# Predictive Modelling of Tribological Systems using Movable Cellular Automata

Nadia Salman

Submitted in accordance with the requirements for the degree of
Doctor of Philosophy

**The University of Leeds**
**Centre of Doctoral Training in Integrated Tribology**
**School of Mechanical Engineering**

October 2019

The candidate confirms that the work submitted is her own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

Chapter 4, on the implementation and the verification of the code, include material published jointly with my supervisors in:

Salman, N., Wilson, M., Neville, A. & Smolin, A. Implementation of MCA in the framework of LIGGGHTS. In *V International Conference on Particle-based Methods – Fundamentals and Applications, Eds: P. Wriggers, M. Bischoff, E. Onate, D.R.J. Owen & T. Zohdi,* pp. 767-777 (2017).

The code was developed by the candidate with technical guidance from A. Smolin. The results were generated, analysed and discussed by the candidate with the general supervision of the co-authors M. Wilson, A. Neville and A. Smolin. The paper was drafted by the candidate and reviewed by the co-authors.

*Dedicated to my Mother and Father, for their strength, endless love and support, and to my three sisters and brother, for always believing in me.*

## Acknowledgements

I would like to express my gratitude to whom without their support this PhD would have not been completed.

I would like to thank my advisor Dr Mark Wilson for his support and guidance I needed to pursue my work throughout the years. Thank you for helping me shape my research skills, for always encouraging me to keep going, and for guiding me while giving me the freedom to pursue and develop my own interests. I would also like to thank my advisors Professor Anne Neville and Professor Ardian Morina. Thank you, Anne, for giving me the opportunity to work on this project, for your guidance during the process, and for always being there for support.

I would like to express my deepest gratitude to my advisor Professor Alexey Smolin and acknowledge the big role he had in the completion of this work. Thank you for your willingness to collaborate on this project. I know how busy you are but you still always had time for me. Thank you for your time and great effort to ensure its completion, and for your continuous patience and support throughout the years. I could not have done this without you and consider myself very lucky to have had you as my mentor.

I am grateful for the director of the CDT in Integrated Tribology, Professor Rob Dwyer-Joyce, thank you for giving me the opportunity to join the CDT and for all the great times I had being part of it. I especially will always remember all the conferences and events I joined as part of the CDT. I would also like to thank, Kim, the manager of the CDT. I cannot thank you enough for all the support and kindness you have showed since day one. It was always comforting to know you are always there for us. I am also very grateful to my colleagues at both the University of Leeds and the University of Sheffield, for all the times we shared and the connections we built.

I cannot end this without expressing my greatest gratitude and appreciation for everything my parents did for me throughout my life, and especially for their support, encouragement and motivation during these past few years. Thank you for everything, you have shaped who I am today and I will always be grateful. I also want to thank my siblings. Amal and Zahraa, thank you for being the best younger sisters anyone could hope for and for giving me a reason and motivation to be the best I can be, to always keep going and never give up. Samira, thank you for being the best older sister one could hope for, the one I can go to with anything and know you will understand and have my back. Yunes, thank you for all always having a caring heart. Last but not least, my best friends, Meriam and Ola, thank you for always being there and for being my rock even though we are miles apart.

## Abstract

In the science of tribology, where there is an enormous degree of uncertainty, mathematical models that convey state-of-the-art scientific knowledge are invaluable tools for unveiling the underlying phenomena. A well-structured modelling framework that guarantees a connection between mathematical representations and experimental observations, can help in the systematic identification of the most realistic hypotheses among a pool of possibilities.

This thesis is concerned with identifying the most appropriate computational model for the prediction of friction and wear in tribological applications, and the development of a predictive model and simulation tool based on the identified method. Accordingly, a thorough review of the literature has been conducted to find the most appropriate approach for predicting friction and wear using computer simulations, with the multi-scale approach in mind. It was concluded that the Movable Cellular Automata (MCA) method is the most suitable method for multi-scale modelling of tribological systems.

It has been established from the state-of-the-art review in Chapter 2 of this thesis, that it is essential to be able to model continuous as well as discontinuous behaviour of materials on a range of scales from atomistic to micro scales to be able to simulate the first-bodies and third body simultaneously (also known as a multi-body) in a tribological system. This can only be done using a multi-scale particle-based method because continuum methods such as FEM are none-predictive and are not capable of describing the discontinuous nature of materials on the micro scale. The most important and well-known particle-based methods are molecular dynamics (MD) and the discrete element methods (DEM). Although MD has been widely used to simulate elastic and plastic deformation of materials, it is limited to the atomistic and nanoscales and cannot be used to simulate materials on the macro-scale. On the other hand, DEM is capable of simulating materials on the meso/micro scales and has been expanded since the algorithm was first proposed by Cundall and Strack, in 1979 and adopted by a number of scientific and engineering disciplines. However, it is limited to the simulation of granular materials and elastic brittle solid materials due to its contact configurations and laws. Even with the use of bond models to simulate cohesive and plastic materials, it shows major limitations with parametric estimations and validation against experimental results because its contact laws use parameters that cannot be directly obtained from the material properties or from experiments.

The MCA method solves these problems using a hybrid technique, combining advantages of the classical cellular automata method and molecular dynamics and

forming a model for simulating elasticity, plasticity and fracture in ductile consolidated materials. It covers both the meso and micro scales, and can even "theoretically" be used on the nano scale if the simulation tool is computationally powerful enough. A distinguishing feature of the MCA method is the description of interaction of forces between automata in terms of stress tensor components. This way a direct relationship between the MCA model parameters of particle interactions and tensor parameters of material constitutive law is established. This makes it possible to directly simulate materials and to implement different models and criteria of elasticity, plasticity and fracture, and describe elastic-plastic deformation using the theory of plastic flow. Hence, in MCA there is no need for parametric fitting because all model parameters can be directly obtained from the material mechanical properties.

To model surfaces in contact and friction behaviour using MCA, the particle size can be chosen large enough to consider the contacting surface as a rough plane, which is the approach used in all MCA studies of contacting surfaces so far. The other approach is to specify a very small particle size so that it can directly simulate a real surface, which allows for the direct investigation of material behaviour and processes on all three scale levels (atomic, meso and macro) in an explicit form. This has still been proven difficult to do because it is too computationally extensive and only a small area of the contact can be simulated due to the high numbers of particles required to simulate a real solid. Furthermore, until now, no commercial software is available for MCA simulations, only a 2D MCA demo-version which was developed by the Laboratory of CAD of Materials at the Institute of Strength Physics and Materials Science in Tomsk, Russia, in 2005. The developers of the MCA method use their own in-house codes.

This thesis presents the successful development of a 3D MCA open-source software for the scientific and tribology communities to use. This was done by implementing the MCA method within the framework of the open-source code LIGGGHTS. It follows the formulations of the 3D elastic-plastic model developed by the authors including Sergey G. Psakhie, Valentin L. Popov, Evgeny V. Shilko, and the external supervisor on this thesis Alexey Yu. Smolin, which has been successfully implemented in the open-source code LIGGGHTS. Details of the mathematical formulations can be found in [1]–[3], and section 3.5 of this thesis.

The MCA model has been successfully implemented to simulate ductile consolidated materials. Specifically, new interaction laws were implemented, as well as features related to particle packing, particle interaction forces, bonding of particles, and others. The model has also been successfully verified, validated, and used in simulating indentation. The validation against experimental results showed

that using the developed model, correct material mechanical response can be simulated using direct macroscopic mechanical material properties.

The implemented code still shows limitations in terms of computational capacity because the parallelization of the code has not been completely implemented yet. Nevertheless, this thesis extends the capabilities of LIGGGHTS software to provide an open-source tool for using the MCA method to simulate solid material deformation behaviour. It also significantly increases the potential of using MCA in an HPC environment, producing results otherwise difficult to obtain.

# Table of Contents

# List of Figures

# List of Tables

# List of Notations

| | |
|---|---|
| $a_i$ | acceleration of particle *i* |
| $A_{ij}$ | area of interaction / contact area between particles *i* and *j* |
| $A_{ij}^0$ | initial contact area between particles *i* and *j* at time-step=0 |
| $c_p$ | Longitudinal (pressure) wave velocity |
| $d_i$ | size/diameter of particle *i* |
| $D_{\alpha\beta}$ | stress deviator |
| $D'_{\alpha\beta}$ | corrected stress deviator |
| $E$ | Young's/elastic modulus |
| $E_h$ | strain work hardening modulus |
| $F_i$ | total force acting on particle *i* from surroundings |
| $F_{ij}$ | pair-wise component of the total force acting on particle *i* from *j* |
| $F_{ij}^{\Omega}$ | volume-dependent component of total force acting on particle *i* from *j* |
| $F_{ij}^{n}$ | normal interaction force between particles *i* and *j* |
| $F_{ij}^{t}$ | tangential interaction force between particles *i* and *j* |
| $F_f$ | frictional force |
| $F_n$ | applied normal force |
| $g$ | gravitational acceleration |
| $G$ | shear modulus |
| $h_{ij}$ | overlapping parameter of particles *i* and *j* |
| H | hardness |
| $H$ | plastic modulus / work hardening |
| $\hat{J}_i$ | moment of inertia of particle *i* |
| k | wear coefficient |
| $k_c$ | adhesion stiffness parameter |
| $k_n$ | elastic stiffness parameter |
| $k_r$ | rolling stiffness parameter |
| $k_t$ | frictional stiffness parameter |

| | |
|---|---|
| K | tangent modulus |
| $K$ | bulk modulus / volume compression modulus |
| $K_{ij}$ | torque in the particle pair *i-j* due to relative rotation |
| $l_{ij}$ | relative shear displacement of particles *i* and *j* |
| $m_i$ | mass of particle *i* |
| $M_i$ | resultant torque acting on particle *i* from surroundings |
| $M_{ij}$ | momentum of the force of interaction between particles *i* and *j* |
| $M_{ij}^r$ | torque caused by rolling friction |
| $M_{ij}^t$ | torque caused by tangential force |
| M | coefficient of stress drop |
| $\vec{n}_{ij}$ | unit vector directed along the line between mass centers of *i* and *j* |
| $N$ | number of particles |
| $N_c$ | coordination number |
| $N_i$ | number of neighbouring particles of particle *i* |
| $P$ | pressure |
| $q_{ij}$ | distance from mass center of *i* to center of area of interaction with *j* |
| $q_{ji}$ | distance from mass center of *j* to center of area of interaction with *i* |
| $r_i$ | radius of particle *i* (also referred to as Ri) |
| $r_{ij}$ | distance between mass centers of particles *i* and *j* |
| $r_{ij}^0$ | initial distance between mass centers of particles *i* and *j* at time-step=0 |
| $r_{list}$ | total neighbour list range radius |
| $r_{cut}$ | cut-off radius |
| $r_{shell}$ | shell thickness radius |
| s | sliding distance |
| t | time |
| $t_{ij}$ | specific force of the tangential interaction of the particle pair *i-j* ($F_{ij}^t$) |
| $T$ | temperature |
| $T_R$ | Raleigh time |
| $v_{ij}$ | relative velocity between particles *i* and *j* |

| | |
|---|---|
| $v_{ij}^{n}$ | normal component of the relative velocity between particles *i* and *j* |
| $v_{ij}^{t}$ | tangential component of the relative velocity between particles *i* and *j* |
| V | volume of material |
| $'$ | denoted scaled value of considered parameter in algorithm of Wilkins |
| *i, j* | sub/superscripts denoting particle number |
| *cur* | superscripts denoting current time-step |
| *pre* | superscripts denoting previous time-step |
| $\alpha, \beta$ | superscripts denoting axes in local coordinate system of pair of particles |

# List of Acronyms

| | |
|---|---|
| BEM | Boundary Element Method |
| CA | Cellular Automata |
| CEB | Chang-Etsion-Bogy |
| COF | Coefficient of Friction |
| DEM | Discrete Element Method |
| DMT | Derjaguin, Muller and Toporov Model |
| EAM | Embedded Atom Method |
| EHL | Elastohydrodynamic |
| EPSD | Elastic-Plastic Spring-Dashpot Model |
| FCC | Face-Centered Cubic |
| FEM | Finite Element Method |
| GW | Greenwood and Williamson |
| HCP | Hexagonal Closed-Packed |
| HPC | High Performance Computer |
| JKR | Johnson, Kendall and Roberts Model |
| LAMMPS | Large-Scale Atomics/Molecular Massively Parallel Simulator |
| LIGGGHTS | LAMMPS Improved for General Granular and Granular Heat Transfer Simulations |
| MAAD | Macroscopic, Atomistic, Ab Initio Dynamics |
| MCA | Movable Cellular Automata |
| MD | Molecular Dynamics |
| MML | Mechanically Mixed Layer |
| MPI | Message Passing Interface |
| OMP | Open Multi-Processing |
| PBC | Periodic Boundary Conditions |
| SC | Simple Cubic |
| ZMC | Zhao-Maietta Chang Model |

# Chapter 1
# Introduction

## 1.1 Motivation

Tribology is the science that deals with bodies in contact and in relative motion, and although it is an important and long studied science, tribological systems, their behaviour and phenomena are still poorly understood. It is still considered a difficult subject due to its multi-disciplinary, multi-scale and multi-physical nature [4]. Initially the science of tribology was based on solid and fluid mechanics theories, but the discrete pattern of contacts has recently been considered, due to the nature of tribological contacts and their physical complexities [5], [6].

When a body is moving tangentially to a surface such as in sliding or rolling motion, frictional forces are produced leading to energy loss, wear and deformation of surfaces, limiting the lifetime of mechanical systems. The modelling of friction and wear phenomena is very complex and non-linear, and despite the vast interest in the field and the increasing capabilities of computational modelling, no practical and comprehensive friction models exist that can show and predict all the aspects of friction dynamics observed experimentally. Models available for friction and wear are mostly empirical models that are very limited and only work for specific materials and test conditions.

Modelling efforts of frictional contacts appear to follow two mutually exclusive philosophies; phenomenological and physics-based models. The first depends on experimental observations and conditions to get the tangential force and the relative displacement between two contacts to obtain a general friction behaviour by curve fitting an equation to the obtained data [7]. These models do not involve any information on the underlying mechanisms, are very hard to generalize, and hence lack predictive capabilities for different conditions. On the other hand, physics-based models consider the various aspects involved such as; the material's mechanics and mechanical properties, and the interface chemistry; to develop an understanding of the frictional behaviour from the local physics of the system on different scales; from atomic to macro scales, as shown in Figure 0-1.

The most commonly used phenomenological model of frictional contacts is Coulomb's model for friction. Coulomb's law states that there is a linear relationship between the tangential frictional forces ($F_f$) and normal forces ($F_n$) at a contact, where the proportionality constant is the coefficient of friction:

$$\mu = \frac{F_f}{F_n} \tag{1.1}$$

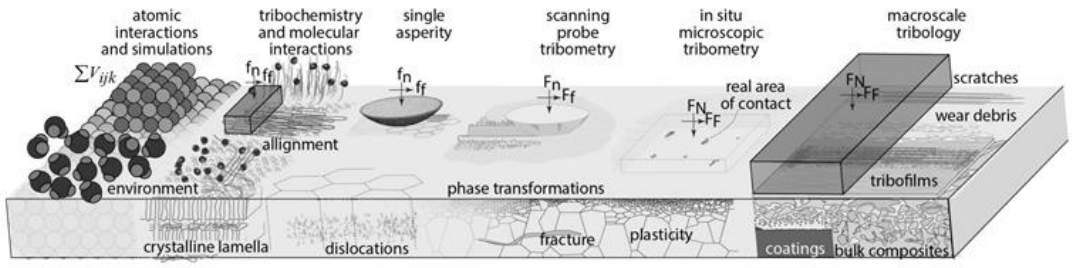



**Figure 0-1: Diagram showing the multi-scale complexity of interaction in frictional contacts [8]**

The coefficient of friction has no physical meaning and the need for determining its value beforehand, limits the predictive capability of the model immensely.

These models are also constricted to static and elastic cases, however when tangential forces are applied, elastic-plastic deformations occur as well as slip over the contact which all greatly affects the frictional behaviour of the system. Hence friction should be described in some locally distributed, more fundamental form than just the coefficient of friction which is only useful in mechanical design, but not in the fundamental understanding of friction and wear [9].

To overcome these drawbacks, dynamic friction models were proposed [10]–[12] where the surfaces in contact are assumed to have a number of asperities in contact; which are surface irregularities as shown in Figure 0-2, and an average deformation parameter of these asperities is assumed based on experimental observations, which again makes them phenomenological models.

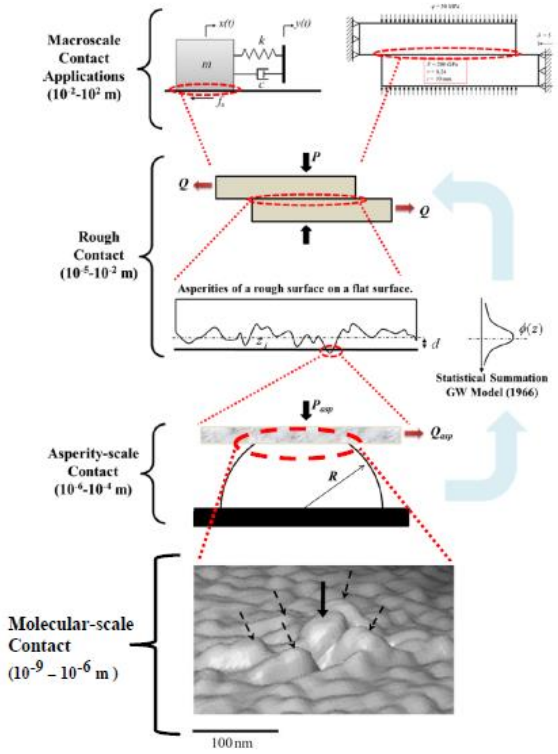



**Figure 0-2: Multi-scale aspect of frictional contacts [13]**

A persisting question in the field of tribology is where does friction come from? What is the origin of friction? The answer is not simply the macroscopic result of blocks interacting with each other, because friction is independent of the apparent contact area. The actual contact occurs at asperity contacts on the microscopic scale as shown in Figure 0-3. The behaviour of contacts at the small scales is fundamentally different from that on the macro scale which is mainly due to the surface interactions. When the surfaces' energy reaches the order of magnitude of other characteristic energies of the system such as elastic strain for example, or kinetic energies, the adhesive effects start to play a major role and the continuum representation of the media is no longer valid. Hence, physics-based models use constitutive laws to relate local stress and strain fields from the fundamental knowledge of the material behaviour. Bowden and Tabor were the first to attempt to describe a friction physics-based model in the 1950s [14] at asperity contacts, where they defined the coefficient of friction as the ratio between the shear strength of the material and its hardness; however they assumed that only elastic deformation occurs; they did not consider the difference between normal and tangential loading.

At the asperity scale, as shown in Figure 0-3, it may be sufficient to study elastic behaviour under steady state contact, but the nature of surfaces on the macroscale is very different with random distributions of asperities, thus scaling up and extending the asperity scale contacts to multi-asperity macro scale contacts becomes very challenging. This is mainly done using statistical summation and fractal characterization techniques where the height distribution of asperities are acquired by measurements of surface roughness and added to the nominal contact area so that the contribution of each asperity is taken into consideration. Here it is assumed that the surface is isotropic and the asperities do not interact with each other, in addition to using empirical superposition of stress-strain relationships under elastic-plastic conditions; which is not a physical representation.

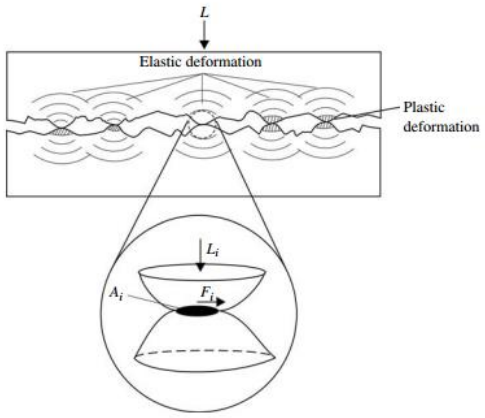



**Figure 0-3: Two rough surfaces in contact; elastic deformation at regions surrounding the contact junctions and plastic deformation occurs at some of the junctions. The expanded view shows the contact area at the contact zone between two contacting asperities [15]**

Furthermore, wear is the amount of material loss that happens during the contact of surfaces, and it is the least predictable phenomenon in tribological systems mainly due to the incomplete knowledge of the wear rate for the appropriate material pair in the system. Modelling of wear has been extensively researched to obtain predictive equations [16]. The first and main phenomenological wear model produced is Archard's wear model [17] which assumes that the volume of material which is removed (V) for a sliding distance (S) is directly proportional to the applied normal load ($F_n$) and the hardness of the softer material (H), where the proportionality constant is the wear coefficient ($k$):

$$\frac{\mathbf{V}}{\mathbf{S}} = \boldsymbol{k}\frac{\boldsymbol{F_n}}{\mathbf{H}} \tag{2.2}$$

When the equation did not apply correctly for a specific case, the model was modified to suit it. For example, for highly elastic materials at asperity contacts, a model [18] was developed where it related the volume of material removed to the coefficient of friction:

$$\frac{\mathbf{V}}{\mathbf{S}} = \boldsymbol{k}\frac{\boldsymbol{F_n}}{\mathbf{H}}\sqrt{\mathbf{1}+\mathbf{3}\mu^{\mathbf{2}}} \tag{3.3}$$

Other models use a qualitative approach using experimental data, however these models are only suitable for specific materials and conditions and thus, again, lack the predictive capabilities [9].

It is also believed that a different approach is needed to be able to predict friction and wear as stated in [9]. They, and many others, suggest to not model wear by following the current conventional wear mechanisms known, which are shown in Figure 0-4, but to consider alternatives. To develop a full picture of what happens at the macroscopic sliding surfaces and how the fragmented particles on the interface move and change, should be represented by a more local and fundamental way other than the coefficient of friction because friction forces change the stresses and temperatures at the interfaces.

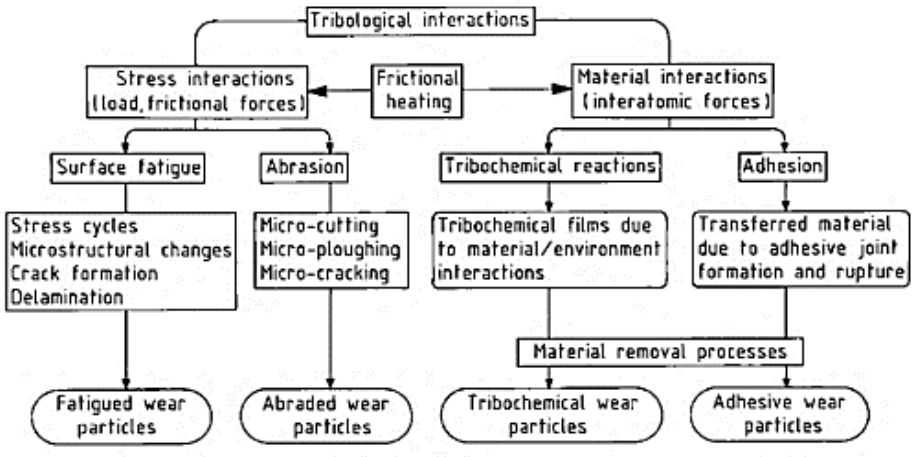



**Figure 0-4: Tribological solid/solid interactions and wear mechanisms [19]**

## 1.2 Scope of Thesis

This thesis is one of many to attempt to explain and model friction and wear phenomenon in dry contact. It works towards the long-term goal of predicting friction and wear behaviour on the macroscale contact of surfaces based on the material and surface properties, while also considering the underlying microscopic mechanisms. The literature review presented in Chapter 2 reveals that the lack of predictive models of friction and wear is mainly due to the lack of understanding of microscopic and macroscopic fracture and plastic deformation in tribological systems. Specifically, the role of plasticity is still poorly investigated and is not included in most macroscale friction models, and only included implicitly in some wear models. The state-of-the-art-review in Chapter 2 shows that a physical particle-based model; such as molecular dynamics (MD) or discrete element method (DEM), is the best approach to model elastic-plastic deformation on different scales, however they are still limited. Hence, an improved particle-based model and simulation tool for the simulation of elastic-plastic deformation of materials on different scales is needed to better understand and predict friction and wear.

## 1.3 Aims & Objectives

In this frame, the immediate goal is to develop a numerical tool that can simulate complex material behaviour on different scales; specifically, plastic deformation. The long-term vision is to use this model to simulate and predict friction and wear behaviour in tribological systems.

The objectives of this project are as follows:

1- Review the relevant computational methods at different scales in literature to identify a suitable approach for practical predictive modelling of tribological systems.
2- Develop a computational code/tool capable of direct numerical simulation of friction and wear (i.e. capturing actual damage to material).
3- Validate the model by comparing against experimental data.
4- Test the model and code to study its sensitivity and performance.
5- Simulate different tribological systems to explore fundamental mechanisms of friction and wear on different scales.

## 1.4 Contributions

The developed simulation tool can be used to simulate elastic-plastic behaviour of solid material on different scales. It provides the community with an open-source 3D mesoscopic particle-based simulation tool based on the Movable Cellular Automata Method (MCA), since currently there only exists a 2D MCA demo version available for the public. The following section attempts to breakdown and place the research into context. The approach for this completed work is shown in Figure 0-5.

- *Review of the state of the art*, and the identification of the most suitable approach for our aim of prediction of friction and wear. MCA was identified as the most appropriate computational method. This is covered in Chapter 2.
- *Computational and Numerical methodology*. Identifying the best platform for implementing the MCA model, which was chosen to be LIGGGHTS open-source code which is based on the Discrete Elements Method (DEM). Presenting the theoretical background of DEM and MCA, and identifying their differences to classify the MCA functionalities that need to be added in LIGGGHTS. This is covered in Chapter 3.
- *Development and implementation of the 3D MCA elastic-plastic model in LIGGGHTS*, and verification of the developed model for the simulation of fracture and plastic deformation. This is covered in Chapter 4.
- *Validation against experimental data and investigating the convergence, sensitivity and computational performance of the model and code.* This is covered in Chapters 5 and 6

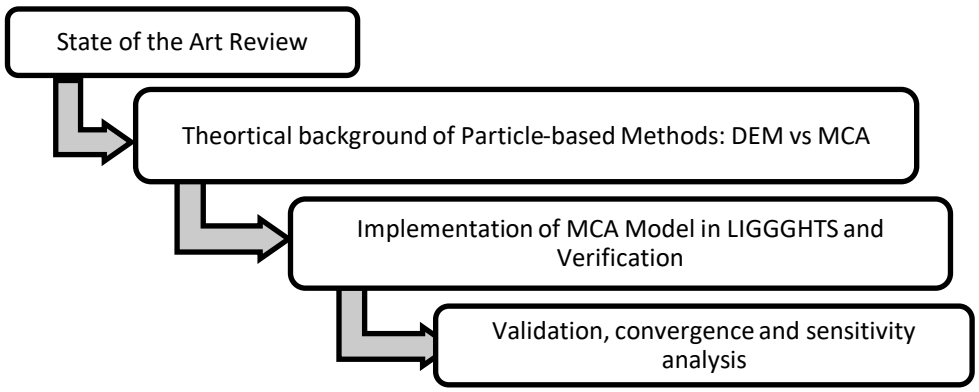



**Figure 0-5: Thesis road map**

# Chapter 2
# State of the Art Review

## 2.1 Introduction

This chapter fulfils the first object of reviewing the literature on the modelling and simulation of tribological systems. Here the advantages and disadvantages of each method is highlighted with the aim to identify the most suitable method for the prediction of friction and wear. This Chapter focuses on the following:

- An overview of the different modelling techniques for material bulk behaviour, with emphasis on multi-scale modelling approaches.
- An overview of the modelling methods and tools used in the study of tribological systems; highlighting the gaps between micro and macro scales.
- Finally, concluding with identifying the most appropriate approach for practical predictive modelling of tribological systems; depending on the most appropriate scale levels.

## 2.2 Materials Simulation and Multi-Scale Modelling

All physical phenomena depend mainly on the materials, their structure, behaviour and reaction to the environment. To understand the basic behaviour of friction and wear in mechanical systems, the ability to properly describe the material behaviour is necessary [20]. As far as modern tribology is concerned, understanding the elementary friction (energy dissipation) and wear (material loss) processes is one aspect of it, and the other is the selection and development of materials using modelling and experimental studies for advanced applications.

In physics-based modelling of materials, there are two main approaches; the continuum approach and the particle-based or discrete approach. They are based on two fundamentally different theories, where each has its advantages and limitations.

From the continuum perspective, materials are described without taking into consideration the inhomogeneities and internal structure of the material; it is assumed that the medium can be infinitely divided without changing its properties [20], which is why it is used in modelling macroscopic behaviour but not capable of describing microscopic behaviours on the smaller scales. The most well-known and widely used continuum modelling method is the finite element method (FEM); which was developed in the 1950s by Argyris and Kelsey [21] and Turner et al [22], other methods are the finite difference and boundary element methods (BEM).

In continuum mechanics, external loads whether forces or displacements, change the shape of the bodies causing deformation of the material. As shown in Figure 2-1, a material of a body ($B$) with a surface ($\partial B$) can be represented as a continuous distribution of an infinite number of continuum particles [23]. This particle or mesh, on the macro scale is a point with no mass or volume just like in a geometrical shape, which is why is not considered to be a small piece of material. However, this continuum particle obtains its properties from a finite-sized region (*l)* on the microscale as shown in Figure 2-1. The properties of these particles can be thought of as an average of the atomic behaviours within this domain. Furthermore, here constitutive laws; determined experimentally or guessed intuitively, are used to describe the material's response and deformation process. One does not need to consider the underlying reasons for these responses. This approach, which is based on fitting information based on observed phenomena, cannot be used to understand fundamental mechanisms or predict behaviour of materials [23].

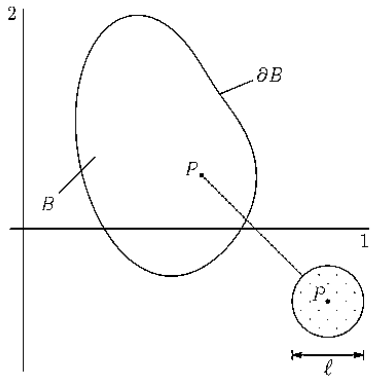



**Figure 2-1: A schematic of a material with body B and a surface ∂B, where P is the continuum particle representing the atomic structure (the dots around P) of length scale *l* [23]**

On the other hand, in the particle-based or discrete approach, the material is modelled as an assembly of separate discrete particles or elements. Particle-based methods actually originate from the molecular dynamics (MD) method at the atomic scale. However, it could be applied on different scales; such as atoms or molecules on the atomic scale, and grains or solid particles on the meso /micro scales and even on the macro scales. This approach is a direct and straightforward way to model mechanical behaviour of materials at different scales, as shown in Figure 2-2, by applying particle-particle interaction laws.

In atomistic modelling, each atom is modelled as an individual particle of the material that cannot be divided any further. The discreteness of the material is explicitly considered, and the associated questions can hardly ever be solved analytically. Thus, numerical simulations used to implement these models are implemented by modelling the motion of the atoms over a certain time span. Monte Carlo and MD methods are the main atomistic methods used; they are used in all

different types of studies that involve the process of diffusion in solids, and the interaction of a material with its environment. MD is the most used and the details of the approach are thoroughly reviewed in the books written by Allen and Tildesley in 1987 [24] and Frenkel and Smith in 1996 [25]. The interaction forces (potential functions) used in the atomistic methods are usually generated by Quantum Mechanics models.

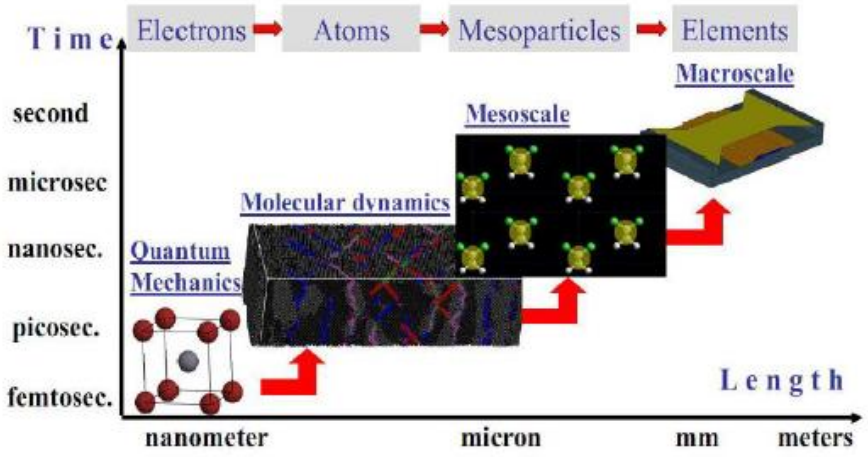



**Figure 2-2: Materials across the different scales [26]**

Even though the physical dimension of an atomic scale simulation is very small; a few nanometres, the system contains huge numbers of particles; maybe billions of atoms. It is impossible to study and predict the behaviour of such systems experimentally; however, these computational studies are very expensive in terms of computational power. Atomistic models are used to probe and gain a better understanding of various fundamental phenomena and mechanisms of materials, their causes and effects; which also helps in developing and optimizing the materials [20]. Its significance also shows because miniaturization has been gaining an increasing interest, and many modern technologies involve the nanometre scale; such as thin films, nano-composites, semi-conductors, etc, as shown in Figure 2-3.

Discrete methods are also used on micro and macro scales involving granular or weakly bonded materials. The most well-known method is the discrete element method (DEM) originally developed by Peter Cundall in the 1970s [27], [28], and it is considered very closely related to MD, however it is not capable of investigating phenomena at the atomic or molecular scales.

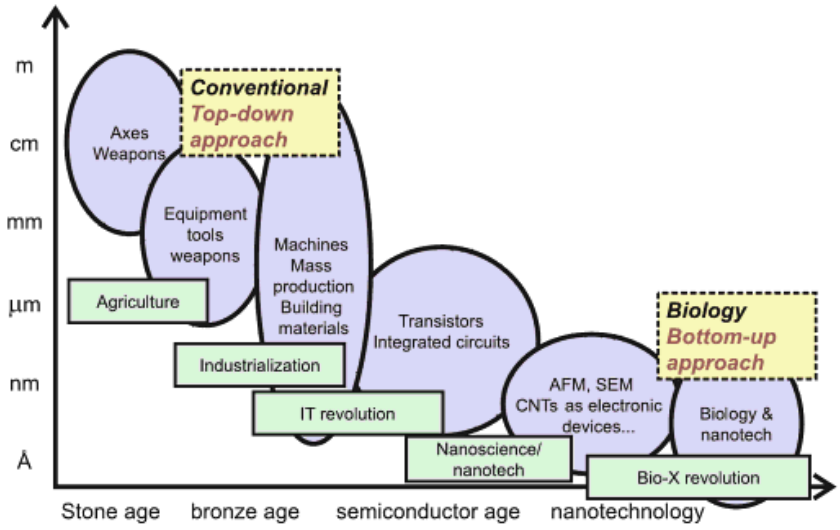



**Figure 2-3: The characteristic material scales of technological eras [5]**

DEM uses the equations and laws of motion to manage the interaction between the particles in the system. The Equations of motion manage the conservation while the laws of interactions play the role of constitutive equations. Despite the variations of DEM, all models rely on three main parts: 1- time integration of the mechanics' equations 2- detection of contact between system particles 3- computation of interaction forces [5].

The disadvantages of these methods are their high computational costs and time, and difficulty to validate experimentally in great detail. However, the increasing speed of computers and the simplifications possibly made in the models made it much more popular. Discrete simulations are also good starting points to model multi-scale phenomena and could possibly be linked to continuum methods.

Engineers mostly use tools that are based on the continuum mechanics theories such as the finite element method (FEM) [20] as shown in Figure 2-4, and only recently, in the 1980s, scientist and engineers started to consider atomistic and discrete descriptions in their models. For many applications and phenomena; including tribological systems; which will be shown later on, the two approaches are actually complementary and if bridged properly it will have a great impact on the understanding of material behaviour and complex processes.

Multi-scale modelling aims to bridge the two viewpoints to be able to bridge the materials scales, which is usually done by introducing the intermediate mesoscopic methods. Another strong motivation for the use of multi-scale models is saving computational cost and time; making it possible to access the different length and time scales needed.

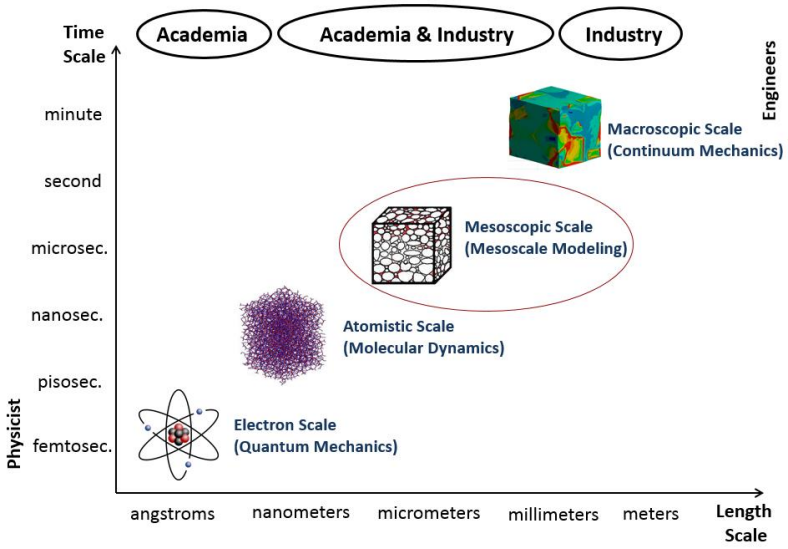



**Figure 2-4: Computational modelling methods across the scales**

There are two main distinguished approaches for multi-scale modelling; the hierarchical and the concurrent approaches [29]. In the hierarchical methods, also known as the coupling of scales, different methods and tools are used on the different scales in sequence, starting from the atomic scale going up to the macro scale. Firstly, the quantum mechanics method is used to find the interatomic potentials and force fields needed to be fed into an MD simulation. Here MD is used to develop an understanding of the microstructure of the material, which will then be fed into a finite element simulation to get macroscopic data. In the concurrent multi-scale methods, the two or more scales are simultaneously simulated, rather than using the data from one scale to generate models on a larger scale. These type of simulations were first reported in the 1990s [20].This is mostly done by dividing the computational domain into different regions where different simulation methods are applied but at the same time, where the information at the small scales are taken and input on the fly into the larger scales.

The difficult and critical problem in these methods is the correct coupling of the different models. The correct and accurate mechanical, physical and thermodynamical transformation between these methods is extremely tricky and challenging, however when validated, are very insightful. One of the pioneering works in multi-scale modelling is called the MAAD approach (macroscopic, atomistic, ab initio dynamics) which was done by Abraham et al. in 1998 [30]. More recently, two review papers were written by Curtin and Miller [31] and Lui et al. [32] were they comprehensively covered the field of multi-scale modelling.

The next section covers the main approaches used in the simulation and modelling of tribological systems and the best approach for the multi-scale prediction of friction and wear is chosen accordingly.

## 2.3 Tribological Triplet Concept

Tribological behaviours are system dependent, and most tribomechanical systems can be divided into several tribological sub-systems. For example, the transmission gearbox in a vehicle is very complex and could be divided into eight tribosystems, some of which are the roller bearings, spur gears and sealing elements, as mentioned in [33]. Each of these tribosystems could be tested or modelled separately, however, all systems are defined by a set of fundamental components as shown in Figure 2-5; which are the counter body and base body, also known as the first bodies, and the interface between them, known as the third body, in addition to their operating conditions; type of motion, load, speed and operating environment. The analysis of the system is very important to be able to select the most appropriate material of each component for the given mechanism, operating conditions and environment.

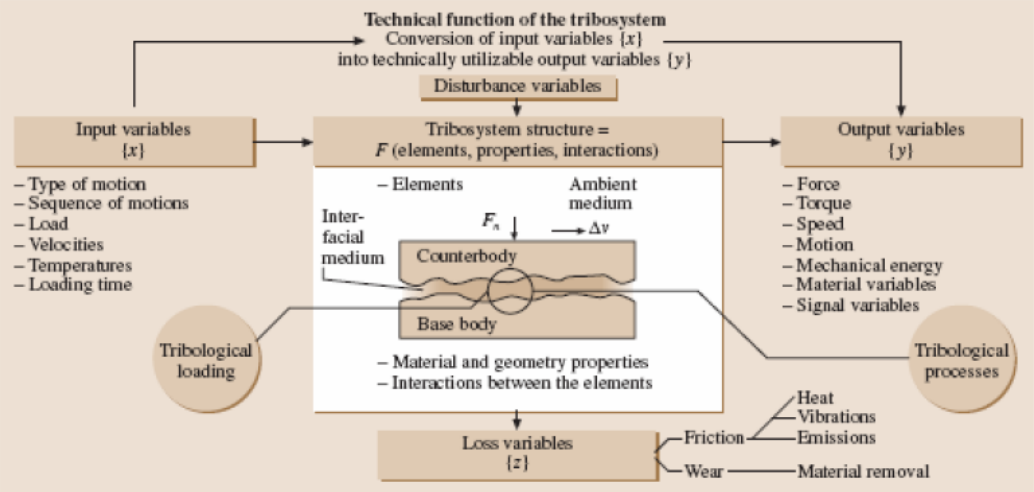



**Figure 2-5: Expanded representation of a tribological system [34]**

The interfacial material called the third body is mainly formed by the degradation of the first-bodies, as shown in Figure 2-6. It can also enter into the contact from the outside. Many mechanical transformations occur in the third body, such as damage, fracture, phase change, plasticity, fatigue, etc., as well as physical phenomena such as chemical reactions, heat production, electrostatic interactions, etc. This is why the third body is particularly complicated to study, including that its confinement makes it hard to study experimentally. However, experimental observations revealed that the third body in various contact conditions may be more or less heterogeneous and continuous. It also revealed that most of the content of the third body come from the degradation of the first-bodies, and that it flows into the contact with an unknown rheology and that it has thicknesses varying from a few nanometres to several micrometres [35]. It is believed that wear is loss of mass from the whole contact and not just from one material. Hence, fatigue, abrasion, adhesion phenomena etc. are not wear but particle detachment mechanisms [36].

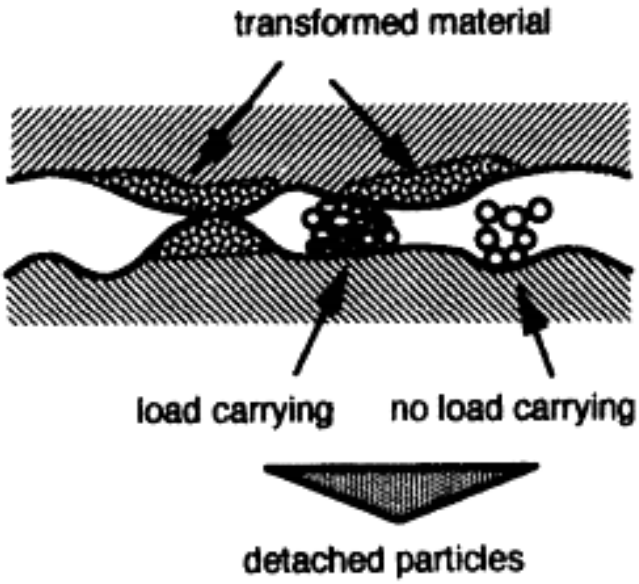



**Figure 2-6: A schematic showing wear and degradation of material in the third-body layer [37]**

Godet and Berthier in the 1980s [38], [39] were the first to propose a concept that was unconventional, discussing the importance of solving tribological problems by considering its components; the bodies in contact (the first bodies consisting of counter body and base body), the mechanism that contains them, and the interface that separates them (the third body) all simultaneously, and they called it the tribological triplet concept [5]. The simultaneous interactions between the three elements are what influence the tribometric characteristics of the system. Here in this framework, three different scales of a frictional contact are investigated which form the tribological triplet which is a multi-scale description of the tribo-system [40] as explained in the following sections.

## 2.4 Modelling of Multi-Bodies: The Multi-scale Approach

In the work done by Peter Blaue in1991 [41], he was interested in answering a fundamental question regarding friction, which was “what are the scale effects in steady-state friction?” and in doing so, three levels of interactions were distinguished as shown in Figure 2-7 below [19]. He mentioned that in modelling friction effectively, the proper level of interaction must be determined, and they could be divided as follows;

1. Level I interaction, is the scale that involves the behaviour of the third body; composed of molecular layers and interfacial films, which are on the molecular/sub-nanometers to tens of nanometers scales. Assuming all the friction is induced within the interface between the first bodies, as it would in an effectively lubricated system, the bounding solids could be neglected.
2. Level II interaction, is the scale involving the asperities contacts and surface layers. If the shear is transmitted to the first bodies (solids) - which is usually the case in dry contacts - they have to be included in the model and their properties have to be taken into consideration.
3. Level III interaction, involves the machine and the surrounding environment. This is of concern when the shear forces are transmitted to the machine and fixtures because the material could not maintain the friction forces. This occurs at the higher micro and macro scales.

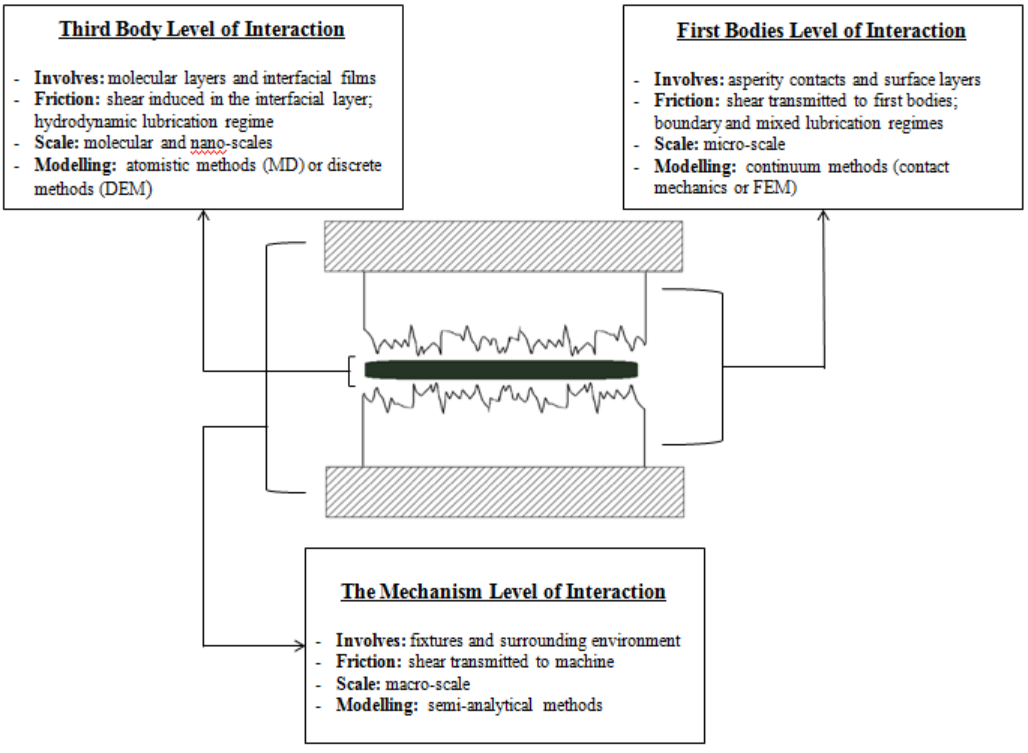



**Figure 2-7: Schematic showing the hierarchy of interaction in tribological systems**

Reflecting back on the tribological triplet concept of the previous section, this means that in studying tribological systems it is crucial to solve the system as a whole; including the three components on their various and different length scales simultaneously. This is where multi-scale modelling comes into play in the study of tribological systems and phenomena. A correct and reliable model should account for these different scales simultaneously as well; a multi-body multi-scale model.

However, it is evident from the literature – which will be shown in section 2.6 - that most studies are conducted on single scales and there is a lack of coupling between the different scales within a single simulation. The level of interactions of the first bodies are on the micro-scale, and hence it was found that they were usually modelled using contact mechanics and continuum mechanics models. As for the third body, since the level of interaction is on the molecular and nano-scale, atomistic models have usually been used to simulate them.

Hugh Spikes in his paper published in 2001 [42] about the expected tribology research advances in the twenty-first century, mentioned the likelihood of the rise of more modelling and simulation studies of multi-body problems due to its importance in understanding many tribological systems and phenomena [5], which did happen. Multi-scale modelling is not a new approach in investigating materials; however, in tribology it is only very recently where tribologists and researchers have tried modelling in a single simulation the different scales of the tribological triplet.

As mentioned in section 2.2, there are two approaches for multi-scale modelling; the hierarchal and concurrent approaches. In many applications where a clear separation between the scales is present in the system, a hierarchal approach is very efficient and possible to use. However, in systems and phenomena where no clear separation between the scales is present, such as in friction, wear and plasticity, a concurrent multi-scale approach is needed which are far more complex where two or more scales are bridged and simultaneously used in a single model [43], [44].

In this case, some use the method of reduction of dimensionality [45] to study the contact and friction mechanics in tribological systems and link between the micro and macro scales, however these are still very abstract concept that are difficult to apply on multi-scale models to predict friction and wear. Other pioneering studies were done to analyse bearings by coupling the mechanism and the bodies in contact by simulating a multi-body model by applying boundary conditions on a FEM simulation coupling it with contact mechanics [46].

Others coupled the smaller scales of the first bodies and the third body to investigate the wear and behaviour of the bodies in contact [47]; to model the interaction levels at the interface, they coupled the discrete and continuum methods in a single

simulation using the movable cellular automata (MCA) method to study dry friction in rail-wheel contact. Others did that as well using a contact homogenization technique [48] where the effects of the deformation of the contact bodies are directly linked to the rheology of the interfaces. Also, a multi-scale study was done using the MCA method to simulate surface topography of dry friction contacts [49] and to study mixed lubrication regime in mechanical seals [50].

Others coupled MD with dislocation dynamics [51], [52] and to model the interface and taking into account the surface conditions, FEM and MD coupled models were done to study the contact problem [53]–[55].While others used the boundary element modelling method where discrete models were coupled to investigate the material degradation in a gear system [56]. However, these methods are also based on approximations and reductions because direct coupling of MD simulations with FEM ones is theoretically impossible due to the $10^{\wedge 8}$ gap in time and length scales.

The mescoscale approaches however, are used to model the scales in between those atomic and macroscales [57]–[61], and is considered by itself a bridge the micro and macro scales. Furthermore, it was shown in literature that friction and wear are also considered mesoscopic phenomena, which is explained in more detail in the following section.

## 2.5 The Role of the Meso-scale: The Gap between the Scales

A study was conducted by Psakhie and Popov in 2012 [58] where they discussed the mesoscopic nature of friction. The mesoscale is known as the gap between the macro and nano scales. The role of the mesoscale on friction and wear was also discussed in [60]. As mentioned in the introduction, Coulomb's law of friction is always used when dealing with friction and wear, however, it is a very rough approximation and it is concluded that all the scales contribute to friction. It is impossible to have a clear separation of macroscopic and microscopic scales in frictional systems [45], [58] because in surfaces there is no gap in the wave vector space as shown in Figure 2-8. Thus, friction is typically a mesoscopic phenomenon which acts similarly to fracture and plastic deformation phenomena, and hence has a mesoscopic multi-scale nature, which was also recognized by Panin in 1998 [57].

Usually, engineers and physicists have very different approaches to study friction, and neither account for its mesoscopic nature. The engineering approach chooses the wave vector separating the macro and micro scales as shown in Figure 2-8(a). This means they only describe the system at the macroscopic scale as a whole by methods on single scales such as the FEM method, and the other scales are not explicitly taken into consideration and they describe the system dynamics using a "friction law" which is highly system independent. On the other hand, many study friction using microscopic models, where the most important scale is chosen for the system and they calculate the dynamics explicitly in this scale as shown in Figure 2-8(b). Although these allow a qualitative understanding of friction at a specific scale, it does not have a quantitative predictive power. Finally, molecular physicists study frictional forces on the molecular scale totally ignoring the intermediate scales as shown in Figure 2-8(c) [45], [58].

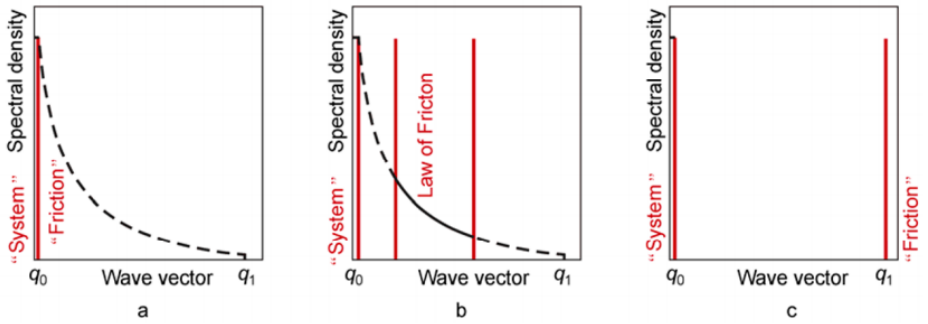



**Figure 2-8: Paradigm in the physics of friction, the world of the (a) engineer (b) friction physicist and (c) molecular physicist [45], [58]**

This separation was clear to notice when reviewing the literature as shown in the following section. The tribology community is either focused on the engineering macroscopic continuum approach, or the physical microscopic atomistic approach.

## 2.6 Modelling and Simulation in Tribology

This section presents a literature review on the numerical and computational simulation studies conducted in the field of tribology. This will follow the guidelines of the tribological triplet concept, thus it will be divided into methods that modelled the first bodies, and the ones that modelled the third body, taking into account the type of computational modelling method used; atomistic, continuum, discrete or multi-scale. At the scale of the mechanism, semi-analytical models are used [5], but it is of no interest in our study since we are only interested in simulating the interaction between the first bodies and the third body.

### 2.6.1 Modelling of Surfaces and Contacts: The First Bodies

The science of tribology; friction, wear and lubrication is all about understanding how two contacting surfaces behave. Whenever two surfaces are in contact, deformation will occur. The deformation may be elastic or partially involve plastic deformation and thus a permanent change in shape. These deformations of element surfaces may be observed both on the macro (rolling parts of a bearing) and the micro (roughness, asperities) level.

No surface is ideally smooth on the micro level, so actual contact will only occur on a certain number of asperities as shown in Figure 2-9, which are deformed by loading. The sum of the micro-contact surfaces is known as the real contact surface and is relatively small in comparison to the nominal or geometric surface, usually only amounting to a couple of percentage points.

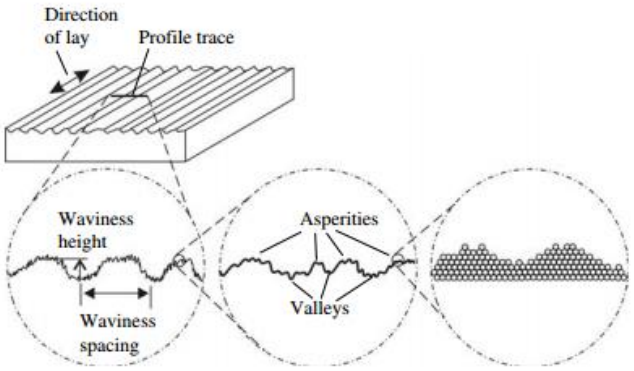



**Figure 2-9: A surface consisting of asperities and valleys magnified to a small scale [15]**

There are two main approaches to model contacts of rough bodies. The first is to represent the rough surface of the body as a set of simple figures, such as spheres or columns, with heights that are normally distributed [62], solved using the analytical Hertzian solution of contact problems. These are reliable in the cases when investigating contact stiffness but cannot be used in the cases where there are arbitrary shaped bodies involved or when the evaluation of stress-strain state of real

asperities is needed. The second approach considers the surface to be made up of a number of finite elements; i.e. continuum modelling, and the shape of these elements are determined by profilometric data taken from experimental data and then are solved numerically.

#### 2.6.1.1 Contact Mechanics

Reliable contact models can improve the understanding of friction and wear, but one of the main difficulties in modelling real rough surfaces is accounting for the effects of surface topography. For decades studies have been conducted to try using simple models in the field of contact mechanics to describe the behaviour of the relative motion between two bodies.

Hertz, after many decades of attempts was the first to propose a contact model [63]. Initially the model comprised the contact between a flat surface and a cylinder but was later extended to include contact between two spheres of varied radii [64]. The model used the elasticity of the materials in contact in order to represent a simplified local body deformation, as well as providing the stress field in the area of contact using the normal force, the curvature radius of the two contacting bodies and the elastic moduli. The model assumes that the contact is frictionless, the contact area is small relative to the size of the bodies, the stresses are localized at the contact zone and less than the elastic limit, and finally a continuous and smooth distribution of pressure in the contact area. Although many later models used the original hypotheses, extensions were proposed to include other geometries [65], [66] and to account for friction and inelastic behaviour [67].

More theories based on the Hertz smooth contact theory were presented taking elastic deformation in single-asperities contact into account by Johnson, Kendall and Roberts (JKR) [68] and by Derjaguin, Muller and Toporov (DMT) [69]. However, in the case of normal contact of inelastic solids, Johnson's core model of elastic-plastic indentation is used [67].

Furthermore, in the 1960s, rough surfaces have been studied using a static contact model which was developed by Greenwood and Williamson [64] to describe multi-asperity contact as shown in Figure 2-10. Statistical distribution of asperities was taken into account and deformations are based on Hertzian contact theory. This resulted that for low normal loads the asperities provided a larger effective radius and a lower contact pressure than predicted by Hertz. In 1975 Bush et al. upgraded the Greenwood and Williamson model and statistical distribution of asperities' heights is still taken into account, but they proposed an elliptical, instead of round, shape of asperities. In 2006 Greenwood simplified Bush et al. model and obtained similar results.

The GW model was modified to include elastic-plastic deformation by many researchers, however the first model to introduce a statistical contact model which realizes both elastic and fully plastic behaviour of asperities was the Chang-Etsion-Bogy (CEB) model [70]. The Zhao-Maietta Chang (ZMC) model [71] developed the CEB model further to include the elastic-plastic deformation regime. They derived their expressions by using Hertz theory for the elastic deformation behaviour, and the Abbot-Firestone [72] and Tabor [73] models for fully-plastic deformation. Persson's theory [74] was also introduced where the stress probability distribution is a function of the surface resolution. Both GW and Persson's models are used in the tribology community to model rough contact.

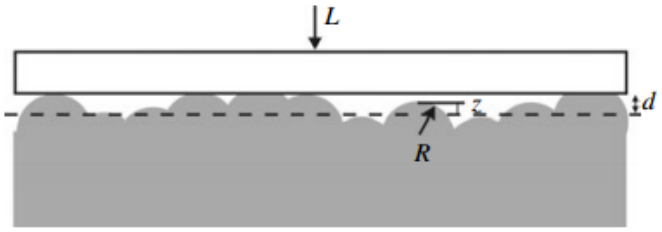



**Figure 2-10: Contact geometry assumed in the Greenwood and Williamson model [15]**

However, all mentioned models for real contact surface are statistical models. Based on this we cannot get an exact picture of real contact surfaces, but they can give us a good approximation of them. There also exist fractal (Pawlus, McCool, Buchner, Sodergerg, etc.) and deterministic models (Whitehouse, Blunt, Tomanik, etc.) which are more advanced and complex [75]. None of these models can exactly describe real contact surface properties, that is why this field still represents one of the most mysterious fields of tribology. This was also mentioned in a recent review paper of the modelling and simulation approaches in tribology across scales [43].

#### 2.6.1.2 Continuum Mechanics: FEM & BEM

FEM is the main continuum mechanics method used in modern simulations as mentioned previously. Conventionally, tribology has been approached using analytical and semi-analytical methods, however during the late 1980's journals were published regarding numerical methods to approach tribological problems [76], specifically using FEM [77]. Although at first it was regarded as computationally expensive compared to the analytical methods, an increase of interest occurred, because though it is complex with many inputs, many complex problems may be solved numerically and almost instantly, as opposed to taking hours using conventional analytical techniques. For this reason, the complexity of the computational methods of analysis may be tolerated, in addition to the fact that optimization methods are constantly being developed to reduce its complexity. The other distinguished method in continuum mechanics is the Boundary Element

Method (BEM) [78]. BEM is less versatile but more efficient than FEM because it does not require volume discretization and only the surface degrees of freedom need to be solved.

In tribology, FEM is used to study the behaviour of the first bodies in contact because contact problems are generally nonlinear even if the contact is linearly elastic with frictionless and non-adhesive properties because the contact area is never known *a priori*; except for the simple rigid flat cases of full contact. FEM solves this problem because an explicit relationship between stresses and strains can be defined which enables considering the arbitrary constitutive material models; whether they are simple linear elastic or complex crystal plasticity models [43]. For example, FEM was applied in the modelling of brake systems and the method was able to predict the brake dynamic instability through generating and analysing the squeal vibrations and local contact stresses, they were reviewed in [79].

On the other hand, BEM uses a formulation that assumes that locally the material solid is a flat half-space, which makes it more efficient but limits its field of application. Although BEM has been used in the study of rough surface contact mechanics [80], true contact area evolution, interface permeability, electric and thermal contact in the linear material laws, a broad area of systems falls outside its field of limitation where FEM then must be used. Large deformations, large sliding or contacts which involve large nonlinear behaviour, fracture at the interface, or indentation involving strong plastic deformations, are all difficult to represent by the BEM framework. However, material nonlinear behaviour can be included in the BEM framework; such as elastic-plastic [81], [82] and viscoelastic [83], [84] behaviour, as long as the deformations remain small. Severe plasticity behaviour however is related to wear which must be included in the model. This is discussed in more detail in section 2.3.3.

Although continuum mechanics models are extremely helpful in understanding the mechanism of tribological systems and the reasons behind their behaviour, they are unable to act as a predictive tool. This is because they require finding constitutive laws which relate the stress and strain fields where predicting the transitions between elastic, plastic and viscous behaviour is particularly difficult and require explicit experimental data. They are also unable to provide information at the micro scales and at the third body level of interaction scales, they are constrained in size and need very computational expensive demands for any transient analysis needed [43]. Thus, FEM and BEM are useful for understanding the behaviour of the tribological interactions but not for predicting them.

#### 2.6.1.3 Crystal Plasticity

Crystal plasticity is a method that has been well-established for the modelling of material heterogeneous plastic deformation when metals experience large deformations, by assuming that plastic deformation results from the plastic slip of specific crystallographic slip systems within a single crystal or individual grains in polycrystalline aggregates [85] [86]. It was first formulated in the 1920-1930s by Taylor [87], [88] and then it was later developed to include elastic-plastic behaviour [86], [89] and finite strain formulations [90] based on modern continuum mechanics [91]. However, crystal plasticity is a continuum theory and is not applicable on small scales describing the nucleation and propagation of dislocations; molecular dynamics and discrete dislocation dynamics are more appropriate for this which is explained more in section 2.6.2. It can also not capture important phenomena on the larger scales related to plastic deformation such as the formation of dislocation structures or grain refinements, even though attempts in that direction have been made. Since plastic deformation is an inhomogeneous multi-scale phenomena, many of the related phenomena cannot be described using crystal plasticity [43].

In tribology, when dealing with rough surface contacts, crystal plasticity is the relevant constitutive framework to use. It is important to note, that it is only relevant here if the size of contact points is comparable to the grain size in a polycrystalline material. The effects of plastic anisotropy, crystal orientation, pile-up, sink-in patterns, etc, all influence the real contact area evolution in rough contacts. Surprisingly however, only very few studies using crystal plasticity in tribology have been found. The studies are limited to the analysis of asperity flattening and sliding contact [92]–[95], and indentation hardness [96], [97].

