

SWANSEA UNIVERSITY

MASTER OF SCIENCE THESIS

Discrete element simulations with LIGGGHTS

Author:

Carles BOSCH PADRÓS

Supervisor:

Professor Y. T. FENG

*A thesis submitted in partial fulfilment of the requirements
for the degree of Master of Science in Computational Mechanics*

in the

Zienkiewicz Centre for Computational Engineering (ZC2E)
College of Engineering

June 2014



Swansea University
Prifysgol Abertawe



Declaration of Authorship

I, Carles BOSCH PADRÓS, declare that this thesis titled, 'Discrete element simulations with LIGGGHTS' and the work presented in it are my own. I confirm that:

- A declaration that the work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.
- A statement that the thesis is the result of your own investigations, except where otherwise stated and that other sources are acknowledged by footnotes giving explicit references and that a bibliography is appended.
- A statement that the student gives consent for the thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed:

Date:

It takes a distinctive approach, emphasising that modelling is more art than science.

Anonymous

SWANSEA UNIVERSITY

Abstract

College of Engineering

Master of Science in Computational Mechanics

Discrete element simulations with LIGGGHTS

by Carles BOSCH PADRÓS

The granular material simulations improve the understanding of granular flows, common actions that are becoming more popular every day in the research and industrial world. The eruption of the Discrete Element method and all kinds of particle simulation software is playing a crucial role in the development of this field, still with room for a lot of progress. While it is possible to simulate with certain precision all kinds of particle flows and visualize all the details of the motion including forces, velocities, temperatures and so on, the already published papers about the Discrete Element Method prove the fact that the behaviour of some basic computational aspects in the simulations is yet to be analysed. Some of these aspects are going to be studied in this thesis through different examples of relatively simple situations with the help of an open source DEM code.

Acknowledgements

I would like to express my profound gratitude to Professor Y. T. FENG for being my thesis supervisor. He has been very helpful in guiding me and introducing me to a new method, without which, this thesis would not have been possible.

I am also thankful to European Commission for sponsoring this Erasmus Mundus Master Program and also to International Centre for Numerical Methods in Engineering (CIMNE), Barcelona, for making last year an inflection point in my career.

I would also like to thank my friends Mahendra, Farhad, Matthew and other well established friendships for making my days in Swansea a remarkable lifetime.

I owe my loving thanks to my parents, Glòra and Salvador for giving me everything I needed and supporting me throughout the Master program. I would also express my deep gratitude to my two brothers, Marina, CIMNE classmates and the rest of my close friends and family. Without their encouragement and understanding it would have been impossible for me to finish this work.

Contents

Declaration of Authorship	ii
Abstract	iv
Acknowledgements	v
Contents	vi
List of Figures	ix
List of Tables	xi
Abbreviations	xiii
Physical Constants	xv
Symbols	xvii
1 Introduction	1
1.1 What is a granular material?	1
1.2 The Discrete Element Method	3
2 The theory of the discrete element method	5
2.1 The discrete element method	5
2.1.1 The two body impact	5
2.1.1.1 Molecular dynamics	7
2.1.1.2 The continuum deformable model	7
2.1.1.3 Soft contact model	8
2.1.1.4 Hertz contact model	9
2.2 Computational Issues	10
2.2.1 Introduction	10
2.2.2 Object representations	10
2.2.3 Initial packing configuration	11

2.2.4	Contact Detection	12
2.2.4.1	Normal forces of the ellipse-ellipse contact	23
2.2.5	Interaction Laws	23
2.2.5.1	Hertz contact model	23
2.2.5.2	Adhesive elastic contact: JKR Model	24
2.2.5.3	DMT model	25
2.2.5.4	Van der Waals	25
2.2.5.5	Liquid Bridge models	25
2.2.5.6	The energy dissipation	26
2.2.6	Dynamic equations	30
3	LIGGGHTS Open Source Code	33
3.1	LIGGGHTS Open Source Code	33
3.1.1	Getting LIGGGHTS	34
3.1.2	LIGGGHTS input file	35
3.1.2.1	Organization of the input file	35
3.1.2.2	Main LIGGGHTS commands	37
3.1.3	Running LIGGGHTS	39
3.1.4	Post processing	40
3.1.4.1	Paraview	41
4	SAND IMPACT ON SURFACE	43
4.1	Introduction	43
4.2	Overview of the process	44
4.3	Governing equations	46
4.4	Case 1	47
4.4.1	LIGGGHTS input file	48
4.5	Case 2	52
4.6	Case 3	53
5	Cohesion	61
5.1	Introduction	61
5.1.1	Cohesion model	61
5.1.2	Overview of the process	61
5.1.3	Governing Equations	62
5.1.4	LIGGGHTS input file	62
5.1.5	Models	63
6	Conclusions	67
Bibliography		69

List of Figures

1.1	Bulk material	2
2.1	Basics of the body motion	6
2.2	Two spherical bodies before impact	6
2.3	Contact between the two bodies	7
2.4	The two bodies after the impact	7
2.5	Two discretised spherical bodies before the impact	8
2.6	Collision between two discretised spherical bodies	8
2.7	The two discretised spherical bodies after the impact	8
2.8	Contact between two soft particles	9
2.9	The most popular object representations in the DEM, spheres and ellipses	11
2.10	Polygon and polyhedron	11
2.11	Object representation formed by the gathering of various shapes	11
2.12	An example of initial packing configuration	12
2.13	The idea of the contact detection	12
2.14	Circular bounding box	13
2.15	Axis aligned bounding box	14
2.16	Object oriented bounding box	14
2.17	Brutal search: analysing the particles one by one	14
2.18	Cell/grid based search	15
2.19	NBS search	16
2.20	D-cell method	17
2.21	Bounding boxes in contact	18
2.22	Bounding boxes and objects in contact	18
2.23	Disk-disk contact	19
2.24	Contact between segment and segment	19
2.25	Contact between 2 ellipsoids	20
2.26	Polygons and Polyhedrons in contact	21
2.27	Polygon-polygon contact models	22
2.28	The rounding technique to solve the corner to corner contact	22
2.29	Normal direction in an ellipse-ellipse contact	23
2.30	Contact representation in the Hertz model	24
2.31	Van der Waals intermolecular forces	26
2.32	The liquid bridge model	26
2.33	Damping model	27

2.34	Representation of the coefficient of restitution	27
2.35	Rheological model: a spring mass dashpot representation of the particle interaction	28
2.36	Friction diagram. The F_t is the friction force, going against the displacement direction	29
2.37	Rolling friction diagram	30
2.38	Spinning friction model	31
4.1	Domain of the simulation	44
4.2	Overview of the process	45
4.3	Bottom surface formed by two triangular mesh elements	46
4.4	Distribution of Velocities at timestep 120. From top to bottom, left to right: $e=0.1, e=0.3, e=0.6$ and $e=0.9$	55
4.5	Distribution of Velocities at timestep 120. From top to bottom, left to right: $f=0.1, f=0.3, f=0.6$ and $f=0.9$	56
4.6	Distribution of Velocities at timestep 120. From top to bottom, left to right: $f=0.1, f=0.3, f=0.6$ and $f=0.9$	57
4.7	Distribution of Velocities. From top to bottom, left to right: $ts=90, 100, 120$ and 150	58
4.8	Distribution of Velocities. From top to bottom, left to right: $ts=90, 100, 120$ and 150	59
5.1	Motion of the granular material at different time steps. From top to bottom, left to right: $ts=230, ts=400, ts=550$ and $ts=700$	64
5.2	Motion of the granular material with cohesion at different time steps. From top to bottom, left to right: $ts=230, ts=400, ts=550$ and $ts=700$	64
5.3	Motion of the granular material with cohesion at 14 rad/s. From top to bottom, left to right: $ts=230, ts=400, ts=550$ and $ts=700$	65
5.4	Motion of the granular material with extremely low value of cohesion with the wall material. From top to bottom, left to right: $ts=230, ts=400, ts=550$ and $ts=700$	66

List of Tables

Abbreviations

DEM Discrete Element Method

LIGGGHTS Lammps Improved General Granular Heat Transfer Simulations

LAMMPS Large-scale Atomic Molecular Massively Parallel Simulator

Physical Constants

Pi π = 3.14159 (exact)

Symbols

m	mass	Kg
\vec{a}	acceleration	$\frac{m}{s^2}$
\vec{F}	Force	N
P	Interaction force between two particles in Hertz contact model	N
δ	Overlap/deformation	m
Y	Young's Modulus	Pa
d	Distance between centres of 2 particles	m
R	Radius	m
v	Poisson's ratio	
E	Stiffness	
a	Critical contact radius	m
$\Delta\gamma$	Surface energy	$\frac{J}{m^2}$
c_n	Damping force	N
d_n	Viscous damping	
ξ	Damping coefficient	
μ	Friction coefficient	
G	Shear modulus	$\frac{m}{s^2}$
k_n	Elastic constant for normal contact	
k_t	Elastic constant for tangential contact	
e	Coefficient of restitution	
ρ	Density	$\frac{Kg}{m^3}$
ω	angular frequency	rads ⁻¹

Dedicated to my parents

Chapter 1

Introduction

1.1 What is a granular material?

A granular material is a group of many discrete solid particles interacting with each other through collisions, heat transfer and other interaction types dispersed over a domain. When it comes to assign granular materials to a state type, neither solid, liquid or gas can represent the complete behaviour of the group of particles. That is the reason why this kind of materials is called the fourth state of matter. Imagine a pile of sand resting in a flat surface, its behaviour would respond to a solid one but, what if the flat surface starts to move creating a slope? The response would be really similar to a fluid as the pile would start to flow. Finally, the particles of a granular material behave like gaseous particles when they are under highly agitated systems.

Several variations of granular materials appear in nature, like sand or snow, but they can also be found in the industry with the shape of powders, chemicals or groups of products that are processed together. The popularisation of the DEM as a solid method and the appearance of different codes and softwares ready to solve problems related with granular materials have increased the appearances of this kind of simulations in all the engineering fields and day after day they are becoming more and more part of the aspects taken in account for scientists and engineers.

Different experimental studies have been carried out in the past to understand the behaviour of granular materials. Nowadays, as the processors in computers



FIGURE 1.1: Sand pile

have remarkably increased their capabilities, the simulations of granular flows are affordable, reliable and useful at the same time.

The present work is focused on reproducing different approaches to simple yet characteristic problems involving granular materials hitting a surface and emphasizing computational aspects directly affected by the nature of the problem, showing that, although the DEM is a well-established method, there are some aspects that can be studied and could modify and simplify the simulations.

It is well known that issues like the coefficient of restitution, coefficient of friction or cohesion have a direct influence in the impact behaviour, nevertheless, this thesis pursues the understanding of how these basic parameters perform in basic situations.

In order to do that, there will be an extended explanation of the Discrete Element Method theory including the different types of body impact, interaction laws, time integration schemes and other aspects related to it. An introduction and a guide on how to use the LIGGGHTS code and the visualization tool Paraview will also be provided right before the studied cases are presented. Each and every experiment will be preceded by a theoretical explanation and study of the case together with the simulation strategy and the governing equations, right before a final discussion over the obtained results.

1.2 The Discrete Element Method

Introduction:

The discrete element method is a numerical method used to compute different characteristics of systems that can be represented by particles. The philosophy behind the DEM is to model the actions that happen at a microscopic level and study how these actions affect the evolution of the motion of a whole media in a macroscopic level. This method needs mainly a contact detection strategy, time-discretised equations of motion governing the particle displacements and interaction laws describing the inter-particle relations.

Historically, it was Cundall in 1971 who created the first approach to granular methods by presenting the distinct element method. What this method proposed is, in a granular media, substitute the contact points with an spring. This spring could be normal or shear, also normal or shear dashpots and different kind of joints and sliders depending on the force that needs to be represented. The distinct element method was developed 14 years later, in 1985, into the generalized discrete element method by Mujinza and Owen.

The DEM is especially useful for granular media as its main feature is that it is modelled by particles, so either each particle simulated is going to be a single element or in case it is required to apply it to a continuous media (not usual) the material would be discretized into particle shaped elements. While the behaviour of such small elements as particles may not be valuable enough to study in a microscopic way, the gathering of all these particles acting together involves a certain magnitude of facts more than valuable enough to study in a macroscopic basis.

In order to understand this method, it is important to know the basic principles that rule it. For the DEM, the old and famous Newton's Laws, formulated by Isaac Newton in 1697, would be the starting point. It is widely known that Newton's Second Law governs the motion of bodies under the action of various forces. This law is going to be basic as DEM problems involve a lot of bodies in movement interacting with each other.

The other crucial fact of this method is the impact between 2 bodies, an action that is going to take place several times in the computations providing the most important information for the problem being solved. The law that will control these actions is the most important one in the DEM. Before getting into more

detail, only the integration scheme is missing for the simulation. For the DEM, the central difference approximation.

Chapter 2

The theory of the discrete element method

2.1 The discrete element method

Second Newton's Law:

One of the most famous laws in physics is basic to understand the motion of any particle group. It describes the relation between a body, the forces acting on it and the effect it has on its motion. The second Newton's Law states that:

$$m\vec{a} = \vec{F} \quad (2.1)$$

Where m is the mass of the body, a is the acceleration and F is the total force that acts on the previous mass. Although it may seem basic, each of the granular bodies has a motion and that motion, as any other body, responds to this principle in terms of motion and force^{2.1}. It is also a quite common fact that granular bodies or particles have a combined motion formed by translation and rotation.

2.1.1 The two body impact

The most repeated action in a granular media simulation is the impact between particles, what makes it a really important part of the method. Recalling the

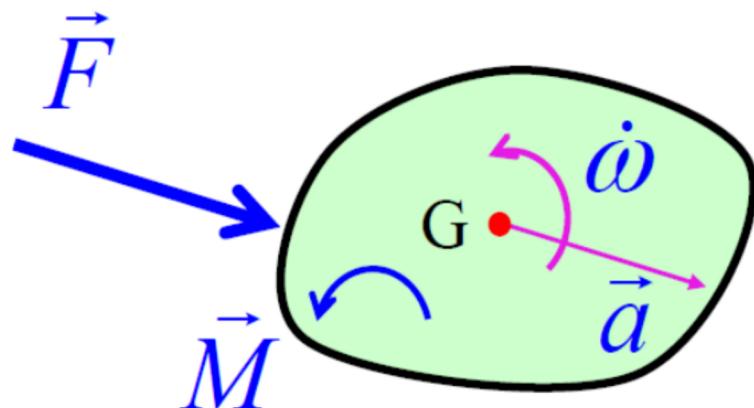


FIGURE 2.1: Basics of the body motion

theory of simple impacts in physics, when two bodies with a certain mass and velocity each one collide, the post-collision behaviour can be studied in terms of the initial velocity and mass of the two bodies and the energy lost during the impact.

When the two bodies start the contact, the energy can be dissipated in terms of sound, deformation, vibration and heat. Depending on the material, the deformations could reach the plastic state and be permanent. If the behaviour is elastic, it will be observed that from the point where the contact starts until it ends, the deformation in time will follow a pyramid shape going up in the first half of the contact until a maximum deformation point, when the two bodies will start to separate and deformation will accordingly decrease.

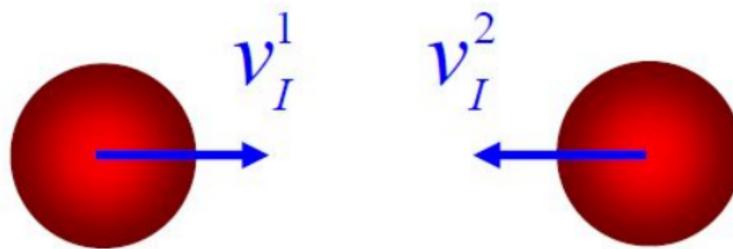


FIGURE 2.2: Two spherical bodies before impact

The force created in these impacts is essential to compute the stresses and strains affecting the particles of our system. In order to study the interaction of particles through the impacts, different models have been created proposing different strategies to solve these situations each one offering different output.

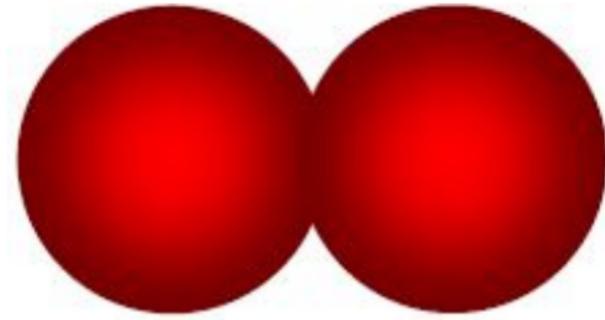


FIGURE 2.3: Contact between the two bodies



FIGURE 2.4: The two bodies after the impact

2.1.1.1 Molecular dynamics

One of the main characteristics of this method is that all particles are assumed rigid. When two particles collide, there is a relation between the velocity before and after the impact. This relation is called coefficient of restitution. This coefficient can be either given or computed and it can take values from 0 to 1. The bigger the coefficient is, the more the energy is conserved until one can obtain a total energy conservation for a value of $e=1$.

