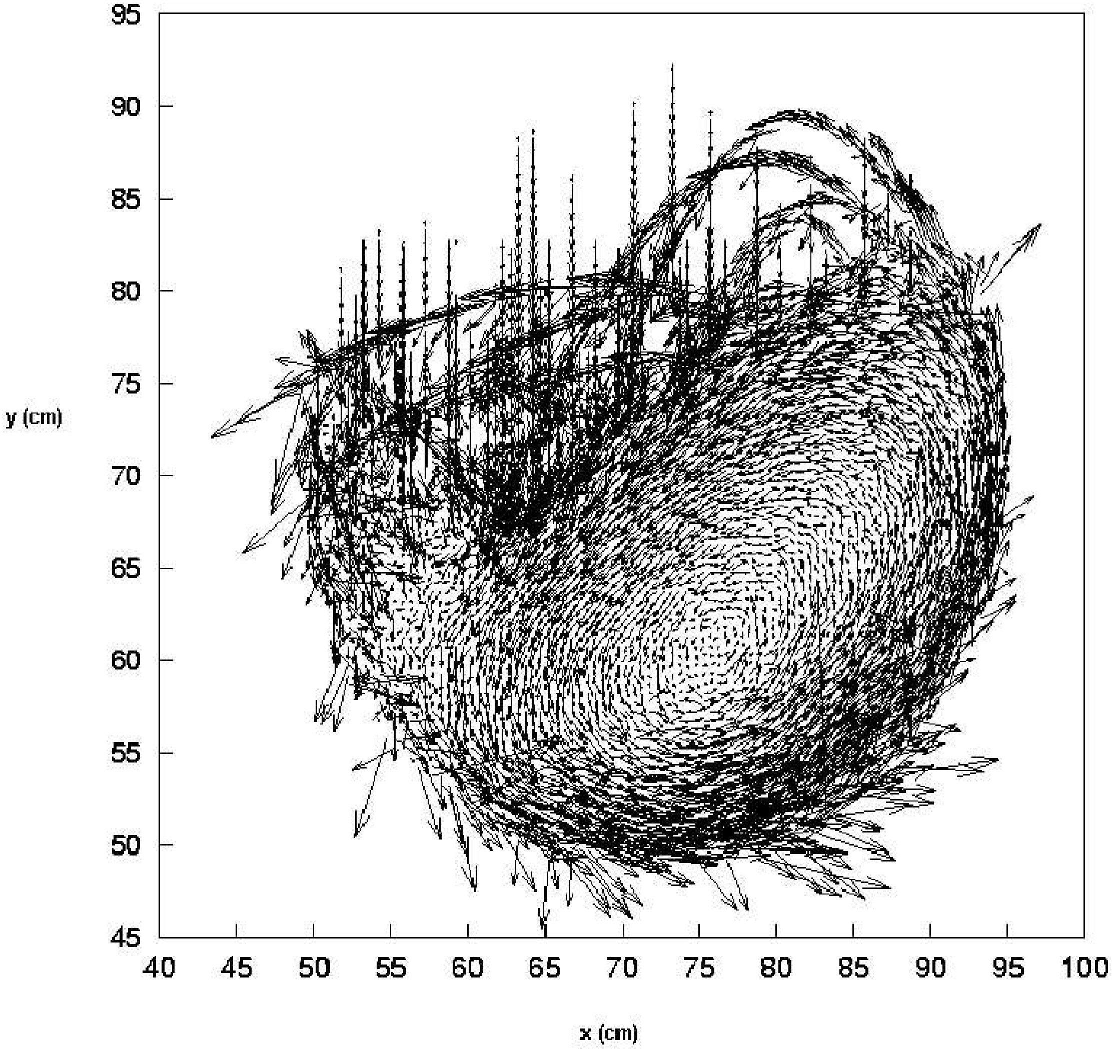


Figure 4.16: Time Averaged Velocity Field for Amplitude $A_y = 1.0$ cm and Frequency $\omega_y = 200$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

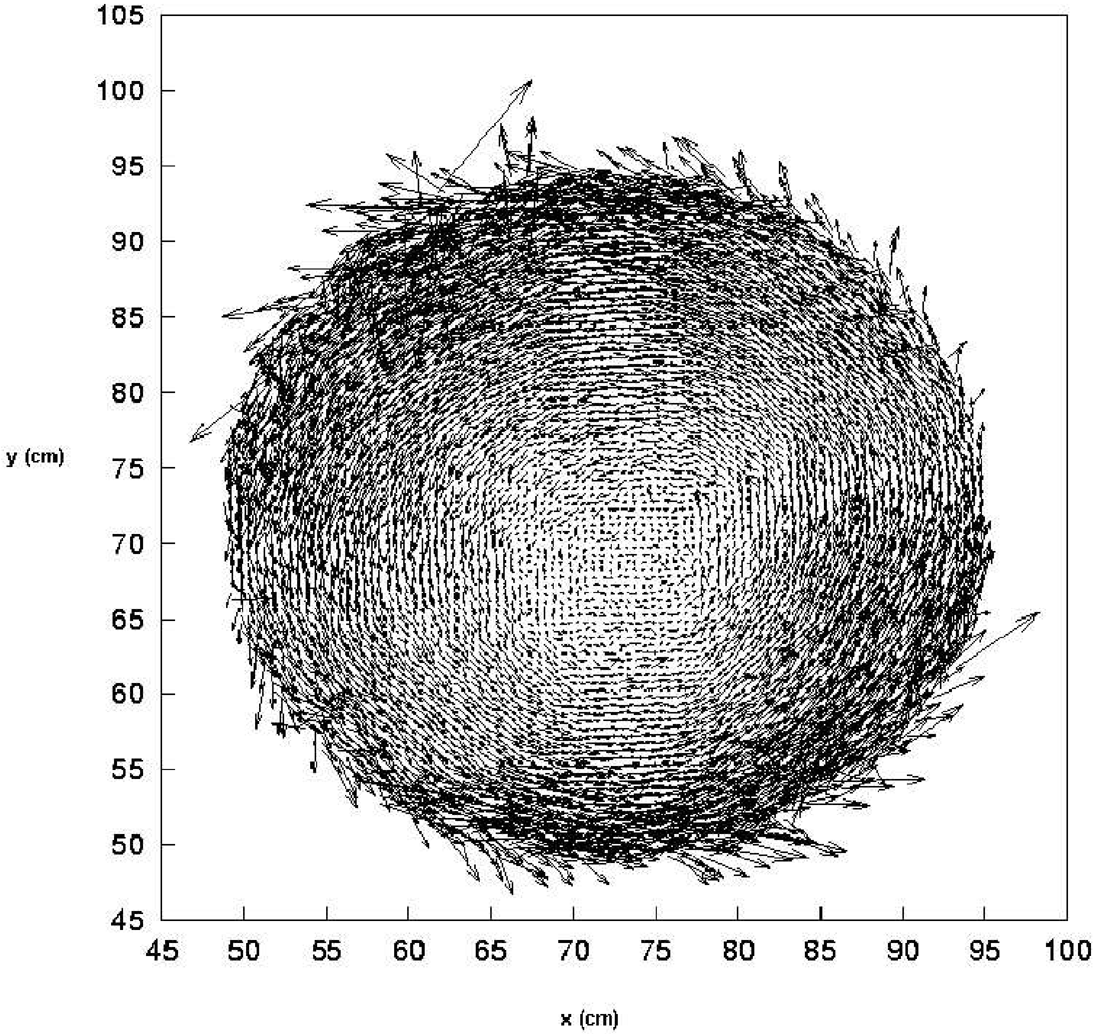


Figure 4.17: Time Averaged Velocity Field for Amplitude $A_y = 1.0$ cm and Frequency $\omega_y = 400$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

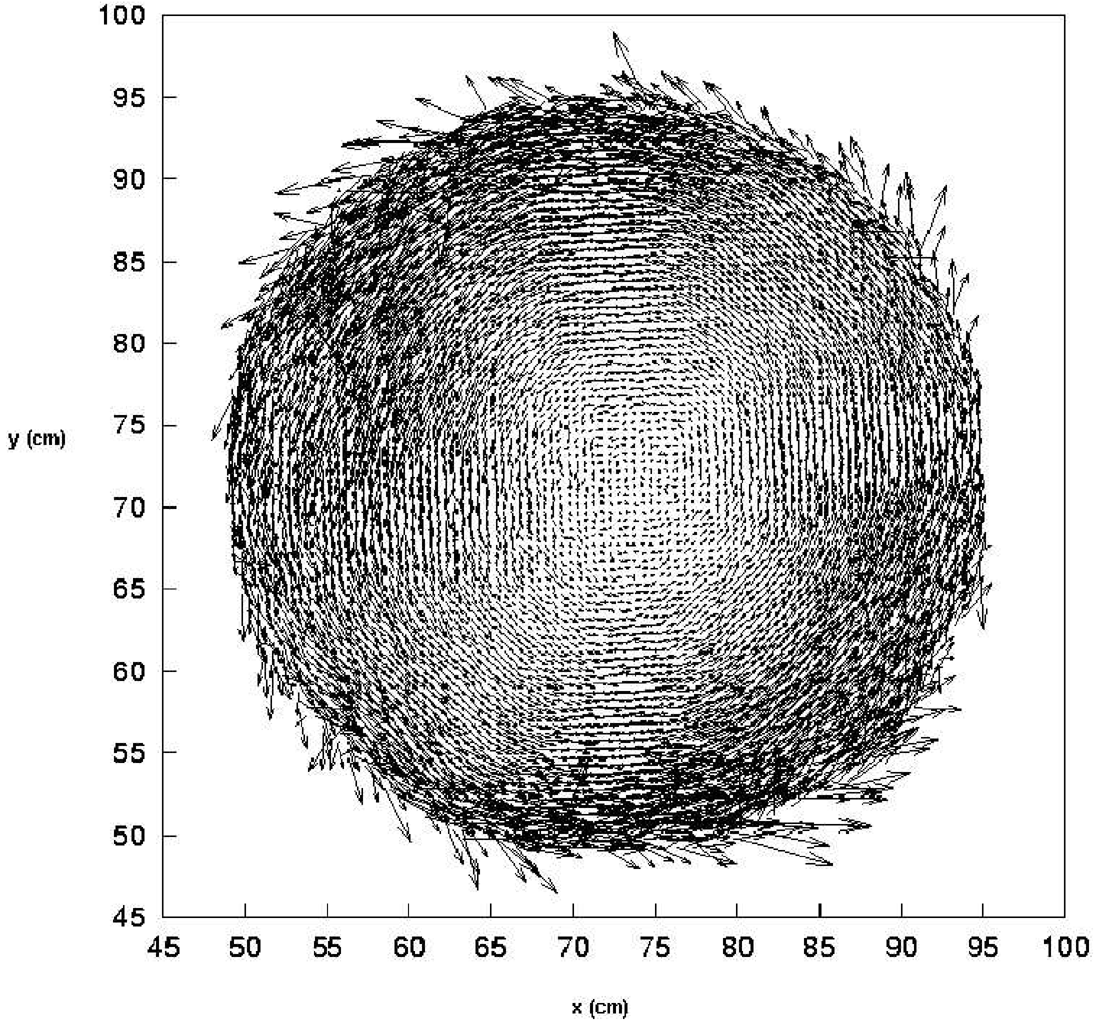


Figure 4.18: Time Averaged Velocity Field for Amplitude $A_y = 1.0$ cm and Frequency $\omega_y = 600$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

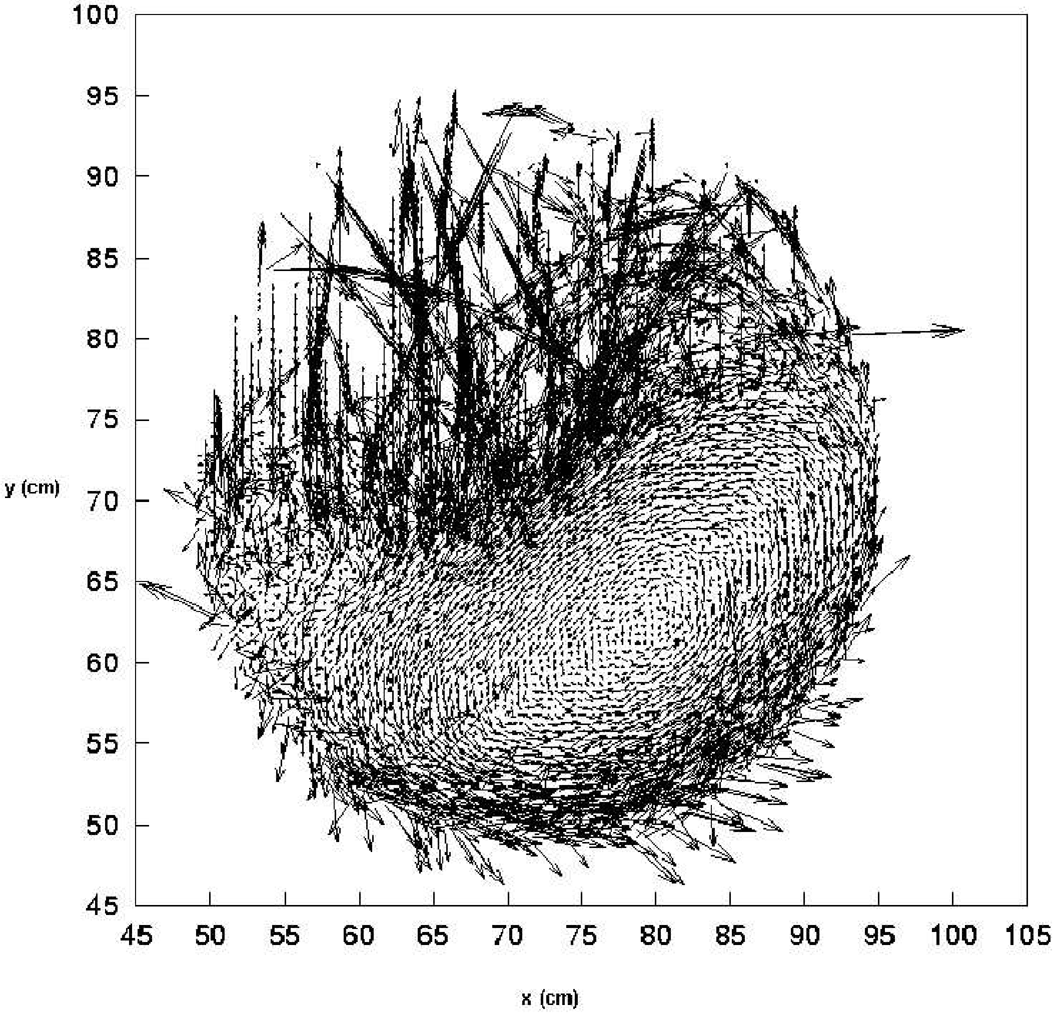


Figure 4.19: Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 126$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

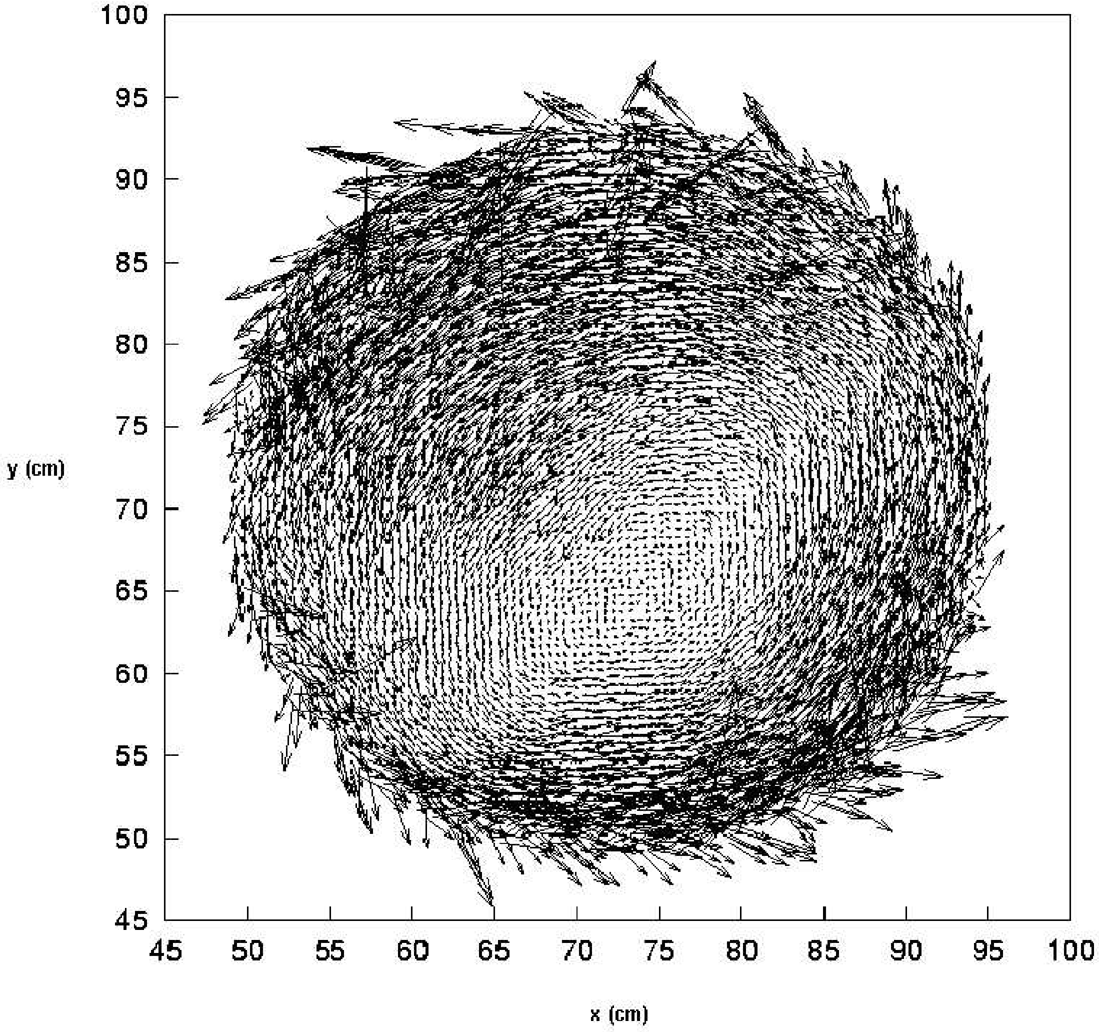


Figure 4.20: Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 200$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

