A new simulation tool for the predictive assessment of fluidized and circulatory granular flow behaviour

Arnón López Marrero

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

March, 2020

Contents

List of Figures

List of Tables

Nomenclature

Bulk Properties

ϵ Void fraction, see equation (3.1), page 18

μ Internal coefficient of friction, see equation (3.7), page 21

ϕ_s Sphericity, see equation (3.6), page 20

ρ_b Bulk density, see equation (3.2), page 18

ρ_f Density of the fluid, see equation (3.5), page 19

ρ_p Density of the solid, see equation (3.5), page 19

σ Normal load applied on the column of powder, see equation (3.7), page 21

τ_s Shear strength, see equation (3.7), page 21

T_a Tensile strength, see equation (3.7), page 21

DEM

$(\Delta n_A)_{t2}$ Relative displacement increments at contact point A, see equation (3.15), page 28

$(\Delta n_B)_{t2}$ Relative displacement increments at contact point B, see equation (3.16), page 29

$(\Delta n_C)_{t2}$ Relative displacement increments at contact point C, see equation (3.17), page 29

$\ddot{x}$ Acceleration disc X, see equation (3.13), page 27

$\ddot{y}$ Acceleration disc Y, see equation (3.13), page 27

$\Delta \vec{F_n}$ Increment in normal force, see equation (3.11), page 27

$\Delta \vec{n}$ Overlap on disc, see equation (3.10), page 26

Δn Normal components of the relative displacement, see equation (3.23), page 31

Δs Tangential components of the relative displacement, see equation (3.24), page 32

Δt Time increment, see equation (3.10), page 26

$\Delta \vec{F}_n$ Increments of the normal force, see equation (3.25), page 32

$\Delta \vec{F}_s$ Increments of the tangential force, see equation (3.26), page 32

$\dot{\theta}$ Angular velocity, see equation (3.17), page 30

$\dot{n}$ Normal components of the relative velocity, see equation (3.21), page 31

$\dot{s}$ Tangential components of the relative velocity, see equation (3.22), page 31

$\dot{x}$ Velocity disc X, see equation (3.14), page 28

$\dot{y}$ Velocity disc Y, see equation (3.14), page 28

κ_n Normal stiffness, see equation (3.11), page 27

$\vec{v}$ Velocity of the wall, see equation (3.10), page 26

m_x Mass disc X, see equation (3.13), page 27

m_y Mass disc Y, see equation (3.13), page 27

Experiments

ϵ Strain, see equation (4.1), page 40

μ Coefficient of Static Friction, see equation (4.3), page 45

ν Poisson's Ratio, see equation (4.0), page 40

σ_n Normal stress, see equation (4.0), page 40

τ Shearing stress, see equation (4.0), page 40

E Modulus of Elasticity, see equation (4.0), page 39

U Velocity of approachn, see equation (4.4), page 52

v Velocity of departure (rebound), see equation (4.4), page 52 e Coefficient of Restitution, see equation (4.3), page 49

Stress

S_n Stress normal to the surface, see equation (7.7), page 159

S_{tc} Stress tangential to the contour, see equation (7.7), page 159

S_{ty} Stress tangential to the y-axis direction, see equation (7.7), page 159

u_n Normal unit vector, see equation (7.3), page 152

u_t Tangential unit vector, see equation (7.4), page 152

Acronyms

AFM Abrasive Flow Machining.

AM Additive Manufacturing.

AoR Angle of Repose.

CFD Computational Fluid Dynamic.

CoM Center of Mass.

DEM Discrete Element Method.

DMT Derjaguin-Muller-Toporovs.

EPSD Elastic-Plastic Spring-Dashpot.

EPSRC Engineering and Physical Sciences Research Council.

FEA Finite Element Analysis.

FEM Finite Element Method.

HPC High-Performance Computing.

JKR Johnson-Kendall-Roberts.

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

MAF Magnetic Abrasive Finishing.

MD Molecular Dynamics.

MF Mass Finishing.

MMR Rate of Metal Removal.

PIV Particle Image Velocimetry.

Acknowledgement

Firstly, I want to thank Professor Michael Morgan for trusting me to carry out his project. Without him this would not have been possible. I also want to express my gratitude to Dr. Mikdam Jamal and Dr. Daniel Peavoy to have gently opened the MTC's door to me and granted me access to the HPC.

I take this opportunity to specially mention Dr. Matteo Villa for always pointing me in the right direction. Especially in those moments when I was a drifting ship.

Thanks to Dr. Mehdi Seddighi and Khaled Takrouri for giving me access to the LJMU's HPC and letting me use their resources to finish this work.

To the *Nasa Team*: Dr. Xiaoxiao Liu, Mr. *Professor* Peter Moran and Mr. Alan Smith. Thanks guys for the infinite times you supported me and all the laughs we have had in these three years.

This acknowledgement would not be such an acknowledgement if I did not mention my friends Carl Rusthon, Jonny Reid and Simon Montgomery who on so many occasions have been there for me, making my life in Liverpool more interesting and diverse. Cheers mates!

María García also deserves a line in this section for being such a good person and for loving me nearly as much as I love her.

To Dr. Luis Padrón from The University of Las Palmas de Gran Canaria: Thank you for the light of your knowledge that so many times has taken me out of the darkness of my ignorance.

I want to thank my parents Elena Marrero and Emilio López for their successes but also for their mistakes, and for their endless lack of selfishness. To my sister and brother, Idriss and Sidi because I am not One without Two.

Lastly, to Alba for being honest, for being brave, for being sensible, for disagreeing with me, for accepting me, for not giving up, for loving me. *Gracias*.

Abstract

The Mass Finishing (MF) technologies are an area within the manufacturing processes that have been poorly investigated. The flow behaviour of the media within the work chamber is complex to establish. Deriving predictive process models or developing strategies for optimisation remains challenging. This is mainly because of the difficulty of measuring the phenomenon without altering it. Thus, there are many unanswered questions concerning the efficiency and capability of the MF technologies and uncertainty concerning the actions by which to optimise these processes. The work presented in this thesis employs a numerical approach, the Discrete Element Method (DEM) to shed light on these matters. This method has successfully revealed the mechanisms of the media flow within the rotary disc finisher and the importance of the friction between the container and the media. Furthermore, an in depth analysis of the interaction between particles and a fixed specimen within the bulk was achieved. This analysis predicts the behaviour of the stresses beyond the surfaces of the specimen and what features of the specimen are more likely to be affected by the media impact. Moreover, a full defined set of experiment is discussed as a means to obtain all the mechanical properties involve in the DEM model. A complete study of the movement governing the vibratory trough is also presented on this thesis.

Chapter 1

Introduction

This PhD proposal is framed within the Engineering and Physical Sciences Research Council (EPSRC) - Industry funded project ‘Process Design for Next-Generation Mass Finishing Technologies’ EP/N022998, led by Professor Michael Morgan.

1.1 Mass finishing brief introduction

MF has not been explored for a very long time and as a result it is a relatively unknown technology. Fortunately, at the moment, it is a rapidly developing surface refinement process and technology. Interest in MF is accelerating globally with the continued introduction of exciting new functional materials, the demand for improved process economics and surface quality and the potential of environmentally efficient machining. It is an enabling process for many precision engineering and manufacturing activities. MF is often now a final added-value operation for critical components in the automotive, auto-sport, aerospace, biomedical, and space industries. The MF process is undergoing change in response to the new demand. New machine tool technologies such as Drag finishing and High–Speed centrifuge finishing are being introduced together with advanced abrasive technologies including thermally treated recycled glass. Such advances are widening process applicability, efficiency and achievable performance. As a result there is a strong need to study the process in order to improve designs and capabilities.

MF describes the wide range of processes used to modify and enhance the surfaces of engineered parts by immersion in a flow of loose abrasive media (granular matter), Figure 1.1.

(a) Spherical ceramic media

(b) Conical plastic media

(c) Cylindrical porcelain media

Figure 1.1: Loose abrasive media

Interest in mass finishing is gathering pace internationally. This is due to batch-scale economies, the rapidly increasing demand for finer tolerances on surface roughness and geometric features, the introduction of new materials and fabrication processes and the requirements for assured surface quality.

Although, a small number of experimental and theoretical studies have been reported on mass removal and deburring effects[1, 2] and the contact forces caused by the media[3, 4], it remains difficult to predict the outcome of a mass finishing cycle on the quality of a surface without extensive pre-production testing. Moreover, as mass finishing is becoming more widely used, it remains extremely challenging to understand how to modify or refine an operation to deliver outcomes appropriate for different parts and/or target criteria, due to the relatively large number of process variables that need to be controlled, including: media (shape, grit material, size), compound solution (flow rate, composition), workpiece material (hardness, composition) and bowl characteristics (capacity, shape, acceleration, frequency).

A key feature of success with the process is achievement of a consistent and evenly distributed relative motion of the media over the surfaces of the component. In previous work, it has been observed that complex flow patterns, such as recirculation and local eddies can occur in vibrated media that affect the performance and thus the uniformity of surface finish[5]. It was successfully demonstrated[6] that the 2-D flow trajectories of media elements, initially measured from a small scale method using Particle Image Velocimetry (PIV), can be predicted using The DEM.

1.2 Contact Mechanic's brief introduction

Contact mechanics relates to the study of the behaviour of solid bodies that contact each other. It is a well-researched topic with early articles dating back to the 1880's when Heinrich Hertz, subsequent to his experiments and observations, wrote his paper "*On the contact of elastic solids*" in which his theory of elastic contact was revealed. Since then, many different contact models have been proposed[7]:

The most important contact models are:

- Hertzian model: This contact model describes the relationship between the stress and the area of contact between two elastic spheres of different radii by neglecting the adhesive surface forces.(Nonadhesive contact)
- Johnson-Kendall-Roberts (JKR) model: This model is based on the Hertz model and integrates the effect of adhesion on the area of contact. (Adhesive contact)
- Derjaguin-Muller-Toporovs (DMT) model: The contact profile is the same as in the Hertzian model, however this model contemplates additional interaction outside the area of contact as well as the adhesion on the contact area. (Adhesive contact)
- Rolling resistance models. These models incorporate a rotational torque to account for the rolling friction.

1.3 DEM brief introduction

There are two different approaches to numerically define a granular flow: the continuum (Eulerian) approach and the discrete (Lagrangian) approach. The continuum approach treats the particles as an artificial continuum and is based on the conservation equation using Computational Fluid Dynamic (CFD) techniques. This approach is computationally efficient although it does not provide information in relation to the local behaviour of the individual particles.

The discrete approach provides information with respect to the individual particles and tracks them at every instant. It does not rely upon continuum mechanics. This approach is highly computationally demanding. The most important discrete approach is the DEM.

DEM arose as a tool to understand the interaction of particles within a granular medium and to predict its flow behaviour. DEM uses several different models for contact behaviour based principally on those models referred to above. DEM is a numerical method of solving bulk particle effects through inter-particle relationships to give solutions in either 2 or 3 dimensions. This is difficult to achieve using a continuum model, representative of flow in media such as rocks, soil or ice masses. This method has primarily been used to help understand and predict flow behaviours in granular or discontinuous materials such as powders and particles with favourable rheological properties.

The first steps towards the DEM date back to 1959, Alder and Wainwright[8]. They were the first to use an electronic computer to numerically calculate the behaviour of several hundred interacting classical (spherical) particles. Twenty years laters, Cundall and Strack[9] pushed forward the understanding of the granular media by developing the Linear Spring and Dashpot model, where the magnitude of the normal force between two particles is considered the sum of spring force and damping force. Originally, the algorithm was proposed for two-dimensional simulations, but more recently efficient programs for three-dimensional simulations have been elaborated[10, 11].

However, the two most common models used in DEM software are the Linear Spring Dashpot model based on Cundall and Strack's work and the Hertz-Mindlin contact model. The latter is a non-linear elastic model with path dependence due to slip motion neglecting the relative roll and torsion between the contact particles.

1.3.1 DEM Applications

In recent years, advances in computing speed and power as well as improvements in programming have enabled the Discrete Element Method to broaden its field of application. It is demonstrated by the broad variety of applications reported in literature and the wide range of industries where it has been applied, such as agriculture and food handling, chemical, pharmaceutical, civil engineering, mining and mineral processing, ceramics, metal - to name but a few.

Many DEM simulations have been published in literature that describe modelling of diverse granular processes such as granular flow in a hopper[10, 11], comminution, fracture of agglomerates[12], granulation, bulk compression of particles[13], flow in screw extruders and conveyors, vibratory screening, filling of dragline bucket, conveyor belt design, earth-mover bulldozer plate design and grinding process[14]. Another significant application for DEM simulations has been to study the mixing of granular materials and granular mixing in blenders.

Although DEM is now being widely applied, new challenges arise and new approaches are needed. MF, due to the granular character of the abrasive media, constitutes one of the industrial areas where DEM simulation has not been used in depth yet and consequently can have a deep impact in the understanding of the MF process as well as in the development of more valuable MF machines.

1.3.2 LIGGGHTS

LAMMPS Improved for General Granular and Granular Heat Transfer Simulations (LIGGGHTS) is a parallel C++ DEM code based on the Molecular Dynamics (MD) code LAMMPS[15] (http://lammps.sandia.gov) distributed by the authors via a dedicated web page (www.cfdem.com)[16]. LIGGGHTS is used for simulating industrial particle processes, such as hoppers discharge, transfer chutes, continuous blending mixers, etcetera.

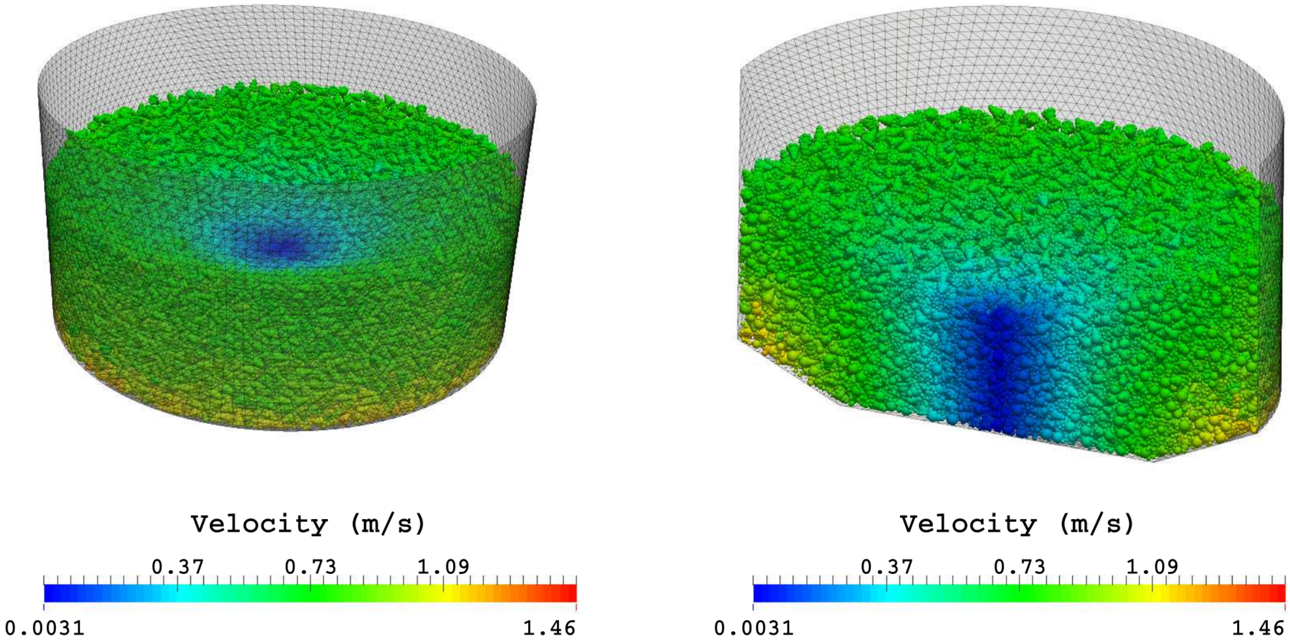
LIGGGHTS can run in parallel, allowing distribution of high demanding computational resources between different processors, which are very powerful when having access to High-Performance Computing (HPC). It is an open-source distribution providing the researcher with a huge flexibility when implementing new features and functionality. It runs from an input script which facilitates later configurations of the simulation. Several granular models are available to use depending on the characterization of the phenomenon to be studied. Among these models are: the Hertz model, a simplified version of the JKR model with two different approaches to calculate the contact area between particles. There are also different implementations of the Elastic-Plastic Spring-Dashpot (EPSD) based on the research work carried out by Chen et al.[17].

Due to all these features, LIGGGHTS has been chosen to simulate the granular flow of mass finishing media contained in the work chamber and to learn its behaviour in order to obtain an insight into this technology.

Before using LIGGGHTS it is important to determine whether the problem is framed within the range of particle processes, if so, then a script[1] has to be defined where all the initial values of the problem are established (boundary condition, contact forces particle-particle, wall-wall, cohesion force, particle material, geometry ...) as well as the desirable output (velocity, position, heat transfer ...).

[1] All the scripts can be found in the Appendix.

Once the script is configured then it is submitted to LIGGGHTS which processes all the calculations. The results obtained from LIGGGHTS are visualised using a different software, Paraview. Paraview is an open-source visualization application. More information is displayed at https://www.paraview.org/.



(a) Disc machine simulation

(b) Disc machine simulation (section view)

Figure 1.2: Early Simulation

Figure 1.2 shows an early simulation of a centrifugal disc machine using LIGGGHTS. The walls of the work chamber are frictionless, the constituent particle of the bulk is a conical shape non-cohesive plastic particle, and the disc spins at 200rpm. It is observed that the distribution of the velocity within the bulk where the particles velocity increases as does the distance from the centre of the work chamber. This model is to be built up by adding friction to the walls, changing the shape of the work chamber, using different media, shapes and materials.

1.4 Problem definition

The area of MF has attracted very little research over the years. As a consequence there remain many unanswered questions concerning the efficiency and capability of the many varying MF process technologies and uncertainty concerning the methods by which to optimise the processes. Furthermore, the flow phenomenon and its relationship with performance and capability has not been widely researched. Thus, the MF systems are lacking in technological insight and control and somewhat lacking in exploitation as a result. This project proposes to gain improved understanding and insight into these critical matters through development of novel DEM simulation methods supported with well-considered experimental studies. This will deliver to industry the necessary technology and understanding to rapidly advance MF.

1.5 Project Aim and Objectives

This project directly addresses fundamental manufacturing research questions. It proposes a series of innovative research programs in MF that is underpinned by solid and extensive knowledge and experience. It will aid in the delivery of next generation machining systems and improve the reliability and performance able to be derived from current practice.

1.5.1 Aim

The principal aim of the project is to secure and deliver to industry the necessary scientific grounding required to advance and exploit the MF process through the design and development of simulation and modelling tools.

1.5.2 Objective

Project objectives will help establish the foundation upon which many future researches studies may be advanced. Stated as:

- To design, build and evaluate a 2D DEM simulation model for MF that provides a framework for further 3D development.

- Using PIV of a transparent 2D trough, demonstrates that DEM is a useful method to predict flow patterns in vibrated media. This will be based on a key measure of the flow, such as the circulation and to compare the experimental data to simulate results for a range of process settings.

- To study the vibration of the vibratory trough (Figure 2.2a) in order to understand the behaviour of the system when working under different loading conditions and excitation energy . The vibration data obtained from this analysis will be used as input for the simulation.

- To establish a range of experiments to obtain the properties of the media particles: Friction coefficient, rolling friction, coefficient of restitution.
- To develop a 3D model for the vibratory tub.
- To establish a 3D model for rotary-fluidized flows (Figure 2.2c), using spherical and conical particles.
- To study the impacts distribution of the media on a specimen's surface and the stress distribution within the specimen.

1.6 Scope of the Dissertation

The structure followed in this thesis initially presents a brief introduction of the two main elements of the current work. On one hand, the mass finishing process as the object of study. On the other hand, DEM as the tool used to investigate the object of study. The second and third chapters delve deeper into this two subjects highlighting the current state of art.

The relevant mechanical properties of the material involved in the mass finishing process must be adequately defined in order to obtain realistic simulations. The fourth chapter details the experiments carried out to retrieve these properties.

The three following chapters inform on the simulations executed to shed light on the mass finishing process. The first simulation studied the vibratory trough; the results of this simulation were inconclusive, however a full vibration analysis of the system was completed. A different simulation was created to study the rotatory disc mass finishing process, a novel approach using a wireless sensor was performed to validate the simulated media flow within the bowl. This simulation provided valuable information on the distribution of velocity within the bulk, also a flow pattern was identified. The last simulations, the more complex, were built on the *shoulders* of the previous one. They were designed to study the interaction between the media particles and the workpiece. A number of novel and interesting results were exposed, such as the relevance of the position of the specimen in relation with the bulk, the distribution of stresses beyond the surface, how deep the stress travels in the workpiece, what features of the workpiece, faces or edges, are more affected by the media impacts.

The eighth chapter discusses the main finding exposed throughout this thesis. Fundamentally, the use of DEM as a tool to investigate the mass finishing, its advantages and limitations. The ninth chapter summarises the most relevant conclusions highlighted on the previous chapters.

Lastly, the tenth chapter proposes a number of ideas to continue studying in the direction initiated in the present work.

Chapter 2

Literature Review: General Background

This chapter will inquire into the beginning of MF technology and how it has been developed until today. As a consequence, it will be necessary to define the concept of granular material and its characteristics as this is the fundamental part of MF technology.

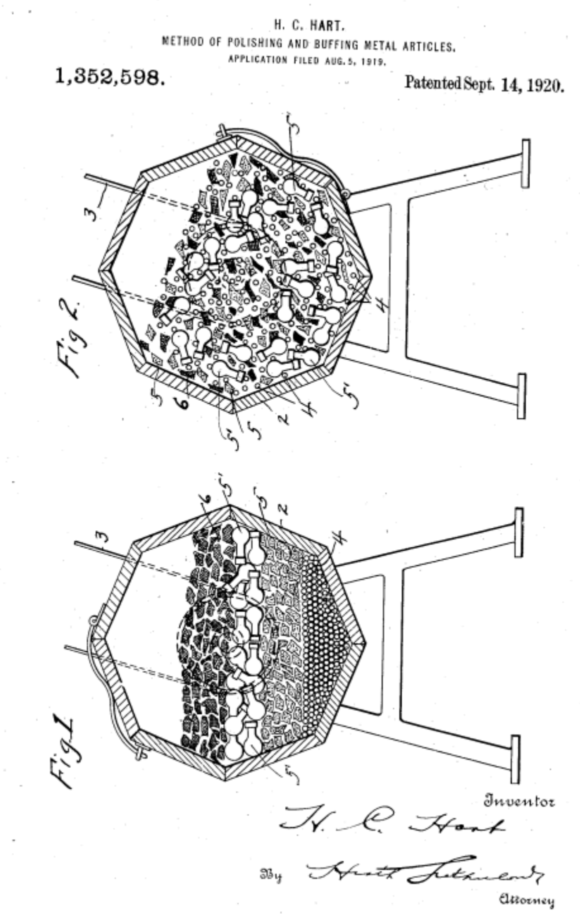
2.1 Mass Finishing

The Industrial Revolution resulted in an enormous growth of the industry and an essential change in the means of production. Machines were taking over many works that were typically carried out by men. Where once a craftsman produced goods in low volumes, now there were factories and assembly lines enabling mass production.

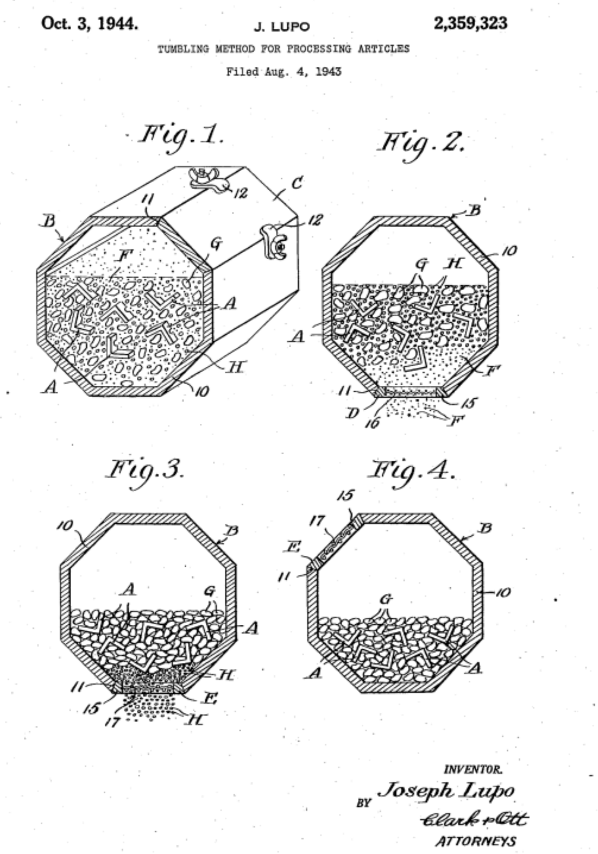
Industry allowed the fast and systematic manufacturing of parts such as buckles, boxes, buttons, etc. At this point a new technique was necessitated to enable repeatable and uniform edge and surface finish effects which could be applied to a high volume of parts.

Lupo[18] developed the MF techniques, pioneering dry barrel finish and tumbling techniques in the early decades of the twentieth century, Figure 2.1. Lupo found that "shoe pegs" made an ideal MF medium for smoothing and polishing plastic parts in rotary barrel finish equipment.

A weakness inherent to the industry as it developed from this time period onwards, is that, very often, successful finish processing is highly dependent on the knowledge held by a few individuals,



(a) Lupo's drawing (1920)



(b) Lupo's drawing (1943)

Figure 2.1: US Patent Office drawings

frequently the technicians in charge of the processes. This knowledge has been passed along as oral tradition and is usually based on extensive trial and error experience. Little has been done to develop and secure comprehensive knowledge that allows a foundation for future application process development. This weakness has only begun to be corrected in relatively recent years.

With the development of the technology, a wide variety of parts can be mass finished, including: die-castings, forgings, extrusions, castings, machined parts and additive manufacturing parts, as well as parts made from wood, composite materials, ceramics and plastic. Nowadays, there are many different MF processes in use including: vibratory trough (Figure 2.2a), vibratory bowl (Figure 2.2b), centrifugal disc (Figure 2.2c), drag finishing machine (Figure 2.2d) among others. However vibratory finishing is the most common due to its versatility and consistency. In such processes, the kinetic energy of the vibrating walls is transferred to the media particles which then impact and slide on the surfaces of the immersed part. The energy transfer is responsible for material removal and the range of finishing actions from surface cleaning to superfinishing. The energy consumed is largely determined by machine type, vibration amplitude, combined masses, media type and lubrication condition. The media type used is dictated by whether the objective is to accomplish a Rate of Metal Removal (MMR) or impart a smooth, lustrous finish.

(a) Vibratory trough

(b) Vibratory bowl

(c) Centrifugal disc

(d) Drag finishing machine

Figure 2.2: Mass finishing processes

MF processes are continually being developed and enhanced. Originally they were used exclusively for simple deburring operations on mass-produced parts. At the present new applications have been found such as smoothing, precision finishing, polishing, rounding, descaling and cleaning, grinding, etc. Currently MF is present in many different industries, examples below:

- Automotive industry where an appropriate surface finish can reduce the friction between parts due to an improvement of the contact areas and thus augmenting the life of the component.
- Aerospace industry, in which, for instance, the roughness of the surface of a turbine blade has a direct impact on the efficiency of the system.
- Medical devices and pharmaceutical industry where the parts are highly customizable (tibias, femoral shafts, prosthetic sockets, ear molds, etc.) and must meet the highest standards in order to maximize comfort. MF provides a fast and cost-effective surface finishing as well as consistently finish.
- Huge numbers of ceramic and plastic parts are manufactured every day, finishing them manually is simply not viable.
- Jewellery industry where parts can be very intricate and very difficult to polish manually.
- Additive manufacturing industry that produces large quantities of parts as well as complex parts impossible to manufacture using any other fabrication method.

Although MF is quite extended within the industry, it continues to be a largely empirical process, thus it is still one of the least-understood, under-appreciated manufacturing technologies. It is necessary to investigate this technology in order to not rely on a few people's practical knowledge, being able to spread the knowledge across industry and establish the foundation of the future generations of MF machines.

2.2 Mass Finishing Technologies in the literature

Hashimoto et al.[19] completed a full classification of the abrasive finishing technologies as well as presenting an up-to-date directory of the scientific research completed for the different abrasive finishing technologies. Furthermore, the material removal mechanism for the different technologies was discussed and the fundamental parameters established. In the classification carried out by Hashimoto, two main families of technologies were identified in accordance with the state of the abrasive material: The bonded abrasive and the unbonded abrasive families. The MF technologies fall into the first family where the abrasive material is bonded to a substratum (abrasive media particles).

Whereas other categories that are driving attention from the researcher community, such as the Abrasive Flow Machining (AFM) and the Magnetic Abrasive Finishing (MAF) categories, fall into the second family.

Very little scientific research has been published in the 50+ years of MF technology development. In the second half of the 1990's, Hashimoto[20] established the basic rules of vibratory finishing and developed a mathematical model to predict the roughness and stock removal of finished parts. Wang et al.[4] explored the relation between the normal contact forces for different working conditions in a vibratory finishing machine and the resulting changes in surface roughness and hardness of aluminium workpieces. They stated that the changes in hardness appeared to be dependent mostly on the lubrication condition and the media roughness while the media diameter produced relatively smaller changes in the hardness of the workpiece. They also observed that the impact parameters such as the average force, maximum force did not vary appreciably for dry and water-wet conditions. Yabuki et al.[21] observed simultaneously the normal and tangential forces generated by the media within the vibratory finishing bowl, in an attempt to pursue a better insight into the contact behaviour and its effects on roughness and hardness occurring in the finishing process for both water-wet and dry conditions. To this end, they used a miniature video camera within the workpiece and a scanning electron microscopy to take individual micrographs of the impact craters. It was concluded that the media contact occurred in three different modes: free impact; rolling of media; a piece of media stationary on the surface while adjacent media particles roll over it. The largest normal impact forces were associated with the third mode. They also observed that the greatest impact force and the frequency of impact occurrence are much higher in the dry condition than in the water-wet condition. This latter statement opposes what S. Wang et al. concluded in their paper. A more general characterization of the impact conditions within the work chamber was offered by Ciampini et al.[3]. They concluded that the impact velocity of the media in a vibratory finisher is of fundamental importance in the analysis of the erosion and surface plastic deformation during burnishing and work hardening. Domblesky et al. [1, 22] conducted an experimental investigation of the vibratory finishing process, and presented the model of material removal rate as a function of bowl acceleration. They observed that large differences in density between the work-pieces and the media result in greater material removal rates. In later works, Hashimoto et al.[23, 24] presented a model of the vibration system in a vibratory finishing as well as a mathematical model for the material removal mechanism in vibratory finishing. They also discussed the optimization of the key parameters in the vibration system to improve its performance. Uhlmann et al.[25] combined a robot and a vibratory bowl. The robotic arm introduced the workpiece within the work chamber and executed different path and speeds. They presented a model for the vibratory finishing bowl that could be applied to other finishing scenarios (different workpieces, media, workpiece speed, etc) due to the generic approach adopted to develop it. It is of note that most, if not all studies have been constrained to the spherical media.

There has been some effort investigating the impact of chemical solutions on the material removal rate as well as the surface roughness. Michaud et al.[26] investigated the advantages of using chemical solutions to accelerate the superficial treatment of the workpiece when using a vibratory finishing bowl. Song et al.[27] observed that in order to achieve an adequate surface finishing it is imperative that the etching speed of the chemical solution is well balanced with the mechanical material removal.

Chapter 3

Literature Review: DEM

3.1 Introduction

Granular material is defined as a large conglomeration of discrete macroscopic particles. These particles can go from 1 μm up to few metres, e.g. sand, powders, etc. According to Woodcock and Mason [28]:

> “A bulk solid (granular material) consists of many particles (granules) of different sizes (and possibly different chemical compositions, densities, shapes) randomly grouped together to form a bulk. The ‘nature’ of such material (...) is thus dependent on many factors (...)”

Although the constituent particle of a bulk is a solid particle, it may behave differently from ordinary solids, liquids and gases. This has led many to characterize granular materials as a new form of matter.

(a) Red lava rock bulk (b) Tablets bulk (c) Finishing media bulk

Figure 3.1: Granular bulks

When studying a bulk it is important to distinguish between the properties of the bulk itself; for example, its flow properties, cohesion, adhesion, etc. and the properties of the constituent particles such as particle density, particle hardness, surface area, etc.

3.1.1 Bulk properties

Although a bulk and its constituent particles have different properties, most of the time the latter determines the properties of the bulk itself. Some of the most common properties of the bulks and their constituent particles are briefly defined below as well as the differentiation of particle and bulk properties.

3.1.1.1 Voidage and density

As defined previously, a bulk is a conglomeration of solid particles. However there are free spaces between particles when these are packed together in a random configuration. The percentage of free space within a bulk is known as *voidage* or *void fraction* (ϵ). Thus:

$$\epsilon = \frac{V_{voids}}{V_{solids} + V_{voids}} \tag{3.1}$$

The volume of *solid particle* or *fractional solids content* is then $(1 - \epsilon)$

Although many authors use the term *porosity* to refer to the voidage, it will not be used in this document as it could be confused with the porosity of the constitute particle.

Bulk density(ρ_b) is then defined as:

$$\rho_b = \frac{m_{solids} + m_{voids}}{V_{solids} + V_{voids}} \tag{3.2}$$

Writing ρ_p as the density of the solids and ρ_f as the density of the fluid in the voids spaces, the expression for the bulk density becomes:

$$\rho_b = \rho_p \cdot (1 - \epsilon) + \rho_f \cdot (\epsilon) \tag{3.3}$$

For a dry bulk in which the fluid density is negligible then the density of the bulk is:

$$\rho_b = \rho_p \cdot (1 - \epsilon) \tag{3.4}$$

The density of a constituent particle is defined as the mass of the constituent particle divided by its volume.

$$\rho_p = \frac{m_p}{V_p} \tag{3.5}$$

3.1.1.2 Size and size distribution

Size is the term used to describe the average dimension among the constituent particles. The use of this term is vague and varies from one industry to another. Nonetheless, bulks can be sorted depending on the particle size. Table 3.1 sets out approximate ranges according to Woodcock and Mason[28].

Table 3.1: Qualitative terms used to describe the size of bulk solids

Descriptive term	Typical size range	Examples
Coarse solid	5 - 100 mm	Coal, aggregates, etc
Granular solid	0.3 - 5 mm	Granulates sugar (0.3 - 0.5mm); rice (2 - 3 mm)
Particulate solid:		
coarse powder	100 - 300 μm	Table salt (200-300 μm)
fine powder	10 - 100 μm	Icing sugar ($\cong$ 45μm)
superfine powder	1 - 10 μm	Face powder
ultrafine powder	<1 μm	Paint pigments

Determining the size of the constituent particles that define the bulk when these are spherical, it is easy to obtain the average particle diameter, however most of the bulks have non-spherical particles as constituent particles. For these cases it is necessary to establish an *equivalent dimension*, such as *equivalent diameter* which would be the diameter of the sphere that behaves in a similar way to the non-spherical particle. A different equivalent would be *volume diameter* corresponding to the diameter of the sphere having the same volume as the original particle. A similar concept is applicable to surface diameter.

The definition of particle diameter has to be specific depending upon the use of the bulk.

3.1.1.3 Shape, surface area and hardness

Constituent particles are usually non-spherical particles, in some industrial processes it is important to define the shape of the particles. Although defining the shape in mathematical terms is not easy some shape factors have been developed. The most commonly used is the *sphericity* ϕ_s, defined as the distance of the non-spherical particle to a sphere of the same volume. The mathematical expression is then:

$$\phi_s = \left.\frac{\text{surface area of sphere}}{\text{surface area of particle}}\right\}\text{of the same volume} \tag{3.6}$$

The hardness of the material is an important property, especially in MF. As a common rule, the harder the constituent particles are the more abrasive the bulk is.

In the literature can be found many different ways to perform hardness tests depending upon the material to be evaluated.

3.1.1.4 Cohesion and adhesion

Cohesion is described as the resistance of a bulk solid to shear at zero compressive normal stress. The Figure 3.2 illustrates a column of bulk solid at a density achieved by the consolidation conditions when not subjected to an imposed normal and loaded at a later stage by a Normal Load (σ).

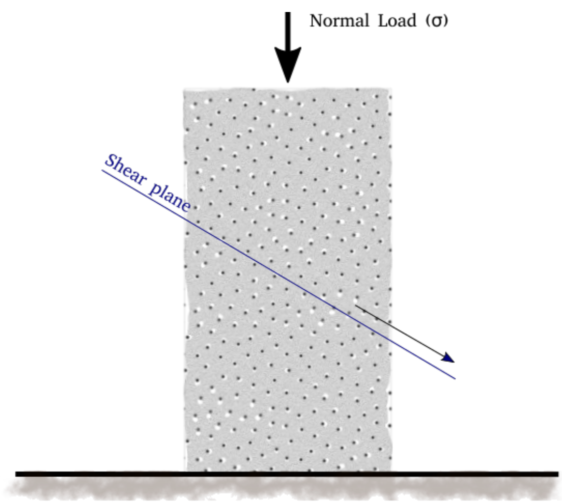


Figure 3.2: Column of consolidated and unsupported powder

The column of bulk solid possesses a yield locus (Figure 3.3) which defines a range of stresses that can be applied without causing a permanent deformation. If the σ keeps augmenting it will reach a magnitude where the constituent particles of the column of powder will slide (permanent deformation). The sliding plane is known as Shear Plane. Shear strength is defined as the maximum value that the shear stress can reach just before sliding and its mathematical expression is:

$$\tau_s = \mu(\sigma + T_a) \tag{3.7}$$

Where:

- τ_s is the Shear Strength.

- μ is the Internal Coefficient of Friction.

- σ is the Normal Load applied on the column of powder

- T_a is the Tensile Strength.

For a non-cohesive bulk when $\sigma = 0$ then $\tau = \mu \cdot T_a = 0$. The figure 3.3 plots this function in green. It can be observed that the function passes through the origin, meaning that when either the Shear Strength or the Normal Load is zero then the other is also zero.

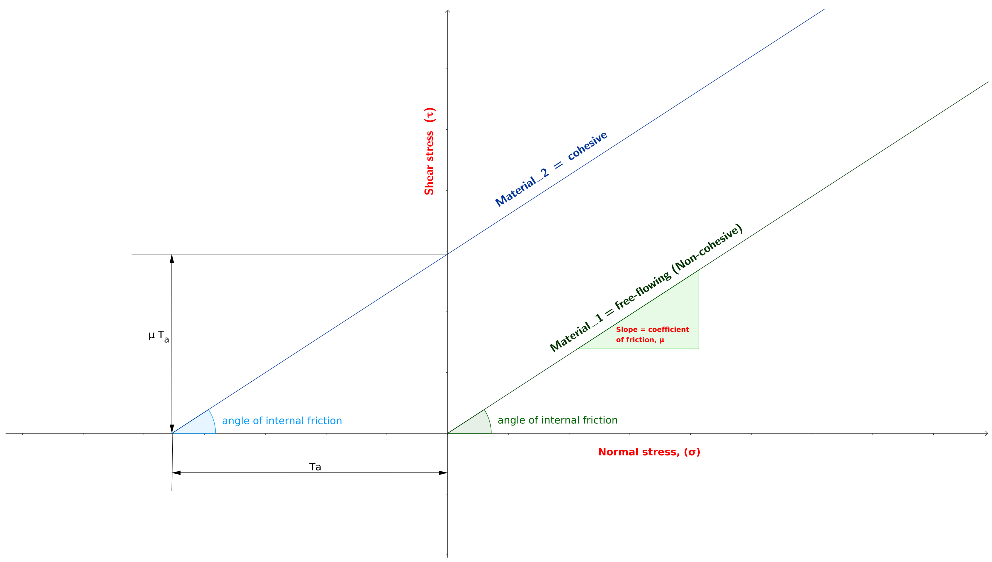


Figure 3.3: Coulomb model for shear strength of a particulate material

For a cohesive material, plotted in blue, when the $\sigma = 0$ then $\tau = \mu \cdot T_a \neq 0$.

The more cohesive a bulk is, the less *flowability* it has. The cohesion of the bulk depends on the constituent particles but also the electrostatic charging and the moisture can affect the cohesiveness of the bulk. For instance, dry sand is a non-cohesive bulk with a high *flowability* (free-flowing), on the contrary wet sand is a cohesive bulk and its *flowability* is very poor (cohesive).

A cohesive bulk occurs when its constituent particles have a tendency to glue together with more or less intensity, whereas adhesion is defined as the tendency of the solid particles to stick to the surface of the containing body.

Summarizing, cohesion is related to particle-particle properties while adhesion is related to particle-wall properties.

3.1.1.5 Angle of repose (AoR)

The Angle of Repose (AoR)(ϕ_r) is one of the most important macroscopic parameters in characterising the behaviour of granular materials; "*One of the most interesting phenomena is the evolution of a pile of granular material. When sand grains are dropped on the top of a pile its slope does not depend on the number of grains as long as the heap is not too small. The avalanches going down the surface of the heap have no characteristic size, they are only limited by the size of the entire pile*',[29]. The AoR is then defined as the angle of inclination of the free surface to the horizontal of a bulk solid heap, Figure 3.4.

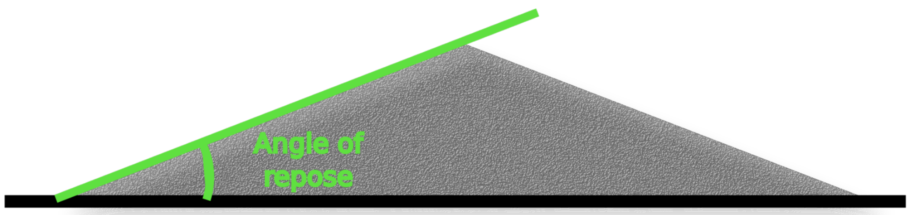


Figure 3.4: Angle of repose of generic bulk solid pile

The AoR is related to many important phenomena in different areas of knowledge, such as geology where it has an important role in the stratification phenomenon. Also in civil engineering where the study of the AoR is primordial in the design of slopes. The angle of repose is one of the easiest parameters to measure and gives a rough estimation of the cohesiveness of bulk solids. There are disagreements in the literature with respect to the usefulness of the angle of repose as a flow property with some favouring its importance as a flow property while others do not. In general, the angle of repose is accepted and widely used.

3.1.1.6 Moisture content

Although moisture levels do not always have a critical impact on the bulk behaviour, there are occasions where it can affect the mechanical properties of the bulk. In 1995 Amidon and Houghton[30] demonstrated that moisture can change significantly the mechanical properties of microcrystalline cellulose. Yi et al[31] demonstrated how moisture modifies certain characteristics of the continuous rootzone sand mixtures bulk, such as bulk density, shear modulus, consolidation index , etc. Moisture can have an important influence on the behaviour of bulk solids, therefore moisture analysis is one of the most frequently performed tasks in their characterization.

The mathematical expression of the moisture content is as follows:

$$\text{moisure content} = \frac{\text{mass of water}}{\text{mass of dry solids}}(\times 100\%) \tag{3.8}$$

In this case the moisture content is expressed in terms of the percentage of water to dry solids. There is also a mathematical expression for the percentage of water to wet solids:

$$\text{moisure content} = \frac{\text{mass of water}}{\text{total mass of solids and water}}(\times 100\%) \tag{3.9}$$

It is important to distinguish the two forms in which water can be found in a bulk:

- Surface moisture: present only in the surface of the constituent particles.

- Inherent moisture: present within the structure of the constituent particles.

The simplest way to find out the moisture content of a bulk solid is to weigh a sample and place it in the oven for an appropriate length of time and then weigh again. The differences in the weight is the mass of water.

Alder and Wainwright[8], in the late 50s, understood the difficulties to solve a many-body (bulk) problem and the necessity of using computing power to solve problems where the number of calculations were too big to be accomplished by human hands:

> “One of the great difficulties in the present day theoretical attempts to describe physical and chemical systems is the inadequate mathematical apparatus which has been available

> to solve the many-body problem(. . .) although the properties of an isolated molecule are well established and the elementary processes which occur when two such molecules interact are described by well-known laws, the behaviour of systems of many interacting molecules cannot, in general, be dealt with theoretically in an exact way(. . .) a three-particle system presents great analytical difficulty. Since these difficulties are not conceptual but mathematical, high-speed computers are well suited to deal with them.”

They established a numerical method by which it was possible to calculate the behaviour of systems with a few hundred interacting particles (spherical particles). One of the benefits of this method is that more detailed information is available from calculations compared to what it is available from real experiments. Since the individual particles can be tracked knowing their motion at any time. The main constraint of the method was due to the limited power of those primitive computers, only being able to handle a few hundred particles.

According to Alder and Wainwright, a brief description of the method would be:

> “(. . .)one could at any instant calculate the force on each particle by considering the influence of each of its neighbours. The trajectories could then be traced by allowing the particles to move under a constant force for a short-time interval and then by recalculating a new force to apply for the next short-time interval, and so on. This method could also handle particles(. . .) with rotational and other degrees of freedom(. . .) The accuracy of such calculations would depend on the length of the time interval.”

This method will not be discussed here further, however a more extensive description can be found on *Studies in Molecular Dynamics. I. General Method*[8].

Several other people have continued with the development of the numerical methods as a way to better understand the granular matter, and in a final instance to be able to predict the behaviour of any granular system. The DEM introduced by Cundal and Strack, enabled them to develop a numerical model and validate it by comparing the force vector plots obtained from a computer program (BALL) with the corresponding plots obtained from a photoelastic analysis. A summarized description of the method is displayed below.

3.2 The Distinct Element Method (DEM)

Analytical and physical approaches have been considered to describe the mechanical behaviour of assemblies of discs and spheres and to determine the interior stresses within the assembly. In 1979 Cundall and Strack published a paper[9] where they proposed a numerical approach. The method was validated by comparing force vector plots obtained numerically with the corresponding plots obtained from a photoelastic analysis (physical approach). This numerical approach allowed different loading configurations, as well as, particle sizes, size distribution and physical properties.

According to the authors the principles on which the method is based are:

> “In the distinct element method, the equilibrium contact forces and displacements of a stressed assembly of discs are found through a series of calculations tracing the movements of the individual particles. These movements are the result of the propagation through the medium of disturbances originating at the boundaries (...) The speed of propagation is a function of the physical properties of the discrete medium.”

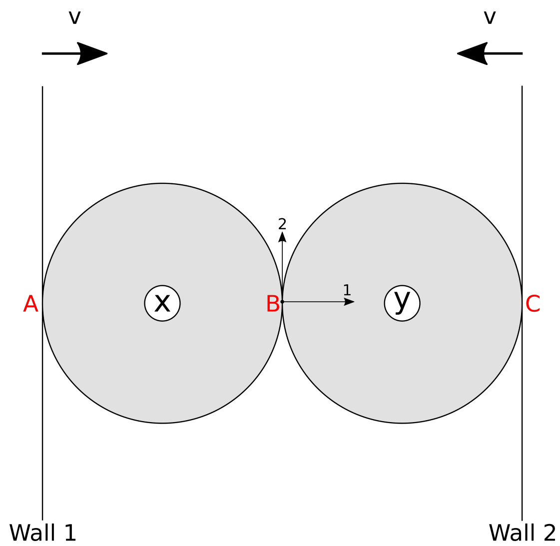
In order to develop the method certain considerations must be fixed:

> “The distinct element method is based upon the idea that the time step chosen may be so small that during a single time step disturbances cannot propagate from any disc further than its immediate neighbours. Then, at all times the resultant forces on any disc are determined exclusively by its interaction with the discs with which it is in contact. It is this key feature of the distinct element method which makes it possible to follow the non-linear interaction of a large number of discs.”

Two laws are used in the distinct element method: Newton’s second law is used to determine the motion of the particles resulting from the forces acting on them. The force-displacement law uses the displacement to find contact forces.

> “The deformations of the individual particles are small in comparison with the deformation of a granular assembly as a whole. The latter deformation is due primarily to the movements of the particles as rigid bodies. Therefore, precise modelling of particle deformation is not necessary to obtain a good approximation of the mechanical behaviour(...) the particles are allowed to overlap one another at contact points. This overlapping behaviour takes the place of the deformation of the individual particles(...) these overlaps are small in relation to the particle sizes.”

To illustrate how forces and displacements are determined during a calculation cycle, Cundall and Strack presented two weightless discs, labelled as disc X and disc Y, Figure 3.5. The discs are squashed by two moving walls located at both ends. At a time step $t = t_0$ the whole arrangement is in physical contact, however no contact forces are displayed.



Time step $\Longrightarrow t = t_0$:

The discs and walls are in contact.

No contact forces exist.

No displacements exist.

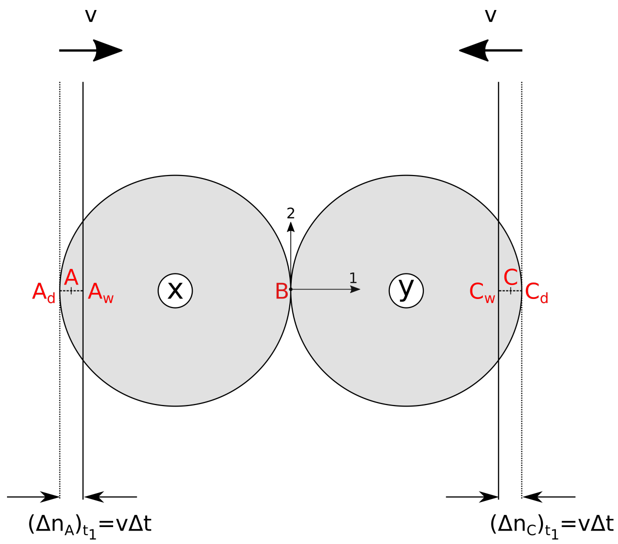
No accelerations exist.

Figure 3.5: Time step equal to $t = t_0$

At a later time step(Δt), $t_1 = t_0 + \Delta t$, the two walls commence moving towards each other squashing the elastic discs(κ). In accordance with the authors assumptions where any disturbance cannot travel beyond a single disc within one time step, both discs remain in the original position during this time step, while an overlap(Δn) takes place at both ends where the walls are in contact with the discs, Figure 3.6, point A and point C.

The expression for the overlap at $t = t_1$ stands as follows:

$$\Delta \vec{n} = \vec{v} \cdot \Delta t \tag{3.10}$$



$t_1 = t_0 + \Delta t \Rightarrow t_1 > t_0$

Overlap in points A and C are:

$$\Delta\vec{n} = \vec{v} \cdot \Delta t$$

The increment in normal force is:

$$\Delta\vec{F}_n = \kappa_n \cdot \Delta\vec{n}$$

Accelerations exist. No variation on discs positions.

Figure 3.6: Time step equal to $t_1 = t_0 + \Delta t$

Where v is the velocity at which the walls are moving, the velocity remains constant during the time step. The force-displacement law is used for finding the expression of the contact forces at the discs:

$$\Delta\vec{F}_n = \kappa_n \cdot \Delta\vec{n} \tag{3.11}$$

Where κ_n is the normal stiffness and $\Delta\vec{F}_n$ represent the increment in normal force.

Taking as positive the 1 direction as pointing from disc X to disc Y and based upon the Equation 3.11 two expressions are obtained for the overlaps during this time step:

$$\vec{F}_{(x)_1} = \kappa_n \cdot (\Delta\vec{n})_{t1}, \qquad \vec{F}_{(y)_1} = -\kappa_n \cdot (\Delta\vec{n})_{t1}, \tag{3.12}$$

It is possible to find the acceleration for the discs X and Y using Newton's second law ($\vec{F} = m \cdot \ddot{x}$).

$$\ddot{x}_1 = \frac{\vec{F}_{(x)_1}}{m_x}, \qquad \ddot{y}_1 = \frac{\vec{F}_{(y)_1}}{m_y} \tag{3.13}$$

The accelerations found from Equation 3.13 are assumed to be constant over the time interval from $t_1 = t_0 + \Delta t$ to $t_2 = t_0 + 2 \cdot \Delta t$ and may be integrated to yield velocities.

$$\dot{x}_1 = \left[\frac{\vec{F}_{(x)_1}}{m_x}\right]\Delta t, \qquad \dot{y}_1 = \left[\frac{\vec{F}_{(y)_1}}{m_y}\right]\Delta t \tag{3.14}$$

The above equations are the mathematical expressions for the velocity of the discs X and Y during the interval from t_1 to t_2. The Figure 3.7 displays the overlaps within the discs-wall arrangement for time t_2. It can be observed, in addition to the overlaps at both ends, points A and C, there is a new overlap at point B.

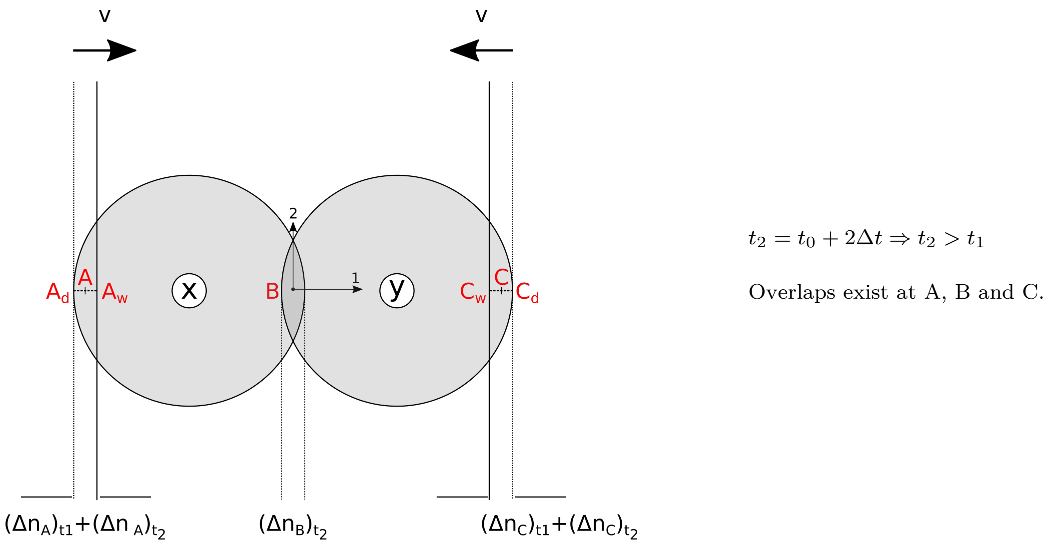


Figure 3.7: Time step equal to $t_2 = t_0 + 2\Delta t$

The relative displacement increments at contact points at time t_2 are found from the following equations:

For the case of the contact disc-wall at the Point A, the expression would be as follows:

$$(\Delta n_A)_{t2} = \left[v - \left(\frac{F_{x1}}{m_x} \cdot \Delta t \right) \right] \Delta t \tag{3.15}$$

Where:

$(\Delta n_A)_{t2}$ = The relative displacement increments at contact point A.

v = The constant velocity of the moving walls.

$\frac{F_{x1}}{m_x} \cdot \Delta t$ = The velocity obtained from Newton's Second Law for the disc X (Equation 3.14).

In the case of the point B, disc-disc contact, the expression would be:

$$(\Delta n_b)_{t2} = \left[\left(\frac{F_{x1}}{m_x} \cdot \Delta t \right) - \left(\frac{F_{y1}}{m_y} \cdot \Delta t \right) \right] \Delta t \tag{3.16}$$

Where:

$(\Delta n_B)_{t2}$ = The relative displacement increments at contact point B.
$\frac{F_{x1}}{m_x} \cdot \Delta t$ = The velocity obtained from Newton's Second Law for the disc X (Equation 3.14).
$\frac{F_{y1}}{m_y} \cdot \Delta t$ = The velocity obtained from Newton's Second Law for the disc Y (Equation 3.14).

For the contact point C, disc-wall:

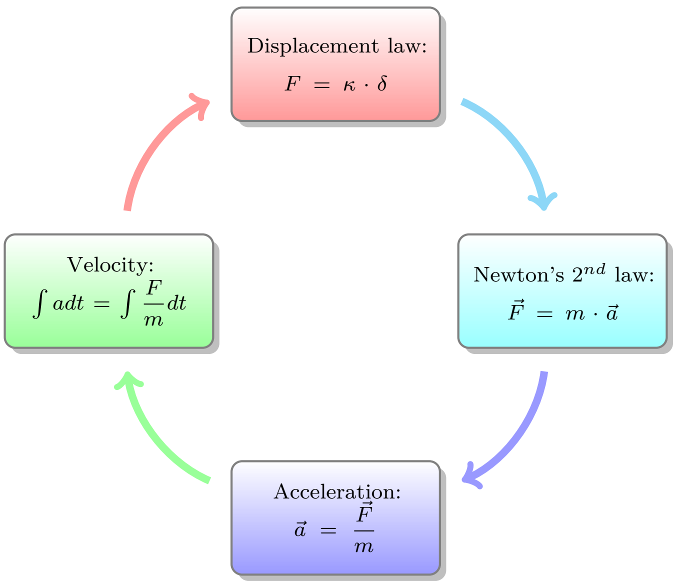
$$(\Delta n_C)_{t2} = \left[\left(\frac{F_{y1}}{m_y} \cdot \Delta t\right) - (-v)\right] \Delta t \quad (3.17)$$

Where:

$(\Delta n_C)_{t2}$ = The relative displacement increments at contact point C.
$\frac{F_{y1}}{m_y} \cdot \Delta t$ = The velocity obtained from Newton's Second Law for the disc Y (Equation 3.14).
v = The constant velocity of the moving walls.

Δn_a, Δn_b and Δn_c are taken as positive for compression.

This cycle consists in using the velocity(v) and the timestep(Δt) to know the displacement(Δn) of the point under study, then the force-displacement law ($F = \kappa \cdot \Delta n$) is used to obtain the forces($\vec{F}$) acting on that point, Equation 3.11. Once the force is known, by mean of Newton's Second law, the accelerations($\ddot{x}$) for the two discs are found, Equation 3.14. Therefore, the velocity($\dot{x}$) and position(x) of discs are also know.



"The above example is an illustration of the cycling through a force-displacement law and the law of motion. In the general case of an assembly of many discs, the force-displacement law is applied at each contact of any disc and the vectorial sum of these contact forces is determined to yield the resultant force acting on that disc. When this has been accomplished for every disc, new accelerations are calculated from Newton's second law."

The force-displacement law is presented below for the case of two discs overlapping each other. The Figure 3.8 illustrates this case.

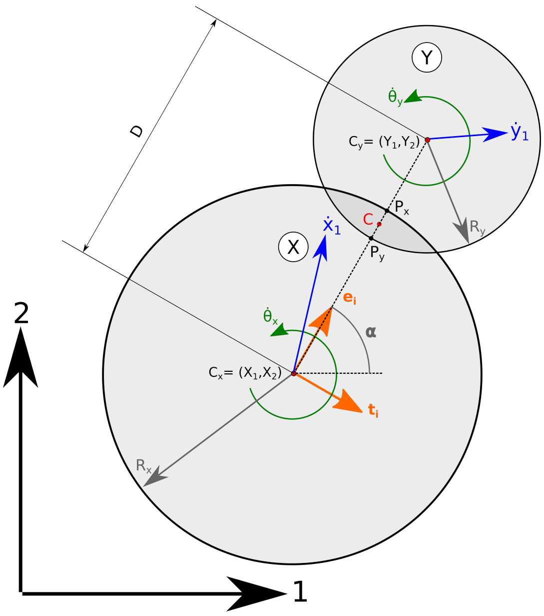


Figure 3.8: Force displacement law for two discs in contact

Where:

X and Y	$\Rightarrow$	Disc's label.
C_x and C_y	$\Rightarrow$	Centres of the discs.
R_x and R_y	$\Rightarrow$	Radii.
$\dot{\theta}_x$ and $\dot{\theta}_x$	$\Rightarrow$	Angular velocities.
$\dot{\mathrm{x}}_1$ and $\dot{\mathrm{y}}_1$	$\Rightarrow$	Velocity vectors.
m_x and m_y	$\Rightarrow$	Masses of the discs.
P_x and P_y	$\Rightarrow$	Points of intersection on the line connecting the disc centres with the boundaries of the discs x and y.

For obvious reasons, two discs are considered to be in contact only when the next inequation is satisfied.

$$\mathrm{D} < \mathrm{R}_x + \mathrm{R}_y \tag{3.18}$$

The relative velocity at the contact point **C** is defined as the velocity of P_x with respect to P_y, the relative displacement at the contact is determined by integration of the relative velocity. The unit vector $\vec{e_i} = (cos\,\alpha,\ sin\,\alpha)$ is introduced as pointing from the centre of disc X to the centre of the disc Y.

$$\vec{e_i} = \frac{\vec{y_i} - \vec{x_i}}{\vec{D}} \tag{3.19}$$

The unit vector $\vec{t_i}$ is obtained by a clockwise rotation of $\vec{e_i}$ through 90$^{\mathrm{o}}$, Figure 3.8.

The relative velocity $(\dot{X}_i)$ of point P_x with respect to P_y may be expressed as:

$$\dot{X} = (\dot{x}_i - \dot{y}_i) - (\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y)\vec{t_i} \tag{3.20}$$

The expression $(\dot{x}_i - \dot{y}_i)$ represents the relative velocity due to the linear velocity of the discs, whereas the expression $((\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y)\vec{t_i})$ represents the relative velocity due to the angular velocity of the discs.

Therefore, the expressions for the normal $(\dot{n})$ and tangential $(\dot{s})$ components of the relative velocity are:

$$\dot{n} = \dot{X} \cdot \vec{e_i} \rightarrow (\dot{x}_i - \dot{y}_i)\vec{e_i} - (\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y)\cancelto{0}{\vec{t_i} \cdot \vec{e_i}} \Rightarrow \dot{n} = (\dot{x}_i - \dot{y}_i)\vec{e_i} \tag{3.21}$$

$$\dot{s} = \dot{X} \cdot \vec{t_i} \rightarrow (\dot{x}_i - \dot{y}_i)\vec{t_i} - (\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y)\cancelto{1}{\vec{t_i} \cdot \vec{t_i}} \Rightarrow \dot{s} = (\dot{x}_i - \dot{y}_i)\vec{t_i} - (\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y) \tag{3.22}$$

The normal (Δn) and tangential (Δs) relative displacement increment are obtained integrating the corresponding relative velocity.

$$\Delta n = \dot{n}\Delta t \Rightarrow \Delta n = [(\dot{x}_i - \dot{y}_i)\vec{e_i}]\Delta t \tag{3.23}$$

$$\Delta s = \dot{s}\Delta t \Rightarrow \Delta s = [(\dot{x}_i - \dot{y}_i)\vec{t_i} - (\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y)]\Delta t \tag{3.24}$$

The relative displacement increments are used with the force displacement law to calculate increments of the normal($\Delta\vec{F}_n$) and shear forces($\Delta\vec{F}_s$).

$$\Delta\vec{F}_n = \kappa_n \cdot \Delta n \Rightarrow \Delta\vec{F}_n = \kappa_n \cdot \{(\dot{x}_i - \dot{y}_i)\vec{e}_i\}\Delta t \tag{3.25}$$

$$\Delta\vec{F}_s = \kappa_s \cdot \Delta s \Rightarrow \Delta\vec{F}_s = \kappa_s \cdot \{(\dot{x}_i - \dot{y}_i)\vec{t}_i - (\dot{\theta}_x \cdot R_x + \dot{\theta}_y \cdot R_y)\}\Delta t \tag{3.26}$$

Where κ_n represents the normal stiffness and κ_s the tangential stiffness.

At each time step the increments, $\Delta\vec{F}_n$ and $\Delta\vec{F}_s$, are added into the sum of all force increments determined for previous time steps:

$$(\vec{F}_n)_N = (\vec{F}_n)_{N-1} + \Delta\vec{F}_n \qquad (\vec{F}_s)_N = (\vec{F}_s)_{N-1} + \Delta\vec{F}_s \tag{3.27}$$

Sign convention for $(\vec{F}_n)$ and $(\vec{F}_s)$ can be found in the Figure 3.9.

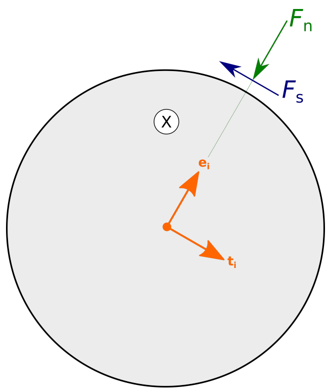


Figure 3.9: Sign convention

Once the normal and shear forces acting on the disc have been calculated for every single contact of the model and the resultants in the normal and tangential directions have also been obtained then they are resolved into components in the 1 and 2 axis, Figure 3.8. The resultant moment acting on a disc is expressed as $\Sigma\vec{M} = \Sigma\vec{F}_s \cdot R$, being R the radius of the disc.

The resultant moments and forces acting on the disc are used with Newton's second law to calculate new acceleration and numerical integrating velocities and then positions.

Cundall and Strack developed the discs model, implemented the algorithm and developed the computer program giving birth to the Discrete Element Method. Further details on the method proposed by them are available in the literature[9].

Further work has been built based on the DEM, developing new models to incorporate features to allow more realistic results in the simulation as well as solving more complex problems. Many researchers have agreed in the importance of the effect of rolling friction on granular flow. When rolling resistance is absent unrealistic results are obtained. Iwashita and Oda[32] proposed a modified model of the conventional DEM that takes the rolling resistance into account. Tordesillas et al. examined the influence of the rolling resistance in the constitutive response of a semi-infinite material to indentation by a rigid flat punch on the material boundary, concluding that models of granular materials need to incorporate at least particle rotations and rolling resistance, to provide a reasonably accurate representation of real particles behaviour. Goniva and Kloss[16] studied the effect of rolling friction on the dynamics in a single spout fluidized bed using DEM coupled to CFD. They proved that simulation results can be improved significantly when applying a rolling friction model.

3.3 DEM application

With the advance of the algorithms used in the DEM softwares and the improvement in parallel computing more commercial software with DEM capabilities has been developed. DEM simulations are increasingly used in many research and industrial fields as it provides an insight into industrial processes that are inaccessible to or difficult to investigate by means of experimental measurements.

Process manufacturing techniques in the pharmaceutical industry require the optimization of powder flow properties to decrease segregation, agglomeration, material loss and assure content uniformity in the final product. Many researchers have used DEM as an approach to gain an insight in these complex processes. Dubey[33] applied DEM to study the powder blending step as this is critical in the process to manufacture pharmaceutical components. Mukherjee et al.[34] reported the response of different pharmaceutical powder under the influence of various humidity conditions. Bhalode and Ierapetritou[35] studied the relation of DEM material parameters with feed factor parameters for a feeder unit to identify the important material properties using DEM. These have each proved beneficial to the industry.

The mining industry involves the loading, transport, crushing, smelting and abrasion of materials that are considered granular matter. In this field, DEM has been used to explore the trajectory of the material inside the mine machinery such as belt conveyors and transfer chutes[36--38]. Other applications of interest in this field are the modelling of the wear of the equipment in contact with the ore[39--41].

DEM is used in the agricultural industry, for example, to study the effect of the filling method on the pressure distribution at the bottom of a silo[42] or to predict the behaviour of the seeds during the discharge and filling processes of the silo[43--45]. It is also used to gain insight into the tillage forces of mouldboard ploughs[46].

The food industry uses DEM to investigate the mechanical behaviour of granular food in complex food processes, such as the cryogenic grinding process used to produce spice powders[47] or the drying and shrinkage process in a fluidised bed[48].

The manufacturing industry with its broad field of applications has used DEM to gain insight into many different industrial applications. Specifically, the mixing processes in rotary drums have been extensively investigated [49--55]. However, since the advent of the Additive Manufacturing (AM) technology more effort has been focused on this topic from the researcher community. Parteli and Pöschel[56] developed a numerical tool based on DEM to study the mechanical behaviour of the powder system, and its geometric and dynamic aspects, during the AM process. Zhao et al.[57] used a similar approach to simulate the powder bed generation. In the same field, Haeri[58] applied DEM to investigate the optimisation of the profile of a blade type spreader for a particle bed fusion process so that this could deliver a powder bed quality comparable or superior to the quality delivered by the counter-rotating roller type for the same operating conditions. Ge et al.[59] used an AM process to print agglomerates and then DEM to simulate the agglomerate crushing process.

3.3.1 DEM applied to mass finishing

Vijayaraghavan et al.[60] presented a paper describing the state-of-the-art in numerical modelling in the MF processes. The DEM approach had some presence in this paper although the experimental approaches were more numerous. Other works are now also emerging however, DEM full capabilities in this field are far from being realised.

DEM has also been used to simulate MF processes. The majority of studies in this field are related to the vibratory finishing machines. The most common approach before the employment of DEM was to study the behaviour of the bulk from the point of view of CFD[61], and as such the bulk was treated as continuum medium. This approach provides minor information on the modes of the impacts that the workpiece undergoes when immersed in the bulk. Therefore the stresses produced on the surface and its vicinity as a consequence of these impacts remain unknown, and the material removal mechanism is not yet fully characterized. Mullany et al.[62] used the PIV method to observe the velocity distribution of the media around a workpiece fixed within the bulk, they successfully predicted the 2D velocity distribution of the media in the vicinity of the workpiece. This approach provided information of the distribution of velocity in the free surface of the bulk. However, the most important limitation of this method is that no information with respect to the velocity in other regions of the bulk was provided nor was insight on the complex 3D media flow presented. Naeini and Spelt[63] were the first in developing a two-dimensional discrete element model to model granular flow in different vibratory beds. They then compared the results with experimental measurements of bulk flow velocity and bed expansion. Uhlmann et al.[64] were early users of DEM with application to the MF processes. They investigated the material removal mechanism in a vibratory finishing machine by combining empirical knowledge and the numerical simulations. They attempted to use DEM to simulate the number of contacts between media particles and workpiece as well as the type of contact and the intensity of these contacts. For their experiments they used two types of media spherical ceramic media and triangular cross-section media. However, in their simulation they used only spherical particles. Furthermore, no information with respect to the vibrational analysis of the machine and the input of it into the DEM model was presented nor detail of the experimental test needed to calibrate the mechanical properties of the media.

Salvatore et al.[65] presented a study which aimed to investigate the evolution of the surface roughness around a part, depending on the abrasive media and cycle duration. To this end, they developed a numerical simulation employing the DEM approach to identify the local condition of contacts . In this simulation, all the bodies were defined as non-deformable and the media particles were characterized as hard spheres with pure friction or friction and restoring energy. They concluded from the simulation that at the front zone, the sliding speed is close to zero and the pressure due to impacts is maximum. In the lateral zone, the pressure is almost zero and the sliding speed is maximum. It was also noticed that due to the incapacity of DEM to predict roughness, a wear model could be implemented so that the output from the numerical simulation could provide an insight into the roughness evolution. Kang et al.[66] applied DEM to understand the dynamic mechanisms of the media system in a vibratory finishing machine. In their model, they used spherical media, its mechanical properties were assumed and no discussion with regards to the method followed to identify the correct properties was discussed. However, they were able to describe the pattern performed by a media particle within the machine's work chamber, through no extensive experimental validation was offered. Wang et al.[67] presented a co-simulation method to study the forces acting on the surface of the workpiece attached across the bottom of the work chamber under the influence of spherical media. However, the tensional state of the workpiece was not discussed. The vibrational excitation of the system was experimentally measured. Furthermore, the experiments required to determine the mechanical properties of the media were not reported. At the same time, the DEM model was not validated beyond basic measurements. Makiuchi et al.[68] proposed a material removal simulation model combining DEM and Preston's law. They used PIV and a rope glued to the workpiece to validate the DEM 3D trajectories through the bowl. However no extensive information about the DEM model was provided.

Only a few articles appear in relation to other MF methods such as the centrifuge disc finisher or drag finisher. Uhlmann et al.[69] presented a DEM model to determine local contact intensities on the workpiece surface and between particles in a drag finishing system. They identified that the contact forces tend to increase with the depth. However, there was not an extensive analysis on where the particles impact the workpiece, the type of impact or the force of direct impacts. Nor was there information regarding the tangential forces produced by the impacts. Li et al.[70] studied the kinematic behaviour of the particles in a centrifugal barrel finishing machine and combined theoretical analysis with DEM simulations. The simulation was only run for spherical particles and no exhaustive details were given with respect to the DEM model developed for the study. However, their study yielded initial information regarding the distribution of velocities within the drum which can be valuable in the design process of new barrel machines. Zanger et al.[71] simulated the tangential velocity of the particles and normal forces on the workpiece's surface using DEM as an attempt to determine the material removal mechanism behind the MF technology, more specifically the stream finishing process. They shed light on the contact behaviour between particles and workpieces. Interesting conclusions were drawn from this investigation. The influence of the depth in the normal forces displayed on the specimen surface was domostrated. They also highlighted the importance of the immersion angle in the efficiency of the process. However, the shape of the media employed was spherical which is not commonly used in industry. Furthermore no discussion regarding the tensional state of the workpieces was presented. Sutowski1 et al.[72, 73] described numerical and experimental methods that focused on determining the kinetic energy distribution of the working medium in the centrifugal disc finishing process. This simulation was conducted also using spherical media particles. They obviously concluded that the velocity distribution of the particle is related to the velocity at which the machine's rotor spins and also that the results of the simulation give grounds to estimate the location of the regions with the highest kinetic energy within the bulk.