2.1.1.2 The continuum deformable model

As previously stated, the information about the contacts is very important in terms of knowing the stresses and strains. How to obtain it? For a Finite Element discretisation, each particle of the granular media being studied is deformable and would be discretized into various elements.

After an impact, this method would give precise results about deformation, impact forces and velocities as the discretisation would be more accurate in the areas where the impact is held.[2.6](#) On the other hand, particle simulations usually involve a large number of bodies, which means even a larger number of elements, deriving in a huge computational cost.

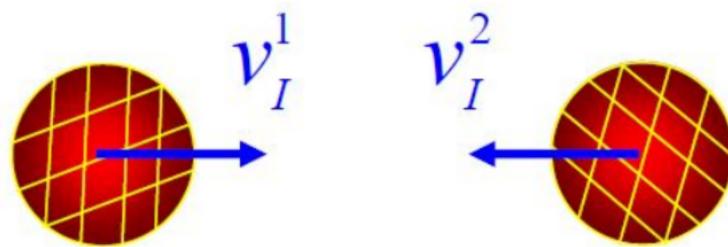


FIGURE 2.5: Two discretised spherical bodies before the impact

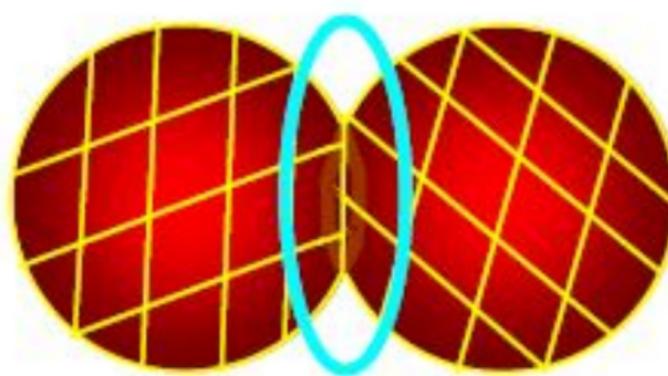


FIGURE 2.6: Collision between two discretised spherical bodies



FIGURE 2.7: The two discretised spherical bodies after the impact

2.1.1.3 Soft contact model

This model is based in the classic Newtonian Mechanics. If every particle in the simulation is treated as a single rigid element with small overlap allowed, the interaction force can be defined by a contact law and the motion of the bodies can be defined by dynamics equations. [2.8](#)

That method gives less information about the process but at the same time the computational cost is less than the continuum model as the number of elements is reduced to the number of particles studied. This model is the starting point of the Discrete Element Method.

Another essential component of the DEM are the previously introduced contact laws. These laws will provide the information related to the impact between two bodies. The most important one is the Hertz contact model.

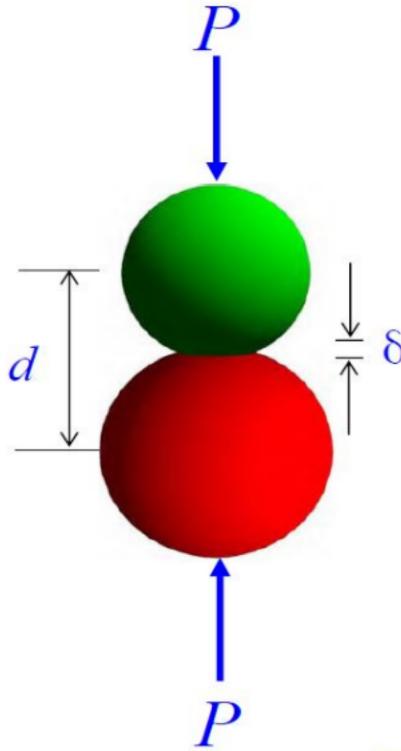


FIGURE 2.8: Contact between two soft particles

2.1.1.4 Hertz contact model

This law states the relation between the compression or overlap of the particles and the force created when they collide. Suppose two rigid and elastic spheres with specific young modulus, radius and velocity. The contact force generated by the impact of these two particles will be a straight line passing through the point where they established contact and the centres of the particles. The value of the force will be given by this equation:

$$P = \frac{4E^* \sqrt{R^*}}{3} \delta^{\frac{3}{2}} \quad (2.2)$$

$$\delta = R_1 + R_2 - d \quad (2.3)$$

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2} \quad (2.4)$$

$$\frac{1}{E^*} = \frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2} \quad (2.5)$$

Where each R corresponds to each radius and the d stands for the distance between the centres of the two particles.

For the Hertz contact model, no tension force is allowed in the contact area, neither is friction and the overlap can never be bigger than the radius of the particles. The Hertz contact law is not the only law possible in DEM problems: This model stands for the forces caused by the physical contact between two particles due to an impact, but there are other contact laws to be considered such as the liquid bridge, Van der Waals, electromagnetic forces and so on. All of them are going to be discussed further ahead.

2.2 Computational Issues

2.2.1 Introduction

It has been stated in the introduction of the method that nowadays, with the power of the available computers, discrete element can be easily implemented and simulated. The use of computational techniques is basic when it comes to solve even simple problems. To understand how the computer can understand and solve such problems, there are a lot of issues to take in account. Some of the bases of the method have been established previously, but the strategy can be organised in 3 main aspects, yet to be discussed: Object interpretation, contact detection and dynamic solutions.

2.2.2 Object representations

This section will talk about the different geometric descriptions of objects that can be found in the Discrete Element Method and how can they be represented and set for a packing configuration.

There are 4 main types of object representations:

1.Spheres and their family^{2.9}:

Spheres and ellipses are the most common shape in DEM, also the simplest one in terms of computational issues.

2.Polygon and polyhedron^{2.10}:

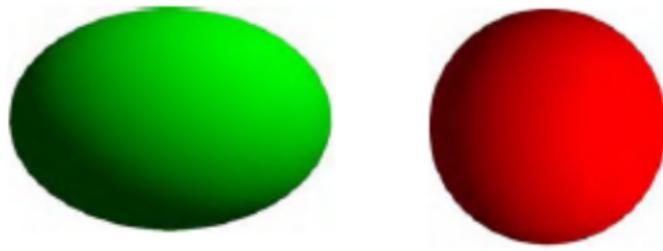


FIGURE 2.9: The most popular object representations in the DEM, spheres and ellipses

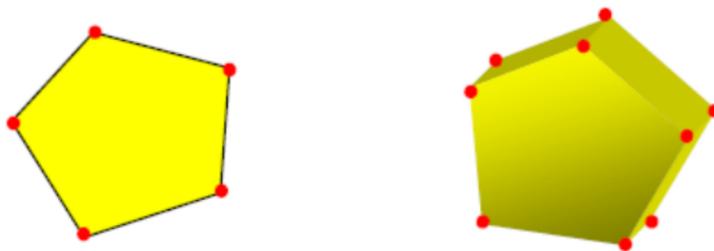


FIGURE 2.10: Polygon and polyhedron

3. Compound form [2.11](#): In some cases simple objects bounded together to model more complicated objects.

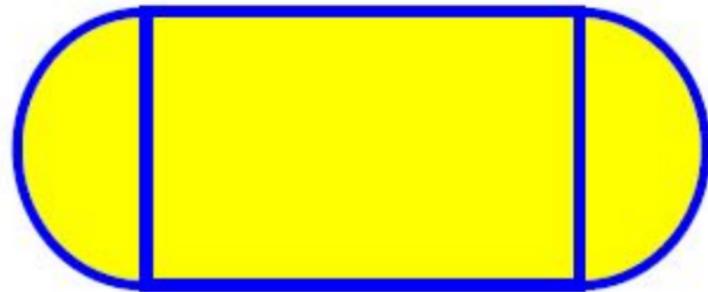


FIGURE 2.11: Object representation formed by the gathering of various shapes

4. Hierarchical: The cases when the objects are represented by a finite element discretisation.

2.2.3 Initial packing configuration

The initial particle configuration is something determined by the problem that has to be represented as accurate as possible because in most of the problems it would

play a crucial role. It can involve the initial position, velocity or distribution of the particles in the domain.[2.12](#)

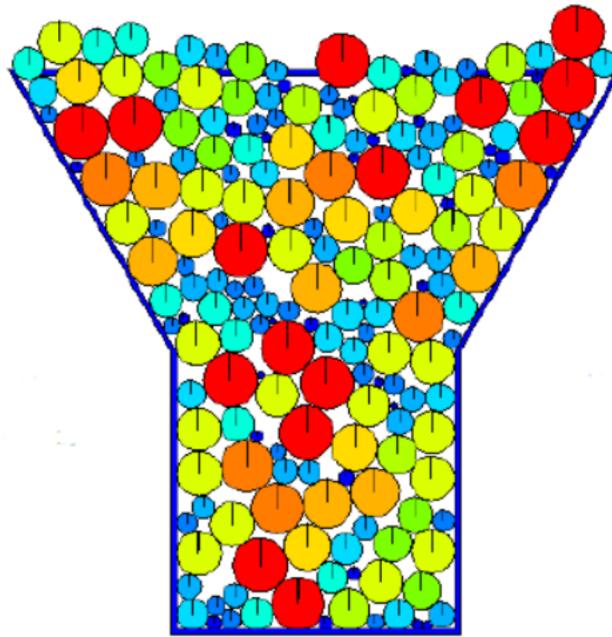


FIGURE 2.12: An example of initial packing configuration

2.2.4 Contact Detection

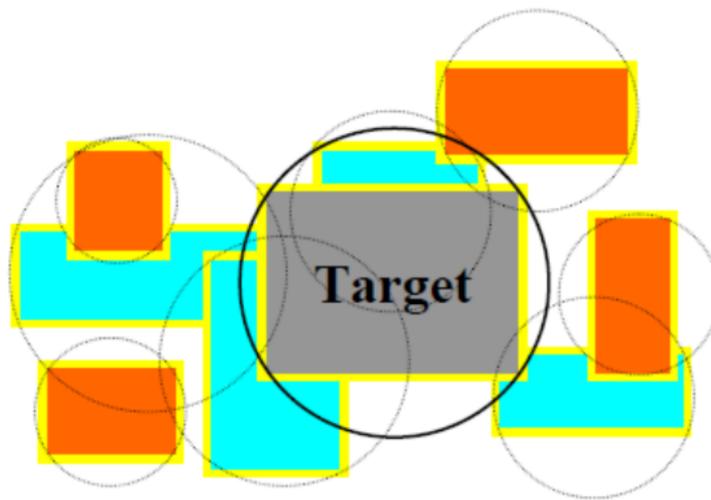


FIGURE 2.13: The idea of the contact detection

Contacts are one of the most important aspects of the Dem and at the same time the most characteristic issue. Locating and solving them are aspects that need to be faced to obtain a realistic representation with the less computational cost. This

part will set the algorithm in charge of saying whether a particle is or is not in contact with another. Notice that the contact search can take up to 90 percent of the computation time as this action has to be done for every time step, following the evolution of the simulation taking in account the large number of particles involved and their specific shapes.

The contact detection is formed by three main steps:

1. The global contact search:

The first step, its mission is to locate all the possible contacts in the whole domain. In order to do that, each particle will be assigned a bounding box according to its shape. The global contact search will determine which bounding boxes overlap with a previously selected box called the target.

Bounding box:

The bounding boxes are volumes surrounding the particles that arbitrarily represent their shapes. Even if they are arbitrary geometries, each particle would have the bounding box with the closest geometry to its shape. The most common options are spheres, axis aligned and oriented bounding boxes.

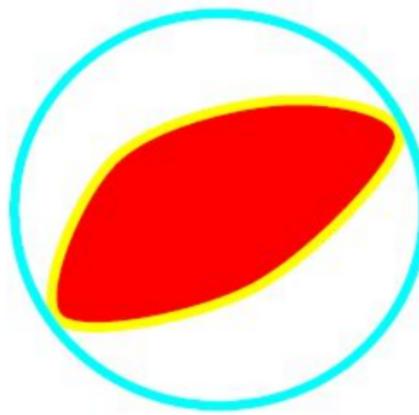


FIGURE 2.14: Circular bounding box

2. Locating possible contacts:

The following step is locating the possible contacts. There are several search methods:

Brutal search: Check the relation between all the bounding boxes. It requires a lot of memory, computationally expensive.[2.17](#)

Cell/Grid based search: For that method the domain is going to be divided into a grid. The first step in this search method is finding the biggest particle in terms

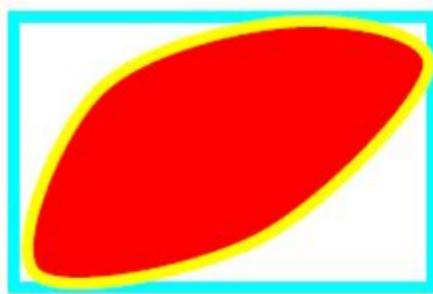


FIGURE 2.15: Axis aligned bounding box

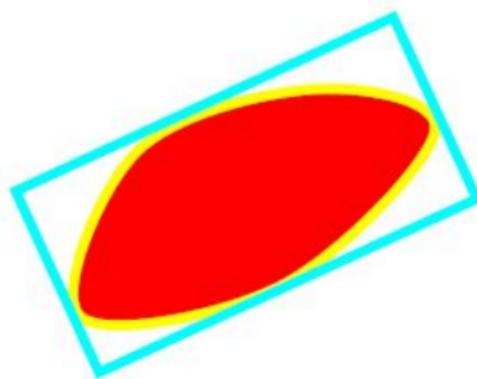


FIGURE 2.16: Object oriented bounding box

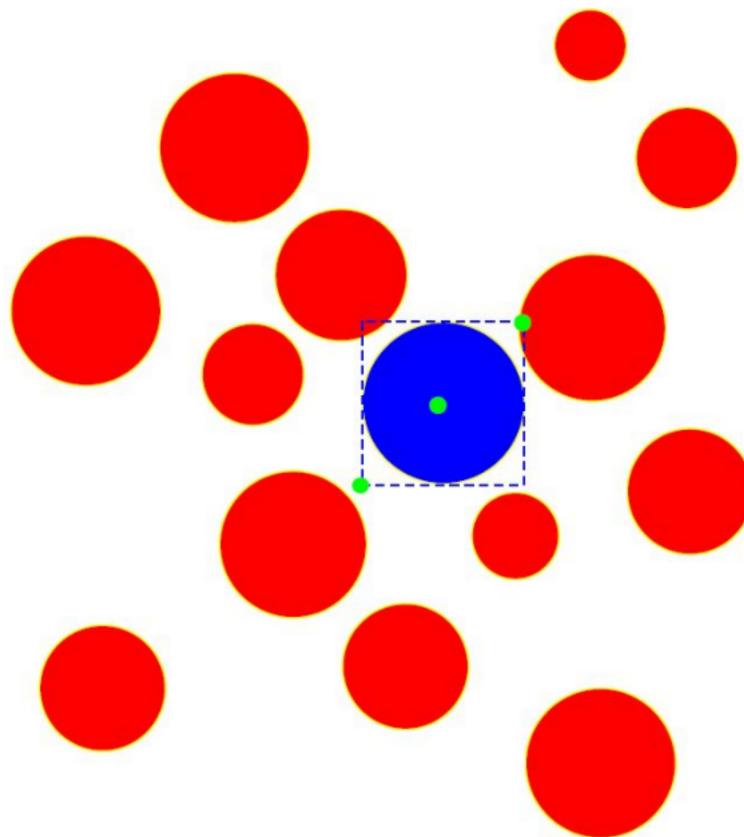


FIGURE 2.17: Brutal search: analysing the particles one by one

of size in order to give its measures to set the size of the cells. Afterwards, each particle has to be assigned into a cell in a step called the mapping stage.