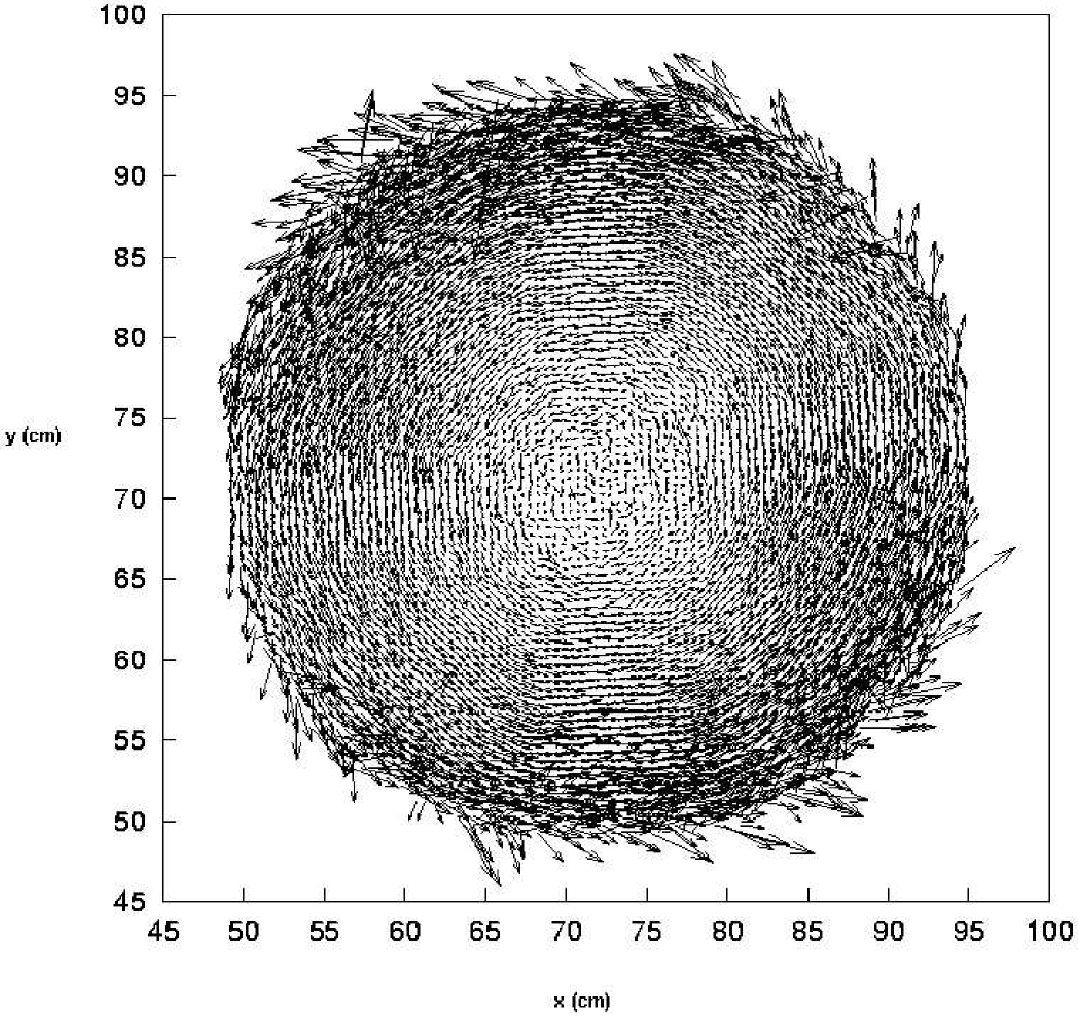


Figure 4.21: Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 400$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

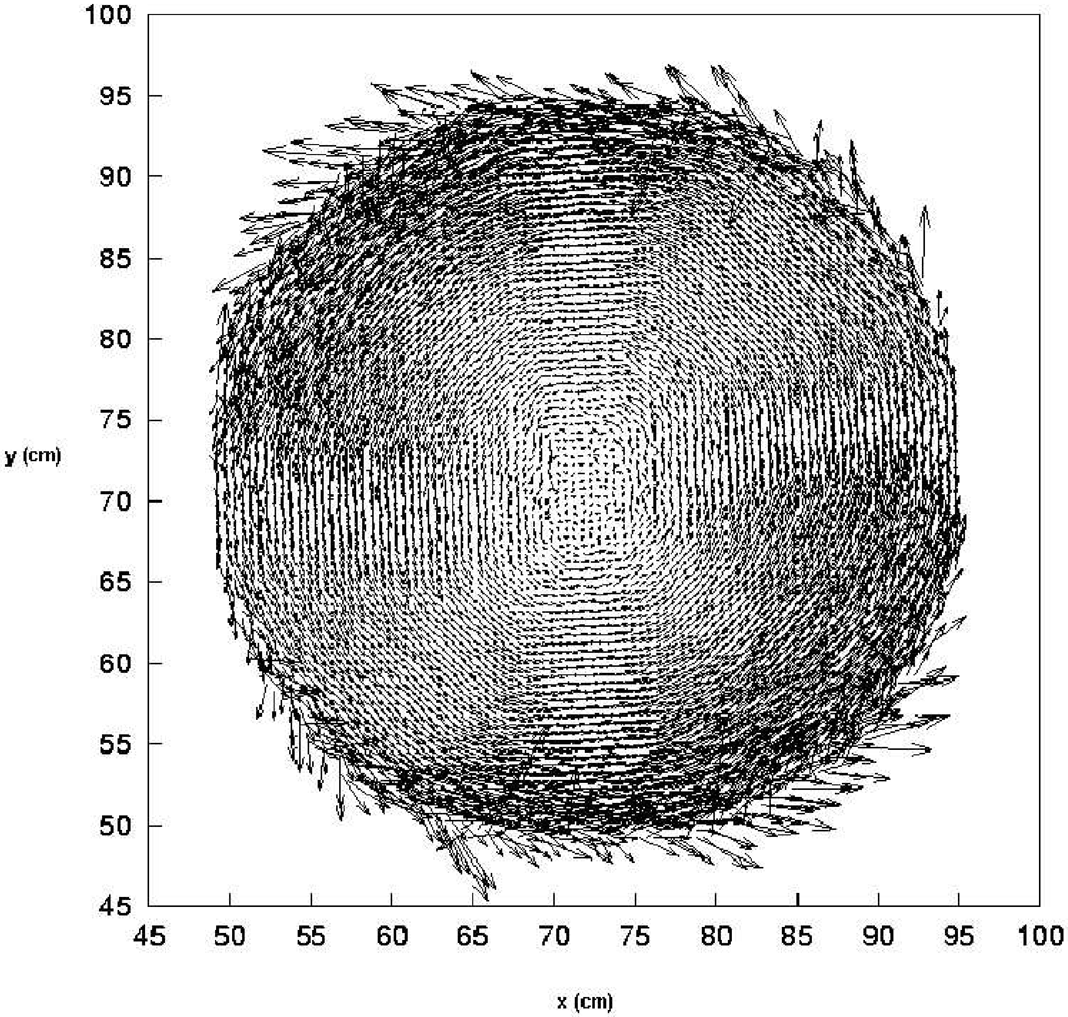


Figure 4.22: Time Averaged Velocity Field for Amplitude $A_y = 1.5$ cm and Frequency $\omega_y = 600$ rad/s: $e_0 = 0.4$, $e_W = 1.0$

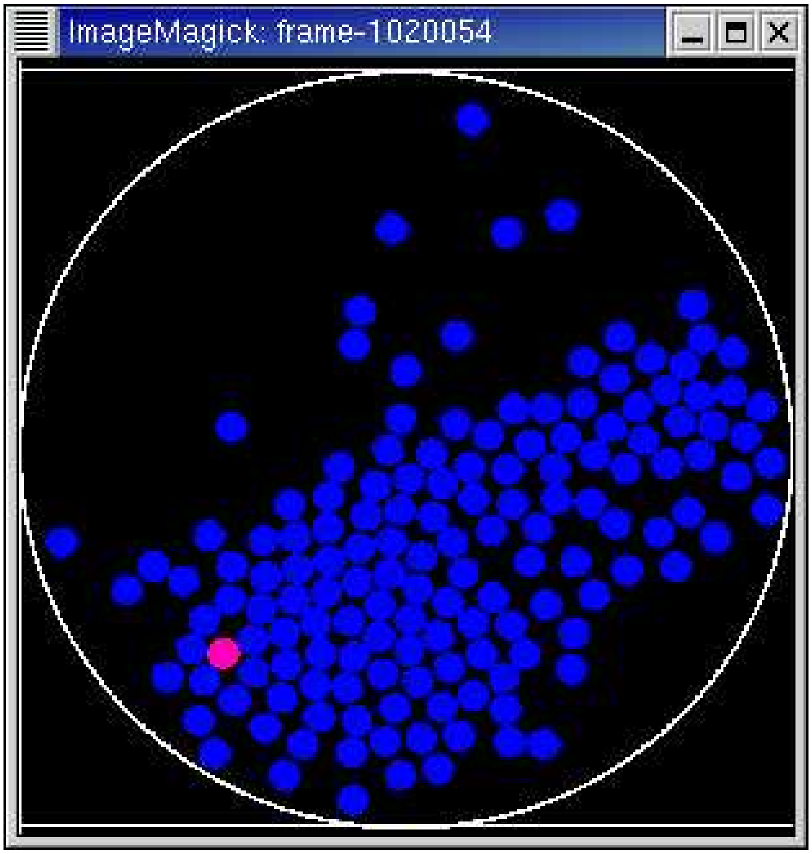


Figure 4.23: Snapshot Frame 1 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disk is coloured in pink.

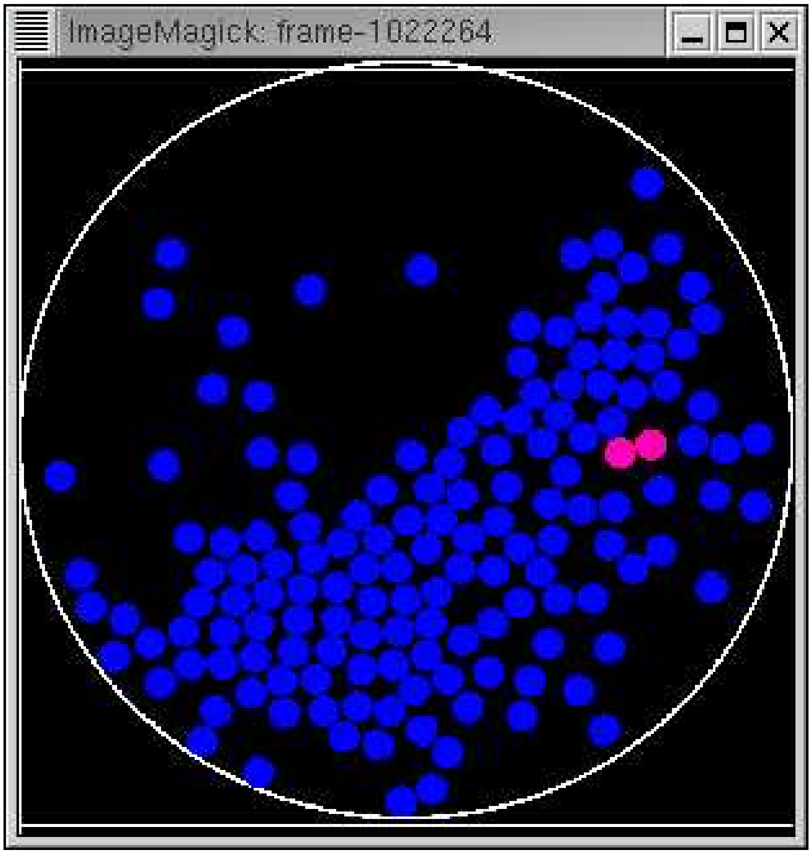


Figure 4.24: Snapshot Frame 2 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink.

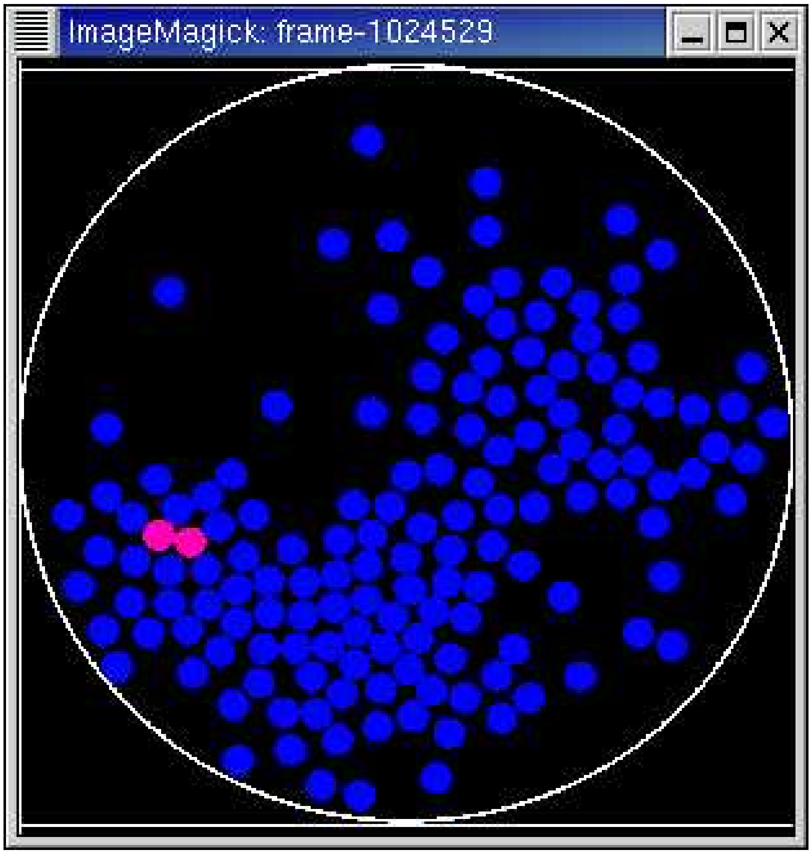


Figure 4.25: Snapshot Frame 3 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink.

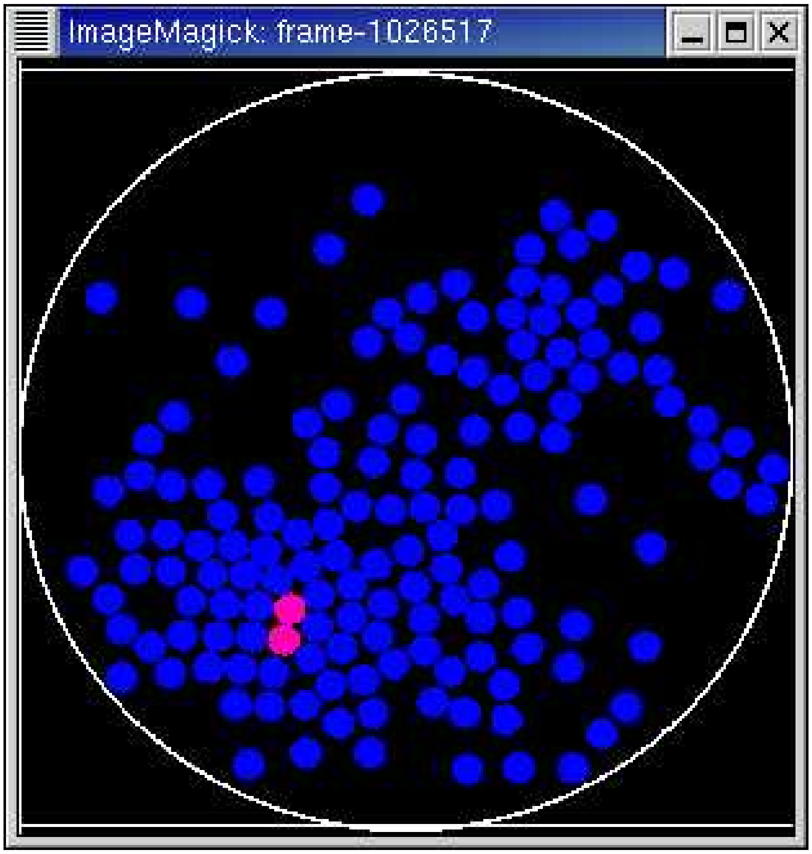


Figure 4.26: Snapshot Frame 4 Showing Evolution of Granular Gas: $\omega_y = 600$ rad/s, $A_y = 0.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink.

