Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

Escola Tècnica Superior d'Enginyers de Camins
Canals i Ports de Barcelona

Departament de Resistència de Materials i
Estructures a l'Enginyeria

Master in Numerical Methods in Engineering

M.Sc. Thesis

Kratos DEM, a Parallel Code for Concrete Testing Simulations using the Discrete Element Method

Author:
Miquel Santasusana

Supervisors:
Dr. Eugenio Oñate
Eng. Miguel Ángel Celigueta

November 21, 2013

Abstract

The Discrete Element Method is a relatively new technique which has nowadays an intense research in the field of numerical methods. In its first conception, the method was designed for simulations of dynamic systems of particles where each element is considered to be an independent and non deformable entity which interacts with other particles by the laws of the contact mechanics and moves following the second Newton's law. This first approach for the DEM has obtained excellent results for granular media simulations or other discontinua applications where actual particles take part on it.

Nowadays, the DEM is still not able to be a discretization method when dealing with continuum simulations, especially in the field of solid mechanics, where lots of efforts have been done. Even though it has many drawbacks and lack of reliability in terms of simulation of solids, the method is promising for applications of rock excavations for instance, where phenomena such as multifracturing are nowadays impossible to reproduce by any classical FEM-based method.

The aim of the thesis is to show the limitations of the method for the general solid mechanics problem as well as the capabilities of the method for the specific problem of fracturing geomaterials, namely rock or concrete. As will be shown, under some specific conditions the DEM can be a good choice for solving the described problem.

The final goal would be developing a DEM tool for engineering problems in rock excavation applications, drilling, fracking, demolition, etc. Firstly, a set of analysis have to be established as benchmark test for different phenomena that permit the validation of the method. Uniaxial Compressive Strength test, Brazilian Tensile Strength test, Triaxial test for concrete, etc. are performed with the developed software and the presented theory in order to validate the method with experimental results.

The thesis supports the development of simulation software called KDEM, coded in C++ language in the open-source Kratos platform (*http://www.cimne.com/kratos*). The platform provides a framework for developing multiphysics FEM-based codes and offers several tools for coupling different codes. In this sense, the aim of the author of the thesis is to make use of this possibility to implement the coupling between the two methods once the DEM application will be reliable enough.

Resum

El Mètode dels Elements Discrets és un mètode relativament nou el qual avui dia és objecte d'una intensa recerca en el món dels mètodes numèrics. Originalment el mètode fou concebut per a la simulació de sistemes dinàmics de partícules on cada element és considerat com a entitat independent i indeformable que interacciona amb les altres seguint les lleis del contacte mecànic i es mou segons la segona llei de Newton. Aquest primer plantejament sobre el M.E.D. ha tingut molts bons resultats per a simulacions de medis granulars o qualsevol assimilable a un medi discontinu.

Avui dia, el M.E.D. encara no és capaç d'ésser un mètode de discretització quan ens referim a simulacions de medis continus, especialment en l'àmbit de la mecànica de sòlids computacional malgrat els esforços realitzats en aquest sentit. Tot i que presenta grans desavantatges i pobres resultats quan s'aplica a simulacions de sòlids, el mètode és prometedor per aplicacions d'excavació de roca per exemple, on fenòmens com ara la multifractura son presents i es desconeix la forma de reproduir-los avui en dia mitjançant mètodes clàssics basats en elements finits.

El propòsit de la tesis és mostrar les limitacions del mètode per aplicacions genèriques de mecànica de sòlids, alhora que exposar les capacitats del mètode per al problema específic de la fractura en geomaterials, com son la roca o el formigó. Així doncs, es mostrarà com, sota certes condicions, el M.E.D. és una bona opció per resoldre el problema plantejat.

L'objectiu final seria el desenvolupament d'una utilitat amb el M.E.D. per aplicacions enginyerils en problemes d'excavació de roca, perforació, fractura hidràulica, demolició, etc. Primerament, un conjunt d'anàlisis han d'establir-se com a bancs de probes on diferents fenòmens es puguin reproduir experimentalment i poder així validar el mètode proposat. El Test de Compressió Uniaxial, el Test Brasiler, el Test Triaxial per al formigó, etc. seran realitzats mitjançant el programari desenvolupat per mitjà de la teoria aquí descrita per tal de validar el mètode comparant amb resultants experimentals.

La tesis recolza el desenvolupament d'un programa de simulació anomenat KDEM, escrit amb llenguatge C++ en la plataforma de codi obert Kratos (*http://www.cimne.com/kratos/*). La plataforma és un marc de treball per al desenvolupament de codis basats en Elements Finits de caire multifísic I ofereix diverses utilitats que permeten l'acoblament de varis codis. En aquest sentit, el propòsit de l'autor de la tesis és fer ús d'aquesta plataforma per tal de poder implementar en un futur l'acoblament dels dos mètodes (M.E.F. i M.E.D.) un cop el M.E.D funcioni de forma adequada.

Acknowledgements

This document is the thesis of my studies of the *Master in Numerical Methods In Engineering - Màster en Mètodes Numèrics en l'Enginyeria* hold in the engineering school *Escola Tècnica Superior d'Enginyers de Camins Canals i Ports de Barcelona* in junction with the center *Centre Internacional de Mètodes Numèrics en l'Enginyeria (CIMNE).*

This thesis is result of one year and a half of work in the field of the Discrete Element Method. I started getting to know the method in the course of the over mentioned Master and I had the opportunity, at that time, to join the DEM team in CIMNE. At that time, the author of the available DEM code in CIMNE left the group; *Dempack*, his code, has been used for real engineering projects but has not been further developed. My objective was to implement a new general purpose Discrete Element Method code in the open-source platform *Kratos.* The mainly differences against the previous code, *Dempack* are two: On one hand, the possibility now to paralelize the code in both shared memory and distributed memory technology, on the other hand the capacity to combine the code with other FEM-based codes in any field, i.e fluid dynamics or computational solid mechanics in *Kratos.*

The work done during the first year is collected in my Bachelor Degree Thesis - *Continuum modelling using the Discrete Element Method. Theory and implementation in an object-oriented software platform.* In that document the basic idea of the implementation of the code in Kratos is presented as well as the philosophy of the continuum theory applied to the Discrete Element Method. Obviously is highly recommended to take a look on that document before proceding with the present thesis which covers more advanced topics about the use of the DEM to continuum fields and more specifically its application to concrete test simulations.

Like most of written texts, which although individually written are collaborative works, this study has been possible thanks to the help of many people who I would like to thank. First of all I would like to thank Eugenio Oñate Ibañez de Navarra for the trust placed in me when giving me the opportunity of forming part of the *Kratos* team in CIMNE. Also I want to thank Miguel Angel Celigueta for being my supervisor and providing me instructive comments and directing the steps of the *KDEM.* Among all the people who have collaborated with this project or helped me doing my work in CIMNE I would like to mention Pooyan Davdan and Carles Roig Pina.

Last but not the least, I would like to thank my parents, Maria Àngels Isach and Josep Santasusana, and my sisters Montserrat and Marina for all the support that they have given to me through all my studying life. They have always provided me the necessary ambition needed to accomplish my goals.

Contents

Chapter 1

Introduction

This dissertation is the result of the implementation of a Discrete Element Method code in an open source object-oriented software platform called Kratos developed in CIMNE (Barcelona). The result of this work is the so-called *Kratos DEM-Application (KDEM)* (*http://www.cimne.com/Kratos/*), which is the program that has been coded by the author forming part of a team of engineers in CIMNE.

After the introduccion and the objectives, a brief review on the Discrete Element Method in its basic conception is presented in the third chapter of the document while the theoretical developments and discussions for the application of the method to continous media, specially to conctrete testing simulation, can be found in the fourth chapter. Also here, special attention is paid to the capabilities of this method when it is applied to continuous media; several numerical analisis are performed here to show the possibilities and limitations of the method in the fifth chapter.

In the sixth chapter, the Kratos framework is introduced and the basic structure of the developed application is explained. The implementation of the utilities that permit, from a user-friendly interface, perform concrete test simulations can be found here. Furthermore, the possibilities and advatages that the Kratos framework provides the DEM-Application are shown as well as examples of how the code behaves in terms of High Performance Computing. Additionally, examples of coupling of this application with other codes and work done by other researchers in the institution are here presented. In the seventh chapter, the results of the different concrete simulation tests can be found as a verification of the code comparing with the results obained by Dempack. Finally, in the last chapter some conclusions and future work is presented.

The objective of the KDEM, in its conception, was to have a base program for the DEM coded in a very powerful and versatile platform, the open-source multiphysics code Kratos. This permits different researchers extending and improving the code as well as using as a closed package for projects and simulation by advanced users and engineers.

Previous to this, in CIMNE, another code based on the Discrete Element Method has been used for engineering projects, *Dempack*. When the KDEM code started the purpose was to provide CIMNE a general DEM application able to be parallelized and to be combined with other fluid or structure applications. At the same time, a lot of implementation has been done in order to substitute, and improve if possible, the existing code Dempack; part of this thesis focuses on the concrete test simulation, a field where CIMNE has currently projects ongoing.

The result of this work accomplishes the objective of providing Kratos a general purpose paralelizable Discrete Element Method application which is the of the interest of CIMNE. In a second term, it presents the first promising results in the field of Concrete Test Simulation, following the objective of substituing in the future, the existing application, Dempack.

Chapter 2

Objectives

The present thesis involves work in two layers; one one hand theoretical research in the DEM, on the other hand, code implementation in a computer application. Additionally, validation of the code and comparison with experimental data is an important point that completes the objectives presented in the following list:

Theoretical research on the method

1. Basis of the model
   (a) Contact search and characteritzation
   (b) Constituitive modelling
   (c) Integration of the equations of motion
2. Advanced issues
   (a) Continuum modelling of frictional cohesive materials
   (b) Fracture criteria
   (c) Stress and strain tensors calculation

KDEM Software implementation

1. General purpose DEM software
   (a) Discrete media simulation
   (b) Continuum solid mechanics applications
   (c) Particle interaction with other media (fluid, structure, etc.)
2. Implementation in Kratos platform
   (a) Kratos structure, C++ and Python language
   (b) Performance, code optimitzation
   (c) Coupling with fluids, solids, etc.
   (d) HPC: Shared memory and distributed memory parallelitzation
3. User friendly interface with GID
   (a) Preprocess: Model definition, mesh, applying conditions and options menu
   (b) GiD - KDEM interaction
   (c) Postprocess: Visualization of results, nodal-wise, particle-wise and contact-wise

Validation and code testing

1. Concrete test simulation
   - (a) Performance of Uniaxial Compressive Strenght test
   - (b) Performance of Brazilian Tensile Strenght test
   - (c) Performance of Triaxial tests
2. Validation examples and verification against Dempack
   - (a) Mesh dependence
   - (b) Convergence
   - (c) Linear elasticity
   - (d) Quasi staticity

Chapter 3

The Discrete Element Method - The Basis

3.1 Introduction

The Discrete Element Method (DEM) was firstly introduced by Cundall (1971) [2] for the analysis of the fracture mechanics problems and, afterwards, it was applied to solids by Cundall and Strack (1979) [3]. The DEM in its basical conception simulates the mechanical behaviour of a system formed by a set of particles arbitrarily disposed. The method considers the particles to be discrete elements forming part of a higher more complex system. Each distinct element has an independent movement; they interact each other due to the contacts. The overall behaviour of the system is determined by the cohesive/frictional contact laws.

Although the DEM is an effective and powerful numerical technique for reproducing the behaviour of granular material, in recent years the DEM has also been object of intense research for its applictaion to the study of multifracture and failure of solids involving geomaterials (soils, rocks, concrete), masonry and ceramic materials, among others. Some key developments including the contributions of CIMNE researchers, can be found in [5], [12], [17], [15], [20], [26].

In the continuum sense, the contact law can be seen as the formulation of the material model on the microscopic level. Cohesive bonds can be broken, which allows to simulate fracture of material and its propagation. The analysis of solid materials with the DEM poses, however, a number of difficulties for adequate reproducing the correct constitutive behaviour of the material under linear (elastic) and non linear conditions.

Basically, the Discrete Element Method algorithm, from a computational point of view, is based on three basic steps:

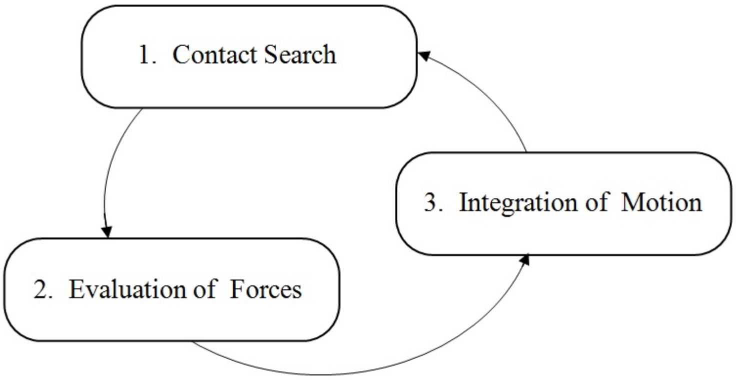


Figure 3.1: Basic computational scheme for the DEM

3.2 Contact Search

In contrast with the FEM-based methods, the DEM, has no connectivities between the nodes by means of the elements. So the transfer of properties such as forces, and in consequence, accelerations, velocities and displacement are possible for the contact criteria between the pair of elements involved. The contact is determined, in the most classical way, when one body belonging to a discrete element intersects with another body which defines another discrete element. The contact search is a very important part of the method in terms of computational cost (range 60%-90% of simulation time in simulations with large number of particles) and it is possibly the most difficult part to treat when dealing with particles that have no spherical/circular shape.

The contact detection basically consists in determining, for every target object, what other objects are in contact with, and then, judge for the correspondent interaction. Normally, objects move freely and the contact is determined when an overlap occurs, and so, then they must interact. It is usually desired to have a very low overlapping 0.1%∼1% to have realistic results, but of course, it depends on the time step selection and the dynamism (velocity) of the particle/objects. Besides this, and due to the fact that this is an expensive step, as previously said, its logical to limit the search of neighbours/contacts only when it is necessary. Obviously there is no need to update the contacts at every time step of the calculation (if the time step is considerably small, the neighbours will be the same from several time step calculations) but, if the search is too much delayed, it can happen to suddenly find large indentation on a new contacting pair; so the repulsion forces would be too large, therefore there would be a huge amount of “created” energy that would lead to numerical inestabilities or unrealistic results. This phenomenon and the recommended solution by the author (buffer zone) are explained on the previous work [28].

The most naïve method of search that can be set is the *brute search*. For every element, the method does a loop for all other elements checking for the contact. The order of the number of operations needed is quadratic: n^2. Normally, the strategy to avoid such an expensive scheme is to divide the contact search in two basic stages, a global search and a subsequent local resolution of the contact. In this case the computation time of the contact search is proportional to $n \log n$:

Global Contact Search

It consists on locating the list of potential contact objects for each given target body. There are two different basic schemes: the Grid/Cell based algorithms or the Tree based ones. There are numerous different methods and variations for each type.

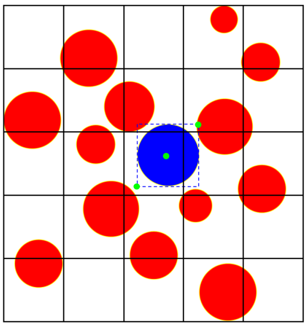


Figure 3.2: Grid/Cell-based algorithm structure

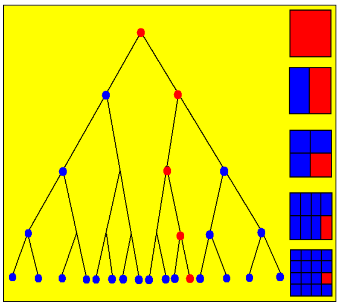


Figure 3.3: Tree-based algorithm structure

In the *Grid based algorithms* a general rectangular grid is defined dividing in cells the entire domain; a unified bounding box (rectangular or spheric) is adopted to represent the discrete elements that can be of any shape; the potential contacts are determined by selecting the surrounding cells where each target body is centred on.

In the *Tree based algorithms* each element is represented by a point. Starting from a centred one, it splits the domain in two sub domains obtaining points that have larger X coordinate in one sub domain and points with smaller values of the X in the other sub domain. The method proceeds for next points alternating every time the splitting dimension and obtaining a tree structure like the one shown in Figure 3.3 tree structure. Once the tree is constructed, for every particle, the nearest neighbours have to be determined following the tree in upwind direction.

Local Resolution Check

The objective is to establish the actual contact configuration. Starting from the potential contacts or areas found in the global contact search, now the contact is analysed in detail. This is the difficult and expensive part of the contact detection; even for simple polygonal shapes the detection criteria is not trivial. The complexity is much higher for 3D cases, which are the most frequent ones. For spheres, however, this is fast and easy.

For a more extensive discussion about this topic and information of how the KDEM application does the search refer to the previous work of the author [28].

3.3 Evaluation of Forces

3.3.1 Decomposition of the contact force

Once contact between a pair of elements has been detected (Figure 3.4), the forces occurring at the contact point are calculated. The interaction between the two interacting spheres can be represented by the contact forces $\mathbf{F}^{ij}$ and $\mathbf{F}^{ji}$ (Figure 3.5), which satisfy the following relation:

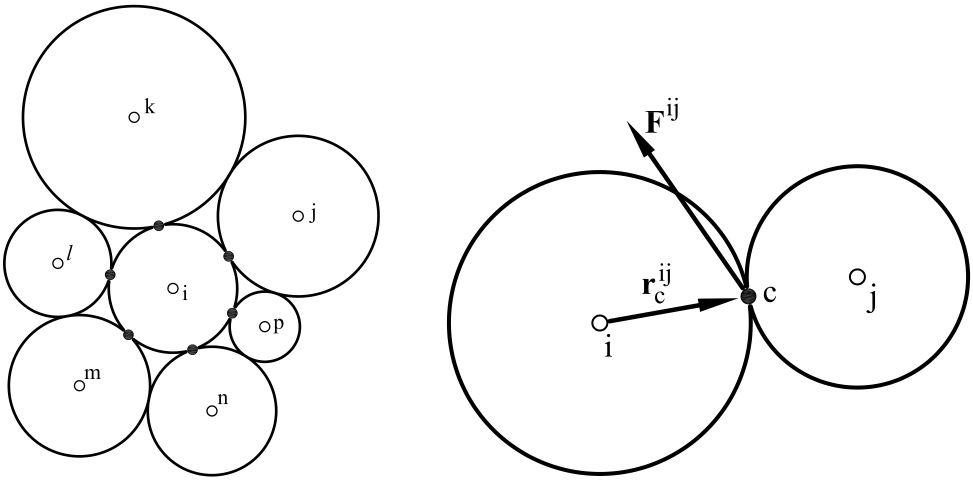


Figure 3.4: Force $\mathbf{F}^{ij}$ at the contact interface between particles i and j

$$\mathbf{F}^{ij} = -\mathbf{F}^{ji} \tag{3.1a}$$

The force can be decomposed $\mathbf{F}^{ij}$ into the normal and shear components, $\mathbf{F}_n^{ij}$ and $\mathbf{F}_s^{ij}$, respectively (Figure 3.5)

$$\mathbf{F}^{ij} = \mathbf{F}_n^{ij} + \mathbf{F}_s^{ij} = F_n \mathbf{n}^{ij} + \mathbf{F}_s^{ij} \tag{3.1b}$$

where $\mathbf{n}^{ij}$ is the unit vector normal to the particle surface at the contact point between particles i and j. This implies that it lies along the line connecting the centers of the two particles and directed outwards from particle i.

$$\mathbf{n} = \frac{\mathbf{x}_j - \mathbf{x}_i}{||\mathbf{x}_j - \mathbf{x}_i||} \tag{3.1c}$$

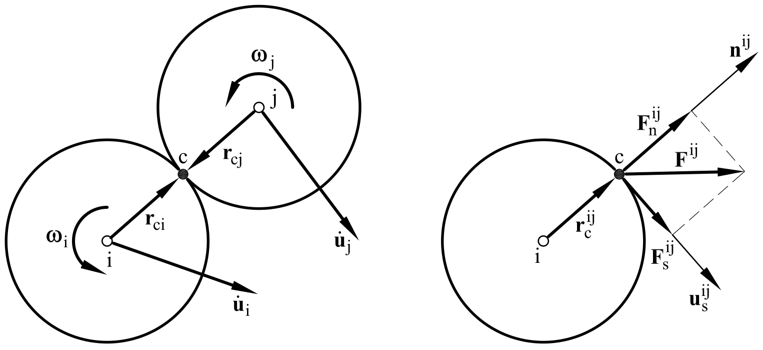


Figure 3.5: Decomposition of the contact force into normal and tangential components

The contact forces F_n, F_{s_1} and F_{s_2} are obtained using a constitutive model formulated for the contact between two rigid spheres (or discs in 2D). The contact interface for the simpliest formulation is characterized by the normal and tangential stiffness k_n and K_s, respectively, a friccional device obeying the Couloumb law with friccion coefficient μ, and a dashpot defined by the contact damping coefficient c_n as shown in the next figure 3.6:

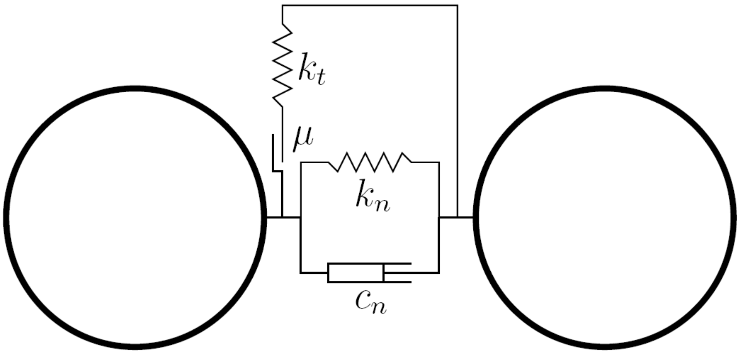


Figure 3.6: Rehological model of the contact

By now, just the simple DEM model is presented; this one is valid for the discrete media simulation and in general any particle interaction phenomena. Obviously every specific model can define more complicated rheologies in order to reproduce the physics of the problem. In this work, for instance the non linear friccional contact with the damage, plasticity and fracture criteria will be presented and explained in detail in the next chapter 4.

3.3.2 Normal interaction force

In the basic DEM the normal contact force F_n is decomposed into the elastic part F_{ne} and the damping contact force F_{ne}:

$$F_n = F_{ne} + F_{nd} \tag{3.2}$$

The damping part is used to decrease oscillations of the contact forces and to dissipate kinetic energy. This is necessary in most DEM codes since an explicit time scheme is used for the integration of motion (see section 3.4.2) acting as a numeric damping; furthermore this can also represent a physical damping on the system defined by the material properties.

Normal elastic force

The elastic part of the normal compressive contact force F_{ne} is, in the basic model, proportional to the normal stiffness k_n and to the indentation (or penetration) of the two particle surfaces, i.e.:

$$F_{ne}^c = k_n u_{rn} \tag{3.3}$$

The superindex c indicates that these may vary on each contact. The indentation is calculated as:

$$\dot{u}_{nc} = (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_i) \cdot \mathbf{n}^{ij} \tag{3.4}$$

where $\mathbf{x_j}, \mathbf{x_i}$ are the center of particles, $\mathbf{n}$ the normal unit vector between the particles (Eq. 2.7), and r_j, r_j their radius. If there is no cohesion allowed (chapter 4), no tensile normal contact forces are allowed and hence:

$$F_{ne} \leq 0 \tag{3.5}$$

So, If $u_{rn} < 0$, Eq. 3.3 holds, otherwise $F_{ne} = 0$. The contact with cohesion will be considered in the next chapter 4.

Normal contact damping

The contact damping force is assumed to be of viscous type and given by:

$$F_{nd} = c_n \cdot v_{rn} \tag{3.6}$$

where v_{rn} is the normal relative velocity of the centres of the two particles in contact, defined by:

$$v_{rn} = (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_i) \cdot \mathbf{n} \tag{3.7}$$

The damping coefficient c_n is taken as a fraction ξ of the critical damping $\bar{c}$ for the system of two rigid bodies with masses m_i and m_j connected by a spring of stiffness k_n with:

$$c = \xi\bar{c} = 2\xi\sqrt{m_{ij}k_n} \tag{3.8}$$

with $0 < \xi \leq 1$ and m_{ij} is the equivalent mass of the contact,

$$m_{ij} = \frac{m_i m_j}{m_i + m_j} \tag{3.9}$$

The fraction ξ is related with the *coefficient of restitution* c_r, which is a fractional value representing the ratio of speeds after and before an impact, through the following expression (see [11]):

$$\xi = \frac{-\ln c_r}{\sqrt{\pi^2 + \ln^2 c_r}} \tag{3.10}$$

In this work, when dealing with the continuum simulations, the adopted value is $\xi = 0.9$, which corresponds to a $c_r \approx 1 \cdot 10^{-3}$, assuming a quasi-static state for the simulated processes.

3.3.3 Tangential frictional contact

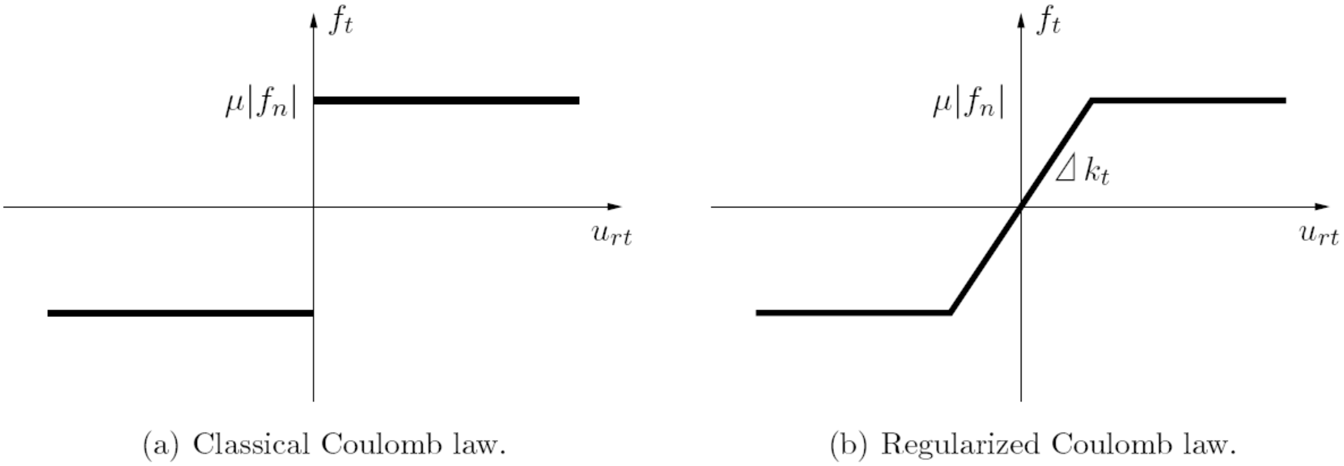
In the absence of cohesion (if the particles were not bonded at all or the initial cohesive bond has been broken) the tangential reaction $\mathbf{F}_t$ appears by friction opposing the relative motion at the contact point. The relative tangential velocity at the contact point vrt is calculated from the following relationship:

$$\mathbf{v}_{rt} = \mathbf{v}_r - (\mathbf{v}_r \cdot \mathbf{n})\mathbf{n} \tag{3.11}$$

with

$$\mathbf{v}_r = (\dot{\mathbf{u}}_j + \omega_j \times \mathbf{r}_{cj}) - (\dot{\mathbf{u}}_i + \omega_i \times \mathbf{r}_{ci}) \tag{3.12}$$

where $\dot{\mathbf{u}}_i, \dot{\mathbf{u}}_j$, and ω_i, ω_j are translational and rotational velocities of the particles, and $\mathbf{r}_{ci}$ and $\mathbf{r}_{cj}$ are the vectors connecting particle centres with contact points.



(a) Classical Coulomb law.

(b) Regularized Coulomb law.

Figure 3.7: Friction force vs. relative tangential displacement

The relationship between the friction force $\|\mathbf{F}_s\|$ and the relative tangential displacement u_{rT} for the classical Coulomb model (for a constant normal force F_n) is shown in Fig. 3.7a. This relationship would produce non physical oscillations of the friction force in the numerical solution due to possible changes of the direction of sliding velocity. To prevent this, the Coulomb friction model must be regularized. The regularization procedure chosen involves decomposition of the tangential relative velocity into reversible and irreversible parts, $\mathbf{v}_{rT}^{\mathrm{r}}$ and $\mathbf{v}_{rT}^{\mathrm{ir}}$, respectively as:

$$\mathbf{v}_{rT} = \mathbf{v}_{rT}^{\mathrm{r}} + \mathbf{v}_{r}^{\mathrm{ir}}\,. \tag{3.13}$$

This is equivalent to formulating the frictional contact as a problem analogous to that of elastoplasticity, which can be seen clearly from the friction force-tangential displacement relationship in Fig. 3.7b. This analogy allows us to calculate the friction force employing the standard radial return algorithm. First a trial state is calculated:

$$\mathbf{F}_s^{\mathrm{trial}} = \mathbf{F}_s^{\mathrm{old}} - K_s \mathbf{v}_{rT} \Delta t\,, \tag{3.14}$$

and then the slip condition is checked:

$$\phi^{\mathrm{trial}} = \|\mathbf{F}_s^{\mathrm{trial}}\| - \mu|F_n|\,. \tag{3.15}$$

If $\phi^{\mathrm{trial}} \leq 0$, we have stick contact and the friction force is assigned the trial value:

$$\mathbf{F}_s^{\mathrm{new}} = \mathbf{F}_s^{\mathrm{trial}}\,, \tag{3.16}$$

otherwise (slip contact) a return mapping is performed giving:

$$\mathbf{F}_s^{\mathrm{new}} = \mu|F_n| \frac{\mathbf{F}_s^{\mathrm{trial}}}{\|\mathbf{F}_s^{\mathrm{trial}}\|}\,. \tag{3.17}$$

3.4 Integration of Motion

3.4.1 Basic equations

The translational and rotational motion of rigid spherical or cylindrical particles is described by means of the standard equations of rigid body dynamics. For the *i*-th particle we have (Figure 3.8)

$$m_i \ddot{\mathbf{u}}_i = \mathbf{F}_i \,, \tag{3.18}$$

$$I_i \dot{\boldsymbol{\omega}}_i = \mathbf{T}_i \,, \tag{3.19}$$

where $\mathbf{u}$ is the element centroid displacement in a fixed (inertial) coordinate frame $\mathbf{X}$, $\boldsymbol{\omega}$ – the angular velocity, m – the element mass, I – the moment of inertia, $\mathbf{F}_i$ – the resultant force, and $\mathbf{T}_i$ – the resultant moment about the central axes.

Vectors $\mathbf{F}_i$ and $\mathbf{T}_i$ are sums of: (i) all forces and moments applied to the *i*-th particle due to external loads, $\mathbf{F}_i^{ext}$ and $\mathbf{T}_i^{ext}$, respectively, (ii) contact interactions with neighbouring spheres $\mathbf{F}_i^c$ (Figure 3.4), $j = 1, \cdots, n_i^c$, where n_i^c is the number of elements being in contact with the *i*-th discrete element, (iii) forces and moments resulting from external damping, $\mathbf{F}_i^{damp}$ and $\mathbf{T}_i^{damp}$, respectively, which can be written as:

$$\mathbf{F}_i = \mathbf{F}_i^{ext} + \sum_{j=1}^{n_i^c} \mathbf{F}_i^c + \mathbf{F}_i^{damp} \tag{3.20}$$

$$\mathbf{T}_i = \mathbf{T}_i^{ext} + \sum_{j=1}^{n_i^c} (\mathbf{r}_i^c \times \mathbf{F}_i^c + \mathbf{q}_i^c) + \mathbf{T}_i^{damp} \tag{3.21}$$

where $\mathbf{r}_i^c$ is the vector connecting the centre of mass of the $i - th$ particle with the contact point c (Figure 3.4) and $\mathbf{q}_i^c$ are torques due to rolling or torsion (not related with the tangential forces).

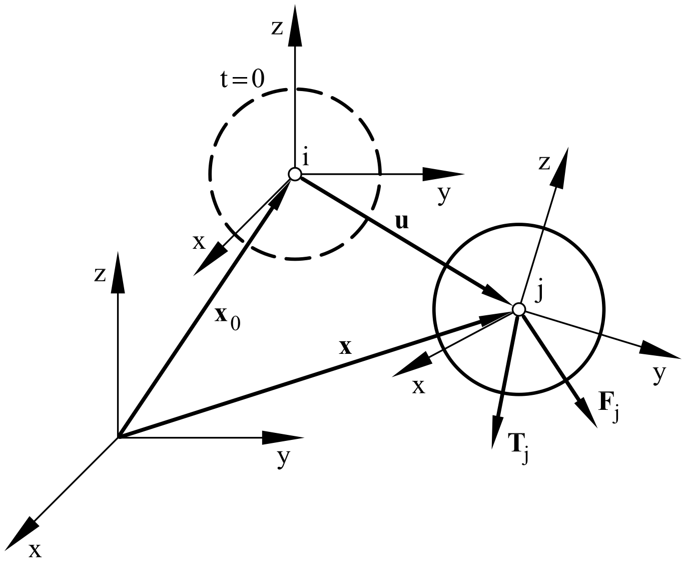


Figure 3.8: Motion of a rigid particle

Note that the form of the rotational equation (3.19) is only valid for spheres and cylinders (in 2D). It is simplified with respect to a general form for an arbitrary rigid body with the rotational inertial properties represented by a second order tensor. In the general case it is more convenient to describe the rotational motion with respect to a co-rotational frame $\mathbf{x}$ which is embedded at each element since in this frame the tensor of inertia is constant.

3.4.2 Integration of the Equations of Motion

Equations (3.18) and (3.19) are integrated in time using a simple central difference scheme [21]. The time integration operator for the translational motion at the n-th time step is as follows:

$$\ddot{\mathbf{u}}_i^n = \frac{\mathbf{F}_i^n}{m_i}\,, \tag{3.22}$$

$$\dot{\mathbf{u}}_i^{n+1/2} = \dot{\mathbf{u}}_i^{n-1/2} + \ddot{\mathbf{u}}_i^n \Delta t \tag{3.23}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n + \dot{\mathbf{u}}_i^{n+1/2} \Delta t \tag{3.24}$$

The first two steps in the integration scheme for the rotational motion are identical to those given by Eqs.(3.22) and (3.23):

$$\dot{\boldsymbol{\omega}}_i^n = \frac{\mathbf{T}_i^n}{I_i}\,, \tag{3.25}$$

$$\boldsymbol{\omega}_i^{n+1/2} = \boldsymbol{\omega}_i^{n-1/2} + \dot{\boldsymbol{\omega}}_i^n \Delta t \tag{3.26}$$

The vector of incremental rotation $\Delta\boldsymbol{\theta} = [\Delta\theta_x, \Delta\theta_y, \Delta\theta_z]^{\mathrm{T}}$ is calculated as:

$$\Delta\boldsymbol{\theta}_i = \boldsymbol{\omega}_i^{n+1/2} \Delta t \tag{3.27}$$

Knowledge of the incremental rotation suffices to update the tangential contact forces. It is also possible to track the rotational position of particles, if necessary. Then the rotation matrices between the moving frames embedded in the particles and the fixed global frame must be updated incrementally using an adequate multiplicative scheme [15, 21].

Explicit integration in time yields high computational efficiency and it enables the solution of large models. The disadvantage of the explicit integration scheme is its conditional numerical stability imposing the limitation on the time step Δt [34], i.e.:

$$\Delta t \le \Delta t_{\mathrm{cr}} \tag{3.28}$$

where Δt_{cr} is a critical time step determined by the highest natural frequency of the system ω_{max} as

$$\Delta t_{\mathrm{cr}} = \frac{2}{\omega_{\mathrm{max}}} \tag{3.29}$$

If damping exists, the critical time increment is given by:

$$\Delta t_{\mathrm{cr}} = \frac{2}{\omega_{\mathrm{max}}} \left(\sqrt{1+\xi^2} - \xi \right) \tag{3.30}$$

where ξ is the fraction of the critical damping corresponding to the highest frequency ω_{max}. Exact calculation of the highest frequency ω_{max} requires the solution of the eigenvalue problem defined for the whole system of connected rigid particles. In an approximate solution procedure, an eigenvalue problem can be defined separately for every rigid particle using the linearized equations of motion. The maximum frequency is estimated as the maximum of natural frequencies of mass–spring systems defined for all the particles with one translational and one rotational degree of freedom.

$$\omega_{max} = \max_i \omega_i \tag{3.31}$$

and the natural frequency for each mass-spring system (contact) is defined as

$$\omega_i = \sqrt{\frac{k}{m_i}} \tag{3.32}$$

with k the spring stiffness and m_i the mass of particle i. Now, for the case with no damping, it is possible to rewrite the critical time step as:

$$\Delta t_{cr} = \min\ 2\sqrt{\frac{m_i}{k}} \tag{3.33}$$

The effective time step is considered as a fraction of the critical time step:

$$\Delta t = \beta \Delta tcr \tag{3.34}$$

with:

$$0 \leq \beta \leq 1 \tag{3.35}$$

The value of β has been studied by different authors. A good review can be found in [22], where the author recommend values close to $\beta = 0.17$ for 3D simulation, and $\beta = 0.3$ for the 2D case.

Chapter 4

Discrete Element Method for Continuum Media - Advanced Issues and Developments on the method

4.1 Introduction to the DEM for continuum simulation

Is the DEM a discretization method? Can we model the macro behaviour introducing micro parameters in the contacts? and, why use the Discrete Element Method prior to other FEM-based methods for these problems? These are key questions that need to be answered if we want to use DEM for continuum media analysis.

The main aspiration is to have a general computational method for unified modelling of the mechanical behaviour of solid and particulate materials, including the transition from solid phase to particulate phase. Within the DEM, the individual particles are modelled as stiff bodies which interact via contact forces. This simplification has the advantage of the complicated microscopic behaviour being represented by a simple system of linear equations based on a relatively small number of parameters. This is what makes DEM a really atractive method.

What is agreed by the comunity is that the Discrete Element Method is a great numerical method to simulate the discontinuous media as a system of independent particles in dynamic motion. However, when dealing with the continuum, nowadays, results are not completely satisfactory even though a lot of research has been done. There have been, indeed, a vast number of different approaches for this question: How should the contact models be characterized (micro scale parameters) in order to obtain the macro scale continuum behaviour? The challenge in all DEM models is finding an objective and accurate relationship between the DEM parameters and the standard constitutive parameters of a continuum mechanics model, namely the Young modulus E, the Poisson ration ν and the stress and strain tensors, $\boldsymbol{\sigma}$ and $\boldsymbol{\epsilon}$ respectively.

Basically two different approaches can be followed for determining the DEM constitutive parameters, namely the global approach and the local approach. In the global approach uniform global DEM properties are assumed in the whole discrete element assembly. In this case, the values of the global DEM parameters can be found using different procedures; Some authors [10, 12] have used numerical experiments for determining the relationships between DEM and continuum parameters expressed in dimensionless form. This method has been used by the authors in previous works [16]–[21], [26]–[27]. This approach has been followed also by C. Labra [15] in the Dempack software. Other procedures for defining the global DEM parameters are based on the definition of average particle size measures for the whole discrete particle assembly and then relating the global DEM and continuum parameters via laboratory tests.

A second approach, followed in this work, is to assume that the DEM parameters depend on the local properties of the interaction particles, namely their radii and the continuum parameters at each interaction point. Many alternatives for defining the DEM parameters via a "local approach" have been reported by different authors in recent years [5, 6, 9, 23, 30, 31].

Appart from the possibility of defining the contact parameters locally different at each contact or globally as an average value, most of the recognized codes and researchers in the field assume that the DEM is a **phenomenological** mehtod. So it does F.-V. Donzé when applies the DEM to the simulation of triaxial concrete test [31] using the YADE platform. It will be discussed later on with more detail in Section 5.1.3 but basically asumes that even for the linear elasticity the method needs the calibration for every material of some constant parameters afecting the stiffness in the contacts.

In this work we present a procedure for defining the DEM parameters for cement material using the local approach. In the next section 4.4 a description of how the local elastic parameters are defined can be found. Then it is also defined the appropriate local failure criteria at the contact interface using an elasto-damage model for the normal tensile stress and an elasto-plastic model for the normal compressive stress.

4.2 Discretization scale

The DEM, as well as other particle methods, can be a discretization or representation method. It is a key issue to decide what is more appropiate for each analysis.

Normally, when the DEM is applied to discrete media simulation, the representative scale of each particle in the simulation is larger than the actual particle; the technique used in that case is the so-called up-scalling [8]. Discretizing a continuus media implies, as well, deciding which is the scale that each element/particle will represent; that scales can be classified as micro-scale, meso-scale or macro-scale.

Micro-scale: Applications in Molecular Dynamics (MD), consisting on a computer simulation of physical movements of atoms and molecules in the context of multi-body simulation. The atoms and molecules are allowed to interact for a period of time, giving a view of the motion of the atoms where forces between the particles and potential energy are defined by molecular mechanics force fields. In concrete simulations, this would be the case of studying the cristaline microstructure of the cement components (Figure 4.1).

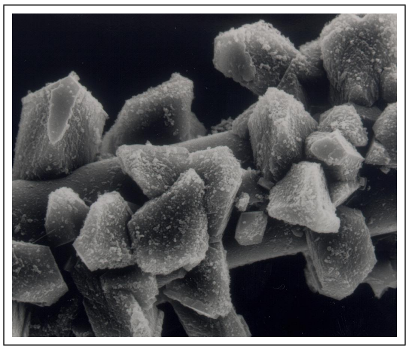


Figure 4.1: Calcium Hydroxile, microscope image. *Source: Google images.*

Meso-scale: This normally consist in a larger scale which can represent larger structures and patterns with different behaviour present in the media that is beeing analysed. In concrete simulations, the meso scale analysis would consist in simulaing the paste as a homogeneous material, the aggregates as particles and the voids (Figure 4.2).