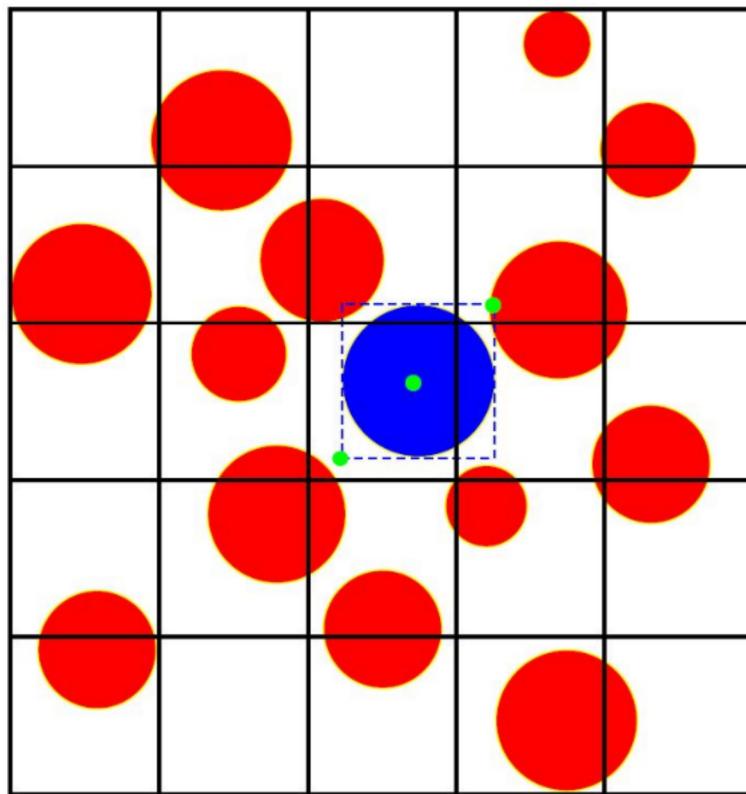


FIGURE 2.18: Cell/grid based search

In order to avoid confusion when it comes to assign particles into cells, it has to be specified which part of the particle counts, for example, assigning particles to cells depending on their centre, lowest or highest part is. Once all the particles are assigned a cell, looping over all cells, the overlap between objects in the same cell is checked, followed by checking the overlap between objects from different cells. This method is quite simplistic and effective, especially for small simulations as the computational cost is directly related to the number of cells evaluated.

NBS search: The first step is again finding the maximum space an element can occupy in order to set the grid size with that measure. Afterwards, assign each particle to the cell containing its centre. Until that point the strategy is the same as the previous method. The following step is creating for the whole domain linked lists of particles that are in the same row of cells. For every linked list, check overlapping in its row and also the adjacent. Loop over all elements.

This method features no looping over all the cells and it processes two rows at a time, which reduces computational cost. Hence, it is a valid strategy for large simulations.

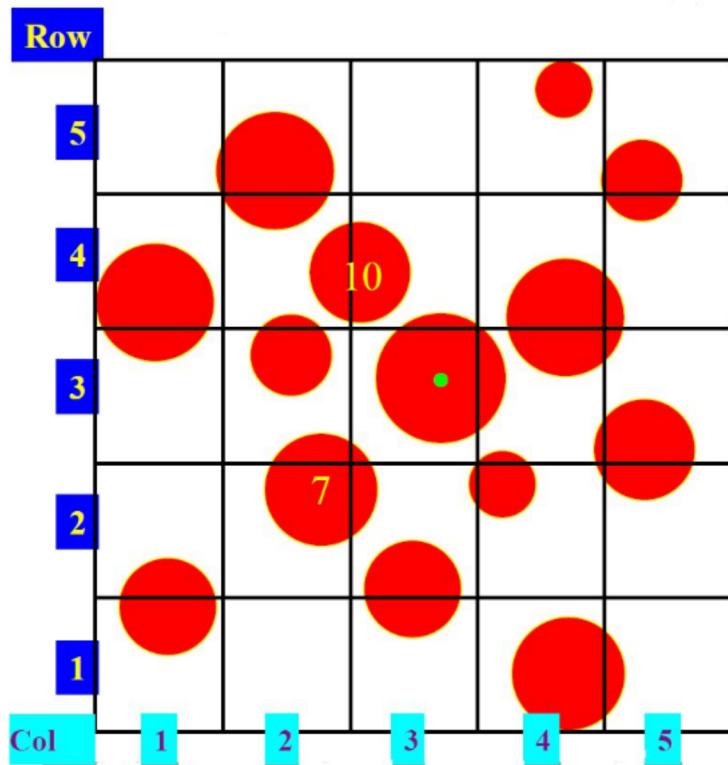


FIGURE 2.19: NBS search

D-cell: For this method the domain needs to be divided into rows. Map each particle to a row according to the lowest corner of the bounding box and create a linked list. Each particle is assigned to a cell from the current row and they start to be processed from the left. An object is migrated to the following row if the upper y coordinate is greater than its lower y coordinate of the next row. These elements can be also migrated to the following cell if their upper x coordinate is bigger than their lower x coordinate.

The detection consists in, taking row by row, check the overlapping between all boxes according the previous restrictions. In this case the computational costs depend on different issues: cell size, object distribution, packing density and obviously the computer capabilities.

Dgrid: For this method, the overlapping is checked one row and one column at a time, with no loop over cells. The main feature of the strategy is the arbitrary setting of the cell size. The Dgrid is very effective for large simulations as the distribution of the particles does not affect the development of the method.

3. Local Resolution check

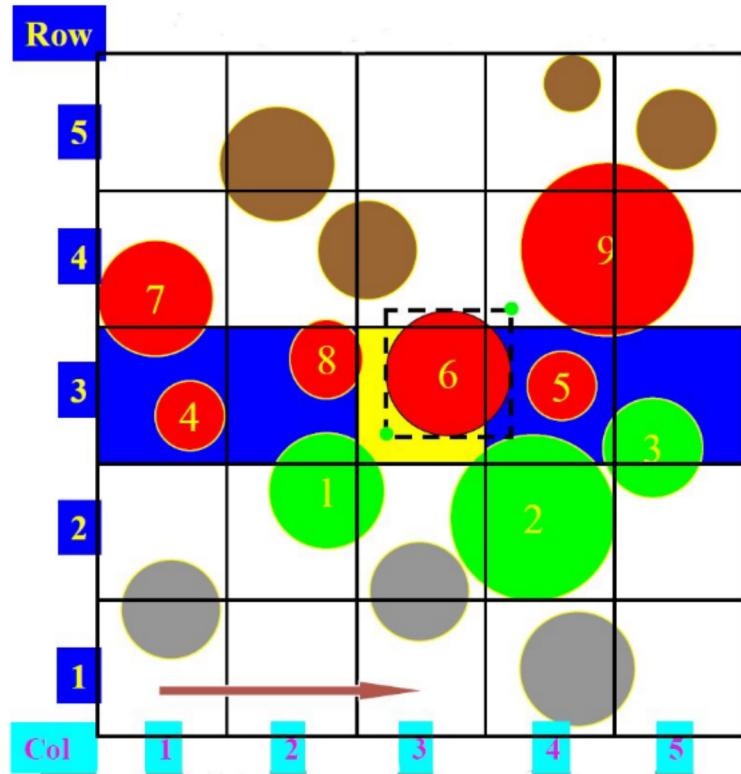


FIGURE 2.20: D-cell method

The actual local configuration with all the contacts detected has to be defined. Once the two overlapping boxes are located, it is time to confirm that the two particles involved are in contact according to their real shape and apply the corresponding contact interaction laws afterwards.[2.21](#) This procedure could involve detecting the normal and tangential direction of the contacts, locating the contact point and the type of overlap occurred.

Type of contacts:

The contacts can be divided in two big groups:

-Long range contact: These forces can appear without the physical contact between particles. They contribute the adhesion and coalescence of particles, mainly at the microscopic level. Van der Waals and liquid bridge are the most common examples.

-Short range contact: due to the direct impact between two elements. All its aspects commented below.

If it is confirmed that two particles are in contact, the normal and tangential contact detections, the contact point and the characteristics of the overlap such as penetration, contact width and area have to be determined too. As previously stated, the contact detection part is the one taking the most computational time,

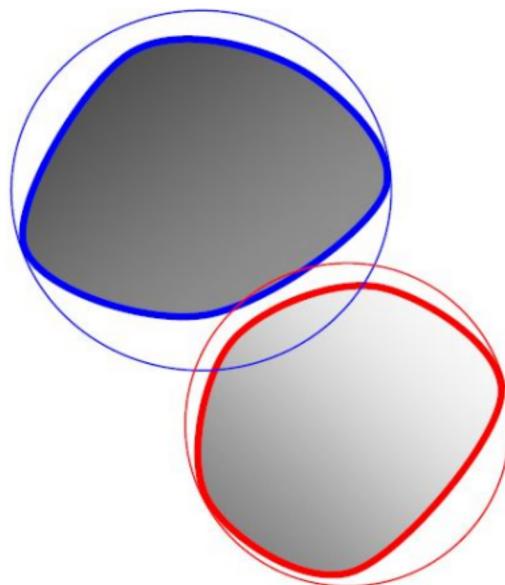


FIGURE 2.21: Bounding boxes in contact

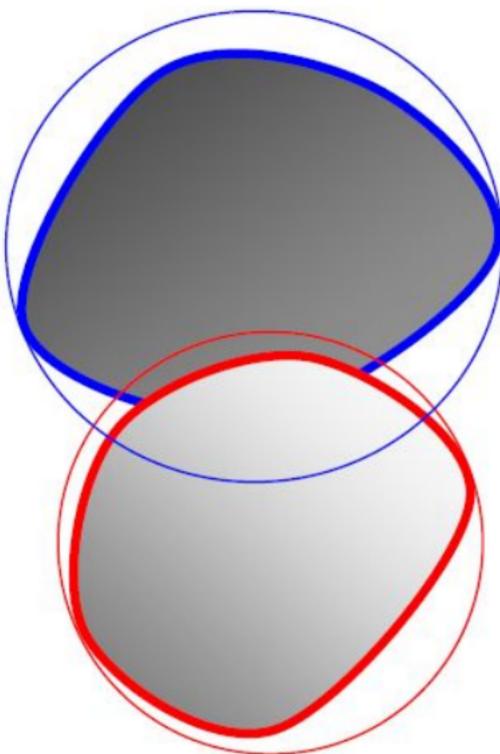


FIGURE 2.22: Bounding boxes and objects in contact

up to 90 percent but, inside this part, the local contact resolution is the most time consuming step.

Type of contact pairs (There are also a few more complex contact pairs highly non-linear, Super ellipsoids and poly ellipsoids):

Disk-disk sphere-sphere

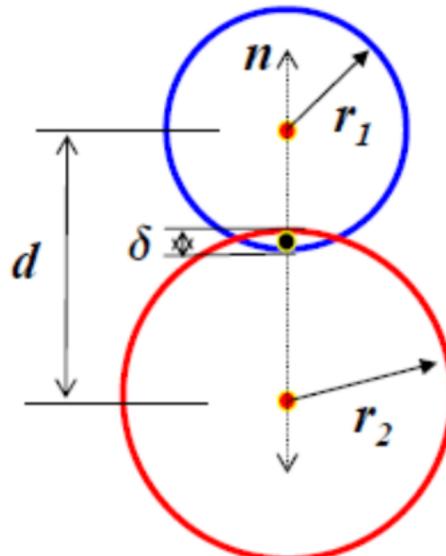


FIGURE 2.23: Disk-disk contact

Node-segment segment-segment

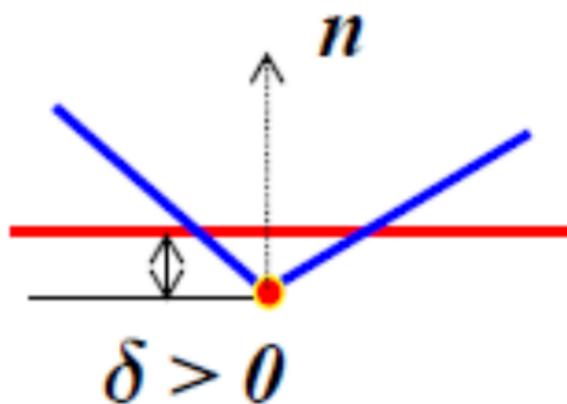


FIGURE 2.24: Contact between segment and segment

Ellipse-ellipse ellipsoid-ellipsoid

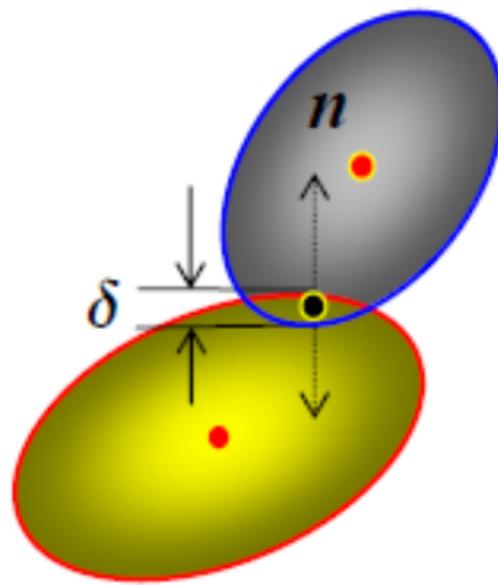


FIGURE 2.25: Contact between 2 ellipsoids

Plus all the possible combinations between them. Notice that each type of contact pair will have its own local resolution.

Contact resolution of ellipses: Method 1

The contact of two ellipses can be determined by the simultaneous solution of two non-linear quadratic equations. The use of an iterative procedure, such as Newton Raphson method is usually required. This part is computationally very expensive. Two solutions of the system of equations would imply two intersections between the ellipses while no solutions would mean there is no contact. To solve the problem, approximate solutions procedures are usually simpler and preferred.

Contact resolution: Temporal coherence

Due to the use of explicit time integration schemes for dynamic analysis where the time step is generally very small, the difference between the two contact configurations at two consecutive time instants are normally very small. Thus some contact characteristic at the current time step can be used as a very good initial guess for the next time step to significantly increase the solution convergence of the contact resolution. It should be exploited wherever possible.

Contact resolution of ellipses: Approximation

The contact resolution reduces to trivial cases of disk-disk (sphere and slightly more complex rotational body formed by the two middle arcs in 3d cases) contacts, although identification of correct contact pairs is required. The scheme faces a

minor problem at the four connection points (or 2 rings in 3d) when the curvature is needed in calculating the normal force.

Basic type of contact pairs part 2:

Polygon-polygon

Polyhedron-polyhedron

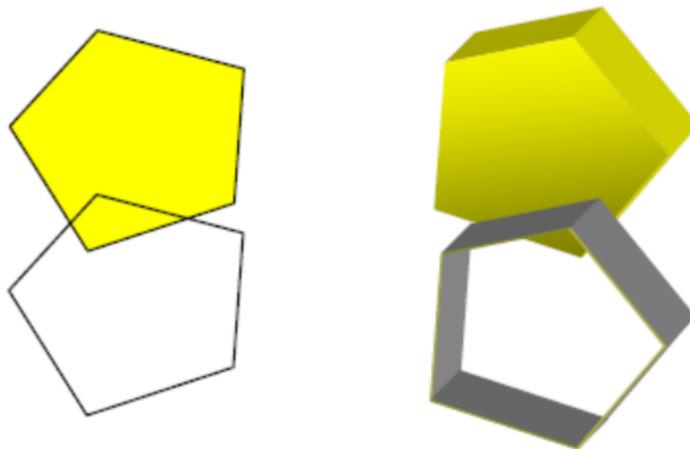


FIGURE 2.26: Polygons and Polyhedrons in contact

The overlap region created, which is also a polygon or polyhedron, can be determined by existing algorithms

Polygon-polygon contact: The most current applications of the DEM method are related to cases where the particles have a common and simplistic shape, but cases with more complicated geometries exist. The fundamentals on contacts between polygons follows:

The first assumption that needs to be done is that the polygons are rigid and all the impacts will behave as elastics. Following these principles it is obvious that the energy is conserved but not the momentum. There will not be enough equations to determine the lineal and angular velocity of the bodies. Two types of physic contact can be found: corner to flat and corner to corner.

Corner to corner is one of the biggest problems in DEM as the direction of the tangential remains unknown while in the corner to flat contact the tangential will depend on the flat surface interacting in the impact. In terms of the overlapping the situation is really similar: For the corner to flat case the overlapping is the distance between the flat part and the corner that impacted.

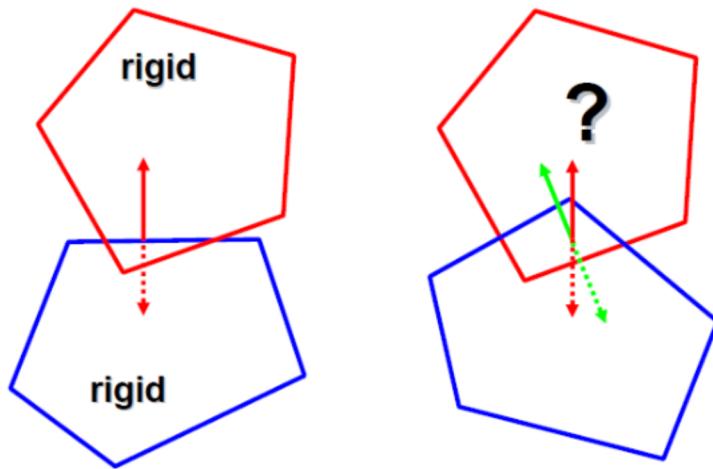


FIGURE 2.27: Polygon-polygon contact models

For the corner to corner case, the problem is again complex as the interpenetration creates another complex geometry. One possible solution to the corner to corner problem is the corner rounding.[2.28](#) The transformations of the impacted corners into arcs can overcome the complexity but the results would not be reliable as they would change depending on the radius chosen.