3.4 Knowledge gap

All the papers found in the current literature with respect to the MF systems and the application of DEM as a technique to gain insight into the behaviour of the granular flow and the interaction between the media particles and the workpiece, present the same limitation. They have each simplified the multiple and diverse shapes of the media particles into spherical particles. This has a beneficial impact on the computing time needed to obtain the simulation results as well as decreasing the complexity of the media characterization. However, the spherical simplification of the problem does not deliver the most accurate simulations. For example, the packing density of the bulk is profoundly affected when substituting spherical geometries for conical geometries. A realistic simulation is necessary to fully understand the bulk's behaviour. Furthermore, realistic geometries are highly significant when studying the interaction between particles and workpieces. The impacts described by spherical particles are different to those described by other media shapes.

Moreover, the current literature presents a knowledge gap related to the stresses produced within the workpiece due to the impacts of the media particles. The distribution of stresses within the specimen is not predicted and it is not understood how they relate to the normal and tangential forces produced by the media impacts. Furthermore, no information has been found with respect to the regions of the bulk where the specimens are more likely to receive *quality* impacts and also which regions or features of the specimens, such as edges, faces or holes, receive the highest number of impacts.

Chapter 4

Material properties definition

A knowledge of the mechanical properties of the media is essential to obtain accurate results when using DEM. As a consequence of the unlikelihood of obtaining any information regarding the mechanical properties of the media material from the manufacturers or the literature, several experiments were performed in an attempt to shed light upon this matter.

4.1 Elastic properties

4.1.1 Objective

This document informs on the works employed to determine experimentally the *Modulus of Elasticity* and the *Poisson's Ratio* for general purpose plastic media specimens.

Figure 4.1: General purpose plastic media

4.1.1.1 Definitions

Modulus of Elasticity The *Modulus of Elasticity* (E), also know as Young Modulus, is the ratio between the Normal Stress(σ) and the Tensile Strain (ϵ). The corresponding unit for this magnitude according to the SI is the Pascal (Pa).

Poisson's Ratio In discussing the deformation of an elastic body, the *Poisson's Ratio* (ν) is defined as the reduction of the cross section of the body when it undergoes an extension due to the action of a normal stress. *Poisson's Ratio* is dimensionless.

Stress When a force is applied to a face of a rigid body, which has enough constraint to prevent the body from moving, so no displacements of particles are possible without a deformation of it, a tension will be developed within the body. This tension is known as Stress. According to the direction of the stress two forms can be differentiated: Normal stress (σ_n), acting perpendicularly to the face where the force is applied. Shearing stress (τ) acting on the face where the force is applied.

Strain It is defined as the ratio between the elongation of the specimen and its original length.

$$\epsilon = \frac{(l - l_0)}{l_0} \tag{4.1}$$

Where l represents the longitude of the elongated specimen and l_0 the initial longitude.

4.1.2 Methodology

A machining process was identified as a successful way to procure the necessary specimens.

The length of the specimens wes determined by the largest media particle available in the market, and also, the brittleness of the material significantly restricted the shape of the samples. Unfortunately, these limitations negated the possibility of a three-point bend test.

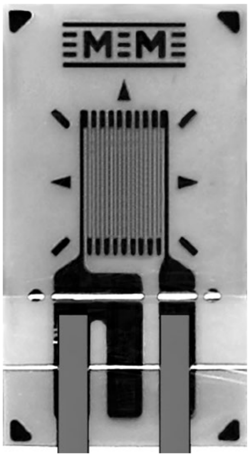
An added problem to the fabrication of the samples was the abrasiveness of the material, rapidly wearing the cutting tools.

Subsequently, the Tensile Test is the method that was selected to determine the *Modulus of Elasticity*, while the Compression Test was used to determine the *Poisson's Ratio*.

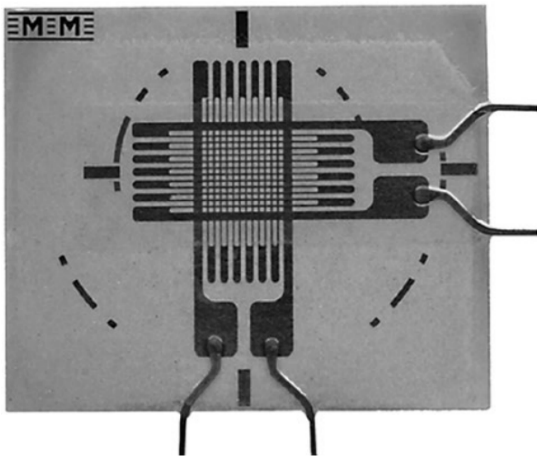
The equipment used for test was a Tinius Olsen H5KS Benchtop.
A loading cell was used for measuring the force applied to the specimen.

Two types of strain gauges were used to determine the strain. For the *Modulus of Elasticity* a linear pattern strain gauge was used. The *Poisson's Ratio* was computed using a T-Rosette strain gauge, Figures 4.2.

The strain gauges were attached to the specimens using a two-component epoxy adhesive supplied by Micro Measurement. The cure time for this type of adhesive is 6 hours at 24°. A 24 hours cure period was given to all specimens.



(a) Strain Gauge (C2A-062LW)

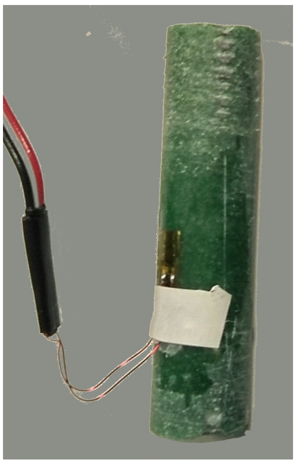


(b) Tee Rossette Strain Gauge (120WT)

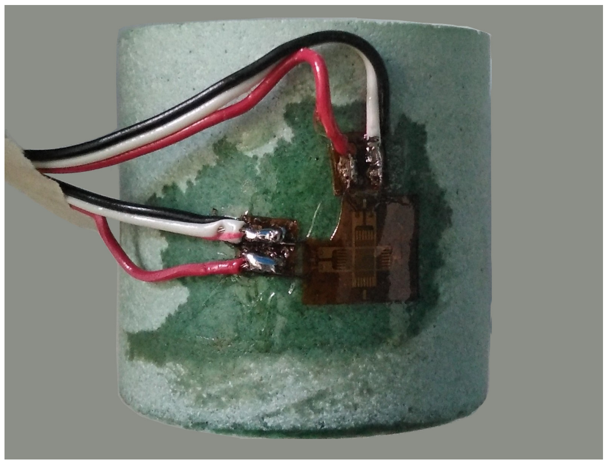
Figure 4.2: Employed Strain Gages

4.1.3 Calculation

Despite the aforementioned difficulties, two different types of specimens were obtained for each of the tests, by machining several conical media particles into cylindrical parts, Figure 4.3.



(a) Young's Specimen



(b) Poisson's Specimen

Figure 4.3: Employed Strain Gages

The tensile specimens dimensions are as follows:

Length (m)	Diameter (m)	Cross section (m^2)
$50e-3$	$11e-3$	$9.5e-5$

The dimensions of the specimens for the compressions test are:

Length (m)	Diameter (m)	Cross section (m^2)
$33e-3$	$35e-3$	$9.6e-4$

4.1.4 Results and Discussions

4.1.4.1 Modulus of Elasticity (E)

The relation between the stress and the strain have been established experimentally. This relation is known as Hooke's law. The mathematical expression for this relation is:

$$\epsilon_y = \frac{\sigma_y}{E} \longleftrightarrow E = \frac{\sigma_y}{\epsilon_y} \quad \text{Pa} \tag{4.2}$$

The subscript 'y' makes reference to the direction in which the specimens were loaded. As a result of the tensile test, the y-direction represents the direction parallel to the axis of the cylinder.

The graph below depicts with a solid blue line the relation between stress and strain obtained during the tensile test. A first degree polynomial ($y = mx + n$) was used, during the post-processing to generate the fitted curve of the experimental data, which is shown by a dotted red line. According to Hooke's law, (Equation 4.2), the slope of the polynomial represents the *Modulus of Elasticity*.

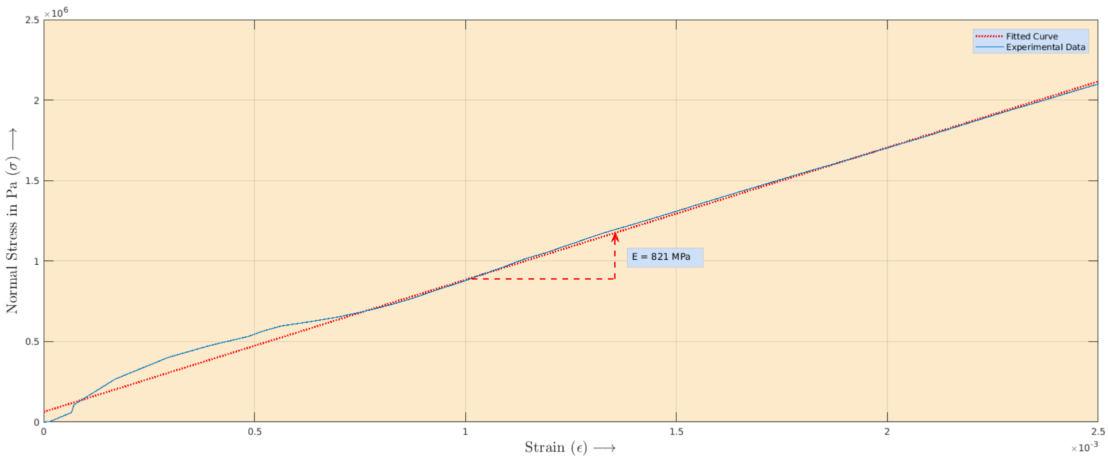


Figure 4.4: Stress-strain diagram for the general purpose media material

The average value for the *Modulus of Elasticity* for the generic purpose plastic media was:

$$E = 821 \times 10^6 \quad \text{Pa} \Rightarrow \boxed{E = 821 \quad \text{MPa}}$$

Due to the brittle characteristic of the media material the tensile test was never carried out efficiently as the specimens always failed where the clamping device was holding them.

4.1.4.2 Poisson's Ratio (ϵ)

As mentioned previously, a T-Rosette strain gauge type has been used to collect the data. This type of strain gauge provides information related to the displacements in two directions. The specimens used for this test had cylindrical geometry, all the test specimens were loaded to fail in axial compression. The Y-direction is considered to be parallel to the specimen axis while the X-direction is parallel to the cross section of the specimen.

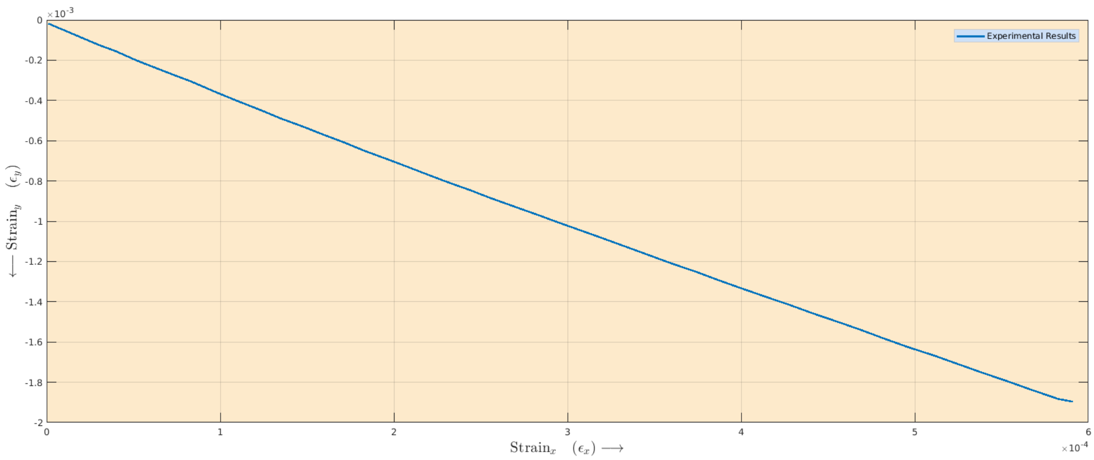


Figure 4.5: Strain$_\text{x}$-Strain$_\text{y}$ diagram for the general purpose media material

Figure 4.5 displays the relation between the axial strain (ϵ_y) and the transversal strain (ϵ_x) within the elastic limit of the material. The ratio of these two values is the constant known as *Poisson's Ratio*. The mathematical expression for the *Poisson's Ratio* is as follows:

$$\epsilon_x = -\nu \underbrace{\frac{\sigma_y}{E}}_{\epsilon_y} \Rightarrow \epsilon_x = -\nu\epsilon_y \Rightarrow \nu = -\frac{\epsilon_x}{\epsilon_y} \tag{4.3}$$

It is evident from the Equation 4.3, the relation between ϵ_y and ϵ_x is linear, as it is depicted in the graph. The minus sign represents the lateral contraction, for this case due to the stress being negative as a compressive load the value of the ratio will be positive.

The average value for the *Poisson's Ratio* for the generic purpose plastic media was:

$$\nu = 0.2887$$

4.1.5 Discussion of Experimental Uncertainty

From the Theory of elasticity it is known that within the elastic limit the *Modulus of Elasticity* and the *Poisson's Ratio* in compression are the same as in tension.

According to the Tensile Tests the average value in tension for the *Modulus of Elasticity* is:

$$E = 821 \quad \text{MPa} \Rightarrow E = 0.82 \quad \text{GPa}$$

From the Equation 4.3, it can be deduced:

$$E = -\nu \frac{\sigma_y}{\epsilon_x}$$

The average value of the *Modulus of Elasticity* obtained from the compression test is:

$$E = 4.6 \quad \text{GPa}$$

The Experimental Results disagree by:

$$\left| \frac{0.8 - 4.6}{4.6} \right| \times 100 = 82\%$$

4.1.6 Conclusion

The values obtained from the Tensile Test are considered erroneous, as all tests failed considerably early in comparison to the Compression Test, hence only the values obtained with the compression test are considered to determine the properties of the media. As a result the values for the *Modulus of Elasticity* and the *Poisson's Ratio* are as follows:

$$E = 4.6 \quad \text{GPa} \quad \text{and} \quad \nu = 0.2887$$

4.2 Frictional properties

Characterizing the effect of friction for both particle-particle (P-P) and particle-wall (P-W) interactions is highly important to achieve realistic simulations. The effects of friction on dense systems, like MF processes are crucial in contrast to disperse systems. In order to determine the correct value of the friction coefficient for both interactions, several experiments were undertaken. This included the manual 'pseudo' pin-on-disk approach. However, it was concluded that this method was unreliable due to inherent variation in applied pressure and velocity.

4.2.1 Objective

This section informs on the works conducted to virtually determine the *Coefficient of Static Friction* between the general purpose plastic media and several materials involved in the MF process. These materials are:

· The material of which the media is made. During the MF process the media particles are continuously interacting with each other.

· The lining material. The media particles also interact with the wall of the work chamber.

· The material of which the sensor case is made.

For the purpose of this thesis, the following definitions applies:

Coefficient of Static Friction (μ) between two bodies is the friction that exists between a stationary object and the surface on which it's resting.

4.2.2 Principle

Characterizing the effect of friction for both particle-particle (P-P) and particle-wall (P-W) interactions is highly important to achieve realistic simulations. The effects of friction on dense systems, like MF processes are crucial in contrast to dispersed systems. In other to determine the correct value of the friction coefficient a parametric approach was employed using several DEM simulations[1] as a starting point.

These consisted of a flat surface upon which a conical shape media particle was at rest, Figure 4.6. The coefficient of friction between the particle and the wall was initially set at 0.1. Then, the surface

[1] All simulations have been carried out using LIGGGHTS

was given inclination slowly enough to neglect the inertial forces. At a certain angle the particle started to slide. A total of ten simulations were carried out where all the properties remained constant except the value of the coefficient of friction which was incremented by 0.1 on each consecutive simulation.

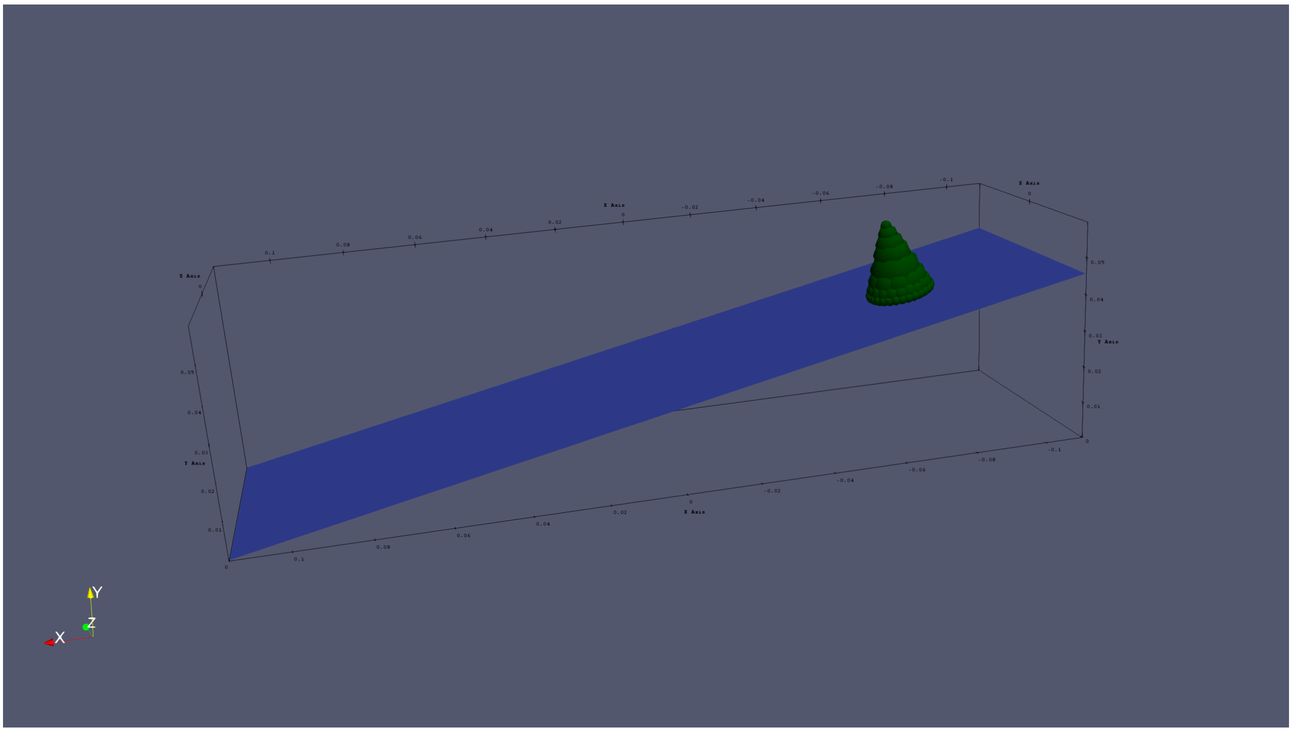


Figure 4.6: Virtual sliding plane

The angles on which the particle started sliding in each case were recorded. The graph in the Figure 4.7 depicts the progression of the sliding angle as the coefficient of friction is increased. A python script was used to identify the exact moment the particle began to slide and then extract the angle at which the plane was set. The code can be found in the Appendix A.

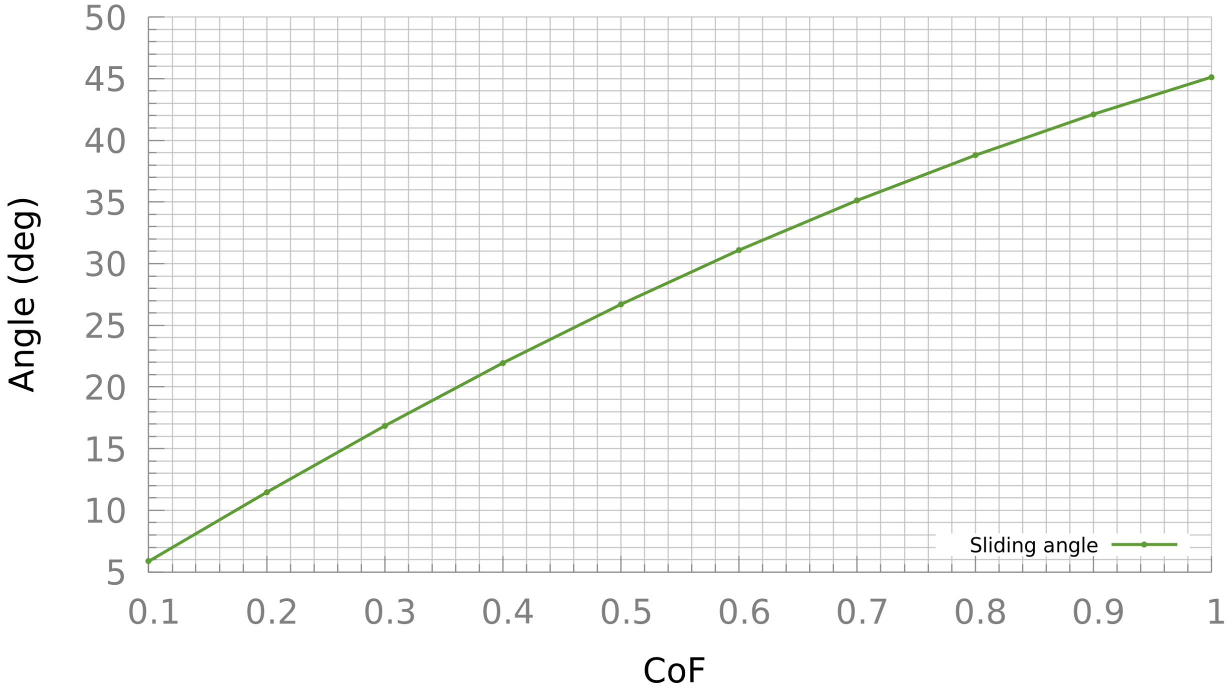


Figure 4.7: Sliding angle

4.2.3 Apparatus

A device was then designed to replicate the simulations. The Figure 4.12 depicts the device and a media particle in contact with a flat surface of media which was previously machined down into a media disc with two perfect parallel faces. The flat surface was inclined progressively until the particle started sliding. The angle was found using a high precision protractor. These experiments were repeated for all the material pairs commented on within this thesis. The angles were retrieved and friction coefficients identified using the graph.

4.2.4 Results

The Table 4.1 presents the values of the coefficient of friction for the different pairs of materials. The pair Sensor-Sensor was not computed as there is only one particle in the simulation, therefore it will never impact any other particle with the same properties.

Table 4.1: Coefficients of Friction

Coefficient of friction for each pair				
Media-Media	Media-Sensor	Media-Wall	Sensor-Wall	Sensor-Sensor
0.58±0.039	0.34±0.051	0.86±0.094	0.86±0.073	n/a

Figure 4.8: Inclining plane device

Valid results have been achieved using these parameters.

4.3 Weight studies

The density of the virtual bulk material needs to be adjusted to obtain the correct bulk density of the real material. Three different approaches were needed in order to accurately establish density values for the media material, all of them based on the relation between mass and volume.

1. The first approach considered the media particle as an ideal cone, the volume of which can be determined by the expression $V = \frac{1}{3} \times \Pi \times r^2 \times h$. A high precision laboratory scale was used to obtain the mass, the value of which was established as the average of 100 media particles individually weighed. The simulations run using the value retrieved from this approach were wrong to the naked eye.

2. In this approach, the volume of the particle was obtained using a digital caliper to retrieve the relevant dimensions of the particle and a CAD package to model a more realistic geometry. The value retrieved considerably improved the results of the simulations.

3. This approach was set in an attempt to validate the results obtained in the previous approach. It established the volume by simplifying the geometry of the media particle. A lathe was used to machine down several conical media particles into perfect cylinder specimens. The mass of the new geometries was obtained via a high precision scale. The average of the densities of the different specimens was set as the density value retrieved from this approach.

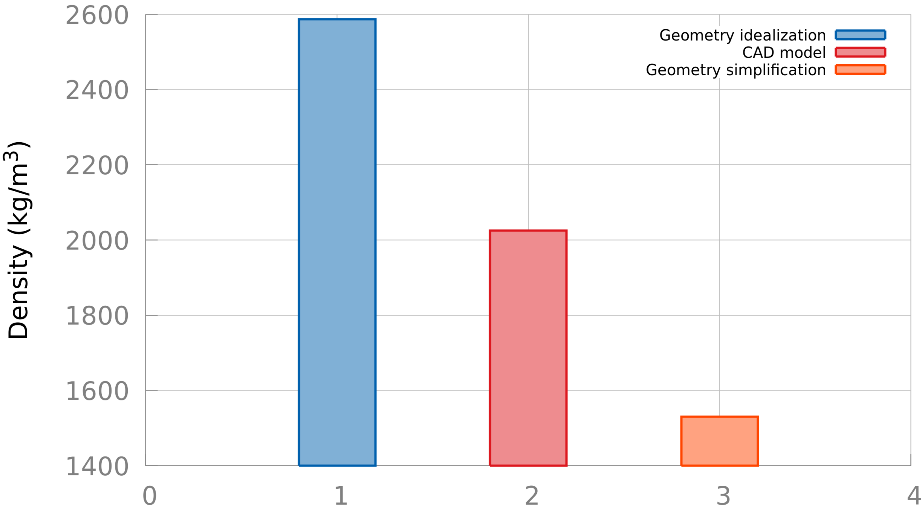


Figure 4.9: Plastic media density

Figure 4.9 depicts the values of the densities obtained by the different approaches. As can be observed, the differences in magnitude are considerable. The latest value was set as default value for the density and used in the simulation commented on within this thesis. It can be concluded from these results that the media are not truly conical.

4.4 Damping properties

The Coefficient of Restitution (e) between two impacting bodies is defined as the relative velocity of departure divided by the relative velocity of approach. This is one of the most challenging parameters to establish. The morphology of the commercial media particles as well as its condition of being a brittle material complicate the task of identifying an appropriate test.

4.4.1 Objective

This document informs on the works conducted to determine experimentally the *Coefficient of Restitution* between the general purpose plastic media and several materials involved in the MF process. These materials are:

- The material of which the media is made.
- The lining material.
- The material of which the work-piece is made. Two different material have been taken into consideration in these tests.
  - Steel.
  - Additive Manufacture Titanium Alloy.

4.4.2 Definitions

For the purpose of this report, the following definitions applies:

Coefficient of Restitution (e) **between two impacting bodies** relative velocity of departure divided by the relative velocity of approach.

4.4.3 Principle

Determination of the coefficient of restitution by dropping a sphere machined out of a conical media particle from a fixed height onto the test specimens and measuring the height of the rebound.

4.4.4 Apparatus

Media sphere

Sphere of diameter (25.50$\pm$0.1) mm.

Sphere-release apparatus

It consists of a clamping device connected by two steel ϕ6mm rods to an aluminium base. An adjustable ruler is used to measure both the dropping and the rebound heights. The clamping device, Figure 4.10, runs along the rods to set the right height for every dropping scenario, the position from which the sphere is released can also be adjusted horizontally, so the sphere and the centre of the specimen can be aligned.

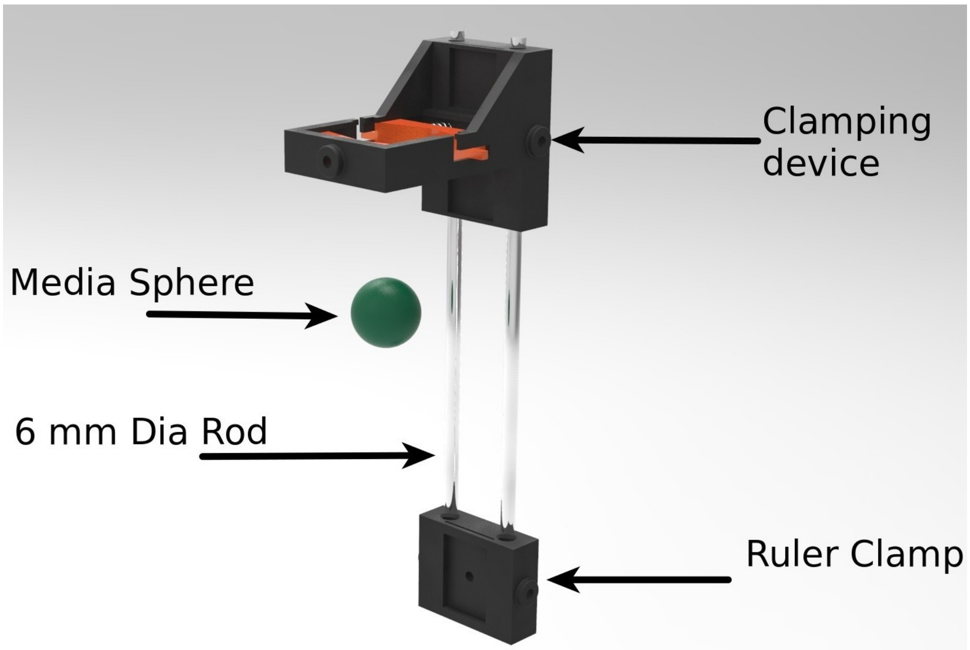


Figure 4.10: Sphere-release apparatus

High-speed camera

A high speed camera has been used to record the movement of the sphere and to collect the height of the rebound, Figure 4.11.

Figure 4.11: Phantom VEO 710S

Laptop

A laptop was connected to the camera to collect and process the recordings.

4.4.5 Test specimens

One specimen was used for each of the tests. Over 50 drops were used for each of the specimens. The dimensions of the specimens varied depending upon the material.

For the media-media impact a cylinder of media was machined out of one of the biggest cones. The dimensions were $\varnothing 35 \pm 0.1$ mm and 35 ± 0.1 mm height.

The specimen used for the impact between media and the additive manufactured titanium alloy was a solid block of 25.05 ± 0.05 mm $\times$ 25.05 ± 0.05.

The steel specimen consisted in a rectangular section bar $46.5 \pm 0.05 \times 6.25 \pm 0.05$.

The roughness of the impact surfaces was treated to be low within the materials' limitations.

4.4.6 Procedure

Several steps are to be followed to successfully conduct the test:

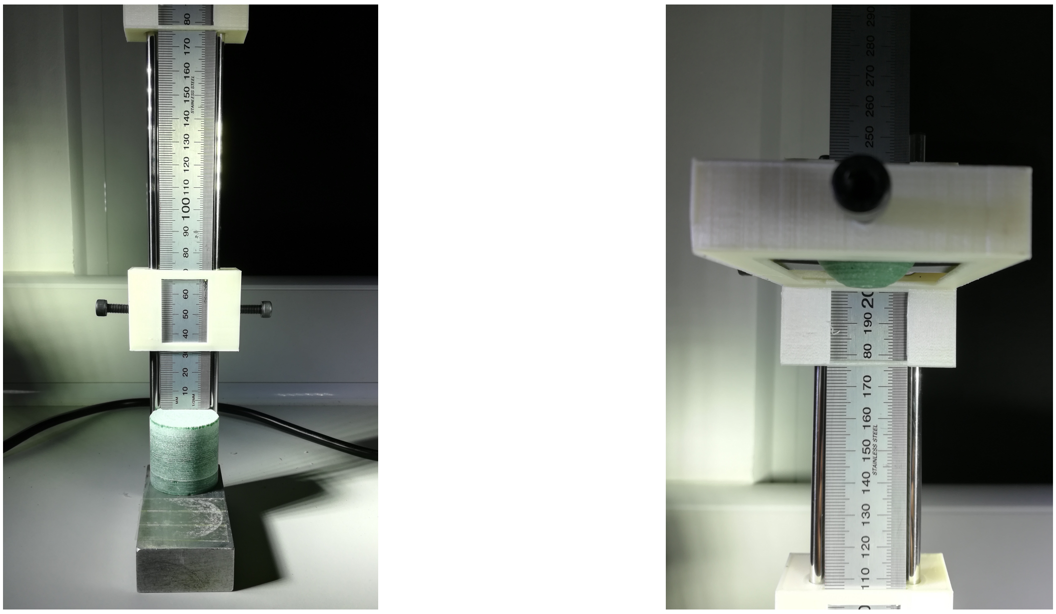
1. Adjust the position of the ruler according to the height of specimen so the bottom of the ruler is flush with the impact surface, Figure 4.12. **2.** Glue the specimen to the aluminium base using a strong industrial adhesive. **3.** Adjust the height of the clamping device to ensure that the sphere is always released at the same height. **4.** Secure the position of all the elements by tightening the lateral and rear screws. **5.** Clamp the sphere using the clamping device. **6.** Activate the camera and release the sphere using the trigger mechanism of the clamping device. **7.** Check the trajectory of the sphere from the recording. **8.** Read and store the rebound height.

4.4.7 Calculation

For a sphere impacting a horizontal static surface, the coefficient of restitution (e) is calculated using the following equation:

$$e = \frac{v}{u} \tag{4.4}$$

where:



(a) Ruler adjustment

(b) Clamping device

Figure 4.12: Sphere-release apparatus

v is the velocity of departure (rebound).

u is the velocity of approach.

$$\frac{mv^2}{2} = mgh_2 \Rightarrow v = \sqrt{2gh_2} \tag{4.5}$$

Where:

m is the mass.

h_2 is the height of the rebound.

g gravity

Thus

$$\frac{mu^2}{2} = mgh_1 \Rightarrow u = \sqrt{2gh_1} \tag{4.6}$$

h_1 is the height of drop.

Hence, substituting the expression of the Equation 4.5 and 4.6 into the Equation 4.4:

$$e = \frac{\sqrt{2gh_2}}{\sqrt{2gh_1}} \Rightarrow e = \sqrt{\frac{h_2}{h_1}} \tag{4.7}$$

4.4.8 Results

The height of the drop was fixed to 200 mm for all the drops. The rebound height varied depending on the impacting material. The Table 4.2 presents the values of the height in mm for the ten best drops within the 50 conducted drops. The criteria followed to select these drops over the rest of them were:

- No deviation in the trajectory of the rebound.
- No rotation of the sphere.

Table 4.2: Rebound height in mm

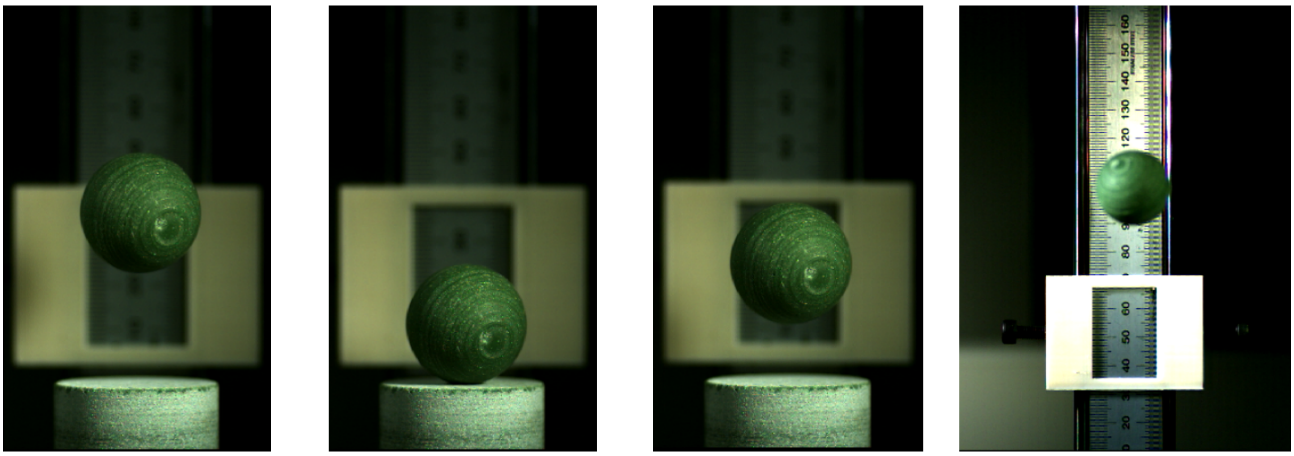
	Impact materials		
	Media	Steel	AM titanium
Rebound height (h_2)	80	83	129
	98	86	120
	94	92	128
	89	101	128
	89	92	124
	88	83	129
	87	93	125
	90	82	128
	87	100	127
	96	81	125

The Coefficient of Restitution is calculated using the Equation 4.7 and the information contained in Table 4.2

Table 4.3: Coefficients of Restitution

	Impacting surface materials		
	Media	Steel	AM titanium
Media material	0.632	0.644	0.803
	0.700	0.656	0.775
	0.686	0.678	0.800
	0.667	0.711	0.800
	0.667	0.678	0.787
	0.663	0.644	0.803
	0.660	0.682	0.791
	0.671	0.640	0.800
	0.660	0.707	0.797
	0.693	0.636	0.791

The Figure 4.13 depicts different moments during the dropping test. The three first sub-pictures are moments of the same test. Per contra, the fourth sub-picture was extracted from a different test due to the impossibility of recording two scenes at the same time. Within the first sub-pictures, it can be appreciated that the sphere does not rotate.



(a) Before the impact **(b)** At the impact **(c)** After the impact **(d)** Rebound height

Figure 4.13: Drop sequence

Table 4.5 collects the average value for the Coefficient of Restitution between the media and all the specimens.

Table 4.4: Coefficients of Restitution (average)

	Media	Steel	AM titanium
Media	0.667	0.668	0.795

4.4.9 Standard Deviation

The Standard Deviation of the results was calculated using the results displayed on Table 4.3 are as follows:

Table 4.5: Standard deviation from the mean

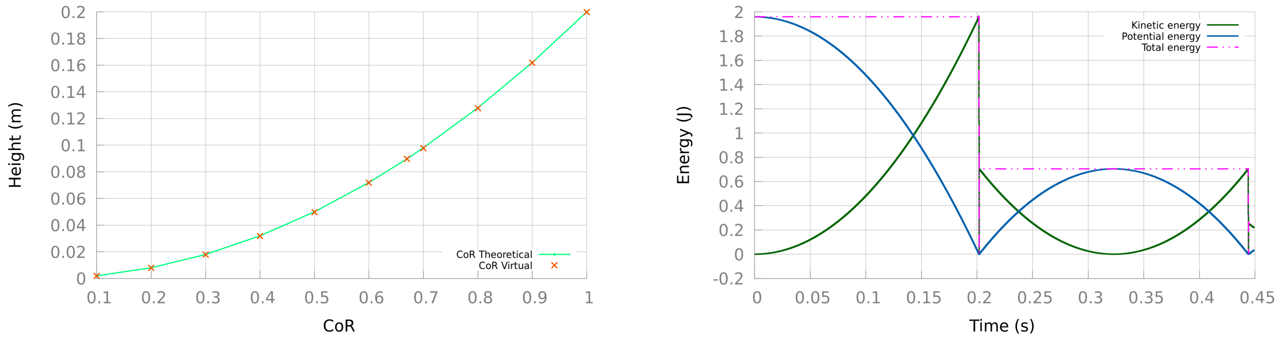
Media	Steel	AM titanium
Average		
0.667	0.668	0.795
Standard Deviation		
0.019	0.028	0.13

It can be observed that when the media collides with AM titanium alloy or the bowl lining material the dissipated energy is lower than in the collision with the rest of the materials. The Standard Deviation for the AM Titanium is greater than the others due to the roughness of its surface.

4.4.10 Experimental approach versus virtual approach

This experiment is a very time-consuming process, moreover the high-speed camera is an expensive article, not always available in the engineering laboratories. On the other hand, the pairs of materials under study might not have the adequate geometry for the test or the possibility of being malleable enough to obtain the right shape. These limitations were the stimulus to consider a different approach.

Firstly, two different methods were considered. The height of the first bounce was first calculated theoretically, using Equation 4.7. Then the free fall was simulated using LIGGGHTS. A python code was developed in order to calculate the theoretical height and also to extract the information from the data files outputted by the DEM software. Figure 4.14a depicts the heights of the first bounce. The green curve plots the results obtained from the theoretical approach while the orange crosses represent the heights obtained virtually through DEM. It can be observed that the results are nearly identical, which means DEM obtains the same result as the classic physic using a different approach.

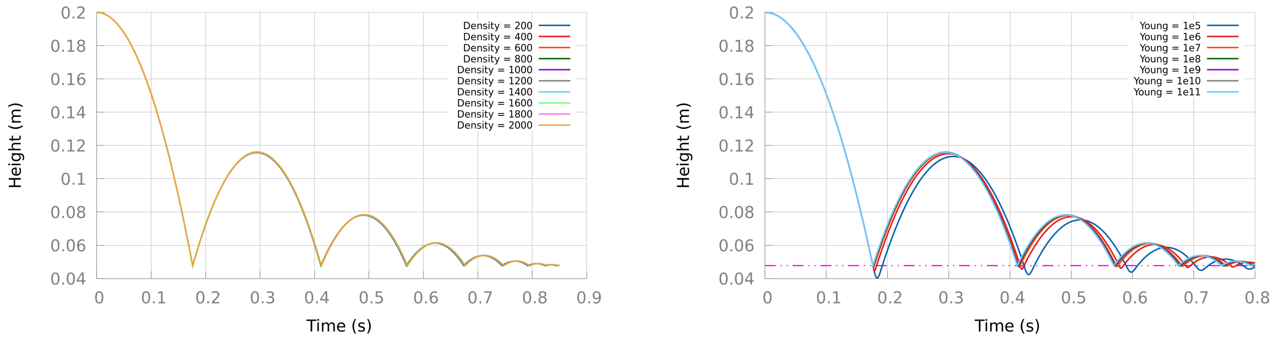


(a) Theoretical vs Virtual CoR

(b) Energy Balance for DEM simulation

Figure 4.14: Theoretical Vs Virtual(DEM) results

The virtual results were studied from the point of view of the law of conservation of energy. Figure 4.14b represents the balance of energy for the system during the two first bounces. As observed the results satisfy the mentioned law. The values expressed in these two graphs validate the result of the DEM against the physical models.

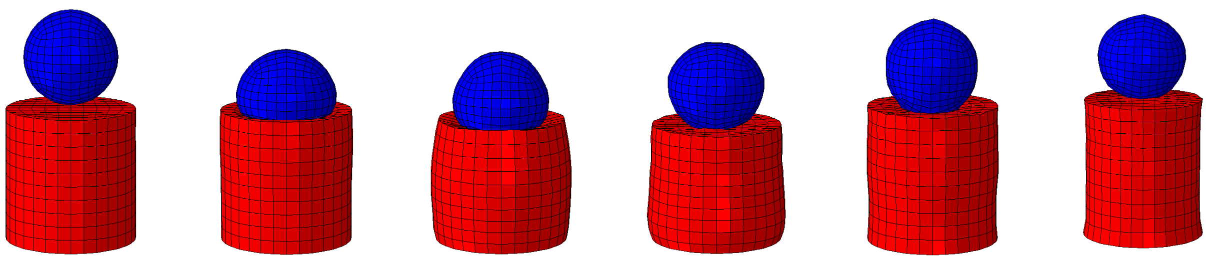


(a) Sphere behaviour depending on its density

(b) Sphere behaviour depending on its Young Modulus

Figure 4.15: Theoretical Vs DEM Virtual results

Secondly, the approach of using the explicit FE method, to recreate the drop test and to virtually obtain the coefficient of restitution, was considered. In this type of simulation the variables are the density of the material, its geometry, the height from where the sphere is released and finally, its elastic properties: Young's modulus ($\mathbf{E}$) and Poisson's Ratio (ν). The first step in this consideration was to study the behaviour of the sphere using DEM when all the variables were constrained and only the density varied. Figure 4.15a depicts the vertical position of the sphere at all times after being released for different values of density. It is observed that for DEM a change in the density does not produce any difference in the behaviour of the sphere so it can be concluded that the CoR is not affected by the density. Figure 4.15b represents the change in the sphere's behaviour when all variables are fixed but the **E**. The horizontal dashed line at the bottom represents the top of the impacting surface. For a high value of the **E** the stresses produced at the impact are not high enough to produce a considerable deformation in the system. On the contrary, for a low **E** the deformation is noticeable. Figure 4.16 shows the impact sequence for a very elastic material. In this case, due to the deformation, the parts are in contact for a much longer period of time. The deformation phenomenon is represented in the graph of Figure 4.15b by the particle travelling beyond the impacting surface. This happens for those values of **E** below 1×10^6Pa. However for DEM this behaviour of the system is independent of the Coefficient of Restitution, although the maximum height after the first bounce is displaced with respect to those with high **E** its value remains consistent. This reveals one of the limitations of the DEM. Section 7.1.1 will explain this in a deeper way.

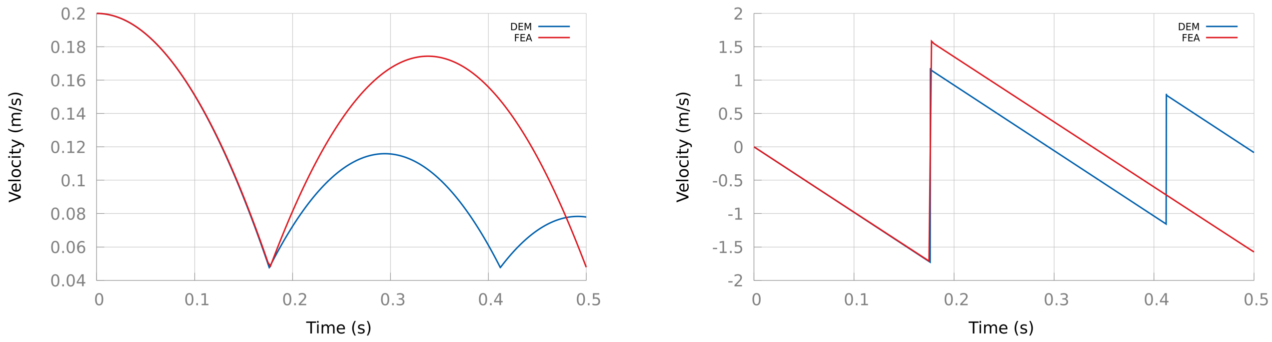


(a) dt=1 **(b)** dt=3 **(c)** dt=5 **(d)** dt=7 **(e)** dt=9 **(f)** dt=11

Figure 4.16: Impact for a $\mathbf{E} = 1 \times 10^5$

In the light of what has been revealed, for values of stresses producing large deformations this approach is inadequate. On the other hand, the behaviour of the sphere for a more rigid material was also investigated. Figure 4.17 shows the comparison between the heights and velocities obtained from DEM and Finite Element Analysis (FEA). For these two simulations identical values were used:

- The sphere and cylinder's geometries were identical to the experimental ones. As was the releasing height.
- The value of the density was obtained from the weight studies.
- The elastic properties were those revealed in Section 4.1. For this case $\mathbf{E} = 4.6 \times 10^9$ Pa and $\nu = 0.2887$.



(a) Heights comparison (DEM vs FEA)

(b) Velocities comparison (DEM vs FEA)

Figure 4.17: DEM Vs FEA

It is observed how the maximum height of the first bounce of the sphere for FEA is significantly greater than for DEM. Consequently, the maximum velocity in DEM is considerably lower in comparison to FEA's result. A difference of 18% is observed between both approaches.

4.4.11 Discussion

The average values of the Coefficient of Restitution obtained by this test are accepted due to the low uncertainty (less of 10% for all the pair combinations). These parameters are used in all the DEM simulations treated in this thesis.

The virtual approach to obtain the Coefficient of Restitution replicating the test via FEA is inefficient for low values of Young's Modulus. Moreover, for higher values the discrepancy is big enough to be cautiously considered. If the system to be studied is very dense in particles then this approach should be discarded, on the contrary if the particles of the system are dispersed then it would provide an insight into the system behaviour, although a serious analysis is recommended using an experimental approach to identify the Coefficient of Restitution.

A convergence study of the mesh in FEA could reduce the deviation between the results. Unfortunately, this study is beyond the current scope.

4.5 Conclusions

There is no one test that will determine all the material properties to fully define the DEM model. Establishing all these properties is a challenging task, mainly because of the difficulty in identifying the suitable set of tests and also due to the reduced number of media particle morphologies found in industry.

Chapter 5

Vibratory mass finishing studies

This chapter investigates the flow behaviour of the media inside the MF system VM375Y, Figure 5.1. This consists in a trough-work-chamber of L1,799 mm × W1,191 mm × H1,125 mm lying on a set of ten springs evenly distributed on a supporting base. An electrical motor of 6kW is attached to the base of the trough. An off-centre weight is fixed to the end of its drive shaft. The direction of the axis of the shaft coincides with the longitudinal direction of the trough. The rotation of the off-centre weight produces a continues vibration of the work-chamber which energizes the media inside. The working cycle range of the motor is $1,500 - 3,000\,\text{rpm}$.

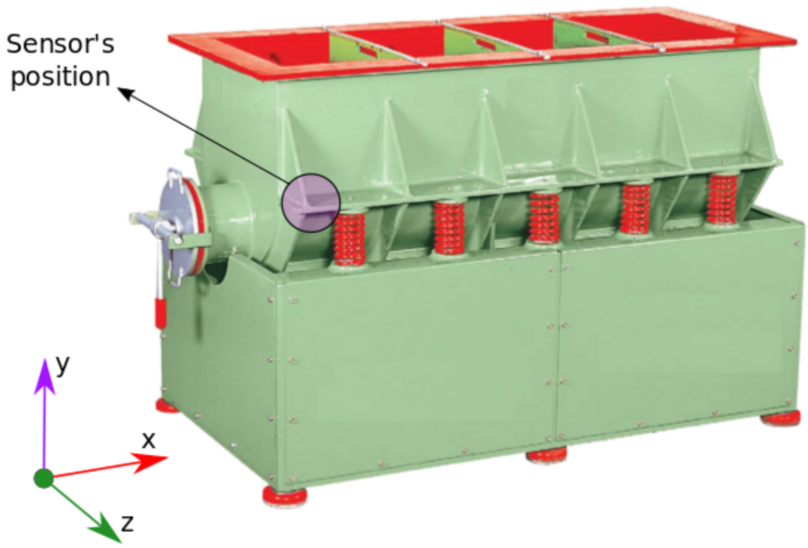
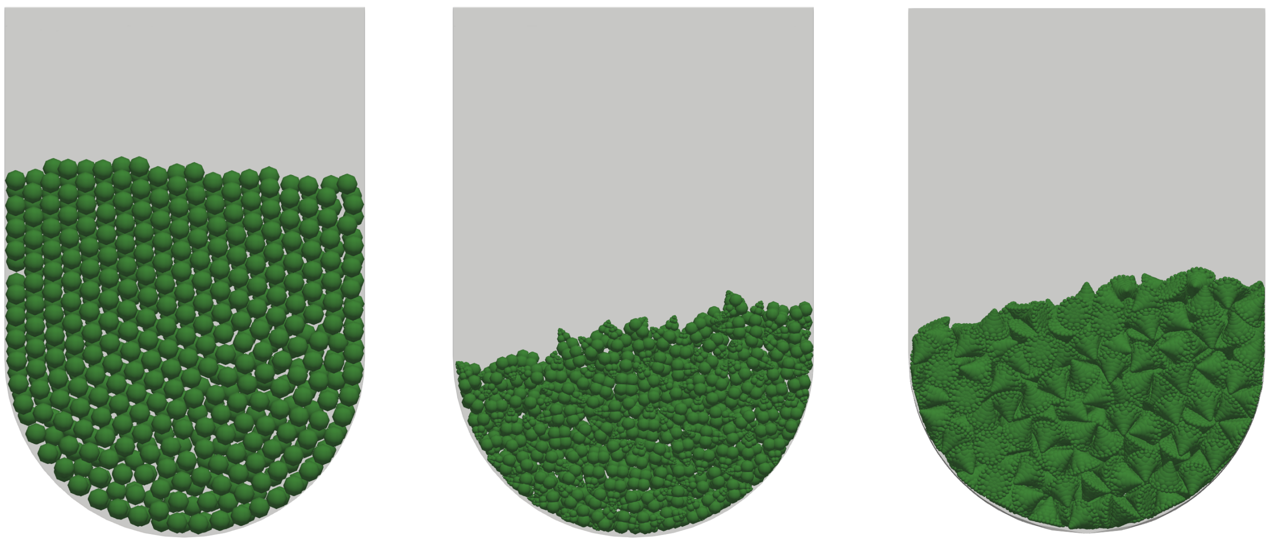


Figure 5.1: Vibratory mass finishing system: VM375Y

The study of this process is a continuation of the work started by Jamal[6] on DEM simulations as a manner to gain insight into the MF processes. Jamal simulated a slice of media within a smaller trough with a few tens of particles. The same trough was used as the starting point of the studies presented in this thesis.

5.1 Particle definition

The first step into the simulation of the media was to correctly define the geometry of the single media particle. The geometry of the actual media particle is conical as shown in Figure 5.3c. To this purpose three different simulations with identical numbers of particles were designed. The first one characterized the single media particle as its equivalent sphere. Figure 5.2a shows the results of this simulation; as can be observed the distribution of particles is a poor approximation of the real distribution. Due to the high voidage, the media density is considerably lower when using this simplification of its geometry. For the second and third simulations the concept of multi-sphere particle was introduced. This consists in an approximation of the real morphology of the media particle by clumping together several spheres, once they are together, they behave as a single body. Figures 5.2b and 5.2c represent two different multi-sphere particles. The first one uses 17 spheres to model the media particle while the second one uses 65 spheres. The distribution of the media changes significantly with respect to the previous simulation. The media density increases as the voidage reduces. It is obvious that a greater number of spheres delivers a more real geometry.



(a) 1 sphere per media particle **(b)** 17-sphere media particle **(c)** 65-sphere media particle

Figure 5.2: Virtual media distributions

The main inconvenience of the multi-sphere technique is the elevated consumption of computational resources. It is key to define the geometry with the minimum possible number of spheres. A 17-sphere media particle produces an acceptable distribution within the trough slice however it does not produce accurate results of its behaviour when they are energized. A 65-sphere media particle provides good results in the distributional sense and also in the behavioural sense.

Figure 5.3 compares the two virtual models with the real media particle. The most abstract model corresponds to the equivalent sphere of the average media particle. The 65-sphere model is a more concrete representation of the media particle. The average weight of the media particles is assigned to both models. All simulations presented in this chapter are based on the 65-sphere model.

(a) Spherical model **(b)** Multi-sphere model **(c)** Media particle

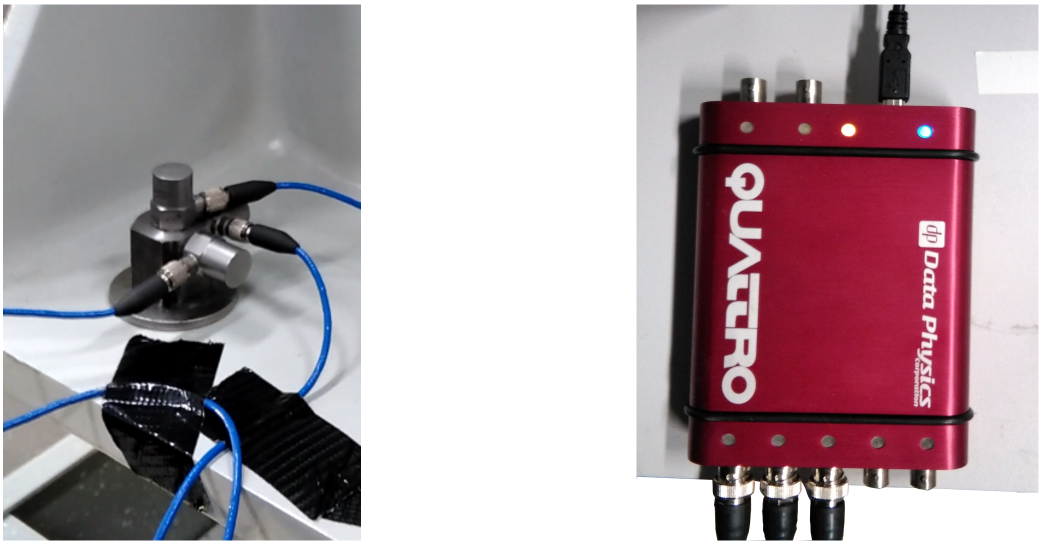
Figure 5.3: Media particle definition

The media simulated in this study is composed by a plastic bonding agent and abrasive material, its composition as well as the sizes and density determine the rate of material removal and the achievable surface finish. As a common rule large, heavy media cuts more aggressively and leaves a rougher surface. On the contrary, small, lightweight media is less aggressive and more suitable for smoother surface requirements. Furthermore, smaller media will provide better coverage. The media employed in this work is the general purpose deburring media.

The properties of the media material used for the simulations discussed in this document were obtained experimentally and are presented in Chapter 4.

5.2 Vibration analysis of the trough VM375Y

The study of the vibration source of the MF system VM375Y is fundamental in the exercise of simulating this finishing process. To this end three high sensitivity uniaxial piezoelectric sensors were used. This type of sensor is usually employed for modal analysis, environmental testing, vibration isolation and automotive crash testing. They were mechanically fixed to the trough using the stud-mount option. A steel block with three orthogonal faces was specially machined for the tests, a sensor was attached to each of those faces and then one of the faces of the steel block was aligned with the longitudinal direction of the trough. Figure 5.4a shows the sensor's arrangement employed in this task. PCB website (https://www.pcb.com) has available the full specifications of the sensor, model number: *352C33*.



(a) Stud-mounted uniaxial piezoelectric sensors

(b) Data set unit

Figure 5.4: Vibration study equipment

The sensors were configured for a sample rate of 100 kHz, to ensure Shannon's theorem[74], and then connected to the Data Acquisition Set, Figure 5.4b. Data acquisition enables gathering signals from the measurement sources and digitizing the signals for storage, analysis, and presentation on the PC. All this data was exported in the form of a large array into a text file that was then post-processed using Matlab®. The Matlab's codes used to post-process all the information can be found in the Appendix B.

This section is divided into two parts. The first part comprehends the result from the modal analysis performed on the empty system. The second part sheds lights into the behaviour of the system when the motor is running at $1,500\,\text{rpm}$, different loading scenarios were considered.

5.2.1 Experimental modal analysis

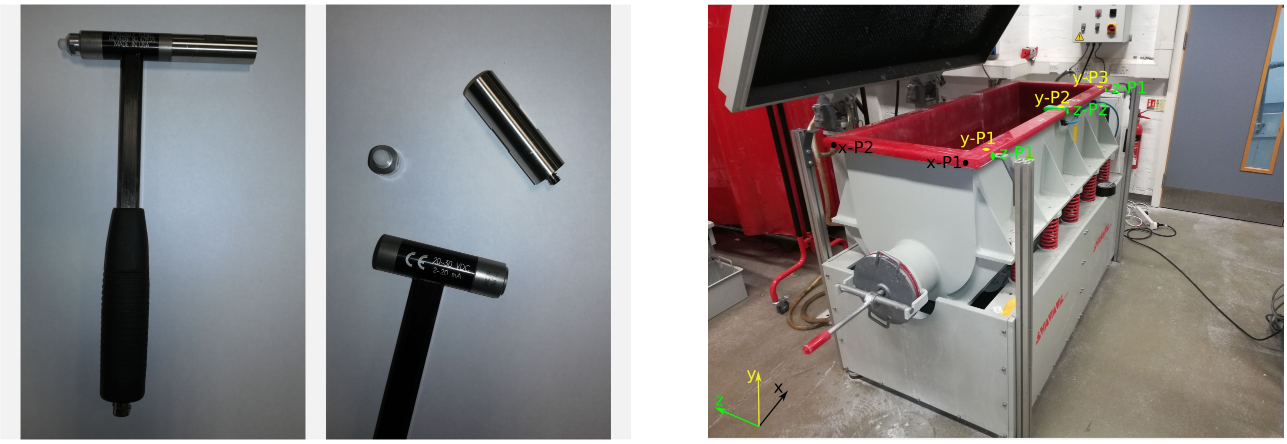
The modal analysis of the system consists in finding the natural frequencies of it. The natural frequency of a system is defined as the frequency at which the system vibrates when it is displaced from its state of equilibrium and it is not under the influence of an external force. Equation 5.1 shows the mathematical expression for the natural frequency f, where m and k refer to the mass of the system and its stiffness, respectively.

$$f = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \quad \text{Hz} \tag{5.1}$$

By definition, to know the natural frequencies f of the MF system it is necessary to displace it from its equilibrium position and count the time needed for a cycle. For the latter, the accelerometer displayed in Figure 5.4a was used. The way to displace system from its equilibrium position is by hitting it. Figure 5.5a shows the impulse hammer used to this aim. The total mass of the hammer was 210 g. The material of the tip was medium-hardness white plastic. The softer the tip is the narrower the range of frequencies that are excited, on the contrary hard tips can energize wider ranges of frequency although more noise is added to the vibrational signal. For this case only the frequencies within the range $[0\,\text{Hz}-150\,\text{Hz}]$ were studied as the displacements of the trough produced by frequencies above this range are negligible to this study.

Several impacts were performed across different points and directions. Figure 5.5b shows these points. The nomenclature used to identify the points is *axis-name*. The first part refers to the direction of the impact and the second part the name of the point in that specific direction. For example, **x-P1**: the **x** represents the x-axis direction thus the impact of the hammer was in this direction. **P1** is just the name given to the impact point.

In order to have a better understanding of the vibratory phenomenon the three orthogonal directions were studied separately. These directions are defined as follows:



(a) Impulse hammer **(b)** Impact points

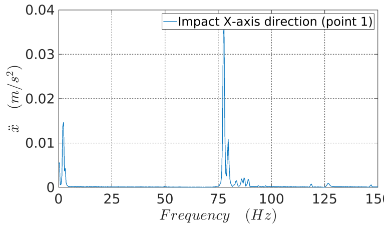
Figure 5.5: Modal study

- The x-axis corresponds to the longitudinal direction of the trough.
- The y-axis corresponds to the vertical direction of the trough.
- The z-axis corresponds to the transversal direction of the trough.

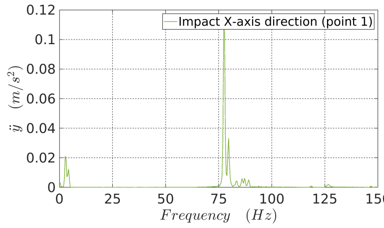
There are two different ways of displaying the results of the test. The time domain graph, that shows the behaviour of the machine over time and the frequency spectrum, that provides information about the frequencies that govern the cyclical behaviour of the movement. Figure 5.6 shows the frequency spectrum in the three-axis directions when the system is impacted in different points and directions. As can be observed there is a clear natural frequency around 80 Hz and another around 120 Hz.

5.2.2 System's vibration when submitted to an external harmonic force

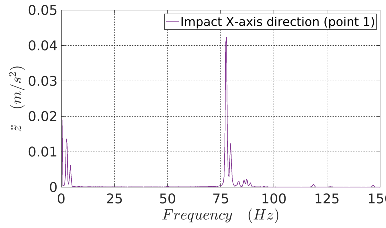
This part of the document studies the effects of a harmonic excitation on the MF system. As previously mentioned the system was studied when the motor was working at 1,500 rpm. The signals produced by the vibration of the system were analysed in order to gain an insight into its behaviour. The signal analysis is a powerful tool that provides a better understanding of how the different conditions affect the way the system performs. Two different sets of data were retrieved from the test. The first set contained the information related to the acceleration over time in the directions of the aforementioned axes. The second set included the frequencies governing the vibration, frequency spectrum.



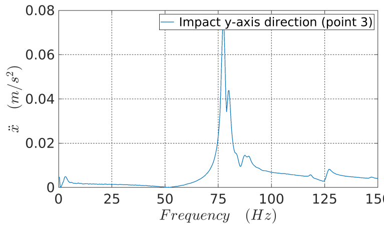
(a) Natural frequencies x-direction (x-P1)



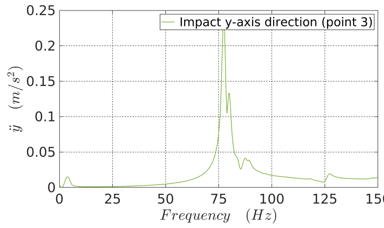
(b) Natural frequencies y-direction (y-P1)



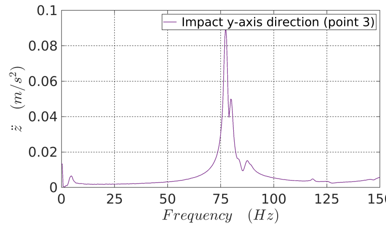
(c) Natural frequencies z-direction (z-P1)



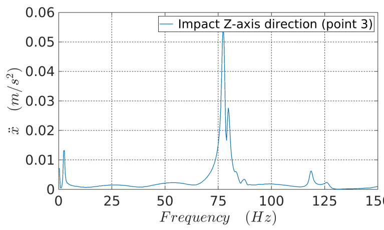
(d) Natural frequencies x-direction (x-P3)



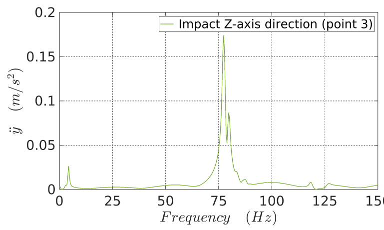
(e) Natural frequencies y-direction (y-P3)



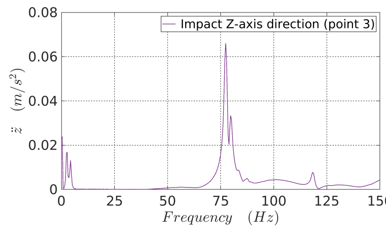
(f) Natural frequencies z-direction (z-P3)



(g) Natural frequencies x-direction (x-P3)



(h) Natural frequencies y-direction (y-P3)



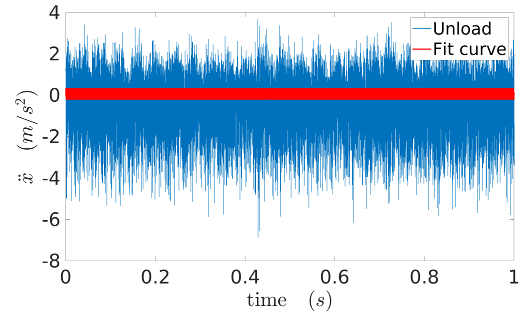
(i) Natural frequencies z-direction (z-P3)

Figure 5.6: Natural frequencies in the different directions

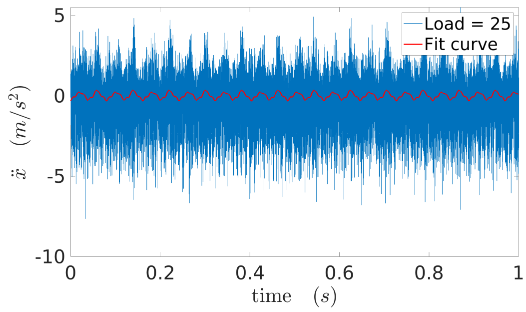
Time domain analysis

Firstly, the time domain was studied. To this end, the data was organized into four columns of values. The first column gathered the time-step at which the sensors had measured the acceleration, the other three columns comprised the values of the acceleration for each of the three directions ($\ddot{X}$, $\ddot{Y}$ and $\ddot{Z}$). These columns were divided into four different arrays. A discrete time signal was obtained by plotting the accelerations over time.

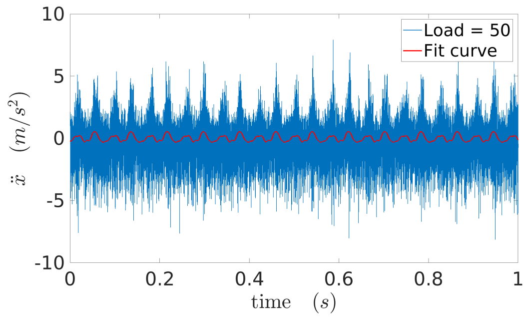
The graphs in Figure 5.7 depict in blue the acceleration in the direction of the x-axis as a function of time, when the motor is working in its steady-state regimen at 1,500rpm. Four different loading scenarios were considered. In the first scenario the system was empty, had no extra load. For the following scenarios 25kg, 50kg, 100kg were consecutively added. As noted, these are complex signals that cannot be represented by a single mathematical expression. However, Fourier series can be used to find a mathematical expression that adjusts to the signal. The red curve represents the Matlab® fit function to fit a Fourier model to the data. The fit curve is governed by a trigonometric expression such as the one presented in Equation 5.2. To define the fit curve in the figures eight terms were



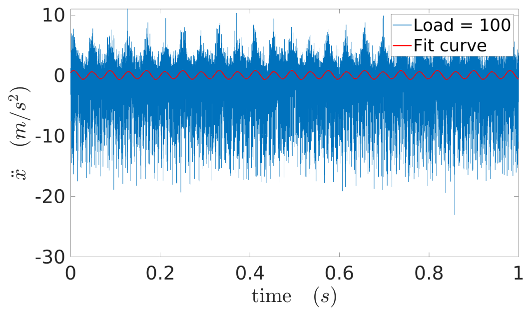
(a) x-axis direction (Unloaded)



(b) x-axis direction (25kg)



(c) x-axis direction (50kg)



(d) x-axis direction (100kg)

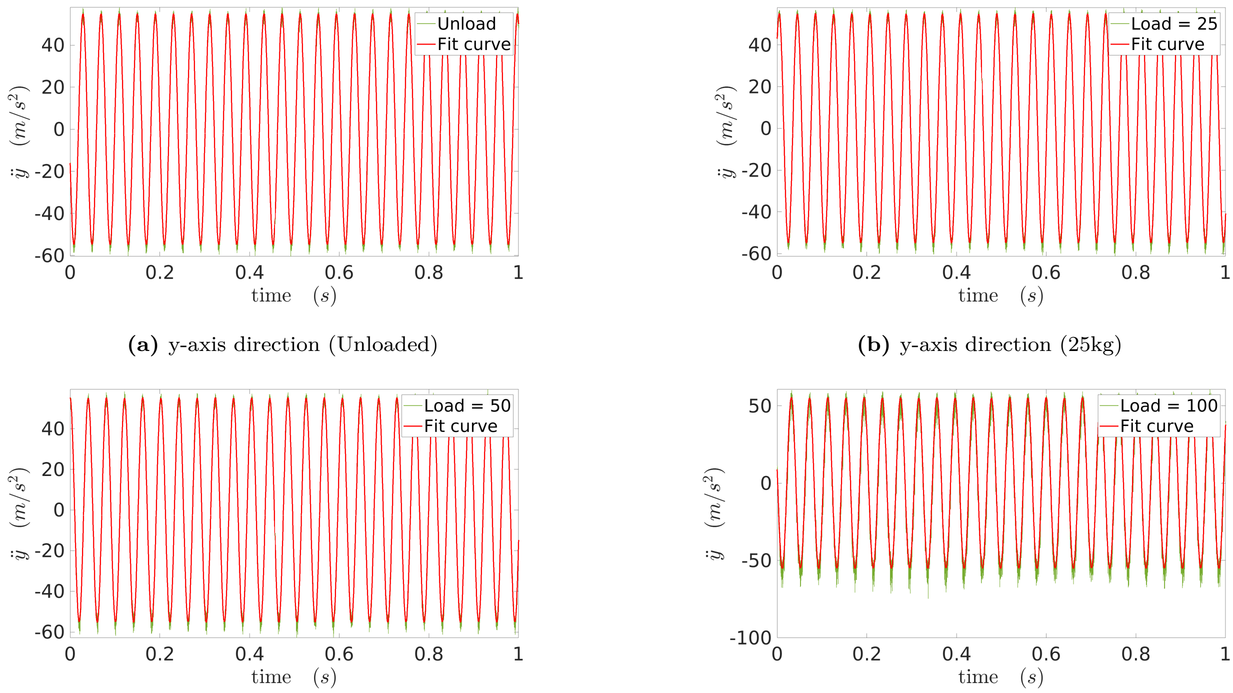
Figure 5.7: Time domain in **x-axis direction** for all loading scenarios (full range)

employed ($n = 8$). The graphs in Figure 5.10 show a zoom into the signal, plotting only the temporal range [0s- 0.04s]. It is noted that eight terms represent a poor fit to these particular signals, for a better fit many more terms are needed.

$$y(t) = a_0 + \sum_{i=1}^{n} a_i \cdot \cos(t \cdot w) + b_i \cdot \sin(t \cdot w) \tag{5.2}$$

These signals shed light on how the load affects the behaviour of the system in this specific direction. When the system in unloaded the amplitude of the acceleration is lower than for the loaded scenarios and the signal is more symmetric with respect to the axis of the abscissas. On the contrary, the more loaded the system is the more erratic it becomes. In the case of the 100kg load the amplitude of the acceleration augments by 50% in respect of the unloaded scenario, as a result of the inertia. For this same reason its signal is less symmetric. The wave length of the signal is difficult to establish for this plotting range. This suggests that the periodic movement is much quicker than in the other directions as is shown below.