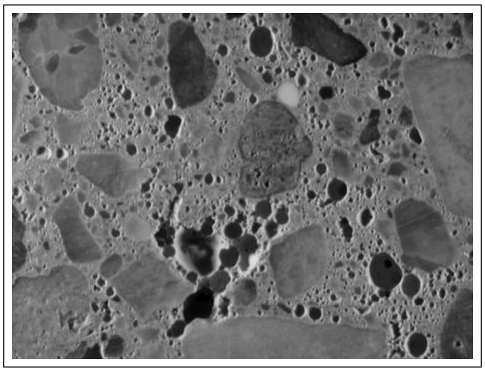


Figure 4.2: Detail of paste, agregate and voids in concrete. *Source: Google images.*

Macro-scale: Large scales where the smallest element discretitzation is defined by means of *Representative Elementay Volumes* or unit cells, which is the smallest volume over which a measurement can be made that will yield a value representative of the whole. The property of interest can include mechanical properties such as elastic moduli, hydrogeological properties, electromagnetic properties, thermal properties, and other averaged quantities that are used to describe physical systems. This the case of the simulations performed in the context of this work, where each element (or particle) represents the average homogeneous behaviour characterized by the macroscopical properties.

In the present work the DEM is used as a discretization method where each sphere represents the mixure of paste, agregate and voids in a certain macroscopical volume surrounding the particle; the mechanical behaviour of it is reached by means of the characteritzation of the contacts with its neighbouring particles.

4.3 Partition of space

In any DEM method, in order to be consistent with the classical continuum theory, the partition of space, which will be normally done by means of spheric elements, has to fill the complete volume of the model without adding extra volume or including unwanted voids.

4.3.1 Spheres packing

The meshes obtained when meshing with spheres regular geometrical 3D objects such as cubes, prisms or cilinders yield a lot of empty space in it. This means that, to complete the continuum described by the model, every particle need to represent a larger volume than its own.

The maximum density sphere packing that can be obtained for a regular mesh come for a distribution in the following fashion:

Start with a layer of spheres in a hexagonal lattice, then put the next layer of spheres in the lowest points you can find above the first layer, and so on − this is just the way you see oranges stacked in a shop. At each step there are two choices of where to put the next layer, so this natural method of stacking the spheres creates an uncountably infinite number of equally dense packings, the best known of which are called cubic close packing and hexagonal close packing. Each of these arrangements has an average density of:

$$\frac{\pi}{3\sqrt{2}} \approx 0.740480189 \qquad (4.1)$$

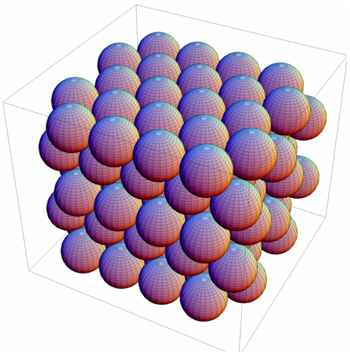


Figure 4.3: So-called cubic packing for spheres. *Source: Wolfram Alpha.*

The *Kepler conjecture* states that this is the best that can be done—no other arrangement of identical spheres has a higher average density.

The optimal (minimum) porosity obtained with particles of the same radius is then in the order of 25%, for higher compacity obviously diferent sizes must be used. However, a considerably large dispersion (small spheres in contact with large ones) yields obvious counterparts in many aspects such as the search algorithm, the contact parameters characterization and the critical times for the explicit schemes.

4.3.2 Mesh generator

The discrete meshes that are used in KDEM are generated using the sphere mesher of *GiD*. It has to be pointed out that since the mesher has some imperfections, gaps, inclusions and some anormal big or small particles will be obtained. This has to be taken into account in the next sections to properly define their properties in the model.

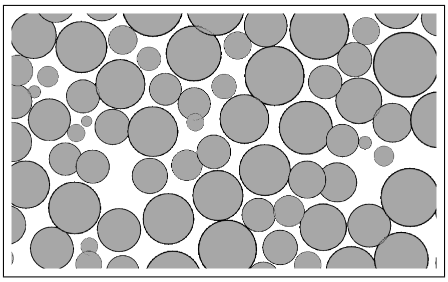


Figure 4.4: Cut view of a 3D sphere mesh (with ”imperfections”) generated by GiD. *Source: GiD sphere mesher.*

on Section 4.6.2, the explanation of how to deal with these imperfections and how to complete the volume modeled can be found.

4.3.3 Interaction range

The overall behaviour of a material can be reproduced by associating a simple constitutive law to each contact interface. The interaction between spherical elements i and j with radius r_i and r_j, respectively is defined within an interaction range. This *range does not necessarily imply that two particles are in contact* and it also allows for some overlapping between particles. Then two spherical particles will interact if:

$$1 - \beta \leq \frac{d_{ij}}{r_i + r_j} \leq 1 + \beta \tag{4.2}$$

where d_{ij} is the distance between the centroids of particles i and j and β is the interaction range parameter (Figure 4.5). The software KDEM will make use of this so-called *delta option* for handling with the prescence of gaps and initial passive indentations. Simply, the equilibirum position for he contacts is set with their initial configuration (which is not always tangential contact of spheres). Once the search is done, the initial distance of each pair is stored as the passive initial contacting state (equilibrium); this works both inwards for inclusions or outwards for gaps. For more information, refer to [28]. In this work the values taken are normally $\beta = 0.15$.

From Eq.(4.2) we deduce

$$d_{ij} = (1 \pm \beta)(r_i + r_j) \tag{4.3}$$

where the plus and minus sign account for the gap and overlapping between two particles, respectively.

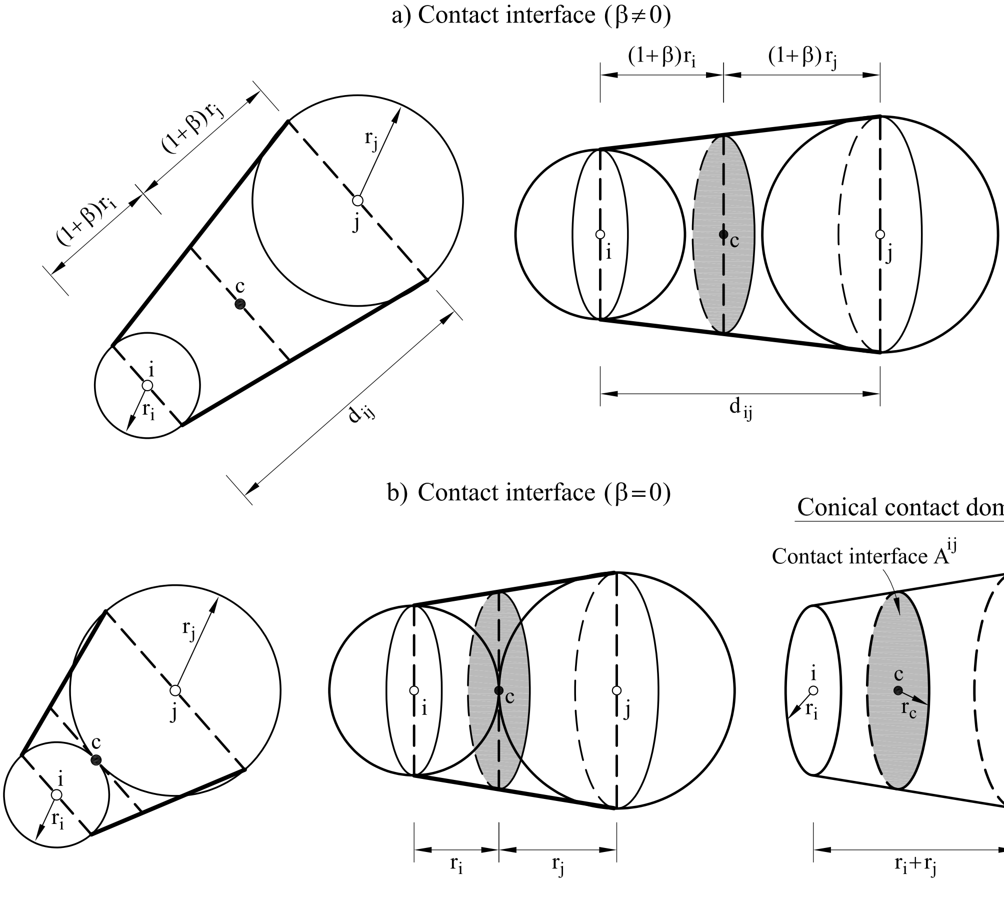


Figure 4.5: Definition of contact interface. (a) $\beta \neq 0$. (b) $\beta = 0$

We will assume that particles i and j are in contact at a point C located at a distance $(1 + \beta)r_i$ or $(1+\beta)r_j$ from the centers of particles i and j, respectively (Figure 4.5). We define the *interaction domain* at the contact point C as a *truncated cone* with circular faces of radius r_i and r_j. The circular section at point C of radius r_c is the *contact interface* between particles i and j.

In the *KDEM-model* these sections are modified in order to avoid overlapping with other conical domains defined by the neighbouring particles. This novel method is developed in section 4.6.2.

4.4 DEM Constitutive Models

A challenge in the analysis of highly non linear material, such as cement, rock or concrete, with the DEM is the characterization of the non linear relationship between forces and displacements at the contact interface accounting for frictional effects, damage and plasticity, and the definition of the limit strengths in the normal and shear directions at these interfaces.

Standard constitutive models in the DEM are characterized by the following parameters:

- Normal and shear stiffness parameters K_n and K_s.
- Normal and shear strength parameters $\mathcal{F}_n$ and $\mathcal{F}_s$.
- Coulomb internal friction angle and coefficient ϕ_1 and μ_1 .
- Coulomb dynamic friction angle and coefficient ϕ_2 and μ_2 .
- Local damping coefficient c_n at the contact interface.
- Global damping coefficient for translational motion, α^t.
- Global damping coefficient for rotational motion, α^n.

Figure 4.6 shows an scheme of some DEM parameters for a 2D model.

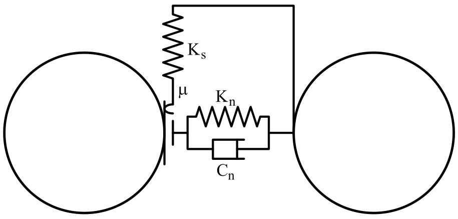


Figure 4.6: Model of the contact interface in the DEM

In this work two different local DEM models for analysis of concrete material are presented which have been implemented in the Kratos DEM-Application.

On one hand, a model developed by CIMNE researchers E. Oñate, F. Zárate et al. reported in [20] that has been implemented in Dempack code and has been used for real engineering projects, hereafter called *Dempack-model*. That *Dempack-model* has been validated using the Dempack software in its application to the analysis of concrete samples for a uniaxial compressive strength (UCS) test, different triaxial compressive strength tests and a brazilian tensile strength (BTS) test. The results obtained with that model compare well with experimental data for the tests provided by the Technical University of Catalonia (UPC) for the concrete samples [29]. In this work these tests are described and analysed, using the Kratos DEM-Application, characterizing the contact parameters with the *Dempack-model*. The tests have been reproduced and the results of the Kratos DEM-Application have been compared against the results obtained by Dempack as well as the experimental results as a validation of the developed code.

On the other hand, a novel methodology presented by the author of the thesis has been developed for the linear elasticity problem with the idea of getting rid of some mesh dependence and improve in some aspects the Dempack model. This novel method is presented in section 4.6 and will be the method used in this thesis for the numerical analysis done in the chapther 5.

4.5 DEM Elastic Constituitive Parameters - *Dempack-model*

Let us assume that an individual particle is connected to the adjacent particles by appropriate relationships at the contact interfaces between the particle and the adjacent ones. These relationships define either a perfectly bond or a frictional sliding situation at the interface.

The following asumes that particles i and j are in contact at a point C located at a distance $(1+\beta)r_i$ or $(1+\beta)r_j$ from the centers of particles i and j respectively (Figure 4.5). The interaction domain at the contact point C is defined then as a truncated cone with circular faces of radius r_i and r_j. The circular section at point C of radius r_c is the contact interface between particles i and j.

4.5.1 Normal contact force

The normal force F_n at the contact interface between particles i and j is given by

$$F_n^{ij} = \sigma_n \bar{A}^{ij} \tag{4.4}$$

where σ_n is the normal stress ($\sigma_n = n_i \sigma_{ij} n_j$) at the contact interface and A^{ij} is the *effective area* at the interface computed as:

$$\bar{A}^{ij} = \gamma_i A^{ij} \quad \text{with} \quad A^{ij} = \pi (r_c)^2 \tag{4.5}$$

A simple algebra gives:

$$r_c = \frac{2r_i r_j}{(r_i + r_j)} \quad \text{and} \quad A^{ij} = \frac{4\pi (r_i r_j)^2}{(r_i + r_j)^2} \tag{4.6}$$

In Eq.(4.5) γ_i is a parameter that accounts for the fact that the number of contacts and the packaging of spheres are not optimal. Clearly, γ_i is a local parameter for the ith sphere. However, in the developed model by Oñate, Zárate et al. [20], a *global definition* of γ_i is used for the whole collection of particles which can be estimated as:

$$\gamma_i = \gamma = 40 \frac{P}{N_c} \tag{4.7}$$

where N_c and P are the average number of contacts per sphere and the average porosity *for the whole particle assembly*. Eq.(4.7) has been deduced by defining the optimal values for the number of contacts per sphere and the global porosity equal to 10 and 25%, respectively. *Note*: see the presented perfect packing of spheres in section 4.3.1.[1]

The normal stress σ_n is related to the normal strain between the spheres, ε_n, by a visco-elastic law as:

$$\sigma_n = E\varepsilon_n + c\dot{\varepsilon}_n \tag{4.8}$$

The normal strain and the normal strain rate are defined as:

$$\varepsilon_n = \frac{u_n}{d_{ij}} \quad , \quad \dot{\varepsilon}_n = \frac{\dot{u}_n}{d_{ij}} \tag{4.9}$$

where d_{ij} is given by Eq.(4.3).

[1] After some analysis done with KDEM-Application and also with Dempack it turned out that this correction of areas is not accurate enough and doesn't solve the mesh-dependence problem. In the simulations performed in chapter 7 this is a input parameter that has been calibrated for every test and every mesh. The new method presented in section 4.6 improves the *Dempack-model* in terms of avoiding mesh-dependence for the normal striffness characterization.

Substituting Eq.(4.9) into (4.8) gives:

$$\sigma_n = \frac{1}{d_{ij}}[Eu_n + c\dot{u}_n] \tag{4.10}$$

In Eqs.(4.8) and (4.10) E is the Young modulus, c is a damping coefficient and u_n and $\dot{u}_n$ are the relative normal (relative) displacement and the normal (relative) velocity at the contact point defined as

$$u_n = (\mathbf{x}_j - \mathbf{x}_i) \cdot \mathbf{n}^{ij} - r_i - r_j \quad , \quad \dot{u}_n = (\dot{\mathbf{u}}_j - \dot{\mathbf{u}}_i) \cdot \mathbf{n}^{ij} \tag{4.11}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the position vectors of the centroids of particles i and j after deformation.

From Eqs.(4.4) and (4.10) we deduce the normal force-normal relative displacement/velocity relationship at the interface between particles i and j as:

$$F_n^{ij} = \frac{\bar{A}^{ij}}{d_{ij}}(Eu_n + c\dot{u}_n) = K_n u_n + C_n \dot{u}_n \tag{4.12}$$

Substituting Eqs.(4.5), (4.6) and (3.8) into (4.12) we find the expression of the stiffness and viscous (damping) coefficients at the contact interface as:

$$\boxed{K_n = \frac{4\pi r_i^2 r_j^2 \gamma_i}{(r_i + r_j)^2 d_{ij}} E} \quad \boxed{C_n = \frac{2\pi r_i r_j \xi \gamma_i}{(r_i + r_j) d_{ij}} \sqrt{m_{ij} K_n}} \tag{4.13}$$

Eq.(4.12) is assumed to hold in the elastic regime for both the normal tensile force F_{n_t} and the normal compressive force F_{n_c}.

4.5.2 Shear forces

The shear force $\mathbf{F}_s^{ij}$ along the shear direction $\mathbf{u}_s^{ij}$ (Figure 4.7) can be written as:

$$\mathbf{F}_s^{ij} = F_{s_1}\dot{\mathbf{u}}_{s_1} + F_{s_2}\dot{\mathbf{u}}_{s_2} \tag{4.14}$$

where F_{s_1} and F_{s_2} are the shear force components along the shear direction $\dot{\mathbf{u}}_{s_1}$ and $\dot{\mathbf{u}}_{s_2}$, respectively.

The shear force modulus F_s is obtained by:

$$F_s^{ij} = |\mathbf{F}_s^{ij}| = (F_{s_1}^2 + F_{s_2}^2)^{1/2} \tag{4.15}$$

A similar approach can be followed for obtaining the relationship between the shear forces and the shear displacements at the contact point.

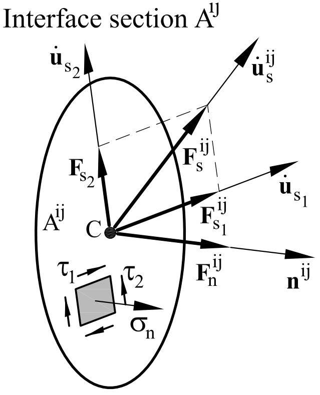


Figure 4.7: Forces and stresses acting on an interface section A^{ij}

The shear forces in the s_1 and s_2 directions (Figure 4.7) are given by

$$F_{s_1} = \tau_1 \bar{A}^{ij} \quad , \quad F_{s_2} = \tau_2 \bar{A}^{ij} \tag{4.16}$$

The shear stresses τ_1 and τ_2 are linearly related to the shear strains γ_1 and γ_2 at the interface by

$$\tau_1 = G\gamma_1 \quad , \quad \tau_2 = G\gamma_2 \tag{4.17}$$

where G is the shear modulus.

A simple definition of the shear strains is:

$$\gamma_1 = \frac{u_{s_1}}{d_{ij}} \quad , \quad \gamma_2 = \frac{u_{s_2}}{d_{ij}} \tag{4.18}$$

where u_{s_1} and u_{s_2} are the components of the relative shear displacement vector at the contact point given by:

$$\mathbf{u}_s^{ij} = [u_{s_1}, u_{s_2}]^T = \mathbf{u}^{ij} - (\mathbf{u}^{ij} \cdot \mathbf{n}^{ij})\mathbf{n}^{ij} \tag{4.19a}$$

with

$$\mathbf{u}^{ij} = (\mathbf{u}_i + \mathbf{w}_i \times \mathbf{r}_{c_i}) - (\mathbf{u}_j + \mathbf{w}_j \times \mathbf{r}_{c_j}) \tag{4.19b}$$

where $\mathbf{r}_{c_i}$ and $\mathbf{r}_{c_j}$ are the vectors connecting the particle centers with the contact point.

Substituting Eqs.(4.17) and (4.18) into (4.16) we find:

$$F_{s_1} = K_{s_1} u_{s_1} \qquad \text{and} \qquad F_{s_2} = K_{s_2} u_{s_2} \tag{4.20}$$

with:

$$\boxed{K_{s_1} = K_{s_2} = K_s = \frac{K_n}{2(1+\nu)}} \tag{4.21}$$

where ν is the Poisson's ratio.

From Eqs.(4.20) we can obtain a relationship between the modulus of the shear force and the modulus of the shear displacement vector as:

$$\mathbf{F}_s = K_s u_s \tag{4.22}$$

with:

$$\mathbf{F}_s = |\mathbf{F}_s{}^{ij}| = \left[(F_{s_1})^2 + (F_{s_2})^2\right]^{1/2} \quad , \quad u_s = |\mathbf{u}_s^{ij}| = \left[(u_{s_1})^2 + (u_{s_2})^2\right]^{1/2} \tag{4.23}$$

Note that the sign of F_{s_k} $(k = 1, 2)$ in Eqs.(38) depends on the sign of the velocity component u_{s_k}, while in Eq.(40) only the modulus of vectors $\mathbf{F}_s^{ij}$ and $\mathbf{u}_s^{ij}$ are involved.

For convenience the upper indices i, j are omitted hereonwards in the definition of the normal and shear forces $\mathbf{F}_n^{ij}$ and $\mathbf{F}_s^{ij}$ at a contact interface.

4.5.3 Global background damping force

A quasi-static state of equilibrium for the assembly of particles can be achieved by application of an adequate global damping to all particles. This damping adds to the local one introduced at the contact interface (Section 6.1). The following non-viscous type global damping has been considered:

$$\mathbf{F}_i^{\text{damp}} = -\alpha^t \left| \mathbf{F}_i^{\text{ext}} + \sum_{c=1}^{n_i^c} \mathbf{F}_i^c \right| \frac{\dot{\mathbf{u}}_i}{|\dot{\mathbf{u}}_i|} \tag{4.24}$$

$$\mathbf{T}_i^{\text{damp}} = -\alpha^r |\mathbf{T}_i| \frac{\boldsymbol{\omega}_i}{|\boldsymbol{\omega}_i|} \tag{4.25}$$

This damping reduces the total unbalanced forces resulting in every partilce. The definition of the translational and rotational damping parameters α^t and α^r is a topic of research. A practical alternative is to define α^t and α^r as a fraction of the stiffness parameters K_n and K_s, respectively. In this work the value taken for the concrete test is $\alpha^t = \alpha^r = 0.2$. Alternative a viscous type damping can be used as described in [15, 21].

4.6 Novel DEM Constitutive Model

Following the same idea introduced in section 4.3, the area that each particle shares with a neighbour has to be calculated in a consistent way. We improve the classical area calculation from the mean radius (or equivalent radius) showed in section 4.3.3 by means of an automatic calibration parameter that corrects the overlappings obtained in the classical method.

4.6.1 Contact parameters derivation

It consists on contact-wise calculate the normal and shear stiffness only with the information of the properties of the two contacting particles. The method proposes to get the elastic characteristic values for the linear spring in the normal direction and for the transversal one from the equivalent axial stiffness and shear stiffness respectively that the correspondent truncated conical volume would yield. The derivation is as follows:

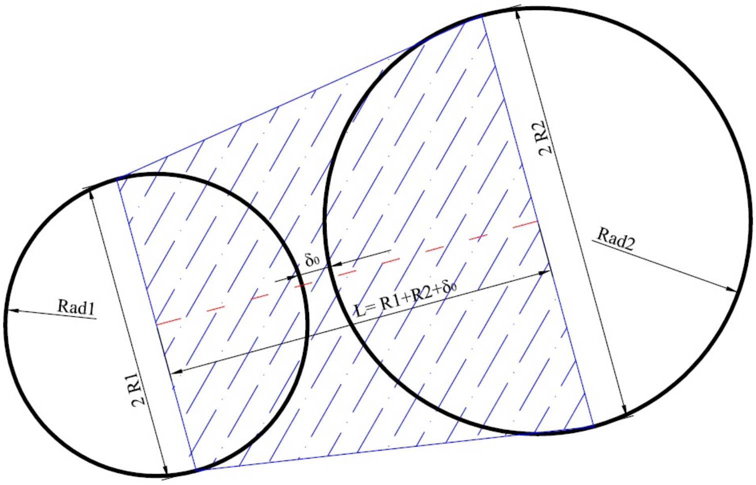


Figure 4.8: Equivalent volume corresponding to by the contact. *Source: M.Santasusana [28].*

$$\delta_x = u_2 - u_1 = \int_0^L \varepsilon dx = \int_0^L \frac{F_x}{EA} dx = \frac{F_x}{E} \int_0^L \frac{dx}{A(x)} \tag{4.26}$$

With a linear variation of the radius:

$$R = R_1(1 + \beta x) \qquad where \qquad \beta = \frac{R_2 - R_1}{R_1 \cdot (R_1 + R_2 + \delta_0)} \tag{4.27}$$

Yields to:

$$F_x = K_n \cdot \delta_x \qquad K_n = \pi E \cdot \frac{R_1 \cdot R_2}{R_1 + R_2 + \delta_0} \tag{4.28}$$

Proceeding similarly, for the shear stress, we get the following:

$$K_s = \pi G \cdot \frac{R_1 \cdot R_2}{R_1 + R_2 + \delta_0} \qquad K_s = K_s \cdot \frac{\mu}{2(1 + \nu)} \tag{4.29}$$

Note that the formulation takes into account linear variation of the stress and stresses along the tronconical differential continuum. The full derivation can be found on Annex [A].

This procedure was an original idea from Professor J. Miquel, and PhD candidates M. Santasusana, M. Celigueta, from CIMNE, and was tested with several simulations using KDEM. It came out that the formulation overestimates the stiffness (similarly to the Dempack model) and it is highly mesh dependent. In the next section 4.6.2 an original idea by the author of this thesis is presented named *Virtual Polihedron Area Correction Method* which is a correction that is able to avoid the mesh dependence and correctly estimates the areas of contact.

4.6.2 Virtual Polihedron Area Correction

Following the ideas presented in Section 4.3, that the area that each particle shares with a neighbour has to be calculated in a consistent way. The idea is to improve the classical area calculation which is based on the mean radius (or equivalent radius) by means of a calibrating parameter that corrects the overlappings obtained in the classical method.

It consists of avoiding the overlaps and give a weight to the stiffness of each contact with a consistent area assignation. Each area of contact comes from the portion of the plane, normal to the line conecting two particles, that is limited by the intersections that this plane forms with the neighbouring planes. These intersections can lead to complex geometries that define irregular polyhedra surrounding every particle; on the other hand, these partial volumes would complete the total volume of the domain.

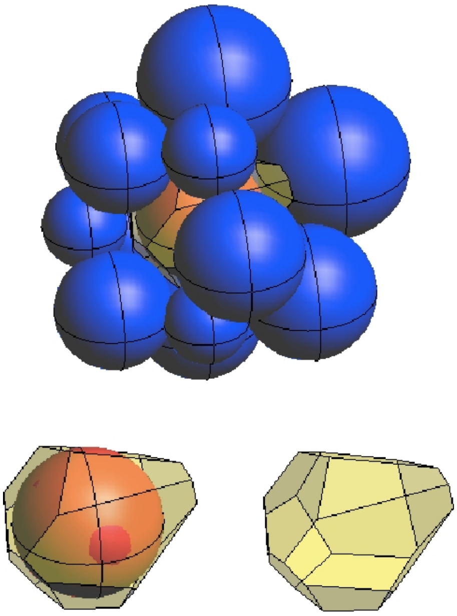


Figure 4.9: Polyhedron associated to a particle. *Source: I.Pouplana [4].*

By doing this, interserctions of n planes should be done varying from 9 to 13 typically the number of neighbours in every particle; in order to find the intersection points, it would be necessary to calculate multiple combinations of 3 planes among them. After that, the convex hull of the intersection points would have to be constructed. These operations would incur in a complex and expensive calculation for every particle. Trying to preserve the simplicity of the method, it turns out that an aproximation of this procedure should be performed instead.

The proposal here is to aproximate the irregular polyhedra of n faces to a virtual regular one of the same number of faces and calculate the total covered surface of this last one. The assumption is that the total area of the resulting polyhedron is similar to the total area of its regular counterpart. It can be shown with examples that an irregular polyhedron surrounding a sphere and a regular one with similar volume and same number of faces have a close value of total covered surface.

However, it is common knowledge that there are only five different convex regular polyhedra, called the *Platonic Solids*: tetrahedron (4 faces), hexahedron (6 faces), octahedron (8 faces), dodecahedron (12 faces) and icosahedron (20 faces).

Since every particle can have any number of neighbours, we need to use the surface area of a fictitious regular convex polyhedron with any number of faces. In order to do so, interpolation over

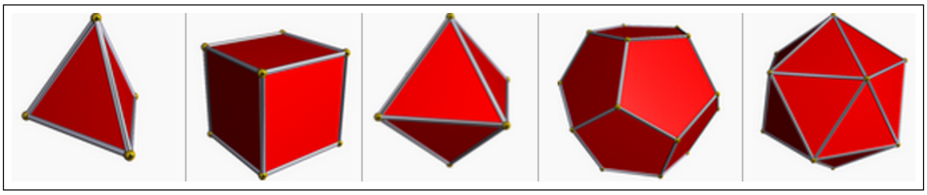


Figure 4.10: Platonic Solids, regular polyhedra. *Source: Wikipedia.*

the already known surface area of the five Platonic solids is performed.

A factor γ_i that modifies the original area assigned to each neighbour A_c (which is based on the equivalent area of each pair of particles) is proposed in equation 4.30:

$$\gamma_i = \frac{A_s, n}{\sum_{c=1}^{n} A_c} \tag{4.30}$$

Where A_s, is the total area of the virtual regular polyhedron of n faces (platonic or interpolated). The denominator of the equation 4.30a is the sum of all the estimated contacting areas that a particle have with each neighbour, A_c:

$$A_c = \pi \cdot R_{eq}^2 = \pi \cdot \left(\frac{2R_1R_2}{R_1 + R_2} \right)^2 \tag{4.31}$$

Note that the presented equivalent area A_c is only correct when $\delta_0 \to 0$ which, for simplicity, can be acceptable, since $\delta_0 << R_i$.

This factor then is applied as a correction to the area for every contact c; for the ith particle, the corrected contact area$\bar{A}_c$ is then:

$$\bar{A}_c = \gamma_i \cdot A_c \tag{4.32}$$

In the Appendix [C] a table containing all the values can be found and also the ratio between the polyhedron surface and the surface of its inscribed sphere. This ratio ranges from $3,308$ in the case of a tetrahedron and tends to 1 when the value of n neighbours increases. The value corresponding to 11 neighbours, which is a usual number of surrounding particles, is still signigicant: 1.410.

From the derived expression 4.28, the following approximation is done:

$$K_n = \pi E \cdot \frac{R_1 \cdot R_2}{R_1 + R_2 + \delta_0} \approx \pi E \cdot \frac{A_c}{R_1 + R_2 + \delta_0} \tag{4.33}$$

Notice that in the derived formula (4.28), instead of A_c, what is present is simply $R_1 \cdot R_2$. However, these two quantities coincide when R_1 and R_2 are similar. Even for a value of $R_2 = 2,5 \cdot R_1$ the relation between those quantities is still of $\beta = 0,\hat{8}$. This factor could be introduced as well as a correction term but, mostly with the γ factor, the error has been significantly reduced and the following can be written[1]:

$$K_n = \gamma_i \cdot \pi E \cdot A_c/(R_1 + R_2 + \delta_0) = \pi E \cdot \bar{A}_c/(R_1 + R_2 + \delta_0) \tag{4.34}$$

An important consideration to be done here is to notice that, given a contacting pair of particles, namely A and B, different area is obtained from each side since each particle has different radius and number of neighbours. In order to avoid this inconsistency and to respect *Newton's Third Law*, the mean value of the the two calculations is the one used in the KDEM code.

Finally, it has to be noticed that special treatment is needed in the ***skin particles***. In this case, the particles aren't surrounded completely by spheres and so, the same assumption of a irregular polyhedron formed by the contacting planes is not valid anymore. In this case, the followed strategy is to assign a proportion of the corresponding interior counterpart to that area of the skin particle pair. Having said that, a clever idea for those skin particles that contact an inner particle is to use for the contact, the area calculated from the inner particle.

2D Case

Just for academic purposes, KDEM includes a 2D module which permits bidimensional simulations using cylinders instead of spheres. The same theory of the calculation of areas applies here, working now with polygons.

In Annex B the corresponding derivation of the correction factor and a table with all the used values in the code can be found.

4.7 Elasto-Damage Model for Tension and Shear Forces

In order to reproduce the behaviour of the fictional cohesive materials like cement, rock or concrete, several important features have been introduced in a DEM model for the continuum by researchers of CIMNE; unidimensional *non-linear elasticity*, *plasticity* and *damage* laws as well as a specific *uncopupled* fracture critera. These models are specially useful for the kind of problems that CIMNE is dealing with, in their current projects using the dempack software, in the field of *concrete test simulation* and *rock mechanics.* These features have been rewritten and tested in KDEM code as well.

In this section, and in the next one (section 4.8), the failure criteria and the constitutives laws implemented are explained. All of these models are compatible with both the so-called *Dempack-model* and the *KDEM-model* which characterize the basic stiffness parameters, K_n and K_s , in a different way.

4.7.1 Normal and shear failure

In this model, cohesive bonds are assumed *to start breaking* when the interface strength is exceeded in the normal direction by the tensile contact force, *or* in the tangential direction by the shear force. The *uncoupled* failure (decohesion) criterion for the normal and tangential directions at the contact interface between spheres i and j is written as:

$$F_{n_t} \geq \mathcal{F}_{n_t} \quad , \quad F_s \geq \mathcal{F}_s \tag{4.35}$$

where $\mathcal{F}_{n_t}$ and $\mathcal{F}_s$ are the interface strengths for pure tension and shear-compression conditions, respectively F_{n_t} is the normal tensile force, respectively and F_s is the modulus of the shear force vector defined in Eq. 4.35.

[1] The same correction applies to the expression derived for the tangential spring in equation A.8.

The interface strengths are defined as:

$$\mathcal{F}_{n_t} = \sigma_t^f \bar{A}^{ij} \quad , \quad \mathcal{F}_s = \tau^f \bar{A}^{ij} + \mu_s |\mathcal{F}_{n_c}| \tag{4.36}$$

where σ_t^f and τ^f are the tension and shear failure stresses, respectively and $\mu_1 = \tan \phi_1$ is the (static) internal friction parameter. These values are assumed to be an intrinsic property of the material. The failure stress σ_t^f is typically determined from a BTS laboratory test. In this work τ^f and ϕ_1 have been taken respectively as the cohesion and the internal friction angle of the Mohr-Coulomb criterion.

The values of τ^f and ϕ_1 can be estimated following the procedure recently proposed by Wang et al. [32] for rocks which gives two analytical expressions for computing τ^f and ϕ_1 as

$$K = \tan^2\left(\frac{\pi}{4} + \frac{\phi_1}{2}\right) \tag{4.37a}$$

$$P = 2\tau^f \tan\left(\frac{\pi}{4} + \frac{\phi_1}{2}\right) \tag{4.37b}$$

where K is the slope of the line that fits the values of the limit axial stress versus the confining pressure for different triaxial tests and P is the value of the limit axial stress (defining the onset of the non linear branch) for the Uniform Compressive Strength (UCS) test. The value of ϕ_1 is obtained from Eq.(4.37a) and then τ^f is obtained from Eq.(4.37b). The derivation of Eqs.(4.37) can be found in [32]. In the DEM in general, and in the *Dempack-model* in particular, however, the practical procedure is to calibrate these parameters phenomenologically to adjust the model to the experimental data as many of the authors do [31].

Figure 4.11 shows the graphic representation of the failure criterium described by Eqs.4.35 and 4.36. Note that this criterium assumes that the tension and shear forces contribute to the failure of the contact interface in a decoupled manner. On the other hand, shear failure under normal compression forces follows a Mohr-Coulomb type constitutive law, with the failure line being a function of the cohesion, the compression force and the internal frictional angle.

Indeed, a coupled failure model in the tension-shear zone can also be used, as shown in the right hand side of Figure 4.11. In this work the uncoupled model has been used.

Figure 4.12 shows the evolution of the normal tension force and the shear force until failure in terms of the relative normal and tangential displacements. The effect of damage in the two constitutive laws is also shown in the figure. The method for introducing the effect of damage is explained in the next section.

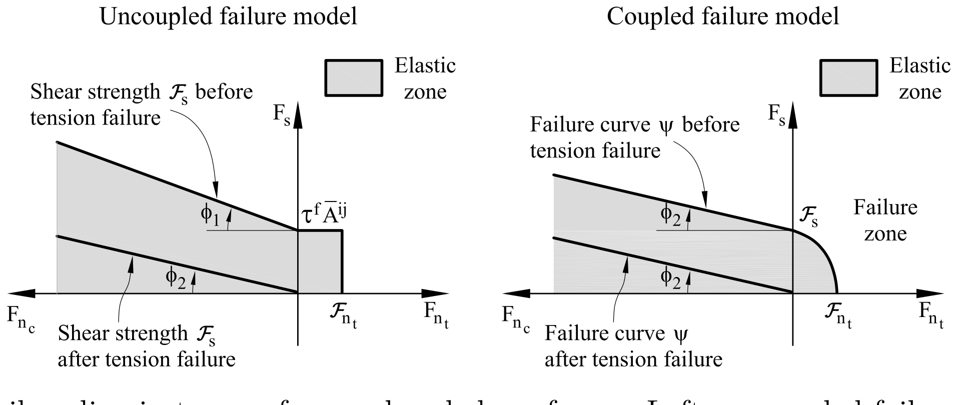


Figure 4.11: Failure line in terms of normal and shear forces. Left: uncoupled failure model. Right: Coupled failure model

4.7.2 Damage evolution law

Elastic damage can be accounted for by assuming a softening behaviour defined by softening moduli H_n and H_t introduced into the force-displacement relationships in the normal (tensile) and tangential directions, respectively (Figure 4.12).

The constitutive relationships for the elasto-damage model are written as

$$\begin{aligned}
&\textbf{Normal (tensile) direction}\\
&\text{For } 0 < d_s \leq 1: \; F_{n_t} = (1-d_n)K_n u_n = K_n^d u_n \qquad \text{with} \qquad K_n^d = (1-d_n)K_n\\
&\text{For } d_n \geq 1: \; F_{n_t} = 0\\
&\textbf{Tangential direction}\\
&\text{For } 0 < d_s \leq 1: \; F_s = (1-d_s)K_s u_s = K_s^d u_s \qquad \text{with} \qquad K_s^d = (1-d_s)K_s\\
&\text{For } d_s > 1: \; F_s = \mu F_{n_c}
\end{aligned} \tag{4.38}$$

where d_n and d_s are scalar damage parameters in the normal (tensile) and shear directions at the contact interface, respectively and K_n^d and K_s^d are damaged elastic stiffness parameters. Damage parameters d_n and d_s are a measure of material damage and loss of mechanical strength at each contact interface. For the undamaged state $d_n = 0$ and $d_s = 0$, while for a damaged state $0 < d_n \leq 1$ and $0 < d_s \leq 1$.

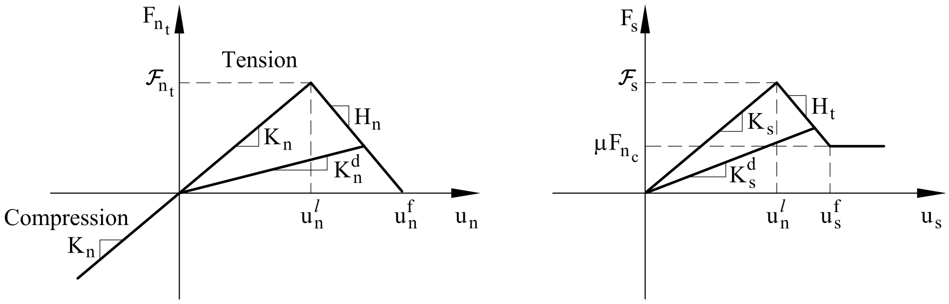


Figure 4.12: Undamaged and damaged elastic module under tension and shear forces

Damage effects are assumed to start when the failure strength conditions 4.35 are satisfied. The evolution of the damage parameters from the value zero to one can be defined in a number of ways using fracture mechanics arguments [19, 21].

In the present work d_n and d_s are defined using a simple linear strain softening law as:

$$d_n = \begin{cases} 0 & \text{for} \quad u_n < \dfrac{\mathcal{F}_{n_t}}{K_n} \\ \dfrac{\left(u_n - \dfrac{\mathcal{F}_{n_t}}{K_n}\right)}{\left(u_n^f - \dfrac{\mathcal{F}_{n_t}}{K_n}\right)} & \text{for} \quad \dfrac{\mathcal{F}_{n_t}}{K_n} \leq u_n < u_n^f \\ 1 & \text{for} \quad u_n \geq u_n^f \end{cases} \tag{4.39}$$

$$d_s = \begin{cases} 0 & \text{for} \quad u_s < \dfrac{\mathcal{F}_s}{K_s} \\ \dfrac{\left(u_s - \dfrac{\mathcal{F}_s}{K_s}\right)}{\left(u_s^f - \dfrac{\mathcal{F}_s}{K_s}\right)} & \text{for} \quad \dfrac{\mathcal{F}_s}{K_s} \leq u_s < u_s^f \\ 1 & \text{for} \quad u_s \geq u_s^f \end{cases} \tag{4.40}$$

where u_n^f and u_s^f are values of the interface displacement increments in the normal and shear directions at failure computed as:

$$u_n^f = d_{ij}\varepsilon_n^f \qquad , \qquad u_s^f = \sqrt{\bar{A}^{ij}}\gamma_s^f \tag{4.41}$$

where ε_n^f and γ_s^f are respectively the failure normal strain and the failure shear strains. These strains are an intrinsic property of each material.

The failure conditions evolve due to damage as follows:

$$F_{n_t} \geq \mathcal{F}_{n_t}^d \qquad , \qquad F_s \geq \mathcal{F}_s^d \tag{4.42}$$

with

$$\begin{aligned} \mathcal{F}_{n_t}^d &= \mathcal{F}_{n_t} - H_n\left(u_n - \frac{\mathcal{F}_{n_t}}{K_n}\right) \\ \mathcal{F}_s^d &= \mathcal{F}_s - H_t\left(u_s - \frac{\mathcal{F}_s}{K_s}\right) \end{aligned} \tag{4.43}$$

where $(\mathcal{F}_{n_t}, \mathcal{F}_{n_t}^d)$ and $(\mathcal{F}_s, \mathcal{F}_s^d)$ are the undamaged and damage interface strengths for pure tension and pure shear conditions, respectively.