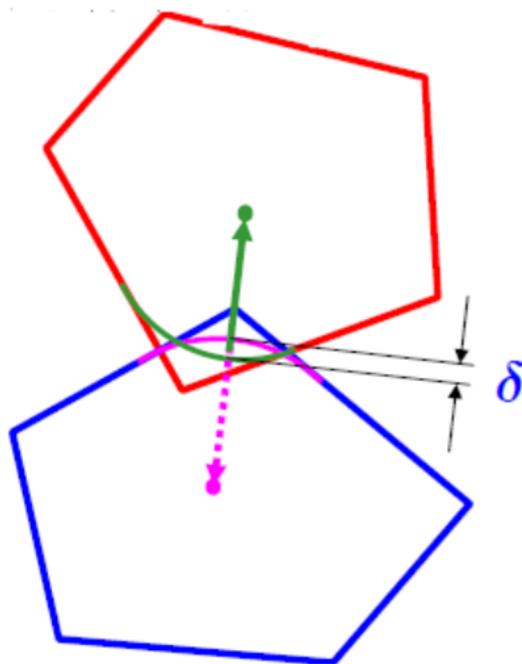


FIGURE 2.28: The rounding technique to solve the corner to corner contact

2.2.4.1 Normal forces of the ellipse-ellipse contact

In case of two ellipses impacting, the normal direction and the overlap can be determined as previously pointed in order to apply the contact models. But this is not the only approach, the shape created by the ellipses in the contact area can be replaced by two regular spheres with the contact features that this means^{2.29}.

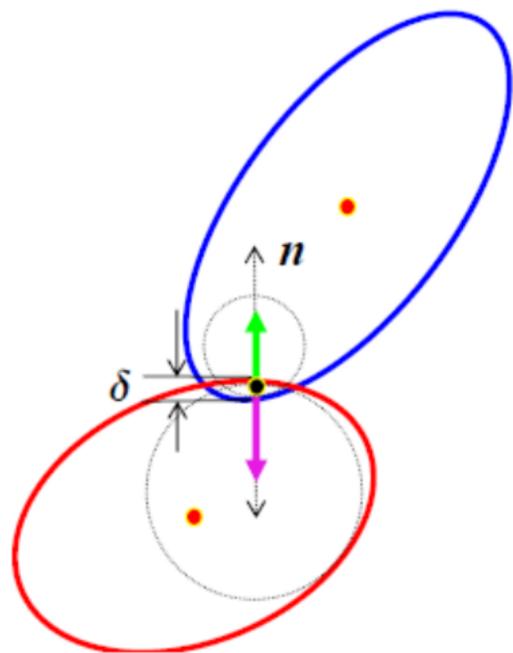


FIGURE 2.29: Normal direction in an ellipse-ellipse contact

2.2.5 Interaction Laws

The interaction laws mainly define the relation between the overlapped distance and the contact force. A lot of variables have an influence on these laws: the physics of the problem, the conditions chosen and so on. Recall that one of the most important issues in the interaction laws is the small overlap allowed as basic assumption of the DEM. The total contact force would take the value of the force obtained with the interaction law plus an obtained damping force. The most common interaction laws are: Linear, hertz and power.

2.2.5.1 Hertz contact model

For this previously introduced method the spheres have to be linear, elastic and isotropic. It is applicable to cases when velocity is really small because it is

usually for static cases. The parameter k depends on the size of the spheres.[2.30](#) The Hertz model stands for an elastic behaviour, but elasto-plastic models can also be obtained with experiments.

$$P = \frac{4E^* \sqrt{R^*}}{3} \delta^{\frac{3}{2}} \quad (2.6)$$

$$\delta = R_1 + R_2 - d \quad (2.7)$$

$$\frac{1}{R^*} = \frac{1}{R_1} + \frac{1}{R_2} \quad (2.8)$$

$$\frac{1}{E^*} = \frac{1 - v_1^2}{E_1} + \frac{1 - v_2^2}{E_2} \quad (2.9)$$

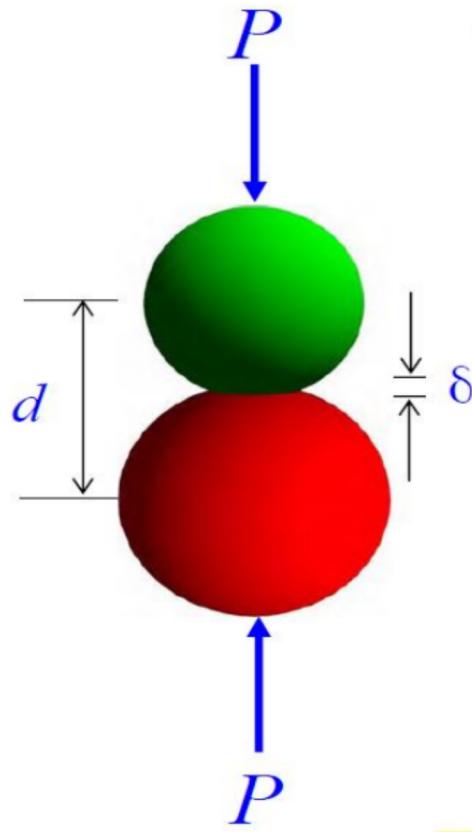


FIGURE 2.30: Contact representation in the Hertz model

2.2.5.2 Adhesive elastic contact: JKR Model

The main feature of that method is the introduction of adhesion to the Hertz contact model. Basically the adhesion forces deform the Hertz contact profile.

The key of that method is the balance between the stored elastic energy and the loss in surface energy. This method is well suitable for soft particles.

$$F = \frac{4E^*a^3}{3R^*} - \sqrt{8\pi\Delta\gamma E^*a^3} \quad (2.10)$$

where a is the critical contact radius and $\Delta\gamma$ is the surface energy.

2.2.5.3 DMT model

In this case, the adhesion forces are added to the hertz contact force instead of deforming the hertz contact profile. That one is more accurate for hard particles.

$$F = \frac{4E^*a^3}{3R^*} - 2\pi\Delta\gamma R^* \quad (2.11)$$

2.2.5.4 Van der Waals

This interaction law is part of the long range contact laws. It is an attractive or repulsive force between particles of the same substance just for the fact that they are close to each other.[2.31](#) Hamaker model:

$$F_n = \frac{AR^*}{6s^2} \quad (2.12)$$

Where A is the Hamaker constant. Another way to compute this force is the Lennard Jones potential:

$$V(s) = 4l\left[\left(\frac{s_{min}}{s}\right)^{12} - \left(\frac{s_{min}}{s}\right)^6\right] \quad (2.13)$$

Where l is the depth of potential well and s_{min} is the zero potential separation.

2.2.5.5 Liquid Bridge models

These forces occur between two wet particles. They mainly comprise capillary and viscous forces. The capillary one appears from the surface tension of the liquid and the hydrostatic pressure difference across the air liquid interface. It depends on the surface tension, the radius of the spheres and their separation and contact angle.[2.32](#)

$$F_n = 2\pi\gamma R \sin\varphi \sin(\varphi + \theta) + \pi R^2 \Delta p \sin^2\varphi \quad (2.14)$$

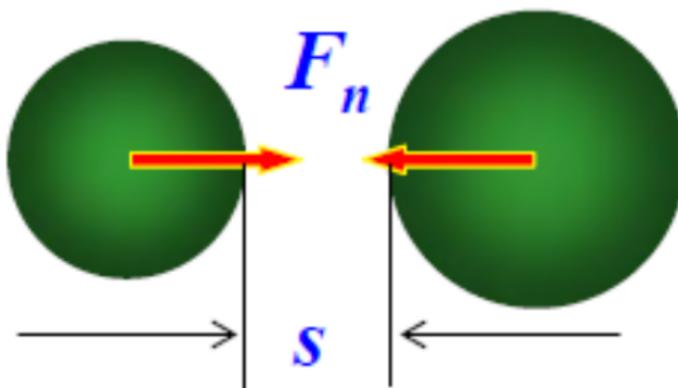


FIGURE 2.31: Van der Waals intermolecular forces

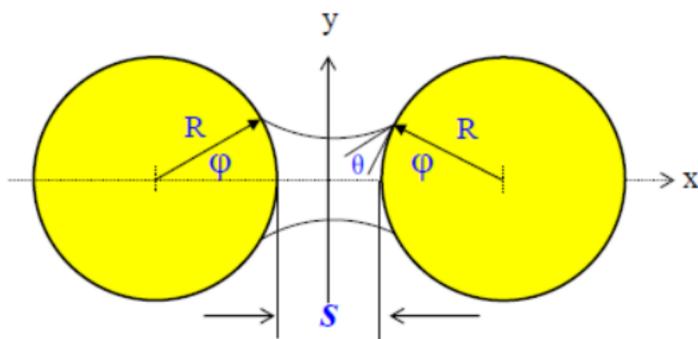


FIGURE 2.32: The liquid bridge model

2.2.5.6 The energy dissipation

The energy dissipated during the impact is the area found between the loading and unloading in the force-deformation diagram. Obviously, the shape of the area will change depending on the case studied. For viscous damping and for restitution:

$$F'_n = F_n(\delta) + c_n \delta \quad (2.15)$$

Rheological models: These models can be really helpful when it comes to understand the interaction laws between two bodies. These models are more specifically spring mass dashpot representations.[2.35](#)

Where the effective mass is:

$$\frac{1}{m^*} = \frac{1}{m_1} + \frac{1}{m_2} \quad (2.16)$$

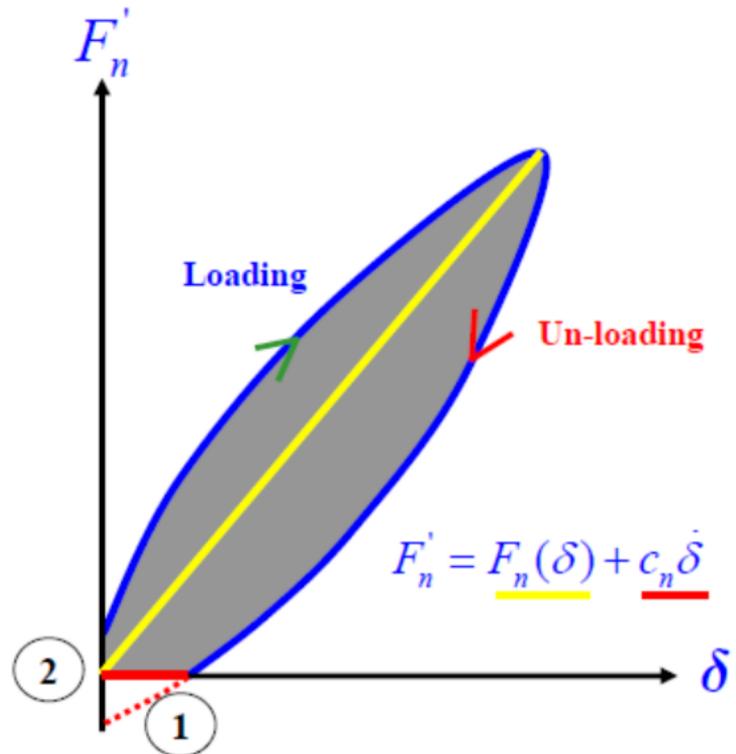


FIGURE 2.33: Damping model

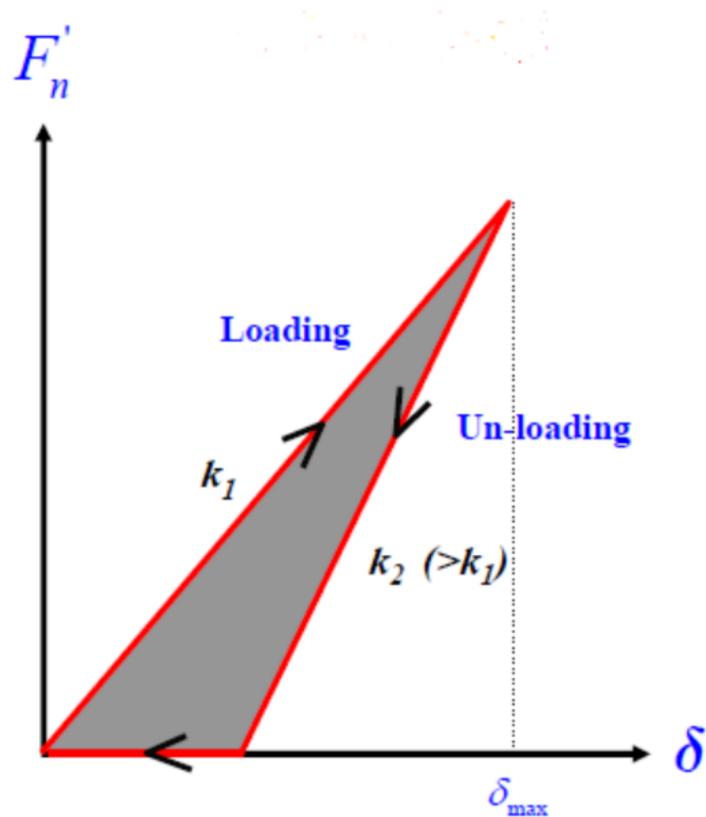


FIGURE 2.34: Representation of the coefficient of restitution

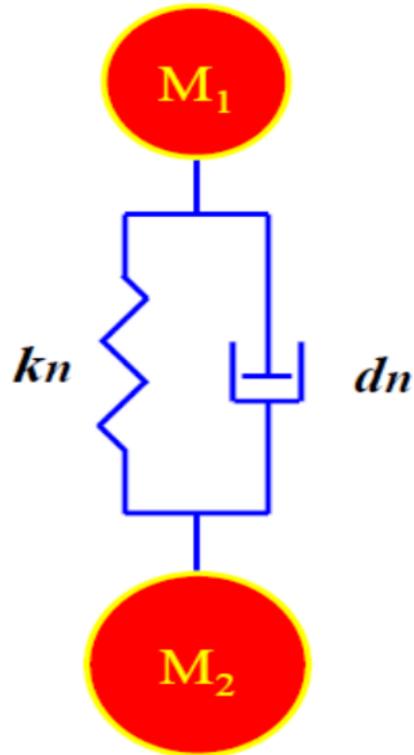


FIGURE 2.35: Rheological model: a spring mass dashpot representation of the particle interaction

And the viscous damping can be expressed as:

$$d_n = 2 \sqrt{\frac{m^*}{k_n \xi \delta}} \quad (2.17)$$

Where ξ is the damping coefficient.

Friction:

Frictional forces appear due to the adhesion of two surfaces that are in contact. The most common frictional model implemented in the DEM is the Coulomb frictional model.

Coulomb friction model: In this type of friction, the slip is a gathering of a restorable elastic and a permanent plastic part.

$$F_t = \frac{\mu F_n \delta_T}{\|\delta_T\|} \quad (2.18)$$

Where δ_T is the sliding velocity.

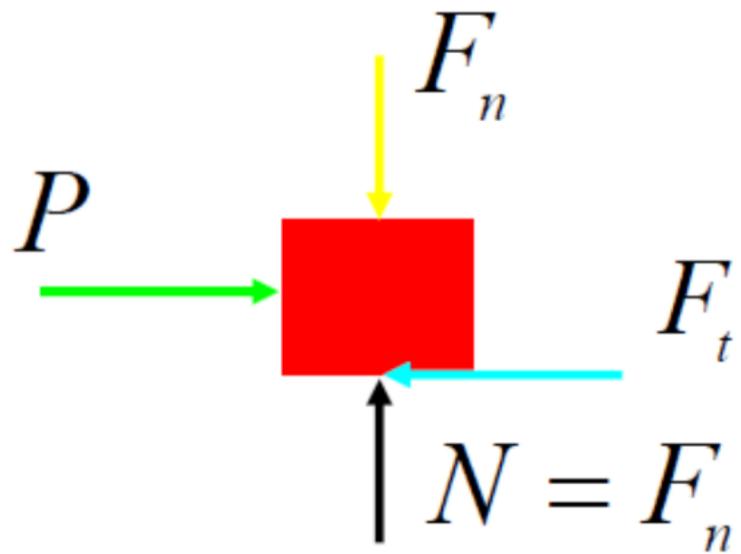


FIGURE 2.36: Friction diagram. The F_t is the friction force, going against the displacement direction

Mindlin-Deresiewicz: This theory refers to a pair of spheres in contact. Following the Hertz model, the tangential forces reduce the contact radius and a small slip occurs in a ring shape near the contact zone.

