A DISCRETE ELEMENT METHODOLOGY FOR THE ANALYSIS OF COHESIVE GRANULAR BULK SOLID MATERIALS

by

Liz Ivoneth Del Cid

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mechanical Engineering).

Golden, Colorado

Date _____

Signed: _____
Liz Ivoneth Del Cid

Signed: _____
Dr. Graham Mustoe
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Gregory Jackson
Professor and Department Head
Department of Mechanical Engineering

ABSTRACT

In bulk handling applications, such as conveying and storage, understanding the effect cohesion has upon the flow-ability of particulate systems at the macroscopic scale is crucial in increasing the avenues of operation unit design improvements and handling scenarios of industrial operational units. This research provides a better understanding of the role cohesion has on the flow-ability of macro materials through the development, implementation and application of a macroscopic elasto-plastic adhesive (MEPA) contact model within an open source general purpose Discrete Element Method (DEM) computer code.

This dissertation outlines the development of a DEM contact law which can model stress history dependent strength behavior of cohesive particulate systems and predict its effects upon the particulate flow. The research tasks in this work are focused in three major areas: 1) cohesive function applications from powders to bulk solids, 2) modeling stress history dependency of cohesive strength, and 3) the prediction of flow properties in test applications that are comparable to experimental results.

For a given bulk handling application, adequately capturing the DEM simulated behavior of cohesive solids is crucial when evaluating its handle-ability. A number of DEM micro mechanically-based cohesive contact laws are available; however, these do not model the stress history dependent behavior physically observed in particulate bulk solids. A study of these micro mechanically-based cohesive models revealed that most of these models are focused on simulating the effects of cohesion in powder systems. A major shortcoming of these micro mechanically-based models is the iterative parametric scaling needed to represent cohesive-like behavior.

When simulating the handling difficulties caused by cohesion, it is apparent that modelling stress history dependency is crucial in consolidated materials with high cohesive strength. This investigation proposed a DEM history dependent particle-particle MEPA

contact model that accounts for both elastic and plastic contact deformations and adhesive attractions. The MEPA model applied herein is a three branched non-linear contact model that simulates the virgin compaction loading, unloading/reloading and adhesion behavior of a particulate solid.

The culmination of this research is a general purpose DEM high performance computer code, LIGGGHTS that includes an enhanced capability for material flow simulations of highly cohesive particulate systems for modeling industrial bulk solids handling applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

General Nomenclature

Greek Letters

area of contact . . . . . $A_{contact}$

surface area of the annulus . . . . . A_d

radius of the contact zone . . . . . a

critical damping constant . . . . . C_r

macroscopic cohesion c

distance between centers of impacting particles . . . . . d

young's modulus of particle 1 . . . . . E_1

young's modulus of particle 2 . . . . . E_2

effective Young's Modulus . . . . . E^*

coefficient of restitution . . . . . e

cohesive force . . . . . F

external force on particle i . . . . . $F_{ext,i}$

force vector on of particle i . . . . . F_i

force on particle i by particle j . . . . . F_{ij}

normal force . . . . . $F_n,\ f_n$

force on particle . . . . . $F_{particle}$

tangential force . . . . . $F_t,\ f_t$

viscous damping torque flag . . . . . f

normal force contact . . . . . f_0^n

cohesive force contribution by DMT cohesive model . . . . . f_{DMT}

cohesive force contribution by JKR cohesive model . . . . . f_{JKR}

force of attraction . . . . . $f_{attraction}$

hysteresis force . . . . . f_{hys}

maximum elastic to plastic loading force . . . . . f_{max}

minimal force of attraction . . . . . f_{min}

effective shear modulus . . . . . G^*

particle stiffness . . . . . k

virgin loading/compacting stiffness . . . . . k_1

unloading/reloading stiffness . . . . . k_2

cohesive stiffness . . . . . k_{adh}, k_{coh}

particle stiffness in the normal direction . . . . . k_n

particle rotational stiffness . . . . . k_r

particle stiffness in the tangential direction . . . . . k_t

total rolling resistance . . . . . M_r

viscous damping torque . . . . . M_r^d

rolling spring torque . . . . . M_r^k

limiting spring torque . . . . . M_r^m

particle mass . . . . . m

effective mass . . . . . m^*

mass of particle 1 . . . . . m_1

mass of particle 2 . . . . . m_2

DEM particle equivalent mass . . . . . m_{eqv}

mass of particle i . . . . . m_i

variable integer . . . . . N

unit vector in the normal direction . . . . . $\mathbf{n}$

number of particulate fines, index parameter . . . . . n

radius of particle . . . . . R

effective radius . . . . . R^*

radius of particle 1 . . . . . R_1

radius of particle 2 . . . . . R_2

radius of particle fine . . . . . r

DEM particle equivalent radius . . . . . r_{eqv}

radius of particle i . . . . . r_i

radius of particle j . . . . . r_j

mean radius of the annulus . . . . . r_{mean}

torsion . . . . . T

unit vector in the tangential direction . . . . . $\mathbf{t}$

normal relative velocity . . . . . $\mathbf{v}_n$

capillary force liquid bridge contact radius . . . . . x_{c1}, x_{c2}

position vector of particle i . . . . . $\mathbf{x}_i$

position vector of particle j . . . . . $\mathbf{x}_j$

acceleration vector of particle i . . . . . $\ddot{x}_i$

capillary force liquid bridge radius . . . . . y_0

viscous damping in the normal direction . . . . . γ_n

surface tension . . . . . γ_{sur}

tangential viscous damping . . . . . γ_t

angle of effective internal friction . . . . . ϕ_{sf}

capillary force pressure difference . . . . . $\triangle p$

increment of time . . . . . $\triangle t$

incremental relative rotation displacement . . . . . $\triangle \theta_r$

separation distance . . . . . δ

equilibrium separation distance . . . . . δ_0

maximum normal overlap . . . . . δ_{max}

minimum cohesive separation distance . . . . . δ_{min}

separation distance in the normal direction . . . . . δ_n

plastic contact deformation . . . . . δ_p

debonding distance . . . . . $\delta_{rupture}$

separation distance in the tangential direction . . . . . δ_t

rate of displacement in the normal direction . . . . . $\dot{\delta}_n$

rate of displacement in the tangential direction . . . . . $\dot{\delta}_t$

safety factor . . . . . ε

capillary force filling angles . . . . . ς_1, ς_2

volume of the liquid bridge, capillary force wetting angle . . . . . θ

relative rolling angular velocity . . . . . $\dot{\theta}_r$

rolling friction coefficient . . . . . μ_r

sliding friction coefficient . . . . . μ_s

Poisson's ratio of particle 1 . . . . . ν_1

Poisson's ratio of particle 2 . . . . . ν_2

principal stresses, consolidation stress . . . . . $\sigma_1,\ \sigma_2,\ \sigma_3$

yield strength . . . . . $\sigma_c,\ \sigma_y$

normal stress . . . . . σ_n

preconsolidation normal stress . . . . . σ_{pre}

shear stress . . . . . σ_t

shear stress . . . . . τ

preconsolidation shear stress, steady-state shear stress . . . . . τ_{pre}

internal angle of friction . . . . . φ

natural frequency . . . . . ω_0

LIST OF ABBREVIATIONS

Control Force . . . . . CF

Computational Fluid Dynamics and Discrete Element Methods . . . . . CFDEM

Colorado School of Mines . . . . . CSM

Coefficient of Rolling Friction . . . . . CoR

Coefficient of Static Friction . . . . . CoS

Discrete Element Method . . . . . DEM

Derjaguin- Muller-Toporov model . . . . . DMT

Johnson-Kendall-Roberts . . . . . JKR

Large-scale Atomic/Molecular Massively Parallel Simulator . . . . . LAMMPS

LAMMPS improved for general granular and granular heat transfer simulations . . . . . LIGGGHTS

Moisture Content . . . . . mc

Macroscopic Elastic-Plastic Adhesive . . . . . MEPA

Overland Conveying Company, Inc. . . . . OCC

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Graham Mustoe for his continued support and encouragement throughout the process of this research. I would like to also thank the Overland Conveyor Co., Inc. team for their technical expertise and support during this degree. To my committee, I would like to express my gratitude for keeping me on track and opening new doors throughout my studies.

I sincerely extend my gratitude to Briana, Matthew, David and Alex for their infinite support, encouragement, and patience over the years. Without their assistance and encouragement, I would not have made it through the endless literature and data management to this end. I gladly call them my family and a prized alliance of creative thought, friendship and cheer.

And most importantly, I would like to thank my family. Their love, patience and optimism has pushed me forward and molded my very structure. They have instilled in me a love for knowledge and exploration. I hope my humble successes have made them proud and hopeful that I will continue to develop with the unlimited enthusiasm of a child into old age.

To my lovely sister, Joselyn. May you always find the will to win, the desire to succeed and the urge to reach your full potential. I love you always and always will.

CHAPTER 1
INTRODUCTION

Characterization of bulks solids is required for the reliable design and operation of industrial processes related to the physical storage and handling units of granular materials. The movement of granular matter which exhibits "sticky" or cohesive phenomena gives rise to a wide variety of different flow behaviors. The changing contact networks and stress distributions produce large fluctuations of forces and reorganization of the contacts. The added influence of cohesive forces creates a challenge in predicting the macroscopic flow behavior of a material. Numerical simulation DEM studies conducted with the industry accepted cohesive models show discrepancies between the bulk responses observed in physical tests and those predicted in numerical simulations. This dissertation describes the development of a DEM particle-particle cohesive contact force model to better simulate the cohesive strength of particulate materials.

1.1 Background

Granular bulk solids are materials that exhibit both the properties of a solid and a fluid [1]. Brown and Richards [2] define bulk solids as: "An assembly of discrete solid components dispersed in a fluid such that the constituents are substantially in contact with near neighbors. This definition excludes suspensions, fluidized beds and materials embedded in a solid mixture." Some examples of bulk solids are coal, sand, ore, mineral concentrate, and crushed oil shale [2]. Bulk solids can be defined numerically through a discrete assembly of interacting particles known as a contact network.

The DEM numerical technique treats granular bulk solid particles as a system of interacting particles. Interactions are considered as forces and couples acting on pairs of individual particles and are expressed through interaction force laws. Each interaction follows a soft

contact approach where an overlap of particles is accepted. The soft contact approach intends to model the deformation of the interacting bodies at a contact point. The particles are treated as rigid bodies and the interactions between them governed by the unilateral contact, energy dissipation by friction and inelastic collisions. The equations of motion as well as the force-displacement relations require a time-discretized form. The implementation is an explicit method accounting for the physical effects of the material [3].

For frictional-cohesive materials, a common issue is the storage and handling difficulties caused by cohesion. Moisture content in bulk solids can slow the flow of material and under certain conditions may stop the flow entirely. This moisture content is characterized as apparent cohesive strength and is dependent on the consolidation stress a material has experienced over time. In macro-modeling, it is important to better understand the macroscopic flow behavior due to cohesive strength in critical state flows and shearing bands. A number of cohesive models are available to simulate cohesive flows using DEM. The more commonly used being the Johnson-Kendall-Roberts, (JKR), and capillary force models [4–6]. However, these cohesive models do not accurately capture the effects of cohesion on the flow behavior observed in bulk solids.

In order to evaluate the flow-ability of cohesive materials, the stress history needs to be considered. For example, high storage stresses in a silo can lead to high cohesive strength of the handled material and may cause blockages such as ratholing or arching near the outlet during discharge (Figure 1.1) [7] . An arch can form over the outlet capable of supporting the entire contents of the silo above when the material has enough strength. Extreme methods of breaking the arch may be required to initiate flow; such methods include sledgehammers, vibrators, and air blasters. The second no flow condition occurs when a stable rathole forms. Some material typically discharges through a preferential flow channel. The formations of these channels depend on the degree of consolidation experienced by the material. In general, the material strength increases with consolidating pressures. However, when the strength is sufficient, the flow channel empties out resulting in a stable rathole and no flow.

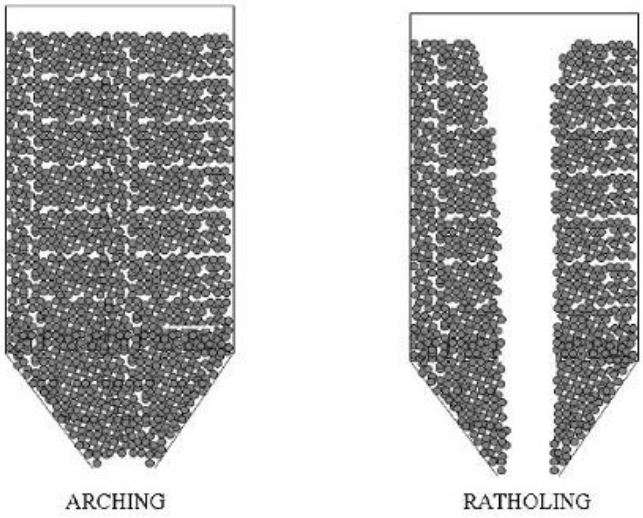
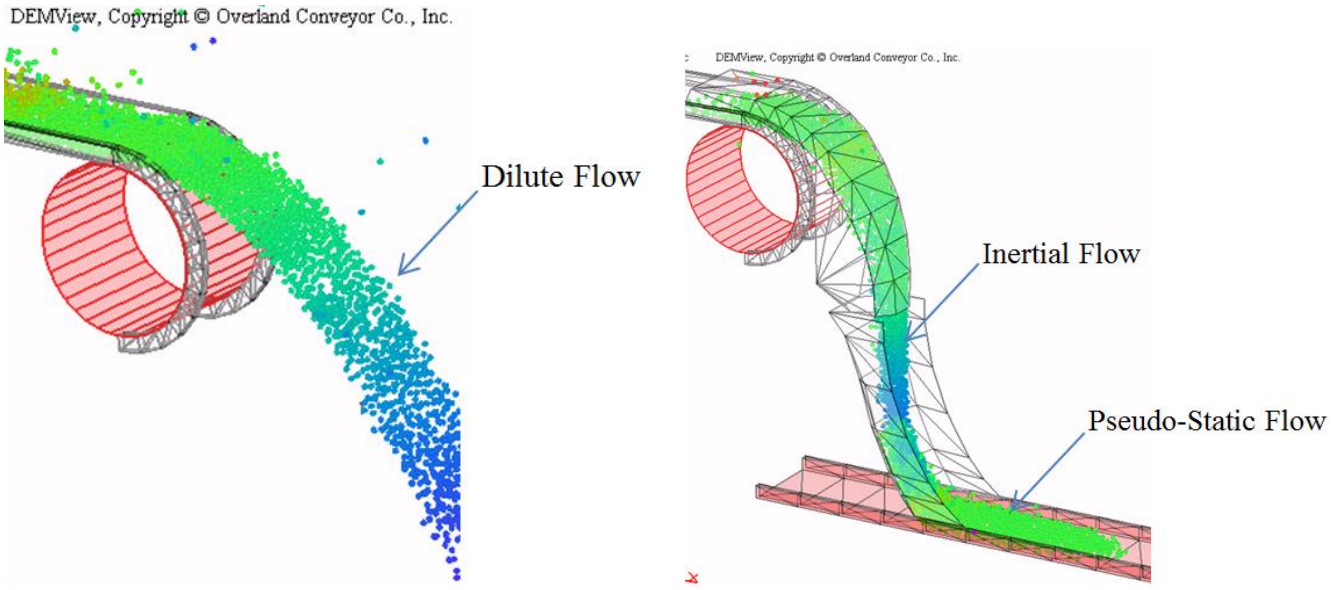


Figure 1.1: Two conditions under which solid flow can stop completely from [7].

The mechanical modeling of particulate systems is technically challenging for a number of reasons: particle shape, non-uniform particle size distribution, moisture effects, and non-linear mechanical interactions including dissipative behavior. Additional complexity is also caused by the variation in particle packing within the particle system and flow behavior. The flow regimes in dense granular flows can generally be classified into three different types: the two extreme regimes of inertial flows and quasi-static shearing flows, and transitional flows [8]. Dense and confined particulate assemblies in extremely slow shear flows are described as solids that abide by elasto-plastic rate independent constitutive laws [9–11]. In the dilute limit (loose contact between particles) the particles interact through instantaneous and uncorrelated collisions. The flow of slightly dissipative particles can be described by the kinetic theory of dense gases [12, 13]. Figure 1.2a and Figure 1.2b show the different flow regimes of granular bulk material.

The flow regimes of transitional flows between quasi-static and inertial flows are still not well understood. Specifically, the development of granular flows from the initially deposited state to fully developed flows with strong interactions is difficult to predict [14]. Nevertheless,



(a) Dilute flow.

(b) Inertial flow and pseudo-static flow.

Figure 1.2: Systems with different flow regimes.

to solve these flow problems, accurate numerical modeling of granular flows by DEM is needed. This numerical model allows us to investigate the particle contact behavior in the different regimes.

Using DEM to simulate the flow-ability of particulate materials requires prior knowledge of all the numerical model parameters. DEM has the capacity of modelling materials at microscopic levels in the different flow regimes and analyzing multiple interacting bodies undergoing large displacements and rotations. It has the ability to capture the phenomena that relates to the particulate nature of granular mediums. These granular solids range in size from 0.1 mm to 3 mm [2]. At the lower end of the size scale, DEM models employing existing cohesive models have the ability to simulate the interactions between fine particles. However, it is very difficult to simulate a significant volume of fine particles with a high volume bulk flow due to the immense computational effort required. To overcome the inability of modeling a vast number of fine particles, the physical input parameters are scaled and applied to a system of larger DEM macro particles. At the macro scale, these models underestimate the

forces of cohesion. This observation comes from the process of simulating cohesive materials. These systems undergo a number of iterative runs until the appropriate cohesive flow behavior is observed within the material. In order to better represent macro particles, the existing DEM cohesive models apply geometric particle scaling to equate the cohesive forces of the micro scale to the bulk particulate scale material. In this dissertation, a DEM history dependent contact model that accounts for both elastic and plastic contact deformations and adhesive attractions is developed for the simulation of the mechanical behavior of bulk solid materials. The material range will cover solids up to 10 mm in size. This range includes most of the materials used in flow property testing laboratory experiments. Henceforth, whenever the term granular material is used, it will refer to this range.

1.2 Purpose

The bulk behavior of a particulate system depends on the collective interactions of individual particles, and hence a realistic representation of a cohesive solid is key to characterizing bulk material response to storage and handling [15]. Using a model that can represent cohesive strength history, allows scenarios of handling difficulties caused by cohesion to be more readily identifiable. The research conducted provides a method of evaluating these conditions for the purpose of providing a tool that can be used within the engineering design of industry storage and handling systems of frictional-adhesive particulate solids.

1.2.1 Research Objectives

The principal aim of this thesis is to create a DEM cohesive model for the study of mechanical flow behavior of bulk solid material. Specifically, the focus is on the development of a macroscopic elasto-plastic adhesive contact model and the numerical implementation within a DEM computer code. It targets an improved understanding of the cohesive strength observed in macro scale bulk solid material flow behaviors. The desired numerical enhancements provide an improved discrete element cohesive contact force model to better simulate the effect of shearing properties on the flow-ability and handle-ability of particulate systems.

Its design simulates the Mohr-Coulomb shear failure envelopes measured in physical flow property tests as comparative cohesive strength testing. This macro-mechanical cohesive model is a novel design across the bulk material handling industry and increases the avenues of operation unit design improvements and handling scenarios available.

1.3 Statement of Work

Flow-ability is a common description of the ease or difficulty with which a bulk solid will flow. The test programs provide comparative cohesive strength testing to determine the effects of moisture and identify the conditions of minimum flow-ability. For numerical simulations, a number of cohesive models are available. These rely on the measured particle properties and micro-mechanics to simulate the bulk materials. As applied to bulk solids, cohesion models simulating macro-mechanical behavior proved to be an unexplored field. Through the course of this research, a DEM contact cohesive model was developed to bridge the gap between the physical material testing and the effective macroscopic flow behavior of cohesive materials.

The literature revealed gaps in the understanding of macroscopic cohesion. These observations established the salient features needed in an improved model and lead to the development of an elasto-plastic adhesive contact algorithm. The cohesive force model was applied and tested using the LIGGGHTS software package. LIGGGHTS is an open source DEM particle simulation software designed for industrial applications by CFDEMproject as an added feature to LAMMPS, a molecular dynamics simulator distributed by Sandia National Laboratories, a US Department of Energy Laboratory [16]. The following section attempts to breakdown and place the research in context.

1.3.1 Scope of Work

The completed tasks provide the proof of concept and a starting platform for the additional research to be performed for industry. The approach for this completed work is available pictorially in Figure 1.3. The following objectives summarize the milestones ac-

complished.

1. *Study of the dominant cohesive models used to simulate cohesive material failure in bulk solids.* An assessment of the existing capillary force model and JKR numerical cohesive models and geometric scaling was performed for its suitability to study bulk wet granular flow.

2. *Development of a macro-mechanical cohesive contact model.* The numerical Macro Elasto-Plastic Adhesion (MEPA) Contact DEM model capable of simulating the mechanical behavior of physical material flow and failure events was created.

3. *Development of the iterative MEPA cohesive contact algorithm.* The algorithm of the MEPA cohesive contact model was developed for all particle interactions along and bounded by the MEPA cohesive bounding curves. The discretized algorithm was then developed for the Aun partition's parallel LIGGGHTS platform of the supercomputer BLUEM at Colorado School of Mines (CSM).

4. *Implementation and validation of the MEPA cohesive contact model.* The cohesive model was coded in LIGGGHTS as part of their granular applications package. A particle drop test was performed to validate the bounding branches of the MEPA cohesive model and provide an improved simulation of material mechanical behavior.

5. *Modeling material failure via the simulation of physical flow property tests.* The flow property test results of copper ore was used to validate the MEPA cohesive model by simulating the observed compressive shearing failure states.

1.3.2 Contributions

The work completed in this dissertation can be summarized by the following contributions:

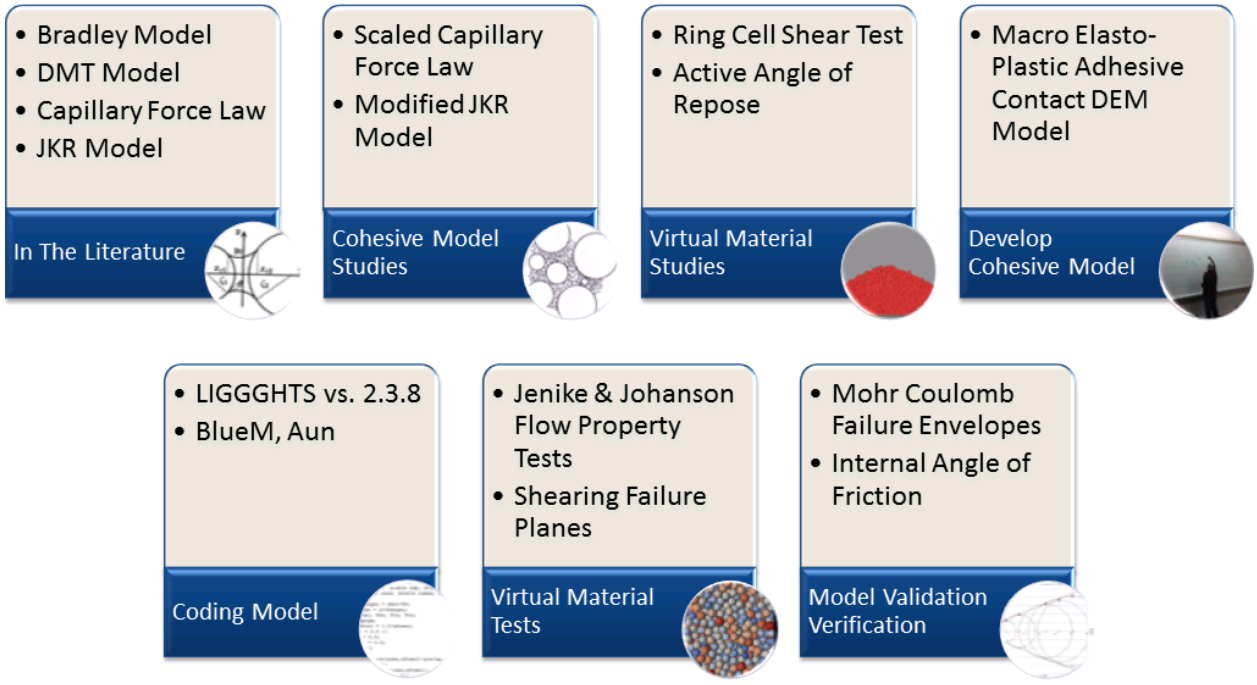


Figure 1.3: Thesis road map for completion.

- *Identification of deficiencies in existing numerical cohesive models.* The studied models provide accurate micro-mechanical results for powder technologies but are alone insufficient to simulate the mechanical behavior of bulk material flow. Even with scaling the models studied illustrated the current limits of cohesive flow modeling.

- *Development and implementation of the MEPA cohesive model.* The numerical MEPA cohesive model capable of simulating physical material flow behavior as part of the granular package of an academic open source code for the study of cohesive granular bulk solid materials. It illustrates the macroscopic cohesive behavior of bulk solids from a macroscopic cohesive model rather than from microscopic cohesive contact laws.

- *Validation of the methodology by simulation of the mechanical behavior of physical data.* The MEPA cohesive model shows positive contributions to the simulation of bulk solid mechanical flow behavior.

CHAPTER 2

DESCRIPTION OF COHESIVE MODELS USED IN PARTICULATE SOLIDS

This research seeks to provide a discrete model and procedure for numerically modeling cohesive bulk solid materials. Plausible solutions decompose cohesive attraction to capillary force bonding or nanoscale attraction forces dominated by fines. In the area of bulk solids, effective cohesive flow behaviors or macroscopic cohesion is an area little explored. The discrete model developed seeks to provide a solution to bulk solid cohesion by employing an elastic, plastic and adhesive model that can perform at the effective cohesion values measured in physical flow property tests.

Through the initial research developed herein, three main concepts are covered: the role cohesion plays in flow behaviors in particulate material, the discretized model through which cohesion is applied, and the main cohesive contact models in use today.

2.1 The Role of Cohesion in Particulate Materials

Cohesive particulate systems show an unusual flow behavior that can be quantified by macroscopic bulk properties. However, these macroscopic properties are controlled by microscopic contact forces and torques and present a challenge in performing realistic quantitative and predictive simulations.

The cohesive strength of a bulk material is a function of its past consolidation stresses. This cohesive stress history in the material presents a common issue affecting the storage and handling of bulk solids. For example, ratholing problems can occur in silo discharge when high stresses during storage result in high material cohesive strength as seen in Figure 1.1. A rathole formed when the material cannot empty completely under the force of gravity alone inducing segregation of the material [17]. In these cases, the discharge behavior differs from the classical mass-flow discharge and transitions of the funnel flow regimes.

DEM has been used to simulate the flow of bulk materials subjected to various loading and unloading conditions. However, the commonly used adhesion models have difficulty predicting the stress history dependent behavior that affects the flow-ability of cohesive solids [17–19]. Cohesive particulate materials have an ability to resist external tensile stresses. This ability enhances the shear strength that is used to assess the cohesion of particulate solids. The cohesive attraction forces between particles manifest themselves through the resistance to separation, shear and rolling between two particles.

2.1.1 Definition of Cohesive Forces

Cohesive forces stem from attraction interactions between particles and can typically be classified into three sources of cohesion: adhesion, capillarity and cementation [3]. Adhesion refers to the attraction interaction between particles of different materials while cohesion to same material interactions. In this study, interactions between particles are limited to surface interactions. These interactions include contacts such as physico-chemical interactions with very short attraction ranges or through solid or liquid bridges at the contact point. Electrical forces such as van der Waals forces are considered negligible because, at the bulk scale, gravitational forces dominate the interactions in the material flow.

The cohesive forces between contacting particles form an association with the contacting normal force and the particle overlap or separation δ_n. Upon the application of a tensile force between the particles, the adhesive force resists separation in the normal contact direction and for a small separation distance the bond still holds. Figure 2.1 shows two particles in contact through loading and unloading [3]. During the loading phase, the contacting particles load elastically or plastically. As they unload, the cohesive force resists the tensile force or the force of separation seen in Figure 2.1b. The distance at which the cohesive bond is broken differs from the distance from which the cohesive contact bond is formed. The distance at which the cohesive bond breaks leads to a hysteresis phenomenon represented as the energy loss per unit area to break the bond in Figure 2.1d. If contact is maintained and the cohesive bond unbroken, as material packs the cohesive bonds increase in strength over

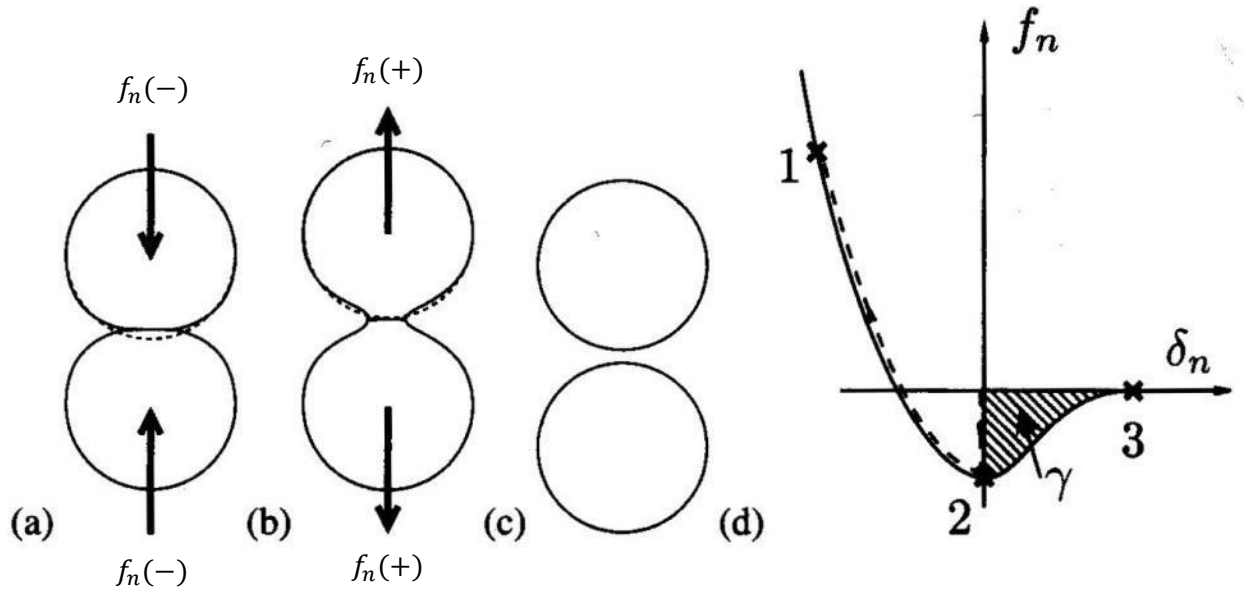


Figure 2.1: Formation of cohesive contacts from [3]. a) Initiation of a cohesive contact. b) Tensile strength due to the presence of cohesion. c) Failure of the cohesive bond. d) The evolution of the normal force f_n as a function of $\delta_n$where γ represents the energy per unit area to break the cohesive contact.

time. This is the cohesive stress history that presents a common issue affecting the storage and handling of bulk solids.

2.1.2 Description of Effective Macroscopic Cohesion

The yield strength of the macro-scale cohesion of particulate materials can be described by Coulomb cohesion. The Mohr-Coulomb failure criterion is derived from the graphical representation of the stress conditions of the material. The material failure is represented by the linear envelope that is obtained from a plot of the material's shear strength versus the applied normal stress and is described by equation 2.1:

$$|\sigma_t| \;=\; \sigma_n \tan \varphi \;+\; c \tag{2.1}$$

Equation 2.1 divides the mechanical strength of the material into the angle of internal friction φ and the macroscopic cohesion of the material c [10]. Here σ_n and σ_t are the normal and

tangential stresses in the material and where a positive (+) value of the normal stress σ_n is considered to be compressive.

The angle of internal friction φ and the effective cohesion c, can be determined with physical tests of shear, compression or tension. Particulate materials are typically tested under compressive loading. In a uniaxial compression test, the yield strength, σ_y, of the material can be derived by:

$$\sigma_y = \frac{\cos \varphi}{1 - \sin \varphi} c \tag{2.2}$$

Figure 2.2 provides a graphical representation of Mohr-Coulomb's failure criterion where the parameters φ and c characterize the strength of the material [3]. The straight sloped line represents the linear failure envelope that is obtained from the shear strength of a material at a given stress state. These tests are used to predict the flow behavior of bulk material and aid in the design of chutes, hoppers and silos.

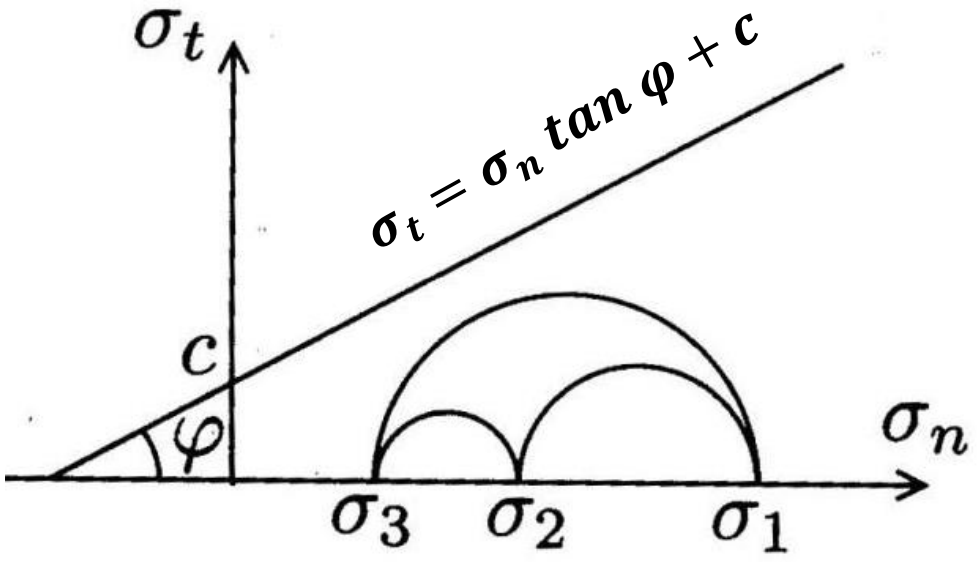


Figure 2.2: Mohr-Coulomb Criterion from [3].

2.2 The Non-linear Contact Law of The Discrete Element Model

In granular mechanics, where particles are moving large distances over time, the discrete element method is most commonly used. Cohesive models are discretized and applied to each grain alongside of Newton's Second Law. The basic mechanics of DEM are described below starting with Newton's Second Law of Motion.

The second law of motion pertains to the behavior of objects under unbalanced forces. The acceleration of these bodies is dependent on the mass and forces acting upon the object. The net force on the object is described by equation 2.3

$$m_i \ddot{\mathbf{x}} = \mathbf{F}_i, \qquad i = 1, \ldots, N \tag{2.3}$$

where N is the number of particles in the system, m_i is the mass of particle i, $\ddot{\mathbf{x}}$ is the acceleration of i and $\mathbf{F}_i$ is the force exerted on that particle. This method is flexible in that the forces $\mathbf{F}_i$ contain the particle-particle interactions as well as the external forces. These force can describe gravity or confining forces due to walls of a container as well as additional effects due to adhesion, cohesion, bond formations and temperature changes. Therefore, we can write:

$$\mathbf{F}_i = \sum_{j=i} \mathbf{F}_{ij} + \mathbf{F}_{ext,i} \tag{2.4}$$

where $\mathbf{F}_{ij}$ is the force exerted by contacting particle j on particle i and the external forces, $\mathbf{F}_{ext,i}$, are most often the force of gravity experienced by particle i. These contact forces are decomposed between the particles into two components:

$$\mathbf{F}_{ij} = F_n \mathbf{n} + F_t \mathbf{t} \tag{2.5}$$

where F_n and F_t are the components of the normal and tangential unit vector ($\mathbf{n}$, $\mathbf{t}$) and represent the tensile/compressive and shearing directions of contact.

When two rigid particles come into contact they do not undergo simulated shape deformation. Instead, the particles rigidly overlap and the interaction force is calculated as a function of that overlap. Figure 2.3 shows two contacting particles. The separation δ_n used to calculate the normal force is:

$$\delta_n = |\mathbf{x}_i - \mathbf{x}_j| - r_i - r_j \tag{2.6}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the centers of the particles and r_i and r_j are their respective radii. When the separation $\delta_n > 0$, the two particles are not in contact and there is no interaction. When $\delta_n < 0$, the two particles are said to overlap and a repulsive contact force is computed that pushes the particles apart.

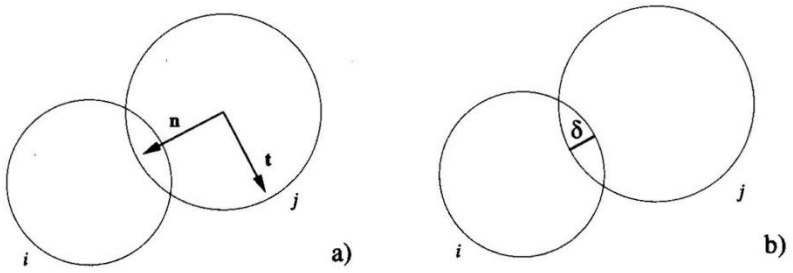


Figure 2.3: Impacting particles i and j. a) Definitions of the unit vectors $\mathbf{n}$ and $\mathbf{t}$. b) The separation δ used to calculate the normal force

Energy dissipation at contact is intrinsic to the characteristics of granular materials. The normal force is modeled as a linear spring with a damping force that opposes the relative velocity of contact for the duration of the interaction. The normal force is therefore described as follows:

$$F_n = \begin{cases} 0, & \delta_n > 0 \\ -k_n\delta_n - \gamma_n\dot{\delta}_n, & \delta_n < 0 \end{cases} \tag{2.7}$$

where k_n characterizes the stiffness of the particles and must be sufficiently large to ensure the overlap between the particles is small. The parameter γ_n is the material's viscous damping.

To avoid attractive forces, F_n is also verified to be non-negative.

The tangential force is modeled in the same way as the normal force arising from a spring stretched by the relative motion of the two particles. Not considering rotation, the tangential force is governed by Coulomb's friction law:

$$|F_t| \leq \mu_s F_n \tag{2.8}$$

where μ_s is the friction coefficient and F_t is the tangential force. To impose this restriction, the following conditions are set:

$$F_t = -k_t \delta_t - \gamma_t \dot{\delta}_t, \quad F_t = \begin{cases} sign\,(F_t)\,\mu_s F_n, & |F_t| \geq \mu_s F_n \\ F_t, & |F_t| < \mu_s F_n \end{cases} \tag{2.9}$$

where k_t characterizes the stiffness of the tangential spring and γ_t is the viscous tangential damping. The general framework discussed thus far considers the contact between two smooth spheres under the assumption of Hertz theory. The stiffness and viscous coefficients, k_n, k_t, γ_n, and γ_t, are calculated as follows from the material properties:

$$k_n = \frac{4}{3} E^* \sqrt{R^* \delta_n} \tag{2.10}$$

$$k_t = 8 G^* \sqrt{R^* \delta_n} \tag{2.11}$$

$$\gamma_n = -2\sqrt{\frac{5}{6}}\beta\sqrt{S_n m^*} \geq 0 \tag{2.12}$$

$$\gamma_t = -2\sqrt{\frac{5}{6}}\beta\sqrt{S_t m^*} \geq 0 \tag{2.13}$$

Here E^*, G^*, R^*, m^*, S_n, S_t and β are the effective Young's modulus, shear modulus, radius and mass respectively and are described as follows:

$$\frac{1}{E^*} = \frac{(1 - v_1^2)}{E_1} + \frac{(1 - v_2^2)}{E_2} \tag{2.14}$$

$$\frac{1}{G^*} = \frac{2\,(2 + v_1)\,(1 - v_1)}{E_1} + \frac{2\,(2 + v_2)\,(1 - v_2)}{E_2} \tag{2.15}$$

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2} \tag{2.16}$$

$$\frac{1}{m^*} = \frac{1}{m_1} + \frac{1}{m_2} \tag{2.17}$$

$$S_n = 2E^* \sqrt{R^* \delta_n} \tag{2.18}$$

$$S_t = 8G^* \sqrt{R^* \delta_n} \tag{2.19}$$

$$\beta = \frac{\ln(e)}{\sqrt{\ln^2(e) + \pi^2}} \tag{2.20}$$

where E is the particle Young's modulus, G the shear modulus, ν is Poisson's ratio, e the coefficient of restitution, and R and m are the particle radius and mass. The relations in the Hertzian nonlinear interaction law come from the calculation of the stress and strain in two elastic bodies maintained in contact.

Discretized, these interaction algorithms must resolve the movement of particles with sufficient precision. This requires the time step to be much smaller than the shortest time-frame that arises from the oscillation of one or two particles in contact. To determine an appropriate time step, the frequency of oscillation is estimated to be proportional to the natural frequency, ω_0, defined by:

$$\omega_0 = \sqrt{\frac{k}{m}} \tag{2.21}$$

This frequency characterizes the behavior of the granular system. Depending on stiffness and damping parameters, k_n, γ_n, and the mass value of the material, oscillations can be damped out or perturbations created in the harmonic character of the model. The time step is selected as:

$$\triangle t \approx \frac{\varepsilon}{\omega_0} \tag{2.22}$$

where ε is a constant that acts as a safety factor. The value depends on the integration algorithm and a value such as $\varepsilon \approx 0.01$ is an acceptable value. It is representative of ten time steps covering the interaction from initial contact to loss of contact. For the LIGGGHTS version 2.3.8 software, ε is approximately 0.0045. This estimated time step is also periodically

checked against the Rayleigh time step. This check is performed for the contact forces given by the Hertz law as the oscillation frequency depends on the interpenetration. The Hertz model is similar to a nonlinear spring that becomes stiffer when compressed, thus increasing the frequency ω_0 with normal force.

To model the rotational inertia and energy loss in rotating particles, a rolling resistance model is applied. The model used is an elastic-plastic spring-dashpot model. When cohesion between particles is present at the interface contact, energy is dissipated when the adhesive bond breaks at the separation point during rolling. When cohesion is present, the resistance to motion can be significant even in the absence of externally imposed pressure [20].

The rotational model used consists of two components: the mechanical spring torque M_r^k and the viscous damping torque M_r^d. The spring torque is dependent on the relative rotation between the contacting particles. It is similar to the loading-unloading stress-strain curve of an elastic perfectly plastic material. Figure 2.4 shows the spring torque of this rolling resistance model [20]. The total rolling resistance model is described as:

$$M_r = M_r^k + M_r^d \tag{2.23}$$

In an incremental manner, the incremental spring torque is giving by

$$\triangle M_r^k = -k_r \triangle \theta_r \tag{2.24}$$

where k_r is the rolling stiffness and $\triangle\theta_r$ is the incremental relative rotation between two particles. The spring torque at time $t + \triangle t$, the next timestep, is found as:

$$M_{r,t+\triangle t}^k = M_{r,t}^k + \triangle M_r^k \tag{2.25}$$

where the next rotational increment is limited by the limiting spring torque M_r^m.

$$\left|M_{r,t+\triangle t}^k\right| \leq |M_r^m| \tag{2.26}$$

Where the limiting spring torque is described by the coefficient of rolling resistance μ_r, the effective radius R^*, and the normal contact force F_n as follows:

$$M_r^m = -\mu_r R^* F_n \tag{2.27}$$

The viscous damping torque M_r^d is assumed to be dependent on the relative rolling angular velocity $\dot{\theta}_r$ and the damping constant C_r as follows:

$$M_{r,t+\triangle t}^d = \begin{cases} -C_r\dot{\theta}_r & if \ \left|M_{r,t+\triangle t}^k\right| < M_r^m \\ -fC_r\dot{\theta}_r & if \ \left|M_{r,t+\triangle t}^k\right| = M_r^m \end{cases} \tag{2.28}$$

where f is set to 0 in the case where there is viscous damping torque only. This is only active before the contact rolling torque is fully mobilized [20]. The viscous damping torque is used to help stabilize and prevent rolling oscillation. The constant f is 1, when viscous damping is present. The elastic-plastic spring-dashpot model was selected because it dissipates kinetic energy, provides stable torques and dense particle packing.

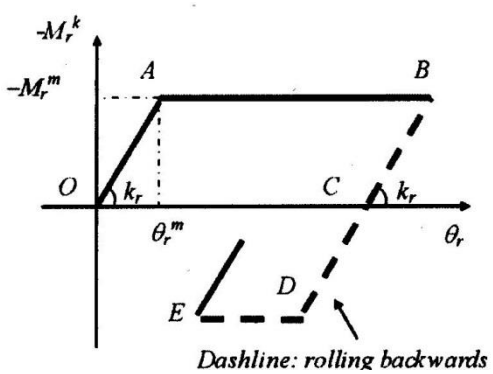


Figure 2.4: Spring torque of elastic-plastic spring-dashpot rolling resistance model from [20].

2.3 Discrete Element Models of Cohesive Contact

The following sections describe different models of cohesive contact behavior that are applied as external forces in Newton's Second Law. These models use spherical particles to simulate three-dimensional systems and make the following assumptions about the contact:

- The particles are perfectly smooth
- The behavior is elastic and isotropic
- The tangential component of the force does not affect the normal component

- The contact deflection is small ($\delta_n \ll a$, the contact zone radius)

It is under these assumptions that the Hertzian contact law is applied.

2.3.1 The Bradley Model

One of the initial cohesive models was proposed by Bradley [21]. This model neglects contact deformations and considers the van der Waals forces. The assumption places the Bradley model in the fine powders application. Bradley showed that the adhesive force for rigid spheres follows a force derived from the Lennard-Jones potential—a mathematical model that approximates the interactions between a pair of neutral atoms or molecules. The force of adhesion between two rigid spheres is described by:

$$f_n = \frac{8\pi\gamma_{suf}\, R^*}{3}\left[\frac{1}{4}\left(\frac{\delta_n}{\delta_0}\right)^{-8} - \left(\frac{\delta_n}{\delta_0}\right)^{-2}\right] \tag{2.29}$$

where δ_0 is the equilibrium separation distance between the particles [21]. Two particles separate when the forces pulling the particles is reached at $\delta_n = \delta_0$. This model uses the van der Waals forces as the interaction that generates the cohesive force. At the macroscopic scale, these forces are negligible when compared to the gravitational force experienced by the particles. Scaling from this level is inaccurate and only serves as a possible qualitative representation of bulk solid cohesion. Figure 2.5 provides an example of the Bradley cohesion model and illustrates the attraction force between particles at a distance [3].

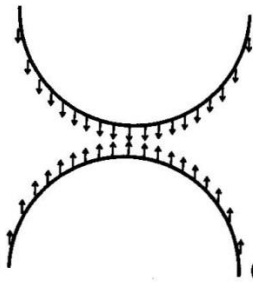


Figure 2.5: The Bradley Model from [3].

2.3.2 The Johnson-Kendall-Roberts Model

The Johnson-Kendall-Roberts, JKR, model is an improvement over the Hertzian contact model, which only represents dry loading and unloading conditions [4]. The JKR model takes into account the surface energy at the contact. This theory correlates the contact area of two contacting particles to the elastic material properties and the interfacial interaction strength. The cohesive force can be formed during the unloading cycle of contact as a force resisting separation. The JKR contact between two particles leads to the radius a of the contact zone to be described by:

$$a^3 = \frac{R^*}{E^*}\left[f_n + 3\gamma_{sur}\pi R^* + \sqrt{6\gamma_{sur}\pi R^* f_n + (3\gamma_{sur}\pi R^*)^2}\right] \tag{2.30}$$

where γ_{sur} is the surface energy in $\frac{J}{m^2}$. The separation of the two particles is obtained from a maximum tensile force given by:

$$f_{JKR} = -\frac{3}{2}\pi\gamma_{sur}R^* \tag{2.31}$$

and does not depend on the elastic moduli of the material. The total normal force of contact with cohesion can be written as:

$$f_n = \frac{4a^2E*}{3R^*} - \sqrt{8\pi E^*\gamma_{sur}a^3} \tag{2.32}$$

This represents a fully elastic model with cohesion between particles in the contact zone [3, 4, 22]. Figure 2.6 provides an example of the JKR model and illustrates the tensile force between the particles in cohesive contact. The JKR approximation is accurate for

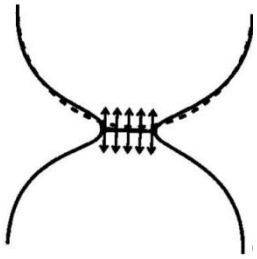


Figure 2.6: The Johnson-Kendall-Roberts contact mechanical model from [3, 4].

large cohesive energies and larger particles with low Young's modulus. The model does not provide resistance in the tangential shearing direction. This limits the effect cohesion has on material flow because material is allowed to slide past each other with little resistance. In the case of a silo rathole, identifying the walls of the no-flow channel would prove difficult if no cohesive sliding resistance is present to form the rathole.

2.3.3 The Derjaguin-Muller-Toporov Model

A distinguishing feature of the Derjaguin-Muller-Toporov model (DMT) is to neglect the contact deflection [23]. It is indirectly taken into account through the calculation of the attraction force and uses the Hertzian contact model. The attraction force between two particles is given by:

$$f_{DMT} = -2\pi\gamma_{sur}R^* \tag{2.33}$$

where R^* is as defined by equation 2.16.

The DMT model is a good approximation of cohesive forces for small particles within the limit of weak cohesion. The attraction force is limited by the separation distance between two contacting particles at which the bond is broken. This limit is smaller for DMT and the particles to which it properly applies to. Figure 2.7 shows a schematic representation of the DMT contact mode [3, 23, 24]. It illustrates how the contact deflection is indirectly taken into account through the attractive force between the particles.