The graphs in Figure 5.8 show the signal in the time domain for the y-axis direction. The range of time is identical to the signal in the x-axis direction, per contra the signals are very different. The first



(a) y-axis direction (Unloaded)

(b) y-axis direction (25kg)

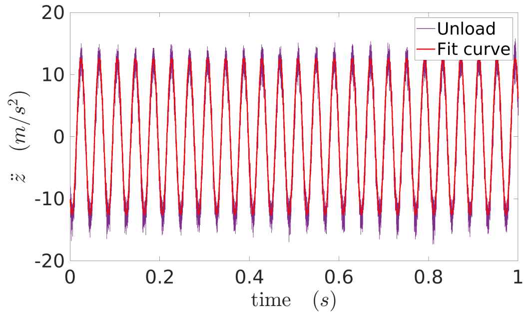
(c) y-axis direction (50kg)

(d) y-axis direction (100kg)

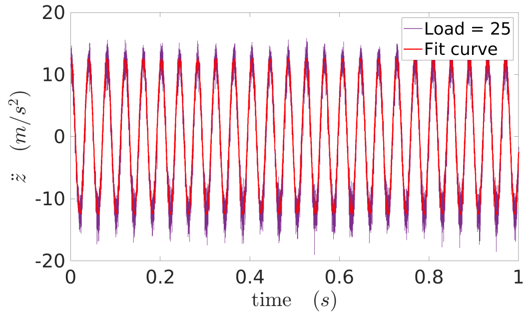
Figure 5.8: Time domain in **y-axis direction** for all loading scenarios (full range)

prominent difference is the amplitude of the signal, in this direction it is considerably greater than in the other direction. It is also appreciated that its magnitude does not significantly vary as the load is incremented. This suggests that the inertia of the weight fixed to the end of the drive shaft of the motor is greater than the inertia of the bulk. Furthermore, the circular trajectory of this weight is contained in the YZ plane which especially determines the behaviour of the signal in the y- and z-axis directions. To support this statement, it is necessary to take a closer look to the fit curve. As before, this used eight terms, in this case they were enough to define the sinusoidal tendency of the signal. As per the graphs, the period of the fit curve is approximately 0.04 s, then its frequency is 25 Hz which is equal to 1,500 rpm. This is the working velocity of the motor. This fact confirms that the inertia of the motor is determined by the frequency of the vibrational signal in a more decisive way than the load: the weight of the bulk. Although the amplitude and the frequency of the fit curve remains unaltered for the different loading steps the weight of the bulk affects the regularity of the signal, as was observed previously, the signal is more erratic as the load increases. Figure 5.11d clearly shows how the curve fits to the signal which weaves from side to side of it.

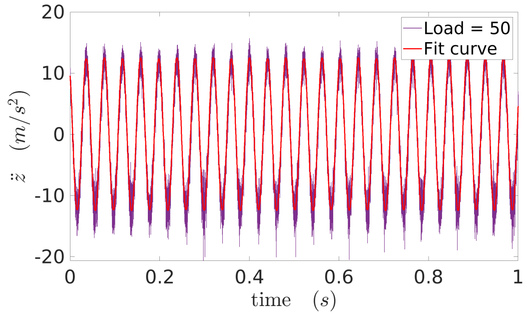
The graphs in Figure 5.9 display the signal in the z-axis direction for the same temporal range as the two previous signals. As was expected its frequency is equal to the signal in the y-axis direction.



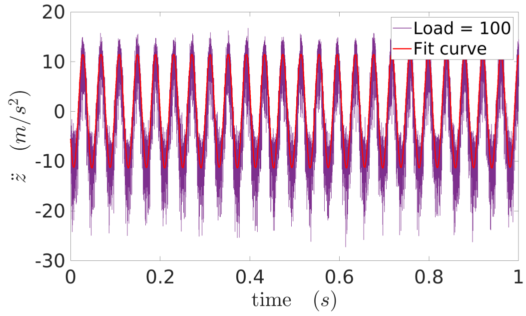
(a) z-axis direction (Unloaded)



(b) z-axis direction (25kg)



(c) z-axis direction (50kg)



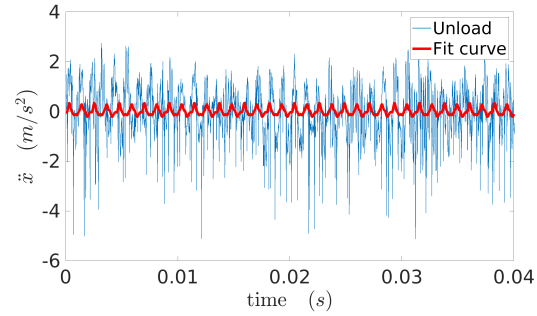
(d) z-axis direction (100kg)

Figure 5.9: Time domain in **z-axis direction** for all loading scenarios (full range)

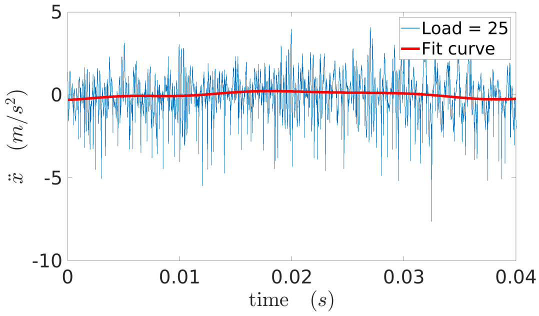
On the contrary, its amplitude is considerably lower. As it can be observed in Figure 5.1 the y-axis direction corresponds to the longitudinal axis of the springs while x- and z-axis conform the plane perpendicular to its axis. The amplitude of the signal is greater in the direction of the axis of the spring than in the transversal directions.

In addition, Figures 5.10a and 5.12a show the signals for the x- and z-axis directions when the system is unloaded. It is appreciated in both graphs that in those directions perpendicular to the springs' axes the vibrational movement by itself produces a more erratic signal. Furthermore, the erratic behaviour aforementioned it is also found, to a greater or lesser degree.

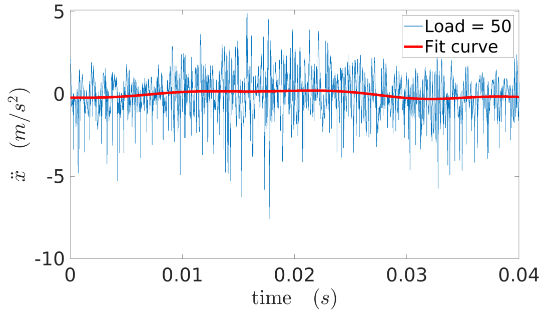
Until this point the signal has been analysed from the time perspective. It has been highlighted how it behaves over time. However, it is also necessary to analyse from the perspective of the frequencies governing the vibratory movement. In this way the most significant frequencies can be identified as this is a necessary step in the characterization of vibrational behaviour of this MF process.



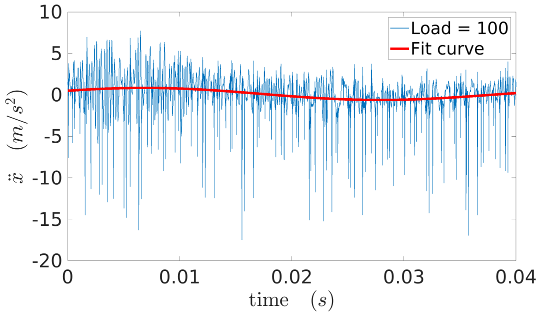
(a) x-axis direction (Unloaded)



(b) x-axis direction (25kg)

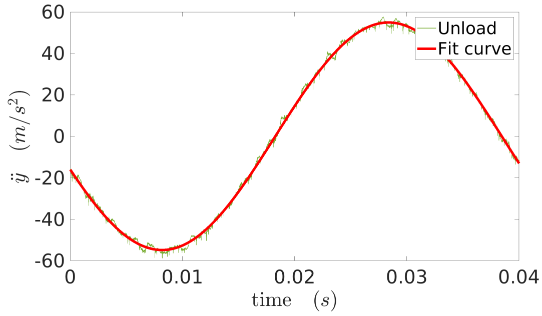


(c) x-axis direction (50kg)

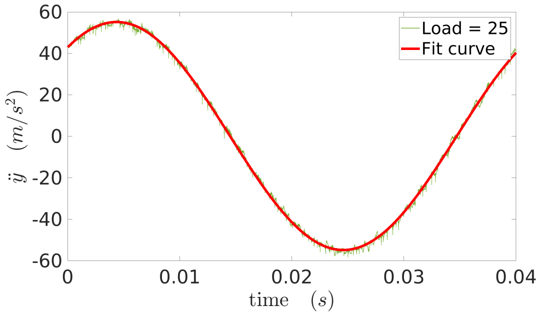


(d) x-axis direction (100kg)

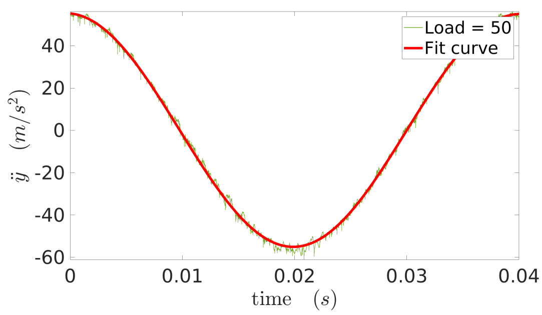
Figure 5.10: Time domain in **x-axis direction** for all loading scenarios (time range = $[0 : 0.04]$)



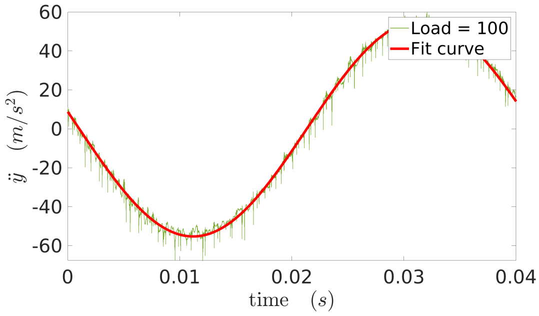
(a) y-axis direction (Unloaded)



(b) y-axis direction (25kg)

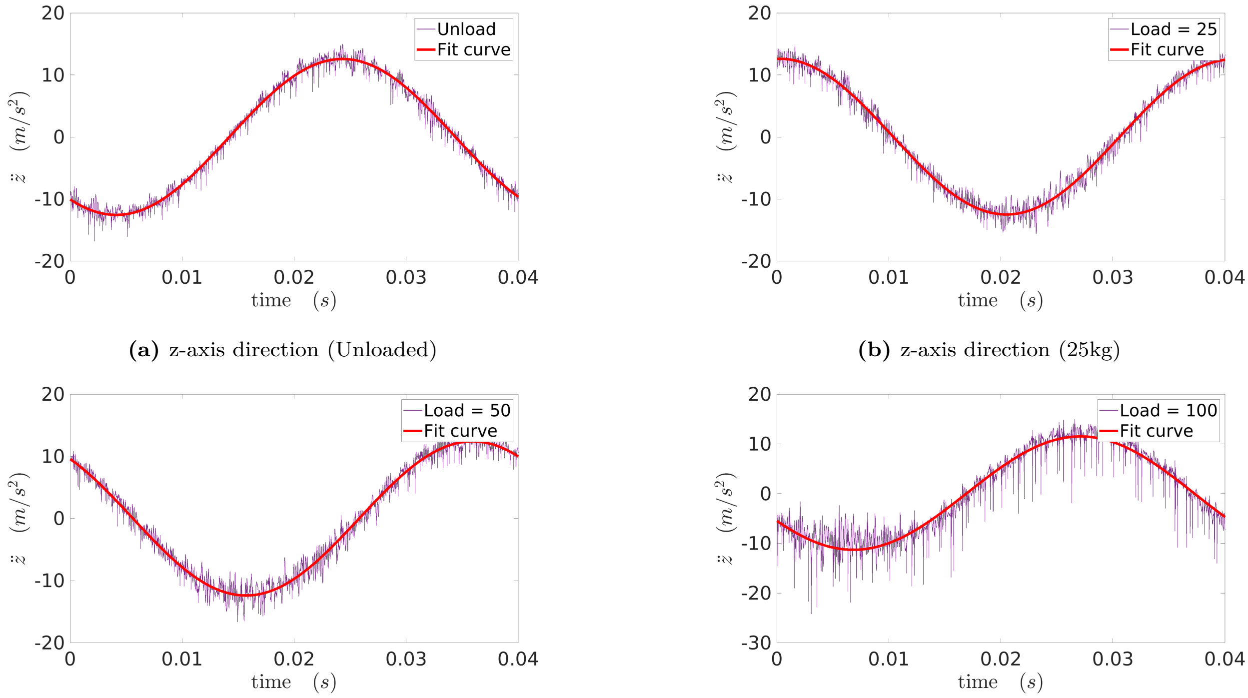


(c) y-axis direction (50kg)



(d) y-axis direction (100kg)

Figure 5.11: Time domain in **y-axis direction** for all loading scenarios (time range = $[0 : 0.04]$)



(a) z-axis direction (Unloaded)

(b) z-axis direction (25kg)

(c) z-axis direction (50kg)

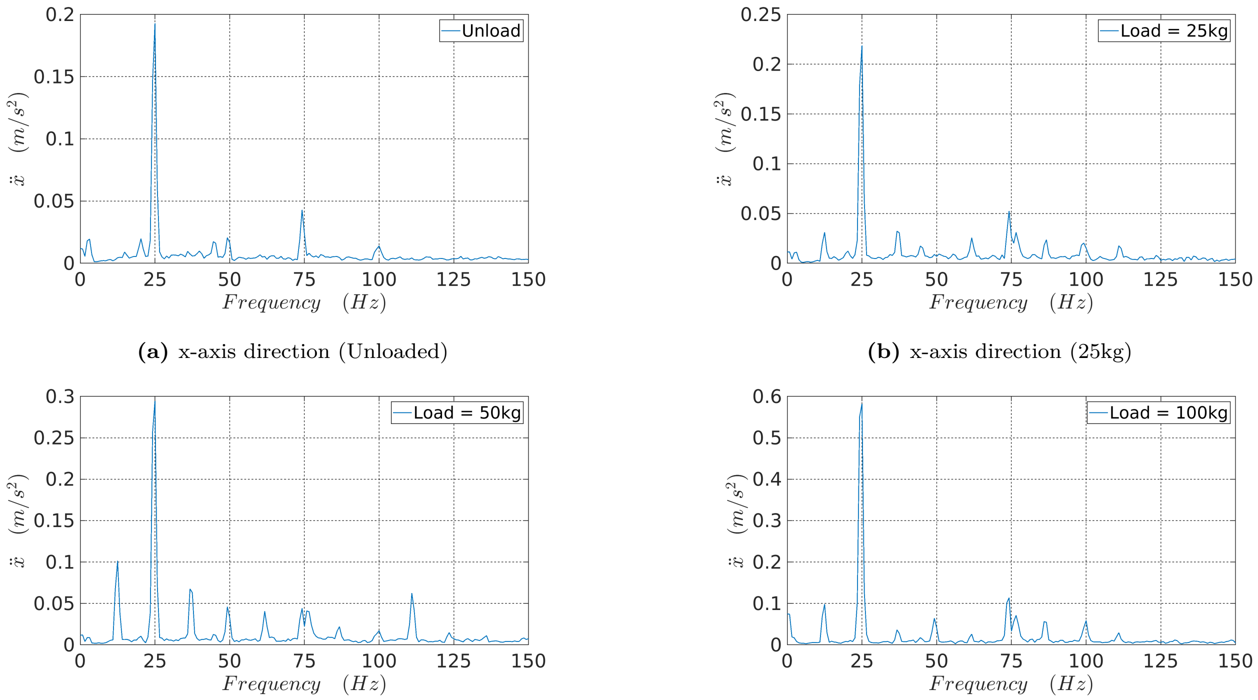
(d) z-axis direction (100kg)

Figure 5.12: Time domain in **z-axis direction** for all loading scenarios (time range = [0 : 0.04])

Frequency domain analysis

The data set used for these measurements also provided the analysis of the signal from the perspective of the Frequency Domain. The graphs in Figures 5.16, 5.17 & 5.18 present the more important frequencies in the three different directions for the different loading scenarios. As can be noticed the prominent frequency is at 25 Hz which coincides with the frequencies of the fit curve used in the y- and z-axis directions when analysing the signal from the perspective of the time domain. This frequency clearly corresponds to the frequency of the excitation source as the motor was always working at 1500 rpm.

The possibility the system had been designed to match its natural frequency and the frequency of the excitation was initially considered. Thus, less energy is required to obtain the same magnitude of displacements. This hypothesis was discarded once the results of the frequency spectra were compared. As can be observed in Figures 5.13, 5.14, & 5.15 the most relevant frequencies, those with the greater magnitudes, do not coincide with the natural frequencies presented in Figure 5.6.



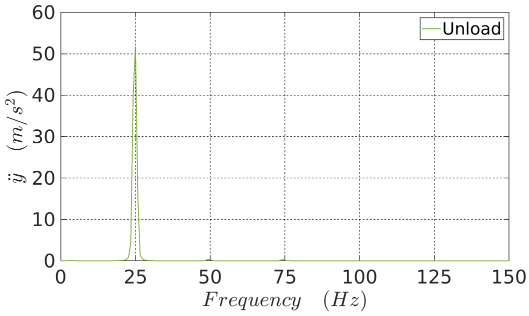
(a) x-axis direction (Unloaded)

(b) x-axis direction (25kg)

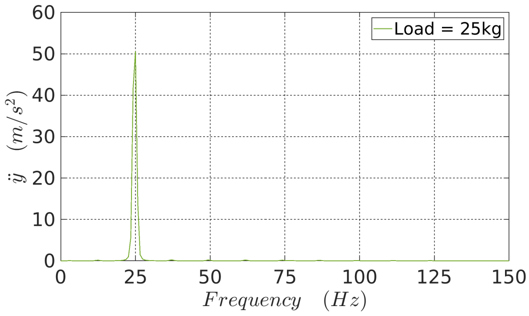
(c) x-axis direction (50kg)

(d) x-axis direction (100kg)

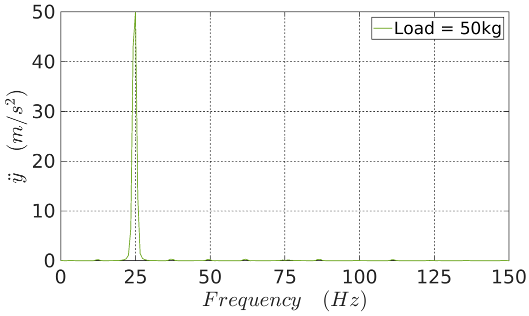
Figure 5.13: Frequency domain in **x-axis direction** for all loading scenarios (acceleration full range)



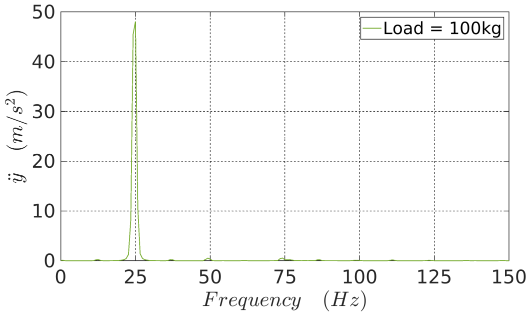
(a) y-axis direction (Unloaded)



(b) y-axis direction (25kg)



(c) y-axis direction (50kg)

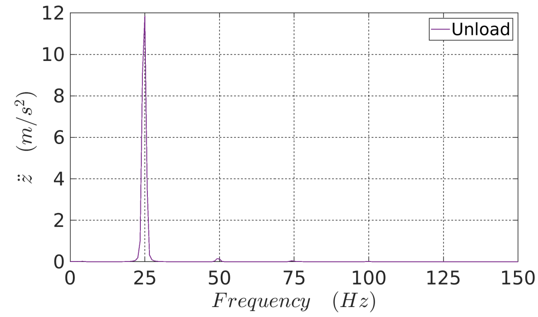


(d) y-axis direction (100kg)

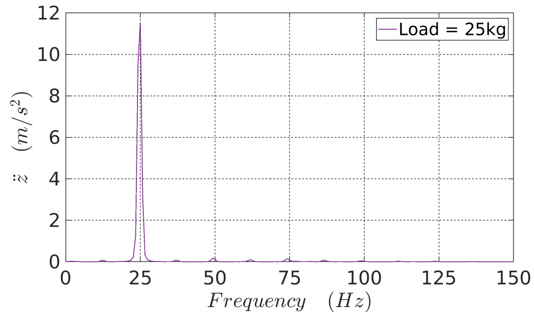
Figure 5.14: Frequency domain in **y-axis direction** for all loading scenarios(acceleration full range)

As per Equation 5.1 the natural frequency diminishes as the mass increases. It can be observed in Figure 5.16, 5.17 & 5.18 that the frequency around 80 Hz barely changes within the different loading steps. The mass of the system is $mass = mass_{trought} + mass_{media}$, for all the loading steps the $mass_{media} << mass_{trough}$. As a conclusion, to notice a change in the natural frequency a much bigger load is necessary.

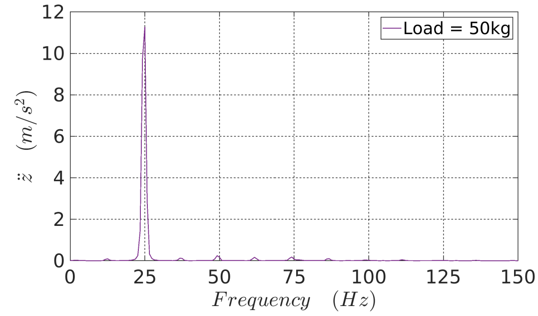
It is observed that the unloaded scenario has a much clearer signal where the fundamental frequency and its harmonics can be easily identified. For example, Figure 5.18a perfectly depicts the fundamental frequency at 25 Hz but also its harmonics at 50 Hz, 75 Hz and 100 Hz. The frequency spectrum becomes more complex when the load is added. Although this phenomenon has been previously observed[67] it remains unexplained. It is obviously related to the load and the way it moves within the work chamber. Further work is required to clarify the significance of theses new frequencies. To this end, the system should be studied in two different scenarios. Firstly, when it is loaded with a monolithic mass. And secondly, when is loaded with a granular mass. These two masses must be equal in magnitude.



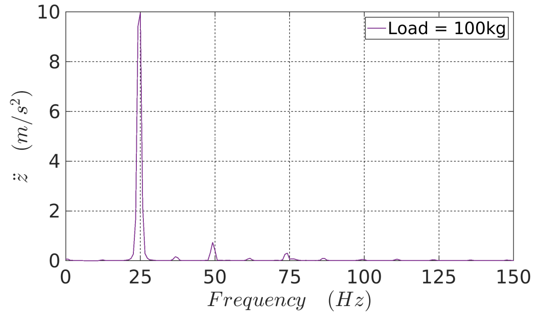
(a) z-axis direction (Unloaded)



(b) z-axis direction (25kg)

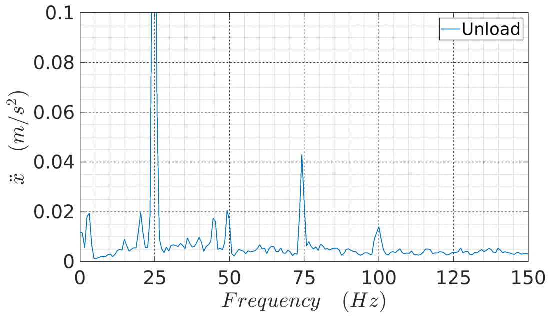


(c) z-axis direction (50kg)

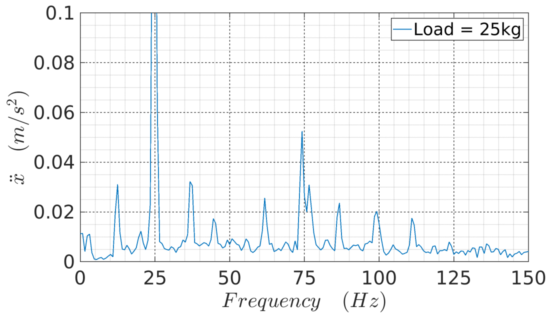


(d) z-axis direction (100kg)

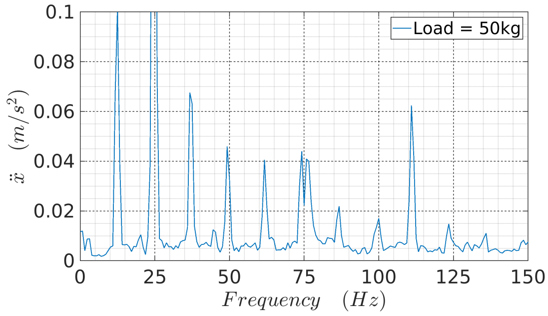
Figure 5.15: Frequency domain in **z-axis direction** for all loading scenarios (acceleration full range)



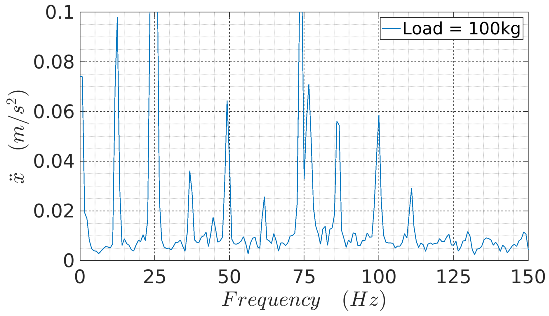
(a) x-axis direction (Unloaded)



(b) x-axis direction (25kg)

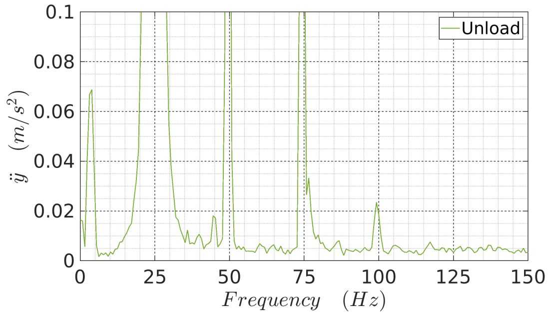


(c) x-axis direction (50kg)

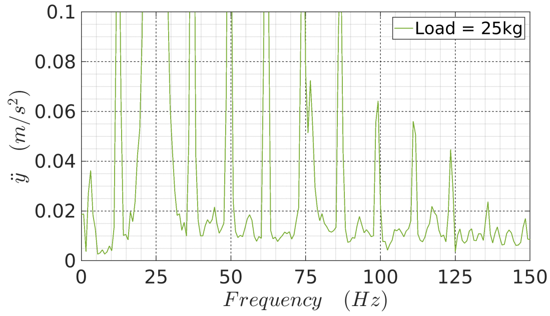


(d) x-axis direction (100kg)

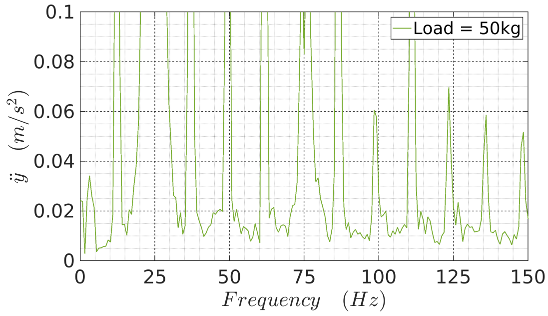
Figure 5.16: Frequency domain in **x-axis direction** for all loading scenarios(acceleration range = $[0 : 0.1]$)



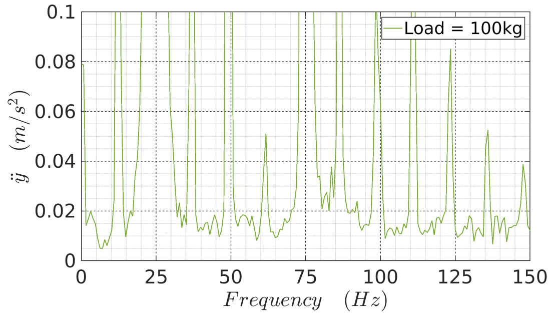
(a) y-axis direction (Unloaded)



(b) y-axis direction (25kg)

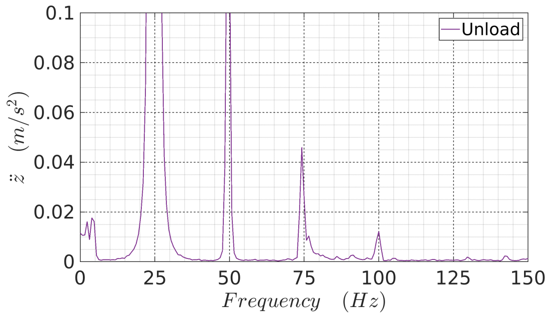


(c) y-axis direction (50kg)

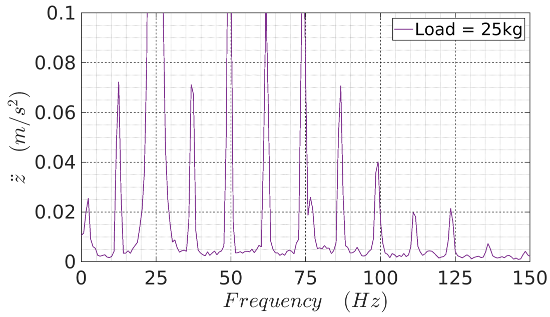


(d) y-axis direction (100kg)

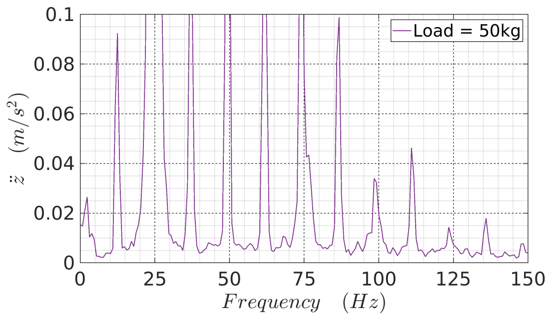
Figure 5.17: Frequency domain in **y-axis direction** for all loading scenarios (acceleration range = [0 : 0.1])



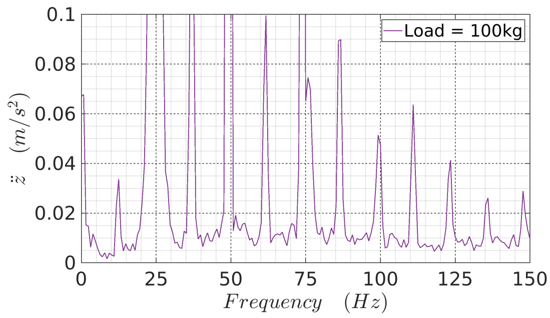
(a) z-axis direction (Unloaded)



(b) z-axis direction (25kg)



(c) z-axis direction (50kg)



(d) z-axis direction (100kg)

Figure 5.18: Frequency domain in **z-axis direction** for all loading scenarios (acceleration range = [0 : 0.1])

5.3 Trough definition

The boundary conditions in this simulation , apart from the excitation source, are the walls of the trough. They maintain the particles within the working chamber. Figure 5.19 presents the CAD model of the whole chamber. To fill this trough to its working condition, approximately, half a ton of media is needed. In simulation terms the mass of half a ton of media is equivalent to a few millions of particles. This is because every real media particle is created in the simulation by clumping 65 single particles. This large quantity increases considerably the computational cost of the simulation. For this reason, only a region of the work chamber has been studied. It consisted in a transversal clip of 100 mm situated at the middle of the trough. A periodic boundary was assigned to both ends, thus the media particles can interact across the boundary, and they can exit at one end and re-enter at the other end when they are travelling in the x-axis direction.

The meshing of the part was not a critical aspect on this simulation as the objective of it always was to gain an insight of the media-flow behaviour and not the interaction between particle and wall.

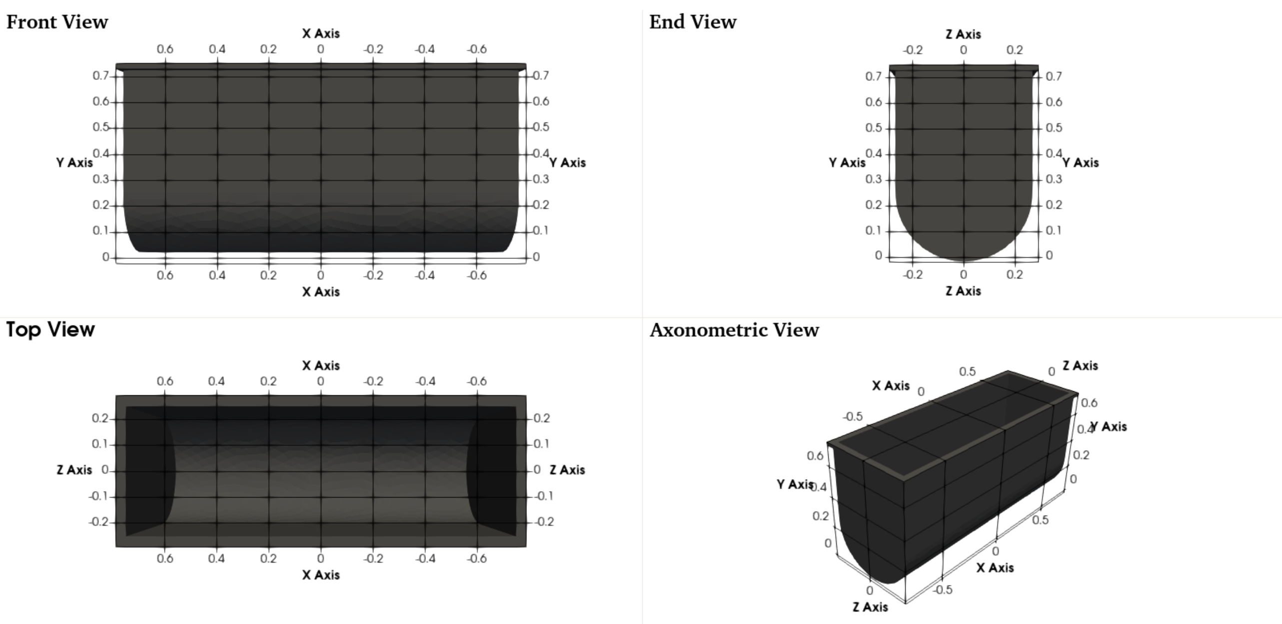


Figure 5.19: VM375Y work chamber

5.4 Vibratory movement definition

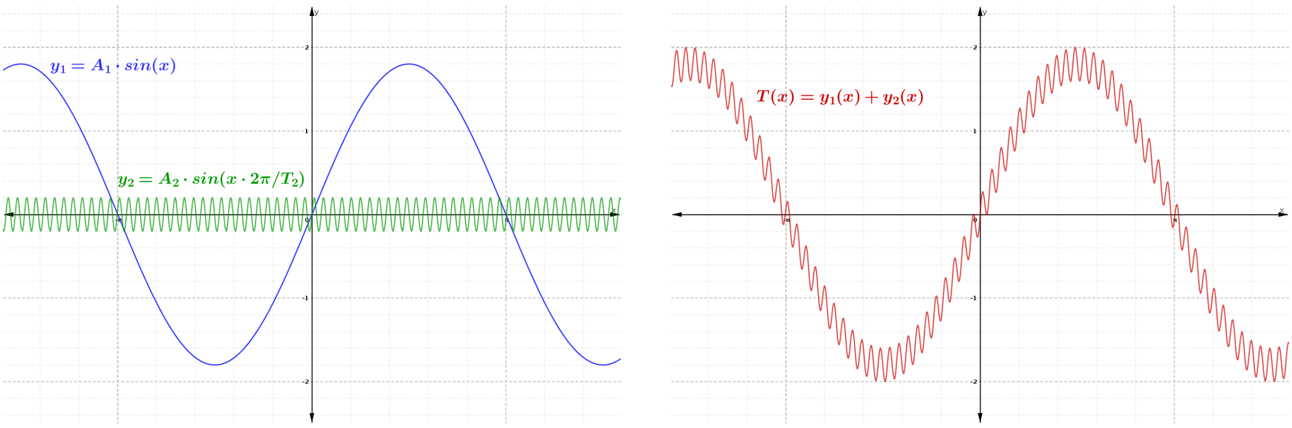
The appropriate definition of the vibration is a decisive parameter in this DEM simulation. LIGGGHTS uses the *wiggle command* to move atoms in an oscillatory way, so that their position $X = (x, y, z)$ as a function of time is given in vector notation as expressed in Equation 5.3.

$$X(t) = X_0 + Asin(\omega \times \Delta) \tag{5.3}$$

Where:

- X_0 represents the position of the particles (x_0, y_0, z_0) at the time the command is specified.
- A represents the amplitude vector (A_x, A_y, A_z) in distance units.
- ω represents the angular velocity ($\omega = \frac{2\pi}{T}$).
- Δ represents the time elapsed since the command was specified.

The representation of this function in a Cartesian coordinate system produces a sinusoidal graph similar to the two functions plotted in Figure 5.20a. This type of functions is far for representing the vibrations depicted in the previous section. However, Figure 5.20b shows a little more complicated sinusoidal graph, this consists in the sum of the two other functions. Moreover, the sum of two sinusoidal functions produces a more complex sinusoidal function so the greater the sum of sinusoidal functions is, the more complicated the resultant function will be.



(a) $y_1(x)$ and $y_2(x)$ **(b)** $T(x) = y_1(x) + y_2(x)$

Figure 5.20: Sinusoidal functions

The appropriate quantity and quality of functions produces a resultant function identical to any of the studied signals. Therefore, the complexity resides in finding the individual signals hidden in the intricacy of the resultant signal. This is exactly what the Fourier transform does. It decomposes a function of time as those displayed in Figure 5.20a into its constituent frequencies. The latter is known as the Frequency Domain, depicted in Figure 5.20b. This provides the amplitude of the signal (A) and its frequency (f) as known $T = f^{-1}$. So, from the expression of the angular velocity:

$$\omega = \frac{2\pi}{T} \Rightarrow \omega = 2\pi f \tag{5.4}$$

Thus, A and ω are known for each of the functions comprehended in the resultant signal. These parameters are enough to fulfil the requirement of Equation 5.3.

The consistency of units in physic is critical to obtain the right results. LIGGGHTS inputs the amplitude as a displacement, this is distance unit (m). Contrarily, the **Data Set Unit** provides the amplitude as the acceleration (m/s^2). Thus the acceleration must be transformed into a distance unit so the software can input the suitable value. Equation 5.5 establishes the relation between the displacement (U_x) and the acceleration ($\ddot{x}$). The frequency (f) is in the denominator, the greater this one is the smaller the displacement will be. For this reason, when inputting the vibration in the software, the frequencies greater than 150 Hz were neglected.

$$U_x = \frac{\ddot{x}}{2\pi f} \tag{5.5}$$

The approach followed to input the vibratory excitation in the script consisted of: Firstly, it has been presented that the vibration's signal is a complex signal that can be decomposed in a number of simpler signals. Secondly, the vibratory motion of the system was individually studied for each of the three-axis directions. Summarising, a total of three different vibrational signals were obtained, one for each of the defined axes. At the same time these signals were simplified into the sum of a number of simpler signals. A *wiggle command* was needed for each of these simpler signals. The script displayed in Figure 5.21 comprises the commands used to defined the vibratory movement so the software could interpret it.

To define the signal in the x-axis direction nine commands were needed, therefore the original signal was decomposed into nine simpler signals. The y- and z-axis directions were also simplified, however, due to the complexity of these two signals fewer commands were needed. The whole script can be found in the Appendix B.4.

```
#------------------------- Trough starts vibrating --------------------------#
# Definition of the vibration in x-axis direction
fix harmonicX1  all move/mesh mesh wall wiggle amplitude  1.589262e-05    0     0     period 8.000000e-02
fix harmonicX2  all move/mesh mesh wall wiggle amplitude  2.358719e-05    0     0     period 4.000000e-02
fix harmonicX3  all move/mesh mesh wall wiggle amplitude  6.802588e-07    0     0     period 2.723404e-02
fix harmonicX4  all move/mesh mesh wall wiggle amplitude  6.736165e-07    0     0     period 2.031746e-02
fix harmonicX5  all move/mesh mesh wall wiggle amplitude  5.226244e-07    0     0     period 1.347368e-02
fix harmonicX6  all move/mesh mesh wall wiggle amplitude  3.070838e-07    0     0     period 1.306122e-02
fix harmonicX7  all move/mesh mesh wall wiggle amplitude  1.923558e-07    0     0     period 1.163636e-02
fix harmonicX8  all move/mesh mesh wall wiggle amplitude  1.486267e-07    0     0     period 1.000000e-02
fix harmonicX9  all move/mesh mesh wall wiggle amplitude  6.028109e-08    0     0     period 9.014085e-03

# Definition of the vibration in y-axis direction
fix harmonicY1  all move/mesh mesh wall wiggle amplitude  0    1.949234e-03    0     period 4.000000e-02

# Definition of the vibration in z-axis direction
fix harmonicZ1  all move/mesh mesh wall wiggle amplitude  0    0    4.038371e-04    period 4.000000e-02
fix harmonicZ2  all move/mesh mesh wall wiggle amplitude  0    0    7.744550e-06    period 2.031746e-02
```

Figure 5.21: Definition of the vibratory movement in LIGGGHTS' script

All the simulations presented in this document employed the signal obtained for the 100kg loading scenario to describe the vibratory movement energising the particles.

The greatest accelerations were found for the vibratory movement in the y- and z-axis directions. Thus, as it was to be expected, the displacement in the x-axis direction is very little in comparison with the displacements in the y- and z-axis direction.

5.5 DEM model

The contact model employed was the Hertz model without cohesion. This model takes into consideration the normal and tangential forces when two particles are in contact. At the same time, the normal force has two terms: a spring force and a damping force. Furthermore, the tangential force also has two terms: a shear force and a damping force. All of them were taken into consideration in

Table 5.1: Parameters' summary

Parameter		Value
Vibratory finishing machine	Type	VM375Y
	Rotation speed	1,500 rpm
	Two eccentric masses	--
Media	Weight of loaded media	22.8 kg
	Average height	24 mm
	Average base diameter	20 mm
	Media particle average weight	0.0047 kg
	Shape	Conical media
	Material	Plastic
Mechanical properties of the material	Media density	1665.71 kg/m^3
	Media Poisson's ratio	0.265
	Media Young modulus	4.60×10^8 Pa
	Lining Poisson's ratio	0.25
	Lining Young modulus	2.50×10^7 Pa
	Media-media static friction coefficient	0.58
	Media-lining static friction coefficient	0.86
	Media-media restitution coefficient	0.67
	Media-lining restitution coefficient	0.159

the simulations discussed in this document.

According to published results, the average contact force does not show any consistent changes with the degree of lubrication (dry or water-wet)[4]. For that reason, the process will be considered dry in terms of simulation.

Table 5.1 summarizes the essential data of the MF machine as well as the media used in the simulation.

5.6 Case study

This case study emerged as an attempt to study the particles' behaviour in the trough. The vibratory signal employed in this simulation was the one corresponding to the loading scenario of 100 kg. The mechanical properties of all the materials involved in this process were experimentally defined as reported in Chapter 4.

The Liverpool John Moores University provided a local HPC to run the simulations. It has a 24h running time limitation, after this time, the job automatically cancelled itself so other users can also run their jobs. For this reason, several scripts were written to carry out this simulation. Thus, a multi-script simulation was needed. Every script corresponds to a simulation step. A total of fifteen steps were required for the whole case study. Table 5.2 displays these steps.

Table 5.2: Simulation detail

Step	Job ID	CPU	Time simulated per step (s)	Computing time (h)	KinEng (J)	Mass (kg)	Media particles
1	3443	280	1	13	5.0643234	19.055114	4065
2	3445	280	0.5	24	0.0069762791	22.892223	4878
3	3447	280	0.5	18	0.33748957	22.892223	4878
4	3448	280	0.5	17	0.5507044	22.892223	4878
5	3449	280	0.5	18	0.20708356	22.892223	4878
6	3450	280	0.5	16	0.54368689	22.892223	4878
7	3451	280	0.5	18	0.12851976	22.892223	4878
8	3452	280	0.5	17	0.4773752	22.868759	4873
9	3453	280	0.5	16	0.28966449	22.859373	4871
10	3454	280	0.5	18	0.54493101	22.859373	4871
11	3455	280	0.5	18	0.16632626	22.85468	4870
12	3456	280	0.5	18	0.55831688	22.849987	4869
13	3457	280	0.5	15	0.39305107	22.849987	4869
14	3458	280	0.5	15	0.54181779	22.849987	4869
15	3459	280	0.5	16	3.40E-05	22.845294	4868

The first two steps were used firstly to fill the simulation domain and secondly to let the media settle within it. From the third to the fourteenth step the trough was continuously energized. The last step was used to remove the excitation source and let the particles rest.

Figure 5.22 shows the distribution of the media within the studied region. A total of 4,878 particles were needed for this simulation.

Figure 5.23 shows the distribution of velocity in the x-axis direction at a moment in time when the system is working in its steady regimen. This capture suggests the media particles within the bulk do not travel along this direction as the majority of the media is in yellow which represents velocity null in this axis direction. The free particles at the top seem to move along this direction without a clear tendency.

Figure 5.24 shows the distribution of velocity in the y-axis direction at the same moment in time as the previous snapshot. It is noted here that the lower part of the bulk is pushing upwards while the upper part is pushing downwards. A mixture of velocities is observed in the free particles of the top of the bulk.

Figure 5.25 shows the distribution of velocity in the z-axis direction at the same instant. The red particles are moving towards the left while the blue particles are going in the opposite direction. There can be recognized on the free particles at the surface of the top a clear tendency of dividing the bulk into two flow currents rotating in opposite directions. However, in the region below the surface no clear tendency is observed.

Figure 5.26 shows the distribution of velocities in the x-axis direction over a period of time approximately equal to $4 \times T$. Each row on the figure divides the time of a period(T) in four snapshots evenly distributed. So, every column captures the same instant in different periods. On this occasion the particles are represented by the vertical component of the velocity vector(V_y) at its centre of mass. The size of the vectors depends on the magnitude of the velocity at that instant. Thus, the greater its magnitude, the bigger the vector is printed out. The red vector represents those particles travelling in the positive direction of the x-axis while the blue vectors are the particles travelling in the negative direction. Yellow vectors are those that do not move in this direction. As can be observed, the particles tend to not move in this direction. Yellow vectors are dominant in magnitude and quantity over the others for all sixteen snapshots.

Figure 5.27 uses the same layout to present the distribution of velocities in the y-axis direction for the same time-steps. Fundamentally, the red-colour vectors represent those particles ascending within the slice while the blue-colour vectors display the descending particle.

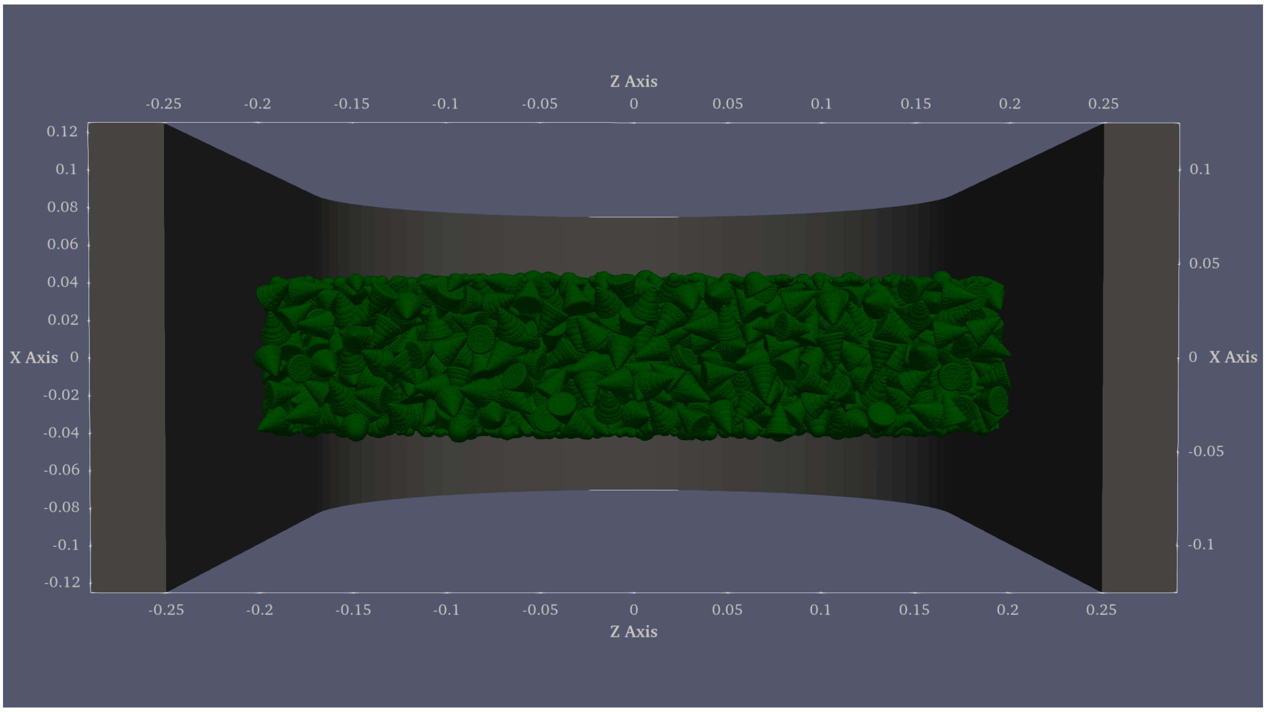
A pattern can be identified along the different snapshots. Firstly, the vibratory movement pushes the particles towards the free surface of the bulk ($dt = 0.01\,\mathrm{s}$), this lasts approximately for $0.02\,\mathrm{s}$. It coincides with the positive side of the acceleration in the signal presented in Figure 5.11d. Then, the particles start travelling downwards as the acceleration becomes negative.

Figure 5.28 depicts the distribution of velocities in the z-axis direction. The reddish vectors represent those particles moving right. On the contrary, blueish vectors represents the particles travelling to the left. Yellow represent no movement in the z-axis direction. Differently to what it was observed for the velocity in the x-axis, the yellow colour is present but far for being dominant. Particles travel backward and forward within the slice. No pattern or trend is apparent in this direction.

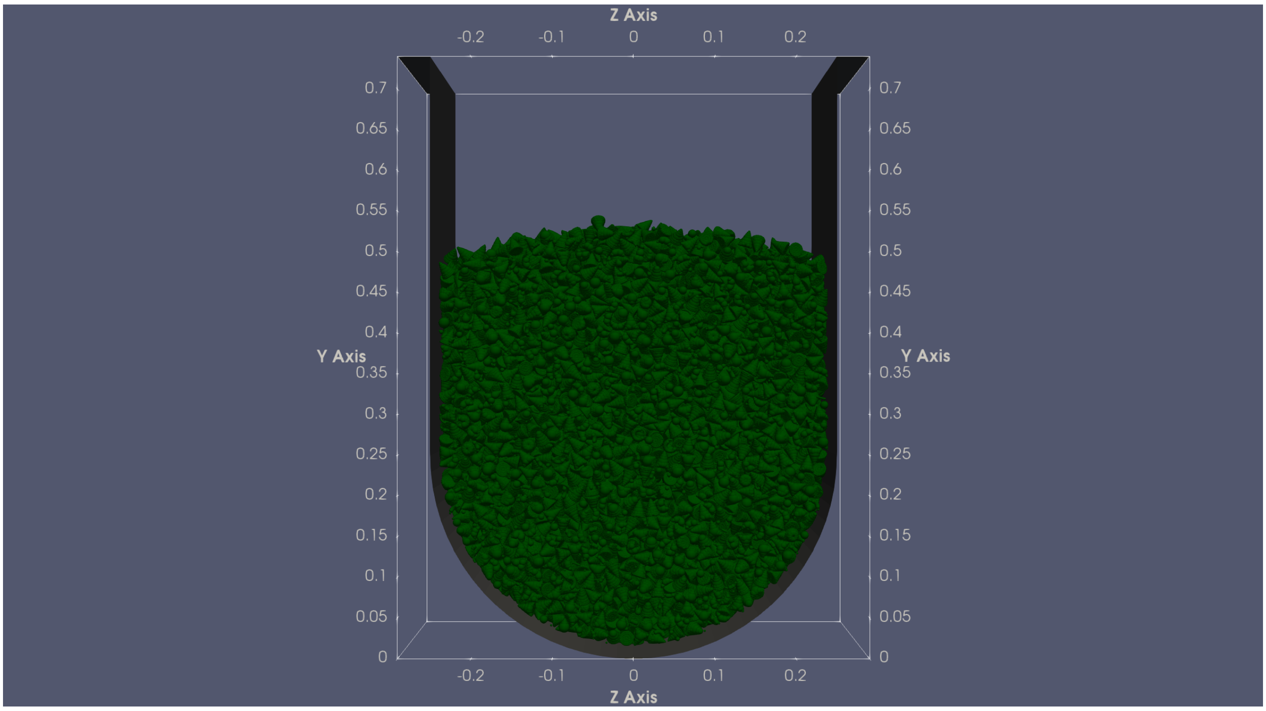
The flow behaviour defined by the simulation does not correspond to what is observed in the reality. Figure 5.29 shows the real distribution of media within the bulk at an instant when the system is working in its steady state. The white arrows indicate the flow trends observed in the Laboratory. On the other hand, the slope created at the free surface of the bulk it is not observed in any of the simulation results.

Nonetheless, when in the laboratory, the system was overloaded, the behaviour of the media was erratic and the form adopted by the free surface at the top of the bulk was similar to what is observed in the Figure 5.23b. Instead of a single well-defined slope, there were two, a positive and a negative slope with similar steepness. For this reason, a different simulation was considered. In this new case, everything remained the same but the quantity of particles, which was reduced by 75%. Figure 5.30 presents the media distribution for this simulation. Firstly, the distribution of the particles is very close to the real one. Secondly, although a single slope was also observed, its steepness was considerably lower than the one observed in the real phenomenon. Thirdly, the media flow behaviour obtained from this simulation did not change considerably from the full load simulation, the same erratic behaviour was identified and therefore no clear tendencies were found.

The vibration measurement was repeated twice and similar results were obtained in both cases. In principle, the reading are fine. Although a more extensive analysis is required to dispel doubts. It is also possible that imputing complex vibratory movement to the DEM software is not yet an option and although the approach followed to describe the vibration is valid from the mathematical point of view, it is not necessarily valid from the software's perspective.

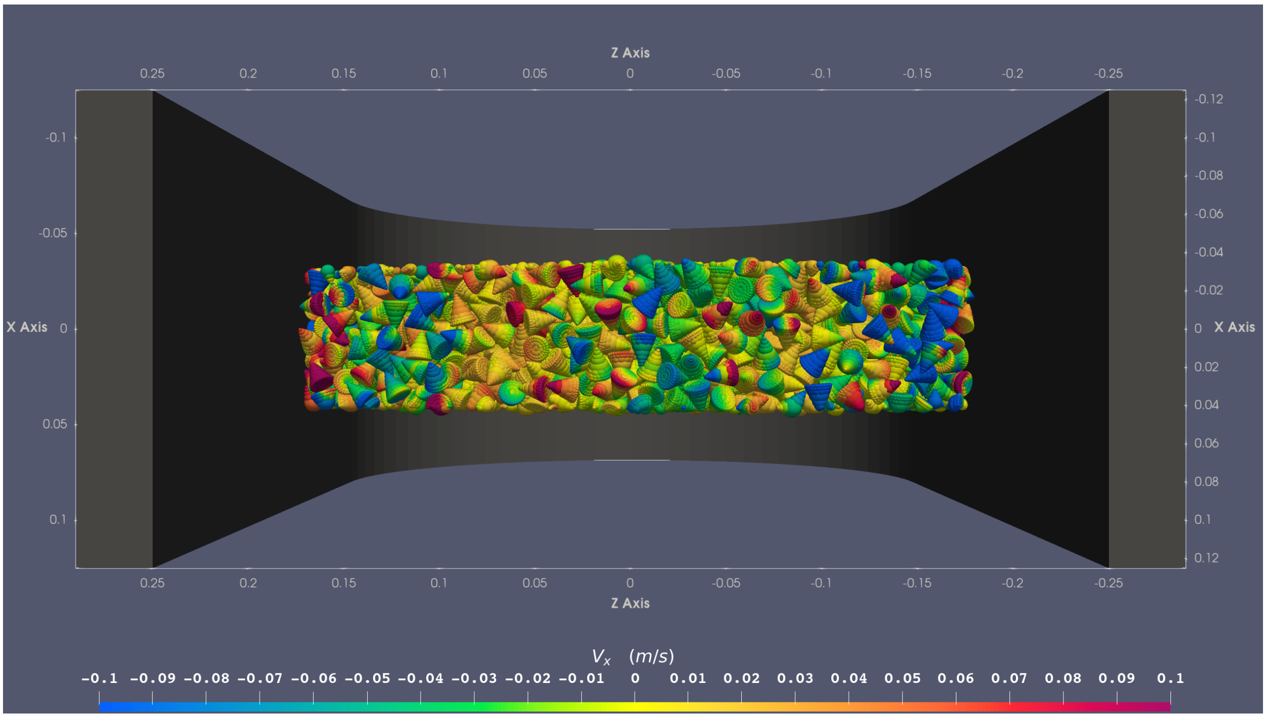


(a) Settled particles (Top View)

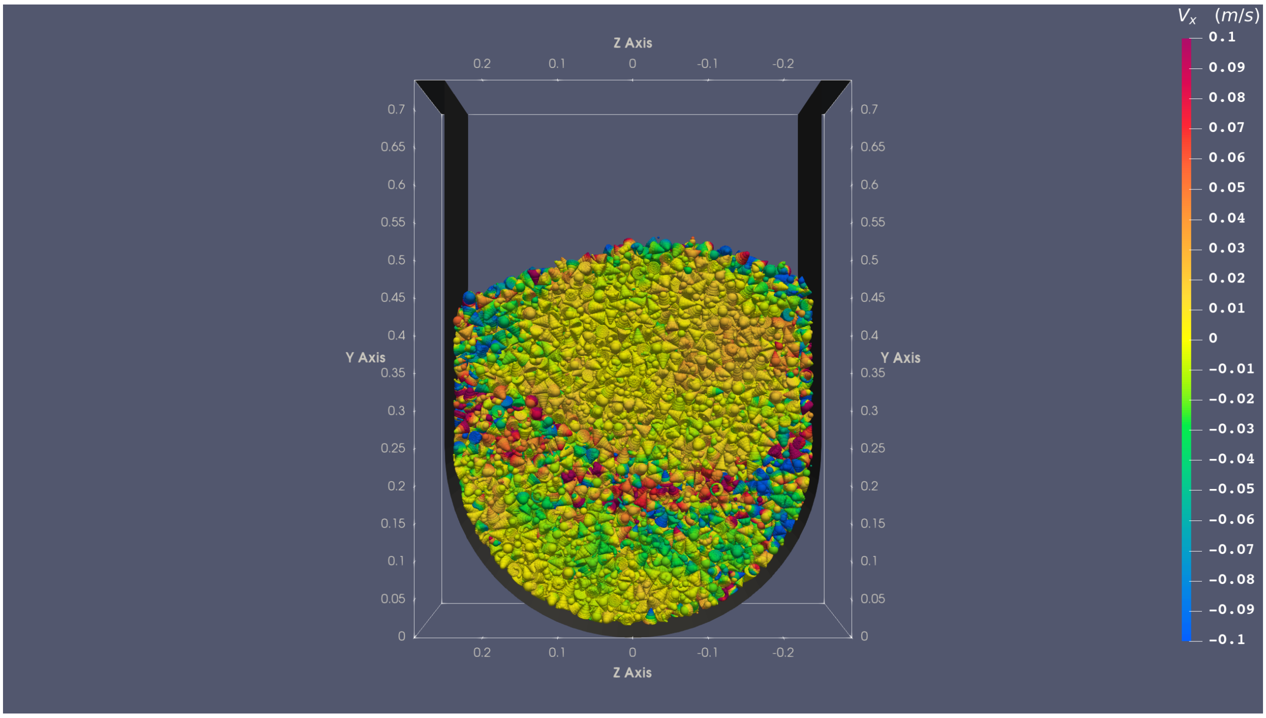


(b) Settled particles (Front View)

Figure 5.22: Media particles distribution before energising the trough

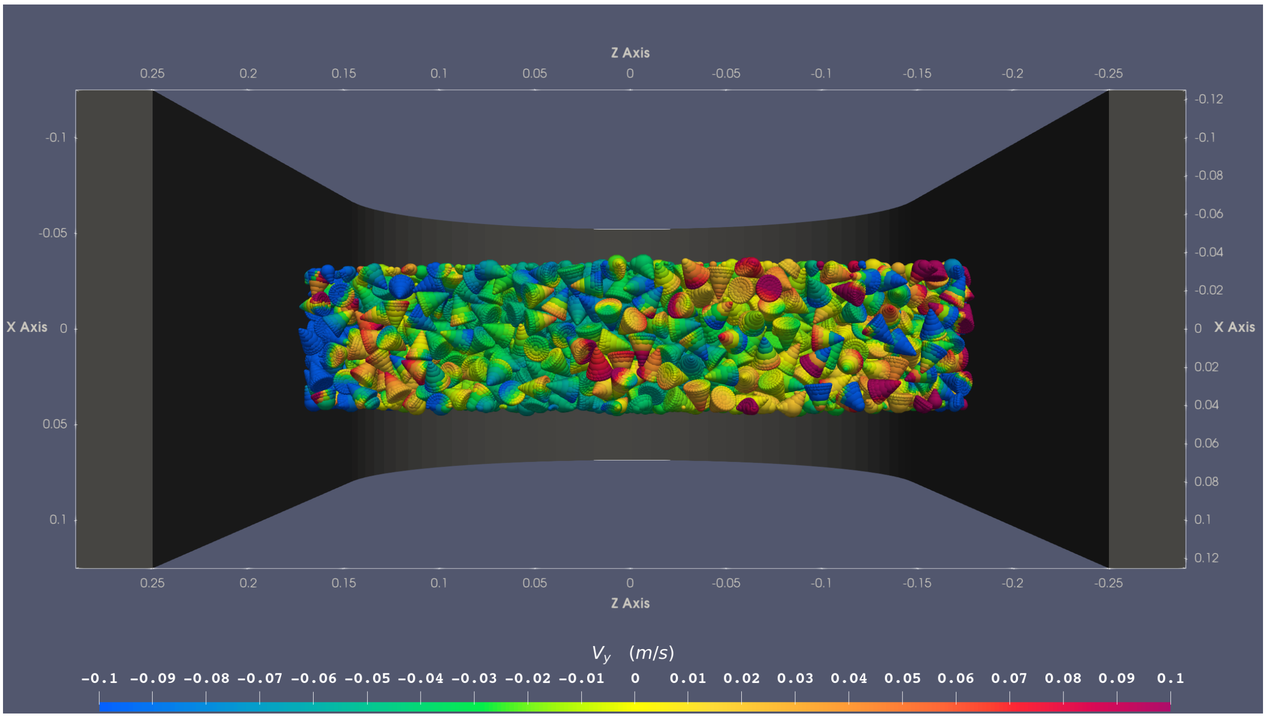


(a) Velocity distribution in the x-axis direction (Top)

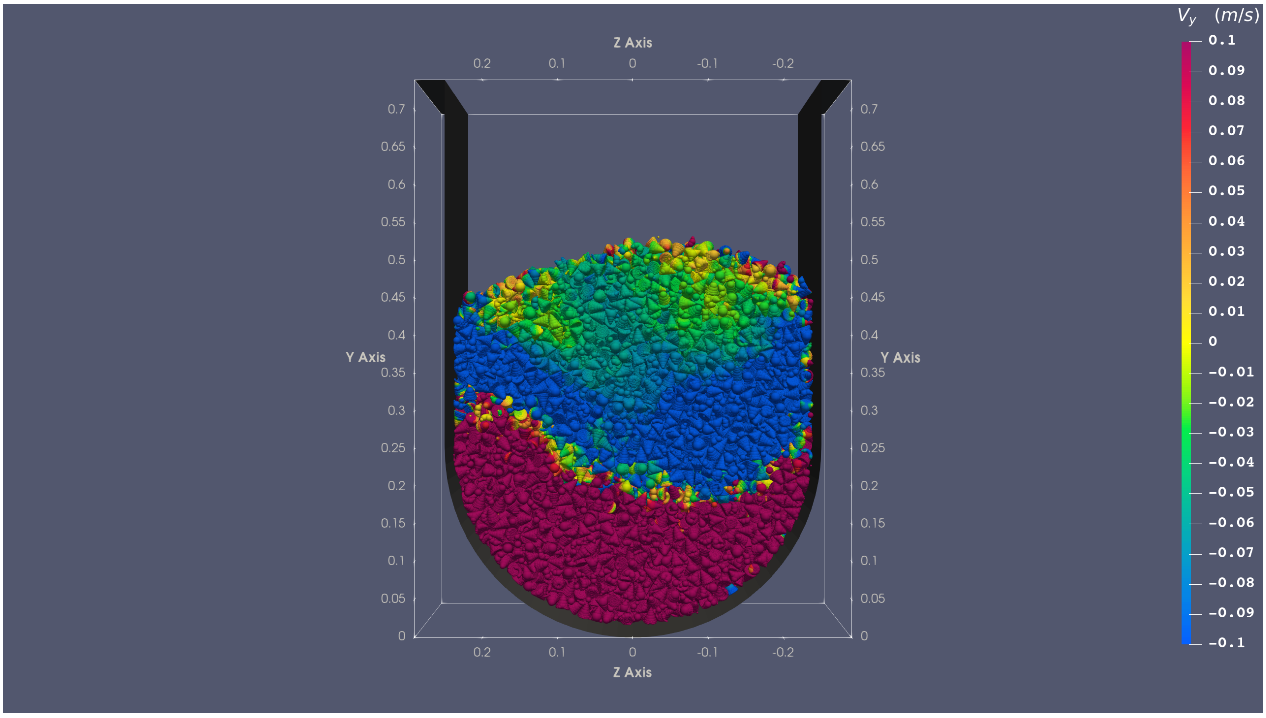


(b) Velocity distribution in the x-axis direction (Front)

Figure 5.23: Velocity distribution (x-axis direction)

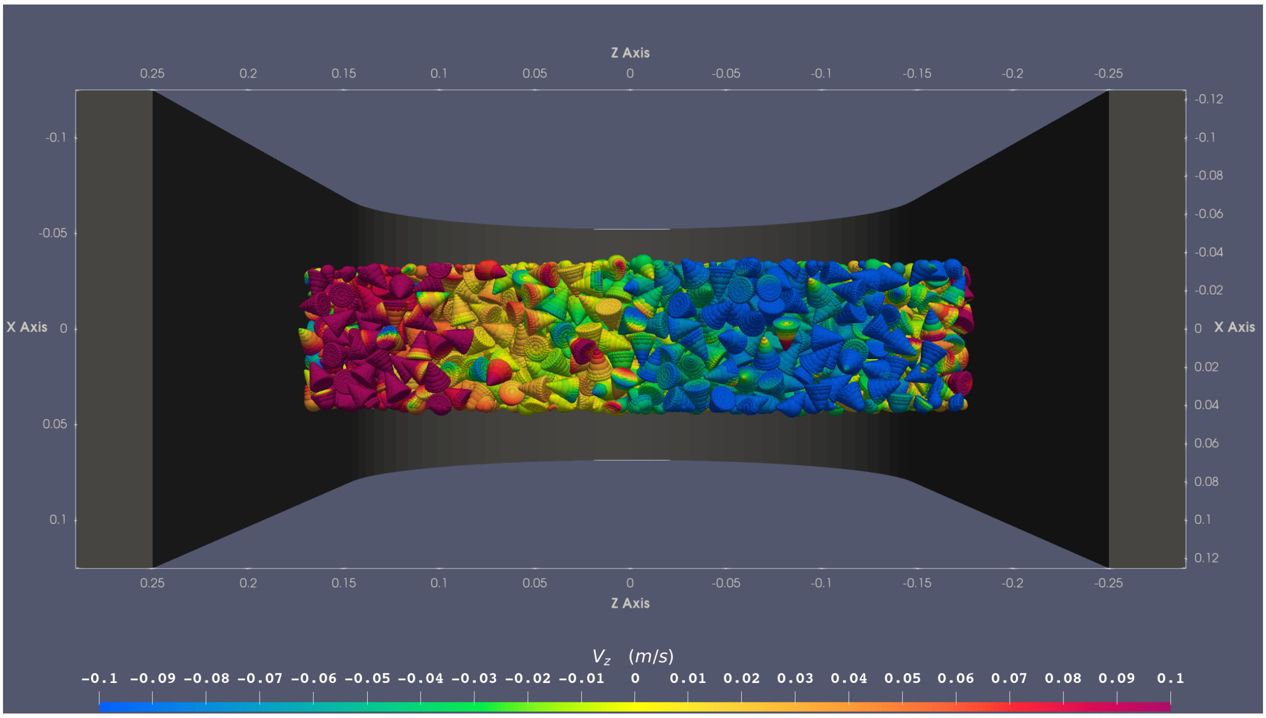


(a) Velocity distribution in the y-axis direction (Top)

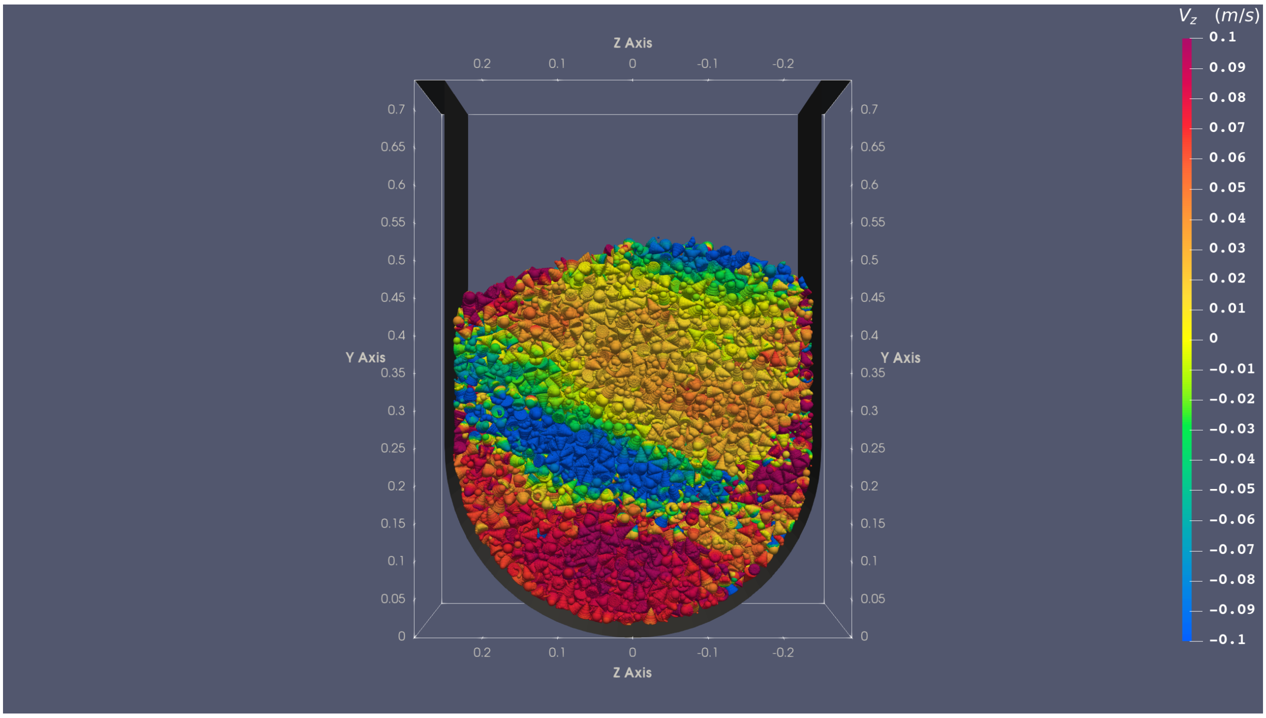


(b) Velocity distribution in the y-axis direction (Front)

Figure 5.24: Velocity distribution (y-axis direction)



(a) Velocity distribution in the Z-axis direction (Top)



(b) Velocity distribution in the y-axis direction (Front)

Figure 5.25: Velocity distribution (z-axis direction)