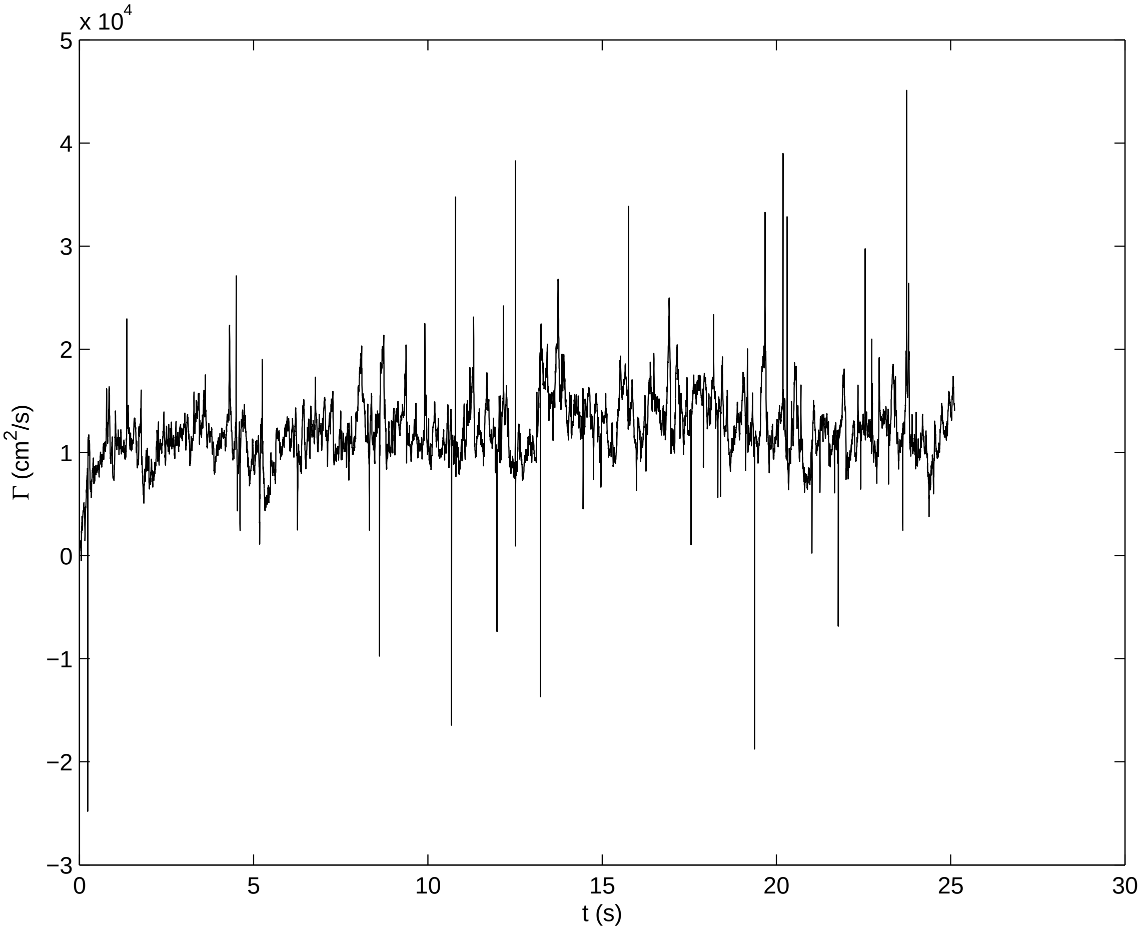


Figure 4.27: Net Circulation (Γ) vs Time (t): $\omega_y = 126$ rad/s, $A_y =$ 1.5 cm, $e_0 = 0.4$, $e_W = 1.0$

4.8 Forces Analysis

The forces in terms of collision energies were analyzed for runs that were in a phase of reasonable circulation (net circulation Γ greater than 12,000 cm^2/s). Contour plots[2] of collision energies within the container (see Figure 4.28) and plots showing the distribution of collision energies versus the angle θ along the container wall, the number of hits versus θ along the wall, and the average collision energy of the hits versus θ along the wall (see Figures 4.30, 4.31 and 4.32) were produced. θ is along the wall of container as in a unit circle. See Figure 4.29. In all three figures, it is sensible that there were two high collision energy peaks in both frequency and amplitude between π and 2π which is located at the bottom of the container. This is because the disks would collide at the left and right bottom arcs of the container as shown in Figure 2.7. There was also a peak between 0 and π which is a result of the cluster of disks *riding up* and hitting the right side of the container. This is also shown in Figure 2.7. This unexpected double peak articulates where are the mechanical stresses on the container. These impact stresses could be significant for the design of the container. Note that this double peak in the bar charts of the disk-container collision energies is consistent with the gap at the bottom middle in the contour plot of the disk-disk collision energies. That is, where there are high disk-disk collision energies, there are high disk-container collision energies. This indicates that the disks bouncing off the container transmit the energy to the layer of disks above. As far as the gap is concerned, keep in mind that these are *time averaged* disk-disk collision energies, so at this point there are fewer disk collisions and thus the time averaged collision energies at this location will be smaller. The high values at the bottom right side of the contour plot could possibly be explained by the motion of the disks riding up the right side of the container.

---

[2] Grid spaced data is required by the contour plot algorithm of *gnuplot* and *Matlab*. So, since the collision energies did not occur on a grid, *Shepard interpolation* [26] was used to interpolate the collision energies onto a grid before plotting the contour. See the appendix for an explanation of *Shepard interpolation*.

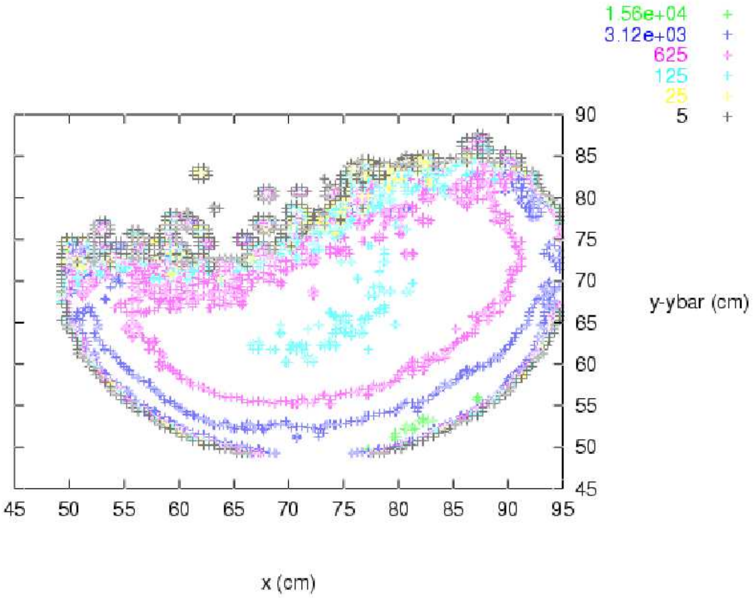


Figure 4.28: Contour Plot of Time Averaged Disk Disk Collision Energies (CE): $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The contour levels are in units of 10^{-4} J/kg.

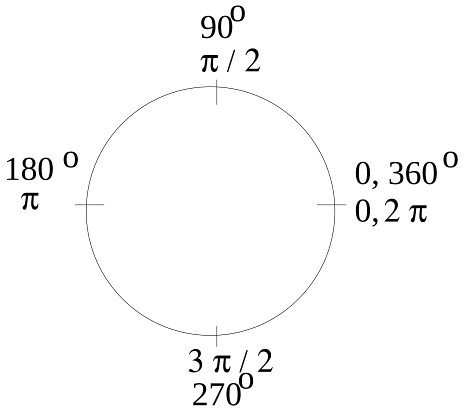


Figure 4.29: Unit Circle Orientation of θ Along Wall of Container

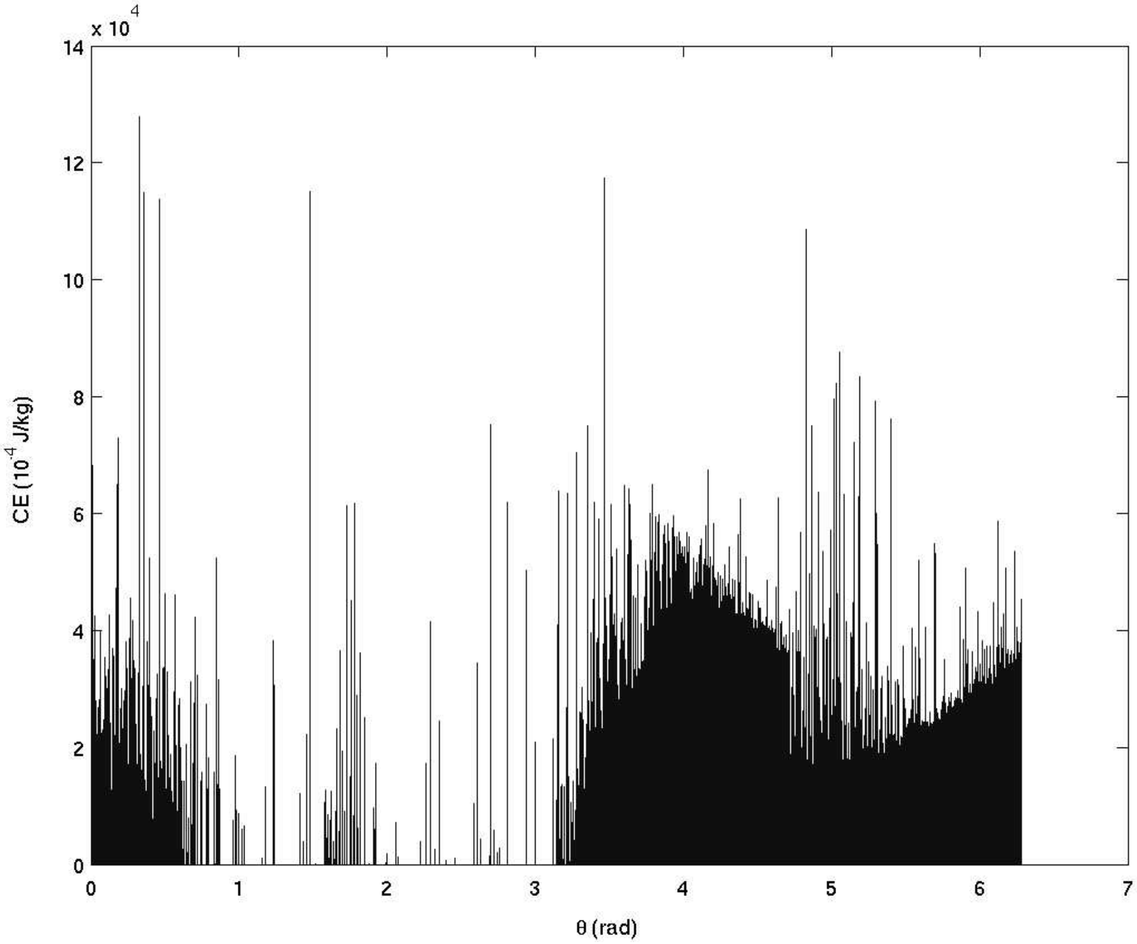


Figure 4.30: Bar Chart of Normalized Disk Container Collision Energies (CE/M) vs θ: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

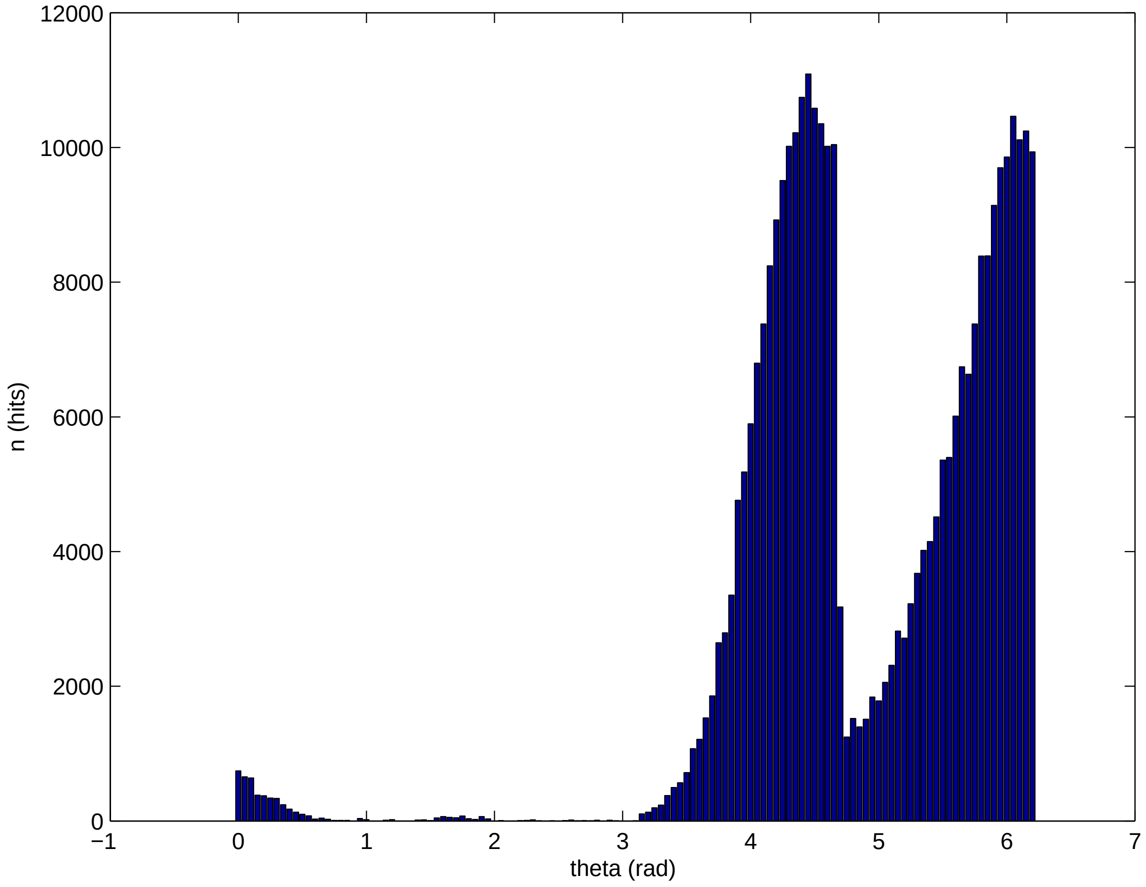


Figure 4.31: Bar Chart of Number of Hits of Disks Against Container (n) vs θ: $\omega_y = 126\,\mathrm{rad/s}$, $A_y = 1.5\,\mathrm{cm}$, $e_0 = 0.4$, $e_W = 1.0$

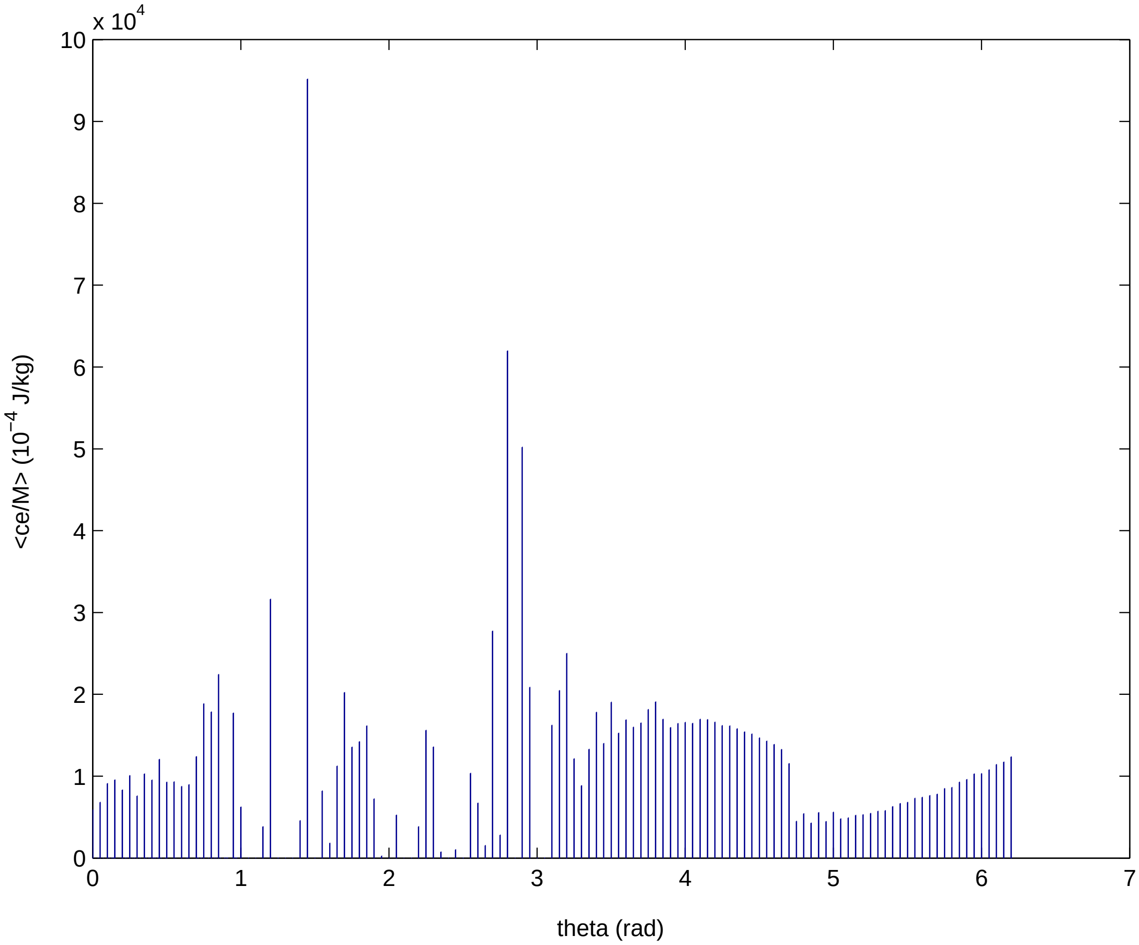


Figure 4.32: Bar Chart of Average Normalized Disk Container Collision Energies (<CE/M>) vs θ: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

Comparing the contour plot with the bar charts, one sees that the highest disk disk collision energies are in the same locations as the highest disk container collision energies (along the bottom of the container). One can see this is evident from the velocity field shown in Figure 4.33. At the locations of the highest collision energies, the velocity vectors are largest.