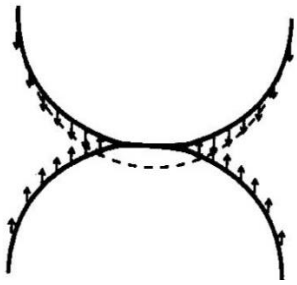


Figure 2.7: The Dejaguim-Muller-Toporov model from [3].

2.3.4 The Simplified JKR Model

From the original JKR model, the contact radius can be determined as a function of the contact overlap δ_n. The contact area between two particles is not a simple calculation to perform, therefore the simplified JKR model approximates the radius a of the contact zone with:

$$a^2 \approx R^* \delta_n \tag{2.34}$$

Simplifying equation 2.32, the normal force is written as:

$$f_n = E^* R^{\frac{1}{2}} \delta_n^{\frac{3}{2}} - U_a E^{*\frac{1}{2}} R^{\frac{3}{4}} \delta_n^{\frac{3}{4}} \tag{2.35}$$

where

$$U_a = \sqrt{6\pi\gamma_{sur}} \tag{2.36}$$

This eliminates the computation of radius a of the contact zone while still providing an explicit expression of the force as a function of the overlap.

2.3.5 Capillary Cohesion

The capillary cohesive model results from the formation of a liquid bridge between two particles. Its geometry between particles of different sizes is complex in shape. The bridge is dependent on the particle radii R_1, R_2, its geometric complexity by the filling angles ς_1, ς_2, wetting angle θ and the distance between the particles δ_n, as illustrated in Figure 2.8. A capillary cohesive model assumes the particles are perfectly spherical and smooth. The moisture content is sufficiently low that water is present as unconnected liquid bridges. The liquid bridges are not deformed under gravity and its effects are neglected. The viscosity of the liquid can be neglected under a quasi-static regime.

The capillary force between the particles can be described by the pressure difference and shape of the liquid bridge as follows:

$$\triangle p = \gamma_{sur} \left[\frac{y^{''}(x)}{(1 + y'^2(x))^{\frac{3}{2}}} - \frac{1}{y(x)\sqrt{1 + y'^2(x)}} \right] \tag{2.37}$$

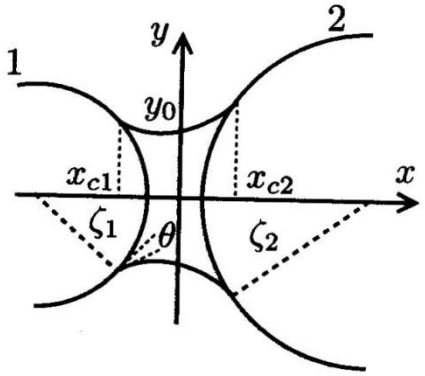


Figure 2.8: Geometry of a liquid bridge between two particles of different sizes from [3].

The pressure difference $\triangle p$ is related to the curvature of the liquid bridge and the surface tension of the liquid γ_{sur} [3, 25, 26]. The capillary force is then described by:

$$F = 2\pi\gamma_{sur}y_0 + \pi\triangle p y_0^2 \tag{2.38}$$

where the radius of the bridge is denoted by y_0 [5, 27]. In discrete numerical simulations, the normal cohesive force is an explicit expression. Several expressions have been proposed by researchers such as Soulie, Scholtes et al. [26]. These models have a failure criterion of the debonding distance $\delta_{rupture}$. One such failure criterion proposes the following relationship between the debonding distance $\delta_{rupture}$, the volume of the liquid bridge V and the wetting angle θ [28].

$$\delta_{rupture} = \left(1 + \frac{\theta}{2}\right) V^{\frac{1}{3}} \tag{2.39}$$

In particulate materials, the volume of all the liquid bridges cannot be the same. Yet moisture is allocated to each grain as a fraction of the total volume of liquid [28].

2.4 Comparison of the Cohesive DEM Models with Macroscopic Behavior

DEM with cohesive interactions is based on the assumption that the particles are rigid discrete elements. These cohesive DEM models can only compute the cohesive interactions between particles. At the microscopic scale, results are qualitatively and quantitatively

similar to the physical phenomena taking place between grains [3, 29, 30].

In the cohesive DEM models presented, the Bradley model computes a van der Waals interaction. These are negligible at the bulk scale dominated by gravitational forces. Van der Waals interactions are formed and broken continuously and only effective over a very small distance between particles. The weak adhesive force computed between particles is insufficient to overcome gravity and often observed in systems of bulk conveying, cannot generate material clumping or material adhesion to system walls.

The JKR model, the simplified JKR model and the DMT model compute cohesion using the surface energy between particles in contact. The contact zone and the surface energy determine the tensile force required to separate and break the cohesive bond between the particles. At the macroscopic scale, these models can determine weak cohesive forces. The approximations computed by the JKR models are more accurate with material of low Young's modulus or low stiffness. The DMT model approximation is more accurate with small fine particles. The cohesion force estimated by these models follows a ratio between the contact zone radius and the adhesion strength at failure. With bulk transfer systems conveying large particles with high stiffness, these models have a limit for which the force is accurate and stable.

The capillary force model computes a liquid bridge between particles. It assumes smooth spherical particles with low water content in a quasi-static regime where the viscosity of the liquid can be neglected. The volume of water is distributed equally to each contacting particle pair. At the microscopic scale, capillary forces provide accurate representation of cohesive forces as a function of the distance between interacting particles. At the macroscopic scale, the proposed geometry of the liquid bridge no longer properly models the cohesive contacting force. The volume of liquid, debonding distance and the wetting angles do not provide a physically interpretable form of the capillary law.

With the cohesive models presented here, there is sensitivity to the size of the particles. At the macroscopic scale, these models provide weak interaction forces that can be numerically

dominated by the friction properties of dry material. The cohesive numerical methods will always yield a result but the resulting flow-ability computed for the material should be compared to the physical experiments and operational systems. In many cases, the effect that cohesion has on the flow of material is underrepresented and the macroscopic data indeterminate in predicting crucial issues. Meaning that the physical model parameters need to be scaled to simulate the mechanical behavior of the material observed.

CHAPTER 3

DEM STUDIES PERFORMED UNDER THE CAPILLARY FORCE AND JOHNSON-KENDALL-ROBERTS MODELS

Functional DEM cohesive models can be seen in industry in the form of capillary forces and surface energy models. The previous chapter discussed the models most commonly applied to cohesive materials. The studies performed here center on the two most widely applied cohesive DEM models: the capillary force model and the Johnson-Kendall-Roberts model. Material was simulated under various flow property tests as a means of measuring the effective cohesion in a system.

This chapter presents the results of the simulated flow property tests performed. The models required the selection of the material's surface tension and particulate fines and were scaled and tuned to describe the desired cohesive material flow. These sections illustrate the shortcomings of the cohesive models to accurately represent cohesiveness in macroscopic particles without a means of scaling.

3.1 Modeling Cohesion with Capillary Forces

This research continued the survey of macroscopic cohesion with a study of the capillary force model used by the bulk solids handling team Overland Conveyor Company, Inc. (OCC) [31]. The force model studied represents the capillary bond formed between two uneven-sized spheres and was presented by Pierrat et al (1997) [32]. The capillary force consists of two components: one given by the surface tension of the liquid and the second by the curvature of the liquid bridge which creates a pressure difference across the surfaces. The total capillary force on the particle is the sum of the two components and is represented by:

$$F = 2\pi\gamma_{sur}R\sin(\theta)\left[\sin(\varsigma+\alpha)+\frac{R\sin(\varsigma)}{2}\left(\frac{1}{r_1}-\frac{1}{r_2}\right)\right] \tag{3.1}$$

where γ_{sur} is the surface tension, ς is the filling angle of the particle, R is its radius and α is the contact angle [32].

At the macroscopic scale, the proposed geometry of the liquid bridge observed in Figure 2.8 no longer properly models the cohesive contacting force. Hence, a single macroscopic particle was modeled after a number of fine particles. Figure 3.1 illustrates a single DEM macro-particle modeled by N finer particles. This approach is used in order to maintain the capillary force within a range in which the capillary law was physically interpretable.

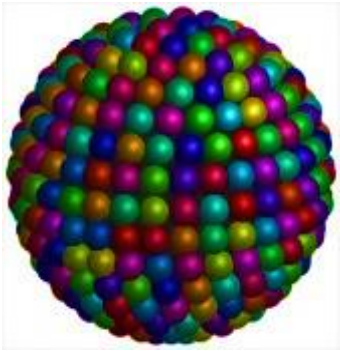


Figure 3.1: A group of fine particles modeled as a single DEM particle.

The macroscopic particle described by Figure 3.1 was given a cohesive force equivalent to the scaled capillary forces of the fine particles. The applied scaling is described as:

$$F_{particle} = \left(\frac{R}{r}\right)^3 F \tag{3.2}$$

where R is the macro-particle radius, r is the fine particle radius and F is the capillary force as defined by equation 3.1. The scaled capillary force showed favorable results in scaling the debonding distance for larger particles. However, the force required to simulate the macroscopic material flow presented its challenges. The scaling parameter is dependent on the chosen radius of the finer particles and to obtain the desired material flow condition, the fine particle radius is carefully selected to match the effective cohesion of the system. To observe the cohesive flow behavior and see its apparent impact on operational system components, the damage due to abrasion of cohesive materials on a conveyor transfer was

examined. Initially, material is placed on a conveyor belt. A conveyor belt is a flexible band placed around two or more pulleys for the purpose of transmitting material from one point to another [33]. Conveyor transfers often require a high level of maintenance, and in many instances are the primary causes of conveyor and plant downtime [34]. Figure 3.2 shows the material flow through a funnel transfer. The funnel is a component that directs the material from all directions onto the center of the next transfer belt.

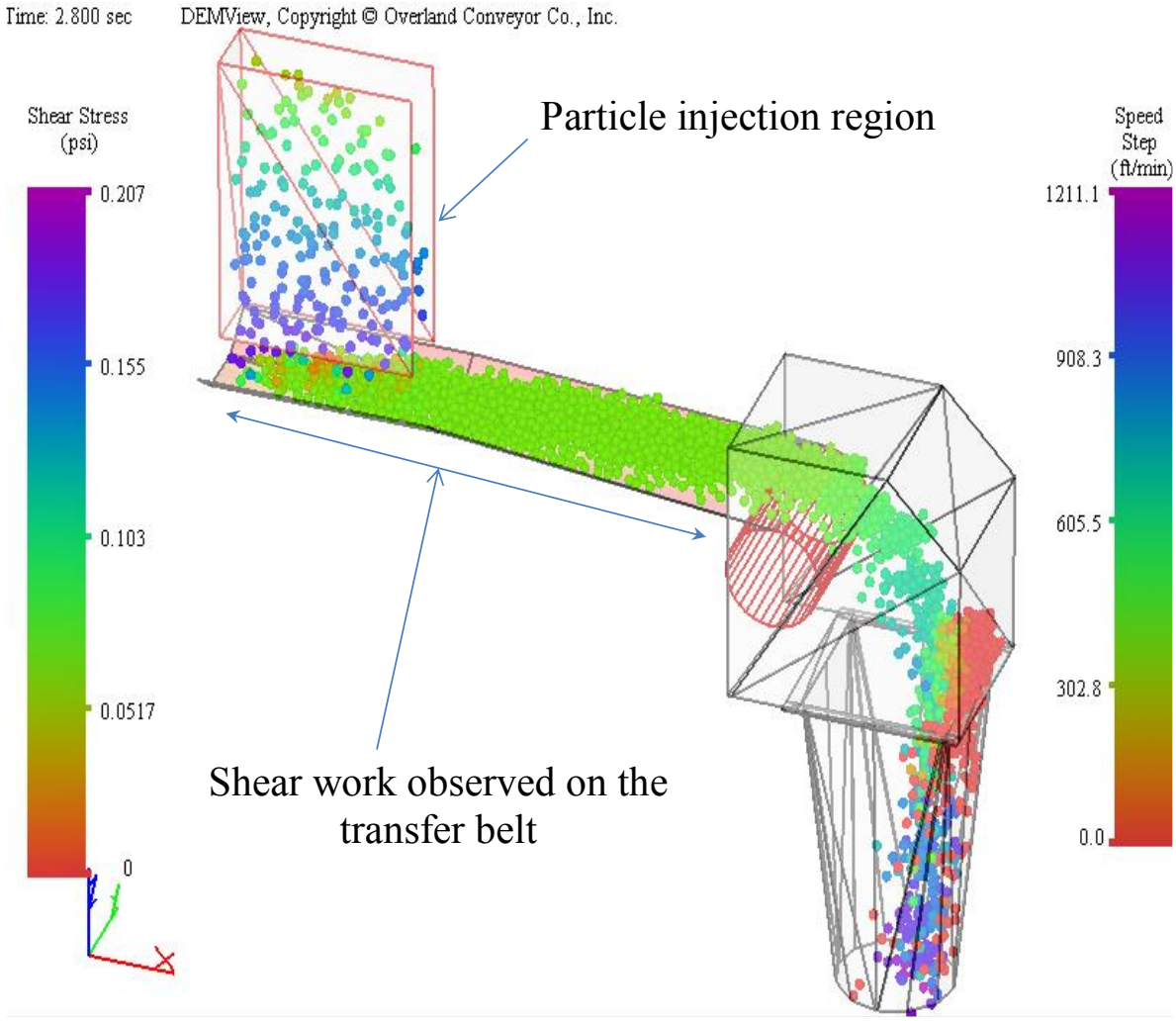


Figure 3.2: Conveyor funnel transfer chute under abrasive wear by cohesive bulk solid material [31].

Figure 3.3 shows the impact the material has on the shear work experienced by the transfer belt at different cohesive levels. The cohesion index observed in Figure 3.3 is a term used by OCC [31] and is graphically represented in Figure 3.4 as a function of the surface tension of water at ambient temperature. It is a qualitative measure of how much material adheres together. It describes the apparent cohesiveness of the material using the surface tension of water as the liquid bridge property in the capillary force model. A low cohesive index represents a system with little moisture, while a high cohesive value indicates cohesive cementation present in materials such as clay. The cohesive index range used for this study is representative of material lightly sprayed with water for dust control to material agglomerations where there is enough moisture in the system for fine particulates to bond larger material together.

Shear work is described by OCC [31] as the work generated by accumulating the incremental work generated by the sliding friction force and the slip distance of the particulate material in contact with a surface that is part of the conveyor system. The highest shear work is observed with dry material. The more moisture is added to the system the less damage is observed. The material starts to adhere to the belt rather than slip. These values can be used to predict linear replacement life and expected belt wear. In addition to belt and surface wear, damage can be examined to determine the level of material degradation that leads to dust generation. With this capillary force model, however, the amount of cohesion in the system is a behavior determined by the tuning and scaling of the surface tension of water and the radius of the particulate fines. This leads to a lack of a quantitative measure of cohesion related a real physical parameter. Engineers can provide little confidence in the cohesive values presented for the material in bulk handling solutions.

3.2 Modeling Effective Cohesion with a Modified Johnson-Kendall-Roberts Model

This research continued the examination of macroscopic cohesion with the study of the JKR model through simulated conveying system applications. In the conveying systems, it was found that the cohesive force computed underestimated system behavior. Given the

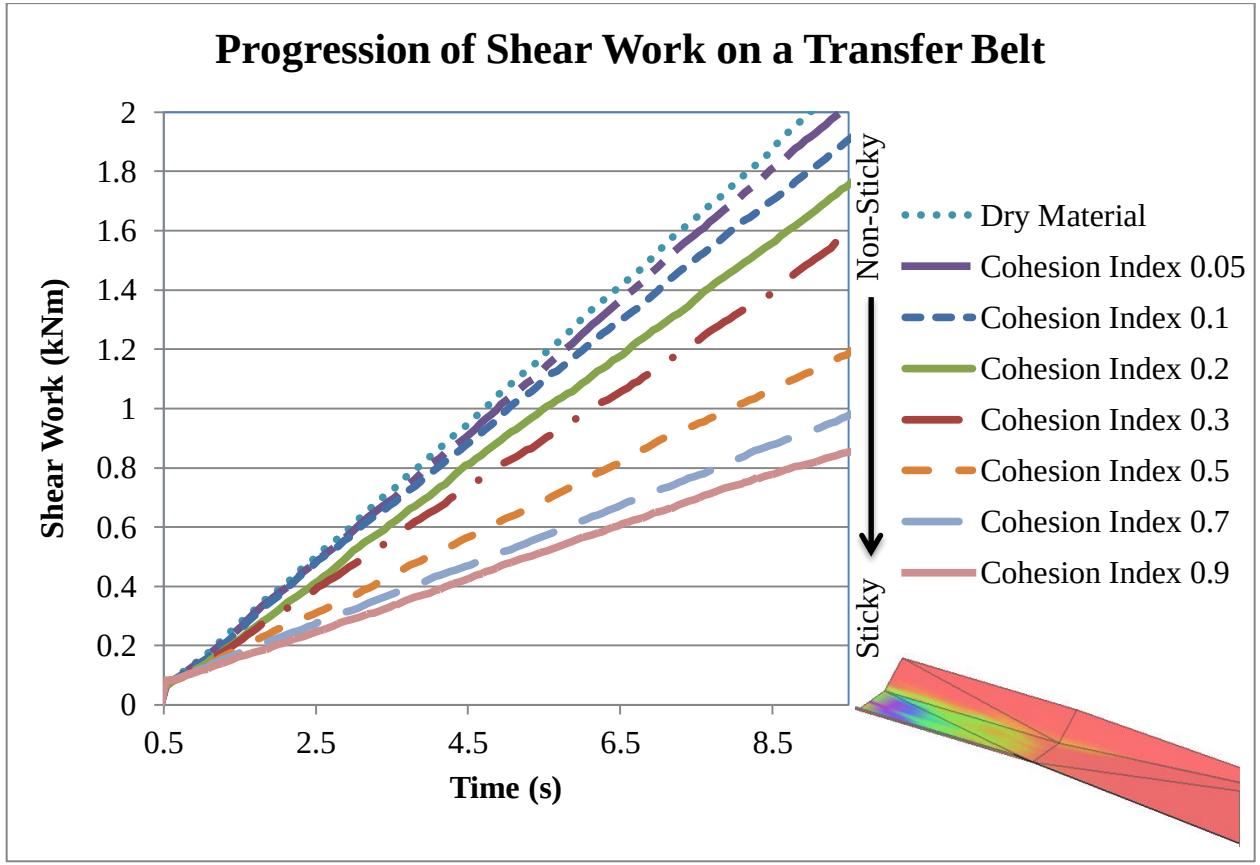


Figure 3.3: Shear work on the transfer belt at varying cohesive levels.

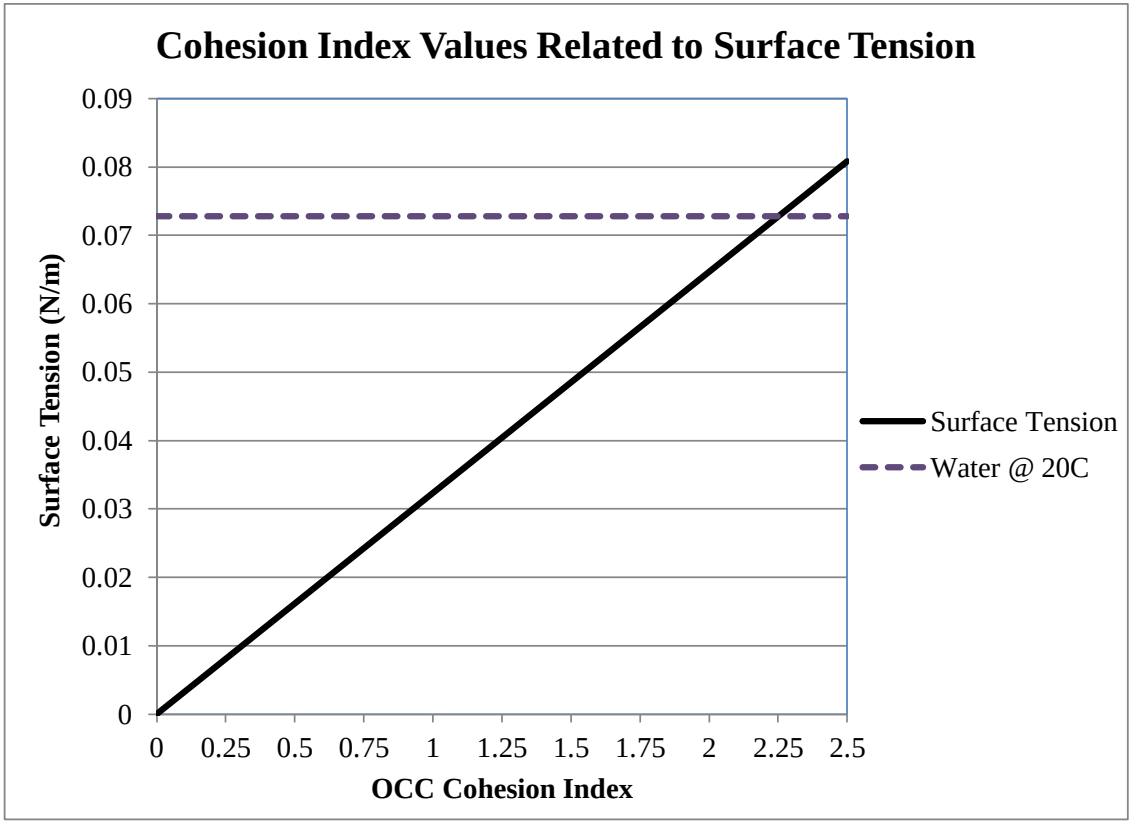


Figure 3.4: Overland Conveying Co, Inc. definition of Cohesion Index [31].

material properties, the cohesive force could not provide a force great enough to overcome the force due to gravity. The existing JKR model described in equation 2.32 was then modified under the squared ratio of the equivalent particle radius to provide the desired cohesive flow behavior. This approach is similar to the model proposed by Kruse (2005) [34] for scaling capillary cohesion in that the geometry is scaled by the number of fines in the system. The scaling law proposed by Kruse [34] also scales the surface tension to an equivalent macroscopic surface tension and this differs in this modified model since the surface tension is not scaled. The modified JKR model was tested with two simulated flow property tests: an active failure system to observe how cohesion affects the angle of repose and a ring cell shear tests to observe the effect cohesion has on shear stress.

The modification made to overcome the inability to model all fine particles and their cohesive forces was grouping fine particles to model a single macro DEM particle. The macro particle is given the JKR cohesive force which results from the combined forces of the fine particle. One macro particle is represented by n particles. Therefore, the equivalent mass of the single DEM particle is described by:

$$m_{eqv} = n \cdot m \tag{3.3}$$

and its equivalent radius by:

$$r_{eqv} = r\, n^{\frac{1}{3}} \tag{3.4}$$

where r and m are radius and mass of the fine particles and m_{eqv} and r_{eqv} are the equivalent mass and radius of the single macro particle [34]. The equivalent cohesive force for the larger particle is found by a squared particle radius ratio given by:

$$\left(\frac{r_{eqv}}{r}\right)^2 \tag{3.5}$$

Therefore, the total JKR cohesive force on a DEM particle is given by:

$$F_{total} = n \left(\frac{r_{eqv}}{r}\right)^2 F_{JKR} \tag{3.6}$$

This allows for the attraction between two fine particles to be combined to amass the equivalent macro scale attraction force of a particle. Though the underlying attraction force at the fines level is physically sound, the effective cohesion of the system was a behavior induced by the tuning and scaling of the radius of the particulate fines. The more fines representing the system, the greater the surface energy computed using the JKR model scaled to the equivalent DEM particle size.

3.2.1 Cohesive Flow Behavior through a Conveyor Transfer System

To further study the effective cohesion of the modified JKR model, wet material was loaded onto a plane and rotated. This was done to simulate material splashed onto the surface of a hooded chute and the material build-up that can occur. The planes are illustrated in Figure 3.5 for a set of material sizes. The results showed that even with scaling the modified JKR model cannot consistently approximate the same mass of material adhering to the plane for various particle sizes.

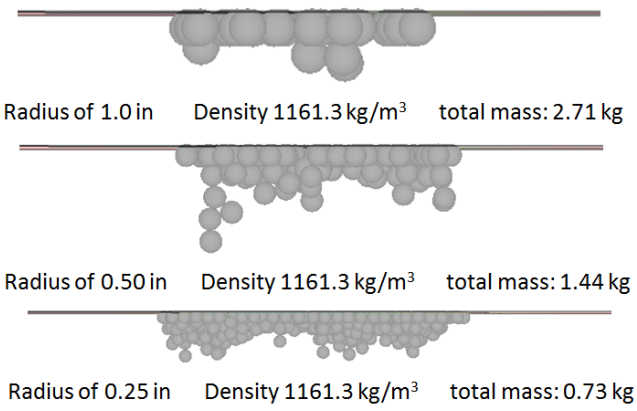


Figure 3.5: Particle plane adhesion

Prior to scaling, the JKR model was solely governed by the Young's modulus describing the stiffness of the material and the surface energy between particles. This approach to the original JKR cohesive model would lead to cases of no cohesion or extreme clustering of particles without the proper tuning of the surface energy. However, even with scaling,

it was difficult to obtain consistency in the cohesive behavior. To illustrate the resulting cohesive forces of the scaled JKR model, a high tonnage transfer chute and rockbox were modeled. Figure 3.6 and Figure 3.7 illustrates the dynamic behavior and cohesive effects of the material. In the high tonnage transfer, the cohesive material starts to build up. This accumulation of cohesive material is represented by the red stationary particles. As the system continues to operate, material sloughs off the surfaces of the head chute and funnel. The head chute directs material into the funnel which then centers the material on the conveyor belt below. In highly cohesive models, the material will begin to adhere to the head pulley and cause a change in the material trajectory. The head pulley is the last pulley on the carry side of the conveyor belt, which is described as the head of the conveyor. The carry side is the top side that carries the material. Excessive build-ups along the walls and in the funnel can cause plugging or reduction of the desired tonnage transfer. Similarly, in Figure 3.7 material build up can occur on the walls, rockbox and bottom dead box. Dead boxes are used to take the direct impact of material discharged from a conveyor into a head chute. Changes of flow direction can also be accomplished through the use of a dead box and are often used to redirect materials through changing levels in a plant.

3.2.2 Material Shear Failure using the Schulze Ring-Shear Cell Tester

To further test the shear properties of cohesive particles, ring cell shear tests were performed. The following section outlines the procedure implemented using the modified JKR model in LIGGGHTS. The test geometry is modeled after a Schulze ring shear tester and the test procedure is modeled after the ASTM D 6773 standard [35, 36]. The simulation geometry for the ring-shear test simulation is shown in Figure 3.8. The inner and outer walls, as well as the lid and base, are modeled as steel (E = 180 GPa). The inner and outer walls are frictionless, whereas the lid and base have a particle-wall static sliding coefficient of friction (μ_p) equal to 0.5. The ring cell is modified such that both the lid and base have eight "fins" each that have a height of 10 mm. The fins prevent the bulk material from sliding directly on the lid and base surfaces as the material is sheared. They represent the

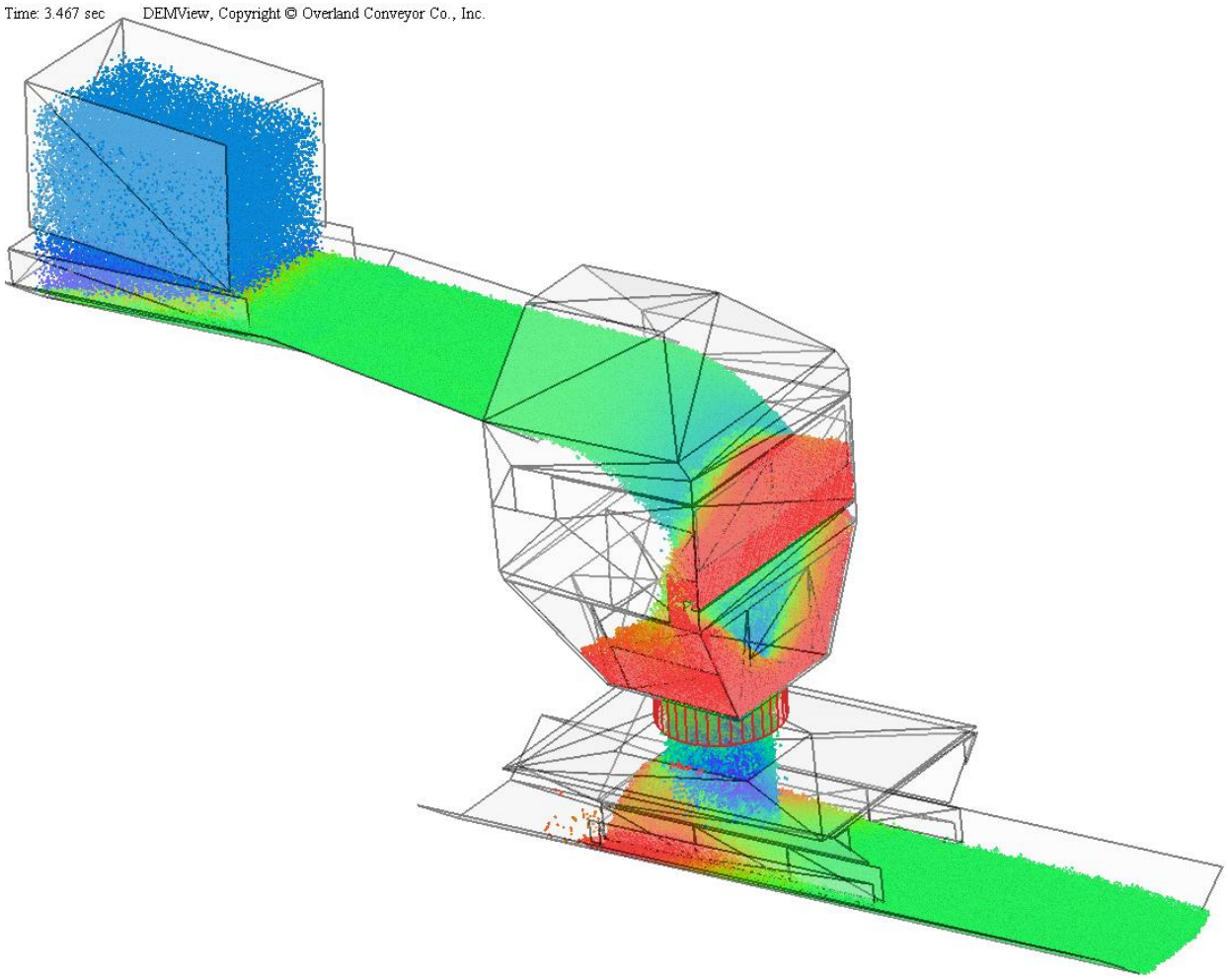


Figure 3.6: Wet cohesive material flowing through a high tonnage transfer chute [31].

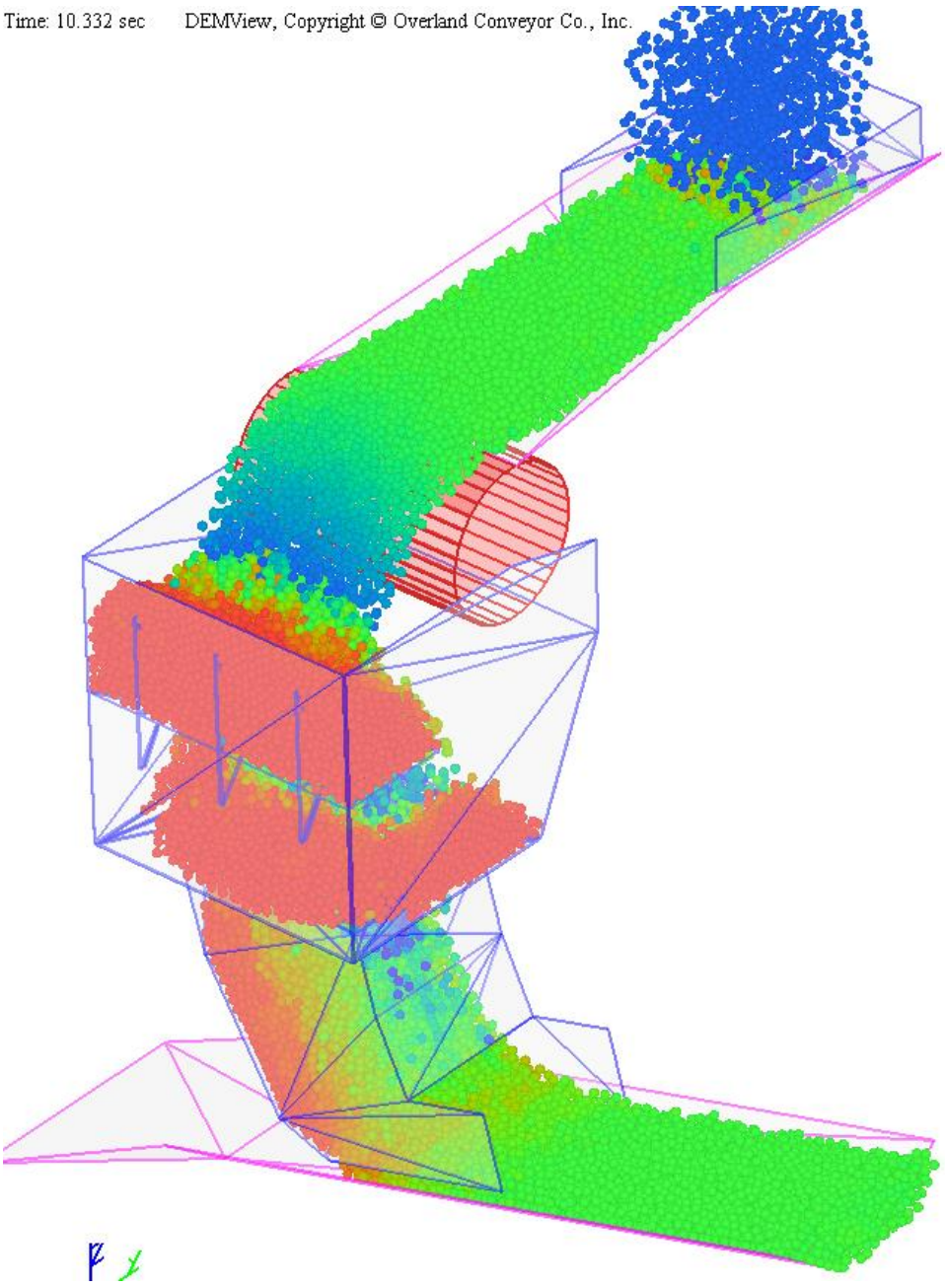
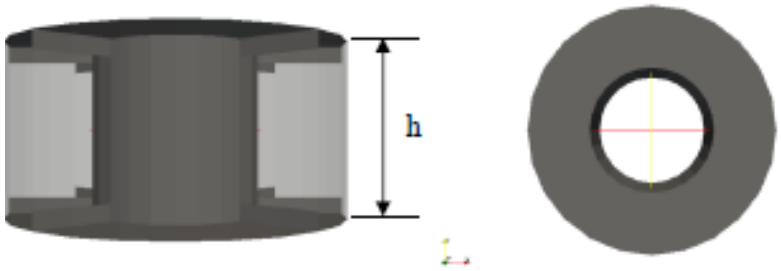


Figure 3.7: Wet cohesive material through a rockbox and dead box transfer chute [31].

corrugated surfaces of the physical ring cell. The dimensions of the cell geometry are shown in Table 3.1. In this table, OD, ID and h refer to the outer and inner diameter of the annulus and the height of the tester, respectively. The value associated with A_d refers to the surface area of the annulus.



(a) Side View (b) Top View

Figure 3.8: Geometry of the virtual ring cell shear tester.

Table 3.1: Dimensions of the ring cell shear test geometry.

OD (mm)	ID (mm)	A_d(m^2)	h (mm)
200	100	0.0236	100

The first step shearing material to failure is filling the cell with loose unconsolidated material. It is then compressed by the lid to a specified pre-shear normal stress value (σ_{pre}) and sheared. A proportional controller is used to maintain σ_{pre} throughout the remainder of the simulation. The shearing base is rotated at 0.03 rad/s to begin shearing of the bulk material until failure. The resulting material shear stress is calculated as torsional data gathered as follows:

$$\tau = \frac{T}{r_{mean} A_d} \tag{3.7}$$

The shearing stress is calculated using the torque exerted by the particles on the lid (T). The moment arm is the mean radius of the annulus (r_{mean}). Once the material undergoes shearing, the shear stress increases until it reaches some steady-state value (τ_{pre}) representing material failure. The test ends once steady-state shearing has been achieved. The shearing

process is repeated for three sets of cohesive values represented as cohesive energy densities (k_{coh}) and are in the units of $\frac{J}{m^3}$ for the now consolidated material.

The angle of effective internal friction of the material at steady-state (ϕ_{sf}) is the measurement of the incipient flow of the material. It is described as the arctangent of the ratio between the steady-state shear stress (τ_{pre})and the pre-consolidation stress (σ_{pre}) :

$$\phi_{sf} = \arctan\left(\frac{\tau_{pre}}{\sigma_{pre}}\right) \tag{3.8}$$

This value provides the angle at which consolidated material is likely to fail to induce flow. The material property of the simulations is given in Table 3.2. The number of particles is chosen such that a bed of material approximately 50 mm tall is created within the cell. The Hertzian contact model is used and the time-step for the simulation is chosen to be on the order of 10^{-5} sec for numerical stability. Rolling friction (μ_r) is set to 0.3 and the coefficient of restitution, e, to 0.2. These values are representative of damped rough free flowing material.

Table 3.2: Testing Sample Material Properties

# of particles	Particle diameter (mm)	Young's modulus (GPa)	Density $\left(\frac{kg}{m^3}\right)$
10,000	6.35-3.17	111	1957.5

In this study, six simulations were performed on the material sample at different levels of cohesion and the internal angle of friction computed. Starting with dry material with zero moisture and hence no cohesion, the internal angle of friction was computed as approximately 33.7 degrees. When moisture is introduced to the system, it is observed that the effective internal angle of friction in reduced. This predicts that the frictional angle is dependent on the level of moisture in the system. Table 3.3 summaries the results of these simulations at different levels of cohesion. The levels of cohesion model material with little moisture to material clumping. Figure 3.9 shows the shear stress on the system for the different levels of cohesion. Also plotted are the shearing stresses for the different consolidation stresses along with the failure envelope developed from the different shearing points for dry material

in Figure 3.10 and Figure 3.11. Similar results are plotted for the most cohesive material sample in Figure 3.12 and Figure 3.13. It was observed that the failure stress points of each cohesive material condition is dependent on the cohesive forces present. This is expected and is representative of the differing internal angles of frictions computed representing the pile angles at which each cohesive material would start to flow.

Table 3.3: Angle of internal friction with a pre-shear stress σ_{pre}, of 42.4 kPa and pre-shear force F_{pre}, of 1000 N.

$K_{coh}(\mathrm{kJ/m^3})$	Sliding, Rolling Friction (μ_s, μ_r)	τ_{pre}(kPa)	ϕ_{sf}(deg)
0.00	0.5, 0.3	28.28	33.68
1141.55	0.5, 0.3	28.45	33.85
1995.38	0.5, 0.3	26.08	31.57
2592.04	0.5, 0.3	25.25	30.75

The k_{coh} value present in Table 3.3 and the accompanying figures is a cohesive energy density value. The JKR model defines the force of separation as the normal contact force plus the attractive adhesive forces between contacting surfaces in equation 2.32. The attraction component is represented by

$$f_{attraction} = \sqrt{8\pi E^* \gamma_{sur} a^3} \tag{3.9}$$

which the JKR model applies as

$$f_{attraction} = k_{coh} A_{contact} \tag{3.10}$$

We model the contact area as:

$$A_{contact} = \pi a^2 \tag{3.11}$$

where a represents the radius of the contact region between particles. The cohesive energy density value can then be represented as:

$$k_{coh} = \frac{\sqrt{8\pi E^* \gamma_{sur} a^3}}{\pi a^2} \tag{3.12}$$

Figure 3.14 shows the contact radius between impacting particles. It displays the contact radius as the intersection of the spheres lying on a plane which in profile is a circle of radius

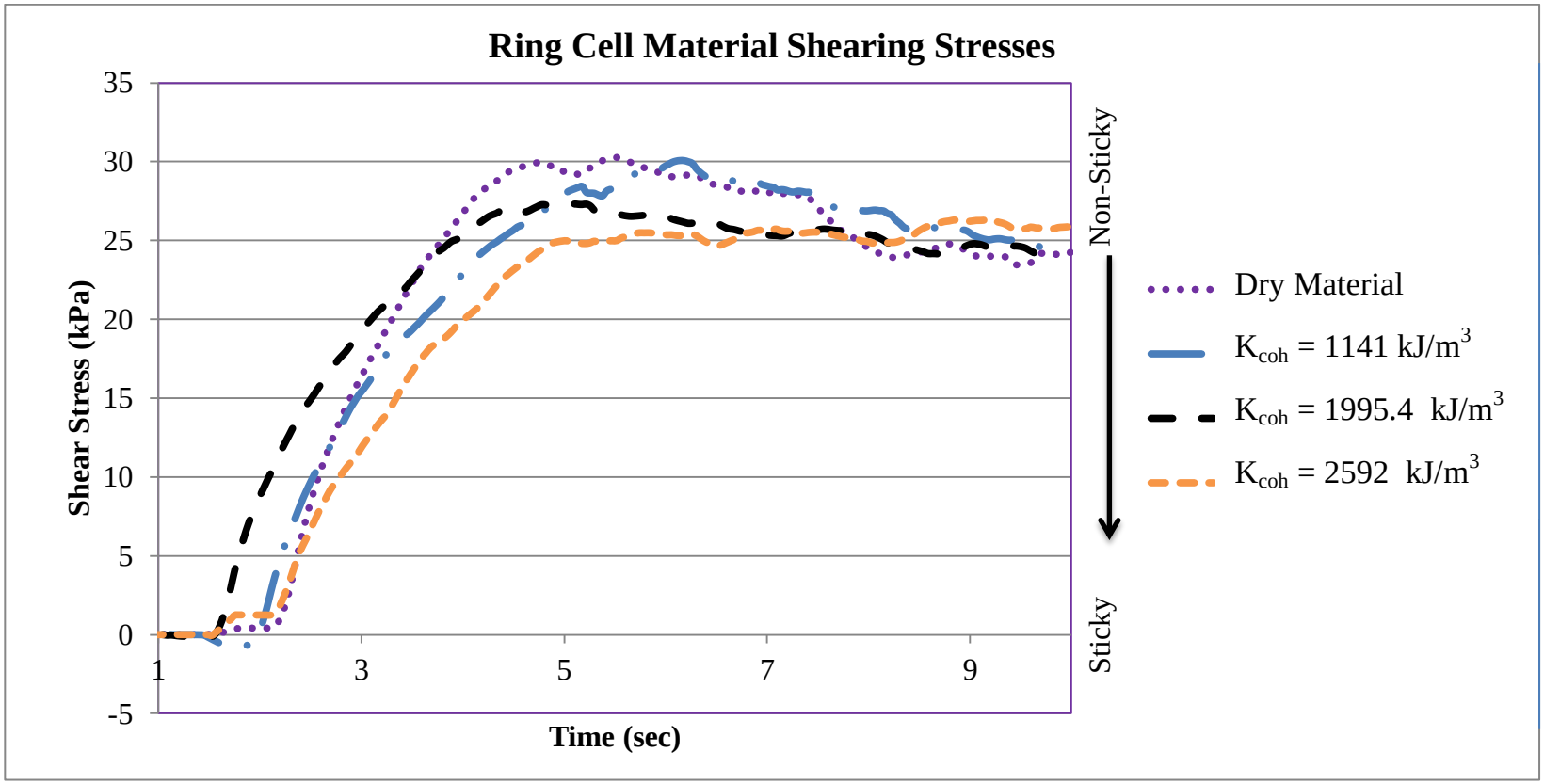


Figure 3.9: Ring cell shearing stress developed over time for $\sigma_{pre} = 42.4\,\text{kPa}$ ($F_{pre} = 1000\,\text{N}$)

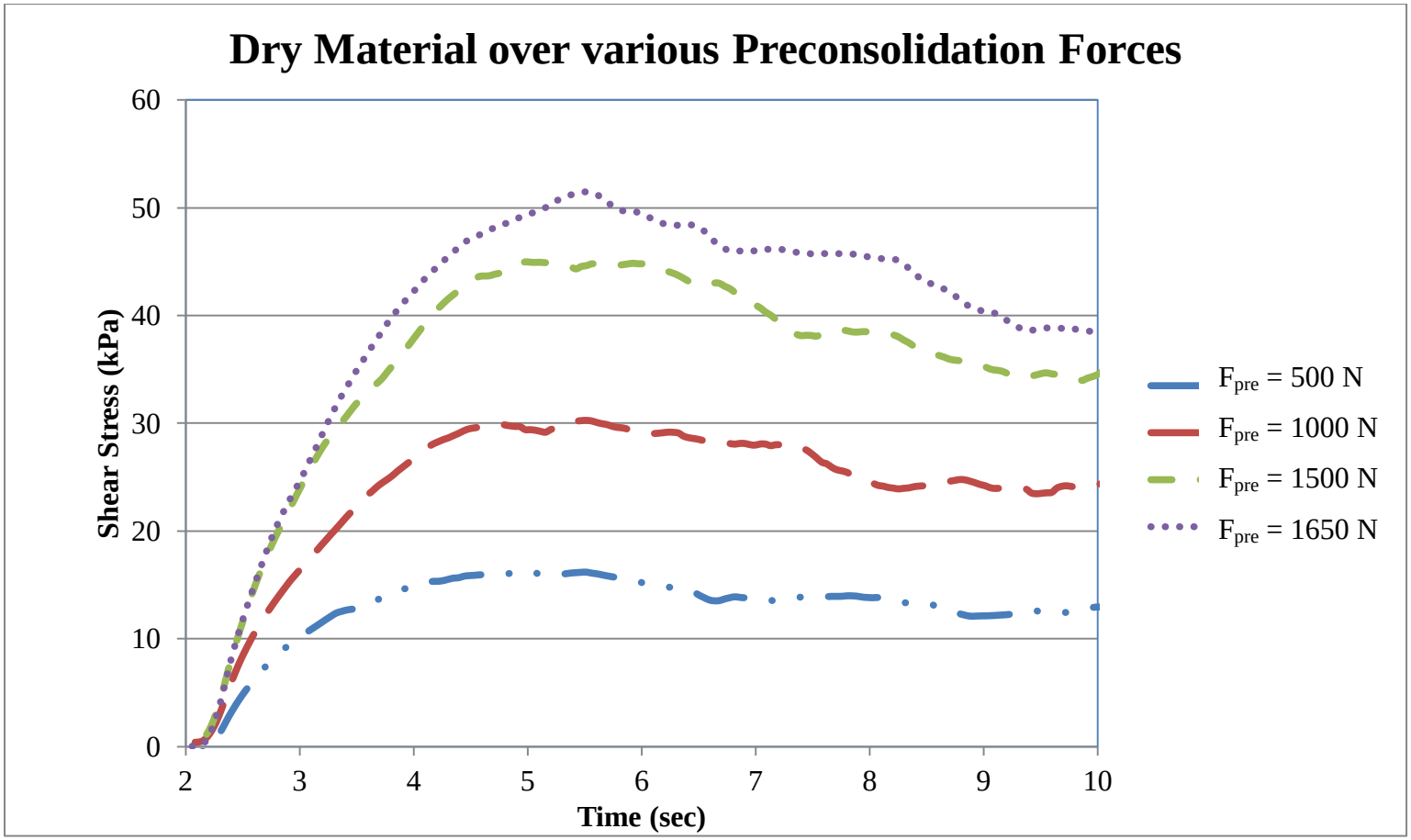


Figure 3.10: Shearing stress of dry material.

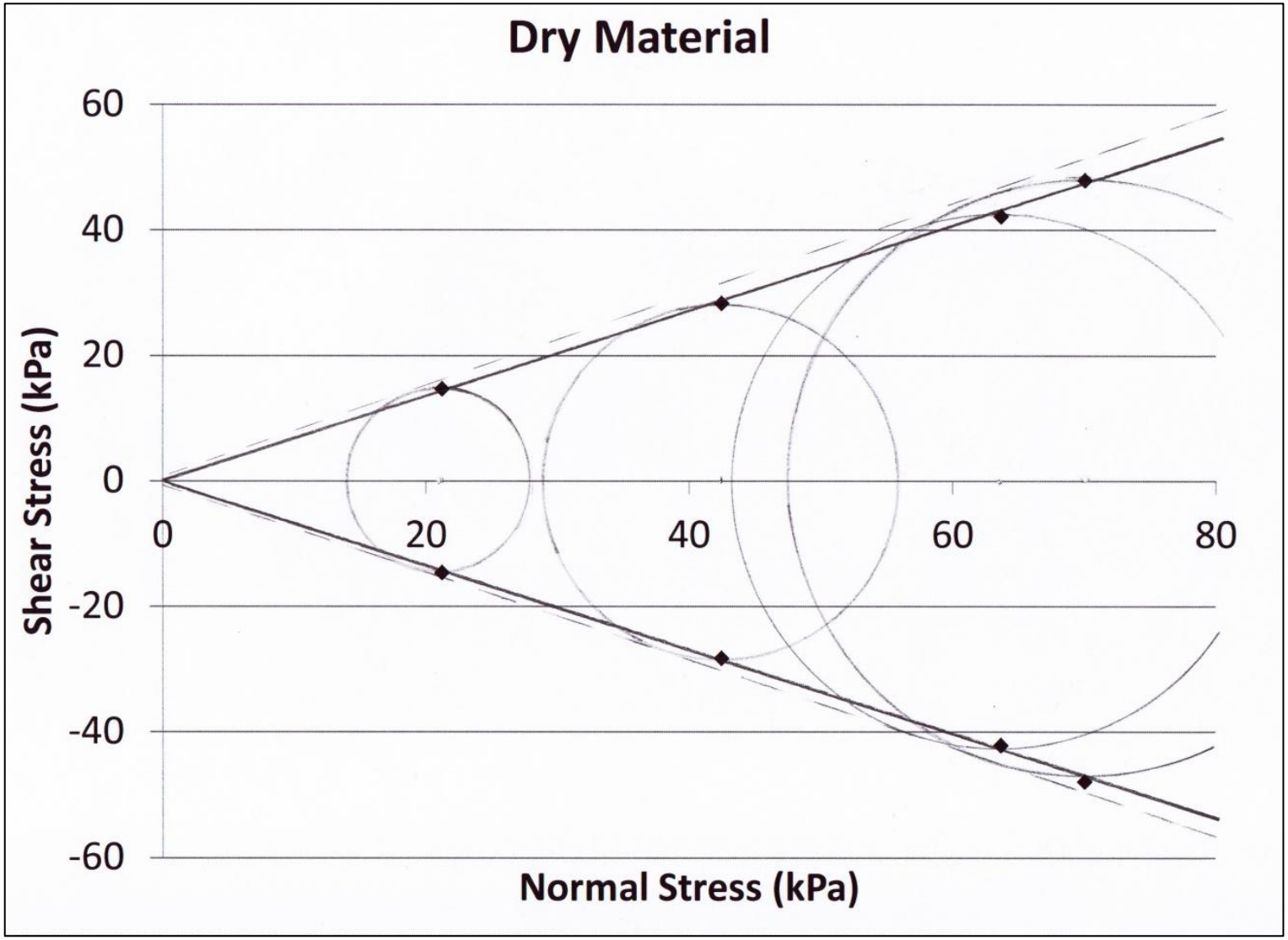


Figure 3.11: Mohr-Coulomb failure envelope of dry material.