4.7.3 Post-failure shear-displacement relationship

Following the failure in the normal or shear directions the shear forces at the contact interface are related to the normal compressive force by Coulomb law as:

$$F_{s_1} = \mu_2|F_{n_c}|\frac{\dot{u}_{s1}}{|\dot{\mathbf{u}}_s^{ij}|} \quad , \quad F_{s_2} = \mu_2|F_{n_c}|\frac{\dot{u}_{s2}}{|\dot{\mathbf{u}}_s^{ij}|} \tag{4.44}$$

or

$$\mathbf{F}_s^{ij} = \mu_2|F_{n_c}|\frac{\dot{\mathbf{u}}_s^{ij}}{|\dot{\mathbf{u}}_s^{ij}|} \quad \text{and} \quad F_s = \mu_2|F_{n_c}| \tag{4.45}$$

where F_n^c is the normal compressive force and μ_2 is the so-called dynamic Coulomb friction coefficient of the material. Note that $\mathbf{F}_s^{ij}$ acts in the direction of the relative shear velocity at the contact point.

Figure 4.13 shows the evolution of the failure lines from the undamaged to the fully damaged state for the uncoupled model of Figure 4.11a.

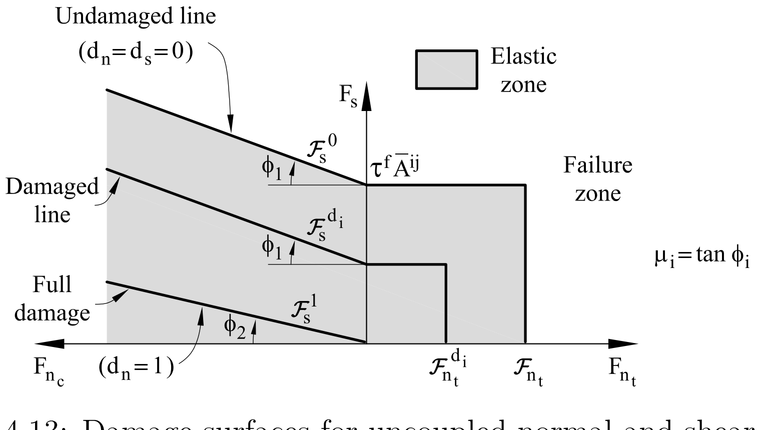


Figure 4.13: Damage surfaces for uncoupled normal and shear failure

4.8 Elasto-Plastic Model for Compressive forces

The compressive stress-strain behaviour in the normal direction for frictional materials such as cement and concrete is typically governed by an initial elastic law followed by a non-linear constitutive equation that varies for each material. The compressive normal stress increases under linear elastic conditions until it reaches a limit value defined by the compressive axial stress σ_{el}. This is defined as the axial stress level where the $\sigma - \varepsilon$ curve starts to deviate from the linear *elastic behaviour*. After this point the material is assumed to yield under *elastic-plastic* conditions.

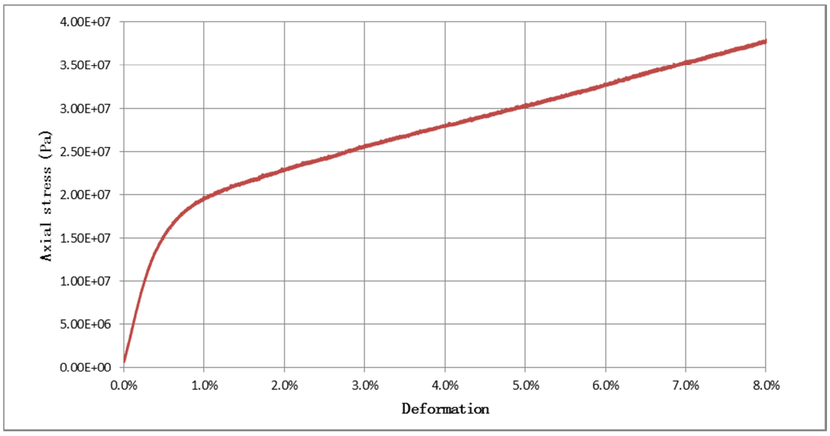


Figure 4.14: Uniaxial strain compaction test in cement sample [14]. Normal total compressive stress-axial strain relationship

Figure 4.14 shows the normal stress-strain relationship for cement [1] material as deduced from the *uniaxial strain compaction* (USC) test in experimental data [14]. The curve shows an initial elastic branch and a (elasto-plastic) hardening branch.

From this observation comes the necessity of coding a law that permits a non-linear behaviour for high values of stresses and includes as well plasticity in terms of irrecuperable deformation. The strategy followed by most of the DEM codes for continuum simulations in concrete or cohesive frictional materials is to phenomenollogically identify the parameters that define a generic and simple non linear and plasticity law on the normal contacts. An important reference in this topic is [31], a work done by V.T. Tran, F.-V. Donzé et al. where a general non-linear elasto-plastic model (Figure 4.15) is calibrated for the simulation of concrete under high triaxial loading.

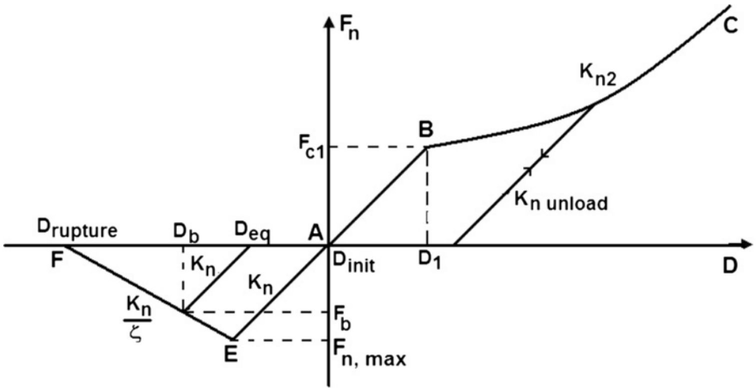


Figure 4.15: Non-linear elastoplastic and damage constituitive law defined by Donzé et Al.

[1] In this specific case of Figure 4.14 another special effect takes part on, the test was performed on a staurated porous cement speciment and there, the effect of the variation of water pressure in the pores should be taken into account; refer to the CIMNE report [20] for more information about how to deal with saturated cement samples.

In the work presented here a simple model is introduced where the *elasto-plastic relationships* in the normal compressive direction are defined as:

Loading path

$$dF_{n_c} = K_{T_n} du_n \tag{4.46a}$$

Unloading path

$$dF_{n_c} = K_{n_0} du_n \tag{4.46b}$$

In Eqs.(55) dF_{n_c} and du_n are the increment of the normal compressive force and the normal (relative) displacement, K_{n_0} is the initial (elastic) compressive stiffness for a value of $E = E_0$ (Figure 4.16), and K_{T_n} is the tangent compressive stiffness given by

$$K_{T_n} = \frac{E_T}{E_0} K_{n_0} \tag{4.47}$$

where E_T is the slope of the normal effective stress-strain curve in the yield branch (Figure 4.16).

Note that plasticity effects in the normal compressive direction have an indirect effect in the evolution of the tangential forces at the interface, as the interface shear strength is related to the normal compression force by Eq. 4.36.

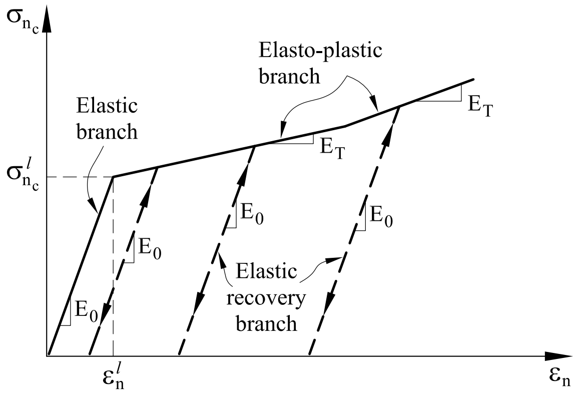


Figure 4.16: Uniaxial normal compressive stress-normal axial strain for elasto-plastic material

The method is generic and permits defining different level of stresses or strains at which the law changes in compression to a different *elastic* slope; the same applies to the strain (or stress) at which the plasticity starts and so the unloading follows the elastic unloading slope instead of going over the non-linear loading path.

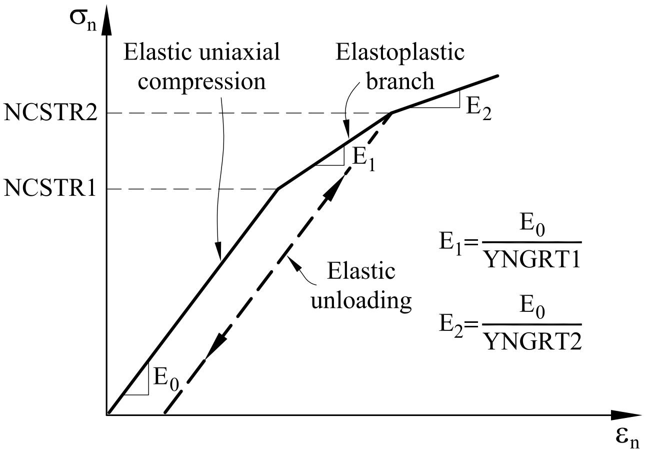


Figure 4.17: Axial stress-strain curve for UCS and Triaxial tests

Chapter 5

Numerical Analysis of the method

The models that are going to be used for the numerical simulations present in this work have been described in the previous chapters. The objective of this chapter is to numerically analyse the DEM applied to continuum simulation by means of doing simple test and checking basic issues of the method such as convergence, mesh dependence, etc. This will be done using the novel method, developed by the author of the tesis, for the characteritzation of the contacts presented in section 4.6.

5.1 Linear Elasticity

5.1.1 Introduction

The main goal of the usage of the DEM in the field of simulating cohesive frictional materials, such as concrete or rock, is to be able to reproduce the typical multifracturing pattern of these materials and capture the strains and stresses which they are subjected to. The aspiration is to have a general computational method for unified modelling of the mechanical behaviour of solid and particulate materials, including the transition from solid phase to particulate phase. A necessary first step for the method to assess is to be able to reproduce the linear elasticity for obvious reasons.

Unfortunatelly, as it has been introduced, there is not a direct unique general way to achieve that, despite the efforts that have been done in the DEM. As it has been previously said in section 4.1, two basic approaches are followed in the characterization of the contact parameters: the global one and the local one. The global uses averaged properties and it is based usually on adimensional parameters. On the other hand, the local approach tries to find a relationship between micro and macro parameters from a local definition. As an introduction to this topic, one example of each approach is referenced here to show the limitations of the DEM in the linear elasticity.

Dimensional Analisis - Global Approach

Huang [12] used dimensionless laws in order to estimate the mechanical behaviour of an assembly of particles governed by the following set of characteristics parameters: K_n , K_s, r, e, ρ, L, V; where r is the average radius, ρ the density, L is the sample length, V is the load velocity and e the porosity of the assembly, as an indirect measure of the particle size distribution and contact density. Later, Yang et al. [33] showed that the porosity e may not be a good index to represent the particle size distribution, or in a most general sense the influence of the particle assembly, and can be replaced by a different parameter Φ representing the geometrical characterization of the assembly like particle size distribution, coordination number, etc.

Since there are seven parameters and three independent dimensions, according to the Buckingham theorem [18] four independent dimensionless parameters govern the elastic response of the assembly.

$$\left\{\frac{K_s}{K_n}, \Phi, \frac{r}{L}, \frac{V}{\sqrt{K_n/\rho}}\right\} \tag{5.1}$$

It is assumed that an enough number of particles is considered, the ratio $(r/L << 1)$ can be ignored. The same can be assumed for the velocity, considering the condition of quasi-static loading $(V/\sqrt{K_n/\rho} << 1)$. The dependence of the elastic constants on the micro-scale parameters can thus be reduced to the following scaling laws:

$$\frac{E \cdot l}{K_n} = \Psi_E \left(\frac{K_s}{K_n}, \Phi\right) \tag{5.2}$$

$$\nu = \Psi_\nu \left(\frac{K_s}{K_n}, \Phi\right) \tag{5.3}$$

Note that according to the scaling laws, the elastic constants are completely determined if the shear and normal stiffnesses K_s and K_n are known for a given size distribution of the particles. This means that the curves hold for a specific assembly of particles, with a given configurations, and can not be scaled to a different one. These is bad news, because that means the method is mesh dependent and needs calibration.

This work was followed by Labra [15] in Dempack. Here some results of what he obtained are shown:

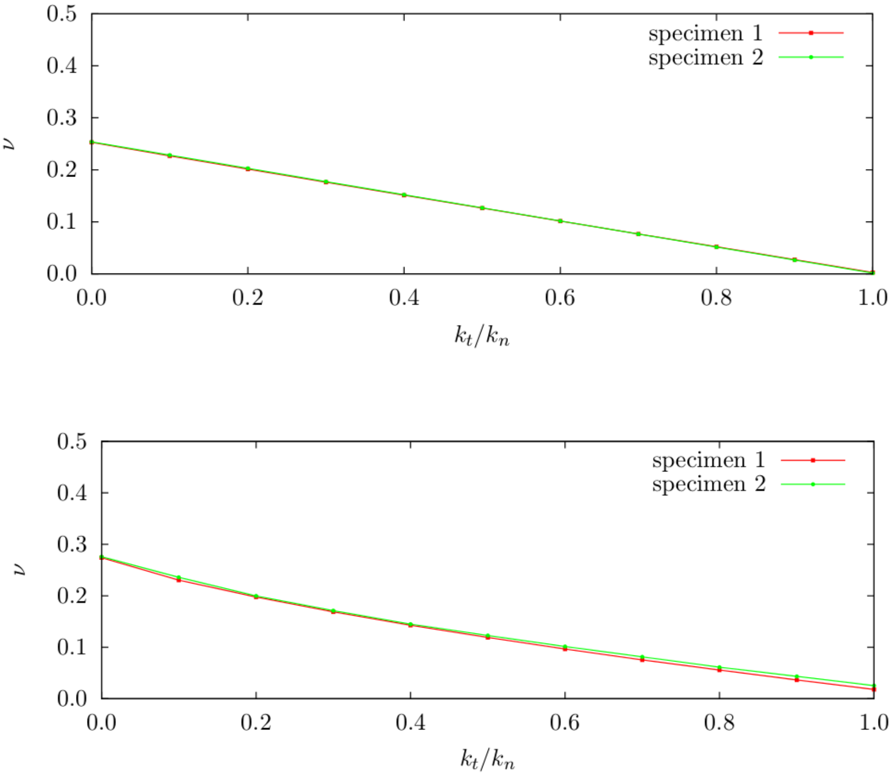


Figure 5.1: Poisson's Ratio for different values of K_s/k_n in 2D and 3D UCS test. *Ref.: C. Labra [15]*

Labra found out that, for a given assembly, the K_s/k_n ratio is key in order to determine the macroscopical poisson ratio of the model. As it can be seen, there exist a limitation on the maximum value of Poisson ratio to the value of 0.25 in 2D case and nearly 0.3 in 3D. Similar results are obtained in the alternative, local approach as it is shown next.

Regular - Local Approach

An interesting study was perform by Tavarez and Plesha [30] with a local definition of the contact parameters in a regular mesh. Their attempt was to theoretically establish the inter-element normal and tangential stiffnesses (and failure parameters, which are not in the present discussion) as functions of element size and commonly accepted material parameters including Young's modulus and Poisson's ratio. This was done by developing a DEM model of a unit cell of material.

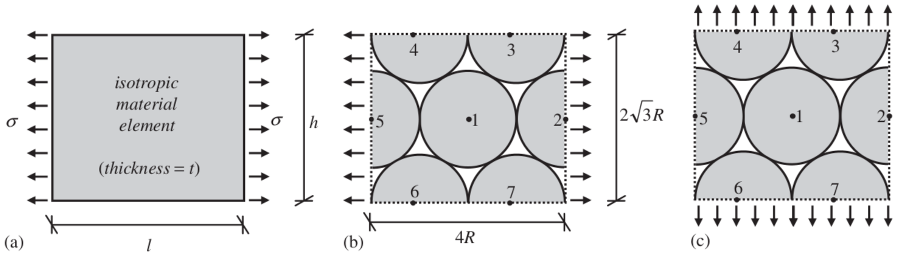


Figure 5.2: Close-packed DEM unit cell for determination of inter-element spring constants. *Ref.: F. A. Tavarez and M. E. Plesha [30]*

Figure 5.2(a) shows an isotropic solid material element (with known elastic modulus and Poisson's ratio) subjected to uniaxial stress. The volume of material is then modelled using the DEM close-packed unit cell with the loading shown in Figure 5.2(b) and the one in 5.2(c) as a check to investigate the possible anisotropy. The unit cell contains seven elements having three degrees of freedom (d.o.f.) per element (two translations and one rotation). Due to the symmetry of loading, all rotations in the unit cell are zero. Therefore, the matrix equation for the 14 translational d.o.f. can be expressed as:

$$[\mathbf{K}] \cdot \mathbf{d} = \mathbf{f} \tag{5.4}$$

Expressing the stiffnes matrix $[\mathbf{K}]$ as a function of K_n and K_s and the geometry and solving for a known case with determined vectors $\mathbf{f}$ and $\mathbf{d}$, the normal and tangetial elastic stiffness for this assembly can be found:

$$K_n = \frac{1}{\sqrt{3}(1-\nu^*)} \cdot E^* t \tag{5.5}$$

$$K_s = \frac{1-3\nu^*}{(1+\nu^*)} \cdot K_n \tag{5.6}$$

Where E^* and ν^* are E and ν for the 2d *plane stress* or $E/(1-\nu^2)$ and $\nu/(1-\nu)$ respectively in *plane strain* case.

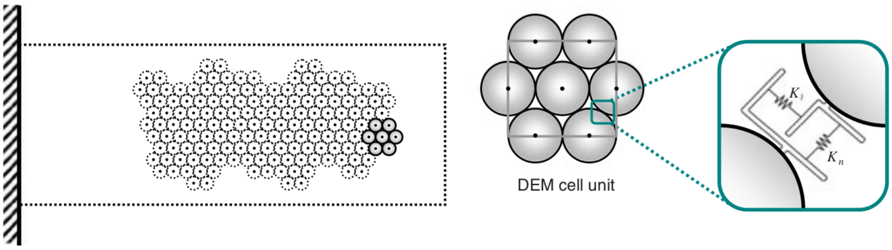


Figure 5.3: DEM discretization and unit cell used in Tavarez and Plesha work. *Ref.: F. A. Tavarez and M. E. Plesha [30]*

The normal and tangential stiffness obtained in numerical simulations by this methodology agree exactly with Equations 5.5 and 5.6. Assuming the shear stiffness must be non-negative, it is interesting to note that these equations limit the maximum value of Poisson's ratio to 1/3 for plane stress and 1/4 for plane strain.

This results are surprisingly similar to the ones that were obtained by the dimensional analysis showed previously. In the global approach, both the Young's modulus and Poisson's ratio should be calibrated for each mesh; in the local approach this is not necessary anymore since an analytical expression has been derived for a given mesh; the disatvantage of the last method is that the multi-fracture path are predefined by the mesh as well. In both methods the poisson is limited to maximum values of 1/4 and are mesh dependent. The method followed in this thesis is analysed next, which, from a local approach and using irregular meshes, aims to obtain linear elasticity for different values of Young's modulus and Poisson's ratio; obviosly the method will face the same limitations explained in this section.

5.1.2 KDEM results of the area calculation. Mesh-Independence.

In this section several examples are done with the purpose of checking whether the method with the correction of areas explained in section 4.6 estimates correctly the area of the geometries.

The following, are 5 different meshes in 2D of a rectangular geometry of 5 cm width and 10 cm height. The meshes go from a regular assembly of discs to a higlhy heterogeneous distribution of the particle radius.

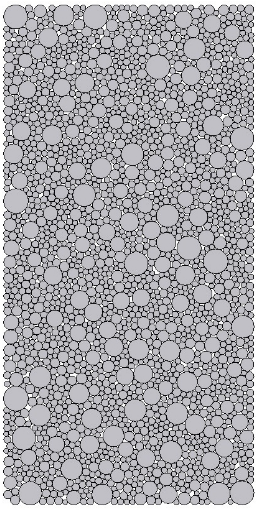


Figure 5.4: Mesh 1

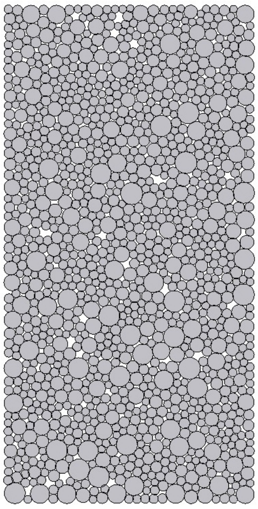


Figure 5.5: Mesh 2

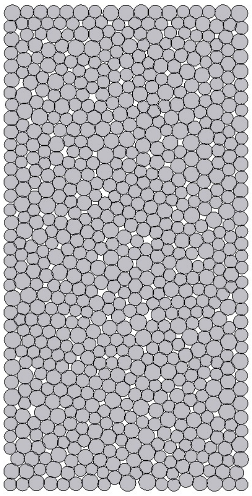


Figure 5.6: Mesh 3

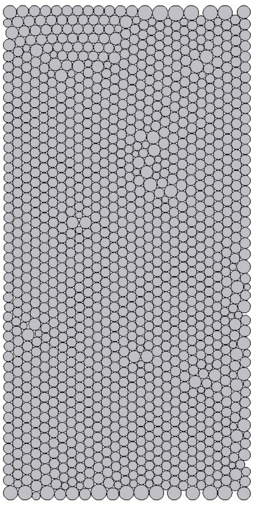


Figure 5.7: Mesh 4

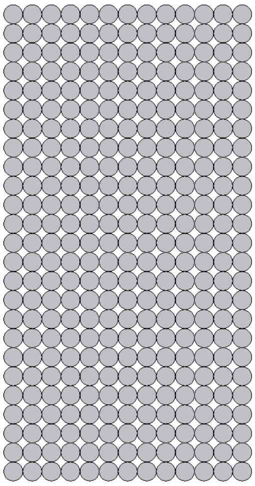


Figure 5.8: Mesh 5

Theoretically, in every contacting pair a characteristic area, $\bar{A}_c$, in 3D, or lenght in 2D, (Section 4.6.2) is assigned so that there are, in average, no overlaps.

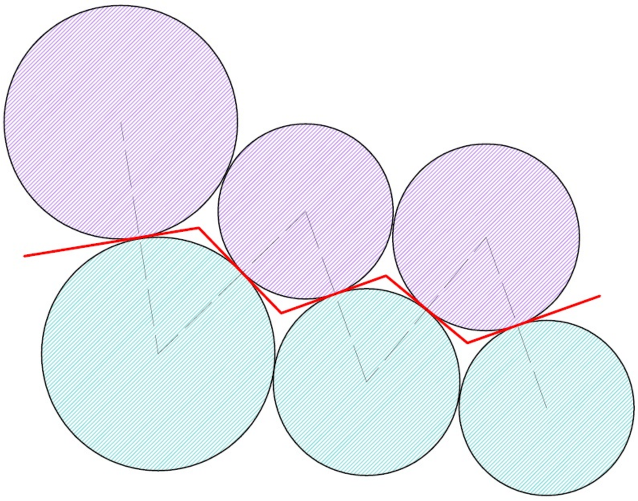


Figure 5.9: Theoretical contact area (lenght in 2D) associated to each contact.

In order to check the goodness of the area assigned to the contacts the following strategy is done: defining several horitzontal strips of particles and calculating the total area projected in vertical direction that every particle of the strip assigns to the contacts with their neighbouring particles out of the strip. The total value is compared against the geometrical width of the model (5cm). Several strips are set in order to average the values obtained.

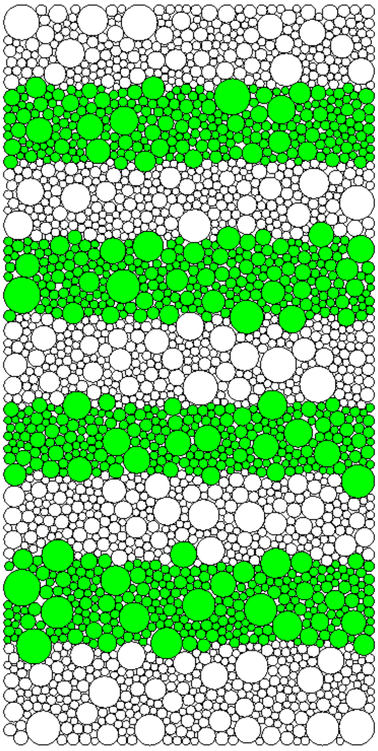


Figure 5.10: Example of strips defined in mesh 1.

The same check has been done in two 3D meshes in order to proof that the method is generic and works fine in 3D too. The geometry meshed is a 15x30 cm cylindrer. Next, the meshes with its monitoring strips is presented.

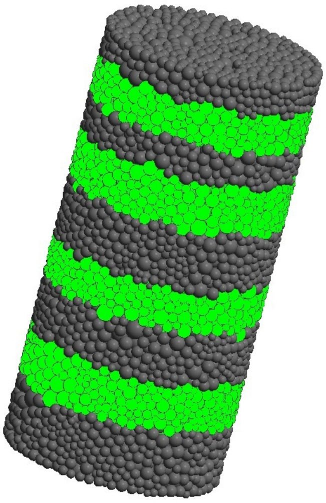


Figure 5.11: Mesh 3D 1 with strips

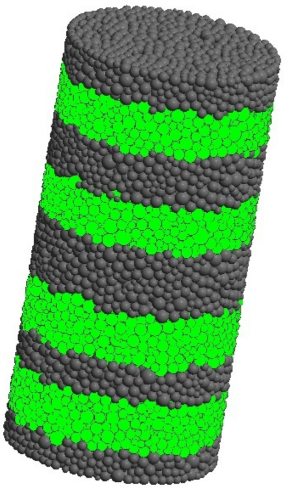


Figure 5.12: Mesh 3D 2 with strips

In the following table 5.1 the results are summarized and the properties of each mesh are presented.

Mesh	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 3D 1	Mesh 3D 2
Num of Elements	2260	1262	725	1343	325	10511	13500
Mean Radius (mm)	$0,72$	$1,02$	$1,42$	$1,02$	$2,0$	$4,21$	$3,86$
Rel. Stand. Deviation[1]**(%)**	$45,65$	$31,16$	$10,24$	$9,23$	$0,00$	$25,25$	$25,83$
Coordination Number[2]	$5,08$	$5,14$	$5,24$	$5,79$	$3,76$	$10,98$	$10,97$
Mesh Porosity[3]**(%)**	$11,04$	$9,39$	$6,77$	$11,14$	$18,31$	$25,44$	$25,73$
Num. Trans Area (cm/cm^2)	$5,17$	$5,13$	$5,11$	$4,93$	$5,00$	$179,4$	$178,3$
Relative Error (%)	$3,3$	$2,6$	$2,2$	$-1,4$	$0,0$	$1,5$	$0,9$

Table 5.1: Properties of the meshes and results of the calculation of area

The results are quite satisfactory since the main goal here was to obtain a method to weight the area assigned to the contacts in a way that the real geometry was respected. The maximum relative error in the measure is 3,3% which is a very low value. It can be seen that this method works nicely in 2D and 3D for homogeneous meshes and for heterogeneous meshes too.

Note on the implementation: This novel methodology does not make the method more expensive in terms of CPU time since the correction of areas should be done only in the first time step and, thereafter, those stored values values should be used. Here it is considered that the contacting area doesn't change, that is obviously valid for small deformations, *i.e.* before fracturing. In KDEM code this correction is not performed for new contacts generated from particles that have broken bonds and interact with new particles. In this case, frictional contact appears and the stiffness of the contact is not a key issue. However, those particles that are in contact with their initial neighbouring particles, even if their bonds are broken, keeping the same stored area.

[1] Relative Standard Deviation: the ratio of standard deviation over the mean value times 100; this way a dimnesionless value in % is obtained, and so, a good idea of the variation is obtained.

[2] Coordination Number: represents the average number of neighbours per particle. It is calculated by the following expression: $NC = 2 \cdot N_{Contacts}/N_{Particles}$.

[3] Mesh porosity: the complementary of the volume fraction of particles in the mesh over the model volume.

5.1.3 Young's Modulus and Poison Ratio

Input parameters - Calibration study

In the previous section it has been proved that the area assigned to the contact, $\bar{A}_c$, was consistently calculated; the results in linear elasticity however, are not good enough if the found expressions 4.34 are used and vary for different meshes. In this chapter a study of how a modification of the derived expressions incur in the results is done:

$$\begin{aligned} K_n &= \alpha \cdot \pi E \cdot \hat{A}/(R_1 + R_2 + \delta_0) \\ K_s &= \beta \cdot \pi G \cdot \hat{A}/(R_1 + R_2 + \delta_0) \end{aligned} \tag{5.7}$$

To assign the values of the local constitutive parameters, a calibration procedure can be used; this procedure has been followed by CIMNE researchers with the *Dempack-model* and other researchers on the field, see Labra [15], Donzé et al. [9, 31]. The proceedure It is generally based on the simulation of quasi-static uniaxial and triaxial compression/traction tests; in the case studied in this section only the normal and tangential stiffness have to be calibrated to obtain a correct linear elastic response.

The test will be done by changing the values of *alpha* and *beta* in the constituitive contact coefficients 5.7.

Output quantities - Macro parameters analysis

The purpose of this section is to obtain numerical values of the Young's modulus and Poisson's ratio simulated by the KDEM application for different meshes. The input values, for academic purposes are $E = 20Mpa$ and $\nu = 0.25$. The model corresponds to the meshes presented previously, a $5cm \times 10cm$ rectangular domain.

The test consist on applying uniaxial strain in the vertical direction and measuring macroscopically the stresses in the same directions as well as the strains in horitzontal direction due to the Poisson's effect. In order to have a uniform stress-strain state on the whole model, no other boundary conditions or external forces (gravity, etc.) are applied.

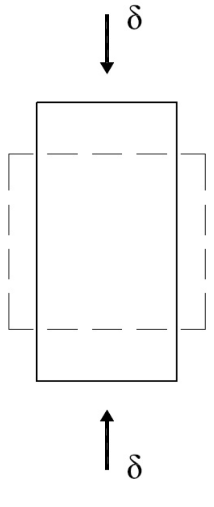


Figure 5.13: Uniaxial strain on the rectangular model.

From linear elasticity theory its known that:

$$
\begin{aligned}
\boldsymbol{\varepsilon}_x &= \frac{1}{E}\left[\boldsymbol{\sigma}_x - \nu(\boldsymbol{\sigma}_y + \boldsymbol{\sigma}_z)\right] \\
\boldsymbol{\varepsilon}_y &= \frac{1}{E}\left[\boldsymbol{\sigma}_y - \nu(\boldsymbol{\sigma}_x + \boldsymbol{\sigma}_z)\right] \\
\boldsymbol{\varepsilon}_z &= \frac{1}{E}\left[\boldsymbol{\sigma}_z - \nu(\boldsymbol{\sigma}_x + \boldsymbol{\sigma}_y)\right]
\end{aligned}
\tag{5.8}
$$

In the uniaxial compression problem in y direction, in both 2D and 3D, the meaningful equations simplify to:

$$
\begin{aligned}
\boldsymbol{\varepsilon}_y &= \frac{1}{E}\boldsymbol{\sigma}_y \\
\boldsymbol{\varepsilon}_x &= -\frac{1}{E}\nu\boldsymbol{\sigma}_y
\end{aligned}
\tag{5.9}
$$

And so, assuming uniform stress and strain in the complete model, a measure of the macroscopic poison ratio is:

$$
\nu = -\frac{\boldsymbol{\varepsilon}_x}{\boldsymbol{\varepsilon}_y} \tag{5.10}
$$

Some monitoring particles have been marked in order to get the stress-strain curves and the poisson ratio. See how this is done in section 6.5.

Results obtained in the different meshes

A set of tests have been done while varying the different calibration parameters ranging between the following values:

Values of α in the analysis: $[\ 1.00, 1.20, 1.50, 1.75, 2.00, 2.50\ ]$
Values of β in the analysis: $[\ 0.00, 1.00, 2.00, 2\cdot(1.00+\nu), 3.00\ ]$

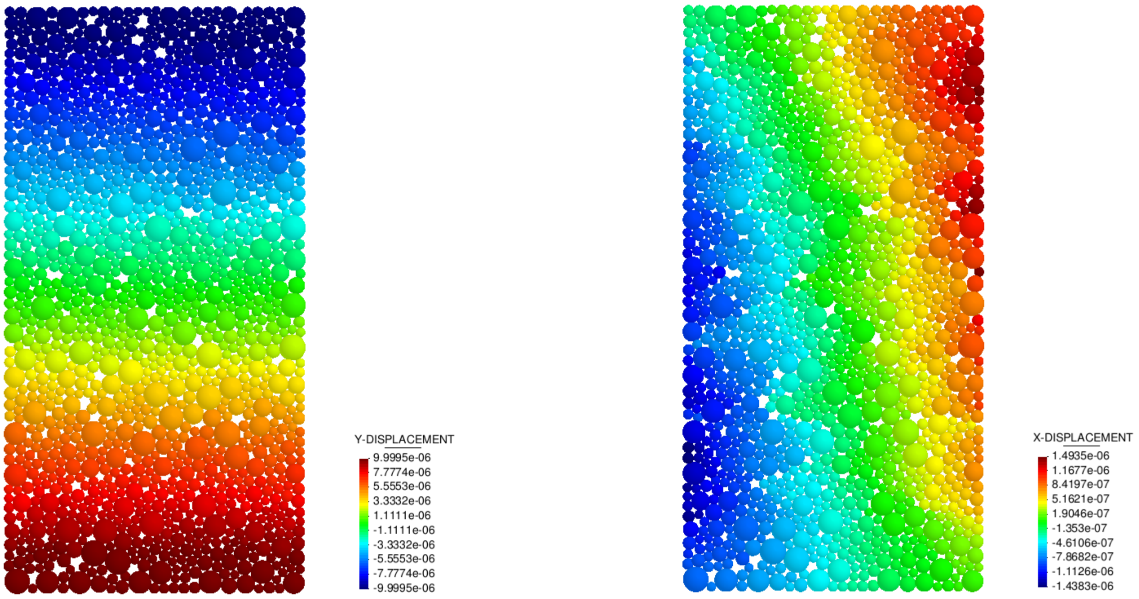


Figure 5.14: Y displacement and X displacement in Mesh 2 for $\alpha = 1.21$ and $\beta = 1.00$.

In Figure 5.14 a view of the deformation in X and Y can be seen for a specific mesh. Due to the heterogenity of the mesh some twist occurs and so, the measured poisson may not be highly accurate.

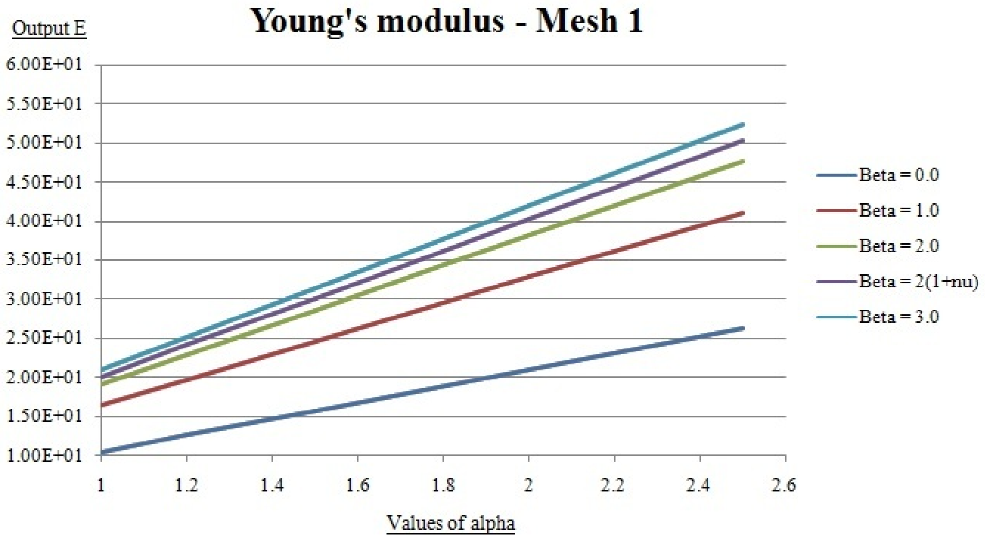


Figure 5.15: Young's modulus [Mpa] for the mesh 1

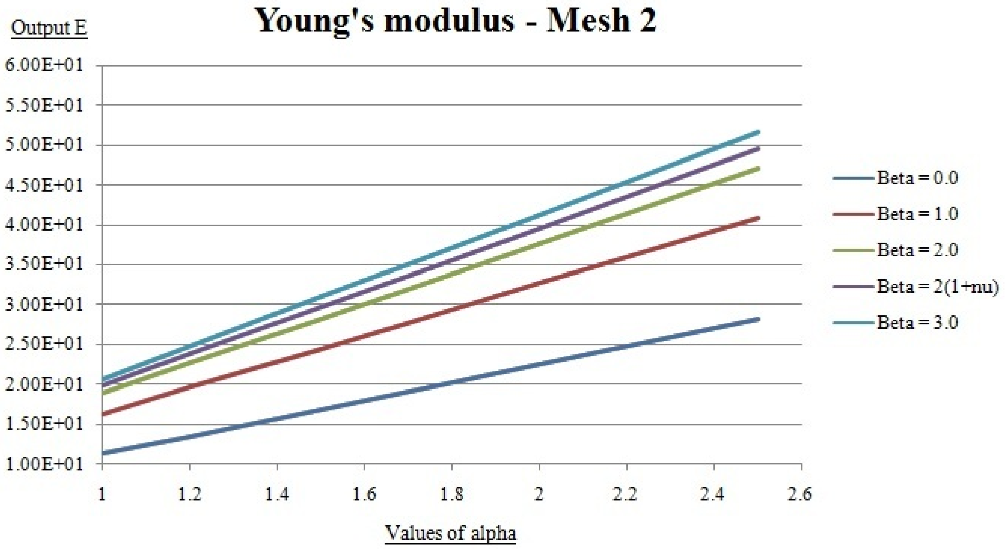


Figure 5.16: Young's modulus [Mpa] for the mesh 2

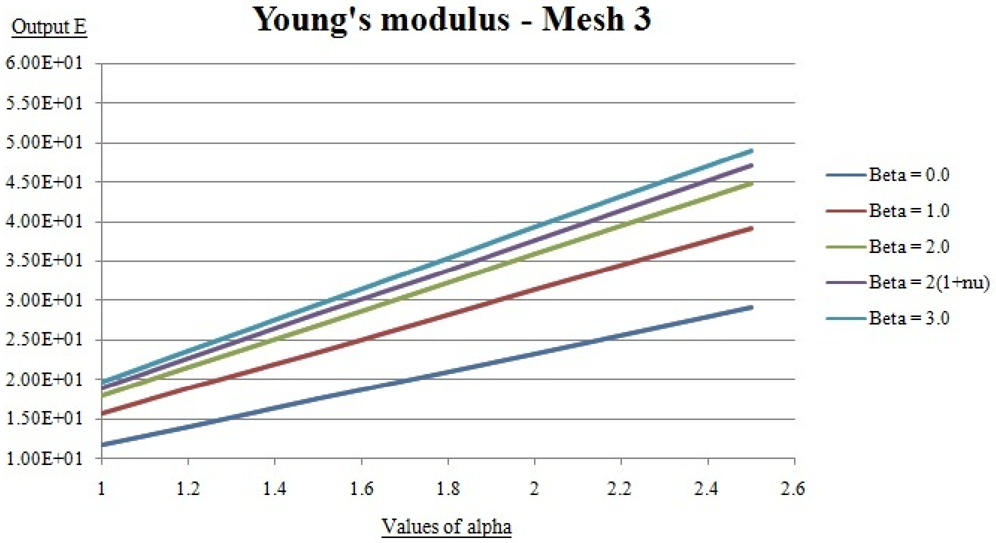


Figure 5.17: Young's modulus [Mpa] for the mesh 3

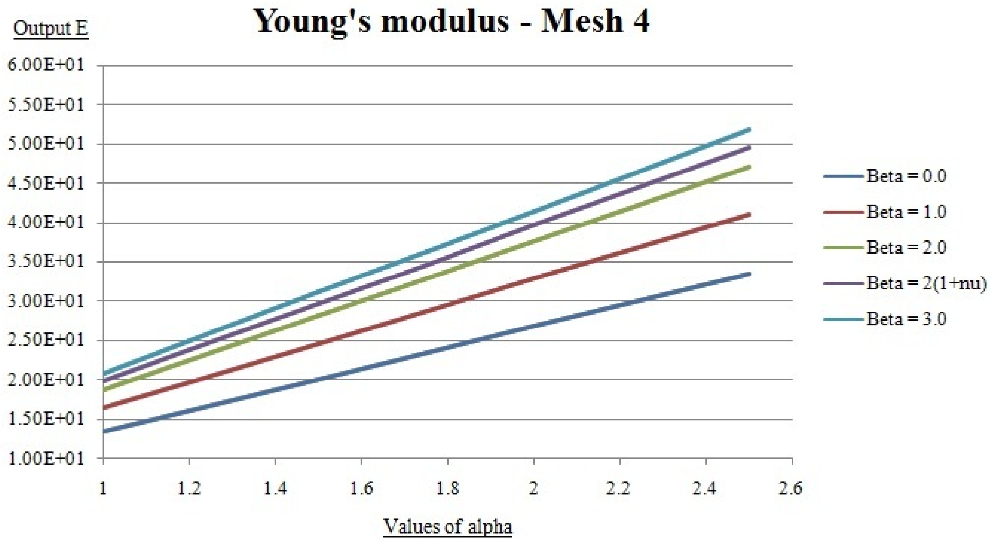


Figure 5.18: Young's modulus [Mpa] for the mesh 4

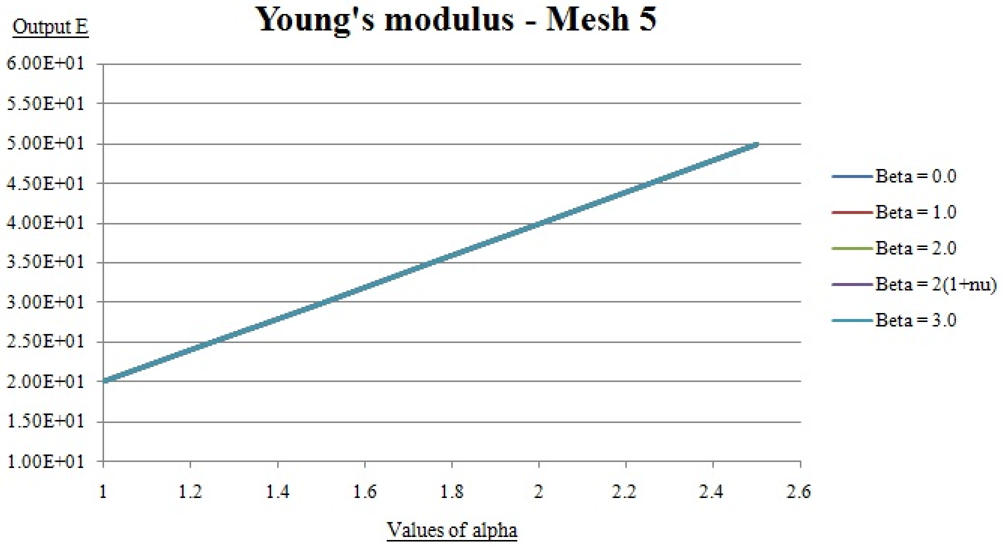


Figure 5.19: Young's modulus [Mpa] for the mesh 5

It can be seen in the results from the output Young's modulus that they evolve in a similar way for evey mesh (except for mesh 5 which is regular). The following points is what can be concluded analysing the presented cases.