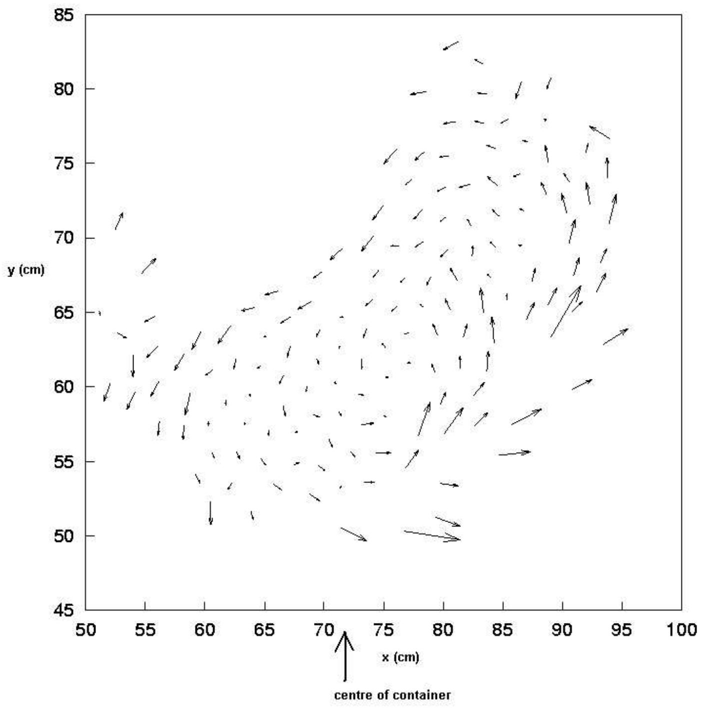


Figure 4.33: Velocity Field Snapshot for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

From the contour plots, one can calculate the area A of the contour that contains a collision energy above a specified threshold modulus of toughness value and compare this area to the total area $\tilde{A}$ of all the contours. Data files were also created for each run that list for each collision, the disk identification number, position, and collision energy, so that one can trace how many different disks experience a collision energy above a specified threshold modulus of toughness value. Plots of n_1, n_2, n_3, n_4 and n_5 vs time were produced where n_i is the number of disks that experience the mean collision energy (between a disk and a disk or between a disk and the container) i times. See Figure 4.34.

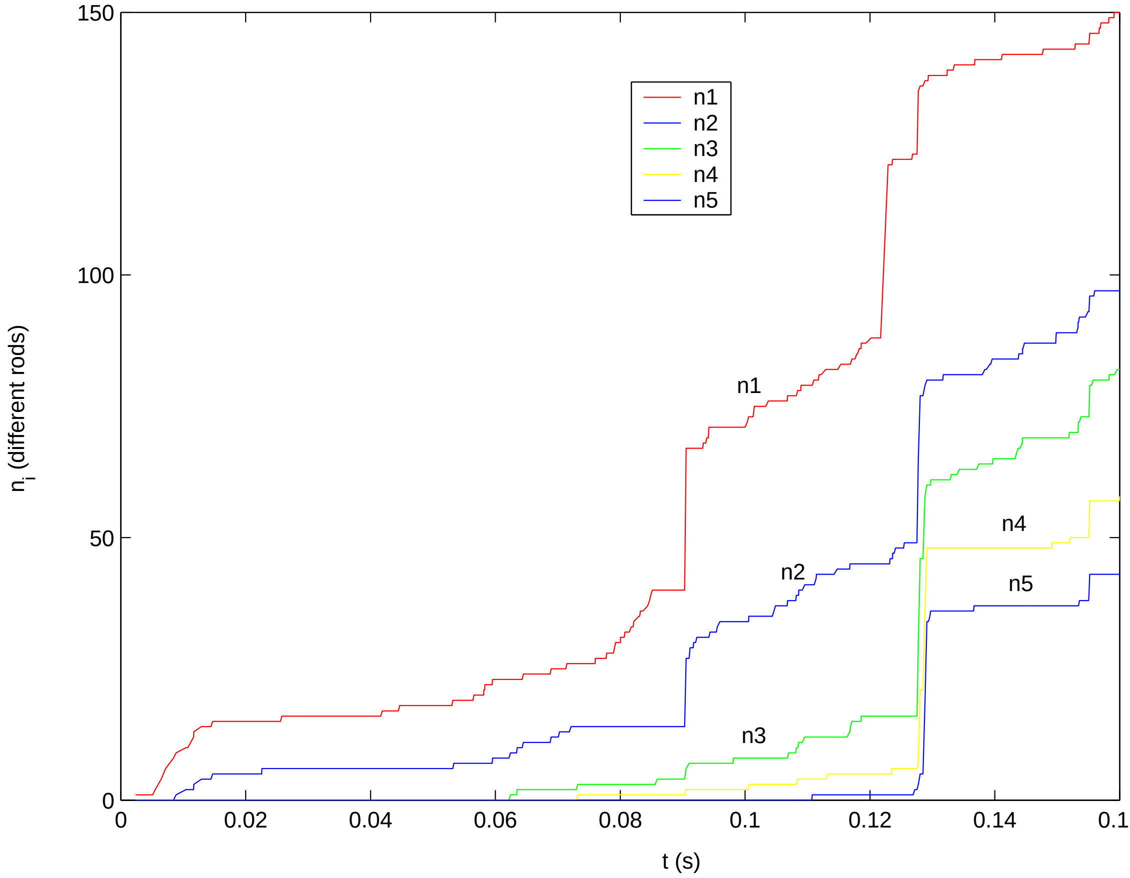


Figure 4.34: Number of Different Disks That Experience Mean Collision Energy (n_i) vs Time (t) for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. There are 150 disks in total.

From this plot one can read off the *saturation time* when n_i is equal to the total number of disks and since the program creates a data file listing each collision's dissipated energy over time, one can then sum the dissipated energy up to this saturation time. Thus, we have established a recipe to take a specific frequency and amplitude of oscillation and a specific medium to be crushed and calculate optimum ratings because we can tell when all the disks achieve some threshold energy (the experimentally measured modulus of toughness for the material to be crushed, or alternatively the simulated mean collision energy) an arbitrary number of times. The value of this threshold energy and the desired number of times that it is to be achieved (see Figure 4.34) will of course depend on the type of material to be crushed. Using the total rod-container dissipated energy (see Figures 4.35 and 4.36) one can obtain a value for the power consumption (dividing the dissipated energy by the duration of the simulation) and relate this to a dollar cost (using a rate of so many dollars per unit of power consumption). We can tabulate this dollar cost as a function of the frequency and amplitude of oscillation. One would want to compare the saturation time with the mixing time (see section 4.5 for a description of the mixing time). One would choose the larger of these two times for computing the total dissipated energy and power consumption. This will ensure that the rods are well-mixed and that all the rods experience the mean collision energy. The power consumption calculation could then be calibrated with the experimentally measured power consumption. All these considerations are part of the grinding mill design process.

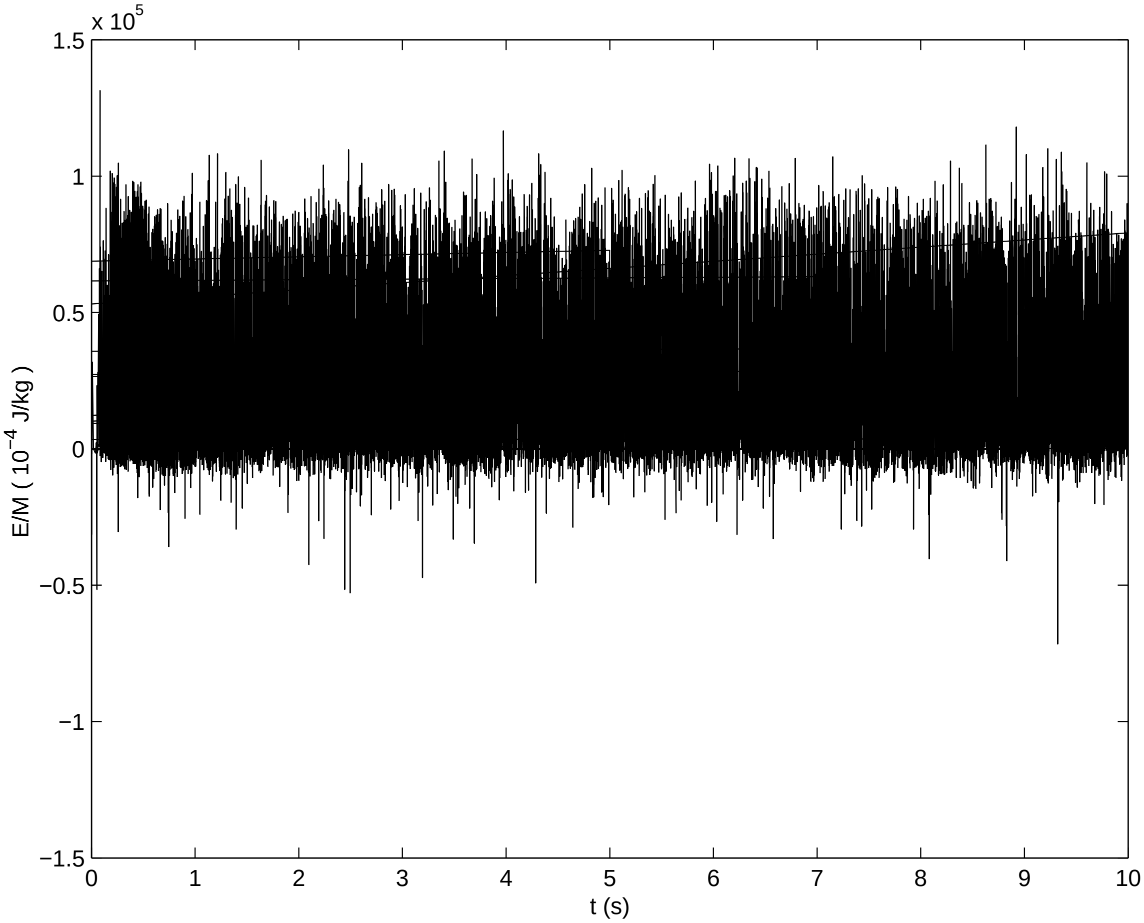


Figure 4.35: Normalized Dissipated Energy (E/M) for Rod-Container Collisions vs Time (t) for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$.

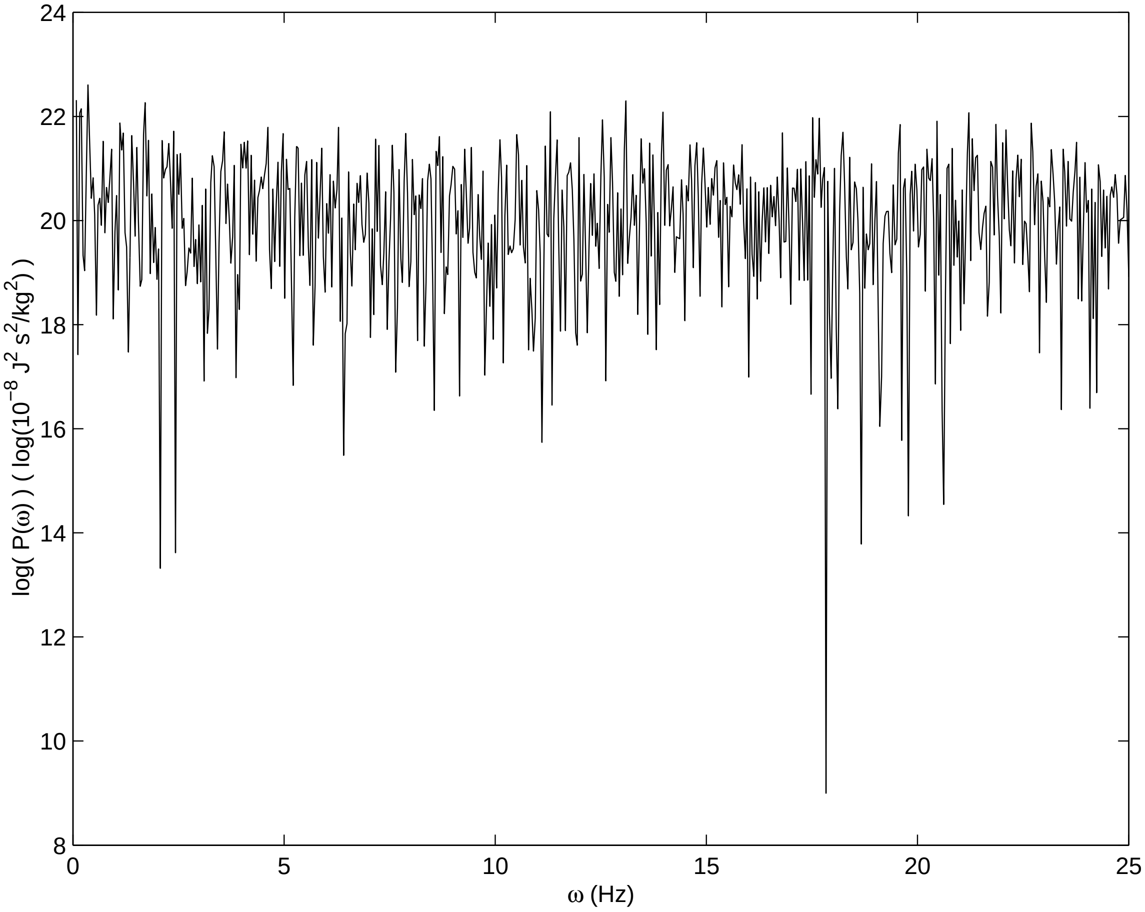


Figure 4.36: Power Spectrum of Normalized Dissipated Energy ($\mathcal{P}(\omega)$) for Rod-Container Collisions for Typical Simulation: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$

4.9 Force Chains

An interesting phenomenon known as *force chains* or *stress chains* [27] was observed in the animated visualizations. Force chains are basically like *lightning bolts* of force travelling through a link of adjacent disks. Force chains were visualized by rendering the colliding disks a different color from the non-colliding disks. See Figure 4.37 for a snapshot of an animation showing a force chain.

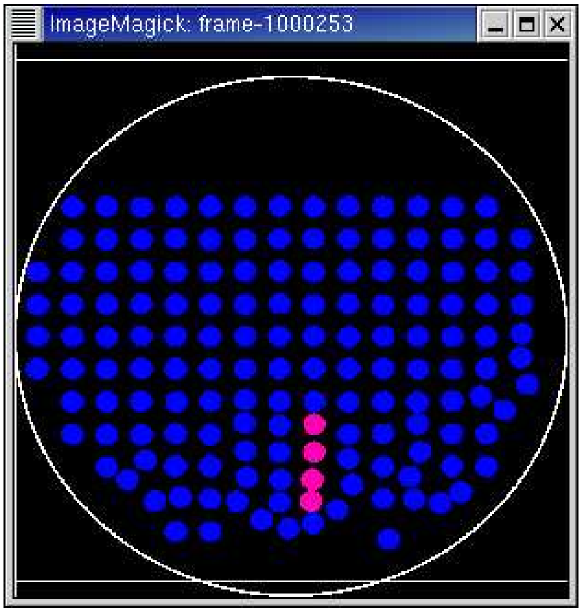


Figure 4.37: Snapshot of Animation Showing Force Chain: $\omega_y = 126$ rad/s, $A_y = 1.5$ cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink.

4.10 Avalanche Movement and the Angle of Repose

G. H. Ristow best explains the avalanche movement which is given verbatim as follows:

> Imagine yourself building a sandpile by slowly pouring dry grains constantly at the same spot...Due to *local* grain rearrangements, the pile will become steeper and steeper in time until a critical surface slope is reached, called the *angle of marginal stability*...When this angle is reached, a *global* grain motion sets in, namely an avalanche detaches, which can transport grains all the way down the slope of the pile...When the grain motion has stopped, the slope of the pile has reached the *angle of repose* of the material...[1]

When the simulation was run during a steady state, the disks clustered together and circulated such that the cluster of disks would avalanche and show an angle of repose. See Figure 4.38 for a snapshot of an animation showing avalanche movement and an angle of repose.

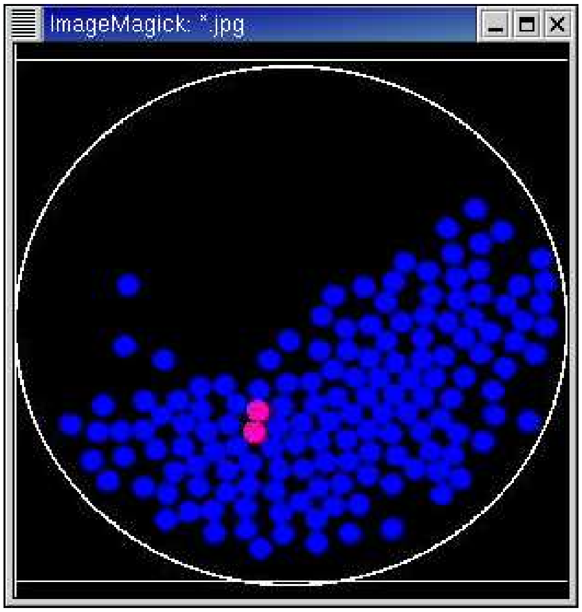


Figure 4.38: Snapshot of Animation Showing Avalanche Movement and Angle of Repose: ω_y = 126 rad/s, A_y = 1.5 cm, $e_0 = 0.4$, $e_W = 1.0$. The colliding disks are coloured in pink.

Chapter 5

CONCLUSIONS AND RECOMMENDATIONS

Following is a summary of what has been presented in this thesis. First, the subject of granular materials was introduced along with examples from published papers. The particular grinding mill industrial crushing application was then described and an event-driven algorithm was explained which was used to model this application. Verification of the program against published works and an independently written simulation was presented and the parameters for the simulation were discussed. Results from the simulation were next analyzed which included steady state, mixing, phase diagrams, circulation, quantifying crushing forces as collision energies, force chains, avalanche movement and the angle of repose. By quantifying the crushing forces in terms of collision energies and studying circulation and mixing, this thesis has outlined a thorough systematic approach to studying the grinding mill industrial crushing problem.