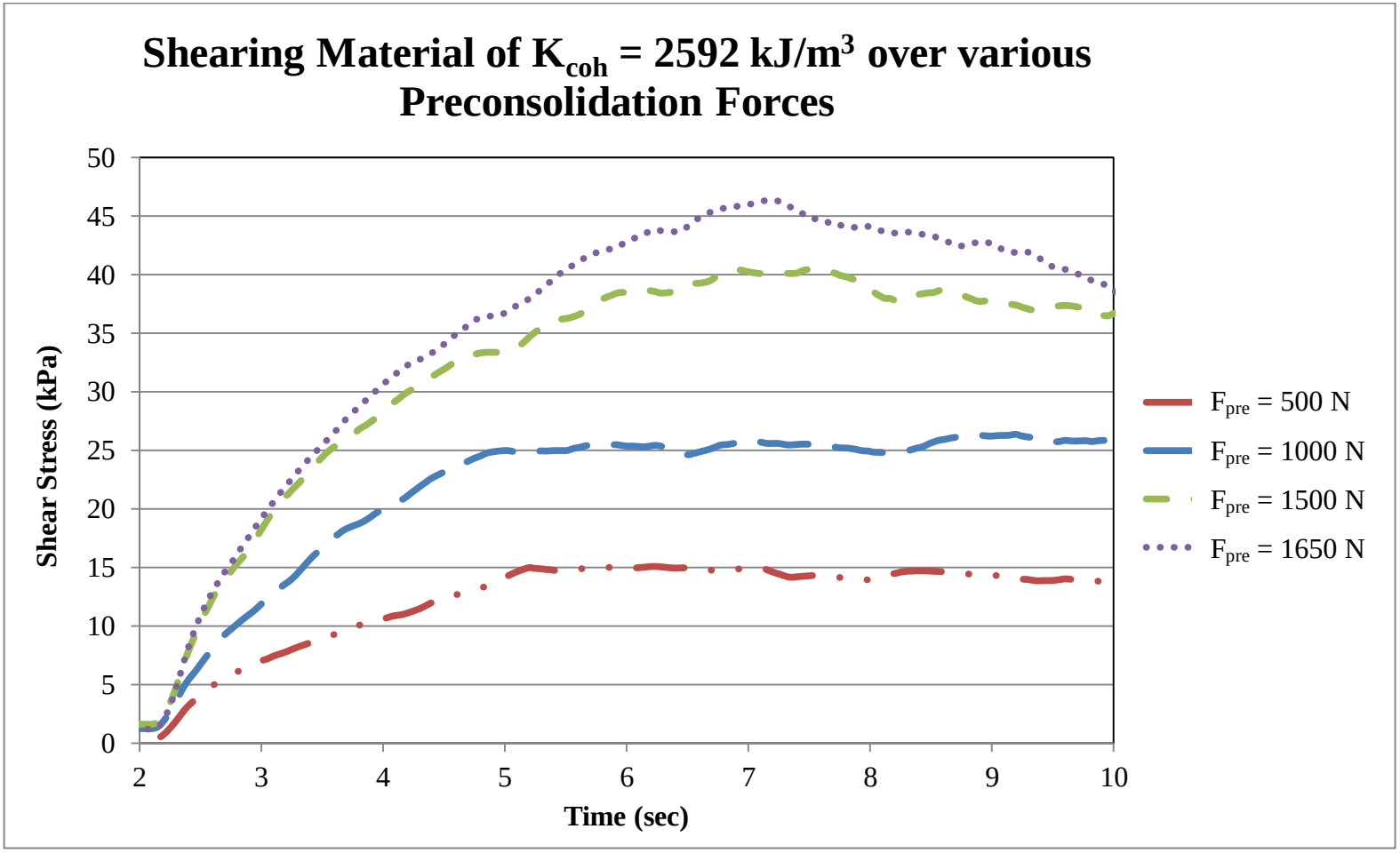


Figure 3.12: Shearing stresses of cohesive material.

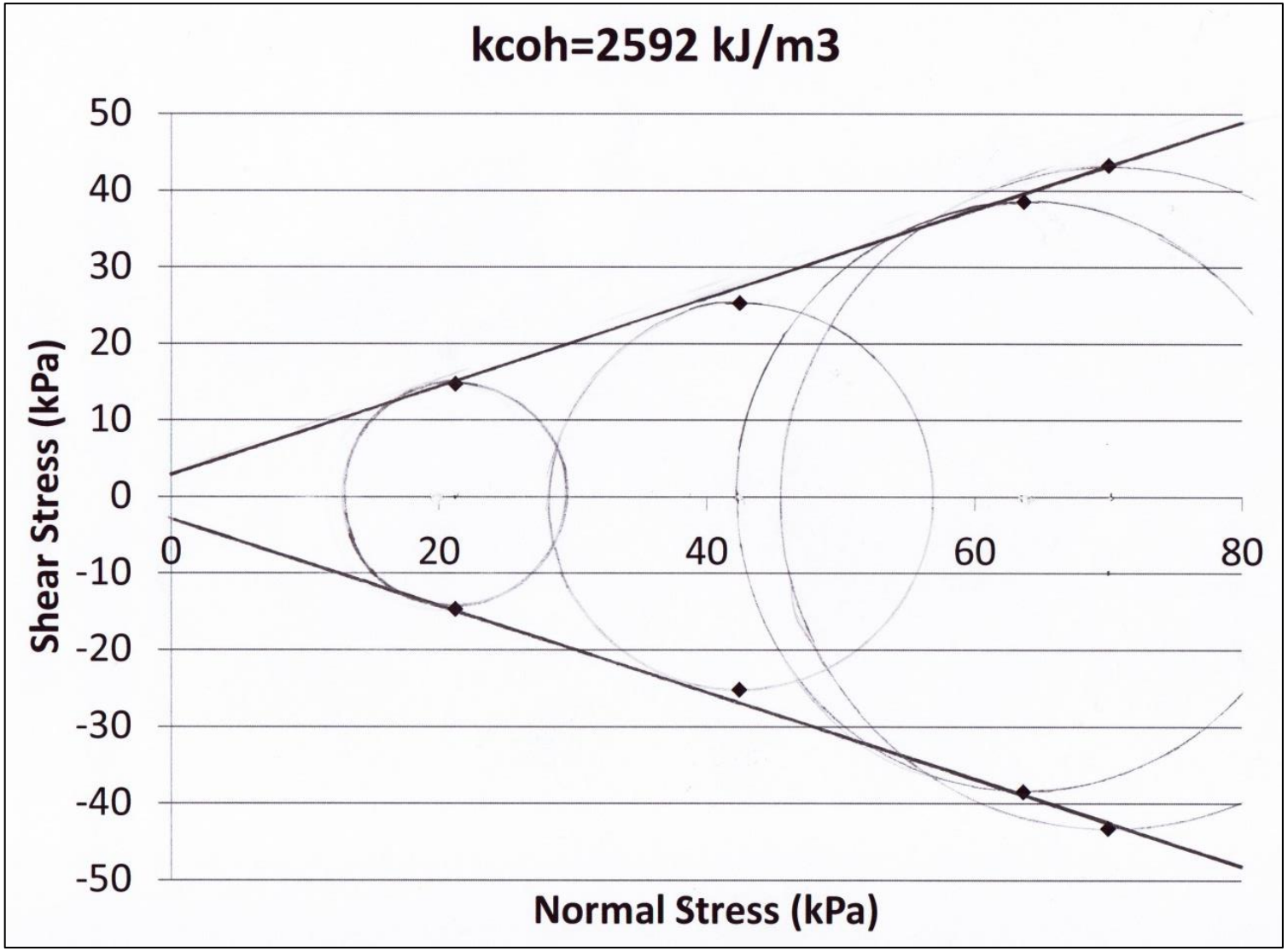


Figure 3.13: Mohr-Coulomb failure envelope of cohesive material.

a. Basic geometry provides the solution for the radius as:

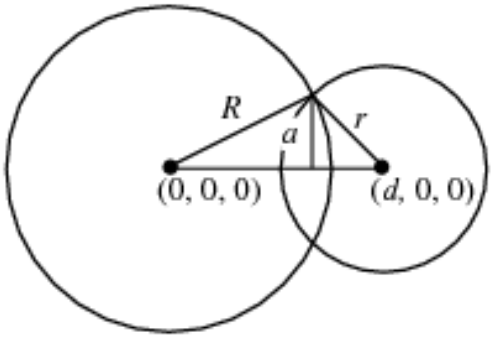


Figure 3.14: Contact radius between impacting particles from [4].

$$a = \frac{1}{2d}\sqrt{4d^2R^2 - (d^2 - r^2 + R^2)^2} \tag{3.13}$$

where d is the distance between centers of the impacting particles.

The current procedure allows for a parametric study of the angle of internal friction at steady-state σ_{sf}. The angle of internal friction is the angle produced by the straight line drawn from the origin to the shear stress at pre-shear τ_{pre}. At this time, only the steady-state properties are calculated due to the difficulty of modeling incipient flow using DEM. The simulation cannot produce the yield locus at a given pre-shear normal stress (σ_{pre}). If the complete state of stress at steady-state is known (i.e. the normal and shear stresses in two mutually orthogonal planes), one can plot the Mohr's circle. This would enable us to compute the effective cohesion and the characteristic consolidation stress. Properties such as the unconfined yield strength and effective cohesion cannot directly be determined using this model [37]. The graphical representations provided here are of pre-consolidation pressures to obtain a rough estimate of the failure envelope. The Hertzian contact model along side of the modified JKR can not model stress history dependent stiffness and is purely elastic.

3.2.3 The Angle of Repose via Active Failure

The simplest material flow test that can be performed is arguably the angle of repose test. This test is used to determine the contour of a pile of unconsolidated bulk solid material

and requires a container of loose material to be filled, the material released and the angle at which it settles measured. A numerical study of the angle of repose of cohesive material was performed under the active failure model proposed by Zhou et al. (2002) [15]. Simulations were carried out in a rectangular container with a fixed middle plate and two side outlets used for discharge. The geometrical details are shown in Figure 3.9. The container size can be scaled up or down corresponding to the particle diameter used. For this test, a material diameter size of ranging from 6.35 mm to 3.17 mm was selected. A simulation was started with the random generation of spheres without overlaps in the container above the fixed middle plate with two outlets closed, followed by a gravitational settling process of 1.0 second to form a stable packing (Figure 3.15a). Once settling is verified the zero remaining kinetic energy in the system, the instantaneous opening of the outlets starts a discharging process in which spheres drop into the bottom of the container under gravity. Some spheres remain on the middle plate after the discharging, forming a stable sandpile (Figure 3.15b). The angle of repose is determined from the surface profile of the pile [15]. Table 3.4 lists the variables considered in this study. For convenience, the effect of the cohesion variable was examined within a range on 0.0 to 144.2 $\frac{\text{kJ}}{\text{m}^3}$, while the other variables were fixed. The dry material provides the base condition.

Table 3.4: Variables and parameters considered.

Name of Parameter/Variable	Symbol	Base Value
Time step	$\triangle t$	1.1573×10^{-5}sec
Particle Diameter	d	$[6.35 - 3.175\,\text{mm}]$
Rolling Friction Coefficient	μ_r	0.3
Sliding Friction Coefficient	μ_s	0.5
Density	ρ	1277.5 $\frac{\text{kg}}{\text{m}^3}$
Poisson's Ratio	ν	0.3
Young's Modulus	E	5.140 MPa
Coefficient of Restitution	e	0.2

For the parameters listed in Table 3.4, five simulations were performed. The first simulation models dry material while the remaining four modeled material with increasing co-

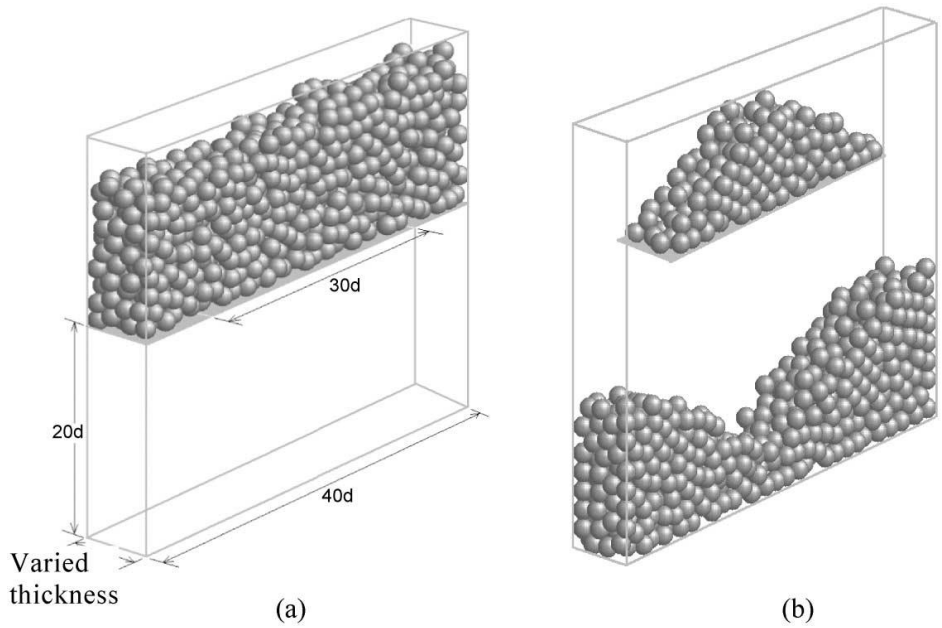
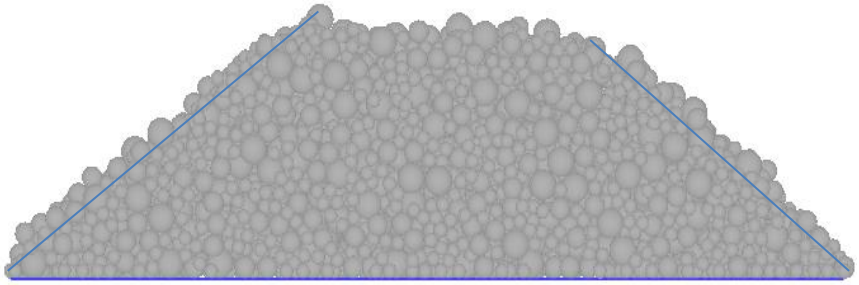


Figure 3.15: Geometry and formation of a sandpile: d is the particle diameter: a) before discharging b) after discharging. Image from [15].

hesion. Figure 3.16 shows the sandpile formations for the active failure angle of repose test performed.

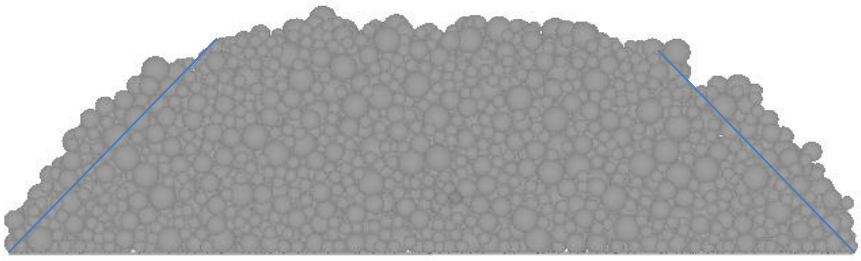
The angle of repose is a macroscopic parameter characterizing the behavior of granular materials. It is related to phenomena such as avalanching and segregation. From the performed numerical simulations, it has been found that the angle of repose strongly depends on material properties such cohesion. However, sandpile formation is dependent on the method of forming a sandpile. The slump test is also a popular approach to determining the angle of repose. In this method, material is placed in a cylinder on a horizontal plane. The cylinder is lifted and the material allowed to settle into the sandpile form. A quantitative description of the dependence that can be used generally in engineering practice is not available and we seek a better understanding of the effect of cohesion in a system to provide a quantitative description the behavior of particulate materials.

Angle of repose: 38 $\pm$ 3 degrees
Cohesion energy: 0.0 kJ/m^3
Dry Material



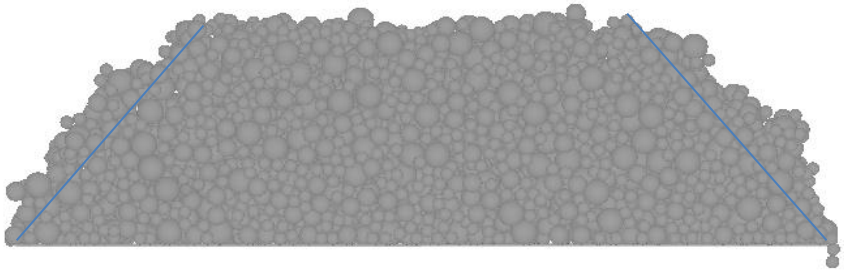
(a)

Angle of repose: 47 $\pm$ 3 degrees
Cohesion energy: 70.7 kJ/m^3



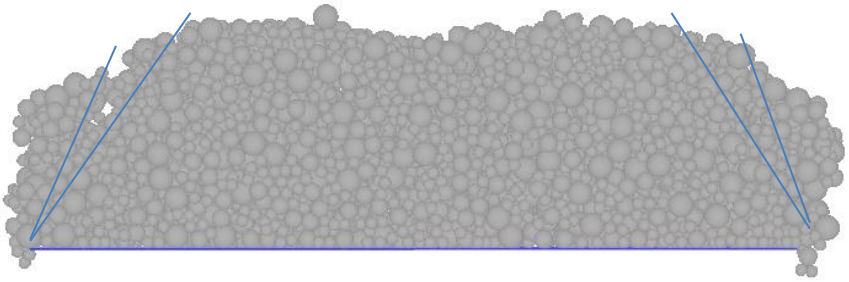
(b)

Angle of repose: 52 $\pm$ 3 degrees
Cohesion energy: 100.8 kJ/m^3



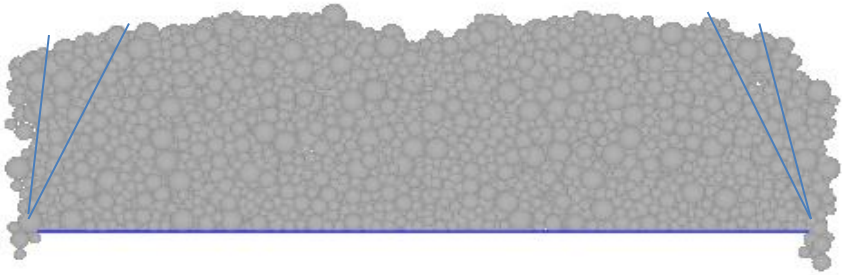
(c)

Angle of repose: 59.5 $\pm$ 6.5 degrees
Cohesion energy: 124.2 kJ/m^3



(d)

Angle of repose: 75$\pm$ 13 degrees
Cohesion energy: 144.2 kJ/m^3



(e)

Figure 3.16: Sandpiles generated via numerical simulations using a distribution of particle sizes with different cohesive values.

3.3 Testing the Flow Properties of Bulk Solids

For free flowing material, the angle of repose is performed as a method of determining how unconsolidated bulk solid material can settle in pseudo-static-flow. When an unconsolidated (loose) bulk solid is deposited on a horizontal surface so as to form a pile and the velocity of the stream onto the top of the pile is negligible, the particles of the solid roll down the pile and the slope of the pile forms an angle of repose with the horizontal. Typically, the angle of repose assumes values between 30 and 40 degrees for dry material [38]. If a solid contains a wide range of particle sizes, it segregates: the fines collect along the trajectory of the pile while the coarse particles roll to the periphery of the pile. When materials drops onto a pile from some height, the fines along the trajectory pack under the impact of the larger particles, gain strength, and form a slope angle steeper than the angle of repose [38].

For a stacked or confined solid, pressures arise under the weight of the superimposed mass and moisture. As particles pack closer together, air is forced out, the particles are brought closer together and cohesive forces develop: the solid consolidates and gains strength. Physically, the ring shear cell can reproduce the higher pressures material can come under. It provides a measurement of how greater consolidation pressures can increase the strength of the bulk solid material. However, consolidation is time dependent and the studied numerical models to not account for this condition. Numerically testing bulk material flow becomes difficult. Improving the DEM numerical models and parameters of cohesive flow for history dependent stresses of bulk solid materials is the center point of this dissertation.

It is necessary to note that further complication arise if during the time of consolidation at rest, the moisture content or the temperature of the material changes. The gain in strength at rest may be caused by any one or a combination of the following factors and are not considered in this research [38]:

- Escape of entrained air with corresponding increase of density.
- Migration of water.

- External vibrations which cause a rearrangement of particles and an increase of density.
- Evaporation of free water with concurrent precipitation of dissolved salts which cement the particles.
- Break-up or softening of particles or crystals under pressure, causing an increase in the surface of contact and cohesion.
- Changes in the surface of particles.

In this study, moisture content is considered uniform throughout the material and particle break-up will be observed by the strength and shearing behavior of the material. For the shear cell test, measurements are made under ambient temperature for two or four moisture contents and one or two times of consolidation. The flow-ability of bulk solid material containing a range of sizes is governed by the flow properties of the fine fraction. This is explained by the fact that during flow the shearing takes place across the fines. In the numerical study, as fines are not modeled, the shearing and flow properties are governed by the frictional properties and shape of the material. In testing the modified versions of the capillary force and JKR model, simulations of the ring cell shear test were unable to provide a direct correlation of the cohesive force to the effective cohesion of the system. The findings have not been satisfactory in modeling the effect cohesion has on flow-ability without some modification to the interaction forces computed for cohesion.

3.4 Numerical Discrete Element Model Parameters

For each of the studies performed, a thorough investigation of the material parameters was performed. The first set of parameters listed describe a material's stiffness, size, shape, surface condition, among others and guide the stability of the system. Table 3.5 lists the main driving parameters for DEM simulations under two sections. The parameters computed by the system are functions of the main material properties and control the damping and natural frequency of the system.

Table 3.5: Numerical DEM model parameters.

Simulation Parameters Given to System	Simulation Parameters Computed by System
Poisson's Ratio, ν	Shear Modulus, G (Pa)
Young's Modulus, E (Pa)	Particle Viscous Damping Coefficient, γ_{damp}
Particle Radius, R (m)	Particle Normal Spring Stiffness, k_n $\left(\frac{\mathrm{N}}{\mathrm{m}}\right)$
Particle Density, ρ $\left(\frac{kg}{m^3}\right)$	Particle Mass, m (kg)
Coefficient of Restitution, e	Contact Area Radius, a (m)
Particle Static Friction Coefficient, μ_s	Penetration Distance, δ_n (m)
Particle Rolling Resistance Coefficient, μ_r	Time Step, $\triangle t$ (sec)
Wall Friction Coefficient, $\mu_{s\,wall}$	Particle Surface Energy, γ_{sur}
Wall Rolling Resistance Coefficient, $\mu_{r\,wall}$	Particle Viscous Damping Torque, M_r^d
Particle Fine Radius, r (m)	Particle Tangential Spring Stiffness, k_t $\left(\frac{N}{m}\right)$

CHAPTER 4
FLOW MODELING OF COHESIVE BULK SOLIDS WITH THE MACRO ELASTO-PLASTIC ADHESIVE (MEPA) MODEL

In the present chapter, the development and implementation of the elastic-plastic adhesive model used to simulated the macroscopic shear flow behavior of cohesive, frictional bulk solids is outlined. The MEPA cohesive model follows the elastic and plastic regimes. It is capable of modeling material yielding through hysteresis and steady-state flow. This chapter describes the details of the contact duration with cohesive attraction for each particle contact.

4.1 The MEPA Cohesive Contact Model

The flow behavior of bulk solids under large deformations and displacements is difficult to model with a particle-particle force law that is solely based upon micro-mechanical considerations. More knowledge at the macro-mechanical scale is necessary to gain insight into flow states that can lead to flow obstructions or no flow. Here, the proposed model uses a maximum force-based failure. It determines the maximum displacement of the contact with a material stiffness described by the material's tensile strength, elastic modulus and Poisson's ratio. The developed contact stresses are also particle size dependent and are computed using the effective radius of the elements in contact [39]. The novel aspect of this MEPA model is that material behavior is described by macro and micro-mechanics such as the material yield limits and physical properties. This model simulates the mechanical behavior of material physical data in a shear test rather than the micro and molecular mechanics laws. It simulates the graphical results from the physical testing used to develop the Mohr-Coulomb shear failure criterion. This work complements studies in cohesive, frictional bulk solids for micro and macro-models of different materials [40–46].

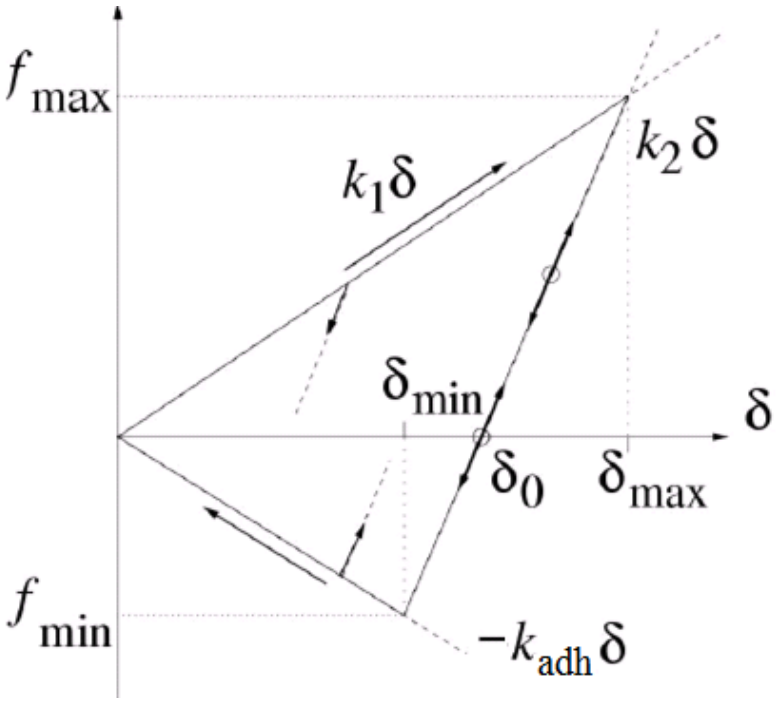
In the implementation of the MEPA model within the DEM algorithm non-uniform sized spherical particles are used throughout. The material roughness is mimicked with a rolling resistance model that applies additional torques and resistances as described in section 2.2 [20]. The MEPA cohesive model implemented in DEM aims to provide a better understanding of the macroscopic flow behaviors of bulk solid materials. In this work, the cohesive bond strength is viewed as a fundamental material parameter and used to numerically resolve macro-mechanical behaviors experimentally determined under the Mohr-Coulomb theory with flow property tests.

4.2 Particle Contact Constitutive Model

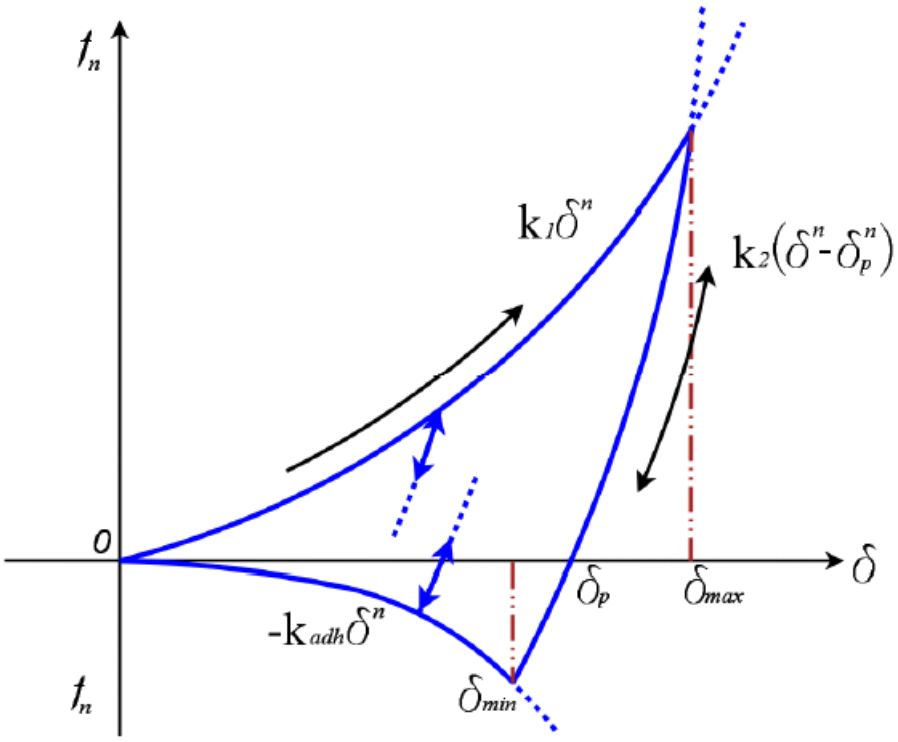
The following sections detail the MEPA model and its implementation. The simulation tests performed to represent physical testing results using this model for copper ore are presented and compared with experimental data in chapter 5. The simulation tests performed in this chapter verify that the macro-flow behaviors of a bulk solid are modeled in detail.

4.2.1 Mathematical Description of the MEPA Contact Model

DEM studies of particulate systems illustrate how the macroscopic response depends on various properties, such as particle size, contact cohesion, friction, and stiffness [47]. However, the realistic modeling of the internal mechanics of a particle contact is too complex to implement. Hence, the relationship between the interaction force and the normal overlap δ of two rigid DEM particles is established to simplify the contact mechanics as seen in Figure 2.3. The interaction force developed during contact is decomposed into a normal and tangential component that is applied. DEM particles are "rigid" but the bulk solid "deforms" the particle pair. The force-overlap diagram for this model is shown in Figure 4.1. The MEPA model takes into account plastic contact deformation and cohesive attraction. As two particles are pressed together the particle contact undergoes elastic and plastic deformations. As they continue to be pressed together the pull-off force increases with the increase of the plastic contact area.



(a) Linear MEPA cohesive contact law.



(b) Non-linear MEPA cohesive contact law.

Figure 4.1: Different MEPA contact models from [7].

The loading, unloading/re-loading and cohesive branches seen in the MEPA model are represented by four parameters: the virgin loading parameter k_1, the unloading and reloading parameter k_2, the cohesive parameter k_{adh} and the index parameter n, controlling the order of the system [43–45]. In the initial loading of the contact, the force increases with stiffness k_1. A linear viscous damping dash-pot is used for energy dissipation during contact. Elasticity is added by a spring, with a larger stiffness, k_2. The stiffness increases due to the irreversible, plastic contact deformation. Cohesion between the contacts is represented by cohesive stiffness k_{adh}, which allows for attractive forces up to a minimal force f_{min}. The non-linearity or shape of the three branches is controlled by the index parameter n. When $n = 1$, the model becomes linear and is represented by the branched model of Figure 4.1a. If k_1 is set equal to k_2, the model is reduced to the linear or Hertzian contact model previously discussed. Each branch can be expressed by the following sets of bounding equations:

$$f_1(\delta) = k_1 \delta^n \tag{4.1}$$

$$f_2(\delta) = k_2\left(\delta^n - \delta_p^n\right) \tag{4.2}$$

$$f_3(\delta) = -k_{adh}\delta^n \tag{4.3}$$

where $f_1(\delta)$ represents the virgin loading branch, $f_2(\delta)$ the re/unloading, and $f_3(\delta)$ the cohesive attraction. The branched relationship as a whole can be expressed as:

$$f_{hys} = \begin{cases} f_1(\delta) & if\ f_2(\delta) \geqslant f_1(\delta) \\ f_2(\delta) & if\ f_1(\delta) > f_2(\delta) > f_3(\delta) \\ f_3(\delta) & if\ f_3(\delta) \geqslant f_2(\delta) \end{cases} \tag{4.4}$$

The normal force on particle i is described by:

$$\mathbf{f}_0^n = -\gamma_n \mathbf{v}_n + f_{hys}\mathbf{n} \tag{4.5}$$

with the normal direction unit vector $\hat{n}$ directed from the center of particle j to particle i. The variable $\mathbf{v}_n$ describes the normal relative velocity of the particle and γ_n the viscous dissipation of the system. The tangential force includes dissipation due to Coulomb friction and tangential elasticity that allows for stick-slip behavior at the contact level [40, 45, 46].

The tangential force is related to the normal force via Coulombs law in equation 4.6.

$$f^t \leq \mu_s f_{hys} \tag{4.6}$$

The overall solution of the non-linear DEM problem is obtained by incrementally solving Newton's equations of motion with the MEPA cohesive model.

4.2.2 DEM Implementation of the MEPA Contact Model

In this study a value of $n = \frac{3}{2}$ is used. This converts the MEPA cohesive model into a non-linear hysteretic spring contact model. The maximum adhesion is determined by the stiffness parameters and the maximum normal overlap δ_{max}. The tangential stiffness is calculated based on the contact stiffness k_t, which is set to the value of k_1. The tangential force is calculated from the product of the tangential stiffness and the tangential displacement, subject to the frictional limit according to Coulomb's law. This contact model has been implemented through LIGGGHTS version 2.3.8, an open source code by CFDEMprojects. Following the branches of the MEPA cohesive model, during initial compressive loading, the contact force increases proportionally with the contact overlap. At the maximum contact overlap, δ_{max}, the contact stiffness increases instantaneously to the value k_2. Further loading and un-loading is defined by the force-displacement relation $f = f_2(\delta)$. Elastic unloading to a zero contact force leads to a non-zero contact overlap equal to the maximum plastic contact indentation, $\delta = \delta_p$, which is recorded and updated over the contact lifetime. When the contact overlap is further decreased as the particles separate, the contact force enters the tensile regime. The maximum tensile contact force $f_{max} = -k_{adh}\delta_{min}^n$ that the contact can experience corresponds to a contact displacement $\delta = \delta_{min}$. The final part of the tensile regime created by the cohesive attraction force is characterized by a softening branch, with the tensile contact force decreasing from its maximum value towards zero in accordance with $f = -k_{adh}\delta$. In addition to the loading and unloading branches shown in Figure 4.1a and b, loading and unloading may also occur within the bounding branches. Any loading stage within the bounding branches loads in accordance to the stiffness k_2 [47].

4.3 Development of the Model System

The development of the contact model consisted of four subsequent stages, namely:

1. Coding development of the iterative MEPA algorithm.

2. Particle impact test of the loading stages.

3. Compressive loading under short-range particle interactions with material yielding.

4. Loose sandpile formations.

4.3.1 Coding Development of the Iterative MEPA Algorithm

The implementation of the MEPA cohesive contact model in C++ on the parallel platform of LIGGGHTs allows for larger, more complex problems to be solved. This section describes the DEM implemention of the algorithm on the Aun partition of the supercomputer BLUEM at CSM. The files of customized code developed can be found in Appendix C. The appendix provides the code used for this study for both particle-particle and particle-plane contact. The code is embedded in the structure of contact laws for LIGGGHTS version 2.3.8. A sample of the script files used for each of the testing simulations are also provided in Appendix C.

The following section describes the logic programmed to describe the algorithm's loading and unloading branches. A flow diagram of the code structure is given by Figure 4.2 for clearity of particle interaction conditional flow. The first step is to update the particles in the system and then compute the interactions.

1. *No Particles in contact.* As particles initially come into contact the particle displacement is determined and the new penetration computed. Recall, the penetration is computed by equation 2.6. If the penetration is negative, then the particles are not in contact and plastic deformation history between the particles is cleared and the interation force set to zero.

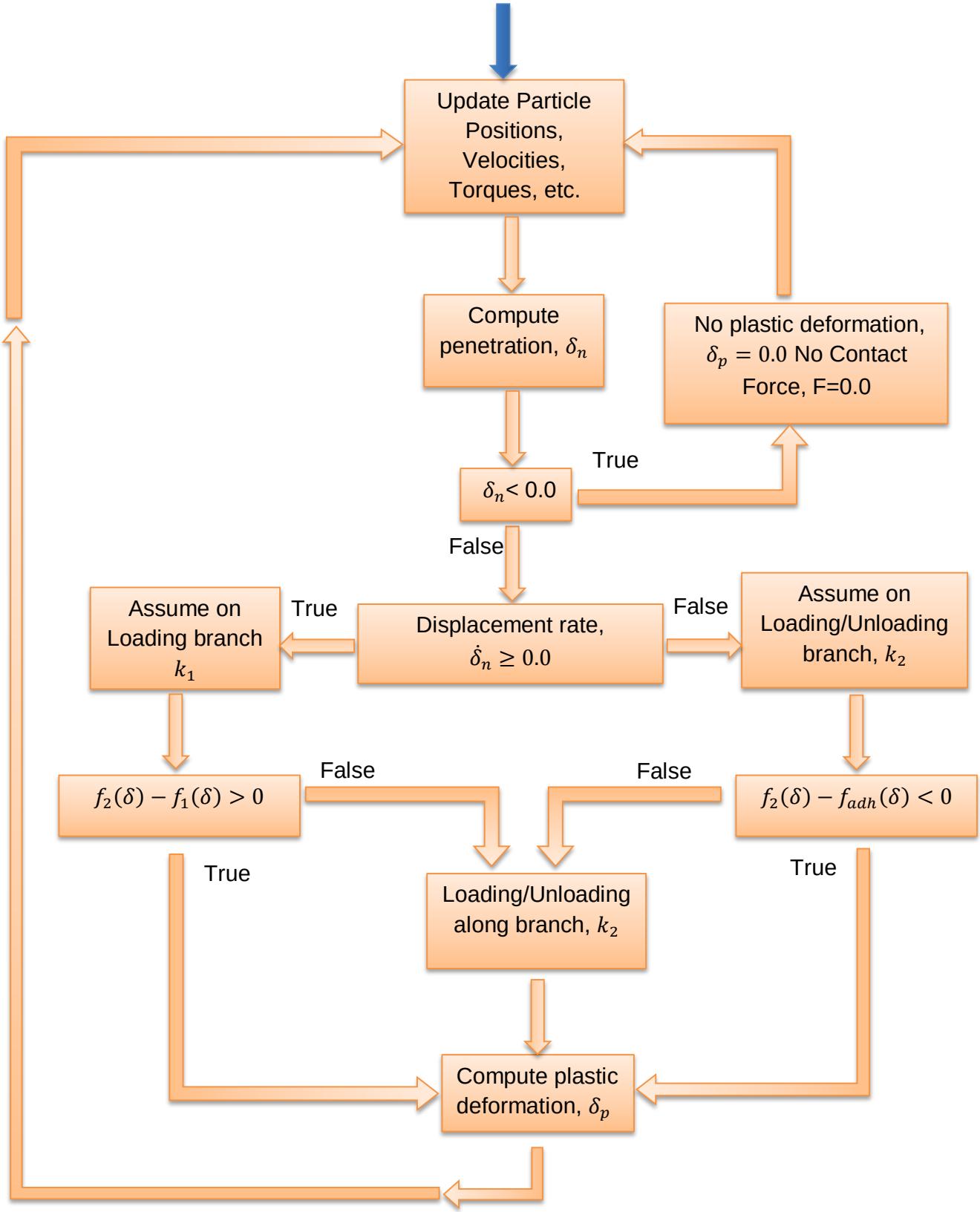


Figure 4.2: The MEPA cohesive contact algorithm flow diagram.

2. *Virgin loading.* If the particles are in contact, $\delta_n > 0$, then it is first assumed that the interaction is loading or unloading along a branch with slope k_2 as seen in Figure 4.3 from point P to point Q. If the particles are in loading (approaching each other), have no contact history and the plasticity deformation history of the interaction does not exceed the maximum allowable contacting loading force of the current penetration, then the particles load along the virgin loading curve, k_1. The deformation between the particles is computed and stored. The system will then return to update the timestep and location of the particles.

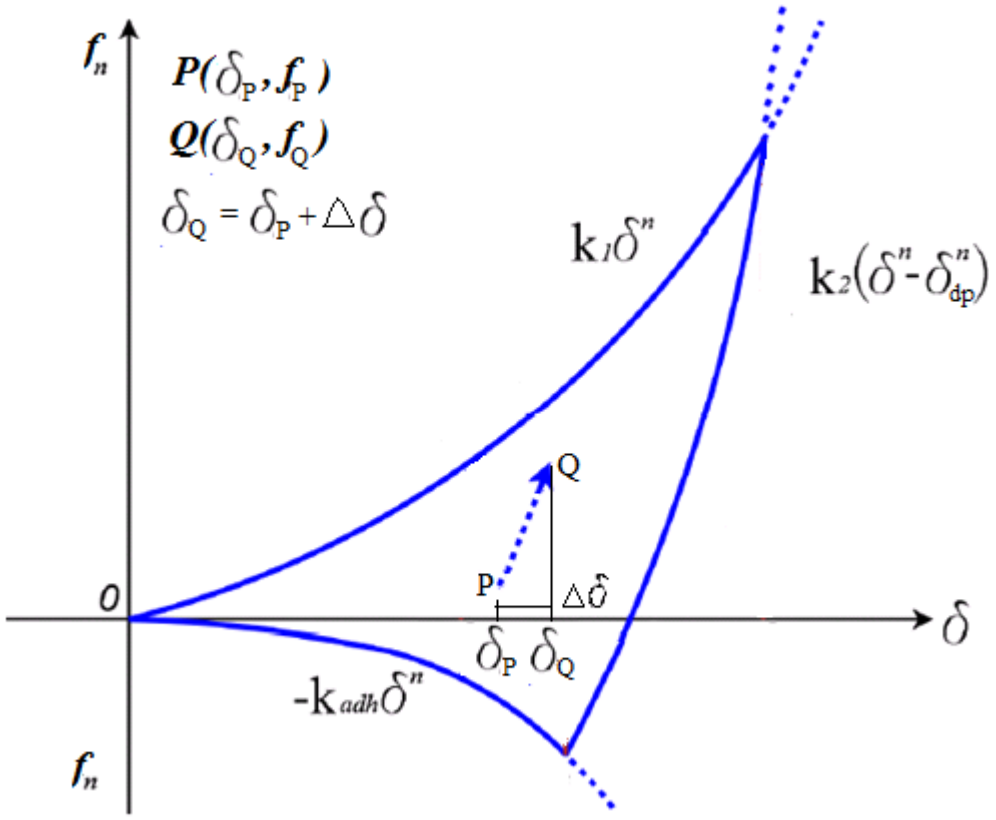


Figure 4.3: Loading along slope of $k_2$within the bounding branches of the MEPA cohesive contact model.

3. *Contact reloading.* If particles with contact history start to reload (further approach each other), the contact starts at point P within the bounding curves and reloads to point Q as seen in Figure 4.3. If, however, the reloading from P exceeds the bounds

of the MEPA cohesive contact model (point Q') for that interaction, a correction is performed to bring it to point Q as seen in Figure 4.4. The contacting force on the particles is now based on the penetration, $\triangle\delta_1$, along the reloading curve, k_2, and the penetration, $\triangle\delta_2$, along the bounding curve, k_1, which illustrates the deformation history and the particle contact penetration.

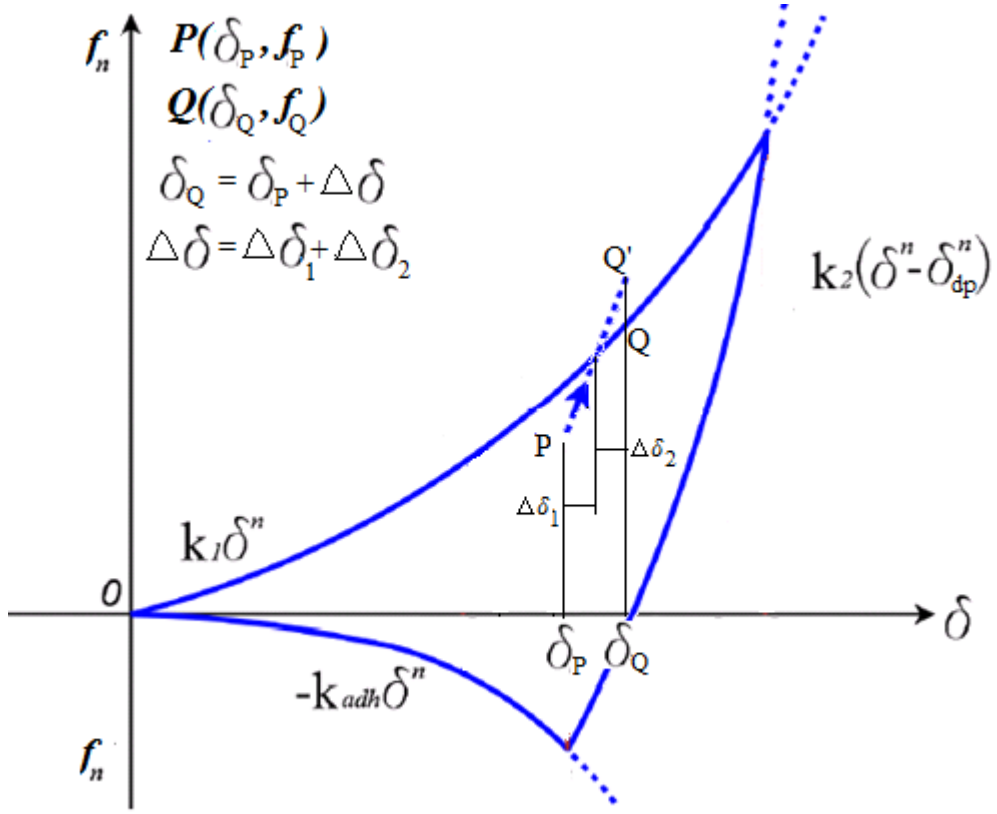


Figure 4.4: Loading along slope of $k_2$with a correction on the transition of bounding branches of the MEPA cohesive contact model.

4. *Contact unloading.* If the particles are unloading, a similar approach to reloading is taken to determine along which branch, $k_2$or k_{adh}, the particles are unloading. If during unloading, the unloading force computed exceeds the bounds of the unloading curves, then the transitioning plasticity and penetration is computed and the system force updated.

The coded structure is as described as follows:

compute change in penetration: $\triangle\delta = \dot{\delta}_n \triangle t$

compute new particle penetration: $\delta = \delta_{old} + \triangle\delta$

if $\delta <$ 0.0 then no contact between particles and $\delta_{dp} =$ 0.0 and $f_{hys} = 0.0$

else contact

compute $f_2\,(\delta,\, \delta_{dp})$ assuming the particles are in loading or unloading

if $\dot{\delta}_n \geq 0$ then particles are loading

check were on bounding branches the contact is located: $f_2\,(\delta,\, \delta_{dp})$ -$f_1\,(\delta) >$ 0.0

if $f_2\,(\delta,\, \delta_{dp})$ -$f_1\,(\delta) >$ 0.0 then need to make deformation correction: $\delta_{dp} = \delta \cdot \left(\frac{k_2 - k_1}{k_2}\right)^{-1/n}$

check against maximum deformation

if maximum deformation not reached then $f_{hys} = f_1\,(\delta)$

otherwise back to loading along k_2 and $f_{hys} = f_2\,(\delta,\, \delta_{dp})$

else loading along k_2 and $f_{hys} = f_2\,(\delta,\, \delta_{dp})$

if the particles are unloading

check were on bounding branches the contact is located: $f_2\,(\delta,\, \delta_{dp})$ -$f_3\,(\delta) >$ 0.0

if $f_2\,(\delta,\, \delta_{dp})$ -$f_3\,(\delta) <$ 0.0 then need to make deformation correction: $\delta_{dp} = \delta \cdot \left(\frac{k_2 + k_{adh}}{k_2}\right)^{-1/n}$

check against maximum deformation

if maximum deformation reached then $f_{hys} = f_3\,(\delta)$

otherwise back to unloading along k_2 and $f_{hys} = f_2\,(\delta,\, \delta_{dp})$

else loading along k_2 and $f_{hys} = f_2\,(\delta,\, \delta_{dp})$

The full files of customized code developed can be found in Appendix C.