1. The correct value of Young's modulus is recovered when the tangential stiffness becomes larger for every mesh. This is due to the correction of areas which ensures that the static forces are well calculated in the system formed by linear springs (weighted by the areas).

2. There is an absolute linear relationship between the local α value and the global stiffness of the model as it was expected in small deformations.

3. The values obtained for the different families of β values yield similar results for all the meshes except for the completely regular one.

4. Given a mesh and a value of β that calibrates the Poisson ratio, a value of alpha which recovers the correct macroscopical behaviour in terms of Young's modulus can be found in the range of $[1.0 - \infty]$.

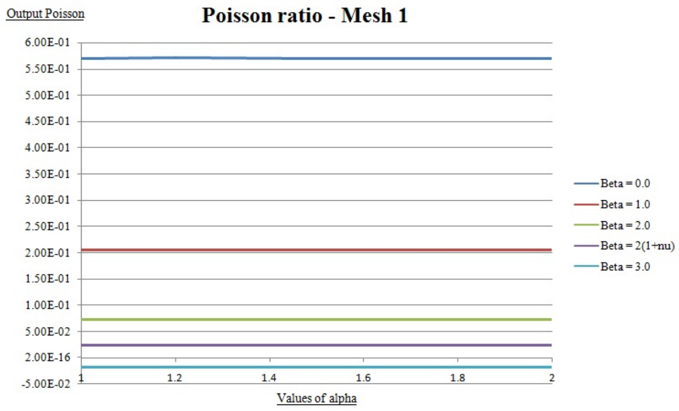


Figure 5.20: Poisson ratio mesh 1

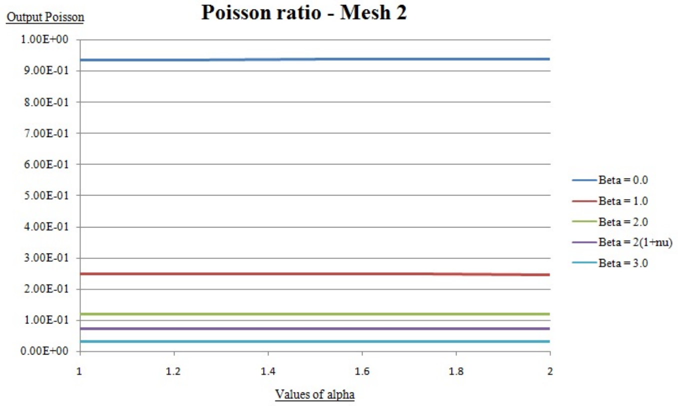


Figure 5.21: Poisson ratio mesh 2

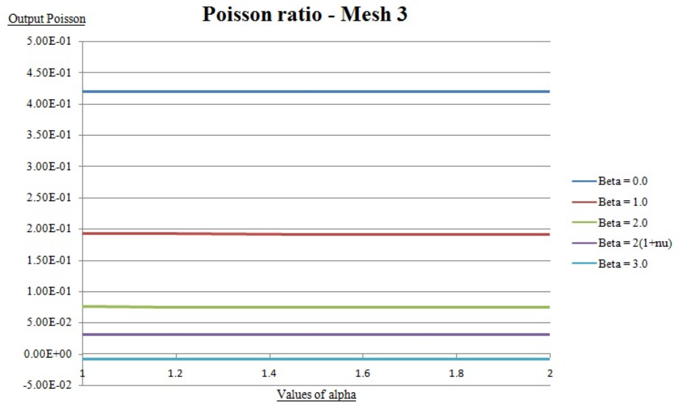


Figure 5.22: Poisson ratio mesh 3

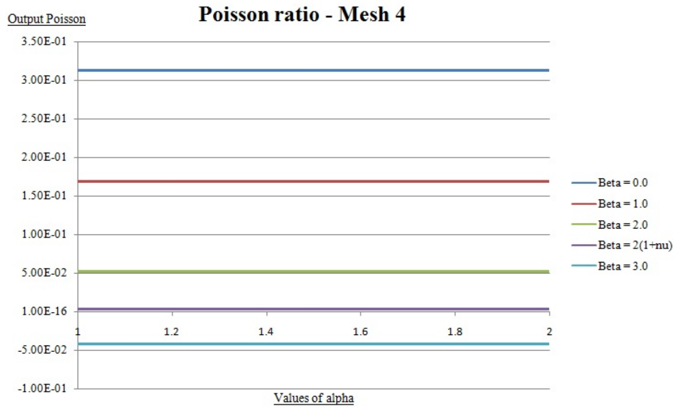


Figure 5.23: Poisson ratio mesh 4

Results of Poisson Ratio of mesh 5 have not been included since that mesh yields a 0.0 value in any case. From these results the following can be concluded:

1. The values of Poisson ratio don't depend on the value of α, which is good news.

2. It seems that the more heterogenous the mesh the higher the ratios that can be obtained. It is debatable which mesh is more heterogenous in terms of distribuiton, mesh 1 or mesh 2. Mesh 2 yields a ratio of 0.9 for $\beta = 0.0$.

3. The values of Poisson ratio seem to converge to some value near zero (that can be slightly negative) when the value of β increases.

4. The maximum Poisson ration is obtained for $\beta = 0.0$, it highly depends on the mesh configuration and it can be larger than 0.5 what is physically impossible.

5. The values obtained for $\beta = 1.0$ yield a good aproximation of the poisson ratio for all the meshes except for mesh 5 which yields obviously a null Poisson ration.

It seems a good strategy to choose a high value of β, or in the extreme, restrict the tangential displacement in the contacts, in order to obtain an exact Young's modulus value (without using α) and a natural null or negligible poisson ratio. Once that is obtained, Poisson deformation should be introduced by some other mechanism depending on the stresses and strains present at the system. This is a topic under research at the moment by the author.

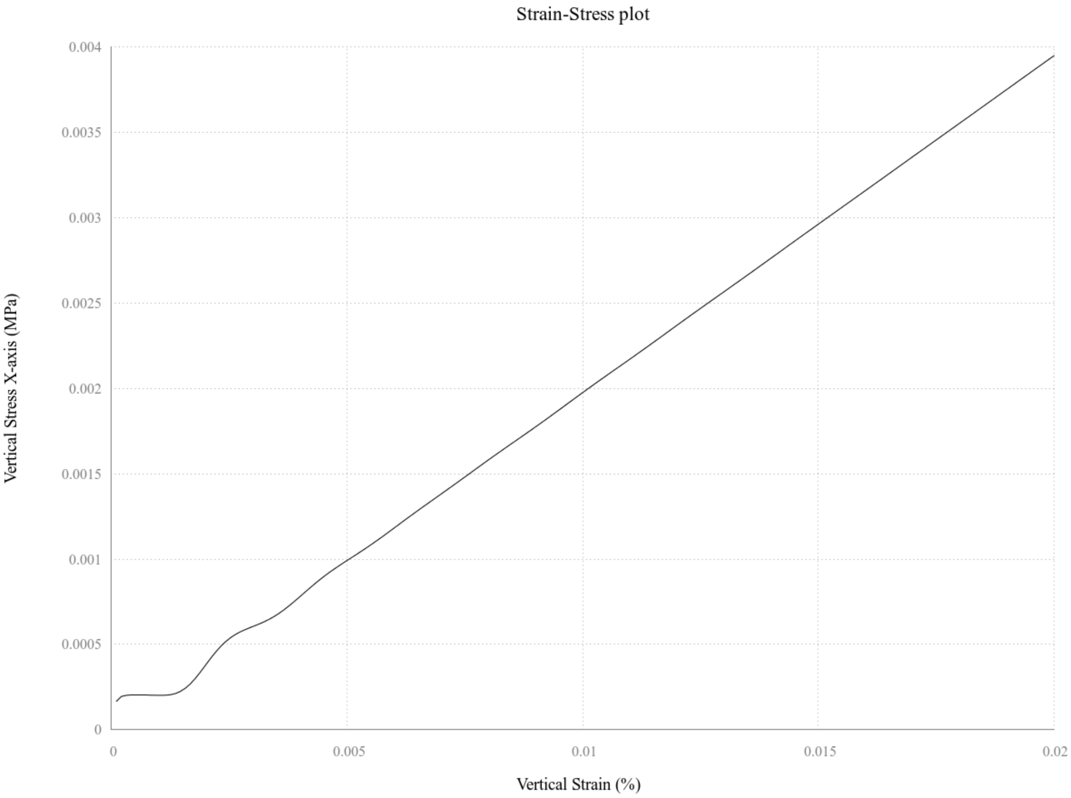


Figure 5.24: Stress-Strain plot of Mesh 2 for $\alpha = 1.21$ and $\beta = 1.00$.

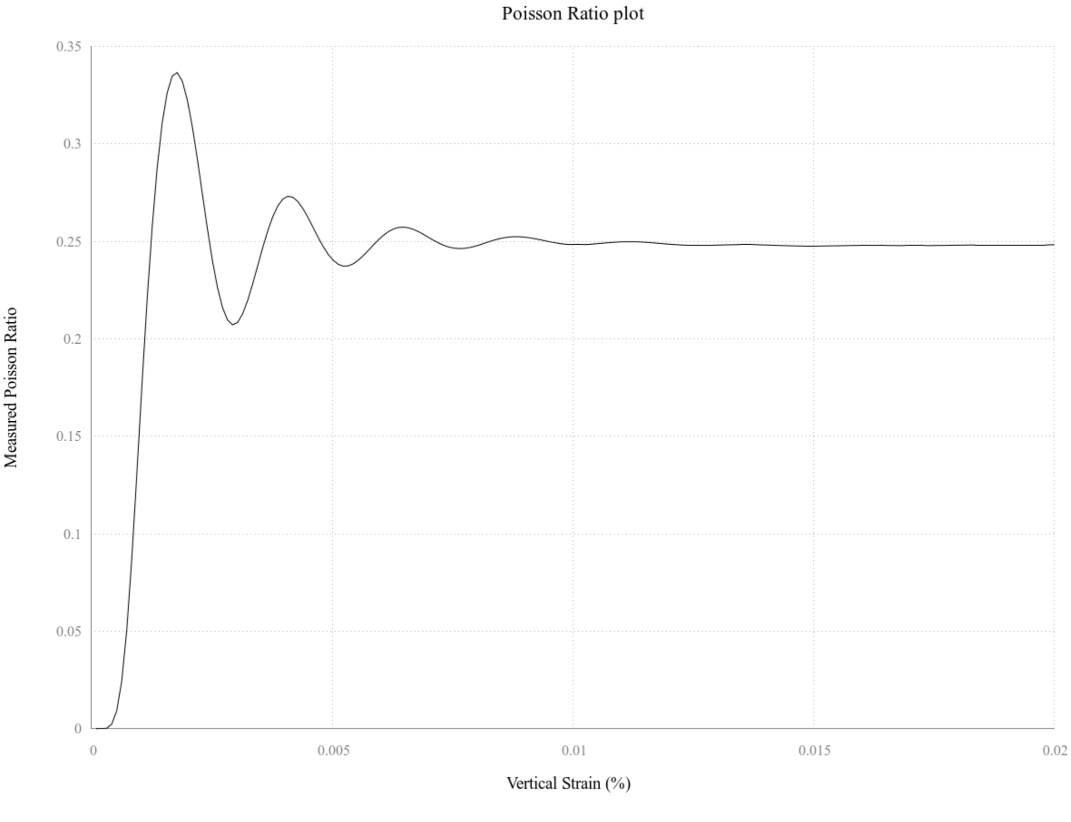


Figure 5.25: Poisson Ratio plot of Mesh 2 for $\alpha = 1.21$ and $\beta = 1.00$.

These plots belong to one of the cases that were run during the simulations. In mesh 2 for a $\beta = 1.00$ the exact poisson ratio is recovered after some oscillation (due to dynamic loading) in the firsts time steps. The same oscillation occur in the Youngs modulus curve. The value that yields the correct value of the Young modulus is in this case: $\beta = 1.21$.

5.2 Mesh dependence

The Mesh-dependence has been already shown also for the simple case of linear elasticity where, using the same *micro* parameters K_n and K_s in the contacts, different Young's modulus and Poisson ratio are obtained, *macro* parameters, are measured.

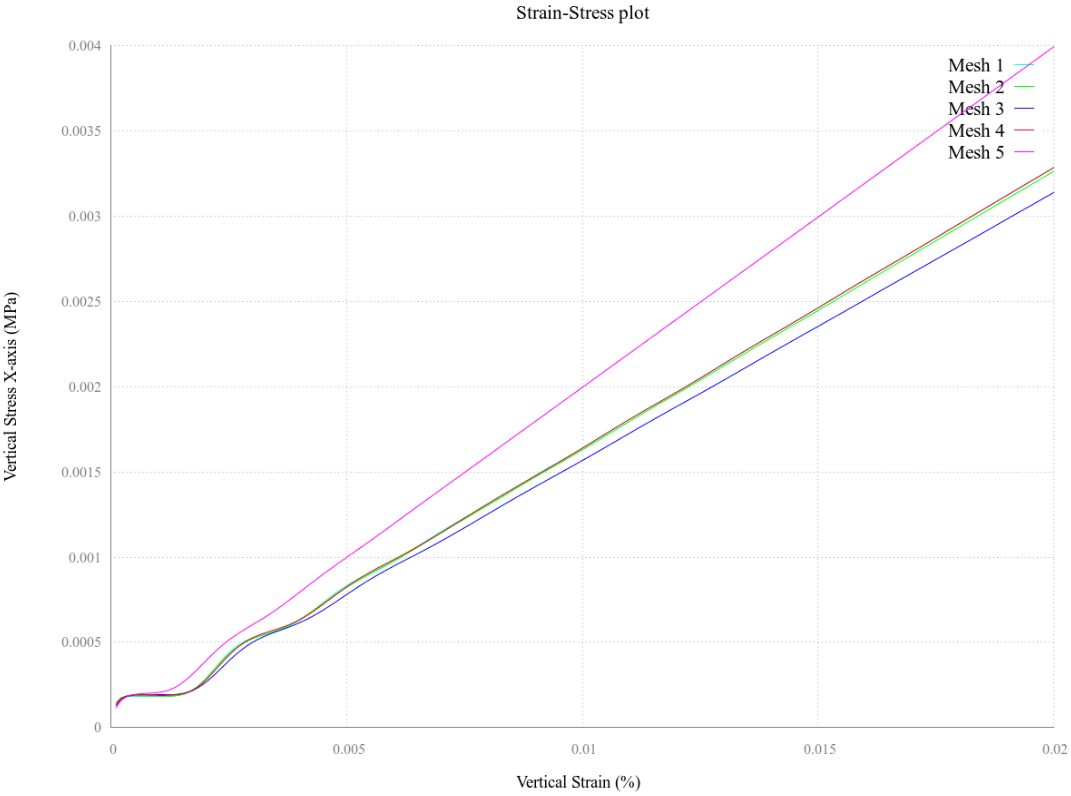


Figure 5.26: Stress-Strain plot for all meshes with $\alpha = 1.00$ and $\beta = 1.00$.

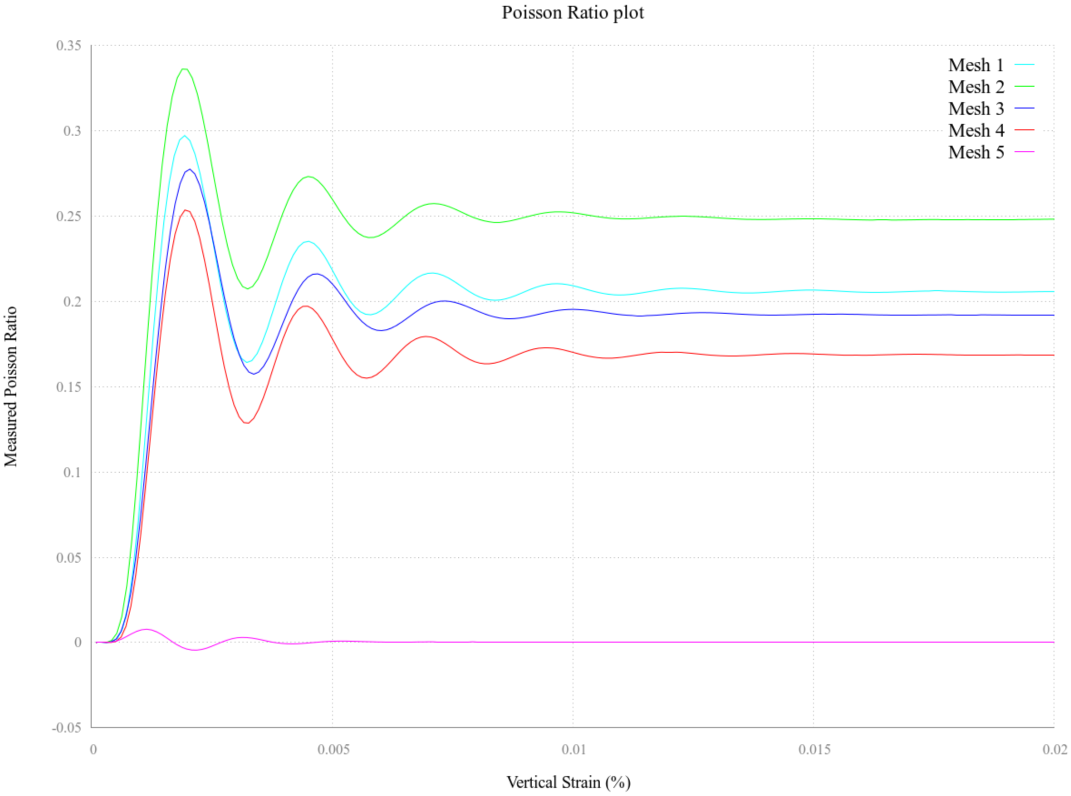


Figure 5.27: Poisson Ratio plot for all meshes with $\alpha = 1.00$ and $\beta = 1.00$.

5.3 Convergence

In order to analyse convergence of the method, this will be done analysing a case which is of the interest of this thesis and also can be compared against experimental data; a UCS test for concrete reported in [29] and performed by Dempack software as well in the CIMNE report [20].

For this case, the *Dempack-model* is applied with a constitutive law that includes non-linear elasticity, plasticity and damage. The parameters of the simulation are detailed in Section 7.

The convergence analysis will be done from three prespectives: Number of elements, explicit time step selection and velocity of loading.

5.3.1 Convergence in number of elements

The strains and stresses state in a specimen during a UCS simulation are more or less homogeneous in all the model. So a general remeshing with larger number of elements should give results that converge to a more accurate one if the method converges in number of elements.

The test has been done with the following meshes:

Mesh	*100*	*250*	*500*	*1k*	*5k*	*13k*	*70k*
Number of Elements	107	251	497	1004	4959	13500	71852
Mean Radius (mm)	18.75	14.27	11.50	9.20	5.40	3.86	2.26
Coordination Number	7.64	8.32	8.87	9.29	9.63	9.55	10.15

Table 5.2: Meshes used in the convergence analysis

Time step selection

In this case a comparison of the same problem with the same characteristics is performed with meshes of different sizes. Every other variables have to be the same in order to compare in a fair situation; the time step, however, has to be modified for the $70k$ mesh which has a large number of elements (70000) due to stability requirements.

The critical time step can be estimated by means of the expression presented in Section 3.4.2:

$$\Delta t_{cr} = \min 2\sqrt{\frac{m_i}{k_i}} \tag{5.11}$$

It is interesting to analyse which is the dependence of the natural frequency on the particle radius (using k as the normal stiffness corresponding to a sphere contacting with another of the same characteristics):

$$\omega_i = \sqrt{\frac{m_i}{k_i}} = \sqrt{\frac{\frac{3\pi R_i^3}{4} \cdot \rho_i}{\pi E \cdot \frac{R_1^2}{2 \cdot R_1 + \delta_0}}} \text{ simplifying for } \delta_0 \to 0 \ \omega_i \propto \sqrt{R_i^2} \propto R_i \tag{5.12}$$

Taking the $13k$ mesh as the reference one , the mean radius obtained was of 3.50 mm and the minimum one 1.00 mm. A fair approximation of the reference citical radius for the simulation can be 2.00 mm since the case of two particles of values arround 1.00 mm contacting is difficult to be present in the model. Also with the Young's modulus of 2.7Gpa and desity of $\rho = 2500kg/m^3$ the following estimation of the critical time can be done:

$$\Delta t_{cr} \approx 6e^{-7} \tag{5.13}$$

In this simulation the value of $\Delta t_{cr} = 1e-7$ was used and turned out to be stable.

Having said that, a special time step will be used for the mesh $70k$. An estimation of the new time step can be done by scaling the times in funcion of the ratio of the number of elements.

$$\frac{\Delta t_{cr2}}{\Delta t_{cr1}} \propto \frac{\hat{R}_2}{\hat{R}_1} \tag{5.14}$$

$$\frac{\hat{R}_2}{\hat{R}_1} \propto \sqrt[3]{\frac{V_2}{V_1}} \propto \sqrt[3]{\frac{N_1}{N_2}} \tag{5.15}$$

And so the critical time can be estimated as:

$$\Delta t_{cr_{70k}} =\propto \Delta t_{cr_{13k}} \sqrt[3]{\frac{13000}{70000}} = 0.57e-7 \tag{5.16}$$

For comodity and safety, a time step of $0.5e-7$ has been used.

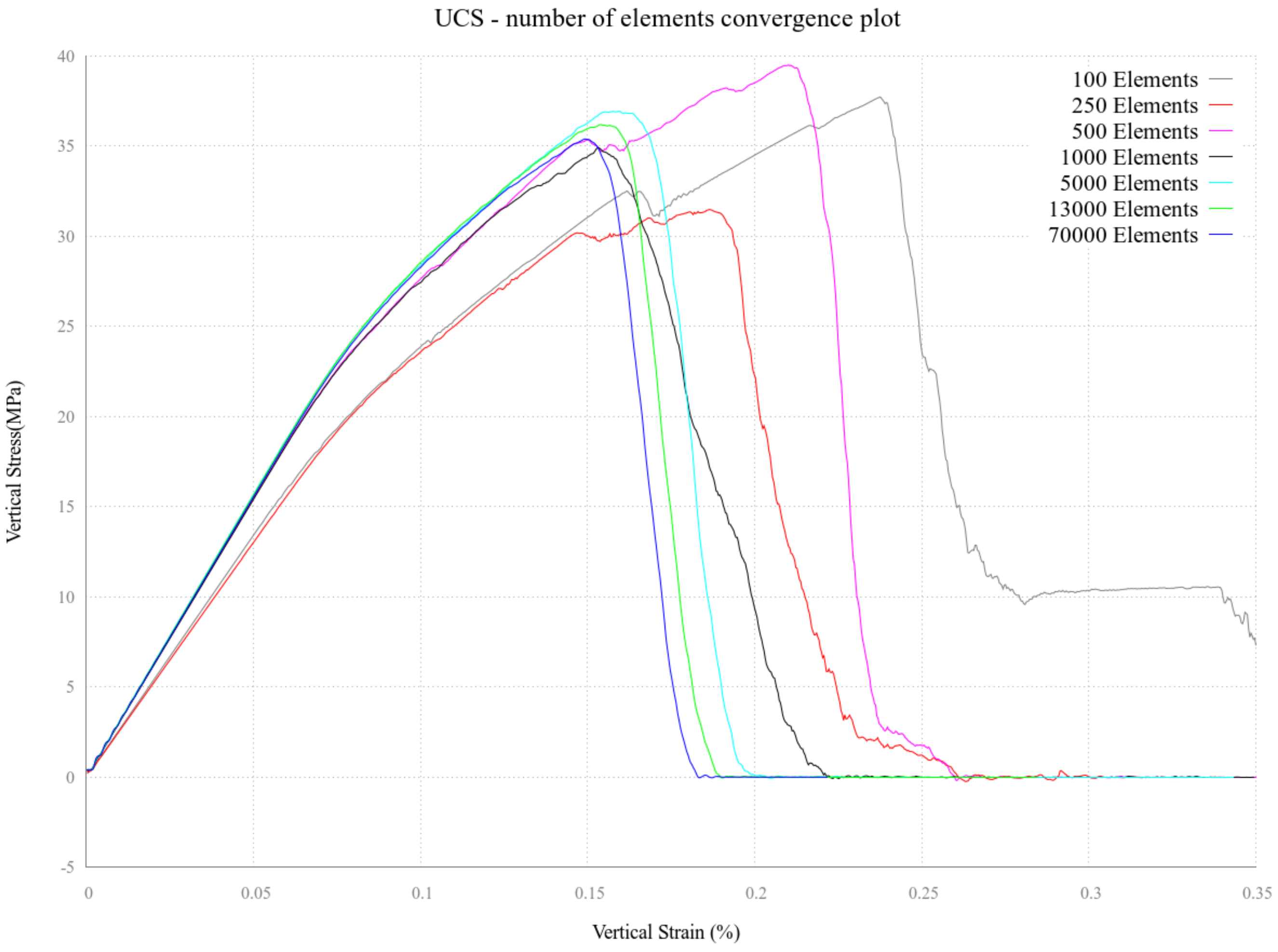


Figure 5.28: Convergence analysis for the number of particles in the discretization.

Although the results seem to converge increasing the number of elements, several more test have to be done to have a prediction of the order of the convergence for the different interesting variables such as displacement or stresses. Appart from the plots and the macroscopical measures, the visualitzation of the results and the cracks tracking is obviously better defined for larger meshes. The next figure, where the displacement field of the 70k mesh is displayed, gives an impressive view of the failure mode in the model which pretty much matches the real one for the UCS tests performed on concrete.

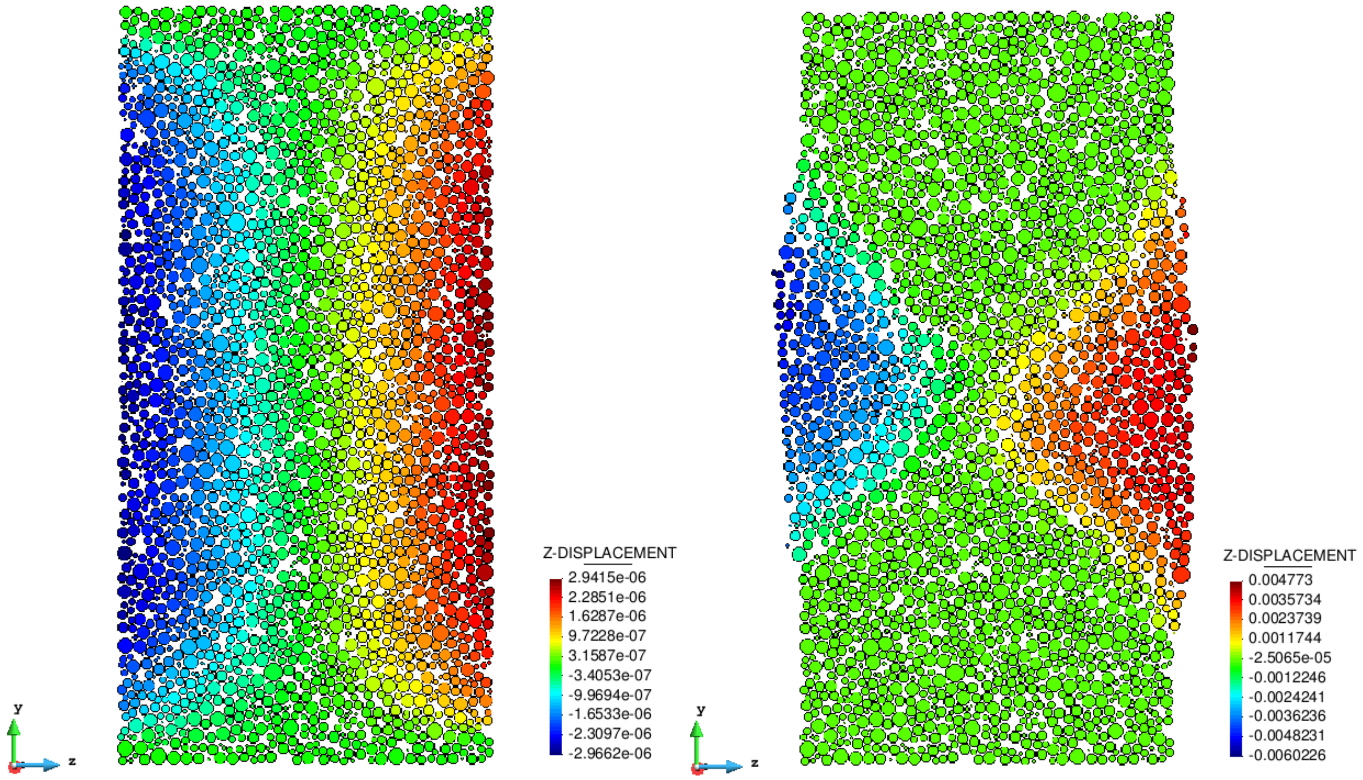


Figure 5.29: Z Displacement of a centered section of the specimen for the 70k mesh at the begining of the loading and after failure (deformation 2 times enlarged).

5.3.2 Convergence in time step

As it has been pointed out in the previous section, the explicit integration scheme has a stability limit in the time step selection which has been analysed in Section 3.4.2 that has to be respected. On the other hand the smaller the time step selected, the more accurate the results and the less the gain of energy that the penalty method produces systematically in a explicit time discretization scheme.

The way this critical timestep is calculated, approximating the highest frequency of the system to the hypothetic highest one, normally leads to an unsafe value; it has been said already in Section 3.4.2 that for practical cases normally a safety factor is used with a value arround 0.17 afecting the estimated timestep.

$$\Delta t = \beta \Delta tcr \tag{5.17}$$

Another consideration has to be done about the critical time step; in many cases and in the case presented here too, where some particles have imposed velocities, the critical time step may let the particles have an excesive indentation if the imposed velocity is high enough and so the make the calculations unstable.

The value for the critical time step obtained in the previous section for the $13k$ mesh is of: $\Delta t_{cr} \approx 6e^{-7}$. The following values have been used in the analysis: $[1e^{-8}, 5e^{-8}, 1e^{-7}, 5e^{-7}]$.

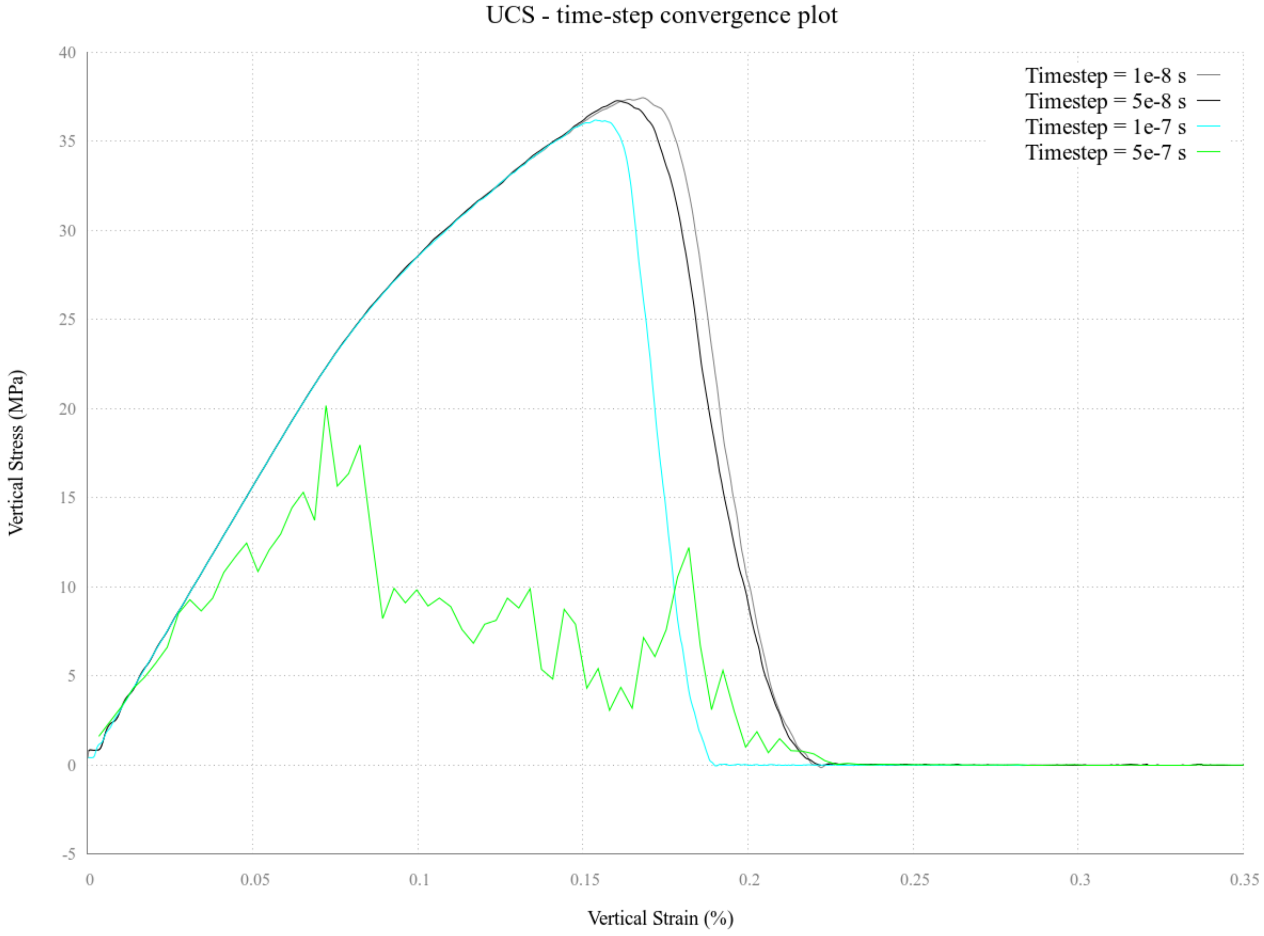


Figure 5.30: Convergence analysis for the time step selection.

The results corroborate that a time step of slightly lower than the critical one $(5e-7)$ is not enough for the stability of the system. However, using a time step of $1e-7$, which is more or less the value that yields from applying the reduction of the critcial step with the safety vaule of $\beta = 0.17$, the calculation is stable. The solution does converge when the time step is smaller.

5.3.3 Convergence in quasi-staticity

The DEM applied to continuum fields is able to simulate dynamic problems naturally since it comes from the basic DEM formulation which is completelly dynamic in its conception. However, in many engineering applications such as the ones presented in this thesis, the concrete test simulations, it is desired to simulate static problems. The problem however is solved in the case of the UCS imposing strain along the speciment and measuring the forces on the top and bottom of the speciment; in this sense, the solution is found by tracking a explicit time evolving solution. It has been reported from the laboratory experiments obtained [29] that the tests simulated in this tesis (UCS, Brazilian, Triaxial) can be considered *quasi-static* since the loading velocity is arround $0.0006 mm/s$.

The same UCS test have been done modifying the loading velocity in the range: $[0.002, 0.020, 0.100, 0.200, 1.000]$ m/s. This particular case has been designed with no damage in the constituitive law in order to check if the model can reproduce the brittle behaviour.

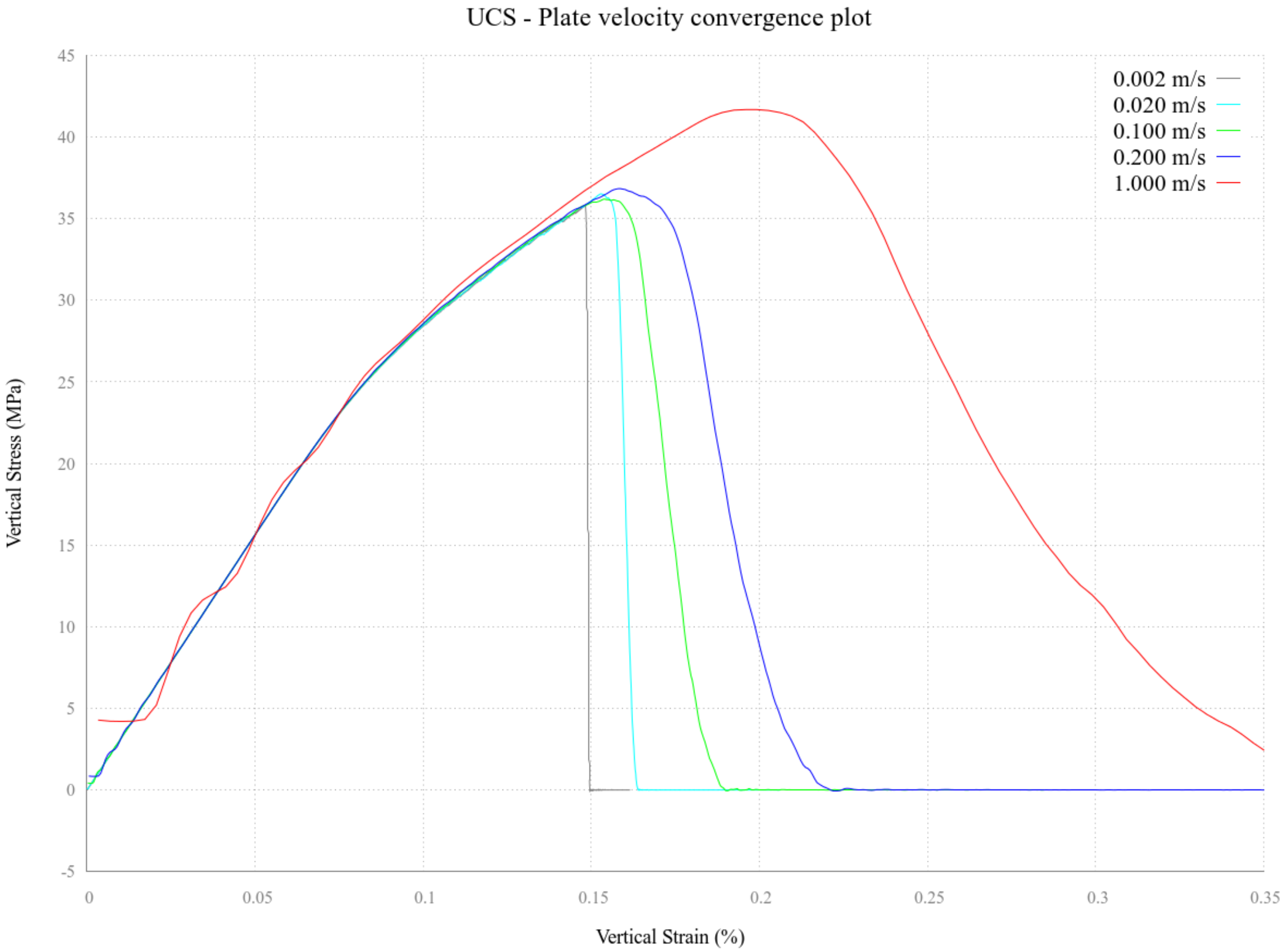


Figure 5.31: Convergence analysis for the loading velocity.

In the *quasi-static* problems, a few special considerations have to be done:

- The experimental loading velocities of $0.0006 mm/s$ are not feasible to simulate using time step value of the order of $1e-7s$. Considering that the test ends at a deformation of 0.25%, which corresponds to $0.75mm$ in a $30cm$ specimen, over 20 minutes of simulation would be needed to reach this deformation with the set loading velocity. This would yield to $1,25e10$ number of calculation steps. The numerical velocity for the loading is taken arround $0,2m/s$ instead.
- Appart from the numerical damping included to cut down the gain of energy of the explicit method, extra damping is needed in the contacts for killing the dynamic effects. In this sense,

the damping is set to a value near the critical (0.9), killing all the local contact unbalanced forces.