### 2.6.2 Modelling of Interfacial Film: The Third Body

The concept of the third body was introduced in the 1980's by Godet [38] to describe the discontinuous and heterogeneous interface at the contact between two bodies. This thin layer, which ranges from several nanometres to several micrometres in thickness with a characteristic thickness of 1-10 µm [40], has its own rheology which depends on the material properties, contact conditions and sometimes other unknown parameters. Godet aimed to unify the problems of friction and wear in dry contacts with the theory of lubrication.

The third body can generally be defined as the zone of material which its composition is different from the bulk material of the first bodies in contact. This zone could be fully present, such as in a fully lubricated system, or empty such as in dry contacts [19]. These third bodies, whether solid or liquid, are fed tangentially and/or normally to the contacts [38]; tangentially by the relative motion of the first bodies, and normally by the wear occurring in the first bodies. There are three main functions related to the third body; it accommodates speed between the first bodies, it supports the normal load applied to the first bodies, and it separated the first bodies to limit the degradation due to direct contact between the materials [5], [98], [99]. On the macroscopic scale, where two surfaces are sliding separates by a lubricant film, the frictional behaviour is characterized by three different lubricating regimes described by the Stribeck curve [100] as shown in Figure 2-11.

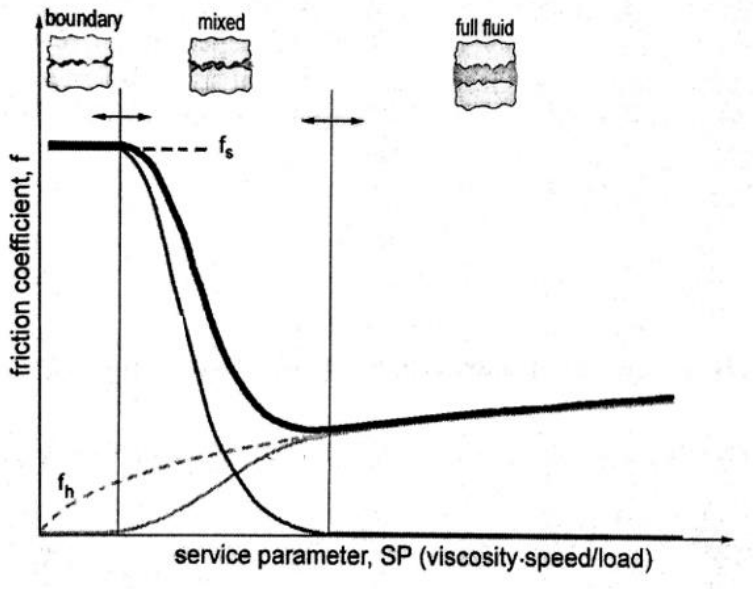



**Figure 2-11: The Stribeck curve and lubrication regimes (a) boundary lubrication (b) elastohydrodynamic and mixed lubrication (c) hydrodynamic lubrication [101]**

$f_s$ is the friction coefficient of the contact between asperities, and $f_h$ is the friction coefficient of a full lubricated contact. The hydrodynamic lubrication regime is determined by an increase in friction and film thickness with an increase in sliding

velocity, a decrease in load and an increase in viscosity [19]. If the system is fully lubricated, the elastic deformation effects start to occur, which happens in the elastohydrodynamic (EHL) and mixed lubrication regimes. In the mixed lubrication boundary, the load is shared between the contact asperities and the lubricant film. The behaviour of these regimes is governed by contact mechanics and continuum mechanics.

In the boundary lubrication regime, the first bodies are separated by a very thin film, whereby increasing sliding velocity, viscosity or decreasing load, the film thickness decreases as well, which is the case of solid/solid interactions. In the solid/solid contact cases, the stress interactions are accompanied by materials interactions too. Here the smaller atomic and micro scales govern the behaviour. Many mechanical components work in the mixed and boundary lubrication regimes and the film thickness continues to reduce to micro and nano-meters [102].

However, even without a lubricant or interfacial fluid, Godet and Berthier [36], [38], [37] showed that in many applications a medium at the interface still exists and it consists of detached particles or pollutant elements from outside the contact as mentioned earlier and shown in Figure 2-6. To form a fundamental understanding of wear, these systems have to be understood and be able to be predicted before moving on to the more complex lubricated systems; which is the focus of this thesis.

Continuum methods, specifically FEM, has been used for the simulation of the third body and wear [103]–[105]. Here two solids contact each other only at one point of the contact surface. Then, to calculate the mass that should be removed from the materials Archard-type law is used, and accordingly the surfaces are modified at next time step. This may have been efficient in some applications, however in general they are unsatisfactory due the assumptions of Archard's law. It assumes that the detached particles are immediately ejected out of the contact and that they have no part in any degradation process. However, proof exists that the third body particles plays a major role in both friction and wear problems, and the solid third body is usually discontinuous, heterogeneous and anisotropic.

Hence, continuum mechanics are no longer valid in this case and discrete particle-based methods are much more promising even though there is a need for further efforts to define micro-to-macro correlations or bridge the gap between micro and macro scales [43]. These methods include individual particle dynamics and interactions, and are capable of describing microscopic physical, chemical and thermal interactions at the frictional contact, especially for solid lubrication or dry contacts, including plastic deformation and wear [99].

#### 2.6.2.1 Molecular Dynamics (MD)

As previously mentioned, MD are an atomistic physical particle-based model and is considered the basis for all the developed discrete or particle-based models. In literature a large variety of studies can be found using MD on the atomic and molecular scales but also solving larger nano-scale phenomena. MD is used in studying atomic scale friction, which is extensively reviewed in [106], [107], and in the molecular aspects of boundary lubrication [108], mixed lubrication [109], dry contacts [110], [111] and adhesive wear as shown in Figure 2-12 [112] capturing the fracture-induced debris formation. It is also used to model tribochemical interactions [55], which started in 1994 [113], investigating the formation and breaking of bonds.

Furthermore, Tomlinson [114] and Frenkel- Kontorvora [115] models were used to investigate atomic friction where two crystalline surfaces were in contact. Although they are capable of representing a crystalline layer of absorbed atoms, it involves many assumptions for simplicity. Also, atomic scale asperities were investigated by MD models that are AFM-like [116], [117] using large parallel computing. Nano-scale simulations were also able to model sliding of intermediate motion with stick-slip systems [118], [119]. Other studies of indentation also used MD in the 1990s [120], [121] where new insight was given into the phenomena that happen when two metallic surfaces come into contact, and then the normal force is initially almost not present, but then it suddenly becomes highly attractive.

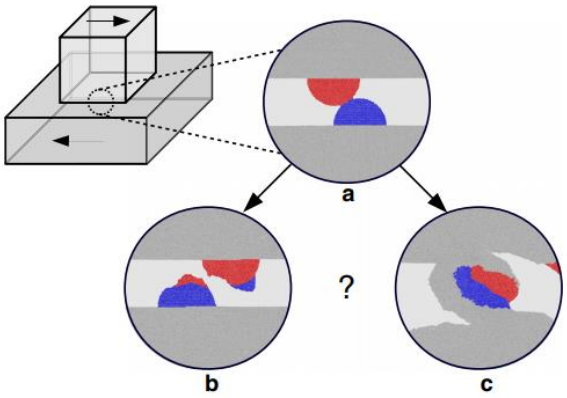



**Figure 2-12: Schematic representation of two possible asperity-level adhesive wear mechanisms (a) the wear process occurs via either (b) a gradual smoothing mechanism by plastic deformation or (c) a fracture-induced debris formation mechanism [112]**

Although MD is a very successful method in studying third body interactions, its limitations are that they can only be used on limited length and time scales, they cannot model larger micro scale phenomena, are difficult and sometimes impossible to validate experimentally, and are computationally expensive. Also, they are very sensitive to the potential functions used to define the interactions between the atoms; which are based on complex quantum mechanics models.

#### 2.6.2.2 Discrete Element Method (DEM)

DEM is a numerical solution to describe the mechanical behaviour of discontinuous bodies and was developed as a tool to simulate mechanical behaviour of granular matters and particulate systems, e.g. some kinds of soils, grains, fragmented rocks, etc., by Cundall and Strack in 1979 [122]. Unlike continuum mechanics methods, DEM is particularly attractive for modelling media that has a great number of interfaces such as geomaterials and granular materials due to its ability to construct a medium that involves discontinuities. The elements interact with each other at contact points, making it possible to model voids, imperfections and heterogeneities. DEM was developed as an extension of MD to model macroscopic slightly deformable solid grains. The primary difference is that in DEM the analysed discrete particles have a shape, size and rotational degrees of freedom, in contrast to MD where mass points are usually considered in the calculations. Hence, other interactions between particles can be analysed, such as rotation, contact plasticity, friction, etc.

For these reasons, DEM has been widely used to study tribological problems such as wear where the material has a continuous part (the first bodies), a continuous part with cracks and a discontinuous part which is considered to be a mixture of abrasive and wear particles [123]. As explained, these are also called first and third bodies, respectively. For example, it was first used to model third-body flow in the 1990's by simulating bearing powder lubrication [124] after ten years from simulating fluid lubrication for the first time using DEM [125], where rigid spheres moving between two rough inclined planes were used to represent the third body. The first few studies after that studied the effect of different numerical [126], geometrical and shape related parameters [127] on the mechanical response of the medium, but were just exploratory studies. Later on, better representative laws were used involving phenomena within the interface using the JKR contact model [128], [129].

More recent, Fillot et al [98], [99] studied the flow of third body particles inside a dry contact by modelling the degradation of the material, i.e. formation and movement of fragmented particles. They used DEM to simulate the behaviour of wear with adhesive particles and obtained some interesting results. However, as they stated their goal was to understand and not to predict, moreover, the work was not experimentally validated. Following that, extensive studies on the wear mechanisms with simple shear gave rise to wear laws in the discrete approach [98], [130]. Some work also accounted for the particle deformation in simple shear [131], [132], however, these models do not represent dense granular flows which are found in experimental observations, and also of course in non-granular materials.

This is why, although DEM is well adapted to simulate discontinuities, it is not suitable to simulate continuous behaviour of materials [123]. DEM has been proven to be very efficient in solving processes involving granular materials such as sand, grains, powders, particles, etc. However, these are all discontinuous materials. When solving problems involving continuous materials such as metals, plastics, etc, DEM has often failed because it fails to model a cohesive structure. This is due to the contact configurations and laws used in DEM to describe individual interactions which are constitutive laws obtained experimentally or using hierarchical multiscale approaches [43]. Furthermore, unlike FEM, continuous mechanical behaviour laws cannot be directly introduced into the DEM formulation [123]. This will all be explained in more detail in Chapter 3.

#### 2.6.2.3 Movable Cellular Automata (MCA)

The MCA method is another particle-based method used in simulations in tribology. MCA was first introduced by Psakhie et al, in 1995 [133], as a simulation tool within the framework of mesomechanics. It is a hybrid particle-based method based on the classical cellular automata (CA), discrete element (DEM) and molecular dynamics (MD) methods; combining their advantages. This method allows the modelling of complex materials behaviour and processes on the mesoscopic length scale, such as crack generation and growth, mass mixing, friction and wear in real systems, phase transformation, etc. Many developments in MCA have been made since 1995, and the latest description of the method can be found in [3]; where MCA is presented as a discrete approach to model the behaviour of materials on different scales and is used as a multi-scale modelling approach.

Cellular Automata (CA) is a mathematical concept that was first established in 1940 by von Neumann [134] and was later more developed by him and his colleagues in the 1960s [135]. In CA, initially each cell is either On or Off, and the evolution of the systems depends on the general rule. For example, rule 30 as shown in Figure 2-13, states that if an Off cell has two neighbouring Off cells, then they will produce an On cell, and if an Off cell has an Off cell on its left and an On cell on its right, then they will produce an On cell as well, and so on [136]. This will lead to the development of a complex system, which means the evolution of a complex system depends on the initial state of a cell and the state of its neighbours. Afterwards in the 1980s, Stephen Wolfram [137], [138] noticed this complexity of CA patterns and introduced the idea that complex physical systems can also be represented by CA, where space, time and physical quantities are discrete. The media is divided into an ensemble of elements (automata); where their locations and neighbours are fixed, and their state could be passive or excited. An external effect will trigger the

evolution of the system, such as an external load or temperature change, and this evolution is governed by a general transition rule [137]. However, this method does not take into consideration the possible and different interactions between the automata, thus is only applicable in the description of continuum media.

Later on, MCA introduced the concept of state of pair automata (interacting pairs of automata) to the classical CA method. In MCA, a material of interest is divided into a set of elements (automata) of finite size representing grains or particles of the material, where the particles interact with each other based on interaction laws. This way, MCA was extended from CA by introducing the principles of particle-based modelling; atomistic and discrete modelling, applying the equations of motions on all the automata, as well as describing the interaction laws between their neighbours. The mobility of the automata made it possible to study material behaviour, fracture, mass mixing, crack propagation, phase transformation and damage generation.

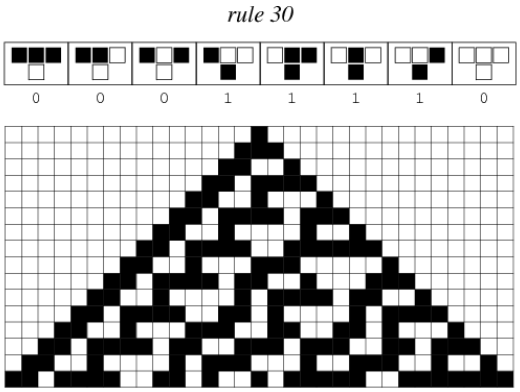



**Figure 2-13: Rule 30 from Wolfram cellular automata [136]**

MCA has been used in many studies to explore the behaviour of different complex and heterogeneous material systems under loading. Initially MCA was mainly used in the study of fracture mechanisms for different materials such as composites (contrasting media) [139] , ceramics (with and without pores) [140]–[143], coatings [144]–[146], alloys [147], etc. It also allows for the simulation of friction forces as a function of material, loading parameters, surface topography and wear. Most studies are done by Popov et al. studying wear in combustion engines [148], friction coefficient in rail/wheel contact [149], quasi-fluid nano-layers [150] and surface topography [151]–[153]. More studies were done by Österle et al. looking at friction of the automotive braking system and formation of tribofilms [154], [155]. Furthermore, studies were done using the MCA method to investigate the behaviour of the third body [156]. The simulations allow the study of the formation of the third body but are still far too simple to show all effects that might occur in a lubricated tribo-contact [157].

The advantage of MCA over other particle-based methods is that it describes material on the mesoscale and one of the difficulties that face discrete simulations of the third body is accounting for the mesoscale which is formed of the sub-100nm scales. As explained earlier, most fundamental tribological phenomena such as friction and wear, occur at the mesoscale [58], [60] and are intensively discontinuous due to the damage, formation and accumulation processes that are involved. This is solved using the MCA method which is a mesoscale modelling method [158] [159]. The other major advantage is that MCA provides a fracture mechanics modelling approach entirely compatible with continuous classical theory. All MCA parameters can be directly derived from material properties without the need for parametric fitting (like in DEM). This is only possible using MCA and will also be explained in detail in Chapter 3.

## 2.7 Movable Cellular Automata in Tribology: a Literature Review

Since the particle-based mesoscopic MCA method seems to be the best approach for the predictive modelling of tribological systems, it is thoroughly reviewed in this section to establish its applicability to tribological systems and the tribological triplet concept.

One of the main advantages of the MCA method is it being a discrete/particle-based method having a many-body particle interaction forces form. This means that the relevant size of the automata can be chosen to be as small or as large as needed. To model surfaces in contact and friction behaviour using MCA, two approaches are used as shown in Figure 2-14 [160]: 1- to specify an automata size that is very small so that it can directly simulate a real surface, which allows for the direct investigation of material behaviour and processes on all three scale levels (atomic, meso and macro) in an explicit form, or 2- to specify larger automata sizes that would consider the contacting surface as a rough plane, which depends on the micro-parameters of the materials.

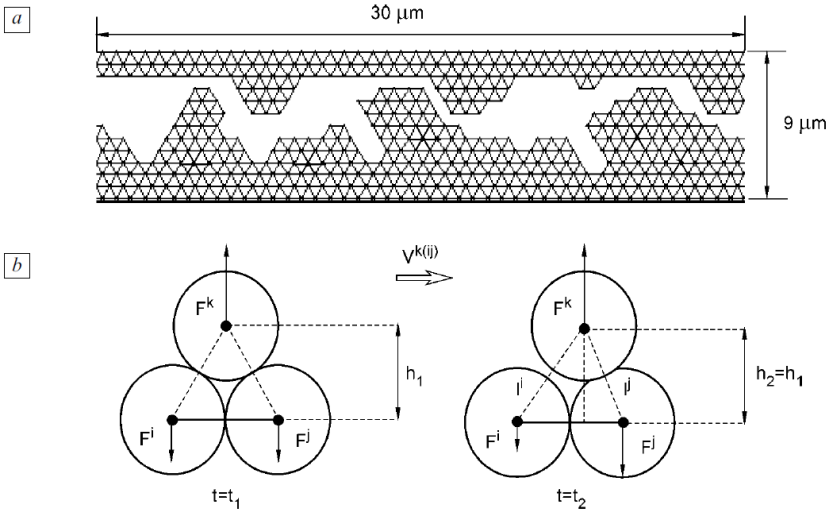



**Figure 2-14: Two approaches to describe contacting surfaces: (a) direct setting of the surface roughness (microlevel) (b) indirect setting based on the segment approximation within the framework of the MCA method (mesolevel) [160]**

The first studies that have been conducted of surfaces in contact with the MCA method, used the second approach because even with using HPCs, the first approach is too computational extensive and only a small area of contact can be simulated due to the high numbers of particles required to simulate a real solid. The first approach, however, can be used to study mechanisms occurring at surface layers, such as crack propagation, damage generation and accumulation, compositions near the surface, changes in surface profiles and others. These studies can help understand the mechanism and reasons behind those phenomena at the mesolevel, because this scale level correspond to these characteristic sizes, but is not sufficient to describe the interaction of surfaces at the macro-level.

The two main applications of the MCA model found in literature within the tribological context, is the simulation of pad-on disk in brake systems and rail-wheel contact. There are many brake pad materials available commercially for the braking system in the automotive industry and they all have one thing in common; which is them having a complex material structure and composition, and they are also very environmentally incompatible [161]. In the following studies [155], [162]–[164], steel on steel contact has been investigated, at a single asperity scale and the tribofilms were investigated too. Experimental observations showed that the friction layers differ in properties from the bulk material structure of the pad and disc. They revealed that the material at the upper most layers at the interface between the pad and disc will change drastically due to tribological processes and will form a thin layer (< 1µm) of the order of 100nm thick, with grain sizes of the order of 10nm. The formation of these films depends on the friction properties of the surfaces and can be thought of as a layer of wear debris due to mechanical mixing of wear particle which are all linked to each other within the friction layer.

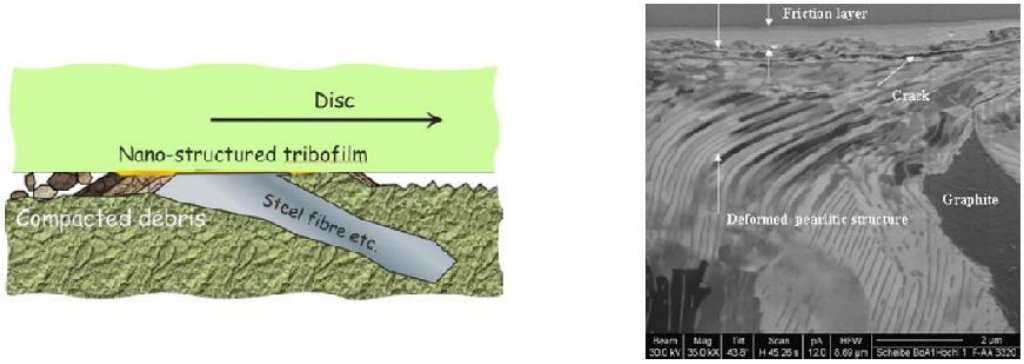



**Figure 2-15: (a) Schematic of pad/disc interface (b) cross-section of friction layer on brake disc [163]**

It was also observed experimentally that when a certain amount of soft inclusions (graphite here) is introduced to the iron oxide layer, the frictional behaviour changed completely due the formation of a mechanically mixed layer (MML) where velocity accommodation takes place. Thus due to the high mass mixing occurring between the components of the pad and disc materials, this layer is also formed (also of thickness < 1µm) with grain size (100nm) which contains sever plastic deformations as shown in Figure 2-15. Thus, this study also looked at the effect of the concentration of these inclusions on the coefficient of friction (COF).

The steel-steel contact was simulated at the single asperity scale which is typically of about a few micrometers of size. The supporting material in the disc is pearlitic steel and that of the pad is ferritic steel, and the friction layer is formed of a matrix of iron oxide with and without graphite particles as shown in Figure 2-16. Graphite was used here to represent any solid lubricant just because the properties of graphite

are well known and were randomly distributed. The mechanical properties of the materials at room temperature are in Table 2-1 and their stress-strain characteristics are used for simulating the materials; they are an approximation of the experimental diagram, so the mechanical properties of the components are the same as the real materials.

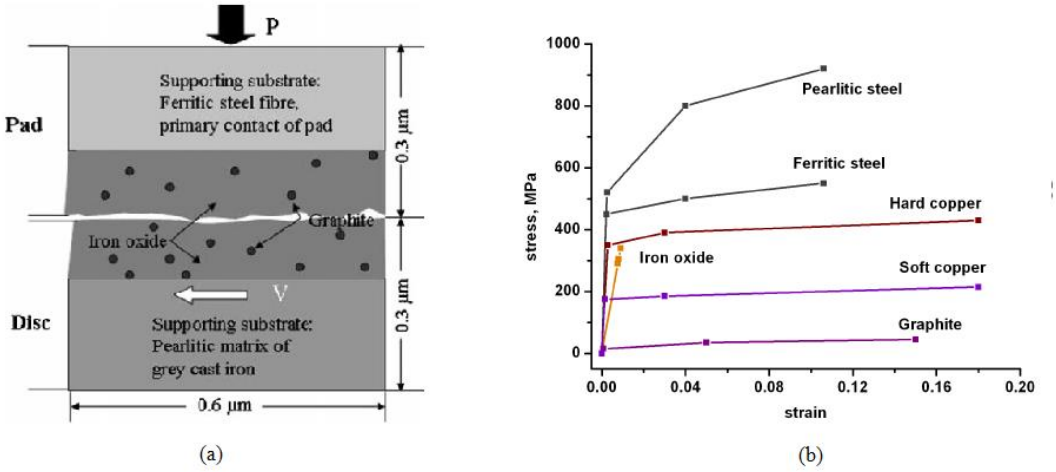



**Figure 2-16: (a) Schematic of pad-disc interface (b) response functions of the materials considered [163]**

**Table 2-1: Materials parameters for modelling [163]**

| | Young's modulus $E$, GPa | Poisson ratio $\nu$ | Elastic limit $\sigma_{y1}$, MPa | Yield strength $\sigma_{y2}$, MPa | Ultimate tensile strength $\sigma_s$, MPa | Strain at yield strength $\epsilon_{y2}$ | Breaking strain $\epsilon_s$ |
|---|---|---|---|---|---|---|---|
| Ferritic steel | 206 | 0·28 | 450 | 500 | 550 | 0·04 | 0·106 |
| Pearlitic steel | 206 | 0·28 | 520 | 800 | 920 | 0·04 | 0·106 |
| Iron oxide | 380 | 0·3 | 290 | 305 | 340 | 0·008 | 0·009 |
| Graphite | 20 | 0·3 | 15 | 35 | 45 | 0·05 | 0·15 |

In the simulated model, the automata size was 10nm, which corresponds to the smallest grain size of the tribofilms as typically observed experimentally during automotive braking. The disc has a constant sliding velocity of 10m/s before braking, and the pad has a constant normal force of 20-100 MPa acting on the top layer of automata of the pad. This load corresponds to the contact pressure (P) which is higher than the nominal pressure because it is assumed that the real contact area is small. The time step of the simulation was $10^{-13}$ s and the automata are assumed to be linked initially. The composition of the tribofilm throughout this study was kept constant with 13% soft (graphite) inclusions. Periodic boundary conditions were applied.

The switching criteria for the transition from linked to unlinked and vice versa were set for the different compositions. In terms of breaking bonds, when two different materials are linked, they become unlinked once the stress intensity reached the strength of the softer material. For metal-metal pairs, the automata can go from unlinked to linked (re-linking) easily, but it is forbidden for the graphite-graphite,

oxide-oxide, graphite-oxide, metal-oxide and metal-graphite because they are considered as having brittle behaviour. Compression can happen for metal particles if plastic deformation occurs under pressure, and for oxides but only if elevated pressures and temperatures are involved. These are prerequisites for the formation of third-body and the mechanically mixed layer (MML), because it assures that any particles released from the third-body film will stay contained within the MML layer [155].

In each time step, the response of each automata is calculated by getting the stresses and strains of each pair-automata assuming plane stress approximations as explained before. The Von Mises stress intensity is then calculated which is used to determine whether the links break or not, and also whether the transition back to the link state occurs. Then new positions and velocities of all the automata are calculated, if enough automata become unlinked, larger movements occur. It cannot be assumed that the response functions of the materials at bulk behaviour is the same for the nanoparticles, this cannot be validated experimentally, however, the simulation results show great agreement of velocity accommodations of pad-disc rotating contacts.

The results of pure oxide layers where simulated against the ones with 13% graphite inclusions as shown in Figure 2-17. After a running-in period, breaking of bonds occur at a narrow zone at the interface due to deformation, fracture and mass mixing, but they do not propagate to the bulk material. This shows proper formation of the friction layers. The simulations show that although oxide layers prevent adhesion and micro-welding, there is no smooth sliding behaviour and no MML is formed and they produce high friction fluctuations as shown in Figure 2-18.

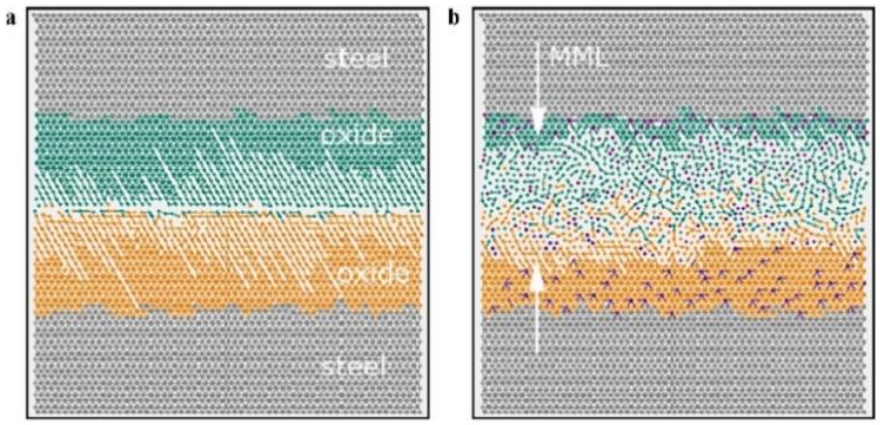



**Figure 2-17: Bonding states of automata after simulation (a) pure oxide layers; (b) oxide layers with 13% graphite nanoinclusions. Automaton size: 10 nm, contact length: 0.5 μm [161]**

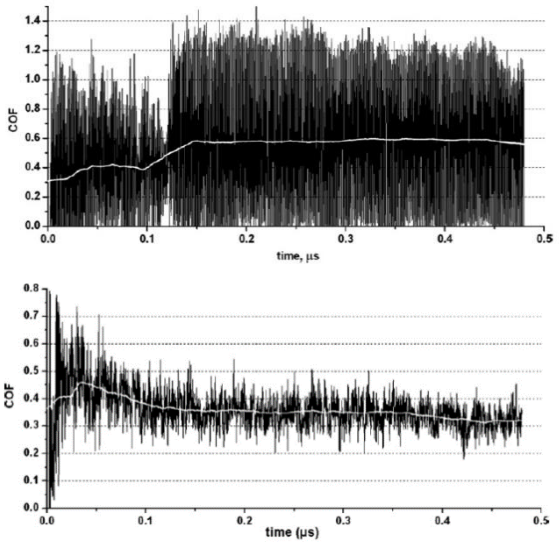



**Figure 2-18: Dynamic COF for (a) pure oxide layers (b) oxide layers with 13% graphite [161]**

They also looked into whether dry contacts will produce difference results. Thus Figure 2-19 shows the results for oxide-oxide contact and Figure 2-20 shows the results for metal-metal contact. seethe figures show that dry friction produces much higher friction because no tribofilms or MML layers are formed.

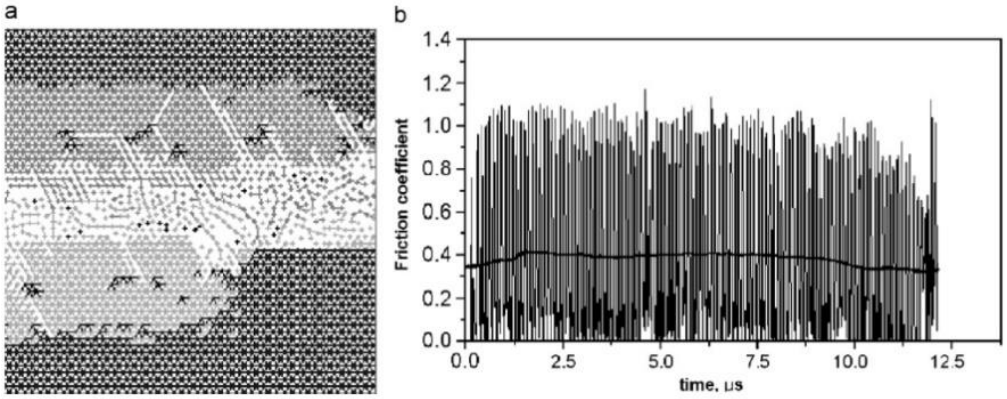



**Figure 2-19: Oxide-Oxide contact (a) simulation model (b) dynamic COF [162]**

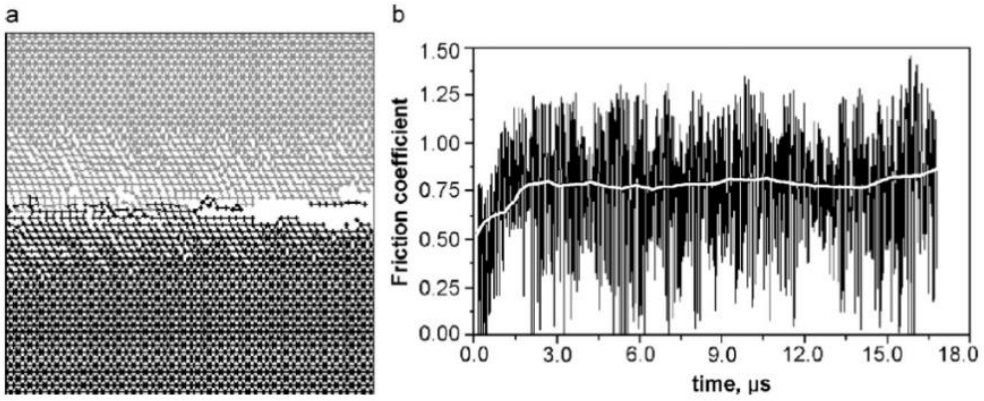



**Figure 2-20: Metal on metal contact (a) simulation (b) dynamic COF [162]**

They also looked into the effect of temperature on the mechanical properties of tribofilm constituents and how they affect modelling results [155]. At elevated temperatures, the iron oxides undergo a brittle–ductile transition and graphite strength slightly increases. The stress-strain behaviours of both the iron oxides and the graphite are shown in
Figure 2-21, where $T_0$=room temp up to $T_4$=800°C.

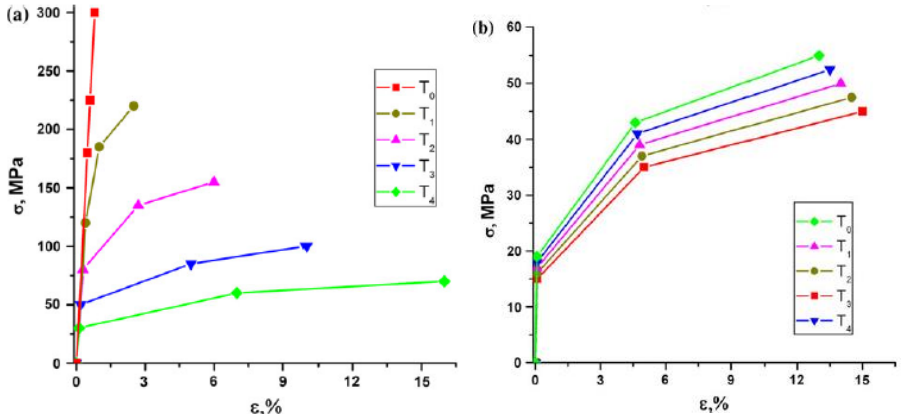



**Figure 2-21: The stress-strain behaviours of (a) iron oxides and (b) graphite at elevated temperatures [155]**

The results are shown in Figure 2-22, and it was observed that the MML layer increases with increasing temperature (pressure) until a certain point and then the tribofilm increases decreasing the MML layer. The corresponding friction characteristics can be seen in Figure 2-23.

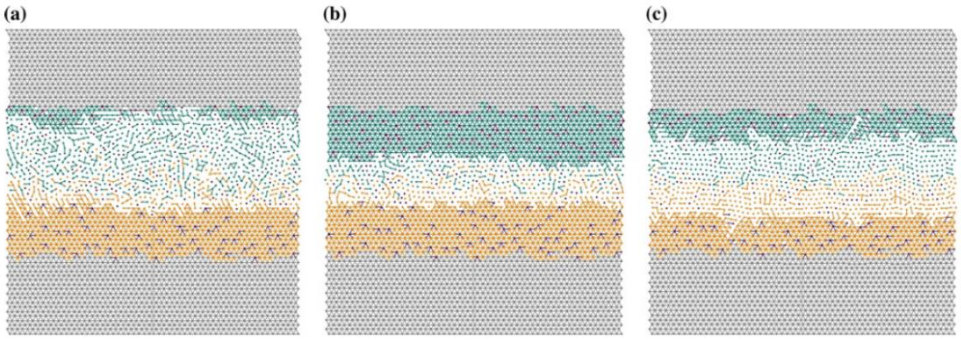



**Figure 2-22: Results at t=0.48 µs at (a) $T_0$ (b) $T_2$ (c) $T_4$ [155]**

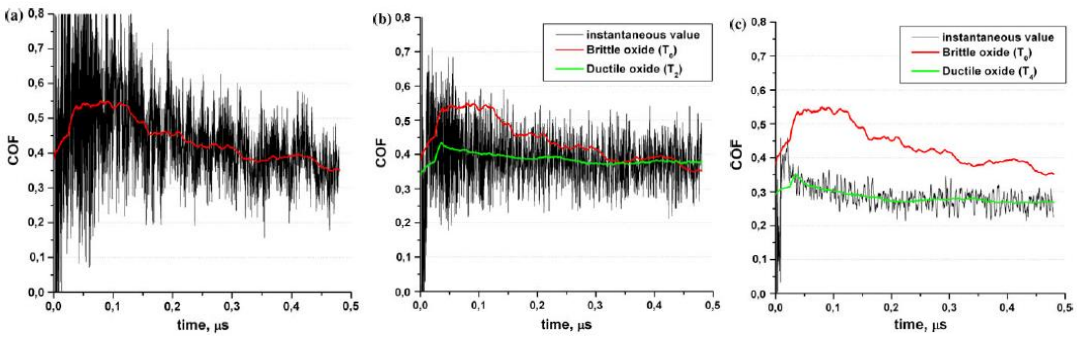


**Figure 2-23: The time dependencies of COF for various assumed oxide properties: (a) $T_0$ (b) $T_2$ and (c) $T_4$. The upper curves (red) always represent mean values determined from instantaneous values of the left curve ($T_0$) for comparison [155]**

Another study shown in Figure 2-24 added copper nanoparticles to the graphite inclusions, which lead to similar friction behaviour as shown in Figure 2-25. But the results show that the inclusions have a great impact on the sliding conditions at room temperatures as well as elevated ones. Table 2-2 shows all the different contact situations that have been studied using MCA.

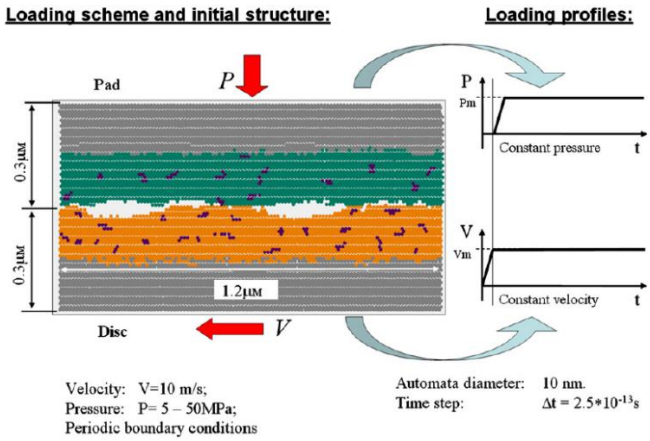



**Figure 2-24: Schematic of MCA model structure and loading parameters showing the boundary conditions [165]**

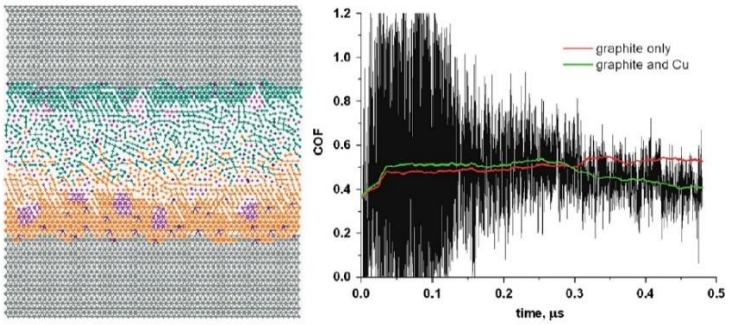



**Figure 2-25: (a) Simulated model for structure shown in Figure 2-24 (b) different between COF for graphite only and graphite and copper inclusions [165]**

**Table 2-2: Results of MCA-modelling with different contact situations [161]**

| **Description of Contact** | **COF-range** | **COF-evolution** |
|---|---|---|
| Steel on steel [a] | >0.75/0.6 | Unstable/partly stable |
| Hard copper on steel (ambient temp.) | >0.5 | Unstable |
| Soft copper on steel (elevated temp.) | >0.5 | Partly stable |
| Oxide on Oxide | >0.5 | Unstable |
| Oxide on oxide with 5% graphite | >0.45 | Unstable |
| Oxide on oxide with >10% graphite | 0.3-0.4 | Stable |
| Oxide on oxide with >10% soft copper (elevated temp.) | 0.3-0.4 | Stable |

| | | |
|---|---|---|
| Oxide on oxide with soft and hard inclusions | 0.45-0.50 | Stable |
| Steel on graphite | 0.2 | Stab |

[a] The matric of cast iron corresponds to pearlitic steel.

## 2.8 Summary, Research Gap and Conclusion

This chapter delivered the first contribution mentioned in section 1.4 of identifying the deficiencies in existing numerical/computational models and identifying the most suitable approach for our aim. As mentioned in the aims and objectives in Chapter 1, what is required for a more fundamental understanding of tribological systems is an improved constitutive model that would capture the most important scale levels and the physics involved between contacts at those scales.

In problems of wear or fracture, material is composed of continuous parts and discontinuous interfaces, and simulating a multi-body system requires simulating the first bodies (which are continuous bulk solids) and the third bodies. By investigating the state of the art in the field of modelling in tribology, it has become clear that a multi-body model is required which can simulate both the first bodies and the third body simultaneously. This has been proven to be difficult because most studies use the continuum approach to model the first bodies the macro-scale, and the third body is modelled using particle-based or discrete approaches on the micro and nano scales. Since the continuum approaches are incapable of describing the inhomogeneities of the microscale which are crucial for the simulation of plastic deformation and wear, only a particle-based model would be able to provide a tool for the simulation of multi-bodies and the prediction of friction and wear.

It has been established from the previous sections that MD is not applicable on simulating materials on larger scales, hence the two most relevant particle-based approaches in our case are DEM and MCA. The important point to note here is that to simulate the bulk continuum behaviour of a material by a particulate system, it depends on the collective interactions of individual particles, and hence a realistic representation of a cohesive solid is key to characterizing bulk material and understanding their behaviour which is not the case for granular DEM models [166]. Furthermore, it is assumed that DEM particles are made of nondeformable and unbreakable granular matter which does not work in the case of third body simulation were the material is relatively soft and brittle [5], [167].

It was also shown that mescoscopic modelling bridges the gap between the atomistic and continuum viewpoints; because a clear separation of scales is principally impossible for triblological phenomena. Friction and wear mechanisms are essentially mesoscopic multi-scale phenomena that act very similar to fracture and

plastic deformation phenomena. The only two mesoscopic scale methods that are mentioned in the literature review are crystal plasticity and MCA. However, the continuum crystal plasticity models for frictional contacts are not applicable on the the (sub)micron scale, because they miss a length scale capable of capturing size-dependence. Hence, the most relevant mesoscopic particle-based method found for the simulation of multi-bodies for the prediction of friction and wear is the MCA method.

MCA gives the possibility of choosing the scale of the simulation [168] which helps investigate elementary processes in tribological systems. Due to its unique capabilities it allows for the simulation of formation and accumulation of damages, fracture processes, formation of pores and cracks and most importantly the simulation of friction and wear. The results obtained from these simulations in literature have been proved to be reliable compared to experimental data. Furthermore, MCA is capable of simulating the first bodies and the third body simultaneously with direct use of continuum descriptions. It is capable of simulating both continuous and discrete behaviour of materials without the limitations of DEM.

The main limitation of using the MCA method is that it may be insufficient to simulate the behaviour of the material at the macro-level because it would be too computational expensive to reduce the automata size that much to describe real solids. However, this can be solved by coupling MCA with continuum models such as FEM models and produce a more multi-scale model that would predict the behaviour at the macro. This was done by Gong et al. in 2013 [169] where they investigated the sliding contact of sealing rings in the macro and micro scales using a multi-scale FEM-MCA coupling method, and were able to visualize and investigate several types of frictional behaviours. They could also be coupled with atomistic models, as described in [59]. The other possibility is providing a large-scale parallel computing code which is computationally powerful enough to cover more scales and reduce the automata size.

As mentioned before, there are two ways to model surfaces in contact and friction behaviour using MCA; as shown in Figure 2-14. Firstly, to specify an automata size that is very small so that it can directly simulate a real surface, which allows for the direct investigation of material behaviour and processes on all three scale levels (atomic, meso and macro) in an explicit form. The second approach is to specify larger automata sizes that would consider the contacting surface as a rough plane, which depends on the micro-parameters of the materials and makes it possible to cover the meso and macro scales.

It was found that, all the studies that have been conducted so far of surfaces in contact with the MCA method, used the second approach because even with using

HPCs, the first approach is too computational extensive and only a small area of contact can be simulated due to the high numbers of particles required to simulate a real solid. The first approach however, can be used to study mechanisms occurring at surface layers, such as crack propagation, damage generation and accumulation, compositions near the surface, changes in surface profiles and others. These studies can help understand the mechanism and reasons behind those phenomena at the mesoscale because this scale level corresponds to these characteristic sizes but is not sufficient to describe the interaction of surfaces at the macro-level.

This problem is solved by implementing MCA in a large parallel computing simulation tool called LIGGGHTS (a DEM open-source code) – which is covered in Chapter 4. By doing so, the first approach of directly simulating a real surface can be achieved. To simulate materials using the MCA method, currently, only a 2D MCA demo-version [170] exists for public use, which was developed by the Laboratory of CAD of Materials at the Institute of Strength Physics and Materials Science in Tomsk, Russia, in 2005. The developers of the MCA method use their own in-house codes. Thus, this project provides a 3D MCA simulator for the scientific and tribology communities to use because the code is open-source.

It is also important to mention here that the challenges and difficulties present in the modelling and simulation of tribological systems and phenomena were also recently discussed at the Lorentz Center workshop held in Leiden at the beginning of 2017 [171], which included many experts in the field from different backgrounds and countries across Europe. This workshop produced a review paper [13] and their main conclusion was that the modelling community in tribology have properly addressed the elastic problems at various scales however significant efforts are still needed for the understanding of plasticity, adhesion, wear, lubrication and surface chemistry in tribological models. They also believe that there is still a need to address and include processes such as crack nucleation and propagation, chemical reactions and fluid-solid interactions which would help with the understanding of rough contact under shear but are not taken into account in contact mechanics studies.These processes should be modelled and understood on their own before including them into multi-scale or multi-physical models. They concluded by saying that only by pursuing these two research aspects simultaneously will there be a chance of a fundamental understanding of frictional interfaces and have simple but comprehensive models which would benefit industrial processes by optimizing and controlling these behaviours [13]. In the end they suggested a collaboration platform for all tribologists where the different research groups would provide open-source software for the community to use, contact problems results and the inputs used for

the simulations/experiments and the assumptions made, and also a list of simulation and testing facilities with links to their websites and laboratories.

The work in this thesis contributes to their outcomes.

# Chapter 3
# Particle-Based Numerical & Computational Methodology: DEM vs MCA

## 3.1 Introduction

In the previous chapter the concept of physical particle-based models was introduced, as well as their importance in the development of a prediction tool for friction and wear. We also briefly discussed the differences between the MD, DEM and MCA methods, and the reasons behind choosing MCA in our study. Now the objective is to develop a 3D MCA model and simulation tool, which requires developing code.

This chapter presents the theoretical, numerical and computational foundations of the 3D MCA elastic-plastic model as described by [1], [3], [172]. It was decided – for reasons that will be explained in this chapter and chapter 4 – to implement the MCA model in an open-source code and simulation tool called LIGGGHTS. LIGGGHTS is an open-source code widely used as a DEM simulator and provides a strong parallel computation platform required for large scale simulation of particles. Due to the similarities between DEM and MCA, LIGGGHTS is a very good platform for developing the 3D MCA model. This is done by adding key functionalities of MCA to LIGGGHTS, by adding new C++ code in the open-source code to develop a 3D computational code/tool capable of direct numerical simulation of friction and wear.

Thus, this chapter will present a detailed theoretical description of the principles of particle-based modelling and simulation within the DEM (specifically the soft particle approach) and MCA frameworks; highlighting their differences in terms of theory and practice. This is crucial to identify which formulations of LIGGGHTS code will need to be modified or added in order to fully implement the MCA model within LIGGGHTS, and the implementation is covered in Chapter 4.

Furthermore, to be able to understand the difference between the models and also, to implement the code, verify and validate it against experimental results in the following chapters, a few key material background information needs to be introduced first in the following section.

## 3.2 Material Mechanical Properties and Characterisation

The mechanical properties of materials govern the material response to various loading conditions. There are several properties that can be measured for a particular material each suitable for particular combinations of stresses. For example, Young's or elastic modulus ($E$) governs the relationship between elastic strain and stress in normal loading condition, whereas, the shear modulus governs the same relationship in a shear loading state. Therefore, it is important to understand the system, loading conditions, geometry and involved material parameters in order to be able to predict the deformation behaviour of the material.

Standardized testing of materials is used for determining material properties and the differences between them. Most commonly, the tensile test shown in Figure 3-1 is used to determine the following properties [173]:

- Young's modulus ($E$): is derived from Hooke's law for elastic springs. It is the slope of the elastic region of the tensile test and governs the ratio between stress and strain in the elastic region. The elastic deformation of a material is recoverable or reversible, meaning when the load is removed the material will recover to its original form. The area under the elastic region determines the amount of elastic energy that can be stored in the material.
- Yield Stress ($\sigma_y$): The yield stress of the material, also known as the elastic limit, determines the point beyond which deformation becomes plastic, i.e. permanent deformation of the material. Even a plastically deformed material when unloaded would only recover the elastic portion, which is usually negligible depending on the material type.
- The maximum tensile stress ($\sigma_u$): The ultimate stress that a material can carry before failing.
- Poisson's ratio ($\nu$): In 3D, when an elastic material is stretched in one direction, it tends to get thinner in the other two directions. Poisson's ratio is a volume conservation ratio that determines the ratio between lateral and vertical deformation for material under uniform unidimensional stress. For metals, the value is typically around 0.3.
- Toughness: measured as the area under the curve of the tensile test and typically indicates how much energy can be stored in the material before failure.

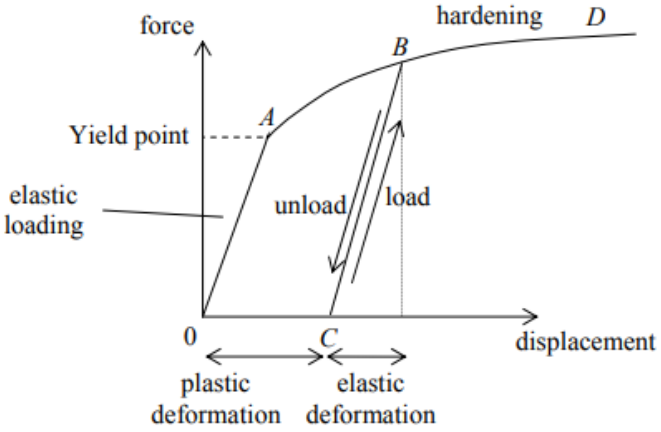



**Figure 3-1: A typical metal tensile test showing the elastic and plastic regions and the path taken during unloading [173]**

Plasticity in crystalline materials occurs due to dislocation movements. Dislocations are crystalline defects that occur during forming of the material and can be described as voids in the atomic arrangements inside the crystal. When crystalline materials are being loaded beyond the elastic limit, dislocations in the crystal lattice begin to slide allowing the grains to change their shape, this results in plastic or permanent deformation of the material. When dislocations begin to pile-up, near grain boundaries or when they encounter a slip system at a different orientation, the load required to further deform the material increases. This phenomenon is known as strain or work hardening of the material [173]. Ceramics, polymers and other classes of materials experience a similar behaviour; however, the strengthening mechanisms differ based on how the material is structured in the atomic scale. In the tensile test, this behaviour is displayed as the increase in force required to deform the material beyond the elastic limit. The relationship between stress and strain in the elastic region is typically not linear [174].

The theory of plasticity makes a few assumptions that attempt to simplify the plastic behaviour of materials. These are:

- The strain-rate does not affect the response of the material
- The response is independent of the Bauschinger effect
- The plastic response of the material is assumed incompressible
- Hydrostatic pressure is ignored and yield stress is independent of it.
- Material behavior is independent of direction; i.e. material is isotropic.

By using these assumptions, the material behaviour can be simplified into one or combination of the 4 behaviours shown in Figure 3-2. These behaviours are described below [173]:

a) Linear Elastic-plastic: This behaviour is common in many engineering metals that are deforming at room temperature.
b) Elastic/perfectly- plastic: This behaviour is more common for elevated temperatures such as during hot forming processes.
c) Rigid/Linear hardening: This is typically common for materials that do not have an elasticity potential; such as ceramics.
d) Rigid Perfectly-Plastic: Is a behaviour common for materials that have no elasticity or strain hardening potential.

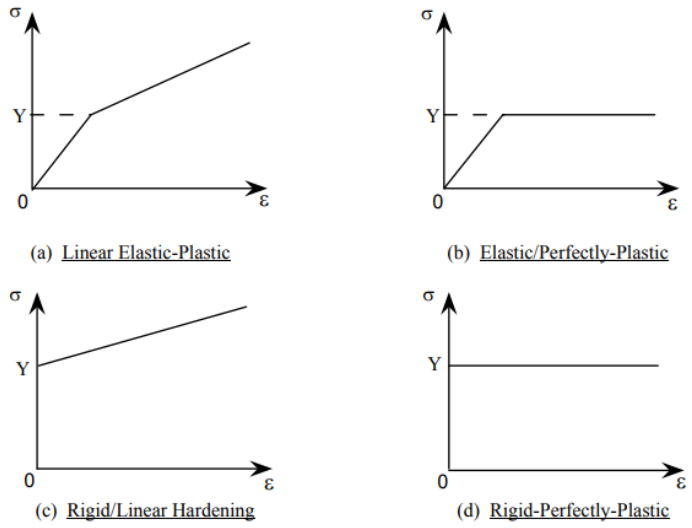



**Figure 3-2: Shows the different types of material elastic and plastic behaviours [173]**

One of the major challenges of plasticity is to find the most realistic way to describe the strain-hardening potential and include it in material modelling. The strain-hardening can be modelled by an increase in the von Mises yield surface of a material as the material yields as shown in Figure 3-3 below. In the case of isotropic hardening, the tangent modulus is used to determine the increase in yield stress with increasing strain [175]–[178]. The tangent modulus shown in Figure 3-4 is calculated by measuring the slope of the tangent to the stress-strain curve.

In the plastic region deformation is both elastic and plastic and can be modelled as [179]:

$$d\varepsilon = d\varepsilon^e + d\varepsilon^p \qquad (\mathbf{3.1})$$

The plastic modulus (H) also known as work hardening, defined as the ratio between stress ($\sigma$) and plastic strain ($\varepsilon^p$), as follows [179]:

$$d\sigma = \mathrm{H} d\varepsilon^p \qquad (\mathbf{3.2})$$

By solving between the equations, the tangent modulus (K) as:

$$\frac{1}{\mathrm{K}} = \frac{1}{\mathrm{E}} + \frac{1}{\mathrm{H}} \qquad (\mathbf{3.3})$$

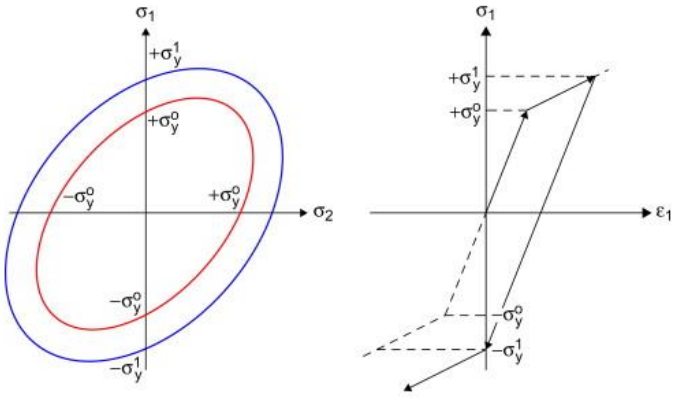



**Figure 3-3 The yield surface of a material expanding isotropically as a result of strain hardening [179]**

Then the tangent modulus (*K*) can be defined and calculated from the tensile test as the slope of the tangent to the curve as shown in Figure 3-4 below.

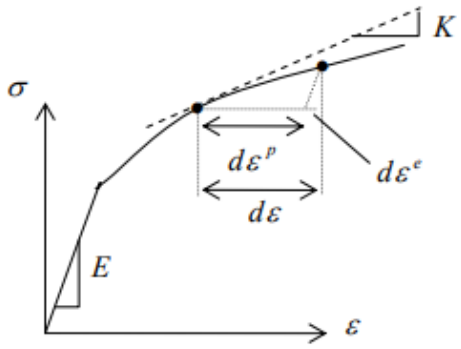



**Figure 3-4: Determination of tangent modulus by determining the slope in the plastic region [176]**

The hardening behaviour of materials differ based on the loading conditions present. In general for viscoplastic materials one of these models can be used [176]–[178]:

- Isotropic hardening: In this case, the yield surface of the material is not translated or rotated, but rather expanded uniformly in all directions. This model works best for materials that do not change loading mode.
- Kinematic hardening: If the Bauschinger effect is taken into consideration, i.e., when the loading on a material is changed from tension to compression; it introduces anisotropy in the material which shifts the yield surface in a direction making it weaker in compression.

The discussion of plasticity and hardening contacts is essential for identifying real contact area. In the statistical theoretical models for measurement, Hertzian elastic contact theory is the basis of the elastic contact. Defining the elastic limit in contacts and how material behaviour may change after that is still a matter of debate between researchers.

## 3.3 General Principles of Particle-Based Modelling

Particle simulation concerns any system that can be modelled as a large collection of bodies/nodes/points or simply particles. Here the interactions that occur between particles are checked contact by contact, and the motion is calculated particle by particle, or particle by wall, unlike continuum-based models, as explained in Ch.2.

The structure of a particle-based program is generally the same, whether simulating a system of atoms on the atomic scale, molecules on the nano- scale (e.g. using MD), particles on the meso-scale (e.g. using MCA) or granular particles on the macro-scale (e.g. using DEM). Both LAMMPS and LIGGGHTS are such programs but from here on further we will only refer to LIGGGHTS because it is the platform used to implement the 3D MCA elastic-plastic model.

As shown in Figure 3-5, a particle-based program consists of a time-loop to calculate the forces on the particles and their positions at a certain time-step. This time-loop starts with the initial positions and velocities of the particles, the systems dimensions, geometry and boundary conditions. Then the forces on the particles are calculated due to internal interactions and external forces, and then the positions and velocities are updated to simulate the evolution of the system over time by integrating the equations of motions. This is repeated by calculating the new forces based on the new positions and velocities until the last time-step is reached.

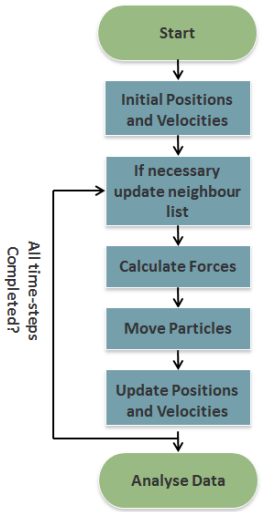



**Figure 3-5: Flowchart showing the particle-based simulation loop [180]**

To calculate the forces, here the classical Newtonian laws of mechanics are used as an approximation to relativistic mechanics of particles moving with speeds much less than the speed of light. The forces depend on the relative distances and velocities between the neighbouring pairs, triplets, etc. of particles. The larger the number of particles, the more distances and velocities need to be calculated making this step the most computational and time-consuming part of the program. It is also

the most important step and the accuracy of the force calculations defines the accuracy of the model. However, to reduce the computational cost, nearby particles are identified using neighbour lists which may be updated at every time-step.

Every particle-based method and simulation tool involves these main functionalities. However, depending on the scale the method describes and the accuracy of the model, the details of each method and the material behaviour they are capable of modelling may vary enormously. Hence, the numerical/theoretical aspects of DEM and MCA methods are described in sections 3.4 and 3.5 respectively. But first the basics of particle-based simulations are covered here, and the computational aspects, such as the integration, time-stepping, the thermodynamics of the system and the boundary conditions, etc. The computational aspects are quite similar for both MCA and DEM and used within LIGGGHTS platform, hence these are first described before going into detail of the numerical aspects.

### 3.3.1 Discretization and Initial Configuration

When starting a particle-based simulation, the first requirement is that the system at hand must be discretized into a series of small elements of finite size. This is done by specifying the type, shape and size of the particles, and then a crucial step is to choose the particles initial configuration. Initial assumptions are also made to simplify the system and design the output. The discretization depends on the method of choice and the system at question. As mentioned before, the particles could be spherical atoms/molecules in MD simulations, or have different particle shapes in DEM simulations, etc. Furthermore, the size of the particles is defined corresponding to the characteristic particle size of the phenomena of study.

In terms of particle configuration, the particles are either put in a lattice structure configuration or inserted randomly. When dealing with an atomic crystal structure, the particles are placed on the appropriate lattice structure corresponding to the minimum energy configuration of the material under the desired thermodynamic conditions [180]. The three most common lattice configurations used for spherical particles are the simple cubic (*sc*), body-centred cubic (*bcc*) and face-centred cubic (*fcc*), as shown in Figure 3-6(a). Even when the system is not a crystal but a liquid or dense gas, the crystal lattice structures are often initially used to avoid potential problems with overlapping particles.

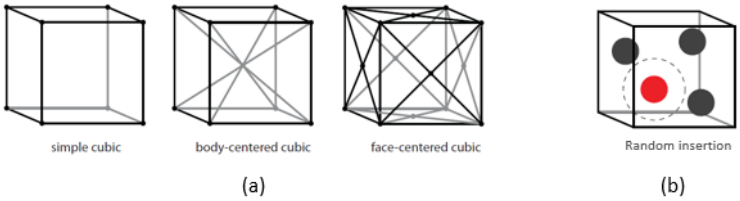



**Figure 3-6: (a) the three main crystal lattice structures (b) random insertion [180]**

However, if the particles shapes are not spherical, a large amount of time is required which causes artificial errors, in which case random insertion of the particles, is a better option as shown in Figure 3-6(b), which is the case in most DEM models.

### 3.3.2 Particle Interaction Forces

In particle-based simulations, each particle interacts with its neighbouring particles or the boundary walls. The forces applied to it are modelled using some form of formulas which differ depending on the scale in question. These interparticle forces can be short-range forces which tend to be molecular repulsive forces and are negligible when the distance between the particles is greater than a few angstroms. They can also be long-range forces that are attractive forces such as van der Waals forces, responsible for physical properties such as friction, surface tension [181].

In MD, the interatomic forces are calculated using potential functions which represent the energy bonding between atoms and have their origins from quantum mechanics. The potential functions are expressed as a summation of the interactions between the particles. Hence, the force ($\vec{F}_i$) in the equation of motion can be expressed as the gradient of the potential energy ($V$).

$$\vec{F}_i = -\nabla_i V \tag{3.4}$$

$$-\frac{dV}{dr_i} = m_i \frac{d^2\vec{r}_i}{dt^2} \tag{3.5}$$

where ($V$) is the potential energy of the system and the change in position as a function of time are derived from Newton's equation of motion. To calculate the trajectories of the particles, the initial positions of the particles and their initial distribution of velocities and acceleration are needed. The calculation of total energy of the system is very important because it defines all information about the structure, properties and thermodynamics of the system. Since the total energy of the system depends on the interaction between the particles; which depends on the structure of the solid lattice, the accurate calculation of these interactions is very crucial and is the main concern in modelling [182].

A crucial point to mention here is that there are many models available for calculating the forces between individual particles in particle-based simulations, but in general they take two different forms which determines the accuracy and complexity of the model. They either take a pair approximation form or a many-body form, which goes back to the principles of Schrodinger equation and the Born-Oppenheimer approximation, but this is beyond our scope. In short, the Born-Oppenheimer approximation states that to calculate the potential energy of the system ($V$) we can get rid of the electrons and only consider the effective interaction of nuclei which is called the potential energy surface *V(R)*.

The pair potential approximation assumes that the total potential energy of the system *V(R)* is equivalent to the sum of pair potentials of the surrounding atoms. In MD, the most common pair potential is Lennard-Jones potential [183]. The pair-potential approximation takes the following form:

$$V(R) = \frac{1}{2} \sum_{i,j(j \neq i)} V(R_{ij}) \qquad (\mathbf{3.6})$$

Here any one atom at any specific instance, can have only one bond with a single neighbour, and the strength of one bond between pairs is independent of any other bonds with the surrounding neighbours. This raises a great error because the bonds between atoms or particles do depend and affect each other. Furthermore, the strength of the bond decreases with increasing density. Pair-potentials are especially not capable of correctly describing metallic bonding, there has been a lot of evidence throughout the years that metallic bonding are not pair-wise. Pair-potentials also have a tendency to form closed packed structures, incorrectly describe the surface energies, has no stability against shear and no angular dependency [139], [140].

Alternatively, many-body potentials are coordination-dependent; they manifest themselves in the fundamental properties of the solid material [186]. Here energy of atoms depends non-linearly on the surrounding atoms (number and distance) and uses electron density as a measure of the surrounding atoms. One of the main reasons why many MD models provide a very well understanding of different fundamental solid material behaviours is because they use a many-body potential form, however it can only describe material on the atomic scale [69].