Experiments could also be and are performed using steel rods and the *Vibradrum* grinding mill machine (see Figure 1.6). By running this simulation using finer test matrices and comparing results with experimentation, one should be able to calibrate simulation results to experiments, perform measurements to compare the percentages of the medium that was crushed and compare this to the $n1$, $n2$, ... and saturation times shown in Figure 4.34 for various simulations using different amplitudes and frequencies of oscillation. This will determine the frequency and amplitude for an optimum oscillation, that is, to minimize the energy required to get a well-mixed container with all the rods experiencing the mean collision energy. This is the solution to the task outlined in this thesis.

Recommendations for future work in this study include:

- incorporating horizontal container oscillations
- incorporating rod rotations

- using a mixture of different sized rods
- using different quantities of rods and different container sizes
- exploring different container wall boundary conditions that cause multiple vortices
- parallelizing the code with each sector run by a different processor (for a large number of rods)
- implementing three-dimensional simulations
- studying the behaviour of the grinding medium as it is being crushed between the rods: materials that have a larger strain at rupture would be more elastic (squashier) and thus would significantly dampen the collision energy when being crushed between the disks. Mechanisms to consider include "(a) distortions or change in shape of the grinding medium, (b) fracturing and/or crushing of the grinding medium, and (c) sliding and/or rolling of individual particles" [28]. For instance, "[p]lastic deformation may be induced in assemblies of cohesionless particles with a resistance that is a function of the interparticle friction" [29]

Appendix A

Numerical Methods Used to Study Granular Materials

In this chapter the most common simulation techniques for modelling granular systems are summarized as presented in G. H. Ristow's book, *Pattern Formation in Granular Materials* [1]. The underlying simplifications as well as the limitations of each method are discussed.

A.1 Monte Carlo Method

This method uses random numbers to calculate and find particle positions which minimize the potential energy of the system. The limitations of the Monte Carlo Method are explained by Ristow below:

> The limitations of this method are: (i) no physical time scale enters the model, since the collision time is assumed to be zero in order to minimize the potential energy during each particle move. It is therefore difficult to relate the physical material properties to the random-walk process, which might be the reason why this method has not attracted much attention in the field of granular materials in recent years [1].

A.2 Diffusing-Void Model

As explained by Ristow:

> This method focuses on the overall geometry of highly packed granular materials through which voids diffuse under the influence of gravity. It is based on a random-walk process on a discrete lattice [1].

Ristow outlines the limitations of the Diffusing-Void Model below:

> The model contains many simplifications and limitations, e.g. (i) the void diffusivity which enters the model is not correlated to any physical or

> geometrical quantity, (ii) the voids maintain their integrity as they move upwards whereas they disperse in real granular systems and (iii) stresses do not appear in the model but they play a crucial role in reality. [1]

A.3 Method of Steepest Descent

As explained by Ristow:

> [This method] ... is motivated by the method of ballistic deposition and should be understood as a *toy-model* to study the geometrical effects in granular materials. Particles follow the path of steepest descent and undergo a sequence of *rolling* and *falling* steps before they reach a local surface minimum. New particle contacts are treated as completely inelastic collisions. [1]

This model has been criticized because it lacks a stochastic element. Other limitations as explained by Ristow are given below:

> [T]his model is only capable of studying geometrical effects. When a particle reaches a local minimum it gets stuck forever. Inertia and elasticity are thus neglected, which can be viewed as the limit of large friction and low restitution coefficient. Consequently, the angle of repose of a sandpile or in a rotating drum is unphysically high.
>
> Also, the simulation time step is not connected to the physical time but rather acts in the rotating drum as a source of vertical vibration to allow small structural rearrangements. These facts make this model inadequate to describe the full dynamics of granular materials having different material properties, e.g. restitution coefficients. [1]

A.4 Cellular Automata

As explained by Ristow:

> [C]ellular automata have a number of basic defining characteristics:
>
> - discrete in space and time

> - discrete states, i.e. a finite number of possible values at each site
> - homogeneous, i.e. all cells are equal
> - synchronous updating
> - deterministic rule
> - spatially and temporally local rule.
>
> The use of cellular-automaton models to study granular materials dates back to the introduction of the concept of *self-organized criticality*, where sandpile avalanche statistics were used as one example. ... In these models, space is discretized into cells which have the size of the particles and can either be occupied or empty. The particle dynamics are modelled by a set of particle-collision rules which apply when certain conditions are fulfilled, e.g. the local surface angle (slope) exceeds a threshold value. ...
>
> It is also possible to choose collision rules that make the system fulfil the Navier-Stokes equations, which is then called a *lattice-gas* automaton. ... This seems to be a promising starting point to model granular materials, when the rules are modified to include the important components gravity, dissipation and static friction. [1]

Limitations of using Cellular Automata to model granular materials are described by Ristow below:

> [A]s with most models working on lattices, the surface angle is mostly given by the topology of the underlying lattice and only identical particles were studied so far. A direct connection of the update time and the physical time is also missing, which greatly reduces the scope of the cellular automaton approach. [1]

A.5 Event-Driven Simulations

As explained by Ristow:

> For low-density particulate systems, particles will spend most of their time during free flights and not during collisions or contacts. From a computational point of view it might be favourable in such a system to avoid

> using a constant time step, but rather to calculate all the collision times in the system and update all particle positions, velocities and accelerations to the shortest collision time calculated. This is especially simple for hard spheres, where the collisions are instantaneous, i.e. the contact time is zero, and only binary collisions occur. After the colliding particles have been updated according to the collision model, the collision table is calculated again and the circle starts anew. ... This algorithm is termed *event-driven* ... and ... [a]n efficient serial algorithm was given by Lubachevsky [8]. However, for large particle numbers and low resititution coefficient, the system can undergo an infinite number of collisions in finite time, leading to clustering, which was termed *inelastic collapse.* ... In such a case the event-driven algorithm breaks down and care has to be taken to detect such a situation. ...
>
> By comparing event-driven and classical molecular dynamics simulations, a perfect agreement of the results was found in the low-density limit in one-dimensional as well as two-dimensional systems. For higher densities, care must be taken that the constant time step used in the molecular dynamics simulation is small enough compared to the collision time. ...
>
> Since the contact detection is only simple for hard-core particles, the classical molecular dynamics approach which uses a constant time step is more flexible and especially advantageous when the softness of the particles has to be considered. [1]

A.6 Time-Driven Simulations (Molecular Dynamics)

As explained by Ristow,

> [This method, a]s the name implies, ... is widely used to study the dynamics of molecules but can be adapted to granular systems in a straightforward fasion which will be outlined below. There it is also referred to as a *discrete-element method.*
>
> One seeks a description of the dynamics of the granular system on a particle basis for each individual grain, starting from Newton's second

law of motion which reads for a system of grains

$$\mathbf{F}_i = m_i \mathbf{a}_i = m_i \frac{d^2 \mathbf{x}_i}{dt^2}, \tag{A.1}$$

where $\mathbf{F}_i$ denotes the force acting on particle i. $\mathbf{x}_i$ and $\mathbf{a}_i$ stand for the position vector and the acceleration of the ith particle, respectively.

A powerful numerical technique which allows us to follow the full dynamics of a system is a molecular dynamics (MD) simulation. The time t is discretized in constant steps δt and a Taylor expansion for the particle positions for times close to τ reads

$$\text{forward } \mathbf{x}_i(\tau + \delta t) = \mathbf{x}_i(\tau) + \delta t \mathbf{v}_i(\tau) + \frac{\delta t^2}{2} \mathbf{a}_i(\tau) + ..., \tag{A.2}$$

$$\text{backward } \mathbf{x}_i(\tau - \delta t) = \mathbf{x}_i(\tau) - \delta t \mathbf{v}_i(\tau) + \frac{\delta t^2}{2} \mathbf{a}_i(\tau) - ..., \tag{A.3}$$

Adding equations (A.2) and (A.3) leads to the expression for the position of particle i at a later time $\tau + \delta t$ (all odd powers of δt drop out)

$$\mathbf{x}_i(\tau + \delta t) = 2\mathbf{x}_i(\tau) - \mathbf{x}_i(\tau - \delta t) + \delta t^2 \mathbf{a}_i(\tau) + ... \tag{A.4}$$

If one is interested in the velocities, e.g. to calculate the granular temperature which is related to the kinetic energies of the particles, one simply has to subtract equation (A.3) from equation (A.2) yielding

$$\mathbf{v}_i(\tau) = \frac{\mathbf{x}_i(\tau + \delta t) - \mathbf{x}_i(\tau - \delta t)}{2\delta t} + ... \tag{A.5}$$

By knowing the positions of all particles for two successive times, or alternatively the positions and velocities at a given time, the future behaviour of the system is fully determined by the forces $\mathbf{F}_i$ acting on each particle. These are the basic ideas of the *Verlet algorithm*.

Appendix B

Sample and Hold

Here, we discuss the effects of the sample and hold technique on the Fourier transform as explained in Sophocles J. Orfanidis's book entitled *Introduction to Signal Processing* [31]. Figure B.1 shows an ideal situation in which the sample points are evenly spaced, compared with a situation where the data is unevenly spaced and a sample and hold technique is required to create an output of evenly spaced points. Evenly spaced points are required as input to the Fourier transform. The output $\hat{x}(t)$ in the ideal case consists of a linear superposition of impulses occurring at the evenly spaced sample times:

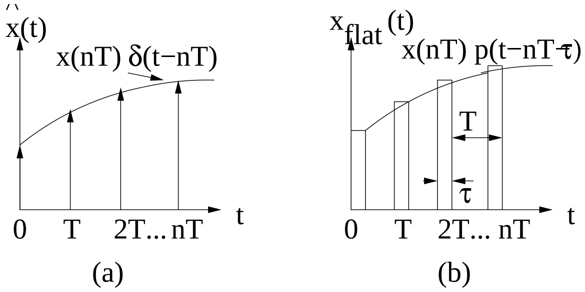


Figure B.1: (a) ideal case and (b) sample and hold case [31]

$$\hat{x}(t) = \sum_{-\infty}^{\infty} x(nT)\delta(t - nT). \tag{B.1}$$

In the sample and hold case, each sample is held constant for a short period, τ seconds, to convert the sample to evenly spaced points. In this case the output is:

$$x_{\text{flat}}(t) = \sum_{-\infty}^{\infty} x(nT)p(t - nT - \tau), \tag{B.2}$$

where $p(t)$ is a flat-top pulse of duration τ seconds, with $\tau \ll T$.

In the ideal case, the spectrum $\hat{x}(t)$ is the Fourier transform:

$$\hat{X}(\omega) = \int_{-\infty}^{\infty} \hat{x}(t)e^{-i\omega t}dt. \tag{B.3}$$

Inserting Equation (B.1) into Equation (B.3) and interchanging integration and summation, we get:

$$
\begin{aligned}
\hat{X}(\omega) &= \int_{-\infty}^{\infty} \sum n = -\infty^{\infty} x(nT)\delta(t - nT)e^{-i\omega t}dt && \text{(B.4)} \\
&= \sum_{n=-\infty}^{\infty} x(nT) \int_{-\infty}^{\infty} \delta(t - nT)e^{-i\omega t}dt && \text{(B.5)} \\
&= \sum_{n=-\infty}^{\infty} x(nT)e^{-i\omega nT}. && \text{(B.6)}
\end{aligned}
$$

In Equation (B.6) we used the property of the delta function that:

$$\int_{-\infty}^{\infty} \delta(x - a)f(x)dx = f(x - a). \tag{B.7}$$

In the sample and hold case we have:

$$\hat{X}(\omega) = \int_{-\infty}^{\infty} x_{\text{flat}} e^{-i\omega t}dt, \tag{B.8}$$

and upon substituting Equation (B.2) into Equation (B.8) we finally get:

$$
\begin{aligned}
\hat{X}(\omega) &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT)p(t - nT - \tau)e^{-i\omega t}dt && \text{(B.9)} \\
&= \sum_{n=-\infty}^{\infty} x(nT) \int_{-\infty}^{\infty} p(t - nT - \tau)e^{-i\omega t}dt. && \text{(B.10)}
\end{aligned}
$$

It is to be noted that in the simulation τ was on the order of 10^{-5} and T was 10^{-3}, so the error should be minimal.

Appendix C

Rotations

If one wanted to incorporate rotations, one would introduce roughness to the disks. Tildesley [15] states that rough particles differ from smooth particles only in their collision dynamics: the techniques used to locate future collisions remain the same. As Tildesley explains, rough particles are characterized by a diameter σ, a mass m, and a moment of inertia I or, alternatively, a parameter $\kappa = 4I/m\sigma^2$. For a rod, $I = \frac{1}{2}mr^2$ where $r = \sigma/2$ and hence for a rod $\kappa = \frac{1}{2}$. As with smooth rods, rough rods have translational velocities $\mathbf{v}$ but in addition they have angular or 'spin' velocities ω. An elastic collision between two rods will conserve total kinetic energy (rotational plus translational), total linear momentum, and total angular momentum defined by

$$\mathbf{J} = \mathbf{L} + \mathbf{S} = \sum_i m\mathbf{r}_i \times \mathbf{v}_i + \sum_i I\omega_i. \tag{C.1}$$

Fig. C.1 shows the two points on the rod surfaces that come together at the moment of impact. The relative velocity vector of these points of contact is

$$\mathbf{v}_{ij}^{imp} = \mathbf{v}_i^{imp} - \mathbf{v}_j^{imp} = (\mathbf{v}_i - \mathbf{v}_j) - \frac{1}{2}(\omega_i + \omega_j) \times \mathbf{r}_{ij}. \tag{C.2}$$

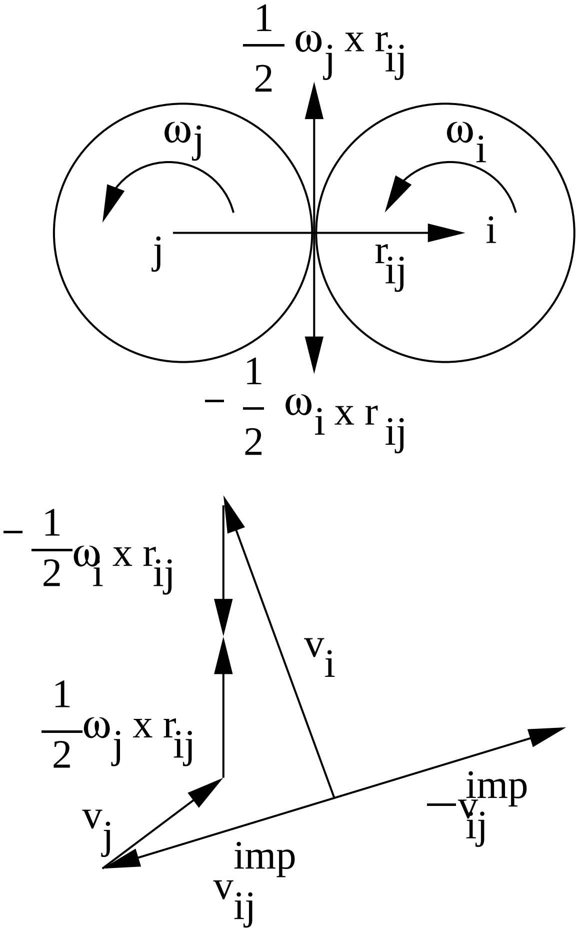

$\frac{1}{2}\omega_j \times r_{ij}$
ω_j
ω_i
j
i
r_{ij}
$-\frac{1}{2}\omega_i \times r_{ij}$
$-\frac{1}{2}\omega_i \times r_{ij}$
v_i
$\frac{1}{2}\omega_j \times r_{ij}$
$-v_{ij}^{imp}$
v_j
v_{ij}^{imp}

Figure C.1: A rough disk collision [15]

The $\mathbf{v}_{ij}^{imp}$ vector can be resolved into components parallel and perpendicular to the line of the centers of the two colliding disks. Since the disks are hard, the parallel component of $\mathbf{v}_{ij}^{imp}$ is reversed on impact. For the smooth disks, the perpendicular component is not changed because for smooth disks there is no change in the angular velocities, but for the rough disks, both the parallel and perpendicular components of the relative velocity vector $\mathbf{v}_{ij}^{imp}$ are reversed on impact. Thus

$$\delta\mathbf{v}_{ij}^{imp} = \mathbf{v}_{ij}^{imp}\,(after) - \mathbf{v}_{ij}^{imp}\,(before) = -2\mathbf{v}_{ij}^{imp}\,(before) \tag{C.3}$$

Using conservation laws, one can obtain an expression for the impulse as

$$\delta\mathbf{p}_i = \frac{1}{2}m\left(\delta\mathbf{v}_{ij}^{imp\,\|} + \frac{\kappa}{1+\kappa}\delta\mathbf{v}_{ij}^{imp\,\perp}\right). \tag{C.4}$$

The changes in velocities and angular velocities then become

$$\begin{aligned} m\mathbf{v}_i\,(after) &= m\mathbf{v}_i\,(before) + \delta\mathbf{p}_i \\ m\mathbf{v}_j\,(after) &= m\mathbf{v}_j\,(before) - \delta\mathbf{p}_i \\ I\omega_i\,(after) &= I\omega_i\,(before) - \frac{1}{2}\mathbf{r}_{ij}\times\delta\mathbf{p}_i \\ I\omega_j\,(after) &= I\omega_j\,(before) - \frac{1}{2}\mathbf{r}_{ij}\times\delta\mathbf{p}_i \end{aligned} \tag{C.5}$$

Walton in his chapter on *Numerical Simulation of Inelastic, Frictional Particle-Particle Interactions* [30] introduces a coefficient of friction μ and a parameter β^* that controls the choice between a rolling or sliding contact. This has been successfully implemented by Baran [3].