- Another damping, reported in this thesis in the section 4.5.3, which is a non-viscous type global damping has been considered:

$$\mathbf{F}_i^{\text{damp}} = -\alpha^t \left| \mathbf{F}_i^{\text{ext}} + \sum_{c=1}^{n_i^c} \mathbf{F}_i^c \right| \frac{\dot{\mathbf{u}}_i}{|\dot{\mathbf{u}}_i|} \tag{5.18}$$

  This damping reduces the total unbalanced forces resulting in every partilce. A value of $\alpha = 0.2$ is used in the analysis.

- The mass of the particles is also modifiable since the accelerations are not an interesting result in the *quasi-static* simulation. In this particular analysis the mass value has not been modified.

- The velocity of the loading is a key factor that determines as well as the above mentioned parameters the *quasi staticity* of the simulation.

The results clearly show that the loading velocity has influence on the results. The elastic part is well calculated even for the 1.0 m/s case, where the dynamic effects can be seen in the elastic waves produced by the excessively fast loading. The failure however, gives higher peak values and higher deformation ranges and yields a ductile behaviour. On the other hand, the slowest case of $0.002 m/s$ yields a extremelly brittle behaviour as it should be.

5.3.4 Stress evaluation and failure criteria

This section has the objective of demonstrating that the currently used failure criteria, which is applied by many of the researchers and codes, have limitations in terms of not considering well the real tridimensional stress and strain states in the continuum.

It has been already said that the majority of the codes, and also the work presented here, consider DEM as a phenomenological method which its failure parameters of a given model have to be calibrated; this is done by performing different typified tests and tunning the parameters that fit the curves of experimental results [31]. The methods considered in most of the cases however, which are based on unidimensional failure criteria on the contacts, don't suffice to represent the real behaviour of the failure mechanisms in the continua.

Evaluation of stress and strain tensors

Although in the present work the evaluation of stresses and strains has been done by means of macroscopial measures, it is possible to evaluate locally an average strain and stress tensor to analyze the field of inter-particle contacts. The definition of the average stress tensor is widely discussed in literature for granular materials and discrete media (see [1, 13]). Normally it is defined in terms of the force acting on the contact between particles and the geometry of the assembly. The derivation of the micromechanical stress tensor and the averaging procedure can be found in Annex D.

The average stress tensor in a domain defined by the volume V is written as:

$$\bar{\sigma}_{ij} = \frac{1}{V} \sum_{c \in V} f_i^c l_j^c \tag{5.19}$$

where l_j is the called branch vector, which joins the center of the particles in contact ($x_q - x_p$).

In the case of the strain tensor, different formulations can be found in the literature. As a general formulation, the following can be used to define the average strain tensor:

$$\bar{\varepsilon}_{ij} = \frac{1}{V} \sum_e \delta_i^e d_j^e \tag{5.20}$$

where d_i^e is a characteristic vector related to the relative displacement, and complementary to the branch vector.

This utility hasn't been implemented in the KDEM-Application yet. Appart from being a good visualization tool, it may be very useful to determine the average constituitive elaticity matrix of each assembly. It seems a good idea to make use of this information in order to correct the behaviour of the model in order to better simulate the linear elasticity. This is a topic under research by the author of the thesis and some researchers in Cimne.

But most important, an inmediate application of this would be the consideration of more complex 3D failure mechanisms that take into account the fact that the material failure can not be defined by 1D criteria in the contact directions and the complete stress state in those evaluation points (contacts) should be used.

Results of 1D criteria in hydrostatic tests.

To show the idea presented in this section a simple test has been performed, a cylindrical concrete speciment discretized by DEM particles has been subjected to a hydrostatic pressure simulating a hydrostatic compression test.

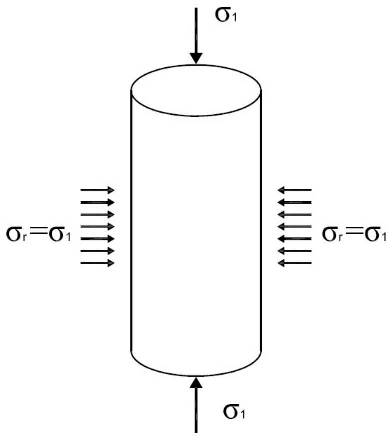


Figure 5.32: Hydrostatic.

Figure 5.33 shows the value of the variable "*FAILURE_CRITERION_STATE*" represented on the contact elements (linear elements representing the contact between the spheric particles) for a hydrostatic test at $30Mpa$. This failure state is calculated as the maximum value between two quantities. In the case of shear it is calculated as the current shear state over the shear strength (τ_{now}/τ^f) which depends on the confinement of the contact σ_n or equivalently in forces $F_s/\mathcal{F}_s$. In the case of tension, the ratio is calculated as the current tension state in the contact over the tension strength, in forces $F_{n_t}/\mathcal{F}_{n_t}$, or in the case when the damage law is applied, it is calculated as the tensile deformation over the maximum damaged deformation u_n/u_n^f. This ratio takes a value of 1.0 in the broken bonds.

$$FCS = \begin{cases} \max(\frac{F_s}{\mathcal{F}}, \frac{F_{n_t}}{\mathcal{F}_{n_t}}) & \text{for} \quad \text{Uncoupled Mohr-Coulomb without damag law.} \\ \max(\frac{F_s}{\mathcal{F}}, \frac{u_n}{u_n^f}) & \text{for} \quad \text{Uncoupled Mohr-Coulomb with damage law.} \\ 1.0 & \text{for} \quad \text{Broken bonds} \end{cases} \tag{5.21}$$

The problem is obviously that the contact only captures the confinement in one direction, the normal one, and so depending on which is the internal friction angle, an increase of shear stress would lead the contact to failure.

In Figure 5.33 the values are ranged $[0.0 - 0.3]$ and it can be seen that in some contact elements, values near the 30% of the failure criterion have been reached. At this point of confinement of 30 Mpa the triaxial test would start with the uniaxial loading; it is clearly unrealistic to have some contacts which are closer to the failure after a hydrostatic confinement which would have to do the opposite effect on the microstructure, push the failure point further.

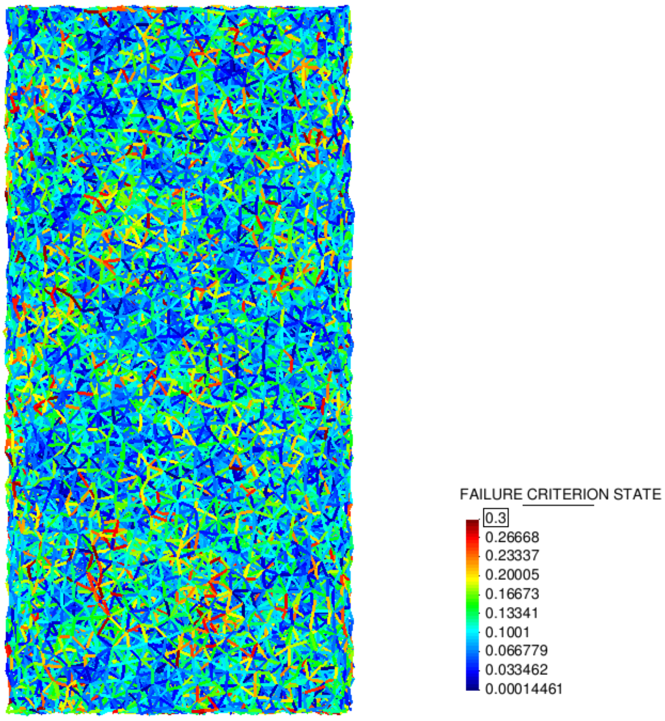


Figure 5.33: Failure criterion state.

Chapter 6

Kratos DEM-Application - Code Implementation and usage

6.1 Kratos-Multiphysics

6.1.1 What is Kratos?

Kratos is a framework for building multi-disciplinary finite element programs. It provides several tools for easy implementation of finite element applications and a common platform providing effortless interaction between them. Kratos has an innovative variable base interface designed to be used at different levels of abstraction and implemented to be very clear and extendible. It also provides an efficient yet flexible data structure which can be used to store any type of data in a type-safe manner.

The Python scripting language is used to define the main procedure of Kratos which significantly improves the flexibility of the framework in time of use.

The kernel and application approach is used to reduce the possible conflicts arising between developers of different fields. Also layers are designed to reflect the working space of different people, considering their programming knowledge. It permits to create your new application starting from a template for every basic generic part of your program. The application connects to the main Kratos general structure and it benefits of its data base common utilities for general FEM-like engineering programs ready to be used, the IO structure to interact with graphical interfaces and the powerful and optimized usage of the combined C++ and python languages.

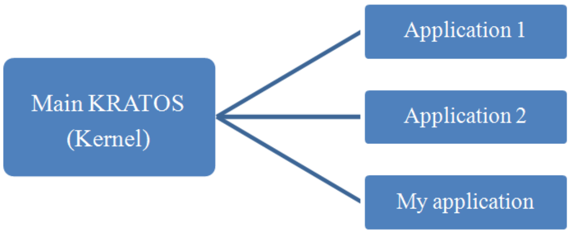


Figure 6.1: Kernel and application apporach.

6.1.2 Who may use Kratos?

Some potential users of Kratos are:

- Research engineers: these developers are considered to be more expert in numerical methods and engineering calculation methods, from the physical and mathematical points of view, than in C++ programming. For this reason, Kratos provides their requirements without involving them in advanced programming concepts.

- Application Developers: these users are less interested in finite element programming and their programming knowledge may vary from very expert to higher than basic. They may use not only Kratos itself but also any other applications provided by finite element developers, or other application developers. Developers of optimization programs or design tools are the typical users of this kind.

- Package Users Engineers: and designers are other users of Kratos. They use the complete package of Kratos and its applications to model and solve their problem without getting involved in internal programming of this package. For these users Kratos has to provide a flexible external interface to enable them use different features of Kratos without changing its implementation.

6.1.3 Who is Kratos?

The Kratos structure, due to its multidisciplinary nature, has to support the wide variety of algorithms involved in different areas. That's the principal reason that explains the variety of people, mostly engineers, composing the Kratos Community, which encourages you to visit the website to learn more about Kratos *http://www.cimne.com/kratos*.

6.1.4 What makes Kratos useful?

- Kratos is MULTI-PHYSICS. One of the main topics in engineering nowadays is the combination of different analysis (thermal, fluid dynamic, structural) with optimising methods in one global software package with just one user interface and, even more, the possibility to extend the implemented solution to new problems.

- Kratos is FINITE ELEMENT METHOD (FEM) based. Many problems in engineering and applied science are governed by Partial Differential Equations (PDE), easily handled by computer thanks to numerical methods. The FEM is one of the most powerful, flexible and versatile existing methods.

- Kratos is OBJECT ORIENTED. An integration of disciplines, in the physical as well as in the mathematical sense, suggests the use of the modern object oriented philosophy from the computational point of view. The modular design, hierarchy and abstraction of these approaches fits to the generality, flexibility and reusability required for the current and future challenges in numerical methods.

- Kratos is OPEN SOURCE. The main code and program structure is available and aimed to grow with the need of any user willing to expand it. The GNU Lesser General Public License allows using and distributing the existing code without any restriction, but with the possibility to develop new parts of the code on an open or close basis depending on the developers.

- Kratos is FREE because is devoted mainly to developers, researchers and students and, therefore, is the most fruitful way to share knowledge and built a robust numerical methods laboratory adapted to their users' needs. Free because you have the freedom to modify and distribute the software. The one thing you're not able to do with free software is take away other people's freedom. Read the license for more detailed information in Kratos webpage.

6.1.5 Kratos structure

An object-oriented structure has been designed to maximize the reusability and extensibility of the code. This structure is based on finite element methodology and many objects are designed to represent the basic finite element concepts. In this way the structure becomes easily understandable for developers with a finite element method background. In this design, Vector, Matrix, and Quadrature represent the basic numerical concepts. Node, Element, Condition, and DoF are defined directly from finite element concepts. Model, Mesh, and Properties are from the practical methodology used in finite element modelling complemented by ModelPart, and Spatial Container, for organizing better all data necessary for analysis. IO, LinearSolver, Process, and Strategy represent the different steps of a finite element program flow. Finally Kernel and Application are defined for library management and its interface definition.

6.1.6 Basic tools

Different reusable tools have been implemented to help developers in writing their applications in Kratos. Several geometries and different quadrature methods are provided and their performances are optimized. Their flexible design and general interface make them suitable for use in different applications. Their optimized performance makes them appropriate not only for academic applications but also for real industrial simulations.

An extensible structure for linear solvers has been designed and different common solvers have been implemented. In this design the solver encapsulates only the solving algorithms and all operations over vectors and matrices are encapsulated in space classes. In this way solvers become independent of the type of mathematical containers and can be used to solve completely different types of equations systems like symmetric, skyline, etc. This structure also allows highly optimized solvers (for just one type of matrices or vectors) to be implemented without any problem.

6.1.7 Subversioning

Apache Subversion, SVN, is a software versioning and revision control system distributed under an open source license. Kratos main developers are attached into a subversion sharing network in order to up-load their developments and up-date the current and historical versions of files from the basic code or from new application parts being developed by others. This way, a new integration method, for instance, can be developed by anyone and included in Kratos database; after that, any other user or developer can update the modified parts of their code and they get instantaneously the integration method. In order to avoid conflicts between implementations from different people, there is also a benchmarking system checking for the correct functioning and compilation of any new implementation.

6.1.8 Benchmarking system

Every night, the cluster of CIMNE automatically updates Kratos using the versioning system, after that, it compiles everything and runs different preset cases for each application. These cases are tests that have been specially designed for each application. They consist on a simple application usage to calculate a predefined problem that has a predetermined known solution. If the cluster doesn't get the expected solutions when running the case or, moreover, if the cluster is not able to compile the code after a new contribution from a developer onto the versioning system, everyone gets a warning reporting the problem. If this is the case, the last uploads have to be revised for the good functioning of every application.

6.1.9 Advantages of DEM-Application belonging to Kratos

Eventhough Kratos is a FEM-based codes developing tool, many other applications fit perfectly in its structure and data base. The Discrete Element Method solves a physical problem, it does have elements which have nodal values and elemental properties; those nodes have boundary conditions, basic PDE have to be solved for the motion of the discrete elements, a search has to be done and the discrete particles are normaly involved in media that can be modelled by finite elements.

- The kernel of Kratos: All the database and the structure of the Kratos core allows the KDEM developer dedicate more time in the algorithms and the strategy for solving the problem with the method of Discrete Elements than spending time organizing the infrastructure for a DEM or FEM code, the management of data, the coding of the node, element, properties objects, the solvers, the search utilities, the numerical tools and libraries, the programing interface, the IO system etc. Everything comes with the package Kratos, ready for the engineers to develop their application and do their research.

- The graphical interface: The applications developed in Kratos automatically can take advantage of the Input and Output data system that allows using GiD as a preprocessor and postprocessor for the model desing and the results visualization.

- The python layer: Is the layer between the closed package ready to be use from a graphical interface and the core of the application which uses C++ language. Python reads the input data that GiD generates from the models designed by the users and does the callings to the functions, in the C++ code, that solve the problem. In this python layer most of the variables in the inner calculation are accessible and can be modifyied in time while the process is running and apply special conditions at a determined point of the calculation. This brings the application a versitile tool that doesn't need to be compiled which uses a programming language that can be used for any engineer whithout programming knowledge.

- The HPC tools: If your code belongs to Kratos, then you can easily parallelize your code with the OpenMP and MPI technology. Also Kratos gives you tools for debugging your code, checking errors, profiling, etc.

- The advantage of belonging to a comunity: The Kratos Community is pleased to incorporate more researchers, students and engineers in their group; belonging to a community makes the troubleshooting easier because many other users of Kratos may have had the same problems that a new user faces and offer their help. Also Kratos is a meeting point for knowledge sharing and cooperation that makes people learn from others and improve their applications.

- The force of teamwork: Thanks to the subversioning system and the benchmarking, more than one researcher can work developing the same application. Since the structure is well defined and the coding follows the Kratos template, there are no conflicts in developments that are added in an application by different people.

- Easy coupling: No doubt this is one of the principal advantages that Kratos brings to the KDEM-Application as well as to the rest of applications. Since every application developed in Kratos, CFD, CSM, Particle-based, etc. is set in the same framework and structure, it is fairly easy to couple different applications. In this sense, once the KDEM-Application was able to run on its own, it was inmediately combined with finite element applications for solids and fluids. So DEM-FEM applications emerge using the DEM and the FEM applications without rewriting those codes, simply creating a common strategy that does the combination and the correspondent interface.

6.2 Code implementation

In this section just a brief idea of how the code is organized is presented. For more details about the implentation, the author recommends the lecture of the previous thesis done from this application [28]. Obviously, since the code is free and open-source, if you are interested in taking a closer look on the files or directly testing the application, it is available to download in the Kratos repository: *https://kratos.cimne.upc.es/projects/kratos/repository*.

6.2.1 Basic computational sequence for a Discrete Element code

The first algorithm was proposed by Cundall [3] and it doesn't differ so much from the Kratos DEM-Applciation one . The code developed, as most of the commercial codes that use the Discrete Element Methods, has a basic sequence calculation that is roughly based in the same steps.

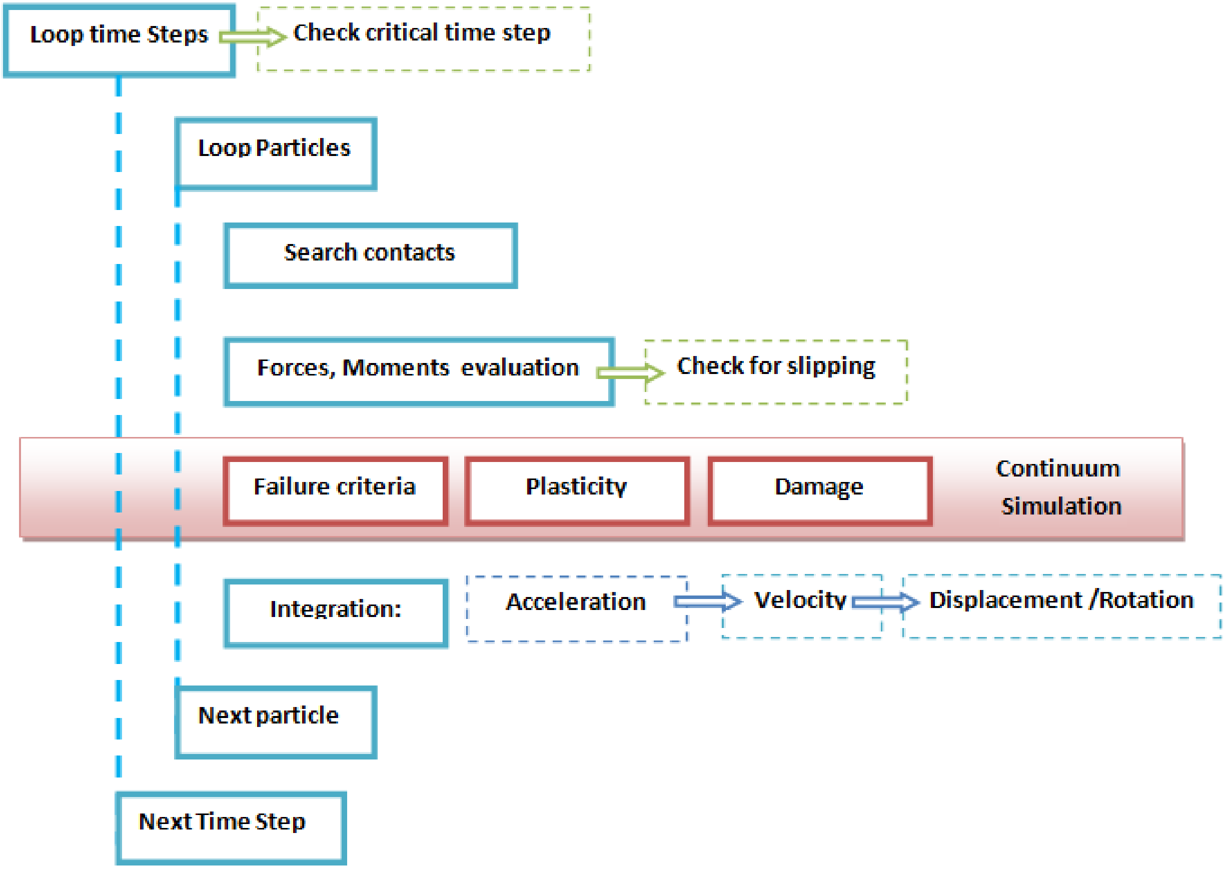


Figure 6.2: Basic sequence of the code.

6.2.2 Basic structure for the KDEM-Application

It has been commented in the section 6.1.4 that the platform is oriented to implement FEM-based applications; some of the applications that are already implemented are: Incompressible fluid App., Solid Mechanics App., PFEM App., Meshing App., ThermoMechanical App, KElectrostatic App., etc. However, a Discrete Element-based method has been also implemented, without problems, in the Kratos platform. The differences are minor comparing the structure of our application and the one from other applications because one of the principal recommendations when implementing in Kratos is trying to keep the same format in order to be more accessible to the other developers. The other fundamental reason is to be able then to couple easily two different applications. In that sense, a complete restructuration of the DEM-Application was made when the DEM-Application project started; In this sense with a little effort on matching the application to the dictated structure brings the great outcoume of a more versatile application.

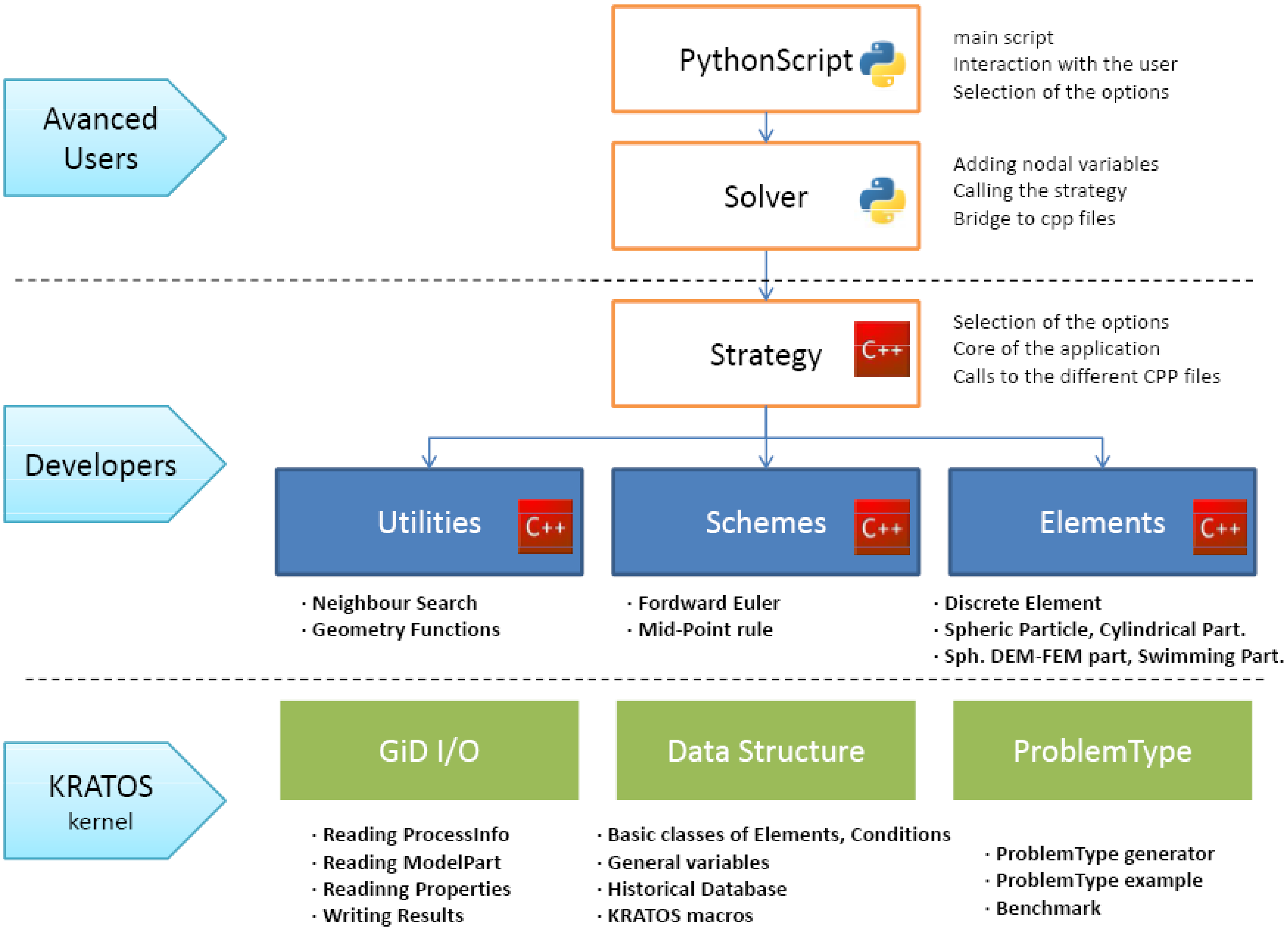


Figure 6.3: Basic structure of the application.

6.2.3 Main differences with respect to Dempack

In terms of the algorithm, the basic difference with respect to the previous code is that in KDEM the basic entity is the **particle** instead of the **contact** in Dempack. This introduces advantages and drawbacks.

On the plus side, looping over particle is more versatile and more generic. From the point of view of parallelization it is optimal, since the memory can be divided geometrically and the comunication minimized. On the other hand looping over the contacts is a clever strategy that minimizes the calculations in the basic DEM and specially in the continuum. However, it is not a general approach, for instance, when the particles are coupled with fluids where a singe particle has to calculate every time step the total forces that yield from its interaction with the fluid without existing any contact there.

One of the major drawbacks of looping over particles is that the calculations of the contacting forces are done twice in the algorithm, one from each particle forming part of the contact. Appart from that, in the continuum simulations, a new strategy has to be done which make the algorithm much slower. There are a lot of extra variables that have to be stored and a mapping procedure has to be done every time step that new neighbours are computed in order to identify which ones are old neighbours, which are initial ones or which are new ones. Dealing with contacts, the damage internal variables, the forces, the fracture flag and everything needed for the calculation can be stored as a member variable of the contact element; in the sphere approach every sphere has an array of values for each of these variables and the mapping has to be done to identify which new neighbour has which values from the previous or initial time step.

6.2.4 Utilities for the continuum - delta option and continuum option

Following the same previous idea, several special features have been developed to provide this general apprach the necessary tools to deal with the continuum:

- **Extended search**: Since the sphere meshers are not perfect and can produce some little voids between the spheres (appart from indentations) an extended search which takes a radius larger than the real one is used for every particle to find the contacting neighbours. The tolerance can be specified by the user.

- **Initial passive distance**: The initial gaps that these initial neighbours have, also the initial indentation in the inclusion case, is stored as a relative passive distance where the particles start to interact. Every particle stores an array of the lenght of its neighbouring particles with these initial distance values.

- **Cohesive groups**: The interface allows defining for each particle, or group of particles, a value for the variable *continuum group*. These groups are used to identify in the first time step which particles have cohesive bonds and which have the simple frictional contact. There is a group, with value 0, reserved for all the particles (that can be of different materials) that are non-cohesive. The rest of the groups make the particle be cohesive among the ones that share the same group. This permits simulate two cohesive entities of the same material that has no cohesion between the two bodies. For example the interaction betweeen two concrete blocks.

- **Historical variables**: The historical variables needed for the plasticity and damage are stored in an array of the neighbours size for evey particle also with the initial passive distance, the failure flag of the bonds, etc. Also the local forces of the last time step are stored since the constituitive law for the force evaluation is incremental. Every time that the neighbours are computed, a procedure is called which maps every neighbour found to the correspondent old neighbours and initial neighbour and transfers the needed values for the calculation. This is the most important issue in terms of (bad) performance with respect to the contact approach where all this procedure is not necessary.

Detailed information of how this works in terms of implementation of the code and explanatory examples can be found on the first thesis done by the author about the KDEM-Application [28].

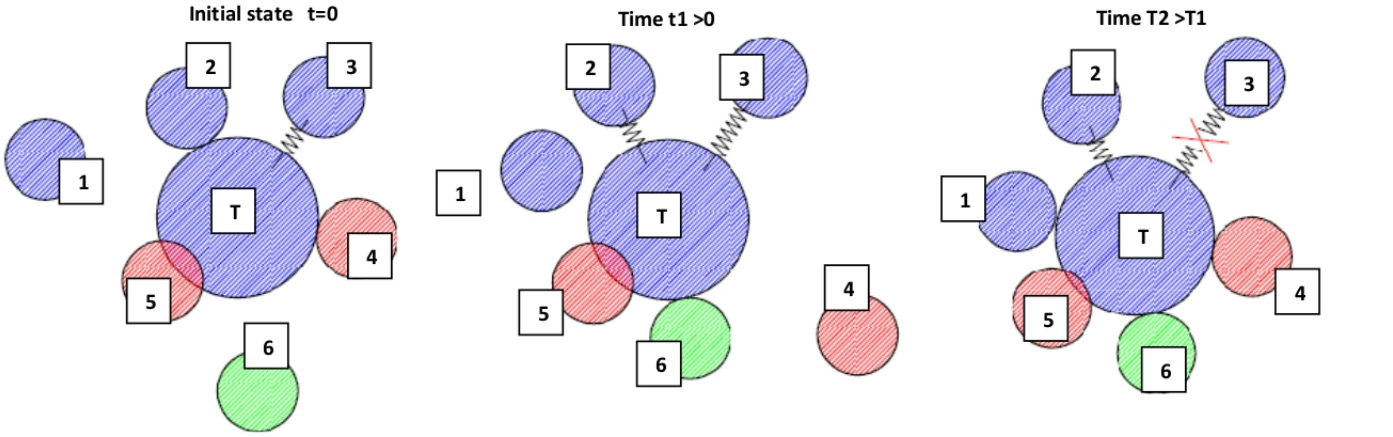
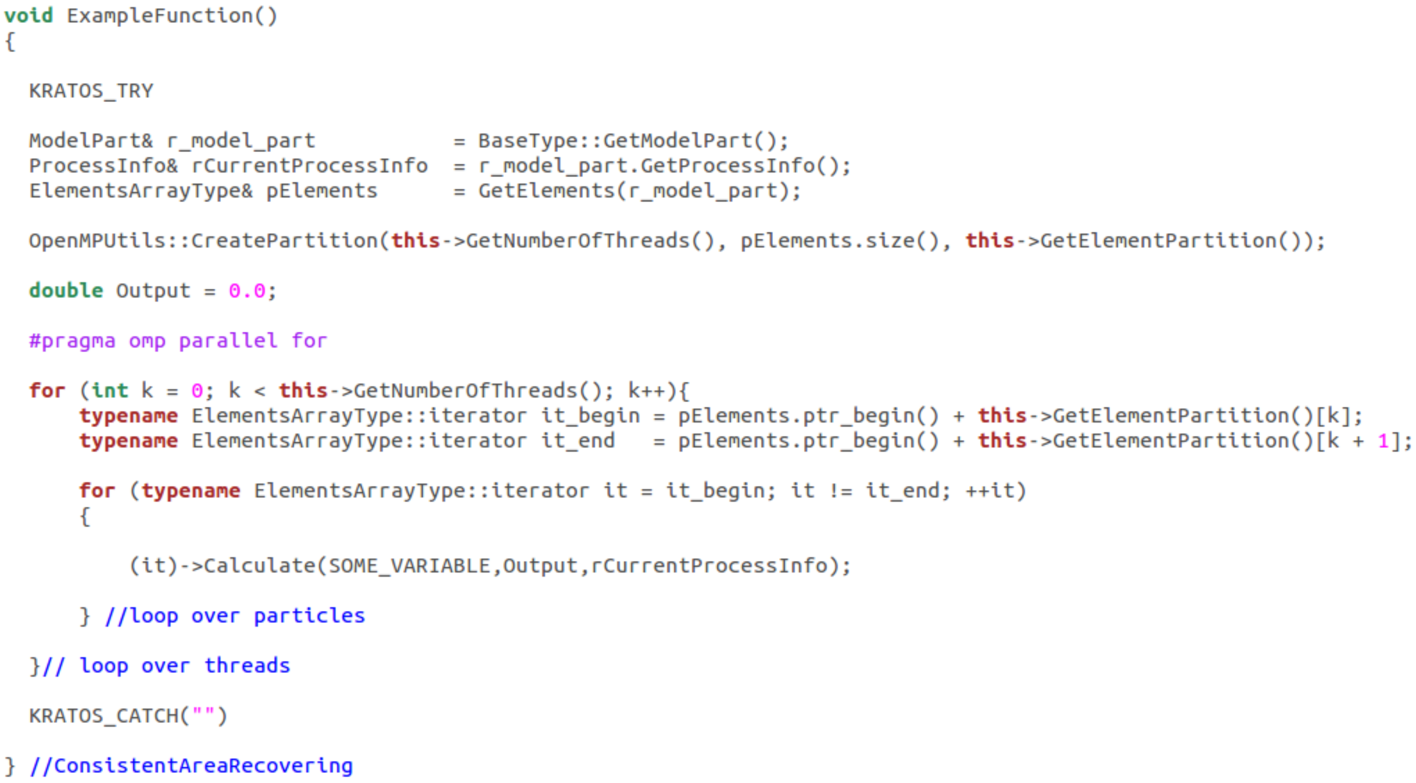


Figure 6.4: Different cases of neighbouring particles that KDEM deals with. *Souce: Miquel Santasusana [28].*

6.2.5 Parallelization of the code

A Discrete Element Method code without parallelization has a very limited use in practice; the reality is that for considerably large amount of particles (common simulations) the code needs to be parallelized to be competitive against other methods. The good thing of DEM is that the parallelization is quite easy to achieve; the method in its original conception is based on calculating each particle independently, i.e. from the forces that we obtain on a target particle, it evolves in an explicit time step scheme, independently from the other particles. In this sense, the main processes in the computational scheme: force calculation, evolve motion, search neighbours can be parallelized. There exist two types of remarkable architectures for cluster of computers, the Shared Memory Machines and the Distributed Memory Machines. In computer science, Distributed Memory refers to a multiple-processor computer system in which each processor has its own private memory. Computational tasks can only operate on local data, and if remote data is required, the computational task must communicate with one or more remote processors. In contrast, a Shared Memory multi processor offers a single memory space shared by all processors.

There are two widespread techniques of parallelization suitable for C++ language, OpenMP and MPI, which can be implemented in Kratos; OpenMP is already available for the code and MPI is currently being introduced to KDEM-Application. The suitable technique for SMM is Open MP (Open Multiprocessing); it permits parallelizing the loops of the process by using compilation directives so the code runs in serial until the is a parallelizable loop, runs the loop on parallel and then reverts back to serial. This can be done by splitting the loop and calculating each part by the different CPU of the same computer; OpenMP runs on a shared memory system so most part of the personal computers would permit parallelizing the calculation and saving time. OpenMP works fine if every unit step of the loop (normally a loop over the particles) is independent from the others so can be split without problems; the DEM permits doing so because a particle is independent from another in a explicit scheme. Next, an example of parallelization by OpenMP for the DEM-Application is presented. It is a partitioning of the loop over the particles for the different threads of the computer.



```
void ExampleFunction()
{

  KRATOS_TRY

  ModelPart& r_model_part            = BaseType::GetModelPart();
  ProcessInfo& rCurrentProcessInfo   = r_model_part.GetProcessInfo();
  ElementsArrayType& pElements       = GetElements(r_model_part);

  OpenMPUtils::CreatePartition(this->GetNumberOfThreads(), pElements.size(), this->GetElementPartition());

  double Output = 0.0;

  #pragma omp parallel for

  for (int k = 0; k < this->GetNumberOfThreads(); k++){
      typename ElementsArrayType::iterator it_begin = pElements.ptr_begin() + this->GetElementPartition()[k];
      typename ElementsArrayType::iterator it_end   = pElements.ptr_begin() + this->GetElementPartition()[k + 1];

      for (typename ElementsArrayType::iterator it = it_begin; it != it_end; ++it)
      {

          (it)->Calculate(SOME_VARIABLE,Output,rCurrentProcessInfo);

      } //loop over particles

  }// loop over threads

  KRATOS_CATCH("")

} //ConsistentAreaRecovering
```

Figure 6.5: Example of typical particle loop parallelized in the KDEM code.

For DMM architecture the suitable technology is the MPI (Message Passing Interface); this would permit running a case, usually with large number of particles in a computer cluster where hundreds, thousands or more CPUs intervene in the calculation. With MPI the entire code is launched on each node which would store the data in its own memory. The transfer of information and the synchronization of the calculation can be controlled. It is also possible to combine MPI with OpenMP to get the best of every technology and adapt to the specific architecture of each cluster.

Figure 6.6: Cluster of Distributed Memory Machines. *Source: Google Images.*

Speed-up chart in KDEM using OMP

In this section a speed-up test has been done with OMP parallelitzation of the code for a continuum case. The example corresponds to a 13000 elements UCS test on concrete material. For this analysis only 2000 time steps were performed and noted down the times obtained with 1, 2, 4, 8, 12 and 16 processors in a SMM cluster in CIMNE.

Number of proc.	1	2	4	8	12	16
Time (s)	$325,56$	$187,59$	$105,67$	$51,45$	$45,38$	$38,03$
Real Speed-up	1	$1,73$	$3,08$	$5,30$	$7,17$	$8,56$
Ideal Speed-up	$1,00$	$2,00$	$4,00$	$8,00$	$12,00$	$16,00$

Table 6.1: Times for the analysis in different number of processors

The results are encouraging since the times keep scaling up even for 16 cores. Also the scalability has a good ratio still for 8 or 12 processors with more than 5 and 7 times respectivelly of speed-up. It has to be noted that with the future implementation of the MPI parallelitzation, a combination of both technologies that can be adapted to every cluster technollogy would yield to a high performance in the code. The following plots show the up-scaling of the time versus the number of processors (Figure 6.7) and the performance compared against the ideal scaling curve (Figure 6.8).

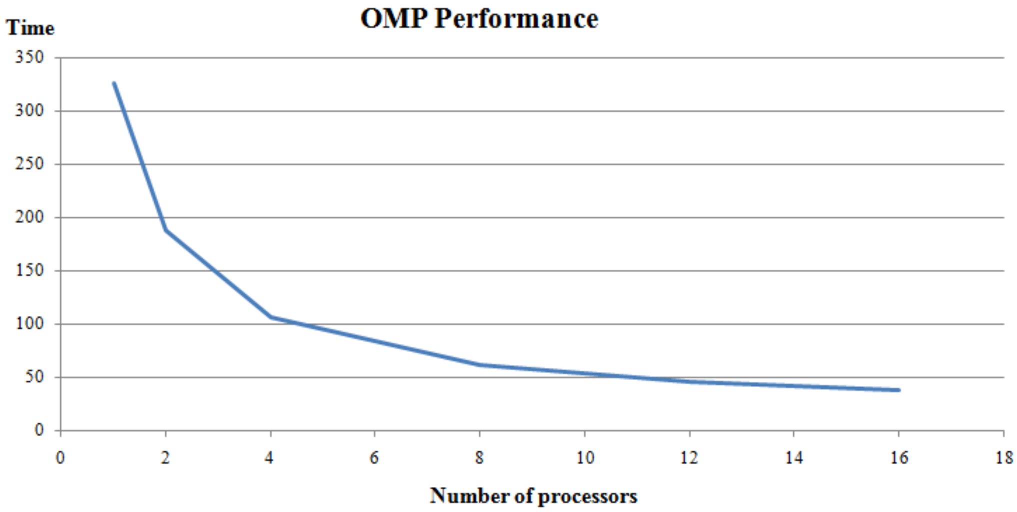


Figure 6.7: Simulation time for different number of processors in a SMM system with OMP parallelization.

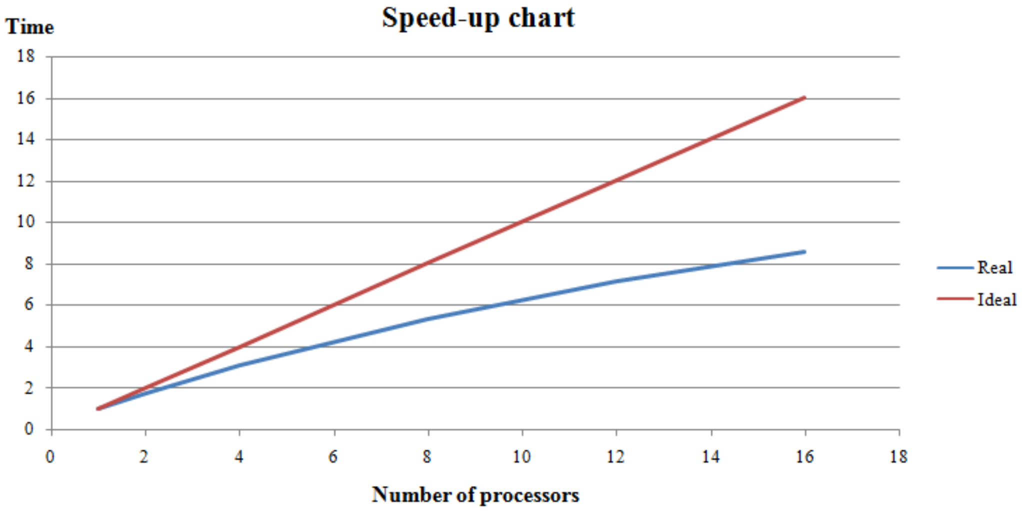


Figure 6.8: Speed-up chart, Ideal times against real times.