The most known many-body potential used is the embedded atom model (EAM) [186]. Here the potential energy of an atom *i* is generally expressed as follows:

$$V_i(\mathrm{R}) = \sum_{j \neq i} \varphi(r_{ij}) + \sum_{i} F(\bar{\rho}_i) \qquad (\mathbf{3.7})$$

where it depends on the pair interaction potential ($\varphi$) as a function of distance between atom *i* and *j*, and an electron charge density-dependent function ($\bar{\rho}_i$) which is the sum of contributions of neighbour atom *j* to the local density of atom *i*:

$$\bar{\rho}_i = \sum_{j \neq i} \rho_j(r_{ij}) \qquad (\mathbf{3.8})$$

DEM uses an approximated pair-wise interaction form which is one of the reasons why DEM fails to model continuous cohesive materials, while MCA solved this by developing a many-body interaction form based on the framework of the many-body embedded atom model [186] which is why MCA carries the advantages of MD models as well. This will all be explained in the following sections.

### 3.3.3 Integration of Equations of Motion

As mentioned before, in particle-based methods the motion of each particle in a system of $N$ number of particles is traced to define the behaviour of the system as a whole. The translational and rotational motion of each particle is governed by Newton's second law of motion as follows:

$$\vec{F}_i = m_i \vec{a}_i = m_i \frac{d\vec{v}_i}{dt} = m_i \frac{d^2\vec{r}_i}{dt^2} \qquad (\mathbf{3.9})$$

$$I_i \frac{d\vec{\omega}_i}{dt} = M_i \qquad (\mathbf{3.10})$$

where ($\vec{F}_i$) is the sum of all forces acting on particle $i$ both from the interaction with its surrounding neighbours and for external forces, ($m_i$) is the mass of $i$, ($\vec{a}_i$) is the acceleration of $i$,($\vec{v}_i$) is the translational velocity, ($\vec{\omega}_i$) is the rotational velocity, ($I_i$) is the moment of inertia and ($M$) is the resultant torque acting on the particle.

The equations of motion are used to model the evolution of the system over time, and obtaining the new positions, velocities and accelerations of all the particles due to the forces on the system by integrating the equations of motion. The important thing to mention here is that, how the forces between the particles are obtained is the main factor that distinguishes one method or model from the other, and how accurate or realistic the model is.

The equations of motion lead to a system of differential equations that cannot be solved analytically, and a numerical integration method must be used to advance the system from time $t$ to time $t + \Delta t$. The Velocity-Verlet algorithm [187] is considered the most well-known time integration algorithm used in particle-based simulations to calculate the positions and velocities of the particles at every time-step. It is a very stable integrator compared to the simple Euler method. It is also time reversible, conserves volume in phase space, has good numerical precision, efficient and is relatively easy to implement which is why both LAMMPS and LIGGGHTS use it to solve the equations of motion.

According to the Velocity-Verlet integration scheme [187], as shown in Algorithm 1 below, the first step is to update the velocities by a half-step, then the positions by one step. Then calculate the acceleration according to the interaction. Then update the velocities by another half-step. Of course, boundary conditions (walls or periodic boundaries) are applied.

The Velocity-Verlet algorithm requires the current forces to be calculated before the first time-stepping. These forces are computed according to the particle-interaction form as explained in the previous section 3.3.2.

**Algorithm 1:** Velocity-Verlet integration scheme

1- Calculate $\vec{v}\left(t+\frac{1}{2}\Delta t\right)=\vec{v}(t)+\frac{1}{2}\vec{a}(t)\Delta t$

2- Calculate $\vec{x}(t+\Delta t)=\vec{x}(t)+\vec{v}\left(t+\frac{1}{2}\Delta t\right)\Delta t$

3- Derive $\vec{a}(t+\Delta t)$ from the interaction forced using $\vec{x}(t+\Delta t)$

4- Calculate $\vec{v}(t+\Delta t)=\vec{v}\left(t+\frac{1}{2}\Delta t\right)+\frac{1}{2}\vec{a}(t+\Delta t)\Delta t$

It is common to not really use the velocities themselves in the Velocity-Verlet algorithm to solve Newton's equations, but instead use the positions of the particles in the current time step and the previous ones, combined with the knowledge of the forces acting on the particles ($F=ma$), to predict the position and acceleration at the next time step [187]. This is relevant to the code development in Chapter 4.

In DEM, the calculations alternate between two main calculations. The first is Newton's second law which determines the motion of each particle using the contact forces. The second is the force-displacement law which updates the contact forces due to the relative motion between two particles. This is not the case in MCA which will be explained in section 3.5.

### 3.3.4 Time-Step Determination

Following from the previous section, knowing the particles positions, velocities and other dynamic information at time $t$, the algorithm calculates the new position, velocities etc. for a time $t + \Delta t$ where the step size $\Delta t$ is kept constant for the specified total time. Hence, choosing the correct time increment is an essential step and one of the most important parameters in particle-based simulations. It influences how long a simulation runs and how much computational power it requires. Furthermore, small time steps make for a more continuous motion and more realistic simulation, but large time steps reduce the computational cost.

In the end, simulations are said to be valid when the simulation time for the properties of interest are greater than the relaxation time of those same properties. To avoid discretisation errors, the timestep must be chosen small enough to match the kinetics of the natural process and to capture the phenomenon of energy transmission by wave propagation. Furthermore, the simulations should be relevant to the time scales and be comparable to the natural kinetics. Hence, a few parameters should be taken into account to minimise the cost and duration of the simulation to a reasonable amount. This includes the time-step, the simulation size and the total duration of the simulation. In DEM, Rayleigh time is usually used to determine the time-step. Rayleigh time is the time taken by the Rayleigh wave to propagate through a solid particle, and it can be approximated as follows [188]:

$$T_R = \pi R \left(\frac{\rho}{G}\right)^{1/2} / (0.1631v + 0.8766) \qquad (\mathbf{3.11})$$

where ($R$) is the particle's radius, ($\rho$) is the density, ($G$) is the shear modulus and ($v$) is the Poisson's ratio. A time-step between the range of $0.1T_R$ - $0.3T_R$ is recommended.

For MCA, it is recommended for the time step to be smaller than the time of longitudinal sound propagation through the particle, and so we use a maximum time-step of:

$$\Delta t = \frac{d}{2c_p} \quad , \quad c_p = \sqrt{\frac{K + {}^4/_3\, G}{\rho}} \qquad (\mathbf{3.12})$$

where ($c_p$) is the longitudinal (pressure) wave velocity, ($K$) is the volume compression modulus or bulk modulus, ($d$) is the diameter of the automaton. In the case of various properties of automata, the maximum ($c_p$) is chosen from all the sample automata.

### 3.3.5 Thermodynamics - Ensembles

The thermodynamic state of a system is defined by three main parameters; temperature (*T*), pressure (*P*) and number of particles (*N*). Other properties can be derived from the equations of state, however, for particle-based simulations, three main ensembles were defined. "An ensemble is a collection of all possible systems which have different microscopic states but have identical macroscopic or thermodynamic states" [23]. These three ensembles are [23]:

1- NVE – the microcanonical ensemble – here the Number of atoms, Volume and Energy are kept constant. This corresponds to an isolated system where energy and entropy are conserved. It is an adiabatic process with no heat exchange. The equations of motions are solved without temperature or pressure control. However, realistic systems change energy, volume and particles with it surrounding which makes it more complicated and which is why the next ensemble is most frequently used.

2- NVT – the canonical ensemble – here the Number of atoms, Volume and Temperature are kept constant. Here, the energy of endothermic and exothermic processes is exchanged with a thermostat. For example, a very popular one is the Nose –Hoover thermostat which controls the temperature and includes velocity rescaling.

3- NPT – the Isobaric-Isothermal ensemble – here the Number of atoms, Pressure and Temperature are kept constant.

LAMMPS uses these three ensembles to define the thermodynamic state of a system of particles. However, LIGGGHTS only uses the NVE ensemble for granular particles, and we will use it in LIGGGHTS-MCA as well.

### 3.3.6 Boundary Conditions

Any particle-based simulation occurs within a box or container with a specific size and shape. What happens inside a system is determined by the conditions that apply at the boundaries of the system. The boundary conditions are what define the surroundings containing the system; they set the conditions at the boundaries to imitate different mechanical loading systems such as compression, tension, shear, vibration, etc. Here we control the velocities, pressures and temperatures set on the system, so that the sample is simulated as a whole under real operating conditions.

If the topic of interest is the bulk properties of a system, then the wall boundaries should be as far away as possible from the region of interest because in reality walls are present over a much larger length scale and the simulation system is very small compared to the actual matter. Also, most atoms will be located on the surface which is also unrealistic. This could be solved by a rigid or periodic boundary conditions (PBC). The PBC is basically a replication of the box which encloses the particles, where the particles are considered to be identical, repeating and infinite as shown in Figure 3-7, creating an infinitely large grid of simulation boxes. In other words, particles exiting from the boundary on one end will re-enter the boundary on the other end, creating periodic movement. Moreover, particles within a cut-off distance from the boundary on one end interact with the particles that are the same distance from the boundary on the other end [189]. LIGGGHTS applies this while calculating the forces and updating the positions and velocities of the particles. However, if the boundary conditions are chosen to be rigid; then the particles at the microscopic level are affected by the edge and wall of the system.

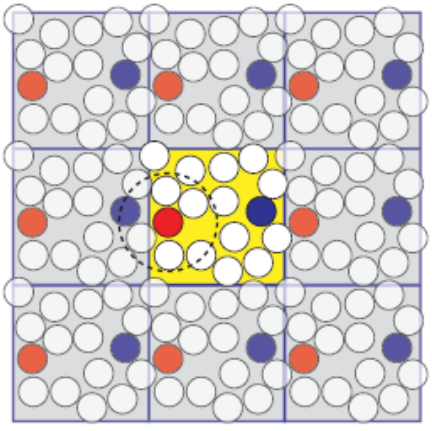


**Figure 3-7: Schematic showing periodic boundary conditions applied on a simulation box with a defined cut-off distance [180]**

### 3.3.7 Generation and Updating Neighbour List

The calculation of the forces is the most time-consuming part of a simulation code because it involves calculating the interactions between neighbouring pairs, triplets, etc. of particles (pair-interactions or many-body interactions). The higher the number of particles, the relative distances and velocities between the neighbouring pairs, the higher the computational time to evaluate the forces between them. Hence, an important step of any particle-based simulation is defining which particles are in contact; and the forces will only be calculated for these particles, to reduce computational cost. This is also called contact detection in DEM and is the second most time-consuming part of a simulation.

In most cases the particles have a certain interaction range beyond which the interactions are zero, or so small that they may be neglected. This range is called the cut-off distance $r_{cut}$. Moreover, in one time step the particle positions do not change much, so a particle will be surrounded by the same set of closest neighbours for a considerable amount of time. The computational time is reduced by using this cut-off distance and creating a neighbour lists to identify the nearby particles and only update and calculate the forces on the particles within the neighbour area within a given time step. LIGGGHTS uses the neighbour list technique. As shown in Figure 3-8, considering the red particle, the neighbour list consists of all the particles within the cut-off distances which are particles 1 to 4.

The drawback is that the neighbour list needs to be updated frequently, which is undesirable because it is a time-consuming process to generate the list. To avoid this, the additional shell of thickness ($r_{shell}$) is necessary to already include neighbours in the neighbourlist that may enter the cut-off radius at a later time as they may move towards the central particle. In Figure 3-8 those would be particles 5-7. The larger the ($r_{shell}$) the longer the same neighbour list can be reused over several time-steps; usually 10 to 50 time-steps. However, a thicker shell also implies that the total list range is larger ($r_{list} = r_{cut} + r_{shell}$), and that a larger number of pair distances need to be evaluated each time step.

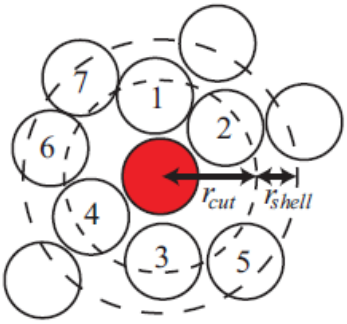



**Figure 3-8: Schematic showing how the neighbour list is generated according to the cut-off distance and shell thickness [180]**

### 3.3.8 Parallelisation and Communication

The main constraint of particle-based simulations is that they require many calculations and a lot of CPU time. The time taken to run the simulation grows exponentially as the size of the system increases. To overcome this, most particle-based simulations are designed to be implemented in parallel. There are several ways to do this parallelisation, but LIGGGHTS uses the spatial decomposition (or domain decomposition) communication-based approach.

The spatial decomposition approach allows for the discretization of the simulation domain into small regions that are then assigned to different processors as shown in Figure 3-9. The regions are then further divided into cells. This means that each processor is responsible for the particles in its own region with the exception of particles that cross boundaries. In this case, communication happens between neighbouring processors providing that information, and hence, the amount of information required to be shared is minimal. This vital information is chosen carefully. Once the forces are calculated for each particle within their domain, they are cross-communicated to their surrounding domains as a resultant force using MPI exchange. Hence, communication between the processes is done via MPI routines where the communicated data consists of information about so-called “ghost particles” which are all the particles that border their subdomain within the range of interaction.

This parallelization method is very efficient, but only when the number of particles within each domain is balanced so that the information exchange happens smoothly. A large number of domains will cause slower computation because the amount of cross communication required will be high; and a very small number of domains will also slow down the computation because the load per processor will be too high. Thus, the optimisation of the numbers of domains and the number of elements in each domain is crucial. The challenge here is to make sure only the needed information is cross-communicated to reduce the memory and communication time.

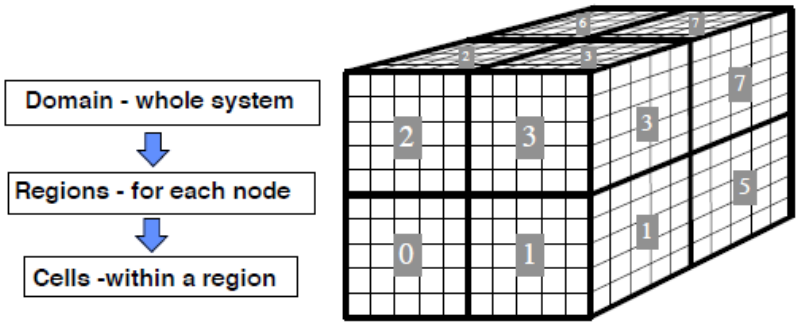



**Figure 3-9: Spatial decomposition parallelisation approach [190]**

## 3.4 Numerical Aspects of DEM Models

Here the numerical aspects of DEM are presented emphasizing its limitations which are overcome by MCA which will be explained in section 3.5. This is important to define the MCA functionalities which will be implemented in LIGGGHTS.

### 3.4.1 DEM Particle Discretisation and Initial Configuration

DEM is concerned with modelling granular materials which is what LIGGGHTS is developed for modelling. A granular material is an assembly of many discrete solid particles, whose typical size ranges from micrometres to centimetres. These granular particles are typically interacting with each other because of dissipative collisions and are usually dispersed in a vacuum of interstitial fluid [191]. Granular particles are very different from solids, liquids or gases, and can be considered as a fourth state of matter. The dynamics of these systems are dominated by gravity and friction effects meaning that without further perturbations the particles will quickly come to complete rest [180]. For example, if a pile of sand is found on an inclined plane with an inclination angle smaller than the response angle, it acts as a solid because of the static friction between the plane and the granular particles. However, when the inclined plane is greater than the response angle, the granular particles start to flow exhibiting a fluid-like behaviour, and this behaviour is called granular flow. Other familiar granular particles are present in rock avalanches, emptying hoppers filled with grains, pneumatic conveying of particles and powders, and mixing and segregation of particles when they are transported and shaken [191].

The term “granular materials” and “bulk solids” are used interchangeably in the literature. Granular bulk solids are materials that exhibit the properties of both solids and fluids, such as coal, sand, ore, mineral concentrate, and crushed oil shale [192]. Since in our daily lives more than fifty percent of all products are either granular in form or involve granular materials in their production, and about forty percent of the value added in chemical industry is linked to particle technology [191], DEM is a very important modelling method and why LIGGGHTS is directed to simulate models involving macroscopic scale processes involving granular materials such as the examples below in Figure 3-10.

However, even though DEM has been successful in modelling granular materials and granular flow, it has failed to accurately simulate continuous cohesive materials. It is worth noting here the distinction between bulk solids in the context of granular materials, as explained above, and bulk continuous cohesive materials such as metals. For example, simulating a tensile test of a cohesive continuous material, using DEM is still difficult to conduct and the relationship between DEM parameters and mechanical parameters are still not defined [123], [166].

Granular materials are characterised by mechanical properties of single particles such as density, Young's Modulus, Poisson's Ratio, Shear Modulus, and by parameters depending on mechanical properties and geometry such as coefficient of restitution, static friction coefficient, rolling friction coefficient, etc. They are also characterized by bulk properties, such as lateral pressure ratio, angle of repose, bulk density, size distribution, wall friction, internal (bulk) friction. All these parameters are obtained empirically or analytically and calibrated which is why all DEM studies involve extensive parametric and calibration studies which plays a crucial role in the accuracy of the results obtained. A unified calibration technique for DEM parameters does not exist, even the particle shape, size and distribution has a huge effect on the outputs [193]–[195]. The accuracy of densely packed models is even more sensitive to initial density, contact orientation, particle size, shape, and distribution [196], [197].

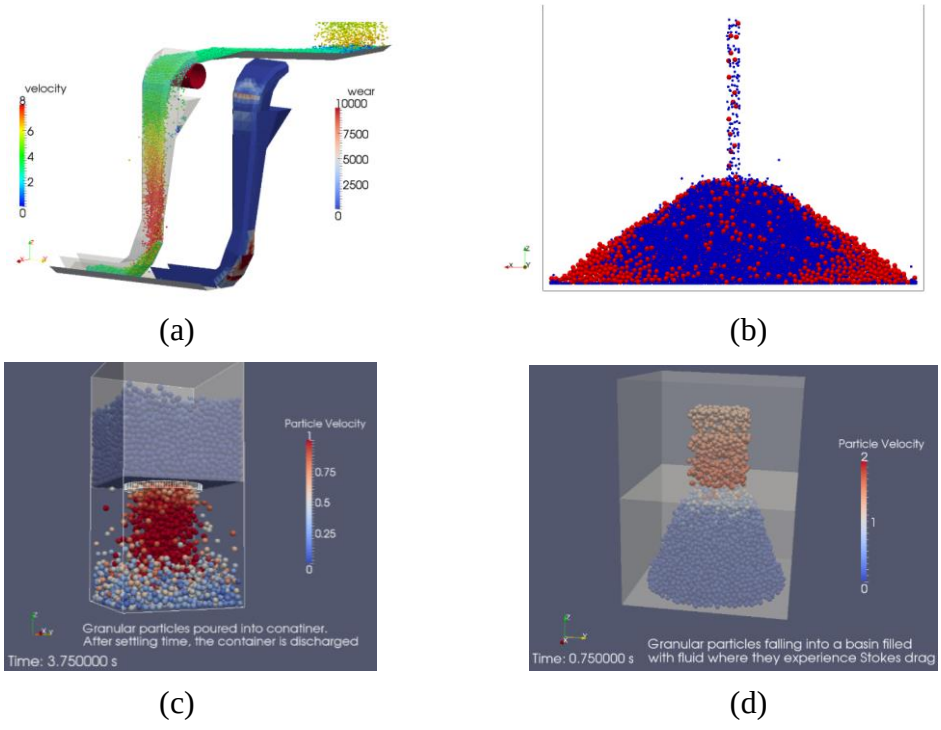



**Figure 3-10: Examples of simulations conducted by LIGGGHTS (a) granular particles moving on a conveyor belt, (b) particle segregation in a slot geometry, (c) a hopper discharge of granular particles, (d) granular particles flowing into a water basin [198]**

Distribution and density of particles are also important because they affect shear and consolidation. Densely packed granular material expands as it starts to flow depending on the level of consolidation. For example, the volume of under-consolidated particles like loosely packed powders decreases during shear [199], while the volume of over-consolidated particles such as tightly packed powders increases, and no volume change occurs in critically consolidates during shear [200].

Regarding the discretization of the subject material in DEM and initial configuration, random insertion of particles is usually used since the particles

usually have different sizes and/or shapes. An example of random insertion in DEM is shown in Figure 3-11. However, obtaining an initial configuration is a crucial and non-trivial task in DEM, because it is very important that the particle packing represents the problem at hand and ensures good assemblies obtaining macroscopic isotropy. Particle packing and distribution directly affects the properties of the bulk solids at the macroscale and should be taken into account during calibration [190].

Many particle packing and generations techniques are available in DEM, but they can be divided mainly into two groups; dynamic techniques and constructive techniques [201]. In dynamic techniques, previous DEM simulations are used to produce the initial packing, however, constructive techniques use initial packing according to geometrical calculations. Some of the most common dynamic techniques are multi-layer compaction, particle growth, isotropic compression and gravitational deposition. While the most common constructive techniques include sequential inhibition, sedimentation techniques and regular arrangements. Depending on the types, shapes, sizes, etc. of the particles, the proper particle packing, and generation technique can be chosen. Details of the techniques are beyond our scope; however, it is clear to see that this is a complex aspect of DEM.

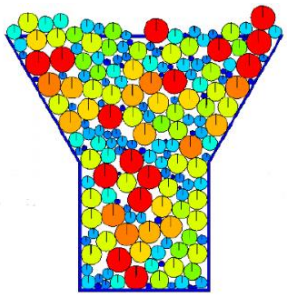


**Figure 3-11: Example of initial configuration in DEM for spherical particles [202]**

For the purpose of this thesis, we are not dealing with granular materials, but we are interested in solid continuous cohesive material behaviour, most importantly metals, to study fracture and/or plastic deformation of materials. Some DEM studies have attempted to do this, but they faced problems involving segregation. The first most common problem is segregation by percolation, which means small particles fall into the void spaces between large particles, the second is segregation by angle of repose, which is the case when the angle of repose for small particles is larger, thus the large particles roll down the heap, and the third is segregation by density-push-away where heavier particles push lighter ones away [200].

Others tried for example using a particle radius expansion model to discretize continuous material to obtain a homogenous assembly [166]. However, even with these many attempts, including parametric studies and calibration methods, DEM still struggles to simulate continuous behaviour due their contact laws and force descriptions[123], [166].

### 3.4.2 DEM Contact Models

In DEM, elements interact through contact forces determined by a contact law describing the material's constitutive behaviour. These laws formulate the material model on the microscopic level and are the most important part of the model [191]. This actually uses the contact mechanics theory described in section 2.6.1.1.

#### 3.4.2.1 Contact of Spheres

The contact models largely depend on the definition of the contact between two particles. As shown in Figure 3-12, there are three approaches to define the contact between particles in DEM. The hard particle approach [203], [204], and the soft particle approach [28] assume that the particles are rigid and do not deform. The hard particle approach assumes the particles to be perfectly rigid and follow a constant motion until a collision occurs. It is event-driven, meaning that only when collision is detected are the forces computed, which is different from the soft approach which are time-driven and forces are computed at every time-step. The hard approach is much simpler because the details of the contact behaviour are ignored, but mainly only used in rapid collisional granular flow [191], [205].

The soft particle approach assumes the particles are rigid bodies but are allowed to slightly deform only at the contact points, and the deformation is described by means of a small overlap of the particles. These small deformations are used to calculate the forces between the particles [206]. Both approaches have limitations due to the assumption of rigidity because most materials involve inelastic deformations. The evaluation of the inter-particle forces based on the overlap in the soft particle approach is not sufficient to account for the inhomogeneous stress distribution inside the particles [205]. Hence, the third approach introduces deformability to the particles and is called the finite discrete element method [207], where the particles are discretized with finite elements. However, this approach is very computationally expensive and cannot be used for a large number of particles.

The soft particle approach is the most common approach and the one used in LIGGGHTS. Pioneering work of DEM, specifically the soft approach started in the field of MD of granular materials, and was introduced by Cundall and others [122], [208]–[210]. However, it follows the MD pair-wise approximation and hence it naturally inherits its limitations and this poses a drawback in the contact law calculations using the soft approach [211].

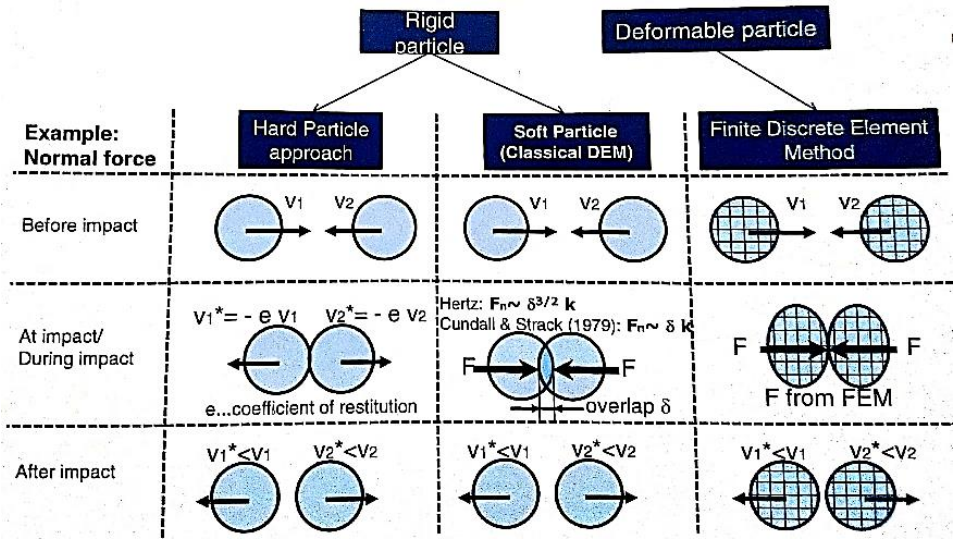



**Figure 3-12: The three types of DEM approaches [200]**

The dynamics of a particle is governed by Newton's equation of motion as mentioned earlier. The force ($\vec{F}_i$) and the torque ($\vec{M}_i$) acting on particle *i* are given by the sum of the pairwise interactions of particle *i* with its neighbouring particles ($N_i$):

$$\vec{F}_i = m_i \frac{d^2\vec{r}_i}{dt^2} = \sum_{j=1.j\neq i}^{N_i} \vec{F}_{ij} \qquad (\mathbf{3.13})$$

$$\vec{M}_i = I_i \frac{d\vec{\omega}_i}{dt} = \sum_{j=1.j\neq i}^{N_i} \vec{M}_{ij} \qquad (\mathbf{3.14})$$

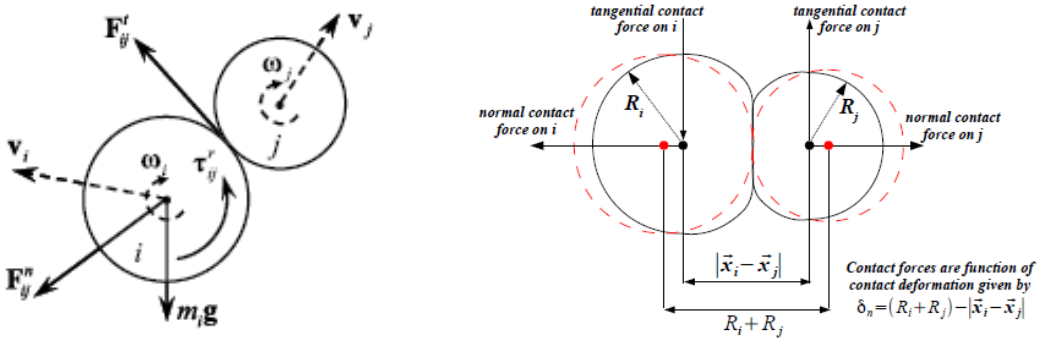
Hence, the total force on a particle is the sum of pair forces with its surrounding neighbours. "The limitation to pairwise interaction is an abstraction which is justified if the particles deform each other only slightly. For stronger interactions one has to take multi-particle interaction into account." [191], which is done by MCA. DEM assumes this problem is solved by choosing a small time-step, smaller than a critical value as explained in section 3.3.4., such that during a single time-step the disturbance cannot propagate from the particle to other particles farther than its immediate neighbours. So at all times the resultant forces on a particle can be determined exclusively from its interaction with the contacting particles for coarse particles, and for fine-particles some non-contact forces can be included, such as van der Waals and electrostatic forces [206]. Assuming that there is a force between two neighbouring particles only when they are in contact makes simulations using spherical particles very numerically efficient since identifying two particles in mechanical contact can be done in a very simple way. Particles are said to be in contact and exerting forces on each other only if the distance between two particles

of radii ($R_i$) and ($R_j$) is less than their contact distance ($R_i + R_j$), as shown in Figure 3-13. Otherwise, there is no force between the particles.

Hence, particles are in contact if:

$$\delta_n = (R_i + R_j) - \|x_i - x_j\| > 0 \qquad (3.15)$$

where ($x_i$) and ($x_j$) are the centers of the particles and ($R_i$) and ($R_j$) are their respective radii. The overlap ($\delta_n$) is also called the mutual compression of particles *i* and *j*. If the shape of the particles are not spherical, the detection of contacts is much more complicated [212], [213].




(a) (b)

**Figure 3-13: A schematic representation of contact (a) before deformation [214] (b) after deformation, where the original, undeformed configuration is indicated in red, and the final configuration in black [215]**

The force between contacting particles is divided into a total normal force component ($\vec{F}_{ij}^n$) which changes of the translational motion and a tangential force component ($\vec{F}_{ij}^t$) which changes of the rotational motion of the particles, as shown in Figure 3-14, and represent the tensile/compressive and shearing directions of contact. The resultant torque comprises two components, the torques caused by rolling friction ($\vec{M}_{ij}^r$) and torque caused by the tangential force ($\vec{M}_{ij}^t$).

$$\vec{F}_i = \vec{F}_{ij}^n + \vec{F}_{ij}^t \qquad (3.16)$$

$$\vec{M}_i = \vec{M}_{ij}^r + \vec{M}_{ij}^t \qquad (3.17)$$

According to the linear spring-dashpot model, as shown in Figure 3-14, the normal force has a contact force and a damping force, while the tangential force has a shear force and a damping force. Once the overlap is detected the contact force is:

$$F = F_{n(elas)} + F_{n(damp)} + F_{t(fric)} + F_{t(damp)} \qquad (3.18)$$

The contact force is the result of the elastic forces ($F_{n(elas)}$), viscous forces ($F_{n,t(damp)}$) and frictional resistance forces ($F_{t(fric)}$) between the moving particles.

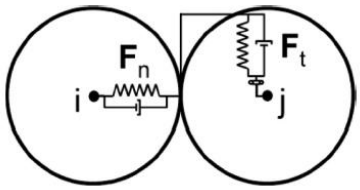



**Figure 3-14: Linear spring-dashpot model [216]**

According to the linear spring-dashpot model [216] which is also what LIGGGHTS [189] uses, these force can be described as:

$$\vec{F}_{ij} = \vec{F}_{ij}^{n} + \vec{F}_{ij}^{t} = \left[k_n \delta_n - \gamma_n (v_{ij}^{n})\right] n_{ij} + \left[k_t \delta_t - \gamma_t (v_{ij}^{t})\right] t_{ij} \qquad (\mathbf{3.19})$$

- $\delta_n$ is the normal overlap vector, $\delta_n = (R_i + R_j) - \|x_i - x_j\|$.

- $\delta_t$ is the tangential displacement or tangential overlap vector between i and j, $\delta_t = \int_{t_0}^{t} v_t dt$, $t_0$ is the time when the two particles just touch and have no deformation, $t$ is the time of collision.

- $n_{ij}$ is the unit vector in the normal direction, $n_{ij} = (x_i - x_j)/\|x_i - x_j\|$, and $t_{ij}$ is the unit vector along the tangential direction, $\hat{t}_{ij} = (v_{ij} - v_{ij}^{n})/\|v_{ij} - v_{ij}^{n}\|$, where $v_{ij} = v_i - v_j$ .

- $v_{ij}^{n}$ and $v_{ij}^{t}$ are the normal and tangential components of relative velocity, $v_{ij}^{n} = (v_{ij} \cdot n_{ij})\, n_{ij}$ and $v_{ij}^{t} = v_{ij} - v_{ij}^{n}$.

- $k_n$ and $k_t$ are the elastic stiffness constants for normal and tangential contacts.

- $\gamma_n$ and $\gamma_t$ are the viscoelastic damping constants for normal and tangential contacts

Calculating $k_n$, $k_t$, $\gamma_n$, and $\gamma_t$ depend on the specific contact model. It will be noted that "a significant degree of variation exists in the literature for the exact values of the contact stiffness coefficients $k_n$ and $k_t$. The same is true for the mass proportional damping coefficients $\gamma_n$ and $\gamma_t$. In fact, the latter are frequently simply chosen sufficiently large to eliminate numerical noise in the DEM simulations" [217], [218]. The next subsections describe models used in LIGGGHTS for modelling normal and tangential forces including calculating the stiffness and damping coefficients. Also the torques, cohesive and non-contact forces are covered. For a more detailed review on interaction forces refer to [191], [213].

#### 3.4.2.2 Normal Force

As shown in equation (3.16) the normal force is calculated as:

$$F_{ij}^{n} = k_n \delta_n - \gamma_n (v_{ij}^{n}) \qquad (\mathbf{3.20})$$

When two particles interact with each other, they produce repulsive forces because of elastic surface deformation. In LIGGGHTS, this is defined by viscoelastic models based on Hertzian or Hookean theories. Here it is assumed that the particles are

perfectly smooth, the behaviour is elastic and isotropic, the tangential component of the force does not affect the normal component, and the contact deflection is smaller than the contact zone radius. Also, no tension or friction forces are allowed in the contact area [219]. In other words, Hertz and Hookean theories only describe the forces caused by physical contact between the particles due to impact, but other models could include cohesion for example, which will be explained in further sections.

LIGGGHTS uses the Hertz-Mindlin contact model which is a non-linear contact model and an improvement of the simple Hertz Model. It is based on an approximation of the Hertz theory proposed by Mindlin and Deresiewicz in 1953 [220], and described as [189], [206], [216]:

$$k_n = \frac{4}{3} E_{ij} \sqrt{R_{ij} \delta_n} \quad , \quad \gamma_n = -2 \sqrt{\frac{5}{6}} \beta_{eff} \sqrt{S_n m_{ij}} \tag{3.21}$$

According to Hook contact model,

$$k_n = \frac{16}{15} \sqrt{R_{ij} Y_{ij}} \left( \frac{15 m_{ij} {V_{ch}}^2}{16 \sqrt{R_{ij}} Y_{ij}} \right)^{\frac{1}{5}} \quad , \quad \gamma_n = \sqrt{\frac{4 m_{ij} k_n}{1 + \left( \frac{\pi}{\ln(\epsilon)} \right)^2}} \geq 0 \tag{3.22}$$

$$\frac{1}{E_{ij}} = \frac{1 - {v_1}^2}{E_i} + \frac{1 - {v_2}^2}{E_j} \quad , \quad \frac{1}{R_{ij}} = \frac{1}{R_i} + \frac{1}{R_j} \quad , \quad \frac{1}{m_{ij}} = \frac{1}{m_i} + \frac{1}{m_j} \tag{3.23}$$

$$\psi = \frac{\ln(\epsilon)}{\sqrt{\ln^2(\epsilon) + \pi^2}} \quad , \quad S_n = 2 E_{ij} \sqrt{R_{ij} \delta_n} \tag{3.24}$$

where ($E_{ij}$) is the equivalent Young's Modulus, ($v$) is Poisson's Ratio, ($R_{ij}$) is the equivalent radius which represents the geometric mean diameter of particles i and j, ($m_{ij}$) is the equivalent mass, and ($v_{ch}$) is the characteristic impact velocity, ($\psi$) is the damping ratio coefficient, ($\varepsilon$) is the coefficient of restitution defined as $\varepsilon = g'/g$, where ($g$) is the absolute normal relative velocity before the collision and ($g'$) is the corresponding post-collision value, and ($S_n$) is the normal constant stiffness.

#### 3.4.2.3 Tangential Force

As shown in equation (3.16) the tangential force is calculated as:

$$F_{ij}^t = k_t \delta_t - \gamma_t (v_t . v_{ij}^t) \tag{3.25}$$

Frictional forces which resist the sliding (tangential) motion of particles are produced where two particles are in contact. A drawback of this model is that it is only valid when the normal force is constant, and no physical meaning can be attributed to the parameters ($k_t$) and ($\gamma_t$).

The spring part of the tangential force ($k_t\delta_t$) is called the 'shear history' which accounts for the tangential overlap of the particles in the duration of time they are in contact. In LIGGGHTS by default, the tangential history is not accounted for. In this case $\delta_t = 0$ and the total tangential force is limited by Coulomb frictional limit:

$$F_{ij}^t = \mu_s F_{ij}^n \qquad (\mathbf{3.26})$$

where ($\mu_s$) is the static friction coefficient. Here ($F_{ij}^t$) only causes a small relative movement termed 'microslip'. If the slip covers a larger area of the contact it is called "gross slip" and the tangential force follows Amonton's law of friction [191]:

$$F_{ij}^t = -\mu_s \left|F_{ij}^n\right| \hat{t}_{ij} \qquad (\mathbf{3.27})$$

If $\delta_t > 0$ and the tangential history effects are taken into consideration, in this case again Hertz and Hooke's theories are used to calculate the constant.

According to Hertz-Mindlin contact model,

$$k_t = 8G_{ij}\sqrt{R_{ij}\delta_n} \quad , \quad \gamma_t = -2\sqrt{\frac{5}{6}}\,\Psi\sqrt{S_t m_{ij}} \geq 0 \qquad (\mathbf{3.28})$$

where $G_{ij}$ is the equivalent Shear Modulus $\frac{1}{G_{ij}} = \frac{2-{v_1}^2}{G_i} + \frac{2-{v_2}^2}{G_j}$

According to Hook contact model, $k_t = k_n$ and $\gamma_t = \gamma_n$.

#### 3.4.2.4 Torque and Rolling Friction

The torque on *i* has two components, the torque due to tangential contact force ($M_{ij}^t$) and the torque due to rolling friction (normal contact) ($M_{ij}^r$).

$$\vec{M}_i = \vec{M}_{ij}^r + \vec{M}_{ij}^t \qquad (\mathbf{3.29})$$

The determination of the contribution of the normal component ($F_{ij}^n$) on rolling friction has actually been considered very difficult in DEM and is considered to be negligible and ignored in most DEM models [206]. However in LIGGGHTS they are calculated as [221]:

$$M_{ij}^t = \vec{R}_i \times \vec{F}_{ij}^t \qquad (\mathbf{3.30})$$

$$M_{ij}^r = \mu_r R_{ij} K_n \left|F_{ij}^n\right| \hat{\omega}_i \qquad (\mathbf{3.31})$$

where ($\mu_r$) is the coefficient of rolling friction and ($\hat{\omega}_i$) is the unit angular velocity of the particle *i*. In LIGGGHT, by default the normal torque is subtracted and does not contribute to the resulting torque. Only by setting a keyword, the full relative rotation contributes to the rolling friction torque. It is also important to mention here that the model in LIGGGHT only track the position and velocity of all the particles at each time step, but it does not track orientation.

An additional torque contribution which is possible to add in LIGGGHTS is the elastic-plastic spring-dashpot model (EPSD) because it dissipates kinetic energy, provides stable torques and dense particle packing. To model the rotational inertia and energy loss in rotating particles, the EPSD rotational model is applied. It consists of two components, a mechanical spring torque ($M_r^k$) and viscous damping torque ($M_r^d$) [192], [222]. The spring torque is dependent on the relative rotation between the two contacting particles and the total resistance model is described as

$$M_r = M_r^k + M_r^d \tag{3.32}$$

$$\Delta M_r^k = -k_r \Delta\theta_r \tag{3.33}$$

where $k_r = 2.25 k_n {\mu_r}^2 {R_{ij}}^2$ is the rolling stiffness, ($\Delta\theta_r$) is the incremental relative rotation between particles, and at the next time-step

$$\Delta M_{r,t+\Delta t}^k = M_{r,t}^k + \Delta M_r^k \quad , \quad \left|\Delta M_{r,t+\Delta t}^k\right| \le |M_r^m| \quad , \quad M_r^m = -\mu_r R_{ij} F_n \tag{3.34}$$

where the next rotational increment is limited by the limiting spring torque ($M_r^m$) which is described by the coefficient of rolling resistance ($\mu_r$), the effective radius ($R_{ij}$), and the normal contact force ($F_n$) as follows

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

$$M_{r,t+\Delta t}^d = \begin{cases} -C_r \dot{\theta}_r & if \ \left|\Delta M_{r,t+\Delta t}^k\right| < |M_r^m| \\ -f C_r \dot{\theta}_r & if \ \left|\Delta M_{r,t+\Delta t}^k\right| = |M_r^m| \end{cases} \tag{3.35}$$

#### 3.4.2.5 Cohesive Force

Phenomenologically, cohesion is the ability of particles to transmit shear stress without transmitting normal stress.Various non-contact forces are associated with cohesive particles, such as capillary forces which are formed due to liquid bridge between particles, and/or van der Waals forces and electrostatic forces which are associated with fine particles. The packing and the flow of the particles are significantly affected by these forces [206], [223]. In DEM, they can be applied as external forces in Newton's second law using Hertzian theory. LIGGGHTS includes cohesive models related to capillary forces and liquid bridges which are irrelevant here, but the Johnson-Kendall-Roberts (JKR) model [68] takes into account the surface energy at the contact as shown in Figure 3-15 and is a modification of Hertz theory. JKR is an elastic-adhesive normal contact model which correlates the contact area of two particles to the elastic material properties and the interfacial interaction strength. The cohesive force can be formed during the unloading cycle of contact as a force resisting separation. It calculates the cohesion force as an additional normal

force which is proportional to the contact area of overlap between the particles [221] and no modification to the tangential force is incorporated.

$$F = F_{n(elas)} + F_{n(damp)} + F_{t(fric)} + F_{t(damp)} + F_{n(coh)} \quad (\mathbf{3.36})$$

$$F_{n(coh)} = CA_{ij} \quad (\mathbf{3.37})$$

where $C$ is the particles cohesion energy density, and $A_{ij}$ is the contact area between particles $i$ and $j$. LIGGGHTS has an 'SJKR' model and a modified 'SJKR2' model which differ in the calculations of contact area [224]:

$$A_{ij(SJKR)} = \frac{\pi}{4} \times \frac{(c_{ij} - R_i - R_j)(c_{ij} + R_i - R_j)(c_{ij} - R_i + R_j)(c_{ij} + R_i + R_j)}{{c_{ij}}^2} \quad (\mathbf{3.38})$$

$$A_{ij(SJKR2)} = 2\pi\delta_n R_{ij} \quad (\mathbf{3.39})$$

where $(c_{ij})$ is the central distance and $(R_{ij})$ is the equivalent distance between *i-j*.

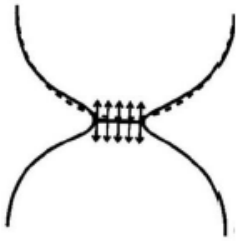


**Figure 3-15: JKR model representing the tensile force between particles in cohesive contact** [68]

The JKR model assumed that the attractive forces are restricted to the area of contact and are non-existent outside. This approximation is accurate for large cohesive energies and large particles with low Young's modulus and stiffness. However, it does not provide resistance in the tangential shearing direction. This limits the effect cohesion has on material flow because particles are allowed to slide past each other with little resistance. The model is very sensitive to the size of the particle [219].

#### 3.4.2.6 Adhesive Force Including Plasticity

The assumptions in DEM break down if the overlap becomes too large. The Hertzian contact model alongside the modified JKR cannot model stress history dependent stiffness and is purely elastic. There is no functionality available in LIGGGHTS to simulate deformable particles or plastic deformation. However, in general there has been some attempts to model plastic deformation using DEM. Although they are not implemented in LIGGGHTS, they are briefly discussed.

Thornton and Ning [225] were the first to introduce a contact model for elastic-perfectly plastic spheres with adhesion. Here a small tensile force is assumed to exist at the point of contact due to molecular attractions between the two particles. This

model proposes that above a certain velocity (the yield point) that a contact becomes plastic and the force-displacement relationship becomes linear (the plastic force-displacement loading curve is tangential to the Hertz curve at the yield point). As a result of this plastic deformation energy dissipation takes places leading to a different unloading path. The unloading curves is calculated from Hertz theory, but the contact radius has been modified to account for the flattening of the contact patch (plastic deformation). The adhesive force in the model is calculated based on the JKR theory with the inclusion of contact flattening [226], [227].

Later in 2008, Luding [228] proposed another elastic-plastic model for adhesive contacts which is a hysteretic model including plastic deformation and history dependent adhesion where

$$F_{n(adh)} = F_{hys} + \gamma_0 v_n \tag{3.40}$$

$$F_{hys} = \begin{cases} k_1\delta & if\ k_2(\delta - \delta_0) \geq k_1\delta \\ k_2(\delta - \delta_0) & if\ k_1\delta\ > k_2(\delta - \delta_0) > -k_c\delta \\ -k_c\delta & if\ -k_c\delta \geq k_2(\delta - \delta_0) \end{cases} \tag{3.41}$$

where ($\gamma_0$) is the viscous dissipation constant, ($k_1$) is the loading stiffness, ($k_2$) is the reloading stiffness, ($k_c$) is the adhesive stiffness/strength, where the value of ($k_2$) is dependent on a maximum overlap value, which depends on a dimensionless plasticity depth parameter ($\Phi_f$), defined relative to the contact radius. The model can revert to a simple linear spring model if $k_1 = k_2$.

However, both Thornton-Ning and Luding models only consider adhesion in the normal direction and failed to account for the torsion or bending strength that may exist between adhesive particles. Furthermore, “Luding’s model contains a shortcoming by which the behaviour of elastic-plastic and adhesive contacts is not realistically simulated. Contacts “break” at zero overlap regardless of loading or unloading history. This implies that plastic deformation has been ignored, which is unrealistic since plastic deformation is permanent and hence detachment must take place” [229]. Walton and Johnson model [230] attempted to solve this by introducing an extra model parameter for stiffness. Here, as the particles experience plastic deformation, there is a flattening of the contact area and there is an increase in the force required to separate the particles, and while Thornton and Ning’s model captures this, the additional resistance to rolling is not captured which means the particles continue rolling until more than one contact has been formed restraining the particle. As such Walton has proposed a contact model consisting of 4 inter-related modes of motion; normal, tangential, twisting and bending.

The different models presented here, both cohesive and adhesive are sensitive to the size of the particles. At the macroscopic scale, these models provide weak

interaction forces that can be numerically dominated by the friction properties of dry material. They will always yield a result but the they should be compared to the physical experiments and operational systems. The physical model parameters need to be scaled to simulate the mechanical behaviour of the material observed which will be explained in section 3.4.4.

### 3.4.3 DEM Bond Model

As mentioned earlier, DEM is capable of modelling and capturing phenomena related to granular materials. With the use of the different contact and cohesive contact models, DEM has been able to study heterogeneous materials such as concrete or rock, and homogeneous materials such as ceramics [192], [231]–[233]. However, the main difficulty for DEM remains to be properly simulating continuous material [123].

In order to similarly successfully model solid materials using DEM that show inherent elastic anisotropy, local anisotropy has to be considered to be able to achieve the correct physics of the damage phenomena [234] including metal, alloys and polymers. This involves including bonding of particles at their contact points [235]. By applying bond models in DEM the complex phenomena such as cracking and fracturing can be represented by the failures of the bonds [166].

It is important to mention here that there is a distinction between bonded particle contact models and adhesive/cohesive contact models in DEM. First, in bonded particles, the particles do not have to be in contact, yet still be connected via a bond between them. A bond is a contact between two particles that has a finite strength value (this can be a tensile, compressive or bending strength) and once this value is exceeded the bond will fail and the particles will no longer be bonded together. Contact can exist between the bonded-particles, but this is governed by the cohesion-less contact models such as Hertz-Mindlin or Linear spring as previously explained. Furthermore, all bonds are formed at an initial timestep but once a bond has broken it cannot reform. Adhesive contact models on the other hand can reform following breakage [226].

There are two main approaches to modelling bonded particles in DEM, the first uses the dual spring model, and the other uses the cohesive beam model [123]. The classical dual spring model has been properly established in literature, but the beam cohesive model has not. The bond model in LIGGGHTS follows the first approach by using the bond-particle model (BPM) developed by Potyondy and Cundall [233]. It was developed as a model for rock where the rock is represented by a dense packing of non-uniform-sized spherical particles. Here the asymmetrical contact

laws are replaced by cohesive interactions that can support tension to form bond between particles. The BPM model as implemented in LIGGGHTS is a combination of a normal and tangential spring-damper system that exist even if overlaps are smaller than zero and are represented by a pair of elastic springs with a constant normal and shear stiffness [236]:

$$F = F_{n(elas)} + F_{n(damp)} + F_{t(fric)} + F_{t(damp)} + F_{n(coh)} + F_{n(bond)} + F_{t(bond)}$$

The elastic force in normal direction is given as [236]

$$F_{n(bond)} = k_n A(r_0 - x_{ij}) \tag{3.42}$$

where $(r_0)$ is the initial particle distance upon bond creation, $(x_{ij})$ is the distance between the particle centres, $(k_n)$ the normal bond stiffness parameter and $(A)$ the beam area, dependent on the beam radius which is a user defined multiple of the minimum particle radius. The elastic force in tangential direction is incrementally defined as

$$F_{t(bond)} = F_{t(bond)-1} + k_t A(v_{ij} - v_{ij,n} n_{ij})\Delta t \tag{3.43}$$

where $(k_t)$ is the tangential bond stiffness parameter. Instead of using a traditional damping mechanism based on the velocity, a dissipative model is used. It reduces the elastic force in normal and tangential direction each time-step

$$F_{t(bond)} = F_{t(bond)} \left(1 - \frac{\Delta t}{\Delta t_d}\right) \Delta t \tag{3.44}$$

where $(\Delta t_d)$ is the dissipation time scale and it should be noted that $(r_0)$ is adapted in normal direction, to reflect this dissipation. The dissipative force is then given as

$$F_{t(damp)} = F_{t(bond)} \frac{\Delta t}{\Delta t_d} \tag{3.45}$$

Finally, care must be taken when contacts end, e.g. bonds are broken, as in that case the elastic potential is instantaneously converted into dissipated energy [236].

However, BMP is still not capable of describing a cohesive continuous material. Cohesive beam bond models [123] is a better way of describing them but it only describes perfectly elastic materials. Furthermore, many studies have attempted to develop new approaches to model continuous materials using DEM. For example, Chen et al. [166] performed tensile test simulations of high-carbon steel by DEM by introducing a new packing theory and using the particle-particle bond model. However, they were still only capable of revealing a relationship between the DEM parameters and the mechanical parameters using a parametric study.

### 3.4.4 DEM Micro-Macro Relations and Parametric Estimation

The predictions of DEM simulations are not only largely dependent on the number of particles, particle size, shape, deformation, time-step, etc. but also on many model related parameters that cannot be directly obtained, or are difficult to obtain from the material properties of the material or from experiments [237]–[239]. The challenge does not only involve the proper quantification and prediction of the properties and experimental validation, but the micro-macro transition is also a huge challenge which is the transition from the micro contact properties to the macro flow behaviours. This transition is crucial for understanding the collective flow behaviour of particles as a function of their contact properties [228]. The use of correct input parameters is crucial in order to ensure the predictivity of the simulations and confidently interpret the results. Simulations should also be validated against experimental results which was been proven to be a difficult task [214]. The developers of LIGGGHTS in [240] also stated that "In Discrete Element Method (DEM) simulations, particle–particle contact laws determine the macroscopic simulation results. Particle-based contact laws, in turn, commonly rely on semi-empirical parameters which are difficult to obtain by direct microscopic measurements"

Table 3-1 below lists the main input parameters in DEM models. Because these input parameters are difficult to obtain experimentally or are sometimes not physically defined, in literature they are often not measured, or values are assumed without proper justification since there is also no robust or standard method for their determination. Often it is also not mentioned whether their values have been measured or calibrated, and the final simulation is also often not validated [194].

This process where the required values of the input parameters for the simulation are determined is called parameter estimation or parametric fitting. This idea is well represented by Potyondy and Cundall who state [233]:

> *"For continuum models, the input properties (such as modulus and strength) can be derived directly from measurements performed on laboratory specimens. For the BPM*[1] *(. . . ) the input properties of the components usually are not known. (. . . ) For the general case of arbitrary packing of arbitrarily sized particles, the relation is found by means of a calibration process (. . . )".*

Estimation of the required input parameters is achieved through a calibration process, which are otherwise unknown. In DEM, this calibration process is achieved through a comparison between the bulk behaviour from a physical experiment and the simulation results. Low levels of parameters independency and high levels of

calibration accuracy can be achieved by performing at least two types of experiments and simulations multiple times. Together, these tests and simulations should help select the optimal set of input parameters [219].

Whilst DEM is increasingly used, the parameter estimation, calibration and validation methods are still difficult, inaccurate and not standardized. To give an example, the developers of LIGGGHTS in their paper in 2016 [240] attempted to develop an Artificial Neural Network for the identification of the DEM simulation parameters to link the microscopic numerical parameters to the macroscopic experimental results. This was done by developing a database by conducting a series of DEM simulations with varying simulation parameters and used them to train a feed-forward neural network to predict the macroscopic behaviour based on the parameters. They state that "For each set of calibration experiments, the neural network needs to be trained only once. After the training, the neural network provides a generic link between the macroscopic experimental results and the microscopic DEM simulation parameters. Based on these experiments, the DEM simulation parameters of any given non-cohesive granular material can be identified." Hence, the computationally expensive DEM simulations to perform parametric estimation and calibrations can be avoided. Even though this, and other techniques [193], [196], [241], are important step forward, they are not commonly used by the scientific community and are limited to specific cases (for example here non-cohesive granular materials). Furthermore, for these reasons, and other reasons, it is also very common to use FEM-DEM or CFD-DEM approaches to model the stress-strain behaviour of each individual particle. A review for different calibration methods in DEM is given here [194].

In MCA parametric fitting or calibration is not needed and the MCA parameters can be directly obtained from the material mechanical properties. This will be explained in the following section 3.5 and highlighted in 3.5.3. This is a major difference and advantage of MCA over DEM.

**Table 3-1: Input parameters for DEM models and their corresponding symbols**

| **Property** | **Symbol** |
|---|---|
| Particle radius | r |
| Contact radius | a |
| Elastic stiffness | $k_n$ |
| Frictional stiffness | $k_t$ |
| Rolling stiffness | $k_r$ |
| Adhesion stiffness | $k_c$ |
| Plastic stiffness | $k_{1,2}$ |

| | |
|---|---|
| Plasticity depth | $\Phi_f$ |
| Normal viscoelastic damping | $\gamma_n$ |
| Tangential viscoelastic damping | $\gamma_t$ |
| Rolling damping constant | $C_r$ |
| Coulomb friction coefficient | $\mu_s$ |
| Rolling friction coefficient | $\mu_r$ |
| Coefficient of restitution | $\varepsilon$ |
| Angle of repose | $\alpha$ |

## 3.5 Numerical Aspects of MCA

While DEM is mainly limited to the study of deformation and fracture of granular or brittle materials, MCA has been developed to study highly consolidated solids and has been proven to be feasible in simulating solid behaviour including viscoelastic and plastic deformation that occur in metals, alloys and polymers, as well as heterogenous materials such as composites and ceramics. The differences between DEM and MCA formulations which makes this possible are explained here.

The theoretical background of the MCA method is explained in detail in this section. As mentioned earlier in section 2.6.2.3, the MCA method was first introduced by Psakhie et al. in 1995 [133], as a simulation tool within the framework of mesomechanics. However, many developments of the method have been made ever since, and the latest description of the method can be found in [3]; where MCA is presented as a discrete approach (i.e. particle-based approach) to model the behaviour of materials on different scales and is used as a multi-scale modelling approach.

The MCA formulations presented in this section, are specifically related to the formulations for the 3D representation of the elastic-plastic MCA model. These are the formulations which will be implemented in LIGGGHTS, as presented later in Chapter 4. These formulations are developed by the authors of the MCA method including Sergey G. Psakhie, Valentin L. Popov, Evgeny V. Shilko, and the external supervisor on this thesis Alexey Smolin and could be found in the following papers [1]–[3].

It is noted that the terms ‘automata’ and ‘particles’ are interchangeably used hereon further.

### 3.5.1 MCA Particle Discretisation and Initial Configuration

In MCA it is assumed that any material consists of finite-sized elementary objects called automata. An automaton is the smallest unit in the simulated object. Each automaton is a sphere having a mass ($m_i$) and a size characterized by a diameter ($d_i$). Particles are allowed to deform, but initially they all have the same size.

The initial configuration of the particles follows an appropriate lattice structure, similar to MD. The automata could be arranged in a simple cubic (SC) form, or hexagonal arrangement (FCC or HCP), however the hexagonal fcc packing is usually used, as shown in Figure 3-16, because it has the highest density of particles and is the most stable in plastic deformation.

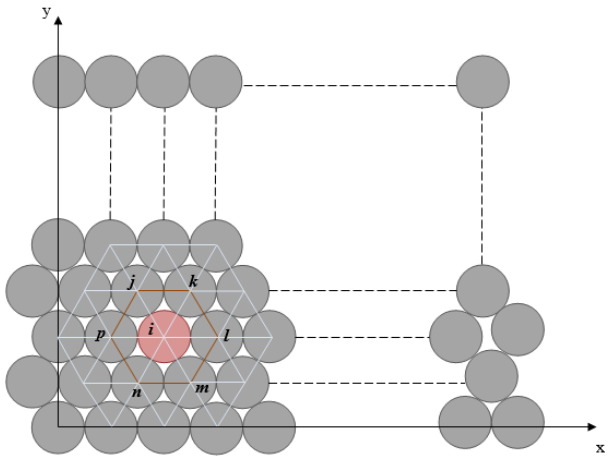



**Figure 3-16: Initial ideal hexagonal packing of automata in the linked state**

### 3.5.2 MCA Inter-Automata Interactions Model

#### 3.5.2.1 Contact of Automata

Each two neighbouring automata form an automata pair. The state of a pair can be of two types; linked (bonded) or unlinked (unbonded). Physically, a linked pair of automata must be in contact, while an unlinked pair can be in contact or non-interacting. Linked pairs can resist both compression and tension, while unlinked pairs only have a resistance to compression. If the simulated specimen is a consolidated solid, then all pairs are assumed to be initially linked (and in contact) which represents the presence of cohesive, adhesive or chemical bonds. The number of links each automaton can have with neighbouring automata depends on the coordination number; it can have a maximum of six links if it is in a (*sc*) lattice structure, or a maximum of 12 links for (*fcc*) or (*hcp*). Defects in the material such as pores/cracks can be included in the initial state by considering them unlinked.

At any given time, in order to determine the state of an automaton, the state and condition of each adjacent pair at the previous time-step are used. Automata also have the ability to change their neighbours by switching of a pair of automata from a linked state to an unlinked state or vice versa, as shown in Figure 3-17. Fracture, wear and damage of material corresponds to the unlinking of automata, while agglomeration, crack healing and micro-welding corresponds to linking of automata from different materials [242].

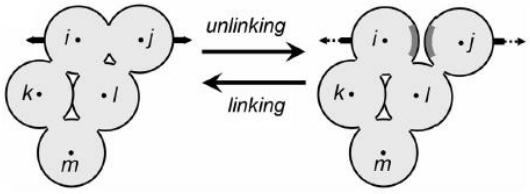



**Figure 3-17: Switching of state of *i-j* from linked (left) to unlinked (right) and vice-versa** [3]

In the initial undeformed state, the initial distance between the centre of automata *i* and the centre of automata *j* is:

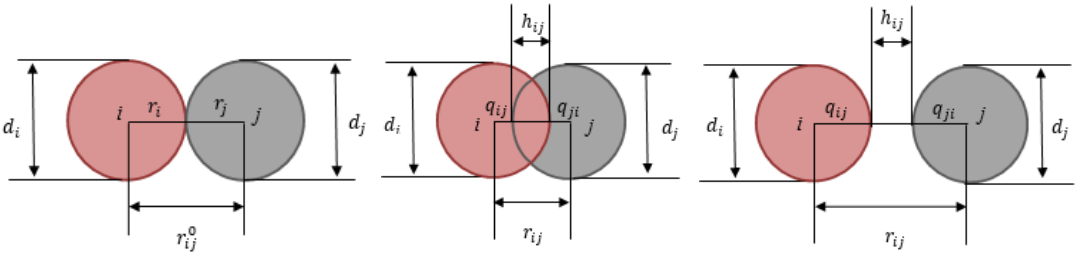
$$r_{ij}^{o} = \frac{d_i + d_j}{2} \tag{3.46}$$

A change in their position or deformation can causes either compression or tension between two neighbouring automata; which is determined by an overlapping parameter ($h_{ij}$) as shown in Figure 3-18, where

$$h_{ij} = r_{ij} - r_{ij}^{o} \tag{3.47}$$

where ($r_{ij}$) is the current distance between the centres of automata *i* and *j*.

A pair of automata *i-j* are linked and in contact when $r_{ij} \leq r_{ij}^{o}$, and linked pairs *i-j* can resist both tension ($h_{ij}$ is positive) and compression ($h_{ij}$ is negative), while unlinked automata only resist compression.




(a) (b) (c)

**Figure 3-18: Automata pair *i-j* in: (a) contact (b) compression state (c) tension state**

The overlapping parameter ($h_{ij}$) describes the relative position between two neighbouring automata. However, each of the automata *i* and *j* can have different material properties, therefore the contribution of each of them to the overlap ($h_{ij}$) may be different depending on their corresponding material properties.

($h_{ij}$) is presented by the distance between the centre of mass of the automaton and the central point of the plane of interaction which are also known as the contact points, which are ($q_{ij}$) and ($q_{ji}$), as shown in Figure 3-19(a):

$$r_{ij} = q_{ij} + q_{ji} \tag{3.48}$$

Of course, the initial values are $q_{ij} = d_i/2$ and $q_{ji} = d_j/2$. However, after deformation it is related to the normal strain of the pair *i-j* ($\varepsilon_{ij}$):

$$\varepsilon_{ij} \equiv q_{ij} - \frac{d_i}{2}/2 \tag{3.49}$$

where in terms of increments (Δ), which corresponds to the parameter during one time-step (Δ*t*), the overlapping parameter is

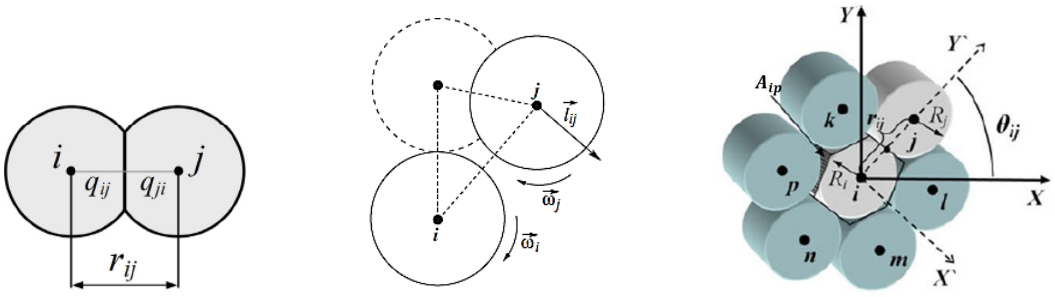
$$\Delta h_{ij} = \Delta q_{ij} + \Delta q_{ji} = (\Delta\varepsilon_{ij} + \Delta\varepsilon_{ij})\big(d_i + d_j\big)/2 \tag{3.50}$$

How to calculate this is explained in section 3.5.2.2. because it depends on the normal forces of interaction ($\vec{F}_{ij}^{n}$).

Interaction of pair of automata *i-j* also cause a shear displacement ($l_{ij}$) not just a normal displacement ($h_{ij}$), as shown in Figure 3-19(b). The shear displacement ($l_{ij}$) occurs at the point of their contact and is caused by the rotation of automata and tangential velocity, of course initially it is zero, however, after deformation it is related to the shear deformation ($\gamma_{ij}$) of the automata.

$$\Delta l_{ij} = \big(\Delta\gamma_{ij} + \Delta\gamma_{ji}\big)r_{ij} \tag{3.51}$$

How to calculate this is explained in section 3.5.2.3. because it depends on the tangential forces of interaction due to relative rotation between automata.