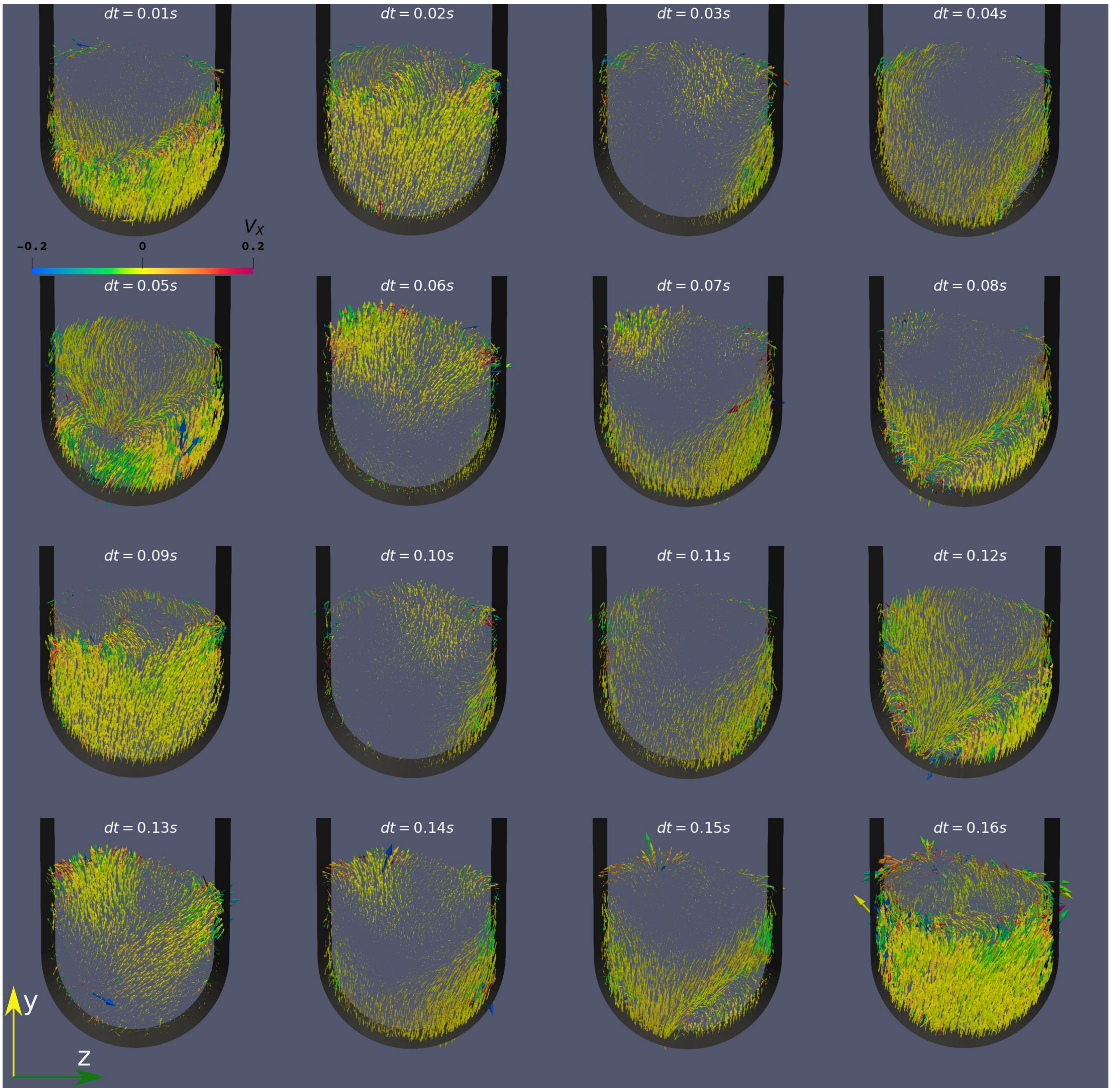


Figure 5.26: Evolution of the V_x over time in a period of 0.16s

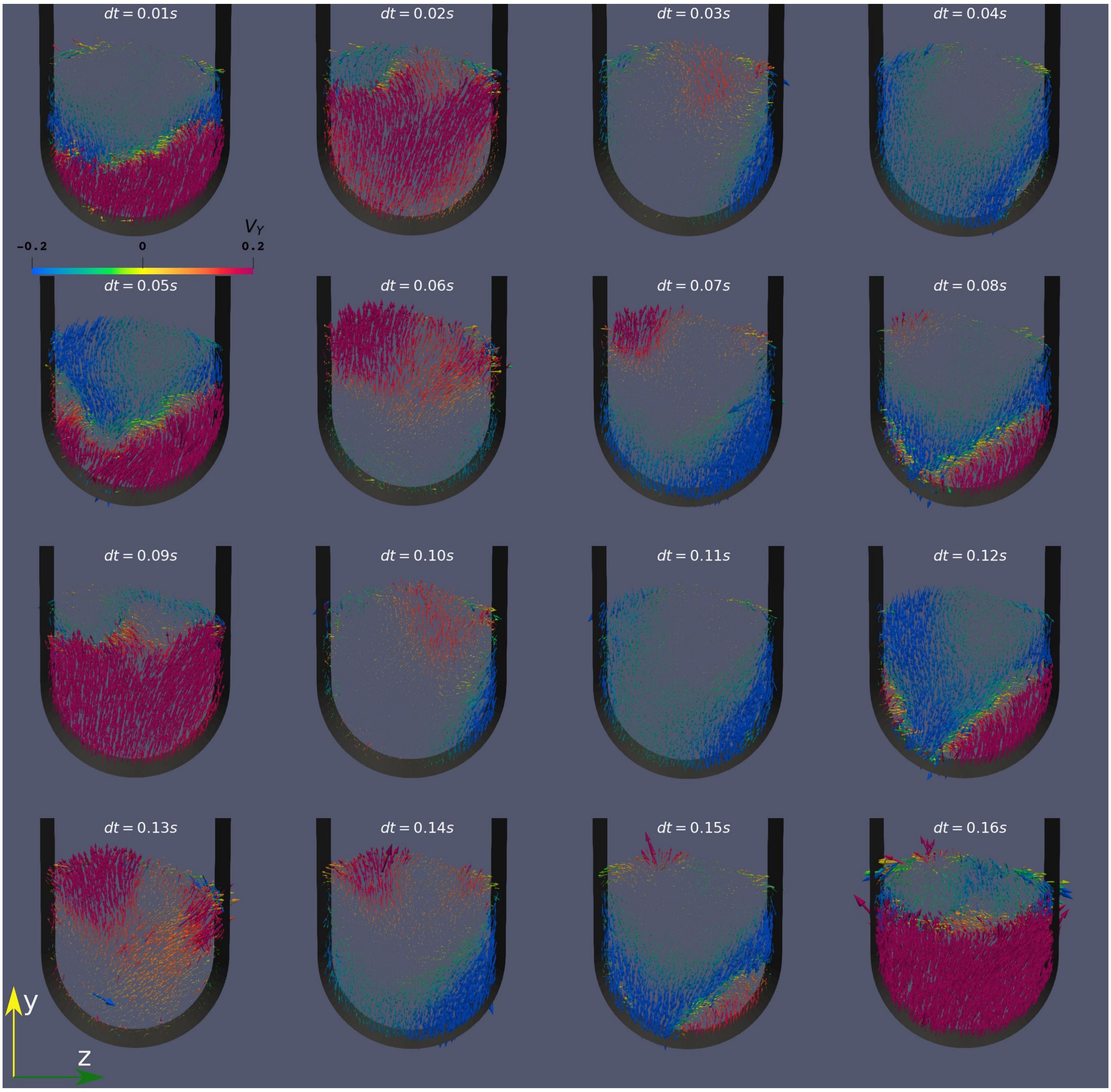


Figure 5.27: Evolution of the V_y over time in a period of 0.16s

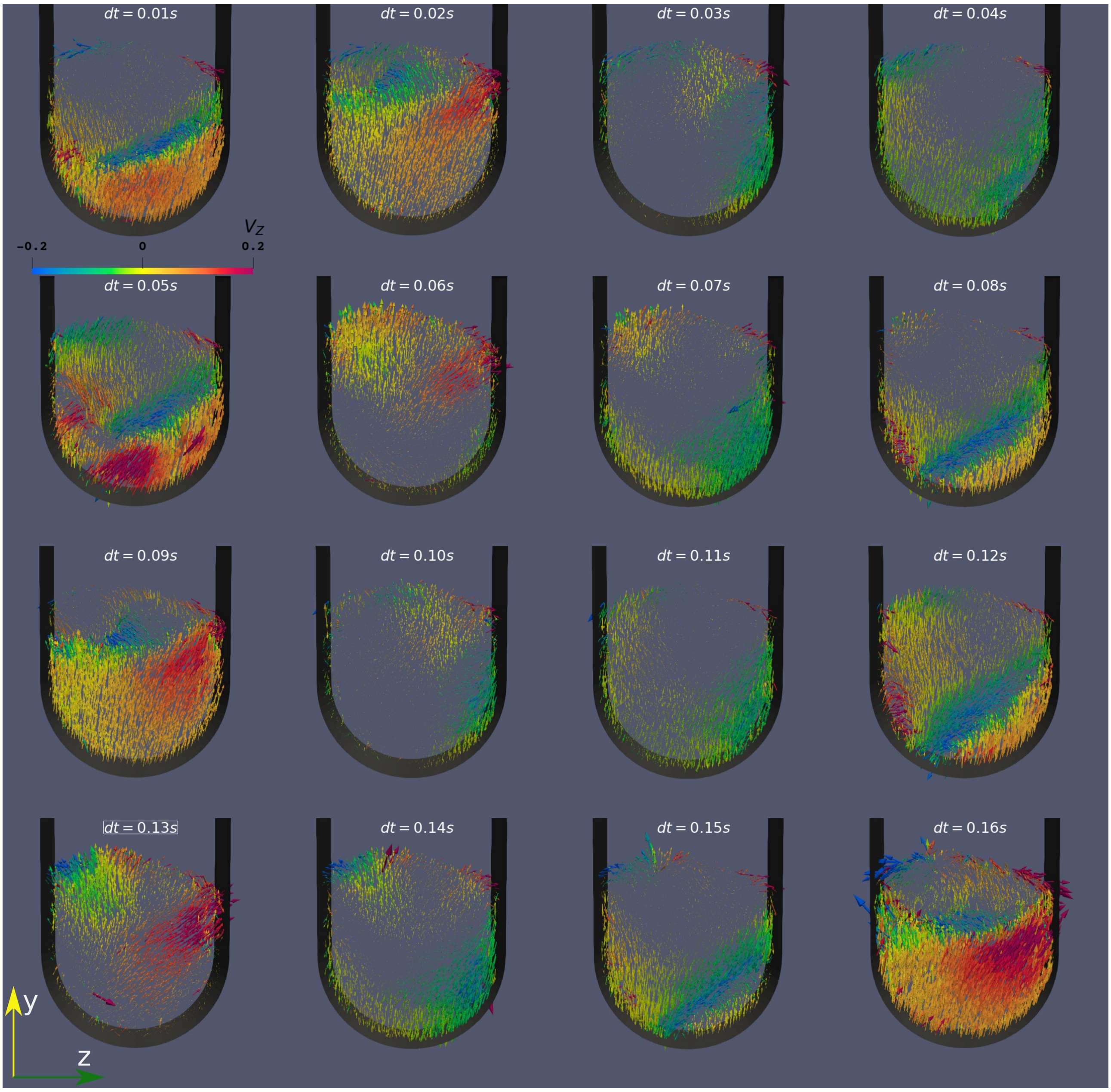


Figure 5.28: Evolution of the V_z over time in a period of 0.16 s

Figure 5.29: Real media distribution within the work chamber (side view)

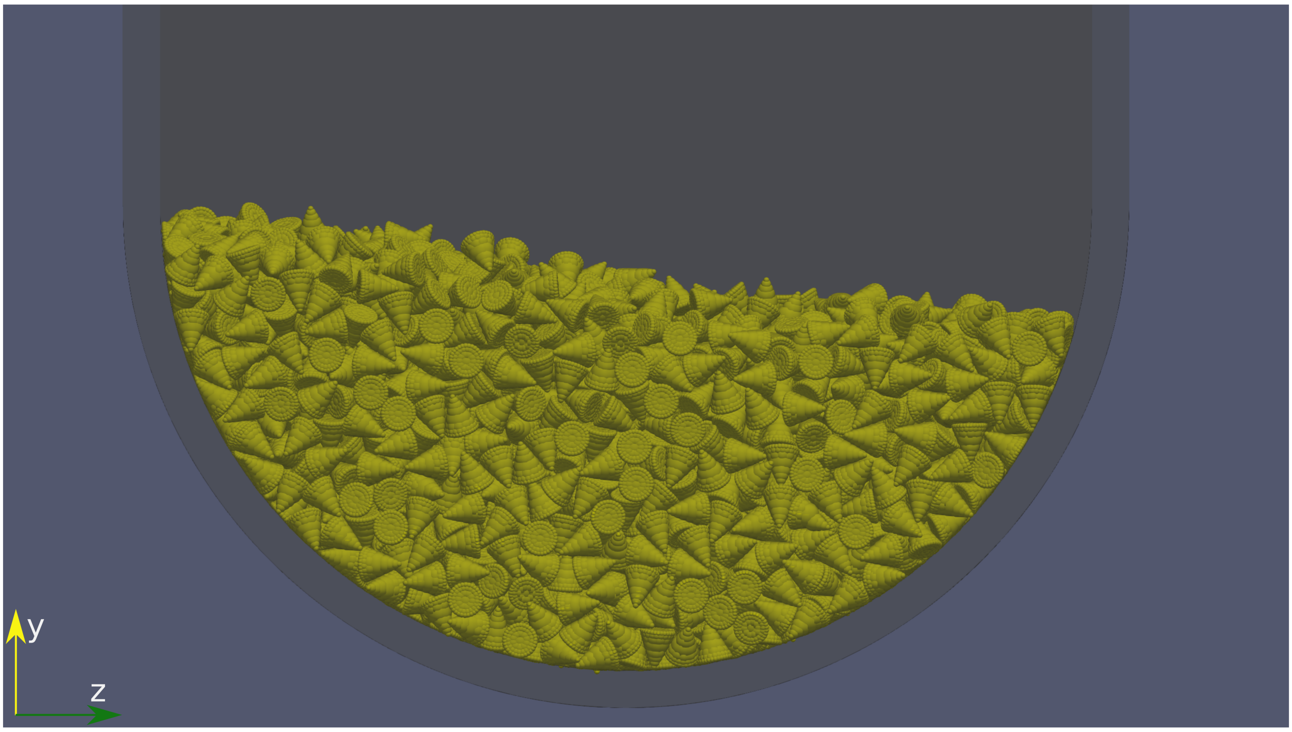


Figure 5.30: Virtual media distribution within the work chamber (side view)

Chapter 6

Rotatory mass finishing studies

This chapter investigates the flow behaviour of the media within the work chamber of the rotatory disc MF machine. This was designed by Otec within the series EF. Figure 6.1 shows the EF-18 model on which are based all the simulations referred in this chapter.

Figure 6.1: Rotatory disc finishing machine (EF 18)

This finishing machine is characterized by possessing a circular work chamber, bowl, supported by a metal frame. At the same time the work chamber is divided into two different parts: the rotatory disc and the surrounding wall. The latter correspond to the physical barrier that keeps the media and workpieces within the work chamber and which features contribute to characterize the media flow. The rotatory disc conforms the lower part of the work chamber, it consists on a disc fitted at the lower

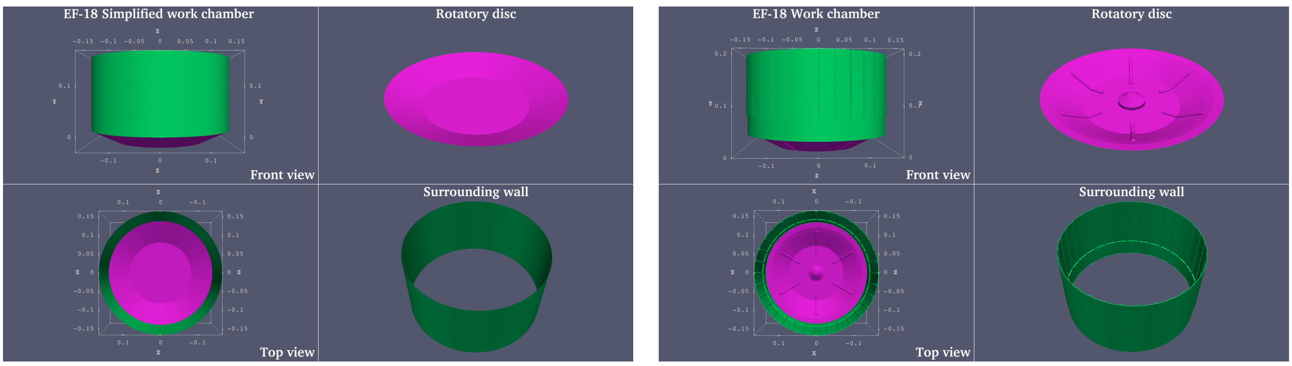
Table 6.1: Technical data

Type	Work chamber		Machine	
	volume (l)	Diameter (mm)	Dimension (W×D×H)	Weight (kg)
EF 18	18	333	620 × 780 × 1,520	115

part of the surrounding wall with a gap of few millimetres between then to allow the compound to exit the work chamber. The disc stops the media particle from leaving the work chamber although its main function is to rotate and energize the media particle in contact with it. To this end an electrical motor is attached to it. The controller of the machine allows to set the spinning velocity of the disc from 0 to 320 rpm. The Table 6.1 summarize the main characteristics of the machine.

6.1 Work chamber model

An abstract model of the bowl was considered in early simulations, Figure 6.2a depicts it. A simple model has a positive impact in the computing time as less elements need to be computed. Unfortunately, it was found that the absence of features in the bowl had a negative impact in the modelling of the granular flow. It was appreciated with naked eye the unusual behaviour of the particles in contact with the walls of the bowl that, at the same time, was transferred to the rest of the particles. The results of this early simulations are briefly discussed in Section 6.4.



(a) Simplified bowl model (EF 18)

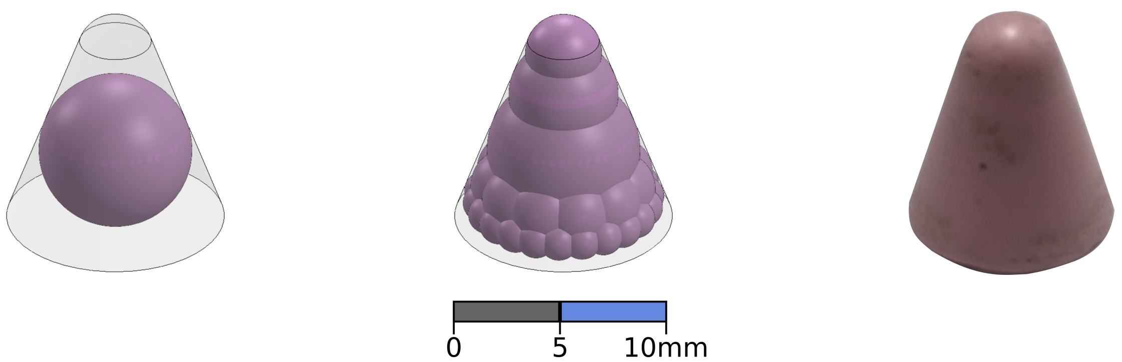
(b) Bowl model (EF 18)

Figure 6.2: Bowl model comparison

A posterior model was created which possessed all the features of the real bowl but the thickness of the walls. A much thinner walls were modelled as it does no affect to the flow behaviour and yet it reduces the computing time. Figure 6.2b shows the model with all the internal attributes found in the real part. The initial results obtained with this model were much more promising thus it was henceforth employed for all simulations.

6.2 Particle definition

Several features have to be taken into consideration when simulating the behaviour of the granular matter; it is crucial to well define the boundary conditions: discussed in the previous section. The mechanical properties of the matter to study, such as Young modulus, Poissons ratio, coefficient of restitution, coefficient of friction, etcetera. However, their geometries are as important as everything aforementioned. The simplest geometry to simulate is a two-dimension disc as in the Distinct Element Method[9]. The sphere is the easiest geometry when simulating a three dimensional space, and a good insight can be obtained using equivalent spheres as an abstraction of the real geometries. However, as it is discussed in Section 6.4, in order to obtain accurate results the simplest geometry need to evolve into the multi-sphere particle. Figure 6.3 represents this process. In this occasion thirty-six spheres were needed to correctly defined a single conical media particle.



(a) Equivalent sphere **(b)** 36 spheres clump **(c)** Media particle

Figure 6.3: Media characterisation

The media employed in this machine is similar to what was discussed in the Section 5.1 of Chapter 5. Its size is the main difference, being this one half the size of those used in the trough simulation. The composition remained identical.

6.3 DEM model

As for the vibratory finishing machine simulation, the Hertz model without cohesion was also the contact model employed for the simulations considered in this chapter. Table 6.2 summarizes the essential data of the MF machine, media and DEM parameters.

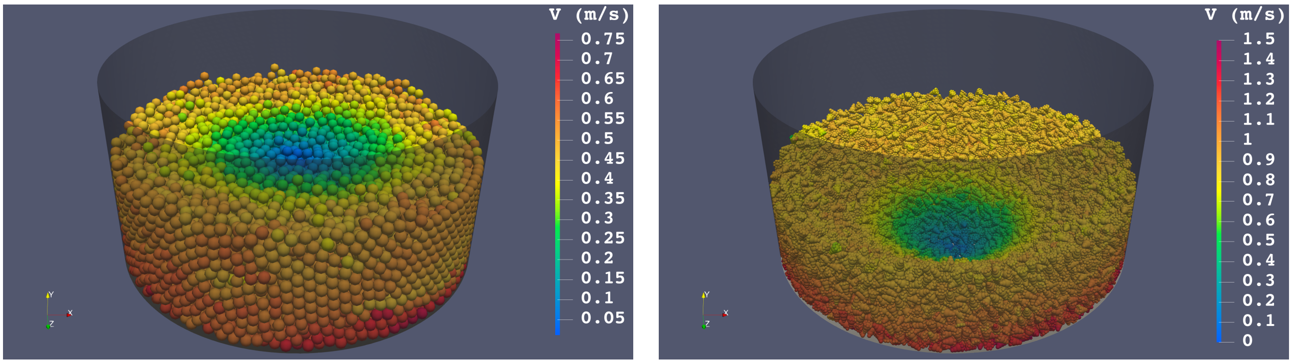
Table 6.2: Bowl simulation Parameters' summary

Parameter		Value
Rotatory disc finishing machine	Type	EF-18
	Rotation speed	200 rpm
	Lining material	Elastomer Polyurethane
Media	Weight of loaded media	7 kg
	Bowl height	210 mm
	Bowl diameter	333 mm
	Media particle average weight	0.00075 kg
	Shape	Conical media
	Material	Plastic
Mechanical properties of the material	Media density	1665.71 kg/m^3
	Media Poisson's ratio	0.265
	Media Young modulus	4.60×10^8 Pa
	Lining Poisson's ratio	0.25
	Lining Young modulus	2.50×10^7 Pa
	Media-media static friction coefficient	0.58
	Media-lining static friction coefficient	0.86
	Media-media restitution coefficient	0.67
	Media-lining restitution coefficient	0.159
	Coefficient of rolling friction (spherical model)	1
DEM specifications	Particle model	Hertz without cohesion
	Time step	2.857×10^{-6} s
	Dumping frequency	90 snapshots per minute

6.4 Early simulation studies

The early simulations of the rotatory disc finishing machine were carried out using the simplest model of the work chamber, Figure 6.2a. Two different models of the media particle were developed in an attempt to characterized the real media particle. The first model was the equivalent spherical model, Figure 6.3a. The average mass of the media particle was assigned to this simplification. The second model of the media particle is presented in Figure 6.3b. This consisted in thirty-six spheres clump together to shape the original geometry of the media particle. The average mass of the media particle was also assigned to this model. Furthermore, the mechanical properties used for both models are displayed in Table 6.2.

The bowl was loaded with 7kg of media. After all the particles were settle in the interior of the bowl the disc commenced to rotate at 200 rpm during a period of five seconds. It was established in the laboratory that the time required by this type of media to reach its steady-state is around a half of a second.



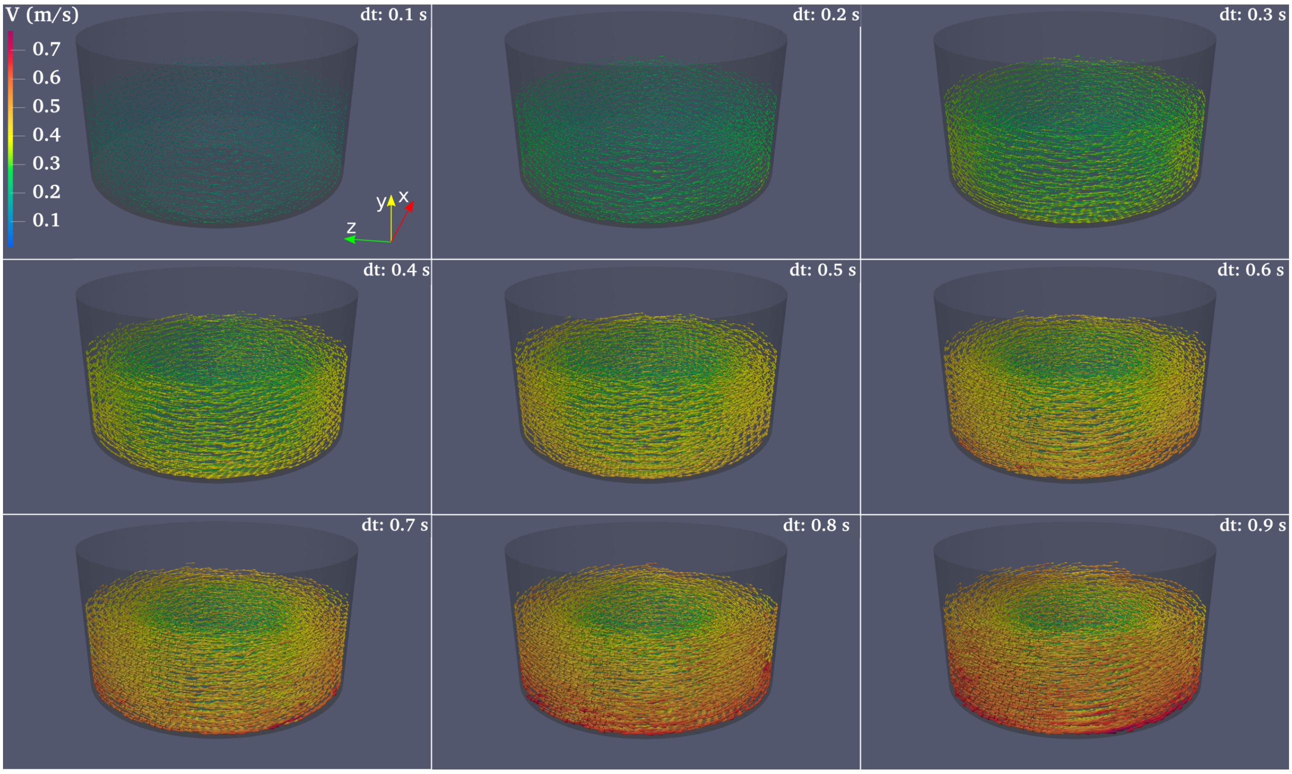
(a) Spherical distribution

(b) Multi-sphere distribution

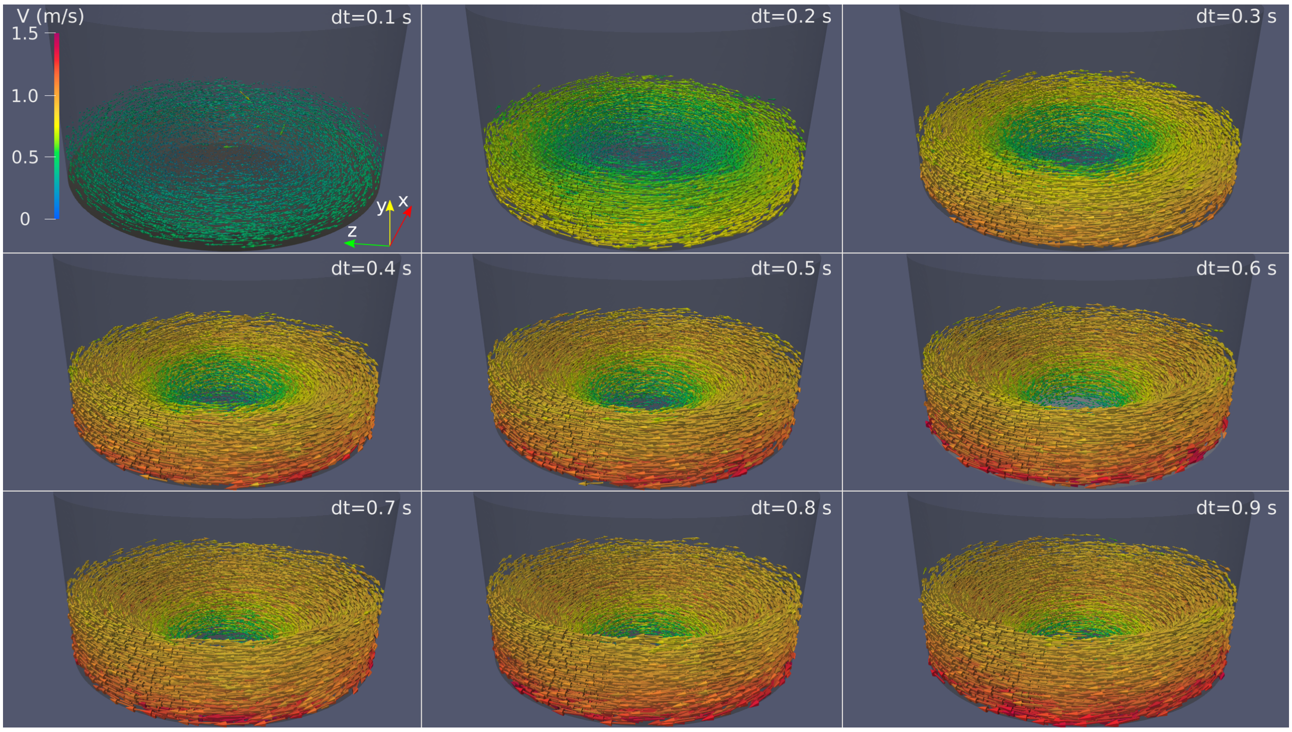
Figure 6.4: Spherical and multi-spherical steady-state flow behaviour

Figure 6.5 displays the distribution of velocities for the two aforementioned models when they are in the work chamber and the bulk is flowing at its steady-state. Although, the mass assigned to these models were identical the spherical model presents a lower packing density than the multi-sphere model. It is observed than the range of velocity differs depending on the model.

Figure 6.5a shows the distribution of velocity for the spherical model. A total of nine snapshots from the same simulation were extracted. The first snapshot, upper left, correspond to the time 0.1 s. The last one, lower right, depicts the instant 0.9 s. In this representation, the media particles were substituted by the velocity vector at their center of mass. The size of the vectors depends of their magnitude. Thus a particle travelling fast will have a bigger vector than those travelling slower. At **dt=0.1 s** the disc is starting to rotate, the particles are energized but the bulk is not yet flowing at its steady-state regimen. Therefore the velocity in the bulk is still slow in comparison to a later



(a) Early spherical bulk velocity distribution



(b) Early multi-sphere bulk velocity distribution

Figure 6.5: Spherical and multi-spherical early simulations

instant in time. It is observed that the dominant color among the vectors is the green which correspond to an approximate velocity of 0.3 m/s. It is not until **dt=0.8 s** that the bulk reaches its steady-state. The distribution of velocities in the bulk after this instant does not change substantially. As it was to be expected those particles in contact with the rotating disc or in its proximity are travelling to a higher velocity that those in other regions of the bulk. For this reason the vector at the bottom are predominantly red and bigger than the rest.

Likewise, Figure 6.5b depicts the velocity distribution for the multi-sphere model at the same range of time. It is observed that at **dt=0.1 s** the predominant color among the vectors is also the green which correspond to an approximate velocity of 0.5 m/s. At **dt=0.6 s** the bulk reaches its steady-state. Several differences are noticed between the two media models:

1. For the same mass the spherical model fills a larger volume: lower density.
2. The time needed for the bulk to reach the steady-state is longer for the spherical model. Besides, this one flows slower than the multi-sphere model.
3. Due to the two previous points the spherical model does not create the vortex observed in the center of the bulk,Figure 6.6, when it flows in these specific conditions.

It was noticed for both simulations that the particles seemed to be travelling in a laminar-like flow during the five seconds that the simulation lasted. All vectors were contained in planes parallel to the xz-plane. According to the simulation, the particles rotates with respect to the y-axis at different velocities but they barely change their position in the y-axis direction. In other words, the particles do not travel upwards and downwards. Furthermore, the size of the vectors near the surrounding walls are bigger than those in the inner part of the bulk.

The steady-state of the real bulk was recorded using a video camera with a framerate of 30 fps. Thus, three frames correspond to a period of time of 0.1 s. Figure 6.6 displays the first of every three frames of this recording. It is observed a vortex is created at the centre of the bowl. Contrarily to what was observed in the simulation using the spherical media. There are particles in the vortex clearly going downwards. Beside, those particles in contact with the surrounding walls or in its vicinity are clearly travelling slower than in other visible regions. The trajectory of the particles observed when the system is working at its steady-state has been printed over the bulk in the frame

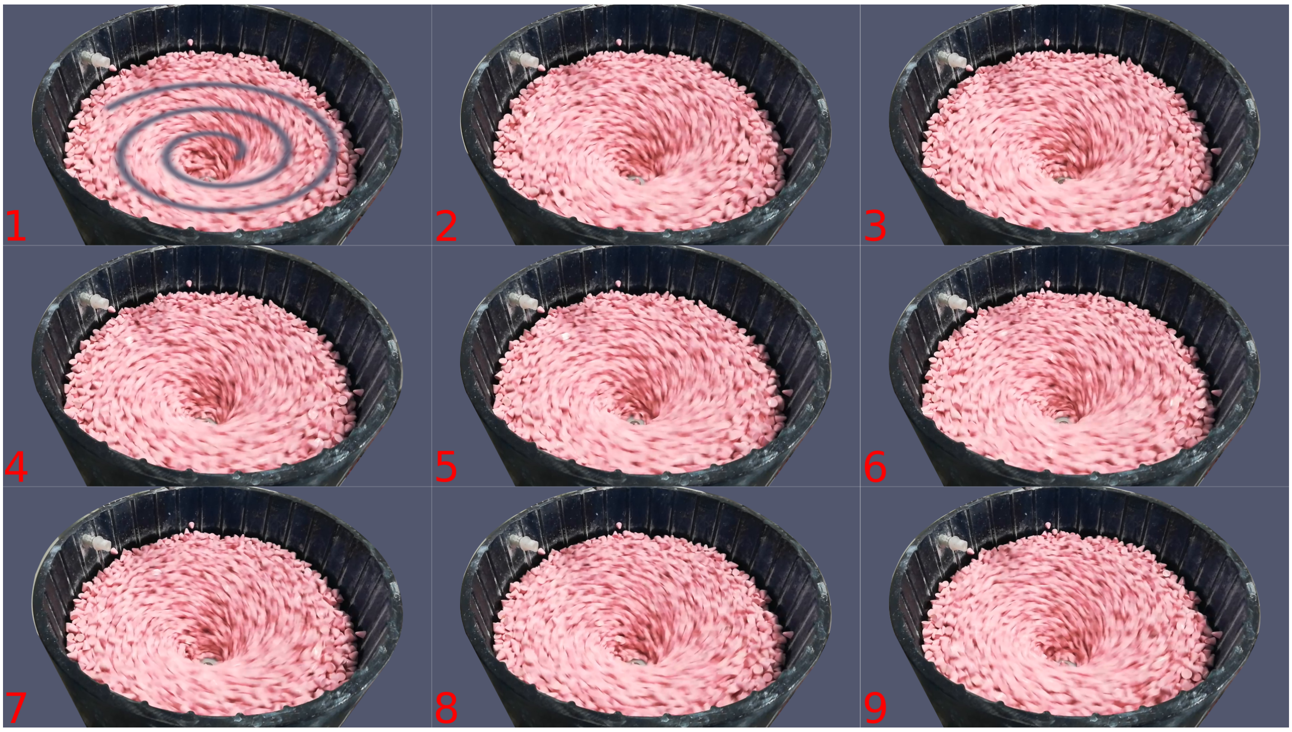


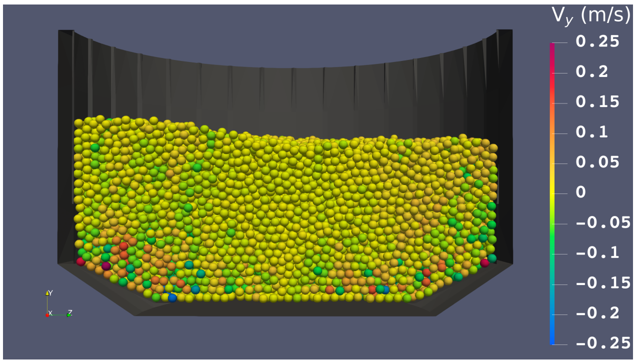
Figure 6.6: Real flow behaviour

presented at the upper left corner.

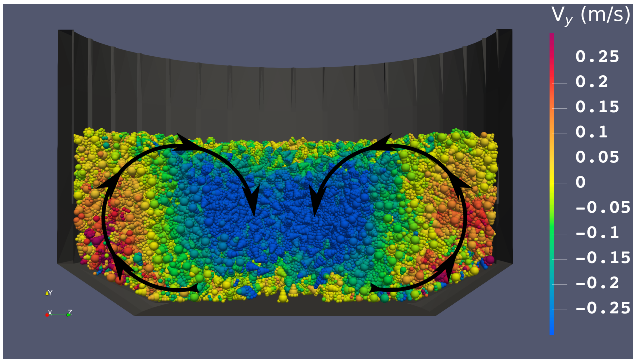
The cause of this unusual behaviour was identified as the poor particle-wall friction due to the lack of ribs on the walls of the work chamber. In an attempt to correct this behaviour the friction coefficient was increased gradually to a value twice the original. No clear benefit was found.

Work chamber advanced model

Two new simulations were developed using the same parameters than the two aforementioned simulations. The only difference was the characterization of the work chamber. A much more detailed bowl was modelled, the ribs on the inner part of the original wall were accurately reproduced in the model. The ribs at the bottom were obviated to diminish the computing time. Figure 6.7 shows the comparison between the velocity distribution in the y-axis direction (V_y) for the spherical and muti-sphere models when in the interior of the work chamber advanced model. The predominant color within the spherical distribution is yellow which represents no displacement in the y-axis direction. A very low percentage of particles are moving upwards or downwards within the bulk. It is also noticed the packing density for the spherical model does not improve with the new bowl model. On the other hand the multi-sphere model produces a much more interesting results. Particles are vertically travelling and a certain trend is identified. Particles descend when they reach the center of the bowl while the area from where the emerge to the surface is at the bottom, closer to the walls. The arrows in Figure 6.7b represent this flowing trend.



(a) Spherical distribution



(b) Multi-sphere distribution

Figure 6.7: Comparison between particle models

As a conclusion, the spherical model provides a poor insight of the behaviour of the media within this scope. It can be used to have a general idea of the most apparent trends of the media but it is insufficient to gain a valuable understanding of the process. Wang et al.[67] successfully simulated spherical media within the work chamber of a rotary vibrator using a spherical media model. However, other morphologies of media, such as the conical media employed in these simulations, have to be correctly characterized using multi-sphere model as a spherical model would not produce an accurate representation of the real process.

The multi-sphere model produces more realistic results. The patterns describe by it apparently match the trends observed in the real phenomenon. For this reason the thirty-six sphere model was established as the valid model for all the simulations commented in the rest of this chapter.

The detailed characterization of the work chamber is essential to retrieve accurate results from the simulation.

6.5 Case study I

The experiment followed to validate the simulation consisted in using a wireless sensor to retrieve information regarding its position and velocity and then simulating the same conditions and comparing the results. The sensor employed was the MMC – MetaMotionC from Mbientlab, it consisted of a cluster of sensors on a PCB with the ability to stream live the raw sensor data. A protector case made of POM-C was designed in order to protect the sensor from the media impacts and the lubricant compound. The dimensions of the case were ϕ43.0 mm and 12.50 mm height. Figure 6.8 shows the case and the sensor within.

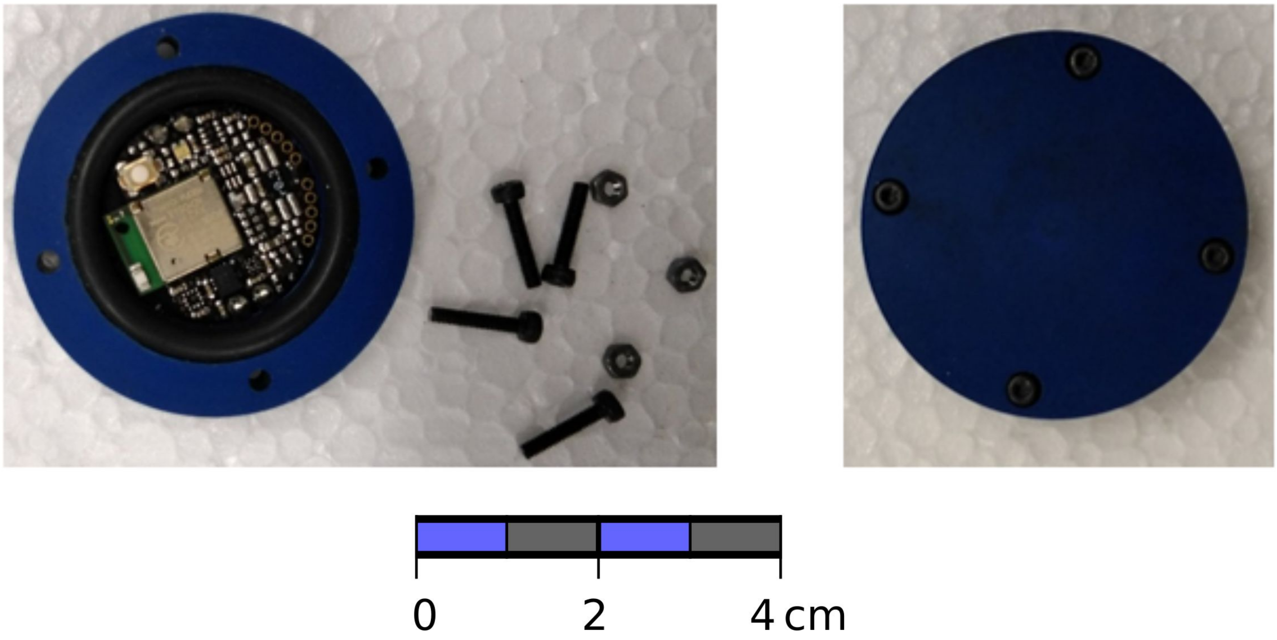


Figure 6.8: Wireless sensor and is protective enclosure

The data retrieved from the sensor shed light on the position and velocity of the sensor at any time during the length of the experiment. All this information was post-processed and filters applied to reduce the level of noise in the signal in order to extract a truthful output[EPSRC EP/N022998].

Due to the importance of the geometry definition all features were considered for the bowl model used in this simulation, not only the ribs on the wall were detailed but also all the features on the bottom disc. The approach that was followed to minimize the computing time was to reduce the quantity of mesh elements, which has increased as a consequence of the level of detail of the model, by reducing the thickness of the wall to the extent it became a shell. In order to define the sensor case in the simulation an identical process to the one used for the media particle was employed. A total of forty-three spheres were used to characterize the virtual case. The density assigned to the virtual case was the one that produced a virtual weight identical to the weight of the real assembly. The media particle definition remained unaltered. The mechanical properties of the materials as the DEM specification and the working velocity of the system are stated on Table 6.3. The simulation script written for this case study can be found in Appendix .

Table 6.3: Simulation parameters

Description	**Value**
Time-step	2.86×10^{-6} s
Total mass	7 kg
No inserted media particles	14,259
Rotational velocity	200 rpm
Media density	1,665.71 kg/m^3
Sensor + case density	1,810 kg/m^3
Media-Media friction	0.58
Media-Sensor Case friction	0.34
Media-Lining friction	0.86
Sensor Case-Lining friction	0.86
E (Media)	4.6×10^8 Pa
ν (Media)	0.2887
E (Sensor Case)	4.6×10^8 Pa
ν (Sensor Case)	0.35
Media-Media restitution	0.67
Media-Sensor Case restitution	0.35
Media-Lining restitution	0.159
Sensor Case-Lining restitution	0.159

The simulation commenced with the insertion of 7 kg of media in the bowl and this was allowed to settle, subsequently the sensor case was introduced to the surface of the media bulk. The driving bottom disc was rotated at 200 rpm for 5 seconds. Table 6.4 summarises the different steps needed to obtain the simulation. A total of 108 hours and 216 HPC CPUs were needed to produce a 4.7 s long simulation.

Table 6.4: Simulation summary

Step	Job ID	CPU	Time simulated (s)	Computing time (h)	KinEng (j)	Mass (kg)	Media Particles	Task
1	3426	216	0.5	15	1.02e-04	7.03	14,265	Filling the bowl
2	3428	216	0.2	6	1.35e-05	7.03	14,265	Media settles
3	3430	216	1	21	3.429	7.03	14,259	Disc starts rotating
4	3432	216	1	22	3.458	7.02	14,257	Disc keeps rotating
5	3433	216	1	21.5	3.468	7.02	14,257	Disc keeps rotating
6	3434	216	1	22.5	3.431	7.02	14,257	Disc keeps rotating

Figure 6.9, at the left hand side shows the virtual bulk and sensor case prior to the start of rotation. At the right hand side it shows the evolution of the vertical position of the Center of Mass (CoM) of the case during the length of the simulation and below the magnitude of its velocity.

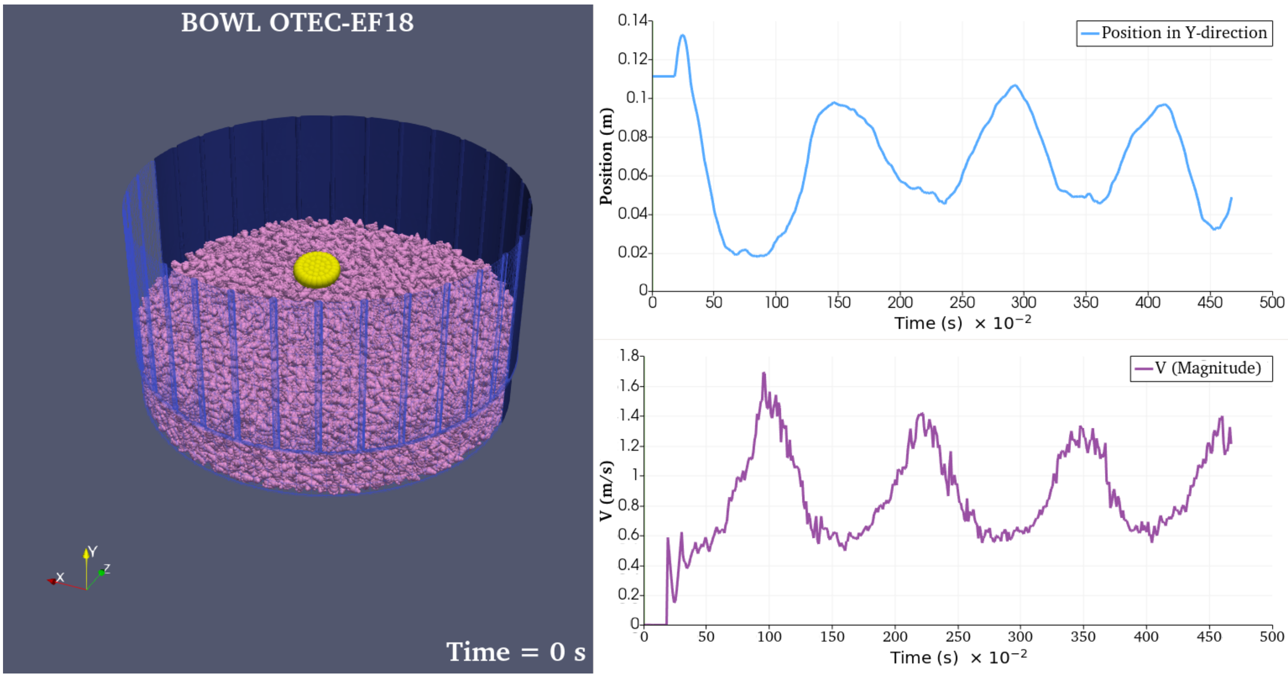


Figure 6.9: Media and Sensor Case at rest

Figures 6.10 and 6.11 show different snapshots of the simulation. The vertical green lines on the graphs indicate the time at which the snapshot was taken. Initially, it can be observed there is a correlation between the peaks in the position graphs and the valleys in the velocity graph and viceversa. The lower the position of the case, the higher the magnitude of the velocity. As previously mentioned, this is due to the fact that the particles located in the vicinity of the rotating disc are travelling faster than those in other regions of the bowl.

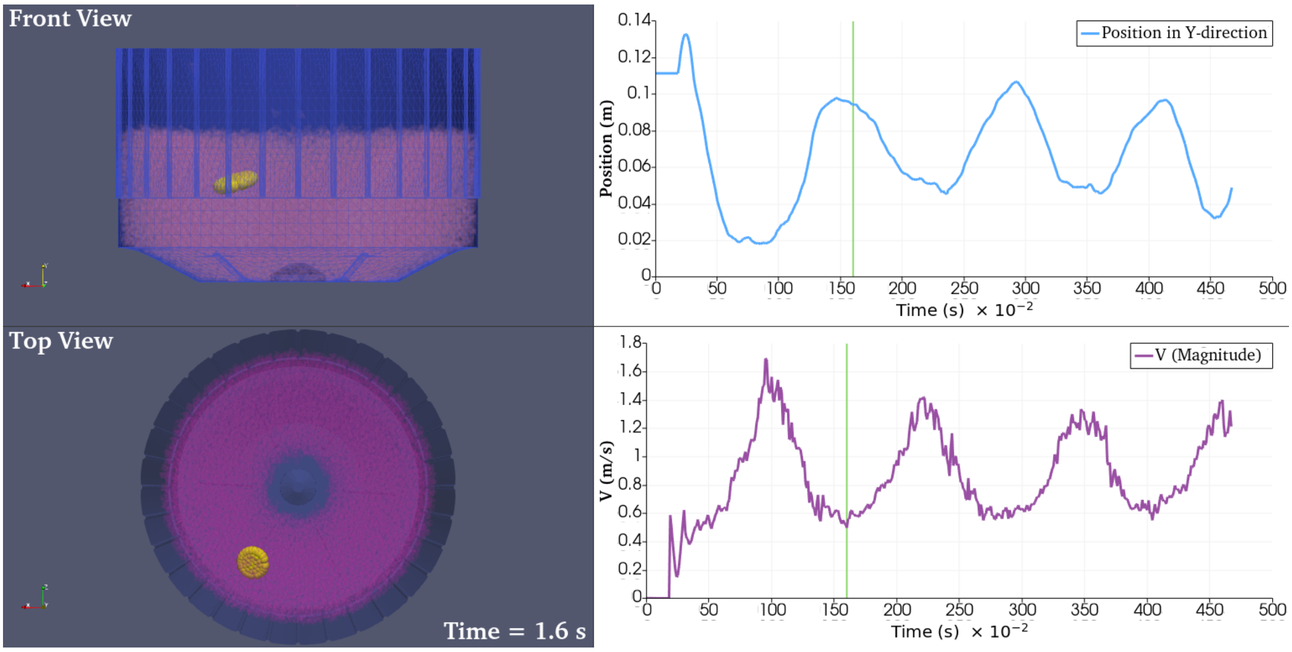


Figure 6.10: Media and Sensor Case at 1.6 s after the media is energised

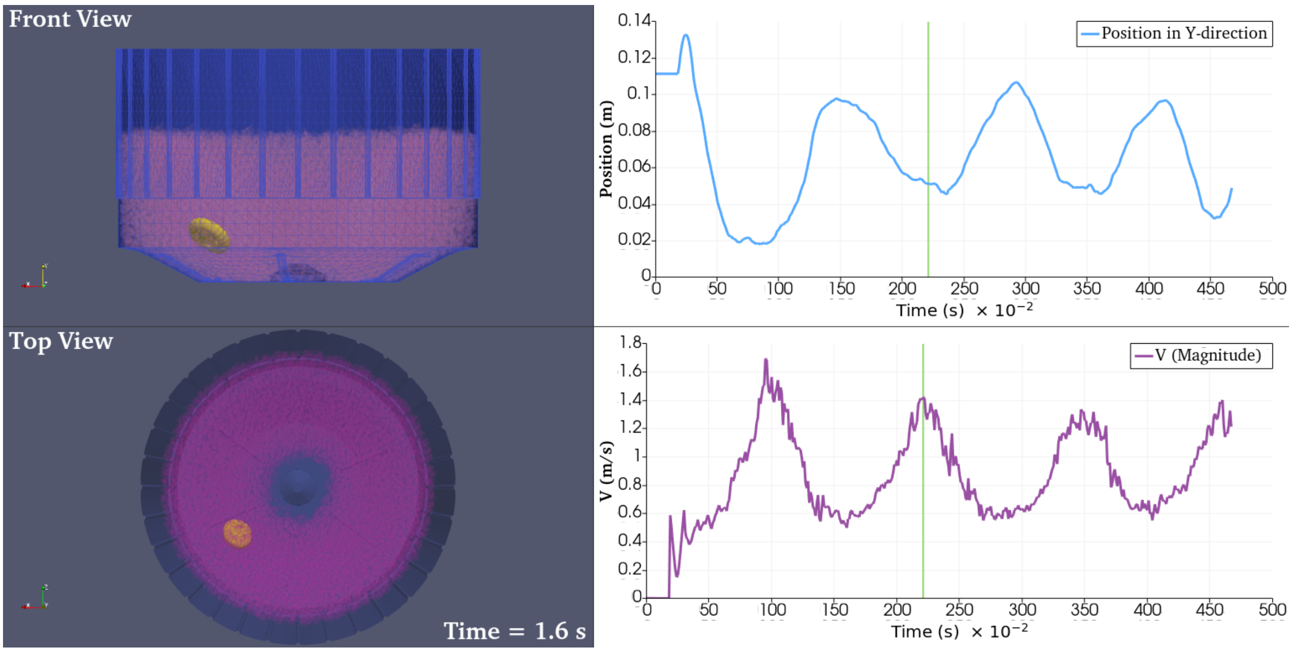


Figure 6.11: Media and Sensor Case at 2.2 s after the media is energised

6.6 Results and Discussion of DEM Simulations

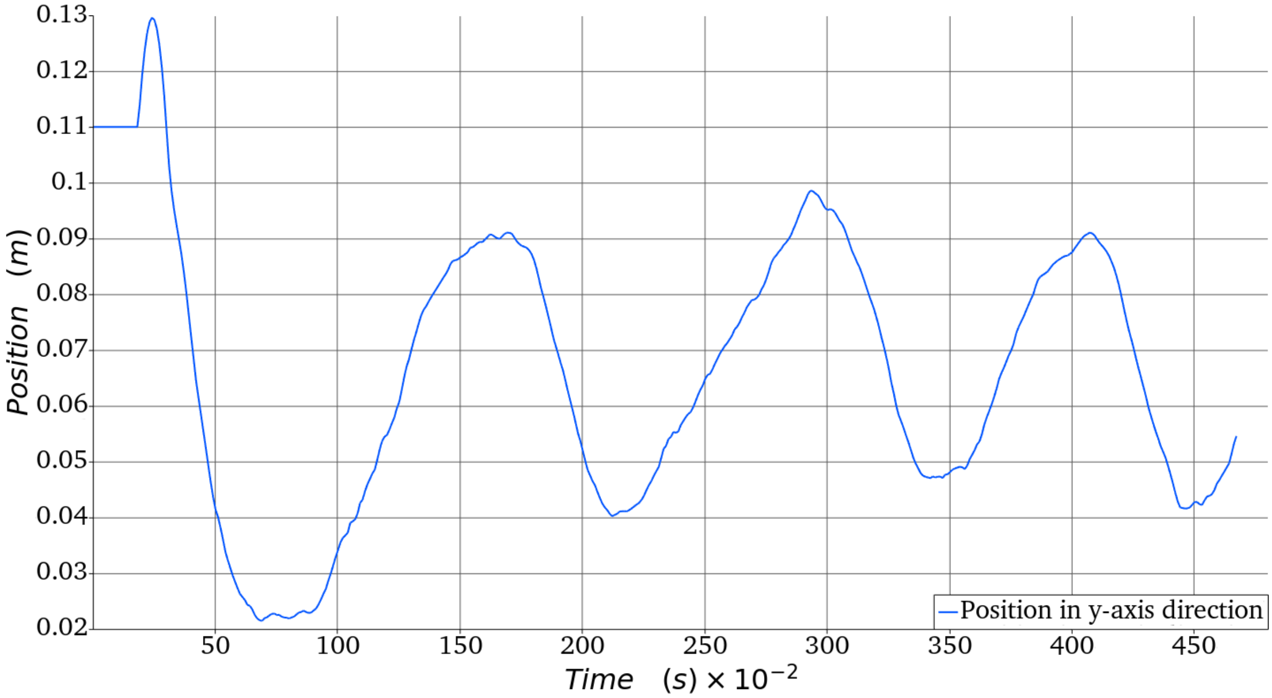
This section reports on the results obtained in the simulation and also, by way of comparison, those obtained from the sensor. In order to distinguish the results, the term virtual sensor is used for the simulation result while real sensor is used to refer to the results obtained with the sensor. All the data obtained from the simulation refers to the CoM of the sensor case.

6.6.1 Comparative trajectories

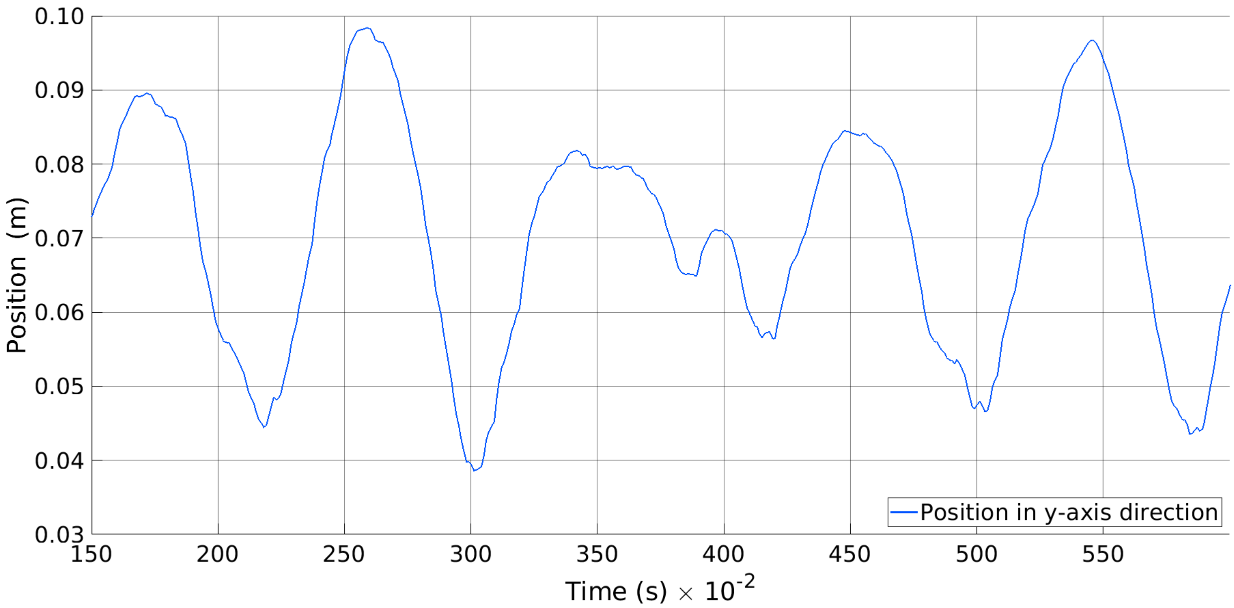
Figure 6.12a shows the position in the vertical direction (Y) of the virtual sensor during the simulation. The initial height of the virtual sensor is 0.11m from the bottom of the bowl. When the bulk media is energized all particles including the virtual sensor are observed to accelerate vertically reaching the highest position during the process. After 0.02 seconds the bulk is running at its normal regimen. The virtual sensor position changes with time in a pseudo-sinusoidal wave. It takes approximately one second to complete a cycle. The amplitude of the wave is approximately 0.07m. Figure 6.12b depicts the position of the real sensor in the vertical direction obtained from the sensor readings. The reference point of the real sensor is the sensor itself. Due to this reason it was originally positioned in the centre of the bowl and half way to its bottom. As can be appreciated in Figure 6.12b the vertical movement of the real sensor compares to a pseudo-sinusoidal wave. The amplitude of this wave is 0.07m as it was for the simulated wave. The real sensor was immersed in the bulk for 20 seconds, four times longer than the duration of the simulation. It was observed that within a 5 second period the number of full cycles was very similar to the simulated data. Figure 6.13a shows the position of the virtual sensor in the plane XZ. It is observed that the range in the X direction is -0.12m to 0.11m (delta: 0.23m) and the range in the Z direction is -0.118m to 0.122m (delta:0.24).

Figure 6.13b displays the position in the horizontal plane of the real sensor. It can be observed how the range of displacement in the X and Z direction closely matches the simulated ranges.

From this comparison it can be concluded that the real sensor and the virtual sensor follow very similar trajectories when within the bulk.

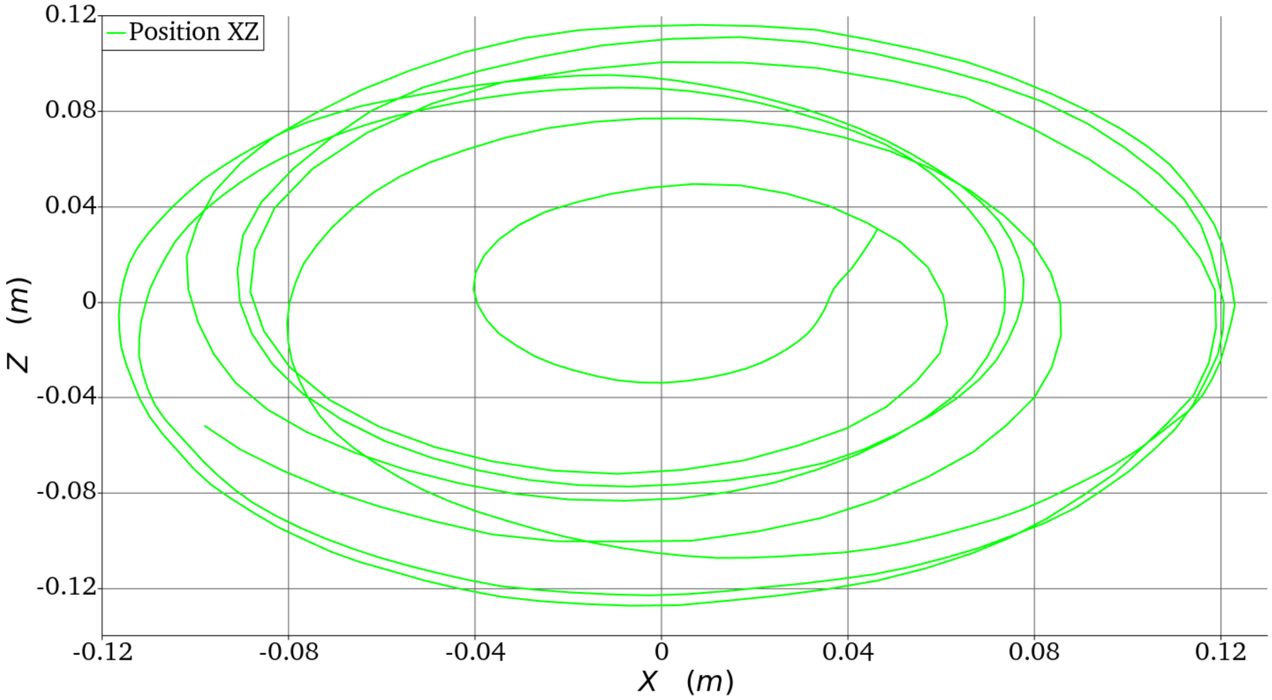


(a) Virtual sensor

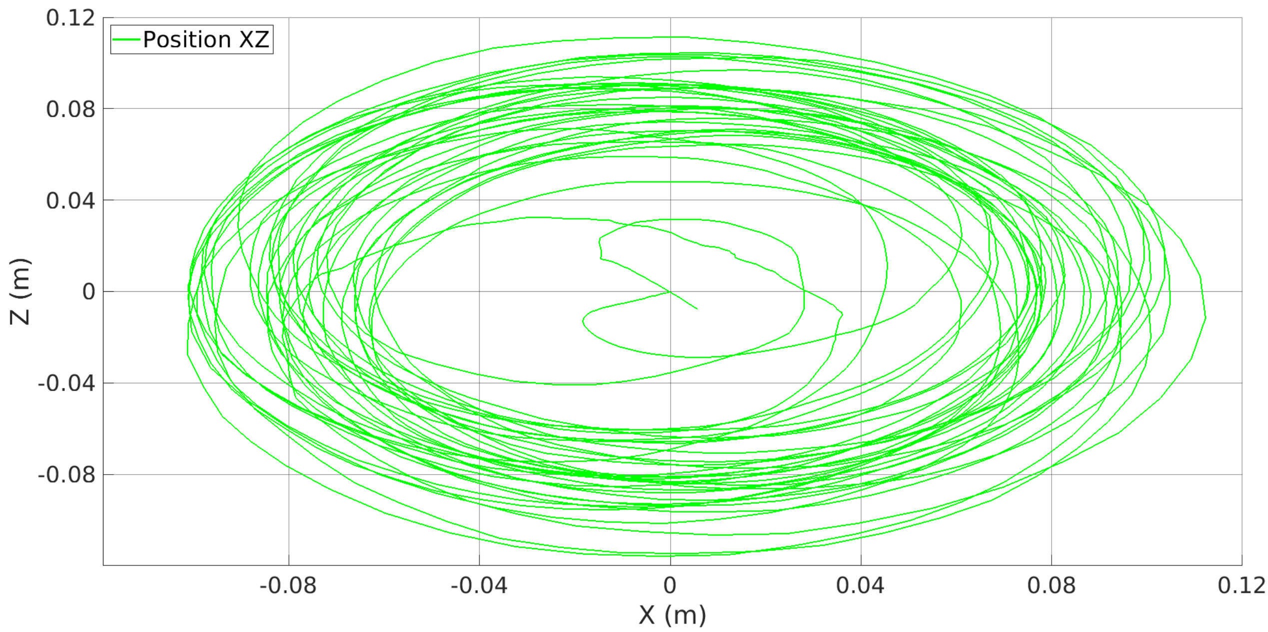


(b) Real sensor

Figure 6.12: Position of the sensor in the y-axis direction



(a) Virtual sensor



(b) Real sensor

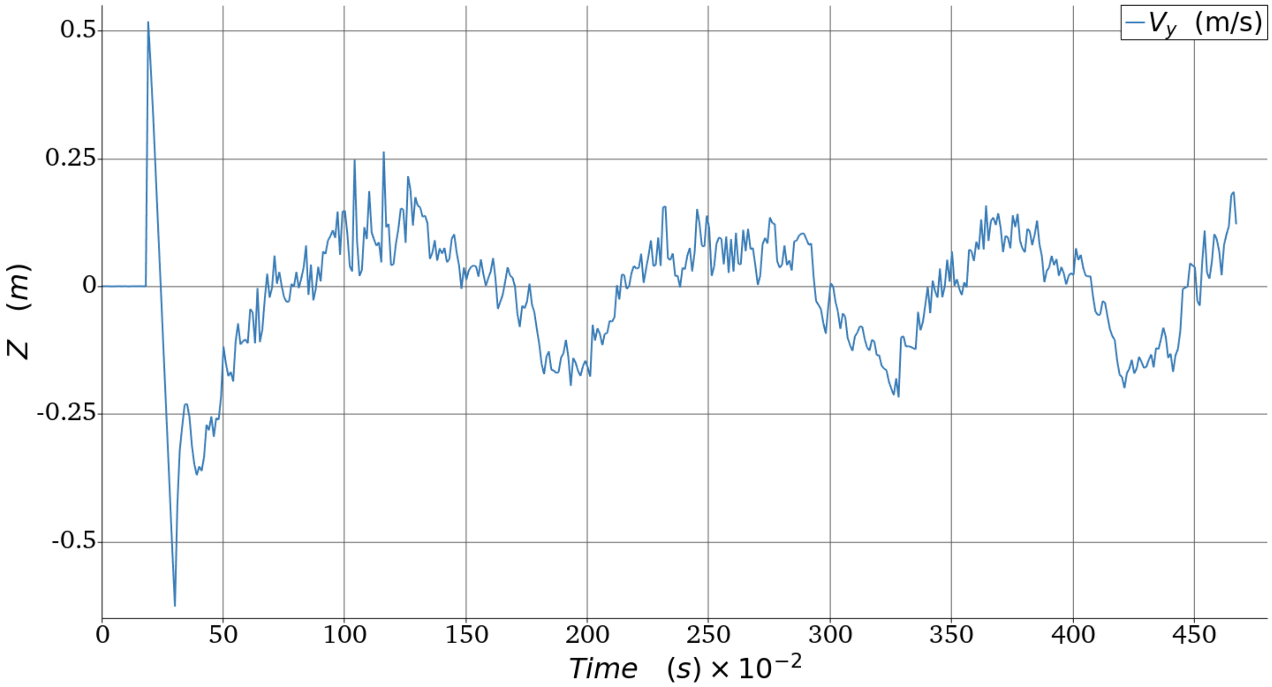
Figure 6.13: Orthogonal projection of the sensor's position onto the horizontal plane (XZ)

6.6.2 Comparative velocities

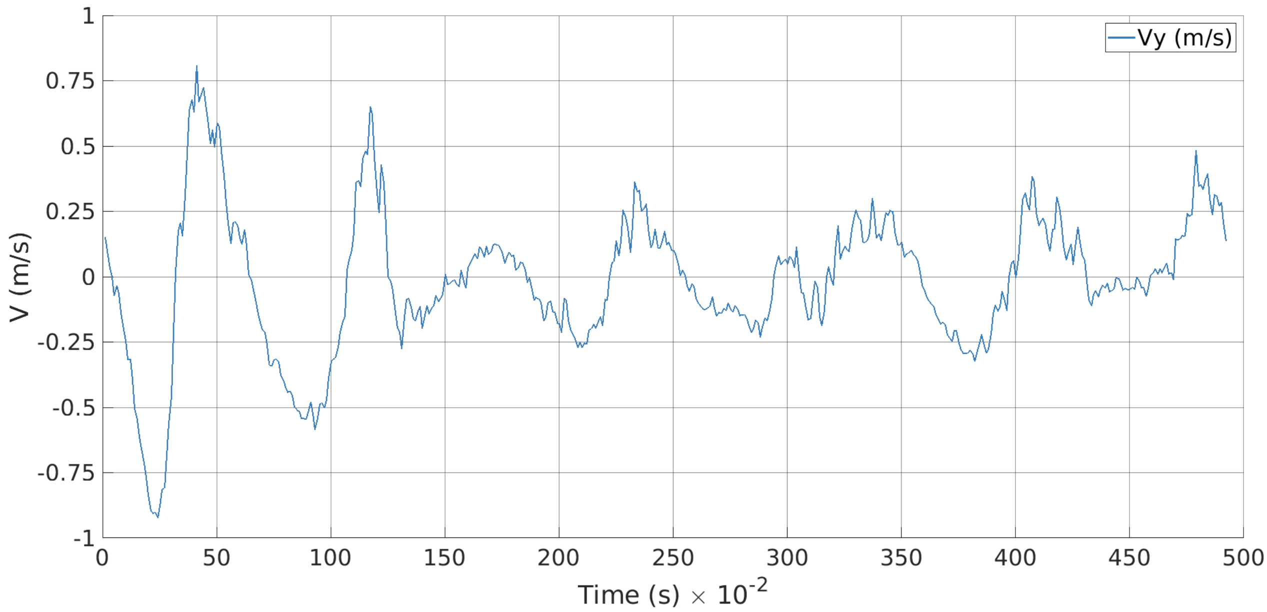
The sensor employed to study the particle flow also provided data with respect to the velocity. Figure 6.14b shows the Y components of the velocity vector during a five second period. The vector in the Y direction represents the movement of the real sensor up and down in the bowl. As can be observed the signal oscillates around zero. The range of velocity for the Y direction is 0.77 m/s to 0.86 m/s. However, after one second the velocity is stabilized to a range of velocity below 0.5 m/s in both directions. A similar behaviour is observed for the virtual sensor, Figure 6.14a. In this case the range of velocity is also maximum at first and then stabilizes. Although the velocities in this direction are always slightly lower than the velocities obtained from the real sensor, the velocity cycle is sustained for approximately 1 second in both cases.

The work chamber is perfectly symmetric with respect to the XY and ZY planes. Hence the signal produced by the sensor for the velocities in the X and Z directions is also expected to be symmetric although with a phase difference between them.

Figure 6.15b displays the signal for the three components of the velocity retrieved from the real sensor. It can be observed that the signals for the components X and Z are similar. The range of velocities for the X direction is 1.65 m/s to -1.74 m/s while for the Z direction it is 1.73 m/s to -1.51 m/s. The difference in the phase varies between $20 \times 10^{-2}(s)$ and $27 \times 10^{-2}(s)$, which is sensible as a full cycle of the sensor takes approximately 1s. Figure 6.15a shows the results obtained from the virtual sensor. The velocity signal in the X and Z direction produced by the simulation is more regular, closer to the theoretical results than the one obtained from the real sensor. It can also be observed that a difference in the phase is of the same order as that observed for the real sensor.