Following is Walton's analysis of a collision between two inelastic particles. Suppose we have two particles with mass m_a and m_b with centers located at $\mathbf{r}_a$ and $\mathbf{r}_b$ travelling with velocities $\mathbf{v}_a$ and $\mathbf{v}_b$ before impact and having rotational velocities $\vec{\omega}_a$ and $\vec{\omega}_b$. Let

$$\begin{aligned} \hat{\mathbf{r}}_{ab} &= \frac{\mathbf{r}_b - \mathbf{r}_a}{|\mathbf{r}_b - \mathbf{r}_a|} = \text{unit vector from } a \text{ to } b, \\ \mathbf{v}_{ab} &= \mathbf{v}_a - \mathbf{v}_b = \text{relative velocity}, \\ v_n &= \mathbf{v}_{ab}\cdot\hat{\mathbf{r}}_{ab} = \text{normal component of relative velocity}, \\ \mathbf{v}_n &= v_n\hat{\mathbf{r}}_{ab} = \text{relative velocity in normal direction} \\ \mathbf{v}_t &= \hat{\mathbf{r}}_{ab}\times(\mathbf{v}_{ab}\times\hat{\mathbf{r}}_{ab}) = \mathbf{v}_{ab} - \mathbf{v}_n = \text{relative velocity in tangential direction}, \\ \mathbf{v}_s &= \mathbf{v}_t + \frac{\sigma_a}{2}\left(\hat{\mathbf{r}}_{ab}\times\vec{\omega}_a\right) + \frac{\sigma_b}{2}\left(\hat{\mathbf{r}}_{ab}\times\vec{\omega}_b\right) = \text{relative surface velocity (tangential direction)} \end{aligned}$$

$$\begin{aligned}\hat{\mathbf{k}}_s &= \frac{\mathbf{v}_s}{|\mathbf{v}_s|} = \text{unit vector in direction of incident surface velocity} \\ v_s &= \mathbf{v}_s \cdot \hat{\mathbf{k}}_s = \text{tangential component of relative surface velocity}\end{aligned} \tag{C.6}$$

We use the usual definition of the coefficient of restitution, $e \equiv -v_n'/v_n$, and we define the rotational restitution coefficient as $\beta \equiv -v_s'/v_s$, where the prime denotes post collision values. Then conserving translational momentum, we obtain the changes in normal direction velocities,

$$\Delta\mathbf{v}_{na} = \frac{m_b\,(1+e)}{m_a+m_b}\mathbf{v}_n \tag{C.7}$$

$$\Delta\mathbf{v}_{nb} = \frac{-m_a\,(1+e)}{m_a+m_b}\mathbf{v}_n \tag{C.8}$$

Using conservation of angular momentum about the contact point we can obtain expressions for the changes in rotational velocities $\vec{\omega}_a$ and $\vec{\omega}_b$ and the tangential direction translational velocities $\mathbf{v}_{ta}$ and $\mathbf{v}_{tb}$,

$$\Delta\vec{\omega}_a = \frac{2}{\kappa\sigma_a}\hat{\mathbf{r}}_{ab} \times \Delta\mathbf{v}_{ta} \tag{C.9}$$

$$\Delta\vec{\omega}_b = \frac{2}{\kappa\sigma_b}\hat{\mathbf{r}}_{ab} \times \Delta\mathbf{v}_{tb} \tag{C.10}$$

$$\Delta\mathbf{v}_{tb} = -\frac{m_a}{m_b}\Delta\mathbf{v}_{ta}. \tag{C.11}$$

Here $\kappa = 4I/m\sigma^2 = I/mr^2 = 1/2$ for a disk where I is the moment of inertia, σ is the diameter of the disk and r is the radius of the disk.

The procedure assumes a sliding contact to obtain the changes in tangential and rotational velocities, and then checks if the rotational restitution coefficient resulting from the sliding solution, β^*, given by

$$\beta^* = -1 + \mu\,(1+e)\left(1+\frac{1}{\kappa}\right)\frac{v_n}{v_s} \quad \text{(sliding solution)} \tag{C.12}$$

exceeds a predetermined (constant) value, β_0, where β_0 is between 0 and 1. If $\beta^* > \beta_0$, then a rolling solution is employed with a fixed $\beta = \beta_0$ to determine the post collision velocities.

Below is an algorithm that implements this model using equal size, equal mass, inelastic, frictional disks:

$$\hat{\mathbf{r}}_{ab} = \frac{1}{\sigma}\left(\mathbf{r}_b - \mathbf{r}_a\right) \text{(at contact)}$$

$$
\begin{aligned}
\mathbf{v}_{ab} &= \mathbf{v}_b - \mathbf{v}_a \\
v_n &= \mathbf{v}_{ab} \cdot \hat{\mathbf{r}}_{ab} \\
\mathbf{v}_t &= \mathbf{v}_{ab} - v_n \hat{\mathbf{r}}_{ab} \\
\mathbf{v}_s &= \mathbf{v}_t + \frac{\sigma}{2} \hat{\mathbf{r}}_{ab} \times (\vec{\omega}_a + \vec{\omega}_b) \\
\text{if } \ (\mathbf{v}_s \cdot \mathbf{v}_s) &= 0 \text{ then} \\
\beta &= -1 \\
\text{else}\,(1+\beta^*)^2 &= \mu^2 (1+e)^2 \left(1 + \frac{1}{\kappa}\right)^2 \frac{v_n^2}{\mathbf{v}_s \cdot \mathbf{v}_s} \\
\text{if } \ (1+\beta^*)^2 &> (1+\beta_0)^2 \ \text{ then} \\
\beta &= \beta_0 \text{ (rolling solution)} \\
\text{else} & \\
\beta &= \beta^* \text{ (sliding solution} = \ -1 + \sqrt{(1+\beta^*)^2}) \\
\text{endif} & \\
\text{endif} &
\end{aligned}
$$

The changes in normal, tangential and rotational velocity components for each disk are then given by,

$$\Delta \mathbf{v}_{na} = -\Delta \mathbf{v}_{nb} = \frac{1}{2} (1+e)\, v_n \hat{\mathbf{r}}_{ab} \tag{C.13}$$

$$\Delta \mathbf{v}_{ta} = -\Delta \mathbf{v}_{tb} = \frac{\kappa (1+\beta)}{2 (1+\kappa)} \mathbf{v}_s \tag{C.14}$$

$$\Delta \vec{\omega}_a = \Delta \vec{\omega}_b = \frac{(1+\beta)}{\sigma (\kappa + 1)} \hat{\mathbf{r}}_{ab} \times \mathbf{v}_s \tag{C.15}$$

Appendix D

Direct Measures of Forces Based on Elasticity

During a collision between two disks, one has a normal force between the disks along the line joining their centers, and a tangential force which lies in a plane perpendicular to the normal force. In this thesis we will be concerned with only the normal force since that is the force that does the crushing between the disks. Following is a summary of the equations used to model the normal force between two colliding disks, as referenced from G. H. Ristow's book *Pattern Formation in Granular Materials* [1]. Ristow describes two models of the normal force used in granular material simulations: an elastic restoration force, F_{el}, modelled as a spring, and a viscous friction force, F_{fric}, modelled as a dashpot.

One often models the elastic restoration force as a linear Hookean spring using (see Figure D.1):

$$F_{\text{el}} = -k\left(r_i + r_j - (\mathbf{x}_j - \mathbf{x}_i)\cdot \hat{n}\right) = -kh, \tag{D.1}$$

where k is the spring constant and $h > 0$ is the virtual overlap of the two disks during contact as shown in Figure D.2.

For the viscous friction force one might model it linearly as

$$F_{\text{fric}} = -2\gamma m_{\text{eff}}\left(\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j\right)\cdot \hat{n} = -2\gamma m_{\text{eff}}\dot{h}, \tag{D.2}$$

where γ is the frictional damping coefficient, and $m_{\text{eff}} = \frac{m_i m_j}{(m_i+m_j)}$ is the effective or reduced mass. The dot notation represents derivatives with respect to time; for example, $\dot{\mathbf{x}}_j \equiv \frac{\partial x_j}{\partial t}$.

Using equations (D.1) and (D.2), the equation of motion for the collision becomes

$$\ddot{h} + 2\gamma\dot{h} + \omega_0^2 h = 0, \tag{D.3}$$

where $\omega_0 \equiv \sqrt{k/m_{\text{eff}}}$.

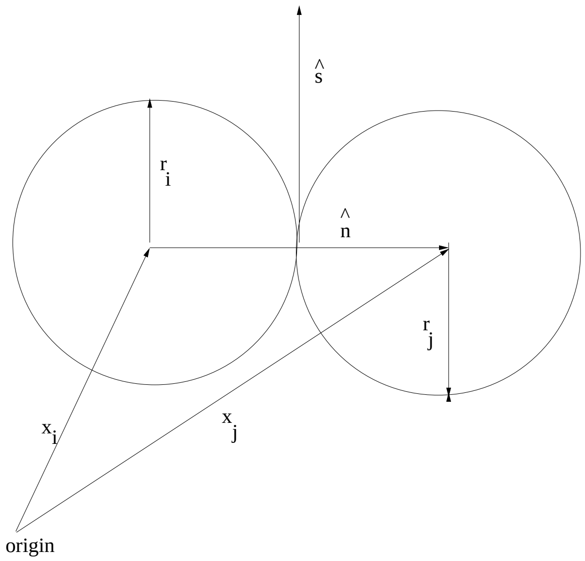


Figure D.1: Sketch of normal $\hat{n}$ and tangential $\hat{s}$ directions during disk-disk contacts [1]

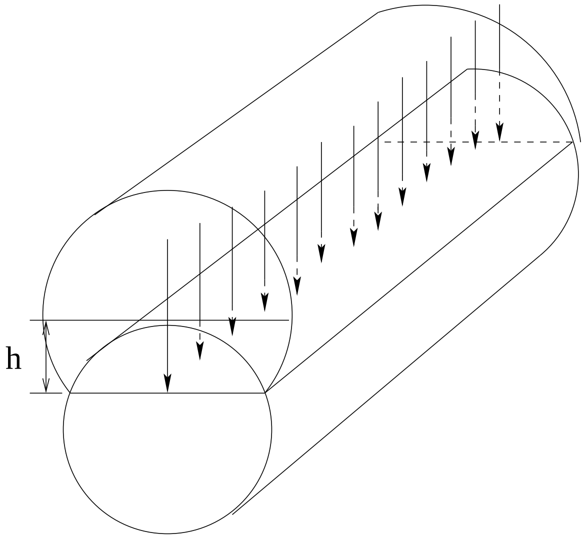


Figure D.2: Disk Overlap During Collision

Using the initial conditions $h(0) = 0$ and $\dot{h}(0) = v_0$, where v_0 is the impact velocity, the solution of equation (D.3) becomes

$$h(t) = \frac{v_0}{\omega} e^{-\gamma t} \sin(\omega t) \text{ with } \omega \equiv \sqrt{\omega_0^2 - \gamma^2}. \tag{D.4}$$

The time it takes for a disk to deform during a collision, t_{C}, is calculated using half the period of the sine wave function in equation (D.4) with $h(t_{\mathrm{C}}) = 0$ as

$$t_{\mathrm{C}} = \frac{\pi}{\omega} = \frac{\pi}{\sqrt{\omega_0^2 - \gamma^2}}. \tag{D.5}$$

One can relate the coefficient of restitution e in the normal direction using [1]

$$e \equiv -\frac{\dot{h}(t_{\mathrm{C}})}{\dot{h}(0)} = \exp\left(-\frac{\pi\gamma}{\omega}\right). \tag{D.6}$$

For particle-wall collisions, one treats the wall with infinite radius and mass, resulting in $r_{\mathrm{eff}} \rightarrow r_i$ and $m_{\mathrm{eff}} \rightarrow m_i$, where r_{eff} and m_{eff} are the effective radius and mass

$$\begin{aligned} r_{\mathrm{eff}} &\equiv \frac{r_i r_j}{(r_i + r_j)} \\ m_{\mathrm{eff}} &\equiv \frac{m_i m_j}{(m_i + m_j)} \end{aligned} \tag{D.7}$$

For particle-wall collisions one uses the substitutions

$$k \rightarrow k r_{\mathrm{eff}} \text{ and } \gamma \rightarrow \gamma \sqrt{r_{\mathrm{eff}}}\, m_{\mathrm{eff}}. \tag{D.8}$$

It is also possible to use different spring constants for the loading and unloading phases

$$F_{\mathrm{n}} = \begin{cases} -k_1 h & , \text{ for loading} \\ -k_2 (h - h_0) & , \text{ for unloading} \end{cases} \tag{D.9}$$

where h_0 is the value of the closest approach. The restitution coefficient is then

$$e = \sqrt{\frac{k_1}{k_2}} \tag{D.10}$$

In equation (D.9), k_2 is a linear function of the maximum force achieved during unloading and this allows the restitution coefficient to be made a decreasing function with increasing impact velocity in accordance with experiments. [1]

One problem with using the above Hookean spring model is that one has to obtain values for the spring constant k and spring constants for materials are not tabulated. Young's modulus of elasticity, E, Poisson's ration ν and the shear modulus of rigidity G however are tabulated. One could in principle use continuum mechanics elasticity theory such as in George E. Mase's book *Schaum's Outline Series in Theory and Problems of Continuum Mechanics* [32] and T. J. Chung's book *Applied Continuum Mechanics* [33] to relate the spring constant k to Young's modulus E, Poisson's ratio ν, and the shear modulus G. This will be developed next.

Figure D.3 shows how the spring constant k can be related to the normal stress σ_{22}.

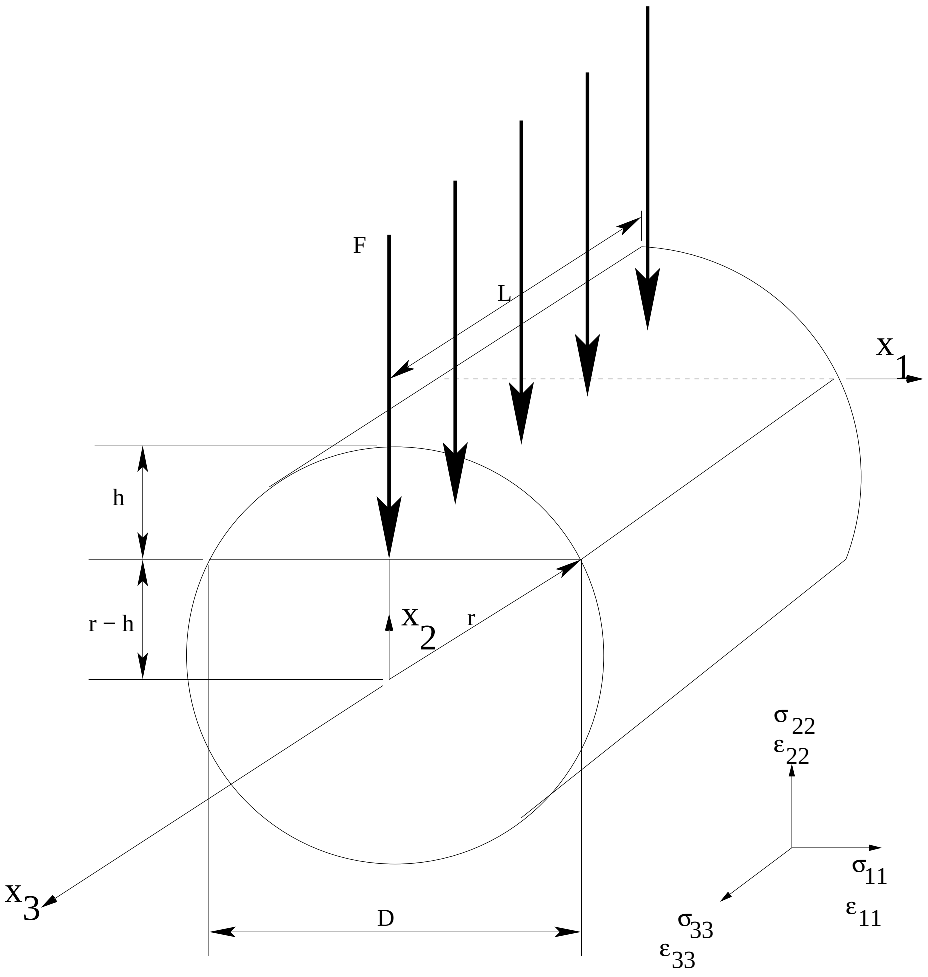


Figure D.3: Relating Spring Constant k to normal stress σ_{22}

The formula that relates the normal stress σ_{22} to the spring constant k is

$$\sigma_{22} = \frac{F}{DL} = \frac{kh}{2\sqrt{r^2 - (r-h)^2}L} \tag{D.11}$$

Then the normal stress σ_{22} can be related to Young's modulus E, Poisson's ratio ν, and the shear modulus G using the analysis developed by Mase [32] and Chung [33] as follows.

Chung defines *Lame* constants λ and μ as

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{D.12}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{D.13}$$

Mase's elasticity analysis reduces to the following equations:

$$\sigma_{22} = \lambda\left(\epsilon_{11} + \epsilon_{22}\right) + 2\mu\epsilon_{22} \tag{D.14}$$

$$\epsilon_{11} = \frac{\partial u_1}{\partial x_1} \tag{D.15}$$

$$\epsilon_{22} = \frac{\partial u_2}{\partial x_2} \tag{D.16}$$

where $\frac{\partial u_1}{\partial x_1}$ and $\frac{\partial u_2}{\partial x_2}$ are the displacement gradients.

Combining these formulae, one finally gets an expression for the normal stress σ_{22} as

$$\sigma_{22} = \frac{\nu E}{(1+\nu)(1-2\nu)}\left(\frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2}\right) + \frac{E}{(1+\nu)}\left(\frac{\partial u_2}{\partial x_2}\right) \tag{D.17}$$

Displacement gradients for a particular geometry can be calculated using an example found in Chung, p. 51. [33] Refer to Figure D.4 for all the possible geometries for disk compressions.