(a) (b) (c)

**Figure 3-19: (a) normal and (b) tangential interaction between automata pair *i* and *j*** [3] **(c) initial structure of automata in hexagonal structure, showing contact area ($A_{ip}$)** [155]

Another essential aspect of MCA is that even though initially the particles are spheres with a specific radius, no particle is ever a perfect sphere and after deformation, the size of the particle cannot be fully described by its radius or diameter. The real shape of an automaton is determined by the area of its contact with its neighbour ($A_{ij}$). Interaction between pair automata happens at plane faces

(face-face interaction), as shown in Figure 3-19(c), and the size of face is chosen so that it fills the voids between polygons. The contact area ($A_{ij}$) is defined in a way to minimize the volume of the voids between the particles in a simulated solid, and it changes during elastic and elastic-plastic deformation. This is essential in defining the interaction forces between automata and is a modification in MCA compared to DEM.

The initial contact area ($A_{ij}^0$) and volume ($\Omega_i^0$) of automata are defined and computed based on the radius and packing. It is assumed that for (*sc*) the automata have a cubic shape hence;

$$A_{ij}^0 = 4R^2 \qquad (\mathbf{3.52})$$

$$\Omega_i^0 = 8R^3 \qquad (\mathbf{3.53})$$

While for (*fcc*) the automata have rhombic dodecahedron shape hence

$$A_{ij}^0 = \sqrt{2}R^2 \qquad (\mathbf{3.54})$$

$$\Omega_i^0 = 4\sqrt{2}R^2 \qquad (\mathbf{3.55})$$

The contact area ($A_{ij}$) also changes during deformation (given in equation 3.55).

Under external load, automata can change their state, position and orientation of their neighbours based on the inter-automata interaction relationships and rules. The state of each automaton in the current step is defined by its state in the previous step as well as the effect of its neighbours. The movement of the automata depends on the interaction forces between the automata and their physical and mechanical properties. The switching of state of pair automata also depends on the interparticle forces. By studying the relative motion between automata and their behaviours, the motion and behaviour of the whole system can be simulated.

Since MCA is a particle-based method, the equations of motion are used to model the evolution of the system over time and calculate the forces acting on all the automata in each time step. It also simulates the motion of a group of particles by Newton's second law of motion. However, as mentioned earlier, one of the fundamental problems with some particle-based methods, including DEM, is the correct representation of the inter-particle interaction forces, which is the most sensitive and time consuming part of any particle-based simulation [172]. In DEM, an approximated pair-wise form is used which assumes that the total energy of the system is just the sum of the pair bonds, which has been proven to fail to describe the material on the macroscale and damage generation at scales lower than the size of the discrete element; this is also why DEM is often coupled with FEM/CFD.

Research showed that this problem can be solved by using a many-body interaction form which provides an isotropic behaviour of a simulated material to form an

accurate description of highly consolidated solids where elastic-plastic deformation occurs, rather than a granular medium. Moreover, stress tensor components can be computed for an automaton considering all forces in the calculation. This permits the realization of different models of the plastic behaviour of materials developed in continuum mechanics [1], [3], [172]. Furthermore, using the formalism of CA allows explicit description of processes such as damage generation and evolution, crack healing, chemical reactions, micro-welding, heat transfer, and phase transitions too. The construction of the force of interactions between particles in many-body approximation was made possible thanks to the use of the hybrid technique combining mathematical formalism of DEM and conventional concept of cellular automata and MD [243], [244] [172].

The authors of the MCA method applied the many-body interaction concept of the embedded atom method (EAM) [186], [245], which widely used in MD and previously discussed in section 3.3.2, to the MCA equations of motion. This allowed them to connect the average stresses and strains for the volume of each particle with the forces of interaction with its neighbouring particles. Meaning each automaton in the system follows the applied constitutive laws, leading to an accurate mechanical response of the whole system, and the capability of correct simulation of irreversible strain accumulation (plasticity) in ductile materials.

As mentioned earlier, in the EAM model the potential energy of an atom *i* depends on the pair interaction potential (φ) as a function of distance between atom *i* and *j*, and depends an electron charge density-dependent function ($\bar{\rho}_i$) which is the sum of the contributions of neighbour atom *j* to the local density of atom *i*, expressed as

$$E_i(\mathrm{R}) = \sum_{j \neq i} \varphi(r_{ij}) + \sum_{i} F(\bar{\rho}_i) \quad , \quad \bar{\rho}_i = \sum_{j \neq i} \rho_j(r_{ij}) \qquad (\mathbf{3.56})$$

By analogy, the MCA equations of motion are described as follows[158]:

$$\vec{F}_i = m_i \frac{d^2 \vec{r}_i}{dt^2} = \sum_{j=1, j \neq i}^{N_i} \vec{F}_{ij} + \vec{F}_i^{\Omega} \qquad (\mathbf{3.57})$$

$$\vec{M}_i = \hat{J}_i \frac{d\vec{\omega}_i}{dt} = \sum_{j=1, j \neq i}^{N_i} \vec{M}_{ij} \qquad (\mathbf{3.58})$$

where $\vec{F}_i$ is the total force acting on an automaton *i* from its surrounding neighbours $N_i$, and $\vec{M}_{ij}$ is the momentum of the force in pair *i*-*j*. It will be noted that this is similar to the equations of motion of DEM, but the total force consists of a pair-wise component ($\vec{F}_{ij}$) which depends on the displacement of *i* relative to its neighbour *j*, and the volume-dependent component ($\vec{F}_i^{\Omega}$) which depends on the combined effects of all the nearest surroundings of the automata *i* which realizes the many-body

interaction form. For locally isotropic media, the volume-dependent component ($\vec{F}_i^{\Omega}$) can be expressed as

$$\vec{F}_i^{\Omega} = A_i \sum_{j=1, j\neq i}^{N_i} \bar{\sigma}_{mean}^{j} A_{ij} \vec{n}_{ij} \qquad (\mathbf{3.59})$$

where ($A_i$) is a material parameter that represent the phase and chemical composition of the material, ($\bar{\sigma}_{mean}^{j}$) is the mean stress in the volume of automaton $j$, ($A_{ij}$) is the area of interaction between the automata.

The pair-wise component ($\vec{F}_{ij}$) could be expressed as the sum of the normal force ($\vec{F}_{ij}^{n}$) and a tangential force ($\vec{F}_{ij}^{t}$) which are characterized by their corresponding specific stress values ($\sigma_{ij}$ and $t_{ij}$) and the contact area ($A_{ij}$), shown in Figure 3-20,

$$\vec{F}_{ij} = \vec{F}_{ij}^{n} + \vec{F}_{ij}^{t} = (\sigma_{ij}\vec{n}_{ij} + t_{ij}\vec{\tau}_{ij})A_{ij} \qquad (\mathbf{3.60})$$

where ($\vec{n}_{ij}$) and ($\vec{\tau}_{ij}$) are the normal and tangential unit vectors from the centre of *i* to *j* respectively, as shown in Figure 3-20. The many-body contribution only affects the normal component of the total force and not the tangential component, which is taken into consideration in the specific force of normal interaction ($\sigma_{ij}$) which is explained in the next subsection. All formulas for forces are written in increments of specific values ($\sigma_{ij}$ and $t_{ij}$) per contact area of automata (traction values).

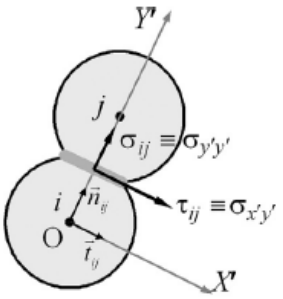



**Figure 3-20: The automata interaction, where X',Y' are the instantaneous coordinate system associated with the current spatial orientation of the contacting pair *i-j*** [155]

### 3.5.2.2 Normal Force

As shown in equation (3.50), the normal force of interaction is described as

$$\vec{F}_{ij}^{n} = (\sigma_{ij}\vec{n}_{ij})A_{ij} \qquad (\mathbf{3.61})$$

where $\vec{n}_{ij}$ is the unit vector in the normal direction, $\vec{n}_{ij} = (\vec{R}_j - \vec{R}_i)/r_{ij}$.

The normal interaction force is directly related to the strain distribution and it is assumed that the response of a linearly elastic isotropic material under the stress state is described using the formula for the diagonal terms of the stress tensor in Hooke's law. Thus, the increment of the specific force of the normal interaction of the automata *i* and *j* in a time step is described as

$$\Delta\sigma_{ij} = 2G_i\Delta\varepsilon_{ij} + \left(1 - \frac{2G_i}{K_i}\right)\Delta\bar{\sigma}^i_{mean} \quad (\mathbf{3.62})$$

where ($G_i$) is the shear modulus, ($K_i$) is the bulk modulus of i, ($\varepsilon_{ij}$) is the normal strain, and ($\bar{\sigma}^i_{mean}$) is the mean stress of i which is the term expressing the many-body particle contribution to the total force on automaton *i*.

At a next time-step, the normal force of interaction is expressed as:

$$\vec{F}^{n+1}_{ij} = (\sigma^{n+1}_{ij}\vec{n}_{ij})A^{n+1}_{ij} \quad (\mathbf{3.63})$$

$$\sigma^{n+1}_{ij} = \sigma_{ij} + \Delta\sigma_{ij} = \sigma_{ij} + 2G_i\Delta\varepsilon^{n+1}_{ij} + (1 - 2G_i/K_i)\Delta\sigma^{n+1}_i \quad (\mathbf{3.64})$$

where ($A^{n+1}_{ij}$) is the contact area at the next time-step which could be deformed

$$A^{n+1}_{ij} = A^0_{ij}\frac{d}{r^{n+1}_{ij}}\left[1 + \frac{1}{2}\left(\frac{\bar{\sigma}^i_{mean}}{K_i} + \frac{\bar{\sigma}^j_{mean}}{K_j}\right)\right] \quad (\mathbf{3.65})$$

where $r^{n+1}_{ij}$ is the current distance between i and j. Now to calculate the ($\sigma_{ij}$) we need $\varepsilon_{ij}$ and $\bar{\sigma}^i_{mean}$.

To calculate the value of the central strain in the pair *i-j* at the next time step *n*+1, the equality of the forces acting on each particle according to Newton's third law is considered:

$$\sigma^n_{ij} + 2G_i\Delta\varepsilon^{n+1}_{ij} + \left(1 - \frac{2G_i}{3K_i}\right)\Delta\bar{\sigma}^{n+1}_i = \sigma^n_{ji} + 2G_j\Delta\varepsilon^{n+1}_{ji} + \left(1 - \frac{2G_j}{3K_j}\right)\Delta\bar{\sigma}^{n+1}_j \quad (\mathbf{3.66})$$

And since

$$\Delta r^{n+1}_{ij} = R\Delta\varepsilon^{n+1}_{ij} + R\Delta\varepsilon^{n+1}_{ji} = \left|\vec{R}^{n+1}_i - \vec{R}^{n+1}_j\right| - \left|\vec{R}^n_i - \vec{R}^n_j\right| \quad (\mathbf{3.67})$$

Then the central strain can be calculated as

$$\Delta\varepsilon^{n+1}_{ij} = \frac{p^n_{ji} - p^n_{ij} + 2G_i\frac{\Delta r^{n+1}_{ij}}{R} + \Delta\bar{\sigma}^{n+1}_j\left(1 - \frac{2G_j}{3K_j}\right) - \Delta\bar{\sigma}^{n+1}_i\left(1 - \frac{2G_i}{3K_i}\right)}{2G_i + 2G_j} \quad (\mathbf{3.68})$$

Calculating the mean stress ($\bar{\sigma}^i_{mean}$) is the most critical step. ($\bar{\sigma}^i_{mean}$) is the local value of pressure of *i* and it is the same as the mean stress ($\bar{\sigma}^i_{mean}$) in the automaton volume. By definition, the mean stress (the same pressure with the opposite sign) is the first invariant of the stress tensor divided by three:

$$\bar{\sigma} = -\frac{\bar{\sigma}_{xx} + \bar{\sigma}_{yy} + \bar{\sigma}_{zz}}{3} \quad (\mathbf{3.69})$$

To calculate the components of the stress tensor in the volume of an automaton, the average stress tensors are calculated using homogenization described in [1] :

$$\sigma^{\alpha\beta}_i = -\frac{1}{\Omega_i}\sum_{j=1}^{N_i} q^\alpha_{ij} f^\beta_{ij} A_{ij} \quad (\mathbf{3.70})$$

where $\alpha, \beta = X, Y, Z$ coordinates, $(f_{ij}^{\beta})$ is the projection of the total (the sum of the central and tangential) specific force at the contact site between the automata *i* and *j* on the axis $\beta$. Thus, for the diagonal terms of the stress tensor we only have a contribution from the central forces, hence

$$\sigma_i^{\alpha\alpha} = -\frac{1}{\Omega_i}\sum_{j=1}^{N_i} q_{ij}^{\alpha} f_{ij}^{\alpha} A_{ij} \qquad (\mathbf{3.71})$$

The modelling practice has shown that the components of the stress tensor are better calculated in the initial configuration (undeformed state), which is determined by the initial packing. As explained before, the number of neighbours is equal to the coordination number $(N_c)$, and the values of $(q_{ij})$ and $(A_{ij})$ are the same for all neighbors. The volume of the automata consists of pyramids with the base $(A_{ij})$ and height $(q_{ij})$, hence $q_{ij}A_{ij}N_c = 3\Omega_i$ and

$$\sigma_i^{\alpha\alpha} = \frac{3}{N_c}\sum_{j=1}^{N_i} \sigma_{ij} \qquad (\mathbf{3.72})$$

Hence,

$$\Delta\bar{\sigma}_{mean}^{i} = \frac{1}{N_c}\sum_{j=1}^{N_i} \sigma_{ij}^{n+1} \qquad (\mathbf{3.73})$$

However, to calculate the increment of the normal force $(\sigma_{ij}^{n+1})$ we need to know the increment of the mean stress $(\Delta\sigma_i^{n+1})$ as shown above. Hence, it is necessary to estimate the increment of the mean stress at the current step $(\Delta\sigma_i^{n+1})$ which could be done by an iterative procedure or by calculating a predictor estimate value, which are used for calculating strain increments and then finally correcting the estimation for mean stress. It has been shown that the iterative procedure works well in cases with high Poisson's ratio ($\sim 0.49$), however for usual materials the predictor estimation method works much better.

The calculation of the predictor estimation of the mean stress $(\bar{\sigma}_{mean}^{i})$ depends on the state of automata pair *i-j* and their neighbours. We will use the symbol of the mean stress as $(\bar{\sigma}_i)$ instead of $(\bar{\sigma}_{mean}^{i})$ for convenience of writing the equations.

In the case of an isolated pair of automata *i-j*, when other neighbours (*N*) of both automata are absent, according to equation (3.63)

$$\Delta\bar{\sigma}_i^1 = \Delta\bar{\sigma}_j^1 = \frac{\Delta\sigma_{ij}^1}{N_c} \qquad (\mathbf{3.74})$$

where the (1) represents the case of an isolated pair of automata *i-j*. According to equation (3.52), the specific force of normal interaction is

$$\Delta\sigma_{ij}^{1} = 2G_i\Delta\varepsilon_{ij}^{n+1} + \left(1 - \frac{2G_i}{K_i}\right)\Delta\bar{\sigma}_i^{1} \qquad (\mathbf{3.75})$$

Then using equation (3.64), the specific force of normal interaction using the predictor estimator for the case of isolated pair of automata *i-j* will be:

$$\Delta\sigma_{ij}^{1} = \frac{2G_i}{1 - \left(1 - \frac{2G_i}{3K_i}\right)\frac{1}{N_c}}\Delta\varepsilon_{ij}^{n+1} \qquad (\mathbf{3.76})$$

$$\Delta\sigma_{ij}^{1} = \frac{2G_iN_c}{N_c - \left(1 - \frac{2G_i}{3K_i}\right)}\Delta\varepsilon_{ij}^{n+1} \qquad (\mathbf{3.77})$$

$$\Delta\sigma_{ij}^{1} = H_{\varepsilon}^{1}\Delta\varepsilon_{ij}^{n+1} \qquad (\mathbf{3.78})$$

where ($H_{\varepsilon}^{1}$) is the "stiffness" of an automaton with one neighbour where

$$H_{\varepsilon}^{1} = \frac{2G_iN_c}{N_c - \left(1 - \frac{2G_i}{3K_i}\right)} \qquad (\mathbf{3.79})$$

The second case takes into consideration the complete environment of each automaton pair *i-j* by considering the neighbours (*N*) that do not shift relative to the automata of the pair *i-j*. This provides the exact expression for the average deformation of the automata of a given pair *i-j*:

$$\Delta\bar{\varepsilon}_i^{c} = \Delta\bar{\varepsilon}_j^{c} = \frac{\Delta\varepsilon_{ij}^{n+1}}{N_c} \qquad (\mathbf{3.80})$$

$$\Delta\bar{\sigma}_i^{c} = 3K_i\frac{\Delta\varepsilon_{ij}^{n+1}}{N_c} \qquad (\mathbf{3.81})$$

Then the corresponding value of the specific force of normal interaction will be

$$\Delta\sigma_{ij}^{c} = \frac{N_c - \left(1 - \frac{3K_i}{2G_i}\right)}{N_c}2G_i\Delta\varepsilon_{ij}^{n+1} \qquad (\mathbf{3.82})$$

$$\Delta\sigma_{ij}^{c} = H_{\varepsilon}^{c}\Delta\varepsilon_{ij}^{n+1} \qquad (\mathbf{3.83})$$

where ($H_{\varepsilon}^{c}$) is the "deformation stiffness" of an automaton with a complete environment of neighbours, where

$$H_{\varepsilon}^{c} = \frac{N_c - \left(1 - \frac{3K_i}{2G_i}\right)}{N_c}2G_i \qquad (\mathbf{3.84})$$

As a predictor for the new value of the increment of the real specific normal force ($\Delta\sigma_{ij}^{p}$), interpolation to the real number of neighbors *N* is used:

$$\Delta\sigma_{ij}^{p} = \left[H_{\varepsilon}^{1} + (H_{\varepsilon}^{c} - H_{\varepsilon}^{1})\frac{N-1}{N_c - 1}\right]\Delta\varepsilon_{ij}^{n+1} \qquad (\mathbf{3.85})$$

$$\Delta\sigma_{ij}^{p} = \left[H_{\varepsilon}^{1}\left(1 - \frac{N-1}{N_c - 1}\right) + H_{\varepsilon}^{c}\frac{N-1}{N_c - 1}\right] \quad (\mathbf{3.86})$$

$$\Delta\sigma_{ij}^{p} = H_{\varepsilon}^{N}\Delta\varepsilon_{ij}^{n+1} \quad (\mathbf{3.87})$$

where $N$ is the number of neighbours and $H_{\varepsilon}^{N}$ is the "total deformation stiffness".

So the predictive normal specific force value is expressed as

$$\sigma_{ij}^{p} = \sigma_{ij}^{n} + H_{\varepsilon}^{N}\Delta\varepsilon_{ij}^{n+1} \quad (\mathbf{3.89})$$

Now using equations (3.69) and (3.58) the estimate for the increment of normal strain can be calculated as:

$$\Delta\varepsilon_{ij}^{p} = \frac{p_{ji}^{n} - p_{ij}^{n} + H_{\varepsilon}^{N}\dfrac{\Delta r_{ij}^{n+1}}{R}}{H_{\varepsilon,i}^{N} + H_{\varepsilon,j}^{N}} \quad (\mathbf{3.90})$$

Then using equation (3.68), the estimated predictor value of the mean stress is

$$\Delta\bar{\sigma}_{i}^{p} = 3K_i\Delta\varepsilon_{ij}^{p} = 3K_i\frac{\sum\Delta\varepsilon_{ij}^{p}}{N_c} \quad (\mathbf{3.91})$$

However, to use this equation (3.74), it is also necessary to estimate the change in the distance to the free surface of the automaton ($\Delta q_i^{\text{free}}$) which is the distance to the point of contact with the new neighbor, and the corresponding change in the deformation ($\Delta\varepsilon_{ij}^{free}$**)**. To do this, the usual boundary condition is used on the free surface of a solid as the equality of normal stresses to the pressure of the external environment ($\Delta P_{ext}$) as follows:

$$\Delta P_{ext} = 2G_i\Delta\varepsilon_{ij}^{free} + (3K_i - 2G_i)\left[\frac{\sum\Delta\varepsilon_{ij}^{n+1}}{N_c} + \frac{N_c - N}{N_c}\Delta\varepsilon_{ij}^{free}\right] \quad (\mathbf{3.92})$$

where

$$\Delta\varepsilon_{ij}^{free} = \frac{\left(1 - \dfrac{3K_i}{2G_i}\right)\dfrac{\sum\Delta\varepsilon_{ij}^{n+1}}{N_c} + \dfrac{\Delta P_{ext}}{2G_i}}{1 - \left(1 - \dfrac{3K_i}{2G_i}\right)\dfrac{N_c - N}{N_c}} \quad (\mathbf{3.93})$$

Then according to equation (3.67), the total predictor increment of the normal strain is

$$\Delta\bar{\varepsilon}_{i}^{p} = \frac{\sum_{j=1}^{N_i}\Delta\varepsilon_{ij}^{p}}{N_c - (N_c - N)\left(1 - \dfrac{3K_i}{2G_i}\right)} + \frac{(N_c - N)\dfrac{3K_i}{2G_i}}{N + (N_c - N)\dfrac{3K_i}{2G_i}}\frac{\Delta P_{ext}}{3K_i} \quad (\mathbf{3.94})$$

Then the final predictor increment of the mean stress can be calculated using equation (3.74) as

$$\Delta\bar{\sigma}_{i}^{p} = 3K_i\Delta\varepsilon_{ij}^{p} \quad (\mathbf{3.95})$$

Substituting it into equation (3.58), we obtain the increments of the normal deformation, and then calculate the values of the central forces using equation (3.54)

$$\sigma_{ij}^{n+1} = \sigma_{ij}^{n} + 2G_i\Delta\varepsilon_{ij}^{n+1} + (1 - 2G_i/K_i)\Delta\bar{\sigma}_i^{p} \quad (\mathbf{3.96})$$

After the first time-step and getting the predictor, the usual formula for calculating the mean stress (3.63)

$$\bar{\sigma}_i = \bar{\sigma}_{mean}^{i} = \frac{1}{N_c}\sum_{j=1}^{N_i}\sigma_{ij}^{n+1} \quad (\mathbf{3.97})$$

and then according to equation (3.50) the vector value of the force can be calculated

$$\vec{\sigma}_{ij}^{n+1} = \sigma_{ij}^{n+1} \cdot A_{ij}^{n+1} \cdot \vec{n}_{ij} \quad (\mathbf{3.98})$$

### 3.5.2.3 Tangential Force

As explained in section 3.5.2.1 and shown in equation (3.50), the tangential force of interaction, is defined by the shear force ($t_{ij}$) or the force of shear strain resistance:

$$\vec{F}_{ij}^{t} = (t_{ij}\vec{\tau}_{ij})A_{ij} \quad (\mathbf{3.99})$$

It lies in the plane normal to the axis connecting the centres of the automata pair *i-j*, as shown in Figure 3-20(b). The direction in this plane is determined by the direction of shear deformation ($\gamma_{ij}$). Since the orientation of both the axis of the pair *i-j* and the shear force varies at all times, it is described by the specific value in relation to the interaction area ($A_{ij}$) and the direction of the axis of the pair ($\vec{\tau}_{ij}$). It can also be written as

$$\vec{F}_{ij}^{t} = \left(\vec{n}_{ij} \times \vec{t}_{ij}\right)A_{ij} \quad (\mathbf{3.100})$$

where the specific shear force ($\vec{t}_{ij}$) is in a vector form and multiplied by a cross product to the normal unit vector ($\vec{n}_{ij}$).

Similar to equation (3.52), the specific force of the tangential interaction of the automata *i* and *j* can be described using the formula for the non-diagonal terms of the stress tensor in Hooke's law:

$$\Delta\vec{t}_{ij} = 2G_i\Delta\vec{\gamma}_{ij} \quad (\mathbf{3.101})$$

where ($\Delta\vec{\gamma}_{ij}$) is the increment of the shear deformation in the pair *i-j*.

At the next time-step,

$$\Delta\vec{t}_{ij}^{n+1} = \vec{t}_{ij}^{n} + 2G_i\Delta\vec{\gamma}_{ij}^{n+1} \quad (\mathbf{3.102})$$

$$\vec{\gamma}_{ij}^{n+1} = \vec{\gamma}_{ij}^{n} + \Delta\vec{\gamma}_{ij}^{n+1} \quad (\mathbf{3.103})$$

Accordingly, the basic equation for calculating the shear force at step *n*+1, similar to equation (3.56), Newton's third law for shear forces in a pair of automata *i-j* is used

$$\vec{t}_{ij}^{n} + 2G_i\Delta\vec{\gamma}_{ij}^{n+1} = \vec{t}_{ji}^{n} + 2G_j\Delta\vec{\gamma}_{ji}^{n+1} \quad (\mathbf{3.104})$$

As shown earlier in equation (3.46) the tangential displacement of automata ($\Delta\vec{l}_{ij}^{n+1}$) occurs at the point of their contact which causes shear in each automaton where

$$\Delta\vec{l}_{ij}^{n+1} = q_{ij}\Delta\vec{\gamma}_{ij}^{n+1} + q_{ji}\Delta\vec{\gamma}_{ji}^{n+1} \quad (\mathbf{3.105})$$

Solving equations (3.85) and (3.86), the increment of shear strain at the current step

$$\Delta\vec{\gamma}_{ij}^{n+1} = \frac{1}{q_{ij}G_i + q_{ji}G_j}\left[G_j\Delta\vec{l}_{ij}^{n+1} + \frac{q_{ij}}{2}\left(\vec{t}_{ij}^{n} - \vec{t}_{ji}^{n}\right)\right] \quad (\mathbf{3.106})$$

To use equation (3.87) it is necessary to calculate the tangential displacement ($\Delta\vec{l}_{ij}^{n+1}$). Since each automaton rotates with its own angular velocity as shown in Figure 3-21, where ($\omega_i$) and ($\omega_j$) is the angular velocities of i and j respectively:

$$\vec{v}_j - \vec{v}_i = \vec{\omega}_{ij} \times \vec{r}_{ij} \quad (\mathbf{3.107})$$

where $\vec{r}_{ij} = (\vec{R}_j - \vec{R}_i)$, $\vec{v}_i = d\vec{R}_i/dt$ , ($v_i$) and ($v_j$) are the translational velocities of i and j respectively, and ($\omega_{ij}$) is the angular velocity of the pair *i-j*, where

$$\omega_{ij} = \frac{\vec{n}_{ij} \times (v_j - v_i)}{r_{ij}} \quad (\mathbf{3.108})$$

the difference between the rotations is responsible for the shear strain:

$$\Delta\vec{l}_{ij} = \frac{(q_{ij}(\vec{\omega}_{ij} - \vec{\omega}_i) \times \vec{n}_{ij} + q_{ji}(\vec{\omega}_{ij} - \vec{\omega}_j) \times \vec{n}_{ij})\Delta t}{r_{ij}} \quad (\mathbf{3.109})$$

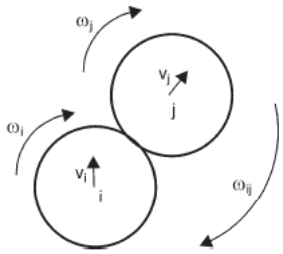



**Figure 3-21: Rotation of a pair of automata *i-j*** [246]

### 3.5.2.4 Torque and Rolling Friction

In 3D representation, rotation also leads to bending and torsion (twisting) strains in the pair *i-j* due to the difference in the automata rotation. The resistance to relative rotation in the automata pair *i-j* causes torque ($\vec{K}_{ij}$). So, the moment of the total force as described in equation (3.48) is given as:

$$\vec{M}_{ij} = q_{ij}\left(\vec{n}_{ij} \times \vec{F}_{ij}\right) + \vec{K}_{ij} \quad (\mathbf{3.110})$$

$$\vec{K}_{ij} = -(G_i + G_j)(\vec{\theta}_j - \vec{\theta}_i) \quad (\mathbf{3.111})$$

where ($\vec{\theta}_i$) and ($\vec{\theta}_j$) are the rotation angles of i and j. At the next time-step,

$$\vec{M}_{ij}^{n+1} = \vec{M}_{ij} + \Delta\vec{M}_{ij}^{n+1} \quad (\mathbf{3.112})$$

Furthermore, a force of dry friction ($F_{ij}^{df}$) can be described between contacting automata. The normal and tangential forces have to be calculated first, but the

tangential force is the one that determined whether there will be a dry friction force or not. The friction force is determined as

$$F_{ij}^{df} = -\mu_{ij} \frac{\sigma_{ij} + \sigma_{ji}}{2} \qquad (\mathbf{3.113})$$

where $\mu_{ij}$ is the coefficient of dry friction for the automata pair *i-j*, and ($\sigma_{ij}$) is the specific force of the normal interaction.

If $F_{ij}^{df} < \left|\vec{t}_{ij}\right|$ where ($\vec{t}_{ij}$) is the specific tangential force, then tangential force is corrected by including the dry friction as follows

$$\vec{t}_{ij}^{df} = \frac{F_{ij}^{df}}{\left|\vec{t}_{ij}\right|} \vec{t}_{ij} \qquad (\mathbf{3.114})$$

#### 3.5.2.5 Elastic-Plastic Forces

All the previous equations describe the mechanical behaviour of linearly elastic materials, and for example in [247] it was shown that these models gave the same results as those who have used usual continuum mechanics equations by finite-difference methods. In [246] it is shown that taking into account the rotation of the particles helps correctly describing the isotropic response of the material.

This proposed approach of building many-body forces of interaction has a great advantage of realizing various different models of elasticity and plasticity within the framework of particle-based methods. Knowing the stress and strain tensor in the bulk of an automaton makes it possible to directly apply conventional fracture criteria written in the tensor form.

Different models can be used for describing elastic-plastic behaviour [1], however in most MCA studies and in this thesis the theory of plastic flow is described by using the von Mises criterion for plasticity and Wilkins algorithm. Von Mises criterion is part of plasticity theory that is mostly used for ductile materials, such as metals. It assumes that yielding of a ductile material starts when the second deviatoric stress invariant reaches a critical value. The von Mises stress or equivalent stress ($\bar{\sigma}_{eq}^{i}$) is a value used to determine if a given material will yield. The von Mises yield criterion states that a material under load will start yielding if the von Mises stress is equal or greater than the elastic yield limit (yield stress $\sigma_y$), as shown in Figure 3-22 [248]. Prior to that the material response is assumed to be elastic or viscoelastic.

This equivalent stress ($\bar{\sigma}_{eq}^{i}$) of a material can be calculated in terms of the stress tensor components as [1]–[3]:

$$\bar{\sigma}_{eq}^{i} = \frac{1}{\sqrt{2}} \sqrt{\left(\bar{\sigma}_{xx}^{i} - \bar{\sigma}_{yy}^{i}\right)^2 + \left(\bar{\sigma}_{yy}^{i} - \bar{\sigma}_{zz}^{i}\right)^2 + \left(\bar{\sigma}_{zz}^{i} - \bar{\sigma}_{xx}^{i}\right)^2 + 6\left[\left(\bar{\sigma}_{xy}^{i}\right)^2 + \left(\bar{\sigma}_{yz}^{i}\right)^2 + \left(\bar{\sigma}_{xz}^{i}\right)^2\right]}$$

where the stress tensor components can be calculated from equations (3.59) and (3.61).

To use the von Mises criterion in particle-based methods, the well-known radial return algorithm of Wilkins can be used for integrating the plasticity equations for isotropic von Mises plasticity [249], [250]. In Wilkins algorithm, first the stress is updated assuming that the response is elastic, then if it is outside the yield surface, the stress is projected or "dropped" to the closest point of the yield surface ($\sigma_{pl}$), as shown in Figure 3-22. If the material is perfectly plastic, the yield surface is constant, but if the yield surface expands during plastic flow, the stress is projected on the expanded yield surface. Hence, if the equivalent stress ($\bar{\sigma}^i_{eq}$) exceeds the plastic stress ($\sigma_{pl}$) which is the current radius of von Misses yield circle, then the elastic problem is corrected by subsequent "drop" of components of stress deviator tensor. More details could be found in [172].
The Wilkins algorithm is formulated in terms of stress deviator tensor as

$$D'_{\alpha\beta} = D_{\alpha\beta} M \qquad (\mathbf{3.115})$$

where ($D_{\alpha\beta}$) is the stress deviator after the elastic solution at the current time step, ($D'_{\alpha\beta}$) is the corrected stress deviator and $M = \sigma_{pl}/\sigma_{eq}$ represents the "drop" where ($\sigma_{pl}$) is the current radius of the von Mises yield circle.

where

$$\sigma_{pl} = \sigma_y + \left(\bar{\varepsilon}^i_{eq} - \frac{\sigma_y}{3G}\right) E_h \qquad (\mathbf{3.116})$$

where $\sigma_y$ is the yield stress of the material, $E_h$ is the plastic work/strain hardening modulus of the material.

The stress deviator tensor can be written in terms of stress as follows for automata *i*:

$$\left(\bar{\sigma}^i_{\alpha\beta}\right)' = \bar{\sigma}^i_{\alpha\beta} M_i \qquad (\mathbf{3.117})$$

$$(\bar{\sigma}^i_{\alpha\alpha})' = (\bar{\sigma}^i_{\alpha\alpha} - \bar{\sigma}^i_{mean}) M_i + \bar{\sigma}_{mean} \qquad (\mathbf{3.118})$$

where $(\bar{\sigma}^i_{\alpha\alpha})'$ and $\left(\bar{\sigma}^i_{\alpha\beta}\right)'$ are the corrected average stress tensors, ($\bar{\sigma}^i_{\alpha\alpha}$) and ($\bar{\sigma}^i_{\alpha\beta}$) are the elastic stress tensor components at the current time step.

Thus, to calculate the plastic forces, the elastic normal and tangential specific forces can be corrected with the use of the current value of ($M_i$), such that the corrected values of normal and tangential specific forces for plasticity will be:

$$\begin{cases} \sigma'_{ij} = (\sigma_{ij} - \bar{\sigma}^i_{mean}) M_i + \bar{\sigma}^i_{mean} \\ \tau'_{ij} = \tau_{ij} M_i \end{cases} \qquad (\mathbf{3.119})$$

where ($\sigma'_{ij}$) and ($\tau'_{ij}$) are the corrected specific forces for plasticity, and then corrected total normal and tangential forces can be calculated according to equations (3.50), (3.51), (3.82) and (3.92).

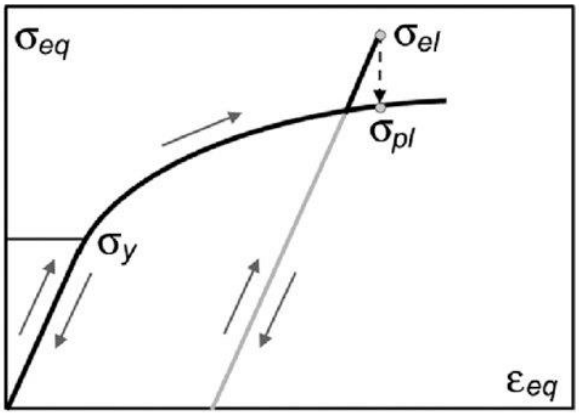



**Figure 3-22: Schematic of functioning of radial return algorithm of Wilkins according to the von Mises criterion** [144]

When modelling a material using MCA, the rheological properties of the automata are defined by assigning a constitutive relation called the mechanical response function of the automata which will be explained further in section 3.5.4. The constitutive relation can be expressed as

$$\bar{\sigma}^i_{eq} = \Phi \bar{\varepsilon}^i_{eq} \tag{3.120}$$

where ($\bar{\varepsilon}^i_{eq}$) is the equivalent strain and can be calculated similarly to the equivalent stress ($\bar{\sigma}^i_{eq}$) as follows:

$$\bar{\varepsilon}^i_{eq} = \frac{\sqrt{2}}{3}\sqrt{\left(\bar{\varepsilon}^i_{xx} - \bar{\varepsilon}^i_{yy}\right)^2 + \left(\bar{\varepsilon}^i_{yy} - \bar{\varepsilon}^i_{zz}\right)^2 + \left(\bar{\varepsilon}^i_{zz} - \bar{\varepsilon}^i_{xx}\right)^2 + 6\left[\left(\bar{\varepsilon}^i_{xy}\right)^2 + \left(\bar{\varepsilon}^i_{yz}\right)^2 + \left(\bar{\varepsilon}^i_{xz}\right)^2\right]} \tag{3.121}$$

And the stress deviator tensor is calculated based on the stress deviatoric tensor as expressed in equations (3.98) and (3.99), where

$$\bar{\varepsilon}^i_{\alpha\alpha} = \frac{\bar{\sigma}^i_{\alpha\alpha}}{2G_i} + \frac{2G_i - K_i}{2G_i K_i}\Delta\bar{\sigma}^i_{mean} \tag{3.122}$$

$$\bar{\varepsilon}^i_{\alpha\beta} = \frac{\bar{\sigma}^i_{\alpha\beta}}{2G_i} \tag{3.123}$$

$$\varepsilon^{ij}_{\alpha\beta} = \frac{\bar{\varepsilon}^i_{\alpha\beta} q_{ji} + \bar{\varepsilon}^i_{\alpha\beta} q_{ij}}{r_{ij}} \tag{3.124}$$

This formulation corresponds to rate-independent plasticity.

The value of the equivalent strain ($\varepsilon_{eq}$) is calculated using the increment of equivalent stress ($\bar{\sigma}^i_{eq}$):

$$(\bar{\varepsilon}^{i}_{eq})_{n+1} = (\bar{\varepsilon}^{i}_{eq})_{n} + \frac{(\bar{\sigma}^{i}_{eq})_{n+1} - (\bar{\sigma}^{i}_{eq})^{final}_{n}}{3G_i} \qquad (\mathbf{3.125})$$

where $(\bar{\sigma}^{i}_{eq})_{n+1}$ is the stress intensity after solving the elastic problem at time step (n+1) and $(\bar{\sigma}^{i}_{eq})^{final}_{n}$ is the equivalent stress at the end of the previous time step.

### 3.5.3 MCA Bond Model and Switching of State of Pair Automata

As mentioned earlier, a pair of automata *i-j* can be in a linked (bonded) state or an unlinked (unbonded state). For unlinked pairs, the interaction only involves resistance to compression at the contact level. Here, the tangential force ($\vec{F}^{t}_{ij}$) is limited by the force of dry friction ($F^{df}_{ij}$) between the unlinked pairs *i-j* as given in equation (3.95). However, on the other hand, for linked pairs, the interaction involves the resistance to both compression and tension. Therefore, in this case, the tangential force is limited by the potential strength of the bond between the automata pair i-j (i.e. strength or yielding conditions), depending on the strength of the bond [1]–[3].

An automata pair can switch between the states. Switching of a pair of automata *i-j* from linked to unlinked state and vice versa results in a changeover in the forces acting on the elements; in particular, unlinked automata would not resist moving away from one another.

The switching of state of pair automata is controlled by different criteria depending on the material type and the physical mechanism of material behaviour. By knowing the stress and strain tensor in the bulk of an automaton, it is possible to directly apply conventional fracture criteria written in tensor form. If different types of material form pair, they might have different criteria, so the possible combinations of criteria that could be used are numerous [242].

To use these criteria, the local stress tensor at the area of contact between a pair of automata *i-j* must be known ($\sigma^{ij}_{\alpha'\beta'}$). To compute the components of this stress tensor more accurately the following approach is proposed. As shown in Figure 3-23, the local stress tensor components ($\sigma^{ij}_{x'y'}$) and ($\sigma^{ij}_{y'y'}$) are equivalent to the specific forces of interaction ($\sigma_{ij}$) and ($\tau_{ij}$) which are applied on the contact area ($A_{ij}$). The other components of the local stress tensors at the area of contact ($\sigma^{ij}_{x'x'}$) and ($\sigma^{ij}_{zz}$) are calculated using linear interpolation of the corresponding values in the centres of the automata:

$$\begin{cases} \sigma^{ij}_{x'x'} = (\bar{\sigma}^{i}_{x'x'} q_{ji} + \bar{\sigma}^{j}_{x'x'} q_{ij})/r_{ij} \\ \sigma^{ij}_{zz} = (\bar{\sigma}^{i}_{zz} q_{ji} + \bar{\sigma}^{j}_{zz} q_{ij})/r_{ij} \end{cases} \qquad (\mathbf{3.126})$$

where $\bar{\sigma}^{i}_{\alpha'\beta'}$ and $\bar{\sigma}^{j}_{\alpha'\beta'}$ are the components of average stress tensor in the volume of particles *i* and *j* associated with their centres.

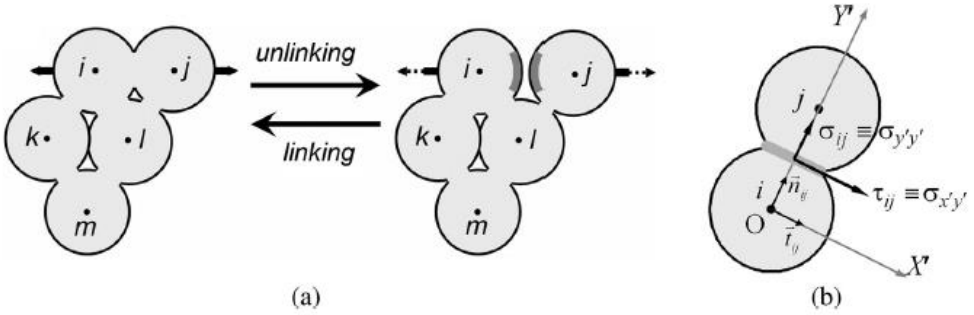



**Figure 3-23: (a) Schematic of switching between linked (at the left) and unlinked (at the right) states of the pair of automata *i–j* and (b) definition of instantaneous coordinate system $X_0Y_0$ associated with the current spatial orientation of the interacting pair *i–j*** [3]

### 3.5.3.1 Bond Breaking: Switching from Linked to Unlinked

Since MCA has a many-body interaction approach which allows for computing the stress tensor components, it is possible to apply well known fracture criteria used in continuum mechanics such as von Mises, Huber-Mises-Hencky, Drucker- Prager and others [3]. Fracture is modelled by means of transition of a pair of automata from a linked state to an unlinked state, with the possibility of further interaction, because unlinked automata could be either in contact or non-interacting.

The switching from linked to unlinked state, also known as bond breaking or fracture, can happen either when the equivalent stress ($\bar{\sigma}^{i}_{eq}$) is reached, or when reaching the equivalent strain ($\bar{\varepsilon}^{i}_{eq}$). Both can be used if the material is elastic and brittle, however, if the material is plastically hardened it is better to use the equivalent stress criterion. If the material performs perfect plasticity, then the use of a deformation criterion should be used. If a pair of linked automata i-j have different materials, then according to the von Mises criteria here, the automata will switch from linked to unlinked when equivalent stress ($\bar{\sigma}^{i}_{eq}$) reaches the fracture strength of the softer material [1]–[3].

Using the Drucker-Prager fracture criterion means that the criteria for tension and compression are different:

$$\sigma^{ij}_{eq} 0.5(a+1) + \sigma^{ij}_{mean} 1.5(a-1) = \sigma_c \qquad (\mathbf{3.127})$$

where $a = \sigma_c/\sigma_t$ is the ratio of compressive strength ($\sigma_c$) and the tensile strength ($\sigma_t$) of the material.

After coming into contact with another automaton, the automaton may bond again, switching from the unlinked to linked state.

#### 3.5.3.2 Bond Formation: Switching from Unlinked to Linked

The transitioning from unlinked to linked pairs, represent the formation or binding of bonds which is a characteristic of healing of cracks, microwelding of particles, etc. In consolidated materials it could also represent cohesion and adhesion of smooth contacting automata under compression, shear and/or friction [3], [251].

Here the switching criteria is controlled by a plastic work produced in the interacted pair of automata. This means that an adjustable amount of plastic deformation of the softer particle is taken as prerequisite for the binding of automata together, thus forming an aggregate of linked particles. Generally speaking different criteria exist involving plastic heat [155].

For example, for a new link (bond) to form in a pair of unlinked (unbonded) automata i-j (i.e. go from unlinked to linked), the criteria could be set based on two threshold values. For example, the switch may occur only when the value of the central compression strength is equal to the yield strength (i.e. the pair i-j experience plastic deformation), and to reach the value of the plastic work (heat) in the automata pair i-j, which means that the forces applied to the pair have to perform the work equal to the energy of the new chemical bond.

### 3.5.4 MCA Micro-Macro Relations and MCA Parameters

The constitutive law $\sigma_{eq} = \Phi(\varepsilon_{eq})$ defines the material response function. It describes the behaviour of the material and its rheological properties. There are generally four types of behaviour as shown in Figure 3-24. The simplest case is the linear elastic behaviour shown in Figure 3-24(a). Here the inter-automata interaction is assumed elastic and linear and follows Hooke's law [158].

When some damages occur at a scale lower than the automata size, the effective Young's modulus decreases due to degradation as shown in Figure 3-24(b). This degradation which is caused by damage generation corresponds to a load with a value higher than ($\sigma_d$), thus there is a linear response in the range $\langle 0 - \sigma_d \rangle$ and after that damages are generated in the range $\langle \sigma_d - \sigma_c \rangle$ where the response function is non-linear. Figure 3-24(a) and (b) could be used for simulating fracture of brittle materials, while irreversible behaviour is shown in Figure 3-24(c) as plastic deformation and Figure 3-24(d) shows plastic flow and material degradation.

By selecting the appropriate response function of the material, we can simulate elastic-plastic deformation and material degradation. The response function for well-

known materials could be found commercially, and if not or if more details are needed then experiments could be used. However, it is necessary to know the specific response function for the materials simulated to be able to simulate the appropriate corresponding behaviour correctly. If the simulated specimen is a composite material, then the response function for each of its constituents should be known.

It has been shown [3], [158], [172] that MCA simulations produce correct macroscopic mechanical response for elastic-plastic models on different length scales by using the material's macroscopic input parameters from the experimental stress–stain curve of any material.

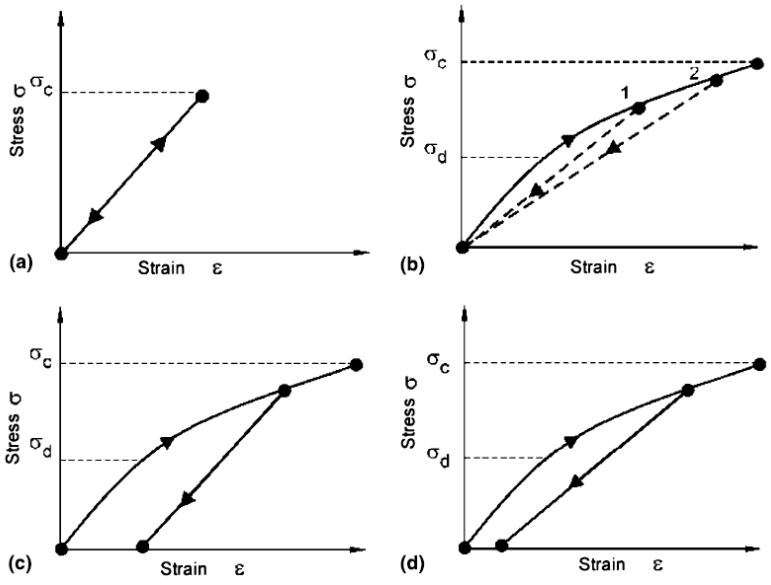



**Figure 3-24: Different types of materials response (a) linear (b) concrete like materials (c) plastic deformation (d) plastic flow and degradation** [158]

As mentioned earlier, the main advantage of the MCA laws of interaction forces is the ease of establishing a relationship between the forces and the tensor parameters of the material constitutive law. MCA particles are characterized by their size ($d_i$), mass ($m_i$), and mechanical properties such as; density ($\rho$), Young's modulus (E), Poisson's ratio (v), (or Shear modulus (G), Bulk modulus (K)) and also Yield strength ($\sigma_y$) and Work hardening modulus ($E_h$) for plasticity. These are all material properties known for any material or can be easily attained experimentally. These values are directly used as input in the simulation without the need for any parametric studies or calibration techniques like in DEM because there are no bulk properties involved such as lateral pressure ratio, angle of repose, size distribution, wall friction, coefficient of restitution, etc.

## 3.6 Summary

This chapter delivered the second contribution mentioned in section 1.4 of choosing the best platform to implement the MCA method and studying the mathematical background of MCA and DEM to identify their differences and functionalities of MCA needs to be added to LIGGGHTS.

As explained in this chapter, what mainly distinguishes one particle-based method from the other, is the manner in which it described and calculated the forces that occur between particles when they interact. The information about the contacts and the forces created in these contacts are essential for computing the stresses and strains of the particles in the system, which in turn we can translate into material deformation on the macro-scale. This is also the most time-consuming and computationally demanding part of any particle-based simulation.

Most researchers when talking about DEM limitations (more specifically the soft particle approach) only talk about its limitation of computational intensity in terms of power and time. However, there is a much bigger limitation which is the lack of established methodology to determine the particle properties and contact models to accurately model a given physical system. It is true that many advances have been made to overcome this problem, but it remains a point of concern. This is why DEM is most commonly used only for simulating granular, brittle and weakly bonded materials, which is due to the inadequate description of the interaction between the elements and its inability to describe a cohesive structure and irreversible processes (plasticity). Furthermore, appropriate representation of the macroscopic properties in DEM is still a challenge and it is sometimes difficult or impossible to obtain a required deformation behaviour [7]. The main challenge is to find constitutive laws that relate the stress and strain fields to the contact laws.

On the other hand, MCA simulates the motion of automata according to a multi-body inter-automata interactions form much like in MD models for metals [9][10], which made it possible to directly and correctly describe plastic deformation [16], [27]–[29]. Using the many-body form also allows it to avoid the artificial effects related to dependence on particle packing, packing related artificial anisotropy of mechanical response, and problems with the correct simulation of irreversible strain accumulation in ductile material, which is all extremely important for modelling elastic-plastic deformation of materials. The other main advantage of MCA is the generalized expression for the forces between particles is establishing a direct relationship between the parameters of particle interactions and tensor parameters of material constitutive law. This makes it possible and easy to implement different models and criteria of elasticity, plasticity and fracture, and describe elastic-plastic

deformation using the theory of plastic flow. As mentioned earlier, in MCA there is no need for parametric fitting because all the MCA parameters can be directly obtained from the material mechanical properties [3], [172].

Specifically, the limitations of DEM as described in LIGGGHTS can be concluded as follows:

1. Particles interact at contact points.
2. Particles are only allowed to overlap at contact point but not deform. Contact forces are only applied when the distance *r* between two particles of radii *Ri* and *Rj* is less than their contact distance *dist=Ri+Ri*, otherwise there is no force.
3. Only cohesion models allow for forces between particles when the particles surfaces do not touch.
4. The rotations of particles are not tracked.
5. Input parameters need to be calibrated because they are related to non-realistic material properties related to granular and granular flow.

The following changes and steps need to be implemented in LIGGGHTS to implement MCA:

1. Add MCA particle and interaction forces parameters
2. Add MCA contact area, mass and inertia tensor
3. Add normal (pressure) and tangential (shear) elastic force calculations
4. Add mean stress predictor
5. Add rotation, bending and torsion torques
6. Add parameters and code for plasticity
7. Add bond breaking and fracture criteria
8. Add bond formation
9. Add interaction with walls
10. Add periodic boundary conditions
11. Add heat generation and transfer
12. Implement Open MP for the main loops and MPI exchange of the new parameters for parallelization

# Chapter 4
# 3D MCA Model Development & Implementation in LIGGGHTS

## 4.1 Introduction

In this thesis the 3D MCA model described in Chapter 3 has been implemented in the DEM software LIGGGHTS. This chapter describes how LIGGGHTS works and presents the development and implementation steps, followed by verification of the model. The theoretical basis for the implementation was covered in Chapter 3 but more details are provided when needed.

Following the nomenclature used in LIGGGHTS, we will hereon further refer to the automata as “atoms” or “particles” interchangeably.

## 4.2 Software and Simulation Platform: LIGGGHTS

LIGGGHTS [198] is an open-source code and simulator written in C++ and is an acronym for “LAMMPS Improved for General Granular and Granular Heat Transfer Simulations”. LIGGGHTS was developed and distributed as an extension to LAMMPS by Christoph Kloss in JKU Linz, Austria, in 2011 to describe coarse-grained granular flow on the micro and macro-scales using DEM [252].

LAMMPS [253] is an acronym for “Large-scale Atomic/Molecular Massively Parallel Simulator”. LAMMPS was first developed by Plimpton and co-workers [254] in Sandia National Laboratories, USA, under GNU General Public License (GPL) and it is one of the main and highly powerful MD simulators used by the scientific and industrial community. Since both MD and DEM are particle-based models and have the same basic functionalities, the extension of LAMMPS to include DEM formulations (the development of LIGGGHTS) worked really well.

As stated by Kloss et al [255], “LIGGGHTS operates on macroscopic particles and tracks the trajectory of each. It is designed around an integration loop which integrates Newton's second law and resolves particle–particle and particle– wall collisions using a soft-sphere approach. Spring-dashpot models are used to compute forces caused by particle–particle interactions (pair forces) and particle–wall interactions. Additionally, volume forces such as gravity are applied.”

Both LAMMPS and LIGGGHTS have great parallelization capabilities which make them very powerful simulators. The parallelisation is implemented using MPI exchange. They can run on one processor on a desktop, as well as thousands of processors on High Performance Computers, allowing the simulation of very large

systems as it is implied by the name. Further information on the capabilities of LAMMPS and LIGGGHTS can be found in the LIGGGHTS documentation [189].

LIGGGHTS can be seen as an improved version of LAMMPS to move from MD to DEM simulations by adding featured from the DEM method that could not be found previously such as the contact force formulations using Hertzian and Hookian theories, also rolling friction, cohesion forces and heat conduction between particles.

This thesis does the same, it extends LIGGGHTS to move from DEM to MCA. This helps describe solid material behaviour (most importantly elastic-plastic deformation) on the meso and micro scales using MCA instead of granular material behaviour, the same way LIGGGHTS extended LAMMPS to implement DEM for describing granular material behaviour instead of atomic/molecular behaviour, as shown in Figure 4-1. The developed code has been named LIGGGHTS-MCA.

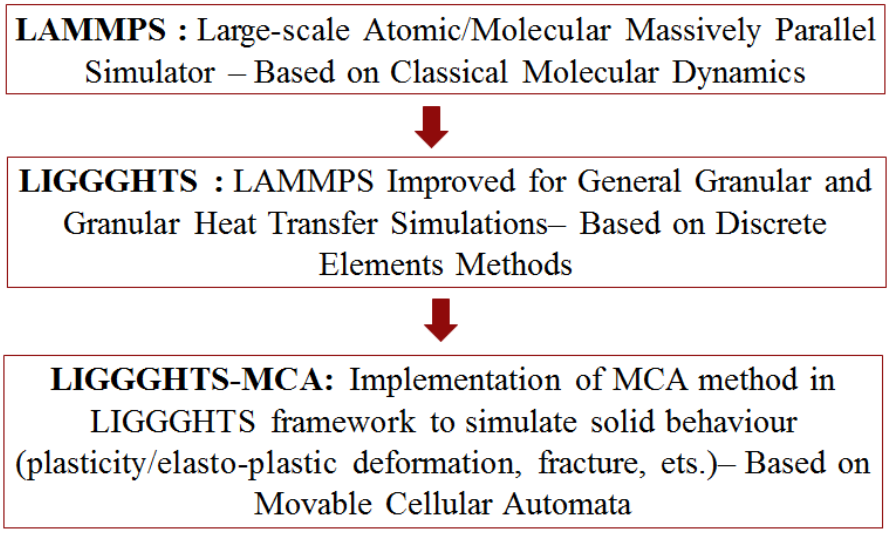



**Figure 4-1: The development of LAMMPS to LIGGGHTS, and now to LIGGGHTS-MCA**

It is important to mention that there is no commercial software available for MCA simulations, only a 2D MCA demo-version [170] which was developed by the Laboratory of CAD of Materials at the Institute of Strength Physics and Materials Science in Tomsk, Russia, in 2005. The developers of the MCA method use their own in-house codes. Thus, this project provides a 3D MCA open-source simulator (LIGGGHTS-MCA) for the scientific and tribology communities to use.

## 4.3 Overview of the Implementation of MCA in LIGGGHTS

Since LIGGGHTS is an open-source code, it can be downloaded from their website [198] or from github [256]. To implement our own functionalities, changes have been made to their source code (i.e. 'src' folder). It is a c++ object-oriented code, so the 'src' folder includes many files based in specific classes, which will be explained in the following sections. All changes have been made to LIGGGHTS Public version 3.3.1. released 23/09/2015.

A LIGGGHTS simulation requires three sets of data which in LIGGGHTS are entirely done via an input script:

1- Particle configuration, and state of contacts and/or bonds.

2- Inter-particle interaction formulas.

3- Defining the simulation set-up and problem definition

When starting the simulation, the input script is first read, which includes commands relevant to the chosen simulation attributes. A broad range of LAMMPS and LIGGGHTS commands are available for the specification of at least:

- any parameters related to the particle discretisation and configuration (e.g. size, insert on lattice structure or random, mass, density, etc.), material parameters (e.g. Young's modulus, Poisson's ratio, etc.), solver parameters (e.g. time step, total number of steps, etc.) and others.
- simulation domain, size and wall definition.
- integrator of choice, particle force interactions (e.g. potential functions in MD/LAMMPS), related thermodynamics information, neighbour lists, etc.
- any additional filters (e.g. temperature control, minimization, etc.)
- output style and configuration.

One of the advantages of LIGGGHTS is that if any command is used in a wrong way or a property is missing, LIGGGHTS will stop running and report an error, and if it is not a critical error it reports a warning message. The error or warning message reports which part of the code it is related to, specifically which class and what line in the code. For users who only use LIGGGHTS as a software and don't interact with the source code, the meaning of the error or warning messages could be found in LIGGGHTS's manual. This was taken into consideration while implementing our own code and our own error and warning messages have been developed as well.

Figure 4-2 shows the class hierarchy of the source code of LAMMPS, which is the same for LIGGGHTS. They are object-oriented codes where the top-level and main class for the entire code is 'liggghts.c'. The classes shaded in blue in the figure are

the top-level classes within LIGGGHTS main class and the ones in red are the virtual parent classes and are called styles [257].

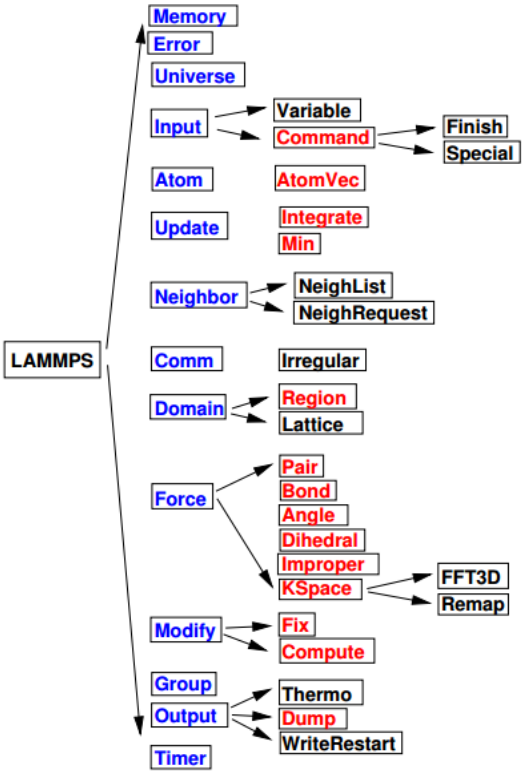



**Figure 4-2: Class Hierarchy in LAMMPS; similar in LIGGGHTS and LIGGGHTS-MCA [257]**

Each command in the input script corresponds to a specific class in the source code which defines the specific functionality. Just to mention the main classes used, the particle type and configuration is defined by an atom_style in the ‘Atom class’, particle-interaction formulas are defined by a pair_style in the ‘Pair class’, bonding of particles by a bond_style in the ‘Bond class’, and everything that happens during a simulation besides force computation, neighbour list construction and output is defined by a fix_style in the ‘Fix class’, including time integration, boundary conditions, force constraints, etc. ‘Computes’ calculate at one timestep, but ‘Fixes’ can alter something during timestep or maintain info from timestep to timestep.

The ‘styles’ define the different types of models that one can choose to use in the simulation. For example, the Atom class has styles such as sphere, granular, molecular, etc. chosen depending on the required type of particles for the type of simulation at hand. Similarly, the Pair class includes different pair styles defining particle-particle interaction laws, and the Fix class includes different fix styles defining the operations applied to the system during simulation (time integration, boundary conditions, external forces on atoms, etc.), as well as materials and interaction properties. This will all be explained in detail in the following sections.

To implement our MCA related functionalities, we introduced our own MCA commands. To create new ‘mca’ commands, new mca styles have to be created according to the corresponding classes and make corresponding changes to the main classes. New mca atom, pair, bond and fix styles have been developed which will be explained in detail in the following section 4.4.

An example input script using the new implemented mca commands can be found in sections 5.2.3 and 6.2.3. The following are the new ‘mca’ commands:

| | |
|---|---|
| *atom_style* | *mca args* |
| *pair_style* | *mca args* |
| *bond_style* | *mca* |
| *fix* | *ID group-ID nve/mca* |
| *fix* | *ID group-ID bond/create/mca args* |
| *fix* | *ID group-ID mca/setvelocity x y z* |

It is recommended by LIGGGHTS developers that when modifying or extending LIGGGHTS, the best way to add a new feature is to find a similar feature in LIGGGHTS and look at the corresponding source and header files to figure out what it does. Depending on how different your new feature is compared to existing features, you can either derive from the base class itself, or from a derived class that already exists [189].

It is worth noting here that LIGGGHTS Short Course was attended on 16 March 2016 in Linz, Austria, where the possibilities and added features were discussed with the developers of LIGGGHTS – specifically Andreas Aigner - to assure the proper use of the code and not adding features or attributes that already exist in LIGGGHTS, whether Public or their commercial code. They agreed that a lot has to be changed in the code to be able to describe plasticity and bonding according to the MCA method, and that it is better to write our own classes related to MCA.

So after careful consideration, the following list shows the new mca classes that were created to be able to create the new ‘mca’ commands listed above. It also shows the existing classes in LIGGGHTS that they were based on, however, most of them have been totally rewritten. Corresponding *.cpp and *.h files were created for each new mca class:

| | |
|---|---|
| atom_vec_mca.h | → atom_vec_sphere.h |
| atom_vec_mca.cpp | → atom_vec_sphere.cpp |
| pair_mca.h | → pair_soft.h |
| pair_mca.cpp | → pair_soft.cpp |
| bond_mca.h | → bond_gran.h |
| bond_mca.cpp | → bond_gran.cpp |

fix_bond_create_mca.h → bond_create_gran.h
fix_bond_create_mca.cpp → bond_create_gran.cpp
fix_bond_exchange_mca.h → bond_propagate_gran.h
fix_bond_exchange_mca.cpp → bond_propagate_gran.cpp
fix_mca_meanstress.h → fix_sph_density_summation.h
fix_mca_meanstress.cpp → fix_sph_density_summation.cpp
fix_nve_mca.h → fix_nve_sphere.h
fix_nve_mca.cpp → fix_nve_sphere.cpp
fix_wall_mca.h → fix_wall_sph.h
fix_wall_mca.cpp → fix_wall_sph.cpp
fix_mca_setvel.h → fix_smd_setvel.h
fix_mca_setvel.cpp → fix_smd_setvel.cpp

The reason behind choosing these LIGGGHTS classes as a basis for the implementation of the MCA features will be explained in detail in the following sections, as well as details of the development of the new code. All the existing classes were available in LIGGGHTS-Public, except for the classes related to the '*bond_style*', they were taken from a separate project developed by Christian Richer called LIGGGHTS-with-bonds [258] because when this project started in February 2016, LIGGGHTS did not have any features related to bonds between particles. Currently, LIGGGHTS-Public has a '*bond_style*' however bond breaking does not depend on forces or stresses between the particles, but just on distance. Also, the '*fix_smd_setvel*' style was taken from LAMMPS not LIGGGHTS.