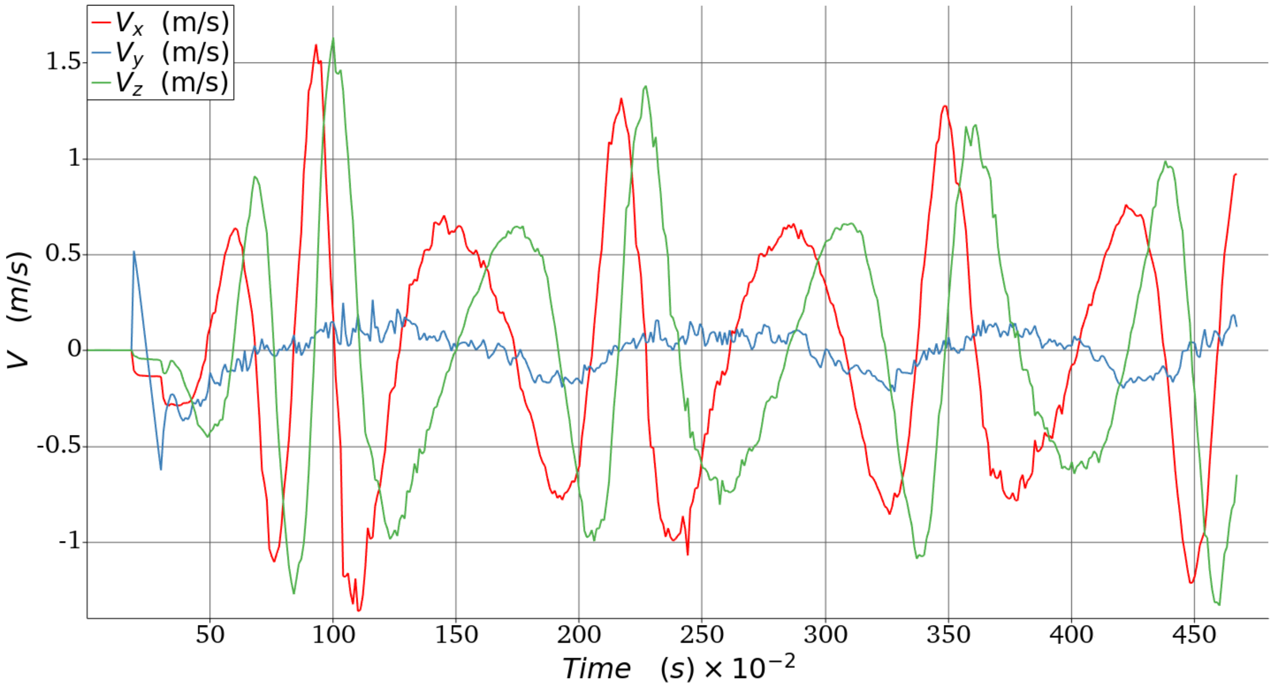


(a) Virtual sensor

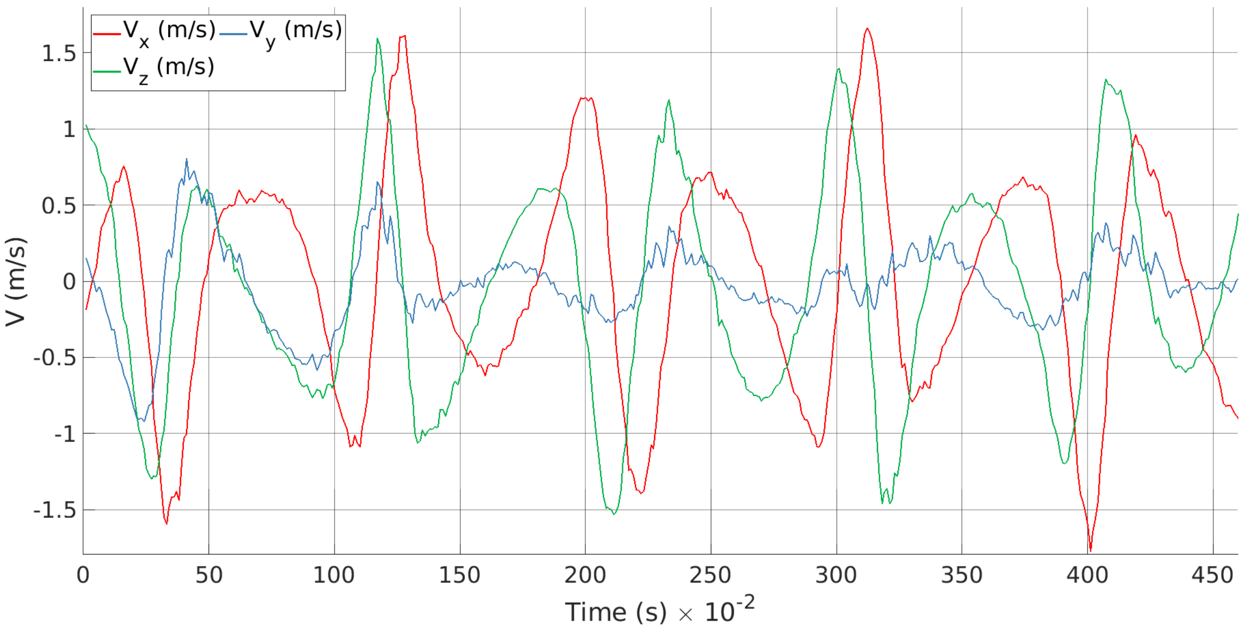


(b) Real sensor

Figure 6.14: Velocity of the sensor in the y-axis direction



(a) Virtual sensor



(b) Real sensor

Figure 6.15: Velocity of the sensor in the x-, y- and z-axis directions

6.6.3 Simulated velocities

DEM simulations provide information of the individual particles within the bulk. Figure 6.16 is a snapshot of the bulk within the work chamber in its normal regimen for an arbitrary moment in time.

The distribution of velocity for the vertical direction is also shown. As can be observed the particles at the centre of the bulk are coloured blue which represents the velocity in the negative direction. The central part of the bulk descends. At the top of the bulk a mix of dark yellow and light green coloured particles are observed. The particles situated closer to the central vortex are commencing to descend while the particles closer to the wall have recently emerged from a deeper position. Figure 6.17a shows six different sections on the bulk at the same moment in time. These are separated by 0.027m from the previous and subsequent sections. The first section (Clip 1 Z=0) divides the bowl in two halves. The incline profile of the bulk at the central region can be appreciated. Here, the particles situated between the wall and the vortex, coloured in red and orange are ascending. The particles in contact with the surrounding wall can be either ascending or descending, depending on the moment in time the snapshot was taken, this is due to the ribs on the wall which prevent the media from flowing creating a mix of direction. A similar distribution of velocity can be appreciated in the second section where the main difference with respect to the previous section is a reduction of the free volume at the centre. The third section is at the edge of the vortex, no free volume is observed and instead, in the central region, there is a clear majority of media particles descending. As observed in the previous sections the particles situated at both sides ascend. The fourth section is situated at 0.027m from the edge of the vortex and it can be observed that the particles descend in the central region at a slower velocity than at the edge of the vortex. The two last sections reference the region at two thirds from the centre of the bowl, it is in this region where the ascension of media particles occurs.

The trend of the flow starts with the particles rotating on the disc moving from the inner part towards the wall, the inclination of the rotating disk launches the media to the surface of the bulk displacing the media in the surface to the centre of the bulk where the vortex is created and particles descend to the bottom where the cycle starts over.

Figure 6.17b displays the vector of velocity within the bulk, it is observed throughout the different sections that the highest velocities are always found near the bottom of the bowl, constituting a 0.04m layer of media particles. The media located above this layer comprise a thinner layer where the velocities are still high, 0.06m to 0.08m from the bottom. The lowest velocities are located where the media is in contact with the surrounding wall, especially in the area above the first quarter of the work chamber as it is there where the ribs commence. If introducing a fixed specimen within the bulk the high speed layer at the bottom of the work chamber should be avoided as the flow pattern would be affected, modifying the efficiency of the process. The ideal spot for fixing the specimen would be in

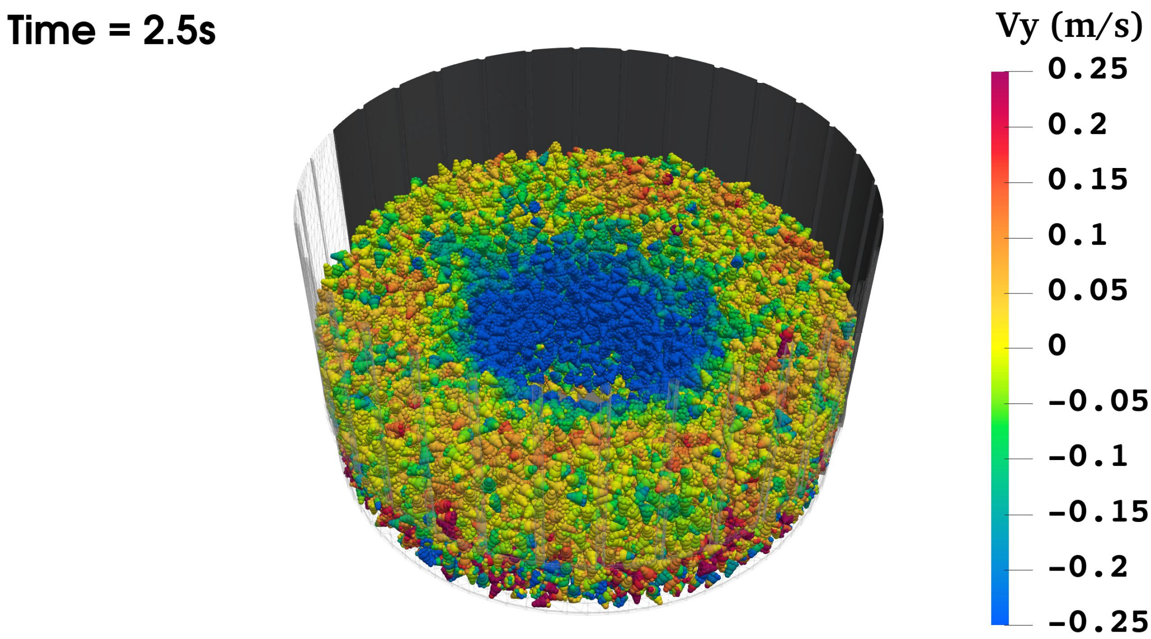
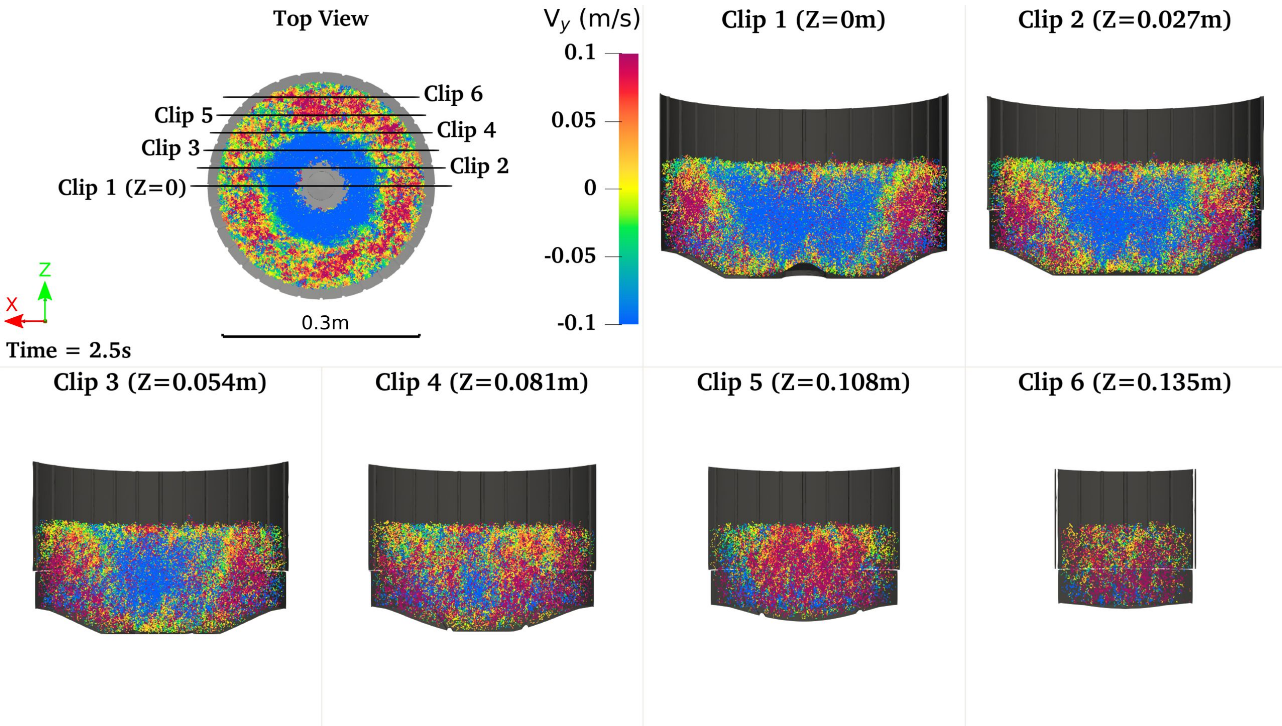
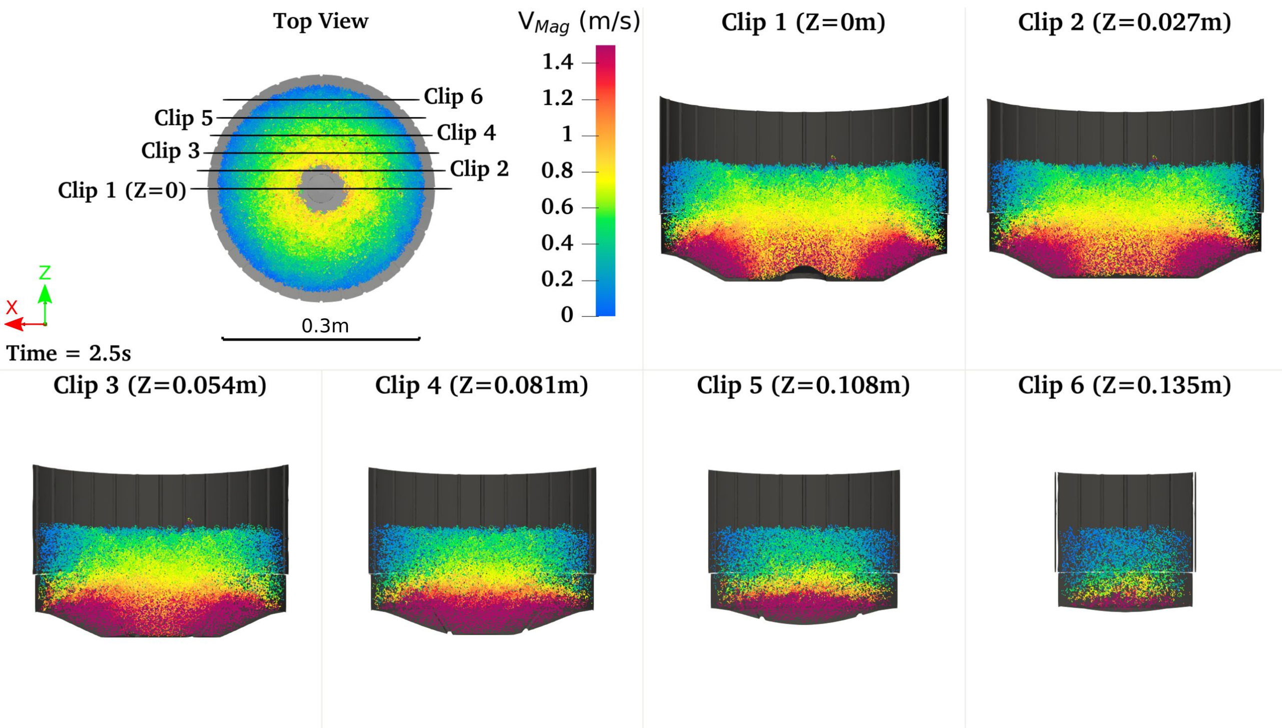


Figure 6.16: Distribution of V_y in the bulk

the layer immediately above the high layer and within 0.027m and 0.054m from the surrounding wall. This important finding will be used to choose the location of the specimens' study in Chapter 7.



(a) Velocity in the y-axis direction



(b) Velocity magnitud

Figure 6.17: Velocity distribution within different sections

6.7 Summary

It has been emphasized over this chapter the importance of using the multi-sphere approach in the task of simulating the complex shape of granular materials in the MF system. It was found that the multi-sphere approximation to the particles' real shape always produced more realistic results than the spherical approximation. Therefore, it was concluded that the spherical approximation does not yield enough accurate data to be used in the design of future machines or to improve the current processes. Moreover, the multi-sphere particles clarified the cyclical flow trends o f the particles when in the work chamber of the rotary machine and the distribution of velocities that could lead to position the work-pieces in the appropriate regions of the bulk.

The novel approach of using a wireless sensor to compare the trajectories and velocities of a real work-piece with a simulated one when travelling within the bulk of the rotatory disc machine yielded successful results. It was concluded that the simulation offered very realistic results and it can be used to understand the flow trends within work chamber, and the impact that the design features had on the behaviour of the flow.

The information extracted from the simulations presented in this chapter was key to progress with the work presented in the following chapter.

Chapter 7

Stress studies on specimen surfaces

LIGGGHTS [16] primarily generates data with respect to the particles such as position $\vec{r}(\vec{x}, \vec{y}, \vec{z})$, velocity $\vec{V}(\vec{V}x, \vec{V}y, \vec{V}z)$, force $\vec{F}(\vec{F}x, \vec{F}y, \vec{F}z)$. It also provides information about the distribution of stresses on the surfaces of the part in contact with the granular flow as a consequence of the particles' impacts. DEM does not provide information on how the superficial stresses affect the subsurface zones of the solids or other areas of the surface not affected by the impact itself but for the propagation of the stress along the surface. To shed light on the behaviour of these zones it is necessary to employ Finite Element Method (FEM),.

It is also important to mention that the public version of LIGGGHTS considers all the solids as fully fixed, therefore there will be no translation of any node in the mesh hence no deformation in the element forming the geometry. For this reason, when the impacts of the particles on the surfaces of the part produce high values of stresses leading to large deformations, either because of the energy contained in the particles, the mechanical properties of the part or the configuration of the boundary conditions, the results obtained would not be realistic. This limitation will be explained in section 7.1.1 on page 126.

In order to analyse the influence of the impact of the particle on the work specimens a two-step process is needed. First the superficial stresses obtained from LIGGGHTS are converted into forces at the nodes of the elements affected by the impact of the particle. Then, the second step translates the geometry, the loading scenario and the boundary condition to a format readable by the FEM, solver.

7.1 Simulation steps

A simulation has several steps before the results can be visualised and interpreted. The diagram depicted in Figure 7.1 shows the compendium of steps and the software used in each step.

The first step involves defining the geometries, comprehending the physical boundaries of the simulated process and the parts interacting with the granular flows. For example, the bowl that contains the media in the mass finishing process would be a physical boundary while the specimens immersed in the flow are parts that modify the behaviour of the flow but do not contain it. This step in a sentence: defining the solid matter interacting with the granular matter. For this purpose, a CAD package is used. FreeCAD (`https://www.freecadweb.org/`), is the software employed in all the parts related to the design works carried out for the completion of this research. This tool is a free and open-source general-purpose parametric 3D CAD modeller software. Figure 7.2a displays the geometry of a simple cuboid.

The following step is to discretize the geometry. This consists in replacing the continuum geometry with a finite set of elements and nodes. GMSH[75] is used for this purpose. Figure 7.2b shows how the geometry initially defined in the CAD package, has been reduced to a set of 12 hexahedra and 36 nodes. The type of hexahedron used for this simulation is a linear element, although a parabolic element, a high-order element, produces more accurate results, in this case is not possible as the element used to mesh the volume must be recognized by the FEA, solver and the VTK converter. The linear hexahedron is the only type of hexahedron available in the three software packages. The discretized volume has to be exported in ASCII stl surface format which is the only format LIGGGHTS is currently reading. Although it is said in its documentation it can also read VTK files, a bug in the code does not allow using the VTK files. As a consequence, transferring the loads from DEM to the FEA, solver involves a high consumption of computational resources. This happens because the solid has many more nodes than the shell, so the IDs of the nodes that initially form the solid are lost when GMSH converts it into a shell. When applying the load back to the solid, the nodes have to be identified by their coordinates and not the IDs. The ID is just a single number to identify a specific node, while the coordinates are a set of three numbers for the node.

The DEM step is where the interactions between the granular matter and the solid matter are defined. For this purpose a simple experiment was configured. It consists in a stainless steel cuboid,

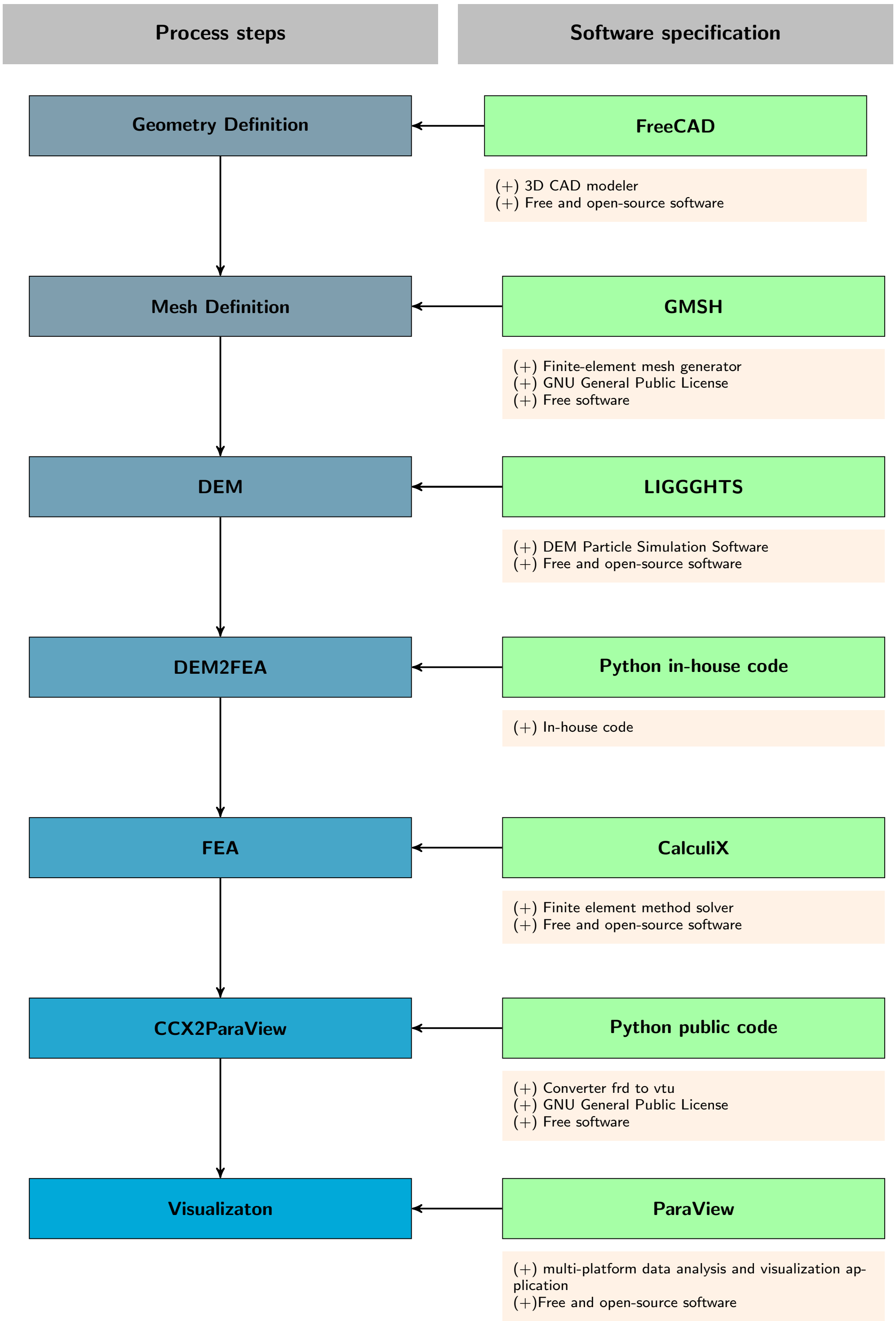
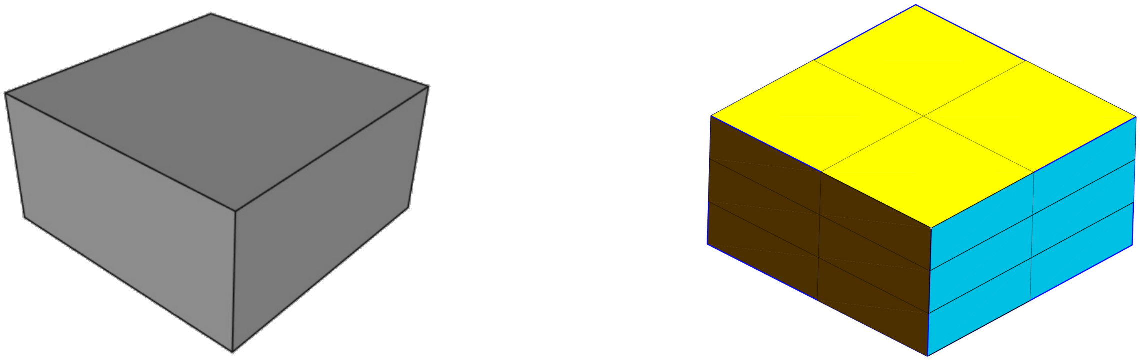


Figure 7.1: Simulation steps and software used

Figure 7.2b, that is impacted by a number of spheres. These spheres have two possible sets of properties. The properties grouped within a set are the density, Young's Modulus, Poisson's Ratio,



(a) cuboid geometry

(b) Discretized cuboid

Figure 7.2: Geometry

the Coefficient of Restitution, the coefficient of friction and the sphere's radius. The impacts of the granular matter on the surface of the cuboid produce on the surface of the latter a distribution of stresses, as shown in Figure 7.3. The right hand side shows a cross section of the cuboid, it can be observed that the cuboid is not a solid-cuboid as it was originally designed, it is a shell-cuboid. And also it is appreciated that the original hexahedron elements have been converted into triangular shell elements. However, the position of the outer nodes of both parts, solid-cuboid and shell-cuboid, remain the same.

In order to transfer the stress calculated by the DEM solver to the FEA, solver, it is first necessary to convert the superficial stress to concentrated loads at the nodes. The Equation 7.1 is used to retrieve the equivalent load acting on the centre of mass of the shell element, then it is distributed over the nodes of the element. This operation has to be repeated for all the elements of the cuboid over all the time-steps. This subject is thoroughly explained in the Section 7.2.

$$S = \frac{F}{A} \tag{7.1}$$

CalculiX (http://www.calculix.de/) is a free and open-source finite element analysis application. It uses a scripting format to input the problems, this one is similar to the commercial software Abaqus. Contrary to DEM, FEA provides information on how the stress on an element of the cuboid affects other elements. Figure 7.4 represents the stress distribution on the solid-cuboid at the same time step as in the previous figure. On this occasion, not only the elements affected by the impact of the granular matter are affected but also the surrounding elements.

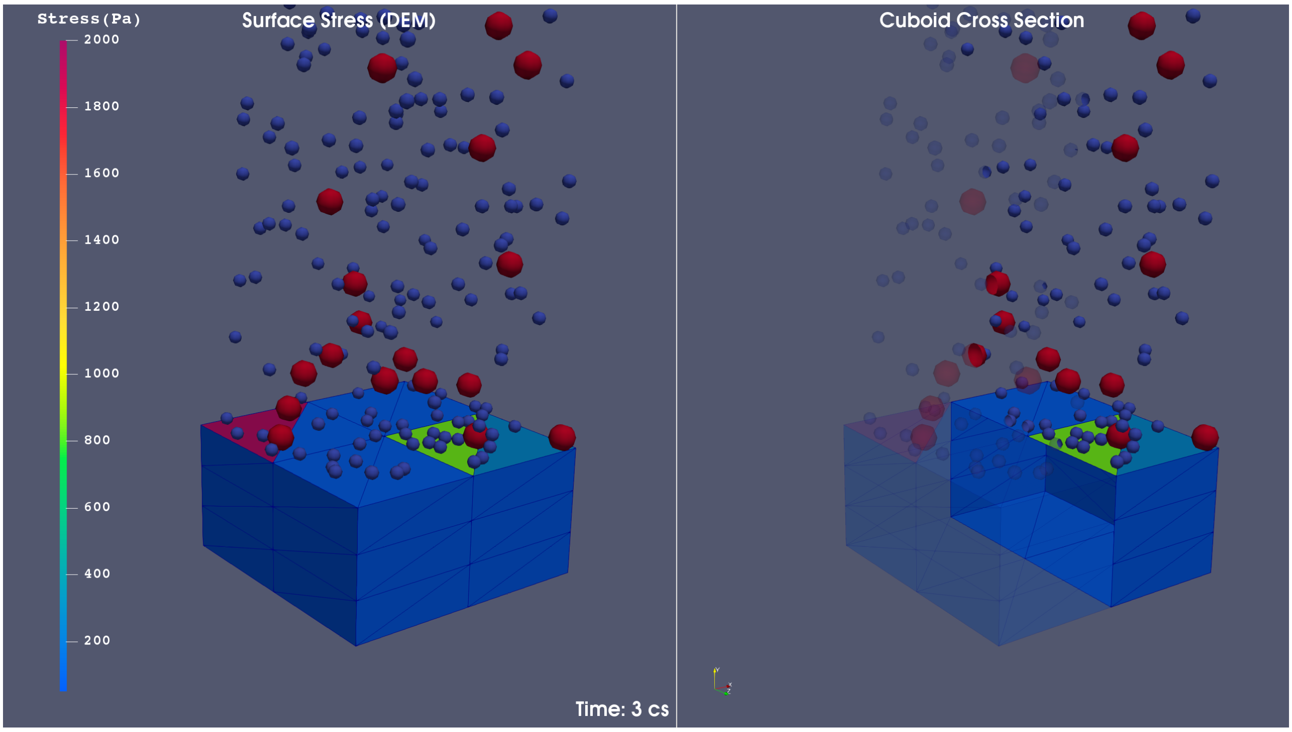


Figure 7.3: Stress on the surface of the shell-cuboid due to the impact of the granular matter

It is necessary, to validate the transference of the loads, to compare the results obtained from the finite element analysis with those obtained from the DEM simulation. Unfortunately, CalculiX GraphiX is limited and makes it a very tedious and unproductive task to compare these results. For this reason, the output has to be converted to VTK so ParaView can post-process all the data and thus efficiently compare the results. For this purpose, a CalculiX to Paraview converter was used: This converter was developed by Ihor Mirzov https://github.com/imirzov/ccx2paraview.

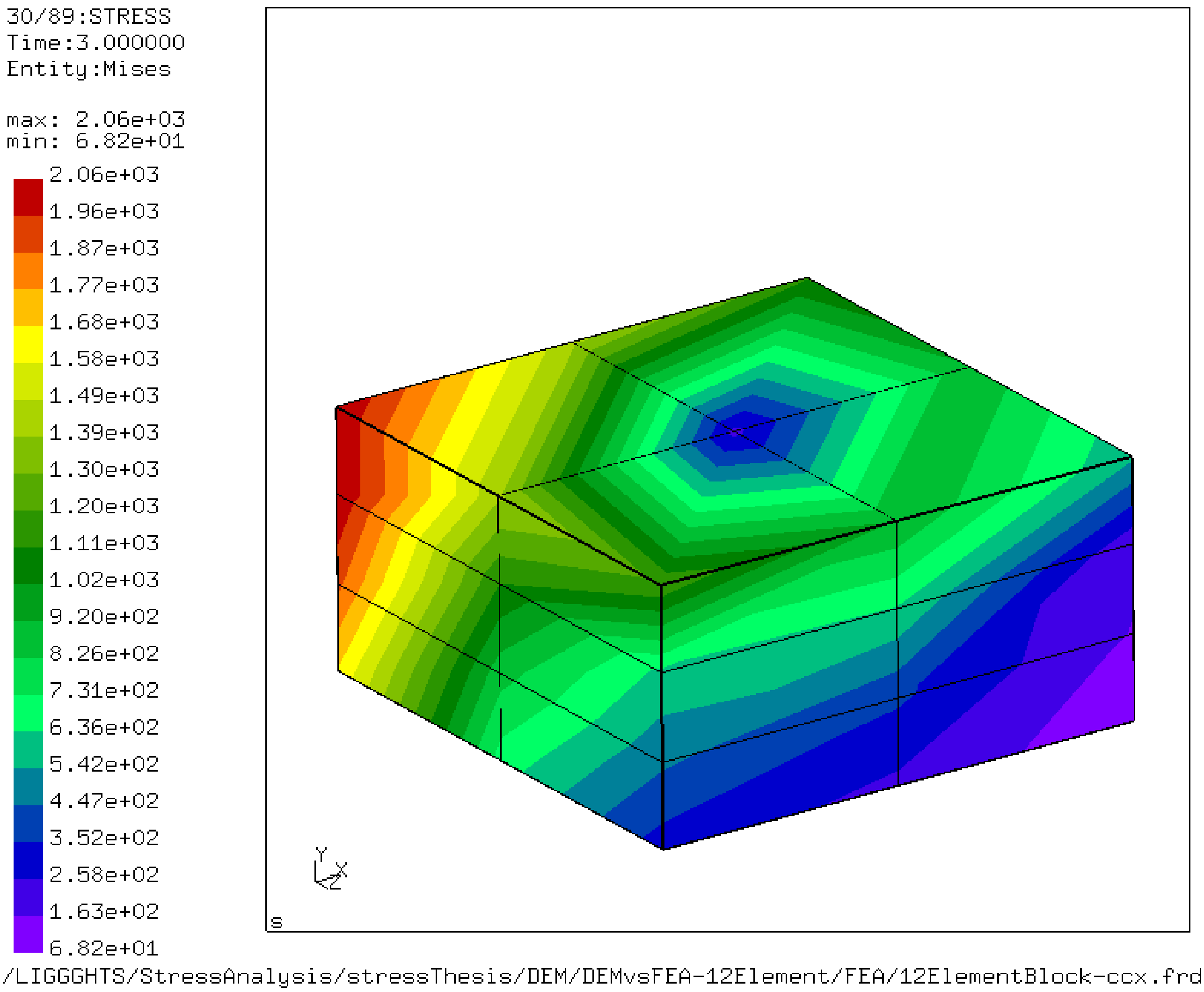


Figure 7.4: Stress distribution on the solid-cuboid due to the impact of the granular matter

Figure 7.5 depicts the stresses for the DEM and for FEA,; as can be observed the range of stresses is very similar and the concentration of stresses is located in the same areas for both cuboids, as it should be. Furthermore, the FEA, output delivers also the state of stresses below the surface. On the contrary, DEM only shows the stress in those elements that have been impacted by the granular matter. FEA, not only provides the stress but also the reactions at the boundary and the deformation of the elements.

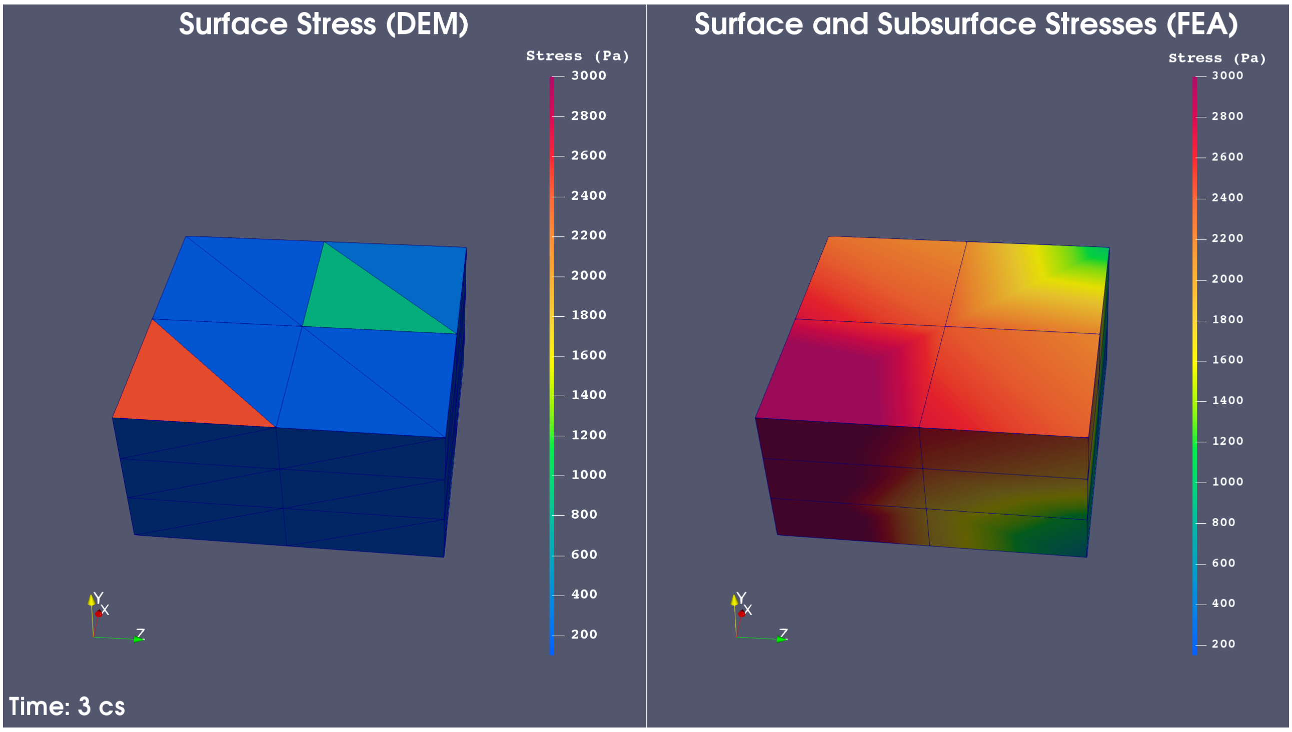
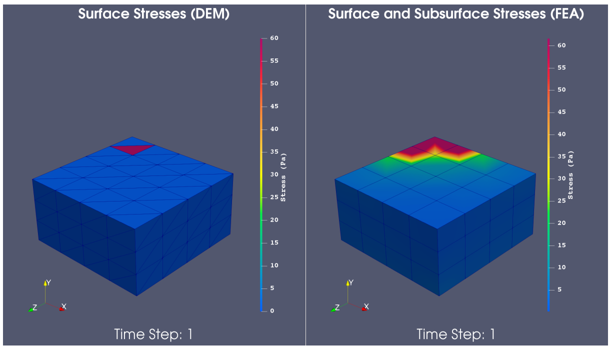
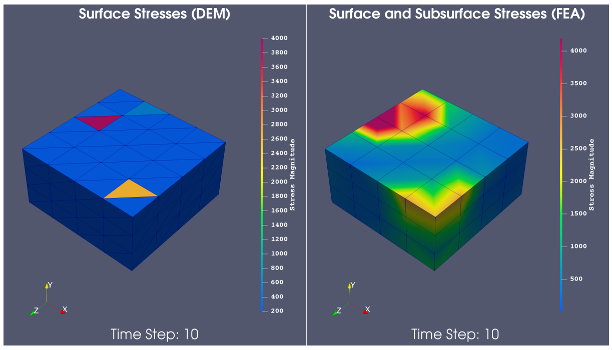


Figure 7.5: DEM stress distribution compared to FEA results

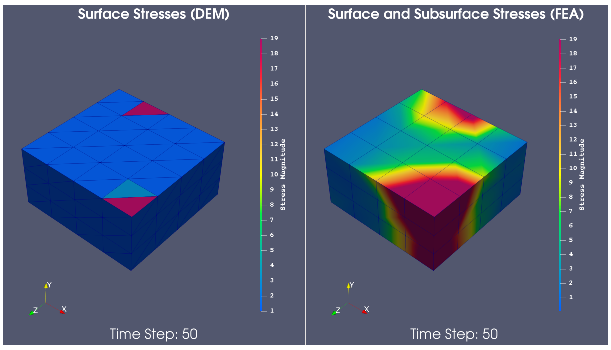
The same experiment was repeated for the same cuboid, only this time it was discretized using 48 hexahedral elements. Figure 7.6 show different time-steps of the same simulation. For this case it can also be appreciated how the stresses have the same range of magnitude and also that the location of the affected elements for the FE cuboid matches those from the DEM cuboid. Furthermore, it is observed that the finer the mesh the more accurate the results are, provided the mesh elements are not smaller than the contact area between the particle and the surface. Unfortunately, the higher the number of elements the more demanding the simulation becomes in terms of computational resources. It is critical to achieve an efficient ratio between the accuracy of the simulation and the computational resources needed to meet the output.



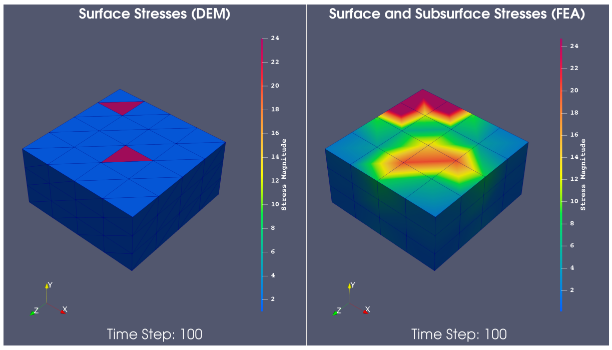
(a) Comparison at time-step 1



(b) Comparison at time-step 10



(c) Comparison at time-step 50



(d) Comparison at time-step 100

Figure 7.6: Comparison DEM v FEA forty-eight elements cuboid

7.1.1 Large deformation

LIGGGHTS considered the cuboids as infinitely rigid bodies. This fact does not affect the results of the two previous experiments because the values of the stresses are insignificant in comparison with the modulus of elasticity of the stainless steel. So the deformations of the specimens were negligible. On the contrary, to consider, on some occasions, the bodies as infinitely rigid could produce unrealistic results.

To study the behaviour of the system when there are large deformations a new experiment was designed. This consisted in a 30mm long, 2mm thick and 20mm wide stainless steel specimen. The left end was encastrated while the right end was free. A total of 2 Kg of granular matter was released 100mm above the free end. The Figure 7.7 shows the result of the experiment. The FEA reports an important displacement at the free end, over 4mm from its original position. However, DEM treats the specimen as an infinitely rigid body therefore the particles bouncing against the specimen keep impacting the surface as it was not deformed. This produces an unrealistic result. LIGGGHTS-Public does not support any type of deformation. All the specimens studied in this thesis can be considered as infinitely rigid without compromising the certainty of the simulation.

As a conclusion, for large deformations a different approach should be considered. The most adequate approach would be to develop a new code for LIGGGHTS and recompile it adding the new code. This new implementation should study the displacement of the nodes when a concentrated load is applied on them, and how this affects the rest of the nodes of the body and then store the new coordinates to redefine the position of the nodes of the body in the next time-step.

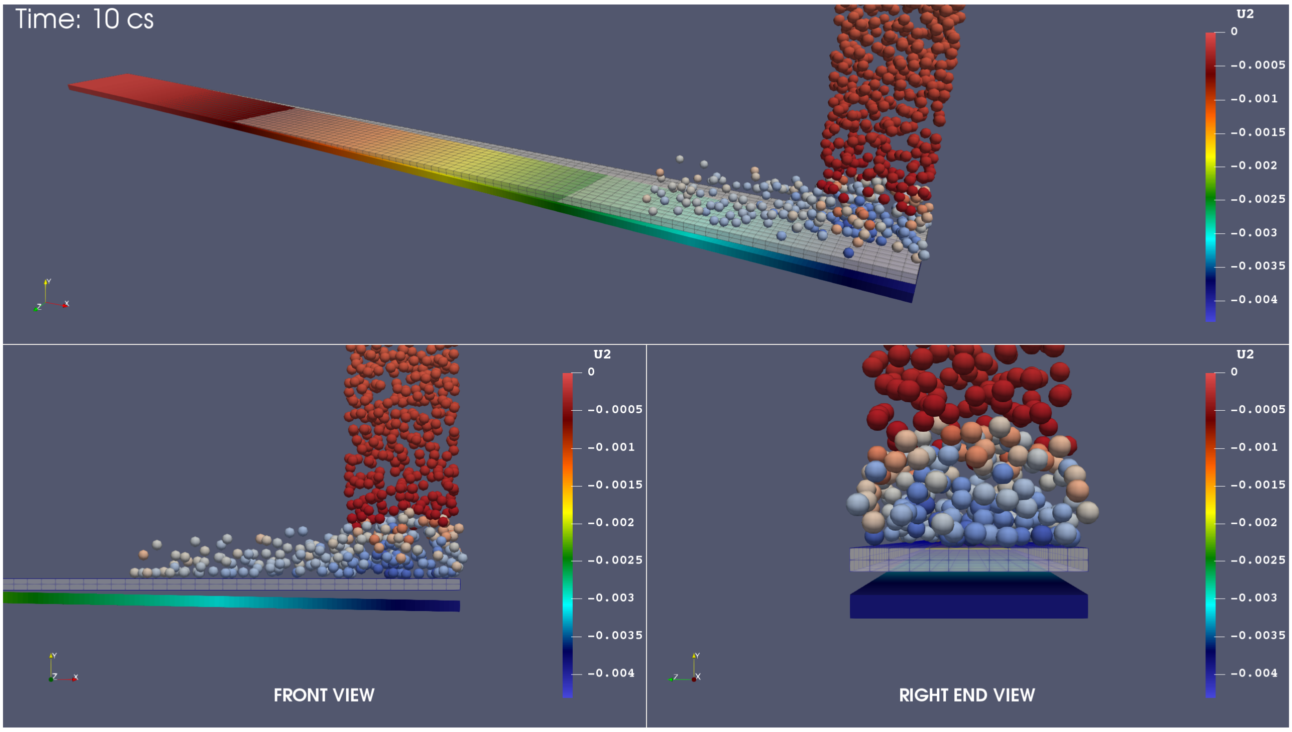


Figure 7.7: Deformation of the plate in the Y-direction

7.2 FEA Solver

CalculiX input system is via text script. Figures 7.8 and 7.9 show examples of an input script. The first part defines the discretized geometry. The second part defines the loading scenario, the boundary conditions and the output request.

As previously mentioned the geometry of the solid is discretized into smaller elements, in this case the type of element employed was C3D8, it consists of a 3D 8-node linear iso-parametric solid element. This means that every element has 8 nodes. Firstly, the IDs and rectangular coordinates of every node

```
*NODE, NSET=Nall
1,0.01,0.0,0.01
2,0.01,0.0,-0.01
3,-0.01,0.0,-0.01
4,-0.01,0.0,0.01
5,-0.01,0.01,0.01
6,0.01,0.01,0.01
7,-0.01,0.01,-0.01
8,0.01,0.01,-0.01
9,0.01,0.0,0.0
10,-0.0,0.0,-0.01
(...)
**
*ELEMENT, TYPE=C3D8,  ELSET=Volume1
69, 4, 12, 25, 11, 13, 26, 35, 28
70, 13, 26, 35, 28, 14, 27, 36, 29
71, 14, 27, 36, 29, 5, 17, 34, 20
72, 11, 25, 10, 3, 28, 35, 30, 18
73, 28, 35, 30, 18, 29, 36, 31, 19
74, 29, 36, 31, 19, 20, 34, 23, 7
75, 12, 1, 9, 25, 26, 15, 32, 35
76, 26, 15, 32, 35, 27, 16, 33, 36
77, 27, 16, 33, 36, 17, 6, 24, 34
78, 25, 9, 2, 10, 35, 32, 21, 30
79, 35, 32, 21, 30, 36, 33, 22, 31
80, 36, 33, 22, 31, 34, 24, 8, 23
*ELSET,  ELSET=EAll
Volume1
*NSET, NSET=boundary
25, 1, 11, 2, 3, 10, 9, 4, 12
*SOLID SECTION,  ELSET=EAll, MATERIAL=Steel
*AMPLITUDE, NAME = Amp1
0,0, 1,1
*MATERIAL,  NAME=Steel
*ELASTIC
2e+11, 0.3
**
*NSET,NSET=nodesSet20
20
```

The parameter NSET is used to assign the nodes to a node set

This option allows nodes and their coordinates to be defined

Defines ten nodes with node IDs and their rectangular coordinates

Defines the kind of element which is being defined

With this option elements are defined

Stores elements in set Volume1

Stores Volume1 in set EAll

This option is used to assign nodes to a node set

Assigns material properties to element sets

Indicates the start of a material definition

Defines the elastic properties of a material

Figure 7.8: Geometry definition in a CalculiX script

within the geometry are listed, then these nodes are grouped by elements. Lastly, the material properties of the part are defined.

The discretization process is very important in order to obtain realistic results. As a common rule when using brick elements as the C3D8, the length, width and height of the element need to be similar. An element with a dimension 5 times greater than any of the other two dimensions would produce inaccurate results. It is mandatory to keep consistency between dimensions. Another thing to take into consideration is the global size of the elements, usually the smaller they are the more realistic the results are. However, there is a limit on the reduction of the size of the element, any diminishing beyond this limit would not produce any difference in the results yet the computational resources would increase. A mesh convergence study would provide an idea where this limit might be.

The second part of the script defines the steps. Here the loading scenario is assigned to the previously defined geometry, also the boundary conditions are applied. The loads extracted from the DEM simulation are transferred to the nodes of the elements affected by the impacts of the granular matter, Section 7.3 on page 131 explains how the transference of loads between DEM and FEA, is done. This process is repeated for every time-step of the DEM simulation, so the number of steps in the FEA, simulation correspond to the number of time-steps obtained in the DEM simulation.

The boundary conditions are defined in such a manner that the nodes of the elements affected by the granular matter are not part of the set of nodes comprising the boundary conditions. For example, in the experiment depicted in Figure 7.7, the left end of the plate was fully fixed while the granular matter was impacting the elements at the free right end. In the case of the experiment shown in Figure 7.6 the upper face of the cuboid, in contact with the granular matter, was not constrained in any way, however the bottom of the part was fully fixed.

The last part of the step defines the output request for either selected nodal variables or selected element variables averaged at the nodal points in an frd file (extension .frd) for subsequent viewing by CalculiX GraphiX.

```
**--------------------------------------------------**
**                       Step 1                     **
**--------------------------------------------------**
** This step correspond to LIGGGHTS Step3 **
*STEP,  NAME=Step-3, NLGEOM=YES, INC=10000
 **
*STATIC
0.1, 1., 1e-05, 0.1
**--------------------------------------------------**
**                BOUNDARY CONDITIONS               **
**--------------------------------------------------**
*BOUNDARY
boundary, 1,6,0
**
*CLOAD, AMPLITUDE = Amp1, op=NEW
nodesSet5, 2, 0.000850
nodesSet34, 1, 0.000130
nodesSet34, 2, -0.012990
nodesSet34, 3, 0.000020
**
nodesSet20, 1, -0.000380
nodesSet20, 2, 0.039880
nodesSet20, 3, 0.000230
**
(...)
**-------------------------------------------------**
**                  OUTPUT REQUESTS                **
**-------------------------------------------------**
*RESTART, WRITE, FREQUENCY=0
**
*OUTPUT
*NODE OUTPUT, NSET=Nall
CF, RF, U
**
*EL FILE
U,S
*END STEP
```

*STEP describes the start of a new STEP, NLGEOM takes geometrically nonlinear effects into account,INC specifies the maximum number of increments in the step

This procedure is used to perform a static analysis

This option is used to prescribe boundary conditions

This option allows concentrated forces to be applied to any node in the model which is not fixed by a single or multiple point constraint

Removes all previous point load forces and applies a force with magnitude 0.039880 and amplitude Amp1 for degree of freedom two for nodes cointained in nodeSet20

It can be used to store results for a later restart

This option is used to print selected nodal variables in file

This option is used to save selected element variables averaged at the nodal points in a frd file

This option concludes the definition of a step

Figure 7.9: Step definition in a CalculiX script

7.3 DEM-FEM Coupling

The stress output file generated by LIGGGHTS is a vtk file. Figure 7.10 depicts at the right hand side a simple example of this type of file, at the left, the image produced by ParaView for the same file.

```
# vtk DataFile Version 3.0
Generated by LIGGGHTS
ASCII
DATASET POLYDATA
POINTS  6 float
-0.01 0 0.01 -0.01 0 0.005 -0.005 0 0.005
-0.01 0 0.01 -0.005 0 0.005 -0.005 0 0.01
POLYGONS 2 8
3 0 1 2
3 3 4 5
POINT_DATA 6
FIELD FieldData 4
normal_stress_average 1 6 double
2.0 2.0 2.0 4.0 4.0 4.0
shear_stress_average 1 6 double
1.0 1.0 1.0 3.0 3.0 3.0
stress 3 6 double
2.0 2.26 1.0  2.0 2.26 1.0  2.0 2.26 1.0
4.0 5.0 3.0 4.0 5.0 3.0 4.0 5.0 3.0
area 1 6 double
1.25e-05 1.25e-05 1.25e-05 1.25e-05 1.25e-05 1.25e-05
```

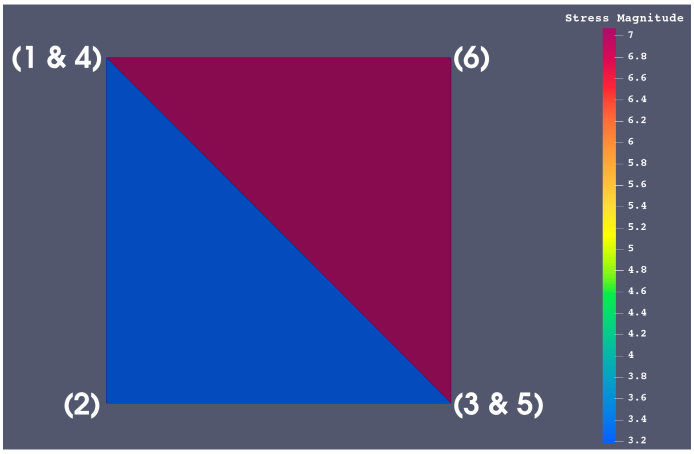


Figure 7.10: Text output and graphic output

The structure of this vtk file is not complicated. The first entry, by the name of `POINTS`, is the rectangular coordinates of the nodes defining the vertexes of the mesh's elements. For this particular case there are six sets of coordinates. The second entry `POLYGONS` refer to the elements. The two following lines define the type of element, in this case the number 3 means a triangular geometry, then the vertexes are pointed out. 0 indicates the first set of coordinates previously defined, 1 the second set and so on. As can be observed in the picture, although the two elements share two nodes, at file level they are different sets of coordinates. Thus the first and fourth sets contain identical coordinates. Exactly the same is observed for the third and fifth sets of coordinates. The `FIELD` entry contains the value of the stresses. These are divided into `normal_stress_average`, `shear_stress_average` and `stress`. The normal and shear are defined at every node. All the nodes belonging to the same polygon have the same value. The stress is also defined at every node however there are three values per node which represent the components of the stress vector. The last entry `area` assigns the area of the polygon to its vertexes.

Writing a CalculiX script to investigate this simple geometry and load distribution is not complicated. However, for more complex geometries with a greater number of elements the output file can easily reach hundreds of thousands of lines. Furthermore, every dump-step has its output file. For example, a 5 second simulation with a dumping ratio of 60 fps, LIGGGHTS would generate 300 output files. This amount of data makes it infeasible to manually write the script. To this end a Python code was developed, Appendix D. It first processes the discretized geometry generated by GMSH and extracts the relevant information: nodes, nodes' coordinates, elements, elements type, etcetera. This information is printed out following the structure of the geometry part of the script previously explained, Figure 7.8. Secondly, it scans LIGGGHTS output files, converts stresses into concentrated loads at the nodes of the polygons and labels all those nodes affected by the stresses. Lastly it transfers the concentrated loads to the geometry for each time-step, and appends this data to the geometry part.

7.4 Case Study I

This case study investigated the behaviour of the bulk when a part is inserted and fixed into it, paying special attention to the average velocity of the bulk. The stress distribution on the specimen was briefly examined. To this end a cylindrical specimen of a ϕ30mm base and 100mm length was simulated. The material employed for this specimen was stainless steel.

In order to ensure stability of the numerical method, a minimum time -step can be determined based on the Rayleigh time and Hertz time approaches. The time-step size for a granular system with no cohesion and dry media must not exceed 20% of the Rayleigh time and 10% of the Hertz time. This percentage depends also on the velocity at which the particles travel, the ratio of skin to the

Table 7.1: Case 1: Simulation details

No Simulation	Step	CPU	Time simulated per step (s)	Computing time (h:m:s)	KinEng (J)	Mass (kg)	Atoms	Bottom rotation	Specimen rotation	Specimen inclination	Total time
1	3	245 (7x5x7)	0.5	10:20:57	4.4664765	7.643192	368388	200	0	0	2
	4	245 (7x5x7)	0.75	14:15:29	4.3907676	7.6424451	368352	200	0	0	
	5	245 (7x5x7)	0.75	14:41:25	4.4248449	7.6424451	368352	200	0	0	
2	6	245 (7x5x7)	0.5	10:37:07	0.90546144	7.6424451	368352	60	0	0	2
	7	245 (7x5x7)	0.75	15:06:26	0.89145316	7.6424451	368352	60	0	0	
	8	245 (7x5x7)	0.75	14:32:50	0.88635412	7.6416981	368316	60	0	0	
3	9	245 (7x5x7)	0.5	08:46:35	4.485529	7.6409512	368280	200	0	30	2
	10	245 (7x5x7)	0.75	14:21:10	4.4579733	7.6409512	368280	200	0	30	
	11	245 (7x5x7)	0.75	14:29:08	4.3650634	7.6409512	368280	200	0	30	
4	12	245 (7x5x7)	0.75	18:59:30	0.84382956	7.6394574	368208	60	0	30	2.25
	13	245 (7x5x7)	0.75	16:28:52	0.85994313	7.6387105	368172	60	0	30	
	14	245 (7x5x7)	0.75	17:14:26	0.8683497	7.6364697	368064	60	0	30	

distance that particles can travel relative to each other in one time-step should be >1, otherwise some interactions may be missed or overlap energy may be generated artificially. As a common rule, the smaller the time-step size is the longer the simulation takes. It is important to determine a time-step size that meet the previous criteria without having to wait longer than necessary. The time-step size satisfying this norm was 2×10^{-6} seconds.

For this simulation the bowl was filled with 7.6 kg of small conical plastic common purpose media. Over 10,200 media particles were simulated. A total of 170 hours were needed to complete the whole simulation.

7.4.1 Location studies

Two different positions were studied. The first position is depicted in Figure 7.11, the axis of the cylinder was parallel to the y axis. In the second position, the axis of the cylinder was inclined 30 degrees with respect to the vertical direction, Figure 7.12.

A multi-script simulation was needed for each different position of the specimen. Every script corresponds to a simulation step. A total of fourteen steps were required for these four configurations. Table 7.1 displays these steps.

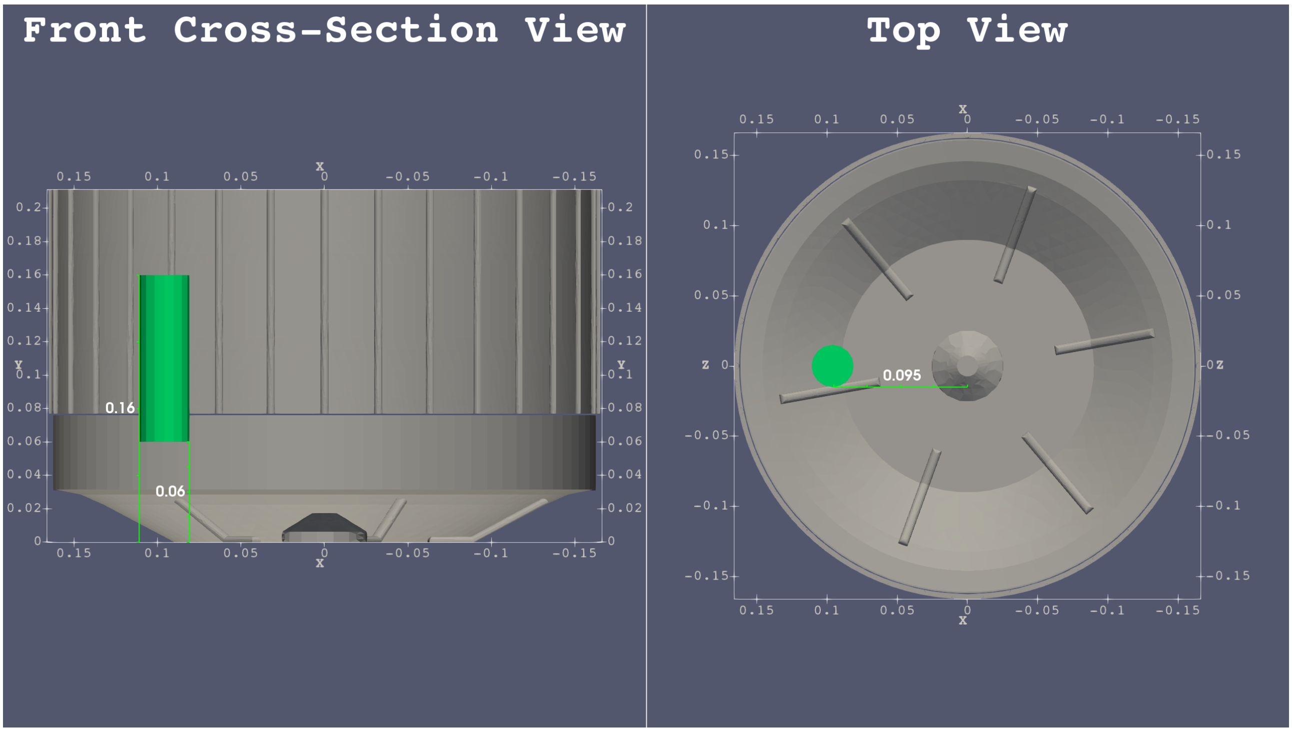


Figure 7.11: Configuration 1 (no inclination)

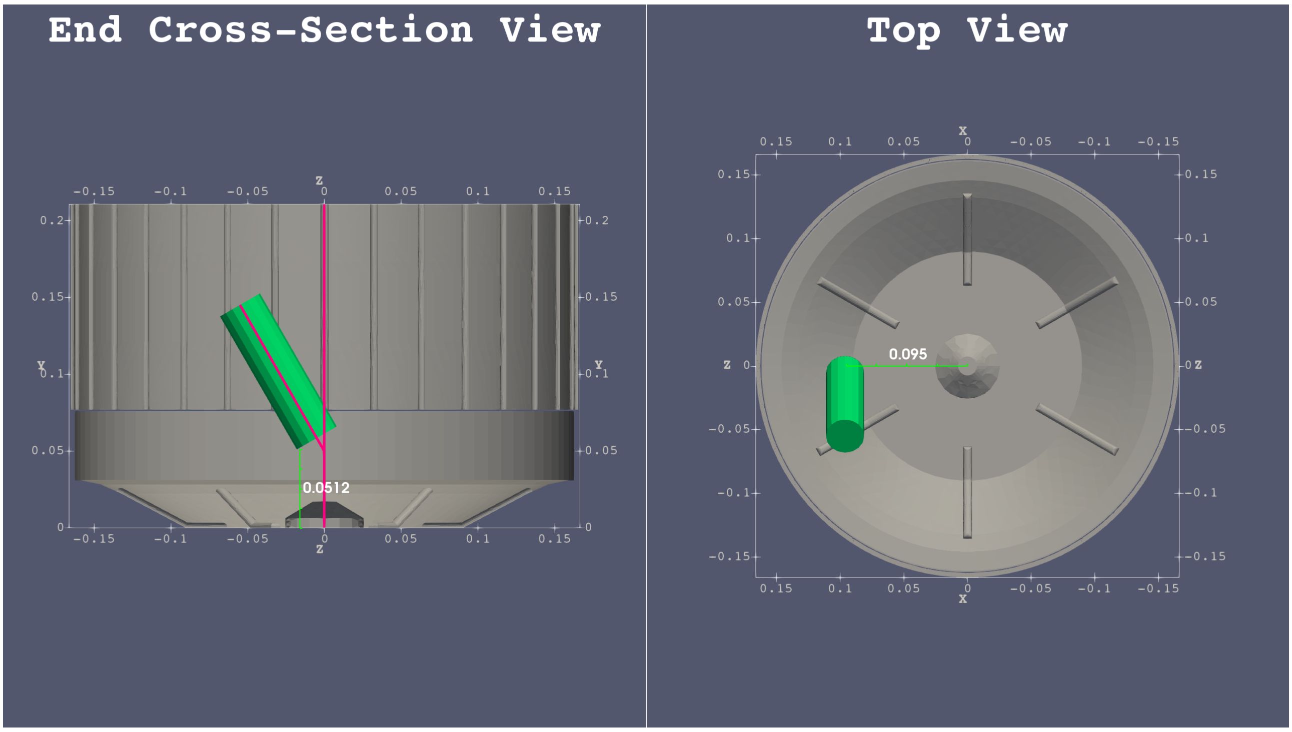


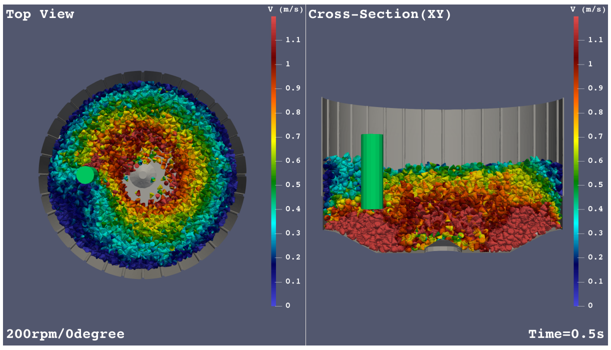
Figure 7.12: Configuration 2 (30 degrees inclination)

Figure 7.13 collects four snapshots of the velocity of the bulk in its steady-state for each of the configurations. In the first configuration the specimen is in its vertical position and the base rotates at 200rpm. In the second configuration the specimen remains in this position and the base spins at 60rpm. For the third configuration the axis of the cylinder is inclined 30 degrees with respect to its vertical position and the base rotates at 200rpm. The fourth configuration also has the axis inclined 30 degrees respect the vertical position and the base rotates at 60rpm. Primarily, the distribution of velocity within the bulk revealed few differences in its behaviour. It is clearly observed how the velocity at which the base rotates characterized the bulk activity. The linear velocity of the particles, in some areas of the bulk, reaches 1.2m/s for the first and the third configurations(200rpm). Furthermore, the vortex at the centre of the bulk is well-defined, although the specimen moves it respect its original position. For the second and fourth configuration (60rpm), it only goes up to 0.7m/s. Contrary to what was observed for the other configurations, the vortex is not fully defined. Besides, apparently the specimen has a deeper impact on the bulk's velocity than in the faster configurations. Finally, it is noticed how the particles' velocity at the top of the bulk is affected in different ways for the two angular velocities. The volume of particles stopped by the specimen is considerable larger for the slower configurations.

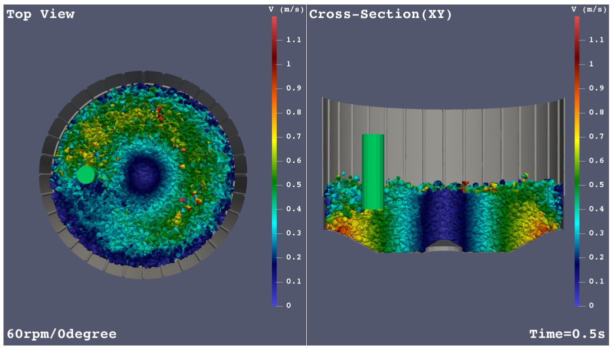
Graphs displayed in Figure 7.14 represent the evolution of the average velocity of the bulk along the simulation. It is confirmed that the bulk velocity is directly affected by the velocity at which the base rotates. It is also noticed that the bulk average velocity is lower for those configuration with inclined specimens, in opposition to the vertical configurations. Thus, the bulk average velocity for the first configuration is higher than the velocity in the configuration number three. In the same way, the second configuration has higher speed than the fourth.

Conversely, graphs displayed in Figure 7.19 represent the average of the Normal stresses (σ), the Shear stresses (τ) and the Magnitude of the stresses along the simulation. It is noticed that the Shear stress is at all times lower than the Normal stress. It also observed that for the configuration with inclined specimens the stresses are greater than in their vertical equivalent.

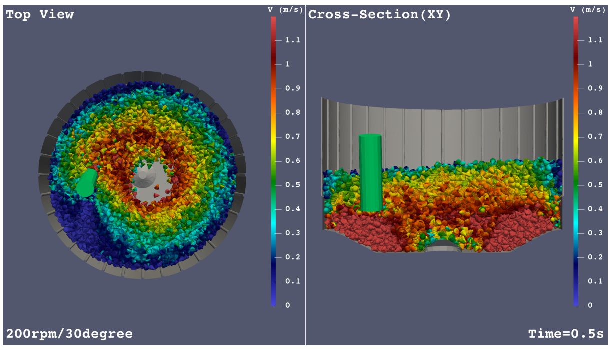
The side of the specimen with the higher concentration of stresses was the one facing the flow, specially the region at the bottom of the part. It is believed this is purely due to the influence of the particles with higher velocity. However, further studies will be held in order to shed light on this



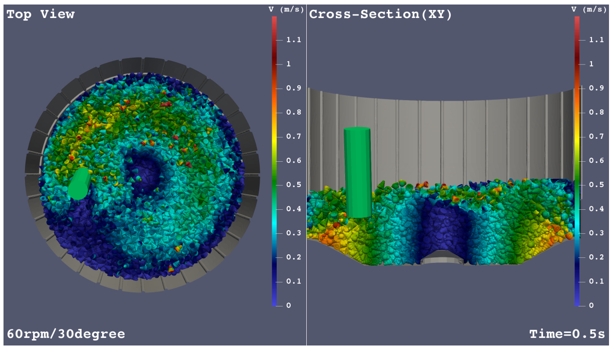
(a) 200rpm -- 0degrees



(b) 60rpm -- 0degrees

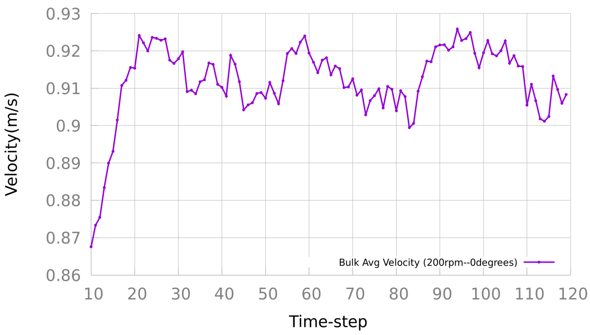


(c) 200rpm -- 30degrees

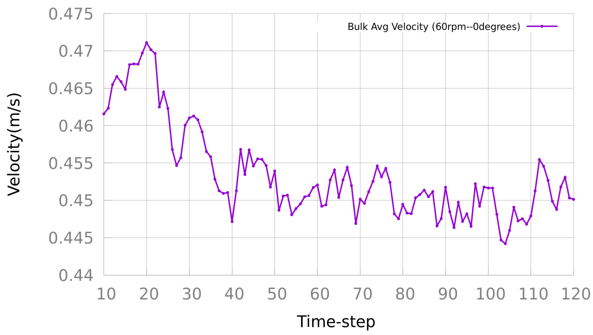


(d) 60rpm -- 30degrees

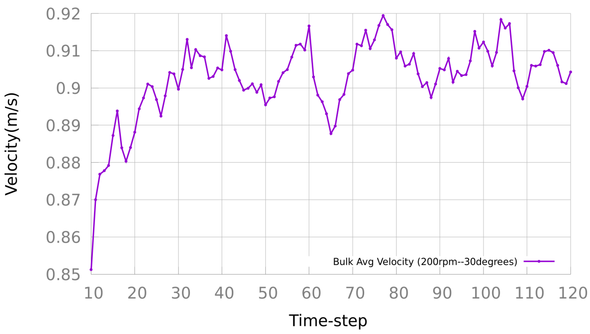
Figure 7.13: Velocity distribution for the different configurations



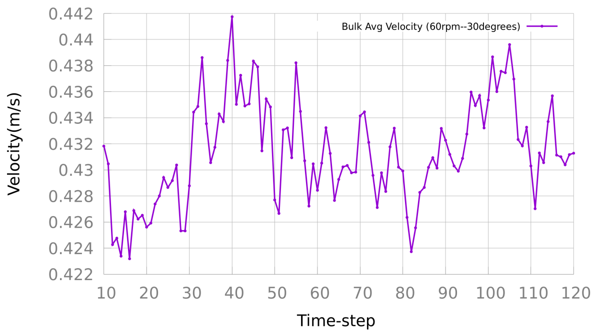
(a) 200rpm -- 0degrees



(b) 60rpm -- 0degrees

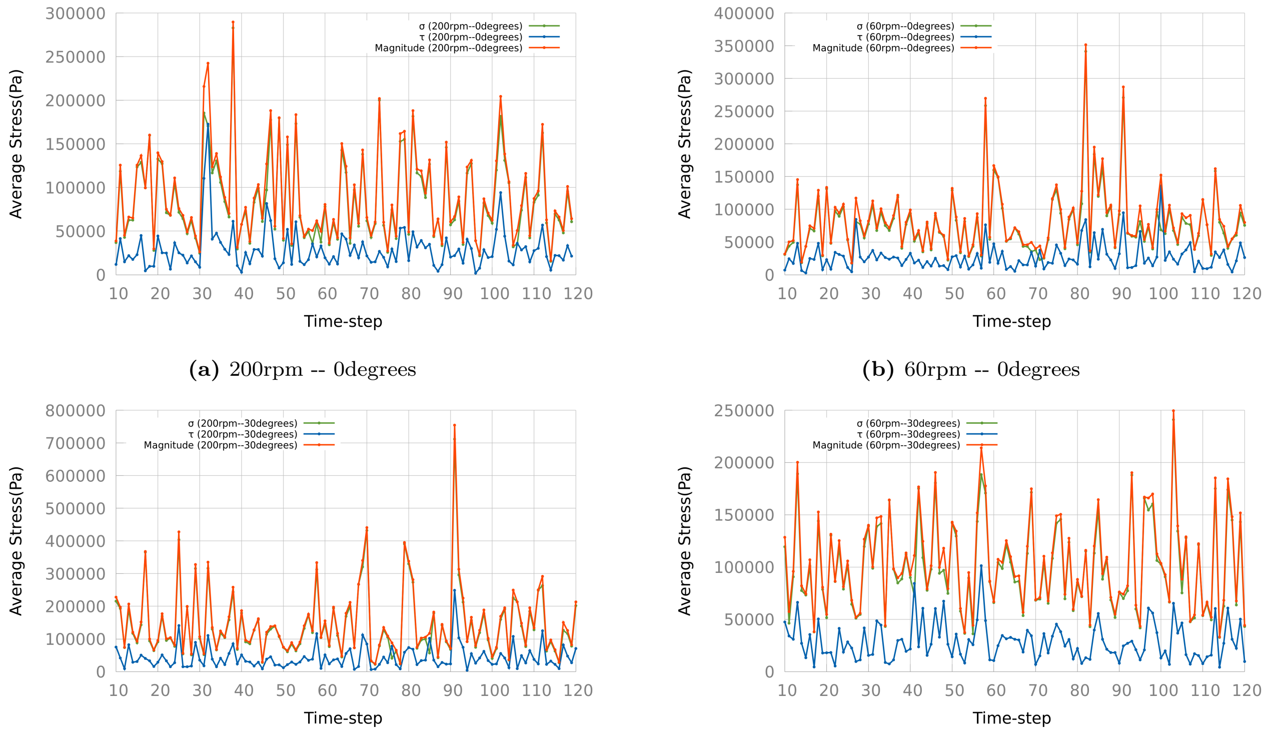


(c) 200rpm -- 30degrees



(d) 60rpm -- 30degrees

Figure 7.14: Velocity average in the bulk for the different configurations



(a) 200rpm -- 0degrees

(b) 60rpm -- 0degrees

(c) 200rpm -- 30degrees

(d) 60rpm -- 30degrees

Figure 7.15: Stress average on specimen for the different configurations

matter. The stress distribution found at the other side of the specimen were either null or negligible.