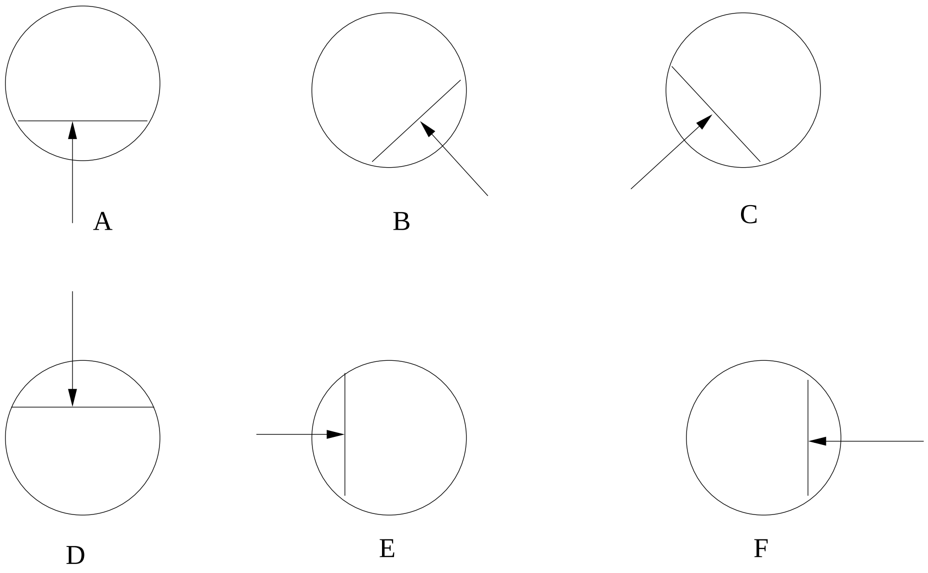


Figure D.4: Disk Compression Geometries

Following is an example calculation for geometry B as shown in Figure D.5.

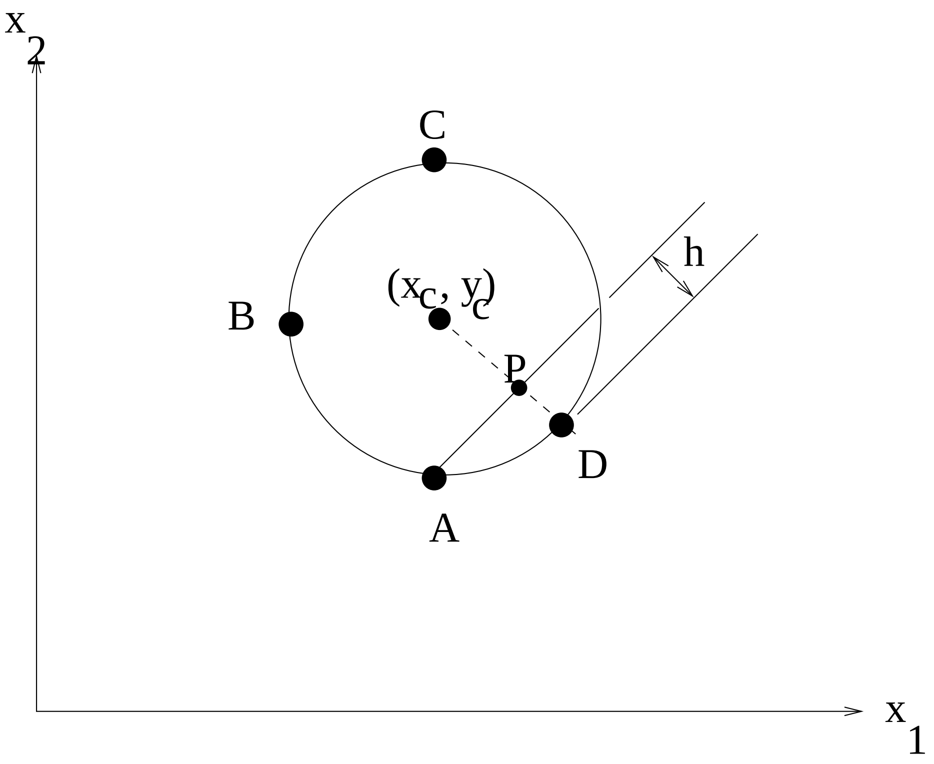


Figure D.5: Figure for Calculating Displacement Field

The most general equation for the displacement field at point i is

$$\begin{aligned} &\text{displacement in } x_1 \text{ direction: } u_1^i = C_1 + C_2 x_1^i + C_3 x_2^i + C_4 x_1^i x_2^i \\ &\text{displacement in } x_2 \text{ direction: } u_2^i = C_1 + C_2 x_1^i + C_3 x_2^i + C_4 x_1^i x_2^i \end{aligned} \tag{D.18}$$

As an example, at A, we have:

$$\begin{aligned} x_1^A &= x_C \\ x_2^A &= y_C - R \\ u_1^A &= 0 \\ u_2^A &= 0 \end{aligned} \tag{D.19}$$

Thus, at A:

$$\begin{aligned} u_1^A &= 0 = C_1 + C_2 x_C + C_3 \left(y_C - R\right) + C_4 x_C \left(y_C - R\right) \\ u_2^A &= 0 = C_1 + C_2 x_C + C_3 \left(y_C - R\right) + C_4 x_C \left(y_C - R\right) \end{aligned} \tag{D.20}$$

A similar analysis can be done at B, C, and D, where at D, $\left(u_1^D\right)^2 + \left(u_2^D\right)^2 = h^2$. The resulting eight equations can be solved for u_1 and u_2. Finally the displacement gradients, $\frac{\partial u_1^i}{\partial x_1}$ and $\frac{\partial u_2^i}{\partial x_2}$, can be obtained by taking derivatives of u_1 and u_2.

This is a lot of work, but at the end, experiments have shown that calculating forces using elasticity results in values that are off by orders of magnitude. Mullier, M., Tüzün, U., and Walton, O. R., in their paper entitled "A single-particle friction cell for measuring contact frictional properties of granular materials" perform an elasticity analysis to calculate the normal deformation between two cellulose acetate spheres. They compare these calculations to experimental values which they obtained from experiments using thermocouples and piezo electric displacement/load transducers. To quote them, they find that "the experimental values of the microdisplacement prior to gross sliding are several folds of those that would result from a perfectly elastic contact" [22]. Furthermore, for our industrial crushing application, an interstitial medium that is to be crushed lies between the disks, and this will affect the deformation calculations. A far better approach is to measure the forces using collision energies and then relate these to the *modulus of toughness* of the material that is to be crushed.

Appendix E

Shepard Interpolation

Below is a description of Shepard interpolation taken almost verbatim from the book *Numerical Algorithms with C* [26]. This technique was used to generate a contour plot of time averaged disk-disk collision energies (see Figure 4.28).

As described in *Numerical Algorithms with C*:

> If the nodes $(x_j, y_j), j = 0, ..., N$, with $(x_j, y_j) \in B \subset \Re^2$, do not form a rectangular grid but are arranged in a completely arbitrary and unordered way, we recommend to use the method of Shepard. This method has proven well suited for the graphical representation of surfaces. Its interpolating function Φ is uniquely determined independently from the ordering of the nodes (x_j, y_j). The function $f : z = f(x, y)$ for $(x, y) \in B$, where B is an arbitrary region of the x, y plane, is interpolated for the given nodes (x_j, y_j) by the function
>
> $$\phi(x,y) = \sum_{j=0}^{N} w_j(x,y) f_j. \tag{E.1}$$
>
> Here $\phi(x_j, y_j) = f(x_j, y_j)$ for $j = 0, ..., N$, where f_j are the given functional values $f(x_j, y_j)$ at the nodes (x_j, y_j), $j = 0, ..., N$, and $w_j = w_j(x, y)$ is a weight function. Setting
>
> $$r_j(x,y) = \sqrt{(x-x_j)^2 + (y-y_j)^2},\ j = 0, ..., N, \tag{E.2}$$
>
> we can define w_j as
>
> $$w_j(x,y) = \frac{\frac{1}{r_j^{\mu}}}{\sum_{i=0}^{N} \frac{1}{r_i^{\mu}}},\ 0 < \mu < \infty. \tag{E.3}$$
>
> With (E.3) the Shepard function ϕ has the representation
>
> $$\phi(x,y) = \frac{\sum_{j=0}^{N} \frac{1}{r_j^{\mu}} f_j}{\sum_{i=0}^{N} \frac{1}{r_i^{\mu}}}. \tag{E.4}$$

The exponent μ in (E.3) can be chosen arbitrarily. If $0 < \mu \leq 1$, the function ϕ has peaks at the nodes. If $\mu > 1$, the function is level at the nodes. The weights w_j satisfy

$$w_j(x,y) = \begin{cases} \geq 0 & , \quad \text{for all } (x,y) \in B, \\ = 1 & , \quad \text{for nodes } (x,y) = (x_j, y_j), \\ = 0 & , \quad \text{for nodes } (x,y) = (x_k, y_k) \text{ with } k \neq j, \end{cases} \tag{E.5}$$

and $\sum_{j=0}^{N} w_j(x,y) = 1$. The interpolation conditions $\phi(x_j, y_j) = f_j$ are satisfied for all j.

The definition of ϕ in (E.1) means, that all $N+1$ interpolation points (x_j, y_j, f_j), $j = 0, ..., N$, are used for calculating each new functional value $\phi(x,y)$, i.e., this is a global method. This can be alleviated by using a local variant. ... [A] modification for calculating the $w_j(x,y)$ ... uses only the nodes (x_j, y_j) within a circle of radius R around the point (x,y) when calculating a new functional value $\phi(x,y)$. The radius R of this circle must be chosen so that enough nodes (x_j, y_j) are situated within the local circle. ...

...

An improved version of the local method is the following: *Local Shepard interpolation with Franke-Little weights:* This very useful local variant uses the Franke-Little weights,

$$\xi_j(x,y) = \begin{cases} 1 - \frac{r_j(x,y)}{R} & , \quad \text{for } 0 < r_j(x,y) < R, \\ 0 & , \quad \text{for } r_j(x,y) \geq R. \end{cases} \tag{E.6}$$

The weight function w_j then becomes

$$w_j(x,y) = \frac{\xi_j^{\mu}(x,y)}{\sum_{i=0}^{N} \xi_i^{\mu}(x,y)}. \tag{E.7}$$

With (E.6) and (E.7), the local Shepard function has the form

$$\phi(x,y) = \sum_{j=0}^{N} w_j(x,y) f_j = \frac{\sum_{j=0}^{N} \left(1 - \frac{r_j(x,y)}{R}\right)^{\mu} f_j}{\sum_{i=0}^{N} \left(1 - \frac{r_i(x,y)}{R}\right)^{\mu}}.[26] \tag{E.8}$$

Appendix F

Compact Disc

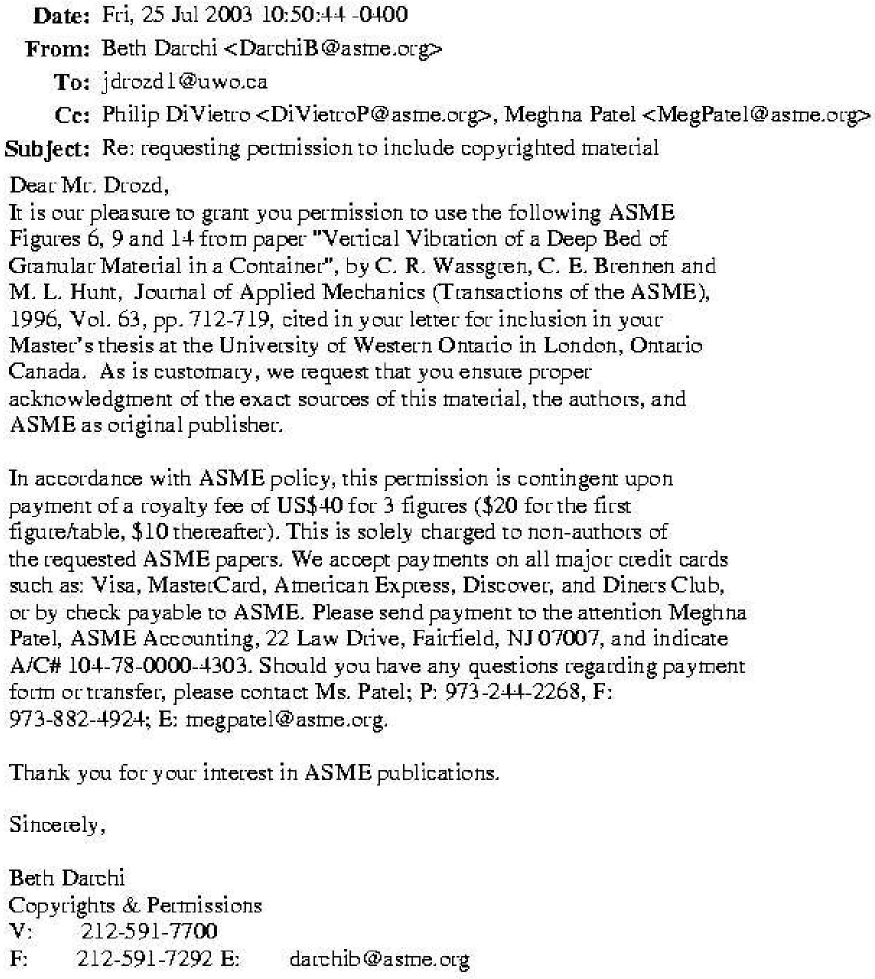
Attached is a compact disc which contains computer animations (mpeg movies) of the simulation, the C++ program code, sample data files and the thesis in pdf, ps, and tex formats along with the postscript figures and latex style files.

Appendix G

Permission to Print Figures

I have written to the American Society of Mechanical Engineers, Elsevier, and the American Physical Society, and I have received confirmation from all three sources that I may reproduce in this thesis Figures 1.1, 1.2, 1.3, 1.4 and 2.4. The American Physical Society asked that I also receive permission from one of the cited paper's authors. I received permission from one of the paper's authors by email. I am still awaiting permission from the publisher of Figure 1.6 but I expect no problems in obtaining their permission. Following are copies of emails granting permission to reproduce these figures.

G.1 Letter from American Society of Mechanical Engineers



Date: Fri, 25 Jul 2003 10:50:44 -0400
From: Beth Darchi <DarchiB@asme.org>
To: jdrozd1@uwo.ca
Cc: Philip DiVietro <DiVietroP@asme.org>, Meghna Patel <MegPatel@asme.org>
Subject: Re: requesting permission to include copyrighted material

Dear Mr. Drozd,
It is our pleasure to grant you permission to use the following ASME Figures 6, 9 and 14 from paper "Vertical Vibration of a Deep Bed of Granular Material in a Container", by C. R. Wassgren, C. E. Brennen and M. L. Hunt, Journal of Applied Mechanics (Transactions of the ASME), 1996, Vol. 63, pp. 712-719, cited in your letter for inclusion in your Master's thesis at the University of Western Ontario in London, Ontario Canada. As is customary, we request that you ensure proper acknowledgment of the exact sources of this material, the authors, and ASME as original publisher.

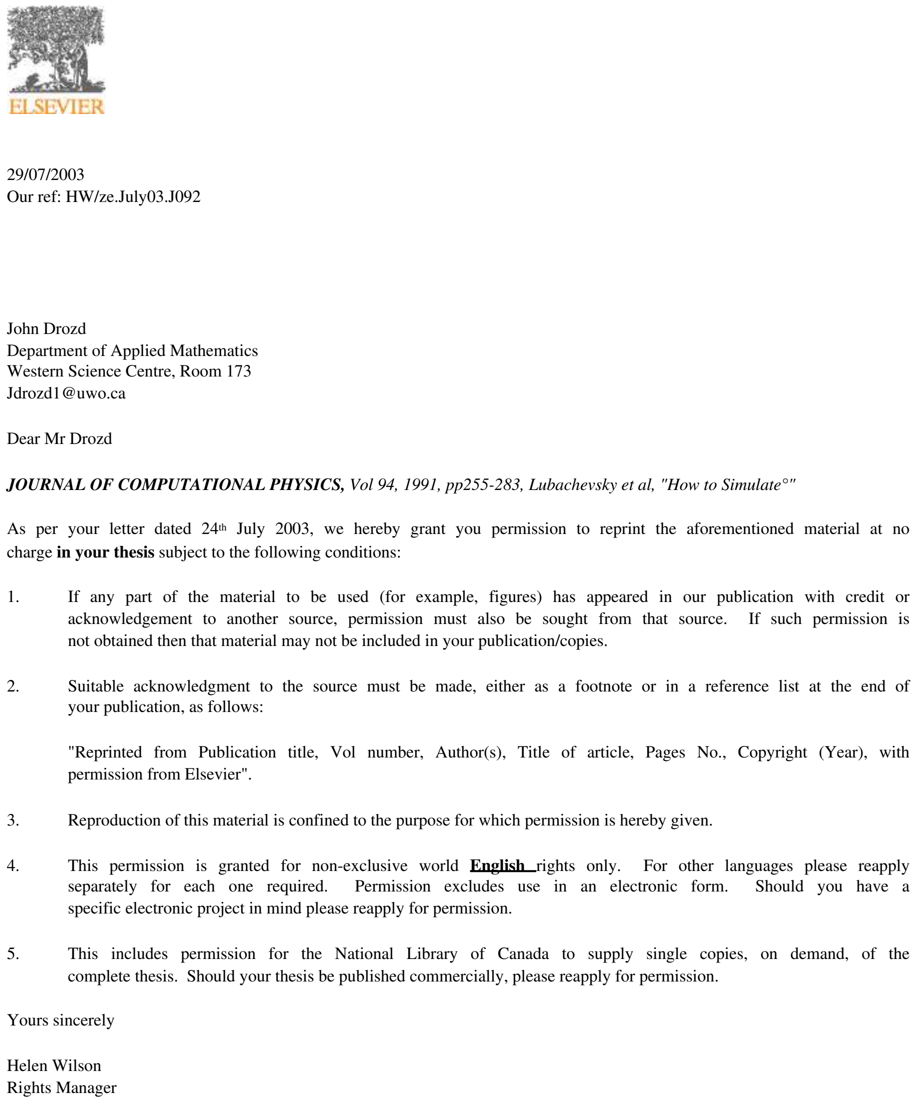
In accordance with ASME policy, this permission is contingent upon payment of a royalty fee of US$40 for 3 figures ($20 for the first figure/table, $10 thereafter). This is solely charged to non-authors of the requested ASME papers. We accept payments on all major credit cards such as: Visa, MasterCard, American Express, Discover, and Diners Club, or by check payable to ASME. Please send payment to the attention Meghna Patel, ASME Accounting, 22 Law Drive, Fairfield, NJ 07007, and indicate A/C# 104-78-0000-4303. Should you have any questions regarding payment form or transfer, please contact Ms. Patel; P: 973-244-2268, F: 973-882-4924; E: megpatel@asme.org.