After defining out new mca styles, they were included in their corresponding main base classes as follows:

| |
|---|
| #include "atom_vec_mca.h" in style_atom.h |
| #include "pair_mca.h" in style_fix.h |
| #include "bond_mca.h" in style_bond.h |
| #include *all* fixes in style_fix.h |

Before going into detail about the development of these new styles in the following section 4.4, it is first important to understand the structure of the LIGGGHTS source code and how the code runs when starting a simulation to be able to properly implement the new classes in their relevant parts of the code.

The flowchart in Figure 4-3 outlines the general structure of the LIGGGHTS program:

1. Read input file: parameters that specify the conditions of the run (number of particles, time step, material, etc)
2. Initialize: the system by setting initial positions and velocities
3. Update pair or neighbour list: to reduce cpu time
4. Compute the forces on all particles: uses the neighbour list
5. Integrate Newton's equations of motion
6. Update positions and velocities
7. Repeat steps 3 to 6 until time steps finished
8. Compute and print the averaged quantities
9. End

Each step corresponds to a respective function in the source code. LIGGGHTS works by calling the main functions in the order shown in the flowchart in Figure 4-3. After reading the input script, the first step is to execute the *init( )* and *setup( )* methods for initialization and setup before the run and actual calculations begin. This includes generating the atoms and initial structure as defined in the input script. Generating means that the necessary structures are allocated in the memory and each atom is assigned with a position and velocity. In case of bonds existing, the bonds between the atoms have to be designated. Also setting boundary conditions, defining neighbours, etc. After all necessary initialisation have been done, the actual simulation, i.e. the time integration of the atomic trajectories, and time-stepping starts following the Velocity-Verlet integration scheme [187], which was earlier explained in section 3.3.3, and revised here below in Algorithm 1.

**Algorithm 1:** Velocity-Verlet integration scheme

1- Calculate $\vec{v}\left(t+\frac{1}{2}\Delta t\right)=\vec{v}(t)+\frac{1}{2}\vec{a}(t)\Delta t$

2- Calculate $\vec{x}(t+\Delta t)=\vec{x}(t)+\vec{v}\left(t+\frac{1}{2}\Delta t\right)\Delta t$

3- Derive $\vec{a}(t+\Delta t)$ from the interaction forced using $\vec{x}(t+\Delta t)$

4- Calculate $\vec{v}(t+\Delta t)=\vec{v}\left(t+\frac{1}{2}\Delta t\right)+\frac{1}{2}\vec{a}(t+\Delta t)\Delta t$

Steps 1 and 2 of Algorithm 1 occur in the *initial_integrate( )* function shown in the flowchart, where the position and velocities are updated by a half time-step. Then step 3 mainly consists of computing the forces which correspond to steps 4-7 on the flow chart. Here mainly information between neighbours are exchanged, and pair and bond forces are computed as defined. The last step in the integration as shown in Algorithm 1, is updating the velocities by another half time-step which corresponds to the *final_integrate( )* method as shown in the flow chart as step 8. Then the time-stepping ends in step 9 and output files are generated with the results.

In all these steps, the calculation of the inter-particle forces is the most time-consuming part. The higher the number of particles, the relative distances and velocities between neighbouring pairs, the higher the computational time to evaluate the forces between them. The computational time is reduced by using cut-off distance, neighbour lists and linked cell list algorithms to identify the nearby particles and only update and calculate the forces on the particles within the neighbour area within a given time step as explained before.

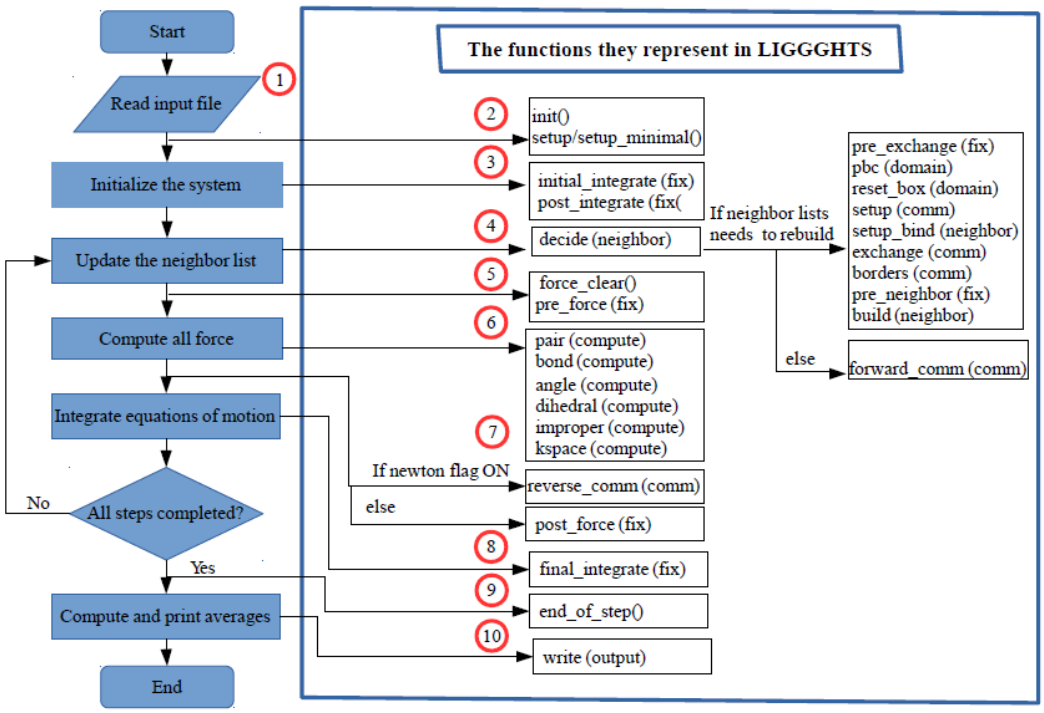



**Figure 4-3: Flow chart of program structure and the relevant functions in LIGGGHTS**

Regarding the new 'mca' classes, Algorithm 2 below shows the main flow of the MCA computation following the order of execution shown in the flowchart in Figure 4-3. Each function is executed in a relevant new MCA class added to LIGGGHTS.

**Algorithm 2:** Programme structure with relevant new MCA implemented classes

| | |
|---|---|
| 1- init( ) / setup( ) | [AtomVecMCA] |
| 2- initial_integrate( ) | [FixNVEMCA] |
| 3- post_integrate( ) | [FixBondCreateMCA] |
| 4- pre_exchange( ) | [FixBondExchangeMCA] |
| 5- pre_force( ) | [FixMeanStressMCA] |
| 6- pair_compute( ) | [PairMCA] |
| 7- bond_compute( ) | [BondMCA] |
| 8- post_force( ) | [FixMCASetvel] |
| 9- final_integrate( ) | [FixNVEMCA] |

## 4.4 Details of the Implementation of MCA in LIGGGHTS

The implemented codes are developed based on the source code of LIGGGHTS Public version 3.3.1. released 23/09/2015. Only the modified and added parts are presented here, other parts of the code which were unchanged are skipped. This section describes the implementation and development of the new styles in detail, following the order of the programme structure described in Algorithm 2.

### *4.4.1* New *Atom_Style* for MCA Particle Discretisation

In LIGGGHTS, the '*atom_style*' defines what type of atoms to use in a simulation, which determines what attributes are associated with the atoms that need to be stored and communicated between neighbouring atoms. The '*atom_style*' sets the parameters that need to be defined before atoms are created. The choice of style affects what quantities are stored by each atom, what quantities are communicated between atoms and processors to enable forces to be computed, and what quantities are listed in the data file. All styles store coordinates, velocities, and atom IDs, and then extra attributes are stored depending on the choice of style.

In LIGGGHTS, the main *atom_style* to simulate granular materials according to DEM is the '*sphere*' atom style which follows the description of section 3.4.1. Instead of using the keyword '*sphere*' in the input script, the keyword '*granular*' can be used [189]. Here spherical geometry of interacting objects is assumed, and they are defined by diameter, density and angular velocity. Throughout the code, the spherical geometry is used to calculate the moment of inertia of the particles, and the contact forces in function of the overlap between particles, assuming circular contact area. This is very different to how automata are described in MCA as described in Chapter 3, section 3.5.1.

To create a new '*atom_style*' for initializing the simulation for the calculations of mca, to compute and communicate between the processors each particle attributes and per-atom arrays, a new mca '*atom_style*' was added to LIGGGHTS. This '*mca*' is the new '*atom_style*' and '*AtomVecMCA*' is the class name defined in the '*atom_vec_mca.h*' and '*atom_vec_mca.cpp*' files. As mentioned, '*atom_vec_mca*' class was first copied by the existing sphere style defined in '*atom_vec_sphere*' class, then it was changed according to the attributes needed for the new atom 'mca' style. It has actually been totally rewritten except for the main LIGGGHTS functions.

The MCA '*atom_style'* command requires four arguments (variables):

1. radius → all automata have the same radius, so need to change from array to single variable.
2. packing → sc or fcc
3. n_bondtypes → the number of bond types which is 1 if all automata are linked, or all automata are unlinked, and 2 if some automata are linked and others are unlinked.
4. bond_per_atom→ number of bonds per atom is an array size for mca neighbours and is defined by coordination number (6 for sc or 12 for fcc).

The syntax of the mca '*atom_style*' in the input script is:

| | |
|---|---|
| ***atom_style*** | ***mca radius ${rp} packing fcc n_bondtypes ${bt} bonds_per_atom ${bpa}*** |
| For example: | |
| ***atom_style*** | ***mca radius 0.0001 packing fcc n_bondtypes 1 bonds_per_atom 12*** |

This is defined in the '*atom_vec_mca.cpp*' file under the '*void AtomVecMCA::settings(int narg, char **arg)*' method which defines how the style is written as a command. If they are not correctly defined in the input script, error messages will be produced to the user. This is used in all new MCA classes to define how the command is written so it will be not be mentioned again.

A big difference between '*atom_vec_sphere*' and '*atom_vec_mca*' is that atoms must store information about their bonds and include bond-related forces and torques in the equations describing their motion. Also, the bonds themselves must be able to store information about atoms they connect and to calculate stresses acting between them. In MCA the bonds are obligatory defined between all interacting pairs of automata in order to specify if the automata belong to one body (linked/bonded pair), or different bodies (unlinked/unbonded pair). This is different from DEM and MD where bonds add forces acting on atoms. Also, the number of neighbours an atom has depends on the packing and coordination number, 6 for SC and 12 for FCC and is defined by the number of bonds per atom. But during deformation other atoms can be in contact with it and the total number of interacting neighbours may be greater than the coordination number.

In the example above, the '*atom_style mca'* defined the particles radius to be 0.0001m which is 0.1mm, the initial structure has an FCC packing structure, the number of bond types is 1 because all particles are initially linked, and each article has 12 bonds and neighbours.

Here, the bonds cannot be used separately, as it has to be used with the '*atom_style mca*', but the bonds themselves and their breaking/formation are handled by the '*bond_style*'.

Other variables related to mca forces calculations need to be stored and communicated for each atom, as well as the rotational velocity, angular momentum and torques. Some members of the class '*atom_vec_sphere*' were used, but many new members have been introduced. The list below lists the new members that were added because they did not exist in '*atom_vec_sphere*':

1. packing → sc or fcc
2. coord_num → coordination number is 6 for cubic and 12 for fcc/hcp
3. mca_radius → single variable not array since all automata have same radius
4. contact_area → initial contact area defined by packing
5. mca_inertia → moment of inertia is scalar for simplicity
6. theta → orientation vector to describe rotation as a first approximation
7. theta_prev → orientation vector at previous time step
8. mean_stress → used for many-body interaction
9. mean_stress_prev → mean stress at previous time step
10. equiv_stress → equivalent (or von Mises, shear) stress, used for plasticity
11. equiv_stress_prev → equivalent stress at previous time step
12. equiv_strain → equivalent (shear) strain, used for plasticity
13. cont_distance → distance to free surface, to determine a new contact
14. bond_mca → local number of bonded automata
15. bond_hist → array including indices of values related to the internal state of the bond.

The listed variables are then implemented within the different methods defined in '*atom_vec_mca*' to ensure that each atom stores and copies this information from one time-step to the next. Some methods are also related to dumping the information of each particle to an output file for postprocessing. Other functions are related to MPI exchange for parallelization and restating of simulations however these were not fully implemented. Care has been taken such that no unnecessary information is transferred to ensure computational efficiency and memory considerations. Furthermore, corresponding atom arrays are also defined in the parent class '*atom.cpp*' and '*atom.h*' so that they are accessible throughout LIGGGHTS when using the mca atom style command. LIGGGHTS made this easy by searching for the word "customize" which finds the locations were the new atom arrays need to be introduced.

Another crucial element, which is also very important for force calculations in the following section, is the definition of '*bond_hist*' (number 15 in the previous list) which is actually the contact history, which stores all relevant information about a bond (contact) of a particle from the previous time-step. The following are indices of values stored in the '*bond_hist*' array:

1. MAX_BONDS → maximum number of bonds
2. STATE → bonded/unbonded/not interacting
3. R → distance to neighbour
4. R_PREV → distance to neighbour at previous time step
5. A → contact area
6. E → normal strain of *i*
7. P → normal force of *i*
8. P_PREV → normal force of *i* at previous time step
9. NX, NY, NZ → unit vector from *i* to *j* in *X, Y, Z* directions respectively
10. NX_PREV, NY_PREV, NZ_PREV → unit vector from *i* to j at previous time step in *X, Y, Z* directions respectively
11. YX, YY, YZ → history of shear force of *i* in *X, Y, Z* directions respectively
12. YX_PREV, YY_PREV, YZ_PREV → history of shear force of i at previous time step in *X, Y, Z* directions respectively
13. SHX, SHY, SHZ → shear strain of *i* in *X, Y, Z* directions respectively
14. SHX_PREV, SHY_PREV, SHZ_PREV → shear strain of *i* at previous time step in *X, Y, Z* directions respectively
15. MX, MY, MZ → bending-torsion torque of *i* in *X, Y, Z* directions respectively
16. SX, SY, SZ → shear force of *i* in *X, Y, Z* directions respectively

Furthermore, in the 'sphere' style the radius is the only parameter describing the size and shape of the particles. Throughout the code, it is used to calculate the contact forces as a function of overlap between particles, to calculate moment of inertia of particles and moments and forces related to the elastic bonds between particles following the description in section 3.4. However, as mentioned in section 3.5, in MCA although the size of an automaton is characterized by a radius, the real shape of the automaton is not always a sphere. It is determined by the area of its contact with its neighbour. This equivalent shape is characterized by a new radius parameter which is calculated from the initial volume of the automata. Hence, in the '*atom_vec_mca*' class, the initial volume of automata and the initial contact area are

also computed, which are calculated based on the radius and packing. It is, however, enough to use the equivalent circular sphere characterized by ($d_i$) and the value of contact area ($A_{ij}$) when calculating the inter-automata interaction forces and torques.

In 3D it is assumed that for sc the automata have cubic shape hence; $A_{ij} = d^2$ and volume $\Omega_i = d^3$. While for fcc the automata have rhombic dodecahedron shape hence; $A_{ij} = d^2/(2\sqrt{2})$ and $\Omega_i = d^3/\sqrt{2}$. $A_{ij}$ also changes during elastic or elastic-plastic deformation.

Also, in 'sphere' the mass is assigned to individual particles on a per-particle basis, while in 'mca' the mass is calculated by multiplying initial volume by density of each particle, hence again it depends on the packing. Also, to simplify the computation of rotation, it is assumed that the automata is a ball and its inertia can be described by one parameter (scalar), where the radius of this ball is calculated from the initial volume of the particle. Thus, the moment of inertia of a ball of radius $R$ and mass $m$ is $\hat{J} = 0.4\, m R^2$. For automata we use effective radius corresponding to the ball of the same volume $4/3\pi R_i^3 = \Omega_i^0$ hence this radius is $R_i = \sqrt[3]{3\Omega_i^0/4\pi}$

Each particle now knows its own attributes which are stored, knows it neighbours and its relationship with them, so time-stepping can begin. This is described in the following section which describes the new integration scheme.

**Related files:** *atom_vec_mca.h* and *atom_vec_mca.cpp*

**Related commands:** *atom_style mca*

### *4.4.2* New *Fix_Style* for MCA Integration Scheme

In LIGGGHTS, the '*fix nve*' command performs constant NVE integration to update the position and velocity for each particle in a group each timestep according to Velocity Verlet scheme as described in section 3.3.3. and Algorithm 1. Furthermore, as explained in section 3.3.5. the NVE microcanonical ensemble is used to simulate an isolated system where energy and entropy are conserved by keeping the number, volume and energy of the system constant. Here the equations of motions are solved without temperature or pressure control.

For each '*atom_style'* there is a relevant '*fix nve'* style. For granular particles in LIGGGHTS, the '*fix nve/sphere*' is used because '*fix nve'* is used in molecular dynamics where atoms are assumed to be point particles and only their position and velocities are updated. However, for granular particles, in addition to their positions and velocities, the equations also need to be updated for their angular velocities. Hence, '*fix nve/sphere'* is used with the '*atom_style sphere'* because it needs, the radius, the torque and angular velocity of each particle to be stored at each time-step which is done by '*atom_style sphere*'.

For MCA, in addition to the position, velocity and angular velocity, the rotation (i.e. orientation vector theta) of each particle also needs to be updated according to equation 3.7. Furthermore, since we introduced our own mca_intertia and theta in 'atom_style mca' as explained in the previous section, the equations need to be redefined. Also, the calculation of velocity and angular velocity in '*fix nve/sphere'* considers the mass coefficient and relative fluid density which is irrelevant to MCA and has to be removed. Thus, a new '*fix nve/mca*' style is developed to be used with '*atom_style mca*' and is defined in the '*fix_nve_mca.h'* and *'fix_nve_mca.cpp'* files.

The syntax of the '*fix nve/mca*' command is:

| |
|---|
| ***fix ID group-ID nve/mca*** |
| For example: |
| ***fix integr nve_group nve/mca*** |

As shown earlier in Algorithm 2, and the flow chart Figure 4-3, the integration occurs in two steps, *initial_integrate( )* and *final_integrate( )* where the forces are updated between the two integration steps. Thus, this command starts the time-stepping by first executing the *initial_integrate( )* method which updates the velocity and angular velocities of the particles by a half time-step, and positions and orientation vectors by one time-step. Then by executing the other half of the integration by *final_integrate( )* which updates the velocities and angular velocities

by another half step, and the positions and orientation vectors by one step to obtain the final location and velocities of the automata.

**Related files:** *fix_nve_mca.h* and *fix_nve_mca.cpp*

**Related commands:** *fix nve/mca*

### 4.4.3 New *Fix_Style* to Create Bonds Between Particles

As mentioned in Chapter 3, all particles are initially bonded, that could mean they are only in contact, or bonded, or unbonded. After that with time-stepping, according to force and bond calculations, the particles can change their positions to be in compression/tension state, and/or switch their state to break or form a bond. This is different from defining bonds in LIGGGHTS where only pair forces are defined between all particles, but bonds are defined between specified pairs of particles and remain in force for the duration of the simulation, unless the bond breaks which is possible in some bond potentials and defined by the '*bond_style*'.

In MCA pair forces and bonds are defined for all particles in the system and the set of interactions can change over time during the simulation such that new bonds can be created during the simulation time even if they were not bonded initially. How these interaction forces are computed is defined in sections 4.4.5 and 4.4.7. related to the '*pair_style'* and '*bond_style'*. But first, at every time-step a check for possible new bonds between particles needs to be done and a list of bonds need to be stored for each particle, which is done by this '*fix bond/create/mca'* style. This has to be done in the *post_integrate( )* phase, which is after the first half of the integration *initial_integrate( )* and before calculating the forces.

It is worth noting that when we started this project in February 2016, LIGGGHTS-Public did not have a '*bond_style*' and did not include any features related to bonding of particles. However, there was a project called LIGGGHTS-WITH-BONDS developed by Richer [258], where his 'bond_style gran' had to be used with 'atom_style bond/gran' which is again different from MCA particles. They also exist in LAMMPS, however, we based all our new mca-bond-related classes on the bond classes developed by Richer as mentioned in section 4.3 but made a lot of changes. Currently, LIGGGHTS does have a '*bond_style*' which computes bond forces only based on distances between particles.

After the particles are generated and pairs are defined by the '*atom_stylce mca'*, and given an initial position, velocity, angular velocity and orientation vector by the '*fix mca/nve'* style, each particle needs to be checked for bonding conditions. If the conditions are fulfilled, bonds will be created and initialized.

This can be used to define regions of different materials within the same simulation, whether within the same block of material, such as a composite or an alloy, or blocks of different materials away from each other with a distance between them such as for example in indentation where you have a sample and an indenter.

Thus a new '*FixBondCreateMCA*' class was created based on the '*FixBondCreateGran*' class and defined in '*fix_bond_create_mca.h'* and

‘*fix_bond_create_mca.cpp’* files, which is executed after the first half of the integration step in the *post_integrate( )* to check, create and store a list of possible bonds for all particles for which bond forces will be calculated.

The syntax of the ‘*fix bond/create/mca’* command is:

| |
|---|
| ***fix ID group-ID bond/create/mca Nevery itype jtype cutoff bondtype bonds_per_atom*** |
| For example: |
| ***fix bondcr all bond/create/mca 1 1 1 ${cutoff} 1 ${bpa}*** |

This means that a check for possible new bonds between atom itype and atom jtype within a specific cut-off distance is performed every Nevery time-steps. If no bonds already exist between atoms *i* and *j*, they exist within the same specified group and they have not reached their allowed maximum number of bonds, then they are listed as a possible bond pair. Of course, more than one particle can fall within the cuttoff distance of a particle, so it might have multiple possible bonds. Hence, each particle checks its list of possible pairs and chooses the closest particle.

In the above example, if *i* and *j* are in the same group, have the same material, have the same bond type, if they fall in the same cut-off distance and have the same maximum number of bonds (meaning they are both sc or both fcc), then they are listed as a bond pair and the list is checked every time-step.

To implement this, a few extra new parameters had to be implemented. The most important ones are the ‘*init_state’* which describes the initial state of the bond, and set it is 1 if unbonded and 0 if bonded which is also the default. ‘*maxbondsperatom*’ is also again defined here based on coordination number, and the most important parameter ‘*cont_distance’*.

‘*cont_distance[i]*’ is the distance to the free boundary of the automaton. Initially it is the radius. Then it is updated based on the mean stress, if the automaton is under compression then ‘*cont_distance’* increases and vice versa. The strain tensor can also be used to compute this distance in the particular direction to the j-th neighbour, but here this simplification is used which works well for small plastic distortions.

So, first we check if the distance is less than the sum of ‘contact_distance[]’. If the distance is larger, then the automata do not touch each other, and new bond cannot be created. If the type of pair of automata is not listed in the ‘*fix bond/create/mca’* command then the bond will also not be created, and it will also not be created if the particle reaches its maximum number of bonds (the coordination number)

**Related files:** *fix_bond_create_mca.h* and *fix_ bond_create_mca.cpp*

**Related commands:** *fix bond/create/mca*

### 4.4.4 New *Fix_Style* for MCA Neighbour List Update

As mentioned in section 3.3.7, because the calculation of forces is the most time-consuming part of a simulation code, neighbour lists are produced and /or checked at every time step to exclude calculating the interaction forces for any far away automata. In LIGGGHTS, the cut-off or skin distance is defined and to create the neighbour list for a particle, it is only necessary to evaluate distances to particles within this distance. This is done by the 'neighbour' command and it is similarly used in MCA, so nothing has been added or modified for that.

However, because new contacts and bonds can be created during the simulation time, the bond contact history also needs to be updated and exchanged. This means that the bonds that are intact are exchanged to the next time-step and if running in parallel also to neighbouring processors, while it removes broken bonds or particles that lost contact. This is done in a new class called '*FixExchangeBondMCA*' based on the *'FixBondPropagateGran'* class and defined in '*fix_bond_exchange_mca.h'* and '*fix_bond_exchange_mca.h'* files. However, this 'fix_style' is turned on automatically when using the '*atom_style mca'* and works for all particles alike so it does not need to be defined in the input script. By default, the neighbour command defines the neighbour list and '*FixExchangeBondMCA*' uses that.

Similar to the '*fix create/bond/mca'*, this needs to be done at every time-step before force calculation. It is done in the *pre-exchange( )* phase, after the list of bonds has been created by '*fix create/bond/mca'.* Furthermore, this fix is also added to the 'delete_atom.cpp' and 'atom_vec_mca.cpp' files to delete any broken particles from the stored information and update their number of bonds.

**Related files:** *fix_bond_exchange_mca.h* and *fix_ bond_exchange_mca.cpp*

**Related commands:** -

### *4.4.5* New *Pair_Style* for MCA Automata Interactions

The '*pair_style*' is one of the most important commands in LIGGGHTS because it defines the interaction laws between particles within a simulation. It is the most time-consuming part and the calculations need to be accurate and efficient.

LIGGGHTS uses pairwise interactions defined between pairs of particles (granulars) which are within a cut-off distance, and the interactions change over time as explained in section 3.4.2. [189]. The main '*pair_style*' used in LIGGGHTS is the '*pair_style gran*' which calculates the normal and tangential components of the forces between two neighbouring particles according to Hertz-Mindlin or Hookean contact laws. For each simulation you can set model values for (hertz/hooke/hertz or hooke stiffness), tangential values (history/no history), and cohesion values. For more details refer to section 5 of LIGGGHTS manual. The forces are calculated based on the properties shown earlier in Table 3-1 and defined in the input script.

As explained in section 3.5, MCA uses completely different formulations to calculate the interaction between forces, so a new '*PairMCA*' class is added and defined in the '*pair_mca.h*' and '*pair_mca.cpp*' files and executed by a new command called '*pair_style mca'*. The '*PairSoft*' was chosen to be used as a base class for implementing our new '*PairMCA*' class instead of '*PairGran*' because of its simplicity and because '*PairGran'* is related to many other commands related to granular particles that will not be used in the MCA formulations. Also, '*pair_soft'* is defined in LAMMPS as well, which will make it easier if this project (the new mca styles) will later be implemented in LAMMPS.

'*PairSoft'* class will help to define the pair-wise part ($\vec{F}_{ij}$) of equation 3.47, however the many-body interaction part ($\vec{F}_i^{\Omega}$) is totally new. Also, as explained in section 3.5.2.2. to calculate the normal forces of interaction a predictor for the mean stress ($\bar{\sigma}_{mean}^{i}$) needs to be defined and calculated first, which is done in a separate class called 'FixMeanStressMCA' and will be explained in the following section 4.4.6.

The syntax for the mca '*pair_style mca'* command is:

| | |
|---|---|
| ***pair_style*** | ***mca cutoff*** |
| ***pair_coeff*** | ***I J args*** |
| For example: | |
| ***pair_style*** | ***mca 2*${d}*** |
| ***pair_coeff*** | ***1 1 ${COF} ${G} ${K} ${Sy} ${Eh}*** |

The MCA '*pair_style mca'* command only requires one argument which is the cut-off distance. As explained in section 3.3.7., this cut-off distance defines the neighbour list for interactions, any atoms outside this distance are not taken into

consideration in the calculation of the forces. The cut-off distance for force calculations can be set as separate value, but if not specified, the global cut-off will be used by default.

The command also needs a '*pair_coeff*' command which is obligatory to define the elastic and/or plastic properties for one or more pair of particles in the simulation domain. This is all defined in the *coeff( )* method in the '*PairMCA'* class. For granular styles there are no additional coefficients to set which is also another reason for using the '*pair_style soft'* as a base for the implementation. The type of pair of atoms *i* and *j* are defined in '*pair_coeff*' command, in the example shown, they both have the same material type '1'. After that, the arguments related to this command are the coefficient of friction ${COF}, which is defined for both elastic and plastic forces, the shear modulus ${G}, and the bulk modulus ${K} for calculating the elastic forces, and yield strength ${Sy}, and plastic work hardening modulus ${Eh} for calculating the corrector for plastic forces.

Algorithm 3 shows the steps for calculating the forces on the particles as defined by the new class 'PairMCA'. The elastic specific forces are calculated first, then the corrector for plasticity is calculated using the equivalent stress as described before, and then the total forces are obtained.

**Algorithm 3:** MCA forces computation algorithm - calculation in every time step

1- Calculate $\sigma_{ij}$, $\vec{\tau}_{ij}$ and $K_{ij}$ at current time step (n+1).

2- Calculate corresponding values of $\bar{\sigma}^i_{xx}$, $\bar{\sigma}^i_{yy}$, $\bar{\sigma}^i_{xy}$, $\bar{\sigma}^i_{zz}$ , $\bar{\sigma}^i_{mean}$, $\bar{\sigma}^i_{eq}$ and $\bar{\varepsilon}^i_{eq}$.

3- Examine the value of $\bar{\sigma}^i_{eq}$ , and correct for plasticity by calculating $\sigma'_{ij}$ and $\tau'_{ij}$.

4- Calculate $\vec{F}^n_{ij}$ , $\vec{F}^t_{ij}$ , $\vec{M}_{ij}$ and $\vec{K}_{ij}$

Algorithm 3 is implemented in the code by the following methods in this order:

- void compute_elastic_force();
- void compute_equiv_stress();
- void correct_for_plasticity();
- void compute_total_force();

First the elastic forces are calculated in the *compute_elastic_force( )* method for each particle by calculating the specific normal force ($\sigma_{ij}$) and normal strain ($\varepsilon_{ij}$) for each particle according to the equations in sections 3.5.2.1. and 3.5.2.2. Then the specific tangential (shear) force ($t_{ij}$) is calculated for each particle, taking into consideration the rotation of particles, using the equations (3.89) to (3.91) in section 3.5.2.3.

To compute the elastic force the shear *${G}* modules and bulk modulus *${K}* defined in the '*pair_coeff*' command are used, which are given as input parameters and calculated from Young's modulus ($E$) and Poisson's ratio ($v$) as follows:

$$G = \frac{E}{2(1+v)} \tag{4.1}$$

$$K = \frac{E}{3(1-2v)} \tag{4.2}$$

After that the equations are corrected by applying friction between particles, using the friction coefficient *${COF}* variable in '*pair_coeff*' command, according to the equations in section 3.5.2.4, and also the calculation of the bending and torsion torques. Rotation is also taken into consideration and added to the code for tangential forces and torques calculation which allows to correctly describe isotropic response of material as shown in [246]. The use of vectors for rotation was defined in a new file '*rotations.h'* which is included in the '*pair_mca.cpp*' file.

The next step is to compute the equivalent stress ($\bar{\sigma}_{eq}^{i}$) and equivalent strain ($\bar{\varepsilon}_{eq}^{i}$) for each particle in the *compute_equiv_stress( )* method to be able to correct the elastic forces for plasticity according to the equations in section 3.5.2.5. To compute the equivalent stress, the mean stress is calculated according to equation (3.63). However, for the calculation of elastic and plastic normal forces, a predictor for the mean stress is defined in '*FixMeanStressMCA*'.

Then the model of plastic flow with von Mises criterion of plasticity is implemented to simulate deformation of locally isotropic elastic-plastic medium using the radial return algorithm of Wilkins as described in section 3.5.2.5. The yield stress *${Sy}* and the plastic work hardening *${Eh}* defined in the '*pair_coeff*' command are used to calculate the plastic corrected forces in *correct_for_plasticity()* method. It is worth noting that the plastic work hardening modulus ${Eh} is the dependency of the yield limit on plastic strain and is here approximated by linear functions. Multi-linear behaviour is not implemented yet. Then the dry friction is re-calculated based on corrected values for plasticity.

Then finally, all total forces and torques are calculated in the *compute_total_force( )* method using the formulas for contact area as shown in equation 3.55.

Throughout 'PairMCA' the type of forces calculated for bonded and unbonded particles differ. For unbonded pairs friction forces are applied and for unbonded pairs no torques are applied due to relative rotation.

The different interaction forces implemented in 'PairMCA' depending on the type of pair interaction for both elastic and plastic forces are summarized in Table 4-1.

**Table 4-1: Types of interactions and their corresponding forces in '*PairMCA*'**

| **Type of Particle Pair** | **Forces between particle Pairs of the same material or different materials** |
|---|---|
| In Contact & Bonded | Central (normal) force<br>Tangential (shear) force<br>Dry friction force<br>Bending Torques<br>Torsion Torques |
| In Contact & Unbonded | Central (normal) force<br>Tangential (shear) force<br>Dry friction force |

For particle interactions with walls, a new '*FixWallMCA*' class is created based on the '*FixWallSph*' class, defined in '*fix_wall_mca.h'* and '*fix_wall_mca.cpp'* files and can be executed by 'fix wall/mca' style in the input script. However, currently the particle-wall interaction is similar to particle-particle interaction, hence it is not fully implemented yet. This should employ rigid boundary conditions.

**Related files:** *pair_mca.h, pair_mca.cpp, rotations.h, fix_wall_mca.h* and *fix_wall_mca.cpp*

**Related commands:** *pair_style mca* and *pair_coeff **

### 4.4.6 New *Fix_Style* for Mean Stress Predictor

Before starting the force calculation on each automaton, the predictor for $\bar{\sigma}^{i}_{mean}$ estimation should be calculated as explained in section 3.5.2.2. because the specific forces of interaction between the automata at the current time step is calculated using $\bar{\sigma}^{i}_{mean}$. At the first step of integration, an estimation predictor for the mean stress is used to calculate the normal force ($F^{n}_{ij}$), and after that current stresses are calculated using the previous ones. Here special predictor estimates for mean stresses are defined to compute the normal specific interaction force.

This is done in the *pre_force ( )* method and defined in a new '*FixMeanStressMCA*' class, following equations 3.64-3.81 in section 3.5.2.2 and described in '*fix_meanstress_mca.h'* and *'fix_meanstress_mca.cpp'* files. '*FixMeanStressMCA*' class consists of two main methods, *swap_prev( )* for swapping some data from previous time-step to the current one related to the bond history list, and *predict_mean_stress( )* for describing the predictor for mean stress.

This '*fix_style'* is turned on automatically when using the '*atom_style mca'* and '*pair_style mca'* and works for all particles alike so it does not need to be defined in the input script. It is based on the existing 'FixSphDensitySummation' class which links the density field and the pressure field, and used to describe the link between mean stress and pressure (normal force) in 'PairMCA'

**Related files:** *fix_meanstress_mca.h* and *fix_meanstress_mca.cpp*

**Related commands:** -

### 4.4.7 New *Bond_Style* for MCA Bond Model

The 'bond_style' loops through all bonds and calculates bond forces on particles. As mentioned, when this project started in February 2016, LIGGGHTS did not have any features related to bonds between particles. Currently, LIGGGHTS-Public has a elastic '*bond_style*' however bond breaking does not depend on forces or stresses between the particles, but just on distance.

Thus, a new mca *'BondMCA'* class was created based on '*BondGran*' class developed by Richer called LIGGGHTS-with-bonds [258] and defined in '*bond_mca.h'* and *'bond_mca.cpp'* files. 'BondGran' uses the formulations described in section 3.4.3. and the bonded-particle model for rock described by Potyondy and Cundall in 2004 in [233]. It uses $(k_n)$ the normal bond stiffness parameter, $(k_t)$ the tangential or shear bond stiffness parameter and $(r_0)$ the initial particle distance upon bond creation to define the bond between the particles. As mentioned, these are parameters that cannot be extracted from experiments and are difficult to calibrate.

In MCA, this is different, as explained in section 3.5.3. and bonds are obligatory defined for all pair particles to describe their contact interaction. Again, this class is totally rewritten because the mca parameters mentioned in the '*AtomVecMCA*' class are used, as well as the bond history list updated by the '*BondCreateMCA*' and '*BondExchangeMCA*' classes, and also to define our own bond fracture criteria.

First a check is done to evaluate if the bond has been already broken, by checking if the pair are still interacting in compression, if not it will be removed by the 'FixBondExchangeMCA'. If they are then the '*bond_state'* is set as '*non_interact'*, if not then computation continues and the bond is broken if the breaking criterion is met, which is all done in the *compute( )* method. As mentioned in section 3.5.3, the equivalent stress $(\bar{\sigma}_{eq}^{i})$ or equivalent strain $(\bar{\varepsilon}_{eq}^{i})$ criterion can be used if the material is elastic and brittle, however, if the material is plastically hardened it is better to use the equivalent stress criterion. If the material performs perfect plasticity, then the use of a deformation criterion should be used such as the Drucker-Prager fracture criterion, Huber-Mises-Hencky criterion or others. If a pair of linked automata i-j have different materials, then according to the von Mises criteria here, the automata will switch from linked to unlinked when equivalent stress $(\bar{\sigma}_{eq}^{i})$ reaches the fracture strength of the softer material [1]–[3]. The syntax for the *'bond_style mca'* is:

| | |
|---|---|
| ***bond_style*** | ***mca*** |
| ***bond_coeff*** | ***N bond_type Fract_Criterion Fract_Param Bond_Criterion Bind_Param*** |
| For example: | |
| ***bond_coeff*** | ***1 1 0.2*** |

'*bond_style mca'* does not have any arguments, but similar to '*pair_style mca'*, '*bond_style mca'* also needs the '*bond_coeff*' which must be defined for each bond type which is implemented in the *coeff( )* method in '*BondMCA*' class.

*N* is the number of bond types. *Fract_Criterion* is an index of the fracture criterion used and *Fract_Param* is a corresponding parameter, and are set as

0 = no fracture, no parameters
1 = ultimate equivalent strain, needs one parameter: strain
2 = ultimate equivalent stress, needs one parameter: stress (Pa)
3 = Drucker-Prager criterion, needs two parameters: tension stress (Pa) and compression stress (Pa).

Then the *Bond_Criterion* and *Bind_Param* are set which are the index and parameter for the bond formation criterion as:

0 = no binding, no parameters
1 = ultimate pressure, needs one parameter: pressure (Pa)
2 = ultimate plastic heat, needs one parameter: plastic heat (J/kg)
3 = combined criterion, needs two parameters: pressure plastic heat

In the example, N is set as '1' which means this bond_coeff will set parameters for bond type '1'. Then *Fract_Criterion* is set as '1' which means it uses the ultimate equivalent strain as fracture criterion and the strain is set as 0.2. Here no *Bond_Criterion* or *Bind_Param* are set (empty arguments) because they are not implemented yet.

**Related files:** *bond_mca.h* and *bond_mca.cpp*

**Related commands:** *bond_style mca* and *bond_coeff **

### 4.4.8 New *Fix_Style* for Setting Velocity Boundary Conditions

After all the forces are computed for each particle due to their interactions, external forces could be applied using boundary conditions, so this is executed by the *post_force( )* method. In LIGGGHTS there are many ways of applying external forces, for example using '*fix addforce'* command, however, instead of adding forces, it is better to add velocities to the particles thus imposing velocity boundary conditions. This mainly enables outputting of forces of the particles depending on the velocity acting on specific regions of the domain.

Thus a new 'FixMCAsetVel' class was created and defined in *fix_mca_stevel.h* and *fix_mca_setvel.cpp* files, based on the 'FixSMDSetvel' class which is part of the USER-SMD package for LAMMPS which is a smooth particle hydrodynamics package [259]. Here velocities are added in the x,y and z directions to a group of particles regardless of the forces acting on them.

It is executed in the input script by the following command:

| |
|---|
| ***fix ID group-ID mca/setvelocity x y z*** |
| For example: |
| ***fix topV_fix top mca/setvelocity 0 0 0.001*** |
| ***fix topB_fix bottom mca/setvelocity NULL NULL -0.001*** |

The velocity in X,Y and Z directions may be specified for specific groups of atoms. For example, the first fix specifies that the particles in the group 'top' will have zero velocity in x and y directions, and an upwards velocity of 0.001m/s in the z direction. The second fix specifies that the particles in group 'bottom' will have zero velocity in x and y direction, and a downwards velocity of 0.001m/s in the z direction. The velocities could also be defined as variables instead of constant velocity and evaluated at each time-step and used to calculate the forces of the particles. Using 'NULL' means that the velocity component in that direction does not change, and it allows the sample to shrink at the loading regions due to Poisson's effect.

This command only works in combination with the '*fix nve/mca'* command.

**Related files:** *fix_mca_stevel.h* and *fix_mca_setvel.cpp*

**Related commands:** *fix mca/setvelocity*

### 4.4.9 Implementation of Periodic Boundary Conditions

As mentioned earlier in section 3.3.6., different types of boundary conditions can be used on simulations. Currently all the MCA implementations and simulations employ non-periodic fixed boundary conditions (defined as ‘boundary fff’ in the input script), which means that particles do not interact across the domain of the simulation and the position of the domain is fixed. If particles move outside of the domain they are lost. Periodic boundary conditions (PBC) can be viewed as surrounding a simulation cell with replicates of the system which is particularly useful when simulating materials on small scales because realistically the domain is much larger than the simulation domain. PBCs are used to avoid boundary effects. In a simulation, PBC allows particles near a boundary to interact with particles on the other side of the domain as if they were nearby, and particles leaving the domain would reappear on the opposite side. This is a common technique for simulating an infinite bulk system.

This is described in LIGGGHTS within the domain decomposition framework using the forward communication buffering methods which executes the flow of communication between locally ‘owned’ particles on a processor and corresponding ‘ghost’ particles on a different processor. To implement PBC for the new MCA functionalities, a few files needed to be changed. The ‘domain.h’ file which is a main class in LIGGGHTS defining the different boundary conditions, is included in the ‘AtomVecMCA’, *‘PairMCA’, ‘FixMeanStressMCA’* and *‘FixBondCreateMCA’* classes. Then the forward communication buffering methods were modified to include PBC including pack_comm, pack_comm_vel, pack_border, and pack_border_vel. These methods are also used in the MPI exchange which will be explained in the following section.

This feature is still not fully implemented and contains some bugs, but nevertheless it provides an invaluable starting point for describing PBC for MCA simulations in LIGGGHTS. At least the implemented code allowed the simulation of a simple shear loading of cubic samples with PBC in one axis.

### 4.4.10 Implementation of MPI Exchange for Parallelization

MPI is a programming technique to parallelize code over a distributed system, for example a HPC, such that the entire program is parallelized over a network of computers, or nodes which cross-communicate information. This means that every parallel process is working in its own memory space in isolation from the others. OpenMP is also a programming technique for parallelization, but instead of the program running in parallel on a distributed system, it runs on shared memory devices, for example a multi-core processor. This means that every parallel thread has access to all the data and a set of operations could run in parallel over a multi-core processor where the cores share the same memory. Thus, communication is relatively cheap and easy. However, LIGGGHTS does not have an OpenMP package which will be explained in the following section.

The parallelisation in LIGGGHTS has its bases from LAMMPS and is implemented fully using MPI exchange as explained in section 3.3.8 [252]. It can run on a single processor on a regular PC or thousands of processors on HPCs. The domain is decomposed into several MPI processes such that each sub-domain contains its ‘owned’ particles and ‘ghost’ particles owned by neighbouring processes. Each domain calculates the forces for its ‘owned’ particles and based on the cut-off distance each process communicates this information to its ‘ghost’ neighbouring particles. So at each time-step it is clear which particles belong to which processor but they vary over time depending on their new positions [216].

This parallelization method is very efficient, but only when well-balanced domains are defined such that the number of particles within each domain is balanced so that the information exchange happens smoothly. The use of a large number of domains will cause slower computation because the amount of cross-communication will be high; and a very small number of domains will also slow down the computation because the load per processor will be too high. Thus, the optimisation of the numbers of domains and the number of particles in each domain is crucial.

As shown in Figure 4-4, the LIGGGHTS integration loop (simplified) according to the Velocity Verlet Scheme begins by inserting particles and transforming meshes. Particles are exchanged between MPI processes, load balancing may occur, and cross-communication occurs, neighbor lists are built, and forces are computed between the two integration steps *initial_integrate( )* and *final_integrate( ).* The parts of the loop that can be parallelized are shown in 4 separate squares and the color codes indicate the contribution of each section to the total timing breakdown [255]. The main time-consuming parts are the calculation of the forces is the main one (pair time) this actually also includes bond forces, then cross-communication, generating and updating neighbor lists.

Regarding our MCA implementation, because of the common code base, some of the functions related to MPI exchange has been inherited and consequently many positive performance characteristics are inherited from LIGGGHTS. However, because most of the main classes which are the most time-consuming have been drastically re-written (AtomVecMCA, PairMCA, BondMCA, BondCreateMCA) MPI exchange is not fully realised in the code. In each class there are methods related to MPI exchange among processors, including pack_comm, pack_comm_vel, pack_border, and pack_border_vel. but these have not been fully realized yet in LIGGGHTS-MCA. It has also been difficult to amend because LIGGGHTS uses meshes to discretize its domain, which is not the case in MCA, which is why maybe it is best to try and look into LAMMPS for this.

The following list summarizes the sequence of important functions during and after MPI communication [255]:

1. Periodically adjust domain boundaries along one or more threads and communicate data if necessary (Optional).
2. Exchange particles and contact history information between MPI processes.
3. Apply spatial sorting of local particle data to each MPI process.
4. Update particle neighbor lists.
5. Exchange mesh triangles and contact history data between MPI processes.
6. Update mesh triangle–particle neighbor lists.

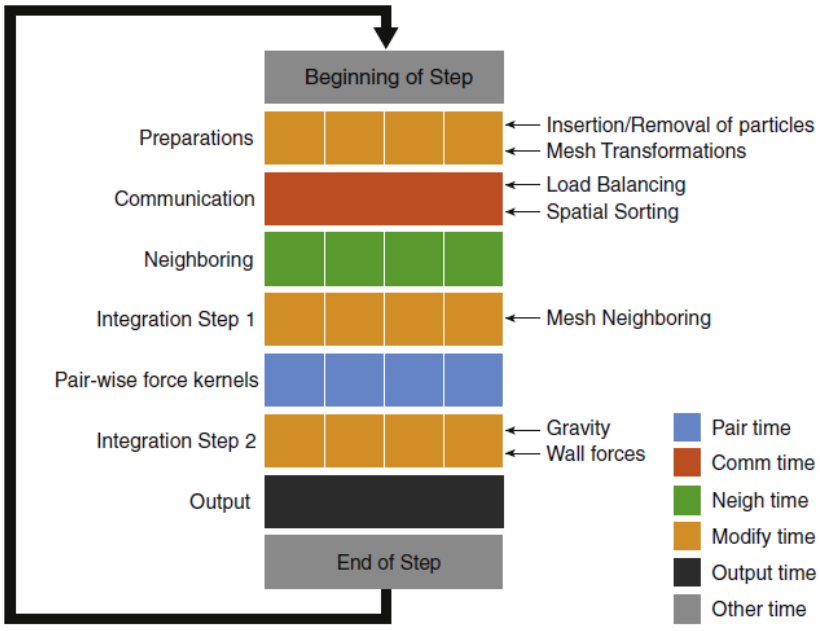



**Figure 4-4: LIGGGHTS integration loop (simplified) according to the Velocity Verlet scheme, showing the parts of the loop that can be parallelized (shown as 4 separate squares) and the color code indicates the contribution of each the total timing breakdown [255]**

### 4.4.11 Implementation of OpenMP for Parallelization

As explained, OpenMP is a different way of parallelization which should provide optimized and multi-threaded versions of many classes of the code. In contrast to MPI, OpenMP allows to compute in parallel on one PC with several cores. Most modern processors have several cores and so it is really useful feature. However, currently there is no OpenMP capability in LIGGGHTS.

OpenMP is based on sequential implementation, which allows the parallelisation of the codes progressively without extensive rewriting of the code. The sequential algorithm runs as a master function that can call subfunctions to distribute the computations on available resources. The master functions retrieve the data from the subfunctions and carries-on. The sequential algorithm is highly efficient for systems requiring long processing time such as in MD and DEM codes.

At the LIGGGHTS short course, Andreas Aigner, one of the developers of LIGGGHTS, said that implementing OpenMP for the new classes will be difficult, that implementing OpenMP has only been done by Richard Berger [260] for only parts of the code implementing a new pair potential and it took him the whole of his PhD to do it. However, after careful consideration and because MPI exchange was difficult to realise, it was decided to try to implement OpenMP for our new classes and it proved to be partially successful.

Figure 4-5 shows an overview of the hybrid MPI/OpenMP parallelization developed by Berger [260]. “The workload of a simulation is first distributed along one or more axes using MPI decomposition. MPI load balancing, which adjusts boundaries over time, can still be used. Each MPI subdomain then further divides its subdomain into partitions of equal workload. All work-intensive algorithms launch multiple threads that work only on particles in a partition assigned to them.”

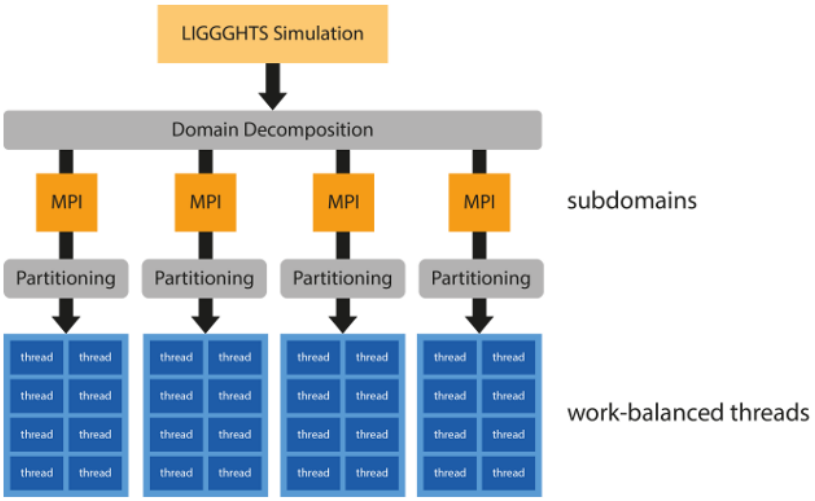



**Figure 4-5: Parallelization in LIGGGHTS [255]**

A similar technique is used to implement OpenMP to the newly developed ‘mca’ code, and was developed by adding OpenMP related code to the following methods because they are the parts that can be parallelized based on Figure 4-4:

- swap_prev( ) → ‘FixMeanStressMCA’ class
- predict_mean_stress( ) → ‘FixMeanStressMCA’ class
- compute_elastic_force( ) →‘PairMCA’ class
- compute_equiv_stress( ) →‘PairMCA’ class
- correct_for_plasticity( ) →‘PairMCA’ class
- compute_total_force( ) →‘PairMCA’ class
- post_integrate( ) → ‘FixBondCreateMCA’ class
- build_bond_index( ) → ‘BondMCA’ class

Then in the ‘Make’ folder of the source code, a file called ‘Makefile.serial_omp’ was added to be able to run the simulations using the OpenMP implemented functions. However, the OpenMP feature is not very efficient, and its efficiency needs to be further tested and optimized. It also contains some bugs but provides an invaluable tool and starting point for running MCA simulations in parallel instead of in series in LIGGGHTS without having to make major changes to the code.

## 4.5 Summary

This chapter so far delivered the third contribution mentioned in section 1.4 of developing and implementing the 3D MCA elastic-plastic model in LIGGGHTS. The following section 4.6 will test the code and perform verification of the model.

The developed code implemented the formulation of MCA described in Chapter 3 section 3.5. New 'mca' commands and styles have been added to LIGGGHTS. The '*atom_style mca*', '*pair_style mca*', '*bond_style mca'* and '*fix nve/mca'* commands all need to be used together in a simulation as one package and defined in the input script and '*fix bond/create/mca'* and *'fix mca/setvelocity'* are optionally used. '*fix bond/exchange/mca'* and *'fix meanstress/mca'* are used by default and do not need to be defined in the input script.

The '*atom_style mca*' defines the particles discretization, generation and attributes according to the description in section 3.5.1. The '*pair_style mca'* defines the particle-particle interactions (i.e. inter-automata interactions) calculations according to section 3.5.2. The '*bond_style mca'* defines the MCA bond model according to section 3.5.3. The '*fix nve/mca*' style defines the integration of equations of motion according to section 3.3.3 incorporating the mca formulations, and the '*fix mca/setvelocity*' style defines velocity boundary conditions for loading.

Looking back at the steps that needed to be implemented mentioned in the summary section 3.6 of Chapter 3:

1. Add MCA particle and interaction forces parameters → in mca atom style
2. Add MCA contact area, mass and inertia tensor →in mca atom style
3. Add normal (pressure) and tangential (shear) elastic force calculations → in mca pair style
4. Add mean stress predictor →in mca pair style and mca fix mean stress style
5. Add rotation, bending and torsion torques →in mca pair style
6. Add parameters and code for plasticity →in mca pair style
7. Add bond breaking and fracture criteria → in mca bond style
8. Add periodic boundary conditions → in mca pair style
9. Implement OpenMP for parallelization
10. Implement MPI exchange for parallelization
11. Add interaction with walls → in mca fix wall/mca style
12. Add bond formation → in mca bond style
13. Add heat generation and transfer → in mca pair style

Steps 1 to 7 have been successfully implemented; which will be verified and validated in the rest of this chapter and the following chapters. Steps 8 and 10 are implemented but are still not working properly and need further testing. Steps 11 to 13 are considered future work of the project explained in Chapter 9.

## 4.6 Verification

At each stage of the development of the code, verification tests were conducted to ensure that the implemented code matches the underlying mathematical model and its solution, and also to check for any computer coding errors. These errors, if not solved, could affect the predictive capability of the tool. To do this, verification is usually done using relatively simple cases which are aimed at checking parts of the model. Validation against experimental results is only done after the whole code is developed, verified and all errors removed, to determine the degree to which the model is an accurate and real representation of the real-world problem.

In [261] verification is defined as: "the procedures in which a model is tested to determine whether it can be made consistent with some set of observations." This means that the model is considered verified if the analytical results matches the simulation results. If they are too far off, then the input parameters should be adjusted by a calibration method and the verification step has to be redone.

As mentioned before, in our case in MCA, no parametric estimation is needed. The macroscopic material properties are used as input parameters and the results demonstrate correct macroscopic response and very close agreement with analytical (presented in this section) and experimental results as well (presented in chapter 5). This is a major advantage of MCA over DEM, and hence the developed code as well if proven to be working correctly according to the described theory in Chapters 3-4.

To verify the developed 3D elastic-plastic MCA model in LIGGGHTS, 3D simulations are conducted mimicking physical tests of solid materials under uni-axial tension and shear using the new implemented 'mca' commands. A set of input parameters are chosen and the mechanical material response is studied. The effect of parameter variations on the material response will then be assessed in the following Chapter 5 as well as validation against experimental results.

### 4.6.1 Computational Environment

LIGGGHTS-MCA can be downloaded from github [262]. As most open-source codes, it is recommended to use a Unix based operative system. You can then compile (i.e. install) it the same way LIGGGHTS is compiled using a command prompt. LIGGGHTS is then executed by reading the commands from the defined input script. After setting up the input script, the simulation can run as a usual LIGGGHTS job, either in serial or in parallel using OMP. When using OMP, the number of desired threads are chosen, if not then by default it will be set to be equal to the number of cores of the computer's processor.

All the simulations presented in this thesis were conducted on a laptop with 8GB RAM memory and 2.10 GHz processor using OpenMP with 6threads. The code has also been used on a Linux CentOS 7 on a PC with 16GB RAM memory and quad core processor at 1.6 GHz and on the Leeds University HPC-ARC3 which has 252 nodes with 24 cores of 128 GB memory each.

### 4.6.2 Uniaxial Tensile and Simple Shear Tests

To test the developed elastic-plastic MCA model, 3D aluminum samples are simulated under uniaxial tension and simple shear loading. The results are compared to the analytical solution and the macroscopic mechanical response of the material is analysed.

So, the following section describes the simulation setup as defined in the input script using the developed MCA commands at relevant parts. In the input script many of LIGGGHTS own commands have been used, but the description below focuses only on the newly developed commands and related commands.

#### 4.6.2.1 Simulation Setup

As explained previously, the first step of the input script is defining the initial structure. Here, a 3D aluminium sample is simulated with a homogenous initial internal structure free of discontinuities (damages or cracks), and all automata are assumed to be initially in contact and bonded. Particles are located on an FCC packing structure, as shown in Figure 4-6.

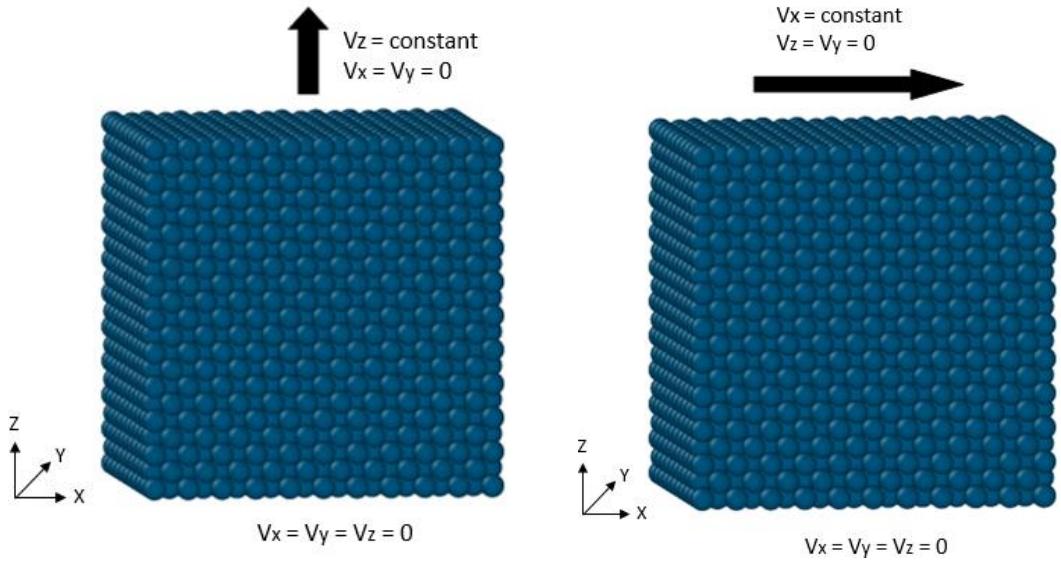



**Figure 4-6: Initial structure of the 3D simulated sample of FCC packing, showing the direction of (a) tension and (b) shear loading**

The initial structure and particle attributes are defined in the '*atom_style mca'* as:

| | |
|---|---|
| ***atom_style*** | ***mca radius 0.254 packing FCC n_bondtypes 1 bonds_per_atom 12*** |

This means that the radius of the particles is equal to *0.254m*. This is an unrealistic particle size, however, it is sufficient for the sake of verification against the analytical solution, but for validation against experimental results and producing true stress/strain values, the particle size must be chosen to be a much smaller size.

Packing is set as *FCC*, the number of bond types is *1* meaning all particles are linked, and the maximum number of bonds per particle is *12* which is the coordination number according to the *FCC* structure.

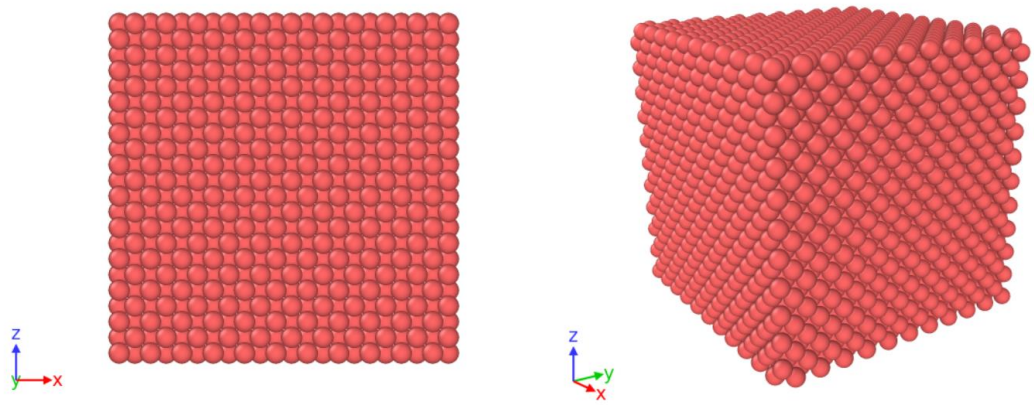
Also, fixed boundary conditions are applied in *x*, *y* and *z* directions so that the particles do not interact across the boundaries and the faces have a fixed position.

Then the simulation domain (specimen size) is defined to be eleven layers of particles, in *x*, *y* and *z* directions to form a cuboid as shown in Figure 4-6. This produces 4631 particles. It is defined in the input script as

| | |
|---|---|
| variable | fcc equal ${d}/0.7071067812 |
| variable | L equal 11*${fcc} |
| region | box block 0 ${L} 0 ${L} 0 ${L} units box |

The dimensions of the sample are defined in terms of multiples of the *FCC* lattice parameter such that a perfect cuboid is simulated, and also to define the layer on which loading conditions will be set. For FCC the lattice parameter ($a$) is defined as $a = r\sqrt{8}$ where ($r$) is the particle radius. To define it in terms of particle diameter $a = d/(\frac{1}{\sqrt{2}})$. If SC lattice structure is used it will just be multiple of its diameter ($L$ equal 11*${$d$}).

The initial structure for the FCC case is shown below in Figure 4-7




(a) (b)

**Figure 4-7: (a) front and (b) side views of the initial structure of sample in FCC packing**

After that the '*pair_style mca'* is set to define the interaction forces laws used in the simulation. It is defined as:

| | |
|---|---|
| ***pair_style*** | ***mca 2*${d}*** |
| ***pair_coeff*** | ***1 1 ${COF} ${G} ${K} ${Sy} ${Eh}*** |

The cut-off distance is set as twice the particle size (diameter). The domain has only one material, so the first two arguments of the '*pair_coeff*' command are set as *1*. The coefficient of friction is set as *0.3*. The shear *${G}* and bulk modulus *${K}* are calculated from the Young Modulus and Poisson Ratio which are set as *70GPa* and *0.3* respectively. For plastic deformation, the yield stress *${Sy}* is set to be *2MPa* and the plastic work hardening modulus *${Eh}* is set as *10GPa*.

Then the 'bond_style mca' and 'fix bond/create/mca' are defined as:

| | |
|---|---|
| ***bond_style*** | ***mca*** |
| ***bond_coeff*** | ***1 1 0.0001*** |
| ***fix*** | ***bondcr all bond/create/mca 1 1 1 2*${d} 1 6*** |

The first argument in the '*bond_coeff*' command is the number of bond types, which is set as *1* here because all automata are linked, the second argument is the choice of fracture criterion which is set as *1* which means it uses the ultimate equivalent strain criterion, and the value of the equivalent strain is 0.0001.

If only elastic deformation is required, *${Sy}* and *${Eh}* are removed from the '*pair_coeff*' command, and only the first argument of the '*bond_coeff*' command needs to be defined to set the number of bond types in the simulation to 1. No fracture criterion is then used because only elastic deformation and no fracture is simulated.

Then the '*fix bond/create/mca'* style is defined where all automata within the cut-off distance are checked every 1 time-step, and all automata are of the same group, the same material, and have a coordination number 6 for SC and for FCC the last argument is set as 12. This is used for elastic and plastic deformation alike.

After that loading is applied to the specimen, mimicking strain-controlled uni-axial tensile and simple shear tests, where the specimen is clamped at the end and pulled/pushed at a constant rate. As shown in Figure 4-6(a) for uni-axially tension, a vertical constant velocity (*Vz)* is applied in the positive z direction (upwards) on the upper row of particles, and zero velocity in the *x* and *y* directions (*Vx)* and (*Vy)*, while applying zero velocity on the lower row of particles in all directions so that the lower row of particles remains fixed and the specimen is uni-axially loaded in the *z* direction. Similar velocity boundary conditions are applied for shear, but as shown in Figure 4-6(a), the horizontal constant velocity (*Vx)* is applied in the

positive *x* direction (to the right) on the upper row of particles instead of in the *z* direction.