Case 1: F_n is constant, F_t increases (loading):

$$\delta_t = \frac{3(2-v)\mu F_n}{16Ga} \left[1 - \left(1 - \frac{F_t}{\mu F_n} \right)^{\frac{2}{3}} \right] \quad (2.19)$$

Case 2: F_n constant, F_t decreases (unloading):

$$\delta_t = \frac{3(2-v)\mu F_n}{16Ga} \left[2 \left(1 - \frac{F_t^* - F_t}{2\mu F_n} \right)^{\frac{2}{3}} - \left(1 - \frac{F_t}{\mu F_n} \right)^{\frac{1}{3}} - 1 \right] \quad (2.20)$$

Rolling friction^{2.37}:

This type of friction appears on a body due to its rotation on a surface.

$$\mu_r = \frac{e_r}{R} \quad (2.21)$$

Spinning friction^{2.38}:

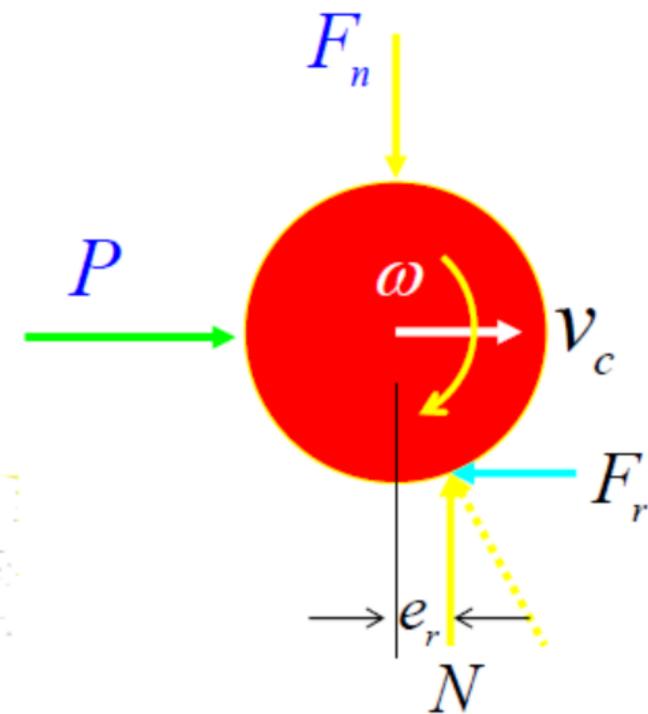


FIGURE 2.37: Rolling friction diagram

One of the main parts of the particle motion is their own spinning. If two particles in contact carry their relative rotation that type of friction is generated through the contact, which generates a momentum in the opposite direction of the relative rotation of the two spheres. Usually not taken in account in the DEM.

2.2.6 Dynamic equations

the task of the constitutive modelling is formulating the dynamic equations and the integration schemes governing the motion of the granular media. Once all the forces and actions happening in the particles are defined it is possible to establish the dynamic equations that, through an appropriate integration scheme, will describe the behaviour of the particles and give the information desired at any part of the simulation.

Dynamic equation assembly:

In order to build up the governing dynamic equation all the forces involved in the process have to be taken in account and properly organised. The general form of the governing dynamic equation will state that the sum of all the contact forces and the forces applied to the system has to be the same as the force due to the

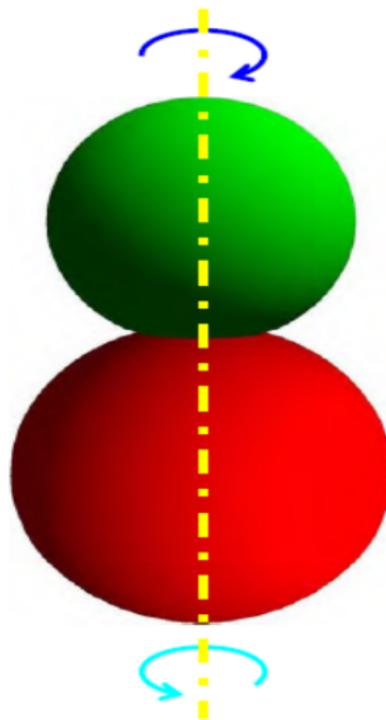


FIGURE 2.38: Spinning friction model

acceleration of a mass plus the damping effect.

$$F(U, \ddot{U}, t) = M\ddot{U} + C\dot{U} \quad (2.22)$$

This equation would be highly dynamic and non-linear and it has to be taken in account that the configuration would be constantly changing.

For the governing equation stated previously there is an specific integration scheme used in the DEM: the central difference.

$$\dot{U}_{(n + \frac{1}{2})} = \frac{(\dot{U}_{(n + 1)}) - \dot{U}_n}{\Delta t_n} \quad (2.23)$$

$$\dot{U}_{(n - \frac{1}{2})} = \frac{(\dot{U}_n) - \dot{U}_{(n - 1)}}{\Delta t_{(n - 1)}} \quad (2.24)$$

$$\Delta \bar{t}_n = \frac{\Delta t_n + \Delta t_{(n - 1)}}{2} \quad (2.25)$$

2nd order:

$$\ddot{U}_n = \frac{(\dot{U}_{(n + \frac{1}{2})}) - \dot{U}_{(n - \frac{1}{2})}}{\Delta t_n} \quad (2.26)$$

$$\dot{U}_n = \frac{(\dot{U}_{(n + \frac{1}{2})}) + \dot{U}_{(n - \frac{1}{2})}}{2} \quad (2.27)$$

Where:

U is the displacement, \dot{U} is the velocity, \ddot{U} is the acceleration, Δt is the time step and n indicates the time step position

The central difference scheme is part of the finite difference method family. It provides the optimization of the approximation for a differential operator and gives the numerical solution of that differential equation.

When it comes to simulate DEM problems, a very important matter is the time step selected. The reliability of the results and the time that the simulation would be running are directly depending on it. The time step has to be selected depending on the time discretization method used and according to the characteristics of the actual problem in order to apply a reasonable value. With the following equation for the central difference the critic time step can be computed for one contact pair and hence, the time step selected will be smaller than the critic value.

$$\Delta t_c r = 2 \sqrt{\frac{m}{k}} (\sqrt{1 + \xi^2} - \xi) \quad (2.28)$$

In case of more than one contact pairs, this equation has to be solved for all the pairs possible and select the critical value. The time step is a function of the mass, penalty parameter and the damping coefficient. The penalty parameter is a function of the young modulus and the radius of the particle. The time step obtained from those calculations would be the critical value, the biggest time step according to the present problem. In order to check the accuracy, a parameter called time step factor taking values between 0.1 and 0.9 will be multiplied into the critical time step to obtain a value with accuracy warranted.

There is an alternative way to compute an appropriate time step, it is called the Courant-Friedrichs-Lowy condition. This strategy will lead to the computation of the critical time step value depending directly on the diameter of the particles, their density, the young modulus and the poisons coefficient.

$$c = \sqrt{\frac{E}{\rho(1 - v^2)}} \quad (2.29)$$

Chapter 3

LIGGGHTS Open Source Code

3.1 LIGGGHTS Open Source Code

LIGGGHTS is an open source Discrete Element Method particle simulation software developed by Sandia National Labs. LIGGGHTS stands for LAMMPS Improved General Granular and Granular-heat transfer simulations, where LAMMPS is a molecular dynamics software simulator that preceded the creation of LIGGGHTS, an improved version that made the move from MD to DEM simulations by adding characteristic features from the method that cannot be found in the previous one such as the contact force formulation involving Hertz/Hooke pair styles, cohesion and rolling friction forces and heat conduction between particles.

Another LIGGGHTS's goal, besides approaching the DEM in a reliable and complete way, is to be used in industrial applications, that is the reason why it also includes the possibility of importing and handling complex CAD geometries, moving meshes to account for moving geometries and much more particle insertion options.

The LIGGGHTS code is written in C++ and can be run either in a single processor or in parallel. Its main script and the functions and libraries called by the main code are not going to be commented as there is not interaction with them in the development of the present work, but the input file is going to be explained in general basis, getting into more detail for each specific case simulated in this thesis.

3.1.1 Getting LIGGGHTS

As previously commented, LIGGGHTS is an open source code, which means that it is available to all for free. It can be downloaded from their website together with the whole LIGGGHTS package, including all the libraries, examples, benchmark problems and documents related to the code.

Starting from the beginning, a Unix based operative system is recommended to work with the LIGGGHTS code because, as most of the open source codes, is created and written in Unix machines. It is also important to take in account the fact of having to interact with the command prompt in order to install the software and run all its applications.

Thinking about the previous facts, this thesis has been developed in a Linux machine, with Ubuntu 12.04, in a laptop that features 8GB of RAM memory and a quad core processor at 3.3GHz. The LIGGGHTS version downloaded is the LIGGGHTS-dev package 3.0.1, actualised the 11th of April 2014. There is also the possibility of downloading the version LIGGGHTS, but the -dev package is prepared to run simulations in parallel, a useful feature definitely recommended to use.

After downloading the tarball from the website, the `liggghts-dev` executable has to be installed from the command prompt. Although it seems to be a simple installation, the libraries and updates required for the machine depend on the version of OS used and it can make the installation much more complicated, having to spend time checking different users forums in order to solve the problems encountered.

The LIGGGHTS website offers a post processing tool to visualize the results and movie of the simulation called Paraview. This tool is an open source data analysis and visualisation platform ready to show the results of particle simulations including forces, particle sizes, different types of velocities and temperatures.

Installing the tool Paraview can become as complex as the `liggghts-dev` process although it is done to be simply installed by calling the command Paraview from the command prompt. The LIGGGHTS code outputs a text file for every time step selected containing the information for every particle, but they are not ready to be read by Paraview yet. In order to visualize these output files in Paraview, they have to be transformed into VTK format through a special conversion tool called `pizza.py`. This conversor is a python tool that prepares the values for Paraview just by calling a simple function named `dump`. Once it is successfully installed the use is straight forward: the output files are transformed in the same folder

and Paraview can open them as one single file automatically giving the options to visualize all the variables selected as output (velocity, forces...).

3.1.2 LIGGGHTS input file

LIGGGHTS is executed by reading commands from a text input file one line at a time. Some commands may have default values while others have to be properly organised in order to use previously stated values. Organisation of the input file:

The LIGGGHTS input file does not follow a strict order, but it is true that there has to be some organisation for some commands to have enough information. The input file usually follows the logic order of going from defining general characteristics such as the atom type used, size of the simulation box to more specific ones like inserting geometries, defining material properties and how the particles are inserted. Obviously, the commands related to the output would be placed at the end of the text file. Here it is how a LIGGGHTS input file should look like:

3.1.2.1 Organization of the input file

A commented example of LIGGGHTS input file follows:

*atom*_{style} granular

*atom*_{modify} map array

boundary m m m

newton off

communicate single vel yes

units si

-This first part sets the basic general data of the simulation. For example, the type of atoms used, how are the imaginary boundaries, if the third newtons law is applied and the units which the data is going to be given in.

region reg block -1 1 -1 1 -1 1 units box

createbox 1 reg

neighbor 0.001 bin

neigh_modify delay 0

-The second part talks about general facts too, but getting a bit deeper. It sets the size of the simulation box and establishes the precision when it comes to establish possible contacts between particles.

fix m1 all property/global youngsModulus peratomtype 200.e9

fix m2 all property/global poissonsRatio peratomtype 0.45

fix m3 all property/global coefficientRestitution peratomtypepair 1 0.1

fix m4 all property/global coefficientFriction peratomtypepair 1 0.5

-This part sets the material properties

pair_style gran model hertz tangential history Hertzian without cohesion

pair_coeff * *

-The fourth part says which kind of interaction law is going to be applied in the simulation

timestep 0.000003

fix gravi all gravity 9.81 vector 0.0 0.0 -1.0

-Here the time step and the gravity are set.

fix cv all mesh/surface file meshes/conveyor.stl type 1 *surfacevel* 2. 0. 0.

fix wall1 all mesh/surface file meshes/wall1.stl type 1

fix wall2 all mesh/surface file meshes/wall2.stl type 1

fix wall3 all mesh/surface file meshes/wall3.stl type 1

fix inface all mesh/surface file meshes/*insertionface3.stl* type 1

fix wall all wall/gran model hertz tangential history mesh *nmeshes* 4 meshes cv wall1 wall2 wall3

-Getting into the physical nature of the simulation, all the walls and CAD geometries are introduced.

fix pts1 all particletemplate/sphere 1 *atomtype* 1 density constant 2500 radius constant 0.015


```
fix pdd1 all particledistribution/discrete 1. 1 pts1 1
```

-This part establishes the physical characteristics of the particles.

```
group nvegroup region reg
```

```
region bc block 0.1 0.3 -0.12 0.12 0.05 0.3 units box
```

-Here the area where the particles are introduced from is set.

```
fix ins nvegroup insert/stream seed 1 distributiontemplate pdd1 maxattempt 300
nparticles 15 particlerate 15 start 400 overlapcheck no vel constant 0. 0. -1.
insertionface inface extrudeLength 0.1
```

-This part defines how the insertion is going to be done.

```
fix integr nvegroup nve/sphere
```

```
fix ts all check/timestep/gran 1000 0.1 0.1
```

```
compute rke all erotate/sphere
```

-Insert the first particles so that first files are not empty

```
run 1
```

```
dump dmp all custom 400 post/dump*.conveyor id type type x y z ix iy iz vx vy
vz fx fy fz omegax omegay omegaz radius
```

```
dump dumpmesh all mesh/gran/VTK 400 post/dump*.vtk id cv inface wall1 wall2
wall3
```

-This last part outputs all the results of the simulation

```
insert particles
```

```
run 4000000 upto
```

3.1.2.2 Main LIGGGHTS commands

LIGGGHTS gives a lot options for all the aspects of the simulation, a huge amount of commands is available in the LIGGGHTS user documentation manual. Besides this fact, there are some commands that have to be always defined in a LIGGGHTS input file, and others that have been used remarkable times or have been significant in the different examples of the present work:

atom_style: This command defines the type of particles that is going to be used in the simulation and what attributes would they have. The most common option is a granular style, which would be defined by diameter, density and angular velocity.

atom_modify: change some characteristics of the atoms already stored

boundary: It determines how the imaginary simulation box is bounded. It can be either periodic, non periodic, fixed, skin wrapped.

Newton: determines whether the third Newton's law is taken in account during the simulation.

Communicate: Establishes if the simulation is going to run in a single processor or using multi processor.

Units: It defines the units that LIGGGHTS will assume the data is given in and the default values are set.

Region: This command creates a defined space that can take the shape of a rectangular box, sphere, cylinder or any basic geometric shape.

createbox: This command assigns the previous region to be the simulation box.

Neighbour: This command defines how the pairs have to be set in terms of distances and assigns them to a processor.

neigh_modify: It sets parameters that affect the use and creation of pairwise neighbour lists.

fix m1 all property/global: The fix command is the biggest command in LIGGGHTS. It can define several aspects of the simulation, that is the reason why it has to be always complemented with the group of atoms that is going to act on and the type of fix it is. In this case, the type of fix is defining material properties for the particles.

pair_style: One of the most important commands for LIGGGHTS, as it defines the interaction laws for the simulation, it can be hertz, hooke, with or without cohesion.

pair_coeff: It specifies the pairwise force field coefficient for the atom types.

Timestep: This command is in charge of setting the time step for the simulation.

Fix gravi all gravity: Another fix command, this time to establish the value and normal direction for the gravity force.

Fix cv all mesh/surface: Showing the wide variety of fix commands, this one inserts a mesh geometry into the simulation box.

fix wall all wall/gran: Once a wall or any geometry has been introduced it has to be defined as a granular wall to control the interaction with the particles.

fix pts1 all particletemplate/sphere: This fix command sets the type of particles with density and radius. One can define as many as desired.

fix pdd1 all particledistribution/discrete: After defining the different particle types, this command establishes how they are combined before inserting them in the box.

Group: It gathers particles from a region or type in order to use them all together further ahead on the simulation.

fix ins *nvegroup* insert/stream: This command establishes how the particles are inserted into the simulation box. It needs an insertion region and an inface to define the starting position of the particles and through where are they going to enter the simulation. It also sets initial conditions such as velocity and number of particles, insertion rate and attempts to perform the insertion.

fix integr *nvegroup* nve/sphere

fix ts all check/timestep/gran: This command informs you of the reliability of the time step selected, giving the relation between the actual time step and the young modulus or rayliegh time.

compute rke all erotate/sphere

run

dump: This command defines how many output files will there be, where are they going to be located and the information they are going to contain, such as position, forces, velocities.

dump dumpmesh all mesh/gran/VTK: This command allows the all the introduced geometries to appear in an output file in order to be visualised in the post processing tool together with the motion of the particles.

3.1.3 Running LIGGGHTS

Once the input file is completely defined, it is time to run the code. To complete that action the code has to be called with a specific command that includes

the name of the input file. LIGGGHTS has to be launched through the command prompt from the directory where the input file, involved meshes and output destinations are located.