6.3 Current developments of DEM in Kratos

It is an honour for the author of the thesis, who started the project of coding a Discrete Element Mehtod in Kratos with the collaboration of researchers Guillermo Casas and Miguel Ángel Celigueta, to see that currently a lot of people joined the *DEM team* in CIMNE and they are using/developing the KDEM application.

Once a first version of the code with the basic features of the DEM (discontinuous) was finished, the interest from many researchers in CIMNE to use this application for specific simulations or coupling with other methods emerged. The KDEM then followed being developed by the author in the field of the continuum media simulation at the same time as the basic DEM module has been used in the following fields:

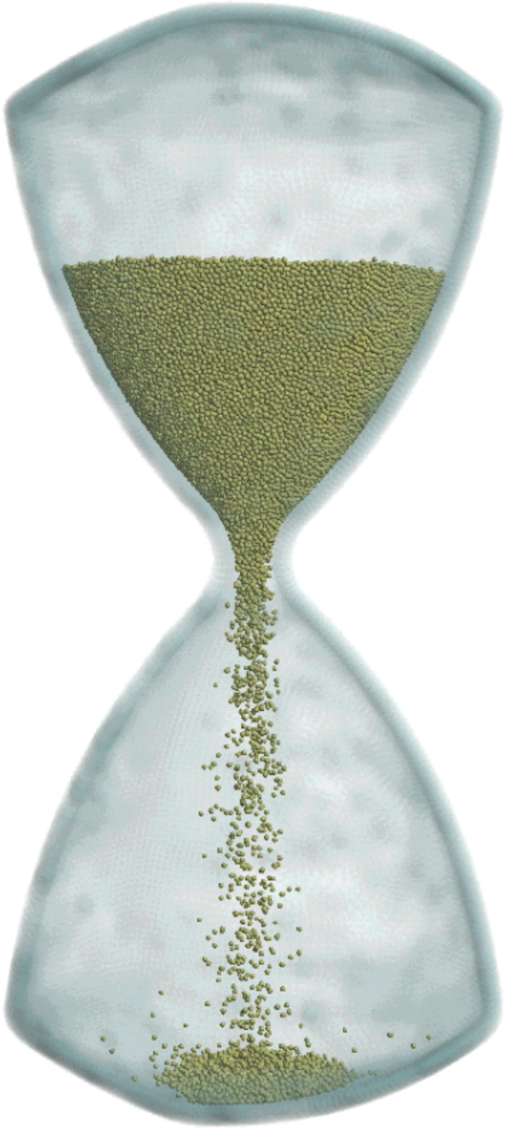


Figure 6.9: Sandclock simulation using KDEM.

Granular material simulation

Authors: Joaquín Irázabal, Miquel Santasusana, M. Ángel Celigueta, Ignasi De Pouplana.

Joaquín Irázabal is an engineer in CIMNE Madrid who is developing utilities and constituitive models for the discrete version of the KDEM for the granular material simulations. This work extends the first stage of development that the initial DEM Team in CIMNE Barcelona had done.

One of the developments that has been done is the visualization of rotation axis in the sphere particles which includes integration of rotations by using quaternions:

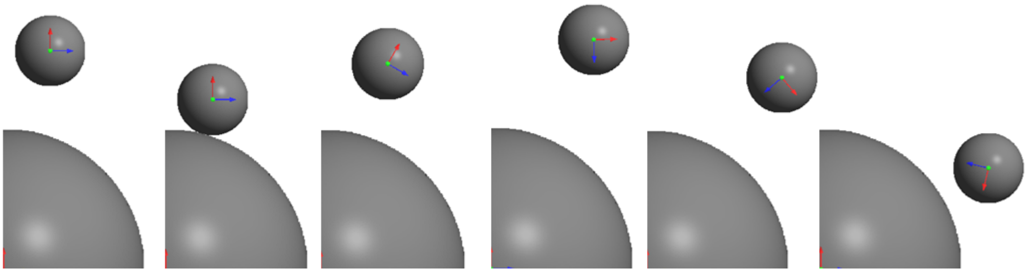


Figure 6.10: Local reference axes tracking the particle rotation in 3D.

Several test have been performed to validate the implemented frictional contact and the implementation of an additional rolling friccion:

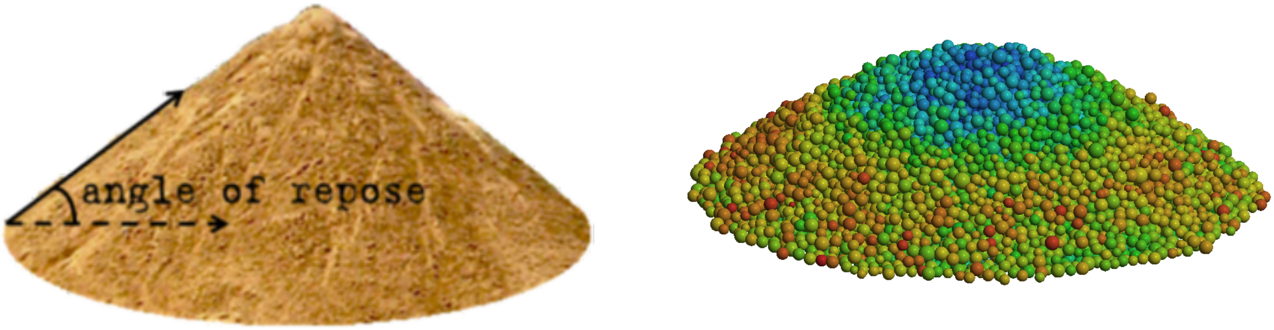


Figure 6.11: Angle of reponse of a sand and simulation with KDEM. *Source left image: Wikipedia*

Currently CIMNE madrid is working in a project with the objective of simulating the behaviour of the ballast under coditions which it is subjected to in the highspeed railways:

Figure 6.12: Ballast in a railway. *Source: Google Images.*

High Performance Computing

Authors: Carlos Roig, Pooyan Dadvand.

As it has been commented the code is already completelly parallelized in OpenMP. A first version of MPI parallelization for the basic discontinuous DEM was acquired and the results were promising. Some computer scientists in CIMNE are currently developing the MPI for the continuum version.

The MPI implementation includes not only the comunication between nodes but also the rebalancing of particles in the nodes for improving the efficiency as it is shown in Figure 6.13. Each colour represents a different processor of the DMM system. The colour in each particle indicates in which processor it is being handled; it can be seen that the particles evolve from one processor to another one while the simulation evolves in order to minimize the communication between processors.

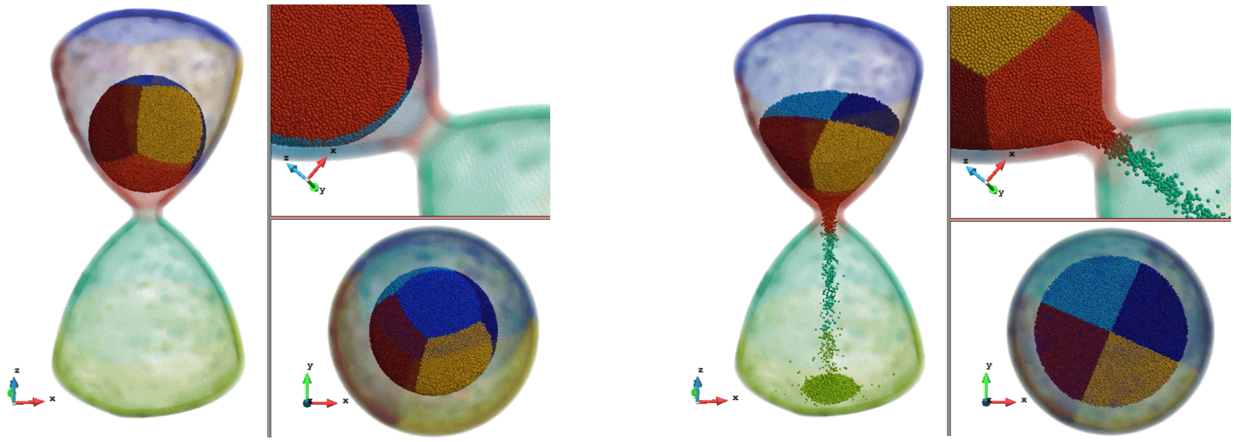


Figure 6.13: Particles in different processors in two different time steps of the sandclock simulation.

Explosion

Authors: Cristobal Garcia, Pooyan Dadvand, José Manuel González.

A basic module of explosions has been coded using KDEM too. The interface of the code, thanks to the Kratos structutre, makes the application so flexible that this simulations and many others can arrise from the KDEM code.

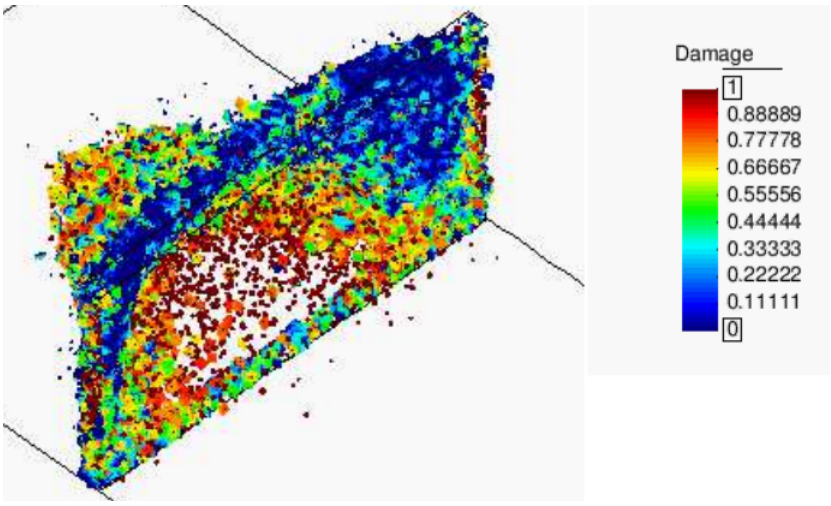


Figure 6.14: Simulation of explosion of a wall using KDEM.

DEM-FEM Coupling

Authors: Feng Chun, Miquel Santasusana.

This is a very important feature of the code, Kratos brings our code the possibility to combine different applications that are developed in the Kratos framework. This is the case of coupling the Discrete Element Method with Finite Element Methods. In the examples that follow, a trianglular finite element mesh discretization out of any geometry interact with the discrete elements. The search algorithm has been extended in order to check contacts between triangle elements and spheric elements additionally to the sphere to sphere contact. Once these contacts are detected, by some contact force characterization, forces from the bounding triangular elements are returned back to the particles and added to their global force vector.

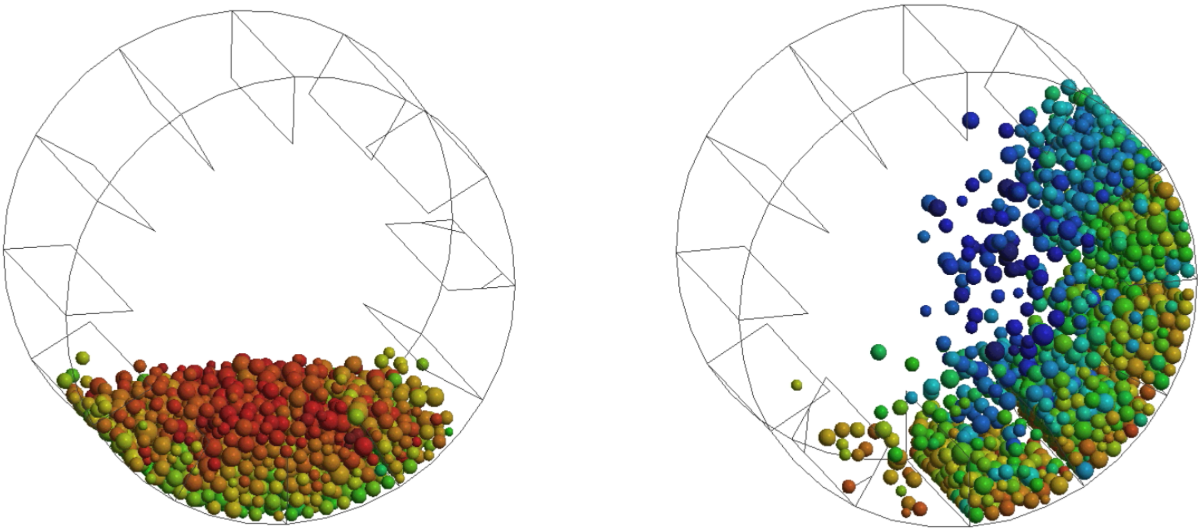


Figure 6.15: Particles interacting in a rotating drum. Simulation using KDEM.

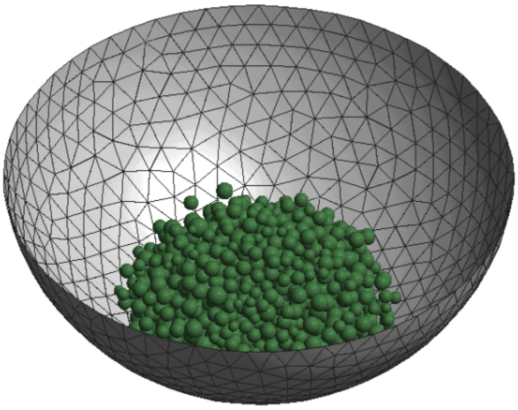


Figure 6.16: Contact detection between particles and any geometry meshed with triangular elements.

The next step in which we are working (currently yielding the first results) is to use those determined contact forces and transmit back them to the triangular elements as well, interpolate and weight the forces on the nodes of the finite elements and let the *Solid-Mechanics-Application* in Kratos to solve the problem in the finite element mesh.

DEM-Fluid Coupling

Authors: Guillermo Casas, M. Ángel Celigueta, Salvador Latorre.

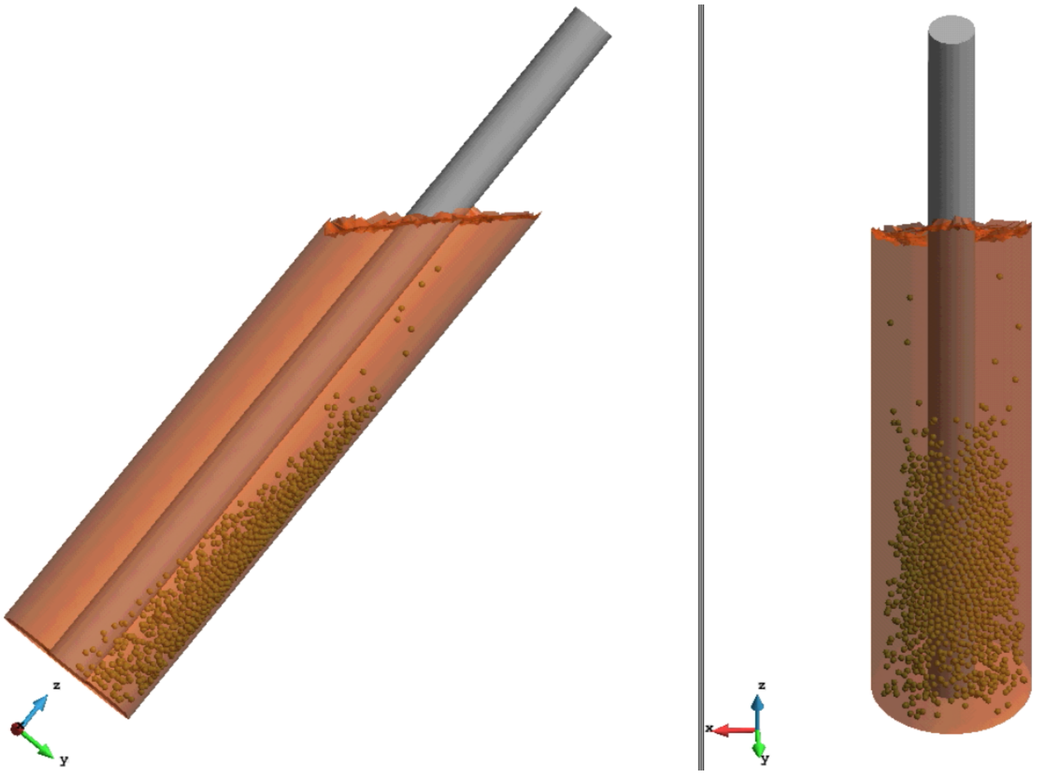


Figure 6.17: DEM particles in a fluid flow inside a cylinder.

A very interesting application of the DEM is its coupling with a CFD application. Some researchers in CIMNE have coupled the KDEM with an Eulerian CFD application in Kratos. The interaction is done projecting the velocities form the velocity field in the fluid to the particles which give back *drag forces* and modify the porosity of the media where the fluid flows depending on the solid fraction (density of particles in the fluid). In order to calculate the *buoyancy force* on the particles, the pressure gradient is projected to the particles.

6.4 Graphic Interface

6.4.1 Gid Pre and Post Processor

GiD is a versatile multipurpose software that provides a graphical support to the pre-process and the post-process stage. Kratos applications are highly compatible with graphical interfaces. The one that CIMNE has been using for Kratos and many other ProblemTypes is GiD interface which has its origins on CIMNE itself (Trial and professional versions can be found in *http://gid.cimne.upc.es/*). In parallel, a new specific graphical interface is being developed for the Kratos package in order to improve the problem definition of the different applications that Kratos supports. In the present section the GiD interface will be introduced and in next section 6.5 the current interface for the KDEM-Application will be presented together with a step-by-step example of how to design a Concrete Test with the application.

Pre-Process

This stage consists on setting the geometry and the data for the problem definition (forces, movements, properties...) as well as imposing boundary conditions and setting the calculation options. the problem definition GiD also dispose of different mesh generators for the FEM and DEM calculation.

- Geometry: GiD Pre Process is a CAD system that assists the geometry definition of the engineering models. Typically geometrical operations can be used as transformations (translations, rotations, etc.), Boolean operations in surfaces and volumes. A complete set of tools are provided for quick geometry definition.

- Mesh: GiD has a large variety of options for the generation of finite element meshes from the defined geometry. GiD has already a sphere mesher that was developed by C. Labra and others [15] and it is used for the DEM simulations of the KDEM. Another special mesher has been developed by the author of this thesis, which generates regular assemblies of spheres from points, lines, surfaces and volumes specially designed for testing the application and permits the coupling of finite elements and discrete elements.

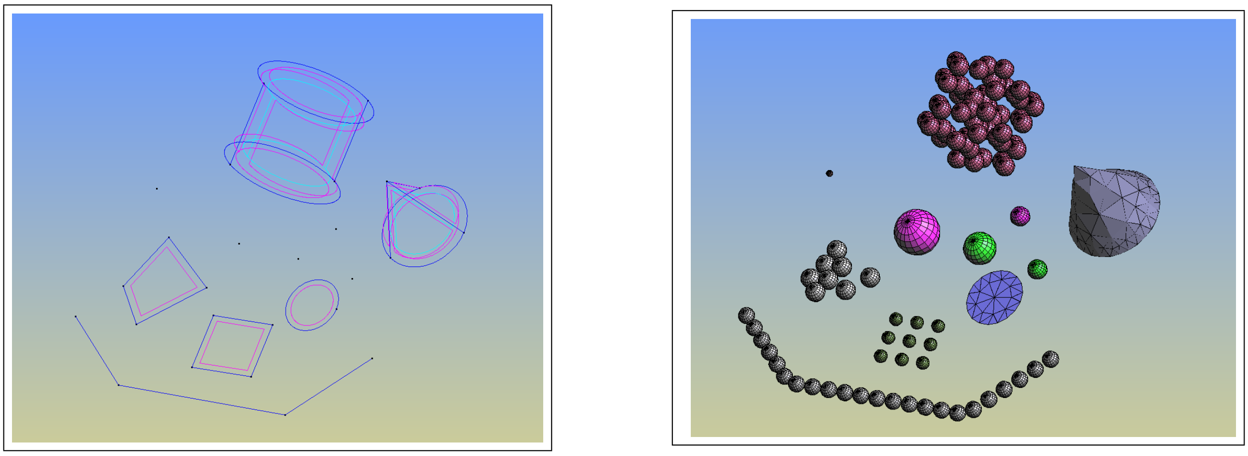


Figure 6.18: Regular and irregular meshes with finite and discrete elements obtained with the developed mesher from different geometrical entities.

- The user can load to GiD a specific ProblemType. When a ProblemType is loaded, the specific options regarding the particular application appear, such as conditions or calculation parameters.

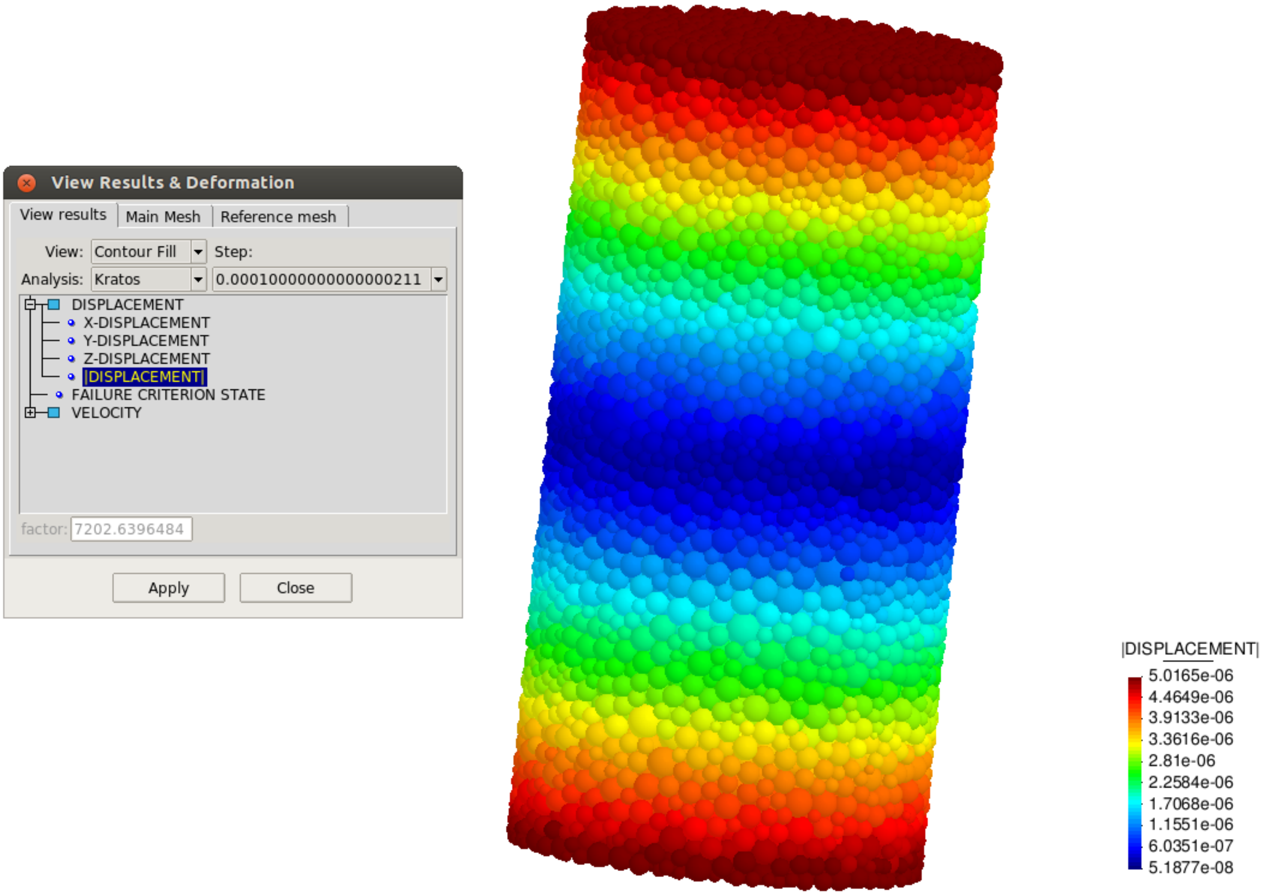
6.4.2 Calculation Process

Once the geometry is drawn and the conditions, loads and parameters that each ProblemType and requires are defined, the calculation shall be done. When the Calculate button of GiD is used, the ProblemType reads the geometry, applies the conditions and so one and calculates the problem. It has to be remarked that GiD doesn't calculate, it triggers the ProblemType inner calculation, in this case, by Kratos.

6.4.3 Post-Process

GiD goes from the Pre to the Post with a simple button click. The options that the user has in each part of the program are different; In the Post there are numerous utilities aiming analyse the results by means of plotting variables or creating graphs for instance.

In the View Results tab many types of visualization are available to represent the accessible output data. The View Results & Deformation window permits to choose the representation of the results either on the original mesh or on the deformed one, selecting a suitable scale. On the View results is possible to select first of all a type of representation; the available representations are Display Vectors, Contour Fills, Contour Lines, etc depending on which one the user considers that is the best for every different type of result.



View Results & Deformation

View: Contour Fill

Step: 0.00010000000000000211

Analysis: Kratos

|DISPLACEMENT|

factor: 7202.6396484

Figure 6.19: Displacement field in a cylindrical specimen during a UCS simulation.

For more information about the specific developments done in the pre and post processor of GiD for the KDEM-Application, refer to [28].

6.5 Example of concrete test design - Step by step with the KDEM problemtype for GiD

In this section as an example of application of the developed software, a triaxial concrete test will be set using GiD and the KDEM problemtype. The model will be prepared so that the KDEM-Application will be able to calculate it. This is the methodology used when the tests included in this thesis were performed.

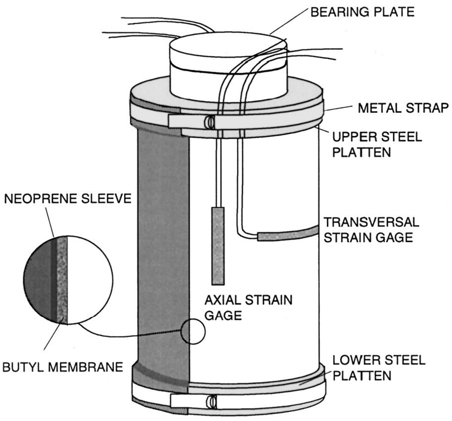
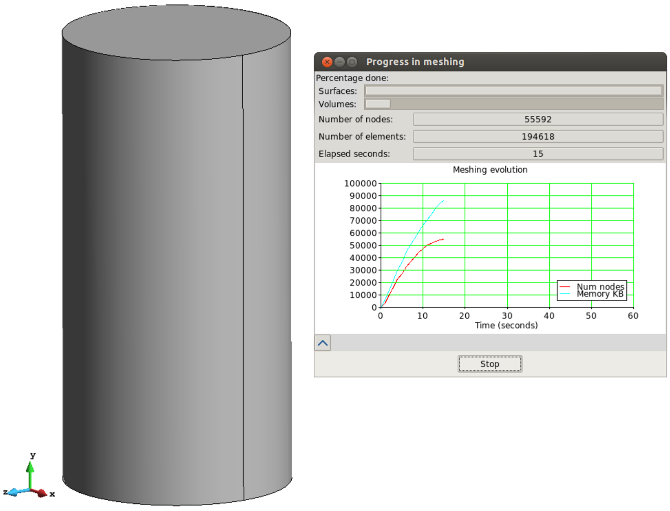


Figure 6.20: Specimen prepared for triaxial testing of concrete [29]

1. Geometry definition and meshing

First step is to open GiD and load the KDEM problemtype. Then the geometry of the model can be drawn. The current tests were performed in a cylindrical speciments of 15cm of diameter and 30cm height that have to be placed centering its bottom face on the coordinates origin.


Progress in meshing

Percentage done:

Surfaces:

Volumes:

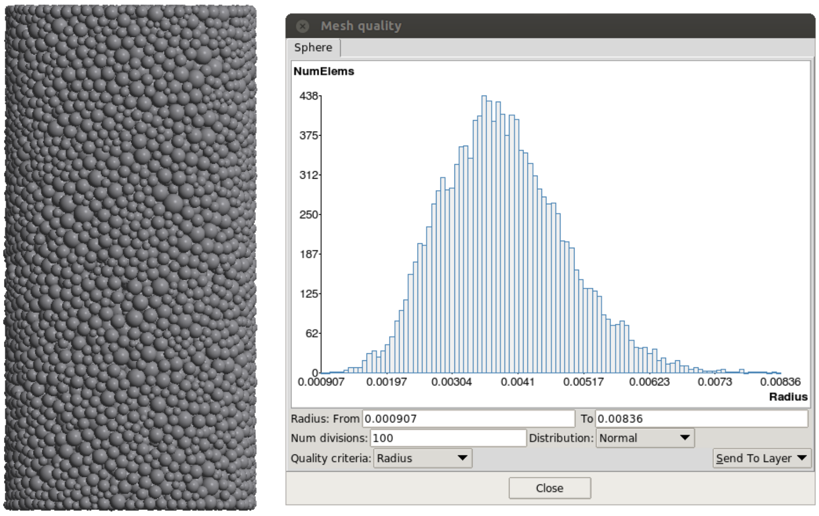
Number of nodes: 55592

Number of elements: 194618

Elapsed seconds: 15

Figure 6.21: Cylinder drawn by gid being meshed with the sphere mesher.

In the general menu of GiD, set *sphere* to the *mesh element type* on the meshing tab and assign it to the cylindrical geometry. The sphere mesher of GiD has a lot of options that allows obtaining more or less regular meshes, obviously several attepts are usually done before obtaining a good mesh which can be taken as the one for the simulation. The sphere meshed has a nice tool to asses the goodness of the meshed models: *mesh quality*.

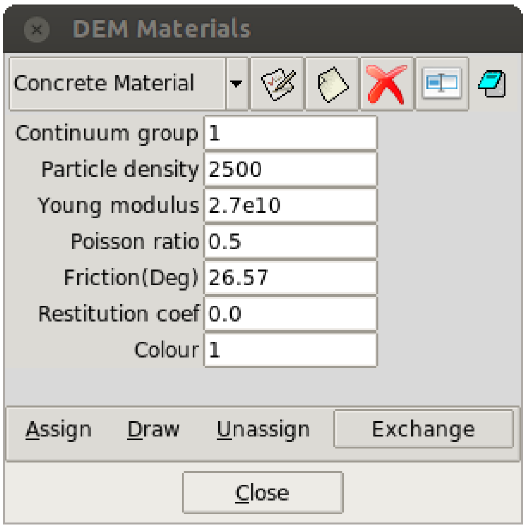


Radius: From 0.000907 To 0.00836
Num divisions: 100 Distribution: Normal
Quality criteria: Radius

Figure 6.22: Distribuition of the radius in the generated mesh using the *mesh quality* tool.

2. Material properties definition

The KDEM application allows modelling in the same simulation different types of cohesive and particulate materials at the same time. With the following menu the principal properties are defined and later assigned to the entities:



Concrete Material

Continuum group 1
Particle density 2500
Young modulus 2.7e10
Poisson ratio 0.5
Friction(Deg) 26.57
Restitution coef 0.0
Colour 1

Figure 6.23: Material menu.

- Continuum group: is an identifier that determines in which cohesive entity the assigned elements belong. Group 0 is reserved for non-cohesive material. This tool allows having elements with different material properties in the same cohesive group, i.e. the same solid. On the other hand it allows having two disconected parts that have the same material properties but aren't cohesive between them even if they are in contact because belong to different bodies.

- Particle density, young modulus and poisson ratio: are basic properties of elastic materials.

- Friction: this value expressed in degrees corresponds to the dynamic friction coefficient of the material. This is the friction that the non-cohesive materials present or, in the continuum case, the one that the particles have once their bonds are broken.

- Restitution coefficient: this value determines the amount of visco-damping applied at the contact level. Review section 3.3.2 for the detailed definition of this parameter.

- Colour: modifies the colour of the default visualization for every material in the post-process.

3. Groups and conditions assignment

In the triaxial test, a uniaxial compression is applied to the speciment after the confining stage (see the description f the experimental tests in Section 7.2.2); this uniaxial compression is numerically simulated by imposing strain on the top and bot lid of the speciment, i.e. fixing the velocity degree of freedom of the particles generated in the top and in the bottom of the model. From the point of view of the stability and the monitoring of the yielding stress-strain curves this is the strategy followed.

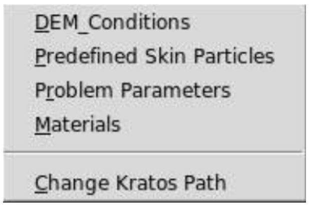
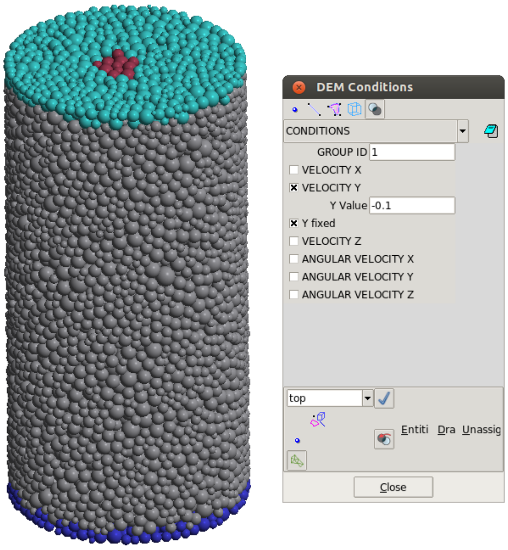


Figure 6.24: KDEM problemtype menu included in GiD.

In the loaded problemtype menu, shown in Figure 6.24, select the *DEM conditons tab*. This window allows the user assigning conditions of velocity to the desired particles as well as a *Group Id* condition.


DEM Conditions

CONDITIONS

GROUP ID 1

□ VELOCITY X

☒ VELOCITY Y

Y Value -0.1

☒ Y fixed

□ VELOCITY Z

□ ANGULAR VELOCITY X

□ ANGULAR VELOCITY Y

□ ANGULAR VELOCITY Z

top

Figure 6.25: Conditions window and view of the different layers in the model.

Prior to the assignation of the conditions, is recommendable to create groups with the particles where the different conditions will be assigned. The first thing to be done is to create 4 new layers, the top particles have to be included in one top layer that will form the *Top* group and the same for the bottom particles forming the *Bottom* group. Note that it is not necessary to restrict this layer to a thickness of just one spheric particle; the mesh is irregular and it is assumed that in some parts the top and bottom group will contain one, two or even three particles. There is a special treatment for this groups in the code that takes this fact into account. As it can be observed in Figure 6.25 two more groups created from the layers are present (only one can be seen in the figure); these are the hereafter called *Top-center* and *Bottom-center* groups. It is **indispensable** for these groups not to share repeated elements. Each element can belong only to a group[1].

[1] This is how the current version of KDEM works, a new problemtype included in the Kratos generic problemtype for GiD is being developed for the KDEM application and this process will be easier. There is the idea of creating a special wizard for the step-by-step procedure of concrete test simulations for professional applications.

The following assignations have to be done to the respective groups:

- *Top group*: $VELOCITY\_Y$ fixed to a negative value. In the example shown in Figure 6.25, a vertical negative velocity of $0.1(m/s)$ is imposed. The marking on *Y_fixed* determines that the degree of fredoom is fixed; otherwise it would be only a initial condition on the velocity. Also the top particles are marked with the identifier *GROUP_ID* set to the value of 1.

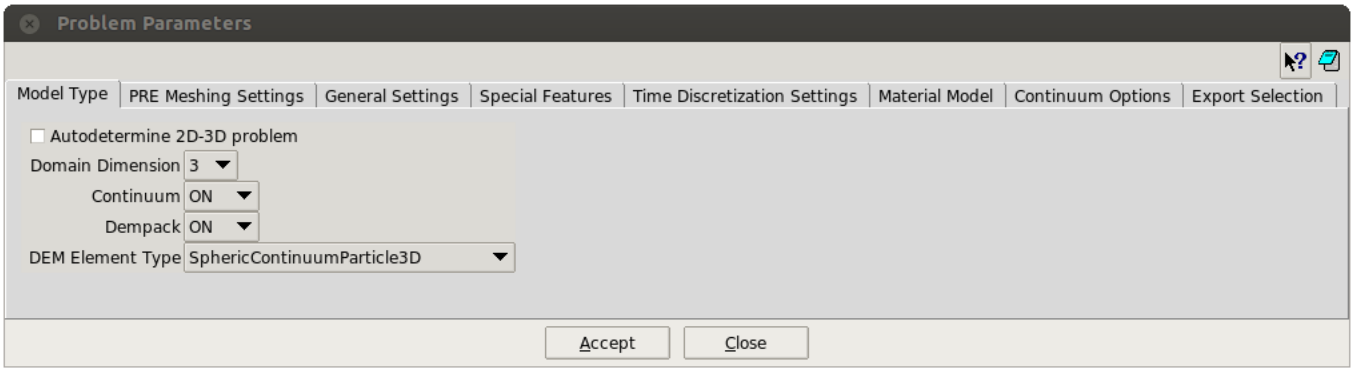
- *Bottom group*: $VELOCITY\_Y$ fixed to a positive value. Imposing velocity in opposite directions in both top and bottom makes the elastic waves that the speciment undergoes for the dynamic loading stabilize twice fast as if only the velocity is imposed on the top and the bottom is fixed to zero velocity. This group is identified with the variable *GROUP_ID* set to the value of 2.

- *Top-center group*: These group have the same assignations as the *Top* group, the velocity in Y direction and the group determined to the value 1. In this group additionally the $VELOCITY\_X$ and $VELOCITY\_Z$ are fixed to zero to avoid that the whole model could fly away in the plane normal to the compression. Eventhough this is a simulation with no forces or velocities imposed in the XZ plane, since this is a numerical simulation with a irregular mesh, it is possible that some unbalanced internal forces apperar in the plane due to numerical errors and this is a way to control it.

- *Bottom-center group*: The same has to be done but now with a positive velocity in Y direction and a 2 as the value of the variable *GROUP_ID*.

The *GROUP_ID* is a identificator for the KDEM applictation to apply special conditions on the elements inside the calculations of the code; in this case, a value of 1 determines which are the particles where the forces for the strain-stress plot are being maesured. It should be marked on the *Top* and *Top-center* particles. However, if the identifyier is placed on the *Bottom* and *Bottom-center* group particles, roughly the same plot is obtained but in negative values. In conclusion, the particles that measure the stress are the ones whose velocity is fixed and so all the force that those particles recieve is due to the reaction of the speciment against the strain that the top and bot lids impose.

4. Problem parameters definition



□ Autodetermine 2D-3D problem
Domain Dimension 3
Continuum ON
Dempack ON
DEM Element Type SphericContinuumParticle3D

Figure 6.26: Problem parameters main menu.

Now it is time to define the parametes of the problem by using the next window on the problemtype menu, the *Problem Parameters*. Next a brief description of the opcions available in each tab of this menu is presented. Please refer to [28] for a much more detailed information about the problemtype options and utilities coded for the application:

ModelType:

- Domain dimension selection, automatic or manually specified.
- Continuum simulation ON or OFF. It activates additional options in the present tabs and a new tab special for the continuum.
- Dempack model option. Explained in Section 4.4, it has been enabled for the simulations present in this thesis.
- Element selection, continuum or basic spheric and cylindrical elements.

PRE Meshing Settings:

- Automatic or predefined skin determination: In the case of concrete testing the automatic skin determination has to be selected. The need of skin determination has two purposes; on one hand the application of the confining pressure in the triaxial test simulations, on the other hand the special treatment for the area calculation. Review Section 4.6.2.
- Tool for cleaning indentations produced by the sphere mesher. In the continuum case, this tool is not used because here the indentations and the gaps will be allowed until a certain tolerance as passive distances.

General Settings:

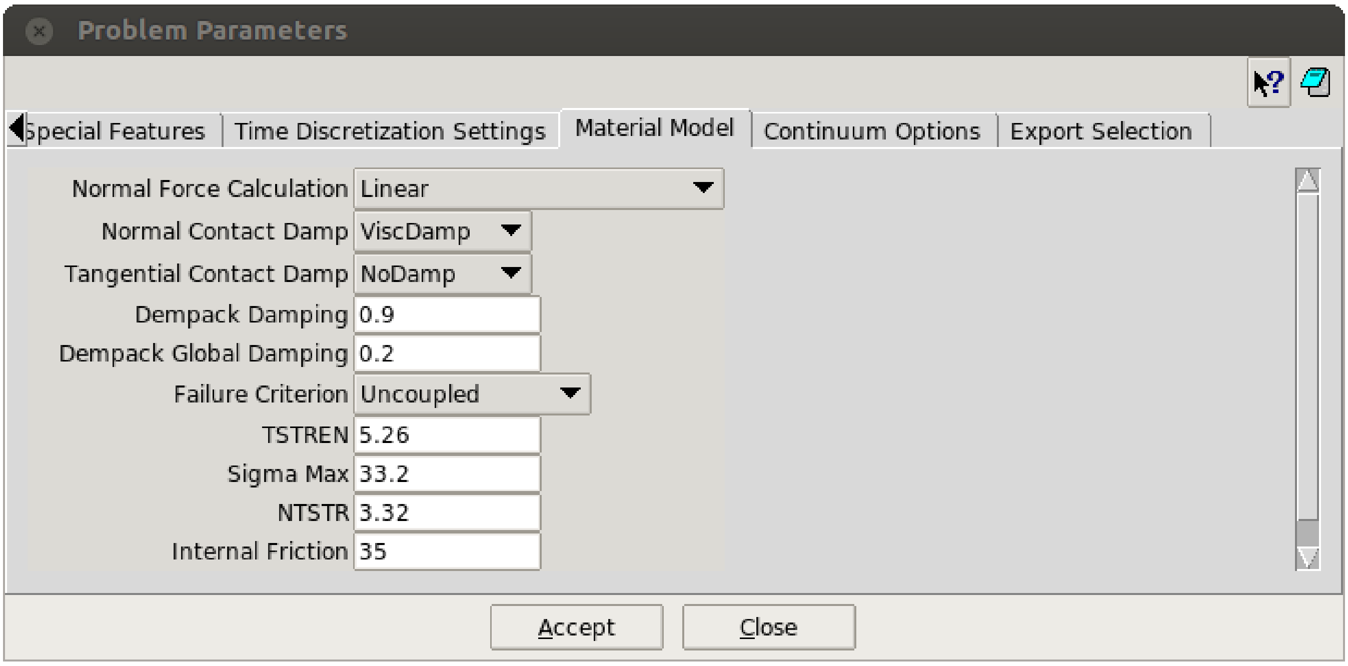
- Number of processors for OpenMP parallel computing.
- Bounding box, automatically created or manually. Destroys particles that leave the initial domain.
- GiD postprocess output in binary or ASCII.
- Rotation Option, activated for realistic applications.
- Gravity
- Other features

Special Features:

- Virtual Mass option. This may help increasing the timestep value for quasi-static problems. In can be used in the concrete test simulations.
- Calibration factors α and β for the contact stiffness. Not used in the concrete test simulations performed in this thesis since the *Dempack-model* is used. The calibration factors were used for the numerical analysis (Section 5).
- Delta Option. This option is always enabled in continuum simulations, the tolerance represents an extension of the radius (in percentage) where the indentations and the gaps are set as the passive initial distances between contacting continuum particles. In the present concrete test simulations this value is set to 0.15.
- Globally specified variables option. This option allows defining the micro parameters of the model from a global approach. See section 5.1.1. 5.1.1.