Also, a region of top and bottom layer of particles must be defined where the velocities will be applied, such that the velocities are only applied on one layer of particles (the $10^{th}$ layer).

This is defined as:

```
variable    Height equal 10*${fcc}-${rp}
region      top block EDGE EDGE EDGE EDGE ${Height} EDGE units box
region      bot block EDGE EDGE EDGE EDGE EDGE 0.0 units box
```

For SC structure variable Height would just be equal 10*${d}.

Then the velocity boundary conditions are defined by the 'fix mca/setvelocity' command, for tension as:

```
fix         topV_fix top mca/setvelocity 0 0 0.01
fix         botV_fix bot mca/setvelocity 0 0 0
```

and for shear as:

```
fix         topV_fix top mca/setvelocity 0 0.01 0
fix         botV_fix bot mca/setvelocity 0 0 0
```

This means that the constant pull velocity is 0.01m/s in the *z* direction for tension, and in the *x* direction for shear. The strain rate can be calculated as cross-head velocity divided by initial length which here will approximately be $0.001s^{-1}$.

The last important part to define is the time-step, simulation time and run-step.

```
variable    dt equal 1.0E-5
variable    fulltime equal 0.1
variable    runstep equal round(${fulltime}/${dt})
timestep    ${dt}
run         ${runstep}
```

The time-step is set to $\Delta t = 10^{-5}$ seconds, the simulation time to 0.1 seconds, and the number of steps is defined as the simulation time divided by the time-step, which in this case gives 10,000 steps.

Also, a recommended time-step message is shown on screen when running the simulation, so it can be adjusted depending on the simulation. This follows the equation (3.9) presented in section 3.3.4. and can be defined in the input script as:

```
variable    TimeStep equal ${rp}/(sqrt((${K}+4.*${G}/3.)/${rho}))
print 'Recommended time step ${TimeStep}'
```

The simulation parameters are summarized below in Table 4-2.

**Table 4-2: Simulation Parameters**

| **Parameter** | **Value** |
|---|---|
| Density ($\rho$) | 2700 Kg/m3 |
| Young's modulus ($E$) | 70 GPa |
| Poisson's ratio ($v$) | 0.3 |
| Yield stress ($\sigma_y$) | 2 MPa |
| Work hardening modulus ($\varepsilon_h$) | 10 GPa |
| Coefficient of friction ($COF$) | 0.3 |
| Particle diameter ($d$) | 0.254m |
| Number of particles | 4631 |
| Coordination number | 12 (FCC) |
| Number of bond types | 1 (linked) |
| Time-step (Δt) | $1 \times 10^{-5}$ s |
| Full simulation time | 0.1 s |
| Boundary conditions | FFF |
| Pull velocity | 0.01 m/s |

#### 4.6.2.2 Post Processing

After running the simulation, output files are produced and analysed. These outputs files are produced and defined at each time-step and their type, content and location are defined in some commands in the input script. The commands are:

```
variable    savetime equal 0.001
variable    filestep equal round(${savetime}/${dt})
dump        dmp all custom ${filestep} post/dump*.liggghts id type x y z vx vy vz
fx fy fz omegax omegay omegaz tqx tqy tqz
fix         outfile all print ${filestep} "${time} ${px} ${py} ${pz} ${pfx} ${pfy}
${pfz}" file cube.dat screen no title
```

The dump command outputs dump*.liggghts text files which are saved in a folder called post at every time-step. They contain information about the particles positions, velocities, forces, etc. at each time-step. The format of dump*.liggghts files can be directly visualized using the visualization tool Ovito. It shows the positions, velocities, forces, etc. of the particles at each time-step and then can be visualized to view the behaviour from the beginning to the end of the simulation.

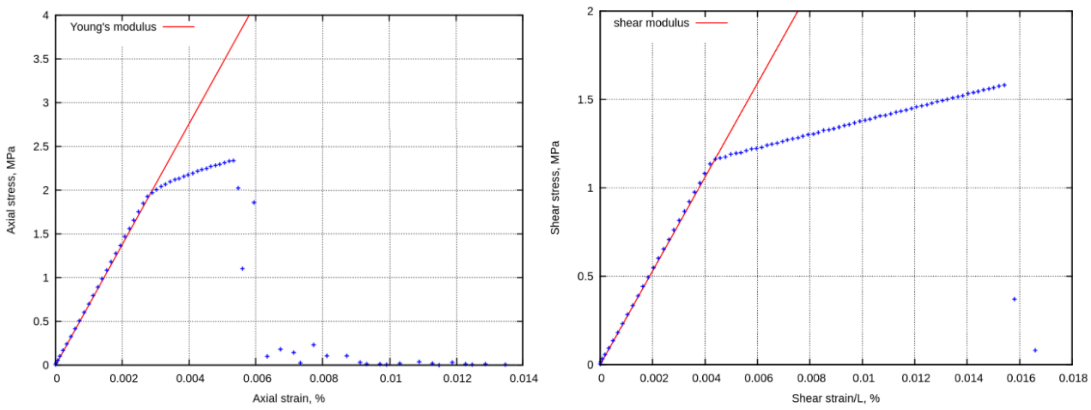
Another valuable visualization tool is Paraview, however the format of dump*.liggghts cannot be directly read by it. To use Paraview, the format has to be changed to VTK format using a conversation tool called pizza.py. After installing this tool the dump files can be read. Instead of outputting these values at every time-step, here they are output at every *${filestep}* which equals 100 steps to save computational time and memory. This can be set to a larger number of steps, depending on the sensitivity of the simulation phenomena, because you do not want to miss any important deformations occurring during the skipped time-steps. However, for our purpose 100 steps is sufficient.

Another, post-processing tool is gnuplot which can plot the stress-strain curves. For this the fix command is used, where it saves a text file called cube.dat which includes the positions and forces in all directions at each *${filestep}*. This is used to plot the stress-strain curves. For the tensile test, the strain is calculated by getting the difference in distance between two particles on each end of the box along the Z (strained) direction ($l - l_0$) and dividing by their initial distance by ($l_0$). This can then be multiplied by 100 to get strain %. The axial force (load) is calculated by the sum of forces on the particles in Z direction, and the axial stress is then calculated by dividing the axial force by the specimen cross section (*A)*. For the Shear test, the values along *X* direction are used instead of the *Z* direction.

The post processing described here is used in the analysis of all the simulations presented in the thesis so it will not be mentioned again unless necessary.

#### 4.6.2.3 Results and Discussion

The results show the macroscopic response simulated using LIGGGHTS-MCA. As mentioned, the stress-strain curves of the simulations were plotted and compared to the analytical solutions. Figure 4-8(a) shows the axial stress for the tensile simulation performed under constant loading conditions as a function of axial strain, and Figure 4-8(b) shows the shear stress for the simple shear simulation performed under constant loading conditions as a function of shear strain. For the uni-axial tensile test the analytical solution is considered to be the Young's Modulus (*E*), and for the uni-axial shear test it is the Shear Modulus (*G*) and they are presented by the red lines, while the simulation results are presented by the blue lines.

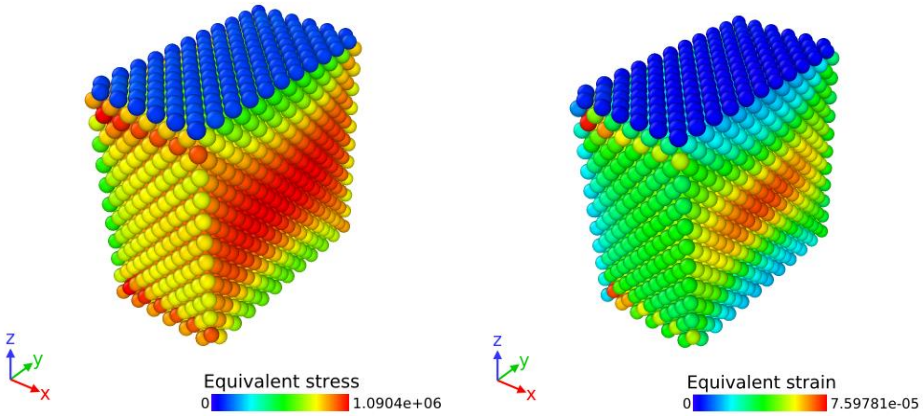



(a) (b)

**Figure 4-8: Stress-strain curve of simulated (a) uniaxial tension test and (b) simple shear test (blue lines) compared to analytical solution (red lines)**

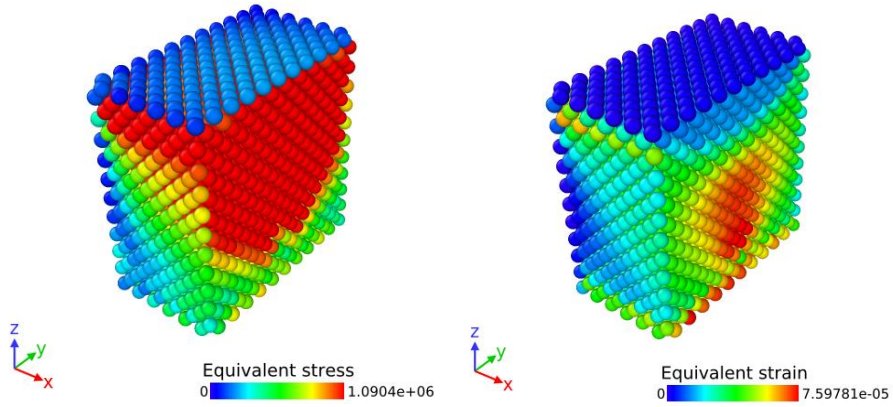
Both tension and shear tests show linear elastic-plastic material deformation behaviour. They show correct macroscopic response in the elastic region and show very close agreement with the analytical solution. Young's modulus and shear modulus of the material after deformation is correctly calculated in the elastic region. Also, as expected the values of the shear forces are lower than the tensile forces. This verifies the implemented normal and tangential elastic interaction forces model.

Figure 4-9 and Figure 4-10 show the elastic and plastic stress and strain fields of the samples of the uniaxial tensile tests visualized using Ovito. Figure 4-9(a) shows the equivalent stress in the elastic region and Figure 4-9(b) shows its corresponding strain field. Figure 4-10(a) shows the equivalent stress in the plastic region and Figure 4-10(b) shows its corresponding strain field. Figure 4-10(b) shows localization of plastic strain in the lower part of the sample because the upper layer of the sample is fixed in XY plane.




(a) (b)

**Figure 4-9: (a) stress and (b) strain fields in the elastic region of the uniaxial tensile test**




(a) (b)

**Figure 4-10: (a) stress and (b) strain fields in the plastic region of the uniaxial tensile test**

The results show how the implementation of the many-body interaction forces form resulted in correct macroscopic response by using direct material mechanical properties as input parameters, which is not possible when only pair-wise interactions are considered such as in DEM. In DEM, many other parameters would have been needed to be defined such as normal and tangential stiffness, coefficient of restitution, etc. which all need calibration and parametric estimation methods as explained in section 3.4.4.

After the yield point, irreversible plastic deformation behaviour is shown, which verifies the implementation of the normal and tangential corrected plastic interaction forces model. It shows the plastic model of the material with piecewise linear hardening. After that fracturing occurs which verifies the implemented bond model and fracture criterion.

### 4.6.3 Uniaxial Tensile and Shear Loading-Unloading Tests

The loading-unloading tests shows the possibility of modelling irreversible deformation, which is also necessary to model indentation and scratching tests later in Chapter 6. Again, the results are compared to the analytical solution and the macroscopic mechanical response of the material is analysed.

#### 4.6.3.1 Simulation Setup

Here, the exact same materials, properties and simulation parameters are used as in the previous uniaxial tensile and shear tests. The only two difference are to turn off the 'bond_style mca' because there is no fracture in this case, and the application of extra loading cycles after the first run of 10,000 steps is done.

This is done by the following commands:

```
# Reverse loading smoothly
variable        vel0 equal 0.01
variable        vel_revers equal ${vel0}/10
unfix           topV_fix
fix             topV_fix top mca/setvelocity 0 0 v_vel_revers
variable        unloadstep equal ${runstep}/2.
run             5000
# Unloading
variable        vel_unload equal -${vel0}
unfix           topV_fix
fix             topV_fix top mca/setvelocity 0 0 v_vel_unload
variable        unloadstep equal ${runstep}/3.
run             ${unloadstep}
```

Variable vel0 is the pull up velocity as used in the previous case which was 0.01m/s. To apply reverse loading smoothly to not cause fracture, this velocity is divided by a factor of 10 to produce a much slower velocity. To apply this on the first layer of particles, the previous fix for loading has to be removed which is done using the unfix command. Then the number of steps for reversing is half that for loading producing 5000 steps, compared to 10,000 steps.

After that unloading is done with the same velocity as the pull velocity but in the opposite direction, and the number of steps for unloading is 1/3 of the full runstep which here will be 3333 steps. Again, the previous fix needs to be unfixed first.

Here only one cycle of loading-unloading is done.

Then the stress-strain curves are produced as described in section 4.6.2.3. evaluate the unloading and reloading paths.

### 4.6.3.2 Results and Discussion

Figure 4-11 shows the stress-strain curve for the uni-axial tensile loading-unloading test by plotting axial stress vs strain % and Figure 4-12 shows the stress-strain curve for the simple shear loading-unloading test plotting shear stress vs strain %. Again, results are compared with the analytical solution which is presented by the linear red line as Young's modulus for tension and shear modulus for shear.

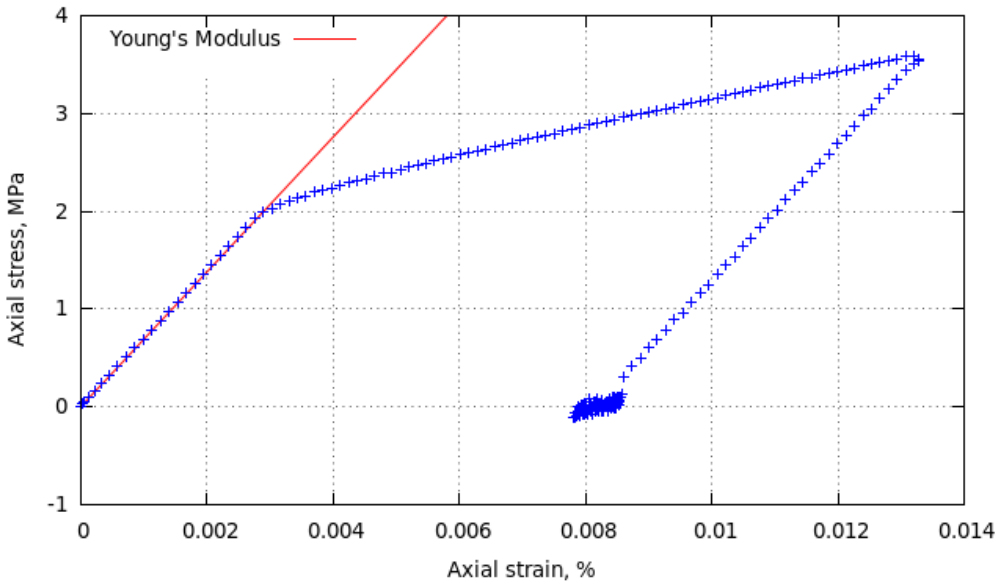



**Figure 4-11: Stress-strain curves of uni-axial tensile loading/unloading test (blue line) compared to analytical solution (red line)**

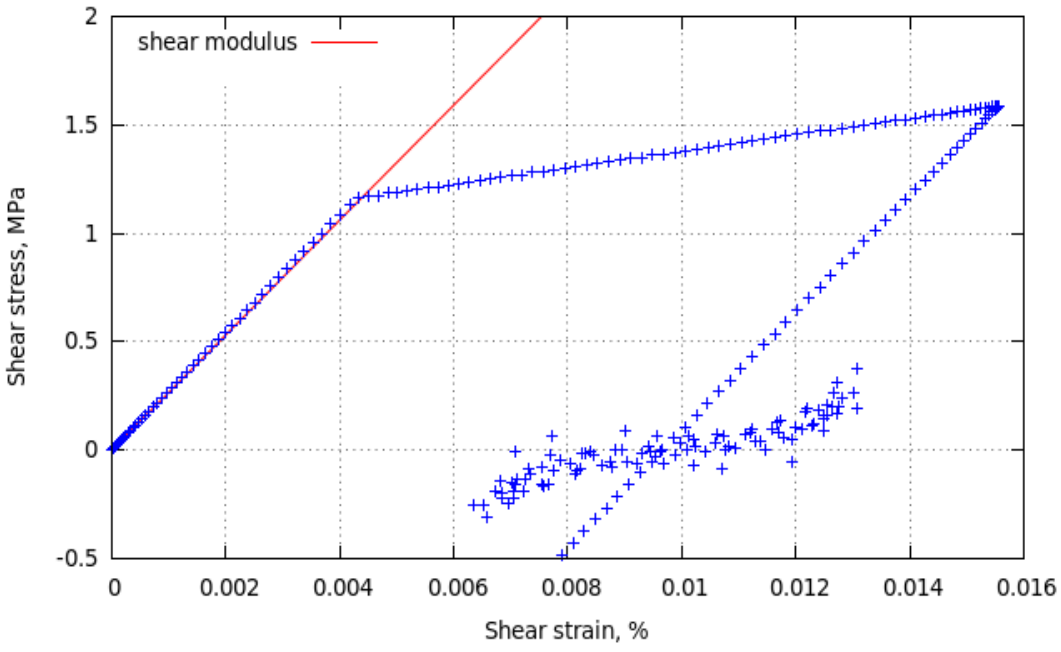



**Figure 4-12: Stress-strain curves of simple shear loading/unloading test (blue line) compared to analytical solution (red line)**

In the case of the uniaxial tensile strength the error between theoretical and actual value is in the order of $10^{-2}$ which can be written as a decimal error or due to the inhomogeneity of natural materials leading slightly different value from the expected perfectly theoretical material. In the case of the simple shear test, the error increases with increasing strain reaching a maximum value of 0.9%. This error can be attributed to the complex stress state that shear forces impose on a material. In other words, the error is most likely due to the same reason as in the uni-axial tensile test, however, due to the complex stress state, the error is slightly magnified.

The unloading at the end of the test shows the elastic recovery of the material even after severe deformation. The slope of the elastic recovery is also equivalent to the Young's modulus of the material. This is the expected unloading path that a material with elastic-potential will go through when going through loading and unloading tests. Both graphs also show some data scatter at the end of the loading, this is due to the oscillation of the loading layer because the velocity of the moving layer at the last step is not exactly equal to zero. Also, after plastic deformation occurs the structure has some residual stresses inside, which also results in some dynamic motion to minimize these residual stresses.

Both tension and shear tests show very close agreement with the analytical solution in the elastic region and show a typical mechanical response for loading-unloading tests, which means the fundamental equations of the model are capable of capturing the deformation behaviour of materials. The comparison between different plastic behaviour of the material is shown and discussed in the following chapter.

### 4.6.4 Conclusion

To conclude, section 4.6 presented cases for testing the correct implementation of the code in LIGGGHTS by running uni-axial tensile and simple shear simulations on an aluminium sample and checking the macroscopic response of the material after loading against analytical solutions. The new 'mca' commands have been used and the verification tests show that the code has been successfully implemented in LIGGGHTS. The results show very good agreement with the analytical solution and show correct macroscopic response. The simulations results produced by the model provided a very close match to analytical solutions.

The next step is to validate the developed model against experimental results and test the convergence and performance of the code.

# Chapter 5
# Validation, Convergence & Parameters Sensitivity Analysis

## 5.1 Introduction

This chapter and the next investigate the predictive capabilities of the developed 3D MCA model in producing the mechanical response of solid materials under loading with direct use of the mechanical material properties as inputs, which are the Young's modulus, Poisson's ratio, yields stress and strain work hardening modulus.

Uni-axial tensile and micro-indentation tests have been simulated mimicking the physical laboratory tests presented in a study performed by the US National Institute of Standards and Technology (NIST) [263]. In their work, they present the results for both uni-axial tensile and micro-hardness tests including micro-indentation of Aluminium 6061 samples. Their study is presented in the paper "High throughput exploration of process-property linkages in Al-6061 using instrumented spherical micro-indentation and microstructurally graded samples" in 2016 [264].

The goal of the NIST study was to establish a relationship between the macroscopic scale standardised tensile tests and the mesoscale spherical micro-indentation stress-strain measurements, which is generally difficult to do because of the gap in length scales, as shown in Figure 5-1. Thus, they conducted both experiments on selected samples to explore the correlations between the measured properties in the two different types of tests.

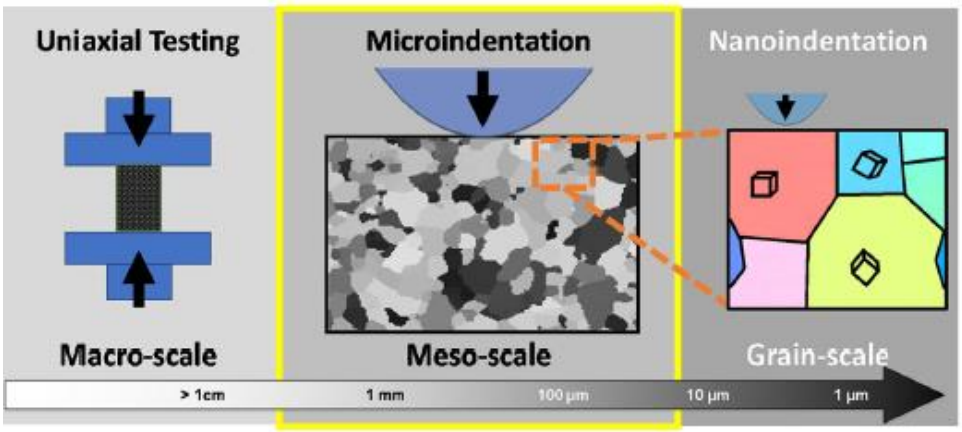



**Figure 5-1: Schematic for different methods of extracting mechanical response of material at different scales as presented by [264]**

The reason for choosing this experimental study to validate our developed MCA model against is because the NIST provided a detailed study as well as results on their website [263], and produced stress-strain curves for both uni-axial tensile test and micro-indentation tests for the same materials. Since the simulation of micro-indentation is, in part, critically dependent on identification of the appropriate

material mechanical properties used as input parameters, the uni-axial tensile tests have first been simulated and validated against the experimental results presented by the NIST study which will be presented in the following section 5.2, and then the material mechanical properties are used in the simulation of micro-indentation to validate against their micro-indentation experimental results, which is presented in Chapter 6.

This chapter also used the same data of the uni-axial tensile tests to perform parameter sensitivity analysis on some of the implemented features and convergence testing of the code. The convergence testing is important to determine some of the sources of errors of the simulated results and the steps that can be taken to reduce the error to an acceptable level. The reduction of error involves increasing simulation time, because of larger number of particles. Hence for applying the model to the study of a specific application, it is important to consider the acceptable level of error so that simulations can run as efficiently as possible.

## 5.2 Validation of MCA Model Against Experimental Uniaxial Tensile Test

### 5.2.1 Experimental Data

In the NIST study [264] uni-axial tensile tests were conducted on Aluminium 6061 samples with different aging parameters. All the specimens were received in the T6 condition and then grounded and polished to 0.06-µm colloidal silica with a final step of electro-polishing. The uniaxial tensile tests were conducted as per ASTM Standard E8-13a with a constant cross head speed to produce a strain-rate of 0.005 $s^{-1}$. The specimens had a diameter of 6.35 mm and a gage length of 25.4 mm.

Their results are shown in Figure 5-2 as stress-strain curves for the samples as received (AR) and at different aging temperatures.

To simulate the tensile test of the aluminium samples presented above for validation, first the mechanical material input parameters have to be defined to be included in the input script. To test the predictive capability of the model to simulate different material deformation behaviour; namely the yielding and the strain hardening, three of the different samples at different aging temperatures from the experimental data shown in Figure 5-2 are simulated. These are the as received sample (AR), the sample at 274°C aging temperature and at 413°C aging temperature. The mechanical response of AR shows almost an elastic-perfectly-plastic behaviour, while the samples at 274°C and at 413°C aging temperatures show elastic-plastic behaviour with linear strain hardening at different slopes.

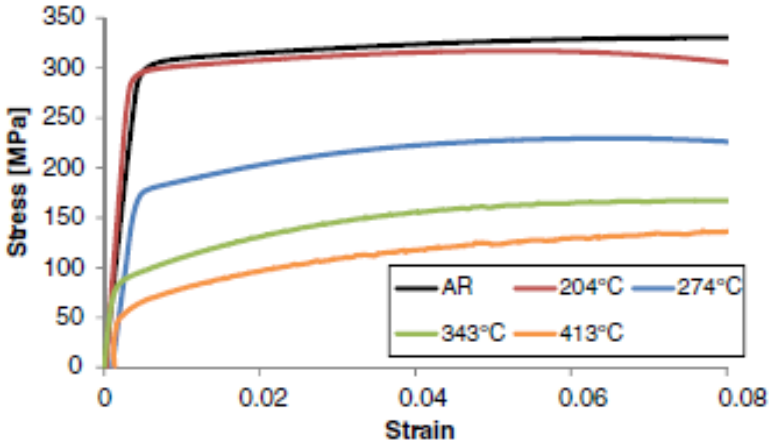



**Figure 5-2: Experimental tensile stress-strain curves for AL-6061-T6 samples as-received (AR) and at different aging temperatures as presented by [264]**

As mentioned in Chapter 4, the four material parameters needed are Young's Modulus ($E$) and Poisson's Ratio ($\nu$) - to calculate the Shear ($G$) and Bulk properties ($K$) - for the calculation of the elastic interaction forces, and Yield Strength ($\sigma_y$) and Strain Work Hardening Modulus ($E_h$) for plastic interaction forces. These are all

commercially known for the material aluminium 6061-T6, but they can also all be directly obtained from the experimental stress-strain curves above in Figure 5-2 to ensure that the same material is being simulated for the comparison between the experimental and simulation mechanical response and material deformation.

For the AR sample, the yield strength ($\sigma_y$) is set as 315 MPa. Young's Modulus ($E$) is obtained from the slope of the curve in the elastic region before the yield limit and is set as 69 GPa. The strain work hardening modulus ($E_h$) can be estimated as the slope of the curve between the yield limit and the strength limit as shown earlier in Figure 3-4 in Chapter 3. So, it can be calculated as

$$E_h = \frac{\sigma_u - \sigma_y}{\varepsilon_u - \varepsilon_y}$$

where ($\sigma_u$) is the ultimate stress, ($\varepsilon_u$) is the ultimate strain and ($\varepsilon_y$) is the strain at yield ($\sigma_y/E$). The ultimate stress and strain can be extracted from the graphs in Figure 5-2, and the strain at yield can also be calculated as ($\sigma_y/E$) for a more accurate estimation. Their values are shown below in Table 5-1, and thus strain work hardening modulus ($E_h$) for AR is calculated and set as 225 MPa.

For the sample aged at 274°C, the yield strength ($\sigma_y$) is set as 189 MPa, the Young's Modulus ($E$) is set as 40 GPa, and strain work hardening modulus ($E_h$) is set as 816.95 MPa. For the sample aged at 413°C, the yield strength ($\sigma_y$) is set as 55 MPa, Young's Modulus ($E$) is set as 66.95 GPa, and strain work hardening modulus ($E_h$) is set as 1076.18 MPa. Poisson's ratio has been set to 0.33 for all three samples and the density to $\rho$=2700 kg/$m^3$.

**Table 5-1: Material mechanical properties extracted from experimental stress—strain curves**

| | **E (GPa)** | **$\sigma_y$ (MPa)** | **$\sigma_u$ (MPa)** | **$\varepsilon_y$** | **$\varepsilon_u$** | **$E_h$ (MPa)** |
|---|---|---|---|---|---|---|
| **AR** | 69 | 315 | 329 | 0.00456 | 0.08 | 225 |
| **274°C** | 40 | 189 | 230.47 | 0.0101 | 0.08 | 816.95 |
| **413°C** | 66.95 | 55 | 138.189 | 0.0027 | 0.08 | 1076.18 |

### 5.2.2 Simulation Setup

After the mechanical properties of the materials are estimated, they can be set in the input script along with the other simulation parameters as described in section 4.6.2.1. The input script for the AR material is shown in the following section 5.2.3.

Cubic samples are simulated with initial FCC structure and homogeneous, free of discontinuities, and all automata are assumed to be initially bonded. The particle size is 0.5 mm and the sample size is 10 mm × 10 mm × 10 mm.

The first top layer of particles is pulled upwards in the Z direction (loading direction) with a pull velocity 0.01m/s while the bottom layer is kept constant. The time step is $4 \times 10^{-6}$ s, and the simulation time is 0.01s to produce a strain rate of about $0.001s^{-1}$. The fracture criterion used here is the ultimate equivalent strain and is set to 0.2. The simulation parameters are shown in Table 5-2, and in the input script in the following section.

**Table 5-2: Input Parameters used in the tensile simulation for validation against exp results**

| **Parameter** | **Values for AR** | **Values for aged at 274°C** | **Values for aged at 413°C** |
|---|---|---|---|
| Density ($\rho$, kg/m$^3$) | 2700 | 2700 | 2700 |
| Young's Modulus (*E, GPa*) | 69 | 40 | 66.95 |
| Yield Stress ($\sigma_y$, MPa) | 315 | 189 | 55 |
| Work Hardening Modulus ($\varepsilon_h, MPa$) | 225 | 816.95 | 1076.18 |
| Poisson's Ratio ($v$) | 0.33 | 0.33 | 0.33 |
| Coefficient of friction (*COF*) | 0.3 | 0.3 | 0.3 |
| Sample Size (mm$^{3)}$ | 10×10×10 | 10×10×10 | 10×10×10 |
| Particle Diameter ($d$) | 0.5 | 0.5 | 0.5 |
| Coordination Number | 12 (FCC) | 12 (FCC) | 12 (FCC) |
| Number of bond types | 1 (linked) | 1 (linked) | 1 (linked) |
| Time-step (Δt, s) | $4 \times 10^{-6}$ | $4 \times 10^{-6}$ | $4 \times 10^{-6}$ |
| Full simulation time (s) | 0.01 | 0.01 | 0.01 |
| Boundary Conditions | FFF | FFF | FFF |
| Pull Velocity (mm/s) | 0.1 | 0.1 | 0.1 |

### 5.2.3 Input Script

The input script for this simulation as described in the previous section is shown below for the AR material. The MCA related commands are highlighted in bold. It is exactly the same for the sample aged at 274°C and 413°C, the only difference is the first section where the material parameters are defined.

---

```
# Material Parameters (Aluminium 6061-T6 as recieved)
variable      rho equal 2700
variable      Y equal 6.9e10
variable      p equal 0.33
variable      G equal $Y/(2*(1+$p))
variable      K equal $Y/(3*(1-2.0*$p))
variable      COF equal 0.3
variable      Sy equal 315e6
variable      Eh equal 225e6

# Atom Parameters
variable      nat equal 1
variable      rp equal 0.00025
variable      d equal 2*${rp}
variable      bt equal 1
variable      bpa equal 12
variable      fcc equal ${d}/0.7071067812

# ------------------------ INITIALIZATION ----------------------------
dimension     3
units         si
boundary      fff
atom_style    mca radius ${rp} packing fcc n_bondtypes ${bt} bonds_per_atom
${bpa}
atom_modify   map array
neigh_modify  delay 0
newton        off
communicate   single vel yes

# ------------------------ Particle DEFINITION ----------------------
variable      L equal 14.41*${fcc}
region        box block 0 ${L} 0 ${L} 0 ${L} units box
create_box    ${nat} box
lattice       fcc ${fcc}
create_atoms  1 region box


# Discretization Parameters
variable      skin equal 2*${d}
neighbor      ${skin} bin
timestep      1.0e-9

# ------------------------ FORCE FIELDS -----------------------------
pair_style    mca ${skin}
pair_coeff    1 1 ${COF} ${G} ${K} ${Sy} ${Eh}
bond_style    mca
bond_coeff    1 -1 1 0.2

mass          1 1.0

# ------------------------ SETTINGS ---------------------------------
# Define Velocity Boundary Conditions
```

```
variable      Height equal 14.41*${fcc}-${d}
region        top block EDGE EDGE EDGE EDGE ${Height} EDGE units box
region        bot block EDGE EDGE EDGE EDGE EDGE 0.0 units box
group         top region top
group         bot region bot

# Loading Parameters
variable      vel_up equal 0.01
variable      vel_down equal 0
fix           topV_fix top mca/setvelocity NULL NULL v_vel_up
fix           botV_fix bot mca/setvelocity NULL NULL v_vel_down

# Create Bonds Between Particles
Variable      cutoff equal ${d}*(1.0+0.02)
fix           bondcr all bond/create/mca 1 1 1 ${cutoff} 1 ${bpa}

# Time Integration
group         nve_group region box
fix           integr nve_group nve/mca

# Simulation Time Parameters
variable      TimeStep equal ${rp}/(sqrt((${K}+4.*${G}/3.)/${rho}))
print         'Recommended time step ${TimeStep}'
variable      dt equal 4.e-6
if            "${TimeStep} < ${dt}" then &
  "print 'Recommended time step ${TimeStep} is smaller than dt= ${dt}'" & quit
variable      fulltime equal 0.1
variable      runstep equal round(${fulltime}/${dt})
variable      savetime equal 0.001
variable      filestep equal round(${savetime}/${dt})
timestep      ${dt}

# Output Settings
thermo_style         custom step atoms
thermo               100
thermo_modify        lost ignore norm no
compute_modify       thermo_temp dynamic yes
dump                 dmp all custom ${filetime} post/dump*.liggghts id type x
y z vx vy vz fx fy fz omegax omegay omegaz tqx tqy tqz

# Plot Output Parameters
variable             mytime equal time
variable             px equal xcm(top,x)
variable             py equal xcm(top,y)
variable             pz equal xcm(top,z)
variable             pfx equal f_topV_fix[1]
variable             pfy equal f_topV_fix[2]
variable             pfz equal f_topV_fix[3]
fix                  outfile all print ${filestep} "${mytime} ${px} ${py}
${pz} ${pfx} ${pfy} ${pfz}" file cube.dat screen no title

# ------------------------- RUN ---------------------------------
fix_modify    bondcr every 0

run    ${runstep}
print  "All done"
```

### 5.2.4 Results and Discussion

For the above described simulation, the stress-strain curves are calculated for all three material samples; as received, at 274°C and at 413°C aging temperature, and the results are plotted against the experimental results from [264] as shown in Figure 5-3.

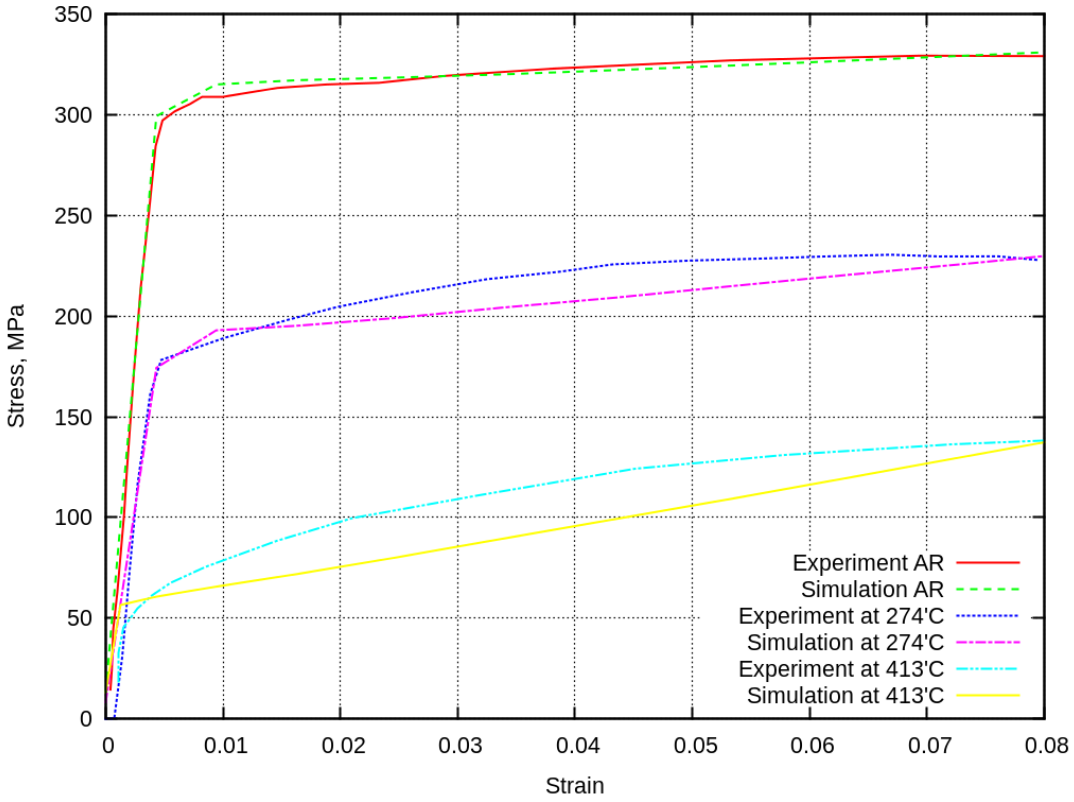



**Figure 5-3: Stress-strain curve for all 3 samples: simulation vs experimental results**

The comparison with the experimental result in Figure 5-3 shows that the model as currently implemented is capable of accurately reproducing results from physical tests such as tensile tests which are commonly used for determining the strength and ductility of materials. However, there is a degree of error.

The experimental result for the AR sample shows elastic-perfectly-plastic behaviour, while the simulation result shows larger plastic deformation. For this case, a solution could be to simulate a larger sample size to avoid the influence of the boundaries on the simulation results, especially from the fixed bottom.

The experimental results for both samples at 274°C and at 413°C aging temperature show elastic-plastic behaviour, however the plastic behaviour is small and linear as is expected due to the accepted model for hardening.

Since the model only describes linear-hardening behaviour which uses the yield stress ($\sigma_y$) and the ultimate stress ($\sigma_u$) to the strain work hardening modulus ($E_h$), the ultimate tensile strength of the material cannot be accurately predicted. This is

particularly visible in the sample of at 413°C aging temperature since it is the material with the highest plastic deformation. In the case of the AR aluminium, since the material has a perfectly plastic response, the model is capable of capturing the material behaviour extremely accurately, except in the transition phase from elastic to plastic deformation. Although, even in this case, the error is negligible and is within range of material inhomogeneity.

For the simulation results here 0.08 is used as the value of the ultimate strain ($\varepsilon_u$) as shown in Table 5-1. A better solution could be achieved by a better fit of the strain work hardening modulus ($E_h$) by using ultimate strain ($\varepsilon_u$) of 0.02 for example.

Another solution could be implementing a multi-linear hardening behaviour or breaking down the plastic deformation region into smaller sections with different strain hardening moduli.

Of course, another source of error could be that the experimental specimens have a diameter of 6.35 mm and a gage length of 25.4 mm, while the simulated samples are 10 mm x 10 mm x10 mm. However, the code presents limitations to simulate the actual size of the experimental specimens and keep the particle size small enough due to the incomplete parallelization of the code which makes this hard to do.

As mentioned in section 4.6.1, all the simulations were conducted on a laptop with 8GB RAM memory and 2.10 GHz processor using OpenMP with 6 numbers of threads. A solution is to run the simulations on the HCP. The code has been tested and works on the HCP however, it still shows some errors due to the incomplete implementation of the MPI exchange.

In general, the results show that the model is capable of accurately simulating the elastic-plastic behaviour of continuous material by directly using macroscopic mechanical properties of materials as inputs of the simulation. This is not possible using DEM models where input parameters include parameters that cannot be obtained experimentally or not easily obtained as shown in Table 3-1 in Chapter 3 and explained section 3.4.4.

## 5.3 Convergence Analysis

This section presents a convergence analysis of the model, where the simulation results are tested with an increase in particle size and dimensions. If the model has been implemented correctly, the different simulations will have similar results.

### 5.3.1 Change in Sample Size

Here the effect of sample size and particle numbers on the mechanical response is investigated. Three different simulation sizes were simulated as shown in Figure 5-4 in their initial state. All the simulation parameters are kept the same except for the increase in the size of the sample twice and four times in all directions.

The sample sizes are 10 mm × 10 mm × 10 mm, 20 mm × 20 mm × 20 mm and 40 mm × 40 mm × 40 mm which produce 12,195, 92,597 and 740,772 particle sizes respectively.

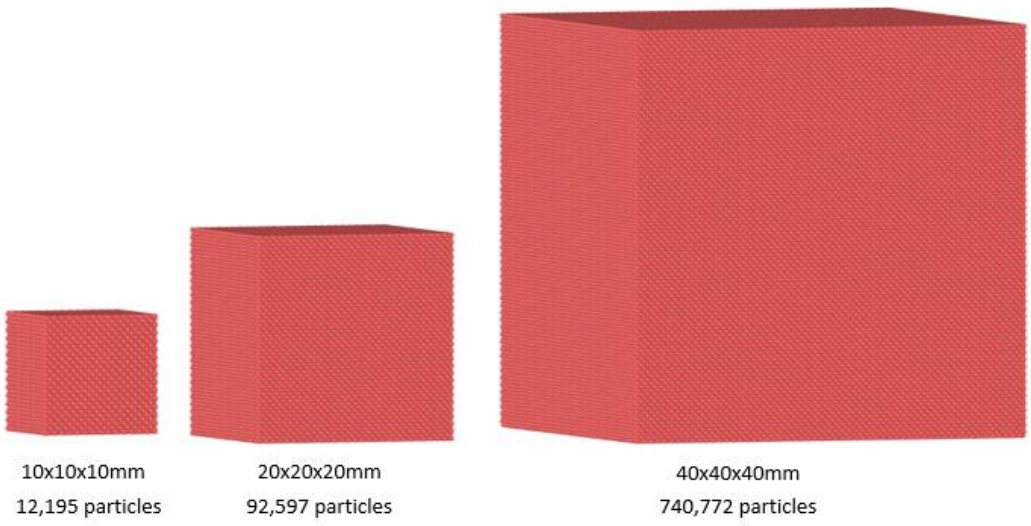



**Figure 5-4: Initial structure of samples with three different sizes**

The results of this convergence test are shown in Figure 5-5.

The main noticeable difference is that with larger sample size the yield stress increased for all the samples. This is likely due to the higher strain rate applied on the larger specimen. Another, reason could be due to the ratio of time-step to strain rate where the increase in strain rate leads to an overshoot of the actual yield stress. This however could not be tested due to computational limitations. Again, a solution is to run the simulations on the HCP, to apply lower pull velocity on the samples and simulate higher strain rates. Beyond yielding, the plastic behaviour is not influenced by the effect of particle size which shows very good convergence of result.

The strain rate in tensile or compression tests have a great effect on the mechanical properties/response of materials, specifically the yield limit. For example, with increasing strain rates the yield stress of a material is expected to increase, and it also influences strain hardening. Any explicit loading rate effects on the yield limit are not explicitly included in the study because the pull velocity is kept the same (0.1 m/s) for all the samples. To increase strain a lot more time-steps are required which is a limitation of the code due to non-complete implementation of the parallelization.

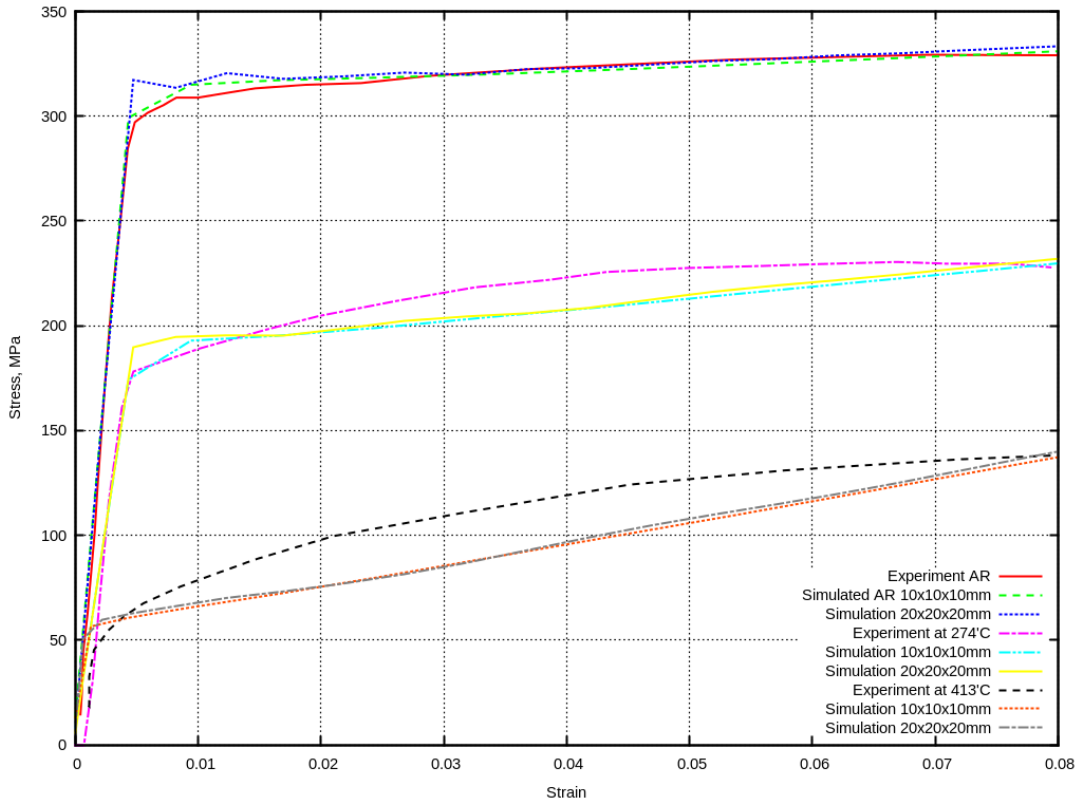



**Figure 5-5: Stress-strain curve for all 3 samples showing the effect of increase in sample size and particle numbers**

To really study the scale effect in crystalline materials, the bevaciour should be observed at the nano-scale where the automata will represent different parts of the crystal grains or grain boundaries. To simulate materials at this scale correctly it is required to introduce the influence of the structure of separate crystals and free/grain boundaries on the inter-automata forces. In the case of plastic deformation, it is also required to account for slip planes similar to the crystal plasticity technique used in FEM.

### 5.3.2 Change in Sample Aspect Ratio

Here the effect of sample aspect ratio on the mechanical response is investigated. Three different samples with different aspect ratios were simulated as shown in Figure 5-6 in their initial state. All the simulation parameters are kept the same except for the increase in the height of the sample twice and four times.

The sample sizes are 10 mm × 10 mm × 10 mm, 10 mm × 10 mm × 20 mm and 10 mm ×10 mm × 40 mm which produce 12,195, 24,389 and 48,778 particle sizes respectively.

The results of this convergence test are shown in Figure 5-7.

Again, similar to before, the velocity in the simulation has not been changed for the different sample sizes, this will influence the applied strain rate on the material. Here, the yield stress is also higher in the larger aspect ratios. This could be attributed to same reasons of strain-rate and time ratio explained in the previous section. At higher loading rate more pronounced effect of elastic waves propagating across the sample is also expected. Furthermore, due to reflection and refraction of these waves on the free boundaries artificial oscillations near the yielding point can be observed. For example, we can see that this effect is much less for the samples with less free surface shown in **Error! Reference source not found.** than in **Error! Reference source not found.**.

Furthermore, the plastic behaviour, though following the same trend, seems to be cyclic. This could be due to the same ratio of time step to strain rate influencing the results where the change in length and the change in cross-sectional is offset. This is seen as a decrease in the calculated stress and strain followed by an increase when the deformation is equivalent again.

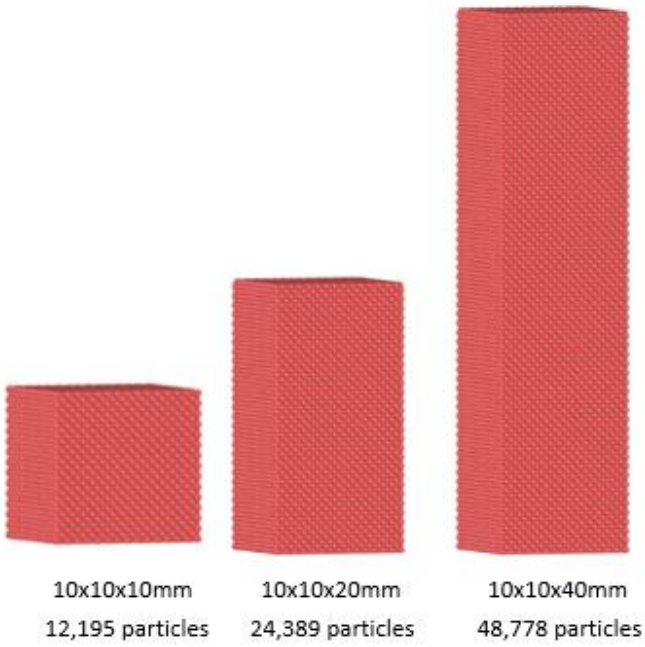



**Figure 5-6: Initial structure of samples with three different aspect ratios**

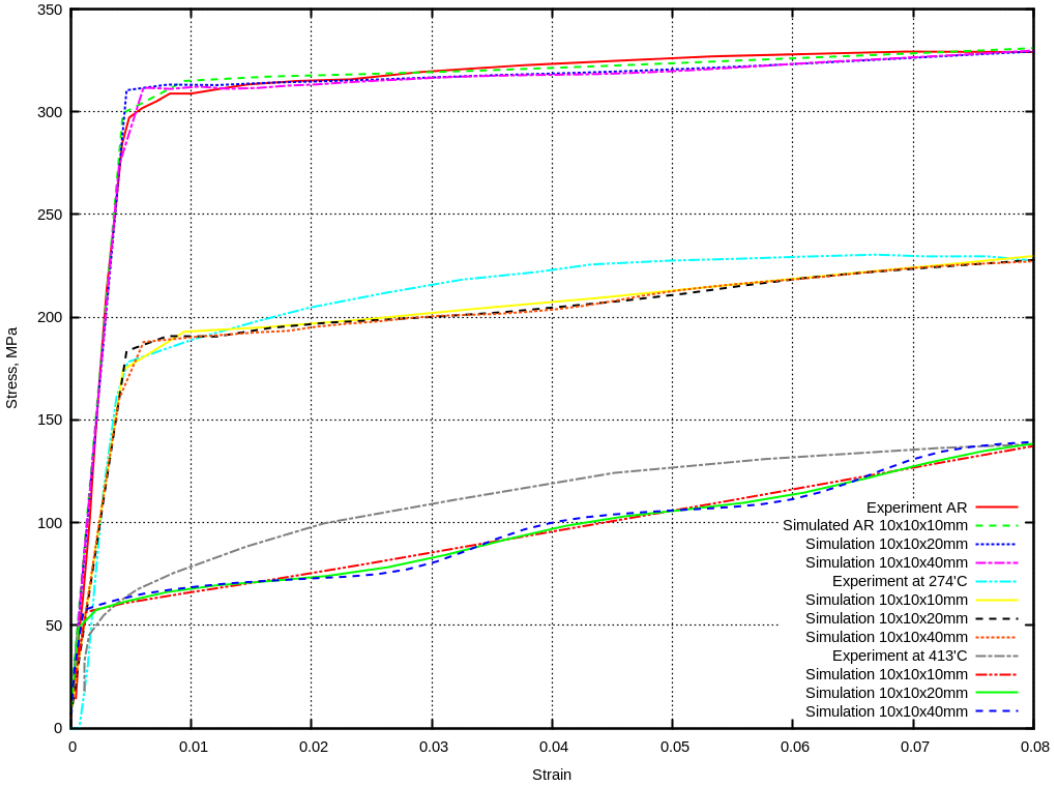



**Figure 5-7: Stress-strain curve for all 3 samples showing the effect of increase in sample aspect ratio**

## 5.4 Performance

In this section the performance of the model is investigated. Table 5-3 shows the simulation time that was taken to run each of the samples mentioned in the previous section. The timings are taken from the time the first output file was produced until the last one was produced at the last time-step. The results are shown in Figure 5-8.

As expected, increasing the number of particles in the simulation drastically increases the computational time. The time should linearly increase with the number of particles with the same number of time steps, which is what the results show.

Excessive performance and scalability tests have not been carried out yet and considered as future work. However, the results show linear increase of time with number of particles that means no unpredictable losing of time for access memory for huge samples.

**Table 5-3: Simulation time for the different sample sizes**

| Sample size (mm) | Nr of particles | Simulation time (min) |
|---|---|---|
| 10×10×10 | 12,195 | 4 |
| 10×10×20 | 24,389 | 13 |
| 10×10×40 | 48,778 | 30 |
| 20×20×20 | 92,597 | 73 |
| 40×40×40 | 740,772 | 600 |

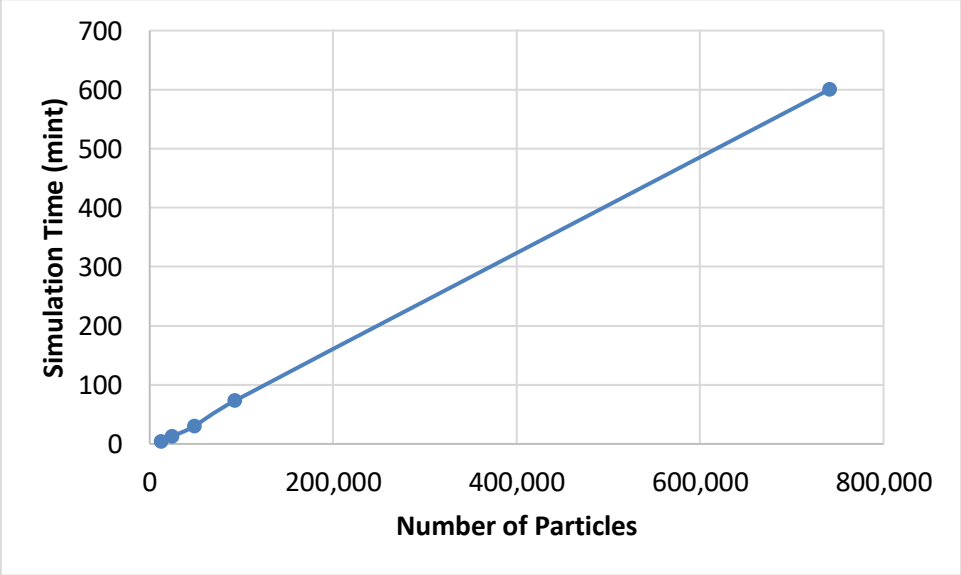



**Figure 5-8: Simulation time vs number of particles showing performance of the code**

## 5.5 Parameters Sensitivity Analysis

In this section, a few simulation parameters are investigated which are related to the developed code. Here we show the difference between using SC packing defined in the ‘atom_style mca’ command and using FCC packing. Also periodic boundary conditions are tests.

### 5.5.1 Simple Cubic vs Face-Centred Lattice Structure

According to the MCA method, particles can be packed in SC, FCC or HCP lattice structure. Only SC and FCC have been implemented in the code. However, MCA is not an atomic scale method, and this does not represent the materials atomic structure but is considered as a method of particle scattering in space. The use of different particle packings in the same simulation is not possible because the calculations of forces depend on the packing and all the automata initially have the same particle size.

In most MCA studies in literature it has been recommended to use the FCC packing, especially when modelling plastic deformation which is why FCC has been used in all the simulations presented in this thesis. However, we would like to investigate the accuracy of the SC packing and its effect on the macroscopic material response because SC packing have much lower particle numbers and this could be helpful in our case because of the computational restriction. It would be much easier to use SC instead of FCC, it will save a lot of computational time and large simulations could be conducted but as shown in Figure 5-9, SC packing only gives an accurate response in the elastic region.

The results below are shown for all the three different materials with the same simulation parameters as described before and shown in Table 5-2. Only difference is using the SC packing structure for the initial structure of the sample instead of FCC. The SC packing

The results for SC packing show relatively accurate results in the elastic region but not in the plastic region. In general, this could be attributed to reason that SC has 1 particle per unit cell, with a coordination number 6, which means that there are 6 number of particles a single atom within a unit cell. So, each particle in the SC structure has 6 neighbouring particles. About 52% of the volume of the cube in an SC structure consists of the volume of the particles and 48% is empty space.

FCC structures have 4 particles per unit cell, with a coordination number 12. FCC is the most efficient cubic structure because the atoms are arranged in such a way that they occupy 74% of the volume of the cube, and only 26% is un-used/empty space.

So, the atoms take up pretty much the whole space and maximize the use of the volume of the cube. This is why it is also known as the cubic closest packed structure, because it is the most efficient use of space when the atoms are arranged.

Thus, the inaccurate result for SC could be attributed to the smaller number of neighbours to interact with and the loosely packed structure. However, in our case, the error in the results for SC is even higher than expected. This could be due to an error in the implementation of the code for SC packing in the estimation of equivalent stress as an “average tangential force” (for sc there are smaller distances to neighbours and hence the low accuracy)

On the other hand, SC requires less simulation time and is about three times faster than FCC simulations because of less number of particles for the same simulation size and because it produces relatively accurate results for elastic deformation, it could be used in simulating elastic or less densely packed materials such as brittle, ceramics or porous materials. In this case, the SC simulation has 8,000 particles instead of 12,195 particles in the case of FCC structure for the same simulation size of 10x10x10mm.

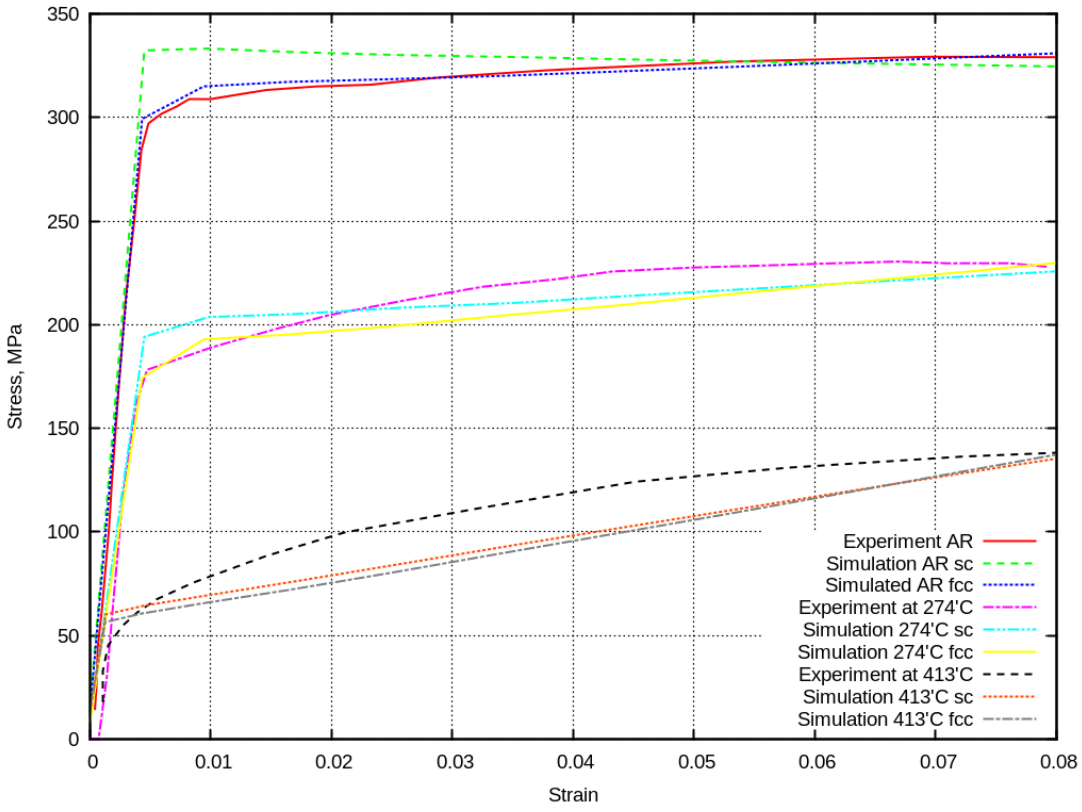



**Figure 5-9: Stress-strain curve for the 3 samples showing particle packing SC vs FCC**

### 5.5.2 Fixed vs Periodic Boundary Conditions

As mentioned, all the simulations so far used non-periodic fixed boundary conditions in all direction (f f f). The three materials were simulated using the periodic boundary conditions by applying them in the *x* and *y* directions but fixed in the z directions which is the direction of loading. When applied tension in *z* direction these conditions actually correspond to the loading in a compression wave where elastic modulus is $K + 4/3\,G$ that much higher than Young's modulus, and equivalent strain increase much higher than the component $\varepsilon_{zz}$.

The results are presented below in Figure 5-10. The sensitivity of the simulation to the periodic boundary effects is clearly shown which is due to the incomplete implementation of this feature in the model as explained in section 4.4.9.

The PBCs as implemented using LIGGGHTS functions forces the particles positions to be inside the cell (within the cut-off distance) by wrapping their positions if they move outside the boundary which makes them appear on the other side of the domain. This causes abrupt changing in position and causes the velocity of the particle to be inconsistent with the change in position within the time-step.

This is believed to cause the artificial effects shown in Figure 5-10., of course in combination with the incomplete implementation.

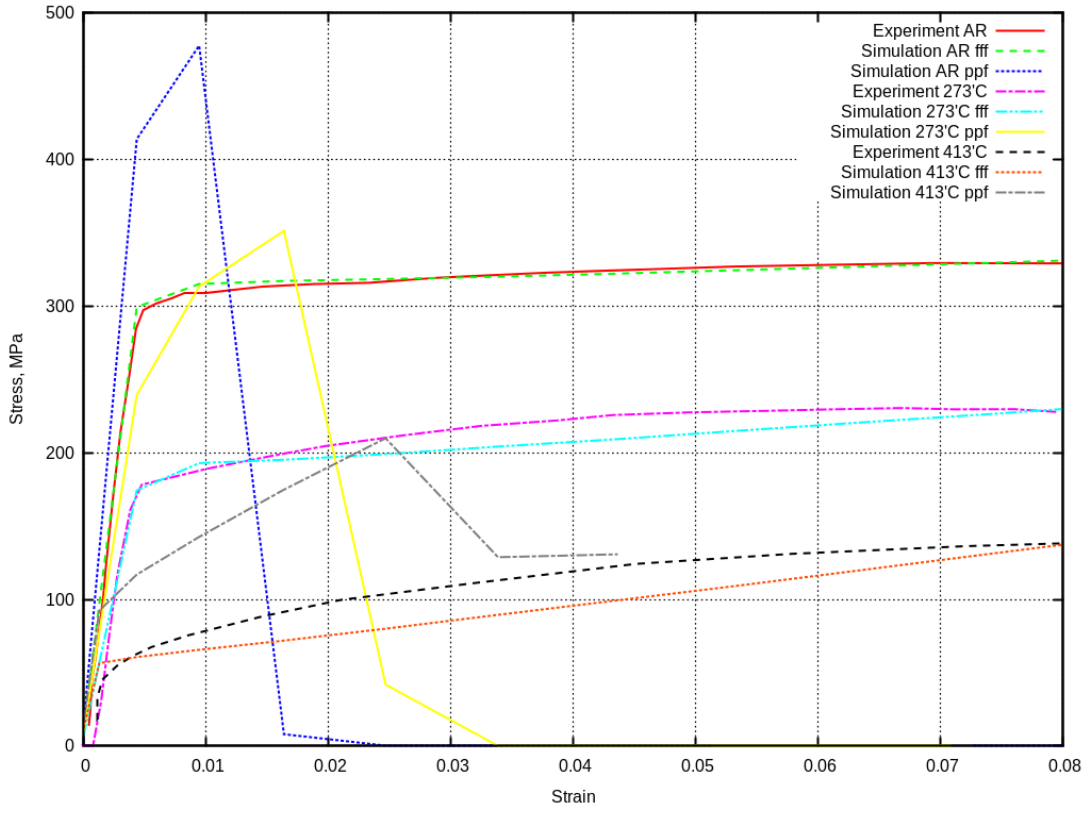



**Figure 5-10: Stress-strain curve for all 3 samples showing fff vs ppf boundary conditions**

## 5.6 Conclusion

This chapter delivered the fourth contribution mentioned in section 1.4 of validation of the code against experimental data and investigating the sensitivity, convergence and performance of the model and code.