Once the job has been submitted for run, the evolution of the computation can be followed in the same command prompt. First, it will show some of the input data such as domain of the simulation, imported meshes or number of particles. After that, the insertion should start. Until that point it is frequent to find some error messages launched by the same LIGGGHTS in the command prompt.

An input file is complex and full of precise data, so it is easy to forget some character or assign some wrong property. If this happens, LIGGGHTS will report the specific problem and stop running immediately. Once the error is known, one can either correct it if it is easy to locate or visit the error message chapter in the manual.

Supposing the particles are being inserted with no problem, the command prompt will show the state of the insertion: Current time step, number of atoms inserted and mass. Some warning messages can appear showing that, even if the program can run, some of the settings given in the input file has not been implemented perfectly. Again, one can locate the error or visit the warning messages chapter in the documentation.

The end of the computation will come when the time steps stop running in the command prompt. The information related to the mass inserted, number of particles and time steps run should appear together with the total number of neighbours created.

3.1.4 Post processing

LIGGGHTS produces, at the end of the simulation, an output file for every time step. All these output files can be found in the folder specified in the input file. Each file contains all the information of all atoms that appear in the present time step but, Paraview can only show one characteristic of the simulation with these output files, the position. All the other data such as velocities, forces, radius, atom types have to be transformed into VTK format in order to be visualised in Paraview. This action can be done by running in the command prompt a python wrapped application from the folder where the output files are. This application is offered in the LIGGGHTS website as a LIGGGHTS post processing tool. Again,

its installation can be complicated in the sense of the packages it requires to work properly.

3.1.4.1 Paraview

Paraview is an open source data analysis and visualization tool. It is basically a post processor capable of understanding the files that the LIGGGHTS post processing tool produces as explained previously. The LIGGGHTS simulation can be visualised, showing the whole movie of the process including surfaces and imported meshes, and the motion of the particles. During that visualisation, it is possible to see the different characteristics of the particles, for example, velocity, forces, radius, atom types, temperatures. Paraview offers the possibility of managing the visualisation parameters at any time in order to see the characteristics of the simulation with precision. As all the information is taken from the data files, it is also possible to see at every time step the data being read. All the results in this thesis have been obtained through that process and the images shown as results are saved screenshots from Paraview.

Chapter 4

SAND IMPACT ON SURFACE

4.1 Introduction

This test is going to talk about different basic aspects of the impact of sand into a surface with the aim of understanding the interaction between a rigid structure and a discrete media and how the properties of the granular material affect the behaviour taking in account different simplistic cases. It is a similar implementation to a paper on discrete element calculations of the impact of sand columns[1].

To do that, the implemented situation will be a planar squared surface just below a group of sand particles that are going to be inserted with a certain velocity during a specified time in order to fill the region. It is also important to notice that the sand is constrained laterally in both X and Y directions by walls placed at the same position of the planar surface edges.

Situation:

These tests are placed in a closed simulation box.[4.1](#) This domain has the shape of an empty block with the dimensions $0.4 \times 0.4 \times 1.5$ made of aluminium. At the bottom, the surface receiving the impact of the sand and at the top, the insertion volume where the particles are going to fall from. Notice that the XY face of the insertion volume is going to be smaller than the domain to avoid unexpected particle behaviors. These behaviours can occur if the insertion face coincides with the dimensions of the insertion region as some of the particles randomly generated can appear close enough to the domain walls to hit them.

The walls of the box are at the same time the no lateral spread boundary condition for the particles. The pressure in the bottom surface will be output in order to study its evolution as the time advances and the region is being filled. If a pressure wants to be obtained, the wall suffering it has to be a granular mesh surface. LIGGGHTS offers the possibility of establishing granular walls to design a region. The meaning of granular is that the wall is assigned an atom type as it is going to interact with the particles following the pair style laws with the data assigned to its material. But granular walls are not enough to output the the pressure. A surface with a triangular mesh is needed. This surface has to be created with an .stl file.

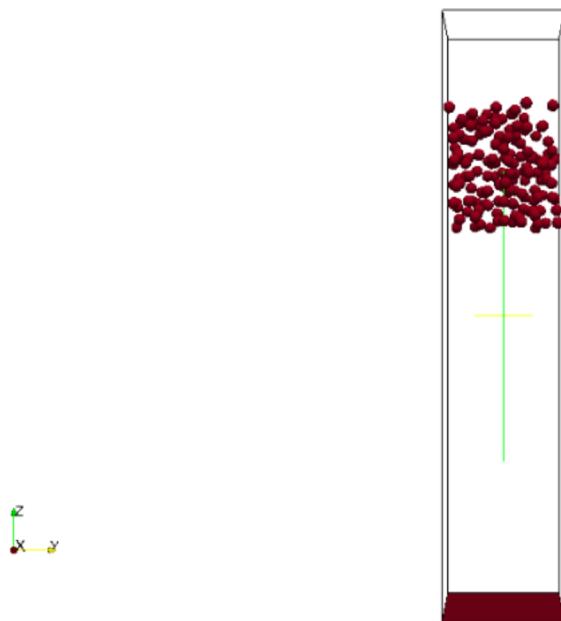


FIGURE 4.1: Domain of the simulation

4.2 Overview of the process

It is a simple simulation, an established number of particles is going to be inserted continuously from a certain height with respect to the target surface. Once the first particles reach the bottom, the observation has to start. The particles are going to stabilize on top of the surface, filling the whole domain and gaining height as more particles are being introduced. The main phenomena to observe in these cases will be the shock width.

The shock width is the distance between the particles that are already settled and the ones that still have the insertion velocity. The width depends on the characteristics of the materials and the simulation. In the following figure the

structure of the process is showed through 6 images that show the state of the simulation at 6 fixed time steps:

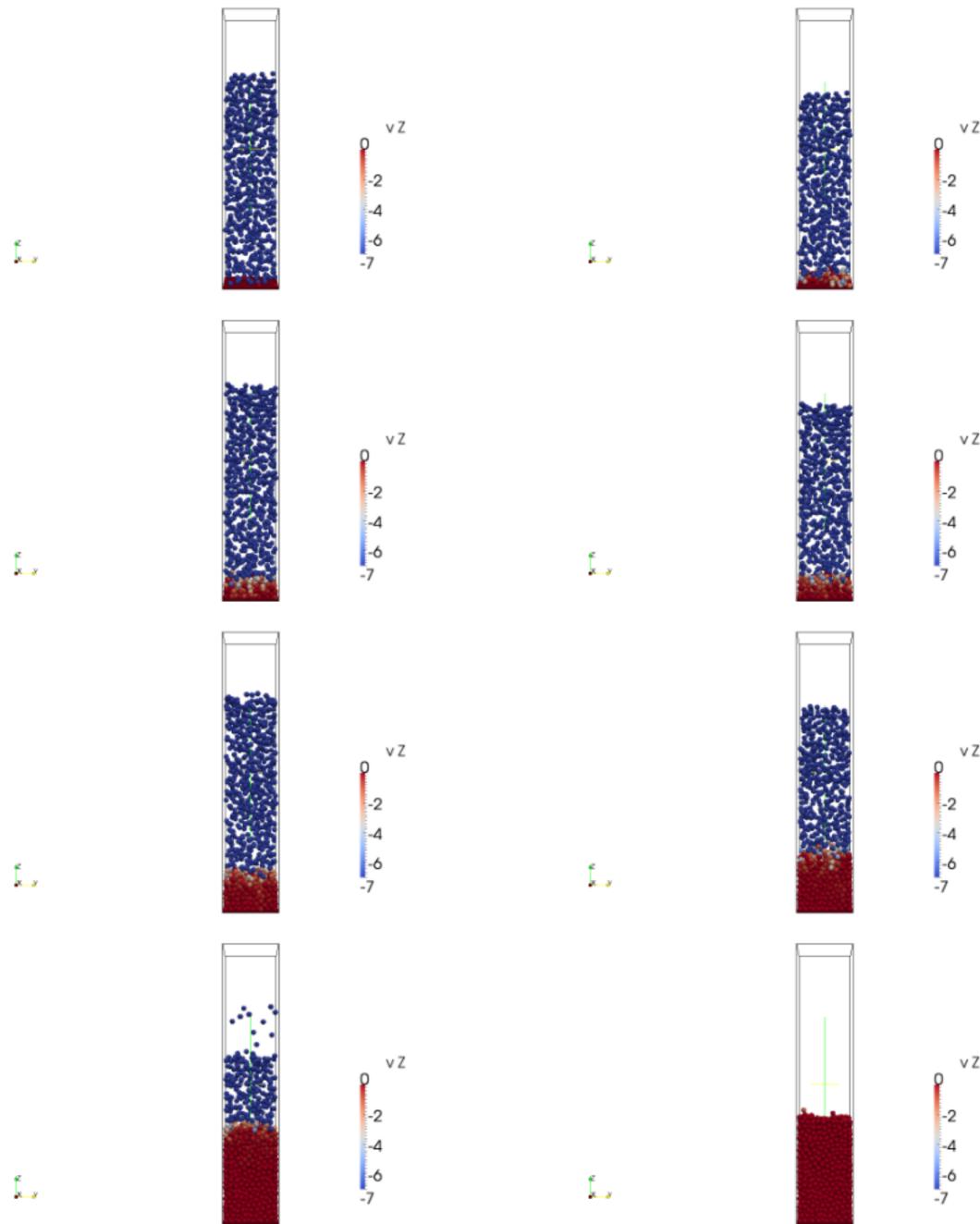


FIGURE 4.2: Overview of the process

For the surface, one of the goals of this document was to show the evolution of the pressure while the particles were being inserted as LIGGGHTS can output the pressure on a granular wall. In order to obtain it, the specified wall has to be build up by a triangular mesh through an STL file. These files create surfaces by

putting together organised triangular elements defined by 3 coordinates. These triangles are going to be the mesh elements.

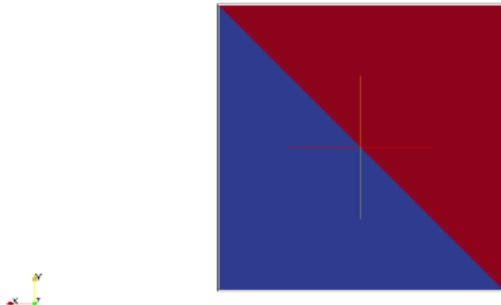


FIGURE 4.3: Bottom surface formed by two triangular mesh elements

4.3 Governing equations

The equations to be taken in account for all the problem cases in this chapter are given by LIGGGHTS and they depend mainly on the pair style used in the simulation specified by the pairstyle command. The equations being solved by LIGGGHTS can be found in the documentation section for pairstyle.

The main formula for the contact between two particles, besides the pair style, is as follows:

$$F = (k_n \delta n_{ij} - \gamma_n v n_{ij}) + (k_t \delta t_{ij} - \gamma_t v t_{ij}) \quad (4.1)$$

Where F is the force between the two particles, k_n is the elastic constant for normal contact, δ_n is the overlap distance between the two particles, γ_n is the viscoelastic damping constant for normal contact, γ_t is the viscoelastic damping constant for tangential contact, δ_t is the tangential displacement vector between 2 spherical particles truncated to satisfy a frictional yield criterion, v_n is the normal component of the relative velocity of the 2 particles and v_t is the tangential component of the relative velocity of the 2 particles.

The type of pair style used affects the computations having different formulas to compute the values stated above depending on the pair style selected. In all this chapter the pair style selected is the Hertz/history, the most common choice.

This style proposes the following values for the variables requested by the main equation:

Elastic constant for normal contact:

$$k_n = \frac{4}{3} Y^* \sqrt{R^* \delta_n} \quad (4.2)$$

Where Y^* is:

$$\frac{1}{Y^*} = \frac{(1 - v_1^2)}{Y_1} + \frac{(1 - v_2^2)}{Y_2} \quad (4.3)$$

Where Y is the Young's Modulus and v is the Poisson's ratio. The viscoelastic damping constant for normal contact stands:

$$\gamma_n = -2 \sqrt{\frac{5}{6}} \beta \sqrt{S_n m^*} > 0 \quad (4.4)$$

Where β :

$$\beta = \frac{\ln(e)}{\sqrt{\ln^2(e) + \pi^2}} \quad (4.5)$$

Where (e) is the coefficient of restitution. The elastic constant for tangential contact is:

$$k_t = 8G^* \sqrt{R^* \delta_n} \quad (4.6)$$

Where G^* is

$$\frac{1}{G^*} = \frac{2(2 + v_1)(1 - v_1)}{Y_1} + \frac{2(2 + v_2)(1 - v_2)}{Y_2} \quad (4.7)$$

And the viscoelastic damping constant for tangential contact is:

$$\gamma_t = -2 \sqrt{\frac{5}{6}} \beta \sqrt{S_t m^*} > 0 \quad (4.8)$$

and finally:

$$S_n = 2Y^* \sqrt{R^* \delta_n} \quad (4.9)$$

$$S_t = 8G^* \sqrt{R^* \delta_n} \quad (4.10)$$

4.4 Case 1

In this first case the impact is going to be caused by only sand particles of the same size.

4.4.1 LIGGGHTS input file

This section is going to show the simulation parameters and explain the main characteristics of the process through the first case input file:

atomstyle granular

atom_modify map array

boundary m m m

newton off

communicate single vel yes

units si

region reg block -0.2 0.2 -0.2 0.2 0.0 1.5 units box

createbox 3 reg

- The simulation domain has been created.

neighbor 0.002 bin

neigh_modify delay 0

'Material properties required for new pair styles'

fix m1 all property/global youngsModulus peratomtype 66.3e9 69e9

fix m2 all property/global poissonsRatio peratomtype 0.15 0.334

fix m3 all property/global coefficientRestitution peratomtypepair 2 0.2 0.1 0.1 0.3

fix m4 all property/global coefficientFriction peratomtypepair 2 0.7 0.6 0.6 0.7

fix m5 all property/global *kfinnie* peratomtypepair 2 1 1 1 1

- The material properties are defined, in this case for sand (particles) and aluminium (walls). There are two types of properties: Peratomtype and peratomtypepair. The first ones are referred to only one material and they are introduced in LIGGGHTS as scalar value. The second one involves two atom styles as this properties represent a relation between two materials. In this case the information has to be introduced in LIGGGHTS in a matrix form following the scheme: $P_{11}, P_{12}, P_{13}, P_{21}, \dots$ and so on until a matrix sized $n_{\text{materials}} \times n_{\text{materials}}$ is created. In this case, as aluminium is exclusively for the walls, the values of friction and

restitution refered to the relation aluminium-aluminium are insignificant because they have no interaction. It is, anyway, required for LIGGGHTS.

'New pair style'

pair style gran model hertz tangential history Hertzian without cohesion

paircoeff * *

- Pair style, used for the simulation.

timestep 0.00001

- Size of the time step.

fix gravi all gravity 9.81 vector 0.0 0.0 -1.0

'the walls'

fix inface all mesh/surface file meshes/inface.stl type 1

fix surface all mesh/surface/stress file meshes/surface.stl type 2 wear finnie

fix boxwall1 all wall/gran model hertz tangential history primitive type 2 xplane 0.2

fix boxwall2 all wall/gran model hertz tangential history primitive type 2 xplane -0.2

fix boxwall3 all wall/gran model hertz tangential history primitive type 2 yplane 0.2

fix boxwall4 all wall/gran model hertz tangential history primitive type 2 yplane -0.2

fix roof all mesh/surface/stress file meshes/roof.stl type 1

fix wall all wall/gran model hertz tangential history mesh n_{meshes} 2 meshes surface roof

- All the walls and surfaces are inserted in the simulation, especifying the material, the pair style and for which walls the pressure will be computed. Notice the selection of which surfaces act as granular walls is done.

'distributions for insertion'

fix pts1 all particletemplate/sphere 1 *atomtype* 1 density constant 2700 radius constant 0.02


```
fix pts2 all particletemplate/sphere 1 atomtype 3 density constant 1200 radius
constant 0.02
```

```
fix pdd1 all particedistribution/discrete 1. 2 pts1 0.5 pts2 0.5
```

- This section specifies the size and density of the particles and, in case of combining different particle styles, the inserted proportion of each particle has to be set too.

'region and insertion'

```
group nvegroup region reg
```

'particle insertion'

```
fix ins nvegroup insert/stream seed 5330 distributiontemplate pdd1 nparticles 2000
massrate 300 insertevery 5000 overlapcheck yes allin no vel constant 0.0 0.0 -8.0
insertionface inface
```

- this command sets the insertion. Having previously selected the particle distribution, the number of particles and the rate of insertion have to be set. In this case 2000 thousand particles are going to be inserted at a pace of 300 Kg each insertion. It is also defined that the particle insertion happens every 5000 time steps. The particles are inserted at a velocity of -8m/s in the Z axis from the surface created exclusively to serve as an insertion face.