Time Settings:

- Calculation Time. Simulation times of $1.0e^{-3}$ are normally enough for the results with the concrete material for the imposed total $0.2m/s$ velocity.
- Time step. $1.0e^{-7}$ is the order of times which will be necessary in this concrete test simulations.
- Output time step. When the results are stored for the post process.
- Time Steps Per Search Step. A value of 10 is sufficiently accurate since the time step is so small.
- Automatic Time Step Reduction, is based on the critical time step calculation discussed on Section 3.4.2.



Normal Force Calculation: Linear
Normal Contact Damp: ViscDamp
Tangential Contact Damp: NoDamp
Dempack Damping: 0.9
Dempack Global Damping: 0.2
Failure Criterion: Uncoupled
TSTREN: 5.26
Sigma Max: 33.2
NTSTR: 3.32
Internal Friction: 35

Figure 6.27: Problem parameters main menu.

Material Model:

- Normal Force Calculation. Here the constituitive modelling of the contact is selected. In the concrete tests simulated
- Selection of the type of damping on the contacts and the globally applied dampings. In the case of the tests the Dempack damping is used in the contact level with a value of 0.9 (almost the critical one) and the also the global damping for quasi-static problems set to 0.2.
- Failure Criterion, coupled and uncoupled Mohr-Coulomb are available. The uncoupled one is the used for the concrete tests.

Continuum Options:

- Amplified Continuum Search Radius Extension. This value is an extension to the intial search radius that allows the particles find neighbours outside their contacting range (this happens when a contacting pair is under tensional strain).
- Concrete Test Option. This option incorporates to the calculation all the necessary features to take into account that a concrete test is being performed.

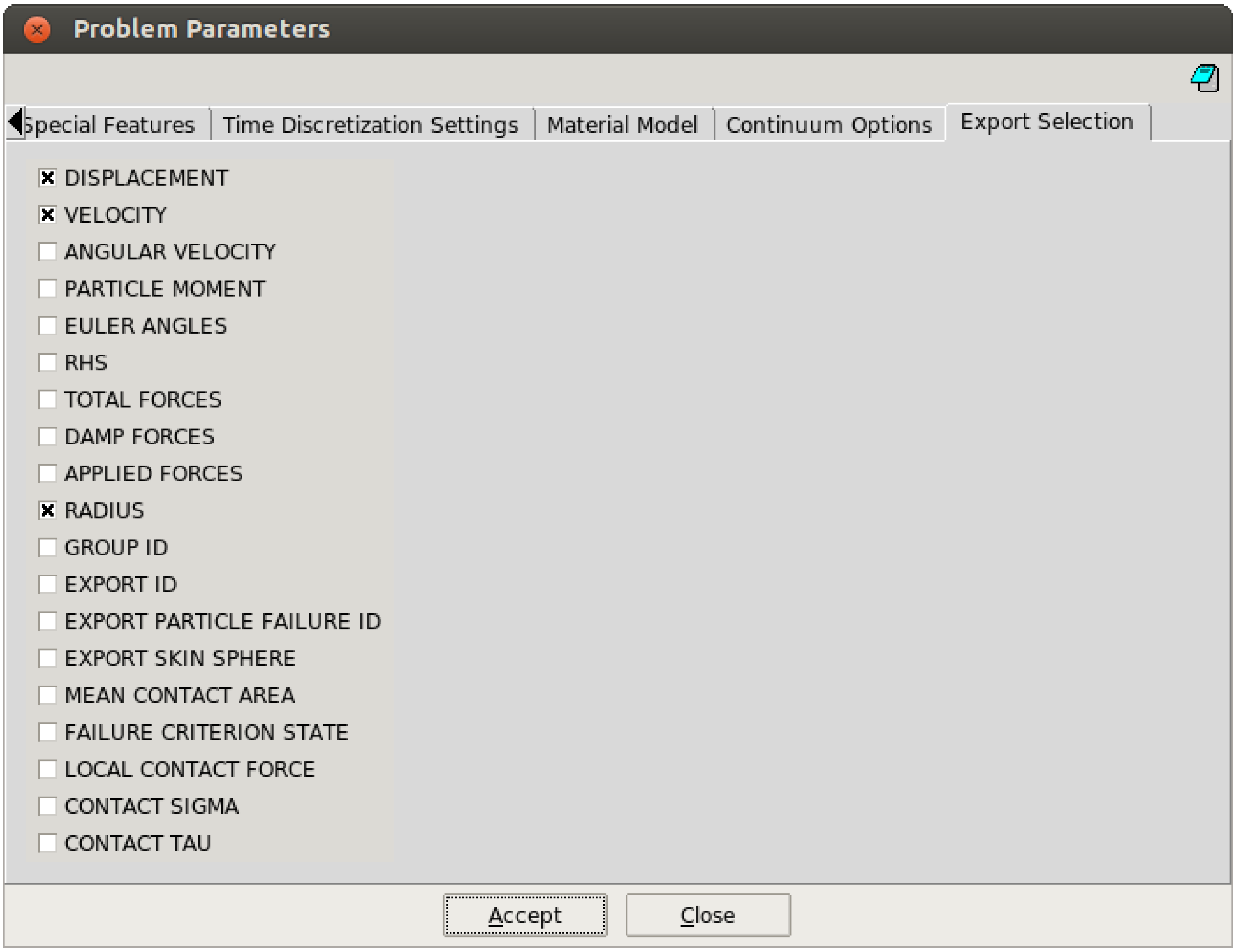
- Graphs Option. Activate this in concrete test simulation or general analysis for the strain-stress curves and poisson ratio evaluation.

- Triaxial Option. This allows introducing a uniform pressure over the *skin particles*. Also this option will disable the restrictions of the velocity degree of freedom while the confinement pressure is acting. Only the elements in the *Top-center* and *Bot-center* groups will keep the XZ velocity fixed to avoid inestabilities. The user can define a percentage of time while the pressure will be increased from zero to the maximum defined value. In the tests presented in Section 7 the triaxial tests are done with 1.5, 4.5, 9.0 30 and 60 Mpa. Also a percentage of the simulation time can be defined here as a rest time; during this time (usually 1%) the sample disipates the dynamic waves due to the confinement and prepares to be compressed by uniaxial vertical strain.

- Fixing horizontal velocity. This option sets the horizontal velocity of all the particles belonging to the *Top* and *Bottom* groups appart form the ones in the centered groups. This option is activated for the test simulations in Section 7 since in the report from the experimental data [29] it is detailed that there exist friction between the plates and the specimen.

Export Selection:

Depending on the options that have been activated in the previous tabs here there will be more or less variables that can be exported to the post process in the output time steps selection. All the variables available are shown in the next figure 6.28.


Problem Parameters

Special Features | Time Discretization Settings | Material Model | Continuum Options | Export Selection

☒ DISPLACEMENT
☒ VELOCITY
□ ANGULAR VELOCITY
□ PARTICLE MOMENT
□ EULER ANGLES
□ RHS
□ TOTAL FORCES
□ DAMP FORCES
□ APPLIED FORCES
☒ RADIUS
□ GROUP ID
□ EXPORT ID
□ EXPORT PARTICLE FAILURE ID
□ EXPORT SKIN SPHERE
□ MEAN CONTACT AREA
□ FAILURE CRITERION STATE
□ LOCAL CONTACT FORCE
□ CONTACT SIGMA
□ CONTACT TAU

Figure 6.28: Variables available to print in the post process.

Chapter 7

Concrete Test Simulation Examples

7.1 Numerical Simulation using KDEM

7.1.1 Description of the DEM parameters

Table 1 shows the basic material properties of the cement material, as required for the DEM analysis of the UCS and triaxial tests in the way they appear in the problemtype of the KDEM-Application.

Confining pressure p (psi)	DENSI (g/cc)	STAFRC	DYNFRC	YOUNG (GPa)	POISS	NSTR (MPa)	NCSTR1 (MPa)
	NCSTR2	YNGRT1	YNGRT2	TSTREN (MPa)	AXIFRAC	GAMMA	PYNGRT

Table 7.1: DEM parameters for UCS and triaxial tests

The meaning of the different DEM parameters in Table 1 is the following:

DENSI: Density (g/cc)

STAFRC: Static Coulomb friction parameter (μ_1)

DYNFRC: Dynamic Coulomb friction parameter (μ_2)

YOUNG: Young modulus (GPa)

POISS: Poisson ratio

NSTR: Normal tensile failure stress σ_f^t (MPa)

NCSTR1: First limit compressive stress at the interface (MPa)

NCSTR2: Second limit compressive stress at the interface

NCSTR3: Third limit compressive stress at the interface

YNGRT1: First reduction coefficient for initial Young modulus

YNGRT2: Second reduction coefficient for initial Young modulus

NCSTR3: Third limit compressive stress at the interface

TSTREN: Initial shear failure stress τ^f (MPa)

AXIFRAC: Extension of damage region ($u_n^f = u_n^l$ (1+ AXIFRAC), 4.12).

GAMMA: Effective area parameter γ_i (Eq.(4.7))

PYNGRT: Young modulus for elastic unloading path in the elasto-plastic branch.

7.1.2 Description of DEM analysis of UCS, Brazilian and triaxial tests

The simulation of a triaxial test with the DEM reproduces the experiment as follows.

a) The confining pressure is applied up to the desired hydrostatic testing pressure. The uniaxial normal compressive stress-axial strain curve follows the branch OA of Figure 7.1.

b) Apply a prescribed axial motion at the top of the specimen until this fail or axial strain strain reaches a desired amount of strain while confining pressure is held constant. The uniaxial normal compressive stress-axial strain follows the elastic branch AB of Figure 7.1 and then the elasto-plastic branch BC of the same figure.

For the uniaxial compressive strength (UCS) and the Brazilian Tensile Strength (BTS), the process starts by step (b) above described with *zero confinement pressure*. The constitutive curve relation the normal compressive stress and the axial strain follows the simple elasto-plastic law of Figure 7.2, where the meaning of parameters NCSTR1, YOUNG (E_0) and YNGRT1 as shown. A high value for NCSTR2 is chosen in this case.

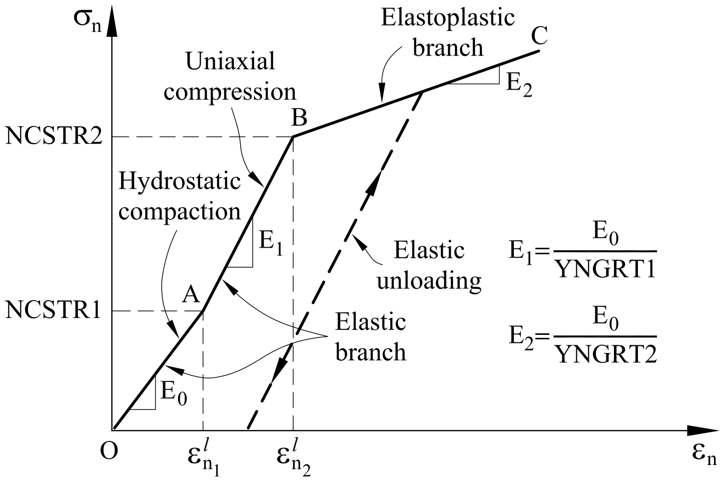


Figure 7.1: Axial strain-axial stress curve for analysis of triaxial tests

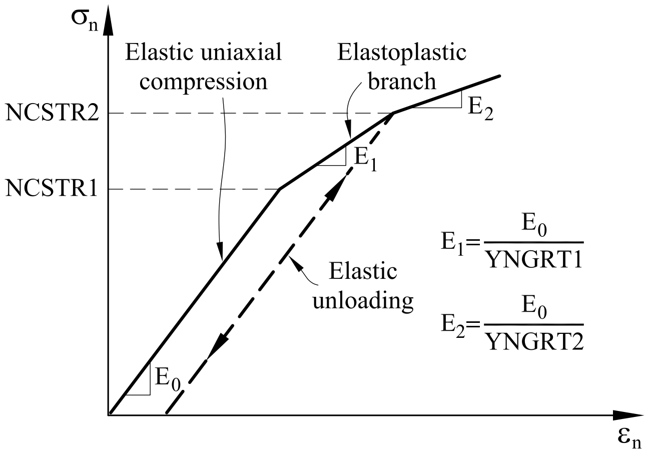


Figure 7.2: Axial stress-strain curve for UCS and BTS

7.2 Triaxial and Uniaxial Compressive Tests On Concrete Specimens

7.2.1 Definition of tests

The experimental tests were carried out at the laboratories of the Technical University of Catalonia (UPC). Details of the test are given in [29]. The concrete used in the experimental study was designed to have a characteristic compressive strength of 32.8 MPa at 28 days. Standard cylindrical specimens (of 150 mm diameter and 300 mm height) were cast in metal molds and demolded after 24 h for storage in a fog room. The triaxial tests were prepared as shown in Figure 7.3, with a 3-mm-thick butyl sleeve placed around the cylinder and an impermeable neoprene sleeve fitted over it. Before placing the sleeves, two pairs of strain gages were glued on the surface of the specimen at mid-height.

Steel loading platens were placed at the flat ends of the specimen and the sleeves were tightened over them with metal scraps to avoid the ingress of oil.

The tests were performed using a servo-hydraulic testing machine with a compressive load capacity of 4.5 MN and a pressure capacity of 140 MPa. The axial load from the testing machine is transmitted to the specimen by a piston that passes through the top of the cell.

Several levels of confining pressure were used in order to study the brittle-ductile transition of the response: 1.5, 4.5, 9.0, 30 and 60 MPa. First the prescribed hydrostatic pressure was applied in the cell, and then the axial load was increased at a constant displacement rate of 0.0006 mm/s.

Two specimens were tested at each confining pressure, and all tests were performed at ages of more than 50 days to minimize the effect of aging response. In addition to the triaxial tests, uniaxial compression tests were also performed. The experimental curves presented in the results are the mean curve of the two obtained for every test.

Confining pressure (psi)	DENSI (g/cc)	STAFRC	DYNFRC	YOUNG (GPa)	POISS	NSTR (MPa)	NCSTR1 (MPa)	NCSTR2 (MPa)
(UCS)	2.5	0.90	0.20	2.7	0.2	3.2	15	25

NCSTR3	YNGRT1	YNGRT2	YNGRT3	TSTREN (MPa)	AXIFRAC	GAMMA
30	3	7	15	15	0.10	0.80

Table 7.2: DEM parameters for UCS and triaxial tests in cylindrical concrete samples for confining pressures of 1.5, 4.5, 9.0, 30 and 60MPa

Another parameteres have to be aknowledged for the necessary damping to achieve a *quasi-static* simulation, global damping with a reduction factor of 0.2 afecting the global forces (see Section 3.3.2) and a local contact damping of 0.9 of the critical one (see Section 4.5.3).

The tests performed specially for this thesis are focused only in comparing the results obtained with the same parameters in the two codes, Dempack and the new one implemented in the context of this thesis, Kratos DEM-Application. It has to be said that the original tests reported in the report by Oñate et al. [20] with a simplified model have obtained results closer to the experimental ones by using different parameters (and a simplified model with only two reductions of the normal stiffness).

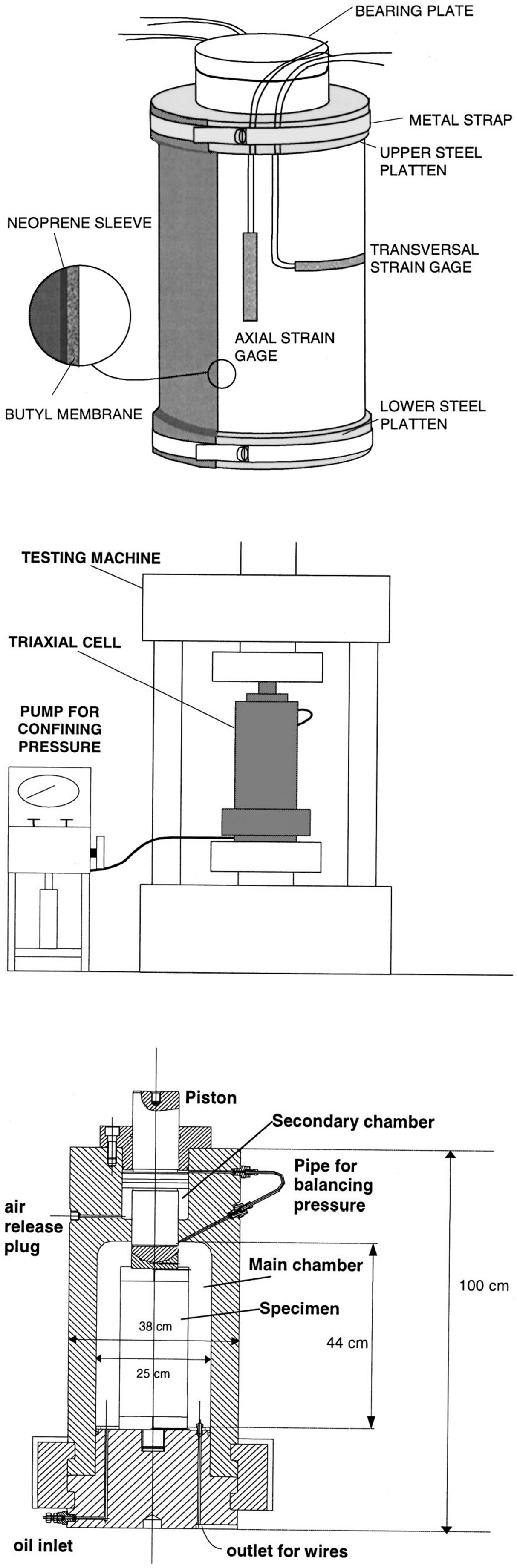


Figure 7.3: Specimen prepared for triaxial testing of concrete [29]

7.2.2 Description of the experimental tests

The general procedure for the triaxial compressive strength tests in the laboratory is as follows:

1. The specimen is placed between two endcaps and a heat-shrink jacket is placed over the specimen.

2. Axial strain and radial strain devices are mounted in the endcaps and on the lateral surface of the specimen, respectively.

3. The specimen assembly is placed into the pressure vessel and the pressure vessel is filled with hydraulic oil.

4. Confining pressure is increased to the desired hydrostatic testing pressure.

5. Specimen assembly is brought into the contact with a loading piston that allows application of axial load.

6. Increase axial load at a constant rate until the specimen fails or axial strain reaches a desired amount of strain while confining pressure is held constant.

7. Reduce axial stress to the initial hydrostatic condition after sample fails or reaches a desired axial strain.

8. Reduce confining pressure to zero and disassemble the sample.

7.2.3 DEM analysis strategy, definition of material parameters and results

The *confining pressure is directly applied to the spheres that lay on the surface of the specimen.* A normal force is applied to each surface particle in the radial direction and vertical direction respectivelly to the lateral particles and the ones on the top and bottom . The magnitude of the force is computed as $F_{n_i} = P\pi r_i^2$ where r_i is the particle radius and P is the confining pressure.

The constitutive equations for this study are based on *the local definition of the DEM* parameters as described on Chapter 4. Table 7.4 shows the DEM parameters for the UCS and triaxial tests for confining pressures of 1.5, 4.5, 9.0, 30 and 60 MPa.

The limits in the compressive normal local stress where the elastic-plastic curve changes its slope and the values of the reduction of the normal stiffness have been determined by adjusting the curves to the experimental ones in a phenomenological characterization procedure, technique reported as well by other authors in the field, for instance F.-V. Donzé [31].

The value of the shear failure stress τ^f and the internal friction angle have been estimated as $\tau^f = 15$ Mpa and $\phi = 42°(\mu_1 = 0.9)$ using the procedure described on Section 4.7.1.

The Coulomb friction coefficient has been estimated from numerical tests as $\mu_2 = 0.90$. This corresponds to DYNFRC in Table 7.4.

The limit tensile stress is deduced from the flexural test as $\tau^f = 4.5$ Mpa. This value corresponds to a value of $\tau^f = 3.2$ Mpa in the BTS test. This assumption has been validated in the numerical experiments.

The model used for the cases is a mesh of 13k spheres (1k=1000 spheres). Figure 7.4 shows the typical distribution of the spheres radius for a discretization, in this case the 13k mesh. Table 7.3 shows some relevant DEM parameters for the discretization used such as is the average radius of all the spheres(r_m), the coordination number, i.e. the average number of contacts per sphere, (N_c), and the porosity of the sphere assembly (p). The description of these parameters were presented in 5.1.2.

DEM Mesh	13k
No. spheres	13500
r_m	4.1140×10^{-3}
N_c	9,4495
p	0,2573

Table 7.3: UPC triaxial test on a concrete specimen. Parameters for the DEM mesh (1k = 1000 spheres)

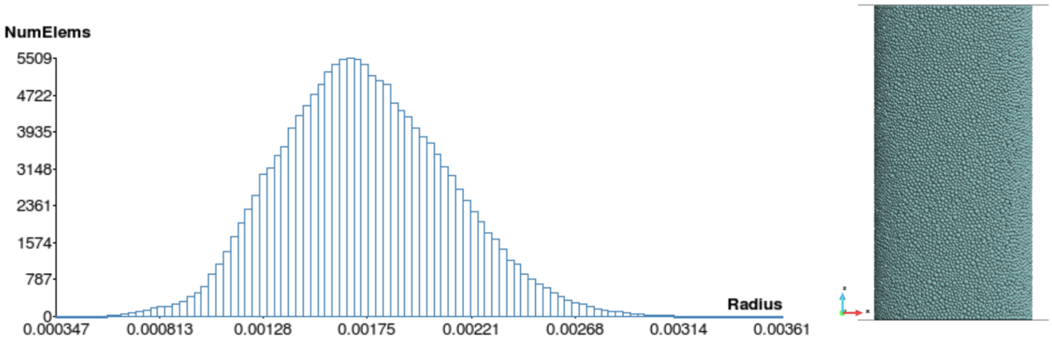


Figure 7.4: Distribution of the spheres radius for the 13k discrete element mesh

Figures 7.5–7.6 show results of the stress-strain curves for the triaxial tests for confining pressures of 1.5, 4.5, 9.0, 30 and 60MPa using a mesh of 4000 spheres. The curves capture with good accurary the peak point in stresses and the deformation point where this occurs. However they show an abrut loss of resistance after the peak that doesn't match the behaviour of the experimental curves for low confinement pressure. They show however a good post-peak resistance tendency.

Figure 7.7 shows the stress-strain curve obtained for the UCS test using the 13k mesh. For this case the experimental curve is not available; However, the experimental peak value (32.8 MPa) and the tangent elastic Young's modulus (2.7 MPa) have good agreement with the experiment values reported in [29].

It has to be pointed out again that the purpose of this results is to verify that the two codes yield similar results. In this sense there is absolute agreement between the two codes which have used the same parameters and conditions, including the time step of the calculation, the velocity of the loading plates, the same mesh, etc. It can be observed though, from the point where the plots start that KDEM yields a slightly overestimated value of the applyied pressure meanwhile Dempack does the contrary; that make the results evolve slightly differently. Having said that, the results are satisfactory and the main goal of the thesis is achieved.

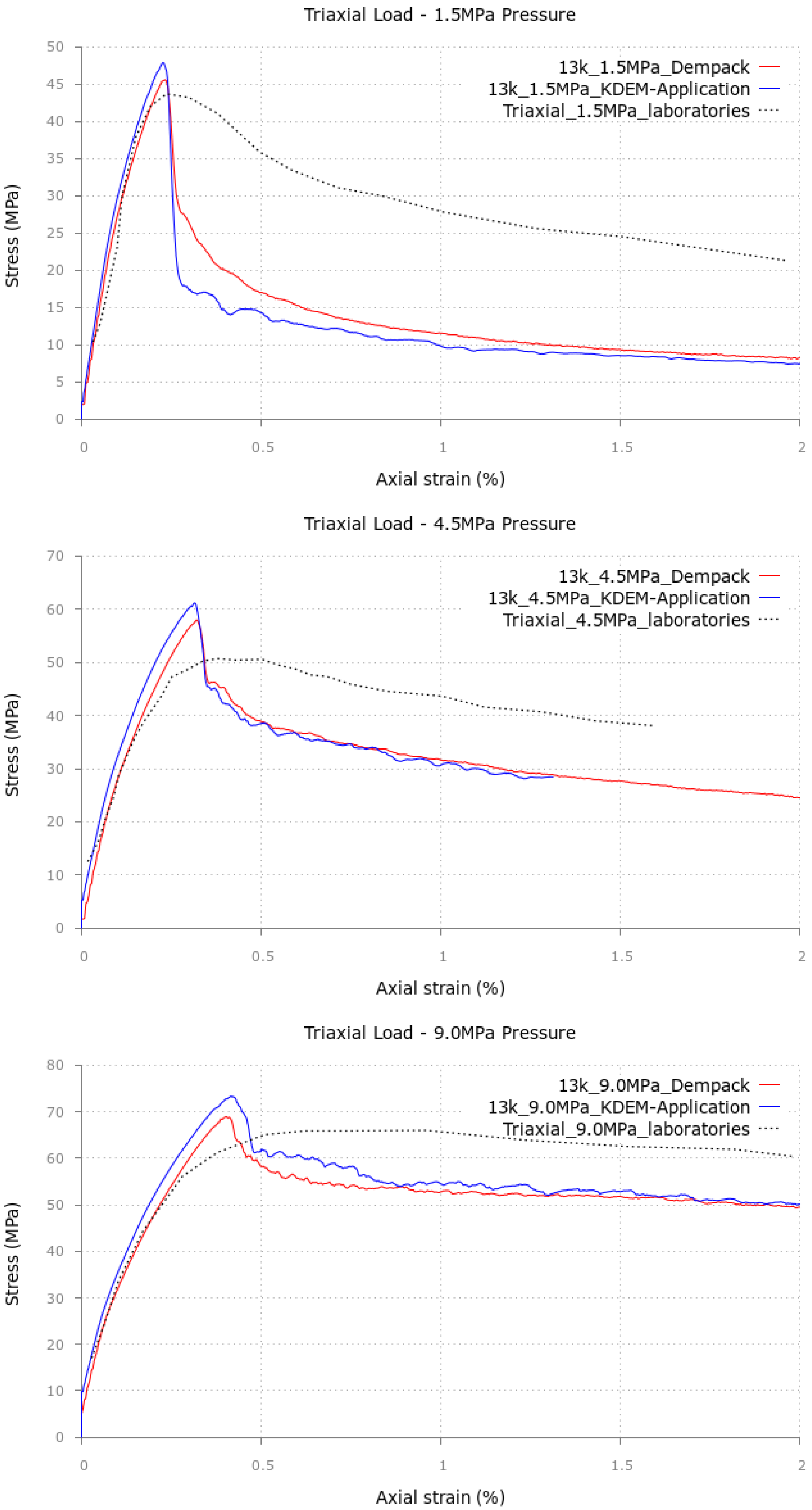


Figure 7.5: Triaxial test on concrete samples. DEM results for the 13k mesh with Dempack and KDEM and experimental values [29] for a confinement pressure of (a) 1.5MPa, (b) 4.0MPa, (c) 9.0MPa

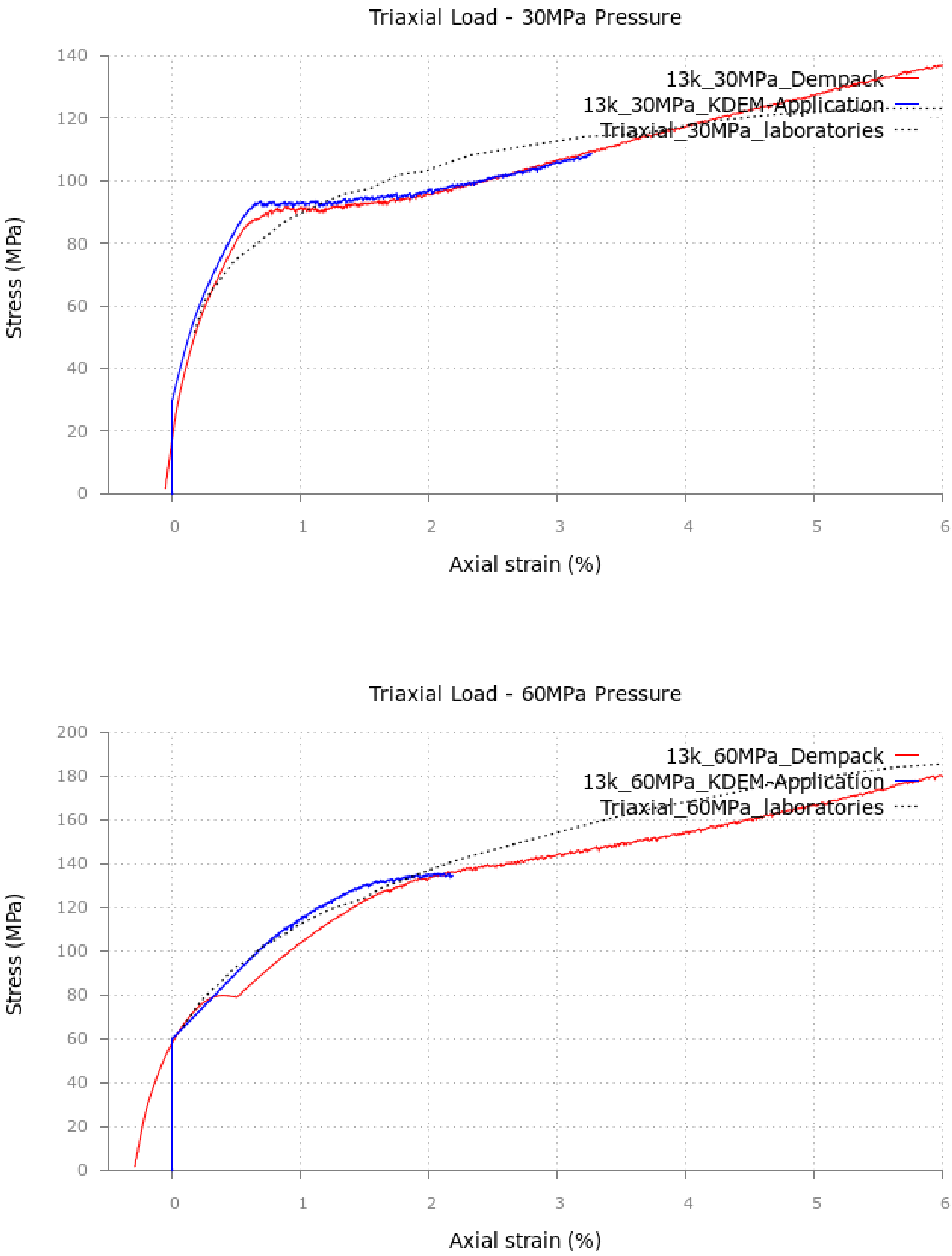


Figure 7.6: Triaxial test on concrete samples. DEM results for the 13k mesh with Dempack and KDEM and experimental values [29] for a confinement pressure of (d) 30MPa, (e) 60MPa

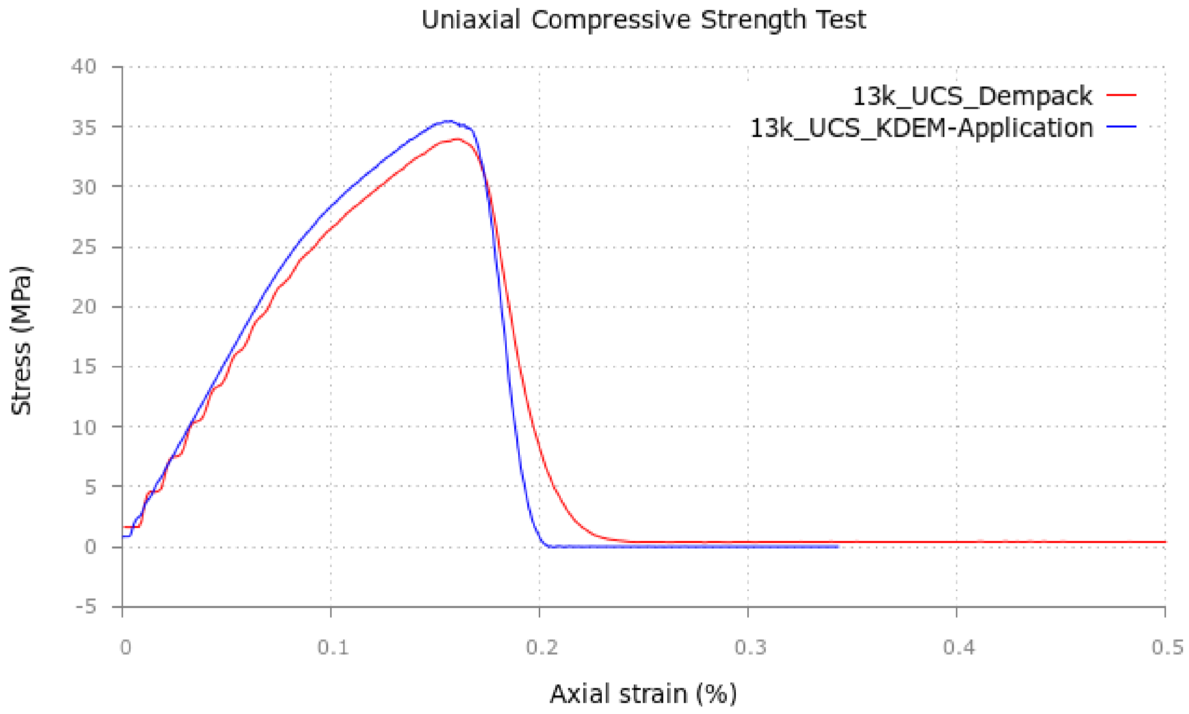


Figure 7.7: Uniaxial compressive strength (UCS) test on concrete sample. DEM results for the 13k mesh in KDEM and Dempack [29]

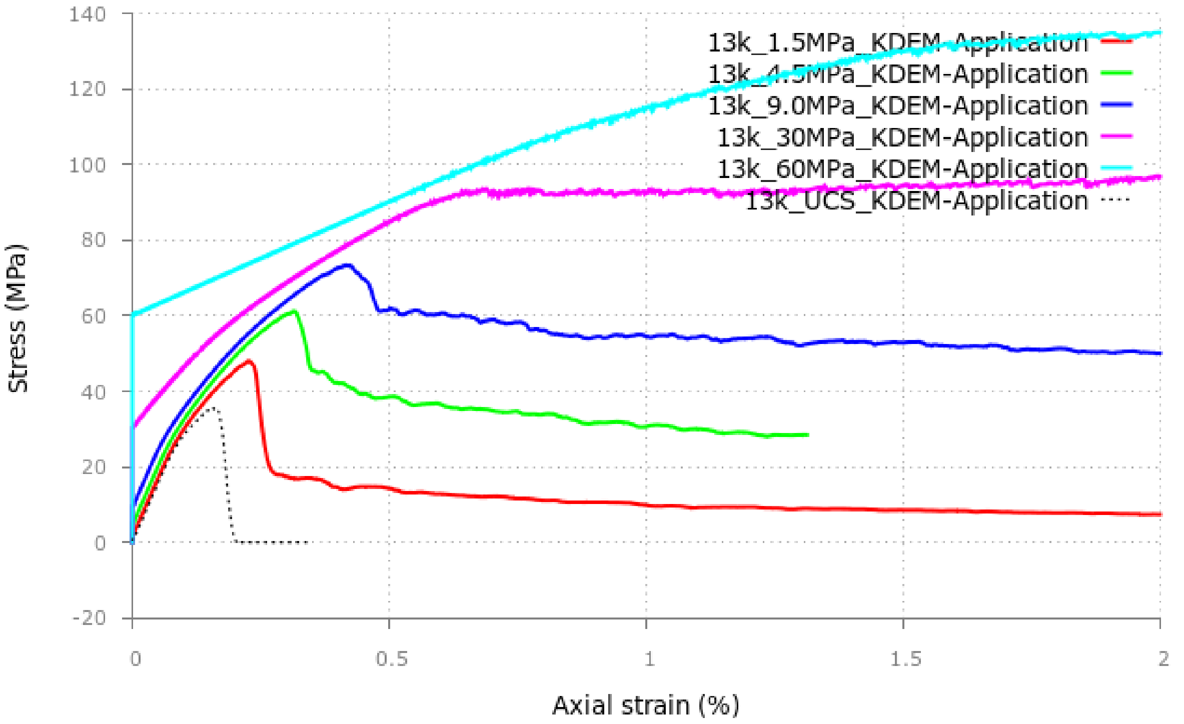


Figure 7.8: Compilation of the UCS and Triaxaial tests performed by KDEM-Application for the 13k mesh

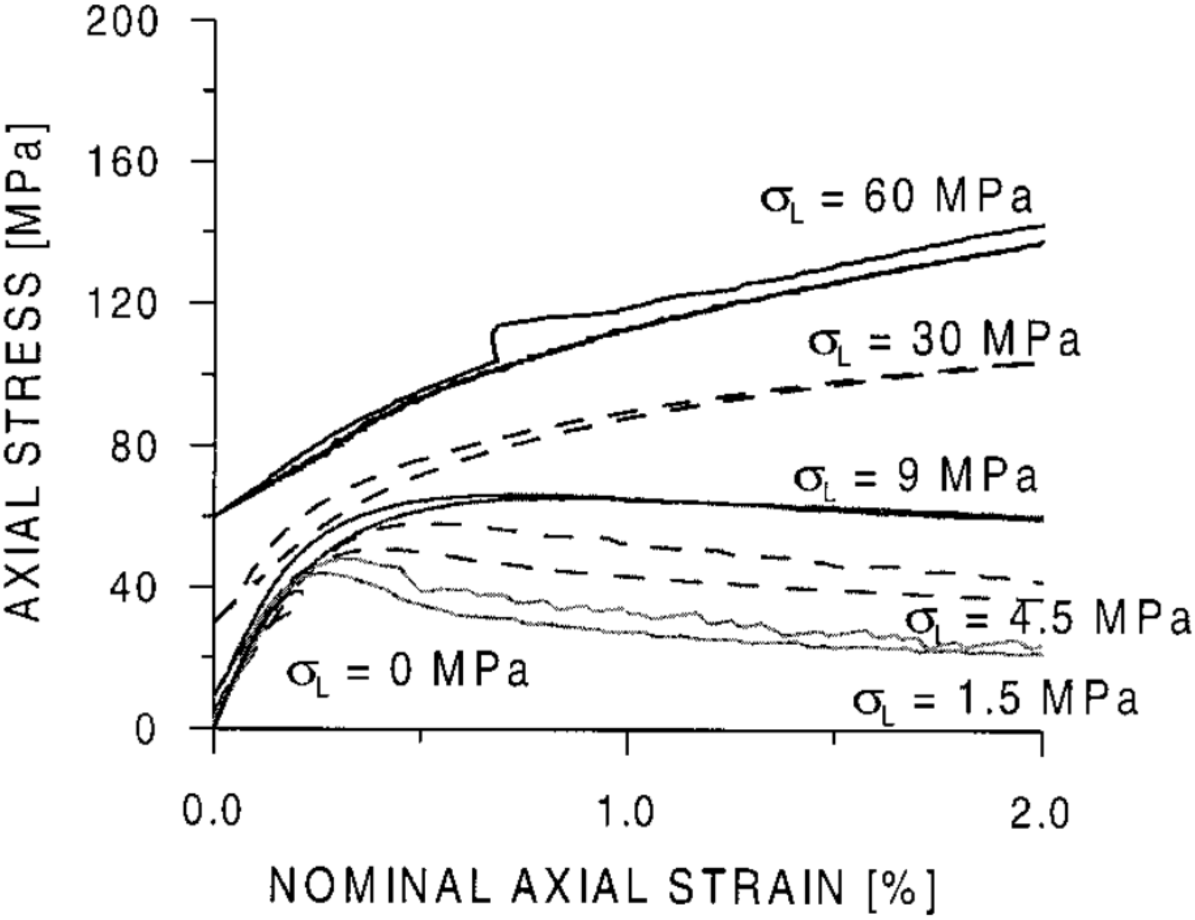


Figure 7.9: Compilation of the same cases in the experimental results published by D. Sfer, I. Carol et al [29]

7.3 Brazilian Tensile Strength Test

7.3.1 Definition of test

In the Brazilian (Indirect Tensile Strength) test, a disc of material is subjected to two opposing normal strip loads at the disc periphery (Fig. 7.11).