4.3.2 Particle Contact Testing of the Loading Stages

To ensure the contacts followed the loading stages, the contact law was first explored for the linear case under particle contact. Using an index parameter $n = 1$ allowed for the simplification and quick identification of the particle contact stage. The particle shape used in this study is spherical. Here, a particle was released from a height of the particle's radius under gravity and the forces along the loading stages observed. The resulting contact forces for the impacting particle can be seen in Figure 4.5. The parameters used in the simulation are listed in Table 4.1. The analysis shows that the implemented system follows the contact MEPA cohesion model proposed. The same particle contact test was performed using an index parameter $n = \frac{3}{2}$ verify the loading stages of a non-linear MEPA model. The particle contact trace can be seen in Figure 4.6. It can be concluded that the resulting loading stages trace the contact system properly and the system has been implemented and is functioning as proposed.

Table 4.1: Input parameters for full contact system.

Particle Density, ρ $\left(\frac{kg}{m^3}\right)$	1957.5
Loading Spring Stiffness, k_1 $\left(\frac{N}{m}\right)$	3.05×10^5
Unloading Spring Stiffness, k_2 $\left(\frac{N}{m}\right)$	4.14×10^5
Cohesive Stiffness, k_c $\left(\frac{N}{m}\right)$	1.20×10^5
Particle Static Friction, μ_s	0.3
Particle Rolling Friction, μ_r	0.2
Particle Radius, r (mm)	50
Simulation Time step, $\triangle t$ (sec)	5.47×10^{-6}

4.3.3 Compressive Loading under Short-Range Particle Interactions with Material Yielding

The flow-ability of bulk solids is usually measured using the relationship between the unconfined yield strength (σ_c) and the consolidation stress (σ_1). Hence, the contact law was then explored under compressive loading to ensure material shear failure. The material simulated represents high rolling bulk solids with a maximum elastic contact stiffness of

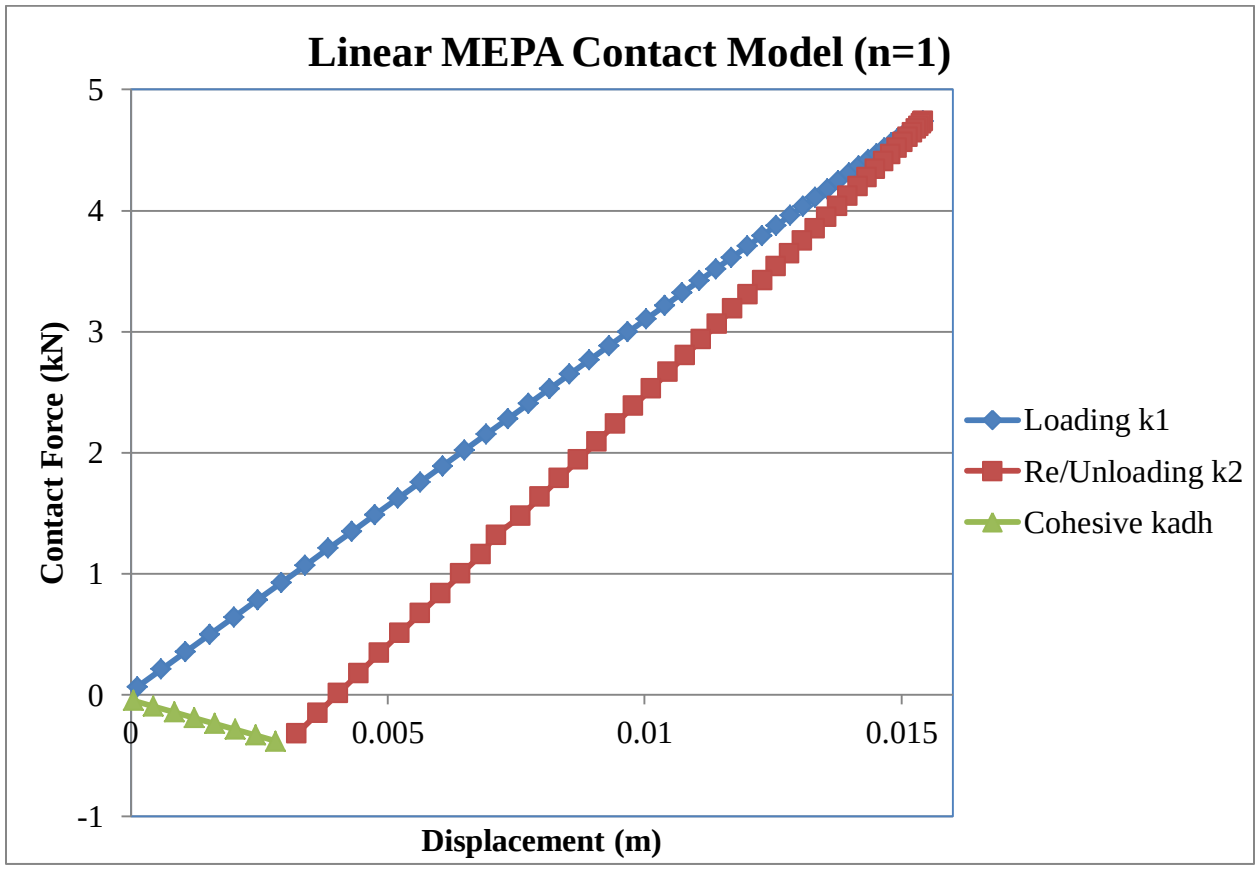


Figure 4.5: Axial strain and force for impacting particle with linear MEPA cohesive contact model.

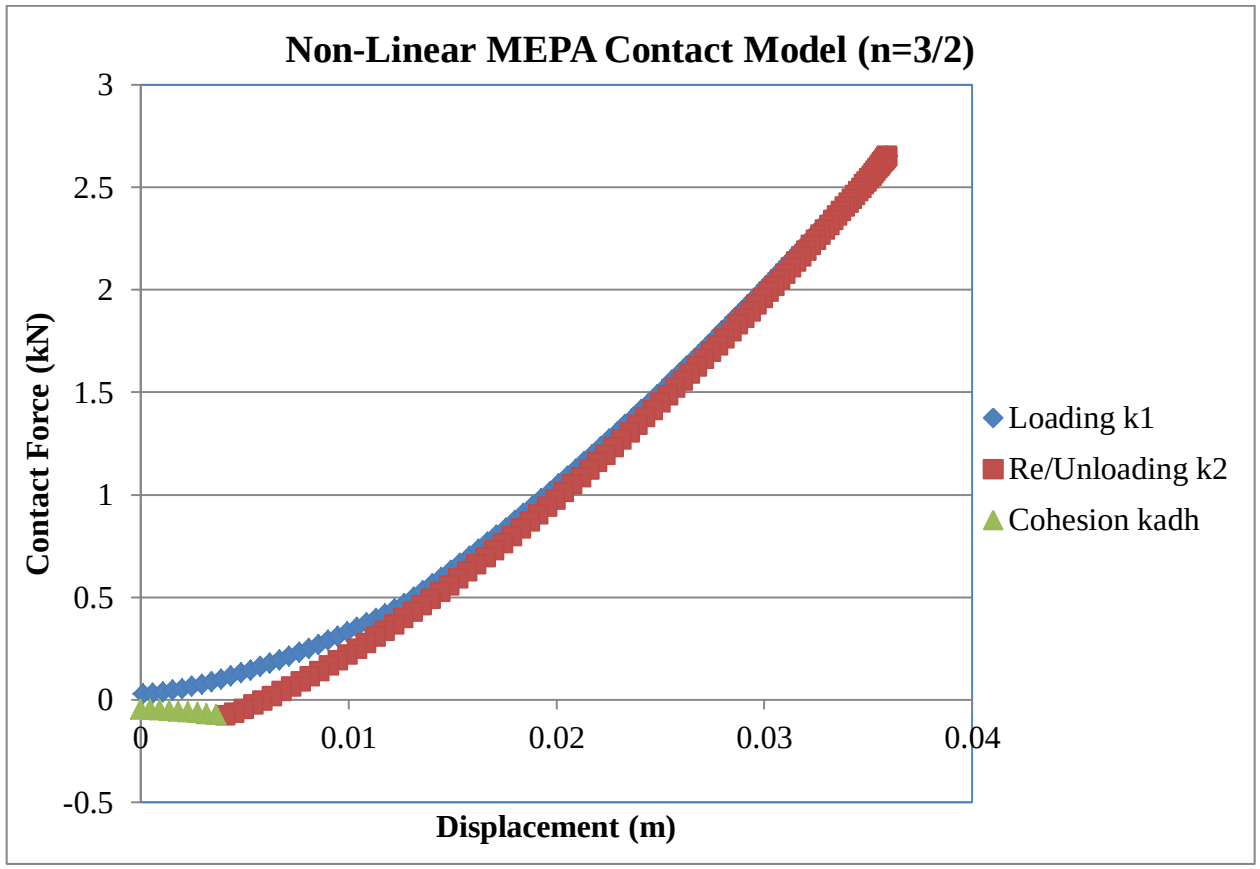


Figure 4.6: Axial strain and force for impacting particle with non-linear MEPA cohesive contact model.

$k_2 = 2.14 \times 10^7 \frac{\mathrm{N}}{\mathrm{m}}$. In this study, k_2 is a value dependent on the maximum overlap that can be experienced by the material and accounts for an increasing stiffness with deformation [48].The initial elasto-plastic stiffness k_1 is 50% of k_2 for this study and is congruent with the the work performed by Luding [47, 49]. The adhesion strength k_{adh} is set to 25% and 50% of k_2 and represents two different material conditions. The first a weak cohesive bond and the second a cohesive attraction equal to the plasticity stiffness k_1. The maximum plastic contact deformation for two contacting particles is computed as:

$$\delta_p = \left(\frac{k_2}{k_2 - k_1} \right)^{\frac{1}{n}} \delta_{max} \tag{4.7}$$

with the maximum elastic overlap δ_{max} equal to 10% of the effect radius of the contacting particles. This value is chosen because it is assumed that the maximum elastic overlap δ_{max} is much smaller than the radius of the contacting particles $\delta_{max} \ll R^*$. The structure followed for the ring cell shearing test is the same as described in section 3.2.2. The numerical sample preparation is carried out by compressing a loose assembly of particles in a shearing cell. The cell in uniaxial compression is set to a specified control force (CF) before slowly shearing the material to failure. The material shearing results can be seen in Figure 4.7 for the material listed in Table 4.2.

Table 4.2: Material properties for consolidated shearing.

Particle Density, ρ $\left(\frac{kg}{m^3}\right)$	1957.5
Loading Spring Stiffness, k_1 $\left(\frac{N}{m}\right)$	1.07×10^7
Unloading Spring Stiffness, k_2 $\left(\frac{N}{m}\right)$	2.14×10^7
Cohesive Stiffness, k_c $\left(\frac{N}{m}\right)$	5.35×10^6, 1.07×10^7
Particle Static Friction, μ_s	0.25
Particle Rolling Friction, μ_r	0.1
Particle Viscous Damping, γ	0.1
Particle Radius, r (mm)	2.86
Simulation Time step, $\triangle t$ (sec)	2.08×10^{-7}

The results show that after some initial loading and deformation the material fails into steady state for each confinement pressure. A Mohr-Coulomb shear failure envelope is

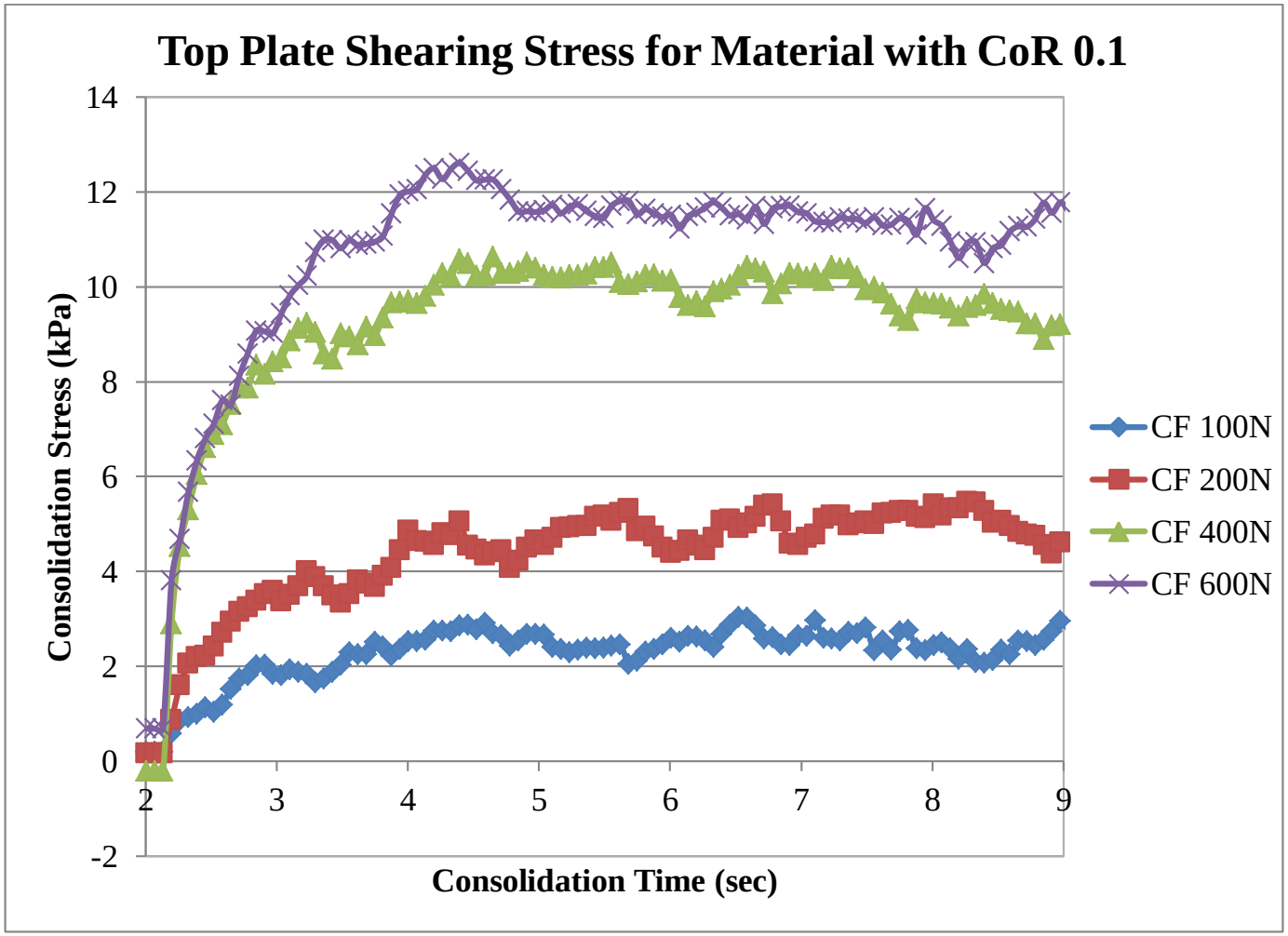


Figure 4.7: Material failure predicted with the MEPA contact model for different consolidation pressures for the servo CF lid.

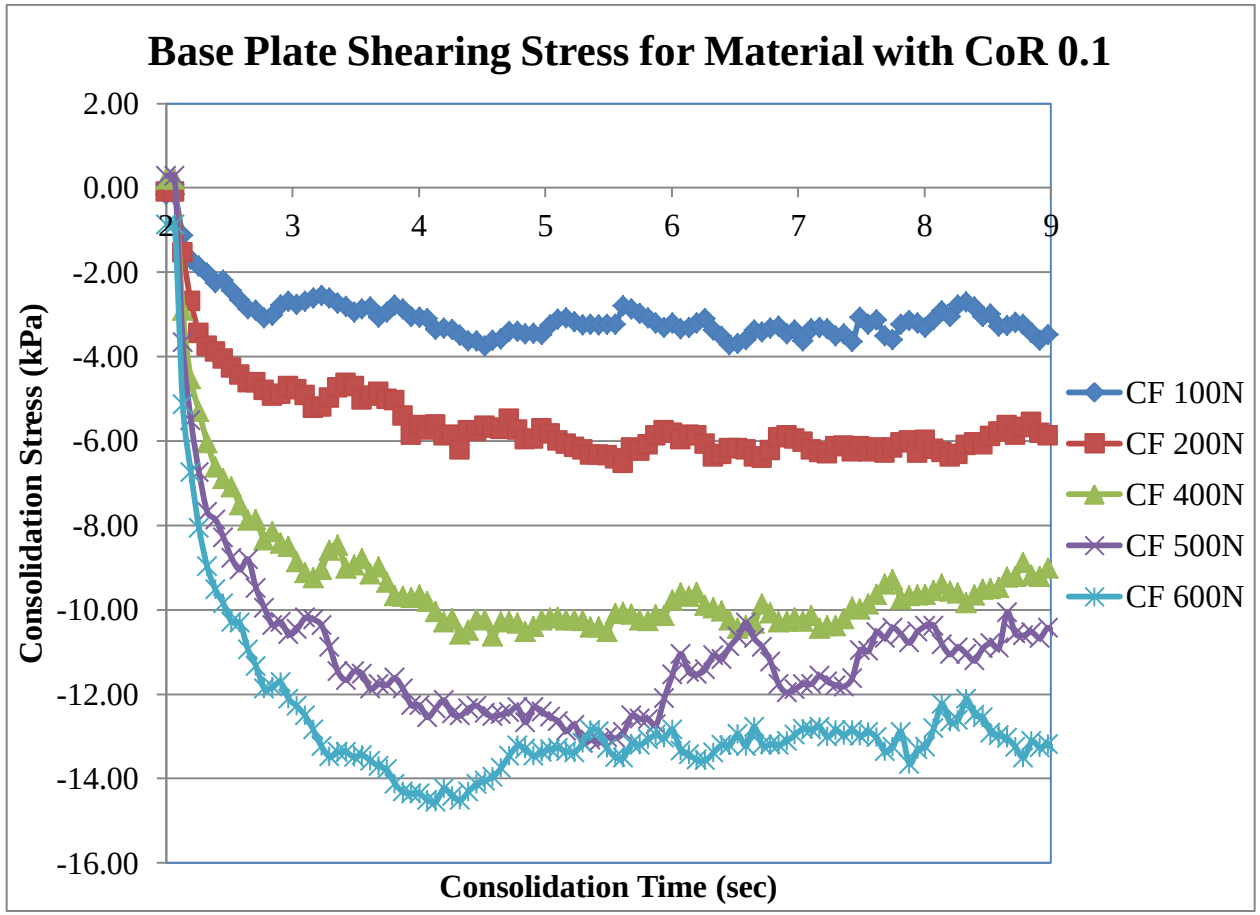


Figure 4.8: Material failure predicted with the MEPA contact model for different consolidation pressures for the shearing base plate.

graphed to show that an effective cohesion can be observed in the system in Figure 4.9. Several more studies were performed for a number of rolling friction conditions as well as for dry material and a cohesive stiffness of $k_{adh} = 1.07 \times 10^7 \frac{N}{m}$. The resulting Mohr Coulomb failure envelopes can be seen in Appendix A. Steady-state shearing is reached for all of the material conditions as expected. The Mohr-Coulomb Shear Failure envelopes also show the expected effective cohesive behavior, such as increased effective cohesion and zero cohesion for the dry material condition. For the variations in the material's rolling friction, as the roughness increases, the shearing failure stress also increases as is expected with the variation of rolling friction.

4.3.4 Contour Formations of Loose Sandpiles

The next validation simulation performed was that of loose sand pile formations for the material described in Table 4.2. This test, also known as the angle of repose test, is used to determine the contour angle of a pile of unconsolidated bulk solid material. The test calls of a container of loose material to be filled and then container walls released. The angle with the horizontal at which the material settles measured. Several simulations with varying rolling friction and cohesive stiffness k_{adh} were carried out in a rectangular container as described in section 3.2.3. A simulation was started with the random generation of spheres without overlaps in the container followed by a gravitational settling process of 1.0 second to allow for stable packing. After discharge, the material remaining on the middle plate forms a stable sandpile. Figure 4.10 shows the result of three simulations with varying rolling friction coefficients. The range studies varied from $0.1 \leq \mu_r \leq 0.9$. This test was also performed for dry material and material under a cohesive stiffness of $k_{adh} = 1.07 \times 10^7 \frac{N}{m}$. The results can be found in Appendix B. The results show that with increased surface roughness, the pile angle also increased as seen in Table 4.3. Measuring the angle with increases cohesion in the system proved more difficult. However, it was be seen that the curvature of the slopes of the piles increase with increased cohesion.

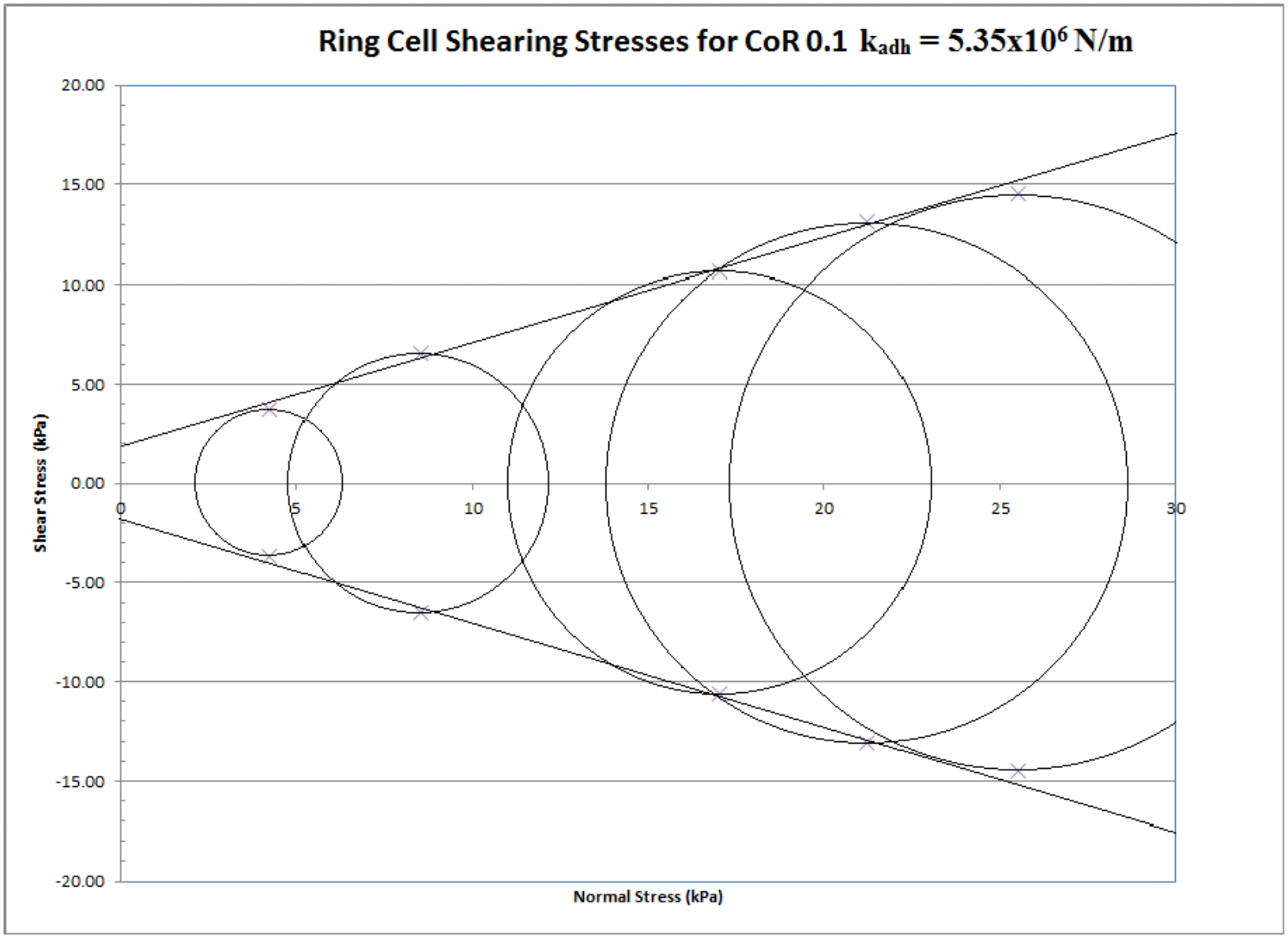
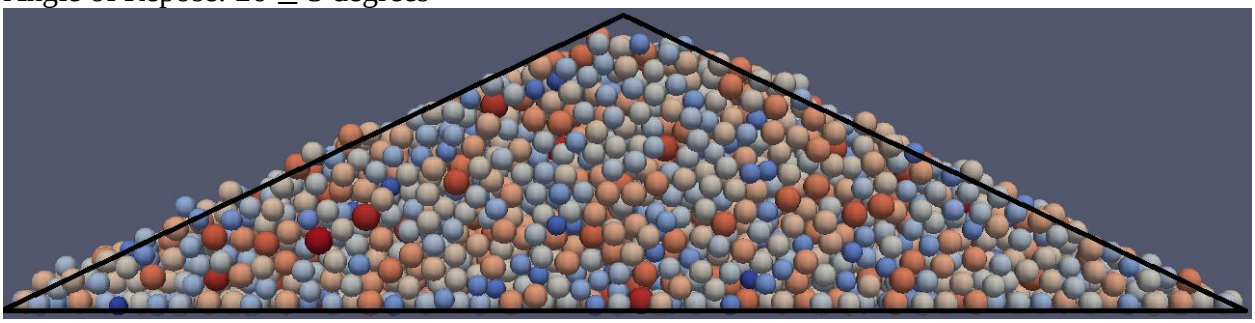


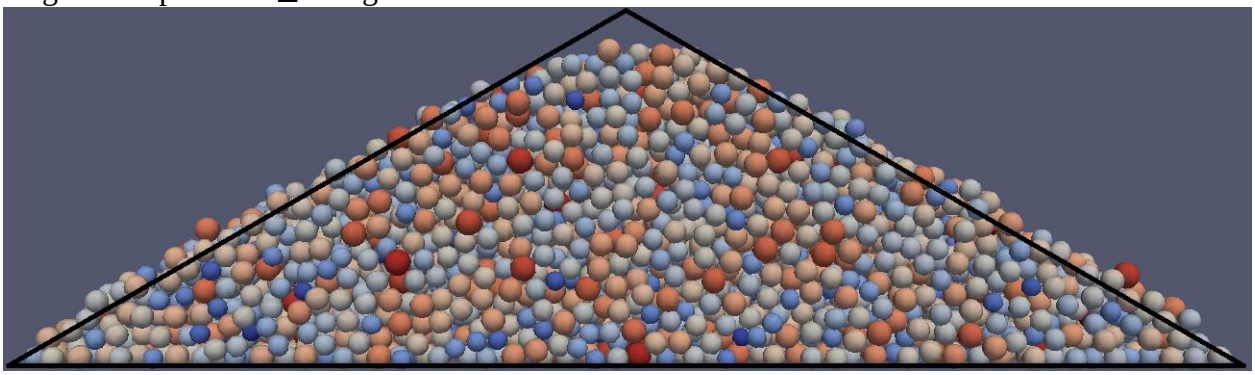
Figure 4.9: Mohr Coulomb Failure envelope for the consolidated sheared cohesive material.

Contact cohesion $k_{adh} = 5.35X10^6 N/m$

Rolling Friction Coefficient, $\mu_r = 0.1$
Angle of Repose: 26 $\pm$ 3 degrees



Rolling Friction Coefficient, $\mu_r = 0.2$
Angle of Repose: 32 $\pm$ 3 degrees



Rolling Friction Coefficient, $\mu_r = 0.4$
Angle of Repose: 39 $\pm$ 3 degrees

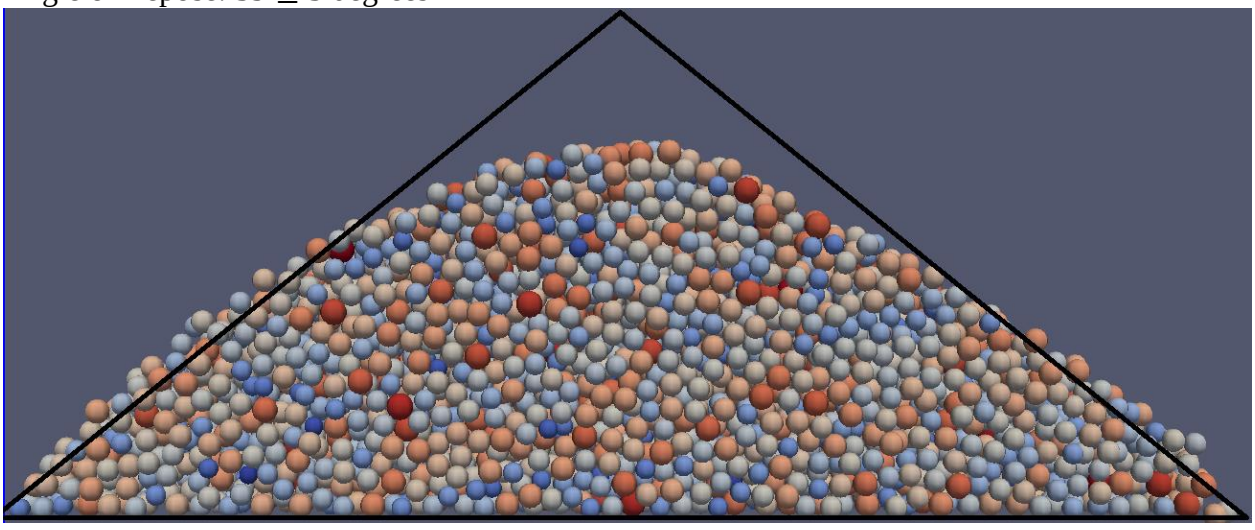


Figure 4.10: Sandpile formation for various rolling friction coefficients.

Table 4.3: Sandpile formation angles measured for dry and cohesive material.

Rolling Friction, μ_r	Dry material angle, ψ (deg)	$k_{adh} = 1.07 \times 10^7 \frac{N}{m}$ material angle, ψ (deg)
0.1	24± 3	26± 3
0.2	29± 3	30± 3
0.4	36± 3	39± 3
0.6	37± 3	42± 3

4.4 Remarks on Application

In summary, the implemented model has three stiffness values denoted as k-parameters that describe the following physical effects:

1. compacting (plastic-like deformation)

2. elastic unloading and re-loading of pre-compacted material

3. adhesive tensile strength of material modeled with cohesion-like behavior

The model involves a non-linear contact stiffness via a choice of index parameter n [47]. Tests were performed for validation by studying the loading system of an impacting particle. A ring cell shear test was performed to ensure shearing failure. The angle of repose test was performed to observe the dynamics of unconsolidated material flow. Each test simulated the results or effective mechanical behavior expected and solidified the usefulness of the MEPA cohesive model to examine flow behavior by simulating the stress history dependent strength and the macroscopic cohesive behavior of bulk solids. Below is a list of the model validations performed to ensure a proper working numerical model:

- Determined the material's effective cohesion using a ring cell shear test with known material properties, and illustrated the material failure modes.

- Determined the angle of repose of the material using loose pile formations, ensuring the variation of the angle with varying material conditions.

The following chapter describes the tests and results of the MEPA cohesive model tested compared with physical material flow property test. Here, a copper ore flow property test performed by Jenike and Johanson Incorporated is used as the test material to evaluate the capability of the cohesive contact model with real physical material properties [37].

CHAPTER 5
DEM SIMULATIONS OF THE MECHANICAL BEHAVIOR OF THE MATERIAL COPPER ORE

This chapter presents the DEM predictions using the MEPA cohesive model for copper ore and compares it to the experimental data of flow property test Report 11040-1 [37].

5.1 Experimental Behavior of Cohesive Granular Materials

The results of a flow property test for copper ore performed by Jenike & Johanson provided by OCC [31] was used as the mechanical testing material data in this study. This data set is used to evaluate the capability of the MEPA cohesive model to simulate the mechanical test data. The following sections describe the copper ore material tested, the mechanical tests and numerical DEM simulations performed, and their procedures.

5.1.1 Copper Ore Testing Material

The primary copper ore material tested was retrieved from the Morenci, Arizona mine. The material was prepared in the following manner:

1. Primary crushing was performed by a gyratory crusher.

2. Secondary crushing was performed by a cone crusher.

3. Tertiary crushing was performed by a hydraulic roll crusher.

Several samples were developed and are presented in the flow property test Report 11040-1 [37]. For this study, the results of Sample 1 are used at 5% and 8% moisture contents (mc). Moisture values are determined by drying small samples at 107 degrees Celsius for two hours in a forced convection oven. The loss in weight of the sample, divided by its original weight before drying, is referred to as the moisture [37]. The particle size of copper ore tested is

6.35mm in diameter with a bulk density of 1042.8-1752.4 $\frac{kg}{m^3}$ for 5% mc and 1350.4-1797.3 $\frac{kg}{m^3}$ for 8% mc. The weight density of an individual particle of copper ore sample 1 is 2481.3 $\frac{kg}{m^3}$. Though test Report 11040-1 [37] performs a series of tests for silo and hopper design, only the effect of moisture content is investigated here. An image of the physical copper ore material with 8% mc can be seen in Figure 5.1.

Figure 5.1: Copper ore material at 8% mc from [37].

5.2 The Testing Model Systems and Methods

The practical determination of the flow behavior of bulk solids is measured with shear testers to determine parameters such as the internal friction of the material. The physical material data used originates from the flow property tests of a translational shear tester. The Jenike shear tester designed for bulk solids and the Schulze ring shear tester will be described below in section 5.2.1.

5.2.1 Physical Translational Shear Tester

The physical test performed in Report 11040-1 [37] uses the Jenike Shear Tester, a translational shear tester. The shear cell is composed of a base located on the frame of the shearing machine. A ring rests on top of the base with a cover or lid. The surface conditions of the bottom of the cover and the inside of the base are rough to increase adhesion of the tested solid. The material is loaded into the base and ring and then covered. A normal force, F_N is applied centrally on the cover and held. The upper part of the shear cell is displaced horizontally against the fixed bottom base by a stem. The measured value is the shearing force exerted by the stem [50]. This procedure ensures a sufficiently uniform distribution of the shearing force across the cell as the material undergoes shear deformation. The normal stress, σ, and the shear stress, τ, acting in the horizontal plane between the top and bottom rings are determined by dividing the normal force, F_N, and shear force, F_S, by the cross-sectional area of the shear cell, A. The standard shear cell is 95.25 mm in diameter with a shearing rate of $4.487 \times 10^{-5} \frac{m}{sec}$ [38, 50]. A schematic of the physical tester with the dimensions defined in millimeters is shown in Figure 5.2.

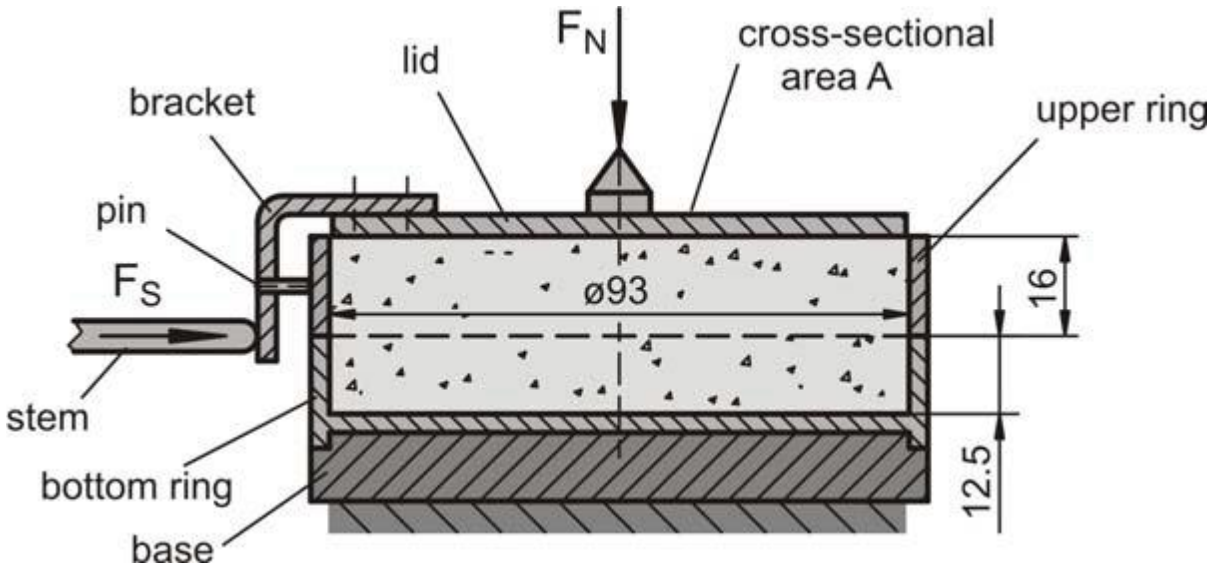


Figure 5.2: Jenike and Johanson direct shear test schematic from [50].

For the measurement of shearing points on a yield locus, the shear cell is initially filled with copper ore. The material is consolidated in what is called a "preshear" and then sheared to failure. The shearing failure provides a point measurement of the yield limit. For preshear, the copper ore is vertically loaded by a controlled normal stress, σ_{pre}, and then sheared. As preshear is initiated the shear stress increases with time. The curve of the shear stress over time becomes flatter settles to steady-state denoting constant stress. Once this state is achieved, the material is considered critically consolidated with respect to the normal stress, σ_{pre} and the shear deformation reversed until the shear stress returns to zero. After releasing the stress, the second step is to reduce the normal stress acting on the material to a value less than the normal stress, σ_{pre}. The material is then again sheared to failure. A second point of the yield limit can be obtained at the shear point of the consolidated material. Several shearing to failure tests are performed in order to measure the course of the yield locus using the same process. The points collected for the yield locus can be plotted through the measured shear points as seen in Figure 5.3.

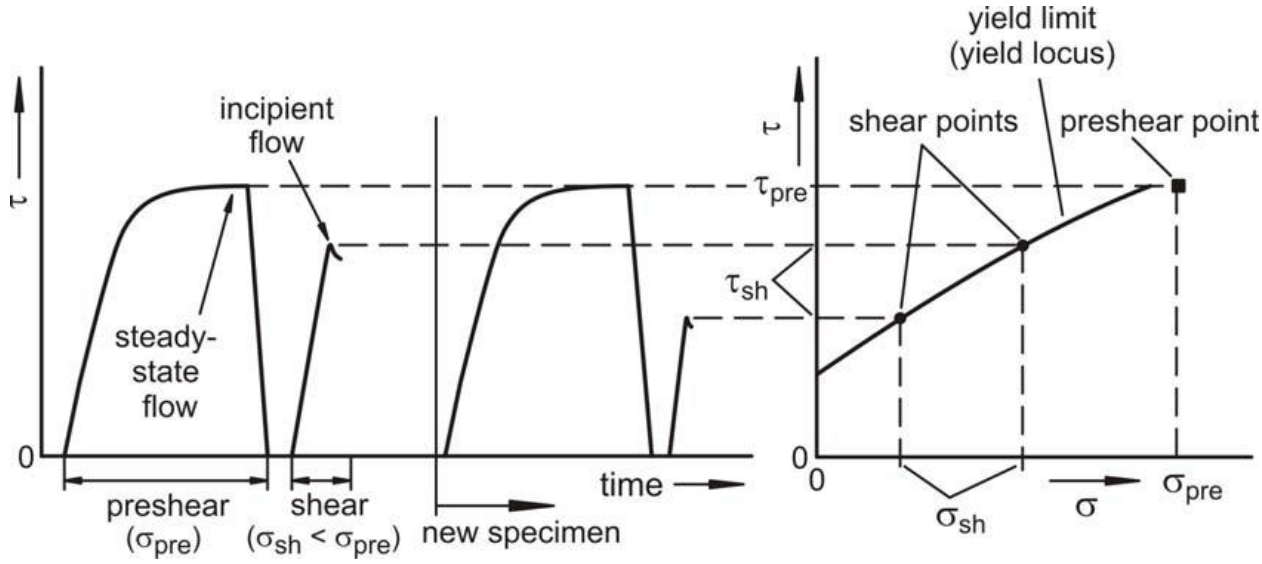


Figure 5.3: Conceptual plot of shear stress vs. time and yield locus from [50].

As a rule, tests are conducted on solids containing water below the point of saturation [38]. With these results, the flow-ability of a bulk solid material can be classified in a

quantitative manner into four stages:

- Very cohesive and non-flowing
- Cohesive
- Easy-flowing
- Free flowing

Although the translational shear tester is widely recognized, a disadvantage is the manual preconsolidation of each specimen [50]. This can be a source of measurement error. In addition, the limited shear displacement available does not allow materials requiring larger shear deformation to attain steady-state flow to be properly tested.

5.2.2 DEM Simulations of a Rotational Shear Tester

In this study experimental data sets obtained with the ring cell shear tester developed by Dietmar Schulze [51, 52] are used to validate and calibrate the DEM simulations. Figure 5.4 shows a schematic of a ring shear tester series RST-01 [36, 50–52]. The ring-shaped bottom ring of the shear cell contains the copper ore sample, while the lid is placed on top of the material and fixed at a crossbeam.

A normal force is exerted to the crossbeam in the rotational axis of the shear cell and transmitted through the lid to the material sample. The counterbalance force, F_A, acts in the center of the crossbeam and counteracts the gravity forces of the lid, the hanger, and the crossbeam [50]. To shear the sample, the lid and the bottom ring of the shear cell rotate relative to each other. This is achieved by rotating the bottom ring while the lid and the crossbeam are prevented from rotating by the connecting tie-rods. Each of the tie-rods are fixed at a load-beam from which the forces acting on cell can be measured. The test procedure is similar to the one performed for the Jenike shear tester. The yield locus can be plotted from the measured shear points as seen in Figure 5.5.

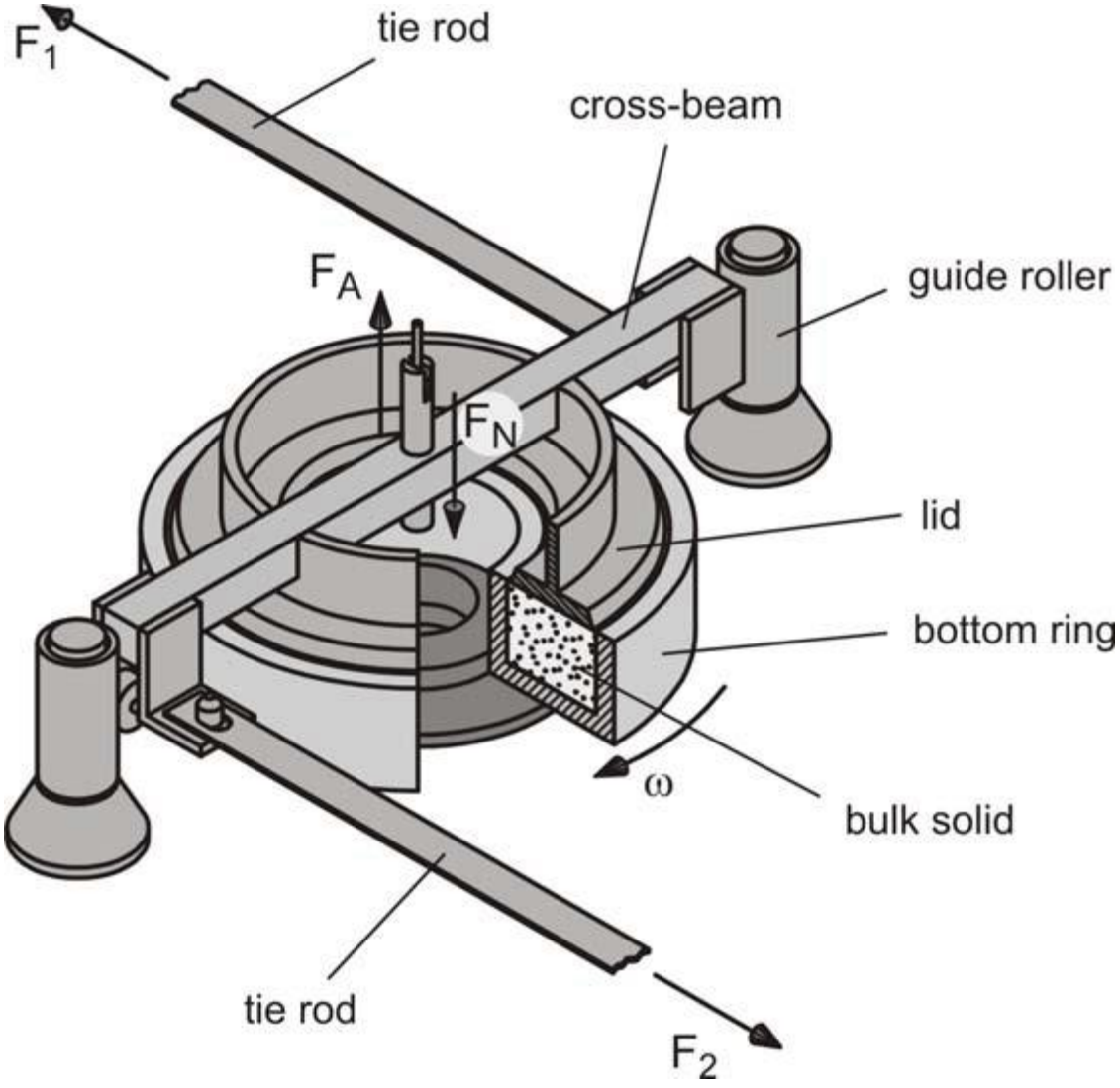


Figure 5.4: Shear Cell of a ring shear tester type RST-01 from [36, 50–52].

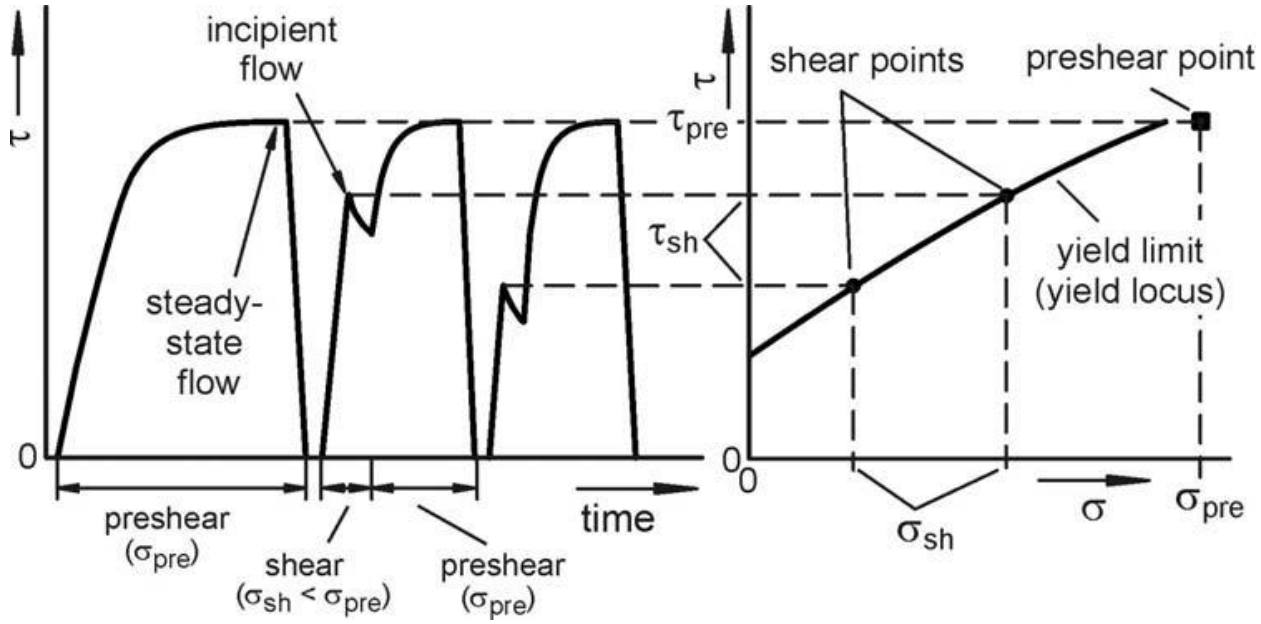


Figure 5.5: Conceptual shear test procedure of a ring shear tester from [50].

5.3 Numerical DEM Model Parameters

The virtual ring cell shear tester described in section 3.2.2 was filled with spherical particles with a truncated Gaussian distribution where the mean particle radius is 3.175 mm, and variation in particle radius is ±10% of the mean particle size. Note, this size variation prevents highly ordered crystal-like packing. A random insertion method was adopted to provide a random packing of the material. Cohesion between particles is accounted for with the k_{adh} value set to an assumed value in the filling process to allow for the development of a similar packing to the physical cohesive material. Static and rolling frictional values are used to account for the roughness of copper ore and the non-spherical nature of the material. The values of material parameters used in the simulations of 5% mc and 8% mc of copper ore are listed in Table 5.1. The material properties are representative of copper ore. The material reloading and unloading stiffness, k_2, is derived from the material properties such as Young's Modulus, Poisson's ratio and particle density. The loading stiffness, k_1, is equivalent to $\frac{1}{5}\,k_2$ as suggested in literature [44, 53]. The cohesive stiffness, k_{adh}, was determined iteratively

for the moisture content specified. Finally, the frictional values selected are representative of abrasive material as copper ore is highly abrasive. All the parameters were kept constant throughout the shearing process.

Table 5.1: Simulation parameters used in the ring shear testers for 5% and 8% mc of copper ore.