'apply nve integration to all particles that are inserted as single particles'

```
fix integr nvegroup nve/sphere
```

'output settings, include total thermal energy'

```
compute 1 all erotate/sphere
```

```
thermo style custom step atoms ke c1 vol
```

```
thermo 1000
```

```
thermo modify lost ignore norm no
```

```
compute modify thermotemp dynamic yes
```

'insert the first particles so that dump is not empty'

```
run 1
```

```
dump dmp all custom 200 post2/dump*.chute id type type x y z ix iy iz vx vy vz
fx fy fz omegax omegay omegaz radius
```

dump dumpstress all mesh/gran/VTK 200 post2/dump*.vtk stress surface

- The dump command specifies how often an output file is going to be produced and what it will contain. In this case, 2 types of files are going to be dump every 200 time steps. The dump type file will contain all the particle information while the vtk file wraps the information of the granular walls and enables them to be visualised in Paraview.

'insert particles'

run 100000 upto

unfix ins

The figures obtained^{4.4} show the state of the simulation at the timestep 120 but with different values of restitution coefficient.

From the figures it can be seen that the bigger the coefficient of restitution is, the less relaxed the particles are. This coefficient establishes the relation between the velocity before and after an impact, the bigger the coefficient, the more velocity the particle would have when it comes to bounce back. When the restitution is low, particles settle quickly at the bottom of the domain and on top of each other, this is why it can be appreciated an increase in the shock width together with e . More precisely, the number of particles with velocity 0 in the $e=0.1$ is much bigger than $e=0.9$. Inversely, the number of particles with a velocity between -1 and -6 is bigger in the $e=0.9$ case, which means a wider shock width.

The same behaviour can be observed when the modified value is the friction coefficient. The following figure shows, for the same time step, the aspect of the process with 4 different values of friction^{4.5}:

For a low coefficient of friction, more energy will be conserved in form of velocity, this is why the bigger the friction, the more settled particles in the bottom of the domain. In this case, the difference is not so evident, what leads to the conclusion that friction does not have the same influence as restitution.

It has not been possible to study the pressure behaviour due to the data showed by Paraview. Although the final value for the pressure coincides with the analytical solution, observing the pressure values at the bottom surface timestep by timestep reveals that the pressure shown during the insertion does not follow a reasonable evolution. The values fluctuate from the first contact until all the particles are settled between values that can go from 0 to three times the expected maximum pressure at the end of the process.

The cause of these unexpected values is the way the pressure is computed originally in LIGGGHTS, could be due to the small number of elements in the surface mesh or the big change on the particle momentum at the impact moment.

The idea behind that was to show the Janssen effect[2], a well known behaviour in granular materials. It accounts for situations where there is a packing with the force of gravity inside a container. In terms of observing the pressure at the bottom of the specified volume, it would be reasonable to think that the more the height of filling, the more pressure will be applied into the surface, but the Janssen effect shows why this statement is not true:

While filling a cylinder, the pressure at the bottom will increase until reaching a saturation level when, even if the height of the particles in the container keeps increasing, the pressure will stop growing. This is due to the friction forces between the particles and the walls of the cylinder, that removes part of the downward force to transform it into friction force.

4.5 Case 2

Now the behaviour of the previous situation is going to be studied with particles of the same material, sand, but two different sizes. The goal is to understand how this difference affects the basic behaviour observed in case 1, modifying the restitution coefficient. The new sizes are going to be 0.015 and 0.03.

In terms of the input file, there is only one setting to be changed: the particle distribution.

'distributions for insertion'

```
fix pts1 all particletemplate/sphere 1 atomtype 1 density constant 2700 radius
constant 0.03
```

```
fix pts2 all particletemplate/sphere 1 atomtype 1 density constant 2700 radius
constant 0.015
```

```
fix pdd1 all particledistribution/discrete 1. 2 pts1 0.5 pts2 0.5
```

The behaviour observed 4.6 is really similar than the case with the equal sized particles. The shock width follows the same pattern but the size is not the same. This is due to having the same number of particles but occupying less surface, which means a slower filling of the domain.

4.6 Case 3

The last case is going to implement two different granular materials as particles, with different characteristics. Half of the introduced particles are going to be made of rubber. The choice of rubber has been made in order to have a completely different material from sand and at the same time computationally cheap with a value of 0.01e9 for the young modulus. The aim is to observe how two kind of particles affect the situation when they are inserted together with the same mass proportion and how the possibility of each material having completely altered and opposite restitution coefficients can affect the filling.

In the LIGGGHTS input file, two things have to be taken in account: The material properties will have now one more value that will have to be set in the matrix form especified.

'Material properties required for new pair styles'

```
fix m1 all property/global youngsModulus peratomtype 66.3e9 69e9 0.01e9
fix m2 all property/global poissonsRatio peratomtype 0.15 0.334 0.49
fix m3 all property/global coefficientRestitution peratomtypepair 3 0.2 0.1 0.2 0.1
0.1 0.2 0.2 0.2 0.3
fix m4 all property/global coefficientFriction peratomtypepair 3 0.7 0.6 0.9 0.6 0.6
0.3 0.9 0.3 1.16
fix m5 all property/global  $k_f$ innie peratomtypepair 3 1 1 1 1 1 1 1 1 1
```

And when the particles are being defined, material 3 has to be assigned to one part of the particles changing the density: 'distributions for insertion' fix pts1 all particletemplate/sphere 1 $atom_type$ 1 density constant 2700 radius constant 0.03 fix pts2 all particletemplate/sphere 1 $atom_type$ 3 density constant 1200 radius constant 0.015 fix pdd1 all particledistribution/discrete 1. 2 pts1 0.5 pts2 0.5

The figure shows, at 4 different time steps, the disposition of the particles with the restitution coefficients by default [4.7](#).

The behaviour is as expected, following the same deposition form as the previous cases. If we change the values of the coefficient of restitution affecting the contact with the same material and we set a completely different value for the two materials of the particles the LIGGGHTS matrices will be as follows:

'Material properties required for new pair styles'


```
fix m1 all property/global youngsModulus peratomtype 66.3e9 69e9 0.01e9
fix m2 all property/global poissonsRatio peratomtype 0.15 0.334 0.49
fix m3 all property/global coefficientRestitution peratomtypepair 3 0.9 0.1 0.2 0.1
0.1 0.2 0.2 0.2 0.1
fix m4 all property/global coefficientFriction peratomtypepair 3 0.7 0.6 0.9 0.6 0.6
0.3 0.9 0.3 1.16
fix m5 all property/global  $k_{finnie}$  peratomtypepair 3 1 1 1 1 1 1 1 1 1
```

The e value for sand-sand has been changed to 0.9, which means that the velocity after the impact is going to be 0.9 times the previous one. For the rubber-rubber restitution 0.1 has been set, meaning that the impact between rubber particles would remove 90 percent of the energy of the particles. The plots are^{4.8}:

From all the plots studied almost no variations can be observed. Even if for each case one material is set with a big restitution and the other material with a really small one, the behaviour is similar to the default situation. The big difference in the coefficients is balanced automatically by the fact that a particle has the same probabilities of experimenting impacts with both materials, hence, with long or short bounce back.

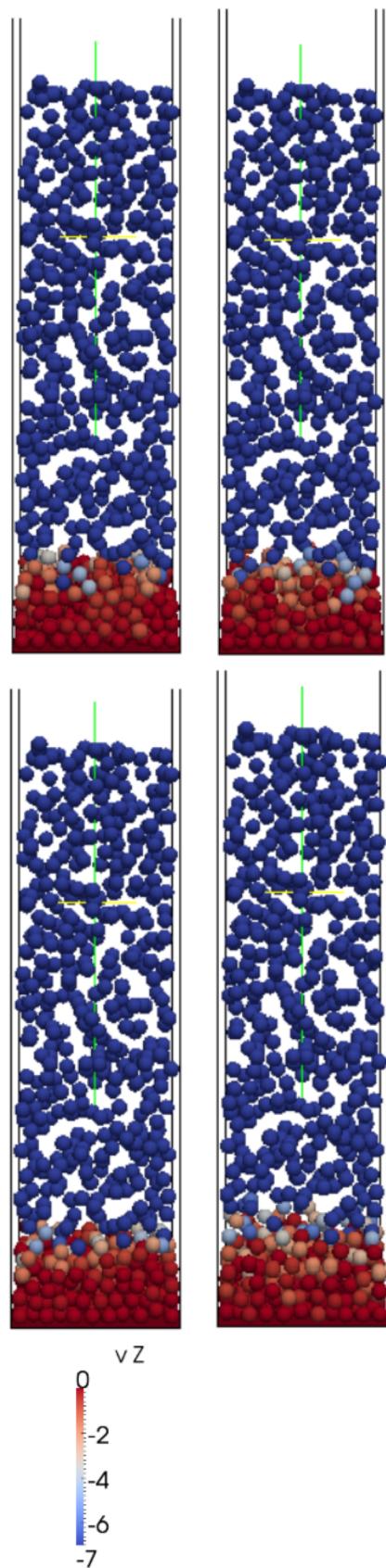


FIGURE 4.4: Distribution of Velocities at timestep 120. From top to bottom, left to right: $e=0.1$, $e=0.3$, $e=0.6$ and $e=0.9$

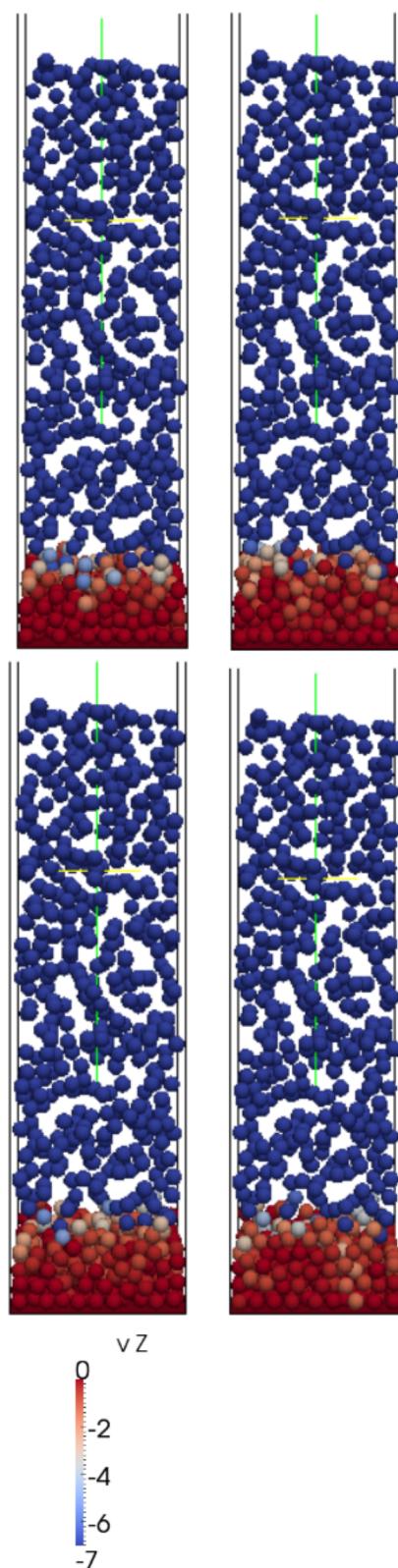


FIGURE 4.5: Distribution of Velocities at timestep 120. From top to bottom, left to right: $f=0.1$, $f=0.3$, $f=0.6$ and $f=0.9$

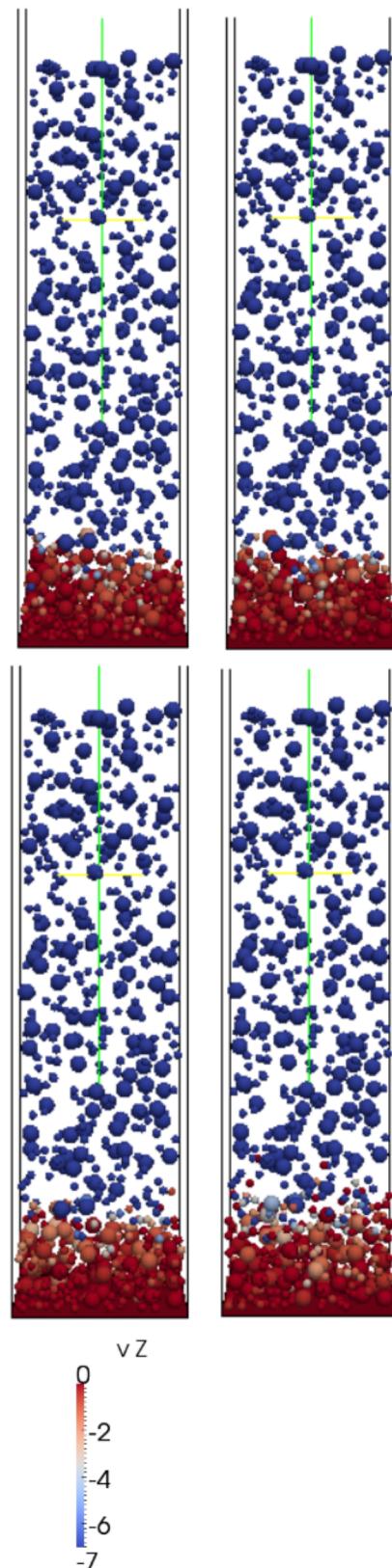


FIGURE 4.6: Distribution of Velocities at timestep 120. From top to bottom, left to right: $f=0.1$, $f=0.3$, $f=0.6$ and $f=0.9$

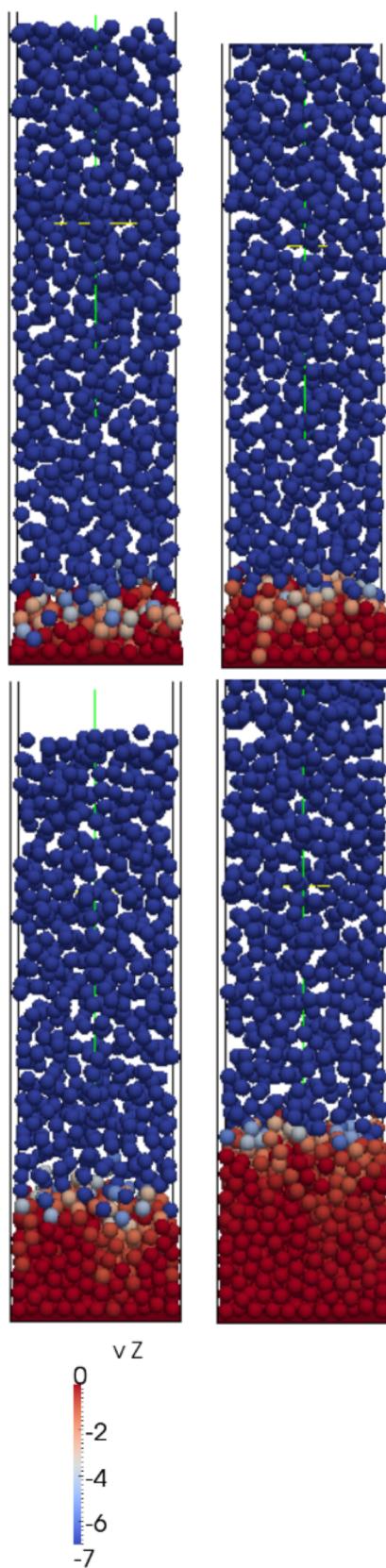


FIGURE 4.7: Distribution of Velocities. From top to bottom, left to right:
 $ts=90, 100, 120$ and 150

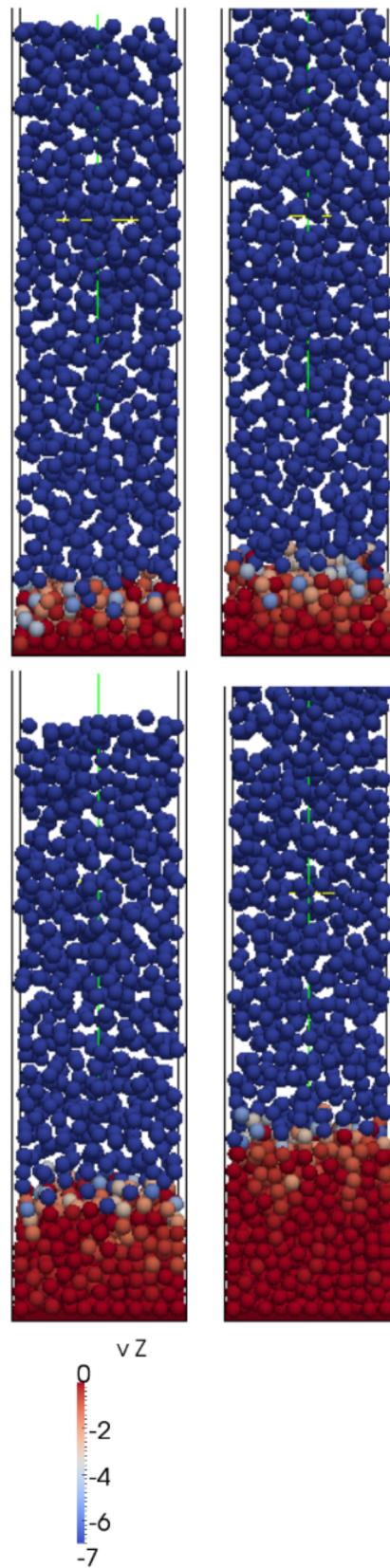


FIGURE 4.8: Distribution of Velocities. From top to bottom, left to right:
 $ts=90, 100, 120$ and 150

Chapter 5

Cohesion

5.1 Introduction

Once the basics of the discrete element method have been discussed through different experiments to test the influence of the most important actions for the DEM, it is the moment to talk about another feature of this method, not complicated to understand and demonstrate and at the same time is one of the most important interaction laws for granular materials:

5.1.1 Cohesion model

Cohesion is the component of shear strength of a granular material. Its existence between particles cause tendency to resist separation if they are from the same material. This phenomena is a big influence in the motion of powders as it affects the three parts of their behaviour: static, quasi static and the flow regimes. The most evident example of that phenomena is the water drops.