7.5 Case Study II

The second case study examines the effect of the media impact on the surface of a cylindrical specimen fixed within the media flow. The purpose of this case study was to identify which parts of the specimen had the higher stresses and also to understand how the superficial stresses affects the sub-surface areas. In order to gain a deeper insight three different positions were studied. DEM computes the force produced by the media impacts on the surface of the cylindrical specimen and then obtains the stress depending on the surface of the mesh element that has been impacted. DEM does not compute how the forces penetrate into the specimen, therefore theFEA is needed.

The specimen is a 100mm high and ϕ30mm stainless steel cylinder with a hole from top to bottom. The hole decreases the number of elements which reduces the computational time required for the simulation and also simplifies the meshing task. It was discretized using two types of hexahedral elements. A finer element was employed for the outer part of the specimen, the part in contact with

Item	Quantity
Nodes	7,813
Lines	624
Outer Hexahedra	2,160
Inner Hexahedra	4,320
Total Hexahedra	6,480

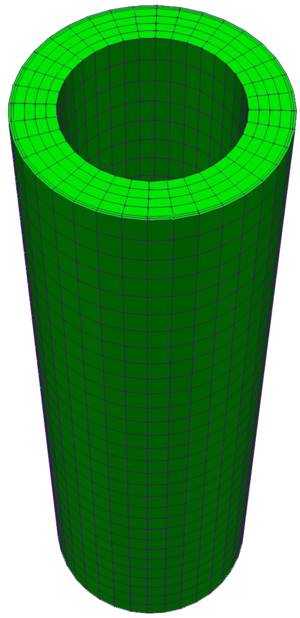


Figure 7.16: Specimen's mesh specifications

the media. A coarser element was used for the inner part of the cylinder. Figure 7.16 presents the mesh statistics. A total of 7,813 nodes and 6,480 elements were necessary to define the whole geometry. A single stress output file for this geometry contains over 35,000 lines. To manually transfer the concentrated loads at the nodes into the CalculiX script would be an almost impossible task.

In relation to the determination of the DEM time-step size and in order to satisfy the criteria explained in the previous case study. The time-step size assigned to this case study was 2×10^{-6} seconds.

The same HPC was used to run the simulations needed for this case study. A total of thirteen steps were required for the whole case study. Table 7.1 displays these steps.

Initially, three identical specimens were fixed within the bowl, Figure 7.17. The bowl was then filled using approximately 7.6 kg of general purpose plastic media. A restart file was written once the media was settled. This file worked as the starting point for all three simulations. In the first one the specimen closer to the centre of the bowl remained and the other two were discarded when the bottom started rotating. This simulation comprised steps 2 to 5. 245 CPUs were running for 87 hours to produce a 3.6 seconds simulation. The second simulation discarded the specimens at both ends while the one at the centre remained. Four more steps and 78 hours were needed to obtain 3.6 seconds. The last simulation comprised the steps 10 to 13, it studied the impacts on the specimen closer to the surrounding wall of the bowl while the other two were discarded. For this simulation the HPC was running for 79 hours and it also produced 3.6 seconds.

The simulations were configured to output the stresses acting on the specimen in sixty files per simulated second, a total of 216 files were produced for each of them. The stress information obtained from DEM was then converted into a CalculiX script using the Python in-house code. This process took 72 hours per simulation. This script was submitted to CalculiX and 24 hours later the FEA, analysis had finished. Then the results were converted into VTK files to be visualized in ParaView. On average, a week was needed for each simulation.

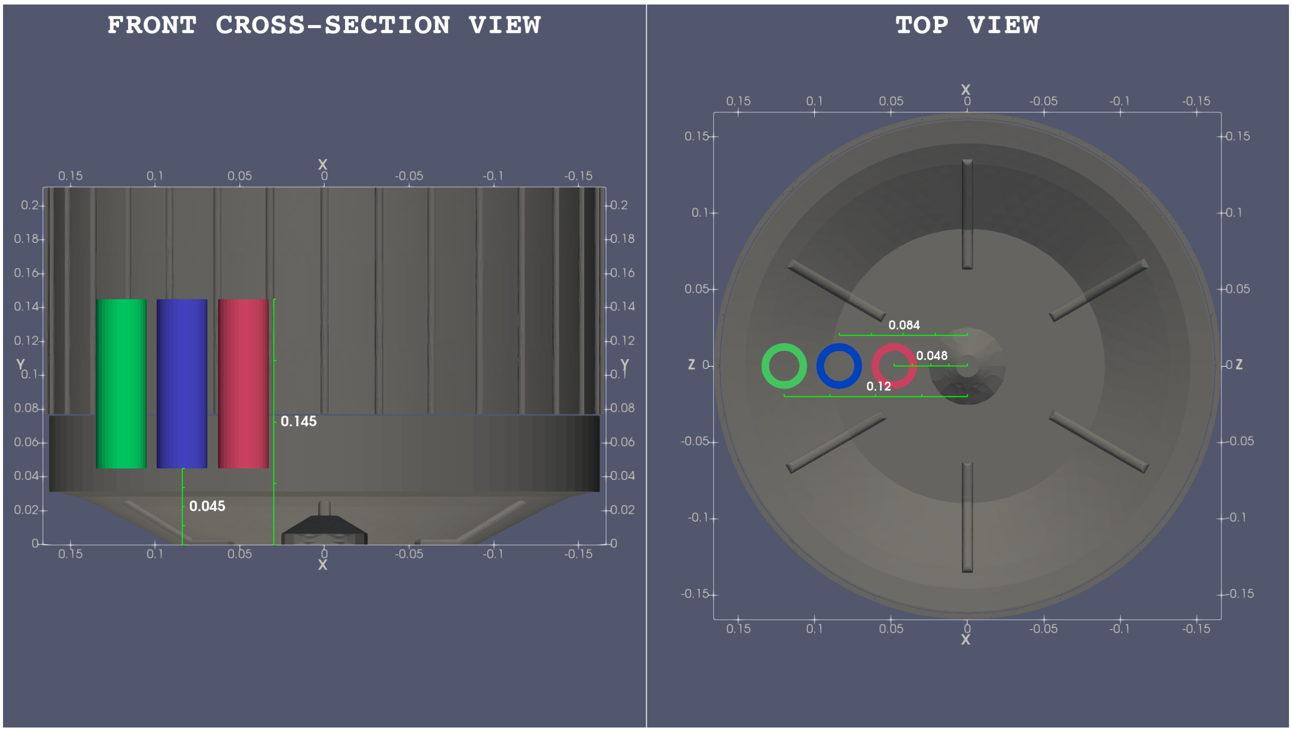


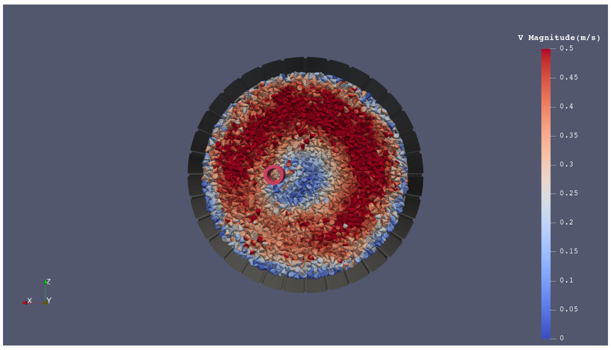
Figure 7.17: Specimen's positions

Table 7.2: Case 2: Simulation details

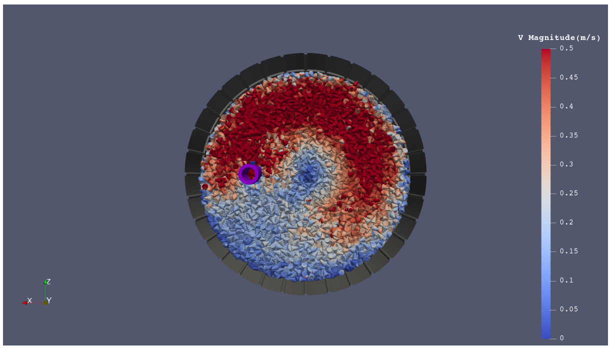
Step	Job ID	CPUs	Time simulated per step (s)	Computing time (h:m:s)	KinEng (J)	Mass (kg)	Media particles
1	run-3574	245 (7x5x7)	1.3	19:36:23	0.043697128	7.6304944	10,216
2	run-3582	245 (7x5x7)	0.05+0.90	23:09:18	9.86E-01	7.6290006	10,214
3	run-3612	245 (7x5x7)	0.9	21:42:11	1.0060745	7.6290006	10,214
4	run-3613	245 (7x5x7)	0.9	21:53:19	0.99325358	7.6275067	10,212
5	run-3615	245 (7x5x7)	0.9	21:12:11	1.0070355	7.6267598	10,211
6	run-3628	245 (7x5x7)	0.9	22:01:11	0.86345713	7.5834387	10,153
7	run-3629	245 (7x5x7)	0.9	18:30:35	0.81721116	7.5819449	10,151
8	run-3632	245 (7x5x7)	0.9	18:23:48	0.83617432	7.5819449	10,151
9	run-3634	245 (7x5x7)	0.9	20:00:14	0.85279846	7.580451	10,149
10	run-3637	245 (7x5x7)	0.9	21:15:17	0.80202778	7.6050993	10,182
11	run-3644	245 (7x5x7)	0.9	19:14:25	0.79799141	7.6050993	10,182
12	run-3642	245 (7x5x7)	0.9	20:01:18	0.79520822	7.6036054	10,180
13	run-3643	245 (7x5x7)	0.9	18:59:28	0.81873654	7.6013647	10,177

7.5.1 Kinematic energy of the bulk

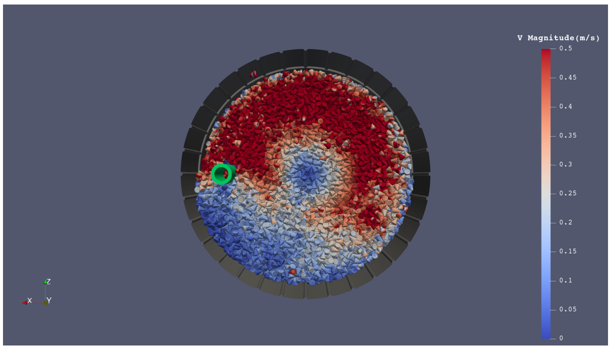
The first observation upon the simulation, before obtaining more detailed output, was the influence of the position of the specimen on the kinematic energy of the bulk. In normal operation, the base of the bowl rotates always at the same speed however the energy differs from one case to another. Figure 7.18d plots the values of the kinematic energy for the steps of the simulation. As can be observed, the energy for the first scenario, *position 1*, where the specimen is closer to the centre of the bowl has the highest values, the two other scenarios, *position 2* and *position 3*, have lower levels of energy. At the vicinity of the first position the velocity of the media is not as high as at the vicinity of the other two positions, furthermore the vortex at the centre of the bowl creates a void so there are parts of the specimen not in contact with the media. At the second and third positions the specimen is fully submerged in the bulk, as a consequence it receive more media impacts diminishing the overall velocity of the bulk affecting its global kinematic energy. Moreover the velocity of the media travelling at the bottom and close to the surrounding wall is higher than at any other part of the bowl. When the specimen is at this position it diminishes the velocity of those particles, this also affects the global



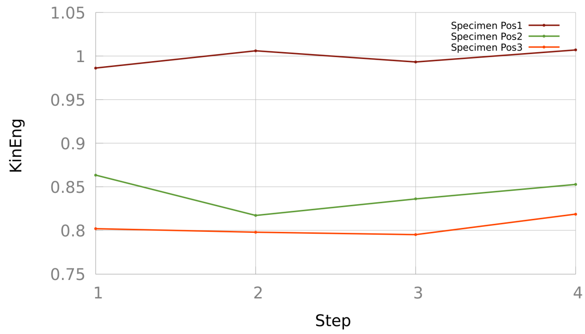
(a) Velocity at the top of the bulk(Position 1)



(b) Velocity at the top of the bulk(Position 2)



(c) Velocity at the top of the bulk(Position 3)



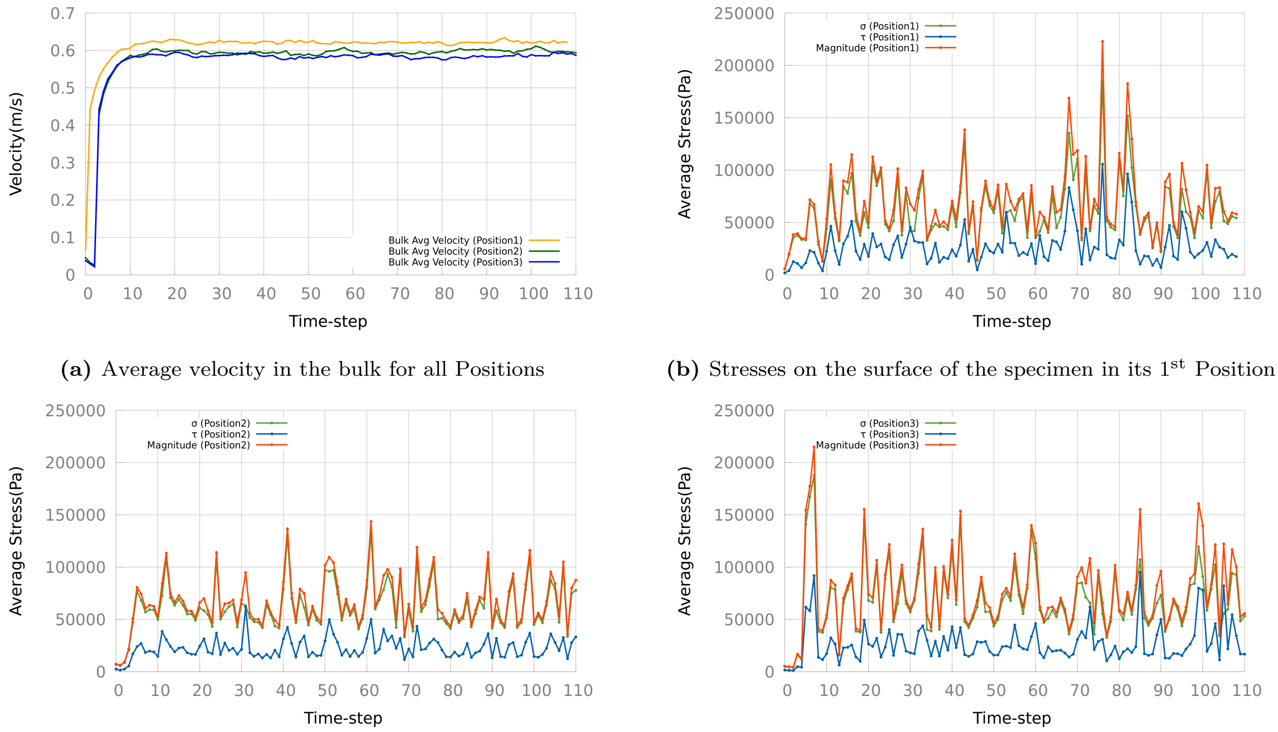
(d) Kinetic energy of the bulk

Figure 7.18: Comparison DEM v FEA forty-eight elements cuboid

energy. As a first hypothesis, the higher the kinetic energy is, the fewer *quality* impacts the specimen receives, therefore the stress should be lower in this case.

Figure 7.19a plots over time the average velocity of the bulk for each position of the specimen. Once the system reaches the steady state, this happens around time-step 10, the velocity remains reasonably constant over time. The standard deviations are significantly low, 0.053 for the first scenario, 0.016 and 0.017 for the second and third scenarios. It is confirmed that the first position of the specimen has a minor impact on the average velocity of the bulk compared to the other two positions.

Figures 7.19b, 7.19c & 7.19d depicts the evolution of the stresses on the surface of the specimen for its different position. It is observed that the stress in the first scenario is lower than for the other two scenarios; as was predicted at the beginning of this section. However, it can be affirmed that inclining the specimen has a more important influence on the stresses than the change in position study in this case.



(a) Average velocity in the bulk for all Positions

(b) Stresses on the surface of the specimen in its 1st Position

(c) Stresses on the surface of the specimen in its 2nd Position

(d) Stresses on the surface of the specimen in its 3rd Position

Figure 7.19: Stress average on specimen for the different positions

7.5.2 Impact studies

DEM graphic output reveals how the same media particle impacts the surface of the specimen and immediately after the impact it slides on the surface of the specimen. So the interaction of the media particle with the specimen has two steps. Impact step and sliding step, the latter comprises of usually more than one impact and its duration is longer than the first step which only includes the initial impact. An initial hypothesis for the study of this interaction was that the kinematic energy of the particle before the impact, is transferred at the impact moment to the specimen in the form of an impact force that produces a concrete stress distribution on the surface of the specimen. Due to the loss of the kinetic energy the impact forces generated by the subsequent impacts, during the sliding step, would not produce higher levels of stresses than at the moment of the impact. On the same premise, the normal component of the stress in the first step would be higher than its tangential component, whereas, in the sliding step the tangential component would be higher than the normal component.

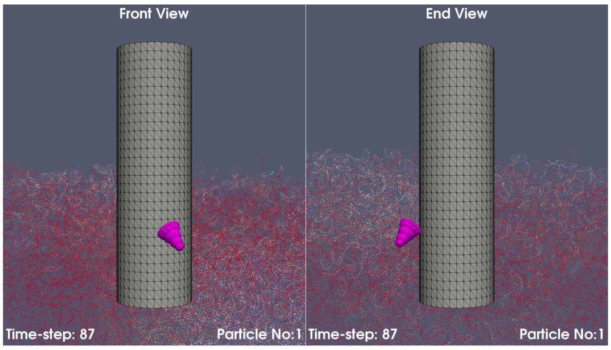
In order to shed light on the previous hypothesis, the interaction between four particles and the surface of the specimen in its first position was studied. It was revealed that the sliding step significantly changes for the different particles. It is not continuous, the particles are not always in contact with the surface of the specimen as it was first believed but it is a discontinuous phenomenon. During the sliding step there are periods of contact followed by periods of lack of contact.

Impact study: Particle 1

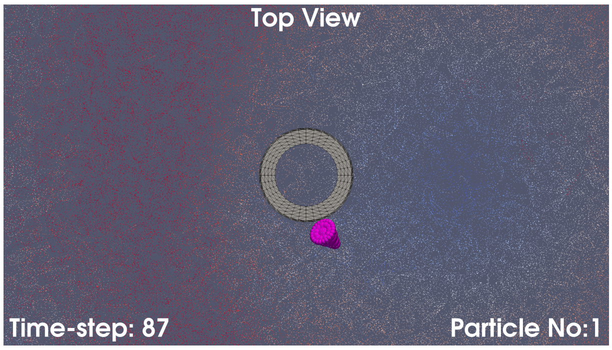
Figures 7.20a & 7.20b show Particle No:1 first impacting the specimen. The bulk is represented on these images as a coloured shade. The media rotates in a clockwise direction. The red colour represents high velocity in opposition to the light blue colour. The vortex is shown in the top view, at the right side of the specimen. The vortex is characterized by slower velocity and lower density of media. The higher media density is found at the left side of the specimen. The location of the particle when it first touches the specimen is approximately at the middle of the submerged part of the specimen. Its trajectory develops towards the vortex.

Figure 7.20c depicts the evolution of the stresses produced on the surface of the specimen by the particle's impacts along the interaction between the two of them. A zero value for the stress represents no contact between the particle and the specimen. The first non-zero value represents the first impact, the rest of non-zero values correspond to impacts within the sliding step. Contrary to what was initially thought, the normal stress (σ) remains at all times higher than the tangential stress (τ). It was also revealed that the first impact is not necessarily the one producing the highest impact force as can be observed in the graph. In this case the highest stress takes place at time-step 95 with a magnitude of 48×10^3 Pa, almost twice the stress produced at the first impact.

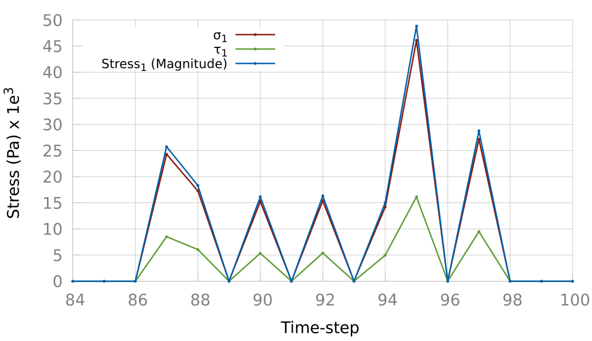
Figure 7.20d compares the relation between the velocity of the particle at the contact point and the impact force with the stress produced on the specimen. As predicted, the velocity drops by over 80% at the first impact and the magnitude of the force at that instant is high. In opposition to what was expected, there are another two occasions where the magnitude of the force is greater than at the impact, at time-steps 95 and 97.



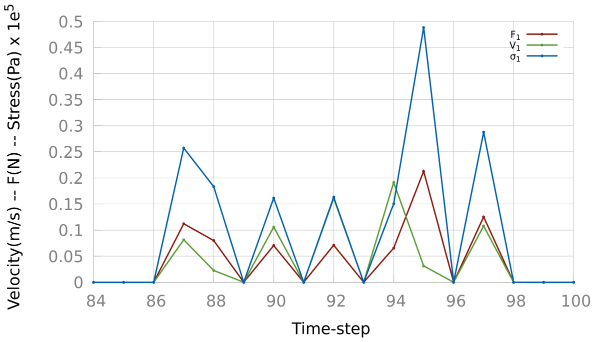
(a) 1st Impact (Front & End views)



(b) 1st Impact (Top view)



(c) Stress evolution on specimen (Position 1)



(d) Velocity & Force evolution on particle (Position 1)

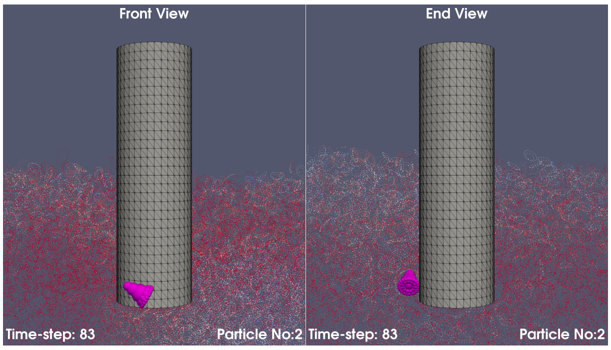
Figure 7.20: 1st Particle impact study

Impact study: Particle 2

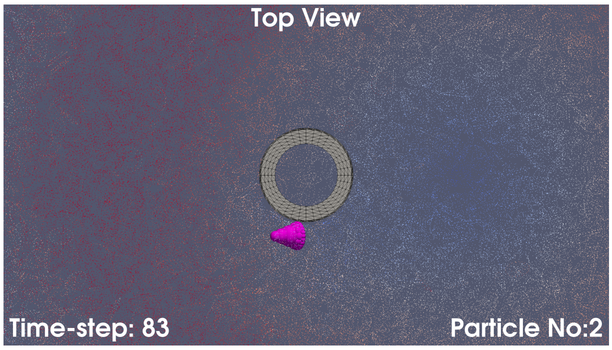
Figures 7.21a & 7.21b show Particle No:2. It impacts the cylinder in its lowest part where it remains during the interaction. In this case, its trajectory develops towards the higher density area. Once the interaction breaks the vortex starts drawing the particle to its centre.

It must be pointed out, Figure 7.21c, that the magnitude of the highest stress produced by this particle is over 120×10^3 Pa, three times greater than those produced by Particle No:1. In contrast with this particle the first impact produces the highest stress.

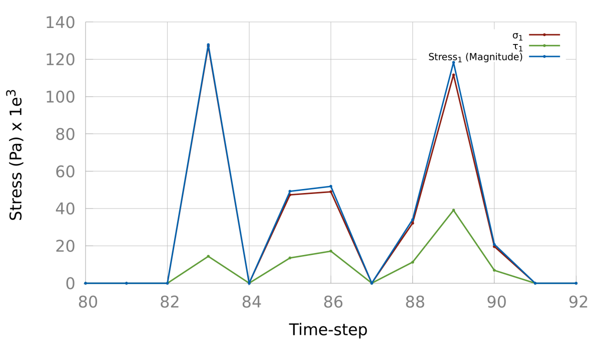
Contrary to what was observed in the first case, the velocity is low at the first impact and it does not drop as much.



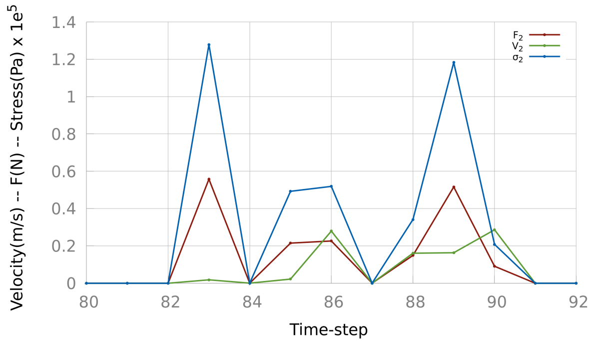
(a) 2^{nd} Impact (Front & End views)



(b) 2^{nd} Impact (Top View)



(c) Stress evolution on specimen (Position 1)



(d) Velocity & Force evolution on particle (Position 1)

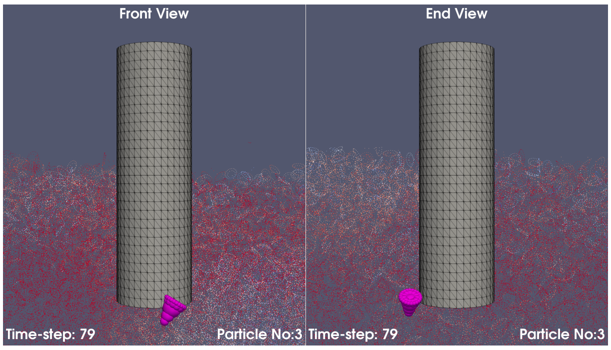
Figure 7.21: 2^{nd} Particle impact study

Impact study: Particle 3

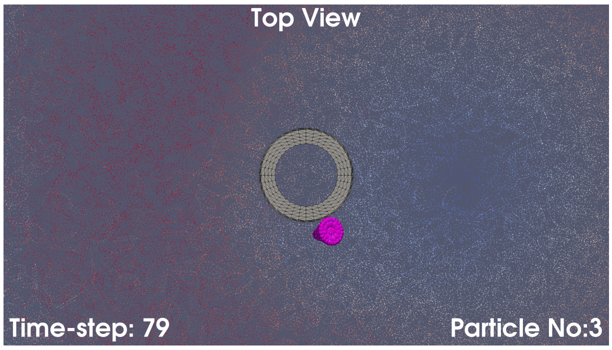
Particle No:3 was the more efficient among them. The values of the stresses produced by the impact forces, Figure 7.22c, were higher than in other cases and the length of the sliding step was considerable longer than the rest. This particle first impacted the specimen at its lower part, then the particles below pushed it towards the surface of the bulk.

At the time of the impact the velocity is highest, however, in this occasion the first impact produces the lowest stress on the surface of the specimen. Consecutively, the velocity of this particle drops drastically after it. The second impact produces the highest stress even when the velocity is lowest.

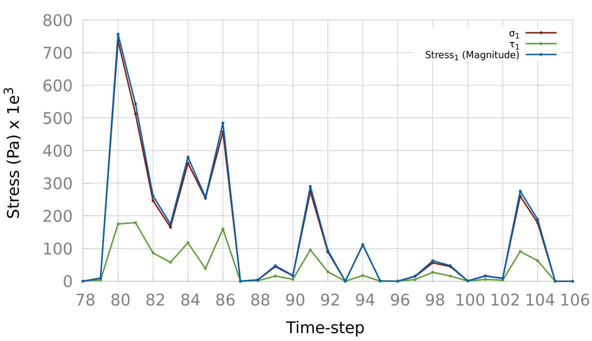
It is clearly observed that the stresses depend solely on the force, as all the mesh elements impacted by the media particle have the same surface. Figure 7.22d shows how the Force has the same pattern as the stress.



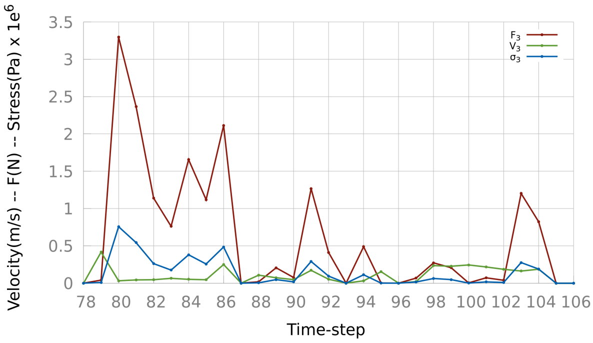
(a) 3rd Impact (Front & End views)



(b) 3rd Impact (Top View))



(c) Stress evolution on specimen (Position 1)



(d) Velocity & Force evolution on particle (Position 1)

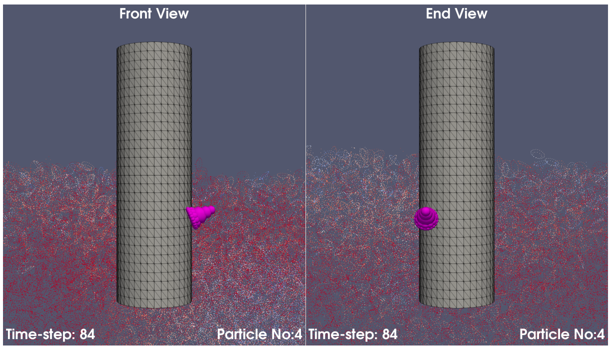
Figure 7.22: 3rd Particle impact study

Impact study: Particle 4

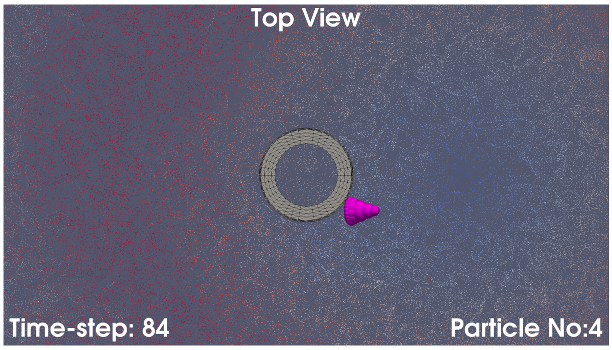
The position of the particle within the bulk determines the duration of interaction between particle and specimen and also the magnitude of the impact forces. In the case of Particle No:4, Figure 7.23, with only two impacts, the particle was between the vortex at the centre of the bowl and the specimen. The vortex drags the particle away from the specimen producing weaker impacts and shorter sliding steps.

As was observed in the previous cases, the normal stress remains above the tangential stress for all the impacts. For the studied cases, the location of the particles does not affect this feature.

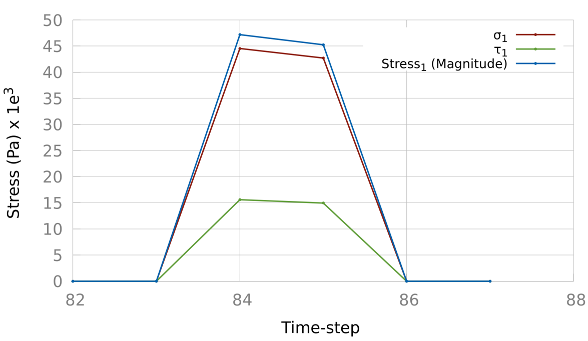
In Figure 7.23d is observed how not only the force shares the stress's pattern, as was previously mentioned, but also the velocity has a very similar pattern. This phenomenon has been noticed before in the study of the first particle, in that case the behaviour of the velocity from the time-steps 86 to 93 keeps close similarity with the one recognized here, although for the last steps that analogy is lost.



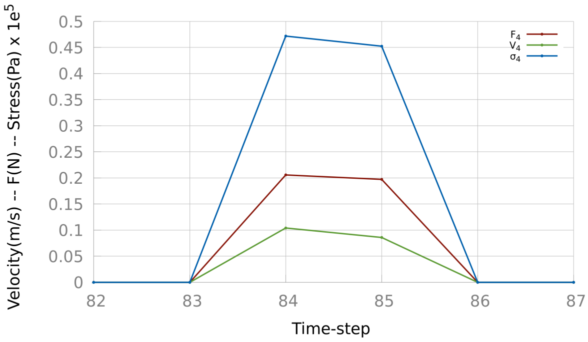
(a) 4^{th} Impact (Front & End Views)



(b) 4^{th} Impact (Top View)



(c) Stress evolution on specimen (Position 1)



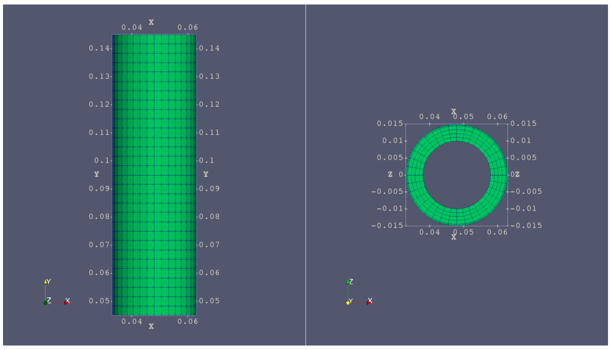
(d) Velocity & Force evolution on particle (Position 1)

Figure 7.23: 4^{th} Particle impact study

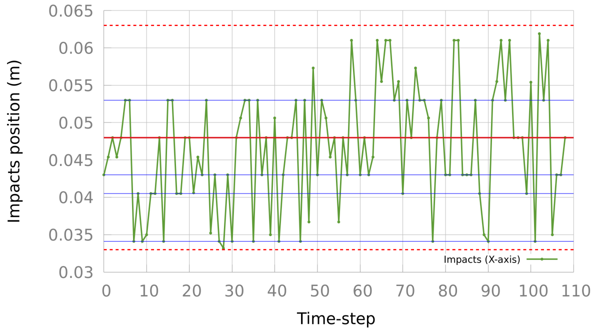
As a conclusion, the kinematic energy of the bulk directly affects the magnitude of the impact force, however the velocity of a single particle impacting the surface is not as determined as it was thought at first, not always the fastest particles produce the highest stresses on the specimen. As was found on the impact study cases. The stresses on the specimen depend on the impact force. This force can be produced by the travelling velocity of the particle, as in the case of Particle No:4, although most of the times it depends on the impacts it suffers from other media particles. This explains why when the particle in contact with the specimen travels through the denser zone of the bulk, the stresses on the surface of the specimen are greater even when the particle's velocity is zero or close to it.

Impact distribution study: Position 1

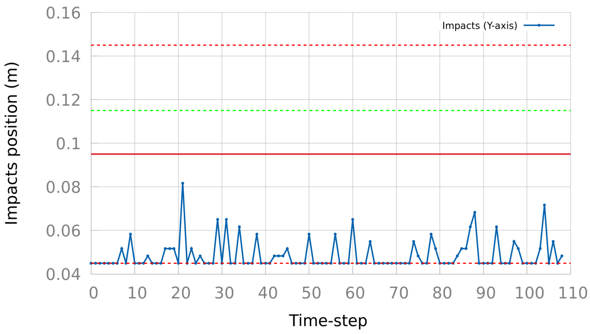
Figure 7.24 displays the distribution of impacts which produced the highest stress at every time-step. The Graph 7.24b plots, for each time step, the position where the highest impact along the x-direction occurred. The solid red line across the graph represents the imaginary line dividing the centre of the specimen along the x-direction. All the points above this line were located at the right side of the specimen, closer to the vortex. The points below this line were at the left side where the media



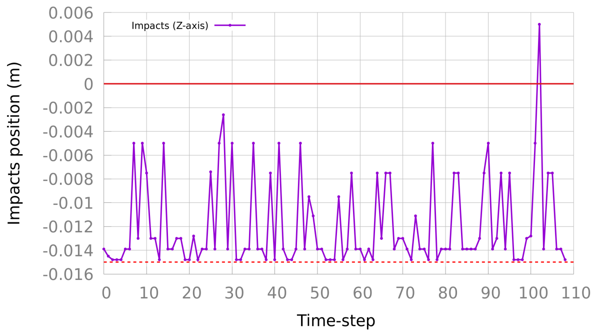
(a) Specimen's Front & Top Views



(b) Impacts' distribution along x-axis



(c) Impacts' distribution along y-axis



(d) Impacts' distribution along z-axis

Figure 7.24: Impact distribution

density is higher. The dashed lines represent the limits of the specimen in both directions. It can be observed that the highest concentration of impacts is located at the left hand side. Furthermore, there is a homogeneous distribution of impacts, the thin blue lines highlight the coordinates that are more frequently impacted. Graph 7.24c shows the impact distribution in the y-direction, the solid red line represents the centre of the specimen whereas the dashed red lines represents the upper and lower limits of it. The green dashed line indicates the surface of the media, above this line no media particle impacts the specimen. It is noticed that the impacts producing the highest stresses are always located at the bottom, quite consistently on the lower edge and rarely further than a half of the distance covered by the media. Graph 7.24d depicts the impacts with respect to the z-direction. The solid red line represents the centre of the specimen, the dashed red line represents the specimen's frontal limit. All points contained between these two lines are located within the half of the specimen facing the media flow. It is observed that the majority of the impacts happen in this half, mostly within three quarters of the front line.

7.5.3 Surface and sub-surface stresses

The loads applied to the specimen's surface were obtained from DEM, and then transferred to the FE model as concentrated loads acting on the nodes of the mesh elements affected by the particles' impacts. Therefore there is a stress distribution on the surface of the specimen and also inside it. The Finite Element Method is not valid to shed light on the stress distribution in the vicinity of the impact, however it is perfectly reliable to clarify the displacement of the nodes due to the action of the concentrated load. Saint-Venant's Principle[76] clarifies this limitation of the FEM,:

> "If the forces acting on a small portion of the surface of an elastic body are replaced by another statically equivalent system of forces acting on the same portion of the surface, this redistribution of loading produces substantial changes in the stresses locally, but has a negligible effect on the stresses at distances which are large in comparison with the linear dimensions of the surface on which the forces are changed."

In the light of what has been revealed in the previous section and to simplify the study of the distribution of stresses, the specimen has been divided into two halves. The uppermost part was discarded as it does not receive an important number of media impacts. Following Saint-Venant's Principle, to learn the stress distribution beyond the surface is key identifying the distances at which the equivalent system of forces have a negligible effect on it. To this end the displacement of the nodes on a specimen's slice were studied. Figure 7.25 depicts at its right hand side the studied slice contained in the plane zy. The vertical white line represents the axis of the specimen. On the left hand side, a zoom into the affected area is displayed. It is observed that the greater displacement of the node is in the order of 1.2×10^{-9}m. Therefore, the highlighted distance of 0.5mm towards the axis of the specimen represents a distance over 400,000 times greater than the node's displacement. Therefore, this study accepts as valid those stress distributions at more than 0.5mm from the specimen's surface.

Likewise, the lowermost half was divided into four cylindrical slices. Figure 7.26 shows the specimen's top view, at the right hand side. In favour of clarity, the specimen has been virtually divided into the *western half*, the side at the left of the red line and the *eastern half* at the other side of the line. The media flow reaches the specimen from the bottom, hence the eastern side is closer to the vortex. The right side of the picture displays the different slices. As can be observed all the slices are concentric to the specimen's axis. The first slice comprises the outermost surface of the specimen, in direct contact with the media. The following slices are separated 5mm from its antecedent. Being

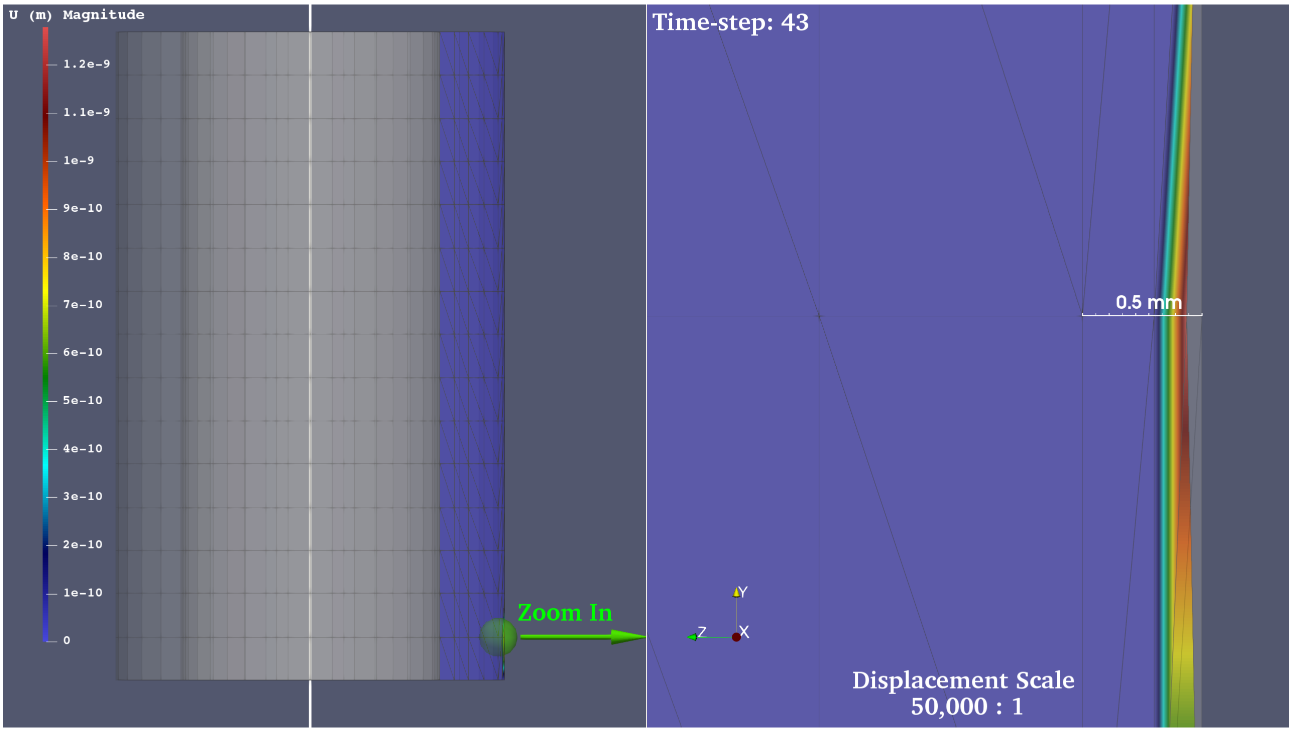


Figure 7.25: Node displacement

the last one at 1.5mm from the first slice.

The FEM, provides the whole stress tensor for each of the nodes of the mesh, as well as the Von Mises stresses and the Principal stresses. The stress tensor by itself does not explain the tangential and normal stress state of the specimen for each of its points. To this end, the stress vector at each point must be calculated. According to the *Cauchy's stress theorem*, the stress vector at a point that belongs to a known surface can be found multiplying the stress tensor by the unit vector normal to the surface at that point. Equation 7.2 shows the mathematical expression of Cauchy's Theorem.

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \begin{Bmatrix} l \\ m \\ n \end{Bmatrix} \tag{7.2}$$

Cauchy's theorem allows one to know the stress vector for an arbitrary orientation plane. The stress vector is attached to a specific point of the geometry and informs one of the pressures acting on it. This vector is not necessarily perpendicular to the surface. Thus, the scalar product between the stress vector and the unit vector normal to the surface has to be calculated in order to find the pressure acting perpendicularly to the surface at that specific point. The same procedure has to be

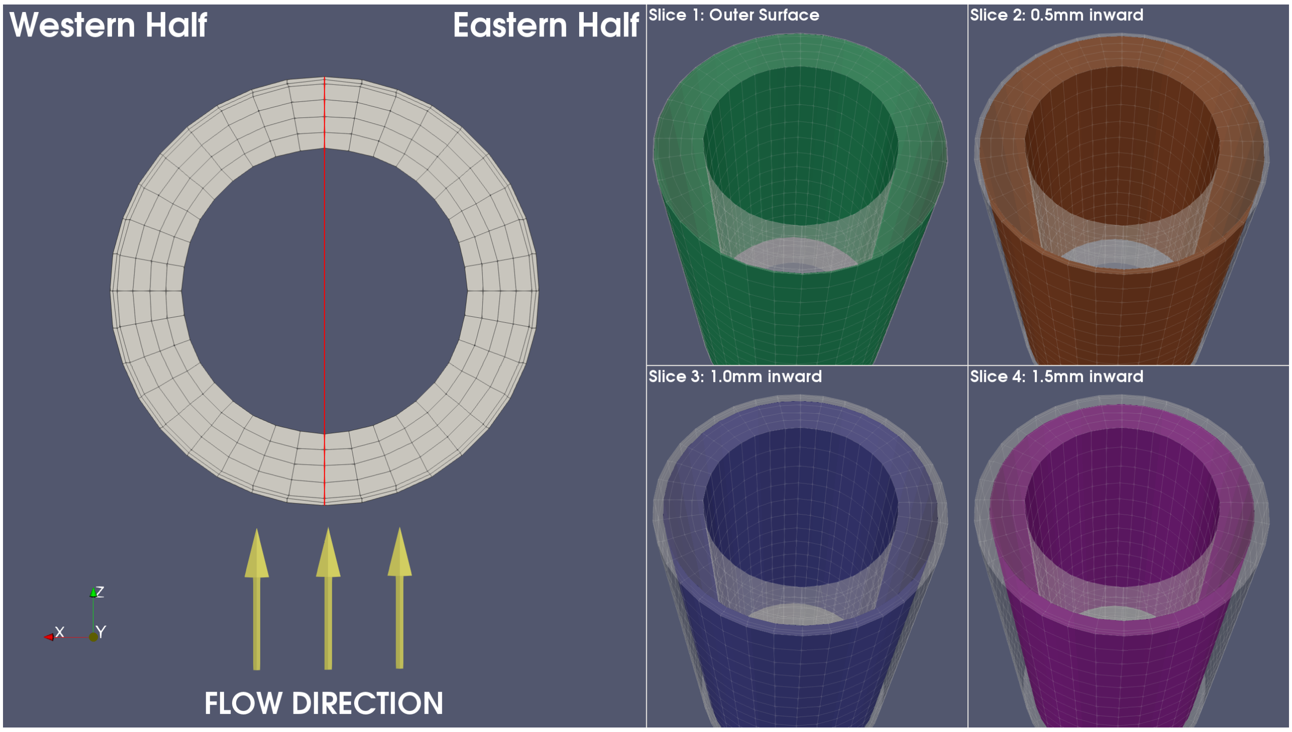


Figure 7.26: Specimen's slices

followed to find the pressure acting tangentially to the surface.

Figure 7.27 shows a section of the cylinder created by an xz-plane parallel to its axis. A specific point, (x_0, z_0), is selected. The blue dashed line symbolizes the tangential plane to the contour at that point. The green vector represents the normal unit vector in its positive direction. The tangential plane contains an infinite number of vectors, all of them tangent to the specimen surface at that point. The orange vector represents the tangential unit vector contained in the xz-plane, the anticlockwise direction is considered as positive.

Equation 7.3 shows the mathematical expression required to obtain the coordinates of the normal unit vector.

$$\begin{bmatrix} u_n \end{bmatrix} = \begin{Bmatrix} u_x \\ u_y \\ u_z \end{Bmatrix} = \begin{Bmatrix} C_x - (1+r)cos(\alpha) \\ 0 \\ (1+r)sin(\alpha) \end{Bmatrix} \tag{7.3}$$

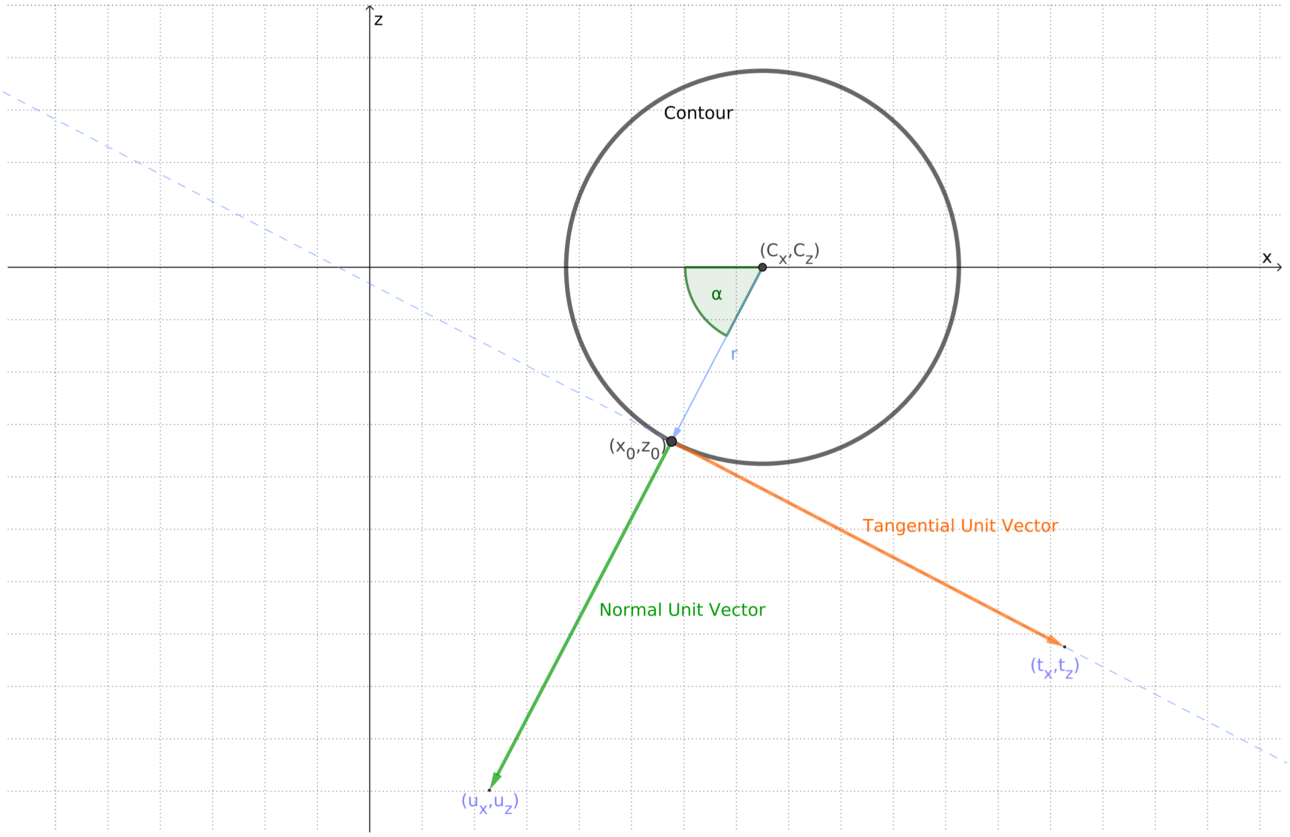


Figure 7.27: Normal and Tangential Unit Vector

Equation 7.4 shows the mathematical expression required to obtain the coordinates of the tangential unit vector.

$$\begin{bmatrix} u_{t1} \end{bmatrix} = \begin{Bmatrix} t_{x1} \\ t_{y1} \\ t_{z1} \end{Bmatrix} = \begin{Bmatrix} x_0 + sin(\alpha) \\ 0 \\ z_0 - cos(\alpha) \end{Bmatrix} \tag{7.4}$$

A second tangential direction was determined in this case study. This direction is parallel to the axis of the cylinder (y-direction). The mathematical expression of the unit vector pointing in this direction is shown in the Equation 7.5. The positive direction of this vector coincides with the positive y-axis.

$$\begin{bmatrix} u_{t2} \end{bmatrix} = \begin{Bmatrix} t_{x2} \\ t_{y2} \\ t_{z2} \end{Bmatrix} = \begin{Bmatrix} 0 \\ (y_0 + 1) \\ 0 \end{Bmatrix} \tag{7.5}$$

Stress study: Slice 1

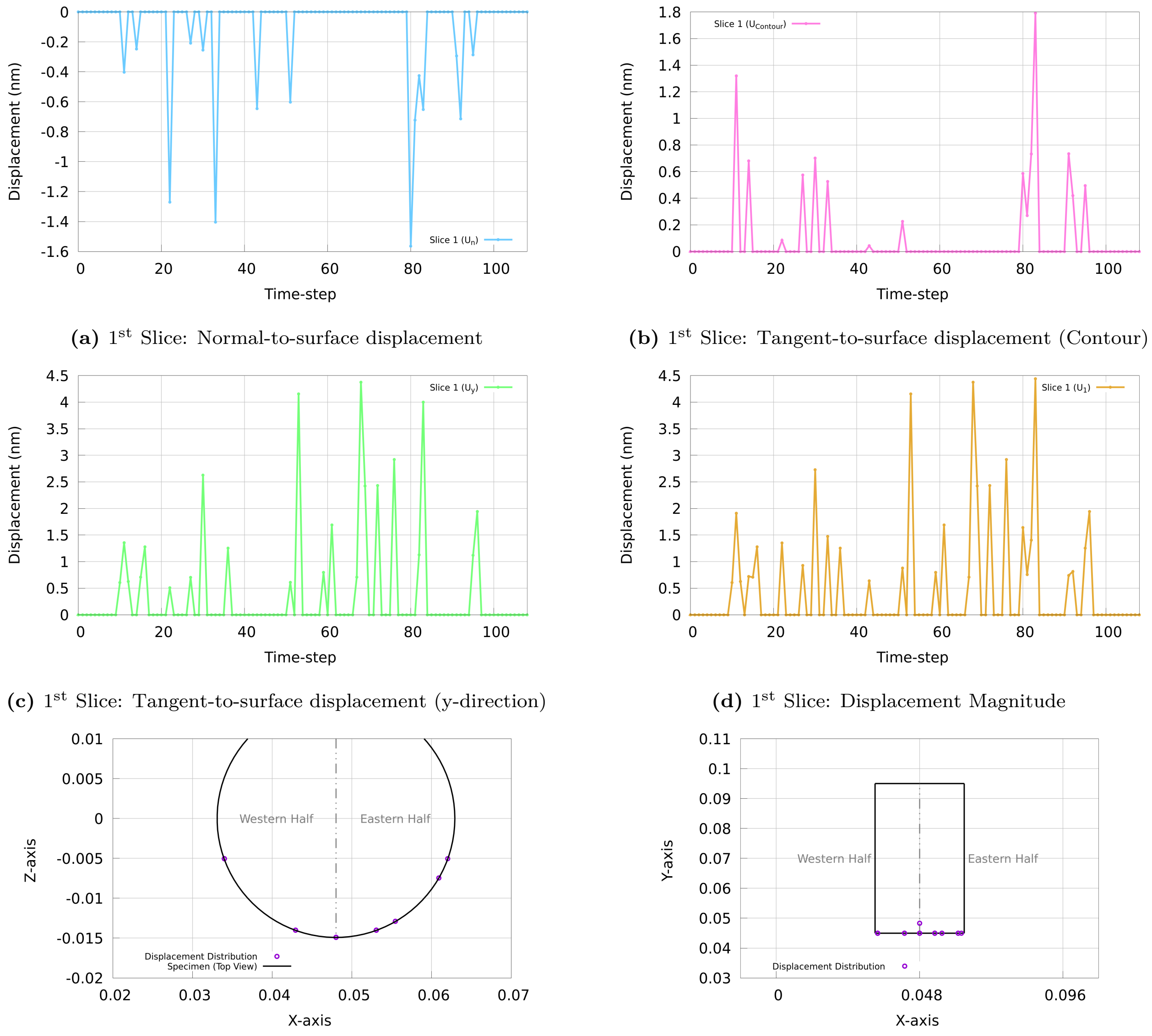
This slice contains all the points where the impacts take place, the specimen's surface. As mentioned previously, FEM, does not provide reliable information with regards to the stress distribution on the specimen's surface. Hence, this slice is used to study the displacement of the node as a consequence of the media impacts.

Figure 7.28 illustrates the greatest deformations in the slice for each time-step of the simulation. More specifically, Figure 7.28a displays the displacement in the direction normal to the specimen's surface. The negative sign of the displacement's magnitude indicates that the deformation is taking place towards the centre of the specimen. It is observed that there is not deformation for each of the time-steps forming the simulation even when it is known as a fact that the media is always in contact with the specimen. This suggests that not all the contacts produce a deformation on the specimen's surface.

Figures 7.28b & 7.28c, respectively present the results for the deformation, in the direction tangent to the specimen's contour and also the tangent direction parallel to the specimen's axis (y-direction). The latter is greater than the deformations in the other two directions. It is noticed that not all the impacts causing deformation produce displacements in the three directions. At times the only component of the deformation is purely tangential. This can be recognized when comparing the displacement magnitude display on Figure 7.28d with the displacement in the three directions. As known, the magnitude of the displacement is obtained from the expression $U = \sqrt{U_n^2 + U_{contour}^2 + U_y^2}$. As observed, there are more non-zero values for the magnitude of the displacement than for the other directions. Therefore, this can only be explained by the fact that the deformation does not always produce displacements in the three directions. The average of the greatest displacement over time is of the order of 1.6nm. The width of the specimen's wall is 5mm. Hence, the displacement represents a $3{\times}10^{-8}\%$ of the width of the specimen's wall.

Figures 7.28e & 7.28f details the location of the impacts producing deformations. As observed, most of them appear on the lower edge of the specimen where the media travels faster than in other regions of the bowl.

As a conclusion, the current loading scenario produces a range of displacement that do not produce plastification in any of the areas of the specimen. The deformations are so small that they can be



(a) 1st Slice: Normal-to-surface displacement

(b) 1st Slice: Tangent-to-surface displacement (Contour)

(c) 1st Slice: Tangent-to-surface displacement (y-direction)

(d) 1st Slice: Displacement Magnitude

(e) 1st Slice: Displacement distribution (Top View)

(f) 1st Slice: Displacement distribution (Front View)

Figure 7.28: Slice 1: Displacement distribution over time

considered negligible. Therefore, the Von Mises and Principal stresses provided by the FEA, are not of use to this study.

Stress study: Slice 2

The Equation 7.8 displays the values of the stress tensor for a specific point and time-step. It also shows the coordinates of the unit vector normal to the surface at that point. The stress vector is obtained as a result of this operation.

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} -207,403 & -16,912.4 & -20,274.8 \\ -16,912.4 & -239,069 & -47,647.1 \\ -20,274.8 & -47,647.1 & -589,339 \end{bmatrix} \begin{Bmatrix} -0.302 \\ 0 \\ -0.952 \end{Bmatrix} \Rightarrow \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} 81,913 \\ 50,477 \\ 567,327 \end{bmatrix} \quad (7.6)$$

In order to find the value of the stress acting on that point in the normal direction, the stress vector has to be multiplied by the unit vector normal, Equation 7.9.

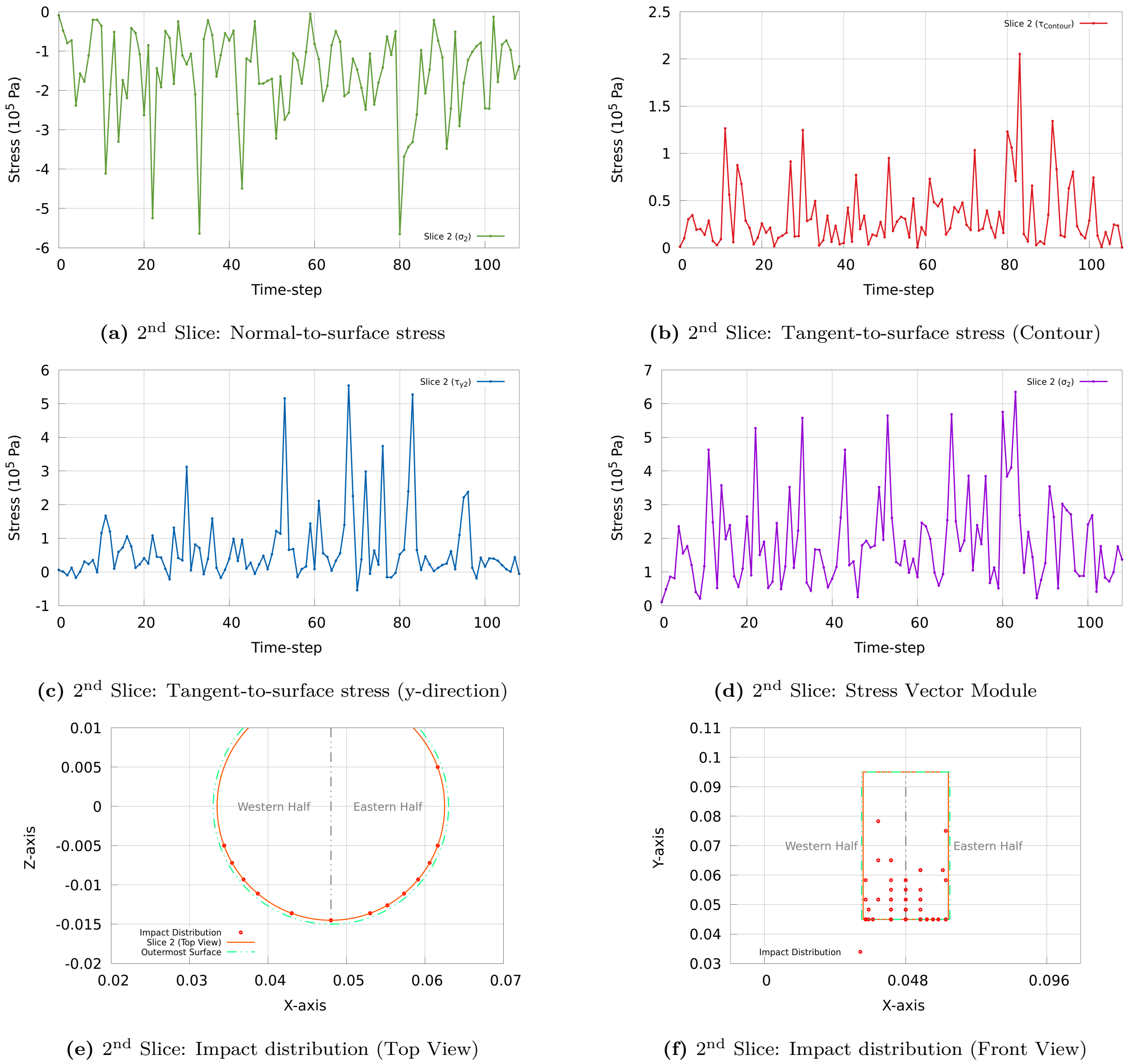
$$\vec{T} \times \vec{u_n} \Rightarrow (T_1 \cdot l + T_2 \cdot m + T_3 \cdot n) \Rightarrow (81,913 \cdot (-0.302) + 50,477 \cdot 0 + 567,327 \cdot (-0.952)) = -564,973 \quad (7.7)$$

Figure 7.29a displays the maximum normal stress for each time-step acting on the first slice located at 0.5mm from the specimen's surface. As is observed these stresses are negative which indicates a compressive effort acting on the surface of the specimen. The average value of the maximum normal stress over time is in the order of 159,000 Pa. Figure 7.29b depicts the stress tangential to the contour of the specimen. The positive direction of the unit vector is anticlockwise which means that the stresses acting on the eastern half of the specimen are positive whereas those acting on the western half are negative. The absolute value of the stress has been plotted in order to facilitate the interpretation of the results. The average tangential stress acting in this direction is in the order of 33,000 Pa. Figure 7.29c shows the tangential stress acting in the vertical direction. It is observed that the majority of the stresses are positive, which indicates the particles tend to impact the specimen from bottom to top. The average stress on this direction over time is of the order of 72,000 Pa. Figure 7.29d presents the values of the module of the vector of the stress, the average value of it over time is around 190,000 Pa.

Figures 7.29e & 7.29f show, respectively, a partial top view of the contour of the specimen and a front view of it. The red dots represent the impacts producing the greatest stresses over time. It is observed that the highest concentration of dots is in the lower part of the specimen and towards the western side. A significant number of impacts take place at the very edge of the specimen.

Stress study: Slice 3

Following the same arrangement as in the previous section. Figure 7.30 shows the key stresses acting on the third slice, at 1.0mm from the surface of the specimen. As is to be expected, the third slice presents lower values of stresses in all the studied directions. The maximum normal stresses remain as compressive stresses, their average value is in the order of 84,000 Pa which represents a 47%



(a) 2nd Slice: Normal-to-surface stress

(b) 2nd Slice: Tangent-to-surface stress (Contour)

(c) 2nd Slice: Tangent-to-surface stress (y-direction)

(d) 2nd Slice: Stress Vector Module

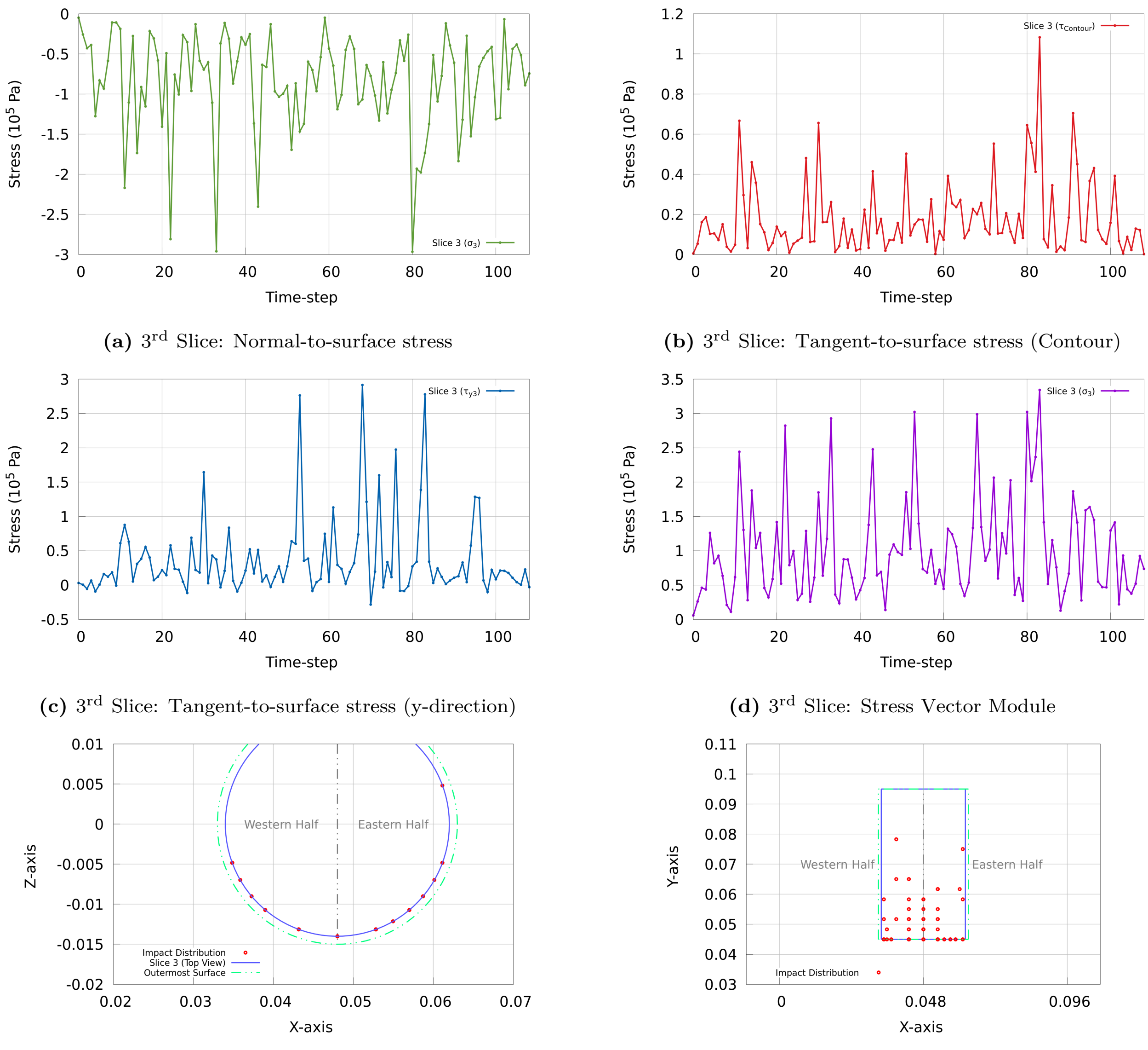
(e) 2nd Slice: Impact distribution (Top View)

(f) 2nd Slice: Impact distribution (Front View)

Figure 7.29: Slice 2: Stress distribution over time

stress reduction in comparison with the previous slice. Similarly, the two tangential stresses were reduced by the same percentage. These results confirm the consistency in the decrement in the stress between the different directions. This can be due to the isotropic property of the specimen's material.

Figures 7.30e & 7.30f do not change significantly with respect to the previous slice. The front lower part of the slice, specially the western half, sustain the higher concentration of maximum stresses over time.



(a) 3rd Slice: Normal-to-surface stress

(b) 3rd Slice: Tangent-to-surface stress (Contour)

(c) 3rd Slice: Tangent-to-surface stress (y-direction)

(d) 3rd Slice: Stress Vector Module

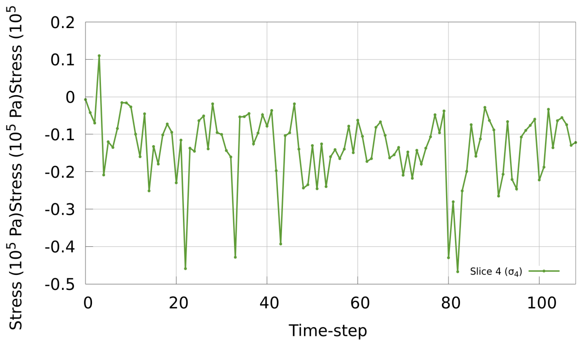
(e) 3rd Slice: Impact distribution (Top View)

(f) 3rd Slice: Impact distribution (Front View)

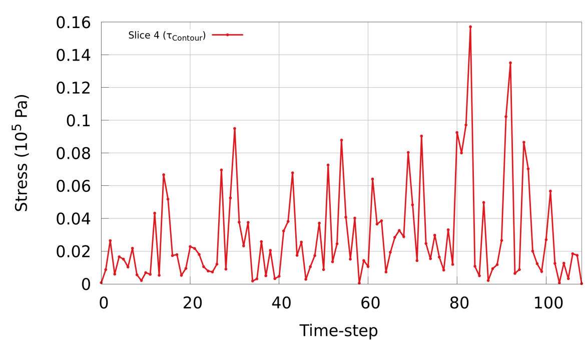
Figure 7.30: Slice 3: Stress distribution over time

Stress study: Slice 4

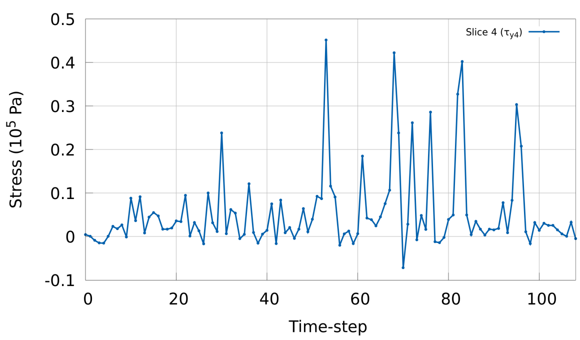
Figure 7.31 displays the results for the 4th Slice. At 1.5mm from the surface the stresses are reduced over 90%. As for the previous slices, the normal stresses plotted in Figure 7.31a correspond to those that are the greatest for each time step. In this case, it is observed that although the majority of the stresses are compressive stresses, there also exist traction stresses that occasionally surpass the compression stresses. On the other hand, the concentration of traction stresses remains in the same area than for the previous slices.