The implementation of the MCA model described in chapter 3 has been validated by direct simulation of tensile tests and comparing the results against experimental data. The tensile test is a destructive test to determine strength and ductility of a material. To successfully model a continuous material in a simulation environment a real tensile test is required to assess the performance of the model.

This chapter showed that the model is capable of accurately predicting the elastic-plastic behaviour of continuous materials by directly using macroscopic mechanical properties of materials as inputs of the simulation. This is not possible using DEM models where input parameters include parameters that cannot be obtained experimentally or not easily obtained as shown in Table 3-1 in Chapter 3 and explained section 3.4.4. A distinguishing feature of the MCA method is calculation of forces acting on automata within the framework of many-particle interaction [5], which provides for an isotropic behaviour of a simulated material regarded as a consolidated body rather than a granular medium. This was shown in the results presented in this chapter.

At the current stage of development of code, the code still faces some limitations mainly in simulating smaller length scales and larger simulations. This is due to the incomplete parallelization of the code as explained in sections 4.4.10 and 4.4.11 of chapter 4. Also, periodic boundary conditions and SC packing do not produce accurate results and are in need of further development and testing.

# Chapter 6
# Micro-Indentation Validation & Case Studies

## 6.1 Introduction

One of the uses of indentation and scratching experimental tests is to accurately measure friction coefficients and wear topography of contacts in the lab. Thus, here we are interested in testing the modelling in the simulation of these contact cases.

Nanoindentation is a powerful technique to determine various mechanical properties of thin layers and also for contact analysis. Unfortunately, with the current limitation of computational power of the implemented code, it is not possible to simulate nano-indentation, instead here micro-indentation is simulated.

As explained in section 5.1, micro-indentation simulations have been modelled to validate against the micro-indentation experimental results conducted by the NIST study in [264]. The micro-indentation test is simulated to show the capability of the implemented model to predict plastic deformation in contact interaction.

## 6.2 Validation of MCA Model Against Micro-Indentation

The microhardness test is similar to a Vickers indentation hardness measurement but on a relatively smaller scale. Micro-indentation method measures loads and displacements on the micro-scale allowing the measurement of many material properties at the macroscale such as elastic modulus, hardness, creep, viscoelastic properties, stress relaxation, fracture toughness, interfacial adhesion etc. [265], [266].

Typically, a very sharp and hard Berkovich diamond micro-indentor is used [267]. The tip is lowered into the material causing elastic deformation at the beginning then plastic deformation. The load is then removed. The load and deformation data are measured, and presented by the loading curve, as shown in Figure 6-1, provide the elastic and plastic properties of the material. The information obtained during the loading curve provides information about plastic and elastic deformations; while the unloading curve shows the elastic recovery in the surface [268]. To determine the micro-hardness maximum indentation load ( $P_{indent(\max)}$), maximum penetration depth ($h_{max}$) and elastic stiffness at unloading ($S$) must be measured. The accuracy of calculated micro-hardness depends on our capability to experimentally evaluate these parameters. Which is why the NIST study and others keep developing new protocols for measuring them.

Another parameter, relevant for micro-indentation measurements, is ($h_f$), which represents the depth of the indent after the unloading [268] shown in Figure 6-1, which is the final depth. To calculate the hardness, the contact depth ($h_c$) has to be determined first. It presents the depth of the indenter in contact with the sample under the applied load. The total penetration depth ($h_{max}$) includes not only ($h_c$) but also the deformation of the sample around the indenter ($h_s$).

The micro-hardness is computed similar to the hardness as:

$$H_{micro} = \frac{P_{indent}}{A_{indent}}$$

where ($P_{indent}$) is the maximum indentation loading and ($A_{indent}$) is the indentation area [269].

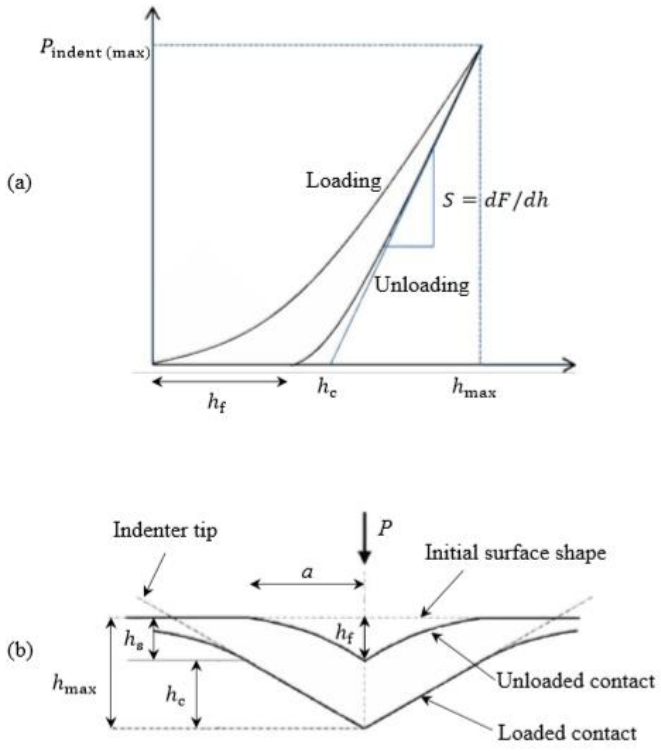



**Figure 6-1: (a) Load vs penetration depth presented as load and unloading curves for a typical micro-indenter test (b) Schematic representation of indenter tip and indentation [269]**

### 6.2.1 Experimental Data

In the NIST study [264] uni-axial tensile tests were conducted on Aluminium 6061 samples with different aging parameters as shown before in section 5.2.1.

They used a spherical tungsten-carbide tip indenter of radius 6.35 mm. The indentation was performed with a constant crosshead speed of 0.1 mm/min with incremental unloading, which is about 50–30 % of the peak force, and reloading cycles until the specified number of cycles was reached.

Their results are shown in Figure 6-2 in terms of loading-displacement curve for the AR material and at the different aging temperatures as presented before.

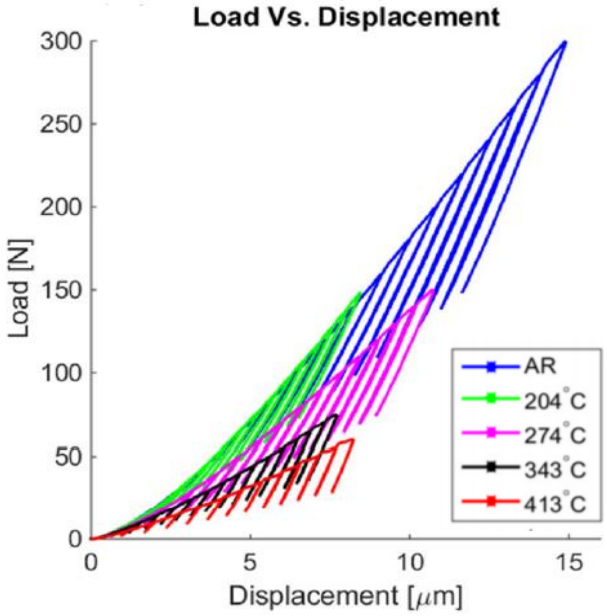



**Figure 6-2: Experimental micro-indentation load-displacement curves for AL-6061-T6 samples as-received (AR) and at different aging temperatures as presented by [264]**

### 6.2.2 Simulation Setup

To simulate the micro-indentation test of the aluminium samples presented above for validation, the mechanical material input parameters from the tensile experiments presented before as shown in Table 5-2 are used. The values of these material parameters have been validated in section 5.2 and produced quite good agreement with the experimental results - especially for the sample as received (AR) - which should help reduce the error in simulating the micro-indentation test. The aluminium samples are cubes of size 7.5 mm × 7.5 mm × 7.5 mm.

The use of the exact mechanical properties for the tungsten-carbide indenter as input is not as significant as the aluminium because it is a much harder material. The material properties of the simulated indenter are assumed to be $\rho$=2800 kg/$m^3$, E=9x$10^5$ MPa, $\nu$=0.18, $\sigma_y$=2x$10^{10}$ MPa and $E_h$=10 GPa.

All particles in the simulation have the same radius of 12.7 µm and the simulation time-step is $10^{-8}$ s. Also, again FCC packing is used here and the initial structures of both aluminium samples and indenter are assumed to be homogeneous, free of discontinuities, and all automata are assumed to be initially bonded.

The size of the spherical indenter is 6.35 mm in the experimental study, and the penetration depth is about 10 µm, which means that Hertz contact radius is about 350 µm.

The shape of the simulated indenter should be a spherical one of size 6.35 mm, as used in the NIST study, which was first simulated as an example shown in Figure 6-3 but not according to actual size. Simulating the actual size of the indenter and sample similar to the experimental study proved to be difficult because to achieve the penetration depth of 10 μm with a spherical indenter of indenter 6.35 mm, the sample size will have to be relatively large. For example, to indent at least 5 layers of particles (which is the minimum for producing a good plot) to a depth of 10 μm and achieve similar plastic deformation as with the Hertz contact radius of 350 μm, the width of the sample will have to be $N \approx 2 \cdot 350 / \left(\frac{10}{5}\right)^3 \approx 42.8 \cdot 10^6$ particles. The current computational limitation of the code will not allow to model this.

To solve this problem, only a part of the spherical indenter is simulated. This is done by simulating a lens-shaped indenter (segment of the sphere) of radius about 400 μm curved such that it forms a contacting part of the sphere of the real indenter radius of 6.35 mm. Figure 6-4 shows the initial structure using the disk indenter according to the proper simulated size.

In terms of loading, the aluminium sample is fixed in place with zero velocity in all directions. Of course, fixed boundary conditions are used (f f f). For the indenter the loading setup of the indenter for simulating micro-indentation resembles the behaviour of the loading-unloading tensile test presented in section 4.6.3. but loading in the opposite direction. Here the indenter moves with a velocity of 0.5 m/s downwards in the z direction while the aluminium sample stays in place for 3000 steps. Then reverse loading is applied by pulling the indenter upwards in the opposite direction for 2500 steps, and then the indenter is moved again upwards by 600 steps.

This is done for all three aluminium samples; as received (AR), at 274°C and at 413°C aging temperatures. The simulation setup is described for the AR sample and is shown in the input script in the following section.

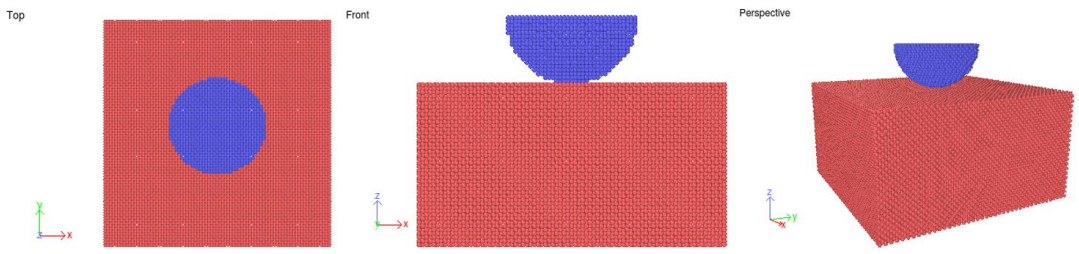



**Figure 6-3: An example of the initial structure of the micro-indentation model with a full spherical indenter but not according to actual size**

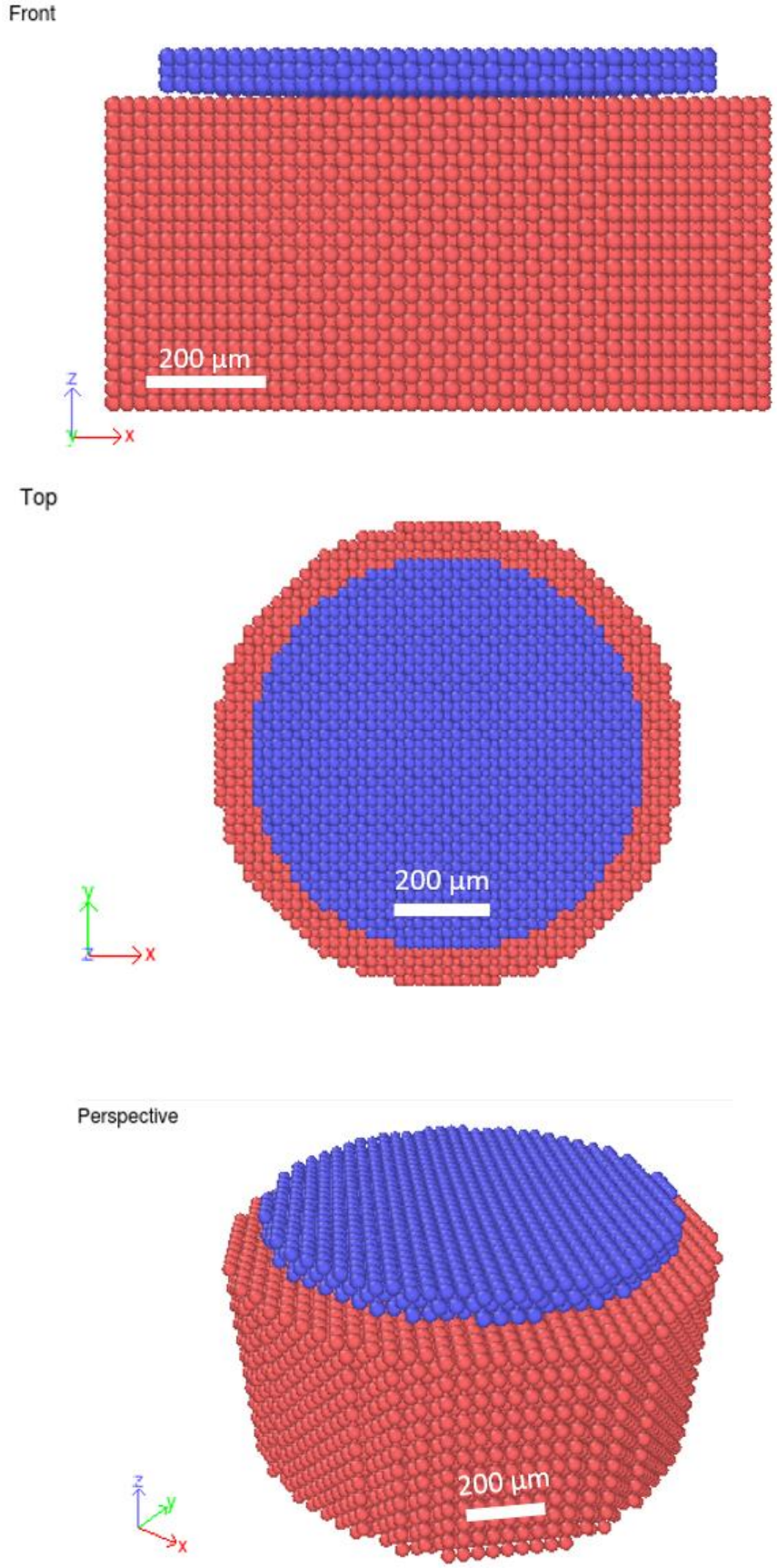



**Figure 6-4: Initial Structure of the micro-indentation model with a lens-shaped indenter mimicking actual contact size and penetration of the experimental study**

### 6.2.3 Input Script

```
# Material Parameters
# aluminium NIST AR
variable      rho equal 2700
variable      Y   equal 6.9e10
variable      p   equal 0.33
variable      G   equal ${Y}/(2*(1+${p}))
variable      K   equal ${Y}/(3*(1-2.0*${p}))
variable      COF equal 0.3
variable      Sy equal 315e6
variable      Eh equal 225e6

# diamond
variable      rho2 equal 2800
variable      Y2   equal 9e11
variable      p2   equal 0.18
variable      G2   equal ${Y2}/(2*(1+${p2}))
variable      K2   equal ${Y2}/(3*(1-2.0*${p2}))
variable      COF2 equal 0.3
variable      Sy2   equal 2.0e16
variable      Eh2   equal 1e10

# Atom Parameters
variable      nat equal 2
variable      rp equal 0.254*5e-5
variable      d   equal 2*${rp}
variable      bt equal 3
variable      bpa equal 20

# ------------------------ INITIALIZATION ----------------------------
dimension     3
units         si
boundary      f f f
atom_style    mca radius ${rp} packing fcc n_bondtypes ${bt} bonds_per_atom
${bpa}
atom_modify   map array
neigh_modify  delay 0
newton        off
communicate   single vel yes

# ----------------------- Particle DEFINITION ----------------------
#sample
variable      latparam equal ${d}/0.7071067812
lattice       fcc ${latparam}
variable      L equal 10*${latparam}+${rp}
variable      2L equal 2*$L

#indenter
variable      Nrings equal 10
variable      X equal 12*${latparam}+${rp}
variable      Y equal ${Nrings}*${latparam}+${rp}
variable      Z equal 1*${latparam}+${rp}
variable      Zhi equal ${L}+1*${latparam}+${rp}

region        box block -${2L} ${2L} -${2L} ${2L} -${2L} ${2L} units box
create_box    ${nat} box bond/types 3
region        BaseBox cylinder z 0 0 ${X} -${Z} ${L} units box
create_atoms  1 region BaseBox
region        indenter cylinder z 0 0 ${Y} ${L} ${Zhi} units box
create_atoms  2 region indenter
```

```
group         Ind region indenter

variable      Height equal ${Zhi}-${latparam}
region        top block EDGE EDGE EDGE EDGE ${Height} EDGE units box
region        bot block EDGE EDGE EDGE EDGE EDGE -${rp} units box
group         top region indenter  #top - this is better for deformed indenter
group         bot region bot

# -- Moving indenter particles to get the shape of the real ball indenter ---
variable      Rind equal 6.35e-3
variable      Rr0 equal ${Nrings}*${latparam}+${rp}
variable      Hr0 equal ${Rr0}*${Rr0}/(2*${Rind})
displace_atoms Ind move 0 0 ${Hr0} units box

variable      Rr1 equal (${Nrings}-1)*${latparam}+${rp}
variable      Hr1 equal ${Rr1}*${Rr1}/(2*${Rind})
region        ring1 cylinder z 0 0 ${Rr1} ${L} ${2L} units box
group         Ring1 region ring1
variable delta equal ${Hr1}-${Hr0}
displace_atoms Ring1 move 0 0 ${delta} units box

variable      Rr2 equal (${Nrings}-2)*${latparam}+${rp}
variable      Hr2 equal ${Rr2}*${Rr2}/(2*${Rind})
region        ring2 cylinder z 0 0 ${Rr2} ${L} ${2L} units box
group         Ring2 region ring2
variable delta equal ${Hr2}-${Hr1}
displace_atoms Ring2 move 0 0 ${delta} units box

variable      Rr3 equal (${Nrings}-3)*${latparam}+${rp}
variable      Hr3 equal ${Rr3}*${Rr3}/(2*${Rind})
region        ring3 cylinder z 0 0 ${Rr3} ${L} ${2L} units box
group         Ring3 region ring3
variable delta equal ${Hr3}-${Hr2}
displace_atoms Ring3 move 0 0 ${delta} units box

variable      Rr4 equal (${Nrings}-4)*${latparam}+${rp}
variable      Hr4 equal ${Rr4}*${Rr4}/(2*${Rind})
region        ring4 cylinder z 0 0 ${Rr4} ${L} ${2L} units box
group         Ring4 region ring4
variable delta equal ${Hr4}-${Hr3}
displace_atoms Ring4 move 0 0 ${delta} units box

variable      Rr5 equal (${Nrings}-5)*${latparam}+${rp}
variable      Hr5 equal ${Rr5}*${Rr5}/(2*${Rind})
region        ring5 cylinder z 0 0 ${Rr5} ${L} ${2L} units box
group         Ring5 region ring5
variable delta equal ${Hr5}-${Hr4}
displace_atoms Ring5 move 0 0 ${delta} units box

variable      Rr6 equal (${Nrings}-6)*${latparam}+${rp}
variable      Hr6 equal ${Rr6}*${Rr6}/(2*${Rind})
region        ring6 cylinder z 0 0 ${Rr6} ${L} ${2L} units box
group         Ring6 region ring6
variable delta equal ${Hr6}-${Hr5}
displace_atoms Ring6 move 0 0 ${delta} units box

variable      Rr7 equal (${Nrings}-7)*${latparam}+${rp}
variable      Hr7 equal ${Rr7}*${Rr7}/(2*${Rind})
region        ring7 cylinder z 0 0 ${Rr7} ${L} ${2L} units box
group         Ring7 region ring7
variable delta equal ${Hr7}-${Hr6}
displace_atoms Ring7 move 0 0 ${delta} units box
```

```
variable      Rr8 equal (${Nrings}-8)*${latparam}+${rp}
variable      Hr8 equal ${Rr8}*${Rr8}/(2*${Rind})
region        ring8 cylinder z 0 0 ${Rr8} ${L} ${2L} units box
group         Ring8 region ring8
variable delta equal ${Hr8}-${Hr7}
displace_atoms Ring8 move 0 0 ${delta} units box

variable      Rr9 equal (${Nrings}-9)*${latparam}+${rp}
variable      Hr9 equal ${Rr9}*${Rr9}/(2*${Rind})
region        ring9 cylinder z 0 0 ${Rr9} ${L} ${2L} units box
group         Ring9 region ring9
variable delta equal ${Hr9}-${Hr8}
displace_atoms Ring9 move 0 0 ${delta} units box

# Discretization Parameters
variable      skin equal 2*${d}
neighbor      ${skin} bin
timestep      5.0e-15

# ------------------------ FORCE FIELDS -----------------------------
pair_style    mca ${skin}
pair_style    mca ${skin}
pair_coeff    1 1 ${COF} ${G} ${K} ${Sy} ${Eh}
pair_coeff    2 2 ${COF2} ${G2} ${K2} ${Sy2} ${Eh2}

bond_style    mca
bond_coeff    *

mass          1 1.0
mass          2 1.0

group         Base region BaseBox
set           group Base mol 1
set           group Base density ${rho}
set           group Ind mol 2
set           group Ind density ${rho2}

# ------------------------ SETTINGS ---------------------------------
# Loading parameters
variable      v_vel_up equal -0.5
variable      ts equal 10000.0*1.E-4*5e-5
variable      vel_up equal ${vel0}*(1.0-exp(-
(2.4*time/${ts})*(2.4*time/${ts})))
variable      vel_down equal 0.0
fix           topV_fix top mca/setvelocity 0 0 v_vel_up
fix           botV_fix bot mca/setvelocity 0 0 v_vel_down

# Create Bonds Between Particles
variable      cutoff equal ${d}*(1.0+0.02)
fix           bondcr1_1 all bond/create/mca 1 1 1 ${cutoff} 1 ${bpa}
fix           bondcr1_2 all bond/create/mca 1 1 2 ${cutoff} 2 ${bpa} state 1
fix           bondcr2_2 all bond/create/mca 1 2 2 ${cutoff} 3 ${bpa}

# Time Integration
group         nve_group region box
fix           integr nve_group nve/mca

# Simulation Time Parameters
variable      dt equal 2.e-4*5e-5
variable      TimeStep equal ${rp}/(sqrt((${K}+4.*${G}/3.)/${rho}))
print 'Recommended time step for aluminium = ${TimeStep}'
if "${TimeStep} < ${dt}" then &
```

```
  "print 'Recommended time step ${TimeStep} is smaller than dt= ${dt}'" &
  quit
variable      TimeStep2 equal ${rp}/(sqrt((${K2}+4.*${G2}/3.)/${rho2}))
print 'Recommended time step for diamond = ${TimeStep2}'
if "${TimeStep2} < ${dt}" then &
  "print 'Recommended time step for diamond ${TimeStep} is smaller than dt=
${dt}'" &
  quit
variable      fulltime equal 3e-5
variable      runstep equal ${fulltime}/${dt}
variable      savetime equal 5e-7
variable      filestep equal ${savetime}/${dt}
timestep      ${dt}

# Output Settings
thermo_style         custom step atoms
thermo               100
thermo_modify        lost ignore norm no
compute_modify       thermo_temp dynamic yes

dump                 dmp all custom ${filestep} post/dump*.liggghts id type x
y z vx vy vz fx fy fz omegax omegay omegaz tqx tqy tqz

# Plot Output Parameters
variable      mytime equal time    # step*${dt}
variable      px equal xcm(top,x)  #x[${lastone}]
variable      py equal xcm(top,y)  #y[${lastone}]
variable      pz equal xcm(top,z)  #z[${lastone}]
variable      pfx equal f_topV_fix[1]     #fx[${lastone}]
variable      pfy equal f_topV_fix[2]     #fy[${lastone}]
variable      pfz equal f_topV_fix[3]     #fz[${lastone}]
fix           outfile all print ${filestep} "${mytime} ${px} ${py} ${pz}
${pfx} ${pfy} ${pfz}" file cube.dat screen no title “


# ------------------------- RUN ---------------------------------
run           1
fix_modify    bondcr1_1 every 0
fix_modify    bondcr2_2 every 0
run           ${runstep}

# Reverse loading smoothly
variable      vel_revers equal ${vel0}*cos((time-${fulltime})*2.0*PI/${ts})
unfix         topV_fix
fix           topV_fix top mca/setvelocity 0 0 v_vel_revers
variable      revstep equal ${ts}/(2.0*${dt})
run           ${revstep}

# Unloading
variable      vel_unload equal -${vel0}
unfix         topV_fix
fix           topV_fix top mca/setvelocity 0 0 v_vel_unload
variable      unloadstep equal ${runstep}/5.
run           ${unloadstep}

quit
```

### 6.2.4 Results and Discussion

Figure 6-5 shows a cross-section view of the simulated micro-indentation model with the disk-shaped indenter after loading to a penetration depth of 10 µm.

Figure 6-6 shows the loading-displacement curve for all three samples and comparing them against the experimental results shown in Figure 6-2.

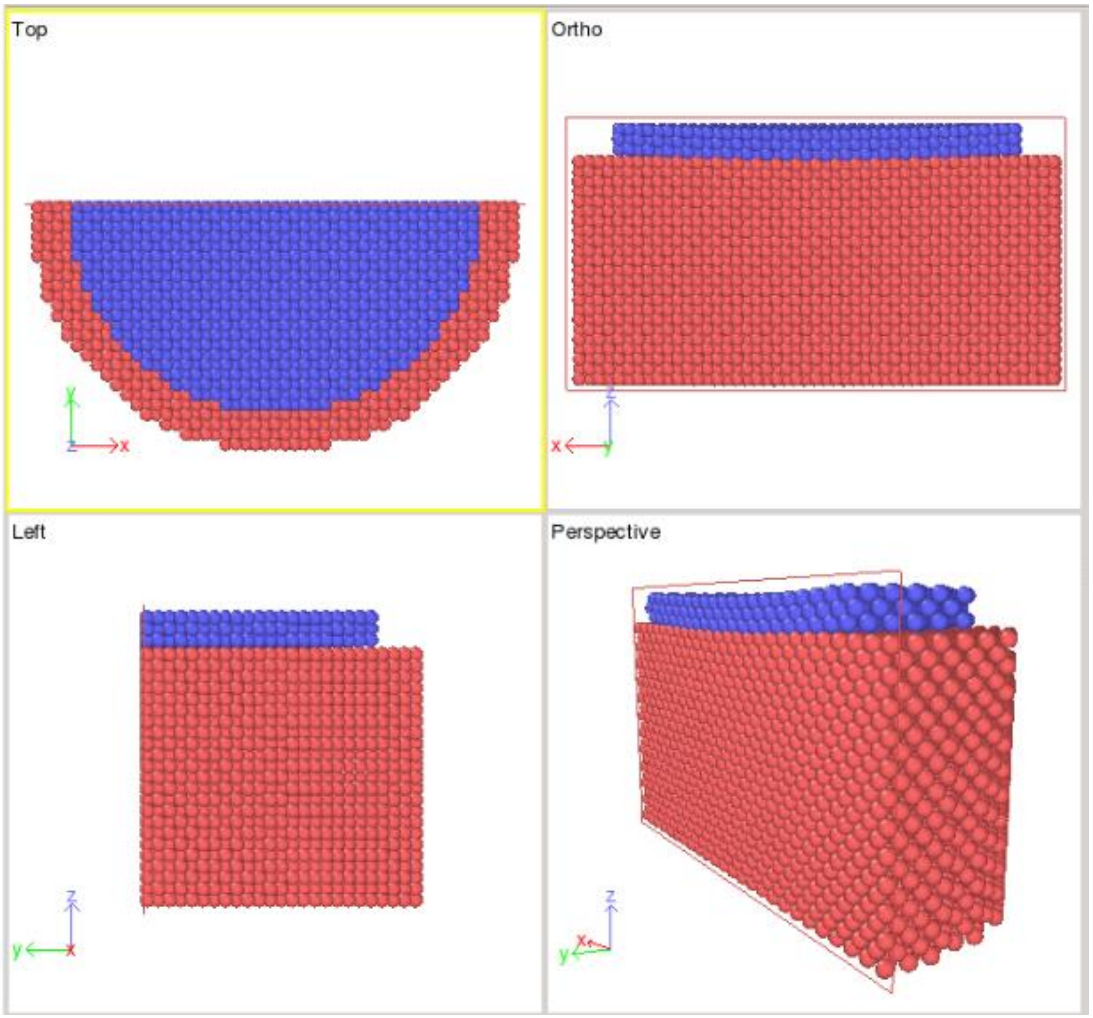



**Figure 6-5: Cross-section view of the micro-indentation model after loading to penetration depth 10µm**

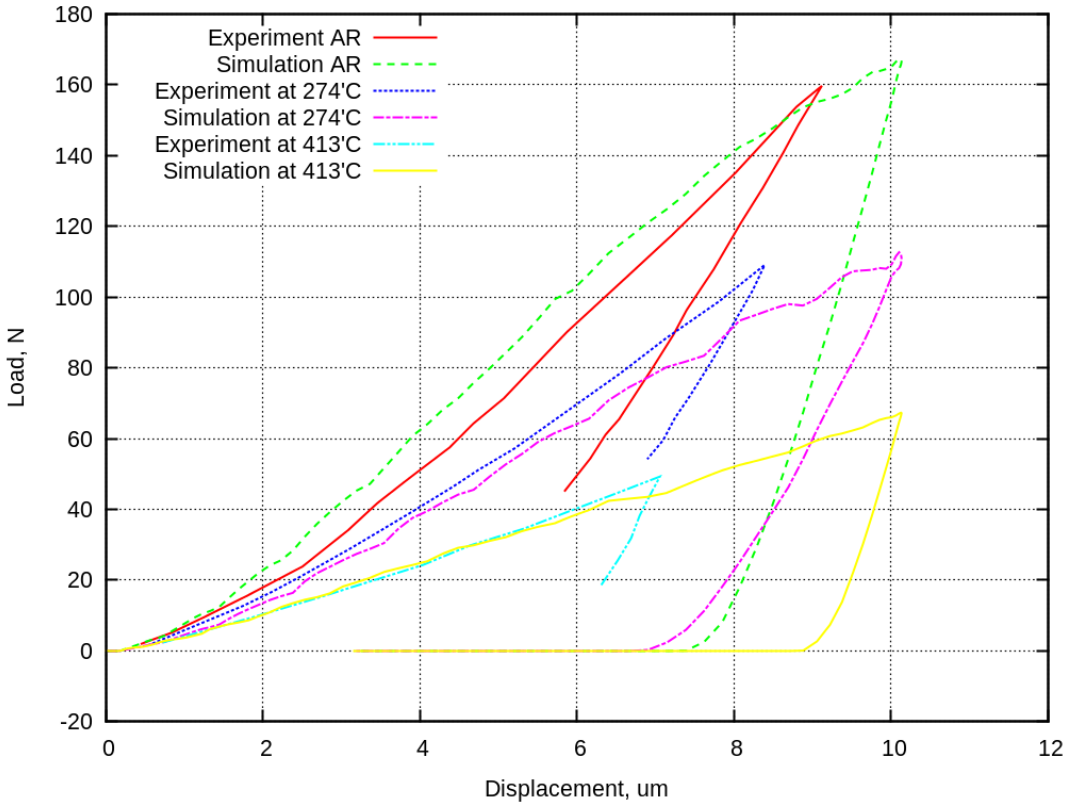



**Figure 6-6: Loading-Displacement curve for all 3 sample: simulation vs experimental results**

The loading-displacement curve shown in Figure 6-6 shows the results of the micro-indentation test described. The three curves calculated for the simulation of micro-indentation of the three different materials show correct behaviour which resembles the behviour of the experimental data, however there is a degree of error.

As explained, due to restrictions in the simulation, the indenter was designed to have a disk shape instead of a spherical one. This will have an effect on the simulation results because the pressure distribution in the contact during the beginning of the test is different and it will affect the behaviour when plastic deformation occurs.

In the case of the AR sample, the simulation measured a smaller indentation depth with a maximum error of 1% at around 160 Newtons. The same error is also present in the unloading curve. On the other hand, in the cases of 274°C and 413°C, the model overestimated the strain, with a maximum error of 0.8% and 0.5% respectively.

In all 3 cases, the general trend of the deformation is conserved.

The differences between the experimental and model results can be attributed to the following reasons:

- The shape of the indenter in the beginning of the test influences not only the pressure distribution in the contact, but also the onset of plasticity and yielding.

- The linear strain hardening model used may not perfectly capture the material behaviour at this scale of deformation.
- Since the experimental aluminium samples have been heat treated, the exact homogeneity of the microstructure at the indenter tip is unknown.
- After displacement of some particles, internal forces may produce extra forces in the interaction with the main sample.

Furthermore, as mentioned in section 5.2.4, for the mechanical properties of the material here 0.08 is used as the value of the ultimate strain ($\varepsilon_u$) for all three materials, as shown in Table 5-1. A better solution could be achieved by a better fit of the strain work hardening modulus ($E_h$) by using ultimate strain ($\varepsilon_u$) of 0.02 for example.

Another possible reason for the error could be due to the effect of surface roughness. All the specimens were experimentally grounded and polished to 0.06-μm colloidal silica, however, the particle size in the model is 12.7 μm which is much higher than the roughness.

Despite the above-mentioned issues, the model is generally capable of predicting the trend of the loading cycle and the expected elastic recovery to a very good degree of agreement.

Further work can be done to obtain better agreement with experimental results, such as:

- simulating the indentation of bigger samples using a spherical indenter bigger sample
- smaller particle size, hence larger number of particles and simulation size
- fitting of the strain work hardening modulus by using lower values of the ultimate strain ($\varepsilon_u$), for example 0.02 instead of 0.08
- implement multi-linear hardening
- study the inhomogeneity of the samples used in the experimental study and incorporate them in the initial structure of the model using the bonding features
- study the effect of slip plane direction
- study the effect of roughness
- properly account for surface interactions

## 6.3 Case Studies

This section presents some further simulations to show the capabilities of developed model. Here no validation against experimental results or detailed analysis is conducted; only the mechanical response is analysed. The tests shown here are only for the purpose of showing the types of simulation and geometries that could be simulated with the use of the new MCA commands in LIGGGHTS-MCA.

### 6.3.1 Case Study I: Indentation using Berkovich Indenter

In this section similar indentation behaviour as described in the previous section is simulated but with a diamond Berkovich Indenter, a much simpler simulation setup and much larger particle size. The Berkovich indenter is commonly used in hardness testing and indentation, so this section explores the capability of the model to describe the behaviour using a pyramid-shaped indenter instead of a spherical one.

Here aluminium samples are indented with a diamond Berkovich indenter. The sample has equal-sized particles of 0.254 mm located in SC and FCC structures. The sample size is 30 mm × 30 mm × 30 mm Figure 6-7 shows the initial structure of the FCC packed sample. Figure 6-8 shows the simulated model after the sample has been indented to its maximum penetration depth which is 5 layers of particles. It shows the deformation of the sample in the z direction of loading. Figure 6-9 shows the map of residual displacement at the surface of the sample at different time-steps; from the initial step to the final step, showing irreversible plastic deformation. Figure 6-11 shows the map of residual displacement at the surface for the final time-step. Figure 6-11 shows the calculated load-displacement curve of the indentation.

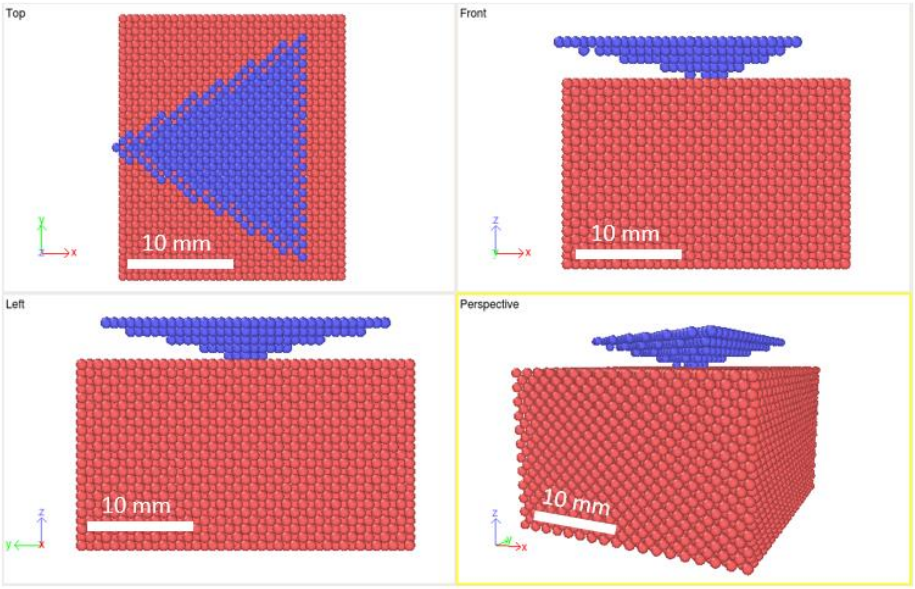



**Figure 6-7: Initial structure of indentation using Berkovich indenter**

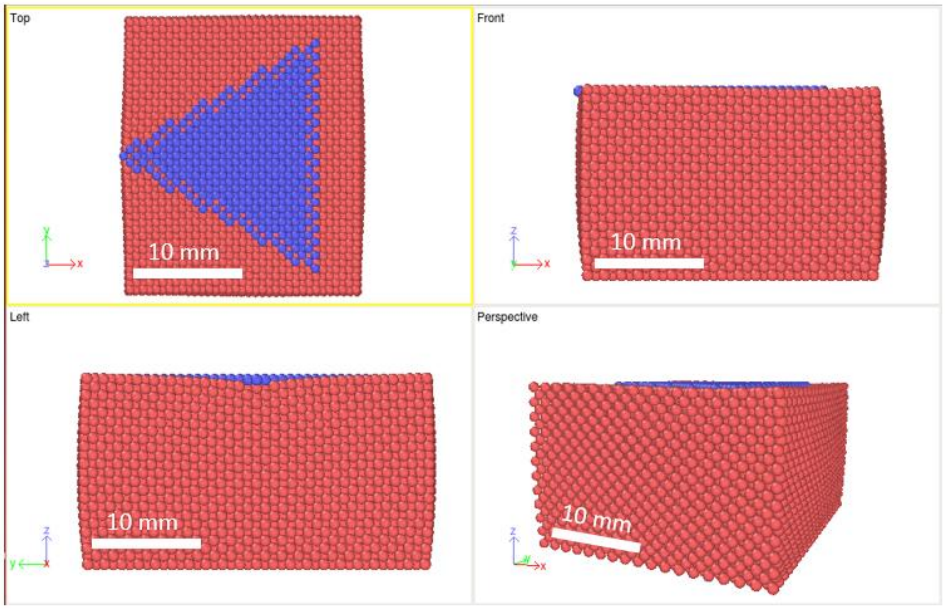



**Figure 6-8: Indentation after loading to maximum penetration depth**

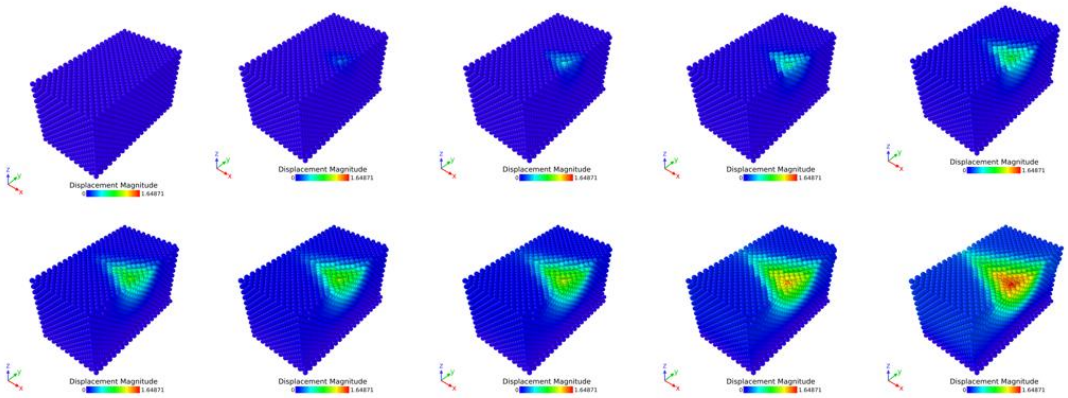


**Figure 6-9: Map of residual displacement at the surface at different time steps from initial to final time-step showing irreversible plastic deformation**

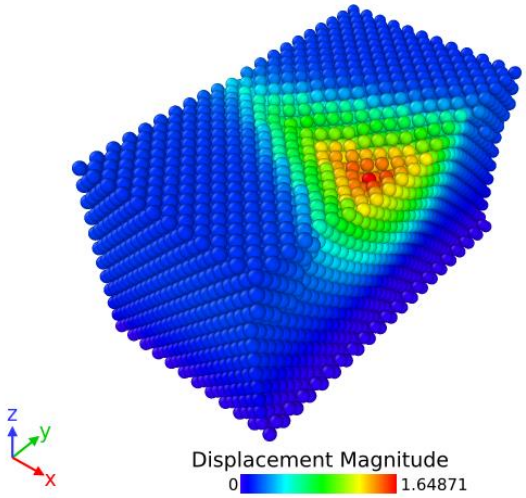



**Figure 6-10: Map of residual displacement at the surface at final time-step**

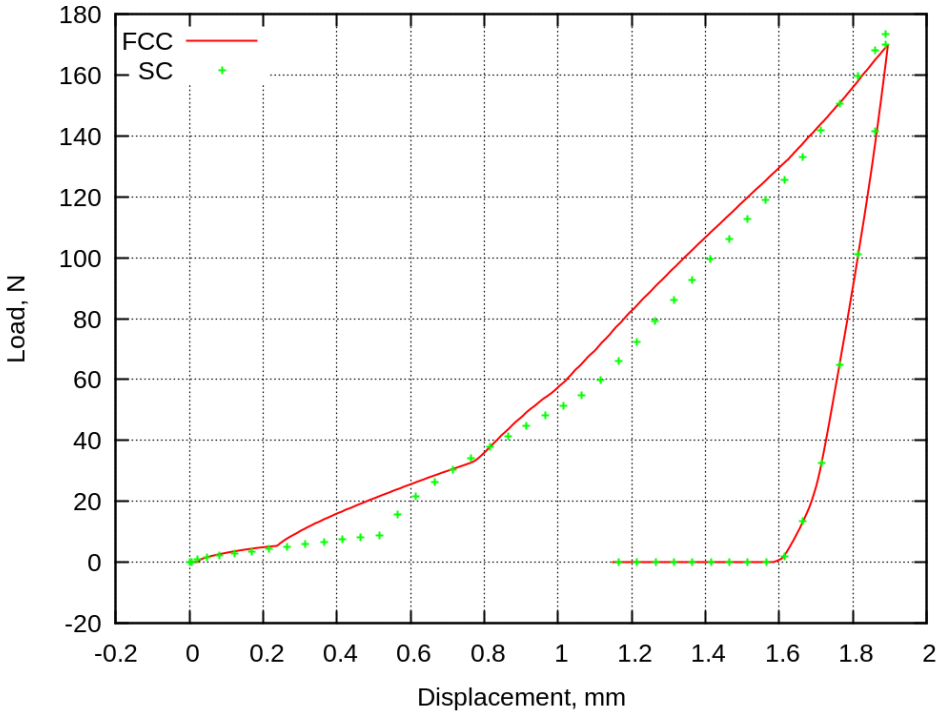



**Figure 6-11: Loading-Displacement curve for indentation with Berkovish indenter showing the difference between results in SC and FCC structures**

The loading-displacement curve of the indentation using the Berkovish indenter is shown in Figure 6-11. Even though the results here are not validated or investigated in detail, the results show that the model is capable of capturing the expected behaviour of material deformation.

The curve shows a step-like behaviour because of the layers of particles in the indenter shape. This is not the case in the simulation with spherical indenter. However, even in the case of indentation using Berkovich indenter the curve will be smoother if smaller particle sizes and more layers of particles are simulated.

The plot also shows the difference between the indentation using SC and FCC packing structures, and both show good behviour. In the case of SC, step like behaviour can be attributed to the compression of the layers of particles due the lower packing efficiency. In the case of FCC, the improved packing provides a much smoother climb. However, both packing structures show close agreement.

### 6.3.2 Case Study II: Scratch Testing

Scratch testing is performed by a controlled continuous loading of the material by moving the indenter, after it is inserted at a certain penetration depth, along the material surface. The material deforms elastically first and then moves into the elastic-plastic range to its limiting state and then further fracture occurs.

Indentation and scratching are essentially similar procedures, only that the latter traces the motion of the indenter not only normally to the surface but also tangentially. Using this coefficient of friction can be analysed.

Here the exact same simulation parameters as described in section 6.3.1. are used, the only difference is that sliding occurs after indentation of one layer of particles. The initial structure of the model is shown in Figure 6-12 but in SC packing structure instead of FCC as shown in Figure 6-7.

Figure 6-13 shows the simulated model after the sample has been indented to its maximum penetration depth which is one layer of particles. It shows very slight deformation of the sample in the z direction of loading because only one layer of atoms is indented, in comparison to larger deformation in the previous case in Figure 6-8 because 5 layers of atoms were indented. Figure 6-14 shows the map of residual displacement of the model with the indenter (on the left) and without the indeter to view the scratch (on the right) showing the irreversible plastic deformation of the aluminium sample. Figure 6-15 shows the results in terms of normal load-displacement curve, and Figure 6-16 shows the tangential loading-displacement curve.

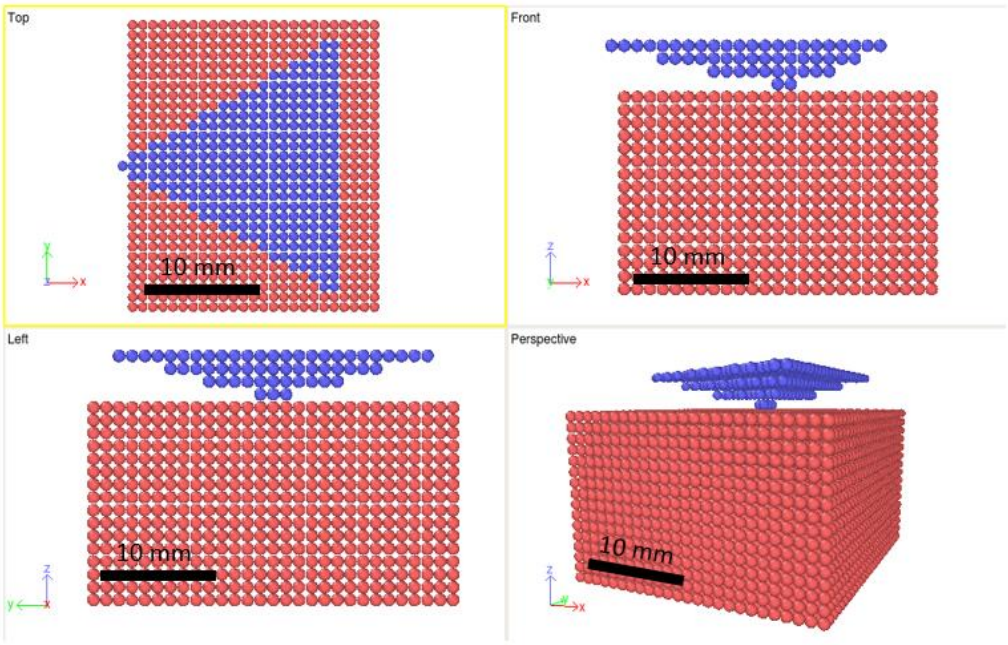



**Figure 6-12: Initial structure of scratch test using Berkovich indenter**

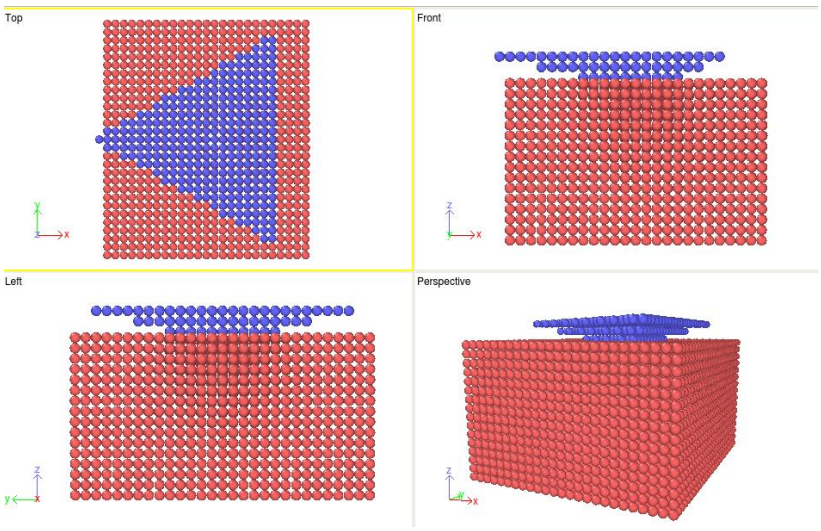



**Figure 6-13: Scratch test after loading to maximum penetration depth**

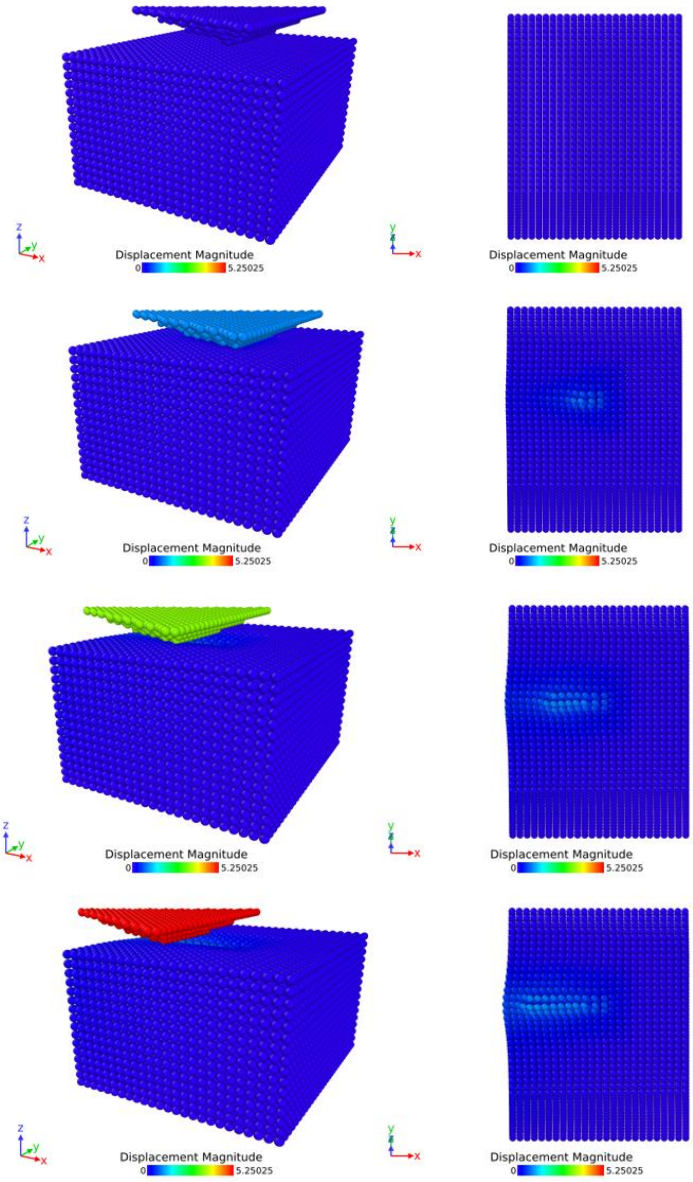



**Figure 6-14: Map of residual displacement from initial state (top two images) to the final state (bottom two images)**

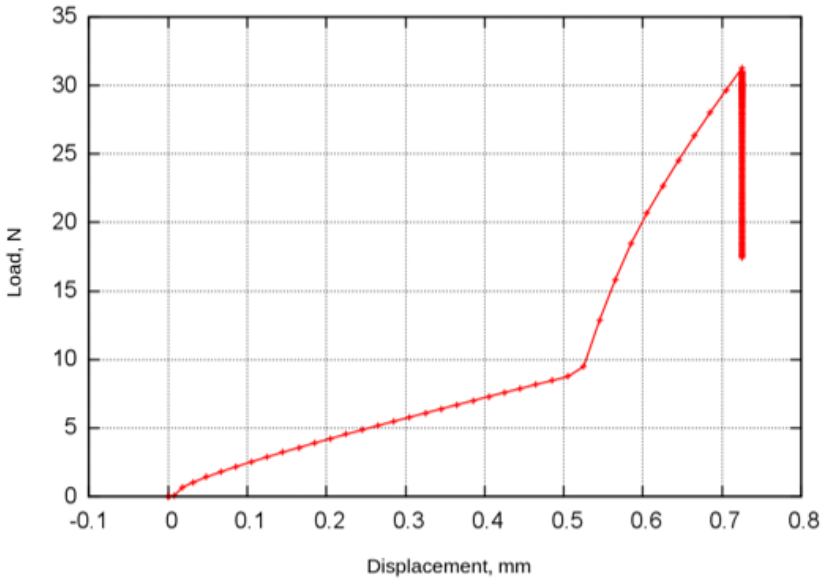



**Figure 6-15: Loading-Displacement curve for scratch test**

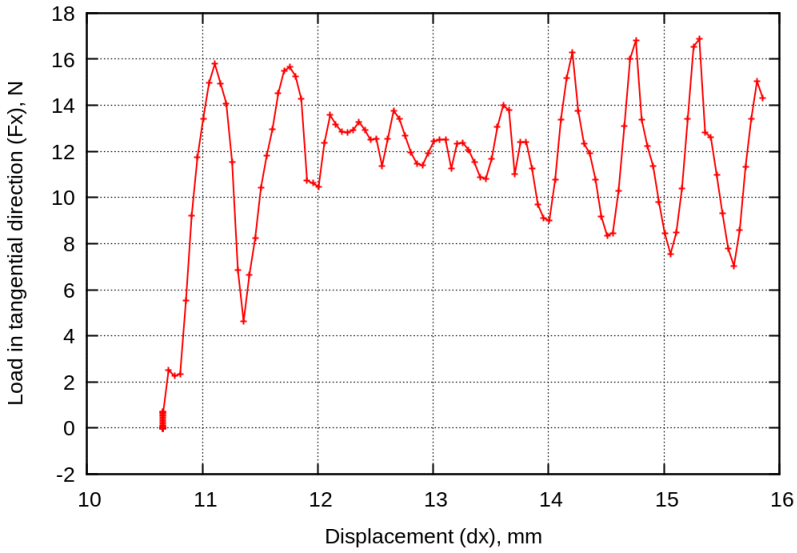



**Figure 6-16: Loading-displacement curve in the tangential direction**

The loading-displacement curve of the scratch test using the Berkovish indenter is shown in Figure 6-15. The results show that model is capable of calculating and tracking the position of the indenter and how the depth and force change with scratching. Figure 6-16 shows the loading-displacement curve in the tangential direction. The coefficient of friction can also be extracted from the plot and calculated as $COF = \frac{F_X}{F_N} \approx 0.4$ which is in very good agreement with the expected value for aluminium. Scratch tests are primarily used to test the adhesion between a coating and a surface. They can also be used to measure nano-scale friction. Further work is required in order to validate the procedure, however, for the purposes of testing the model and what it can measure, the model is capable of simulating scratch tests.

## 6.4 Conclusion

This chapter further delivered the fourth contribution mentioned in section 1.4 of validation of the code against experimental data. MCA simulations of indentation and sliding processes are presented to test this new software for the prediction of tribological behaviour

This was done by simulating micro-indentation tests and comparing them to experimental results. The results show correct mechanical response but show a degree of variation compared to the experimental results. However, a larger degree of agreement with experimental results could be obtained by investigating the points mentioned in section 6.2.4.

# Chapter 7
# Summary & Conclusions

This thesis successfully extended the open-source code LIGGGHTS to move from DEM description to MCA, the same way LIGGGHTS is an extended version of LAMMPS to move from MD to DEM. The reasons behind choosing MCA has been discussed in detail in Chapters 2 and 3. Specifically, this is believed to be a great step ahead towards the prediction of friction and wear, by being able to correctly simulate elastic-plastic deformation of different scale levels by using direct macroscopic material deformation parameters as input parameters.

Our understanding of friction and wear processes is still very limited. How two surfaces come in contact and how they interact is not in any way a simple process. When an object moves tangentially to a surface such as in sliding or rolling cases, frictional forces are produced leading to energy loss, wear and deformation of surfaces, limiting the lifetime of mechanical systems. However, the modelling of friction and wear phenomena is very complex and despite of the vast interest in the field and the increasing capabilities of computational modelling, no practical and comprehensive friction models occur that can show and predict all the aspects of friction dynamics observed experimentally.

Most models available for friction and wear are empirical models that do not involve any information on the underlying mechanisms, are very hard to generalize for different materials and conditions, and hence lack predictive capabilities. This lack of understanding and predictability is due to the multi-scale nature of the phenomena; meaning the origin of friction and wear comes from the atomic scale up to the macro scale. The behaviour on all the scales should be captured, however, this is – until now – not possible. It is a major mathematical challenge and the aim of this thesis was to develop a model that will help bridge the gap between the nano and macro scales for surfaces in contact to study and predict friction and wear behaviour.

Many numerical studies were conducted in tribology; however, they were mostly performed on a single scale and there is a lack of coupling between the different scales within a single simulation. For effective and reliable predictive models of friction and wear to occur, tribological systems should be modelled as a whole system, the surfaces in contact along with the interactions between them. The tribological components and phenomena (friction and wear) exist at many different scales, ranging from the atomic scales to the macro scales.

It was concluded that the multi-scale modelling of tribological systems is crucial for understanding and predicting tribological phenomena. All the scales have an effect on these phenomena, however the mesoscopic scale was found to be the scale most

responsible for their behaviour. Furthermore, the mesoscale could be used as a starting point for bridging the gap between the micro and macro scales.

After extensive research, the movable cellular automata (MCA) method which is a mesoscopic discrete method was chosen for bridging the gap between the atomic and macro scales. MCA is a particle-based method that is capable of directly describing plastic deformation from at different scales from macro mechanical material properties as inputs, due to the many-body interaction form. The main advantage of the MCA method is the generalized many-body formulas for central interaction forces acting between the pair of particles similar to the embedded atom force filed used in molecular dynamics. It is based on computing components of the average stress and strain tensors in the bulk of automaton according to the homogenization procedure described in chapter 3. Use of many-body interaction forces allows correct simulating within discrete element approach such important features of the mechanical behaviour of solids like Poisson effect and plastic flow. Hence, it was chosen to be the best method to attempt multi-scale modelling for the prediction of friction and wear because a clear separation of macroscopic and microscopic scale is principally impossible for triblological phenomena. Mescoscopic modelling bridges the gap between atomistic and continuum viewpoints and friction and wear mechanisms are considered mesoscopic multi-scale phenomena that act very similar to fracture and plastic deformation phenomena.

The MCA method allows for the simulation of friction forces as a function of material, loading parameters, surface topography and wear. Most studies are done by V. L. Popov, S. G. Psakhie and A. I. Dmitriev studying wear in combusion engines, friction coefficient in rail/wheel contact, quasi-fluid nano-layers, surface topography. In the last five years more studies were done by A. I. Dmitriev and W. Österle looking at friction of the automotive braking system and tribofilms. It also gives the possibility of choosing the scale of the simulation which help investigate elementary processes in tribological systems. Furthermore, due to its unique capabilities it allows for the simulation of formation and accumulation of damages, fracture processes, formation of pores and cracks and most importantly the simulation of friction and wear. The results obtained from these simulations have been proved to be reliable compared experimental data.

The MCA method as developed by the authors as listed above and including Sergey G. Psakhie, Valentin L. Popov, Evgeny V. Shilko, and the external supervisor on this thesis Alexey Yu. Smolin, has been successfully implemented in the open-source code LIGGGHTS. This follows the 3D elastic-plastic model found in [1]–[3], and section 3.5 in this thesis. It is important to mention that before delving into the implementation of MCA in LIGGGHTS a good understanding of the use of

LIGGGHTS as a software and also the understanding of its source code was a crucial step to ensure proper implementation and also not to add things which were already available in LIGGGHTS. I believe this was a keystone for the successful implementation within the time frame of the PhD.

The code was successfully implemented within the framework of the open source code LIGGGHTS. Different verification, validation and convergence tests have been conducted.

The implementation of the MCA model described in Chapter 3 has been validated through a number of material characterization simulations. It is validated against experimental tensile test in Chapter 5 and against micro-indentation in Chapter 6. The results show that the model is capable of predicting elastic-plastic deformation for continuous material by directly using the materials macroscopic material parameters. This is only possible due to the MCA formulations described in Chapter 3.

# Chapter 8
# Future Work

It is important to note that this thesis is a starting point for many future possible developments, implementations and use in running simulations and studying material deformation behaviour. I would say 90% of the ground bases have been accomplished and only a few future steps are needed for the MCA method to be totally implemented within LIGGGHTS for the scientific and tribology communities to use. The main limitations of the current code are concerned with the functions related to the cross communication between processors. This is required for parallel processing and periodic boundary conditions. Once this is correctly implemented, smaller particle sizes and larger simulation domains could be simulated which will help study more complex systems. The most important functionalities related to the correct governing of the physical model have been successfully implemented such as particle definition, neighbour list generation and update, force calculations, bonding, integration and looping over time-step. Algorithmic optimization for parallel computing is thus considered as secondary.

The following subsections provide a non-exhaustive list of further developments of the code, as well as possible further analysis and simulations, however many further possibilities may be envisaged, the possibilities listed are in no way limiting.

## 8.1 Further Development of Code

### 8.1.1 Parallelisation

The first and main limitation of the developed MCA model in LIGGGHTS, is that it is still not running completely in parallel. This has limited the type, number and complexity of the simulations that we were able to run. However, with further development of the code's parallelization, different types of materials at different scales and complexity could be simulated including friction and wear.

Ideally, we should have both MPI and OpenMP. As mentioned in section 4.4.10. and 4.4.11. both have not been fully realized. OpenMP is implemented but is not very efficient, and its efficiency needs to be tested and optimized mainly for the '*PairMCA'*, '*BondMCA'* and '*BondCreateMCA*' classes because they are the most time-consuming parts of the code.

Furthermore, currently in some simulation the code produces an error of "Bond atoms … missing on proc …" which usually happens when the wrong size of simulation box is specified because particles can move out of the simulation box and be lost. However, in our case it also sometimes happens when running a simulation

in parallel using OpenMP, but it does not occur in serial mode (one thread). Additional analysis is needed to fix this error.

### 8.1.2 Correcting Code for SC Packing

As explained in section 5.5.1, SC packing only shows good results for the elastic region, but not the plastic region, and the implemented code for calculating the forces for correction for plastic deformation needs to be revised and corrected.