5.1.2 Overview of the process

An effective way to understand the particle interaction laws is to place them inside a rotating drum. There is gravity, velocity and a lot of contacts. In this case particles from the same material but different size are going to be set inside a rotating drum. The idea is to understand the behaviour of the particles with the motion and see the effect of cohesion by increasing or decreasing it in the particles

or walls and keeping the other parameters constant. The influence of cohesion on the simulation will be the variations on the motion of the material.

Situation:

Following the process from the previous tests, the first step is to set the domain. In that case, as the simulation includes an object(the drum), the domain will be as close to the shape of the object as possible. The actual object is a rotating drum, what means that the domain is going to have the shape of a box with the smallest size possible for every dimension depending on the drum body.

Precisely the object is represented as a cylinder, ring or a part of a tube, to make the geometry import and the triangular mesh from cad much easier. The walls enclosing the drum are going to be set in LIGGGHTS as granular walls and will be independent of the movement of the drum.

The particles will be inserted at the beginning of the simulation in the middle of the inside of the drum, while it is already on rotation. This fact will cause some dispersion in the particles at an early stage of the simulation, but such behaviour is not going to be considered.

5.1.3 Governing Equations

In terms of LIGGGHTs, the cohesion model implemented is quite simplistic but yet intuitive. It implements the linear cohesion model, which means that the cohesion force is going to act during the contacts following the equation:

$$F = KA \quad (5.1)$$

where F is the Force, A is the contact area and K is the cohesion energy density.

The rest of the Hertz contact model is going to be applied again with the same equations than the previous chapter of this thesis with the addition of the cohesion force in the normal forces of the contact, trying to retain the contact.

5.1.4 LIGGGHTS input file

Activating the option of a pair style with cohesion in the input file is simple:

'Material properties required for new pair styles'


```

fix m1 all property/global youngsModulus peratomtype 66.3e9 69e9
fix m2 all property/global poissonsRatio peratomtype 0.15 0.334
fix m3 all property/global coefficientRestitution peratomtypepair 2 0.2 0.1 0.1 0.3
fix m4 all property/global coefficientFriction peratomtypepair 2 0.7 0.6 0.6 0.7
fix m5 all property/global  $k_{finnie}$  peratomtypepair 2 1 1 1 1
fix m6 all property/global Densitycohesion peratomtypepair 2 1000 2000 2000 6000

```

- First of all, the density cohesion has to appear as a property of the materials in the simulation.

'New pair style'

pairstyle gran model hertz tangential history cohesion skjr

-Finally, the word cohesion has to present in the pair style command.

5.1.5 Models

In the first case the basic behaviour of the cohesion is going to be studied by comparing the results of the simulation with the same materials and settings just activating cohesion against the process without cohesion. The other data in these models is for a non real material: A Young modulus of 5e6Pa, Poisson ratio 0.45, a coefficient of restitution of 0.3, friction of 0.7 and a density of 1200 $\frac{kg}{m^3}$

Development of the particle motion inside the drum. The model without cohesion:[5.1](#):

The behaviour is as expected. The media contains 1 atom type but two different sizes, what makes the small particles move downwards. The motion follows the direction of the velocity, going up the wall but, due to its granular condition without cohesion and the effect of gravity in the particles that are on top of the media drag a lot of particles to the lowest part of the drum, the starting point. This situation is stable as far as the velocity is maintained.

Now, the motion with cohesion with a value of surface energy density (k) of 100000 [5.2](#):

It can be observed that, due to the cohesion, the material behaves less as a granular one and more as a thick fluid. The material sticks more to the walls until reaching

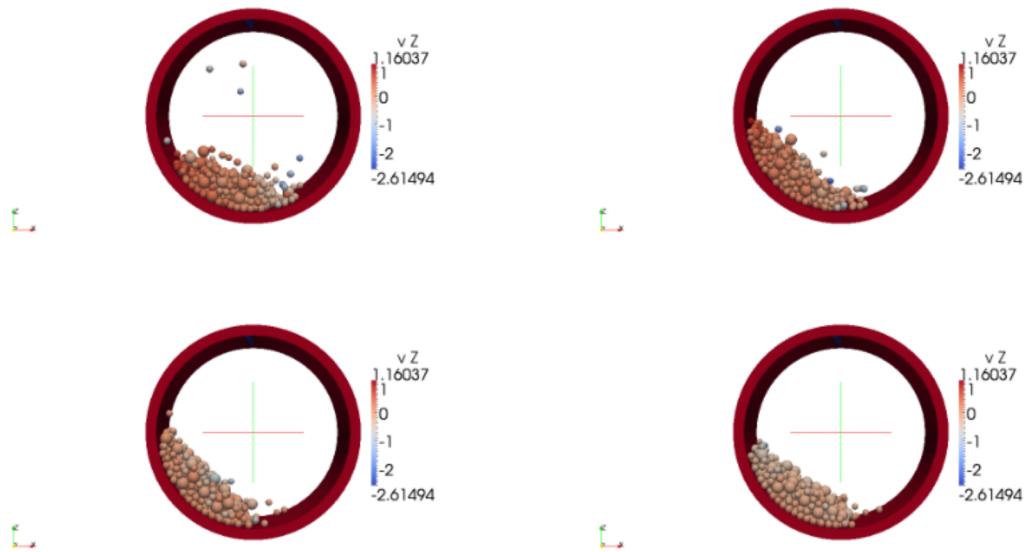


FIGURE 5.1: Motion of the granular material at different time steps. From top to bottom, left to right: ts=230, ts=400, ts=550 and ts=700

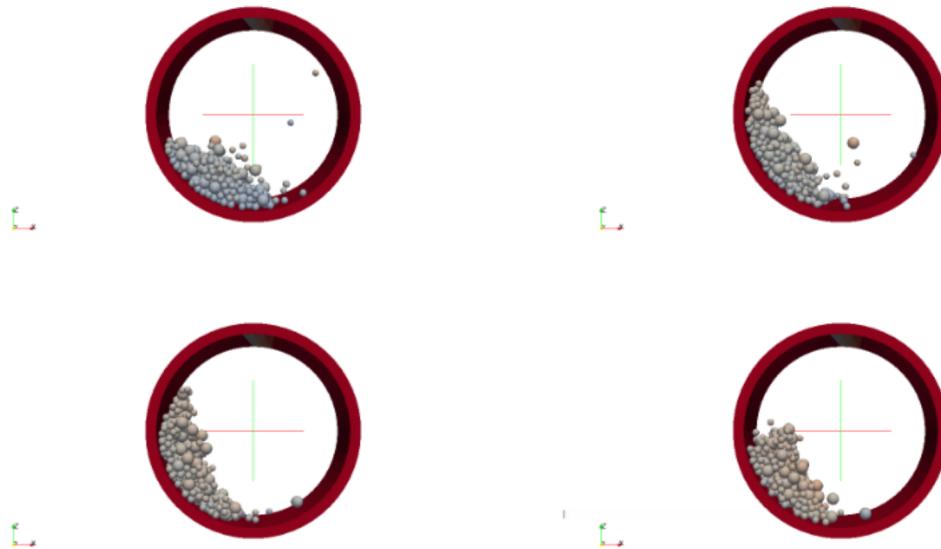


FIGURE 5.2: Motion of the granular material with cohesion at different time steps. From top to bottom, left to right: ts=230, ts=400, ts=550 and ts=700

a higher point than the no cohesion case. At this point, the granular mixture falls due to the effect of gravity.

Effects of cohesion increases as the velocity increases too. In the following case the material is stuck to the wall through the same value for cohesion, but now the

drum is spinning faster, what will force the material to move faster too, as it can be seen here [5.3](#):

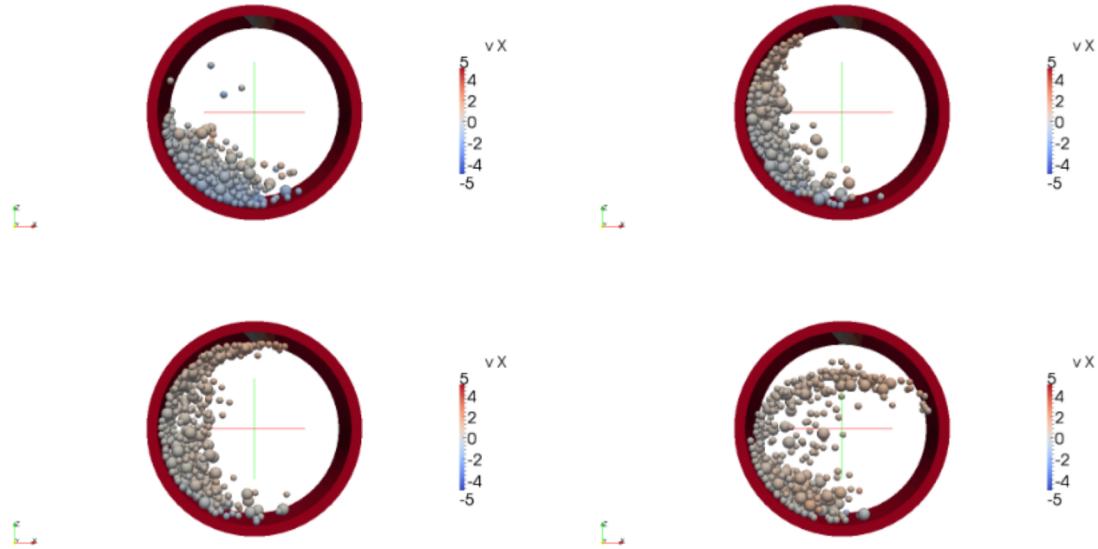


FIGURE 5.3: Motion of the granular material with cohesion at 14 rad/s. From top to bottom, left to right: ts=230, ts=400, ts=550 and ts=700

Finally, the behaviour of a granular material with high cohesion inside a drum with a extremely low cohesion with the material contained is showed. The granular material acts like 1 mass and, against a surface with velocity there is almost no sticking[5.4](#):

The pictures show how the height level of the particles is the lowest one among all the cases studied. The value of cohesion with the wall is really low as other cases seen in this chapter, but the high cohesion of the granular material forces the particles to stay together and makes the drag smaller.

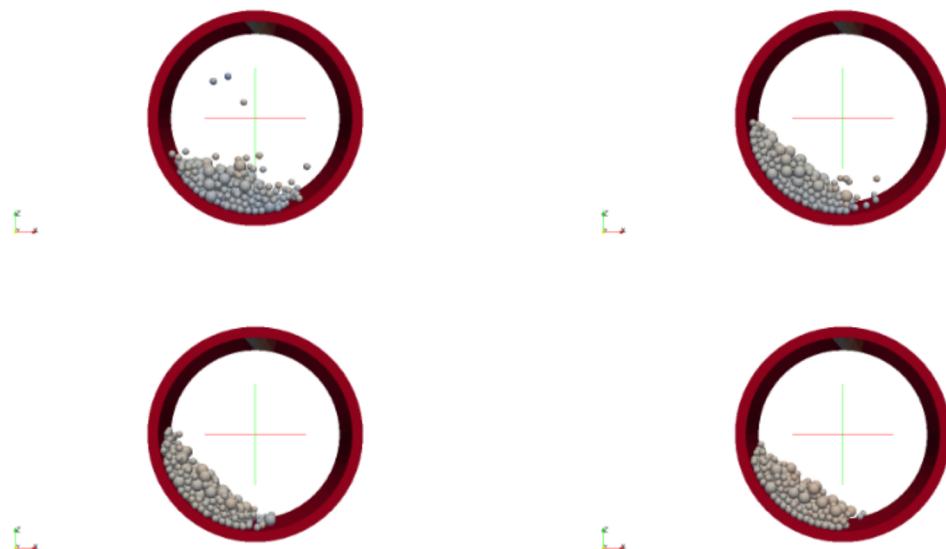


FIGURE 5.4: Motion of the granular material with extremely low value of cohesion with the wall material. From top to bottom, left to right: $ts=230$, $ts=400$, $ts=550$ and $ts=700$

Chapter 6

Conclusions

The main motivation behind this thesis was the possibility to work on a new method that opens the doors to a new field in the computational mechanics. Another initial point was the idea of approaching the method from a practical scope, giving the simulations the majority of the weight of this thesis.

Due to the fact that DEM is not a widely established method, this thesis pursued the implementation of simple cases that could be easily modelled by DEM with the help of an open source code special for particle simulations called LIGGGHTS. According to that, the literature review was mainly related to simulations of granular materials impacting different targets [3–5]. The long term goal of this project was to take the cases implemented in the literature a bit further, especially in the DEM/FEM coupling aspect.

The objectives changed due to the unexpected slow pace to the understanding of the code. The focus was put over the DEM itself, with cases where basic settings or conditions of the simulation could lead to surprising results. It was the case, for instance, of a paper modelling a gas-granular jet impacting a surface at a high velocity and obtaining different relaxation in the parcel material once the particles hit the surface. Unfortunately, the conditions of the gas-particle jet required a superior understanding of the open source code due to the multi-state condition.

The Janssen effect (previously explained in this thesis): The examples introduced by Yasinul Karim and Eric Corwin [6] have been considered to study. Karim et. al. studied a conveyor belt transporting coins with friction in the conveyor and the walls but, if the pressure created by the particles is measured in the direction

of the conveyor by putting a wall, it can be observed that the belt vibration removes the inter-coin tensions, which means no Janssen effect in force driven packing systems. However, their experimentations with such systems leaded to the following statement "Erasing friction with friction". Incrementing massively the coin-wall friction setting up a sawtooth in the lateral walls makes the vibration of the conveyor belt have no effect on the coins. The decision to implement this as part of the thesis was taken, but the model obtained in LIGGGHTS was not realistic enough, requiring more time to adjust the model and take advantage of the open source concept to build a reliable simulation.

Still with the idea of working in the Jansen effect, basic cylinder filling cases were implemented in order to prove this phenomena under gravitational effects, but the understanding of the post processing tool computing techniques has not been enough, requiring more time to learn how the pressure values are obtained.

Simple granular material impact experiments have been implemented, far from the first objectives. However, they have been a good tool to understand the basics of the DEM methodology and a demonstration of the powerful capabilities of LIGGGHTS and Paraview, setting the basis for, with a deeper focusing, reach even further than the challenges in this thesis.

Bibliography

- [1] H.N.G. Wadley V.S. Desphande S.M. Pingle, N.A. Fleck. Discrete element calculations of the impact of a sand column against rigid structures. *International Journal of Impact Engineering*, 139(45):74–89, December 2011.
- [2] Jean-Pierre Hulin Yann Bertho, Frederique Giorgiutti-Dauphine. Dynamical janssen effect on granular packing with moving walls. *Physical Review Letters*, 90(14), April 2003.
- [3] H.N.G. Wadley-V.S. Desphande T. Liu, N.A. Fleck. The impact of sand slugs against beams and plates: Coupled discrete particle/finite element simulations. *Journal of the Mechanics and Physics of Solids*, 24(61):1798–1821, April 2013.
- [4] Tao Liu Vikram S. Desphande Kumar P. Dharmasena, Haydn N.G.Wadley. The dynamic response of edge clamped plates loaded by spherically expanding sand shells. *International Journal of Impact Engineering*, 112(62):182–195, July 2013.
- [5] V.S. Desphande T. Liu, H.N.G. Wadley. Dynamic compression of foam supported plates impacted by high velocity soil. *International Journal of Impact Engineering*, 126(63):88–105, August 2013.
- [6] M. Yasinul Karim and Eric I. Corwin. Eliminating friction with friction: 2d janssen effect in a friction-driven system. *Physical Review Letters*, 112(112):188001, May 2014.