	5% mc Copper Ore	8% mc Copper Ore
Poisson's Ratio, ν	0.34	0.34
Young's Modulus, E (GPa)	119	119
Particle Radius, R (mm)	[2.85-3.49]	[2.85-3.49]
Particle Density, ρ $\left(\frac{kg}{m^3}\right)$	2481.3	2481.3
Loading Spring Stiffness, k_1 $\left(\frac{N}{m}\right)$	4.75×10^8	4.75×10^8
Unloading Spring Stiffness, k_2 $\left(\frac{N}{m}\right)$	2.37×10^9	2.37×10^9
Cohesive Stiffness, k_{adh} $\left(\frac{N}{m}\right)$	3.56×10^8	5.93×10^8
Particle Static Friction, μ_s	0.35	0.35
Particle Rolling Friction, μ_r	0.3	0.3
Wall Friction, μ_s	0.0	0.0
Base Friction, μ_s	0.7	0.7
Simulation Time step , $\triangle t$ (sec)	1.0×10^{-7}	1.0×10^{-7}

5.4 MEPA Cohesive Model Results for Copper Ore Sample 1 with 5% Moisture Content

The yield locus determined through physical experiments for copper ore with a moisture content of 5% is shown in Figure 5.6. The 6.35 mm diameter material is sheared with the Jenike Shear Tester as described in section 5.2.1 and serves as the reference case. The ring shearing tests were performed with 6.35 mm diameter material and a cohesive stiffness of $k_{adh} = 3.56 \times 10^8 \frac{N}{m}$ using the MEPA cohesive model to simulate the mechanical behavior. Table 5.2 shows the determined internal angle of friction of the material as a comparable measurement between tests. The numerical simulation results represent material consolidated at 54.8 kPa for the material conditions listed in Table 5.1. Figure 5.7 shows the simulated yield locus for the 5% moisture content condition obtained from the shearing cell in a comparable graph to the Jenike shear results. It can be concluded from Figure 5.7, for the given material conditions, that a greater consolidation stress is needed to reach the

shearing stresses of the physical system. The test results of the shear test procedure are also shown in Figure 5.8 with the Mohr Coulomb shear failure envelope after the application of a 5-point moving average. Much of the non-linearity of the data gathered is related to the rearrangement of the particles. The static and rolling frictional values together with the selected cohesive stiffness can account for the variation in the expected result.

The graphed Mohr Coulomb shear failure envelope more closely simulated the physical test data than the determined internal angle of friction from the yield limit graph. Upon closer visual examination of the shape of the physical ore material, it appears that further DEM simulation work warrants the analysis of non-spherical shaped particles. This point is discussed in Chapter 6 in section 6.2 describing Further Work. Additional work should be performed to validate the parameters seen in Table 5.1 for the consolidation compaction phase in order to gain more confidence in their use.

Table 5.2: Internal frictional angle, φ, for 5% mc of Copper Ore.

	Internal Frictional Angle, φ (degrees)
Jenike Shear (Physical Test)	47.3
Ring Shear (Mohr-DEM)	44.6
Ring Shear (Yield Limit-DEM)	31.5

5.5 MEPA Cohesive Model Results for Copper Ore Sample 1 with 8% Moisture Content

The yield locus determined through physical experiments for Copper Ore with a moisture content of 8% is shown in Figure 5.9 and serves as the reference case. The ring shearing tests were performed with 6.35 mm diameter material and a cohesive stiffness, $k_{adh} = 5.93 \times 10^8 \frac{N}{m}$ using the MEPA cohesive model to simulate the mechanical behavior. Table 5.3 shows the determined internal angle of friction of the material as a comparable measurement between tests. In agreement with the trend seen with 5% mc copper ore, the results show that the simulated material has a lower frictional angle than the physical material. The numerical DEM experiment results represents material consolidated at 38.7 kPa for the material con-

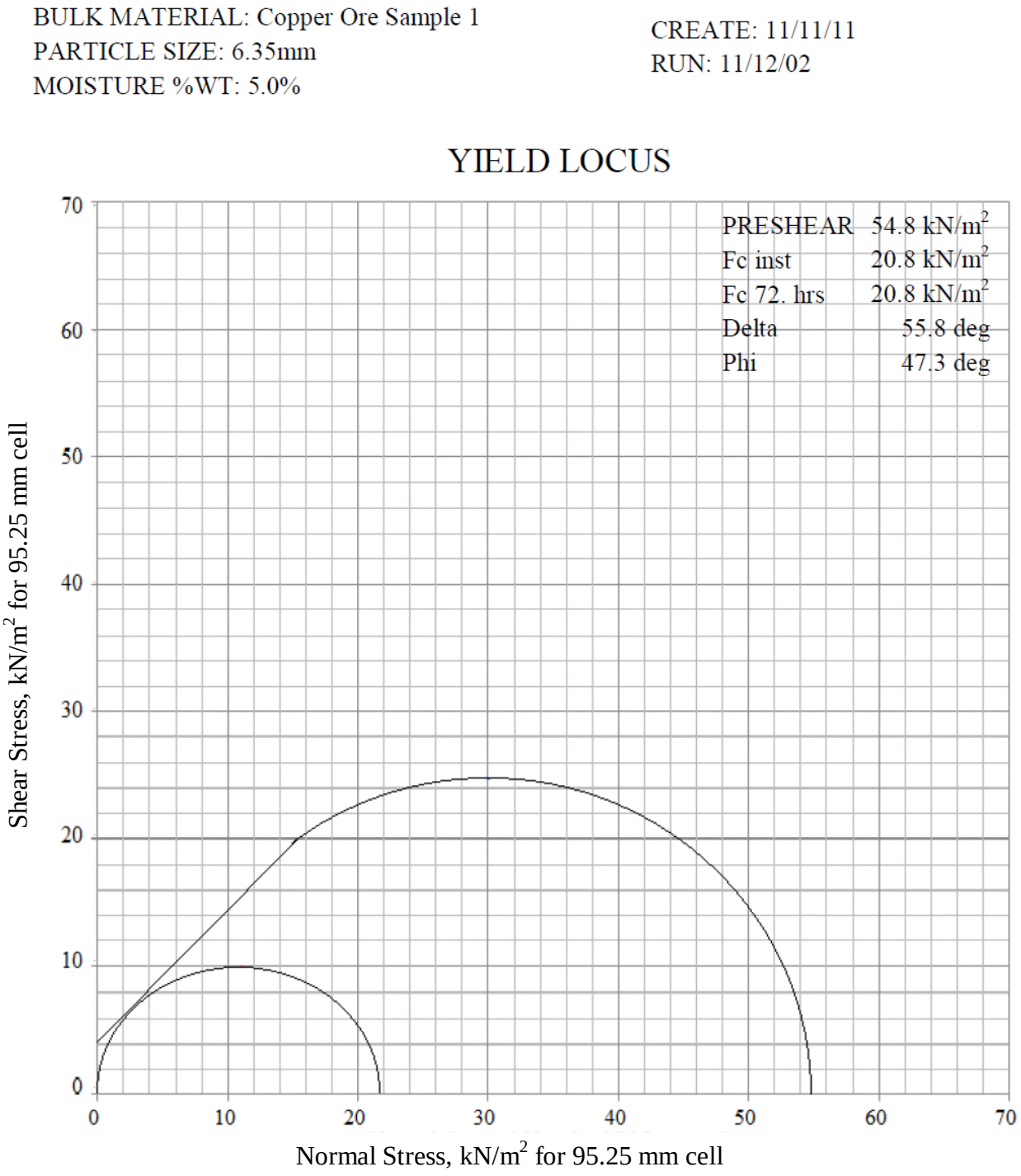


Figure 5.6: Physical testing results of the loading for Copper Ore Sample 1 at 5% mc [37].

BULK MATERIAL: Copper Ore Sample 1
PARTICLE SIZE: 6.35 mm

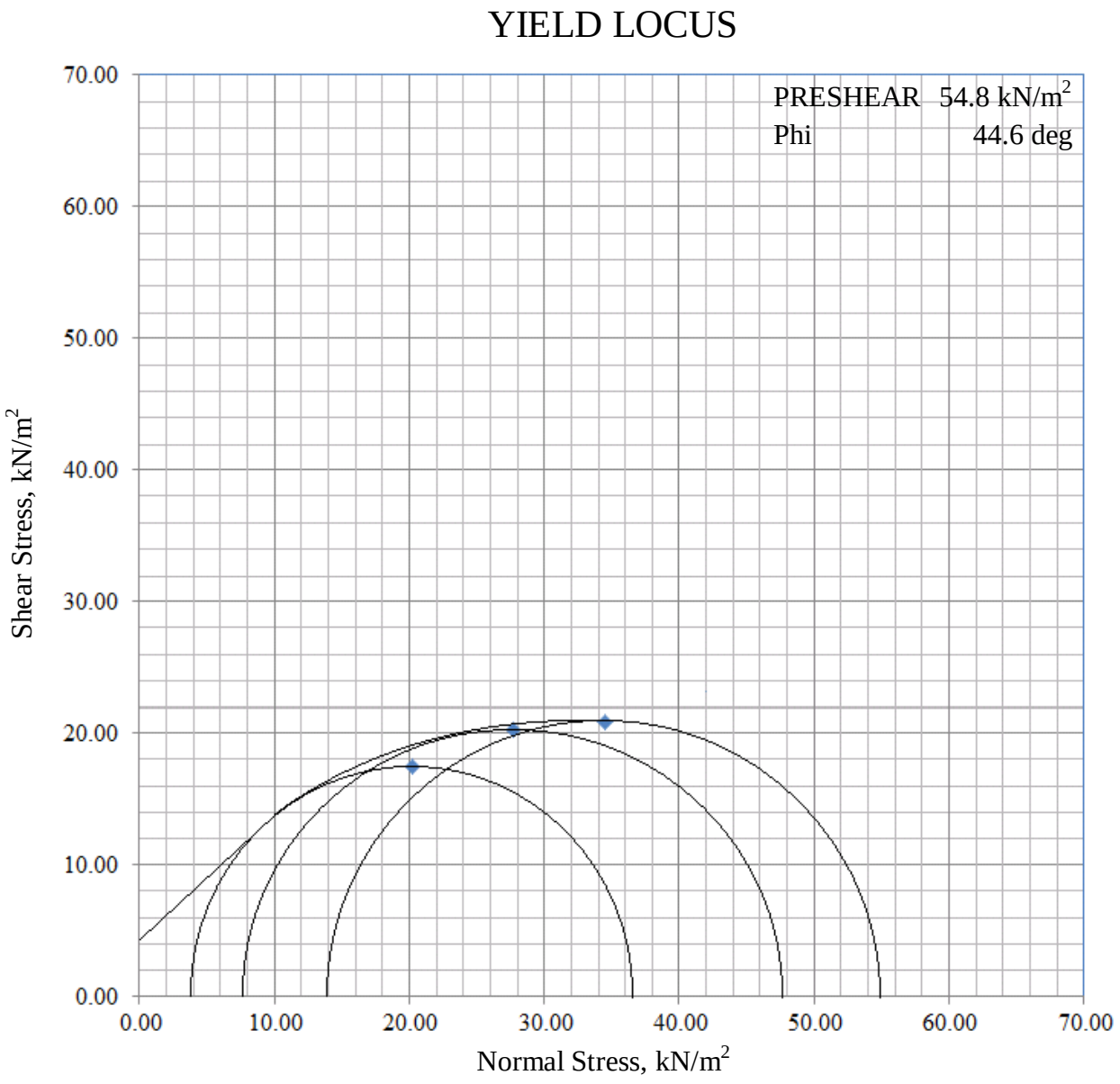


Figure 5.7: DEM testing results of the loading for Copper Ore Sample 1 at 5% mc with cohesive stiffness, $k_{adh} = 3.56 \times 10^8 \frac{N}{m}$.

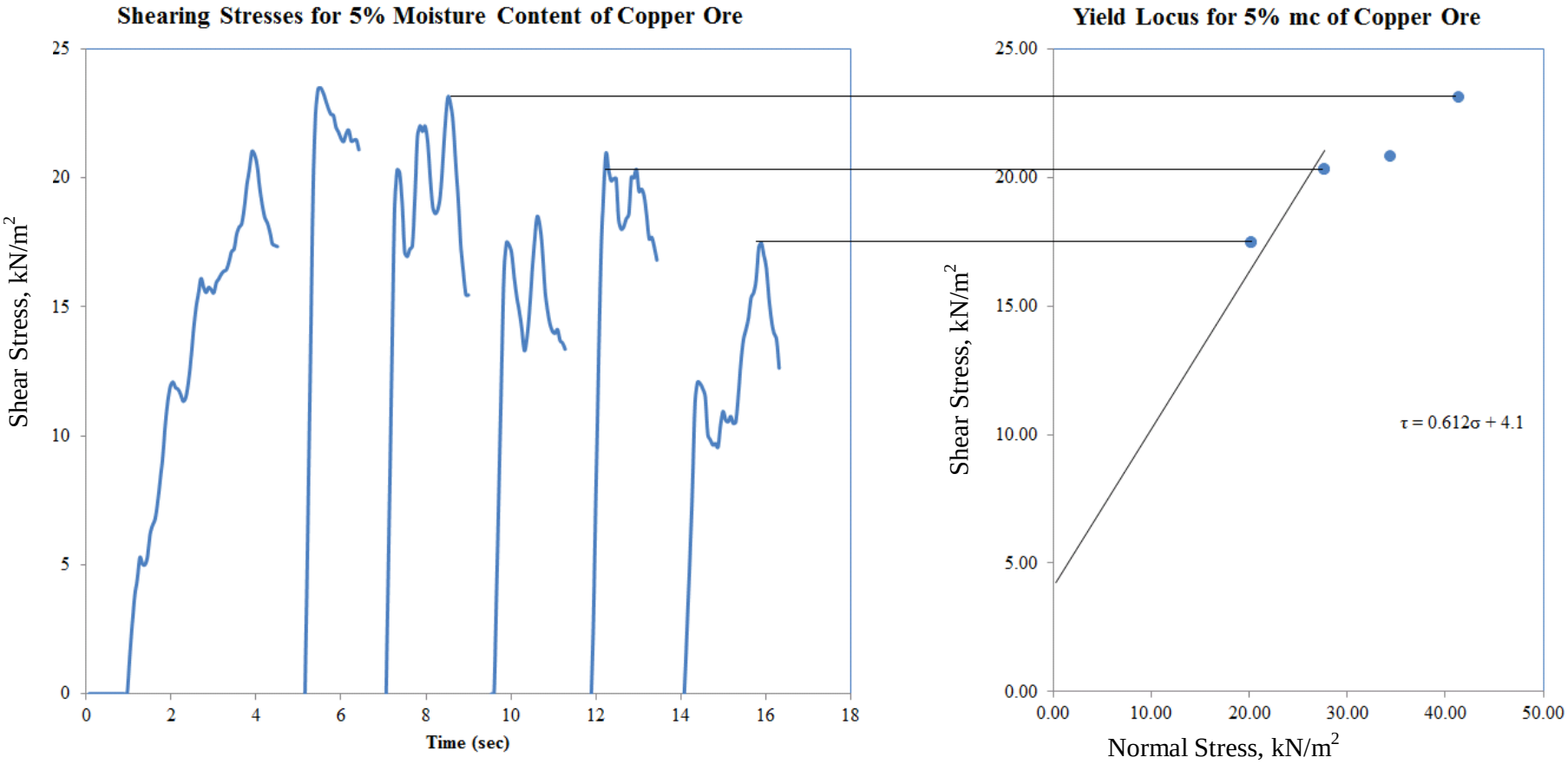


Figure 5.8: DEM determined yield locus of 5% mc Copper Ore Sample 1.

ditions listed in Table 5.1. Figure 5.10 shows the simulated yield locus for the 8% moisture content condition obtained from the shearing cell and most closely simulates the failure envelope of the physical testing data. It can be concluded for the given material conditions, a greater consolidation stress is needed to reach the shearing stresses of the physical system. The test results of the shear test procedure are also shown in Figure 5.11 with the Mohr Coulomb shear failure envelope with 5 point averaging to smooth out the rearrangement of the particles. From the numerical DEM data, it is difficult to determine the shear point in the data as particles rearrange. The variation in the shearing points is more evident in lower consolidation pressures and is a short coming of the servo control of the LIGGGHTs engine. The servo control detects the average force on the ring cell lid and adjusts the distance of the confinement height to keep the average consolidation pressure at the preset servo reference pressure. Improvements in the adaptive control system of the servo control would assist in overcoming the difficulties of simulating shearing material flows.

Table 5.3: Internal frictional angle, φ, measurements for 8% mc of Copper Ore.

	Internal Frictional Angle, φ (degrees)
Jenike Shear (Physical Test)	46.7
Ring Shear (Mohr-DEM)	44.8
Ring Shear (Yield Limit-DEM)	36.3

5.6 Discrete Element Method Application of Copper Ore using the MEPA Cohesive Contact Model

To observe the MEPA cohesive contact model in application, a transfer chute DEM simulation was performed with dry copper ore and 8% mc copper ore. The DEM material paramaters used for copper ore followed those described in Table 5.1 with the cohesive stiffness, k_{adh}, set to zero for the dry material simulation. The simulations were performed to observe the mechanical behavior of the simulated copper ore. Figure 5.12 shows dry copper ore on a transfer belt. It is observed the material is free flowing and displays no cohesive agglomerations. In Figure 5.13, 8% mc copper ore is simulated and compared with the

BULK MATERIAL: Copper Ore Sample 1
PARTICLE SIZE: 6.35 mm
MOISTURE % WT: 8.0%

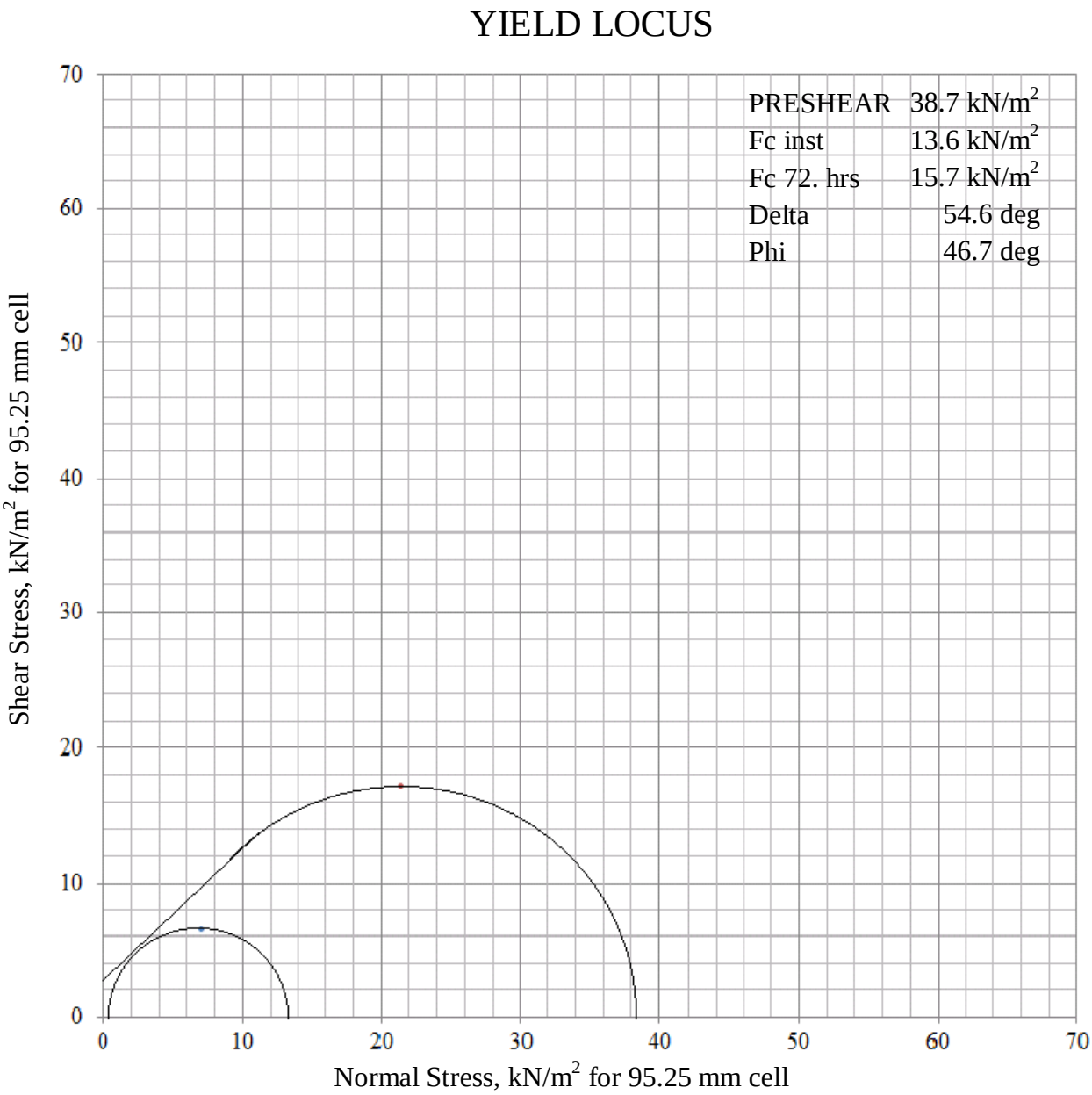


Figure 5.9: Physical testing results of the loading for Copper Ore Sample 1 at 8% mc [37].

BULK MATERIAL: Copper Ore Sample 1
PARTICLE SIZE: 6.35 mm

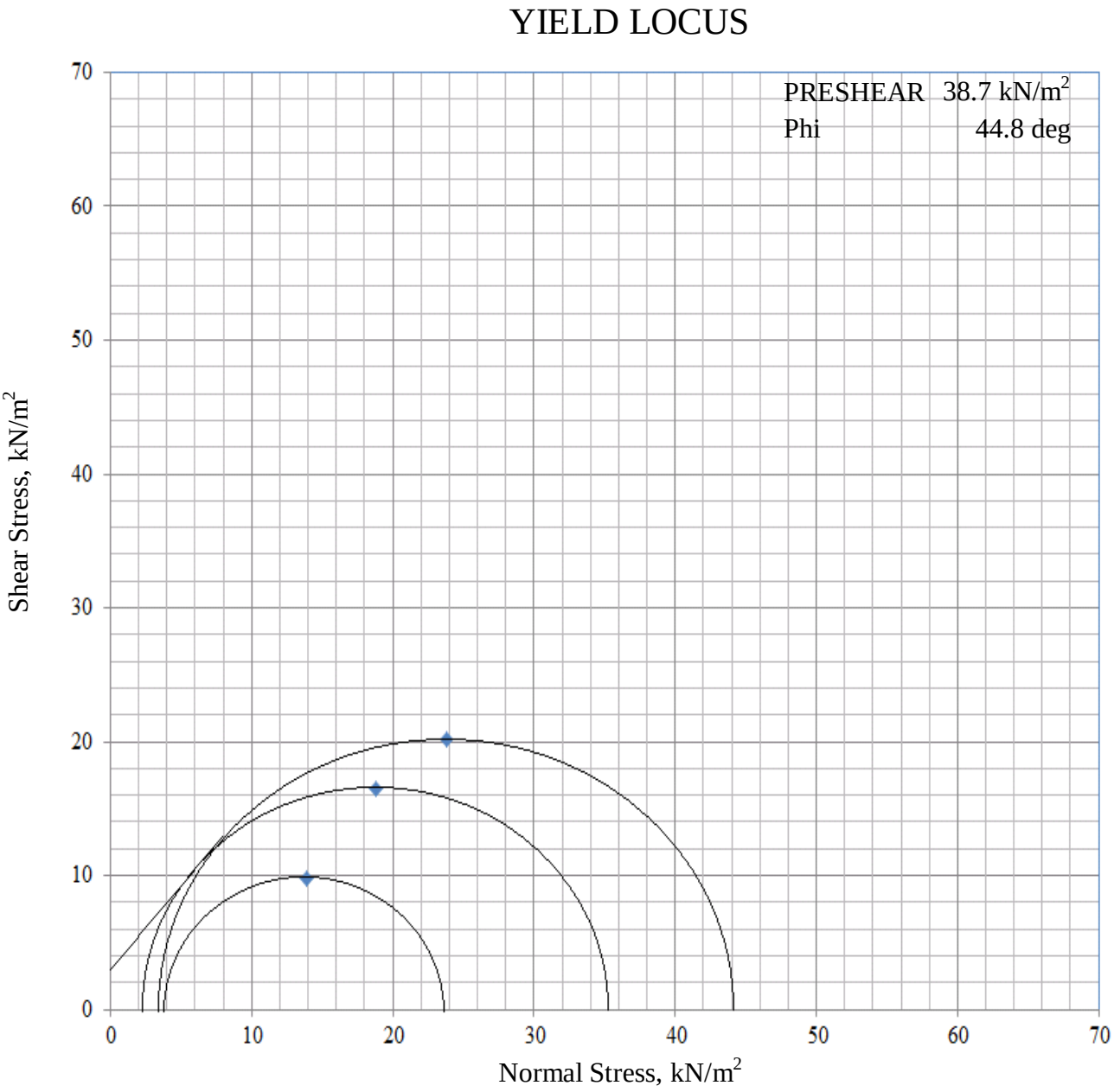


Figure 5.10: DEM testing results of the loading for Copper Ore Sample 1 at 8% mc with cohesive stiffness, $k_{adh} = 5.93 \times 10^8 \frac{N}{m}$

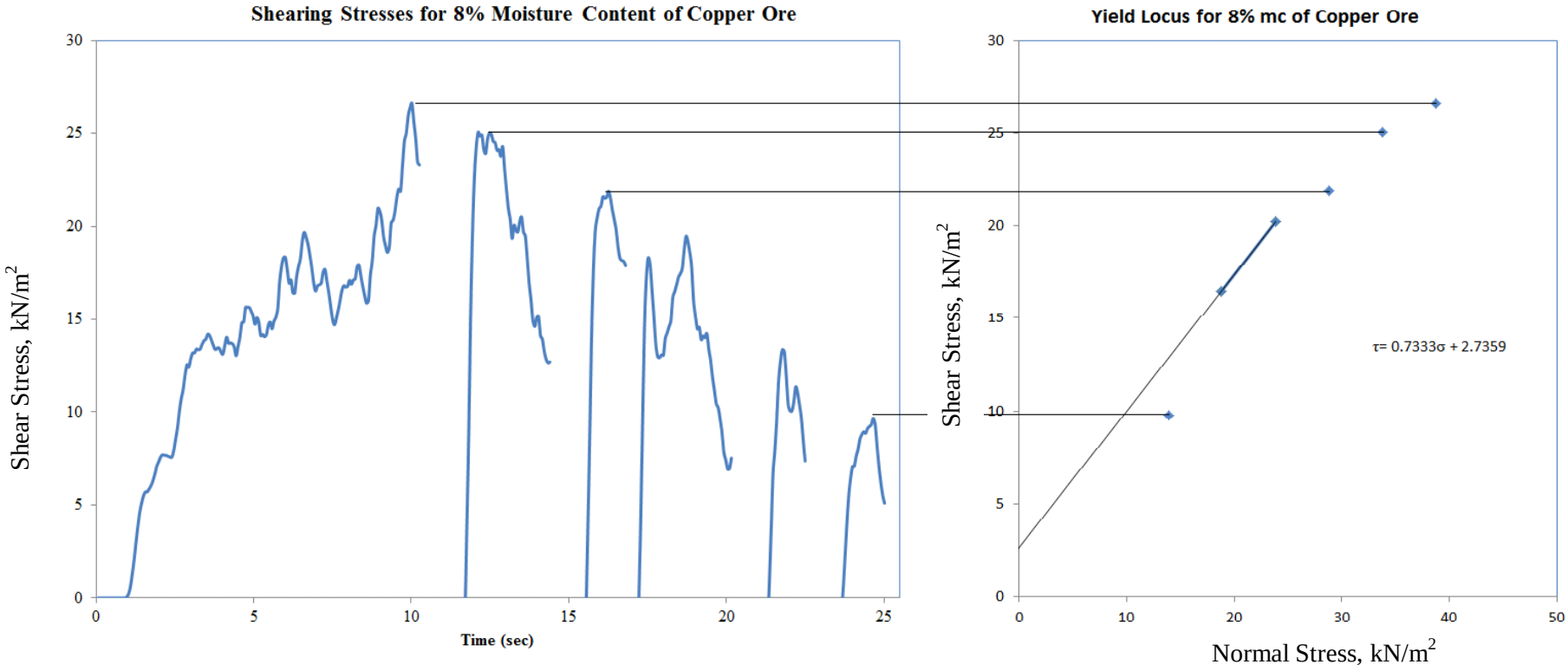


Figure 5.11: Numerically determined yield locus of 8% mc Copper Ore Sample 1.

physical material. In Figure 5.13b, the DEM simulated copper ore displays the mechanical behavior of cohesive materials. Clumps or agglomerations of particles ranging in size from 2.85 mm - 3.49 mm in radius form using the MEPA cohesive contact model and the material parameters described in Table 5.1. The effective cohesive behavior observed illustrates the capabilities of the MEPA cohesive model to simulate material strength described by physical tests. A more accurate representation is expected with further refinement of the material parameters and DEM particles shapes. These topics are further discussed in section 6.2.

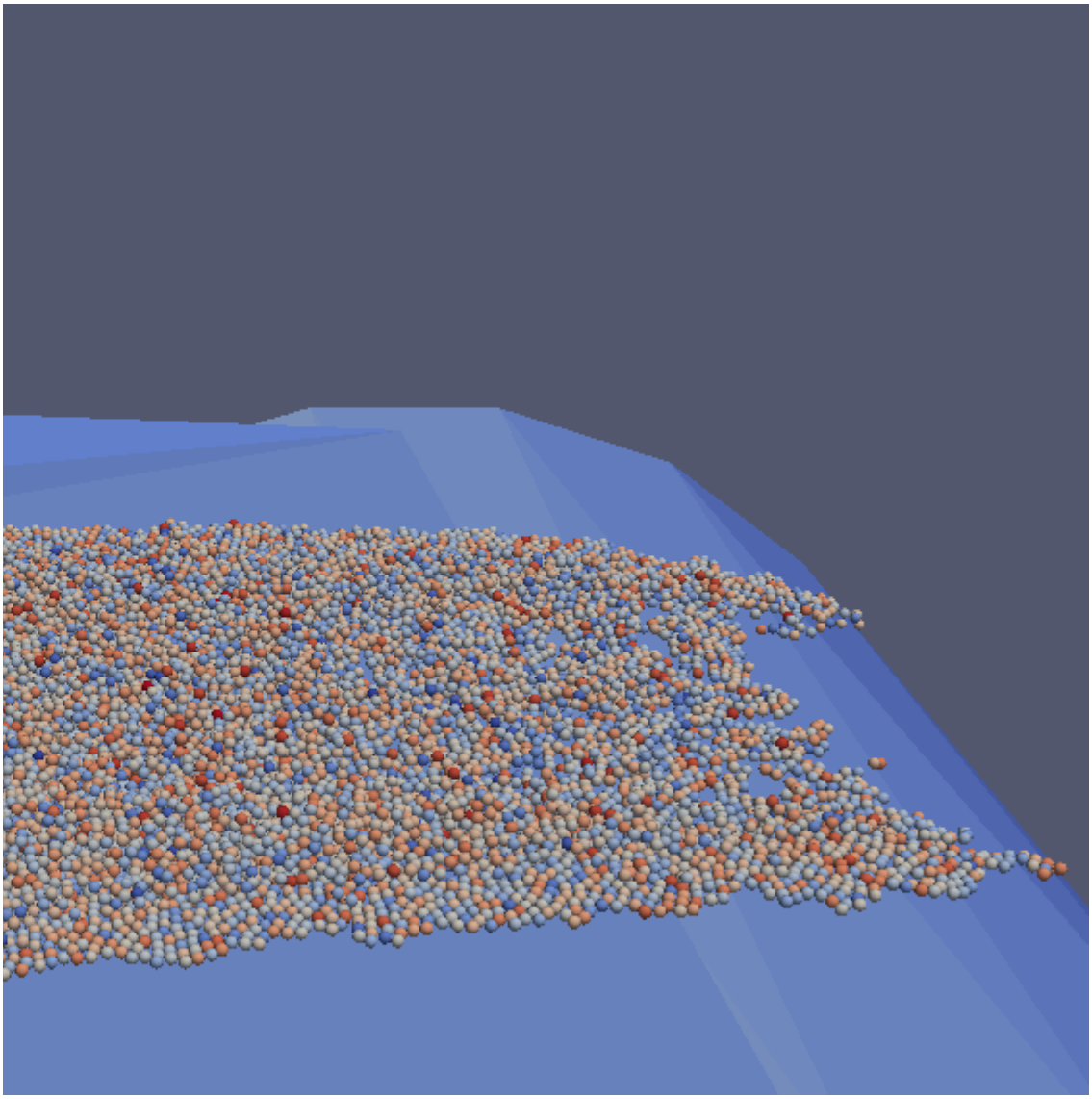
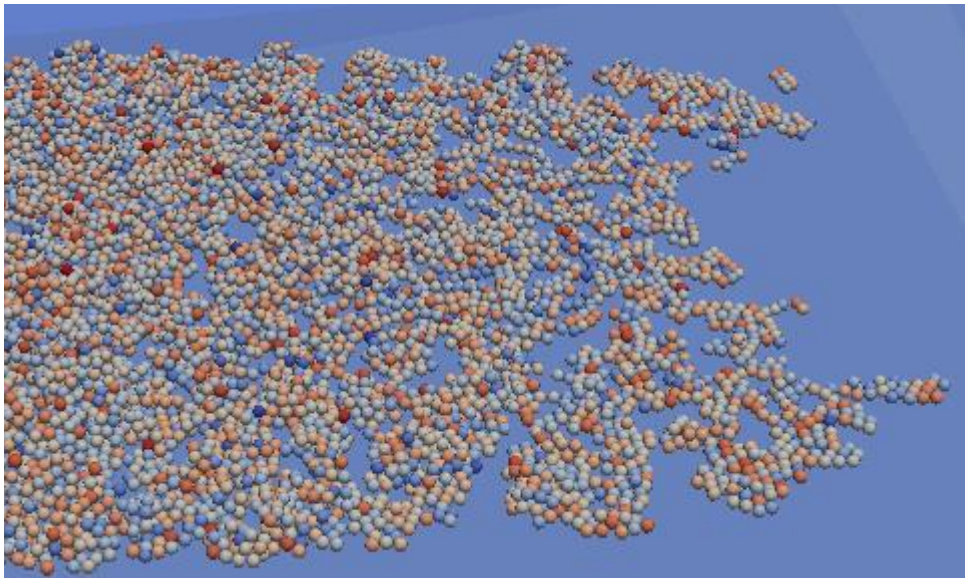


Figure 5.12: Virtual DEM material representation of dry copper ore on conveying belt.

(a) Physical copper ore material at 8% mc from [37].



(b) Virtual DEM copper ore material at 8% mc.

Figure 5.13: DEM simulated material representation of the 8% mc copper ore material.

CHAPTER 6
CONCLUSIONS AND FURTHER WORK

This final chapter summarized the research performed over the course of the dissertation. It reiterates the key concepts and provides context for the use of the MEPA cohesive model. This chapter also suggests some avenues of further work in moving forward with this research and how the MEPA cohesive model can be improved for greater reliability and ease of use.

6.1 Conclusions

The work contained herein can be broken down into six substantial sections: studies of the micro-mechanically based cohesive models, development of the MEPA cohesive model methodology, development of the coded algorithm, the impletation, modeling of material behavior and it's application to simulate large cohesive granular systems. The contributions of each of these concepts culminate to provide an improved understanding of cohesive flow behavior for the geomechanics and bulk material handling communities. The contributions of this study with respect to the proposed scope of work is as follows:

1. *Study of the micro-mechanically based cohesive models used to simulate cohesive material failure in bulk solids.* By investigating the capillary force and JKR models, it was determined that geometric scaling was not modeling the physically observed cohesive flow behavior without additional parametric tuning. The implementation of geometric scaling within the DEM algorithm was accomplished via the creation of a typical DEM particle with an agglomeration of finer particles. The challenging problem was tuning the scaled parameters in an attempt to obtain the desired observed mechanical behavior. Furthermore, simulations of material shearing tests were unable to provide a direct correlation to the effective cohesion of the system.

2. *Development of a macro-mechanical cohesive contact model.* The development of the methodology for this research was motivated by the inability of micro-mechanically based models to accurately simulate cohesive forces in bulk solid materials. The discrete element methodology developed as the MEPA cohesive model uses a maximum force-based failure approach. The model is a three-branch bounded system with stiffness values denoted by k-parameters that describe the following physical effects: compaction (plastic-like deformation), elastic unloading and re-loading of pre-compacted material and adhesive tensile strength of material modeled with cohesion-like behavior. The model is capable of simulating physical observed cohesive behavior for the analysis of cohesive granular bulk solid materials.

3. *Development of the iterative MEPA cohesive contact algorithm.* The development of the MEPA cohesive contact algorithm stepped through the loading, unloading, and reloading interations between particles. The logic developed and programmed illustrates the history of plastic deformation between a pair of contacting particles bound by the laws of the MEPA model. The development of the discretized iterative MEPA model was performed within the LIGGGHTS granular package and uses non-uniformly sized spherical particles.

4. *Implementation and validation of the MEPA cohesive contact model.* The implementation of the MEPA cohesive model performs as follows: as two particles are pressed together the particle contact undergoes elastic and plastic deformations and as they continue to be pressed together the pull-off force increases with the increase of the plastic contact area. To ensure the contacts followed the loading stages, a particle drop test was performed. The contact law was explored for both the linear Hookean-like and non-linear Hertzian-like loading/unloading conditions. The validation process concluded when the resulting loading stages traced the contact system properly as the methodology proposed.

5. *Modeling material failure via the simulation of physical flow property tests.* Cohesive material shearing failure is a challenge to simulate as these systems depend on their consolidation pressure history. To further test the MEPA cohesive model, the effect of moisture content within two copper ore samples was investigated. For this study, the material's yielding limit and internal angle of friction were simulated with some success. In application, the MEPA succeeded in illustrating material shearing failure with cohesion by simulating the mechanical behavior of physical data as seen in Figure 5.7 and Figure 5.10. These graphs more closely modeled the observed mechanical behavior of the copper ore material though it is noted that the shearing stresses and apparent internal angle of friction are lower than expected. The DEM simulation results would see improvement with a refinement on the material's parameters such as the selected static and rolling frictional values.

6. *Application of the MEPA cohesive contact model in bulk transfer.* The mechanical behavior of copper ore in a transfer system was simulated with the MEPA cohesive contact model. For this study, the parameters derived from the modeling of material failure for 8% mc copper ore were used. The simulation properly displayed the cohesive agglomerations observed in a sample of the physical copper ore material.

An advanced DEM based macro scale cohesive contact model has been developed for the analysis of cohesive granular bulk solid materials. The discrete element method approach to the MEPA cohesive contact model methodology has been successfully implemented and applied to the simulation of copper ore at two different cohesive levels. Initial studies of the available micro cohesive contact models in comparison to the qualitave simulation results obtained from the three-dimensional parallel implementation of the MEPA cohesive contant model illustrates the potential of this new methodology to accurately simulate mechanical behavior in granular bulk solids.

6.2 Future Work

The MEPA cohesive model developed and implemented for the analysis of bulk solid materials reproduces realistic looking macroscopic cohesive behavior and stress history dependent strength for simulating cohesive flow behaviors within bulk solids. This methodology establishes a robust approach to simulating cohesive mechanical flow behaviors. However, the current MEPA model can be improved in a number of areas. Some of these areas are discussed in the section below.

6.2.1 Refinement of the Determination of the MEPA Material Parameters

For all the material parameters, additional validation is required for the consolidation compaction phase in order to gain more confidence in their use. It is suggested a genetic algorithm with a similar approach to the one proposed by Garvey (2013) [54] be implemented. This would refine the material response for the model by determining the appropriate material parameters. Application of the genetic algorithm would determine a set of guidelines on the relationship between the input parameters and the macroscopic properties of the bulk material flow [54]. The structure would perform a hybrid crossover as described in the process below:

1. Generate an initial population. The number of sets of random material properties would be determined by computational capacity and availability.

2. Perform a shearing test on each population specimen.

3. Compute the internal angle of friction and macroscopic cohesion for each specimen.

4. Assess the fitness of the population to a measure of error of the macroscopic properties.

5. Identify the elite parameter values that more closely provide the desired macroscopic properties.

6. Perform crossover between the pair of elite parameter values to generate a new set of inputs.

7. Iterate the process starting from step 2 until a termination criterion has been reach.

Once the parametric refinement is complete, the final set of flow properties would be performed to simulate the mechanical behavior of the physical system.

6.2.2 A Detailed Material Parametric Study.

A detailed material parameter sensitivity study should be performed to see the effects of the different parameters in the MEPA model. A sensitivity study can test the robustness of the MEPA cohesive contact results and provide an increased understanding of the relationships between the input parameters such as virgin loading, k_1, sliding friction, μ_s, and rolling friction coefficient, μ_r, and the simulated material strength, stress history dependence and overall mechanical behavior. It would guide and focus the material calibration stage by identifying non-sensitive parameters and optimizing sensitive parameters in the MEPA model.

6.2.3 Smooth Transition of the Loading Branches of the MEPA Model.

Modify the MEPA model so that as the material becomes fully compacted the tangent moduli of the k_1-branch and the bounding k_2-branch is continuous. This should lead to a more realistic modeling of initial loading of a material.

6.2.4 Further Validation and Verification Simulations of Material Testing for Compressive and Unconsolidated Flows

A wider range of simulated tests is also suggested. Flow property tests, such as the one used in this dissertation, also perform compressibility (bulk density as a function of consolidating pressure), wall friction mass-flow angles, and permeability tests. These can be performed as an added validation to the material flow behavior.The physical tests performed

on the copper ore material to determine the mechanical behavior due to cohesive strength are as follows:

1. The Jenike shearing test to determining shearing failure.

2. Dry material sieving method for particle sizing.

3. Wall friction angles for mass flow calculations.

4. Permeability for the determination of critical steady-state flow rates.

5. Bulk density changes as a function of consolidation pressure.

In this study, only the Jenike shearing test data was used as a comparison to the DEM simulation results of the mechanical behavior of copper ore. Further DEM simulations should be performed for a more complete study of the mechanical behavior of the material. The sieving method and the bulk density tests would be a valued addition to the DEM validation and verification simulations of the material's mechanical behavior. The dry material sieving method would provide a particle size distribution after which the truncated Gaussian distribution would be modeled to prevent high order packing structures. The changes in bulk density are of interest as it relates to material handling and packing and affects the flow or no-flow conditions of discharging silos. A more complete set of test results would be significant in improving the simulation of cohesive flow.

The analysis of non-spherical shaped particles also warrants further work. Non-spherical particles carry a greater ability to interlock than classical spherical particles in combination with frictional models to simulate material interlocking during consolidation and shear. Literature indicates that packing is affected by particle shape and it's use in DEM simulations leads to a more accurate determination of the internal angles of friction and angles of repose [55].

6.2.5 Simulation of Moisture Content Changes over Time.

The changes in moisture content in a system is of increasing importance. Through the handling life of a material, it can undergo the evaporation of free water or see precipitation both environmental and mechanically induced for dust control. This capability may be possible with the addition of a visco-elastic component of normal contact force which acts between DEM particles.

6.3 Concluding Comments

As a foundation for continued bulk flow analysis, this research produced the MEPA cohesive model. This research provides a solution to the issues posed by the inability of micro-mechanically based models to accurately simulate cohesive forces observed in bulk solid materials. The research framework was organized into the development, implementation and applicability of a discrete element methodology. The resulting MEPA cohesive model can thus be applied successfully to simulate the mechanical behavior described by the physical data.

REFERENCES CITED

[1] Antony S. J., W. Hoyle, and Y. Ding. *Granular materials: fundamentals and applications*. Royal Society of Chemistry, 2004.

[2] Brown R. L. and J. C. Richards. Principles of powder mechanics. 1970.

[3] Radjaï F., F. Dubois, et al. *Discrete-element modeling of granular materials*. Wiley, 2011.

[4] Johnson K.L., K. Kendall, and A. D. Roberts. Surface energy and the contact of elastic solids. *Proceedings of the Royal Society of London. A. mathematical and physical sciences*, 324(1558):301–313, 1971.

[5] Lian G., C. Thornton, and M. J. Adams. A theoretical study of the liquid bridge forces between two rigid spherical bodies. *Journal of Colloid and Interface Science*, 161(1): 138–147, 1993.

[6] Gröger T., U. Tüzün, and D. M. Heyes. Modelling and measuring of cohesion in wet granular materials. *Powder Technology*, 133(1):203–215, 2003.

[7] Thakur S., J. P. Morrissey, J. Sun, J. F. Chen, and J. Y. Ooi. Particle scale modelling of cohesive powders for bulk handling applications. In *EDEM Conference*, 2011.

[8] Savage S. B. The mechanics of rapid granular flows. *Advances in applied mechanics*, 24:289–366, 1984.

[9] da Cruz F., S. Emam, M. l. Prochnow, J. N. Roux, and F. Chevoir. Rheophysics of dense granular materials: Discrete simulation of plane shear flows. *Physical Review E*, 72(2):021309, 2005.

[10] Nedderman R. M. *Statics and kinematics of granular materials*. Cambridge University Press, 2005.

[11] Tardos G. I. A fluid mechanistic approach to slow, frictional flow of powders. *Powder Technology*, 92(1):61–74, 1997.

[12] Campbell C. S. Rapid granular flows. *Annual Review of Fluid Mechanics*, 22(1):57–90, 1990.

[13] Jenkins J. T. and S. B. Savage. A theory for the rapid flow of identical, smooth, nearly elastic, spherical particles. *Journal of Fluid Mechanics*, 130:187–202, 1983.

[14] Gordon G. D. R. P., Zhou and Q. C. Sun. Three-dimensional numerical study on flow regimes of dry granular flows by dem. *Powder Technology*, 239:115–127, 2013.

[15] Zhou Y. C., B. H. Xu, A. B. Yu, and P. Zulli. An experimental and numerical study of the angle of repose of coarse spheres. *Powder Technology*, 125(1):45 – 54, 2002.

[16] Kloss C. and C. Goniva. Liggghts–a new open source discrete element simulation software. In *Proc. of The 5th International Conference on Discrete Element Methods*, pages 25–26, 2010.

[17] Ketterhagen W. R., J. S. Curtis, C. R. Wassgren, and B. C. Hancock. Predicting the flow mode from hoppers using the discrete element method. *Powder technology*, 195(1): 1–10, 2009.

[18] Anand A., J. S. Curtis, C. R. Wassgren, B. C. Hancock, and W. R. Ketterhagen. Predicting discharge dynamics of wet cohesive particles from a rectangular hopper using the discrete element method (DEM). *Chemical Engineering Science*, 64(24):5268–5275, 2009.

[19] Galanov B. A. Models of adhesive contact between rough elastic solids. *International Journal of Mechanical Sciences*, 53(11):968–977, 2011.

[20] Ai J., J. F. Chen, M. J. Rotter, and J. Y. Ooi. Assessment of rolling resistance models in discrete element simulations. *Powder Technology*, 206(3):269–282, 2011.

[21] Bradley R. S. The cohesive force between solid surfaces and the surface energy of solids. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 13 (86):853–862, 1932.

[22] Johnson K. L. *Contact mechanics*. Cambridge university press, 1987.

[23] Muller V. M., V.S. Yushchenko, and B.V. Derjaguin. On the influence of molecular forces on the deformation of an elastic sphere and its sticking to a rigid plane. *Journal of Colloid and Interface Science*, 77(1):91–101, 1980.

[24] Maugis D. Adhesion of spheres: the JKR-DMT transition using a Dugdale model. *Journal of Colloid and Interface Science*, 150(1):243–269, 1992.

[25] Hotta K., K. Takeda, and K. Iinoya. The capillary binding force of a liquid bridge. *Powder Technology*, 10(4):231–242, 1974.

[26] Soulie F., F. Cherblanc, M. S. El Youssoufi, and C. Saix. Influence of liquid bridges on the mechanical behaviour of polydisperse granular materials. *International journal for numerical and analytical methods in geomechanics*, 30(3):213–228, 2006.

[27] Mikami T., H. Kamiya, and M. Horio. Numerical simulation of cohesive powder behavior in a fluidized bed. *Chemical Engineering Science*, 53(10):1927–1940, 1998.

[28] Gras J. P., J. Y. Delenne, F. Soulié, and M. S. El Youssoufi. Micromechanical analysis of water retention phenomenon. In *Powders and Grain 2009: Proceedings of the 6th International Conference on Micromechanics of Granular Media*, volume 1145, pages 923–926. AIP Publishing, 2009.

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

[30] Cambou B., M. l. Jean, F. Radjaï, et al. *Micromechanics of granular materials*. Wiley Online Library, 2009.

[31] Overland Conveyor Company Inc. Lakewood CO 80228. Bulk Material Handling Solutions. `http://www.overlandconveyor.com/`, 2013.

[32] Pierrat P. and H. S. Caram. Tensile strength of wet granula materials. *Powder Technology*, 91(2):83–93, 1997.

[33] Holloway M. D. and C. NWaoha. *Dictionary of Industrial Terms*. 2013.

[34] Kruse D. and R. Lemmon. Material flow simulation-using the discrete element method as an everyday design tool. *Bulk Solids Handling*, 25(6):358–367, 2005.

[35] Schulze D. *Powders and bulk solids: behavior, characterization, storage and flow*. Springer, 2007.

[36] Standard ASTM. D6773-02: Standard shear test method for bulk solids using the Schulze ring shear tester, ASTM international.

[37] Subramanian A. Report 9.3.2013 : Ring shear testing of dry particles using LIGGGHTS. Overland Conveyor Company Inc. Lakewood, CO 80228, 2013.

[38] Jenike A. W. Storage and flow of solids, bulletin no. 123. *Bulletin of the University of Utah*, 53(26), 1964.

[39] Tavarez F. A. and M. E. Plesha. Discrete element method for modeling penetration. In *ASME/JSME 2004 Pressure Vessels and Piping Conference*, pages 165–171. American Society of Mechanical Engineers, 2004.

[40] Thornton C. and S. J. Antony. Quasi-static shear deformation of a soft particle system. *Powder technology*, 109(1):179–191, 2000.

[41] D'Addetta G. A., F. Kun, and E. Ramm. On the application of a discrete model to the fracture process of cohesive granular materials. *Granular matter*, 4(2):77–90, 2002.