### 8.1.3 Correcting Code for Periodic Boundary Conditions

As explained in section 5.5.2. PBC does not show good results, which is because it has not been fully implemented yet as explained in section 4.4.9. the PBC will be very useful to implement, it will also help with the MPI exchange, however it not fully realised yet. The related function including pack_comm, pack_comm_vel, pack_border, and pack_border_vel, need to be fully understood and revised. Since the periodic boundary feature comes from LAMMPS and the whole communication infrastructure is implemented in the 'basement' of LAMMPS, it might be useful to review this feature in LAMMPS first before trying to implement it in LIGGGHTS, this is also true for the MPI exchange.

It will also be good to test PBCs with smaller particle sizes and study its effect on the macroscopic response.

### 8.1.4 Implementation of Multi-Linear Harding

Currently only linear hardening is implemented for the description of interaction forces for plastic deformation. This shows good agreement with experimental results to some degree, but multi-linear hardening would show more accurate results.

### 8.1.5 Implementation of Bond Formation

Currently, the ‘bond_style mca’ described fracturing or breaking of bonds, meaning the switching of state from linked to unlinked, but not vice versa; which is a very helpful tool in MCA and also is essential for modelling friction zone. The bond formation or binding initial parameters have been defined in ‘BondMCA’ style but force calculation and binding criteria have not been implemented yet.

This can be implemented using the description of section 3.5.3.2.

### 8.1.6 Implementation of Heat Conduction Between Particles

Another limitation is that the current MCA implemented model does not include temperature effects, and therefore cannot adjust parameters due to temperature changes. Particles that are in contact may exchange their thermal energy and have chemical reactions. The heat transfer between particles still needs to be implemented which is also an important step in implementing the bonding formation formulas of MCA and studying effects such as phase change. However, this could be done by incorporating a linear coefficient of thermal expansion ($\alpha_i$) and temperature change ($\Delta T_i$) of automata i in each time-step in the equations of interactions described in section 3.5.2., such that equation (3.54) for calculating the normal force

$$\sigma_{ij}^{n+1} = \sigma_{ij} + \Delta\sigma_{ij} = \sigma_{ij} + 2G_i\Delta\varepsilon_{ij}^{n+1} + (1 - 2G_i/K_i)\Delta\sigma_i^{n+1} \qquad (\mathbf{3.54})$$

can be described as follows to include the thermal expansion and temperature parameters:

$$\sigma_{ij}^{n+1} = \sigma_{ij} + 2G_i\left(\Delta\varepsilon_{ij}^{n+1} - \alpha_i\Delta T_i\right) + (1 - 2G_i/K_i)(\Delta\sigma_i^{n+1} - 3K_i\alpha_i\Delta T_i)$$

This can be implemented in the '*pair_style mca*' but will also need new integrators such as a new '*fix npt/mca*' according to the isobaric-isothermal NPT ensemble, or a new '*fix nph/mca*' according to the isenthalpic NPH ensemble, because the current '*fix nve/mca*' Nose-Hoover thermostat does not have any temperature or pressure control.

### 8.1.7 Enable Restarting of Simulation Runs

LIGGGHTS provides a 'restart' command to restart a simulation after it has stopped. This is useful when a simulation is really long or when the simulation is expected to stop because of an error, so instead of having to run the simulation from the beginning, it is possible to choose at which time-step you want to presume the run. This command produces a binary output file every few time-steps with the information from that time-step needed to continue running later on. The restart files can then be read by a 'read_restart' command to restart the simulation from a particular time-step.

This feature currently does not work with the new MCA commands. This could be implemented by defining '*write & read_restart*" methods related to this function in each of the new classes (atom, pair, bond and fix). The new MCA member classes and attributes need to be defined in these methods for each new MCA class.

### 8.1.8 Numerical Damping

Numerical damping could also be implemented to be able to obtain an equilibrium configuration of the system by using fewer integration steps. In the simulation of quasi-static phenomena, the dissipation of kinetic energy is desired, however, most constitutive laws do not include velocity-based damping, but artificial damping can be used. This is also a technique used in DEM [270], however in MCA a different technique is used. It is related to the way of calculating the current coordination of the automata i and j. Currently, we use the equation (3.57) previously described

$$\Delta r_{ij}^{n+1} = \left|\vec{R}_i^{n+1} - \vec{R}_j^{n+1}\right| - \left|\vec{R}_i^{n} - \vec{R}_j^{n}\right|$$

This takes a lot of computational time, so instead of using actual coordinates of the automata, a so-called implicit factor can be used to calculate the deformation based on predictor values obtained with the help of velocities multiplied by the order of the time step $\zeta\Delta t$, where $\zeta$ is the coefficient of implicitness of the numerical scheme. This means that the current coordinates will be used in a future time-step if the particle velocity has not changed.

The following equation is used to increase the distance between the centers of the automata

$$\Delta r_{ij}^{n+1} = \left(r_{ij}^{n+1} + \zeta\Delta t\left(\vec{V}_{ij}^{n+1}.\vec{n}_{ij}^{n+1}\right)\right) - \left(r_{ij}^{n} + \zeta\Delta t\left(\vec{V}_{ij}^{n}.\vec{n}_{ij}^{n}\right)\right)$$

However, this will probably not be needed if the parallelisation of the code has been accomplished.

### 8.1.9 Test New Code on More Recent Versions of LIGGGHTS

As mentioned, the current MCA styles and code have been implemented in LIGGGHTS-Public version 3.3.1. released 23/09/2015. It would be useful to test it on the most recent version of LIGGGHTS-Public version 3.8.0. released 30/11/2017, in case any users want to use MCA within the more recent versions of LIGGGHTS and also for reasons explained in the following section. This should be relatively easy and no complication are anticipated.

### 8.1.10 Documentation or Manual for Using MCA Model in LIGGGHTS

One of the advantages of both LAMMPS and LIGGGHTS, are that their commands are well documented so that users can easily run simulations correctly. Similar documentation could be written for the new MCA styles which could also help with

possible future submitting of the new features for inclusion in LIGGGHTS-Public as a user package, which is something that LIGGGHTS encourages to do.

According to LIGGGHTS manual [189] “Here is what you need to do to submit a user package or single file for our consideration. Following these steps will save time for both you and us. See existing package files for examples.

1. All source files you provide must compile with the most current version of LIGGGHTS(R)-PUBLIC.
2. If you want your file(s) to be added to main LIGGGHTS(R)-PUBLIC or one of its standard packages, then it needs to be written in a style compatible with other LIGGGHTS(R)-PUBLIC source files. This is so the developers can understand it and hopefully maintain it. This basically means that the code accesses data structures, performs its operations, and is formatted similar to other LIGGGHTS(R)-PUBLIC source files, including the use of the error class for error and warning messages.
3. Your new source files need to have the LIGGGHTS(R)-PUBLIC copyright, GPL notice, and your name at the top, like other LIGGGHTS(R)-PUBLIC source files. They need to create a class that is inside the LIGGGHTS(R)-PUBLIC namespace. I.e. they do not need to be in the same stylistic format and syntax as other LIGGGHTS(R)-PUBLIC files, though that would be nice.
4. Finally, you must also send a documentation file for each new command or style you are adding to LIGGGHTS(R)-PUBLIC. This will be one file for a single-file feature. For a package, it might be several files. These are simple text files which we will convert to HTML. They must be in the same format as other *.txt files in the lammps/doc directory for similar commands and styles. The txt2html tool we use to do the conversion can be downloaded from this site, so you can perform the HTML conversion yourself to proofread your doc page.”

Steps 2 and 3 are completed, except of course for the future development part of the code, as explained in the previous sections, most importantly the parallelisation of the code. Steps 1 and 4 should ready to do.

## 8.2 Further Analysis and Simulation Possibilities

### 8.2.1 Further Performance Analysis

Chapter 5 presented some of the performance analysis done on the present state of the code. However, further analysis could be done, specifically for analysing the computational cost and particle sizes. It is recommended to test the reference case with particle sizes 0.05 mm, 0.005 mm and analyse the corresponding simulation time and accuracy of material deformation response.

It is expected that the particle size below 0.5 mm for the uni-axial tension and shear tests will not produce a much difference in terms of deformation response. However, in the micro-indentation test, and similar further tests it could have a big effect.

It will also increase the possibilities of simulating more complex material deformation phenomena including friction and wear.

### 8.2.2 Simulation of Different Materials

The code has been tested on ductile materials with different yield strengths and strain hardening, however, it is also recommended to test the code on different materials such as ceramics, glass, copper, etc. Since MCA has been proven to be really efficient in the simulation of brittle materials, and the code has shown accurate results for elastic and fracture behaviour, this should not present any difficulties.

### 8.2.3 Multi-Scale Simulation

After the testing of smaller particle-sizes, which will be much easier to do once the code runs in parallel, the code could be tested to run multi-scale simulations. Again, this was already done using the MCA method in other MCA papers including [3], so if the method is correctly implemented within LIGGGHTS, then this should be possible.

### 8.2.4 Simulation of Tribological Systems

In terms of modelling the first bodies, using LIGGGHTS commands, and as shown in the case of modelling indentation, it is possible to define two solids in contact. The material properties and rheology of these bodies could be defined.

In terms of modelling the third body, using the current state of the code it is possible to simulate a “quasi-liquid” layer by creating a layer with specific properties

between two solid bodies in the initial structure of the simulation. High plasticity of this layer will produce easier sliding. Models as described in papers [155]–[162] can be modelled using the current code in 3D instead of just in 2D.

Boundary conditions can then be applied to the three different bodies and the coefficient of friction can be calculated as described in chapter 2, and shown in Figure 2-14, as the vertical components of the force acting on the top layer (block) of particles from the lower particles, divided by the horizontal components of the force acting on the same particles. This describes a dynamics coefficient of friction between the sliding contacts. Wear phenomena could also be studied by analysing damaging and fracturing of particles.

## References


[1] S. Psakhie, E. Shilko, A. Smolin, S. Astafurov, and V. Ovcharenko, "Development of a formalism of movable cellular automaton method for numerical modeling of fracture of heterogeneous elastic-plastic materials," *Frat. ed Integrita Strutt.*, vol. 24, pp. 26–59, 2013.

[2] S. G. Psakhie, E. V. Shilko, A. S. Grigoriev, S. V. Astafurov, A. V. Dimaki, and A. Y. Smolin, "A mathematical model of particle-particle interaction for discrete element based modeling of deformation and fracture of heterogeneous elastic-plastic materials," *Eng. Fract. Mech.*, vol. 130, pp. 96–115, 2014.

[3] E. V. Shilko, S. G. Psakhie, S. Schmauder, V. L. Popov, S. V. Astafurov, and A. Y. Smolin, "Overcoming the limitations of distinct element method for multiscale modeling of materials with multimodal internal structure," *Comput. Mater. Sci.*, vol. 102, pp. 267–285, 2015.

[4] X. You-Bai and C. In, *Theory of Tribo-Systems.* 1852.

[5] M. Renouf, F. Massi, N. Fillot, and A. Saulot, "Numerical Tribology of a Dry Contact," *Tribol. Int.*, vol. 44, pp. 834–844, 2011.

[6] V. E. Panin, "Chapter 12: Discrete computer models of condensed media with an internal strucutre," in *Physical Mesomechanics of Heterogeneous Media and Computer-Aided Design of Materials*, Suffolk, England: Cambridge International Science Publishing, 1998, pp. 249–270.

[7] K. C. Ludema, "Mechanism-based modeling of friction and wear," *Wear*, vol. 200, no. 1–2, pp. 1–7, 1996.

[8] Lehigh University, "The Tribology Laboratory Research." [Online]. Available: http://www.lehigh.edu/~intribos.

[9] H. C. Meng, K. C. Ludema, and H. Meng C., Ludema K.,C., "Wear Models and Predictive Equations: Their Form and Content," *Wear*, vol. 181–183, pp. 443–457, 1995.

[10] P. Dahl, "A solid friction model," *Aerosp. Corp. El Segundo, CA*, vol. 158, p. Tech. Rep. TOR-01 58(3 107-1 8)-1, 1968.

[11] C. de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *Automatic Control, IEEE Transactions on*, vol. 40, no. 3. pp. 419–425, 1995.

[12] H. Dankowicz, "On the Modeling of Dynamic Friction Phenomena," *ZAMM*, vol. 79, no. 6, pp. 399–409, 1999.

[13] M. Eriten, "Multiscale Physics-Based Modeling of Friction," University of Illinois, 2012.

[14] F. P. Bowden and D. Tabor, "Friction, lubrication and wear: a survey of work during the last decade," *Br. J. Appl. Phys.*, vol. 17, no. 12, pp. 1521–1544, Dec. 2002.

[15] C. M. Mate, *Tribology on the Small Scale: A Bottom Up Approach to Friction, Lubrication and Wear*. Oxford University Press, 2008.

[16] K.-H. Zum Gahr, *Microstructure and Wear of Materials*. Amsterdam, the Netherlands: Elsevier, 1987.

[17] J. F. Archard, "Contact and Rubbing of Flat Surfaces," *J. Appl. Phys.*, vol. 24, no. 8, p. 981, 1953.

[18] A. D. Sarkar, *Friction and wear*. Academic Press, 1980.

[19] H. H. Czichos, "Tribology and its many facets: From macroscopic to

microscopic and nano-scale phenomena," *Meccanica*, vol. 36, no. 6, pp. 605–615, 2001.
[20] M. J. Buehler, *Atomistic Modelling of Materials Failure*. New York, USA: Springer, 2008.
[21] J. H. Argyris and S. Kelsey, "Energy theorems and structural analysis," New York Plenum Press, London, UK, 1954.
[22] M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp, "Stiffness and deflection analysis of complex structures," *J. Aeronaut. Sci.*, vol. 23, pp. 805–823, 1956.
[23] E. B. Tadmor and R. E. Miller, *Modeling materials: Continuum, atomistic and multiscale techniques*, vol. 9780521856. 2011.
[24] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. Oxford, UK: Clarendon Press, 1987.
[25] D. Frenkel and B. Smith, *Understanding Molecular Simulation: From Algorithms to Applications*. San Diego, US: Academic Press, 1996.
[26] S. J. Plimpton, "Large-scale Atomic/Molecular Massively Parallel Simulator, LAMMPS." Department of Mechanical Engineering and Materials Science, University of Pittsburgh.
[27] P. A. Cundall, "A computer model for simulating progressive, large scale movement in blocky rock systems," in *ISRM Symposium* , 1971.
[28] P. A. Cundall and O. D. L. Strack, "A discrete numerical model for granular assemblies," *Géotechnique*, vol. 29, no. 1, pp. 47–65, 1979.
[29] W. K. Liu, E. G. Karpov, and H. S. Park, *Nano Mechanics and Materials*. West Sussex, UK: John Wiley & Sons Ltd, 2006.
[30] F. F. Abraham, J. Broughton, N. Bernstein, and E. Kaxiras, "Spanning the continuum to quantum lenth scales in a dynamic simulation of brittle fracture," *Europhys. Lett.*, vol. 44, pp. 783–787, 1998.
[31] W. A. Curtin and R. E. Miller, "Atomistic/conituum couplin in computational materials science," *Model. Simul. Mater. Sci. Eng.*, vol. 11, pp. R33–R68, 2003.
[32] W. K. Liu, E. G. Karpov, S. Zhang, and H. S. Park, "An introduction to computational nano-mechanics and materials," *Comput. Methods Appl. Mech. Eng.*, vol. 193, pp. 1529–1578, 2004.
[33] S. Perić, B. Nedić, D. Trifković, and M. Vuruna, "An Experimental Study of the Tribological Characteristics of Engine and Gear Transmission Oils," *J. Mech. Eng.*, vol. 59, pp. 443–450, 2013.
[34] I. Kovaříková, B. Szewczyková, P. Blaškovitš, E. Hodúlová, and E. Lechovič, "Study and Characteristic of Abrasive Wear Mechanisms," *Mater. Sci. Technol.*, no. APVV-0057-07, VEGA 1/0381/08, APVT-20-011004, pp. 1335–9053, 2009.
[35] G. Mollon, "A numerical framework for discrete modelling of friction and wear using Voronoi polyhedrons," *Tribol. Int.*, vol. 90, pp. 343–355, 2015.
[36] G. Maurice, "Third-bodies in tribology," *Wear*, vol. 136, no. 1, pp. 29–45, 1990.
[37] Y. Berthier, "Maurice Godet's Third Body," *Tribol. Ser.*, vol. 31, pp. 21–30, 1996.
[38] M. Godet, "The Third Body Approach: A Mechanical View of Wear," *Wear*, vol. 100, pp. 437–452, 1984.
[39] Y. Berthier, "Experimental Evidence for Friction and Wear Modelling," *Wear*, vol. 139, pp. 77–92, 1990.
[40] Denape and Jean, "Third body concept and wear particle behavior in dry

friction sliding conditions," *Tribol. Asp. Mod. Aircr. Ind. Trans Tech Publ.*, pp. 1–12, 2014.

[41] P. J. Blau, "Scale effects in steady-state friction," *Tribol. Trans.*, vol. 34, no. 3, pp. 335–342, 1991.

[42] H. Spikes, "Tribology Research in the Tweny-First Century," *Tribol. Int.*, vol. 34, pp. 789–799, 2001.

[43] A. I. Vakis *et al.*, "Modeling and simulation in tribology across scales: An overview," *Tribol. Int.*, vol. 125, pp. 169–199, Sep. 2018.

[44] P. R. Budarapu, J. Reinoso, and M. Paggi, "Concurrently coupled solid shell-based adaptive multiscale method for fracture," *Comput. Methods Appl. Mech. Eng.*, vol. 319, pp. 338–365, 2017.

[45] V. L. Popov, "Method of reduction of dimensionality in contact and friction mechanics: A linkage between micro and macro scales.," *Friction*, vol. 1 SRC-B, no. 1, pp. 41–62, 2013.

[46] F. Massi, J. Rocchi, A. Culla, and Y. Berthier, "Coupling system dynamics and contact behaviour: Modelling bearings subjected to environmental induced vibrations and 'false brinelling' degradation. Mechanical Systems and Signal Processing , .," *Mech. Syst. Signal Process.*, vol. 24 SRC-, pp. 1068–1080, 2010.

[47] F. Bucher *et al.*, "Multiscale simulation of dry friction in wheel/rail contact," *Wear*, vol. 261, no. 7–8, pp. 874–884, 2006.

[48] I. Temizer and P. Wriggers, "A multiscale contact homogenization technique for the modeling of third bodies in the contact interface," *Comput. Methods Appl. Mech. Eng.*, vol. 198, pp. 377–396, 2008.

[49] M. Schargott, "A Multi-scale Approach for Dynamical Simulations of the Surface Topography in High-pressure Frictional Contacts: Bridging Simulation Scales," *Tribol. Lett.*, vol. 39, pp. 9–17, 2010.

[50] A. P. A. P. Nyemeck, N. Brunetière, B. Tournerie, N. Brunetiere, and B. Tournerie, "A Multiscale Approach to the Mixed Lubrication Regime: Application to Mechanical Seals," *Tribol. Lett.*, vol. 47, no. 3, pp. 417–429, 2012.

[51] J. Cho, T. Junge, J. F. Molinari, and G. Anciaux, "Toward a 3D coupled atomistic and discrete dislocation dynamics simulation: dislocation core structures and Peierls stresses with several character angles in FCC aluminum," *Adv. Model. Simul. Eng. Sci.*, vol. 2, no. 1, 2015.

[52] C. GONIVA, C. KLOSS, and S. PIRKER, "Towards fast parallel CFD-DEM: An Open-Source Perspective," *Proc. Open Source CFD Int. Conf.*, no. January, pp. 12–13, 2009.

[53] G. Anciaux and J. F. Molinari, "Contact mechanics at the nanoscale, a 3d multiscale approach," *Int. J. Numer. Methods Eng.*, vol. 79, pp. 1041–1067, 2009.

[54] B. Q. Luan, S. Hyun, J. F. Molinari, N. Bernstein, and M. O. Robbins, "Multiscale modeling of two-dimensional contacts," *Phys. Rev. E*, vol. 74, 2006.

[55] G. Anciaux, S. B. Ramisetti, and J. F. Molinari, "A finite temperature bridging domain method for MD-FE coupling and application to a contact problem," *Comput. Methods Appl. Mech. Eng.*, vol. 205–208, no. 1, pp. 204–212, 2012.

[56] G. K. Sfantos and M. H. Aliabadi, "Multi-scale boundary element modelling of material degradation and fracture," *Comput. Methods Appl. Mech. Eng.*, vol. 196, pp. 1310–1329, 2007.

[57] V. E. Panin, “Overview on mesomechanics of plastic deformation and fracture solids,” *Theor. Appl. Fract. Mech.*, vol. 30, pp. 1–11, 1998.
[58] S. G. Psakhie and V. L. Popov, “Mesoscopic nature of friction and numerical simulation methods in tribology,” *Phys. Mesomech.*, vol. 15, pp. 251–253, 2012.
[59] G. P. Ostermeyer, “The mesoscopic particle approach,” *Tribol. Int.*, vol. 40, pp. 953–959, 2007.
[60] I. I. Garbar, “The Mesostructure of Surface Layers of Metal Under Friction with Relatively High Contact Stress,” in *Fundamentals of Tribology and Bridging the Gap Between the Macro- and Micro/Nanoscales*, London: NATO Science Series, 2001, pp. 529–535.
[61] M. Kalweit, “Molecular Modelling of Meso- and Nanoscale Dynamics,” 2008.
[62] A. A. Olshevsky, A. A. Olshevsky, K. Shevchenko, and V. Sakalo, “The Accounting of Surface Roughness in Contact of Arbitary Shaped Bodies using FEM,” *Xxi Ictam*, no. August, pp. 4–5, 2004.
[63] H. Hertz, *Miscellaneous Papers by H. Herts*. London: Macmillan, 1896.
[64] J. A. Greenwood and J. B. P. Williamson, “Contact of nominally flat surfaces,” in *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 1966, vol. 295, no. 1442, pp. 300–319.
[65] G. M. L. Gladwell, “Contact problems in the classical theory of elasticity,” Alphen aan den Rijn, Sjthoff and Noordhoff, 1980.
[66] K. L. Johnson, “The effect of a tangential contact force upon the rolling motion of an elastic sphere on a plane,” *ASME J. Appl. Mech.*, vol. 25, pp. 339–46, 1958.
[67] K L Johnson, *Contact Mechanics*. Cambridge, New York: Wiley-Blackwell, 1985.
[68] K. L. Johnson, K. Kendall, and A. D. Roberts, “Surface energy and the contact of elastic solids,” in *Proceedings of the Royal Society of London A*, 1971.
[69] B. V. Derjaguin, V. M. Muller, and Y. P. Toporov, “Effect of contact deformations on the adhesion of particles,” *J. Colloid Interface Sci.*, vol. 53, no. 2, pp. 314–326, 1975.
[70] W. R. Chang, I. Etsion, and D. B. Bogy, “An Elastic-Plastic Model for the Contact of Rough Surfaces,” *J. Tribol.*, vol. 109, no. 2, p. 257, 1987.
[71] Y. Zhao, D. M. Maietta, and L. Chang, “An Asperity Microcontact Model Incorporating the Transition From Elastic Deformation to Fully Plastic Flow,” *J. Tribol.*, vol. 122, no. 1, p. 86, 2000.
[72] E. J. Abbott and F. A. Firestone, “Specifying Surface Quality－A Method Based on Accurate Measurement and Comparison,” *J. Mech. Eng.*, vol. 55, pp. 569–572, 1933.
[73] D. Tabor, *The hardness of metals*. Clarendon Press, 2000.
[74] B. N. J. Persson, “Theory of rubber friction and contact mechanics,” *J. Chem. Phys.*, vol. 115, no. 8, pp. 3840–3861, 2001.
[75] M. Ciavarella and G. Demelio, “Elastic multiscale contact of rough surfaces: Archard’s model revisited and comparisons with modern fractal models,” *ASME J. Appl. Mech.*, vol. 68, pp. 496–8, 2001.
[76] G. T. Ham, C. A. Rubin, G. L. Hahn, and G. V Bharghava, “Elasto-plastic finite element analysis of repeated, two-dimensional rolling sliding contacts,” *ASME J. Tribol.*, vol. 110, pp. 44–49, 1988.
[77] O. C. Zienkiewicz and R. L. Taylor, *The finite element method in engineering*

*science*. London, Uk: McGraw-Hill, 2000.
[78] P. K. B. R. Banerjee, “Boundary element methods in engineering science,” *Int. J. Rock Mech. Min. Sci. Geomech. Abstr.*, vol. 21, no. 1, p. A1, 1981.
[79] H. Ouyang, W. Nack, Y. Yuan, and F. C. F, “Numerical analysis of automotive disc brake squeal: a review,” *Int. J. Veh. Noise Vib.*, vol. 1, pp. 207–231, 2005.
[80] A. Bemporad and M. Paggi, “Optimization algorithms for the solution of the frictionless normal contact between rough surfaces,” *Int. J. Solids Struct.*, vol. 69–70, pp. 94–105, 2015.
[81] C. Jacq, D. Nélias, G. Lormand, and D. Girodin, “Development of a Three-Dimensional Semi-Analytical Elastic-Plastic Contact Code,” *J. Tribol.*, vol. 124, no. 4, p. 653, 2002.
[82] F. Sahlin, R. Larsson, A. Almqvist, P. M. Lugt, and P. Marklund, “A mixed lubrication model incorporating measured surface topography. Part 1: Theory of flow factors,” *Proc. Inst. Mech. Eng. Part J J. Eng. Tribol.*, vol. 224, no. 4, pp. 335–351, 2010.
[83] G. Carbone and C. Putignano, “A novel methodology to predict sliding and rolling friction of viscoelastic materials: Theory and experiments,” *J. Mech. Phys. Solids*, vol. 61, no. 8, pp. 1822–1834, 2013.
[84] R. Bugnicourt, P. Sainsot, N. Lesaffre, and A. A. Lubrecht, “Transient frictionless contact of a rough rigid surface on a viscoelastic half-space,” *Tribol. Int.*, vol. 113, pp. 279–285, 2017.
[85] P. R. Dawson, “Computational crystal plasticity,” *Int. J. Solids Struct.*, vol. 37, no. 1–2, pp. 115–130, 2000.
[86] R. Hill and J. R. Rice, “Constitutive analysis of elastic-plastic crystals at arbitrary strain,” *J. Mech. Phys. Solids*, vol. 20, no. 6, pp. 401–413, 1972.
[87] G. I. Taylor and C. F. Elam, “Bakerian Lecture. The Distortion of an Aluminium Crystal during a Tensile Test,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 102, no. 719, pp. 643–667, 1923.
[88] G. I. Taylor, “Plastic strain in metals,” *J. Inst. Met.*, vol. 62, pp. 307–324, 1938.
[89] R. Hill, “Generalized constitutive relations for incremental deformation of metal crystals by multislip,” *J. Mech. Phys. Solids*, vol. 14, no. 2, pp. 95–102, 1966.
[90] R. J. Asaro, “Crystal Plasticity,” *J. Appl. Mech.*, vol. 50, no. 4b, p. 921, Dec. 1983.
[91] F. Roters, P. Eisenlohr, T. R. Bieler, and D. Raabe, *Crystal Plasticity Finite Element Methods : in Materials Science and Engineering*. Wiley, 2011.
[92] H. Li, Z. Jiang, and D. Wei, “Crystal Plasticity Finite Modelling of 3D Surface Asperity Flattening in Uniaxial Planar Compression,” *Tribol. Lett.*, vol. 46, no. 2, pp. 101–112, May 2012.
[93] R. Online, H. Li, Z. Jiang, D. Wei, J. Han, and D. Han, “Crystal plasticity finite element modelling of surface roughness and texture of metals Publication Details,” 2011.
[94] L. Nicola, A. F. Bower, K.-S. Kim, A. Needleman, E. Van Der Giessen, and L. N. Ab, “Multi-asperity contact: A comparison between discrete dislocation and crystal plasticity predictions,” *Philos. Mag.*, vol. 88, pp. 3713–3729.
[95] B. Barzdajn, A. T. Paxton, D. Stewart, and F. P. E. Dunne, “A Crystal Plasticity Assessment of Normally-loaded Sliding Contact in Rough Surfaces and Galling,” *J. Mech. Phys. Solids*, Aug. 2018.
[96] P. A. Sabnis, S. Forest, N. K. Arakere, and V. A. Yastrebov, “Crystal

plasticity analysis of cylindrical indentation on a Ni-base single crystal superalloy," *Int. J. Plast.*, vol. 51, pp. 200–217, Dec. 2013.

[97] E. Renner, Y. Gaillard, F. Richard, F. Amiot, and P. Delobelle, "Sensitivity of the residual topography to single crystal plasticity parameters in Berkovich nanoindentation on FCC nickel," *Int. J. Plast.*, vol. 77, pp. 118–140, Feb. 2016.

[98] N. Fillot, I. Iordanoff, and Y. Berthier, "A granular dynamic model for the degradation of material," *ASME Jounrnal Tribol.*, vol. 126, no. 3, pp. 606–614, 2004.

[99] N. Fillot, I. Iordanoff, and Y. Berthier, "Modelling third body flows with a discrete element method-a tool for understanding wear with adhesive particles," *Tribol. Int.*, vol. 40, no. 6, pp. 973–981, 2007.

[100] S. Hironaka, "Boundary Lubrication and Lubricants ," Three Bond Technical News, Tokyo, 1984.

[101] E. Ciulli, "Friction in Lubricated Contacts: from Macro- to Microscale Effects," in *Fundamentals of Tribology and Bridging the Gap Between the Macro- and Micro/Nanoscales*, London: NATO Science Series, 2001, pp. 725–734.

[102] D. Dowson, *History of Tribology*. London and Bury St Edmunds, UK: Professional Engineering Publishing Limited, 1998.

[103] M. Öqvist, "Numerical simulations of mild wear using updated geometry with different step size approaches," *Wear*, vol. 249, no. 1–2, pp. 6–11, 2001.

[104] P. Podra and S. Andersson, "Simulating sliding wear with finite element method.," *Tribol. Int.*, vol. 32, no. 2, pp. 71–81, 1999.

[105] A. Eleőd, J. Devecz, and T. Balogh, "Numerical modelling of the mechanical process of particle detachment by finite element method," *Period. Polytech. Transp. Eng.*, vol. 28, no. 1–2, pp. 77–90, 2000.

[106] Y. Dong, Q. Li, and A. Martini, "Molecular dynamics simulation of atomic friction: A review and guide," *J. Vac. Sci. Technol. A Vacuum, Surfaces, Film.*, vol. 31, no. 3, p. 030801, 2013.

[107] E. Q. Lin, L. S. Niu, H. J. Shi, and Z. Duan, "Molecular dynamics simulation of nano-scale interfacial friction characteristic for different tribopair systems," *Appl. Surf. Sci.*, vol. 258, no. 6, pp. 2022–2028, 2012.

[108] D. M. Heyes, "Molecular aspects of boudary lubrication," *Tribol. Int.*, vol. 29, pp. 627–629, 1996.

[109] X. Zheng, H. Zhu, A. Kiet Tieu, B. Kosasih, A. K. Tieu, and B. Kosasih, "A molecular dynamics simulation of 3D rough lubricated contact," *Tribol. Int.*, vol. 67, pp. 217–221, 2013.

[110] C. Yang, U. Tartaglino, and B. N. J. Persson, "A multiscale molecular dynamics approach to contact mechanics," *Eur. Phys. J. E*, vol. 19, no. 1, pp. 47–58, 2006.

[111] M. Solar, H. Meyer, C. Gauthier, O. Benzerara, and et al., "Molecular dynamics simulations as a way to investigate the local physics of contact mechanics: a comparison between experimental data and numerical results," *J. Phys. D*, vol. 43, 2010.

[112] R. Aghababaei, D. H. Warner, and J. F. Molinari, "Critical length scale controls adhesive wear mechanisms," *Nat. Commun.*, vol. 7, 2016.

[113] J. A. Harrison and D. W. Brenner, "Simulated tribochemistry: an atomic scale view of the wear of diamond," *J. Am. Chem. Soc.*, vol. 116, pp. 10399–10402, 1994.

[114] G. A. Tomlinson, "A molecular theory of friction," *Philos. Mag.*, vol. 7, pp.

905–939, 1929.
[115] Y. I. Frenkel and T. Kontorova, “On the theory of plastic deformation and twinning,” *J. Exp. Theor. Phys.*, vol. 8, p. 1340, 1938.
[116] M. R. Sorensen, K. W. Jacobsen, and P. Stoltze, “Simulations of atomic-scale sliding friction,” *Phys. Rev. B*, vol. 53, pp. 2101–2013, 1996.
[117] T. Jacobs, K. Ryan, P. Keating, and et al., “The Effect of Atomic-Scale Roughness on the Adhesion of Nanoscale Asperities: A Combined Simulation and Experimental Investigation,” *Tribol. Lett.*, vol. 50, pp. 81–93, 2013.
[118] E. Rabinowicz, *Friction and wear of materials*. New York: Wiley, 1965.
[119] F. P. Bowden and D. Tabor, *The friction and lubrication of solids. 2nd corrected ed.* Oxford: Oxford University Press, Clarendon Press, 1986.
[120] U. Landman and W. D. Luedtke, “Nanomechanics and dynamics of tip-substrate interactions,” *J. Vac. Sci. Technol. B*, vol. 9, pp. 414–423, 1991.
[121] J. A. Harrison, S. J. Stuart, and D. W. Brenner, “Atomic scale simulation of tribological and related phenomena,” in *Handbook of micro/nanotribology*, CRC Press, 1999, p. 525.
[122] P. A. Cundall and O. D. L. Strack, “A discrete numerical model for granular assemblies,” *Géotechnique*, vol. 29, no. 1, pp. 47–65, 1979.
[123] D. André, I. Iordanoff, J. Charles, and J. Néauport, “Discrete element method to simulate continuous material by using the cohesive beam model,” *Comput. Methods Appl. Mech. Eng.*, vol. 213–216, pp. 113–125, Mar. 2012.
[124] H. G. Elrod and D. E. B. DE, “Numerical experiments with flows of elongated granules,” in *Leeds-Lyon Symposium*, 1991.
[125] J. E. Lane, T. H. Spurling, B. Freasier, J. W. Perram, and E. R. Smith, “Inconsistency of the density-functional theory of adsorption when using computer simulations,” *Phys. Rev. A*, vol. 20, pp. 2147–2153, 1979.
[126] A. A. Lubrecht, C. Chan-Tien, Y. Berthier, and D. Dowson, *A simple model for granular lubrication influence of boundaries.* 1995.
[127] A. Ghaouti, M. Chaze, P. Dubujet, and F. Sidoroff, “Particulate and granular simulation of the third body behaviour,” *Tribol Ser*, vol. 31, pp. 355–365, 1996.
[128] K. L. Johnson, K. Kendall, and A. D. Roberts, “Surface Energy and the Contact of Elastic Solids,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 324, no. 1558, pp. 301–313, 1971.
[129] B. Seve, I. Iordanoff, and Y. Berthier, “A discrete solid third body model: influence of the intergranular forces on the macroscopic behaviour,” *Tribol. Interface Eng. Ser.*, vol. 39, pp. 361–368, 2001.
[130] N. Fillot, I. Iordanoff, and Y. Berthier, “Simulation of wear through a mass balance in a dry contact,” *ASME J. Tribol.*, vol. 127, pp. 230–237, 2005.
[131] A. Kabir, V. K. J. VK, C. F. Higgs, and M. L. M, “An evaluation of the explicit finite element method approach for modelling dense flows of discrete grains in a coquette shear cell,” *IMechE Journals*, vol. 222, pp. 715–723, 2008.
[132] A. Kabil, M. Lovell, and C. F. Higgs, “Utilizing the explicit finite element method for studying granular flows,” *Trbological Lett.*, vol. 29, pp. 85–94, 2009.
[133] S. G. Psakhie *et al.*, “Method of movable cellular automata as a tool for simulation within the framework of mesomechanics,” *Russ. Phys. J.*, vol. 38, no. 11, pp. 1157–1168, 1995.
[134] J. von Neumann, “The general and logical theory of automata,” in *Systems Research for Behavioral Science: A Sourcebook*, 2017.

[135] J. T. Schwartz, J. von Neumann, and A. W. Burks, “Theory of Self-Reproducing Automata,” *Math. Comput.*, 1967.
[136] E. W. Weisstein, “Rule 30,” *Wolfrm MathWorld*, 2018. [Online]. Available: http://mathworld.wolfram.com/Rule30.html. [Accessed: 17-May-2016].
[137] S. Wolfram, *A new kind of science*. London: Wolfram Media, 2002.
[138] S. Wolfram, *Cellular automata and complexity: Collected papers*, vol. 45, no. 10–11. 2003.
[139] A. Y. Smolin, E. V Shilko, S. V Astafurov, and I. S. Konovalenko, “Modeling mechanical behaviors of composites with various ratios of matrix-inclusion properties using movable cellular automaton method,” *Def. Technol.*, vol. 11, no. 1, pp. 18–34, 2014.
[140] J. Czopor, D. Aniszewska, and M. Rybaczuk, “The influence of defects on strength of ceramics modeled with Movable Cellular Automata,” *Comput. Mater. Sci.*, vol. 51, no. 1, pp. 151–155, 2012.
[141] D. Aniszewska, “Compression and torsion of ceramic specimens by application of movable cellular automata,” *Theor. Appl. Fract. Mech.*, vol. 62, no. 1, pp. 34–39, 2012.
[142] D. Aniszewska, “Fractal characteristics of defects growth in porous ceramics modeled with the Movable Cellular Automata,” vol. 39, pp. 2409–2415, 2015.
[143] A. Y. Smolin, N. V. Roman, I. S. Konovalenko, G. M. Eremina, S. P. Buyakova, and S. G. Psakhie, “3D simulation of dependence of mechanical properties of porous ceramics on porosity,” *Eng. Fract. Mech.*, vol. 130, pp. 53–64, 2014.
[144] S. G. Psakhie *et al.*, “Modeling nanoindentation of TiCCaPON coating on Ti substrate using movable cellular automaton method.,” *Comput. Mater. Sci.*, vol. 76 SRC-, pp. 89–98, 2013.
[145] A. Y. Smolin, G. M. Eremina, V. V. Sergeev, and E. V. Shilko, “Three-dimensional movable cellular automata simulation of elastoplastic deformation and fracture of coatings in contact interaction with a rigid indenter,” *Phys. Mesomech.*, vol. 17, no. 4, pp. 292–303, 2014.
[146] A. Smolin, G. Eremina, and S. Psakhie, “Modeling Fracture of Nanostructured Bioactive Coatings on Ti-based Materials under Contact Loading,” *Procedia Mater. Sci.*, vol. 3, pp. 621–626, 2014.
[147] M. Wang, D. W. Huang, and R. M. Luo, “Shear failure of tungsten alloy at mesoscale modeled by Movable cellular Automata,” *Theor. Appl. Fract. Mech.*, vol. 56, no. 3, pp. 162–168, 2011.
[148] V. L. Popov, S. G. Psakhie, B. Kehrwald, E. V Shilko, and A. I. Dmitriev, “Wear in combustion engines: experiment and simulation on a basis of movable cellular automaton method,” *Phys. Mesomech.*, vol. 4, no. 4, pp. 71–80, 2001.
[149] V. L. Popov, S. G. Psakhie, and E. V. Shilko, “Friction coefficient in ‘rail - wheel’ ”-contacts as a function of material and loading parameters,” *Phys. Mesomech.*, vol. 5, no. 3, pp. 17–24, 2002.
[150] V. L. Popov, S. G. Psakhie, A. Dmitriev, and E. Shilko, “Quasi-fluid nano-layers at the interface between rubbing bodies: Simulations by movable cellular automata,” *Wear*, vol. 254, no. 9, pp. 901–906, 2003.
[151] A. I. Dmitriev, V. L. Popov, and S. G. Psakhie, “Simulation of surface topography with the method of movable cellular automata,” *Tribol. Int.*, vol. 39, no. 5, pp. 444–449, 2006.
[152] A. I. Dmitriev, M. Schargott, and V. L. Popov, “Direct modelling of surface

topography development in a micro-contact with the movable cellular automata method," *Wear*, vol. 268, no. 7–8, pp. 877–885, 2010.
[153] M. Schargott, V. L. Popov, A. I. Dmitriev, and S. G. Psakhie, "Development of surface topography for the rail-wheel contact.," *Wear*, vol. 265 SRC-, no. 9–10, pp. 1542–1548, 2008.
[154] W. Österle, A. I. Dmitriev, and H. Kloß, "Assessment of sliding friction of a nanostructured solid lubricant film by numerical simulation with the method of movable cellular automata (MCA)," *Tribol. Lett.*, vol. 54, no. 3, pp. 257–262, 2014.
[155] A. I. Dmitriev and W. Österle, "Modelling the sliding behaviour of tribofilms forming during automotive braking: Impact of loading parameters and property range of constituents," *Tribol. Lett.*, vol. 53, no. 1, pp. 337–351, 2014.
[156] A. I. Dmitriev, W. Österle, H. Kloß, and G. Orts-Gil, "A study of third body behaviour under dry sliding conditions. Comparison of nanoscale modelling with experiment," *Esonian J. Eng.*, vol. 18, no. 3, p. 270, 2012.
[157] E. Gnecco and E. Meyer, *Fundamentals of Friction and Wear on the Nanoscale*, Second Edi. Switzerland: Springer International Publishing, 2015.
[158] S. G. Psakhie *et al.*, "Movable cellular automata method for simulating materials with mesostructure," *Theor. Appl. Fract. Mech.*, vol. 37, no. 1–3, pp. 311–334, 2001.
[159] B. C. and M. Droz, *Cellular Automata Modelling of Physical Systems*. Cambridge, UK: Cambridge University Press, 1998.
[160] S. G. Psakhie, G. P. Ostermeyer, A. I. Dmitriev, E. V. Shilko, A. Y. Smolin, and S. Y. Korostelev, "Method of movable cellular automata as a new trend of discrete computational mechanics. 1: Theoretical description," *Phys. Mesomech.*, vol. 3(2), no. 2, pp. 5–12, 2000.
[161] W. Österle and A. I. Dmitriev, "Functionality of conventional brake friction materials - Perceptions from findings observed at different length scales," *Wear*, vol. 271, no. 9–10, pp. 2198–2207, 2011.
[162] A. I. Dmitriev, W. Österle, and H. Kloß, "Numerical simulation of typical contact situations of brake friction materials," *Tribol. Int.*, vol. 41, no. 1, pp. 1–8, 2008.
[163] W. Österle, H. Kloß, and A. I. Dmitriev, "Friction control during automotive braking: experimental observations and simulation at nanometre scale," *Tribol. - Mater. Surfaces Interfaces*, vol. 3, no. 4, pp. 196–202, 2009.
[164] A. I. Dmitriev and W. Österle, "Modeling of brake pad-disc interface with emphasis to dynamics and deformation of structures," *Tribol. Int.*, vol. 43, no. 4, pp. 719–727, 2010.
[165] W. Österle, A. I. Dmitriev, and H. Kloß, "Possible impacts of third body nanostructure on friction performance during dry sliding determined by computer simulation based on the method of movable cellular automata," *Tribol. Int.*, vol. 48, pp. 128–136, 2012.
[166] G. Chen, D. L. Schott, and G. Lodewijks, "Tensile test simulation of high-carbon steel by discrete element method," *Eng. Comput. (Swansea, Wales)*, vol. 33, no. 4, pp. 1224–1245, 2016.
[167] G. Mollon, "A multibody meshfree strategy for the simulation of highly deformable granular materials," *Int. J. Numer. Methods Eng.*, vol. 108, no. 12, pp. 1477–1497, Dec. 2016.
[168] H. Kloss *et al.*, "Computer-aided simulation of material local fracture effects at the area of tribological contacts on the base of methods of discrete

approach," in *ICF11*, 2005, vol. 2, pp. 2–7.
[169] R. Gong, Y. Chen, H. Che, and M. Zhu, "Multiscale simulation of sliding contacts between two rough sealing surfaces," *Proc. Inst. Mech. Eng. Part J J. Eng. Tribol.*, vol. 228, no. 3, pp. 339–351, 2013.
[170] S. G. Psakhie *et al.*, "MCA Demo version: 2D Load Test," *Laboratory of CAD of Materials, Institute of Strength Physics and Materials Science Tomsk, Russia*, 2005. [Online]. Available: http://mech2.pi.tu-berlin.de/popov/software/mca/mca.htm. [Accessed: 21-Jul-2016].
[171] Lorentz Center, "Micro/Nanoscale Models for Tribology: Lorentz Workshop 2017," 2017. [Online]. Available: https://tribomodels2017.sciencesconf.org/.
[172] S. G. Psakhie, E. V. Shilko, A. S. Grigoriev, S. V. Astafurov, and A. V. Dimaki, "A mathematical model of particle – particle interaction for discrete element based modeling of deformation and fracture of heterogeneous elastic – plastic materials," *Eng. Fract. Mech.*, vol. 130, pp. 96–115, 2014.
[173] Piaras Kelly, "Solid Mechanics Part I: An introduction to solid mechanics," Solid mechanics lecture notes, University of Auckland, 2013.
[174] J. Lin, T. Zhu, and L. Zhan, "8 - Constitutive equations for modelling superplastic forming of metals," in *Superplastic Forming of Advanced Metallic Materials*, G. Giuliano, Ed. Woodhead Publishing, 2011, pp. 154–183.
[175] J. Bonet and A. J. Gil, "7 - Finite element modelling of thin metal sheet forming," in *Superplastic Forming of Advanced Metallic Materials*, G. Giuliano, Ed. Woodhead Publishing, 2011, pp. 136–153.
[176] F. Abu-Farha and R. Curtis, "2 - Standards for superplastic forming of metals," in *Superplastic Forming of Advanced Metallic Materials*, G. Giuliano, Ed. Woodhead Publishing, 2011, pp. 34–48.
[177] N. Ridley, "1 - Metals for superplastic forming," in *Superplastic Forming of Advanced Metallic Materials*, G. Giuliano, Ed. Woodhead Publishing, 2011, pp. 3–33.
[178] G. Bernhart, P. Lours, T. Cutard, V. Velay, and F. Nazaret, "3 - Processes and equipment for superplastic forming of metals," in *Superplastic Forming of Advanced Metallic Materials*, G. Giuliano, Ed. Woodhead Publishing, 2011, pp. 49–71.
[179] G. Giuliano, *Superplastic Forming of Advanced Metallic Materials*. Cambridge, UK: Woodhead Publishing, 2011.
[180] J. T. Padding, "Particle-based Simulations," Lecture Notes (6KM59), Technical University Eindhoven, the Netherlands, 2013.
[181] B. W. Ninham, "Long-range vs. short-range forces. The present state of play," *J. Phys. Chem.*, vol. 84, no. 12, pp. 1423–1430, Jun. 1980.
[182] V. E. Panin, "Chapter 13: Interatomic interaction and properties of materials under strong extrenal effects," in *Physical Mesomechanics of Heterogeneous Media and Computer-Aided Design of Materials*, Suffolk, England: Cambridge International Science Publishing, 1998, pp. 271–297.
[183] J. E. Jones, "On the Determination of Molecular Fields. II. From the Equation of State of a Gas," *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 106, no. 738, pp. 463–477, Oct. 1924.
[184] N. Ceder, Gerbrand; Marazi, "Atomistic Computer Modeling of Materials (SMA 5107) | Materials Science and Engineering," *MIT OpenCourseWare*. [Online]. Available: http://ocw.mit.edu/courses/materials-science-and-engineering/3-320-atomistic-computer-modeling-of-materials-sma-5107-spring-2005/. [Accessed: 11-Jun-2016].

[185] Z. Bardber, Ed., *Introduction to Materials Modelling*. The Institute of Materials, Minerals and Mining: Maney, 2005.
[186] M. S. Daw, S. M. Foiles, and M. I. Baskes, “The Embedded-Atom Method: A Review of Theory and Applications,” *Elsevier Sci. Publ.*, vol. N/A, no. September 1992, pp. 251–310, 1993.
[187] N. Martys and R. D. Mountain, “Velocity Verlet algoritm for dissipative-particle-dynamics-based models for suspensions,” *Phys. Rev. E*, vol. 59, no. 3, pp. 3733–3736, 1999.
[188] Y. Li, Y. Xu, and C. Thornton, “A comparison of discrete element simulations and experiments for ‘sandpiles’ composed of spherical particles,” *Powder Technol.*, vol. 160, no. 3, pp. 219–228, 2005.
[189] C. Kloss, “LIGGGHTS(R)-PUBLIC Documentation, Version 3.X — LIGGGHTS v3.X documentation.” [Online]. Available: https://www.cfdem.com/media/DEM/docu/Manual.html.
[190] V. Vedachalam, “Discrete Element Modelling Of Granular Snow Particles Using LIGGGHTS,” PhD Thesis, The University of Edinburgh, UK, 2011.
[191] T. Pöschel and T. Schwager, *Computational Granular Dynamics: Models and Algorithms*. Berlin: Springer, 2005.
[192] Liz Ivoneth Del Cid, “A discrete element methodology for the analysis of cohesive granular bulk solid materials,” PhD Thesis, Colorado School of Mines, 2015.
[193] M. Rackl and K. J. Hanley, “A methodical calibration procedure for discrete element models,” *Powder Technol.*, vol. 307, pp. 73–83, Feb. 2017.
[194] C. J. Coetzee, “Review: Calibration of the discrete element method,” *Powder Technol.*, vol. 310, pp. 104–142, Apr. 2017.
[195] A. P. Grima and P. W. Wypych, “Investigation into calibration of discrete element model parameters for scale-up and validation of particle–structure interactions under impact conditions,” *Powder Technol.*, vol. 212, no. 1, pp. 198–209, Sep. 2011.
[196] Z. Syed, M. Tekeste, and D. White, “A coupled sliding and rolling friction model for DEM calibration,” *J. Terramechanics*, 2017.
[197] T. Roessler and A. Katterfeld, “DEM parameter calibration of cohesive bulk materials using a simple angle of repose test,” *Particuology*, vol. 45, pp. 105–115, Aug. 2019.
[198] Liggghts.com, “LIGGGHTS® Open Source Discrete Element Method Particle Simulation Code.” [Online]. Available: http://cfdem.com/liggghtsr-open-source-discrete-element-method-particle-simulation-code. [Accessed: 21-Jul-2016].
[199] A. Hassanpour and M. Ghadiri, “Characterisation of flowability of loosely compacted cohesive powders by indentation,” *Part. Part. Syst. Charact.*, 2007.
[200] C. Kloss, A. Aigner, “LIGGGHTS Short Course,” Linz, 2016.
[201] C. Recarey *et al.*, “Advances in particle packing algorithms for generating the medium in the Discrete Element Method,” *Comput. Methods Appl. Mech. Eng.*, vol. 345, pp. 336–362, Mar. 2019.
[202] C. Padros, “Discrete element simulations with LIGGGHTS,” Master’s Thesis, Swansea University, 2014.
[203] D. C. Hong and J. A. McLennan, “Molecular dynamics simulations of hard sphere granular particles,” *Phys. A Stat. Mech. its Appl.*, 1992.
[204] D. C. Richardson, K. J. Walsh, N. Murdoch, and P. Michel, “Numerical simulations of granular dynamics: I. Hard-sphere discrete element method

and tests," *Icarus*, vol. 212, no. 1, pp. 427–437, Mar. 2011.
[205] S. Luding, "From molecular dynamics and particle simulations towards constitutive relations for continuum theory," in *Lecture Notes in Computational Science and Engineering*, vol. 71, Springer Verlag, 2010, pp. 453–492.
[206] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, and A. B. Yu, "Discrete particle simulation of particulate systems: Theoretical developments," *Chem. Eng. Sci.*, vol. 62, no. 13, pp. 3378–3396, 2007.
[207] A. Munjiza, T. Bangash, and N. W. M. John, "The combined finite-discrete element method for structural failure and collapse," *Eng. Fract. Mech.*, vol. 71, no. 4–6, pp. 469–483, 2004.
[208] P. K. Haff and B. T. Werner, "Computer simulation of the mechanical sorting of grains," *Powder Technol.*, 1986.
[209] J. C. Gallas, H. J. Herrmann, and S. Sokołowski, "Convection cells in vibrating granular media," *Phys. Rev. Lett.*, 1992.
[210] O. Walton and R. L. Braun, "Viscosity, granular-temperature, and stress calculations for shearing assemblies of inelastic, frictional disks," *J. Rheol. (N. Y. N. Y).*, vol. 30, p. 949, 1986.
[211] S. Luding, N. Rivas, and T. Weinhart, "From soft and hard particle simulations to continuum theory for granular flows," in *Multi Scale Mechanics*, Alliance of Laboratories in Europe for Education, Research and Technology, 2017, pp. 3–42.
[212] A. Podlozhnyuk, S. Pirker, and C. Kloss, "Efficient implementation of superquadric particles in Discrete Element Method within an open-source framework," *Comput. Part. Mech.*, 2017.
[213] J. Shäfer, S. Dippel, and D. E. Wolf, "Force Schemes in Simulations of Granular Materials," *J. Phys. I*, 1996.
[214] M. Marigo, "Discrete Element Method Modelling of Complex Granular Motion in Mixing Vessels: Evaluation and Validation," PhD Thesis, The University of Birmingham, 2012.
[215] D. Mukherjee, "Discrete Particle Simulation Techniques for the Analysis of Colliding and Flowing Particulate Media," PhD Thesis, University of California, Berkley, 2013.
[216] C. Kloss, C. Goniva, A. Hager, S. Amberger, and S. Pirker, "Models, algorithms and validation for opensource DEM and CFD-DEM," *Prog. Comput. Fluid Dyn. An Int. J.*, vol. 12, no. 2/3, p. 140, 2012.
[217] J. Fleischmann and J. Fleischmann, "DEM-PM Contact Model with Multi-Step Tangential Contact Displacement History," 2015.
[218] A. Di Renzo and F. P. Di Maio, "Comparison of contact-force models for the simulation of collisions in DEM-based granular flow codes," *Chem. Eng. Sci.*, vol. 59, no. 3, pp. 525–541, 2004.
[219] M. Jebahi, D. André, I. Terreros, and I. Iordanoff, *Discrete Element Method to Model 3D Continuous Materials*. John Wiley & Sons, 2015.
[220] H. Deresiewicz and Raymond D Mindlin, "Elastic spheres in contact under varying oblique forces," *ASME J. Appl. Mech. Appl. Mech.*, vol. 20, pp. 327–344, 1952.
[221] R. Tayeb, X. Dou, Y. Mao, and Y. Zhang, "Analysis of cohesive microsized particle packing structure using history-dependent contact models," *J. Manuf. Sci. Eng. Trans. ASME*, 2016.
[222] J. Ai, J. F. Chen, J. M. Rotter, and J. Y. Ooi, "Assessment of rolling resistance models in discrete element simulations," *Powder Technol.*, 2011.

[223] H. P. Zhu, Z. Y. Zhou, R. Y. Yang, and A. B. Yu, “Discrete particle simulation of particulate systems: A review of major applications and findings,” *Chem. Eng. Sci.*, vol. 63, no. 23, pp. 5728–5770, 2008.
[224] C. Ramírez-Aragón, J. Ordieres-Meré, F. Alba-Elías, and A. González-Marcos, “Comparison of Cohesive models in EDEM and LIGGGHTS for simulating powder compaction,” *Materials (Basel).*, 2018.
[225] C. Thornton and Z. Ning, “A theoretical model for the stick/bounce behaviour of adhesive, elastic- plastic spheres,” *Powder Technol.*, vol. 99, no. 2, pp. 154–162, Sep. 1998.
[226] J. P. Morrissey, “Discrete Element Modelling of Iron Ore Pellets to Include the Effects of Moistureand Fines,” PhD Thesis, University of Edinburgh, 2013.
[227] J. Loh, W. R. Ketterhagen, and J. A. Elliott, “A hybrid tabulation-scaling implementation of Thornton and Ning’s plastic-adhesive particle contact theory,” *Powder Technol.*, 2014.
[228] S. Luding, “Cohesive, frictional powders: Contact models for tension,” *Granul. Matter*, vol. 10, p. 235, 2008.
[229] M. Pasha, S. Dogbe, C. Hare, A. Hassanpour, and M. Ghadiri, “A linear model of elasto-plastic and adhesive contact deformation,” *Granul. Matter*, vol. 16, no. 1, pp. 151–162, 2014.
[230] O. R. Walton and S. M. Johnson, “Simulating the effects of interparticle cohesion in micron scale powders,” *AIP Conf. Proc.*, vol. 1145, no. 1, p. 897, 2009.
[231] N. Brown, “Discrete element modelling of cementitious materials,” PhD Thesis, University of Edinburgh, 2013.
[232] M. Obermayr, K. Dressler, C. Vrettos, and P. Eberhard, “A bonded-particle model for cemented sand,” *Comput. Geotech.*, vol. 49, pp. 299–313, 2013.
[233] D. O. Potyondy and P. A. Cundall, “A bonded-particle model for rock,” *Int. J. Rock Mech. Min. Sci.*, vol. 41, no. 8 SPEC.ISS., pp. 1329–1364, 2004.
[234] A. Truszkowska, Q. Yu, P. A. Greaney, T. M. Evans, and J. J. Kruzic, “Developing a Crystal Plasticity Model for Metallic Materials Based on the Discrete Element Method,” in *MRS Advances*, 2017.
[235] A. Truszkowska, Q. Yu, P. A. Greaney, T. M. Evans, and J. J. Kruzic, “A discrete element method representation of an anisotropic elastic continuum,” *J. Mech. Phys. Solids*, vol. 121, pp. 363–386, Dec. 2018.
[236] A. Mayrhofer, A. Hager, and C. Kloss, “Energy Balance in a Particle System,” in *V International Conference on Particle-based Methods – Fundamentals and Applications : PARTICLES 2017*, 2017.
[237] M. Latzel, “From microscopic simulations towards a macroscopic description of granular media,” Universty of Stuttgart, 2003.
[238] N. Kumar, “Micro-Macro and jamming transition in granular materials,” PhD Thesis, Universiteit Twente, 2014.
[239] A. Singh, “Micro-macro and rheology in sheared granular matter,” PhD Thesis, University of Twente, 2014.
[240] L. Benvenuti, C. Kloss, and S. Pirker, “Identification of DEM simulation parameters by Artificial Neural Networks and bulk experiments,” *Powder Technol.*, vol. 291, pp. 456–465, 2016.
[241] C. J. Coetzee, “Calibration of the discrete element method and the effect of particle shape,” *Powder Technol.*, vol. 297, pp. 50–70, 2016.
[242] P. Stempfle, A. Domatti, H. A. Dang, and J. Takadoum, “Mechanical and chemical wear components in environmental multi-asperity nanotribology,”

*Tribol. Int.*, vol. 82, no. PB, pp. 358–374, 2015.
[243] A. Mikhailov, *Foundations of Synergetics I. Distributed Active Systems*. Berlin: Springer Verlag, 1990.
[244] N. Wiener and A. Rosenblueth, “The mathematical formulation of the problem of conduction of impulses in a network of connected excitable elements, specifically in cardiac muscle,” *Arch. Inst. Cardiol. Mex.*, vol. 16, no. 3, pp. 205–265, 1946.
[245] M. S. Daw and M. I. Baskes, “Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals,” *Phys. Rev. B*, vol. 29, no. 12, pp. 6443–6453, 1984.
[246] A. Y. Smolin, N. V. Roman, S. A. Dobrynin, and S. G. Psakhie, “On rotation in the movable cellular automaton method,” *Phys. Mesomech.*, vol. 12, no. 3–4, pp. 124–129, May 2009.
[247] S. G. Psakhie, A. Y. Smolin, Y. P. Stefanov, P. V. Makarov, and M. A. Chertov, “Modeling the behavior of complex media by jointly using discrete and continuum approaches,” *Tech. Phys. Lett.*, vol. 30, no. 9, pp. 712–714, 2004.
[248] R. M. Jones, *Deformation Theory of Plasticity*. Blacksburg, Virginia: Bull Ridge Publishing, 2009.
[249] M. L. Wilkins, *Computer simulation of dynamic phenomena*. Berlin: Springer-Verlag Berlin Heidelberg, 1999.
[250] M. L. Wilkins, *Calculations of elastic-plastic flow*, vol. 3. New York: Academic Press, 1964.
[251] A. Y. Smolin, E. V. Shilko, S. V. Astafurov, E. A. Kolubaev, G. M. Eremina, and S. G. Psakhie, “Understanding the mechanisms of friction stir welding based on computer simulation using particles,” *Def. Technol.*, vol. 14, no. 6, pp. 643–656, 2018.
[252] C. Kloss and C. Goniva, “LIGGGHTS - Open Source Discrete Element Simulations of Granular Materials Based on Lammps,” in *Supplemental Proceedings: Materials Fabrication, Properties, Characterization, and Modeling*, 2011.
[253] Sandia.gov, “LAMMPS Molecular Dynamics Simulator.” [Online]. Available: lammps.sandia.gov. [Accessed: 21-Jul-2016].
[254] S. Plimpton, “Fast Parallel Algorithms for Short–Range Molecular Dynamics,” *J. Comput. Phys.*, vol. 117, no. 1, pp. 1–19, 1995.
[255] R. Berger, C. Kloss, A. Kohlmeyer, and S. Pirker, “Hybrid parallelization of the LIGGGHTS open-source DEM code,” *Powder Technol.*, vol. 278, pp. 234–247, 2015.
[256] C. Kloss, “LIGGGHTS.” .
[257] SandiaLaboratory, “Lammps Developer Guide,” 2011.
[258] C. Richer, “LIGGGHTS-WITH-BONDS: LIGGGHTS® DEM software with Bonds enabled.This offering is not approved or endorsed by DCS Computing GmbH, the producer of the LIGGGHTS® and CFDEM®coupling software and owner of the LIGGGHTS and CFDEM® trade marks.,” 2015. [Online]. Available: https://github.com/richti83/LIGGGHTS-WITH-BONDS. [Accessed: 07-Oct-2019].
[259] G. C. Ganzenmueller, “lammps/src/USER-SMD,” 2014. [Online]. Available: https://github.com/lammps/lammps/tree/master/src/USER-SMD.
[260] R. Berger, “Lennard-Jones MD code in Fortran for teaching purposes,” 2013. [Online]. Available: https://github.com/rbberger/ljmd-f.
[261] E. J. Rykiel, “Testing ecological models: The meaning of validation,” *Ecol.*

*Modell.*, 1996.
[262] A. Smolin and N. Salman, “LIGGGHTS-MCA.” [Online]. Available: https://github.com/AlexeySmolin/LIGGGHTS-MCA. [Accessed: 21-Jul-2016].
[263] NIST, “Tensile and Microindentation Stress-Strain Curves of Al-6061,” 2016. [Online]. Available: https://materialsdata.nist.gov/handle/11256/774.
[264] J. S. Weaver, A. Khosravani, A. Castillo, and S. R. Kalidindi, “High throughput exploration of process-property linkages in Al-6061 using instrumented spherical microindentation and microstructurally graded samples,” *Integr. Mater. Manuf. Innov.*, 2016.
[265] F. Fröhlich, P. Grau, and W. Grellmann, “Performance and analysis of recording microhardness tests,” *Phys. status solidi*, 1977.
[266] H. V. Lawn BR, “Elastic recovery at hardness indentations.,” *J. Mater. Sci.*, vol. 16, pp. 2745–52, 1981.
[267] E. S. Berkovich, “Three-Faceted Diamond Pyramid for Studying Microhardness by Indentation,” *Zavod. Lab.*, 1950.
[268] B. Bushan, *Springer handbook of nanotechnology*. Springer Science and Business media, 2010.
[269] W. C. Oliver and G. M. Pharr, “Measurement of hardness and elastic modulus by instrumented indentation: Advances in understanding and refinements to methodology,” *J. Mater. Res.*, 2004.
[270] V. Šmilauer, “Cohesive Particle Model Using the Discrete Element Method on the Yade Platform,” PhD Thesis, Université de Grenoble, Czech Technical University in Prague, 2010.