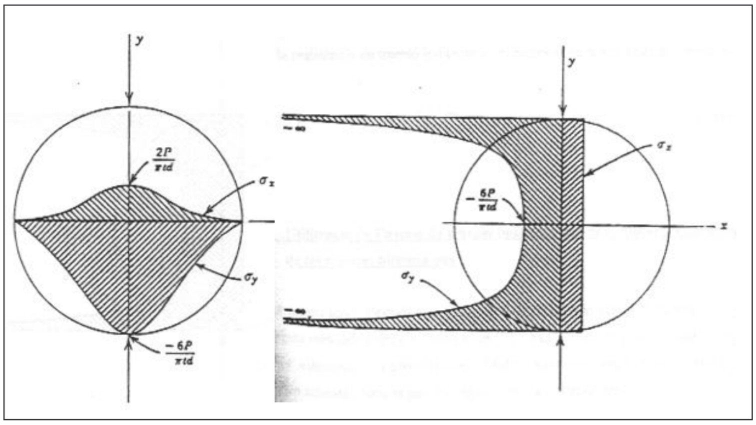


Figure 7.10: Theoretical stress distribution on diametral and vertical planes in the indirect tensile test.

The applied load is P. The rather thin disc has a radius R and thickness L. The tensile strength of concrete or rock, σ_t, is calculated from the equation (diameter $D = 2R$):

$$\sigma_t = \frac{2 \cdot P_{max}}{\pi \cdot t \cdot d} \tag{7.1}$$

Eq. 7.1 is based on the theory of elasticity for isotropic media. The formula gives the tensile stress perpendicular to the loaded diameter at the center of the disc at the time of failure when the applied force is P. Failure initiates at the center of the core and propagates outward along the loading direction.

7.3.2 DEM analysis strategy, definition of material parameters and results

In this case, in order to compare the results in the two codes, the numerical test simulates a cement speciment; its material properties and results were provided by Weatherford company [14].

Confining pressure (psi)	DENSI (g/cc)	STAFRC	DYNFRC	YOUNG (GPa)	POISS	NSTR (MPa)	NCSTR1 (MPa)	NCSTR2 (MPa)
(UCS)	1.7	0.30	0.40	2.85	0.1	2.90	6.7	-

NCSTR3	YNGRT1	YNGRT2	YNGRT3	TSTREN (MPa)	AXIFRAC	GAMMA
-	18	-	-	6.0	1.0	0.80

Table 7.4: DEM parameters for UCS and triaxial tests in cylindrical concrete samples for confining pressures of 1.5, 4.5, 9.0, 30 and 60MPa

The test is performed numerically in KDEM by imposing velocity on two strips of spheric particles upon the cylinder and below. The total imposed velocity of the strips is $0.25 m/s$ and obviuously damping has been used to get quasi-staticity.

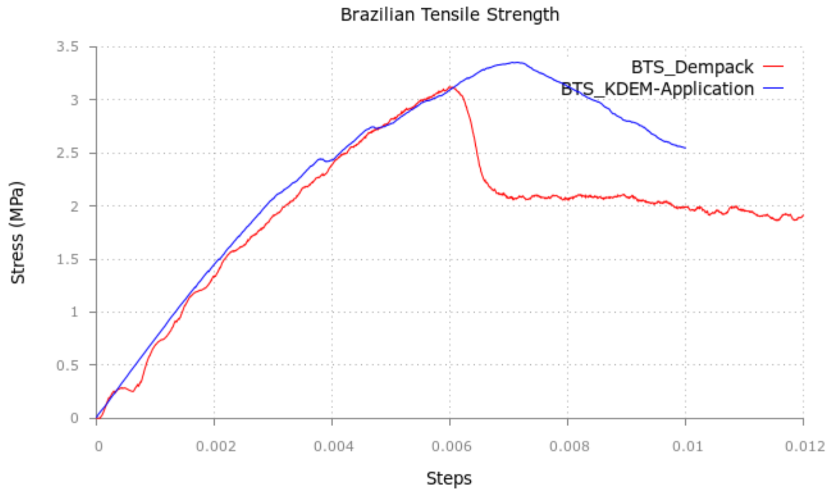


Figure 7.11: DEM results for BTS test in cylindrical cement sample.

The expected value is $\sigma_t = 2.9Mpa$ obtained from the experimental results [14]. It can be seen that Dempack obtains a good approximation of the value while KDEM overestimates the value. This is just one of the first results obtained in this kind of simulations and it has to be noted that the mesh used for this problem was not the same as the one used in Dempack. In general terms, it can be said that the results are good and also the post results show a good behaviour of the model.

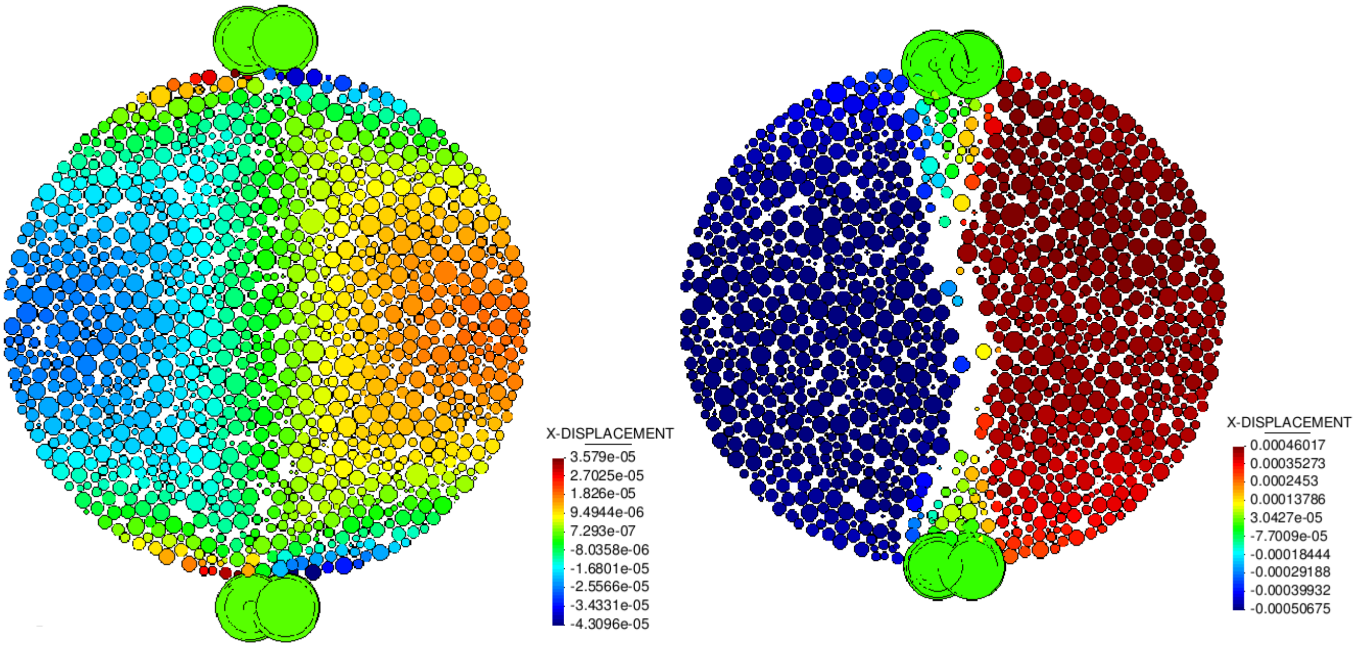


Figure 7.12: X Displacement of a centered section of the specimen at the begining of the loading and after failure (deformation 10 times enlarged).

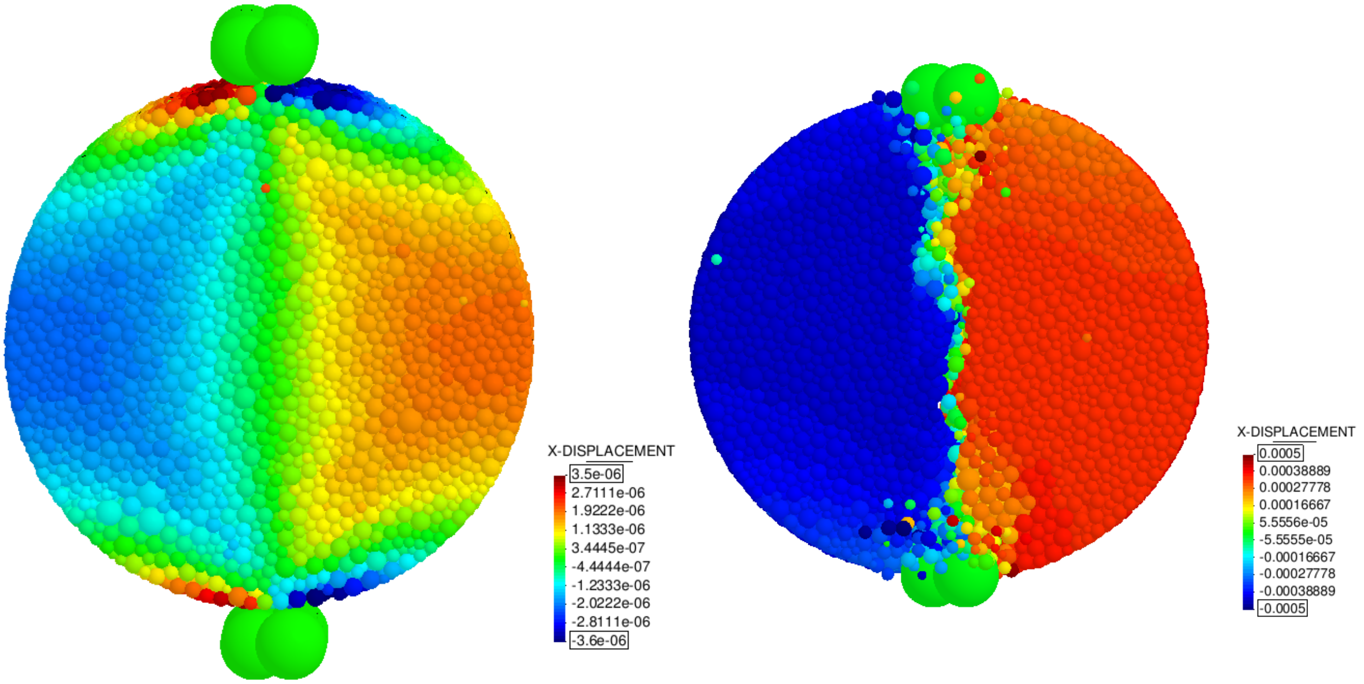


Figure 7.13: X Displacement of frontal view of the specimen at the begining of the loading and after failure (deformation 10 times enlarged).

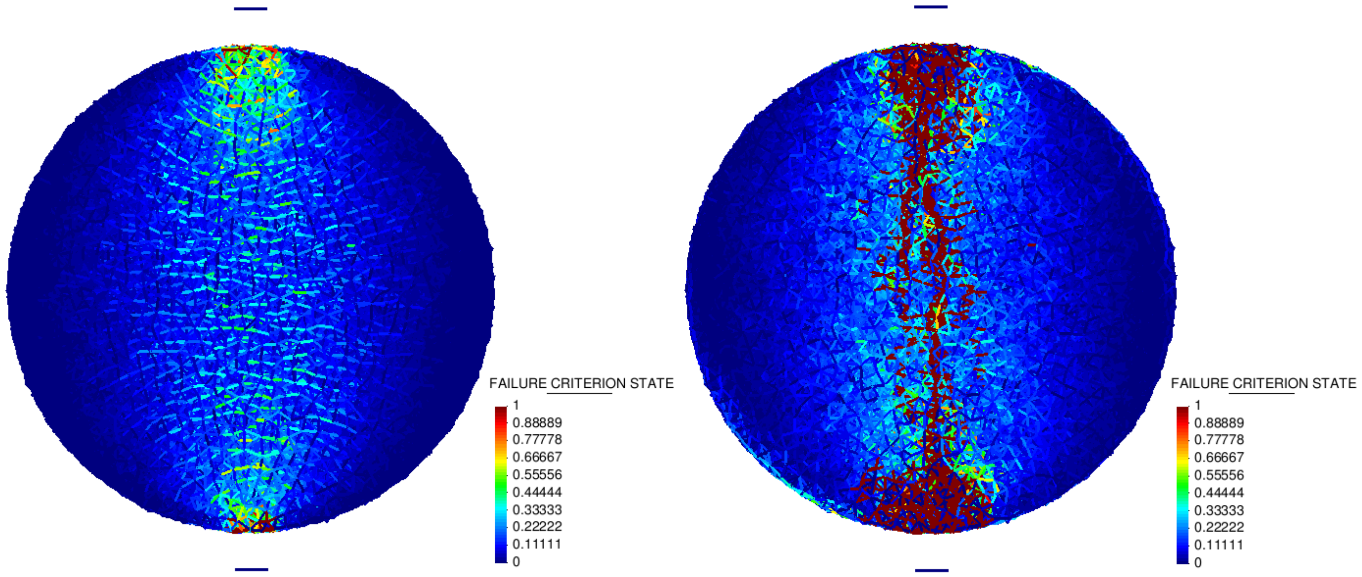


Figure 7.14: Failure Criterion State on the contacts at the begining of the simulation and after failure (real scale deformation).

It can be seen that the fracture is strongly marked on a vertical plane in the center of the speciment as in the experimental results. Also the distribution of the stresses seem to be correct since the tensional stress concentrates in the center diameteral plane where the horizontal contacts are the most loaded ones. The fracture evolves rapidly and localizing in a vertical path. It starts in both extremes and the resistence is over when one of the crack connects the whole diameter. The variable plotted on the contacts *FAILURE_CRITERION_STATE* (find the description in Sectionhydroo) is an output value that gives a proper insight of the contacts which are closer to failure, being 1 the value for the broken ones.

Chapter 8

Conclusions and future work

8.1 Conclusions

Theoretical research and numerical analysis

A brief introduction of the application of the Discrete Element Method to the continuum media simulation has been presented, specially the case of cohesive frictional solid materials such as concrete, cement or rock.

The *pros* and the *contras* of the method in its basic conception for the problem of continuum simulations have been reported and referenced from the state of the art and have been highlighted after with numerical analysis.

Several ideas and developments have been introduced in order to improve the DEM for the above mentioned problems. Some of them resulted successful; this is the case of the proposed correction of areas which makes the problem mesh independent in terms of local contact definition and ensures that there is no overlapping and the areas are correctly calculated.

The model developed in CIMNE by E. Oñate, F. Zárate and other researchers for cohesive frictional materials have been presented in detail.

Code development

A versatile Discrete Element Method application has been coded in the Kratos platform. The software brings the user the possibility to perform both discontinuous and continuous simulations.

A specific, user-friendly, problemtype that interacts with GiD has been developed; the problemtype in the pre-process brings the user tools for the meshing stage with spheric particles, allows introducing the material properties, is able to apply the necessary conditions, and defines the problem parameters and options for the calculation; on the other hand, the post-process allows visualization of a vast diversity of results for the analysis and automatically creates graphs in the case of concrete testing simulation.

The code has been parallelized succesfully in Open MP; the scalability charts have been presented and the results are fairly promising. The code is currently on its way to be parallelized in MPI technology as well.

The code is set in the Kratos framework and has been designed with special attention to be able to couple with other fluid or solid applications.

Numerical results

Several numerical anyalisis have been performed testing the code and using the formulation presented in this thesis.

Lack of reliability in the method has been shown even for linear elasticity problem. It is highly mesh dependent even for the linear elasticity, where the Poisson ratio is strongly related with the heterogeneity of the mesh. Although the results seem to converge increasing the number of elements, several more test have to be done to have a prediction of the order of the convergence for the different interesting variables such as the displacement, stresses, etc. It can be concluded that, by the formulation and the model that the community is using nowadays in DEM for the continuum, the DEM is not a discretization method but a phenomenollogical method that needs a calibration procedure.

The code has been verified by simulating the same test as the reference code Dempack. The results obtained with KDEM are very similar to the ones obtained by Dempack using the same parameters; these results at the same time simulate well the experimental curves presented.

Concrete Testing with the presented formulation

The main results obtained by CIMNE researchers E. Oñate, F. Zárate, F. Arrufat in the report [20] in the structural analysis of the concrete samples with the discrete element method (DEM) have been presented. The assumptions made for modelling the material with the DEM using a local constitutive model have been described in detail. The good behaviour of the model has been validated in the analysis with the DEM of concrete samples under a uniaxial compressive strength test, different triaxial compressive strength (UCS) tests, and a brazilian tensile strength (BTS) test for a cement speciment. DEM results compare well with experimental data for the same tests provided by Weatherford for cement samples [14] and UPC for concrete samples [29].

The methodology however is problem dependent and needs a calibration procedure of the constituitive parameters including the failure strenghts that apply to the uncoupled Mohr Coulomb criterion. Its of the author's opinion that a lot of research has to be and actually can be done in order to develop more sofisticated failure criteria by using the complete information that the stresses and strains give and not focusing only on the contact one-dimensional state. The same applies to the damage and plasticity laws which are now just basic models. Obviously, from the author's point of view, this has to be done keeping the simplicity and genuineness of the DEM. Other choices would make the method not be advantageous against other FEM-based methods.

8.2 Future of DEM Application

Fortunately the Discrete Element Method has a remarkable interest nowadays in CIMNE, as it has been presented in section 6.3, as well as in many research institutions. The Discrete Element Method brings a lot of posiblities in the world of simulation. In this sense, a lot of work will be done in the KDEM with the different presented applications as well as in the application which concerns this thesis, the continuum simulation.

In the continuum simulation field, the future work planned by the autor to be done is:

- Investigate in the *micro-macro* parameter determination. The linear elasticity is still obtained via calibration method and it is highly mesh dependent.
- Use information of the strain and stress state for developing more complete failure criteria.
- Develop a coupling scheme with the finite element method.

The last bullet of the list is a very important topic that would permit use the Discrete Element Method only when the fracture occurs and the Finite Element Method on the rest of the model; FEM is a much more reliable, accurate and fast method dealing whith elasticity, plasticity and most of the constitutive models used to simulate solids. The Discrete Element Method though give interesting results simulating the fracture of cohesive materials and so, the best features of both methods could be obtained by coupling the two methods. See previous work done by CIMNE researchers in [15, 21, 24, 25]

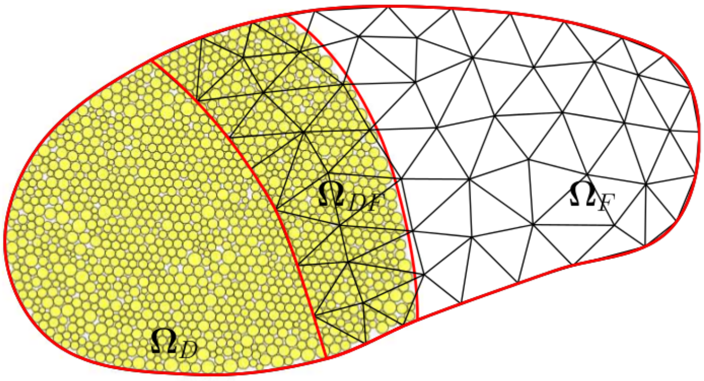


Figure 8.1: Overlap region between DEM and FEM. *Source: Carlos Labra [15]*

Another reason is the application to simulations which deal with large structures where the failure occurs in a very localized region due to an impact or an excavating process for instance. It would be necessary to have a very fine mesh in the area of interest where the DEM would simulate the mutlifracture; on the rest of the domain a coarse finite element mesh would suffice. This way avoid using an innecessary large number of particles that would make the calculation not-feasible in terms of memory and time calculation. The following image 8.3 illustrates this idea, a DEM domain evolves while the initial FEM Elements get close to the failure critera and the coupling scheme allows the transfer of nodal information and lets the DEM deal with the fracture.

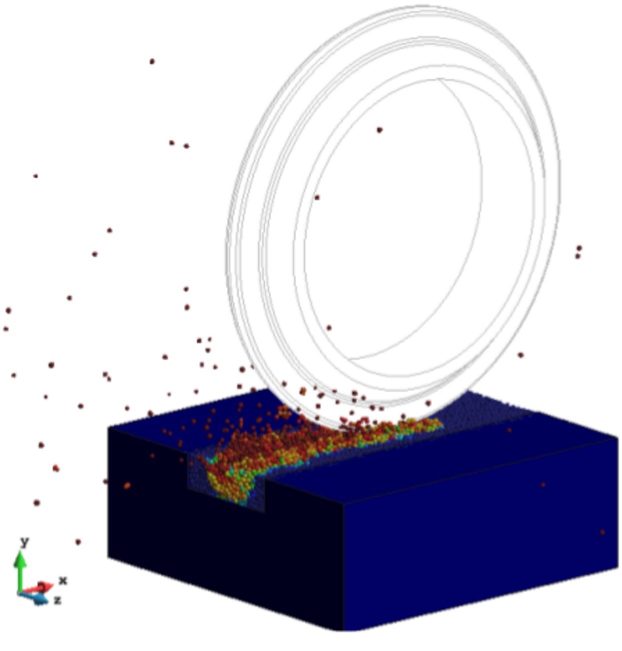


Figure 8.2: Disc cutting a rock simulated by a FEM-DEM coupled code. *Source: Carlos Labra [15]*

Currently CIMNE is working in engineering projects where simulations like the one presented in Figure 8.2 are used. The code used for these simulations is Dempack, where all these capabilites were already implemented and developed by Carlos Labra [15].

Regarding the DEM-Application, obviously it is a young code that still needs a lot of verification and ”tune up” in order to achieve robustness and efficiency. The team have a considerable list of new features and modifications to introduce to the application which is now just the foundation of a promising ambitious project. To name some examples:

- (generic) MPI paralellization of the code.
- (generic) Include the Discrete Element Method in the new problemtype of Kratos for GiD.
- (basic module) Implement new discrete particles, ellipsoids, tablet type, polyhedral shapes, clusters of spheres and DEM blocks concept.
- (continuum module) Stress and strain calculation.
- (continuum module) Implement a implicit algorithm for the quasi-static problems in order to speed the calculation up.
- (continuum module) Coupling the KDEM with a Computational Solid Mechanics code.
- (continuum module) FEM-DEM coupling, transition from FEM to DEM.
- (continuum module) Professional interface for engineering projects.
- (concrete testing simulations) Specific wizard for the concrete tests simulation.

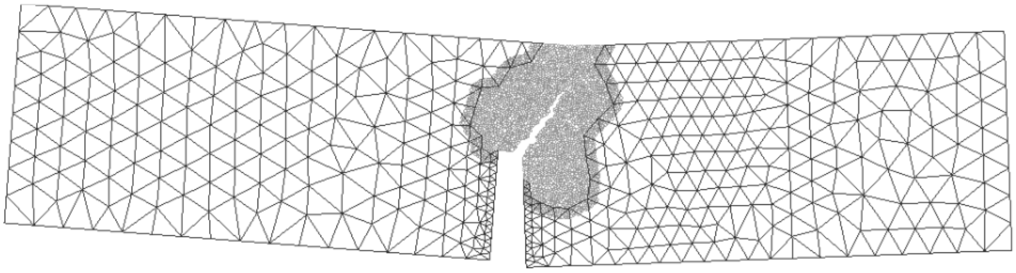


Figure 8.3: 2D nothced beam test where the FEM domain evolves to DEM before fracturing. *Source: Carlos Labra [15]*

Appendix A

Derivation of elastic spring stiffness

Normal stiffness

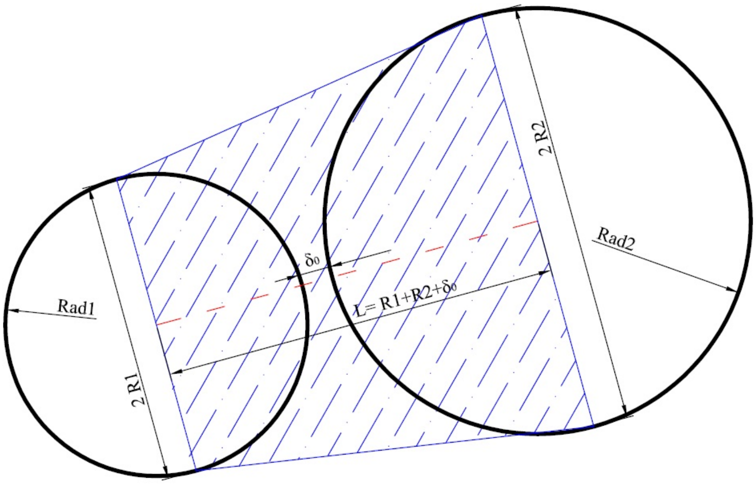


Figure A.1: Equivalent volume corresponding to by the contact. *Source: M.Santasusana [28].*

Imposing that the displacement in the normal direction is the integral of the deformation (considered not constant) of the equivalent truncated conical volume defined by the spehres in contact:

$$\delta_x = u_2 - u_1 = \int_0^L \varepsilon dx = \int_0^L \frac{F_x}{EA} dx = \frac{F_x}{E} \int_0^L \frac{dx}{A(x)} \tag{A.1}$$

With a linear variation of the radius:

$$R = R_1(1 + \beta x) \qquad where \qquad \beta = \frac{R_2 - R_1}{R_1 \cdot (R_1 + R_2 + \delta_0)} \tag{A.2}$$

Yields to:

$$\frac{F_x}{E} \int_0^L \frac{dx}{\pi(1+\beta x)^2 \cdot R_1^2} = \frac{F_x}{\pi E R_1^2 \beta} \int_0^{R_1+R_2+\delta_0} (1+\beta x)^{-2} \cdot dx = \frac{F_x}{\pi E R_1^2 \beta} \left[-\frac{1}{1+\beta x} \right]_0^{R_1+R_2+\delta_0} \tag{A.3}$$

Evaluating the limits and inserting β:

$$\frac{F_x}{\pi E R_1^2 \cdot \frac{R_2 - R_1}{R_1 \cdot (R_1+R_2+\delta_0)}} \left(1 - \frac{1}{1 + \frac{R_2 - R_1}{R_1}} \right) = \frac{F_x}{\pi E R_1 \cdot \frac{R_2 - R_1}{R_1+R_2+\delta_0}} \left(\frac{R_2 - R_1}{R_2} \right) \tag{A.4}$$

which finally yields:

$$\delta = \frac{F_x}{E\pi} \cdot \frac{R_1 + R_2 + \delta_0}{R_1 \cdot R_2} \tag{A.5}$$

And so,

$$F_x = K_n \cdot \delta_x \qquad K_n = \pi E \cdot \frac{R_1 \cdot R_2}{R_1 + R_2 + \delta_0} \tag{A.6}$$

Tangential stiffness

Proceeding similarly, for the shear stress, the integral of the deformation defines the tangential displacement:

$$\delta_y = V_2 - V_1 = \int_0^L \gamma dx = \int_0^L \frac{F_y}{GA} dx = \frac{F_y}{G} \int_0^L \frac{dx}{A(x)} \tag{A.7}$$

The result of the integral is the same, and then:

$$F_x = K_n \cdot \delta_x \qquad K_s = \pi G \cdot \frac{R_1 \cdot R_2}{R_1 + R_2 + \delta_0} \tag{A.8}$$

The relationship between the two stiffness parameters is finally:

$$K_s = K_n \cdot \frac{1}{2(1+\nu)} \qquad or \qquad \frac{K_n}{K_s} = \frac{E}{G} \tag{A.9}$$

Appendix B

Table of coefficients for the correction of area in 2D

The apothem can be used to find the area of any regular n-sided polygon of side length s according to the following formula, which states that the area A is equal to the apothem a multiplied by half the perimeter p.

$$A = \frac{P \cdot a}{2} \tag{B.1}$$

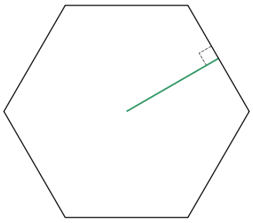


Figure B.1: Apothem of an Hexagon. *Source: Wikipedia.*

This formula can be derived by partitioning the n-sided polygon into n congruent isosceles triangles, and then noting that the apothem is the height of each triangle, and that the area of a triangle equals half the base times the height. An apothem of a regular polygon will always be a radius r of the inscribed circle.

This property can also be used to easily derive the formula for the area of a circle, because as the number of sides approaches infinity, the regular polygon's area approaches the area of the inscribed circle of radius $r = a$.

$$A = \frac{P \cdot a}{2} = \frac{2\pi \cdot r \cdot r}{2} = \pi \cdot r^2 \tag{B.2}$$

The area can be computed as well by trigonometry, knowing the inner angle of the triangles in which the polygon can be divided, i.e. $360/n$. Forming a rectangular triangle as follows:

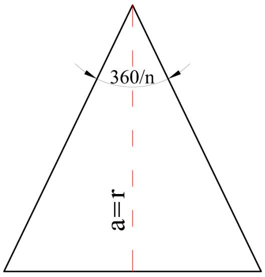


Figure B.2: Apothem in a n portion of the polygon.

The value of the total area of the polygon can be easily obtained as:

$$A = n \cdot r^2 \cdot \tan \frac{180}{n} \tag{B.3}$$

If now the perimeter is isolated from the formula B.1 and the last expresion for the area B.3 is introduced, the perimeter is:

$$p = 2nr \tan \frac{180}{n} \tag{B.4}$$

And so the ratio over the circle perimeter:

$$\alpha_{2D} = \frac{n \tan \frac{180}{n}}{\pi} \tag{B.5}$$

Poligon	Triangle	Square	Pentagon	Hexagon	Heptagon	Octagon	Nonagon
Num of neigh	3	4	5	6	7	8	9
Ratio of Areas	1.654	1.273	1.156	1.103	1.073	1.055	1.043

Poligon	Decagon	Hendecagon	Dodecagon	Tridecagon	Tetradecagon
Num of neigh	10	11	12	13	14
Ratio of Areas	1.034	1.028	1.023	1.020	1.017

Table B.1: 2D Polgon ratios

Appendix C

Table of coefficients for the correction of area in 3D

The same than in Annex B is done here for the 3D case, for polyhedra instead of polygons. In this case, it exists a regular polyhedra for a few cases, the so-called *Platonic Solid*.

The Surface Area presented in the next table correspond to the value of the surface covered by the faces of the polyhedra based on the radius of the inscribed sphere. *Source: Wolfram Alpha.*

Polyhedron	Tetrahedron	Hexahedron	Octahedron	Dodecahedron	Icosahedron
Num of Neigh	4	6	8	12	20
Surface Area	$24{r_i}^2\sqrt{3}$	$24{r_i}^2$	$12{r_i}^2\sqrt{3}$	$\frac{600{r_i}^2}{25+11\sqrt{5}}\sqrt{\frac{5+2\sqrt{5}}{5}}$	$\frac{120{r_i}^2\sqrt{3}}{7+3\sqrt{5}}$
Ratio of Areas	$3,308$	$1,910$	$1,654$	$1,325$	$1,207$

Table C.1: 3D Polyhedron ratios

The Ratio of Areas, γ_{3D}, is the ratio between the area of the regular polyhedron (platonic or interpolated) and the surface of the corresponding inscribed sphere.

Since the value is known only for the 5 regular polyhedra, a linear interpolation has been performed for the rest:

Number of neigh (n)	4	5	6	7	8	9	10	11	12
Ratio of Areas	$3,308$	$2,609$	$1,910$	$1,7827$	$1,654$	$1,572$	$1,490$	$1,410$	$1,325$

Number of neigh (n)	13	14	15	16	17	18	19	20
Ratio of Areas	$1,310$	$1,295$	$1,281$	$1,266$	$1,251$	$1,236$	$1,221$	$1,207$

Table C.2: Interpolations of ratios for different number of neighbours

Appendix D

Derivation of the Stress Tensor

Consider a assembly of discrete elements (spheres) occupying a volume V. The assembly is submitted to external forces T_i^1, T_i^2, ..., T_i^m on its boundary points x_i^1, x_i^2, ..., x_i^m. The average stress of an equivalent continuum of the same V volume under the same loads is:

$$\bar{\sigma}_{ij} = \frac{1}{V} \sum_{k=1}^{m} x_i^k T_j^k \tag{D.1}$$

This expression can be dedduced from the average stress in a continuum proceeding in the following way (Katalin Bagi [1]):

Consider a closed continuous domain with volume V loaded on its boundary S by a distributed force $p_i(x_j)$. Depending on the loads a $\sigma_{ij} = \sigma_{ij}(x_k)$ stress tensor belongs to every point of the domain satisfying the boundary conditions:

$$\sigma_{ij} n_j = p_i \tag{D.2}$$

where n_i is the outwards unit normal vector on S. The volume average of the stress tensor can be expressed - with the help of the Gauss-Ostrogradski teorem - as a surface integral:

$$\bar{\sigma}_{ij} = \frac{1}{V} \iiint\limits_{(V)} \sigma_{ij} \, \mathrm{d}V = \frac{1}{V} \iint\limits_{(S)} x_i p_j \, \mathrm{d}S \tag{D.3}$$

If the domain is divided into subdomains, the average stress tensor can be calculated separately for each subdomain:

$$\bar{\sigma}_{ij}^L = \frac{1}{V^L} \iint\limits_{(S^L)} x_i p_j \, \mathrm{d}S \tag{D.4}$$

where V^L and S^L are the volume and boundary of the L-th subdomain, distributed forces $p_i(x_j)$act on S^L form the neighbouring subdomains and the external boundary. To get a global average, voume-weighted averages of $\bar{\sigma}_{ij}^L$ can be calculated and it results in the same expression as D.3:

$$\bar{\sigma}_{ij} = \frac{1}{V} \sum_L V^L \bar{\sigma}_{ij}^L = \frac{1}{V} \sum_L \left(\iint\limits_{(S^L)} x_i p_j \, \mathrm{d}S \right) = \frac{1}{V} \iint\limits_{(S)} x_i p_j \, \mathrm{d}S \tag{D.5}$$

In those cases when there are concentrated forces instead of the distributed loads acting on the boundary of the domain and between the subdomains, the above expression can be written in a discrete form. Denote the forces acting from outside as F_i^1, F_i^2, ..., F_i^k, ...; they act at boundary points x_i^1, x_i^2, ..., x_i^k, ... Expression D.3 is modified as:

$$\bar{\sigma}_{ij} = \frac{1}{V} \sum_{k=1}^{m} x_i^k T_j^k \tag{D.6}$$

(the index k runs over the external loading forces). Now consider the L-th subdomain; the forces F_i^1, F_i^2, ..., F_i^c, ... act on its boundary at the points x_i^1, x_i^2, ..., x_i^k, ...; (partly from the neighbouring subdomains and partly form outside). So the average stress here is:

$$\bar{\sigma}_{ij} = \frac{1}{V} \sum_{c=1} x_i^c F_j^c \tag{D.7}$$

Since the forces insede cancel out in the summation, the volume-weighted average for the whole domain - as already seen in D.6

$$\bar{\sigma}_{ij} = \frac{1}{V} \sum_{(L)} V^L \bar{\sigma}_{ij}^L = \frac{1}{V} \sum_{(L)} \left(\sum_{(c)} x_i^c F_j^c \right) = \frac{1}{V} \sum_{k} x_i^k F_j^k \tag{D.8}$$

Bibliography

[1] K. Bagi. Stress and strain in granular assemblies.*Mechanics of Materials*, 22:165-177, 1996.

[2] P.A. Cundall. A computer model for simulating progressive, large-scale movements in blocky rock systems. *Synopsium Soc. Internat Mécanique des Roches*, 2:8, 1971.

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

[4] I. De Pouplana Sardà. *Validation study of a new implementation of the Discrete Element Method for an open source multiphysics code.* Undergraduate Final Thesis. ETSECCPB - UPC, july 2013.

[5] F. Donze, F. Richefeu and S. Magnier. Advances in discrete element method applied to soil, rock and concrete mechanics. Electronic Journal of Geotechology Engineering 08, 1 44, Special volume, Bouquet, 2009.

[6] A. Fakhimi and T. Villegas. Application of dimensional analysis in calibration of a discrete element model for rock deformation and fracture. *Rock Mechanics and Rock Engineering*, 40(2):193–211, 2007.

[7] Y. Feng, Discrete Element Methods Theory & Practice. International Symposium / UK-China Summer School on Discrete Element Methods and Numerical Modelling of Discontinuum Mechanics, 2008.

[8] Y.T. Feng, K. Han and D.R.J. Owen. On upscaling of discrete element models: similarity principles. *Engineering Computations*, 26:6, 2009.

[9] S. Hentz, L. Daudeville and F. Donzé. Identification and validation of a discrete element model for concrete. J. of Engineering Mechanics, 130(6):709–719, 2004.

[10] Y.-M. Hsieh, H.-H. Li, T.-H. Huang and F.-S. Jeng. Interpretations on how the macroscopic mechanical behavior of sandstone affected by microscopic properties revealed by bonded-particle model. *Engineering Geology*, 1 10, 2008.

[11] G. Hu, Z. Hu, B. Jian and others. On the Determination of the Damping Coefficient of Nonlinar Spring-dashpot System to model Hertz Contact for Simulation by Discrete Element Method. *Journal of Computers*, 6(5):984–988, 2011.

[12] H. Huang. *Discrete element modeling of tool-rock interaction.* Ph.D. Thesis, December, University of Minnesota, 1999.

[13] N. Kruyt, and L. Rothenburg. Kinematic and static assumptions for homogenization in micromechanics of granular materials. *Mechanics of Materials*, 36(12):1157–1173, 2004.

[14] O. Kwon. Rock mechanics testing & analyses. Cement mechanical testing. *Weatherford Laboratories Report*, WFT Labs RH-45733, March 2010.

[15] C. Labra. *Advances in the development of the discrete element method for excavation processes.* Ph.D. Thesis. Technical University of Catalonia, UPC, July 2012.

[16] C. Labra and E. Oñate. High-density sphere packing for discrete element method simulations. *Communications in Numerical Methods in Engineering*, 25(7):837–849, 2009.

[17] C. Labra, J. Rojek, E. Oñate and F. Zárate. Advances in discrete element modelling of underground excavations. *Acta Geotechnica*, 3(4):317–322, 2009.

[18] H. Langhaar. *Dimensional Analysis and Theory of Models.* Wiley, 1951.

[19] J. Lubliner, S. Oller, J. Oliver and E. Oñate E. A plastic damage model for concrete. *Int. Journal of Solids and Structures*, 25(3):299–326, 1989.

[20] E. Oñate, F. Zárate, F. Arrufat, J. Miquel and P.A. Ubach. *Modelling and analysis of Cement and Concrete Samples under Mechanical Testing with Discrete Element Method.* Internal report of CIMNE, February 2013.

[21] E. Oñate and J. Rojek. Combination of discrete element and finite element methods for dynamic analysis of geomechanics problems. *Comput. Meth. Appl. Mech. Engrg.*, 193:3087–3128, 2004.

[22] C. O'Sullivan and J. D. Bray. Selecting a suitable time step for discrete element simulations that use the central difference time integration scheme. *Engineering Computations*, 21(2/3/4):278–303, 2004.

[23] D. Potyondy and P. Cundall. A bonded-particle model for rock. *International Journal of Rock Mechanics and Mining Sciences*, 41(8):1329 1364, Rock Mechanics Results from the Underground Research Laboratory, Canada, 2004.

[24] J. Rojek and E. Oñate. Unified DEM/FEM approach to geomechanics problems. In: Proceedings of Computational Mechanics WCCM VI in conjunction with APCOM04, Beijing, China, September 5 10, 2004.

[25] J. Rojek and E. Oñate. Multiscale analysis using a coupled discrete/finite element model. *Interaction and Multiscale Mechanics: An International Journal (IMMIJ)*, 1(1):1–31, 2007.

[26] J. Rojek, E. Oñate and F. Zarate, and J. Miquel. Modelling of rock, soil and granular materials using spherical elements. In *2nd European Conference on Computational Mechanics ECCM-2001*, Cracow, 26-29 June, 2001.

[27] J. Rojek, C. Labra, O. Su and E. Oñate. Comparative study of different micromechanical parameters. *Int. J. of Solids and Structures*, 49:1497–1517, 2012.

[28] M. Santasusana Isach. *Continuum modelling using the Discrete ELement Method. Theory and implementation in an object-oriented software platform.* Undergraduate Final Thesis. ETSECCPB - UPC, july 2012.

[29] D. Sfer, I. Carol, R. Gettu, G. Etse. Study of the behaviour of concrete under triaxial compression. *J. of Engineering Mechanics*, 128(2):156-163, 2002.

[30] F. A. Tavarez and M. E. Plesha. Discrete element method for modelling solid and particulate materials. *International Journal for Numerical Methods in Engineering*, 70:379–404, 2007.

[31] V.T. Tran, F.-V. Donzé and P. Marin. A discrete element model of concrete under high triaxial loading. *Cement & Concrete Composites*, 33:936–948, 2011.

[32] H.-C. Wang, W.-H. Zhao, D.-S. Sun and B.-B. Guo. Mohr-Coulomb yiedl criterion in rock plastic mechanics. Chinese Journal of Geophysics, 55(6):733–741.

[33] B. Yang, Y. Jiao, and S. Lei. A study on the effects of microparameters on macroproperties for specimens created by bonded particles. *Engineering Computations: International Journal for Computer-Aided Engineering and Software*, 23(6):607–631, 2006.

[34] O.C. Zienkiewicz, R.L. Taylor. *The Finite Element Method for Solid and Structural Mechanics.* Sixth Edition, Elsevier, 2005.