[42] Kruyt N. P. and L. Rothenburg. Statistics of the elastic behaviour of granular materials. *International Journal of Solids and Structures*, 38(28):4879–4899, 2001.

[43] Luding S., M. Lätzel, W. Volk, S. Diebels, and H. J. Herrmann. From discrete element simulations to a continuum model. *Computer methods in applied mechanics and engineering*, 191(1):21–28, 2001.

[44] Tomas J. Assessment of mechanical properties of cohesive particulate solids. Part 1: particle contact constitutive model. *Particulate science and technology*, 19(2):95–110, 2001.

[45] Luding S. Shear flow modeling of cohesive and frictional fine powder. *Powder Technology*, 158(1):45–50, 2005.

[46] Luding S., R. Tykhoniuk, and J. Tomas. Anisotropic material behavior in dense, cohesive-frictional powders. *Chemical engineering & technology*, 26(12):1229–1232, 2003.

[47] Luding S. and A. S. J. Suiker. Self-healing of damaged particulate materials through sintering. *Philosophical Magazine*, 88(28-29):3445–3457, 2008.

[48] Luding S. Cohesive, frictional powders: contact models for tension. *Granular matter*, 10(4):235–246, 2008.

[49] Luding S., K. Manetsberger, and J. Müllers. A discrete model for long time sintering. *Journal of the Mechanics and Physics of Solids*, 53(2):455 – 491, 2005.

[50] Schulze D. Flow properties of powders and bulk solids. *Braunschweig/Wolfenbu ttel, Germany: University of Applied Sciences*, 2006.

[51] Schulze D. Development and application of a novel ring shear tester. *Aufbereitungstechnik*, 35(10):524–535, 1994.

[52] Schulze D. A new ring shear tester for flowability and time consolidation measurements. In *Proc. 1st International Particle Technology Forum*, pages 11–16, 1994.

[53] Thakur S. C., J. P. Morrissey, J. Sun, J. F. Chen, and J. Y. Ooi. Micromechanical analysis of cohesive granular materials using the discrete element method with an adhesive elasto-plastic contact model. *Granular Matter*, 16(3):383–400, 2014.

[54] Garvey R. J. *A study of unstable rock failures using finite difference and discrete element methods*. Colorado School of Mines, 2013.

[55] Zou R. P. and A. B. Yu. Evaluation of the packing characteristics of mono-sized non-spherical particles. *Powder Technology*, 88(1):71 – 79, 1996.

APPENDIX A - SHEARING FAILURE ENVELOPES FOR VARYING ROLLING FRICTION CONDITIONS

The Mohr-Coulomb Shear Failure envelopes also demonstrate the expected effective cohesive behavior, such as increased effective cohesion and zero cohesion for the dry material condition. For the variations in the material's rolling friction, as the roughness increases, the shearing failure stress also increases as is expected with the variation of rolling friction.

A.1 Shearing Failure Envelopes for Rolling Friction Coefficient 0.1 and Cohesive Cohesive Stiffness $1.07 \times 10^7 \frac{N}{m}$

Figure A.1 with Cohesion Stiffness $k_{adh} = 1.07 \times 10^7 \frac{N}{m}$

A.2 Shearing Failure Envelopes for Rolling Friction Coefficient 0.2 and Cohesive Cohesive Stiffness $1.07 \times 10^7 \frac{N}{m}$

Figure A.2 with Cohesion Stiffness $k_{adh} = 1.07 \times 10^7 \frac{N}{m}$

A.3 Shearing Failure Envelopes for Rolling Friction Coefficient 0.4 and Cohesive Cohesive Stiffness $1.07 \times 10^7 \frac{N}{m}$

Figure A.3 with Cohesion Stiffness $k_{adh} = 1.07 \times 10^7 \frac{N}{m}$

A.4 Shearing Failure Envelopes for Rolling Friction Coefficient 0.6 and Cohesive Cohesive Stiffness $1.07 \times 10^7 \frac{N}{m}$

Figure A.4 with Cohesion Stiffness $k_{adh} = 1.07 \times 10^7 \frac{N}{m}$

A.5 Shearing Failure Envelopes for Rolling Friction Coefficient 0.1and Cohesive Cohesive Stiffness $5.35 \times 10^6 \frac{N}{m}$

Figure A.5 with Cohesion Stiffness $k_{adh} = 5.35 \times 10^6 \frac{N}{m}$

A.6 Shearing Failure Envelopes for Rolling Friction Coefficient 0.2 and Cohesive Cohesive Stiffness $5.35 \times 10^6 \frac{N}{m}$

Figure A.6 with Cohesion Stiffness $k_{adh} = 5.35 \times 10^6 \frac{N}{m}$

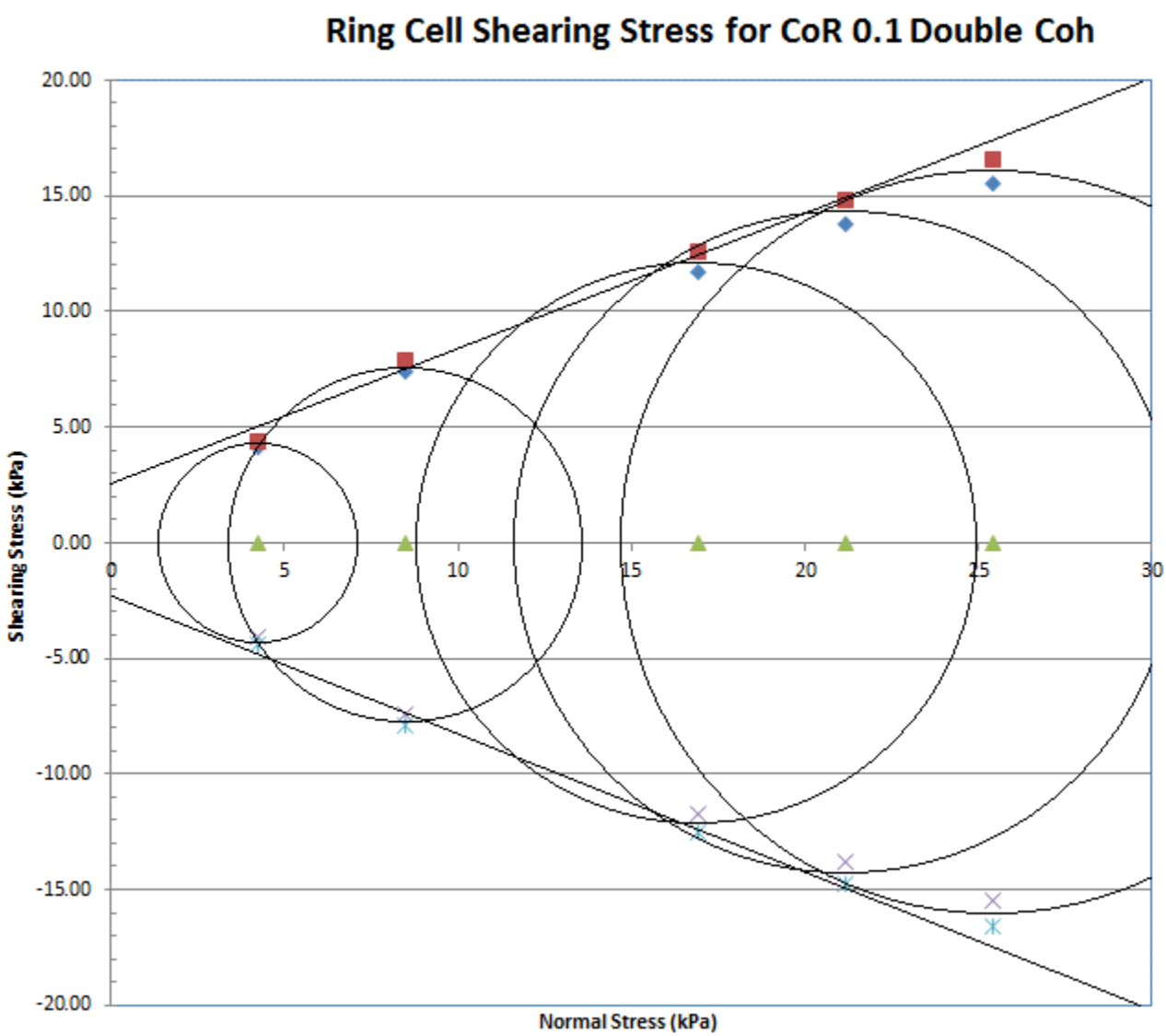


Figure A.1: Mohr Coulomb Failure envelope for the consolidated sheared high cohesive material for CoR 0.1.

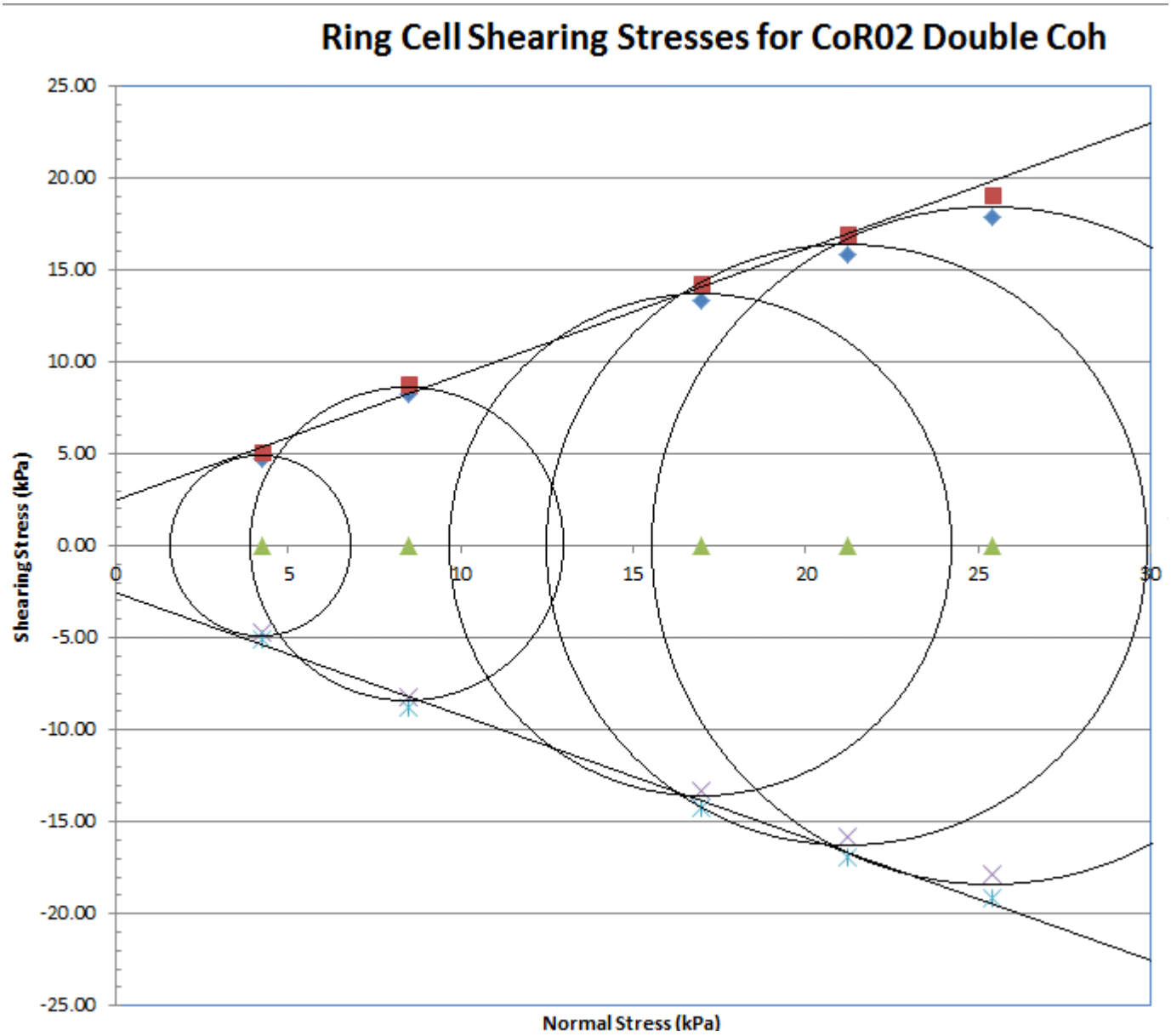


Figure A.2: Mohr Coulomb Failure envelope for the consolidated sheared high cohesive material for CoR 0.2.

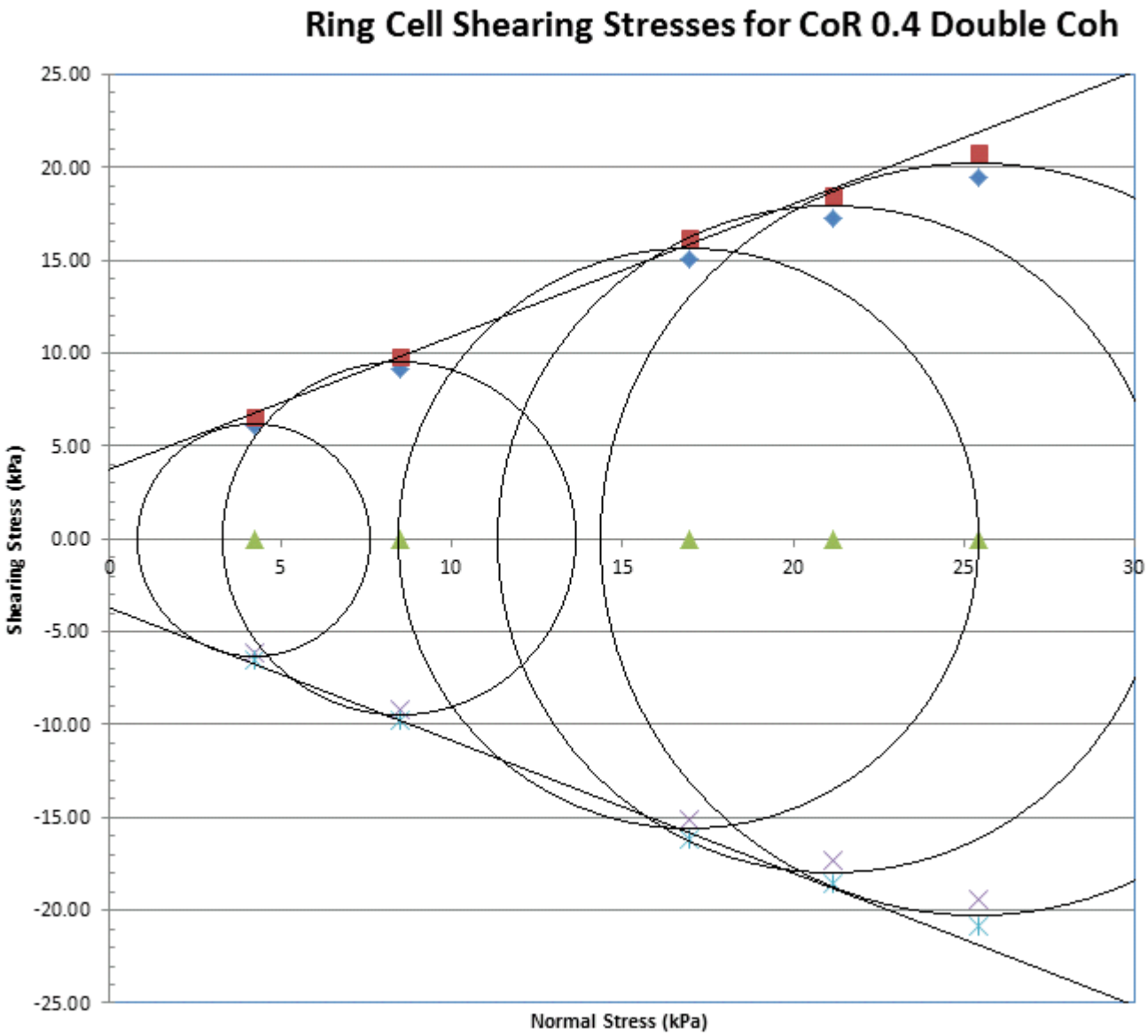


Figure A.3: Mohr Coulomb Failure envelope for the consolidated sheared high cohesive material for CoR 0.4

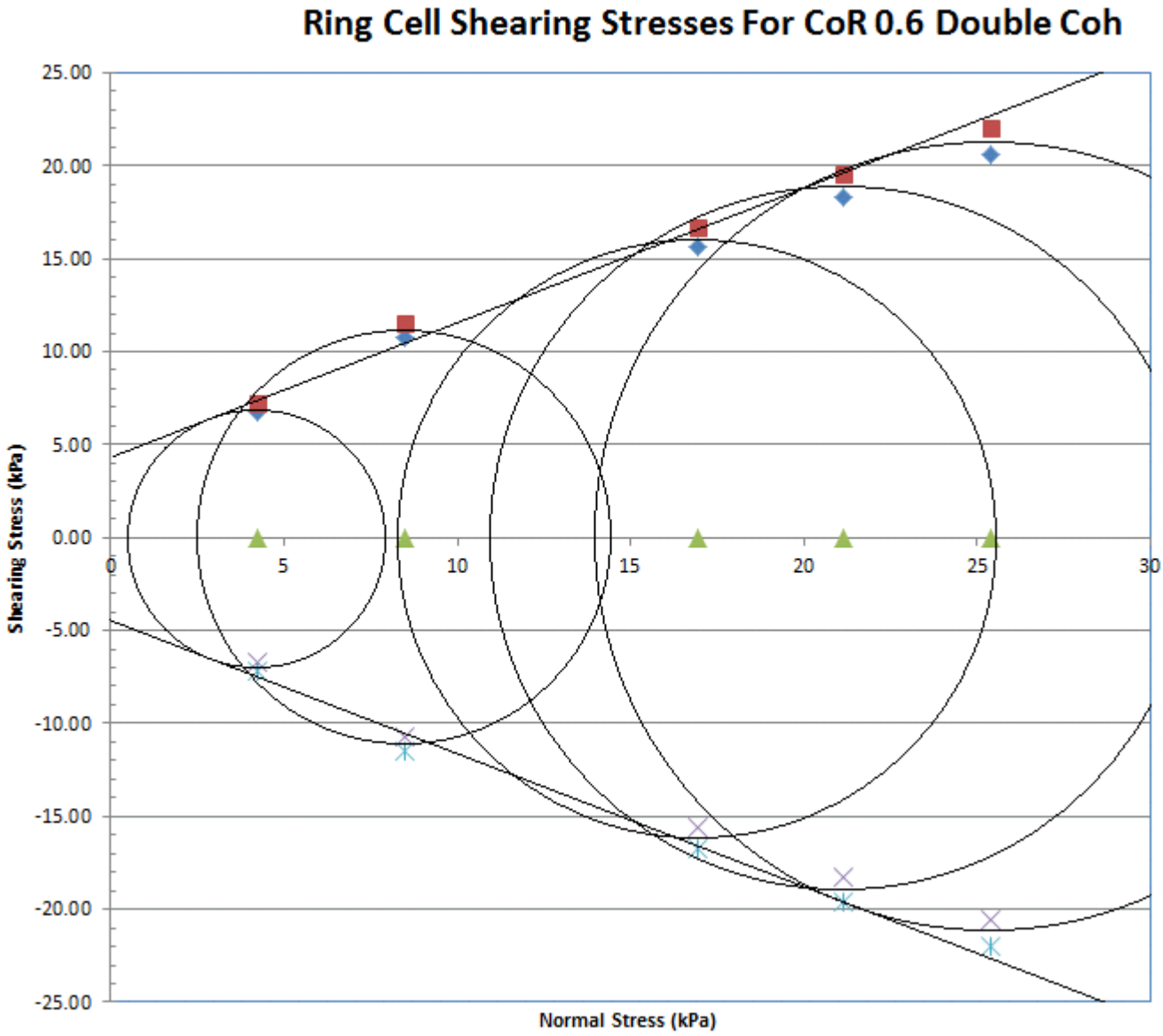


Figure A.4: Mohr Coulomb Failure envelope for the consolidated sheared high cohesive material for CoR 0.6.

A.7 Shearing Failure Envelopes for Rolling Friction Coefficient 0.4 and Cohesive Cohesive Stiffness $5.35 \times 10^6 \frac{N}{m}$

Figure A.7 with Cohesion Stiffness $k_{adh} = 5.35 \times 10^6 \frac{N}{m}$

A.8 Shearing Failure Envelopes for Rolling Friction Coefficient 0.6 and Cohesive Cohesive Stiffness $5.35 \times 10^6 \frac{N}{m}$

Figure A.8 with Cohesion Stiffness $k_{adh} = 5.35 \times 10^6 \frac{N}{m}$

A.9 Shearing Failure Envelopes for Rolling Friction Coefficient 0.1 and Dry Material Conditions

Figure A.9 with Cohesion Stiffness $k_{adh} = 0.0 \frac{N}{m}$

A.10 Shearing Failure Envelopes for Rolling Friction Coefficient 0.2 and Dry Material Conditions

Figure A.10 with Cohesion Stiffness$k_{adh} = 0.0 \frac{N}{m}$

A.11 Shearing Failure Envelopes for Rolling Friction Coefficient 0.4 and Dry Material Conditions

Figure A.11 with Cohesion Stiffness $k_{adh} = 0.0 \frac{N}{m}$

A.12 Shearing Failure Envelopes for Rolling Friction Coefficient 0.6 and Dry Material Conditions

Figure A.12 with Cohesion Stiffness $k_{adh} = 0.0 \frac{N}{m}$

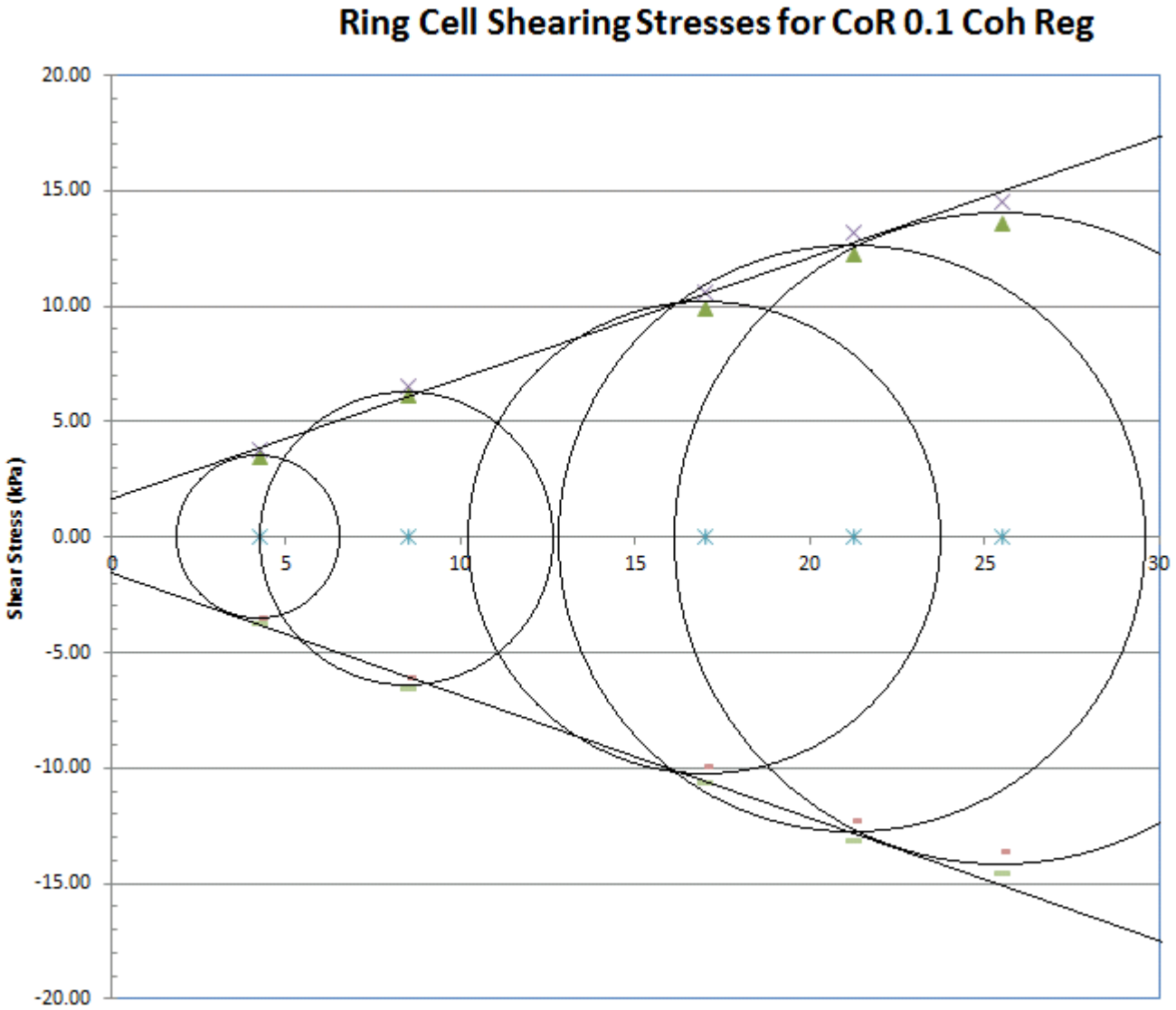


Figure A.5: Mohr Coulomb Failure envelope for the consolidated sheared cohesive material for CoR 0.1.

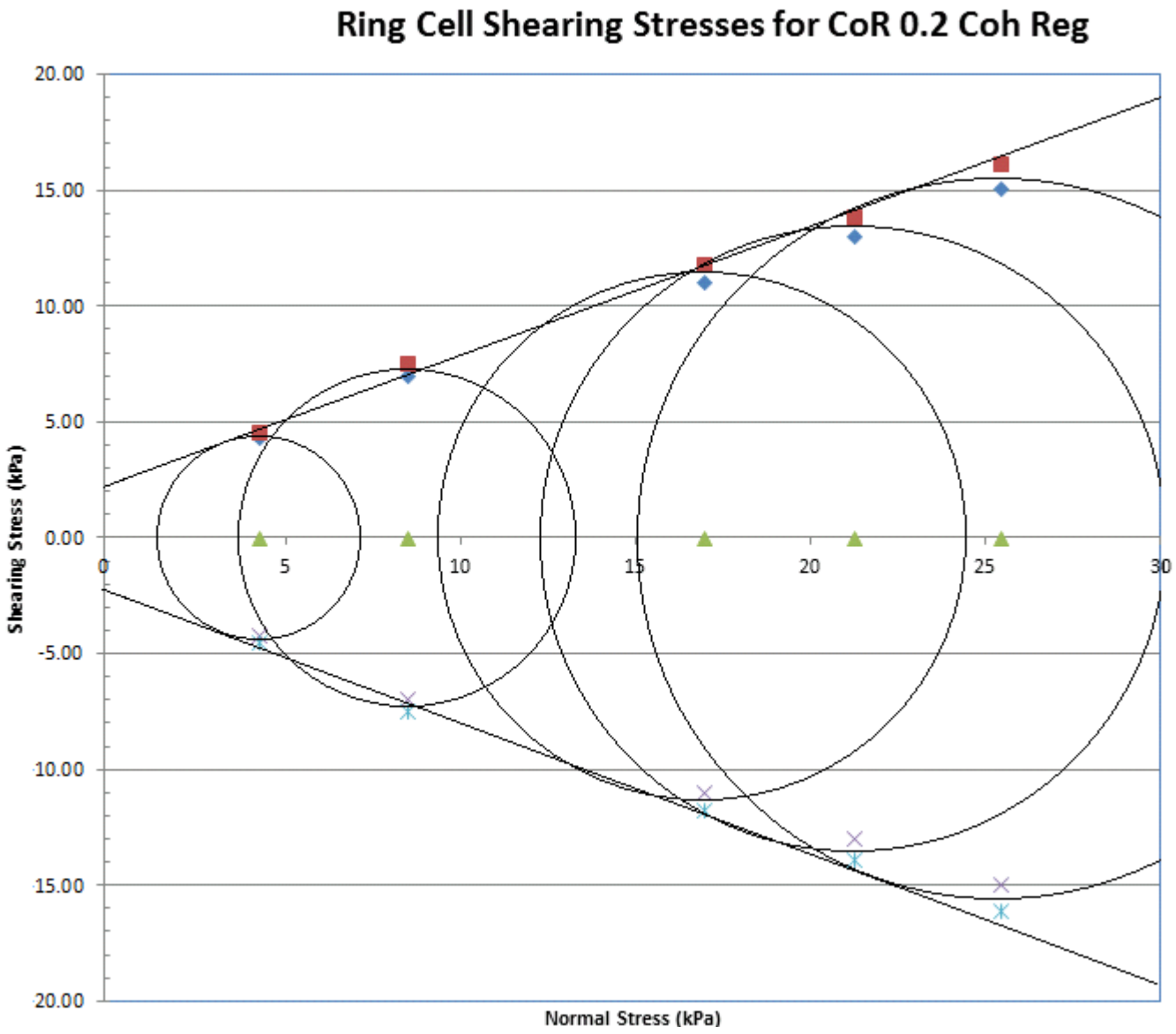


Figure A.6: Mohr Coulomb Failure envelope for the consolidated sheared cohesive material for CoR 0.2.

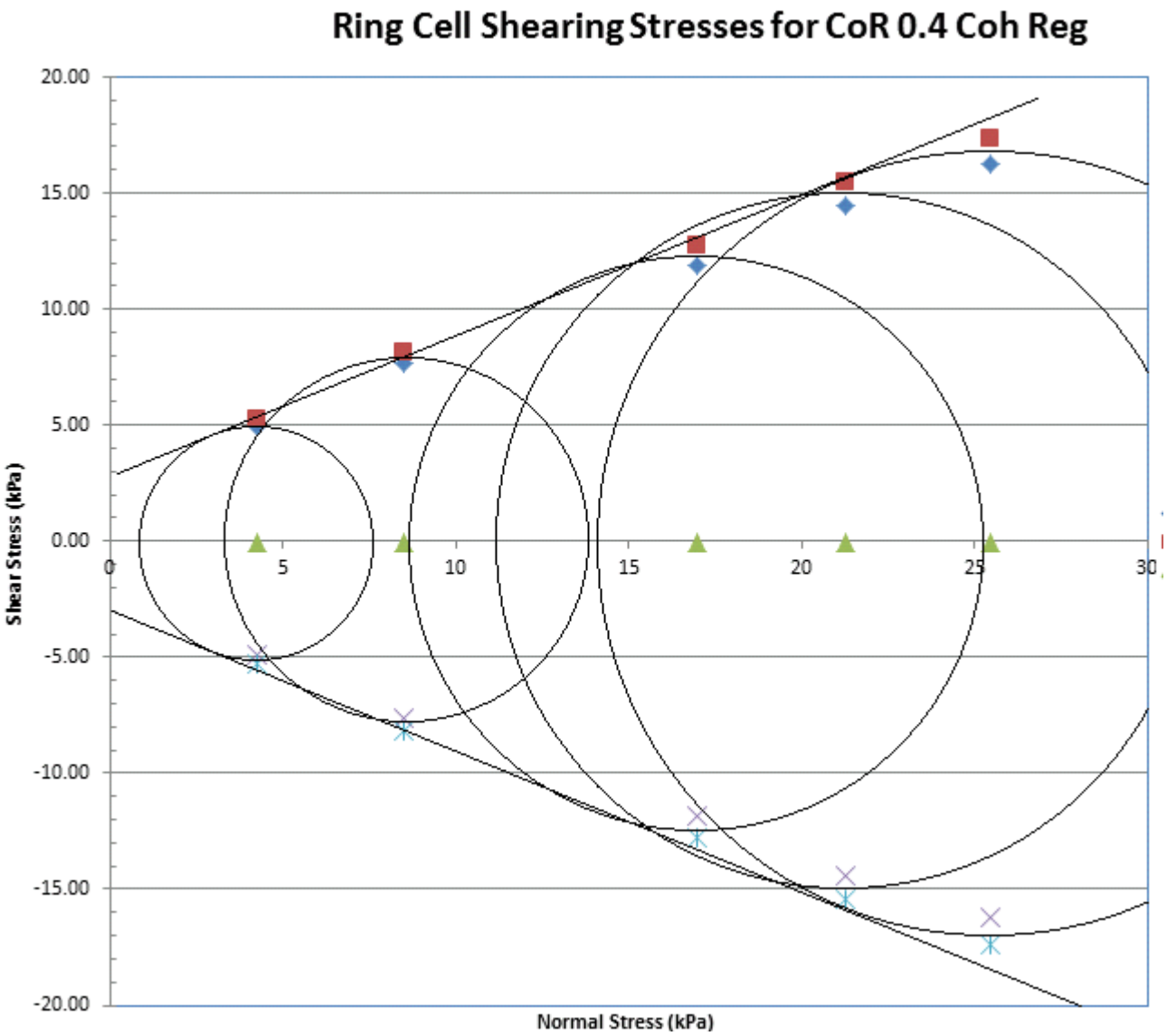


Figure A.7: Mohr Coulomb Failure envelope for the consolidated sheared cohesive material for CoR 0.4

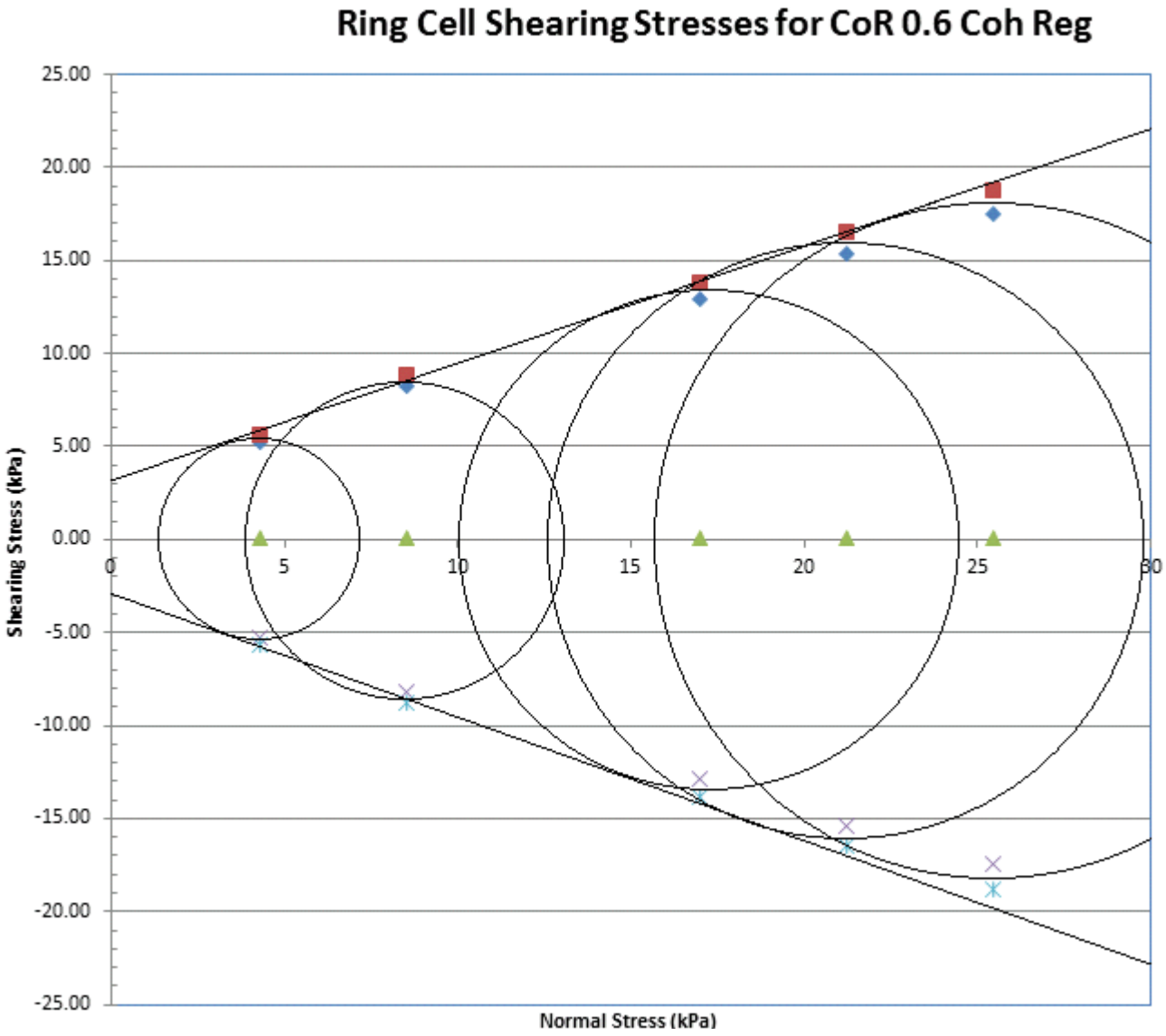


Figure A.8: Mohr Coulomb Failure envelope for the consolidated sheared cohesive material for CoR 0.6.

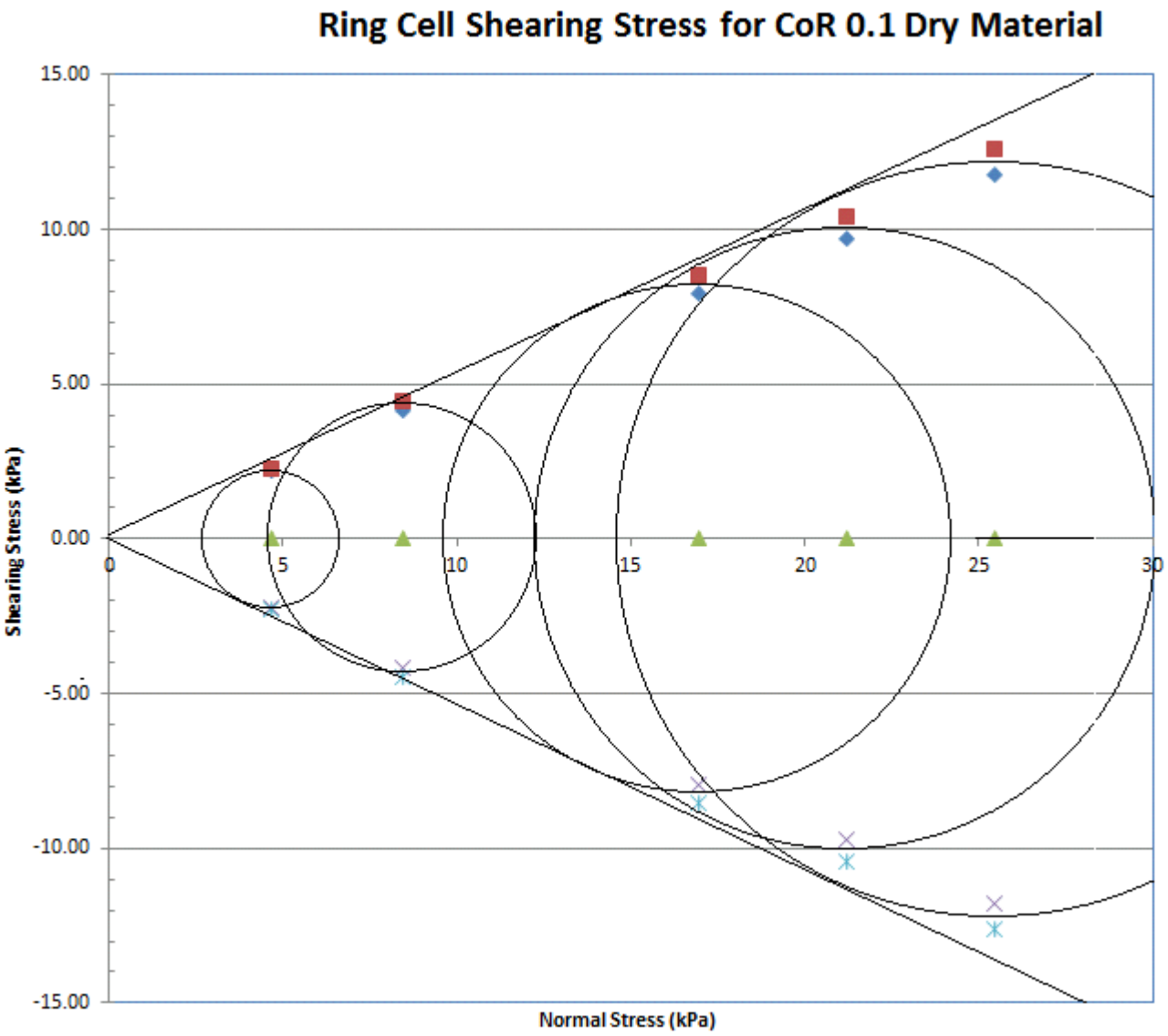


Figure A.9: Mohr Coulomb Failure envelope for the consolidated sheared dry material for CoR 0.1

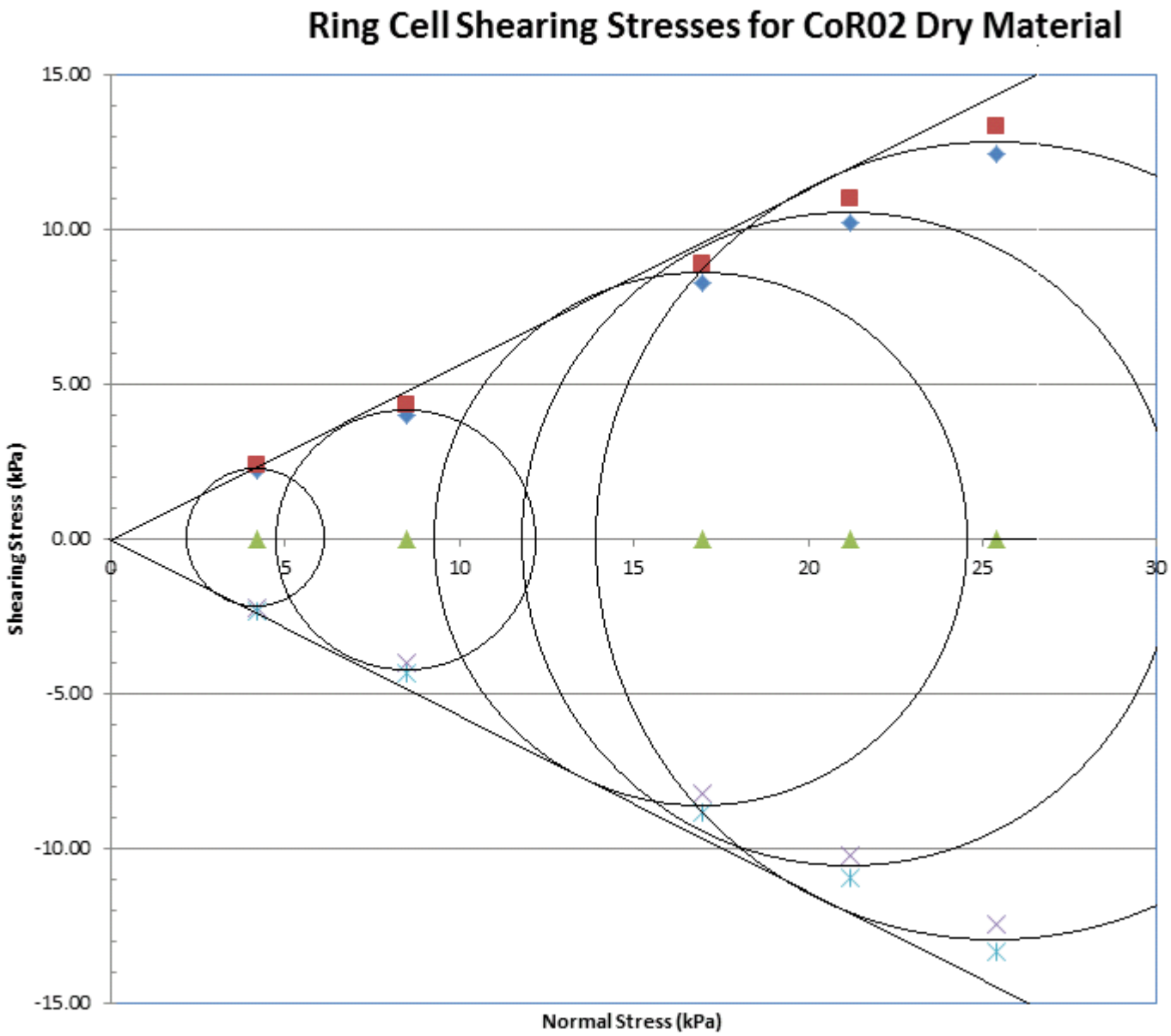


Figure A.10: Mohr Coulomb Failure envelope for the consolidated sheared dry material for CoR 0.2.

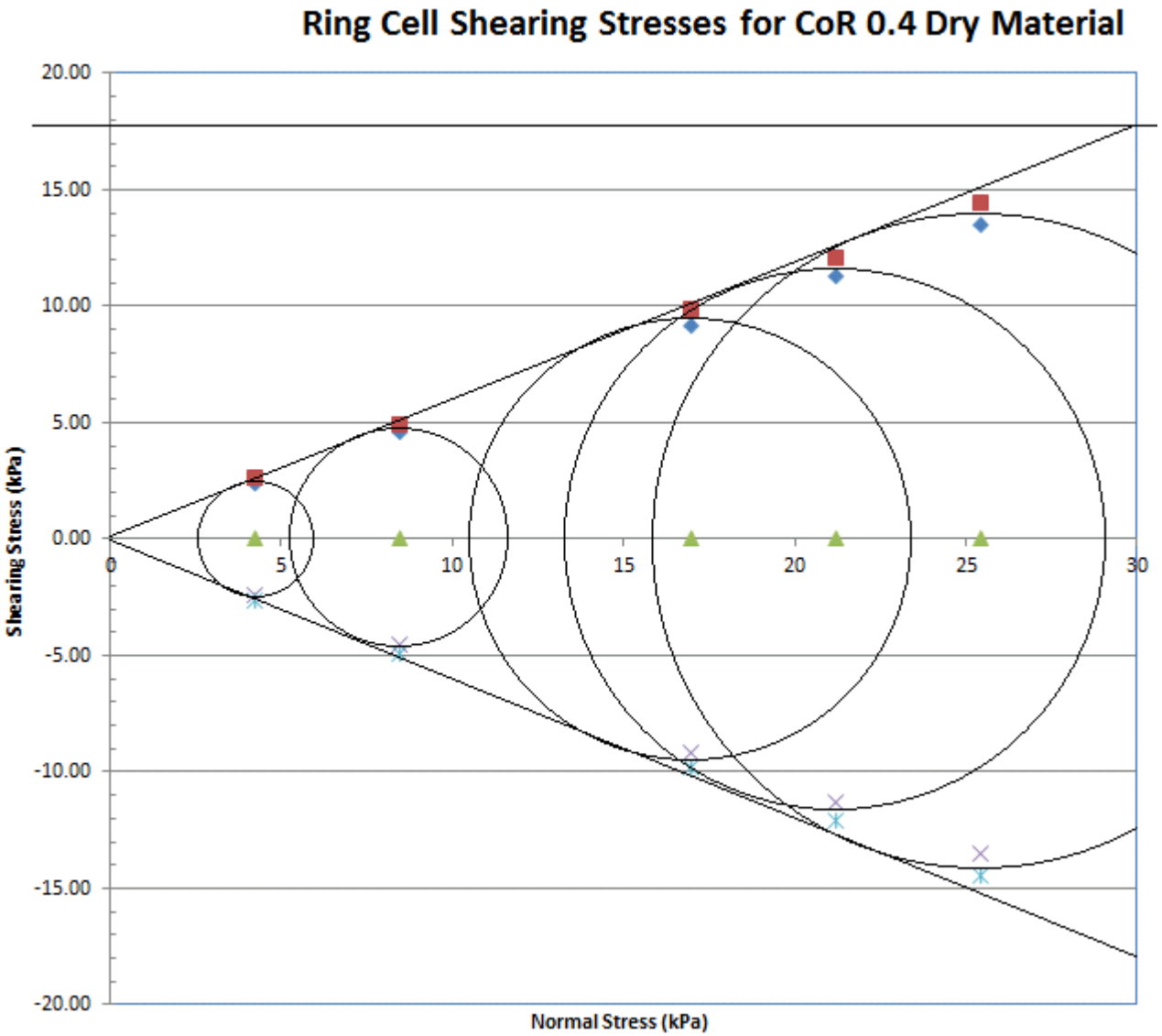


Figure A.11: Mohr Coulomb Failure envelope for the consolidated sheared dry material for CoR 0.4

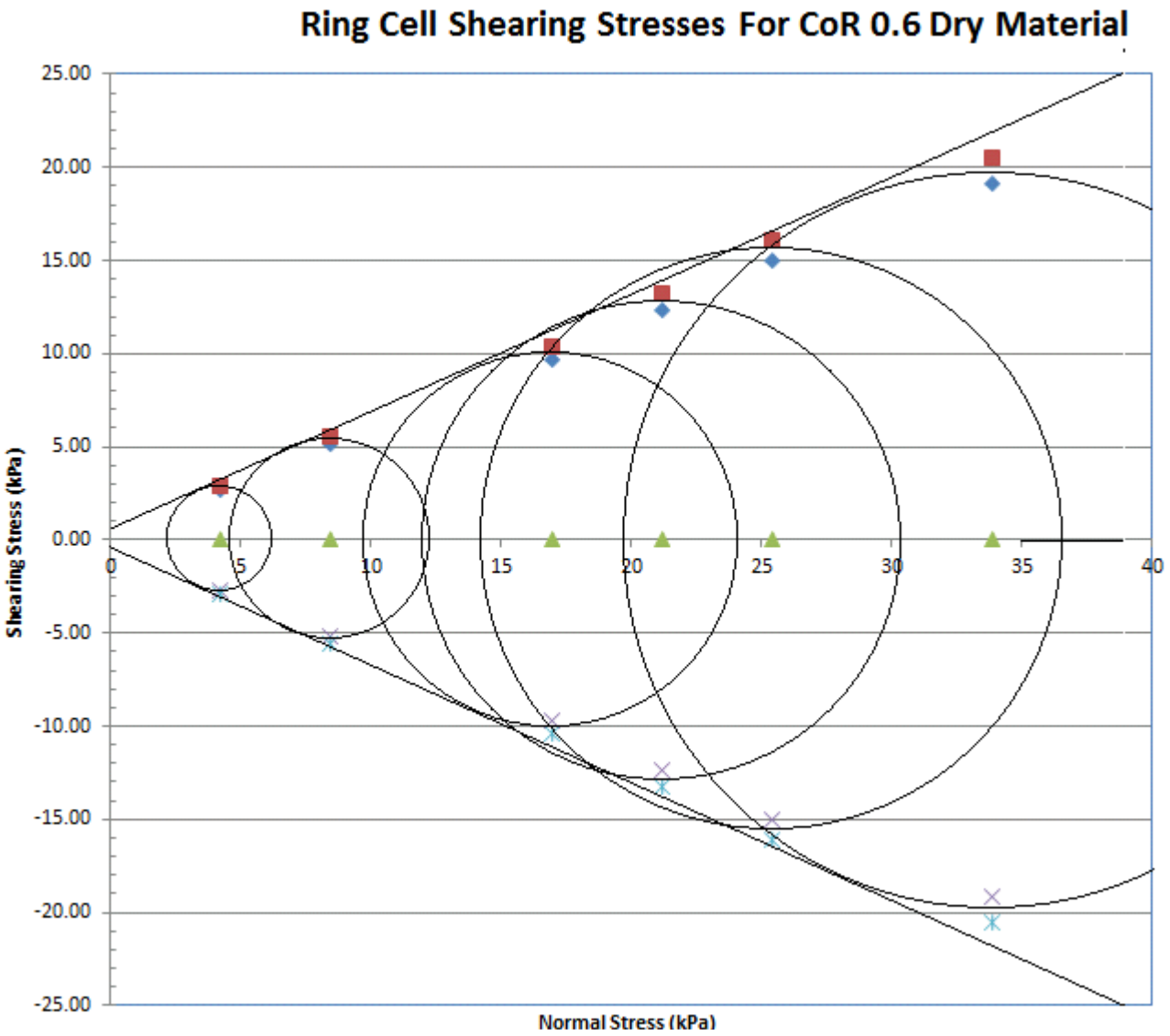


Figure A.12: Mohr Coulomb Failure envelope for the consolidated sheared dry material for CoR 0.6.