Thank you for your interest in ASME publications.

Sincerely,

Beth Darchi
Copyrights & Permissions
V: 212-591-7700
F: 212-591-7292 E: darchib@asme.org

G.2 Letter from Elsevier Publishers



ELSEVIER

29/07/2003
Our ref: HW/ze.July03.J092

John Drozd
Department of Applied Mathematics
Western Science Centre, Room 173
Jdrozd1@uwo.ca

Dear Mr Drozd

JOURNAL OF COMPUTATIONAL PHYSICS, *Vol 94, 1991, pp255-283, Lubachevsky et al, "How to Simulate°"*

As per your letter dated 24th July 2003, we hereby grant you permission to reprint the aforementioned material at no charge **in your thesis** subject to the following conditions:

1. If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies.

2. Suitable acknowledgment to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

   "Reprinted from Publication title, Vol number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier".

3. Reproduction of this material is confined to the purpose for which permission is hereby given.

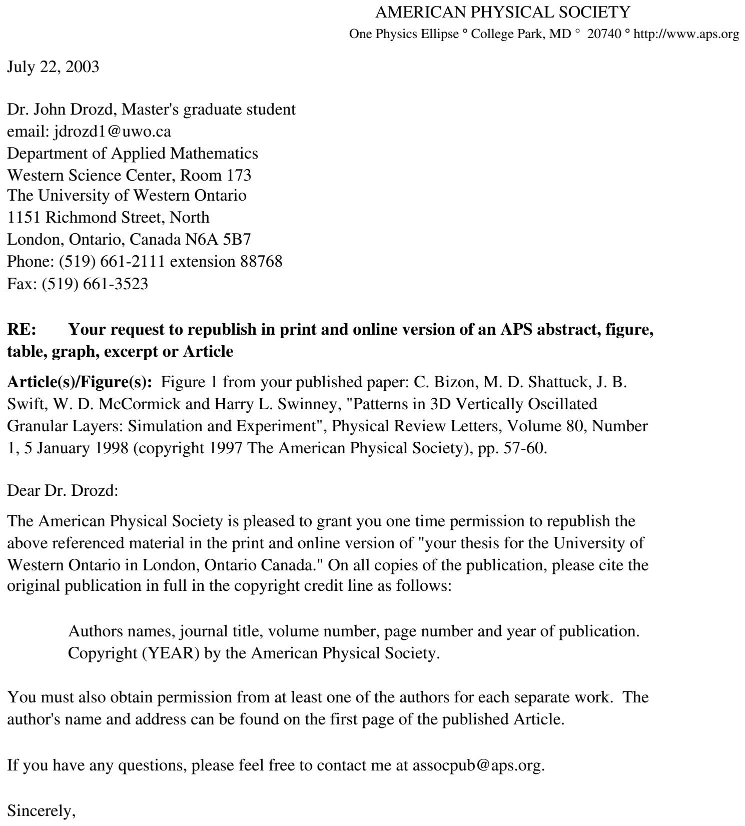
4. This permission is granted for non-exclusive world **English** rights only. For other languages please reapply separately for each one required. Permission excludes use in an electronic form. Should you have a specific electronic project in mind please reapply for permission.

5. This includes permission for the National Library of Canada to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission.

Yours sincerely

Helen Wilson
Rights Manager

G.3 Letter from American Physical Society



AMERICAN PHYSICAL SOCIETY
One Physics Ellipse ° College Park, MD ° 20740 ° http://www.aps.org

July 22, 2003

Dr. John Drozd, Master's graduate student
email: jdrozd1@uwo.ca
Department of Applied Mathematics
Western Science Center, Room 173
The University of Western Ontario
1151 Richmond Street, North
London, Ontario, Canada N6A 5B7
Phone: (519) 661-2111 extension 88768
Fax: (519) 661-3523

RE: Your request to republish in print and online version of an APS abstract, figure, table, graph, excerpt or Article

Article(s)/Figure(s): Figure 1 from your published paper: C. Bizon, M. D. Shattuck, J. B. Swift, W. D. McCormick and Harry L. Swinney, "Patterns in 3D Vertically Oscillated Granular Layers: Simulation and Experiment", Physical Review Letters, Volume 80, Number 1, 5 January 1998 (copyright 1997 The American Physical Society), pp. 57-60.

Dear Dr. Drozd:

The American Physical Society is pleased to grant you one time permission to republish the above referenced material in the print and online version of "your thesis for the University of Western Ontario in London, Ontario Canada." On all copies of the publication, please cite the original publication in full in the copyright credit line as follows:

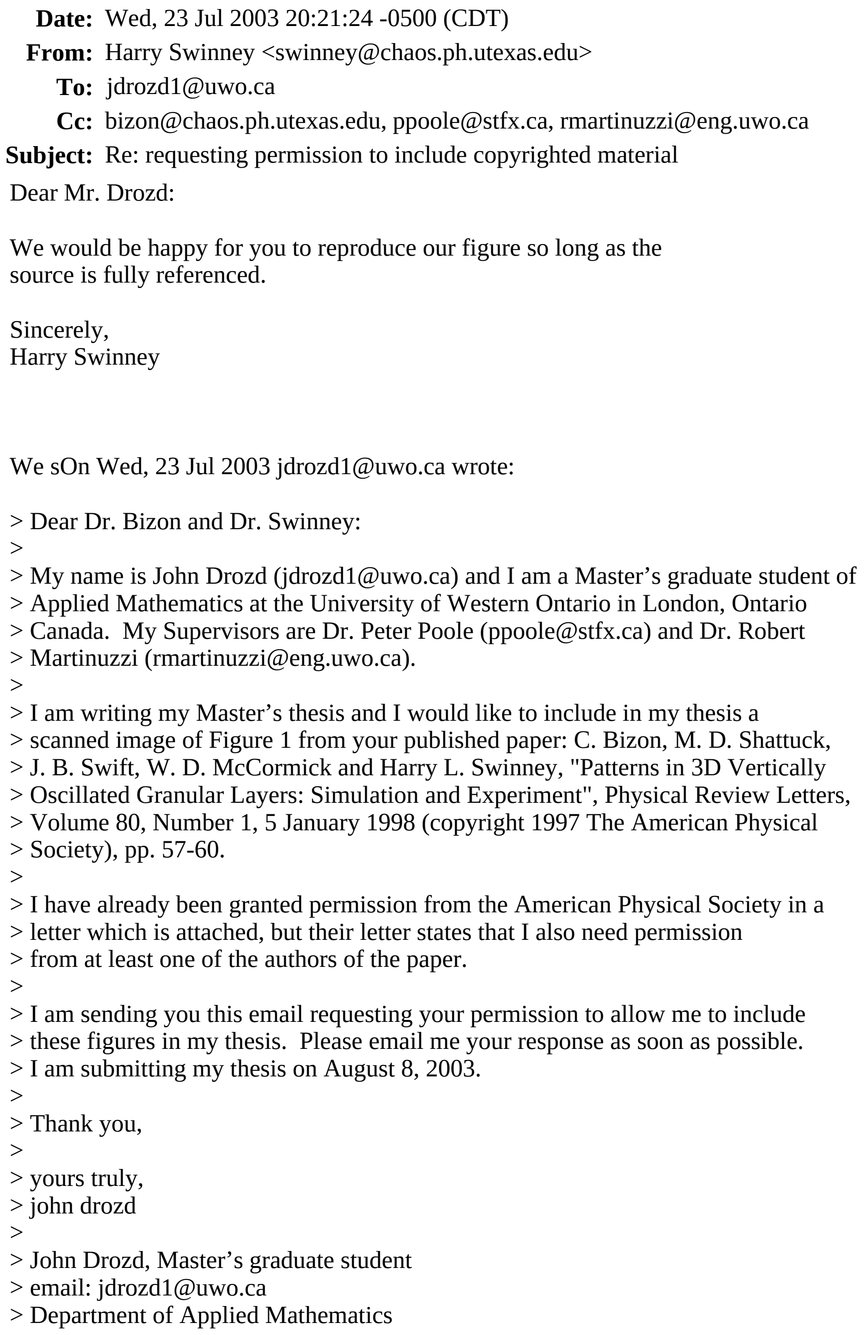
> Authors names, journal title, volume number, page number and year of publication.
> Copyright (YEAR) by the American Physical Society.

You must also obtain permission from at least one of the authors for each separate work. The author's name and address can be found on the first page of the published Article.

If you have any questions, please feel free to contact me at assocpub@aps.org.

Sincerely,

G.4 Letter from Dr. Harry L. Swinney



Date: Wed, 23 Jul 2003 20:21:24 -0500 (CDT)
From: Harry Swinney <swinney@chaos.ph.utexas.edu>
To: jdrozd1@uwo.ca
Cc: bizon@chaos.ph.utexas.edu, ppoole@stfx.ca, rmartinuzzi@eng.uwo.ca
Subject: Re: requesting permission to include copyrighted material

Dear Mr. Drozd:

We would be happy for you to reproduce our figure so long as the
source is fully referenced.

Sincerely,
Harry Swinney

We sOn Wed, 23 Jul 2003 jdrozd1@uwo.ca wrote:

> Dear Dr. Bizon and Dr. Swinney:
>
> My name is John Drozd (jdrozd1@uwo.ca) and I am a Master’s graduate student of
> Applied Mathematics at the University of Western Ontario in London, Ontario
> Canada. My Supervisors are Dr. Peter Poole (ppoole@stfx.ca) and Dr. Robert
> Martinuzzi (rmartinuzzi@eng.uwo.ca).
>
> I am writing my Master’s thesis and I would like to include in my thesis a
> scanned image of Figure 1 from your published paper: C. Bizon, M. D. Shattuck,
> J. B. Swift, W. D. McCormick and Harry L. Swinney, "Patterns in 3D Vertically
> Oscillated Granular Layers: Simulation and Experiment", Physical Review Letters,
> Volume 80, Number 1, 5 January 1998 (copyright 1997 The American Physical
> Society), pp. 57-60.
>
> I have already been granted permission from the American Physical Society in a
> letter which is attached, but their letter states that I also need permission
> from at least one of the authors of the paper.
>
> I am sending you this email requesting your permission to allow me to include
> these figures in my thesis. Please email me your response as soon as possible.
> I am submitting my thesis on August 8, 2003.
>
> Thank you,
>
> yours truly,
> john drozd
>
> John Drozd, Master’s graduate student
> email: jdrozd1@uwo.ca
> Department of Applied Mathematics

REFERENCES

[1] Ristow, G.H., *Pattern Formation in Granular Materials, Springer Tracts in Modern Physics* **64**, (Springer, New York, 2000).

[2] Wassgren, C. R., Brennen, C. E., Hunt, M. L., Vertical Vibration of a Deep Bed of Granular Material in a Container, *Journal of Applied Mechanics* **63**: pp. 712-719, 1996.

[3] Baran, O., Drozd, J.J., Poole, P.H., Sistla, P., Martinuzzi, R.J., Simulation of circulatory motion of granular matter in an agitated cylindrical bed, *to be submitted for publication.*

[4] Bizon, C., Shattuck, M. D., Swift, J. B., McCormick, W. D., Swinney, Harry L., Patterns in 3D Vertically Oscillated Granular Layers: Simulation and Experiment, *Physical Review Letters* **80** No. 1: 57-60, 1997.

[5] *General Kinematics Vibradrum Grinding Mills* brochure available on the internet at http://www.generalkinematics.com.

[6] Drake, T. G., Walton, O. R., Comparison of Experimental and Simulated Grain Flows, *Journal of Applied Mechanics* **62**: 131-135, March 1995.

[7] Luding, S., Clément, E., Rajchenbach, J., Duran, J., Simulations of pattern formation in vibrated granular media, *Europhysics Letters* **36**(4): pp. 247-252, 1996.

[8] Lubachevsky, B.D., How to Simulate Billiards and Similar Systems, *Journal of Computational Physics* **94**: 255-283, 1991.

[9] Bizon, C., Shattuck, M. D., Swift, J. B., Swinney, H. L., Transport coefficients for granular media from molecular dynamics simulations, *Physical Review E* **60**: 4340-4351, 1999.

[10] Goldman, D., Shattuck, M. D., Bizon, C., McCormick, W. D., Swift, J. B., Swinney, H. L., Absence of inelastic collapse in a realistic three ball model, *Physical Review E* **57**: R4831-4833, 1998.

[11] McNamara, S. and Young, W. R., Inelastic collapse in two dimensions, *Physical Review E* **50**: pp. R28-R31, 1994.

[12] McNamara, S. and Young, W. R., Inelastic collapse and clumping in a one-dimensional granular medium, *Physics of Fluids A* **4**: pp. 496-504, 1992.

[13] Goldsmith, W., *Impact* (Edward Arnold Ltd., London, 1960), p. 262-263, 267.

[14] Raman, C. V., The Photographic Study of Impact At Minimal Velocities, *Physical Review* **12**: pp. 442-447, 1918.

[15] Allen, M. P., Tildesley, D. J., *Computer Simulation of Liquids*, (Oxford University Press, New York, 1999), p. 101.

[16] Bizon, C., Shattuck, M. D., de Bruyn, J. R., Swift, J. B., McCormick, W. D., Swinney, H. L., *Journal of Statistical Physics* **93**: 449, 1998.

[17] Shattuck, M. D., Bizon, C., Umbanhowar, P. B., Swift, J. B., Swinney, H. L., in *Powders & Grains 97* (Balkema, Rotterdam, 1997).

[18] Press, William H., Teukolsky, Saul A., Vetterling, William T., Flannery, Brian P., *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition (Cambridge University Press, New York, 1992), p. 353.

[19] Hughes, W. F., Brighton, J. A., *Schaum's Outline Series in Theory and Problems of Fluid Dynamics* (McGraw-Hill Book Company, New York, 1967), p. 106.

[20] Grue, J., Gjevik, B., Weber, J. E., *Rayleigh-Bénard Convection with Weak Shear Flow: Absolute and Convective Instabilities* in *Waves and Nonlinear Processes in Hydrodynamics* (Kluwer Academic Publishers, Boston, 1996), p. 303.

[21] Velard, M. G., Radyadour, Kh. Z., *Interfacial Phenomena and The Marangoni Effect* (Springer-Verlag, New York, 2002), pp. 89, 123.

[22] Mullier, M., Tüzün, U., Walton, O. R., A single-particle friction cell for measuring contact frictional properties of granular materials, *Powder Technology* **65**: 61-74, 1991.

[23] Information on the *Satec* Modulus of Toughness measuring equipment is available at http://fp.uni.edu/rrttc/mts/StiffnessTouphness.asp

[24] Beer, Ferdinand P., Johnston, E. Russell Jr., *Mechanics of Materials*, (McGraw-Hill Book Company, New York, 1981), p. 473.

[25] Pöschel, T., Luding, S., (eds.), *Granular Gases* (Springer, New York, 2001).

[26] Engeln-Müllges, G., Uhlig, F., *Numerical Algorithms with C*, (Springer, New York, 1996), p. 241.

[27] Information on force chains or stress chains is available at http://www.phy.duke.edu/˜dhowell/research.html

[28] Harr, M. E., *Mechanics of Particulate Media*, (McGraw-Hill, Book Company, New York, 1977), p. 291.

[29] Mehta, A., *Granular Matter: An Interdisciplinary Approach*, (Springer-Verlag, New York, 1994), p. 276.

[30] Walton, O. R., in *Particulate two-phase flow*, edited by M. C. Roco (Butterworth-Heinemann, Boston, 1993), p. 884.

[31] Orfanidis, S. J., *Introduction To Signal Processing*, (Prentice-Hall, Englewood Cliffs, New Jersey, 1996), pp. 30-31.

[32] Mase, George E., *Schaum's Outline Series in Theory and Problems of Continuum Mechanics* (McGraw-Hill Inc., New York, 1970), p. 147.

[33] Chung, T. J., *Applied Continuum Mechanics* (Cambridge University Press, New York, 1996), pp. 24, 51.

VITA

NAME: John J. Drozd

PLACE OF BIRTH: Brno, Czech Republic

YEAR OF BIRTH: 1964

POST-SECONDARY EDUCATION AND DEGREES:
Ryerson University
Toronto, Ontario
1985-1989 B. Technology (Architectural Science)

University of Western Ontario
London, Ontario
1996-2001 Diploma in Honors Standing
(Applied Math and Computer Science)

HONOURS AND AWARDS:
Ryerson University
Dean's List
Awarded 1986, 1987, 1988

Ryerson University
Highest Grades in Project Management
Awarded 1987

Natural Sciences and Engineering Research Council
Post Graduate Scholarship A
Awarded 2001

RELATED WORK EXPERIENCE:
Teaching Assistant
The University of Western Ontario
London, Ontario, Canada
2001 - 2003

PUBLICATIONS:

Saujani, S., Drozd, J., and Mallier, R., Nonlinear evolution of singular disturbances to a $tanh^3$ mixing layer, *Australian & New Zealand Industrial and Applied Mathematics Journal, Series B, ANZIAM J.* 43 (2002), pp. 409-427.

Valluri, S. R., Drozd, J. J., Chishtie, F. A., Biggs, R. G., Davison, M., Dhurandhar, Sanjeev V., and Sathyaprakash, B. S., A Study of the Gravitational Wave Form From Pulsars, *Classical and Quantum Gravity, Proceedings of the 4th Edoardo Amaldi Conference on Gravitational Waves*, Perth, Western Australia, 8-13 July 2001, Vol. 19 No. 7 (April 2002), pp. 1327-1334.