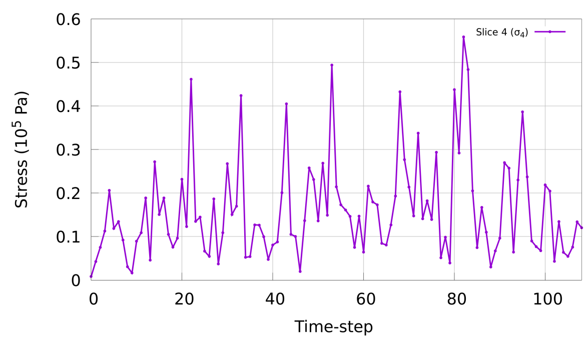
(a) 4th Slice: Normal-to-surface stress



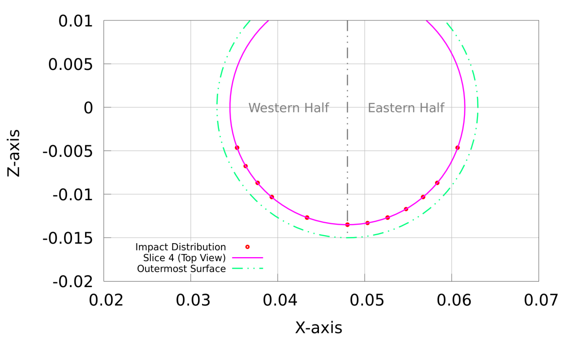
(b) 4th Slice: Tangent-to-surface stress (Contour)



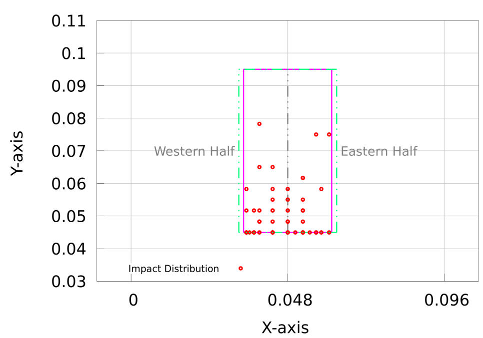
(c) 4th Slice: Tangent-to-surface stress (y-direction)



(d) 4th Slice: Stress Vector Module



(e) 4th Slice: Impact distribution (Top View)



(f) 4th Slice: Impact distribution (Front View)

Figure 7.31: Slice 4: Stress distribution over time

Table 7.3 summarizes the average of the maximum stresses over time in the three directions. Where S_n, S_{tc}, S_{ty} respectively stand for Normal Stresses, Stresses tangential to the specimen's contour and Stresses in the direction of the specimen's axix. It is also shown the reduction of the stress for each slice with respect to the second slice; situated at 0.5mm from the surface. It is observed that the decay of the stress in the interior of the specimen is lineal, which is aligned to the behaviour of isotropic materials.

Table 7.3: Stress summary

Distance from surface (mm)	Key Stresses (Pa)			Reduction from surface (%)		
	S_n(avg)	S_{tc}(avg)	S_{ty} (avg)	S_n	S_{tc}	S_{ty}
0.5	159,168	33,525	72,714	0	0	0
1	84,759	17,905	38,772	47	47	47
1.5	13,744	2,947	5,537	91	91	92

For the conditions established in this case study, beyond 1.5mm from the specimen's surface the stresses in the interior of the specimen are null or negligible.

7.6 Case Study III

This case study continues the examination of the effects of the media impact on the surface of a specimen. As in the previous case studies the part was fixed within the media flow. The specimen is a rectangular prism, its base is parallel to the global coordinate plane xz and it is a perfect square with 15mm side. Its height is 100mm. The mesh element employed to discretize the prism was, as previously, the 8-nodes-brick. On this occasion, three different regions were differentiated to have a better insight into the distribution of stresses. Figure 7.32 displays the specimen's top view. The four corners of the specimen in yellow, comprised the first region, a fine element with L0.8mm $\times$ W0.8mm $\times$ H1.7mm was defined for it. The front, rear and sides, in blue, constituted the second region. A three-times longer element was employed for this region. Finally, the core of the specimen, green-coloured, comprised the third region. The coarsest element, three-times wider than the previous one, was allocated to this region since this part is not in contact with the media. The

Table contained in Figure 7.32 summarizes the total number of hexahedra used for each region.

Item	Quantity
Nodes	32,268
Lines	1,312
Corner Hexahedra	8,640
Lateral Hexahedra	14,400
Core Hexahedra	6,000
Total Hexahedra	29,040

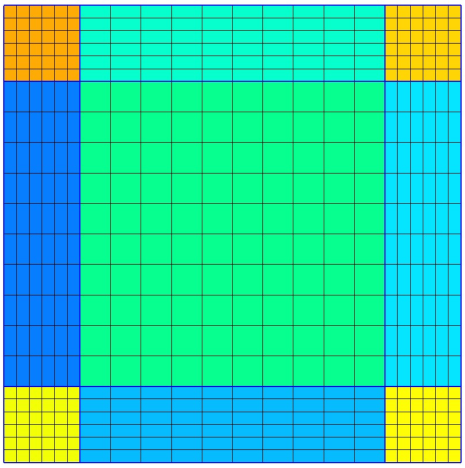


Figure 7.32: Specimen's mesh specifications

The geometry of this specimen is justified by the necessity of knowing whether the concentration of stresses, in the lower edge of the specimen studied in the Case Study II, is due to its relative position within the granular flow, or on the contrary, is due to its condition of edge. At the same time, this geometry facilitates the study of the distribution of stresses in the interior of the part by possessing a face completely parallel to the xy coordinate plane.

In order to keep consistency between Case Study II and III, all the parameters used in the previous case study were repeated in this one. The specimen was fixed at the same coordinates as the *Position I* of the previous case study as shown in Figure 7.17.

7.6.1 Normal stress distribution

Two slices were isolated from the whole specimen to facilitate the study of the normal stresses, Figure 7.33. The slices were created parallel to the side facing the media flow. The first one is situated at 0.8mm from this side. This is exactly the width of the first layer of elements. The second slice was at 1.6mm from it, which it is the equivalent to two layers of elements.

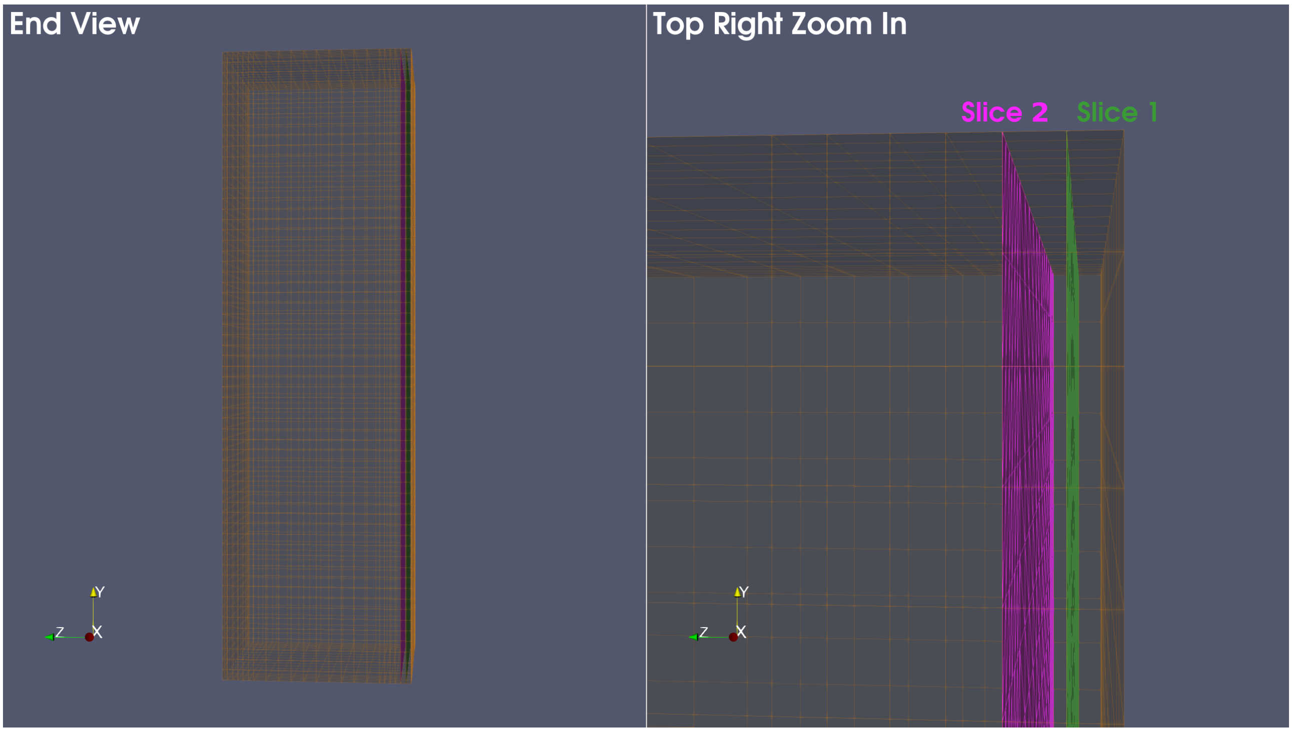


Figure 7.33: Specimen's slices

The distribution of normal stresses was studied for the time-steps: 13, 37, 53, 90. To find the normal stress acting on the nodes of the slice the Cauchy Principle was used to calculated the vector of stress at the nodes and then the normal stress. On this occasion the unit vector normal to the surface makes null most of the stress tensor as shown in Equation 7.8.

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \Rightarrow \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix} \tag{7.8}$$

The normal stress is the scalar product of the stress vector [T] and the unit vector {u}.

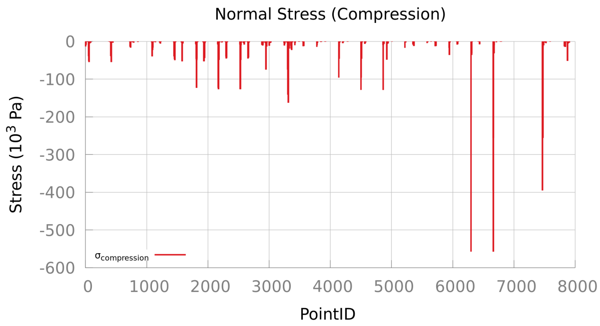
$$\begin{bmatrix} T \end{bmatrix} \times \begin{Bmatrix} u \end{Bmatrix} = \begin{bmatrix} \tau_{xz} & \tau_{yz} & \sigma_z \end{bmatrix} \times \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \Rightarrow \sigma_n = \sigma_z \tag{7.9}$$

The normal stresses were divided into compression stresses (negative) and traction stresses (positive). Stresses in the range [-500Pa, 500Pa] were considered trivial.

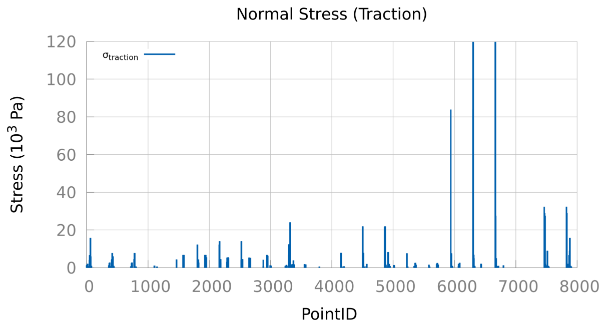
Normal stresses: Slice 1

When the media impacts the surface of the specimen the stresses travel across the matter generating a field of stresses. This field is not pure in its directions but it is a confluence of stresses with different signs and directions. This simulation provides an insight into this field. Figure 7.34 depicts the normal stresses for the first slice over the different time-steps. At the right hand side, in blue, the tensile stresses are represented, the compressive stress at the other side, in red. It is observed that the compressive stresses are greater than the tensile stresses for all the studied time-steps.

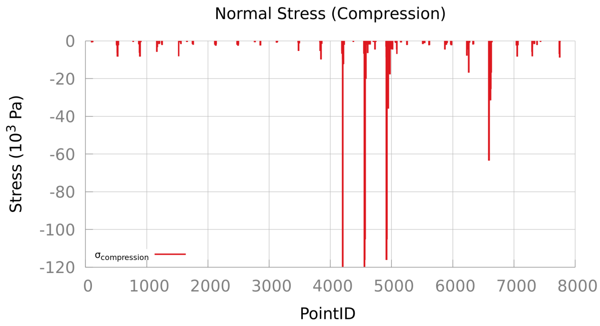
Table 7.4 summarizes the key values of the stress distribution. The average value was used to compare the weight of each stress within the stress field. It was revealed that the compressive stresses were the most significant part of the field, for all the time-steps it comprised more than 65% of the normal stress. It can be stated that at 0.8mm from the surface the majority of the normal stresses are compressive.



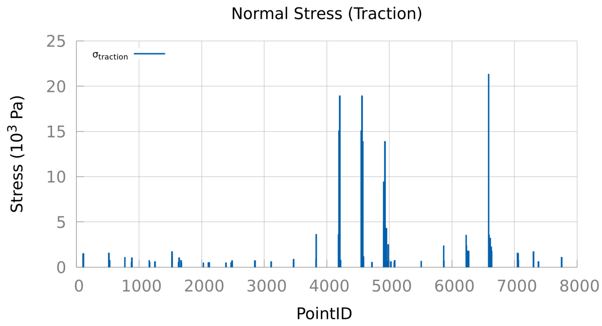
(a) Compression stress distribution (time-step 13)



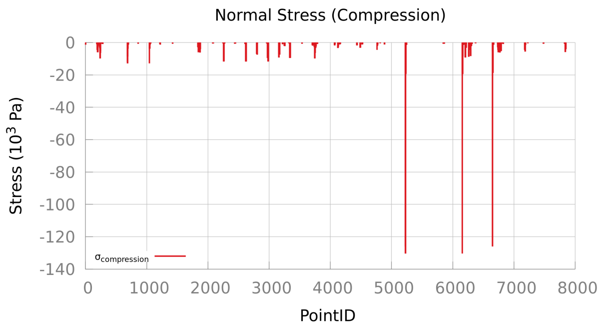
(b) Traction stress distribution (time-step 13)



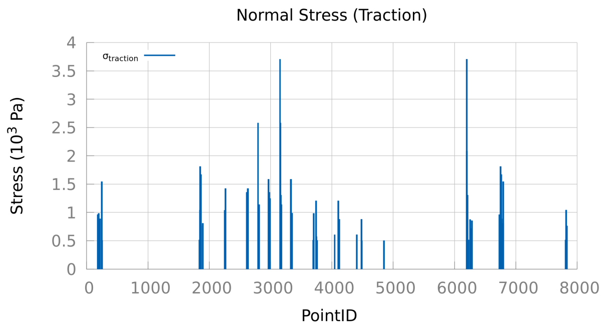
(c) Compression stress distribution (time-step 37)



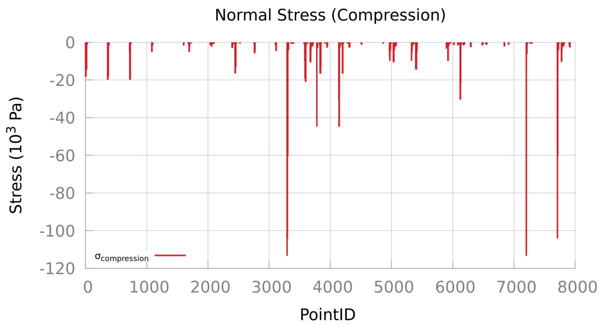
(d) Traction stress distribution (time-step 37)



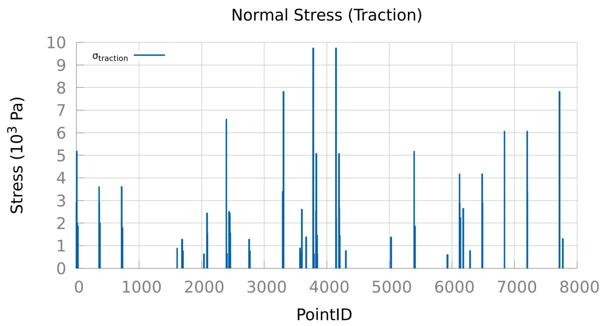
(e) Compression stress distribution (time-step 53)



(f) Traction stress distribution (time-step 53)



(g) Compression stress distribution (time-step 90)



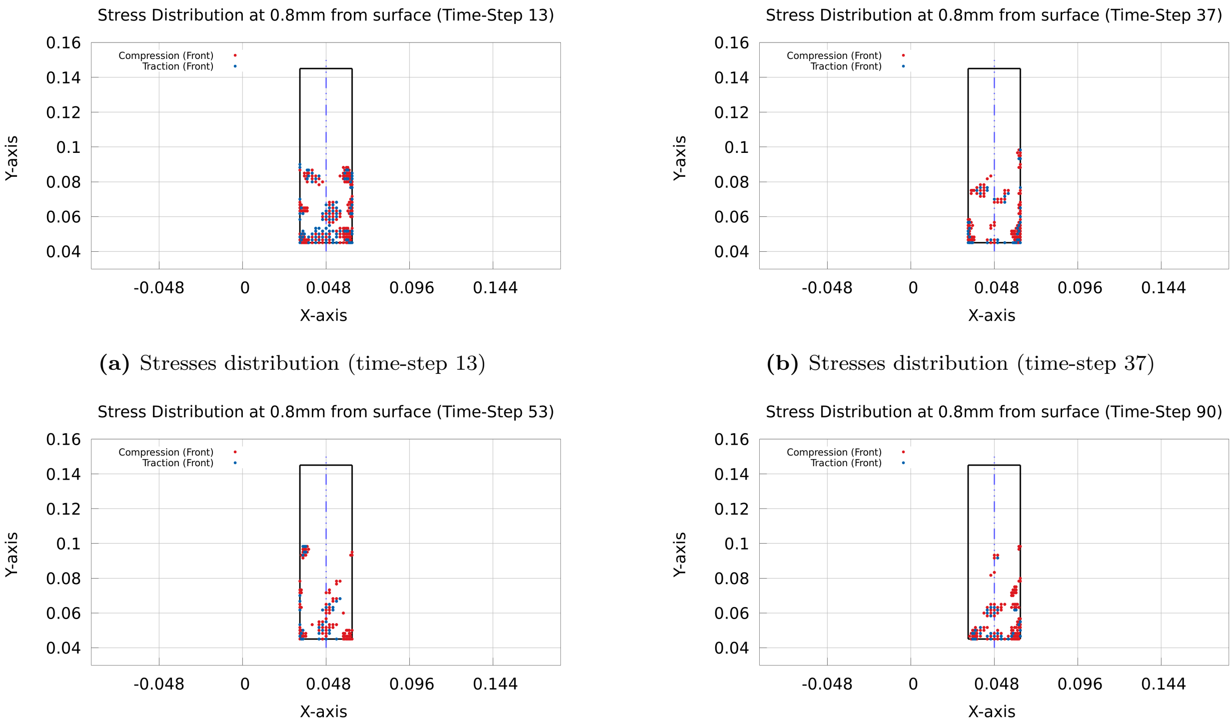
(h) Traction stress distribution (time-step 90)

Figure 7.34: Slice 1: Stress distribution at 0.8mm from surface

Table 7.4: Slice 1: Stress summary

Time-step	Max ($\sigma_{compressive}$)	Max ($\sigma_{tensile}$)	Avg ($\sigma_{compressive}$)	Avg ($\sigma_{tensile}$)	$\sigma_{compressive}$ (%)	$\sigma_{tensile}$ (%)
13	-557,537	119,675	-18,866	6,381	74.7	25.3
37	-119,810	21,377	-8,479	3,132	73.0	27.0
53	-130,446	3,705	-5,159	1,130	82.0	18.0
90	-113,370	9,763	-5,388	2,623	67.3	32.7

Figure 7.35 shows the physical distribution of stresses within the slice. The red dots represents the nodes affected by compressive stresses while the blue dots represents tensile stresses. Only those nodes affected by the stresses outside the range [-500Pa, 500Pa] were plotted.



(a) Stresses distribution (time-step 13)

(b) Stresses distribution (time-step 37)

(c) Stresses distribution (time-step 53)

(d) Stresses distribution (time-step 90)

Figure 7.35: Slice 1: Stress distribution over the surface

It is observed than the concentration of stresses is fundamentally in the lower part of the specimen as this part remains immersed in the media flow. Furthermore, the vicinities of the edges present the highest concentration of normal stresses, especially compressive stresses. This was also observed in the Case Study II, where there was a high concentration of stresses at the lower edge over time. However, it is detected in the current specimen a concentration of stresses around the edges but not necessarily the bottom edge, depending on the time-step the concentration varies from one edge to another.

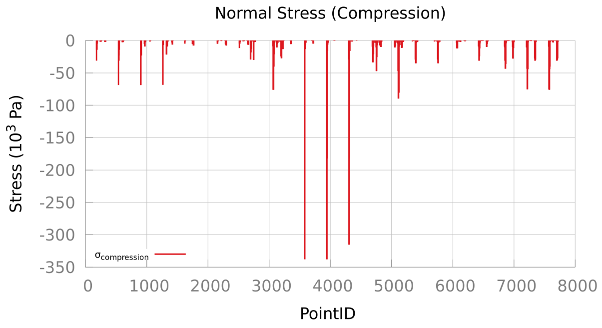
Normal stresses: Slice 2

Following the approach used in the previous slice, the nodes affected by normal stresses outside the range [- 500Pa, 500Pa] were studied. Then the values of the normal stresses were divided into compression and tensile stresses and plotted, Figure 7.36. As expected, the stresses are reduced considerably as the distance increases.

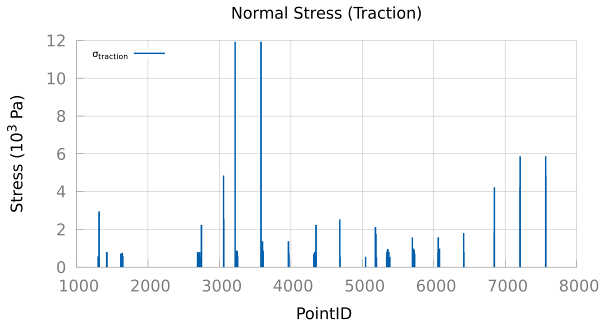
Table 7.5: Comparison of the nodes affected by stresses (Slice 1 vs Slice 2)

Slice 1		**Slice 2**	
Nodes ($\sigma_{compression}$)	Nodes ($\sigma_{tensile}$)	Nodes ($\sigma_{compression}$)	Nodes ($\sigma_{tensile}$)
803	478	1142	191
516	183	684	47
464	162	565	9
570	165	674	10

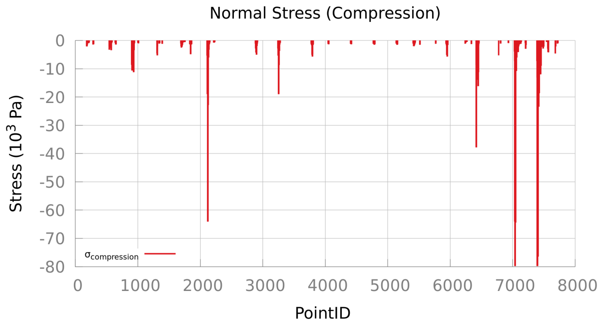
Interestingly, it is observed that the quantity of nodes affected by the compressive stresses increases as the distance increases. On the contrary, the number of nodes affected by the tensile stresses decreases as the distance increases. Table 7.5 collects this data.



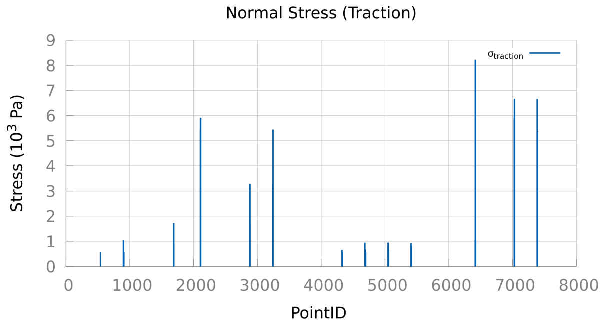
(a) Compression stress distribution (time-step 13)



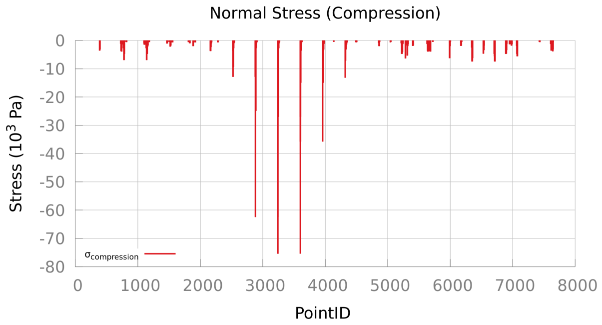
(b) Traction stress distribution (time-step 13)



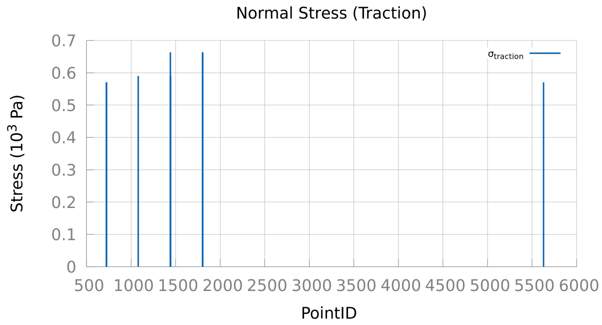
(c) Compression stress distribution (time-step 37)



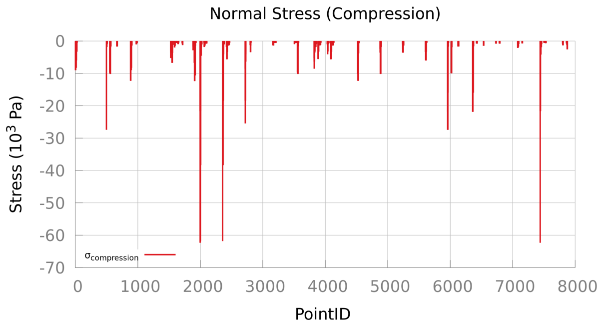
(d) Traction stress distribution (time-step 37)



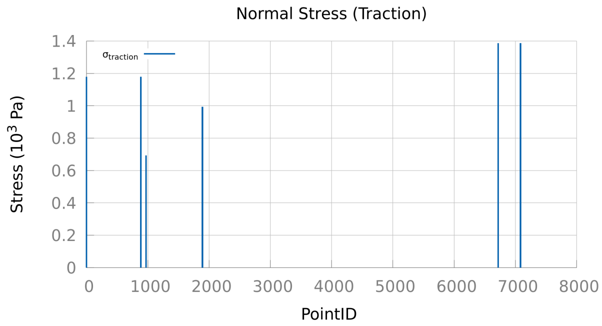
(e) Compression stress distribution (time-step 53)



(f) Traction stress distribution (time-step 53)



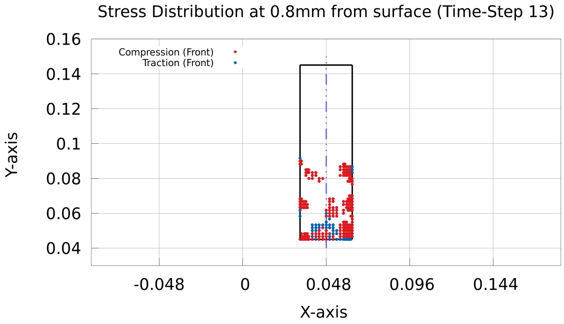
(g) Compression stress distribution (time-step 90)



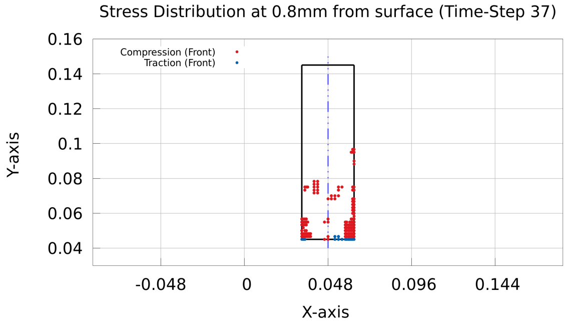
(h) Traction stress distribution (time-step 90)

Figure 7.36: Slice 2: Stress distribution at 1.7mm from surface

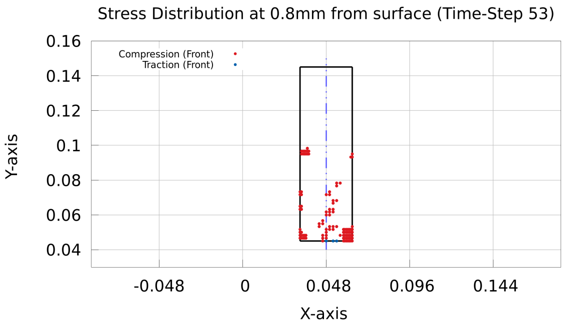
Figure 7.37 shows the nodes affected by the normal stresses. As predicted, the areas affected on the second slice are the same as those affected on the first slice. The essential difference consists in the increment of the proportion of nodes affected by compressive stresses over the nodes affected by tensile stresses.



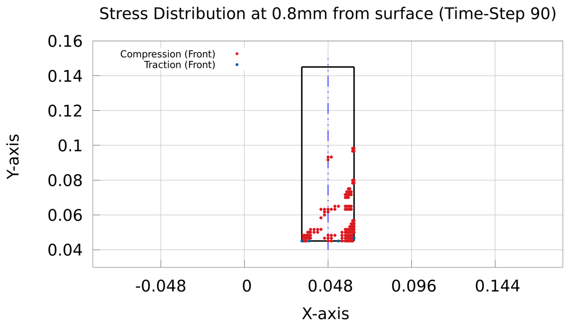
(a) Stresses distribution (time-step 13)



(b) Stresses distribution (time-step 37)



(c) Stresses distribution (time-step 53)



(d) Stresses distribution (time-step 90)

Figure 7.37: Slice 2: Stress distribution over the surface

7.7 Stress analysis summary

This section presents a summary of the most important findings discussed over the case studies.

Case study I

It was concluded that the inclined configuration of the specimen absorbs more energy thus, the velocity of the bulk is lower and the stress distribution on their surfaces is higher. On the other hand, it is essential to consistently rotate the part in order to obtain a surface with a homogeneous material attrition. More simulations would be needed to have a deeper insight into this subject.

Case study II

The media impacts on the surface of the specimen produce a mixture of normal and tangential forces that travel across it producing, on one hand, a distribution of normal stresses characterized by its compressibility, on the other hand a tangential distribution. The two tangential directions that are relevant to this study are the vertical one, in the direction of the axis of the specimen (y-direction) and the direction tangent to the contour of it. Among the tangential stresses the highest values were identified at the positive side of the y-axis. This indicates that the media impact the surface from bottom to top. Moreover, the normal stresses acting on the specimen are always more significant than the tangential ones.

The evolution of the stresses within the specimen corroborates the isotropic behaviour of the material as it is under the influence of a state of stresses weak enough not to surpass the elastic region of the material.

Most of the impacts producing the highest concentrations of stresses are located in the lower part of the specimen where the velocity of the media reaches the highest values and towards the western part of the specimen. This is due to the fact that the density of the media at that side is greater than at the other side where the vortex drags in most of the media flowing around the specimen.

Case study III

It has been revealed that the stress field in the interior of the specimens tends to be more dense on those edges in permanent contact with the media and their vicinities. This indicates that a larger quantity of material would be removed from the edges than from the faces.

It has also been acknowledged that the increment of the distance from the face receiving the media impact to the stress field not only reduces the magnitude of the stresses within the field but it also tends to homogenize the field by transforming it into a compressive stress field. Using a convergence study on the mesh would be necessary to fully affirm the latter observation, through such is beyond the scope of this study.

Chapter 8

Discussions

It was found, throughout this project, that DEM represents a powerful tool to study the behaviour of the granular matter inside the rotary mass finishing processes. It has been a key instrument to understand the flow pattern within the bowl and the range of velocities the media particle reaches in the different regions of the work chamber. Furthermore, this approach predicts the behaviour of the media flow when the features of the work chamber vary. This enables the mechanical designers in industry to accurately define their prototypes without needing to follow a trial-and-error approach which increases the costs of the new designs.

DEM also added a valuable insight on the interaction between particles and fixed workpieces, identifying those areas of the specimens that are more likely to be impacted and with a higher intensity depending on their positions within the bowl and how the impacts on the specimens affect the total kinetic energy of the bulk. This information is helping to contribute knowledge that will help the manufacturers of drag finishing machines to scientifically support their recommendations to clients on how to proceed when using their systems.

The DEM-FEM couple clarify the tensional state of the fixed work-piece behind its surface. This finding helps to understand the possibility of using the mass finishing technology to correct the residual stresses on those work-pieces produced using the additive manufacturing technology. This approach potentially produces a huge beneficial impact on the development of the additive manufacturing technology as at the same time that the surface of the workpiece is treated, the tensional state of its subsurface regions is addressed.

An obstacle of the DEM approach applied to the mass finishing field resides in the often difficult challenge of finding the mechanical properties of the material involved in the process, especially, the damping and friction properties. The morphology and dimensions of the media particles represent a troublesome challenge in obtaining those. A set of experiments were designed to overcome these difficulties. Although the approach followed to identify the damping coefficient is not entirely novel the idea of modifying the geometry of the media to the appropriate shapes is fully original and can be used for researchers in this and other fields that present the same difficulties. On the other hand the method employed to obtain the coefficient of friction between the different pairs (particle-particle and particle-wall) is wholly original, no other researcher has been found using a similar approach. The benefits of this are that the efficiency and simplicity of the mechanical arrangement designed for this end, makes possible to try as many pairs of materials as needed, and with low cost of the experiment.

A real limitation of this approach is the computing resources needed to obtain a few seconds of simulation. It is very important to well define the simulation script thus the simulation time can be reduced to the minimum. Even though, having access to a HPC is mandatory when working with a reasonable quantity of particles providing a realistic insight, unfortunately, this limitation affects industry as not all manufacturing companies can afford the costs of having an in-house HPC. Some commercial software packages have successfully enabled the use of GPUs in order to maximize the usability of their products.

On a different subject, the study of the movement governing the vibratory trough was part of the objective of this project. The possibility that the system had been designed to match its natural frequency and the frequency of the excitation was initially considered. Thus, less energy is required to obtain the same magnitude of displacements. It is believed that this possibility was not considered by the manufacturer. The information retrieved from the analysis can be used by designers in industry to develop more efficient systems.

The trough behaviour was studied under different loading conditions. It was found that the mass of the media does not affect the natural frequency of the system. However, it has an impact on the amplitude of the displacement of the trough. This is key in the vibratory mass finishing process as a low displacement would not energize the particles enough to follow their characteristic flow trend. The information retrieved from the comparison of the system when performing under different loading scenarios can be used for the manufacturers to advise on the maximum load at which the system runs satisfactorily.

Inputting the vibratory movement into the DEM software was unsuccessful, it is believed that although the approach followed is mathematically consistent the algorithm in charge of inputting the vibration has a limitation when the typology of the vibration is more complex than a single sinusoidal signal. This finding identifies a potential lack of functionality that could be addressed by LIGGGHTS software developers.

Chapter 9

Conclusions

This section attempts to summarize the most relevant conclusions found in the different works that comprise this project.

One of the most challenging aspects of the DEM simulation is obtaining the mechanical properties of those materials involved in the process. This mainly occurs because DEM is a fairly new simulation method and not many materials have been investigated when considered as in a granular matter process. Therefore, these properties such as the coefficient of restitution and the friction coefficient are unknown. Furthermore, this properties are not inherent to the material but dependent on the pair of materials under scrutiny. Properties such as Young's Modulus or Poisson's ratio remain fairly unchanged in the absence of a heat source. The density of the materials can also be understood as a constant. However, the coefficient of restitution is a *material-pair property* that changes when either of the materials from the pair changes. For example, a tennis ball does not behave the same on a clay court and a hard court. Similarly the coefficient of friction changes depending on the surface on which the material is sliding. For these reasons it is complicated to find some materials' properties within the current literature.

During the work carried out to retrieve the properties of the material involved in the MF processes it was observed:

1. The tensile test was not appropriate to identify Young's Modulus and Poisson's ratio. This is due to the brittle condition of the media, on the contrary the compression test was adequate to this end.

2. The method proposed to find the coefficient of friction between different surfaces yielded good results. It can be adapted to any pair of surfaces, the time needed to set the experiment up is short and it is not expensive.

3. The method employed to retrieve the coefficient of restitution, contrarily to the method proposed for the coefficient of friction requires some expensive equipment not always available. This inconvenience was the reason to approach a different method, trying the use of the FEM to simulate the experiment and obtain the data in a virtual way. Unfortunately, the results obtained were unsuccessful and investigating more in this direction was beyond the scope of the current project. However, this effort represents a step forward so others can continue investigating this approach.

The excitation source of the vibrational trough was studied to mathematically define it and then input it in the DEM model. It was concluded that the mass of the media in the work chamber alters the movement of the trough, when this is unloaded the amplitude of the acceleration is lower than when loaded and the signal of its vibration is more symmetric with respect to the axis of the abscissas. On the contrary, the more loaded the system is the more erratic its signal becomes. For example, in the case where the system is loaded with 100 kg of media the amplitude of the acceleration augments 50% in respect to the scenario where the system runs empty. This is a consequence of the inertia of the extra mass. For this same reason its signal is less symmetric. Inputting the vibrational signal in the DEM model was unsuccessful, it is believed that the algorithm used to implement this type of movement does not allow it to work with complex vibrations.

It has been highlighted in the task of simulating the complex shape of granular materials in the MF system, the necessity of defining the particles involved in the process using multi-sphere clumps to approximate the real original shape. In the vibratory systems as well as in the rotatory systems the multi-sphere particles always produced more realistic results than the spherical simplification of the original shapes. Therefore, it was concluded that the spherical simplification does not yield enough accurate data to be used in the design of future machines or to improve the current processes. Moreover, the multi-sphere particles clarified the cyclical flow trends of the particles when in the work chamber of the rotary machine and the distribution of velocities that could lead to position the work-pieces in the appropriate regions of the bulk. These findings are potentially significant for mechanical design engineers tasked with the development of drag finishing systems and also to end-users who want to adequately position the parts to be polished.

The approach of comparing the trajectories and velocities of a real work-piece with a simulated one when travelling within the bulk in a rotatory disc machine yielded very interesting result. It was concluded that the simulation offered very realistic results and it helped to understand the flow trends within work chamber, and the impact that the design features have on the behaviour of the flow.

The information extracted from the simulation of the rotary disc machine was key to deciding the position of the specimens on which the stresses produced by the impact of the media particle were studied. Very valuable conclusions were obtained from this study:

1. In the study of two cylindrical specimens fixed in the same position within the bulk travelling in the interior of a rotary disc machine, one of them completely parallel to the axis of the work chamber the other 30 degrees inclined with respect to the same axis. It was concluded that the inclined configuration of the specimen absorbed more energy thus, the velocity of the bulk is lower and the stress distributions on their surfaces are higher. On the other hand, it is essential to consistently rotate the part in order to obtain a surface with a homogeneous material attrition.

2. When studying the media impacts on the surface of a cylindrical specimen it was found that these impacts produce a mixture of normal and tangential forces that travel across the specimen producing, on one hand, a distribution of normal stresses characterized by its compressibility, on the other hand a tangential distribution. The normal stresses acting on the specimen are always more significant than the tangential ones. Most of the impacts producing the highest concentrations of stresses are located in the lower part of the specimen where the velocity of the media reaches the highest values.

3. It was revealed that the stress field in the interior of a square specimen tends to be more dense on those edges in permanent contact with the media and their vicinities. This indicates that a larger quantity of material would be removed from its edges than from its faces.

Lastly, the following few lines summarise the contribution to knowledge of the works presented herein:

- Well defined experiments to retrieve specific mechanical properties.

- Vibrational characterization of the lab vibro-trough.

- Flow model properly validated.

- In depth analysis of the interaction between particles and a fixed specimen.

Chapter 10

Further works

This chapter is aimed to other students or academics who potentially want to look at further research in this field. It is structured in two sections. The first one discusses objectives that were not met. The second one proposes novel works beyond this research.

Unmet objective

As it has been stated in Chapter 5. Inputing the vibrational signal into LIGGGHTS was unsuccessful, as a side work of this project the same simulation was launched using the commercial software EDEM, unfortunately no benefits were found. LIGGGHTS is an open source software, the source code is public and it can be modified and re-compiled to implement new features to the software. For this reason, it is suggested to look into the algorithm applying the vibration movement to the particles and modified in a way such that this type of movement can be accurately introduced. For this task a high command of the programming language C++ and a solid background in vibration analysis is needed.

Due to the time constraints, the works discussed in Section 4.4.10, with respect to the possibility of using FEA to simulate the experiments designed to retrieve the values of the coefficient of restitution, remained unfinished. Not an excessive amount of further work is needed to clarify whether this approach brings some value to the research community.

Further novel work

The granular matter behaves as a fluid in certain conditions, and in the case of a rotatory disc finishing machine the granular flow could be comparable to a fluid flow. On the other hand, at present, DEM demands a huge amount of computational resources, not always available in the research centres. However, CFD simulations requires less computational resources. For this reason other researchers have used CFD in the past. It would be interesting to characterize a fluid which could behave as this granular flow and then validate it against DEM. In the case this approach succeeds, it would save a lot of computing time. Furthermore, most of the CFD packages allow one to couple FEA packages to analyse the interaction between fluids and solids. This could simplify the study of the interaction between the bulk and the workpieces.

One of the limitations of the coupled DEM-FEA developed as part of the works of thesis is that too many steps are needed in order to retrieve the result to then be interpreted. A more advisable approach would be to directly write an algorithm to implement it within LIGGGHTS so the forces are directly transferred to the nodes so that the stresses and displacement are directly computed. It could also be implemented a possibility of studying parts moving freely within the granular fluid which it is not possible to achieve in the current public version of the software. For this task an important knowledge of C++ programming language is essential as well as a strong background in the finite element theory.

As part of this work, the interactions between particles and a fixed workpiece were studied. The next step down this line would be to couple a wear model to the results obtained from the FEA of the workpiece allowing information with respect to the material attrition to be obtained.

References

[1] J. Domblesky, R. Evans, and V. Cariapa. Material removal model for vibratory finishing. *International Journal of Production Research*, 42(5):1029--1041, 2004. ISSN 00207543. doi: 10.1080/00207540310001619641.

[2] A. G. Mamalis, A. I. Grabchenko, A. V. Mitsyk, V. A. Fedorovich, and J. Kundrak. Mathematical simulation of motion of working medium at finishing–grinding treatment in the oscillating reservoir. *The International Journal of Advanced Manufacturing Technology*, 70(1-4):263--276, Jan 2014. ISSN 0268-3768. doi: 10.1007/s00170-013-5257-6. URL http://link.springer.com/10.1007/s00170-013-5257-6.

[3] D. Ciampini, M. Papini, and J. K. Spelt. Impact velocity measurement of media in a vibratory finisher. *Journal of Materials Processing Technology*, 183(2-3):347--357, 2007. ISSN 09240136. doi: 10.1016/j.jmatprotec.2006.10.024.

[4] S. Wanga, R. S. Timsit, and J. K. Spelt. Experimental investigation of vibratory finishing of aluminum. *Wear*, 243(1-2):147--156, 2000. ISSN 00431648. doi: 10.1016/S0043-1648(00)00437-3.

[5] Mohammad Saeid Emami Naeini. *Discrete Element Modeling of Granular Flows in Vibrationally-Fluidized Beds*. PhD thesis, Department of Mechanical and Industrial Engineering University of Toronto, 2011.

[6] Mikdam Jamal. *Characterisation and Evaluation of Thermally Treated Recycled Glass for Mass Finishing and Superfinishing Processes*. PhD thesis, Liverpool John Moores University, 2015.

[7] K L Johnson. Contact Mechanics, 1985. ISSN 07424787. URL http://www.amazon.fr/Contact-Mechanics-K-L-Johnson/dp/0521347963.

[8] B. J. Alder and T. E. Wainwright. Studies in Molecular Dynamics. I. General Method. *The Journal of Chemical Physics*, 31(2):459--466, 1959. ISSN 0021-9606. doi: 10.1063/1.1730376. URL http://aip.scitation.org/doi/10.1063/1.1730376.

[9] P A Cundall and O D L Strack. A discrete numerical model for granular assemblies. *Géotechnique*, 29(1):47--65, 1979. ISSN 0016-8505. doi: 10.1680/geot.1979.29.1.47. URL http://www.icevirtuallibrary.com/doi/10.1680/geot.1979.29.1.47.

[10] P. A. Langston, M. S. Nikitidis, U. Tüzün, D. M. Heyes, and N. M. Spyrou. Microstructural simulation and imaging of granular flows in two- and three-dimensional hoppers. *Powder Technology*, 94(1):59--72, 1997. ISSN 00325910. doi: 10.1016/S0032-5910(97)03288-9.

[11] James W. Landry, Gary S. Grest, and Steven J. Plimpton. Discrete element simulations of stress distributions in silos: Crossover from two to three dimensions. *Powder Technology*, 139(3): 233--239, 2004. ISSN 00325910. doi: 10.1016/j.powtec.2003.10.016.

[12] L. Liu, K.D. Kafui, and C. Thornton. Impact breakage of spherical, cuboidal and cylindrical agglomerates. *Powder Technology*, 199(2):189--196, Apr 2010. ISSN 00325910. doi: 10.1016/j.powtec.2010.01.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0032591010000264.

[13] A. Mehrotra, B. Chaudhuri, A. Faqih, M.S. Tomassone, and F.J. Muzzio. A modeling approach for understanding effects of powder flow properties on tablet weight variability. *Powder Technology*, 188(3):295--300, Jan 2009. ISSN 00325910. doi: 10.1016/j.powtec.2008.05.016. URL https://linkinghub.elsevier.com/retrieve/pii/S0032591008003331.

[14] P. W. Cleary, M. D. Sinnott, and R. D. Morrison. DEM prediction of particle flows in grinding processes. *International Journal for Numerical Methods in Fluids*, 58(3):319--353, Sep 2008. ISSN 02712091. doi: 10.1002/fld.1728. URL http://doi.wiley.com/10.1002/fld.1728.

[15] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics, 1995. ISSN 00219991.

[16] Christoph Kloss, Christoph Goniva, Alice Hager, Stefan Amberger, and Stefan Pirker. Models, algorithms and validation for opensource DEM and CFD-DEM. *Progress in Computational Fluid Dynamics, An International Journal*, 12(2/3):140, 2012. ISSN 1468-4349. doi: 10.1504/PCFD.2012.047457. URL http://www.inderscience.com/link.php?id=47457.

[17] Jun Ai, Jian Fei Chen, J. Michael Rotter, and Jin Y. Ooi. Assessment of rolling resistance models in discrete element simulations. *Powder Technology*, 206(3):269--282, 2011. ISSN 00325910. doi: 10.1016/j.powtec.2010.09.030.

[18] Jr Joseph Lupo. Tumbling Barrel, 1923.

[19] Fukuo Hashimoto, Hitomi Yamaguchi, Peter Krajnik, Konrad Wegener, Rahul Chaudhari, Hans Werner Hoffmeister, and Friedrich Kuster. Abrasive fine-finishing technology. *CIRP Annals - Manufacturing Technology*, 65(2):597--620, 2016. ISSN 17260604. doi: 10.1016/j.cirp.2016.06.003. URL http://dx.doi.org/10.1016/j.cirp.2016.06.003.

[20] F. Hashimoto and D. B. DeBra. Modelling and Optimization of Vibratory Finishing Process. *CIRP Annals - Manufacturing Technology*, 45(1):303--306, 1996. ISSN 17260604. doi: 10.1016/S0007-8506(07)63068-6.

[21] A. Yabuki, M. R. Baghbanan, and J. K. Spelt. Contact forces and mechanisms in a vibratory finisher. *Wear*, 252(7-8):635--643, 2002. ISSN 00431648. doi: 10.1016/S0043-1648(02)00016-9.

[22] J. Domblesky, V. Cariapa, and R. Evans. Investigation of vibratory bowl finishing. *International Journal of Production Research*, 41(16):3943--3953, 2003. ISSN 00207543. doi: 10.1080/0020754031000152550.

[23] Fukuo Hashimoto and Stephen P. Johnson. Modeling of vibratory finishing machines. *CIRP Annals - Manufacturing Technology*, 64(1):345--348, 2015. ISSN 17260604. doi: 10.1016/j.cirp.2015.04.004. URL http://dx.doi.org/10.1016/j.cirp.2015.04.004.

[24] Fukuo Hashimoto, Stephen P. Johnson, and Rahul G. Chaudhari. Modeling of material removal mechanism in vibratory finishing process. *CIRP Annals - Manufacturing Technology*, 65(1):325--328, 2016. ISSN 17260604. doi: 10.1016/j.cirp.2016.04.011. URL http://dx.doi.org/10.1016/j.cirp.2016.04.011.

[25] E. Uhlmann, A. Dethlefs, and A. Eulitz. Investigation into a geometry-based model for surface roughness prediction in vibratory finishing processes. *International Journal of Advanced Manufacturing Technology*, 75(5-8):815--823, 2014. ISSN 14333015. doi: 10.1007/s00170-014-6194-8.

[26] M Michaud, G Sroka, and L Winkelmann. Chemically Accelerated Vibratory Finishing for the Elimination of Wear and Pitting of Alloy Steel Gears Chemically Accelerated Vibratory Finishing for the. *Agma 01Ftm7*, 2001.

[27] Xiaozhong Song, Rahul Chaudhari, and Fukuo Hashimoto. Experimental Investigation of Vibratory Finishing Process. In *Volume 2: Processing*. American Society of Mechanical Engineers, Jun 2014. ISBN 978-0-7918-4581-3. doi: 10.1115/MSEC2014-4093. URL https://asmedigitalcollection.asme.org/MSEC/proceedings/MSEC2014/45813/Detroit,Michigan,USA/267295.

[28] C. R. Woodcock and J. S. Mason. *Bulk Solids Handling An Introduction to the Practice and Technology*. Blackie Academic & Professional, Glasgow, first edit edition, 1987. ISBN 9789401076890. doi: 10.1007/978-94-009-2635-6. URL http://doi.wiley.com/10.1002/9781444305449.

[29] Volkhard Buchholtz and Thorsten Pöschel. Numerical investigations of the evolution of sandpiles. *Physica A: Statistical Mechanics and its Applications*, 202(3-4):390--401, Jan 1994. ISSN 03784371. doi: 10.1016/0378-4371(94)90467-7. URL http://linkinghub.elsevier.com/retrieve/pii/0378437194904677.

[30] Gregory E. Amidon and Michael E. Houghton. The Effect of Moisture on the Mechanical and Powder Flow Properties of Microcrystalline Cellulose, 1995. ISSN 1573904X.

[31] Hojae Yi, B. Mittal, V. M. Puri, A. S. MCNitt, and C. F. Mancino. Measurement of Bulk Mechanical Properties and Modeling the Load Response of Rootzone Sands. Part 2: Effect of Moisture on Continuous Sand Mixtures. *Particulate Science and Technology*, 20(2):125--157, 2002. ISSN 0272-6351. doi: 10.1080/02726350215335. URL http://www.tandfonline.com/doi/abs/10.1080/02726350215335{%}5Cnhttp://www.tandfonline.com/doi/abs/10.1080/02726350215335{#}.U1GRD1cvmSo.

[32] Masanobu Oda and Kazuyoshi Iwashita. Study on couple stress and shear band development in granular media based on numerical simulation analyses. *International Journal of Engineering Science*, 2000. ISSN 00207225. doi: 10.1016/S0020-7225(99)00132-9.

[33] A. Dubey. *Powder flow and blending*. Elsevier Ltd, 2017. ISBN 9780081001806. doi: 10.1016/B978-0-08-100154-7.00003-X. URL http://dx.doi.org/10.1016/B978-0-08-100154-7.00003-X.

[34] Raj Mukherjee, Chen Mao, Sayantan Chattoraj, and Bodhisattwa Chaudhuri. DEM based computational model to predict moisture induced cohesion in pharmaceutical powders. *International Journal of Pharmaceutics*, 536(1):301--309, 2018. ISSN 18733476. doi: 10.1016/j.ijpharm.2017.12.001. URL https://doi.org/10.1016/j.ijpharm.2017.12.001.

[35] P. Bhalode and Marianthi Ierapetritou. *Discrete Element Modeling (DEM) Parametric Study of Feeder Unit in Continuous Pharmaceutical Industry*, volume 47. Elsevier Masson SAS, 2019. ISBN 9780128185971. doi: 10.1016/B978-0-12-818597-1.50054-0. URL https://doi.org/10.1016/B978-0-12-818597-1.50054-0.

[36] Paul W. Cleary and Mark L. Sawley. DEM modelling of industrial granular flows: 3D case studies and the effect of particle shape on hopper discharge. *Applied Mathematical Modelling*, 26(2): 89--111, 2002. ISSN 0307904X. doi: 10.1016/S0307-904X(01)00050-6.

[37] Franz Kessler and Michael Prenner. DEM - Simulation of conveyor transfer chutes. *FME Transactions*, 37(4):185--192, 2009. ISSN 2406128X.

[38] Nicolin Govender, Raj K. Rajamani, Schalk Kok, and Daniel N. Wilke. Discrete element simulation of mill charge in 3D using the BLAZE-DEM GPU framework. *Minerals Engineering*, 79:152--168, 2015. ISSN 08926875. doi: 10.1016/j.mineng.2015.05.010. URL http://dx.doi.org/10.1016/j.mineng.2015.05.010.

[39] J. T. Kalala, M. Bwalya, and M. H. Moys. Discrete element method (DEM) modelling of evolving mill liner profiles due to wear. Part II. Industrial case study. *Minerals Engineering*, 18(15): 1392--1397, 2005. ISSN 08926875. doi: 10.1016/j.mineng.2005.02.010.

[40] Akbar Jafari and Vahid Saljooghi Nezhad. Employing DEM to study the impact of different parameters on the screening efficiency and mesh wear. *Powder Technology*, 297:126--143, 2016. ISSN 1873328X. doi: 10.1016/j.powtec.2016.04.008. URL http://dx.doi.org/10.1016/j.powtec.2016.04.008.

[41] Franco Perazzo, Rainald Löhner, Fernando Labbe, Frederik Knop, and Patricio Mascaró. Numerical modeling of the pattern and wear rate on a structural steel plate using DEM. *Minerals Engineering*, 137(April):290--302, 2019. ISSN 08926875. doi: 10.1016/j.mineng.2019.04.012. URL https://doi.org/10.1016/j.mineng.2019.04.012.

[42] Józef Horabik, Piotr Parafiniuk, and Marek Molenda. Experiments and discrete element method simulations of distribution of static load of grain bedding at bottom of shallow model silo. *Biosystems Engineering*, 149:60--71, 2016. ISSN 15375110. doi: 10.1016/j.biosystemseng.2016.06.012.

[43] C. González-Montellano, Á Ramírez, E. Gallego, and F. Ayuga. Validation and experimental calibration of 3D discrete element models for the simulation of the discharge flow in silos. *Chemical Engineering Science*, 66(21):5116--5126, 2011. ISSN 00092509. doi: 10.1016/j.ces.2011.07.009.

[44] C. González-Montellano, A. Ramírez, J. M. Fuentes, and F. Ayuga. Numerical effects derived from en masse filling of agricultural silos in DEM simulations. *Computers and Electronics in Agriculture*, 81:113--123, 2012. ISSN 01681699. doi: 10.1016/j.compag.2011.11.013.

[45] Darius Markauskas, Álvaro Ramírez-Gómez, Rimantas Kačianauskas, and Evaldas Zdancevičius. Maize grain shape approaches for DEM modelling. *Computers and Electronics in Agriculture*, 118:247--258, 2015. ISSN 01681699. doi: 10.1016/j.compag.2015.09.004.

[46] Mustafa Ucgul, Chris Saunders, and John M. Fielke. Discrete element modelling of tillage forces and soil movement of a one-third scale mouldboard plough. *Biosystems Engineering*, 155:44--54, 2017. ISSN 15375110. doi: 10.1016/j.biosystemseng.2016.12.002. URL http://dx.doi.org/10.1016/j.biosystemseng.2016.12.002.

[47] Bhupendra M. Ghodki and T. K. Goswami. DEM simulation of flow of black pepper seeds in cryogenic grinding system. *Journal of Food Engineering*, 196:36--51, 2017. ISSN 02608774. doi: 10.1016/j.jfoodeng.2016.09.026. URL http://dx.doi.org/10.1016/j.jfoodeng.2016.09.026.

[48] Jannatul Azmir, Qinfu Hou, and Aibing Yu. CFD-DEM study of the effects of food grain properties on drying and shrinkage in a fluidised bed. *Powder Technology*, 360:33--42, 2020. ISSN 1873328X. doi: 10.1016/j.powtec.2019.10.021. URL https://doi.org/10.1016/j.powtec.2019.10.021.

[49] D. Höhner, S. Wirtz, and V. Scherer. A study on the influence of particle shape and shape approximation on particle mechanics in a rotating drum using the discrete element method. *Powder Technology*, 253:256--265, 2014. ISSN 00325910. doi: 10.1016/j.powtec.2013.11.023. URL http://dx.doi.org/10.1016/j.powtec.2013.11.023.

[50] O. O. Ayeni, C. L. Wu, J. B. Joshi, and K. Nandakumar. A discrete element method study of granular segregation in non-circular rotating drums. *Powder Technology*, 283:549--560, 2015. ISSN 1873328X. doi: 10.1016/j.powtec.2015.06.038.

[51] Rahul K. Soni, Rahul Mohanty, Swati Mohanty, and B. K. Mishra. Numerical analysis of mixing of particles in drum mixers using DEM. *Advanced Powder Technology*, 27(2):531--540, 2016. ISSN 15685527. doi: 10.1016/j.apt.2016.01.016. URL http://dx.doi.org/10.1016/j.apt.2016.01.016.

[52] Xiaoyan Liu, Zhou Hu, Weining Wu, Jiesi Zhan, Fabian Herz, and Eckehard Specht. DEM study on the surface mixing and whole mixing of granular materials in rotary drums. *Powder Technology*, 315:438--444, 2017. ISSN 1873328X. doi: 10.1016/j.powtec.2017.04.036. URL http://dx.doi.org/10.1016/j.powtec.2017.04.036.

[53] Wei Ning Wu, Xiao Yan Liu, Rong Zhang, and Zhou Hu. DEM investigation of the power draw for material movement in rotary drums with axis offset. *Chemical Engineering Research*

and Design, 144:310--317, 2019. ISSN 02638762. doi: 10.1016/j.cherd.2019.02.011. URL https://doi.org/10.1016/j.cherd.2019.02.011.

[54] Esmaeil Yazdani and Seyed Hassan Hashemabadi. The influence of cohesiveness on particulate bed segregation and mixing in rotating drum using DEM. *Physica A: Statistical Mechanics and its Applications*, 525:788--797, Jul 2019. ISSN 03784371. doi: 10.1016/j.physa.2019.03.127. URL https://doi.org/10.1016/j.physa.2019.03.127.

[55] Jirí Zegzulka Hlosta, L. Jezerská, Jirí Rozbroj, David Žurovec, Jan Necas. DEM Investigation of the Influence of Particulate Properties and Operating Conditions on the Mixing and Experimental Study. *Processes*, 2020.

[56] Eric J.R. Parteli and Thorsten Pöschel. Particle-based simulation of powder application in additive manufacturing. *Powder Technology*, 288:96--102, 2016. ISSN 1873328X. doi: 10.1016/j.powtec.2015.10.035. URL http://dx.doi.org/10.1016/j.powtec.2015.10.035.

[57] Yufan Zhao, Yuichiro Koizumi, Kenta Aoyagi, Kenta Yamanaka, and Akihiko Chiba. Characterization of powder bed generation in electron beam additive manufacturing by discrete element method (DEM). *Materials Today: Proceedings*, 4(11):11437--11440, 2017. ISSN 22147853. doi: 10.1016/j.matpr.2017.09.023. URL https://doi.org/10.1016/j.matpr.2017.09.023.

[58] S. Haeri. Optimisation of blade type spreaders for powder bed preparation in Additive Manufacturing using DEM simulations. *Powder Technology*, 321:94--104, 2017. ISSN 1873328X. doi: 10.1016/j.powtec.2017.08.011. URL http://dx.doi.org/10.1016/j.powtec.2017.08.011.

[59] Ruihuan Ge, Mojtaba Ghadiri, Tina Bonakdar, Qijun Zheng, Zongyan Zhou, Ian Larson, and Karen Hapgood. Deformation of 3D Printed Agglomerates: Multiscale Experimental Tests and DEM Simulation. *Chemical Engineering Science*, 217:115526, 2020. ISSN 00092509. doi: 10.1016/j.ces.2020.115526. URL https://doi.org/10.1016/j.ces.2020.115526.

[60] V. Vijayaraghavan, S. Castagne, S. Srivastava, and C. Z. Qin. State-of-the-art in experimental and numerical modeling of surface characterization of components in mass finishing process. *International Journal of Advanced Manufacturing Technology*, 90(9-12):2885--2899, 2017. ISSN 14333015. doi: 10.1007/s00170-016-9595-z.

[61] Vikram Cariapa, Hyunjae Park, Jongsoo Kim, Cunjiang Cheng, and Antonio Evaristo. Development of a metal removal model using spherical ceramic media in a centrifugal disk

mass finishing machine. *International Journal of Advanced Manufacturing Technology*, 39(1-2): 92--106, 2008. ISSN 02683768. doi: 10.1007/s00170-007-1195-5.

[62] B. Mullany, H. Shahinian, J. Navare, F. Azimi, E. Fleischhauer, P. Tkacik, and R. Keanini. The application of computational fluid dynamics to vibratory finishing processes. *CIRP Annals - Manufacturing Technology*, 66(1):309--312, 2017. ISSN 17260604. doi: 10.1016/j.cirp.2017.04.087. URL http://dx.doi.org/10.1016/j.cirp.2017.04.087.

[63] S. E. Naeini and J. K. Spelt. Two-dimensional discrete element modeling of a spherical steel media in a vibrating bed. *Powder Technology*, 195(2):83--90, 2009. ISSN 00325910. doi: 10.1016/j.powtec.2009.05.016. URL http://dx.doi.org/10.1016/j.powtec.2009.05.016.

[64] Eckart Uhlmann, Arne Dethlefs, and Alexander Eulitz. Investigation of Material Removal and Surface Topography Formation in Vibratory Finishing. *Procedia CIRP*, 14:25--30, 2014. ISSN 22128271. doi: 10.1016/j.procir.2014.03.048. URL http://linkinghub.elsevier.com/retrieve/pii/S2212827114001917.

[65] F. Salvatore, F. Grange, R. Kaminski, C. Claudin, G. Kermouche, J. Rech, and A. Texier. Experimental and Numerical Study of Media Action during Tribofinishing in the Case of SLM Titanium Parts. *Procedia CIRP*, 58:451--456, 2017. ISSN 22128271. doi: 10.1016/j.procir.2017.03.251. URL http://dx.doi.org/10.1016/j.procir.2017.03.251.

[66] Young Sup Kang, Fukuo Hashimoto, Stephen P. Johnson, and Jerry P. Rhodes. Discrete element modeling of 3D media motion in vibratory finishing process. *CIRP Annals - Manufacturing Technology*, 66(1):313--316, 2017. ISSN 17260604. doi: 10.1016/j.cirp.2017.04.092. URL http://dx.doi.org/10.1016/j.cirp.2017.04.092.

[67] Xiuzhi Wang, Shengqiang Yang, Wenhui Li, and Yanqing Wang. Vibratory finishing co-simulation based on ADAMS-EDEM with experimental validation. *International Journal of Advanced Manufacturing Technology*, 96(1-4):1175--1185, 2018. ISSN 14333015. doi: 10.1007/s00170-018-1639-0.

[68] Yusuke Makiuchi, F. Hashimoto, and Anthony Beaucamp. Model of material removal in vibratory finishing, based on Preston's law and discrete element method. *CIRP Annals*, 68(1):365--368, 2019. ISSN 17260604. doi: 10.1016/j.cirp.2019.04.082. URL https://doi.org/10.1016/j.cirp.2019.04.082.

[69] Eckart Uhlmann, Alexander Eulitz, and Arne Dethlefs. Discrete element modelling of drag finishing. *Procedia CIRP*, 31:369--374, 2015. ISSN 22128271. doi: 10.1016/j.procir.2015.03.021. URL http://dx.doi.org/10.1016/j.procir.2015.03.021.

[70] Wenhui Li, Li Zhang, Xiuhong Li, Shengqiang Yang, and Fengfeng Wu. Theoretical and simulation analysis of abrasive particles in centrifugal barrel finishing: Kinematics mechanism and distribution characteristics. *Powder Technology*, 318:518--527, 2017. ISSN 1873328X. doi: 10.1016/j.powtec.2017.06.033. URL http://dx.doi.org/10.1016/j.powtec.2017.06.033.

[71] Frederik Zanger, Andreas Kacaras, Patrick Neuenfeldt, and Volker Schulze. Optimization of the stream finishing process for mechanical surface treatment by numerical and experimental process analysis. *CIRP Annals*, 68(1):373--376, 2019. ISSN 17260604. doi: 10.1016/j.cirp.2019.04.086. URL https://doi.org/10.1016/j.cirp.2019.04.086.

[72] Paweł Sutowski, Jarosław Plichta, and Paweł Kałduński. Determining kinetic energy distribution of the working medium in a centrifugal disc finishing process—part 1: theoretical and numerical analysis with DEM method. *International Journal of Advanced Manufacturing Technology*, 104 (1-4):687--704, 2019. ISSN 14333015. doi: 10.1007/s00170-019-03937-2.

[73] Paweł Sutowski, Jarosław Plichta, and Paweł Kałduński. Determining kinetic energy distribution of the working medium in a centrifugal disc finishing process—part 2: experimental analysis with the use of acoustic emission signal. *International Journal of Advanced Manufacturing Technology*, 104(1-4):687--704, 2019. ISSN 14333015. doi: 10.1007/s00170-019-03937-2.

[74] C.E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10--21, Jan 1949. ISSN 0096-8390. doi: 10.1109/JRPROC.1949.232969. URL http://ieeexplore.ieee.org/document/1697831/.

[75] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309--1331, Sep 2009. ISSN 00295981. doi: 10.1002/nme.2579. URL http://doi.wiley.com/10.1002/nme.2579.

[76] Jean Claude Barré de Saint-Venan. Mémoire sur la torsion des prismes, avec des considérations sur leur flexion, 1855.