APPENDIX B - ANGLE OF REPOSE RESULTS FOR VARIOUS ROLLING FRICTION COEFFICIENTS AND COHESIVE STIFFNESSES

The angle of repose test, is used to determine the contour angle of a pile of unconsolidated bulk solid material. The test calls of a container of loose material to be filled and then container walls released. The angle with the horizontal at which the material settles measured.

B.1 Angle of Repose for Dry Material

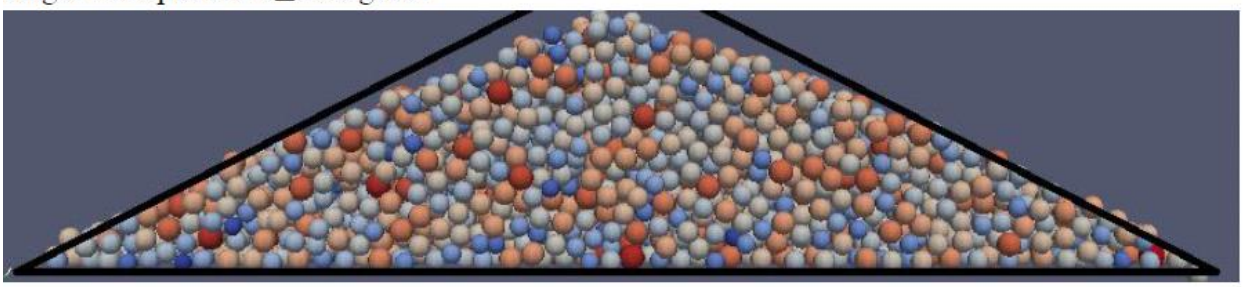
Figure B.1 shows sandpile formations for dry material of CoR 0.1, 0.2 and 0.4.

B.2 Angle of Repose for Cohesive Material at $k_{adh} = 1000\,\frac{N}{m}$

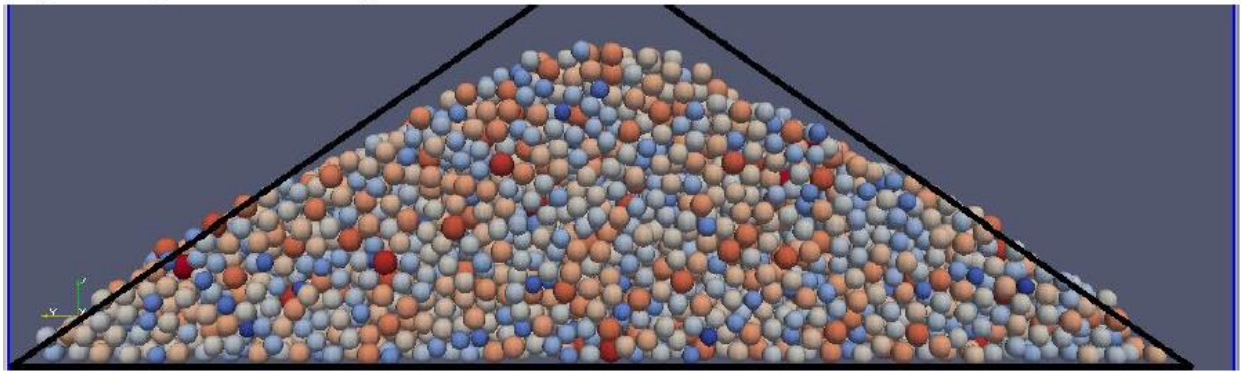
Figure B.2 shows sandpile formations for cohesive material of CoR 0.1, 0.2 and 0.4.

Contact Cohesion $k_{adh} = 0.0\ \frac{\mathbf{N}}{\mathbf{m}}$

Rolling Friction Coefficient, $\mu_r = 0.1$
Angle of Repose: 24 ± 3 degrees



Rolling Friction Coefficient, $\mu_r = 0.2$
Angle of Repose: 29 ± 3 degrees



Rolling Friction Coefficient, $\mu_r = 0.4$
Angle of Repose: 36 ± 3 degrees

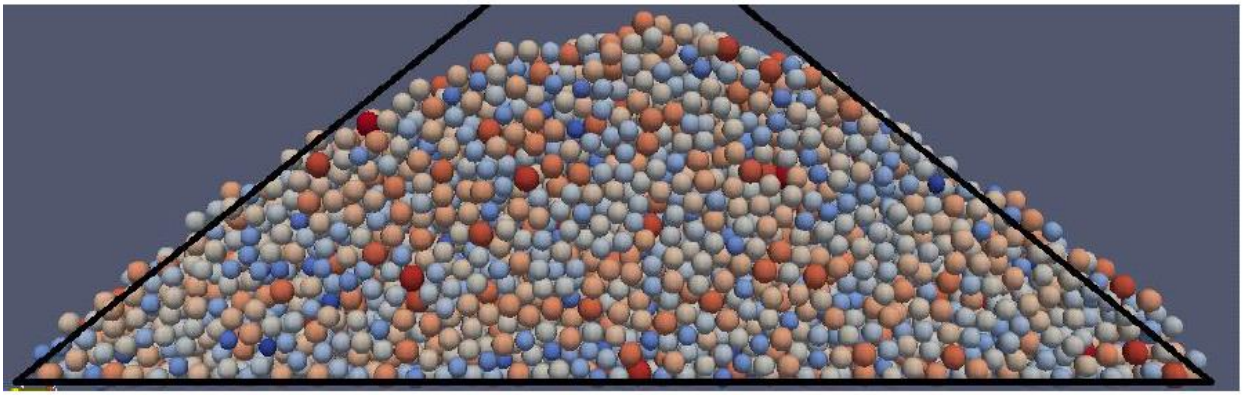
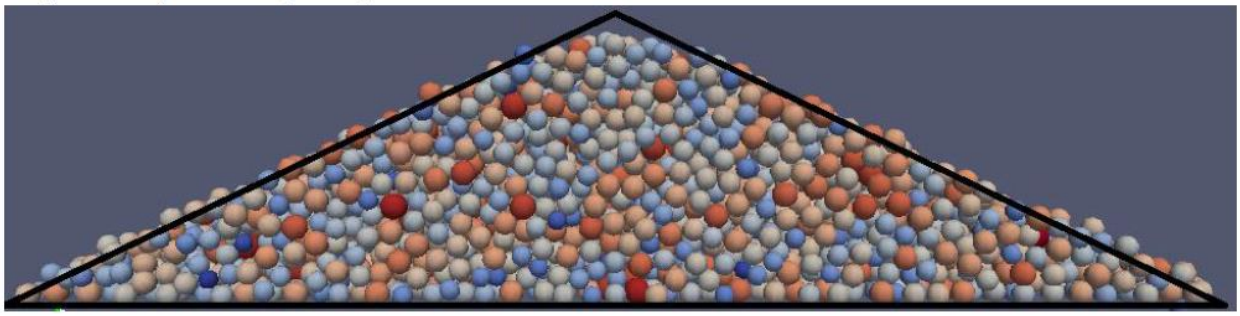


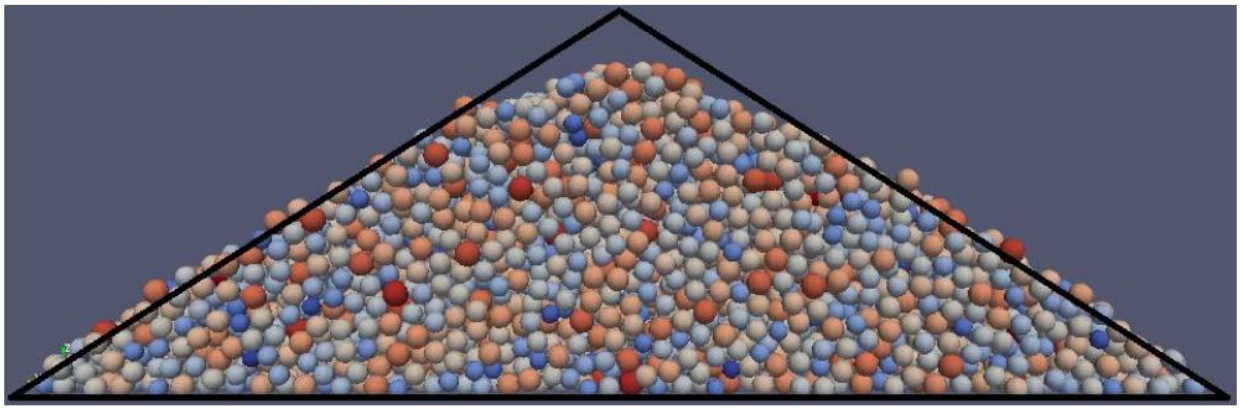
Figure B.1: Sandpile formation for various rolling friction coefficients of Dry Material.

Contact Cohesion $k_{adh} = 1000\ \frac{\mathbf{N}}{\mathbf{m}}$

Rolling Friction Coefficient, $\mu_r = 0.1$
Angle of Repose: 21 ± 3 degrees



Rolling Friction Coefficient, $\mu_r = 0.2$
Angle of Repose: 33 ± 3 degrees



Rolling Friction Coefficient, $\mu_r = 0.4$
Angle of Repose: 39 ± 3 degrees

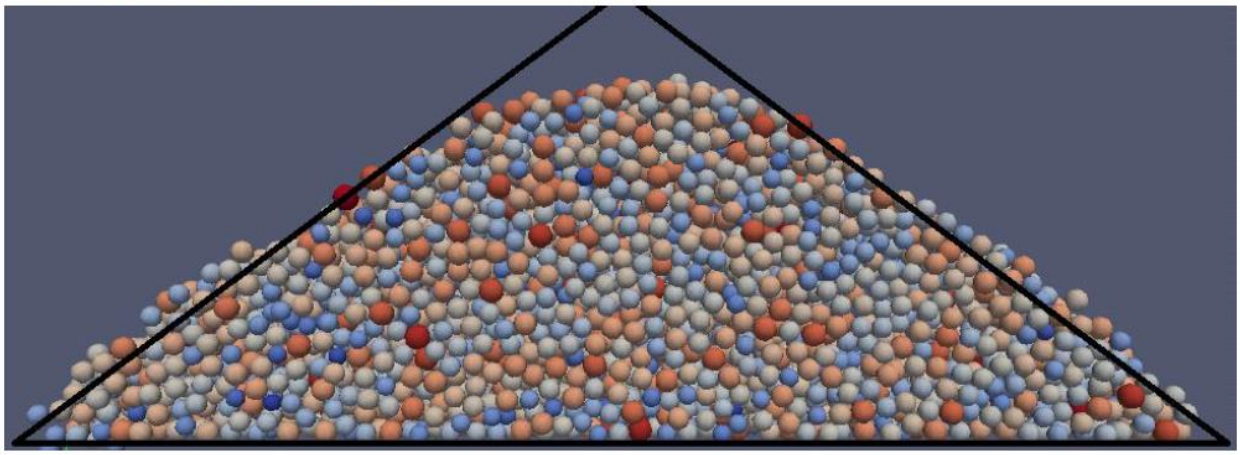


Figure B.2: Sandpile formation for various rolling friction coefficients of cohesive material.

APPENDIX C - THE IMPLEMENTED MEPA MODEL C++ CODE AND SCRIPT FILES FOR LIGGGHTS VS 2.3.8

C.1 Particle-Particle Contact Header File

/∗LIGGGHTS - LAMMPS Improved for General Granular and Granular Heat Transfer Simulations LIGGGHTS is part of the CFDEMproject www.liggghts.com — www.cfdem.com Christoph Kloss, christoph.kloss@cfdem.com Copyright 2009-2012 JKU Linz Copyright 2012- DCS Computing GmbH, Linz LIGGGHTS is based on LAMMPS LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator http://lammps.sandia.gov, Sandia National Laboratories Steve Plimpton, sjplimp@sandia.gov. This software is distributed under the GNU General Public License. See the README file in the top-level directory. ∗/

/∗Contributing authors for original version: Leo Silbert (SNL), Gary Grest (SNL), New Contributions by Liz Del Cid: Adding visco-elasto-plastic adhesive model under the original Luding model found in [45] ∗/

#ifdef PAIR_CLASS

PairStyle(gran/luding/history,PairGranludingHistory)

#else

#ifndef LMP_PAIR_GRAN_LUDING_HISTORY_H

#define LMP_PAIR_GRAN_LUDING_HISTORY_H

#include “pair_gran.h”

class PairGranludingHistory : public PairGran {

friend class FixWallGranludingHistory; friend class FixCheckTimestepGran;

public:

PairGranludingHistory(class LAMMPS ∗); ∼PairGranludingHistory();

virtual void settings(int, char ∗∗); virtual void init_granular();

virtual void compute_force(int eflag, int vflag, int addflag);

template <int ROLLINGFRICTION> void compute_force_eval(int eflag, int vflag, int addflag);

virtual void write_restart_settings(FILE *); virtual void read_restart_settings(FILE *);

protected:

virtual void history_args(char**); void allocate_properties(int);

bool forceoff() { return force_off; }

class FixPropertyGlobal* kload1; //Loading Stiffness

class FixPropertyGlobal* kunload1; //Unloading Stiffness

class FixPropertyGlobal* kcoh1; //Cohesive Stiffness

class FixPropertyGlobal* nPower1; //non linearity

class FixPropertyGlobal* Y1; //Youngs Modulus

class FixPropertyGlobal* v1; //Poisson's ratio

class FixPropertyGlobal*cohEnergyDens1; //Cohesion energy density

class FixPropertyGlobal* coeffMu1; // Fluid viscosity

class FixPropertyGlobal* coeffRestMax1; // Maximum restitution coefficient (for mu=0)

class FixPropertyGlobal* coeffStc1; // Critical Stokes number (10-30 for glass beads)

class FixPropertyGlobal*coeffRest1; //coefficient of restitution

class FixPropertyGlobal* coeffFrict1; //coefficient of (static) friction

class FixPropertyGlobal* coeffRollFrict1; //characteristic velocity needed for Linear Spring Model

class FixPropertyGlobal* coeffRollVisc1; //coefficient of rolling viscous damping (epsd model)

int charVelflag;

class FixPropertyGlobal* charVel1; //characteristic velocity needed for Linear Spring Model

double **kload, **kunload, **kcoh, **Yeff, **Geff, **betaeff,**veff, **cohEnergyDens, **coeffRestLog,**coeffFrict;

double nPower, charVel, **coeffRollFrict,**coeffRollVisc,**coeffMu,**coeffRestMax,**coeffStc;

virtual void deriveContactModelParams(int &ip, int &jp, double &meff, double &deltan, double &kn, double &kun, double &kad, double &kt, double &gamman, double &gammat, double &xmu, double &rmu, double &vnnr);

virtual void deriveContactLudingParams(int &ip, int &jp, double &dmax, double &Fmax, double &Fmin, double &d1max, double &dcmax);

virtual double LudingForce(double &ddot, double &DT, double &u, double &dp, double &up, double &kn, double &kun, double &kad, double &umax, double &u1max);

virtual void addCohesionForce(int &, int &,double &,double &);

int cohesionflag;

int dampflag, rollingflag, viscousflag;

}

#endif #endif

C.2 Particle-Wall Contact Header File

/* ———————————————————————- LIGGGHTS - LAMMPS Improved for General Granular and Granular Heat Transfer Simulations LIGGGHTS is part of the CFDEMproject www.liggghts.com — www.cfdem.com This file was modified with respect to the release in LAMMPS Modifications are Copyright 2009-2012 JKU Linz Copyright 2012- DCS Computing GmbH, Linz LAMMPS - Large-scale Atomic / Molecular Massively Parallel Simulator http://lammps.sandia.gov, Sandia National Laboratories Steve Plimpton, sjplimp@sandia.gov Copyright (2003) Sandia Corporation. Under the terms of Contract DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains certain rights in this software. This software is distributed under the GNU General Public License. See the README file in the top-level directory. */

/*Contributing authors for original version: Leo Silbert (SNL), Gary Grest (SNL) New Contributions by Liz Del Cid: Adding visco-elasto-plastic adhesive model under the original Luding model found in [45] */

#ifdef FIX_CLASS

FixStyle(wall/gran/luding/history,FixWallGranludingHistory)

#else

#ifndef LMP_FIX_WALL_GRAN_LUDING_HISTORY_H

#define LMP_FIX_WALL_GRAN_LUDING_HISTORY_H

#include “fix_wall_gran.h”

namespace LAMMPS_NS {

class FixWallGranludingHistory : public FixWallGran { public: FixWallGranludingHistory(class LAMMPS *, int, char **); ∼FixWallGranludingHistory();

protected: virtual void post_create();

virtual void init_granular();

virtual void init_heattransfer();

void addHeatFlux(TriMesh *mesh,int ip, double rsq, double area_ratio);

virtual void compute_force(int ip, double deltan, double rsq,double meff_wall,double dx, double dy, double dz,double *vwall, double *c_history,double area_ratio);

virtual void addCohesionForce(int &ip, double &r, double &Fn_coh,double area_ratio);

template <int ROLLINGFRICTION>

void addRollingFrictionTorque(int ip, double wr1,double wr2,double wr3,double cr,double ccel,double r,double mi,double rmu,double kun,double kt,double dx, double dy, double dz,double rsqinv,double *c_history,double *r_torque);

virtual void deriveContactModelParams(int ip, double deltan,double meff_wall, double &kn,double &kun,double &kad,double &kt, double &gamman, double &gammat, double &xmu,double &rmu,double &vnnr);

virtual void deriveContactLudingParams(int ip, double &dmax, double &Fmax, double &Fmin, double &d1max, double &dcmax);

virtual double LudingForce(double &ddot, double &DT, double &u, double &dp, double &dcmax, double &kn, double &kun, double &kad, double &umax, double &u1max);

virtual void pre_reset_history(int,double*) {}

int dampflag, cohesionflag, rollingflag, viscousflag;

double **kload, **kunload, **kcoh, **Yeff, **Geff, **betaeff, **veff, **cohEnergyDens, **coeffRestLog, **coeffFrict, **coeffRollVisc;

double nPower, charVel, **coeffRollFrict, **coeffMu, **coeffRestMax, **coeffStc;

// heat transfer

class FixPropertyAtom *fppa_T; class FixPropertyAtom *fppa_hf;

double Temp_wall; double Q,Q_add;

const double *th_cond; double const* const* deltan_ratio; };

}

#endif #endif

C.3 Particle-Particle Contact C++ Code

/* LIGGGHTS - LAMMPS Improved for General Granular and Granular Heat Transfer Simulations LIGGGHTS is part of the CFDEMproject www.liggghts.com — www.cfdem.com Christoph Kloss, christoph.kloss@cfdem.com Copyright 2009-2012 JKU Linz Copyright 2012- DCS Computing GmbH, Linz

LIGGGHTS is based on LAMMPS LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator http://lammps.sandia.gov, Sandia National Laboratories Steve Plimpton, sjplimp@sandia.gov. This software is distributed under the GNU General Public License. See the README file in the top-level directory. */

/* Contributing authors for original version: Leo Silbert (SNL), Gary Grest (SNL) Added New Contributions by Liz Del Cid: Adding visco-elasto-plastic adhesive model under the original Luding model found in [45] */

#include “math.h”

#include “stdio.h”

#include “stdlib.h”

#include “string.h”

```
#include "pair_gran_luding_history.h"
#include "atom.h"
#include "atom_vec.h"
#include "domain.h"
#include "force.h"
#include "update.h"
#include "modify.h"
#include "fix.h"
#include "fix_contact_history.h"
#include "comm.h"
#include "neighbor.h"
#include "neigh_list.h"
#include "neigh_request.h"
#include "memory.h"
#include "error.h"
#include "fix_rigid.h"
#include "fix_property_global.h"
#include "mech_param_gran.h"
#include "compute_pair_gran_local.h"
#include "vector_liggghts.h"
#include "math_extra_liggghts.h"
using namespace LAMMPS_NS;
#define MIN(a,b) ((a) < (b) ? (a) : (b))
#define MAX(a,b) ((a) > (b) ? (a) : (b))
#define sqrtFiveOverSix 0.91287092917527685576161630466800355658790782499663875
/* ---------------------------------------------------------------------- */
```

PairGranludingHistory::PairGranludingHistory(LAMMPS *lmp) : PairGran(lmp) { //flag that we intend to use contact history

history = 1;

dnum_pairgran = 4;

kload = NULL;

kunload = NULL;

kcoh = NULL;

Yeff = NULL;

Geff = NULL;

betaeff = NULL;

veff = NULL;

cohEnergyDens = NULL;

coeffRestLog = NULL;

coeffFrict = NULL;

coeffRollFrict = NULL;

coeffRollVisc = NULL;

coeffMu = NULL;

coeffRestMax = NULL;

coeffStc = NULL;

charVelflag = 0;

force_off = false;

sanity_checks = true; }

/* ———————————————————————- */

PairGranludingHistory::~PairGranludingHistory() {

memory->destroy(kload);

memory->destroy(kunload);

memory->destroy(kcoh);

memory->destroy(Yeff);

memory->destroy(Geff);

memory->destroy(betaeff);

memory->destroy(veff);

memory->destroy(cohEnergyDens);

memory->destroy(coeffRestLog);

memory->destroy(coeffFrict);

memory->destroy(coeffRollFrict);

memory->destroy(coeffRollVisc);

memory->destroy(coeffMu);

memory->destroy(coeffRestMax);

memory->destroy(coeffStc); }

/* ———————————————————————————————————- */

void PairGranludingHistory::history_args(char** args) { //provide names and newton-flags for each history value //newtonflag = 0 means that the value args[0] = (char *) “shearx”;

args[1] = (char *) “1”;

args[2] = (char *) “sheary”;

args[3] = (char *) “1”;

args[4] = (char *) “shearz”;

args[5] = (char *) “1”;

args[6] = (char *) “dplastic”;

args[7] = (char *) “1”;

if (rollingflag == 2 —— rollingflag == 3)

{ args[8] = (char *) “r_torquex_old”;

args[9] = (char *) “1”;

args[10] = (char *) “r_torquey_old”;

args[11] = (char *) “1”;

```
args[12] = (char *) "r_torquez_old";
args[13] = (char *) "1"; } }
/* ---------------------------------------------------------- */
inline void PairGranludingHistory::addCohesionForce(int &ip, int &jp,double &r, double &Fn_coh) { //r is the distance between the sphere's centeres double ri = atom->radius[ip]; double rj = atom->radius[jp]; double Acont;
if(cohesionflag == 1)
Acont = - M_PI/4 * ( (r-ri-rj)*(r+ri-rj)*(r-ri+rj)*(r+ri+rj) )/(r*r); //contact area of the two spheres
else Acont = M_PI * 2. * (2.*ri*rj/(ri+rj)) * (ri + rj - r);
Fn_coh=cohEnergyDens[atom->type[ip]][atom->type[jp]]*Acont; }
/* ---------------------------------------------------------- */
inline void PairGranludingHistory::deriveContactModelParams(int &ip, int &jp,double &meff,double &deltan, double &kn, double &kun, double &kad, double &kt, double &gamman, double &gammat, double &xmu, double &rmu, double &vnnr) {
int itype = atom->type[ip];
int jtype = atom->type[jp];
double rj = atom->radius[jp];
double ri = atom->radius[ip];
double reff=ri*rj/(ri+rj);
double stokes, coeffRestLogChosen;
if (viscousflag) {
coeffRestLogChosen=log(coeffRestMax[itype][jtype])+coeffStc[itype][jtype]/stokes; }
else { coeffRestLogChosen=coeffRestLog[itype][jtype]; } double sqrtval = sqrt(reff*deltan);
double Sn=2.*Yeff[itype][jtype]*sqrtval; double St=8.*Geff[itype][jtype]*sqrtval;
gamman=-2.*sqrtFiveOverSix*betaeff[itype][jtype]*sqrt(Sn*meff);
gammat=-2.*sqrtFiveOverSix*betaeff[itype][jtype]*sqrt(St*meff);
```

```
    kn = kload[itype][jtype]; kun = kunload[itype][jtype];
    kad = kcoh[itype][jtype];
    kt = kn;
    xmu=coeffFrict[itype][jtype];
    if(rollingflag)rmu=coeffRollFrict[itype][jtype];
    if (dampflag == 0) gammat = 0.0;
    // convert Kn and Kt from pressure units to force/distance^2
    /* kn /= force->nktv2p; kun /=force->nktv2p; kt /= force->nktv2p;*/
    return; }
    /* ---------------------------------------------------------- */
    inline void PairGranludingHistory::deriveContactLudingParams(int &ip, int &jp, double
&dmax, double &Fmax, double &Fmin, double &d1max, double &dcmax) {
    int itype = atom->type[ip];
    int jtype = atom->type[jp];
    double rj = atom->radius[jp];
    double ri = atom->radius[ip];
    double reff=ri*rj/(ri+rj);
    double tempdmax;
    double tempinv = 1.0/nPower;
    dmax= reff*0.1;
    //max penetration is 10% of the effective radius double tempPower = pow(dmax,nPower);
    // maximum elastic-plastic loading force and minimum unloading force
    Fmax = ( kload[itype][jtype] * kunload[itype][jtype] ) / (kunload[itype][jtype] - kload[itype][jtype]
) * tempPower;
    Fmin = -kcoh[itype][jtype] * kunload[itype][jtype] / ( kunload[itype][jtype] + kcoh[itype][jtype]
) * tempPower;
    tempdmax = (kunload[itype][jtype] / (kunload[itype][jtype] - kload[itype][jtype]));
```

d1max=pow(tempdmax,tempinv)*dmax;

tempdmax = (kunload[itype][jtype] / (kcoh[itype][jtype] + kunload[itype][jtype])); dc-
max=pow(tempdmax, tempinv)*dmax;

return; }

/* ———————————————————————————————- */

void PairGranludingHistory::compute_force(int eflag, int vflag,int addflag) {

if (rollingflag == 0) compute_force_eval<0>(eflag,vflag,addflag);

else

if(rollingflag == 1) compute_force_eval<1>(eflag,vflag,addflag);

else

if(rollingflag == 2) compute_force_eval<2>(eflag,vflag,addflag);

else

if(rollingflag == 3) compute_force_eval<3>(eflag,vflag,addflag); }

/* ———————————————————————————————- */

template <int ROLLINGFRICTION> void PairGranludingHistory:: compute_force_eval(int eflag, int vflag,int addflag) {

//calculated from the material properties double kn, kun, kad, kt, kr, gamman, gammat, xmu, rmu; double Fn_coh;

int i,j,ii,jj,inum,jnum,itype,jtype;

double xtmp,ytmp,ztmp,delx,dely,delz,fx,fy,fz;

double radi,radj,radsum,rsq,r,rinv,rsqinv,reff;

double vr1,vr2,vr3,vnnr,vn1,vn2,vn3,vt1,vt2,vt3, wr_roll[3],wr_rollmag; double wr1,wr2,wr3;

double vtr1,vtr2,vtr3,vrel;

double mi,mj,meff,damp,ccel,tor1,tor2,tor3,r_torque[3],r_torque_n[3],dr_torque[3];

double fn,fs,fs1,fs2,fs3;

double shrmag,rsht, cri, crj;

int *ilist,*jlist,*numneigh,**firstneigh; int *touch,**firsttouch;

```
double *shear,*allshear,**firstshear;
double Fmax, Fmin, d1max, dcmax, dmax;
double r_inertia,r_coef,r_torque_mag,r_torque_max,factor;
if (eflag —— vflag) ev_setup(eflag,vflag);
else evflag = vflag_fdotr = 0;
double **x = atom->x;
double **v = atom->v;
double **f = atom->f;
double **omega = atom->omega;
double **torque = atom->torque;
double *radius = atom->radius;
double *rmass = atom->rmass;
double *mass = atom->mass;
int *type = atom->type;
int *mask = atom->mask;
int nlocal = atom->nlocal;
inum = list->inum;
ilist = list->ilist;
numneigh = list->numneigh;
firstneigh = list->firstneigh;
firsttouch = listgranhistory->firstneigh;
firstshear = listgranhistory->firstdouble;
// loop over neighbors of my atoms
for (ii = 0; ii < inum; ii++) {
i = ilist[ii];
xtmp = x[i][0];
ytmp = x[i][1];
```

```
ztmp = x[i][2];
radi = radius[i];
touch = firsttouch[i];
allshear = firstshear[i];
jlist = firstneigh[i];
jnum = numneigh[i];
double DT = update->dt;
for (jj = 0; jj < jnum; jj++) {
j = jlist[jj];
j &= NEIGHMASK;
delx = xtmp - x[j][0];
dely = ytmp - x[j][1];
delz = ztmp - x[j][2];
rsq = delx*delx + dely*dely + delz*delz;
radj = radius[j]; radsum = radi + radj;
if (rsq >= radsum*radsum) {
// unset non-touching neighbors
touch[jj] = 0;
shear = &allshear[dnum()*jj];
shear[0] = 0.0;
shear[1] = 0.0;
shear[2] = 0.0;
shear[3] = 0.0; //this holds the plasticity parameter for deformation
if (ROLLINGFRICTION == 2 —— ROLLINGFRICTION == 3) {
shear[4] = 0.0; // this is the r_torque_old
shear[5] = 0.0; // this is the r_torque_old
shear[6] = 0.0; // this is the r_torque_old }
```

```
} else {
r = sqrt(rsq); rinv = 1.0/r;
rsqinv = 1.0/rsq;
// relative translational velocity
vr1 = v[i][0] - v[j][0];
vr2 = v[i][1] - v[j][1];
vr3 = v[i][2] - v[j][2];
// normal component
vnnr = vr1*delx + vr2*dely + vr3*delz;
vn1 = delx*vnnr * rsqinv;
vn2 = dely*vnnr * rsqinv;
vn3 = delz*vnnr * rsqinv;
// tangential component
vt1 = vr1 - vn1;
vt2 = vr2 - vn2;
vt3 = vr3 - vn3;
// relative rotational velocity
double deltan=radsum-r;
cri = radi-0.5*deltan;
crj = radj-0.5*deltan;
wr1 = (cri*omega[i][0] + crj*omega[j][0]) * rinv;
wr2 = (cri*omega[i][1] + crj*omega[j][1]) * rinv;
wr3 = (cri*omega[i][2] + crj*omega[j][2]) * rinv;
// normal forces = Hookian contact + normal velocity damping
// meff = effective mass of pair of particles
// if I or J part of rigid body, use body mass
// if I or J is frozen, meff is other particle
```

```
if (rmass) { mi = rmass[i]; mj = rmass[j]; }
else { itype = type[i]; jtype = type[j]; mi = mass[itype]; mj = mass[jtype]; }
if (fix_rigid) {
if(body[i] >= 0)
mi = masstotal[body[i]];
if(body[j] >= 0)
mj = masstotal[body[j]]; }
meff = mi*mj/(mi+mj);
if (mask[i] & freeze_group_bit)
meff = mj;
if (mask[j] & freeze_group_bit)
meff = mi;
shear = &allshear[dnum()*jj];
deriveContactModelParams(i, j, meff, deltan, kn, kun, kad, kt, gamman, gammat, xmu,
rmu, vnnr);
deriveContactLudingParams(i, j, dmax, Fmax, Fmin, d1max, dcmax);
// normal forces = Hookian contact + normal velocity damping
damp = gamman*vnnr*rsqinv;
double ddot = -1.0*vnnr; double dp = 0.0;
double up= shear[3];
//initial plastic deformation if (up < 0.0) { up=-up;}
//DT=0.5*DT;
ccel =LudingForce(ddot, DT, deltan, dp, up, kn, kun, kad, dmax, d1max) *rinv-damp;
DT = dt;
shear[3]= dp;
if (cohesionflag) { addCohesionForce(i,j,r,Fn_coh); ccel-=Fn_coh*rinv; }
// relative velocities
```

```
vtr1 = vt1 - (delz*wr2-dely*wr3);
vtr2 = vt2 - (delx*wr3-delz*wr1);
vtr3 = vt3 - (dely*wr1-delx*wr2);
vrel = vtr1*vtr1 + vtr2*vtr2 + vtr3*vtr3; vrel = sqrt(vrel);
// shear history effects
touch[jj] = 1;
//shear = &allshear[dnum()*jj];
if (shearupdate && computeflag) {
shear[3] = dp;
shear[0] += vtr1*dt;
shear[1] += vtr2*dt;
shear[2] += vtr3*dt;
// rotate shear displacements
rsht = shear[0]*delx + shear[1]*dely + shear[2]*delz;
rsht *= rsqinv; shear[0] -= rsht*delx;
shear[1] -= rsht*dely; shear[2] -= rsht*delz; }
shrmag = sqrt(shear[0]*shear[0] + shear[1]*shear[1] + shear[2]*shear[2]);
// tangential forces = shear + tangential velocity damping
fs1 = - (kt*shear[0]);
fs2 = - (kt*shear[1]);
fs3 = - (kt*shear[2]);
// rescale frictional displacements and forces if needed
fs = sqrt(fs1*fs1 + fs2*fs2 + fs3*fs3);
fn = xmu * fabs(ccel*r);
// energy loss from sliding or damping
if (fs > fn) { if (shrmag != 0.0) {
fs1 *= fn/fs;
```

```
fs2 *= fn/fs;
fs3 *= fn/fs;
shear[0] = -fs1/kt;
shear[1] = -fs2/kt;
shear[2] = -fs3/kt; }
else fs1 = fs2 = fs3 = 0.0; }
else { fs1 -= (gammat*vtr1); fs2 -= (gammat*vtr2); fs3 -= (gammat*vtr3); }
// forces & torques
fx = delx*ccel + fs1; fy = dely*ccel + fs2; fz = delz*ccel + fs3;
tor1 = rinv * (dely*fs3 - delz*fs2);
tor2 = rinv * (delz*fs1 - delx*fs3);
tor3 = rinv * (delx*fs2 - dely*fs1);
// add rolling friction torque
vectorZeroize3D(r_torque);
if(ROLLINGFRICTION > 0) {
if(ROLLINGFRICTION == 1) { vectorSubtract3D(omega[i],omega[j],wr_roll);
wr_rollmag = vectorMag3D(wr_roll);
if(wr_rollmag > 0.) { // calculate torque
reff=radi*radj/(radi+radj);
vectorScalarMult3D(wr_roll,rmu*kun*deltan*reff/wr_rollmag,r_torque);
// remove normal (torsion) part of torque
double rtorque_dot_delta = r_torque[0]*delx + r_torque[1]*dely + r_torque[2]*delz;
r_torque_n[0] = delx * rtorque_dot_delta * rsqinv;
r_torque_n[1] = dely * rtorque_dot_delta * rsqinv;
r_torque_n[2] = delz * rtorque_dot_delta * rsqinv;
vectorSubtract3D(r_torque,r_torque_n,r_torque); } }
else // ROLLINGFRICTION == 2 —— 3 {
```

```
double wr_roll_n[3],wr_roll_t[3]; double r_inertia_red_i,r_inertia_red_j;
itype = type[i]; jtype = type[j];
// relative rotational velocity vectorSubtract3D(omega[i],omega[j],wr_roll);
// remove normal (torsion) part of relative rotation // use only tangential parts for rolling
torque
double wr_dot_delta = wr_roll[0]*delx+ wr_roll[1]*dely + wr_roll[2]*delz;
wr_roll_n[0] = delx * wr_dot_delta * rsqinv;
wr_roll_n[1] = dely * wr_dot_delta * rsqinv;
wr_roll_n[2] = delz * wr_dot_delta * rsqinv;
vectorSubtract3D(wr_roll,wr_roll_n,wr_roll_t);
// spring reff=radi*radj/(radi+radj);
if(ROLLINGFRICTION == 2)
kr = 2.25*kun*rmu*rmu*reff*reff; else kr = kt*reff*reff;
vectorScalarMult3D(wr_roll_t,update->dt*kr,dr_torque);
r_torque[0] = shear[4] + dr_torque[0];
r_torque[1] = shear[5] + dr_torque[1]; r_torque[2] = shear[6] + dr_torque[2];
// limit max. torque
r_torque_mag = vectorMag3D(r_torque);
r_torque_max = fabs(ccel*r)*reff*rmu;
if(r_torque_mag > r_torque_max) { factor = r_torque_max / r_torque_mag;
r_torque[0] *= factor; r_torque[1] *= factor; r_torque[2] *= factor;
// save rolling torque due to spring
shear[4] = r_torque[0];
shear[5] = r_torque[1];
shear[6] = r_torque[2];
// no damping / no dashpot in case of full mobilisation rolling angle // r_coef = 0.0;
} else {
```

```
// save rolling torque due to spring before adding damping torque
shear[4] = r_torque[0]; shear[5] = r_torque[1]; shear[6] = r_torque[2];
// dashpot only for the original epsd model
if(ROLLINGFRICTION == 2) { // dashpot r_inertia_red_i = mi*radi*radi;
r_inertia_red_j = mj*radj*radj;
if (domain->dimension == 2)
r_inertia = 1.5 * r_inertia_red_i * r_inertia_red_j/(r_inertia_red_i + r_inertia_red_j);
else r_inertia = 1.4 * r_inertia_red_i * r_inertia_red_j/(r_inertia_red_i + r_inertia_red_j);
r_coef = coeffRollVisc[itype][jtype] * 2 * sqrt(r_inertia*kr);
// add damping torque
r_torque[0] += r_coef*wr_roll_t[0];
r_torque[1] + = r_coef*wr_roll_t[1];
r_torque[2] += r_coef*wr_roll_t[2]; } }
}
}
if(computeflag) { f[i][0] += fx; f[i][1] += fy; f[i][2] += fz;
torque[i][0] -= cri*tor1 + r_torque[0];
torque[i][1] -= cri*tor2 + r_torque[1];
torque[i][2] -= cri*tor3 + r_torque[2]; }
if (j < nlocal && computeflag) {
f[j][0] -= fx; f[j][1] -= fy;
f[j][2] -= fz;
torque[j][0] -= crj*tor1 - r_torque[0];
torque[j][1] -= crj*tor2 - r_torque[1];
torque[j][2] -= crj*tor3 - r_torque[2]; }
if(cpl && addflag) cpl->add_pair(i,j,fx,fy,fz,tor1,tor2,tor3,shear);
if (evflag) ev_tally_xyz(i,j,nlocal,0,0.0,0.0,fx,fy,fz,delx,dely,delz); } } }
```

```
}
/* ----------------------------------------------------------------------*/
double PairGranludingHistory::LudingForce(double &ddot, double &DT, double &u, double &dp, double &up, double &kn, double &kun, double &kad, double &umax, double &u1max)
//LudingForce(double &kn, double &deltatemp, double &damp) {
//double Flud = kn*pow(deltatemp,nPower)-damp; //return Flud;
double changeu = ddot*DT;
double unew = u+changeu; double Fcal, F2n, F1n, Fcn; double dpnum; double NPinv = 1.0/nPower;
if (unew < 0.0 ){ dp = 0.0; Fcal = 0.0; return Fcal; }
F2n = kun*(pow(unew,nPower)-pow(up,nPower));
if (ddot >= 0.0){
F1n = kn*pow(unew,nPower);
if ((F2n-F1n) > 0.0){
dpnum = (kun-kn)/kun;
dp = pow(dpnum,NPinv)*unew;
if (dp > umax){
dp = umax; }
Fcal = F1n;
if (unew > u1max){
dp = umax;
Fcal = kun*(pow(unew,nPower)-pow(umax,nPower)); }
}else{
dp = up; if (dp > umax){ dp = umax; }
Fcal = F2n; }
}else{
```

```
Fcn = -1.0*kad*pow(unew,nPower);
if ((F2n-Fcn) < 0.0){
dpnum = (kun+kad)/kun;
dp = pow(dpnum,NPinv)*unew;
if (dp > umax){ dp = umax; }
Fcal = Fcn;
}else{
dp = up;
if (dp > umax){
dp = umax; }
Fcal = F2n; } }
if (dp > umax){ dp = umax; } if (dp < 0.0){ dp = 0.0; }
return Fcal;
}
/* - global settings - */
void PairGranludingHistory::settings(int narg, char **arg) { iarg_ = 0;
// set defaults dampflag = 1; rollingflag = 0; cohesionflag = 0; viscousflag = 0; force_off
= false;
// parse args printf("roll, narg: %d %d \n", rollingflag, narg); bool hasargs = true;
while(iarg_ < narg && hasargs) {
hasargs = false;
if (strcmp(arg[iarg_],"cohesion") == 0) {
if (narg < iarg_+2) error->all(FLERR,"Pair gran: not enough arguments for 'cohesion'");
iarg_++; if(strcmp(arg[iarg_],"sjkr") == 0) cohesionflag = 1;
else if(strcmp(arg[iarg_],"sjkr2") == 0) cohesionflag = 2;
else if(strcmp(arg[iarg_],"off") == 0) cohesionflag = 0;
else
```

error->all(FLERR,"Illegal pair_style gran command, expecting 'sjkr' or 'off' after keyword 'cohesion'");

iarg_++; hasargs = true; }

else if (strcmp(arg[iarg_],"force") == 0) {

if (narg < iarg_+2)

error->all(FLERR,"Pair gran: not enough arguments for 'force'");

iarg_++; if(strcmp(arg[iarg_],"on") == 0) force_off = false;

else if(strcmp(arg[iarg_],"off") == 0) force_off = true;

else error->all(FLERR,"Illegal pair_style gran command, expecting 'on' or 'off' after keyword 'force'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"sanity_checks") == 0) {

if (narg < iarg_+2)

error->all(FLERR,"Pair gran: not enough arguments for 'sanity_checks'");

iarg_++; if(strcmp(arg[iarg_],"on") == 0) sanity_checks = true;

else if(strcmp(arg[iarg_],"off") == 0) sanity_checks = false;

else error->all(FLERR, "Illegal pair_style gran command, expecting 'on' or 'off' after keyword 'sanity_checks'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"rolling_friction") == 0) {

if (narg < iarg_+2)

error->all(FLERR,"Pair gran: not enough arguments for 'rolling_friction'"); iarg_++;

if(strcmp(arg[iarg_],"cdt") == 0) rollingflag = 1;

else if(strcmp(arg[iarg_],"epsd") == 0) {

rollingflag = 2; // dnum_pairgran = 6; dnum_pairgran = 7; }

else if(strcmp(arg[iarg_],"epsd2") == 0) { rollingflag = 3; dnum_pairgran = 7; }

else if(strcmp(arg[iarg_],"off") == 0) rollingflag = 0;

else

error->all(FLERR,"Illegal pair_style gran command, expecting 'cdt', 'epsd', 'epsd2' or 'off' after keyword 'rolling_friction'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"tangential_damping") == 0) {

if (narg < iarg_+2)

error->all(FLERR,"Pair gran: not enough arguments for 'tangential_damping'");

iarg_++; if(strcmp(arg[iarg_],"on") == 0) dampflag = 1;

else if(strcmp(arg[iarg_],"off") == 0) dampflag = 0;

else error->all(FLERR,"Illegal pair_style gran command, expecting 'on' or 'off' after keyword 'tangential_damping'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"viscous") == 0) {

if (narg < iarg_+2) error->all(FLERR,"Pair gran: not enough arguments for 'viscous'");

iarg_++; if(strcmp(arg[iarg_],"stokes") == 0) viscousflag = 1;

else if(strcmp(arg[iarg_],"off") == 0) viscousflag = 0;

else error->all(FLERR,"Illegal pair_style gran command, expecting 'stokes' or 'off' after keyword 'viscous'");

iarg_++; hasargs = true; }

else if (force-> pair_match("gran/luding/history",1) —— force-> pair_match("gran/hertz/history", 1))

error->all(FLERR,"Illegal pair_style gran command, illegal keyword"); }

printf("roll: %d \n", rollingflag); if(cohesionflag && domain->dimension!=3)

error->all(FLERR,"Cohesion model valid for 3d simulations only"); }

/* ——— init specific to this granular substyle ——- */

void PairGranludingHistory::init_granular() { int max_type = mpg->max_type();

allocate_properties(max_type);

//Get pointer to the fixes that have the material properties

kload1= static_cast <FixPropertyGlobal*> (modify-> find_fix_property ("LoadingStiffness", "property/global", "peratomtype",max_type,0,force->pair_style));

kunload1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“UnloadingStiffness”, ”property/global”, ”peratomtype”, max_type,0,force->pair_style));

kcoh1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“CohesiveStiffness”, ”property/global”, ”peratomtype”,max_type,0,force->pair_style));

Y1= static_cast <FixPropertyGlobal*> (modify-> find_fix_property (“youngsModulus”, ”property/global”, ”peratomtype”, max_type,0,force->pair_style));

v1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“poissonsRatio”, ”property/global”, ”peratomtype”,max_type,0,force->pair_style));

nPower1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“NPower”, ”property/global”, ”scalar”,0,0,force->pair_style));

coeffRest1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“coefficientRestitution”, ”property/global”, ”peratomtypepair”, max_type,max_type,force->pair_style));

coeffFrict1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“coefficientFriction”, ”property/global”, ”peratomtypepair”, max_type,max_type,force->pair_style));

if(rollingflag)

coeffRollFrict1= static_cast <FixPropertyGlobal*> (modify-> find_fix_property (“coefficientRollingFriction”, ”property/global”, ”peratomtypepair”, max_type, max_type, force->pair_style));

if(rollingflag == 2)

// damping for original epsd model only

coeffRollVisc1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“coefficientRollingViscousDamping”, ”property/global”, ”peratomtypepair”, max_type, max_type, force->pair_style));

if(viscousflag) {

coeffMu1= static_cast<FixPropertyGlobal*> (modify-> find_fix_property (“FluidViscosity”, ”property/global”, ”peratomtypepair”, max_type, max_type, force->pair_style));

coeffRestMax1= static_cast <FixPropertyGlobal*> (modify->find_fix_property (“MaximumRestitution”, ”property/global”, ”peratomtypepair”, max_type, max_type, force->pair_style));

coeffStc1=static_cast <FixPropertyGlobal*> (modify->find_fix_property (“CriticalStokes”, ”property/global”, ”peratomtypepair”, max_type, max_type,force->pair_style)); }

if(cohesionflag)

cohEnergyDens1= static_cast <FixPropertyGlobal*> (modify-> find_fix_property (“cohesionEnergyDensity”, ”property/global”, ”peratomtypepair”, max_type, max_type, force->pair_style));

if(charVelflag) charVel1=static_cast <FixPropertyGlobal*> (modify->find_fix_property (“characteristicVelocity”, ”property/global”, ”scalar”, 0, 0,force-> pair_style));

//pre-calculate parameters for possible contact material combinations

for(int i=1;i< max_type+1; i++) { for(int j=1;j<max_type+1;j++) {

double kloadi=kload1->compute_vector(i-1);

double kloadj=kload1->compute_vector(j-1);

double kunloadi=kunload1->compute_vector(i-1);

double kunloadj=kunload1->compute_vector(j-1);

double kcohi=kcoh1->compute_vector(i-1);

double kcohj=kcoh1->compute_vector(j-1);

double Yi=Y1->compute_vector(i-1);

double Yj=Y1->compute_vector(j-1);

double vi=v1->compute_vector(i-1); double vj=v1->compute_vector(j-1);

double cor = coeffRest1->compute_array(i-1,j-1);

// error checks on Y, v, e nPower= nPower1->compute_scalar();

if(sanity_checks) { if(strcmp(update->unit_style,”si”) == 0 && Yi < 5e6)

error->all(FLERR,”youngsModulus >= 5e6 required for SI units”);

if(strcmp(update->unit_style,”cgs”) == 0 && Yi < 5e5)

error->all(FLERR,”youngsModulus >= 5e5 required for CGS units”);

if(vi < 0. —— vi > 0.5)

error->all(FLERR,"0 <= poissonsRatio <= 0.5 required");

if(cor <= 0.05 —— cor > 1) error-> all(FLERR,"0.05 < coefficientRestitution <= 1 required"); }

Yeff[i][j] = 1./((1.-pow(vi,2.))/ Yi+(1.-pow(vj,2.))/Yj);

Geff[i][j] = 1./(2.*(2.-vi)*(1.+vi)/ Yi+2.*(2.-vj)*(1.+vj)/Yj);

kload[i][j]=kloadi*kloadj/ (kloadi+kloadj);

kunload[i][j]=kunloadi*kunloadj/ (kunloadi+kunloadj);

kcoh[i][j]=kcohi*kcohj/(kcohi+kcohj); coeffRestLog[i][j] = log(coeffRest1->compute_array(i-1,j-1));

if(viscousflag) { coeffMu[i][j] = coeffMu1->compute_array(i-1,j-1);

coeffRestMax[i][j] = coeffRestMax1->compute_array(i-1,j-1);

coeffStc[i][j] = coeffStc1->compute_array(i-1,j-1);

// error check if(sanity_checks) {

if(coeffRestMax[i][j] <= 0. —— coeffRestMax[i][j] > 1)

error->all(FLERR,"0 < MaximumRestitution <= 1 required");

if(coeffMu[i][j] <= 0.) error->all(FLERR,"coeffMu > 0 required");

if(coeffStc[i][j] <= 0.) error->all(FLERR,"CriticalStokes > 0 required"); } }

betaeff[i][j] =coeffRestLog[i][j] /sqrt(pow(coeffRestLog[i][j],2.)+pow(M_PI,2.));

coeffFrict[i][j] = coeffFrict1->compute_array(i-1,j-1);

if(rollingflag) coeffRollFrict[i][j] = coeffRollFrict1->compute_array(i-1,j-1);

if(rollingflag == 2) coeffRollVisc[i][j] = coeffRollVisc1->compute_array(i-1,j-1);

if(cohesionflag) cohEnergyDens[i][j] = cohEnergyDens1->compute_array(i-1,j-1); //omitting veff here

} }

if(charVelflag) {

charVel = charVel1->compute_scalar(); if(sanity_checks) {

```
if(strcmp(update->unit_style,"si") == 0 && charVel < 1e-2)
error->all(FLERR,"characteristicVelocity >= 1e-2 required for SI units"); } }
// error checks on coarsegraining if((rollingflag —— cohesionflag) && force->cg_active())
error->cg(FLERR,"Granular model with rolling friction and / or cohesion");
// error checks on coarsegraining if((rollingflag —— cohesionflag) && force->cg_active())
error->cg(FLERR,"Granular model with rolling friction and / or cohesion"); }
/* ———————— allocate per-type and per-type pair properties ————————- */
void PairGranludingHistory::allocate_properties(int size) {
memory->destroy(kload);
memory->destroy(kunload);
memory->destroy(kcoh);
memory->destroy(Yeff);
memory->destroy(Geff);
memory->destroy(betaeff);
memory->destroy(veff);
memory->destroy(cohEnergyDens);
memory->destroy(coeffRestLog);
memory->destroy(coeffFrict);
memory->destroy(coeffRollFrict);
memory->destroy(coeffRollVisc);
memory->destroy(coeffMu);
memory->destroy(coeffRestMax);
memory->destroy(coeffStc);
memory->create(kload,size+1,size+1,"kload");
memory->create(kunload,size+1,size+1,"kunload");
memory->create(kcoh,size+1,size+1,"kcoh");
memory->create(Yeff,size+1,size+1,"Yeff");
```

```
memory->create(Geff,size+1,size+1,"Geff");
memory->create(betaeff,size+1,size+1,"betaeff");
memory->create(veff,size+1,size+1,"veff");
memory->create(cohEnergyDens,size+1,size+1,"cohEnergyDens");
memory->create(coeffRestLog,size+1,size+1,"coeffRestLog");
memory->create(coeffFrict,size+1,size+1,"coeffFrict");
memory->create(coeffRollFrict,size+1,size+1,"coeffRollFrict");
memory->create(coeffRollVisc,size+1,size+1,"coeffRollVisc");
memory->create(coeffMu,size+1,size+1,"coeffMu");
memory->create(coeffRestMax,size+1,size+1,"coeffRestMax");
memory->create(coeffStc,size+1,size+1,"coeffStc");
}
/* ----------------------------------------------------------- proc 0 writes to restart file ----------------------------------------------------------- */
void PairGranludingHistory::write_restart_settings(FILE *fp) {
int writeflag = dampflag + rollingflag * 2;
fwrite(&writeflag,sizeof(int),1,fp);
fwrite(&cohesionflag,sizeof(int),1,fp);
fwrite(&viscousflag,sizeof(int),1,fp); }
/* ---------- proc 0 reads from restart file, bcasts ---------- */
void PairGranludingHistory::read_restart_settings(FILE *fp) {
if (comm->me == 0) {
int readflag;
fread(&readflag,sizeof(int),1,fp);
fread(&cohesionflag,sizeof(int),1,fp);
fread(&viscousflag,sizeof(int),1,fp);
dampflag = readflag & 1;
```

rollingflag = readflag & 2; }

MPI_Bcast(&dampflag,1,MPI_INT,0,world);

MPI_Bcast(&cohesionflag,1,MPI_INT,0,world);

MPI_Bcast(&rollingflag,1,MPI_INT,0,world);

MPI_Bcast(&viscousflag,1,MPI_INT,0,world); }

C.4 Particle-Wall Contact C++ Code

/* —

LIGGGHTS - LAMMPS Improved for General Granular and Granular Heat Transfer Simulations

LIGGGHTS is part of the CFDEMproject www.liggghts.com — www.cfdem.com This file was modified with respect to the release in LAMMPS Modifications are Copyright 2009-2012 JKU Linz Copyright 2012- DCS Computing GmbH, Linz LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator http://lammps.sandia.gov, Sandia National Laboratories Steve Plimpton, sjplimp@sandia.gov Copyright (2003) Sandia Corporation. Under the terms of Contract DE-AC04-94AL85000 with Sandia Corporation, the U.S. Government retains certain rights in this software. This software is distributed under the GNU General Public License.

See the README file in the top-level directory. - */

/* - Contributing authors for original version: Leo Silbert (SNL), Gary Grest (SNL) — */

#include “math.h”

#include “stdio.h”

#include “stdlib.h”

#include “string.h”

#include “fix_wall_gran_luding_history.h”

#include “pair_gran_luding_history.h”

#include “atom.h”

#include “force.h”

#include “update.h”

#include “pair.h”

#include “modify.h”

#include “memory.h”

#include “error.h”

#include “fix_property_global.h”

#include “compute_pair_gran_local.h”

#include “fix_property_atom.h”

#include “mech_param_gran.h”

#include “fix_rigid.h”

#include “vector_liggghts.h”

#include “fix_mesh.h”

#include “container.h”

#include <iostream> //for cin and cout using namespace LAMMPS_NS; using namespace FixConst;

#define MIN(A,B) (((A) < (B)) ? (A) : (B))

#define MAX(A,B) (((A) > (B)) ? (A) : (B))

#define SMALL 1e-12

/* ———————————————————————- */

FixWallGranludingHistory:: FixWallGranludingHistory (LAMMPS *lmp, int narg, char **arg) : FixWallGran (lmp, narg, arg) { // parse wall models

// set defaults Temp_wall = -1.;

Q = Q_add = 0.; dampflag = 1;

cohesionflag = 0; rollingflag = 0; viscousflag= 0;

bool hasargs = true; while(iarg_ < narg && hasargs)

{ hasargs = false; if (strcmp(arg[iarg_],”cohesion”) == 0)

{ iarg_++; if(strcmp(arg[iarg_],"sjkr") == 0) cohesionflag = 1;

else if(strcmp(arg[iarg_],"sjkr2") == 0) cohesionflag = 2;

else if(strcmp(arg[iarg_],"off") == 0) cohesionflag = 0;

else error->fix_error(FLERR,this,"expecting 'sjkr' or 'off' after keyword 'cohesion'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"rolling_friction") == 0)

{ iarg_++; if(strcmp(arg[iarg_],"cdt") == 0) rollingflag = 1; else if(strcmp(arg[iarg_],"epsd") == 0)

rollingflag = 2; else if(strcmp(arg[iarg_],"epsd2") == 0)

rollingflag = 3; else if(strcmp(arg[iarg_],"off") == 0)

rollingflag = 0;

else error->fix_error(FLERR,this,"expecting 'cdt', 'epsd', 'epsd2' or 'off' after keyword 'rolling_friction'");

iarg_++; hasargs = true; }

else if (strcmp(arg[iarg_],"viscous") == 0)

{ if (narg < iarg_+2)

error->all(FLERR,"Pair gran: not enough arguments for 'viscous'");

iarg_++; if(strcmp(arg[iarg_],"stokes") == 0)

viscousflag = 1; else if(strcmp(arg[iarg_],"off") == 0)

viscousflag = 0; else

error->all(FLERR,"Illegal pair_style gran command, expecting 'stokes' or 'off' after keyword 'viscous'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"tangential_damping") == 0)

{ iarg_++; if(strcmp(arg[iarg_],"on") == 0) dampflag = 1;

else if(strcmp(arg[iarg_],"off") == 0) dampflag = 0;

else error->fix_error(FLERR,this,"expecting 'on' or 'off' after keyword 'dampflag'");

iarg_++; hasargs = true; } else if (strcmp(arg[iarg_],"temperature") == 0)

{ if(is_mesh_wall())

```
error->fix_error (FLERR,this, "for mesh walls temperature has to be defined for each mesh via fix mesh");
iarg_++; Temp_wall = atof(arg[iarg_++]); hasargs = true; } }
if (cohesionflag < 0 —— cohesionflag > 1 —— dampflag < 0 —— dampflag > 3)
error->fix_error(FLERR,this," "); }
/* ———————————————————————————————————- */
FixWallGranludingHistory::~FixWallGranludingHistory() {
}
/* ———————————————————————————————————- */
void FixWallGranludingHistory::post_create() { FixWallGran::post_create(); }
/* ———————————————————————————————————- */
void FixWallGranludingHistory::init_granular()
{ //get material properties
kload = ((PairGranludingHistory*)pairgran_)->kload;
kunload = ((PairGranludingHistory*)pairgran_)->kunload;
kcoh = ((PairGranludingHistory*)pairgran_)->kcoh;
Yeff = ((PairGranludingHistory*)pairgran_)->Yeff;
Geff = ((PairGranludingHistory*)pairgran_)->Geff;
betaeff = ((PairGranludingHistory*)pairgran_)->betaeff;
veff = ((PairGranludingHistory*)pairgran_)->veff;
cohEnergyDens = ((PairGranludingHistory*)pairgran_)->cohEnergyDens;
coeffRestLog = ((PairGranludingHistory*)pairgran_)->coeffRestLog;
coeffFrict = ((PairGranludingHistory*)pairgran_)->coeffFrict;
coeffRollFrict = ((PairGranludingHistory*)pairgran_)->coeffRollFrict;
coeffRollVisc = ((PairGranludingHistory*)pairgran_)->coeffRollVisc;
coeffMu = ((PairGranludingHistory*)pairgran_)->coeffMu;
coeffRestMax = ((PairGranludingHistory*)pairgran_)->coeffRestMax;
```

```
coeffStc = ((PairGranludingHistory*)pairgran_)->coeffStc;
nPower = ((PairGranludingHistory*)pairgran_)->nPower;
charVel = ((PairGranludingHistory*)pairgran_)->charVel;
// need to check properties for rolling friction and cohesion energy density here
// since these models may not be active in the pair style
int max_type = pairgran_->mpg->max_type();
FixPropertyGlobal *coeffRollFrict1, *cohEnergyDens1, *coeffMu1, *coeffRestMax1, *coeffStc1, *coeffRollVisc1;
if(rollingflag)
coeffRollFrict1=static_cast <FixPropertyGlobal*> (modify->find_fix_property
("coefficientRollingFriction", "property/global", "peratomtypepair", max_type, max_type,style));
if(rollingflag == 2)
// epsd model
coeffRollVisc1= static_cast <FixPropertyGlobal*>
(modify->
find_fix_property ("coefficientRollingViscousDamping", "property/global", "peratomtypepair", max_type, max_type, force->pair_style));
if(cohesionflag)
cohEnergyDens1= static_cast <FixPropertyGlobal*>
(modify-> find_fix_property ("cohesionEnergyDensity", "property/global","peratomtypepair", max_type, max_type,style));
if(viscousflag)
{ coeffMu1=static_cast <FixPropertyGlobal*>
(modify->find_fix_property ("FluidViscosity","property/global", "peratomtypepair", max_type, max_type, force-> pair_style));
coeffRestMax1=static_cast <FixPropertyGlobal*>
```

(modify-> find_fix_property (“MaximumRestitution”, ”property/global”, ”peratomtype-pair”, max_type, max_type, force->pair_style));

coeffStc1= static_cast <FixPropertyGlobal*>

(modify-> find_fix_property (“CriticalStokes”,”property/global”, ”peratomtypepair”, max_type, max_type, force-> pair_style)); }

//pre-calculate parameters for possible contact material combinations

for(int i=1;i < max_type+1; i++)

{ for(int j=1;j<max_type+1;j++) { if(rollingflag)

coeffRollFrict[i][j] = coeffRollFrict1->compute_array(i-1,j-1);

if(rollingflag == 2)

coeffRollVisc[i][j] = coeffRollVisc1->compute_array(i-1,j-1);

if(cohesionflag)

cohEnergyDens[i][j] = cohEnergyDens1->compute_array(i-1,j-1);

if(viscousflag) { coeffMu[i][j] = coeffMu1->compute_array(i-1,j-1);

coeffRestMax[i][j] = coeffRestMax1->compute_array(i-1,j-1);

coeffStc[i][j] = coeffStc1->compute_array(i-1,j-1); } } }

if(cohesionflag)

error->warning(FLERR,”Cohesion model should only be used with hertzian contact laws.”); }

/* ———————————————————————- */

void FixWallGranludingHistory::init_heattransfer()

{ fppa_T = NULL; fppa_hf = NULL; deltan_ratio = NULL;

if (!is_mesh_wall() && Temp_wall < 0.) return;

else if (is_mesh_wall())

{ int heatflag = 0; for(int imesh = 0; imesh < n_meshes(); imesh++)

{ heatflag = heatflag —— mesh_list()[imesh]->mesh()->prop().getGlobalProperty<

ScalarContainer<double> >(“Temp”) != NULL; }

```
if(!heatflag) return; }
// set flag so addHeatFlux function is called heattransfer_flag_ = true;
// if(screen && comm->me == 0)
fprintf(screen,"Initializing wall/gran heat transfer model\n");
fppa_T = static_cast<FixPropertyAtom*>
(modify->find_fix_property("Temp","property/atom","scalar",1,0,style));
fppa_hf = static_cast<FixPropertyAtom*>
(modify->find_fix_property("heatFlux","property/atom","scalar",1,0,style));
th_cond = static_cast <FixPropertyGlobal*>
(modify-> find_fix_property ("thermalConductivity", "property/global", "peratomtype",0,
0,style))-> get_values();
// if youngsModulusOriginal defined, get deltan_ratio
Fix* ymo_fix = modify->
find_fix_property ("youngsModulusOriginal", "property/global", "peratomtype", 0, 0,
style,false);
// deltan_ratio is defined by heat transfer fix, see if there is one
int n_htf = modify-> n_fixes_style("heat/gran/conduction");
// get deltan_ratio set by the heat transfer fix if(ymo_fix && n_htf)
deltan_ratio = static_cast <FixPropertyGlobal*> (ymo_fix)->get_array_modified(); }
/* ---------------------------------------------------------- */
void FixWallGranludingHistory::compute_force(int ip, double deltan, double rsq,double
meff_wall, double dx, double dy, double dz,double *vwall, double *c_history, double
area_ratio)
{ double r,vr1,vr2,vr3,vnnr,vn1,vn2,vn3,vt1,vt2,vt3;
double wr1,wr2,wr3,damp,ccel,vtr1,vtr2,vtr3,vrel;
double fn,fs,fs1,fs2,fs3,fx,fy,fz,tor1,tor2,tor3,r_torque[3];
double shrmag,rsht,rinv,rsqinv;
```

```
double kn, kun, kad, kt, gamman, gammat, xmu, rmu; double cri, crj;
double dmax, Fmax, Fmin, d1max, dcmax;
double *f = atom->f[ip]; double *torque = atom->torque[ip];
ouble *v = atom->v[ip]; double *omega = atom->omega[ip];
double radius = atom->radius[ip]; double mass = atom->rmass[ip];
double cr = radius - 0.5*deltan; double dt = update->dt;
double DT = 0.5*dt; double dp = 0.0; double up = c_history[3];
if(fix_rigid_ && body_[ip] >= 0) mass = masstotal_[body_[ip]];
r = sqrt(rsq); rinv = 1.0/r; rsqinv = 1.0/rsq;
// relative translational velocity
vr1 = v[0] - vwall[0]; vr2 = v[1] - vwall[1]; vr3 = v[2] - vwall[2];
// normal component
vnnr = vr1*dx + vr2*dy + vr3*dz; vn1 = dx*vnnr * rsqinv;
vn2 = dy*vnnr * rsqinv; vn3 = dz*vnnr * rsqinv;
// tangential component
vt1 = vr1 - vn1; vt2 = vr2 - vn2; vt3 = vr3 - vn3;
// relative rotational velocity // in case of wall contact, r is the contact radius
wr1 = cr*omega[0] * rinv; wr2 = cr*omega[1] * rinv;
wr3 = cr*omega[2] * rinv;
//get the parameters needed to resolve the contact
deriveContactModelParams(ip, deltan, meff_wall, kn, kun, kad, kt, gamman, gammat,
xmu, rmu, vnnr);
deriveContactLudingParams(ip, dmax, Fmax, Fmin, d1max, dcmax);
// normal forces = Hookian contact + normal velocity damping
damp = gamman*vnnr*rsqinv; double ddot =-1.0* vnnr;
//printf("c_history: %E \n", c_history[3]); if (up < 0.0) { up=-up;}
ccel = LudingForce(ddot, DT, deltan, dp, up, kn, kun, kad, dmax, d1max)*rinv-damp;
```

```
c_history[3] = dp;
//printf("c_history: %E \n", c_history[3]);
if(cohesionflag) {
double Fn_coh; addCohesionForce(ip, r, Fn_coh,area_ratio);
ccel-=Fn_coh*rinv; }
// relative velocities
vtr1 = vt1 - (dz*wr2-dy*wr3); vtr2 = vt2 - (dx*wr3-dz*wr1);
vtr3 = vt3 - (dy*wr1-dx*wr2); vrel = vtr1*vtr1 + vtr2*vtr2 + vtr3*vtr3;
vrel = sqrt(vrel);
// shear history effects if (shearupdate_ && computeflag_)
{ c_history[3] = dp; c_history[0] += vtr1*dt_;
c_history[1] += vtr2*dt_; c_history[2] += vtr3*dt_;
// rotate shear displacements
rsht = c_history[0]*dx + c_history[1]*dy + c_history[2]*dz;
rsht = rsht*rsqinv; c_history[0] -= rsht*dx;
c_history[1] -= rsht*dy; c_history[2] -= rsht*dz; }
shrmag = sqrt(c_history[0]*c_history[0] + c_history[1]*
c_history[1] + c_history[2]*c_history[2]);
// tangential forces = shear + tangential velocity damping
fs1 = - (kt*c_history[0]);
fs2 = - (kt*c_history[1]); fs3 = - (kt*c_history[2]);
// rescale frictional displacements and forces if needed
fs = sqrt(fs1*fs1 + fs2*fs2 + fs3*fs3);
fn = xmu * fabs(ccel*r);
// energy loss from sliding or damping
if (fs > fn) { if (shrmag != 0.0)
{ fs1 *= fn/fs; fs2 *= fn/fs; fs3 *= fn/fs;
```

```
c_history[0]=-fs1/kt; c_history[1]=-fs2/kt; c_history[2]=-fs3/kt; }
else fs1 = fs2 = fs3 = 0.0; } else { fs1 -= (gammat*vtr1);
fs2 -= (gammat*vtr2); fs3 -= (gammat*vtr3); }
// forces & torques
fx = dx*ccel + fs1;
fy = dy*ccel + fs2; fz = dz*ccel + fs3;
if(computeflag_) { f[0] += fx*area_ratio;
f[1] += fy*area_ratio; f[2] += fz*area_ratio; }
tor1 = rinv * (dy*fs3 - dz*fs2);
tor2 = rinv * (dz*fs1 - dx*fs3); tor3 = rinv * (dx*fs2 - dy*fs1);
// add rolling friction torque vectorZeroize3D(r_torque); switch (rollingflag)
{ case 0: break; case 1:
addRollingFrictionTorque <1> (ip, wr1, wr2, wr3, cr, ccel, r, mass, rmu, kun, kt, dx, dy,
dz, rsqinv, c_history, r_torque);
break; case 2:
addRollingFrictionTorque <2> (ip, wr1, wr2, wr3, cr, ccel , r, mass, rmu, kun, kt, dx,
dy, dz, rsqinv, c_history, r_torque);
break; case 3:
addRollingFrictionTorque <3> (ip, wr1, wr2, wr3, cr, ccel, r , mass, rmu, kun, kt, dx,
dy, dz, rsqinv, c_history ,r_torque);
break; }
if(computeflag_) {
torque[0] -= cr*tor1*area_ratio + r_torque[0];
torque[1] -= cr*tor2*area_ratio + r_torque[1];
torque[2] -= cr*tor3*area_ratio + r_torque[2]; }
if(cwl_ && addflag_)
cwl_->add_wall_2(ip,fx,fy,fz,tor1*area_ratio,tor2*area_ratio,tor3*area_ratio,c_history,rsq);
```

```
}
/* ---------------------------------------------------------- */
void FixWallGranludingHistory::addHeatFlux(TriMesh *mesh,int ip, double delta_n, double area_ratio)
{ //r is the distance between the sphere center and wall double tcop, tcowall, hc, Acont, r;
double reff_wall = atom->radius[ip];
int itype = atom->type[ip]; double ri = atom->radius[ip];
if(mesh) Temp_wall = (*mesh->prop().getGlobalProperty< ScalarContainer<double> >("Temp"))(0);
double *Temp_p = fppa_T->vector_atom;
double *heatflux = fppa_hf->vector_atom;
if(deltan_ratio) delta_n *= deltan_ratio[itype-1][atom_type_wall_-1];
r = ri + delta_n;
Acont = (reff_wall*reff_wall-r*r)*M_PI*area_ratio;
//contact area sphere-wall tcop = th_cond[itype-1];
//types start at 1, array at 0 tcowall = th_cond[atom_type_wall_-1];
if ((fabs(tcop) < SMALL) -- (fabs(tcowall) < SMALL)) hc = 0.;
else hc = 4.*tcop*tcowall/(tcop+tcowall)*sqrt(Acont);
if(computeflag_) { heatflux[ip] += (Temp_wall-Temp_p[ip]) * hc;
Q_add += (Temp_wall-Temp_p[ip]) * hc * update->dt; }
if(cwl_ && addflag_) cwl_->add_heat_wall(ip,(Temp_wall-Temp_p[ip]) * hc); }
/* ---------------------------------------------------------- */
inline void FixWallGranludingHistory::
addCohesionForce(int &ip, double &r, double &Fn_coh,double area_ratio)
{ //r is the distance between the sphere center and wall
double reff_wall = atom->radius[ip];
```

```
double Acont; if(cohesionflag == 1)
Acont = (reff_wall*reff_wall-r*r)*M_PI;
//contact area sphere-wall else Acont = M_PI * 2. * reff_wall * (reff_wall - r);
int itype = atom->type[ip];
Fn_coh=cohEnergyDens[itype][atom_type_wall_]*Acont*area_ratio; }
/* ---------------------------------------------------------- */
template <int ROLLINGFRICTION> void FixWallGranludingHistory::
addRollingFrictionTorque(int ip, double wr1, double wr2, double wr3, double cr, double
ccel, double r,
double mi,double rmu,double kun,double kt,double dx, double dy,
double dz,double rsqinv,double *c_history,double *r_torque)
{ double wrmag,r_torque_n[3]; double radius = atom->radius[ip];
if (ROLLINGFRICTION == 1) {
wrmag = sqrt(wr1*wr1+wr2*wr2+wr3*wr3); if (wrmag > 0.)
{ r_torque[0] = rmu*kun*(radius-r)*wr1/wrmag*cr;
r_torque[1] = rmu*kun*(radius-r)*wr2/wrmag*cr;
r_torque[2] = rmu*kun*(radius-r)*wr3/wrmag*cr;
// remove normal (torsion) part of torque
double rtorque_dot_delta = r_torque[0]*dx+ r_torque[1]*dy + r_torque[2]*dz;
r_torque_n[0] = dx * rtorque_dot_delta * rsqinv;
r_torque_n[1] = dy * rtorque_dot_delta * rsqinv;
r_torque_n[2] = dz * rtorque_dot_delta * rsqinv;
vectorSubtract3D(r_torque,r_torque_n,r_torque); }
} else { double kr,r_inertia,r_coef,r_torque_mag,r_torque_max,factor;
double dr_torque[3],wr_n[3],wr_t[3];
int itype = atom->type[ip]; double dt = update->dt;
// remove normal (torsion) part of relative rotation
```

```
// use only tangential parts for rolling torque
double wr_dot_delta = wr1*dx+ wr2*dy + wr3*dz;
wr_n[0] = dx * wr_dot_delta * rsqinv;
wr_n[1] = dy * wr_dot_delta * rsqinv;
wr_n[2] = dz * wr_dot_delta * rsqinv;
wr_t[0] = wr1 - wr_n[0];
wr_t[1] = wr2 - wr_n[1];
wr_t[2] = wr3 - wr_n[2];
// spring if (ROLLINGFRICTION == 2)
kr = 2.25*kun*rmu*rmu*radius*radius;
else kr = kt*radius*radius;
dr_torque[0] = kr * wr_t[0] * dt;
dr_torque[1] = kr * wr_t[1] * dt;
dr_torque[2] = kr * wr_t[2] * dt;
r_torque[0] = c_history[4] + dr_torque[0];
r_torque[1] = c_history[5] + dr_torque[1];
r_torque[2] = c_history[6] + dr_torque[2];
// limit max.
torque r_torque_mag = vectorMag3D(r_torque);
r_torque_max = fabs(ccel*r)*radius*rmu;
if(r_torque_mag > r_torque_max) {
factor = r_torque_max / r_torque_mag;
r_torque[0] *= factor;
r_torque[1] *= factor;
r_torque[2] *= factor;
// save rolling torque due to spring
c_history[4] = r_torque[0];
```

```
c_history[5] = r_torque[1];
c_history[6] = r_torque[2];
// no damping / no dashpot in case of full mobilisation rolling angle //
r_coef = 0.0;
} else {
// save rolling torque due to spring before adding damping torque
c_history[4] = r_torque[0];
c_history[5] = r_torque[1];
c_history[6] = r_torque[2];
// dashpot only for the original epsd model
if(ROLLINGFRICTION == 2)
{ // dashpot if (domain->dimension == 2)
r_inertia = 1.5*mi*radius*radius;
else r_inertia = 1.4*mi*radius*radius;
r_coef = coeffRollVisc[itype][atom_type_wall_] * 2 * sqrt(r_inertia*kr);
// add damping torque
r_torque[0] += r_coef*wr_t[0];
r_torque[1] += r_coef*wr_t[1];
r_torque[2] += r_coef*wr_t[2]; } } }
}
/* ---------------------------------------------------------- */
inline void FixWallGranludingHistory::
deriveContactModelParams(int ip, double deltan,double meff_wall, double &kn,
double &kun, double &kad, double &kt, double &gamman, double &gammat,
double &xmu,double &rmu,double &vnnr) { double sqrtval = sqrt(atom->radius[ip]);
int itype = atom->type[ip]; double stokes, coeffRestLogChosen;
if (viscousflag) { double reff=atom->radius[ip];
```

```
stokes=meff_wall*vnnr/(6.0*3.1416*coeffMu[itype][atom_type_wall_]*reff*reff);
//Stokes Number from MW Schmeeckle (2001)
coeffRestLogChosen = log ( coeffRestMax[itype][atom_type_wall_] ) + coeffStc[itype][atom_type_wall_]
//stokes;
// Empirical from Legendre (2006) }
else { coeffRestLogChosen = coeffRestLog[itype][atom_type_wall_]; }
kn=kload[itype][atom_type_wall_];
kun=kunload[itype][atom_type_wall_];
kad=kcoh[itype][atom_type_wall_];
kt=kn;
gamman = 0.1 * sqrt ( 4. * meff_wall * kun / ( 1. + ( M_PI / coeffRestLogChosen ) * (
M_PI / coeffRestLogChosen)));
gammat=gamman;
xmu=coeffFrict[itype][atom_type_wall_];
if(rollingflag)rmu=coeffRollFrict[itype][atom_type_wall_];
if (dampflag == 0) gammat = 0.0;
// convert Kn and Kt from pressure units to force/distance^2
/* kn /= force->nktv2p; kun /= force->nktv2p; kt /= force->nktv2p;*/
return; } /* ---------------------------------------------------------- */
inline void FixWallGranludingHistory::deriveContactLudingParams(int
ip, double &dmax, double &Fmax, double &Fmin, double &d1max, double &dcmax)
{ int itype = atom->type[ip]; double reff=atom->radius[ip];
double tempmax; double tempinvr = 1.0/nPower;
dmax= reff*0.1;
//max penetration is 10% of the effective radius
double temppower = pow(dmax,nPower);
Fmax = (kload[itype][atom_type_wall_]*kunload[itype][atom_type_wall_])/
```

```
(kunload[itype][atom_type_wall_]-kload[itype][atom_type_wall_])*temppower;
Fmin=-kcoh[itype][atom_type_wall_]*kunload[itype][atom_type_wall_]/
(kunload[itype][atom_type_wall_]+kcoh[itype][atom_type_wall_])*temppower;
tempmax = (kunload[itype][atom_type_wall_]/
(kunload[itype][atom_type_wall_]-kload[itype][atom_type_wall_]));
d1max=pow(tempmax,tempinvr)*dmax;
tempmax =(kunload[itype][atom_type_wall_]/
(kcoh[itype][atom_type_wall_]+kunload[itype][atom_type_wall_]));
dcmax=pow(tempmax, tempinvr)*dmax;
return; } /* ----------------------------------------------- */
double FixWallGranludingHistory::LudingForce(double &ddot, double &DT,
double &u, double &dp, double &up, double &kn, double &kun, double &kad,
double &umax, double &u1max)
//(double { double changeu = ddot*DT;
double unew = u+changeu; double Fcal, F2n, F1n, Fcn; double dpnum; double NPinv
= 1.0/nPower;
if (unew < 0.0 ){ dp = 0.0; Fcal = 0.0; return Fcal; }
F2n = kun*(pow(unew,nPower)-pow(up,nPower));
//printf("ddot %E \n", ddot);
if (ddot >= 0.0){ F1n = kn*pow(unew,nPower);
if ((F2n-F1n)>0.0){
dpnum = (kun-kn)/kun;
dp = pow(dpnum,NPinv)*unew;
if (dp > umax){ dp = umax; }
Fcal = F1n; if (unew > u1max){
dp = umax; Fcal = kun*(pow(unew,nPower)-pow(umax,nPower)); }
// printf("Load unew, ddot, dp: %E %E %E \n", unew, ddot, dp);
```

```
}else{ dp = up; if (dp > umax){
dp = umax; } Fcal = F2n; }
}else{ Fcn = -1.0*kad*pow(unew,nPower);
//printf("unLoad F2n, Fc: %E %E\n", F2n, Fcn);
if ((F2n-Fcn)<0.0)
{ dpnum = (kun+kad)/kun; dp = pow(dpnum,NPinv)*unew;
//printf("unLoad kun, kad, dpnum, dp: %E %E %E %E\n", kun, kad, dpnum, dp);
if (dp > umax){ dp = umax; } Fcal = Fcn;
}else{ dp = up; if (dp > umax)
{ dp = umax; } Fcal = F2n;
//printf("unLoad F2n, dp: %E %E \n", F2n, dp); }
// printf("unload unew, ddot, dp: %E %E %E \n", unew, ddot, dp);
} if (dp > umax){ dp = umax; }
if (dp < 0.0){ dp = 0.0; }
return Fcal;
}
```

C.5 Ring Cell Shearing Test LIGGGHTs Script File

```
# Ring shear test copper ore 1/31/2015
shell rm out.csv
#If it exists, remove the output file containing total torque on lid so that a new one can
be created
atom_style granular
boundary f f f
newton off
units si
region limits block -0.1397 0.1397 -0.1397 0.1397 -0.0508 0.1524
units box
```

create_box 4 limits

neighbor 12500000E-010

bin neigh_modify

delay 0

Material properties (Injection box, Copper Ore (rough), Side Walls, Top lid + base)

fix Load all property/global LoadingStiffness peratomtype 100.0 474788897.8 718168080.6 718168080.6

fix unload all property/global UnloadingStiffness peratomtype 100.0 2373944489 3590840403 3590840403

fix cohload all property/global CohesiveStiffness peratomtype 0.0 356091673.35 0.0 0.0

fix Youngs all property/global youngsModulus peratomtype 5E+06 119E+09 180E+09 180E+09

fix Poissons all property/global poissonsRatio peratomtype 0.3 0.34 0.3 0.3

fix Rest all property/global coefficientRestitution peratomtypepair 4 &

0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2

fix Friction all property/global coefficientFriction peratomtypepair 4 &

0.0 0.0 0.0 0.0 & 0.0 0.35 0.7 0.7 & 0.0 0.7 0.0 0.0 & 0.0 0.7 0.0 0.0

fix Rolling all property/global coefficientRollingFriction peratomtypepair 4 &

0.3 0.3 0.3 0.3 & 0.3 0.3 0.3 0.3 & 0.3 0.3 0.3 0.3 & 0.3 0.3 0.3 0.3

fix ViscousDamping all property/global coefficientRollingViscousDamping peratomtypepair 4 &

0.1 0.1 0.1 0.1 & 0.1 0.1 0.1 0.1 & 0.1 0.1 0.1 0.1 & 0.1 0.1 0.1 0.1

fix power all property/global NPower scalar 1.5

pair_style gran/luding/history rolling_friction epsd cohesion off tangential_damping on

pair_coeff * *

communicate single vel yes

timestep 1.0E-07

fix gravityFix all gravity 9.81 vector 0. 0. -1

Import material geometry

fix InjFace all mesh/surface/planar file inj_astm.stl type 1 move 0. 0. 0.075

fix SideWallID all mesh/surface file id_astm.stl type 3 curvature 1E-5

fix SideWallOD all mesh/surface file od_astm.stl type 3 curvature 1E-5

with attached fins:

fix Base all mesh/surface/stress file base_astm.stl type 4 curvature 1E-5

fix Lid all mesh/surface/stress file lid_astm.stl type 4 curvature 1E-5

fix SetMeshes all wall/gran/luding/history mesh n_meshes 3 meshes SideWallOD SideWallID Base rolling_friction epsd cohesion off tangential_damping on

Particle attributes 0.003175

fix CuOrefp3 all particletemplate/sphere 1001 atom_type 2 density constant 2481.258432 radius gaussian number 0.003175 1.05833E-4

fix CuOrefp3Injdist all particledistribution/discrete 1001 1 CuOrefp3 1.0

Particle insertion

group nve_group

region limits

fix Inj1 nve_group insert/stream seed 1001 distributiontemplate CuOrefp3Injdist maxattempt 1000 & nparticles 10000 massrate 35.0 start 1 all_in yes vel constant 0 0 -0.1 & insertion_face InjFace extrude_length 0.0254

fix integr nve_group nve/sphere

Set up thermodynamics output

fix timecheck all check/timestep/gran 1000 0.1 0.1

thermo_style custom step atoms ke f_Lid[1] f_Lid[2] f_Lid[3] f_Lid[6] f_Base[6]

thermo 300000

thermo_modify lost ignore norm no

Output

variable t equal f_Lid[6]

variable b equal f_Base[6]

variable ts equal “step*dt”

variable fz equal f_Lid[3]

fix out all print 600000 “${ts}, $t, ${fz},$b” append out.csv screen no title “Time (s), Torque (N-m), Force (N),Torque_Base (N-m)”

dump dmp all custom 600000 post/dump-*.model2 id type x y z vx vy vz fx fy fz omegax omegay omegaz radius mass

#dump dmpBase all mesh/stl 300000 post/stl/dumpstl-Base-*.stl Base

#dump dmpLid all mesh/stl 300000 post/stl/dumpstl-Lid-*.stl Lid

Dump mesh VTK files

dump dmpLidVTKStress all mesh/vtk 600000 post/stl/dumpvtkstress-Lid-*.vtk Lid stress-components stress area

dump dmpBaseVTKStress all mesh/vtk 600000 post/stl/dumpvtkstress-Base-*.vtk Base stresscomponents stress area

Run until particles settle 0.75sec

run 7500000

Move top wall down

unfix Inj1

unfix Lid

undump dmpLidVTKStress

smooth lid:

fix Lid all mesh/surface/stress/servo file lid_astm.stl type 4 com 0 0 0.1 ctrlPV force dim z target_val -1291 vel_max 5.0 kp 0.02 ki 0. kd 0.

fix SetMeshLid all wall/gran/luding/history mesh n_meshes 1 meshes Lid rolling_friction epsd cohesion off tangential_damping on

#dump dmpLid all mesh/stl 300000 post/stl/dumpstl-Lid-*.stl Lid

```
dump dmpLidVTKStress all mesh/vtk 300000 post/stl/dumpvtkstress-Lid-*.vtk Lid stress-components stress area
restart 10000000 post/restart/RingCell100N-*.rsrt run 2500000
# Rotate base and change control parameters; run until pre-shear has been achieved
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period 50
run 55000000 # up to 5.5 seconds for steadystate shear
# Stop base rotation unfix RotBase
# Rotate base in opposite direction to relieve shear stress (0.05 seconds)
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period -25
run 675000 # up to 0.0675 seconds for release of shear
# Stop base rotation unfix RotBase
#reconsolidate and shear material again 1
fix_modify Lid target_val -1130.4
run 2500000 # hold servo for 0.25 seconds
# Rotate base and change control parameters; run until pre-shear has been achieved
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period 50
run 55000000 # up to 5.5 seconds for steadystate shear
# Stop base rotation unfix RotBase
# Rotate base in opposite direction to relieve shear stress (0.05 seconds)
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period -25
run 675000 # up to 0.0675 seconds for release of shear
# Stop base rotation unfix RotBase
#reconsolidate and shear material again 2
fix_modify Lid target_val -970.0464
run 2500000 # hold servo for 0.25 seconds
# Rotate base and change control parameters; run until pre-shear has been achieved
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period 50
```

```
run 55000000 # up to 5.5 seconds for steadystate shear
# Stop base rotation unfix RotBase
# Rotate base in opposite direction to relieve shear stress (0.05 seconds)
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period -25
run 675000 # up to 0.0675 seconds for release of shear
# Stop base rotation unfix RotBase
#reconsolidate and shear material again 3
fix_modify Lid target_val -809.7203
run 2500000 # hold servo for 0.25 seconds
# Rotate base and change control parameters; run until pre-shear has been achieved
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period 50
run 55000000 # up to 4.4 seconds for steadystate shear
# Stop base rotation unfix RotBase
# Rotate base in opposite direction to relieve shear stress (0.05 seconds)
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period -25
run 675000 # up to 0.0675 seconds for release of shear
# Stop base rotation unfix RotBase
#reconsolidate and shear material again 4
fix_modify Lid target_val -649.3942
run 2500000 # hold servo for 0.25 seconds
# Rotate base and change control parameters; run until pre-shear has been achieved
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period 50
run 55000000 # up to 5.5 seconds for steadystate shear
# Stop base rotation unfix RotBase
# Rotate base in opposite direction to relieve shear stress (0.05 seconds)
fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period -25
run 675000 # up to 0.0675 seconds for release of shear
```

Stop base rotation unfix RotBase

#reconsolidate and shear material again 5

fix_modify Lid target_val -474.3593

run 2500000 # hold servo for 0.25 seconds

Rotate base and change control parameters; run until pre-shear has been achieved

fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period 50

run 55000000 # up to 4.4 seconds for steadystate shear

Stop base rotation unfix RotBase

Rotate base in opposite direction to relieve shear stress (0.05 seconds)

fix RotBase all move/mesh mesh Base rotate origin 0. 0. 0. axis 0. 0. 1.0 period -25

run 675000 # up to 0.0675 seconds for release of shear

Stop base rotation

unfix RotBase

C.6 Angle of Repose Test LIGGGHTs Script File

Input file generated by BFA. Do not edit without advanced knowledge of the properties

shell rm out.csv

#If it exists, remove the output file containing total torque on lid so that a new one can be created

Initializing Simulation Properties

atom_style granular

boundary p p f

newton off

units si

region limits block -12700000E-009 12700000E-009 -12700000E-008 12700000E-008 -19050000E-009 27305000E-008

units box

create_box 4 limits

neighbor 0.00125

bin neigh_modify

delay 0

Material properties

fix Load all property/global LoadingStiffness peratomtype 100.0 474788897.8 718168080.6 718168080.6

fix unload all property/global UnloadingStiffness peratomtype 100.0 2373944489 3590840403 3590840403

fix cohload all property/global CohesiveStiffness peratomtype 0.0 356091673.35 0.0 0.0

fix Youngs all property/global youngsModulus peratomtype 5.1E+06 119E+09 180E+09 180E+09

fix Poissons all property/global poissonsRatio peratomtype 0.3 0.34 0.3 0.3

fix Rest all property/global coefficientRestitution peratomtypepair 4 &

0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2

fix Friction all property/global coefficientFriction peratomtypepair 4 &

0.0 0.0 0.0 0.0 & 0.0 0.35 0.7 0.7 & 0.0 0.7 0.0 0.0 & 0.0 0.7 0.0 0.0

fix Rolling all property/global coefficientRollingFriction peratomtypepair 4 &

0.3 0.3 0.3 0.3 & 0.3 0.3 0.3 0.3 & 0.3 0.3 0.3 0.3 & 0.3 0.3 0.3 0.3

fix ViscousDamping all property/global coefficientRollingViscousDamping peratomtypepair 4 &

0.1 0.1 0.1 0.1 & 0.1 0.1 0.1 0.1 & 0.1 0.1 0.1 0.1 & 0.1 0.1 0.1 0.1

fix power all property/global NPower scalar 1.5

End material properties

pair_style gran/luding/history rolling_friction epsd cohesion off tangential_damping on

pair_coeff * *

communicate single vel yes

timestep 1.0E-07

fix gravityFix all gravity 9.81 vector 0 0 -1

Setting up Component Geometry

fix OutletX4d all mesh/surface file OutletX4d.stl type 3 curvature 1E-5

fix PlateX4d all mesh/surface file PlateX4d.stl type 4 curvature 1E-5

#There's no injection box, need to use a fake wall to "use up" atom_type 1

fix CHn15n67aiu14DjfnaFAKEWALLkkao3815nv2ie163hr all mesh/surface file OutletX4d.stl type 1 curvature 1E-5

#Use imported meshes to create walls fix SetMeshes all wall/gran/luding/history mesh n_meshes 2 meshes OutletX4d PlateX4d rolling_friction epsd cohesion off tangential_damping on

Define material distributions fix Coal all particletemplate/sphere 293431846 atom_type 2 density constant 2481.258432 radius gaussian number 0.003175 1.05833E-4

fix CoalPBX4ddist all particledistribution/discrete 875047539 1 Coal 1.0

Insertion Methods group nve_group region limits

region PBX4dRegion block -0.0127 0.0127 -0.127 0.127 0.127 0.254 units box fix PBX4d1 nve_group insert/pack seed 1162669527 verbose no

distributiontemplate CoalPBX4ddist maxattempt 1000 insert_every once start 1 overlapcheck yes all_in yes random_distribute uncorrelated vel constant 0 0 0

omega constant 0 0 0 region PBX4dRegion particles_in_region 10000

fix integr nve_group nve/sphere

Set up thermodynamics output

thermo_style custom step atoms ke

thermo 666719

thermo_modify lost ignore norm no

Output

variable k equal ke

variable ts equal "step*dt"

fix out all print 666719 "${ts}, $k" append out.csv screen no title "Time (s), KineticEnergy"

Number of cylinder components: 0

Insert first particles (else dump is empty) and set up dump (dumping at however many timesteps it takes to get the framerate)

dump dmp all custom 666719 post/dump-*.model2 id type x y z vx vy vz fx fy fz omegax omegay omegaz radius mass

Save restarts every - 500 restart 5000000 post/restart/AOR4d-*.rsrt

Begin Full Run - 432039

SimEvent at Time - 172816

run 19000000 start 0 stop 366678412

fix SetMeshes all wall/gran/luding/history mesh n_meshes 1 meshes PlateX4d rolling_friction epsd cohesion off tangential_damping on

Events Finished - 172816

run 30832119 start 0 stop 366678412

C.7 Passive Failure Test LIGGGGHTs Script File

Input file generated by BFA. Do not edit without advanced knowledge of the properties

shell rm out.csv

#If it exists, remove the output file

Passive Failure # 1/13/2015 2:50 PM

Initializing Simulation Properties

atom_style granular

boundary f f f

newton off

units si

region limits block -12700000E-009 12700000E-009 -10000000E-008 10000000E-008 -19050000E-009 27305000E-008

units box

create_box 4 limits

neighbor 0.00125

bin neigh_modify

delay 0

Material properties

fix Load all property/global LoadingStiffness peratomtype 100.0 474788897.8 718168080.6 718168080.6

fix unload all property/global UnloadingStiffness peratomtype 100.0 2373944489 3590840403 3590840403

fix cohload all property/global CohesiveStiffness peratomtype 0.0 356091673.35 0.0 0.0

fix Youngs all property/global youngsModulus peratomtype 5E+06 119E+09 180E+09 180E+09

fix Poissons all property/global poissonsRatio peratomtype 0.3 0.34 0.3 0.3

fix Rest all property/global coefficientRestitution peratomtypepair 4 &

0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2 & 0.2 0.2 0.2 0.2

fix Friction all property/global coefficientFriction peratomtypepair 4 &

0.0 0.0 0.0 0.0 & 0.0 0.35 0.0 0.0 & 0.0 0.0 0.0 0.0 & 0.0 0.0 0.0 0.0

fix Rolling all property/global coefficientRollingFriction peratomtypepair 4 &

0.0 0.0 0.0 0.0 & 0.0 0.3 0.0 0.0 & 0.0 0.0 0.0 0.0 & 0.0 0.0 0.0 0.0

fix ViscousDamping all property/global coefficientRollingViscousDamping peratomtype-pair 4 &

0.0 0.0 0.0 0.0 & 0.0 0.1 0.0 0.0 & 0.0 0.0 0.0 0.0 & 0.0 0.0 0.0 0.0

fix power all property/global NPower scalar 1.5

End material properties

pair_style gran/luding/history rolling_friction epsd cohesion off tangential_damping on

pair_coeff * * communicate single vel yes

timestep 1.0E-07

fix gravityFix all gravity 9.81 vector 0 0 -1

Setting up Component Geometry

fix PassiveBox all mesh/surface file PassiveBox.stl heal auto_remove_duplicates type 3 curvature 1E-5

fix ServoWall all mesh/surface/stress file ServoWall.stl heal auto_remove_duplicates type 4 curvature 1E-5

#There's no injection box, need to use a fake wall to "use up" atom_type 1

fix CHn15n67aiu14DjfnaFAKEWALLkkao3815nv2ie163hr all mesh/surface file ServoWall.stl type 1 curvature 1E-5

#Use imported meshes to create walls

fix SetMeshes all wall/gran/luding/history mesh n_meshes 2 meshes PassiveBox ServoWall rolling_friction epsd cohesion off tangential_damping on

Define material distributions

fix Coal all particletemplate/sphere 293431846 atom_type 2 density constant 19574997E-004 radius gaussian number 0.003175 1.05833E-4

fix CoalPBX4ddist all particledistribution/discrete 875047539 1 Coal 1.0

Insertion Methods group nve_group region limits

region PBX4dRegion block -0.0127 0.0127 -0.09525 0.09525 0.127 0.254

units box

fix PBX4d1 nve_group insert/pack seed 1162669527 verbose no distributiontemplate CoalPBX4ddist maxattempt 1000 insert_every once start 1

overlapcheck yes all_in yes random_distribute uncorrelated vel constant 0 0 0 omega constant 0 0 0 region PBX4dRegion particles_in_region 3600

fix integr nve_group nve/sphere

Set up thermodynamics output

thermo_style custom step atoms ke f_ServoWall[1] f_ServoWall[2] f_ServoWall[3]

thermo 333475

thermo_modify lost ignore norm no

OutPut

variable k equal ke

variable fx equal f_ServoWall[1]

variable fy equal f_ServoWall[2]

variable fz equal f_ServoWall[3]

variable ts equal “step*dt”

fix out all print 333475 “${ts}, $k, ${fx}, ${fy}, ${fz}” append out.csv screen no title “Time (s), ke, Wall_Fx (N), Wall_Fy (N), Wall_Fz (N)”

Number of cylinder components: 0

Insert first particles (else dump is empty) and set up dump (dumping at however many timesteps it takes to get the framerate)

dump dmp all custom 333475 post/dump-*.model2 id type x y z vx vy vz fx fy fz omegax omegay omegaz radius mass

dump dmpServoWall all mesh/stl 333475 post/stl/dumpstl-ServoWall-*.stl ServoWall

dump dmpServoWallVTKStress all mesh/vtk 32049 post/stl/dumpvtk-ServoWall-*.vtk ServoWall id

Save restarts every - 500

restart 5000000 post/restart/PassiveFailure-*.rsrt

Begin Full Run - 432039 # SimEvent at Time - 172816

run 10000000 start 0 stop 50000000

fix TranslateWall all move/mesh mesh ServoWall linear 0.0 -0.025 0.

run 6000000

unfix TranslateWall

run 333475

Events Finished -