Appendix A

Friction code

```
### LOADING PACKAGES
import pandas as pd
import numpy as np
import csv
import math
import glob
### DEFINING COEFFICIENTO OF FRICTION TO CALCULATE
cof = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
### READING DATA OBTAINED FROM SIMULATIONS
working_directory = "/foo" + "/"
s = "output*.csv"
filenames = glob.glob(working_directory + s)
### POST-PROCESSING DATA OBTAINED FROM SIMULATIONS
slidingAngle = []
for filename in filenames:
    inputData = pd.read_csv(filename, usecols=[1, 2, 6, 7, 8])
    velocity = []
    time = []
    angle = []
    for i in range(len(inputData)):
        vel = math.sqrt(inputData.iloc[i, 2]**2 + inputData.iloc[i, 3]**2 +
                        inputData.iloc[i, 4]**2)
        t = inputData.iloc[i, 0]
        ang = inputData.iloc[i, 1]
        velocity.append(vel)
        time.append(t)
        angle.append(ang)

```

```
    outputData = pd.DataFrame(
        {"time(s)" : time,
         "angle(deg)": angle,
         "velocity(m/s)": velocity
        })

    threshold = 0.0001
    for i in range(len(outputData)):
        a = outputData.iloc[0, 2]
        b = outputData.iloc[i, 2]
        if (i+1) >= len(outputData):
            break
        elif math.fabs(a - b) >= threshold:
            angLimit = outputData.iloc[i, 0]
            slidingAngle.append(angLimit)
            break
### RETRIVING SLIDING ANGLE
initialSlidingAngle = pd.DataFrame(
        {"cof" : cof,
         "angle(deg)": slidingAngle
        })
### EXPORTING SLIDING ANGLE
initialSlidingAngle.to_csv("foo.csv", index=False)
```

Appendix B

Vibrational studies

B.1 Matlab signal-analysis code

```
% This script has been develop to analyse the vibration of the trough for
% different loading states.
%
% Three ortogonal axis were independently studied:
%     − x−axis comprehends the longitudinal direction of the  trough.
%     − y−axis comprehends the vertical direction of the trough.
%     − z−axis comprehends the transversal direction of the trough.

%% Analysis in the Time Domain.
%% Loading data
load('Accempty1.mat');
load('Acc25kg1.mat');
load('Acc50kg1.mat');
load('Acc75kg1.mat');
load('Acc100kg1.mat');

%% Empty trough analysis
Ts0 = Accempty1(:,1);
Acc{1} = Accempty1(:,3);%x4
Acc{2} = Accempty1(:,4);%y2
Acc{3} = Accempty1(:,2);%z3

ft{1} = fit(Ts0,Acc{1},'fourier8');
freq_ft0X = ft{1}.w;
period_ft0X = (2*pi)/freq_ft0X;
```

```
ft{2} = fit(Ts0,Acc{2},'fourier8');
freq_ft0Y = ft{2}.w;
period_ft0Y = (2*pi)/freq_ft0Y;

ft{3} = fit(Ts0,Acc{3},'fourier8');
freq_ft0Z = ft{3}.w;
period_ft0Z = (2*pi)/freq_ft0Z;

%% 25kg Load analysis
Ts25 = Acc25kg1(:,1);
Acc{4} = Acc25kg1(:,3);
Acc{5} = Acc25kg1(:,4);
Acc{6} = Acc25kg1(:,2);

ft{4} = fit(Ts25,Acc{4},'fourier8');
freq_ft25X = ft{4}.w;
period_ft25X = (2*pi)/freq_ft25X;

ft{5} = fit(Ts25,Acc{5},'fourier8');
freq_ft25Y = ft{5}.w;
period_ft25Y = (2*pi)/freq_ft25Y;

ft{6} = fit(Ts25,Acc{6},'fourier8');
freq_ft25Z = ft{6}.w;
period_ft25Z = (2*pi)/freq_ft25Z;

%% 50kg Load analysis
Ts50 = Acc50kg1(:,1);
Acc{7} = Acc50kg1(:,3);
Acc{8} = Acc50kg1(:,4);
Acc{9} = Acc50kg1(:,2);

ft{7} = fit(Ts50,Acc{7},'fourier8');
freq_ft50X = ft{7}.w;
period_ft50X = (2*pi)/freq_ft50X;

ft{8} = fit(Ts50,Acc{8},'fourier8');
freq_ft50Y = ft{8}.w;
period_ft50Y = (2*pi)/freq_ft50Y;

```

```
ft{9} = fit(Ts50,Acc{9},'fourier8');
freq_ft50Z = ft{9}.w;
period_ft50Z = (2*pi)/freq_ft50Z;

%% 75kg Load analysis
Ts75 = Acc75kg1(:,1);
Acc{10} = Acc75kg1(:,3);
Acc{11} = Acc75kg1(:,4);
Acc{12} = Acc75kg1(:,2);

ft{10} = fit(Ts75,Acc{10},'fourier8');
freq_ft75X = ft{10}.w;
period_ft75X = (2*pi)/freq_ft75X;

ft{11} = fit(Ts75,Acc{11},'fourier8');
freq_ft75Y = ft{11}.w;
period_ft75Y = (2*pi)/freq_ft75Y;

ft{12} = fit(Ts75,Acc{12},'fourier8');
freq_ft75Z = ft{12}.w;
period_ft75Z = (2*pi)/freq_ft75Z;

%% 100kg Load analysis
Ts100 = Acc100kg1(:,1);
Acc{13} = Acc100kg1(:,3);
Acc{14} = Acc100kg1(:,4);
Acc{15} = Acc100kg1(:,2);

ft{13} = fit(Ts100,Acc{13},'fourier8');
freq_ft100X = ft{13}.w;
period_ft100X = (2*pi)/freq_ft100X;
coeffsX = coeffvalues(ft{13});
limitX = length(coeffsX)-1;
Ux = coeffsX(2:limitX)'/freq_ft100X^2;

ft{14} = fit(Ts100,Acc{14},'fourier8');
freq_ft100Y= ft{14}.w;
period_ft100Y = (2*pi)/freq_ft100Y;
coeffsY = coeffvalues(ft{14});
limitY = length(coeffsY)-1;
Uy = coeffsY(2:limitY)'/freq_ft100Y^2;
```

```

ft{15} = fit(Ts100,Acc{15},'fourier8');
freq_ft100Z= ft{15}.w;
period_ft100Z = (2*pi)/freq_ft100Z;
coeffsZ = coeffvalues(ft{15});
limitZ = length(coeffsZ)-1;
Uz = coeffsZ(2:limitZ)'/freq_ft100Z^2;

%% Writting the imput vibrational parameters for LIGGGHTS (100kg Load)
txtfile = fopen('scriptInput.txt','w');

for k = 1:size(Ux)
   txtX = 'fix harmonicX%d  all move/mesh mesh wall wiggle amplitude  %e\t      0\t        0\t      period %e \n';
   fprintf(txtfile,txtX,k,Ux(k), period_ft100Y);
   txtY = 'fix harmonicY%d  all move/mesh mesh wall wiggle amplitude  0\t      %e\t        0\t      period %e \n';
   fprintf(txtfile,txtY,k,Uy(k), period_ft100Z);
   txtZ = 'fix harmonicZ%d  all move/mesh mesh wall wiggle amplitude  0\t      0\t        %e\t      period %e \n\n';
   fprintf(txtfile,txtZ,k,Uz(k), period_ft100X);
end

for k = size(Ux):-1:1
   txt1 = 'unfix harmonicZ%d\n';
   fprintf(txtfile,txt1,k);
   txt2 = 'unfix harmonicY%d\n';
   fprintf(txtfile,txt2,k);
   txt3 = 'unfix harmonicX%d\n';
   fprintf(txtfile,txt3,k);
end

fclose(txtfile);

%% Plotting results for all loads and directions (Time Domain)
for i = 1 : length(ft)
    % set line colors
    if i == 1 || i == 4 || i == 7 ||i == 10 ||i == 13
        C = [0 0.4470 0.7410];
        yAxisLabel = '$\ddot{x} \quad (m/s^2)$';
    elseif i == 2 || i == 5 || i == 8 || i == 11 || i == 14
        yAxisLabel = '$\ddot{y} \quad (m/s^2)$';
        C =[0.4660 0.6740 0.1880];
    elseif i == 3 || i == 6 || i == 9 || i == 12 || i == 15
```

```
        yAxisLabel = '$\ddot{z} \quad (m/s^2)$';
        C = [0.4940 0.1840 0.5560];
    end
    % set Legends
    if i == 1 || i == 2 || i == 3
        dataLegend = 'Unload';
    elseif i == 4 || i == 5 || i == 6
        dataLegend = 'Load = 25';
    elseif i == 7 || i == 8 || i == 8
        dataLegend = 'Load = 50';
    elseif i == 10 || i == 11 || i == 12
        dataLegend = 'Load = 75';
    elseif i == 13 || i == 14 || i == 15
        dataLegend = 'Load = 100';
    end
    saveImage(ft{i},Ts0,Acc{i},i, C, dataLegend, yAxisLabel)
end

%% Analysis in the Frequency Domain.
%% Loading data
load('Frq100kg1.mat');
load('Frq100kg2.mat');

%% 100kg
freq100_R1 =  Frq100kg1([1:150],1);
accFreq_R1{1} = Frq100kg1([1:150],3);
accFreq_R1{2} = Frq100kg1([1:150],4);
accFreq_R1{3} = Frq100kg1([1:150],2);

for  i = 1 : length(accFreq_R1)
    if i == 1
        C = [0 0.4470 0.7410];
        yAxisLabel = '$\ddot{x} \quad (m/s^2)$';
        dataLengend = 'Frequency x-axis';
    elseif i == 2
        yAxisLabel = '$\ddot{y} \quad (m/s^2)$';
        C =[0.4660 0.6740 0.1880];
        dataLengend = 'Frequency y-axis';
    elseif i == 3
        yAxisLabel = '$\ddot{z} \quad (m/s^2)$';
        C = [0.4940 0.1840 0.5560];
```

```
        dataLengend = 'Frequency z-axis';
    end
    saveImageFreq(freq100_R1, accFreq_R1{i}, i, C, dataLengend, yAxisLabel)

end

% Finding maximum and minimum values
maxAccX = max(accFreq_R1{1});
minAccX = maxAccX * 0.05;
[accX, freqX] = findpeaks(accFreq_R1{1},freq100_R1(),'MinPeakHeight',minAccX);

maxAccY = max(accFreq_R1{2});
minAccY = maxAccY * 0.05;
[accY, freqY] = findpeaks(accFreq_R1{2},freq100_R1(),'MinPeakHeight',minAccY);

maxAccZ = max(accFreq_R1{3});
minAccZ = maxAccZ * 0.05;
[accZ, freqZ] = findpeaks(accFreq_R1{3},freq100_R1(),'MinPeakHeight',minAccZ);
%% Defining LIGGGHTS vibrational input
% Finding amplitude(m) in the x-, y- and z-axis direction and the period
ux = rdivide(accX , (2*pi*freqX).^2);
tx = rdivide(1,freqX);

uy = rdivide(accY , (2*pi*freqY).^2);
ty = rdivide(1,freqY);

uz = rdivide(accZ , (2*pi*freqZ).^2);
tz = rdivide(1,freqZ);

% Writing LIGGGHTS input script
txtfile = fopen('scriptInput2.txt','w');

txth1 = '\n# Values for the vibration in X-direction\n';
fprintf(txtfile,txth1);
for k = 1:size(ux)
   txtX = 'fix harmonicX%d  all move/mesh mesh wall wiggle amplitude  %e\t      0\t        0\t      period %e \n';
   fprintf(txtfile,txtX,k,ux(k), tx(k));
end

txth2 = '\n# Values for the vibration in Y-direction\n';
fprintf(txtfile,txth2);
```

```

for k = 1:size(uy)
   txtY = 'fix harmonicY%d  all move/mesh mesh wall wiggle amplitude  0\t      %e\t        0\t      period %e \n';
   fprintf(txtfile,txtY,k,uy(k), ty(k));
end

txth3 = '\n# Values for the vibration in Z-direction\n';
fprintf(txtfile,txth3);

for k = 1:size(uz)
   txtZ = 'fix harmonicZ%d  all move/mesh mesh wall wiggle amplitude  0\t      0\t         %e\t     period %e \n';
   fprintf(txtfile,txtZ,k,uz(k), tz(k));
end

txth4 = '\n# Mesh stop vibrating\n';
fprintf(txtfile,txth4);

for k = size(uz):-1:1
   txt1 = 'unfix harmonicZ%d\n';
   fprintf(txtfile,txt1,k);
end

for k = size(uy):-1:1
   txt2 = 'unfix harmonicY%d\n';
   fprintf(txtfile,txt2,k);
end

for k = size(ux):-1:1
   txt3 = 'unfix harmonicX%d\n';
   fprintf(txtfile,txt3,k);
end
```

B.2 Functions

```
function saveImage(F, X1, Y1, P, C, dataLegend, yAxisLabel)

%saveImage(F, X1, Y1, P, C)
% F:   Fit data
% X1:  Var of time data
% Y1:  Var of acceleration data
```

```
% P:  File consecutive number
% C:  Color's code specification
% dataLegend: Legend specification
% yAxisLabel: y-axis label specification


% Change default axes fonts.
set(0,'DefaultAxesFontName', 'Verdana')
set(0,'DefaultAxesFontSize', 30)

% Change default text fonts.
set(0,'DefaultTextFontname', 'Verdana')
set(0,'DefaultTextFontSize', 30)

% Path to folder
path = '/media/arnon/Memoria/PhD/Thesis/graph/png/vibratory-png/';
fileNumber = num2str(P);
fileName = 'plot_';
file = strcat(path, fileName, fileNumber);
% Create figure
fg1 = figure;
p1 = plot(F,X1,Y1);
p1(1).LineWidth = 1;
p1(1).LineStyle = '--';
p1(1).Marker = 'none';
p1(1).Color = C;
p1(2).LineWidth = 4;
xlim([0 .04]);xlabel('$\mbox{time} \quad  (s)$','Interpreter','latex');ylabel(yAxisLabel,'Interpreter','latex');legend(
    dataLegend);

% set all units inside figure to normalized so that everything is scaling accordingly
 set(findall(fg1,'Units','pixels'),'Units','normalized');
 % do not show figure on screen
 set(fg1, 'visible', 'off')
 % set figure units to pixels & adjust figure size
 fg1.Units = 'pixels';
 fg1.OuterPosition = [0 0 2560 1440];
 % define resolution figure to be saved in dpi
 res = 200;
 % recalculate figure size to be saved
 set(fg1,'PaperPositionMode','manual')
```

```
fg1.PaperUnits = 'inches';
fg1.PaperPosition = [0 0 2560 1440]/res;
% save figure
print(fg1,file,'−dpng',sprintf('−r%d',res))
end
```

B.3 Slurm standard script

```
#!/bin/bash
# This line tells the shell how to execute this script, and is unrelated
# to SLURM.
# at the beginning of the script, lines beginning with "#SBATCH" are read by
# SLURM and used to set queueing options. You can comment out a SBATCH
# directive with a second leading #, eg:
##SBATCH −−nodes=1
# Project/Account (use your own)
#SBATCH −A Induction_JOB.%j
# we need 1 node, will launch a maximum of one task and use one cpu for the task:
#SBATCH −−nodes=8
##SBATCH −−ntasks−per−node=28
##SBATCH −−cpus−per−task=28
# we expect the job to finish within 10 hours. If it takes longer than 5
# hours, SLURM can kill it:
#SBATCH −−time=24:00:00
# we want the job to be named "myTest" rather than something generated
# from the script name. This will affect the name of the job as reported
# by squeue:
#SBATCH −−job−name=VM375YSlice1
# when the job ends, send me an email at this email address.
#SBATCH −−mail−type=END
#SBATCH −−mail−user= ArnonLopez.Marrero@the−mtc.org
# both standard output and standard error are directed to the same file.
# It will be placed in the directory I submitted the job from and will
# have a name like slurm_12345.out
##SBATCH −−output=slurm_%j.out
#SBATCH −o process_%j.out
#SBATCH −e process_%j.err
# once the first non−comment, non−SBATCH−directive line is encountered, SLURM
# stops looking for SBATCH directives. The remainder of the script is  executed
# as a normal Unix shell script
```

```
# first we append the folder containing the custom Module:
module use --append ~/ModuleFiles/
# and load the module for the software we are using:
module load  VTK.8.0 liggghts.3.8.0 mpich/ge/gcc/64/3.2
# next we create a unique directory to run this job in. We will record its
# name in the shell variable "RUNDIR", for better readability.
# SLURM sets SLURM_JOB_ID to the job id, ${SLURM_JOB_ID/.*} expands to the job
# id up to the first '.' We make the run directory in our area under $SCRATCH, because at NYU HPC
# $SCRATCH is configured for the disk space and speed required by HPC jobs.
RUNDIR=$HOME/JOBS/${SLURM_JOB_NAME/.*}/output/run-${SLURM_JOB_ID/.*}
mkdir -p  $RUNDIR
echo "The output for the ${SLURM_JOB_NAME}-${SLURM_JOB_ID} can be found in $RUNDIR"
# we will be reading data in from somewhere, so define that too:
#DATADIR=$HOME/JOBS/${SLURM_JOB_NAME/.*}/input
DATADIR=$HOME/JOBS/VM375YSlice1/input
# the script will have started running in $HOME, so we need to move into the
# unique directory we just created
cd $RUNDIR
# define variable with the name of the script to be run
# script="/home/MMEALOPE/JOBS/firstJob/Drag_HPC/in.drag_HPC"
input_file="$DATADIR/in.VM375Y_step01"
echo "The input script for this job is ${input_file}"
# run the executable (sending the contents of my_input_params.inp to stdin)
mpirun -np 280 /home/MMEALOPE/Applications/LIGGGHTS-PUBLIC/src/lmp_auto < $input_file
```

B.4 LIGGGHTS script

```
### 3D VM 375 Y TROUGH
### Arnon Lopez Marrero (General Engineering Research Institute)
### ArnonLopez.Marrero@the-mtc.org
### FILE: in.VM375Y_Case1
###

# This script is designed to study the bulk within the trough.
# Firstly, the trough is loaded with the general purpose media (green media),
# then the work-chamber stars vibrating.

# Description:
# There are two types of Materials.
# 1. Media material, plastic.
```

```
# 2. Lining Material (Elastomer Polyurethane)
#

processors 2 20 7 # To be modified as needed

### INPUT PARAMETERS Input Parameters
include         PRIMES.inc
include         in.DEM
### GENERAL SETTINGS FOR GRANULAR SYSTEMS
atom_style      granular
atom_modify     map array
boundary        p f f
newton          off
echo            both

communicate     single vel yes

units           si
hard_particles  yes
soft_particles  yes
### REGION DEFINITION
region          reg block ${xlo} ${xhi} ${ylo} ${yhi} ${zlo} ${zhi}  units box
create_box      3 reg
neighbor        0.00432 bin
neigh_modify    delay 0

group           bulk initialize
group           sensor initialize
group           media initialize
### MATERIAL PROPERTIES
fix         m1 all property/global youngsModulus peratomtype &
                            ${youngsModP1} ${youngsModP2} ${youngsModW1}
fix         m2 all property/global poissonsRatio peratomtype &
                            ${poissionsRatP1} ${poissionsRatP2} ${poissionsRatW1}
fix         m3 all property/global coefficientRestitution peratomtypepair 3        &
                            ${coeffRestP1P1} ${coeffRestP1P2} ${coeffRestP1W1} &
                            ${coeffRestP1P2} ${coeffRestP2P2} ${coeffRestP2W1} &
                            ${coeffRestP1W1} ${coeffRestP2W1} ${coeffRestNuLL} ##
fix         m4 all property/global coefficientFriction peratomtypepair 3        &
                            ${coeffFricP1P1} ${coeffFricP1P2} ${coeffFricP1W1} &
                            ${coeffFricP1P2} ${coeffFricP2P2} ${coeffFricP2W1} &
```

```
                                ${coeffFricP1W1} ${coeffFricP2W1} ${coeffFricNuLL} ##
### PAIR STYLE
pair_style    gran model hertz tangential history cohesion off #Hertzian without cohesion
pair_coeff    * *
### TIME-STEP
timestep      ${timeStep}
### WALLS
fix           wall  all mesh/surface/stress file Models/VM375YSlice.stl  type 3 curvature 1e-5 stress on
fix           geometry all wall/gran model hertz tangential history cohesion off  mesh n_meshes 1 meshes wall
### GRAVITY
fix           gravi all gravity 9.81 vector 0.0 -1.0 0.0
### PARTICLE DISTRIBUTION AND INSERTION
region        INS1 block -0.05   0.05  0.35  0.75  -0.24  0.24  units box
fix           media_tpt bulk particletemplate/multisphere ${prime1} atom_type 1 density constant ${mediaDensity} &
              nspheres 65 ntry 100000 spheres file Models/65SphereGreenPlasticPolisher scale 1.0 type 1
fix           media_distro media particledistribution/discrete/numberbased ${prime3}  1 media_tpt 1.0 ##
#for 7kg use the variable  ${noMediaParticles}
fix           media_ins media insert/rate/region seed ${prime5} distributiontemplate media_distro    &
              nparticles ${noMediaParticles} massrate 40 insert_every ${insertStep} all_in yes vel        &
              constant 0.0 -2.0 0.0 region INS1 ##
### APPLY MULTISPHERE INTEGRATION
fix             int1 bulk multisphere allow_group_and_set yes
### CALCULATE THE MASS OF INDIVIDUAL MULTI-SPHERE BODIES (clumps)
compute       massms all multisphere property masstotal #body mass (1 value)
variable      massCase equal c_massms
### OUTPUT SETTINGS
variable      m equal mass(all)
variable      n equal atoms
compute       1 all property/atom radius
variable      step equal step
variable      rts equal step*dt
fix           ts all check/timestep/gran ${dumpStep} 0.1 0.1
thermo_style      custom step time atoms  ke f_ts[1]  f_ts[2] v_m
thermo            ${dumpStep}
thermo_modify     lost ignore norm no
compute_modify  thermo_temp dynamic yes
### INSERT FIRST PARTICLES AND START DUMPING
shell         mkdir post
run           1
dump          dmpMedia bulk custom ${dumpStep} post/dumpMedia*.liggghts id type x y z vx vy vz fx fy fz radius
    ↪ mass
```

```
dump            dumpstl all mesh/stl ${dumpStep} post/dumpSurf*.stl
variable        insertionTime equal ${insertStep}*10
run             ${insertionTime}
### SETTLE INSERTED PARTICLES
unfix               media_ins
write_restart   Restart/restart.step01
### PROPERTIES OF INDIVIDUAL CLUMP
compute         bid all rigid property id_multisphere #body id (1 value)
compute         btype all rigid property clumptype #M-sphere type as defined in fix paticleteplate/multisphere
compute         xcm all rigid property xcm #body position (based on CoM) (3 values)
compute         vel all rigid property vcm #body velocity (based on CoM) (3 values)
compute         fcm all rigid property fcm #fbody force (based on CoM) (3 values)
compute         quat all rigid property quat #body quaternion (based on CoM) (4 values)
compute         bmass all rigid property masstotal #body mass (1 value)
### MULTISHPERE COMPUTE VARIABLE
variable        com1 equal c_xcm[1]
variable        com2 equal c_xcm[2]
variable        com3 equal c_xcm[3]
### DUMP MULTI-SPHERE PROPERTIES
dump            rigids bulk local ${dumpStep} data/rigids*.dump c_bid c_btype c_massms c_xcm[1] c_xcm[2]  &
                 c_xcm[3] c_vel[1] c_vel[2] c_vel[3] c_fcm[1] c_fcm[2] c_fcm[3] ##
variable        processCycle equal 2    #[s] seconds the Trough is vibrating
variable        vibratingTime equal round(${processCycle}*${timeStep}^(-1))
### TROUGH STARTS VIBRATING
# Definition of the vibration in x-axis direction
fix harmonicX1  all move/mesh mesh wall wiggle amplitude  1.589262e-05     0      0     period 8.000000e-02
fix harmonicX2  all move/mesh mesh wall wiggle amplitude  2.358719e-05     0      0     period 4.000000e-02
fix harmonicX3  all move/mesh mesh wall wiggle amplitude  6.802588e-07     0      0     period 2.723404e-02
fix harmonicX4  all move/mesh mesh wall wiggle amplitude  6.736165e-07     0      0     period 2.031746e-02
fix harmonicX5  all move/mesh mesh wall wiggle amplitude  5.226244e-07     0      0     period 1.347368e-02
fix harmonicX6  all move/mesh mesh wall wiggle amplitude  3.070838e-07     0      0     period 1.306122e-02
fix harmonicX7  all move/mesh mesh wall wiggle amplitude  1.923558e-07     0      0     period 1.163636e-02
fix harmonicX8  all move/mesh mesh wall wiggle amplitude  1.486267e-07     0      0     period 1.000000e-02
fix harmonicX9  all move/mesh mesh wall wiggle amplitude  6.028109e-08     0      0     period 9.014085e-03
# Definition of the vibration in y-axis direction
fix harmonicY1  all move/mesh mesh wall wiggle amplitude  0     1.949234e-03    0     period 4.000000e-02
# Definition of the vibration in z-axis direction
fix harmonicZ1  all move/mesh mesh wall wiggle amplitude  0     0     4.038371e-04     period 4.000000e-02
fix harmonicZ2  all move/mesh mesh wall wiggle amplitude  0     0     7.744550e-06     period 2.031746e-02
run ${vibratingTime}
### WRITE RESTART FILE
```

```
write_restart    Restart/restart.step02
```

B.5 Multi-sphere media definition

```
# 65 Spheres single conical geometry media particle (Green polisher).
# Arnon Lopez Marrero (General Engineering Research Institute)

################################################################
# Moment of inertia at the center of mass (kg/m^2)             #
# 2.36E+05      0       0                                      #
# 0        1.98E+05  0                                         #
# 0        0        2.36E+05                                   #
################################################################

# Axis     Radius
#X              Y             Z               R           No
0               2.05E-02      0               1.55E-03    #1
0               1.80E-02      0               2.76E-03    #2
0               1.50E-02      0               4.08E-03    #3
0               1.20E-02      0               5.33E-03    #4
0               7.21E-03      0               7.21E-03    #5
-3.80E-03       4.62E-03      0               4.62E-03    #6
-0.003074265    4.62E-03      -0.002233584    4.62E-03    #7
-0.001174265    4.62E-03      -0.003614015    4.62E-03    #8
0.001174265     4.62E-03      -0.003614015    4.62E-03    #9
0.003074265     4.62E-03      -0.002233584    4.62E-03    #10
0.0038          4.62E-03      0               4.62E-03    #11
0.003074265     4.62E-03      0.002233584     4.62E-03    #12
0.001174265     4.62E-03      0.003614015     4.62E-03    #13
-0.001174265    4.62E-03      0.003614015     4.62E-03    #14
-0.003074265    4.62E-03      0.002233584     4.62E-03    #15
-6.40E-03       2.80E-03      0               2.80E-03    #16
-0.006086762    2.80E-03      -0.001977709    2.80E-03    #17
-0.005177709    2.80E-03      -0.003761826    2.80E-03    #18
-0.003761826    2.80E-03      -0.005177709    2.80E-03    #19
-0.001977709    2.80E-03      -0.006086762    2.80E-03    #20
0               2.80E-03      -0.0064         2.80E-03    #21
0.001977709     2.80E-03      -0.006086762    2.80E-03    #22
0.003761826     2.80E-03      -0.005177709    2.80E-03    #23
0.005177709     2.80E-03      -0.003761826    2.80E-03    #24
```

```
0.006086762   2.80E-03   -0.001977709   2.80E-03   #25
0.0064        2.80E-03   0              2.80E-03   #26
0.006086762   2.80E-03   0.001977709    2.80E-03   #27
0.005177709   2.80E-03   0.003761826    2.80E-03   #28
0.003761826   2.80E-03   0.005177709    2.80E-03   #29
0.001977709   2.80E-03   0.006086762    2.80E-03   #30
0             2.80E-03   0.0064         2.80E-03   #31
-0.001977709  2.80E-03   0.006086762    2.80E-03   #32
-0.003761826  2.80E-03   0.005177709    2.80E-03   #33
-0.005177709  2.80E-03   0.003761826    2.80E-03   #34
-0.006086762  2.80E-03   0.001977709    2.80E-03   #35
-8.15E-03     1.53E-03   0              1.53E-03   #36
-0.007971903  1.53E-03   -0.00169448    1.53E-03   #37
-0.007445395  1.53E-03   -0.003314904   1.53E-03   #38
-0.006593489  1.53E-03   -0.00479045    1.53E-03   #39
-0.005453414  1.53E-03   -0.00605663    1.53E-03   #40
-0.004075     1.53E-03   -0.007058107   1.53E-03   #41
-0.002518489  1.53E-03   -0.007751111   1.53E-03   #42
-0.000851907  1.53E-03   -0.008105353   1.53E-03   #43
0.000851907   1.53E-03   -0.008105353   1.53E-03   #44
0.002518489   1.53E-03   -0.007751111   1.53E-03   #45
0.004075      1.53E-03   -0.007058107   1.53E-03   #46
0.005453414   1.53E-03   -0.00605663    1.53E-03   #47
0.006593489   1.53E-03   -0.00479045    1.53E-03   #48
0.007445395   1.53E-03   -0.003314904   1.53E-03   #49
0.007971903   1.53E-03   -0.00169448    1.53E-03   #50
0.00815       1.53E-03   0              1.53E-03   #51
0.007971903   1.53E-03   0.00169448     1.53E-03   #52
0.007445395   1.53E-03   0.003314904    1.53E-03   #53
0.006593489   1.53E-03   0.00479045     1.53E-03   #54
0.005453414   1.53E-03   0.00605663     1.53E-03   #55
0.004075      1.53E-03   0.007058107    1.53E-03   #56
0.002518489   1.53E-03   0.007751111    1.53E-03   #57
0.000851907   1.53E-03   0.008105353    1.53E-03   #58
-0.000851907  1.53E-03   0.008105353    1.53E-03   #59
-0.002518489  1.53E-03   0.007751111    1.53E-03   #60
-0.004075     1.53E-03   0.007058107    1.53E-03   #61
-0.005453414  1.53E-03   0.00605663     1.53E-03   #62
-0.006593489  1.53E-03   0.00479045     1.53E-03   #63
-0.007445395  1.53E-03   0.003314904    1.53E-03   #64
-0.007971903  1.53E-03   0.00169448     1.53E-03   #65
```

B.6 Simulations parameters

```
# Particle and Domain Properties for DEM

### Domain Properties
variable timeStep          equal   1/350000           #[s]        simulation timestep DEM
variable insertStep        equal   ${timeStep}^(−1)/10   #[s]        insertion time
variable diameterBase      equal   0.167              #[m]        base diameter
variable domainWidth       equal   ${diameterBase}/2.0   #[m]        widht of domain  (x)
variable domainLowHeight   equal   −9.98952e−8        #[m]        height of domain (y low en)
variable domainHighHeight  equal   0.26               #[m]        height of domain (y high end)
variable domainDepth       equal   ${diameterBase}/2.0   #[m]        depht of domain  (z)
variable xlo               equal   −0.05 #−0.125      #[m]        length of domain  (−x)
variable xhi               equal   0.05  #0.125       #[m]        length of domain  (x)
variable ylo               equal   −0.02              #[m]        height of domain (y low end)
variable yhi               equal   0.75               #[m]        height of domain (y high end)
variable zlo               equal   −0.30              #[m]        depht of domain  (−z)
variable zhi               equal   0.30               #[m]        depht of domain  (z)
### Particle and Wall Properties
variable mediaDensity      equal   1665.71            #[kg/m3]    particle density
variable caseDensity       equal   1810.00            #[kg/m3]    particle density
variable wgtMediaClump     equal   0.0046929527       #[kg]       media particle wgt for mediaDensity
variable wgtCaseClump      equal   0.0277206          #[kg]       sensor+case weight for caseDensity
variable wkrLoad           equal   25.0               #[kg]       working load
# Young Modulus has been modified to reduce the length of  timestep, hence the simulation time.
variable youngsModP1       equal   4.6e8              #[Pa]       youngs Modulus(media)
variable youngsModP2       equal   2.8e8              #[Pa]       youngs Modulus(sensor case)
variable youngsModW1       equal   0.025e9            #[Pa]       youngs Modulus(wall)
variable poissionsRatP1    equal   0.2887             #[−]        poissons Ratio(media)
variable poissionsRatP2    equal   0.35               #[−]        poissons Ratio(sensor case)
variable poissionsRatW1    equal   0.25               #[−]        poissons Ratio(wall)
### The experimental values of the coefficients of restitution are commented below:
variable coeffRestP1P1     equal   0.670              #[−]        coefficient Restitution
variable coeffRestP1P2     equal   0.35               #[−]        coefficient Restitution
variable coeffRestP1W1     equal   0.159              #[−]        coefficient Restitution
variable coeffRestP2P2     equal   0.20               #[−]        coefficient Restitution
variable coeffRestP2W1     equal   0.159              #[−]        coefficient Restitution
variable coeffRestP1P2     equal   0.35               #[−]        coefficient Restitution
```

```
variable coeffRestNuLL    equal    0.06                    #[-]            coefficient Restitution
### The coefficients of friction has been calculated using the Incline planes simulation
variable coeffFricP1P1    equal    0.58                    #[-]            coefficient Friction particle-particle
variable coeffFricP1P2    equal    0.34                    #[-]            coefficient Friction particle-particle
variable coeffFricP1W1    equal    0.86                    #[-]            coefficient Friction particle-particle
variable coeffFricP2P2    equal    0.2                     #[-]            coefficient Friction particle-particle
variable coeffFricP2W1    equal    0.86                    #[-]            coefficient Friction particle-wall1
variable coeffFricNuLL    equal    0.0                     #[-]            coefficient Friction particle-wall1
### Dump Properties
variable dumpStep equal            round(${timeStep}^(-1)/90)    #[-]    write dumpfiles (90fps)
### Preprocessor
variable noMediaParticles equal     round(${wkrLoad}/${wgtMediaClump})    #No of particle to be inserted
```

Appendix C

Rotary studies

C.1 LIGGGHTS script

```

## 3D CENTRIFUGE MACHINE
## Arnon Lopez Marrero (General Engineering Research Institute)
## ArnonLopez.Marrero@the-mtc.org
## FILE: in.sensor3_step01
## RUN PARALLEL: sbatch in.sensor3_step01
## DATE: 26 APRIL 2019

# This script is designed to study the trajectory of the particles when exited in the bowl.
# Firstly, the bowl is loaded with 7 kg of media, then the sensor case is inserted.
# Lastly the bottom stars spining at 200 rpm.
# Material properties have been obtained using the incline panel device and the dropping ball test

# Description:
# There are three types of Materials.
# 1. Media material (Plastic).
# 2. Sensor Case Material (POM-C (Polyacetal - Copolymer)).
# 3. Lining Material (Elastomer Polyurethane).

processors 6 6 6
### Input Parameters
include         /home/MMEALOPE/JOBS/Sensor_Paper3/input/PRIMES.inc
include         /home/MMEALOPE/JOBS/Sensor_Paper3/input/in.DEM
### General Settings for Granular Systems
atom_style      granular
```

```
atom_modify     map array
boundary        f f f
newton          off
echo            both

communicate     single vel yes

units           si
hard_particles  yes
soft_particles  yes
### Region/Simulationbox Definition and ReadIn Particle Data
region          reg cylinder y 0. 0. ${diameterBase} ${domainLowHeight} ${domainHighHeight} units box

create_box      3 reg

neighbor        0.00432 bin
neigh_modify    delay 0

group           bulk initialize
group           sensor initialize
group           media initialize
### Material Properties
fix             m1 all property/global youngsModulus peratomtype &
                                ${youngsModP1} ${youngsModP2} ${youngsModW1}
fix             m2 all property/global poissonsRatio peratomtype &
                                ${poissionsRatP1} ${poissionsRatP2} ${poissionsRatW1}
fix             m3 all property/global coefficientRestitution peratomtypepair 3         &
                                ${coeffRestP1P1} ${coeffRestP1P2} ${coeffRestP1W1} &
                                ${coeffRestP1P2} ${coeffRestP2P2} ${coeffRestP2W1} &
                                ${coeffRestP1W1} ${coeffRestP2W1} ${coeffRestNuLL} ##
fix             m4 all property/global coefficientFriction peratomtypepair 3         &
                                ${coeffFricP1P1} ${coeffFricP1P2} ${coeffFricP1W1} &
                                ${coeffFricP1P2} ${coeffFricP2P2} ${coeffFricP2W1} &
                                ${coeffFricP1W1} ${coeffFricP2W1} ${coeffFricNuLL} ##
### Pair Style
pair_style      gran model hertz tangential history cohesion off #Hertzian without cohesion
pair_coeff      * *
### Time Step
timestep        ${timeStep}
### Walls
```

```
fix         wall all mesh/surface file /home/MMEALOPE/JOBS/Sensor_Paper/input/Models/wallSimplified3.stl type 3 curvature 1e-5
fix         wall2 all mesh/surface file /home/MMEALOPE/JOBS/Sensor_Paper/input/Models/wallSimplified4.stl type 3 curvature 1e-5
fix         bottom all mesh/surface file /home/MMEALOPE/JOBS/Sensor_Paper/input/Models/bottomSimplified.stl type 3
fix         geometry all wall/gran model hertz tangential history cohesion off mesh n_meshes 3 meshes wall wall2 bottom
### Gravity
fix         gravi all gravity 9.81 vector 0.0 -1.0 0.0
### Particle Distributions and Insertion
region      INS1 cylinder y  0  0  0.160  0.12  0.24  units box #cylinder args = axis of cylinder c1 c2 radius lo hi
region      INS2 cylinder y  0.05  0.05  0.035  0.13  0.19  units box

fix         media_tpt bulk particletemplate/multisphere ${prime1} atom_type 1  &
            density constant ${mediaDensity} nspheres 11 &
            ntry 100000 spheres file /home/MMEALOPE/JOBS/Sensor_Paper/input/Models/36plasticmedia   &
            scale 1.0 type 1 ##
fix         sensor_tpt bulk particletemplate/multisphere ${prime2} atom_type 1  &
            density constant ${caseDensity} nspheres 43 ntry 100000 spheres &
            file /home/MMEALOPE/JOBS/Sensor_Paper/input/Models/43SpheresSensorCase &
            scale 1.0 type 2 ##
fix         media_distro media particledistribution/discrete/numberbased ${prime3}  1 media_tpt 1.0
fix         sensor_distro sensor particledistribution/discrete/numberbased ${prime4}  1 sensor_tpt 1.0
#for 7kg use the variable  ${noMediaParticles}
fix         media_ins media insert/rate/region seed ${prime5} distributiontemplate media_distro &
            nparticles ${noMediaParticles} massrate 40 insert_every ${insertStep} &
            overlapcheck yes all_in yes vel constant 0.0 -1.0 0.0 region INS1 ##
### Apply multisphere Integration to all Particles that are Inserted
fix         int1 bulk multisphere allow_group_and_set yes
### Define Compute to Calculate  the Mass of Individual Multi-sphere Bodies (clumps)
compute     massms all multisphere property masstotal #body mass (1 value)
variable    massCase equal c_massms
### Output Settings
variable    m equal mass(all)
variable    n equal atoms

compute     1 all property/atom radius
variable    step equal step
variable    rts equal step*dt
fix         ts all check/timestep/gran ${dumpStep} 0.1 0.1
```

```
thermo_style     custom step time atoms  ke f_ts[1]  f_ts[2] v_m
thermo           ${dumpStep}
thermo_modify    lost ignore norm no
compute_modify  thermo_temp dynamic yes
### Write restart file
shell          mkdir restartporto
restart        ${dumpStep} restartporto/restart.1 restartporto/restart.2
### Insert the First Particles and Star Dumping
shell          mkdir post
run            1
dump           dmpMedia bulk custom ${dumpStep} post/dumpMedia*.liggghts id type x y z vx vy vz fx fy fz radius
    ↪ mass
dump           dumpstl all mesh/stl ${dumpStep} post/dumpSurf*.stl
variable       insertionTime equal ${insertStep}*3
run            ${insertionTime}
### Settle Inserted Particles
unfix            media_ins
write_restart    Restart/restart.step01
###Insert the Sensor Case
fix            sensor_ins sensor insert/rate/region seed ${prime6} distributiontemplate sensor_distro  &
               nparticles 1 massrate 40 insert_every ${insertStep} &
               overlapcheck yes all_in yes vel constant 0.0 -1.0 0.0 region INS2
run            ${insertStep}
### Settle sensor on top of the media
unfix            sensor_ins
run              ${insertStep}*2
write_restart    Restart/restart.step02
```

C.2 Simulations parameters

```
# Particle and Domain Properties for DEM

### Domain Properties
# I reduced the timestep from 1/2700000 = 3.7e-7 to 1/350000
variable timeStep         equal    1/350000              #[s]          simulation timestep DEM
variable insertStep       equal    ${timeStep}^(-1)/10   #[s]          insertion time
variable diameterBase     equal    0.167                 #[m]          base diameter
variable domainWidth      equal    ${diameterBase}/2.0   #[m]          widht of domain  (x)
variable domainLowHeight  equal    -9.98952e-8           #[m]          height of domain (y low en)
variable domainHighHeight equal    0.26                  #[m]          height of domain (y high end)
```

```
variable domainDepth      equal    ${diameterBase}/2.0    #[m]          depht of domain  (z)
### Particle and Wall Properties
variable mediaDensity     equal    1665.71               #[kg/m3]       particle density
variable caseDensity      equal    1810.00               #[kg/m3]       particle density
variable wgtMediaClump    equal    0.000490733            #[kg]          media particle wgt for mediaDensity
variable wgtCaseClump     equal    0.0277206              #[kg]          sensor+case weight for caseDensity
variable wkrLoad          equal    7.0                    #[kg]          working load
# I modified the Young Modulus to reduce the length of  timestep, hence the simulation time.
variable youngsModP1      equal    4.6e8     #4.6e9       #[Pa]          youngs Modulus(media)
variable youngsModP2      equal    2.8e8     #2.8e9       #[Pa]          youngs Modulus(sensor case)
variable youngsModW1      equal    0.025e9                #[Pa]          youngs Modulus(wall)
variable poissionsRatP1   equal    0.2887                 #[-]           poissons Ratio(media)
variable poissionsRatP2   equal    0.35                   #[-]           poissons Ratio(sensor case)
variable poissionsRatW1   equal    0.25                   #[-]           poissons Ratio(wall)
#The experimental values of the coefficients of restitution are commented below:
variable coeffRestP1P1    equal    0.670                  #[-]           coefficient Restitution
variable coeffRestP1P2    equal    0.35                   #[-]           coefficient Restitution
variable coeffRestP1W1    equal    0.159                  #[-]           coefficient Restitution
variable coeffRestP2P2    equal    0.20                   #[-]           coefficient Restitution
variable coeffRestP2W1    equal    0.159                  #[-]           coefficient Restitution
variable coeffRestP1P2    equal    0.670                  #[-]           coefficient Restitution
variable coeffRestP1W1    equal    0.670                  #[-]           coefficient Restitution
variable coeffRestNuLL    equal    0.06                   #[-]           coefficient Restitution
# The coefficients of friction has been calculated using the Incline planes simulation
variable coeffFricP1P1    equal    0.58                   #[-]           coefficient Friction particle-particle
variable coeffFricP1P2    equal    0.34                   #[-]           coefficient Friction particle-particle
variable coeffFricP1W1    equal    0.86                   #[-]           coefficient Friction particle-particle
variable coeffFricP2P2    equal    0.2                    #[-]           coefficient Friction particle-particle
variable coeffFricP2W1    equal    0.86                   #[-]           coefficient Friction particle-wall1
variable coeffFricNuLL    equal    0.0                    #[-]           coefficient Friction particle-wall1
### Dump Properties
variable dumpStep equal          round(${timeStep}^(-1)/90)     #[-]          write dumpfiles (90fps)
### Preprocessor
variable noMediaParticles equal    round(${wkrLoad}/${wgtMediaClump})    #No of particle to be inserted
```

C.3 Multi-sphere sensor definition

```
# 43 Spheres Sensor Case.
# Arnon Lopez Marrero (General Engineering Research Institute)
# A.LopezMarrero@2016.ljmu.ac.uk
```

```

#X1 Y1  Z1  R
0    6.25E−03     0     6.25E−03      #1
5.12E−03     6.25E−03     0     6.25E−03      #2
2.56E−03     6.25E−03     0.00443405     6.25E−03      #3
−2.56E−03     6.25E−03     0.00443405     6.25E−03      #4
−5.12E−03     6.25E−03     0  6.25E−03      #5
−2.56E−03     6.25E−03     −0.00443405     6.25E−03      #6
2.56E−03     6.25E−03     −0.00443405     6.25E−03      #7
1.02E−02     6.25E−03     0 6.25E−03      #8
8.86E−03     6.25E−03     0.005115     6.25E−03      #9
5.12E−03     6.25E−03     0.00885944     6.25E−03      #10
0.00E+00     6.25E−03     0.01023     6.25E−03      #11
−5.12E−03     6.25E−03     0.00885944     6.25E−03      #12
−8.86E−03     6.25E−03     0.005115     6.25E−03      #13
−1.02E−02     6.25E−03     0     6.25E−03      #14
−8.86E−03     6.25E−03     −0.005115     6.25E−03      #15
−5.12E−03     6.25E−03     −0.00885944     6.25E−03      #16
0.00E+00     6.25E−03     −0.01023     6.25E−03      #17
5.12E−03     6.25E−03     −0.00885944     6.25E−03      #18
8.86E−03     6.25E−03     −0.005115     6.25E−03      #19
1.54E−02     6.25E−03     0     6.25E−03      #20
0.014826961     6.25E−03     0.003972872     6.25E−03      #21
0.01329349     6.25E−03     0.007675     6.25E−03      #22
0.010854089     6.25E−03     0.010854089     6.25E−03      #23
0.007675     6.25E−03     0.01329349     6.25E−03      #24
0.003972872     6.25E−03     0.014826961     6.25E−03      #25
0     6.25E−03     0.01535     6.25E−03      #26
−0.003972872     6.25E−03     0.014826961     6.25E−03      #27
−0.007675     6.25E−03     0.01329349     6.25E−03      #28
−0.010854089     6.25E−03     0.010854089     6.25E−03      #29
−0.01329349     6.25E−03     0.007675     6.25E−03      #30
−0.014826961     6.25E−03     0.003972872     6.25E−03      #31
−0.01535     6.25E−03     0     6.25E−03      #32
−0.014826961     6.25E−03     −0.003972872     6.25E−03      #33
−0.01329349     6.25E−03     −0.007675     6.25E−03      #34
−0.010854089     6.25E−03     −0.010854089     6.25E−03      #35
−0.007675     6.25E−03     −0.01329349     6.25E−03      #36
−0.003972872     6.25E−03     −0.014826961     6.25E−03      #37
0     6.25E−03     −0.01535     6.25E−03      #38
0.003972872     6.25E−03     −0.014826961     6.25E−03      #39
```

```
0.007675     6.25E−03     −0.01329349     6.25E−03     #40
0.010854089     6.25E−03     −0.010854089     6.25E−03     #41
0.01329349     6.25E−03     −0.007675     6.25E−03     #42
0.014826961     6.25E−03     −0.003972872     6.25E−03     #43
```

Appendix D

DEM2FEA

D.1 Source code

```
"""
The script is developed by Arnon Lopez Marrero under the supervision of Matteo Villa,
as part of the works carried out as part of the PhD studies.
E−mail to Arnon: ArnonLopez.Marrero@the−mtc.org; A.LopezMarrero@2016.ljmu.ac.uk
       to discuss technical details.
============================================================================
The script is to transfer the LIGGGHTS output data into a Calculix input file to
further analyses the behaviour of the surfaces.
Python script performs the following three major steps:
    1) reading the geometry file generated by GMSH and retrieving key data;
    2) reading the output files generated by LIGGGHTS and retrieving key data;
    3) combining the data acquired before into a single Calculix input file (∗.inp).
"""

## LOADING MODULES
import glob
import os
import re
import pandas
from colorama import init, Fore, Style
## MODULES TO PROCESS ARRAYS
import numpy as np
import pandas as pd
## MODULES TO PROCESS VTK FILES
import vtk
```

```
from vtk.numpy_interface import dataset_adapter as dsa
import time
import datetime
from functions import *
now = datetime.datetime.now()
start_time = time.time()
## REQUESTING FILES TO USER
 s = input('Series of files to be processed (ie. int*.vtk): ')
## SELECTING FOLDER WHERE TO READ THE DATA AND WHERE TO WRITE THE NEW DATA.
current_working_directory = os.getcwd() + '/'
printOutFolder = current_working_directory + 'Stress/'
# Path to CCX input file
save_to_path = current_working_directory + 'FEA/'
createFolderInSitu(printOutFolder)  # calling function createFolderInSitu to create folder 'Stress/'
filenames = glob.glob(current_working_directory + s)
filenames = sorted(filenames, key=keyfunc)

## Uncomment the two folowing lines to fix a folder(input_path) where to look for files
# input_path = '/test/'
# filenames = glob.glob(input_path + s)

# Iteration over each relevant file in the data folder
fileNo = 1  # used for generating the file names

for filename in filenames:
    print('Reading file: ', filename)
    print('Checking compatibility of: ', filename)
    ## Piece of code to check if the files are compatible
    fo = open(filename, encoding='utf8')
    lines = fo.readlines()

    firstLine = lines[0]
    secondLine = lines[1]
    thirdLine = lines[2]
    fourthLine = lines[3]
    fifthLine = lines[4]

    if firstLine[:22] != '# vtk DataFile Version':
        raise Exception('ERROR: Expecting vtk DataFile, cannot continue')
    elif thirdLine[0:5] != 'ASCII':
        raise Exception('ERROR: Expecting ASCII VTK mesh, cannot continue')
```

```
elif fourthLine[0:16] != 'DATASET POLYDATA':
    raise Exception('ERROR: Expecting DATASET POLYDATA VTK file, cannot continue')
else:
    print(filename, ' Compatible!')
## READ THE VTK FILE (SCALARS)
reader = vtk.vtkPolyDataReader()
reader.SetFileName(filename)
reader.Update()  # Needed because of GetScalarRange
## EXTRACT POINT DATA FROM VTK
#Access the generated data object.
dataRAW = reader.GetOutputDataObject(0)
# Wrap the data object.
data = dsa.WrapDataObject(dataRAW)
## TO ACCESS POINT_DATA(FIELDDATA) ARRAYS, USING THE KEYWORD
# print(data.PointData.keys())*.vtk
stress = data.PointData['stress']
area = data.PointData['area']
## TO ACCESS POINT_DATA(FIELDDATA) ARRAYS, USING THE POSITION
# stress = data.PointData[2]
# area = data.PointData[3]
polydata = reader.GetOutput()
points_coord_array = dsa.WrapDataObject(polydata).Points  # access to POINTS
## USING PANDA TO GENERATE A DATA STRUCTURE
stresslog = pd.DataFrame()
## ASSIGN ARRAYS VALUES TO THE DATA STRUCTURE
i = 0
print('Extracting information from VTK file: ', filename, '\n')
for cell in stress:
    # print('.', end="")
    while i < len(stress):
        # Writing component X of the stress to stress:0 column
        stresslog.loc[i, 'stress:0'] = stress[i, 0]
        # Writing component Y of the stress to stress:1 column
        stresslog.loc[i, 'stress:1'] = stress[i, 1]
        # Writing component z of the stress to stress:2 column
        stresslog.loc[i, 'stress:2'] = stress[i, 2]
        # Writing the area of the element(triangle) to area column
        stresslog.loc[i, 'area'] = area[i]
        # Writing coordinates X of the nodes of the element(triangle) to Points:0 column
        stresslog.loc[i, 'Points:0'] = points_coord_array[i, 0]
        # Writing coordinates Y of the nodes of the element(triangle) to Points:1 column
```

```
            stresslog.loc[i, 'Points:1'] = points_coord_array[i, 1]
            # Writing coordinates Z of the nodes of the element(triangle) to Points:2 column
            stresslog.loc[i, 'Points:2'] = points_coord_array[i, 2]
            i += 1
    # Writing Stress matrix to CSV file
    base_filename = 'Stresslog_Step'
    file_extension = '.csv'
    specname = base_filename + str(fileNo) + file_extension
    fileNo += 1  # file counter
    writeIn = writeOput(specname, printOutFolder)  # Calling function writeOput(outname, outdir)
    stresslog.to_csv(writeIn, index=False, header=True, )
    print('Writing file: ', writeIn)
print('
    ↪ ########################################################################
    ↪ n')
## SETTING DATA FOLDER AND FILE NAME TO READ
input_path = printOutFolder
filenames = glob.glob(input_path + base_filename + "*.csv")
filenames = sorted(filenames, key=keyfunc)
## ITERATION OVER EACH RELEVANT FILE IN THE DATA FOLDER
fileNo = 1  # used for generating the file names
for filename in filenames:
    ## READ CSV FILE OUTPUT BY THE PREVIOUS FUNCTION
    stressMatrix = pd.read_csv(filename, dtype=np.float64)
    ## CONVERT THE STRESS ACTING ON THE ELEMENTS INTO FORCES ACTING AT THE NODES.
    line = 0
    for eachLine in range(len(stressMatrix)):
        while line < len(stressMatrix):
            stressMatrix.iloc[line, 0] = (stressMatrix.iloc[line, 0] * (stressMatrix.iloc[line, 3] / 3))
            stressMatrix.iloc[line, 1] = (stressMatrix.iloc[line, 1] * (stressMatrix.iloc[line, 3] / 3))
            stressMatrix.iloc[line, 2] = (stressMatrix.iloc[line, 2] * (stressMatrix.iloc[line, 3] / 3))
            line += 1
    ## REMOVE AREA COLUMN FROM MATRIX
    forceMatrix = stressMatrix.drop('area', axis=1)
    ## CHANGE HEADERS
    forceMatrix.rename(index=str, columns={'stress:0': 'Fx', 'stress:1': 'Fy', 'stress:2': 'Fz',
                                           'Points:0': 'X', 'Points:1': 'Y', 'Points:2': 'Z'},
                       inplace=True)
    ## REMOVED LINES WITH NO FORCES ACTING AT NODES TO SPEED UP THE LOOPS
    i = 0
    for line in range(len(forceMatrix)):
```

```
        while i < len(forceMatrix):
            if forceMatrix.iloc[i, 0] == 0 \
                    and forceMatrix.iloc[i, 1] == 0 \
                    and forceMatrix.iloc[i, 2] == 0:
                forceMatrix = forceMatrix.drop(forceMatrix.index[i])
                continue
            i += 1
    ## ITERATION TO SUM THE FORCES AT THE NODES.
    print('Calculating forces at nodes')
    i = 0
    for line in forceMatrix:  # loop through the lines
        while i < (len(forceMatrix) - 1):
            j = 1
            for cells in forceMatrix:  # loop through the cells
                index = 0
                j = 1
                while (index) < len(forceMatrix):
                    if i >= len(forceMatrix):
                        break
                    if i >= index:
                        index = i + 1
                        if index >= (len(forceMatrix)):
                            break
                    if forceMatrix.iloc[i, 3] == forceMatrix.iloc[(index), 3] \
                            and forceMatrix.iloc[i, 4] == forceMatrix.iloc[(index), 4] \
                            and forceMatrix.iloc[i, 5] == forceMatrix.iloc[(index), 5]:
                        forceMatrix.iloc[i, 0] += forceMatrix.iloc[(index), 0]
                        forceMatrix.iloc[i, 1] += forceMatrix.iloc[(index), 1]
                        forceMatrix.iloc[i, 2] += forceMatrix.iloc[(index), 2]
                        forceMatrix = forceMatrix.drop(forceMatrix.index[index])
                        index = i
                        j = 0
                        continue
                    j += 1
                    index = i + j
            i += 1
    i = 0
    # To stop dtype=np.float64 having problem for reading empty files
    if forceMatrix.empty:
        cntr = 0
        base_filename = 'ForceAtNodes_Step'
```

```
        file_extension = '.csv'
        emptyStepFullName = base_filename + str(fileNo) + file_extension
        writeIn = writeOput(emptyStepFullName, printOutFolder)
        with open(writeIn, mode='w+', encoding='utf-8') as f:
            # **LIGGGHTS simulation did not computed forces in this step\n')
            forceMatrix.to_csv(f, mode='a', header=True, index=False)
            f.write("0, 0, 0, 0 , 0, 0")
        # Print out a full list of the snapshots with no stresses
        logFullName = 'DEMemptyStep.log'
        writeLogIn = writeOput(logFullName, save_to_path)
        fileList = glob.glob(current_working_directory + s)
        fileList = sorted(fileList, key=keyfunc)
        with open(writeLogIn, mode='w+', encoding='utf-8') as f:
            if cntr < 1:
                f.write("This log contains all the snapshots from DEM with no stresses\n")
                f.write('File: %s\n' % fileList[cntr])
            else:
                f.write('File: %s\n' % fileList[cntr])
        cntr += 1
        fileNo += 1
        continue
    # Creating the path and the file where to print the force matrix values
    base_filename = 'ForceAtNodes_Step'
    file_extension = '.csv'
    specname = base_filename + str(fileNo) + file_extension
    output_path = printOutFolder
    writeIn = writeOput(specname, output_path)
    fileNo += 1
    # Writing force matrix to CSV file
    forceMatrix.to_csv(writeIn, index=False, header=True, )
    print('\nWriting file: ', writeIn)
print(Fore.GREEN + '\nPath to results:\n' + printOutFolder + '\n')
print(Fore.CYAN + "Writing INP file\n")
init(autoreset=True)
inp_path = '/media/arnon/Memoria/PhD/LIGGGHTS/LJMU_HPC/Stream_3/Geometry/Specimens/'
inp_file_name = 'squareSpecimen-HPC-Pos1'
csv_path = printOutFolder
csv_root = 'ForceAtNodes*.csv'
output_file_name = inp_file_name + '-ccx.inp'


```

```
#################################################################
##    RETRIEVING ONLY THE KEY INFORMATION FROM *.inp FILE GENERATED IN GMSH        ##
#################################################################

# Declare an empty list named "lines" where to store whole document
lines = []
# Declare an empty DataFrame named "nodes" to store the nodes for the WHOLE MESH
nodes = pd.DataFrame()
element_volume = pd.DataFrame()
keywords = ['*Node', '*NODE', '*node']  # Substring to search for, in this case *node
element_types_list = ['C3D4', 'C3D6', 'C3D8', 'C3D10', 'C3D15', 'C3D20', 'C3D27']
index = 0
geometry_file = inp_path + inp_file_name + ".inp"
create_folder(save_to_path)
try:
    with open(geometry_file, 'rt') as in_file: # open file and close it after read.
        ## STORE EACH LINE IN A STRING VARIABLE "LINE"
        for line in in_file:
            lines.append(line.lstrip()) # stripping whitespaces at the beginning of the line.

    index = find_word_line(lines, index, keywords)[0] # searching for the keyword
    milestone = index  # sets the starting line for the search of element_types_list
    for element in range(index, len(lines)):  # iterate over lines after 'keyword' was found
        str_line = lines[element]  # store lines in lis 'str_line'

        if str_line[0].isdigit():  # if the first character is a digit then store it in nodes[]
            str_line = str_line.replace(',', ' ')  # remove all commas
            str_line = str_line.split()  # break the string line in several substrings
            ## STORING EACH NODE IN DATAFRAME "NODES"
            nodes.loc[element, 'id'] = str_line[0]  # node ID as str
            nodes.loc[element, 'x'] = float(str_line[1])  # x coordinate
            nodes.loc[element, 'y'] = float(str_line[2])  # y coordinate
            nodes.loc[element, 'z'] = float(str_line[3])  # z coordinate
            milestone += 1
        else:  # when first character is different to a digit then break the loop
            milestone += 1  # Determine the starting line for the next search
            break
    number_of_nodes = str(len(nodes))
    element_list = []
    ## RETURNS THE NO OF THE LINE CONTAINING THE KEYWORD
    index = find_word_line(lines, milestone, element_types_list)[0]
```

```
    ## RETURNS THE FIRST KEYWORD FOUND
    element_type_mesh = find_word_line(lines, milestone, element_types_list)[1]
    for element in range(index, len(lines)):  # iterate over lines after where the 'keyword' was found
        str_line = lines[element]  # store lines in list 'str_line'
        if str_line[0].isdigit():  # if the first character is a digit then store it in nodes[]
            element_list.append(str_line)
        else:  # when first character is different to a digit then break the loop
            break

    print(
        '=========== General information =========== \n' + 'Number of nodes in the geometry: ' +
      ↪ number_of_nodes + '\n' + 'Mesh element type: ' + element_type_mesh + '\n'
        + 'Number of elements: ' + str(element - index) + '\n
      ↪ ========================================== \n')
except FileNotFoundError:
    print("Oops!  File was't found.")

filename = 'CCXInputFile.inp'
loc = save_to_path + filename
print(type(nodes))
# nodes = nodes.astype(float)
extractDataFromINP = find_elements_physical_groups(geometry_file, milestone, save_to_path)
physicalGroupLabel = extractDataFromINP[0]
## REMOVE QUOTATION MARKS IN THE LINES OF THE LIST ELEMENT_LIST
physicalGroupLabel = listToStringWithoutBrackets(physicalGroupLabel)
volumesGeometry = extractDataFromINP[1]
columns_Headers = ['x', 'y', 'z']
# nodes['id'] = nodes['id'].astype(int)
for i in columns_Headers:
    nodes[i] = nodes[i].astype(float).round(4)

nodes_to_txt = nodes  # .round(4)
# nodes_to_txt = round(nodes, 4)  # round the float to 4 digits before printing them out

with open(loc, mode='w+', encoding='utf-8') as f:  # open/create a file named "filename" and located in "
  ↪ save_to_path"
    f.write('** This script has been generated by Arnon Lopez Marrero\n' +
            '** General Engineering Research Institute @ LJMU\n' +
            '** ArnonLopez.Marrero@the-mtc.org\n' +
            '** ' + str(now.strftime('%d-%m-%Y')) + '\n')
    f.write('*NODE, NSET=Nall\n')
```

```
    nodes_to_txt.to_csv(f, mode='a', header=False, index=False)
    f.write('**' + '\n')
    f.write(volumesGeometry)
    f.write('**\n' + '*SOLID SECTION, ELSET=EAll, MATERIAL=Steel\n' + '**\n')
    f.write('**\n' + '*AMPLITUDE, NAME = Amp1\n' + '0,0, 1,1\n' + '**\n')
    f.write('**\n' + '*MATERIAL, NAME=Steel\n' + '*ELASTIC\n' + '2e+11, 0.3\n' + '**\n')


## SELECTING FOLDER TO READ THE DATA AND WHERE TO WRITE THE NEW DATA.
print('Looking for %s files at %s' % (csv_root, csv_path))
filenames = glob.glob(csv_path + csv_root)
filenames = sorted(filenames, key=keyfunc)  #
# ITERATION OVER EACH RELEVANT FILE IN THE DATA FOLDER
fileNo = 1  # used for generating the file names
path_to_loads = save_to_path + 'cLoads.txt'
## OPEN/CREATE A TXT FILE TO STORE ALL THE FORCES ACTING ON THE SETS OF NODES
cLoads = open(path_to_loads, 'w+')
step = 0
step_name = 1
control_list = [-1]
n = 1
index = 0  # resetting variable
print(str(len(filenames)) + ' non-empty files have been found:')


for files in filenames:
    counter = 1  # allows to print Header only when needed
    # forceAtNodes = pd.read_csv(files)
    forceAtNodes = pd.read_csv(files, dtype=np.float64)
    number_of_lines = file_len(files)
    # calling function to remove lines with no forces
    forceAtNodes = no_force_line_removed(forceAtNodes)
    number_of_lines = len(forceAtNodes)
    step += 1

    with open(loc, 'a') as nodesSets:
        if number_of_lines == 0:
            print('Printing Step_' + str(step_name) + ' to file')
            cLoads.write('**-----------------------------------------**\n')
            cLoads.write('**                    ' + 'Step ' + str(step_name) + '                    **\n')
            cLoads.write('**-----------------------------------------**\n')
```

```
        cLoads.write('** This step correspond to LIGGGHTS Step' + str(step_name) + ' **\n')
        # STEP DEFINITION
        cLoads.write(
            '*STEP, NAME=Step-' + str(
                step_name) + ',' + ' NLGEOM, inc=10000, unsymm=YES\n' + ' **\n')
        cLoads.write('*STATIC\n' + '0.1, 1., 1e-05, 0.1\n')
        # BOUNDARY CONDITIONS
        cLoads.write('**\n' + '** BOUNDARY CONDITIONS\n' + '**\n')
        cLoads.write(
            '*BOUNDARY\n' + '** node set, dof, last dof, magnitude\n' +
            str(physicalGroupLabel) + ', 1,6,0\n' + '**\n')
        # LOAD SPECIFICATION
        cLoads.write('*CLOAD, AMPLITUDE = Amp1, op=NEW\n')
        cLoads.write('** IMPORTANT: The line below was imposed to avoid having a step with no load\n' +
                     '** The imposed load is applied on the boundary conditions so the system is not affected \n')
        cLoads.write('** set name, direction, magnitude\n')
        cLoads.write(str(physicalGroupLabel) + ', 1, 0.00001\n')
        # OUTPUT REQUEST FOR CALCULIX
        cLoads.write('** OUTPUT REQUESTS\n' + '**\n')
        cLoads.write('*RESTART, write, frequency=0\n' + '**\n')
        cLoads.write('** FIELD OUTPUT: F-Output-1\n' + '**\n')
        cLoads.write('*OUTPUT, field\n' + '*NODE OUTPUT, NSET=Nall\n' + 'CF, RF, U\n' + '**\n')
        cLoads.write('*ELEMENT OUTPUT,ELSET=EAll\n ' + 'LE, P, S\n')
        cLoads.write('*EL FILE\n' + 'U,S\n')
        cLoads.write('*END STEP\n')
        # counter = 2
        step_name += 1
    if number_of_lines != 0:
        for index in range(len(forceAtNodes)):  # iterate over the nodes and its forces (CSV file)
            force_1 = round(forceAtNodes.iloc[index, 0],
                            5)  # storing forces in variable for the direction 1 from  CSV file.
            force_2 = round(forceAtNodes.iloc[index, 1],
                            5)  # storing forces in variable for the direction 2 from  CSV file.
            force_3 = round(forceAtNodes.iloc[index, 2],
                            5)  # storing forces in variable for the direction 3 from  CSV file.
            for i in range(len(nodes)):  # iterate over the nodes generated for the geometry by GMSH
                # rounding all coordinate to 3 decimal so they can be compare.
                coord_x_force = round(forceAtNodes.iloc[index, 3], 3)
                coord_x_node = round(nodes.iloc[i, 1], 3)
                coord_y_force = round(forceAtNodes.iloc[index, 4], 3)
                coord_y_node = round(nodes.iloc[i, 2], 3)
```

```
coord_z_force = round(forceAtNodes.iloc[index, 5], 3)
coord_z_node = round(nodes.iloc[i, 3], 3)
if coord_x_force == coord_x_node and \
        coord_y_force == coord_y_node and \
        coord_z_force == coord_z_node:
    nodeID = int(nodes.iloc[i, 0])
    if force_1 != 0 and counter == 1 or \
            force_2 != 0 and counter == 1 or \
            force_3 != 0 and counter == 1:
        # print out to file step's header and the general definition for the Step
        print('Printing Step_' + str(step_name) + ' to file')
        cLoads.write('**----------------------------------------**\n')
        cLoads.write('**                  ' + 'Step ' + str(step_name) + '               **\n')
        cLoads.write('**----------------------------------------**\n')
        cLoads.write('** This step correspond to LIGGGHTS Step' + str(step_name) + ' **\n')
        # STEP DEFINITION
        cLoads.write(
            '*STEP, NAME=Step-' + str(
                step_name) + ',' + ' nlgeom=YES, inc=10000, unsymm=YES\n' + ' **\n')
        cLoads.write('*STATIC\n' + '0.1, 1., 1e-05, 0.1\n')
        # BOUNDARY CONDITIONS
        cLoads.write('**\n' + '** BOUNDARY CONDITIONS\n' + '**\n')
        cLoads.write(
            '*BOUNDARY\n' + '** node set, first dof, last dof, magnitude\n' +
            str(physicalGroupLabel) + ', 1,6,0\n' + '**\n')
        # LOAD SPECIFICATION
        cLoads.write('*CLOAD, AMPLITUDE = Amp1, op=NEW\n' +
            '** set name, direction, magnitude\n')
        counter = 2
        step_name += 1
    # note: the if below save the list of nodes set where a force is applied in nodesSets
    if force_1 != 0 or force_2 != 0 or force_3 != 0:
        set_name = 'nodesSet' + str(int(nodes.iloc[i, 0]))
        blacklist(nodeID, control_list, nodesSets, set_name)
    # when Fx is not 0 then print it out
    if force_1 != 0:
        # set_name = 'nodesSet' + str(int(nodes.iloc[i, 0]))
        cLoads.write('%s, 1, %f\n' % (set_name, force_1))
        # blacklist(nodeID, control_list, nodesSets, set_name)
        # after printing out Fx check if Fy and Fz are not 0
        if force_2 == 0 and force_3 == 0:
```

```
                                cLoads.write('**\n')
                                continue
                        # when Fy is not 0 then print it out
                        if force_2 != 0:
                            if force_1 != 0:
                                cLoads.write('%s, 2, %f\n' % (set_name, force_2))
                                # cLoads.write('**\n')
                            else:
                                # set_name = 'nodesSet' + str(int(nodes.iloc[i, 0]))  # set's name
                                cLoads.write('%s, 2, %f\n' % (set_name, force_2))  # print to file F in direction 2
                                # blacklist(nodeID, control_list, nodesSets)
                            if force_1 != 0 and force_3 == 0:
                                cLoads.write('**\n')
                        if force_3 != 0:
                            if force_1 != 0 or force_2 != 0:
                                cLoads.write('%s, 3, %f\n' % (set_name, force_3))
                                cLoads.write('**\n')
                            else:
                                # set_name = 'nodesSet' + str(int(nodes.iloc[i, 0]))
                                cLoads.write('%s, 3, %f\n' % (set_name, force_3))
                                # blacklist(nodeID, control_list, nodesSets)
                                cLoads.write('**\n')
        else:
            continue
        if force_1 != 0 and counter == 2 or \
                force_2 != 0 and counter == 2 or \
                force_3 != 0 and counter == 2:
            # OUTPUT REQUEST FOR CALCULIX
            cLoads.write('** OUTPUT REQUESTS\n' + '**\n')
            cLoads.write('*RESTART, write, frequency=0\n' + '**\n')
            cLoads.write('** FIELD OUTPUT: F-Output-1\n' + '**\n')
            cLoads.write('*OUTPUT, field\n' + '*NODE OUTPUT, NSET=Nall\n' + 'CF, RF, U\n' + '**\n')
            cLoads.write('*ELEMENT OUTPUT,ELSET=EAll\n ' + 'LE, P, S\n')
            cLoads.write('*EL FILE\n' + 'U,S\n')
            cLoads.write('*END STEP\n')
            counter = 3
        step += 1
cLoads.close()

filenames = [loc, path_to_loads]
abaqus_input_script = save_to_path + output_file_name
```

```
with open(abaqus_input_script, 'w') as dump:
    for fname in filenames:
        with open(fname) as infile:
            dump.write(infile.read())
print(Fore.MAGENTA + '\n' + output_file_name + ' has been printed to ' + save_to_path + '\n')
time_elapsed = float((time.time() - start_time) / 60)
print(Fore.GREEN + "Program has successfully finished after %s minutes" % round(time_elapsed, 2))
```

Functions

```
import os
import re
import pandas as pd
import math
import sys
from colorama import init, Fore, Style
init(autoreset=True)


def createFolderInSitu(directory):
    """This function creates a new folder in the path where the code is being executed from,
        unless the folder had been created previously"""
    if not os.path.exists(directory):
        try:
            os.mkdir(directory)
        except OSError:
            print ('Error: Creating directory. ' + directory)
        else:
            print ("Successfully created the directory %s " % directory)
    else:
        print('%s was already created' % directory)


def writeOput(outname, outdir):
    """"This function creates the path and file name that will used to print information out  """
    # outname = 'name.csv'
    # outdir = './dir'
    if not os.path.exists(outdir):
        os.mkdir(outdir)
    fullname = os.path.join(outdir, outname)
```

```
    return fullname


def get_input():
    """"This function requests information from the user """
    my_input = input('Series of files to be processed (ie. int*.vtk): ')
    if my_input[-4:] != '.vtk':
        raise Exception("Invalid or no argument given for the series")
    return (my_input)


class Node(object):
    """This class is meant to create point's attributes
    that will be used to store the Nodes of the mesh
    from a *.inp file"""

    # The class "constructor" - It's actually an initializer
    def __init__(self, id, coord_x, coord_y, coord_z):
        self.id = id
        self.coord_x = coord_x
        self.coord_y = coord_y
        self.coord_z = coord_z

    def nodeAssigment(label, coord_X, coord_Y, coord_Z):
        node = Node(label, coord_X, coord_Y, coord_Z)
        return node


def keyfunc(afilename):
    """Function to sort file read within a folder so the file_10 is not before file_9 """
    nondigits = re.compile("\D")
    return int(nondigits.sub("", afilename))


def blacklist(input_item, input_list, print_to_file, set_name):
    """This function checks the existence of a item within a list,
    when the item is not in the list it gets appended and a message
     is printed to a external file"""
    for n in range(0, len(input_list)):
        if input_item == input_list[n]:
            break
```

```
        elif input_item != input_list[n] and n != len(input_list) - 1:
            n += 1
        else:
            # print_to_file = open('file_name.txt', 'a+')
            print_to_file.write('*NSET' + ',' + 'NSET=' + set_name + '\n')
            print_to_file.write('%d\n' % input_item)
            input_list.append(input_item)
            # print_to_file.close()


def no_force_line_removed(matrix):
    """"This function iterate over the lines of the DataFrame
    and remove those lines with no Forces in X Y & Z directions """
    i = 0
    for l in range(len(matrix)):
        while i < len(matrix):
            if matrix.iloc[i, 0] == 0 and matrix.iloc[i, 1] == 0 and matrix.iloc[i, 2] == 0:
                matrix = matrix.drop(matrix.index[i])
                continue
            i += 1
    return matrix


def create_folder(directory):
    """This function creates a new folder in the path where the code is being executed from,
    unless the folder had been created previously"""
    if not os.path.exists(directory):
        try:
            os.mkdir(directory)
        except OSError:
            print('Error: Creating directory. ' + directory)
        else:
            print("The directory %s has been successfully created" % directory)
    else:
        print(Fore.RED +'%s was already created. Files might be overwritten!!\n' % directory)


def get_file():
    """"This function requests information from the user """
    my_input = input('Please, introduce the Geometry file generated by Gmsh (ie. specimen*.inp): ')
    if my_input[-4:] != '.inp':
```

```
        raise Exception("Invalid or no argument given for the series")
    return (my_input)


def find_word_line(list_of_lines, milestone, keywords):
    """"This function need a list of lines, a milestone which is the line from where to start reading and a list of
    keywords to look for, when it finds one of the words in the keyword list then return the position of the line that
     contains the word and the word found"""
    for i in range(milestone, len(list_of_lines)):
        for word in keywords:
            list_of_lines[i].find(word)
            if list_of_lines[i].find(word) != -1:
                i = i + 1
                return i, word
            else:
                continue


def file_len(fname):
    """"This function returns the number of lines in the input file"""
    with open(fname) as f:
        for i, l in enumerate(f):
            pass
    return i + 1

def split_list(list_name, line_length):
    """"This function split a long single line list into several lines having each line as many elements as specify by
        the input variable line_length. Abaqus only read set nodes when they have no more than 16 numbers per line"""
    pd.options.display.float_format = '{:,.0f}'.format
    a = len(list_name)
    df = pd.DataFrame()
    number_lines = math.ceil(a / line_length)
    line_break = 1
    j = 0
    k = 0
    for i in range(len(list_name)):
        if i != line_length and i != (line_break * line_length):
            df.loc[j, k] = float(list_name[i])
            k += 1
            continue
        k = 0
```

```
        if i == line_length or i == (line_break * line_length):
            j += 1
            df.loc[j, k] = float(list_name[i])
            k = 1
            line_break += 1
            continue
    return df


def retrieving_input_output_data():
    """"This function retrieves from a external text file the information needed for the code to work"""
    print('It is mandatory to provide a input text file with the key values for the program to work.' +
                       '\n' + 'The input text file has to follow the structure showed below:\n')
    print(Fore.BLUE + '\n-----------input and output path-----------\n' +
          "path_to_inp = Path to the inp file,ie: '/path_to_file/'\n" +
          "csv_path = Path to the csv files, ie: '/path_to_file/'\n" +
          "csv_root = Root of the files. ie: 'ForceAtNodes'\n" +
          "save_to_path = Path to output save folder, ie: '/path_to_folder/'\n" +
          "Output_file_name = output file without the extension, ie: 'foo2'\n" +
          '--------------------------------------\n')
    # print(Style.RESET_ALL)
    input_txt_file = input('Please, introduce the path to the input text file,' + '(ie: /path_to_file/foo.dat): ')
    filename = str(input_txt_file)
    try:
        with open(filename, 'r') as info:
            read = info.read()
            input_data = read.split("'")
            path_to_inp = input_data[1]
            inp_file_name = input_data[3] + '.inp'
            csv_path = input_data[5]
            csv_root = input_data[7] + '*.csv'
            save_to_path = input_data[9]
            output_file_name = input_data[11] + '.inp'
        return path_to_inp, inp_file_name, csv_path, csv_root, save_to_path, output_file_name
    except FileNotFoundError:
        print("Oops!  Input Text File wasn't found.\n")


def find_elements_physical_groups(geometry_file, milestone, pathToOutput):
    """"This function need a list of lines, a milestone which is the line from where to start reading and a list of
    keywords to look for, when it finds one of the words in the keyword list then return the position of the line that
```

```
    contains the word and the word found"""

    nodesCounter = []
    with open(geometry_file, 'rt') as in_file:  # open file and close it after read.
        # This type of element is the only available in GMSH, Calculix, ccx2paraview.
        element_types_list = ['=C3D8']
        lines = []
        for line in in_file:  # Store each line in a string variable "line"
            lines.append(line.lstrip())
        linesWithElement = []
        for i in range(milestone, len(lines)):
            for word in element_types_list:
                lines[i].find(word)

                if lines[i].find(word) != -1:  # -1 means Not Exist
                    linesWithElement.append(lines[i])
                    i = i + 1
                    nodesCounter.append(i)

                else:
                    continue
        linesWithNodeSet = []
        linesWithElementSet = []
        indexForNodeSet = []
        indexForElementSet = []
        for i in range(len(lines)):
            if lines[i].find('*ELSET') != -1:  # -1 means Not Exist
                linesWithElementSet.append(lines[i])
                i = i + 1
                indexForElementSet.append(i)

            if lines[i].find('*NSET') != -1:  # -1 means Not Exist
                linesWithNodeSet.append(lines[i])
                i = i + 1
                indexForNodeSet.append(i)

    solidList = []
    elementSetLabels = []
    nodeSetLabels = []
    # Loop to store the Label of the solid defined by the elements(C3D8)
    for i in range(len(linesWithElement)):
```

```
        solidList.append(re.findall(r'ELSET=(\w+)', linesWithElement[i]))
    for i in range(len(linesWithElementSet)):
        elementSetLabels.append(re.findall(r'ELSET=(\w+)', linesWithElementSet[i]))
    for i in range(len(linesWithElementSet)):
        nodeSetLabels.append(re.findall(r'NSET=(\w+)', linesWithNodeSet[i]))


    oldSolidList = []
    newSolidList = []
    for i in range(len(solidList)):
        oldSolidList = ','.join(solidList[i])
        newSolidList.append(oldSolidList)
        solidList.append(oldSolidList)
    newSolidList = ', '.join(newSolidList)


    element_list = []


    for i in range(len(nodesCounter)):
        counter = nodesCounter[i]
        # print(counter)
        if i == 0:
            element_list.append(linesWithElement[i])
        for element in range(counter, len(lines)):  # iterate over lines after where the 'keyword' was found
            str_line = lines[element]  # store lines in list 'str_line'
            if str_line[0].isdigit():  # if the first character is a digit then store it in nodes[]
                element_list.append(str_line)
            elif (i+1) >= len(nodesCounter):
                element_list.append('*ELSET, ELSET=EAll\n')
                element_list.append(newSolidList)
                element_list.append('\n')
                break
            else:  # when first character is different to a digit then break the loop
                element_list.append(linesWithElement[i+1])
                break

    for i in range(len(indexForNodeSet)):
        counter = indexForNodeSet[i]
        # print(counter)
        # a = len(lines)
```

```python
        # print('a=', + a)
        if i == 0:
            element_list.append(linesWithNodeSet[i])
        for element in range(counter, len(lines)):  # iterate over lines after where the 'keyword' was found
            str_line = lines[element]  # store lines in list 'str_line'
            if str_line[0].isdigit():  # if the first character is a digit then store it in nodes[]
                element_list.append(str_line)
            elif (i+1) >= len(nodeSetLabels):
                break
            else:  # when first character is different to a digit then break the loop
                element_list.append(nodeSetLabels[i+1])
                break

    outputFileName = 'geometry.txt'

    with open((pathToOutput + outputFileName), "w") as myfile:
        # Remove quotation marks in the lines of the list "element_list
        lst = map(str, element_list)  # Convert all of the items in lst to strings (for str.join)
        # Join the items together with commas
        geometry = "".join(lst)
        # Write to the file
        myfile.write(geometry)
    return nodeSetLabels, geometry


def listToStringWithoutBrackets(list1):
    return str(list1).replace("[", '').replace("]", '').replace("'", "")
```